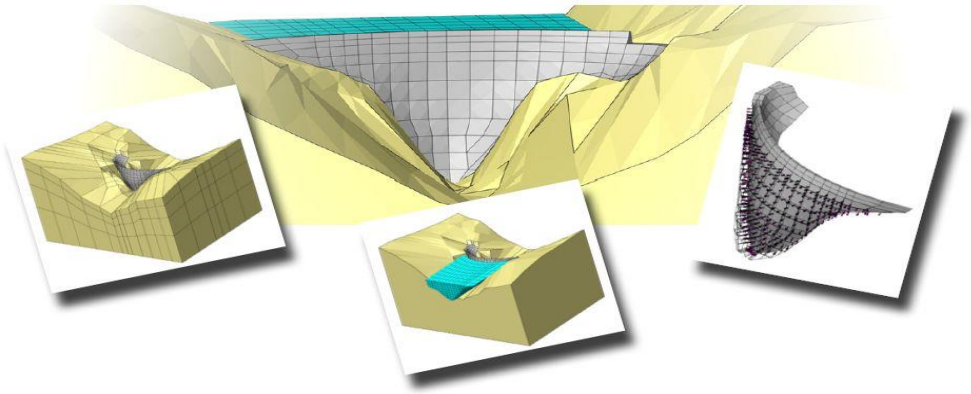
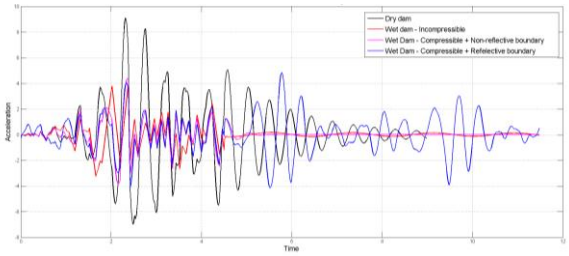
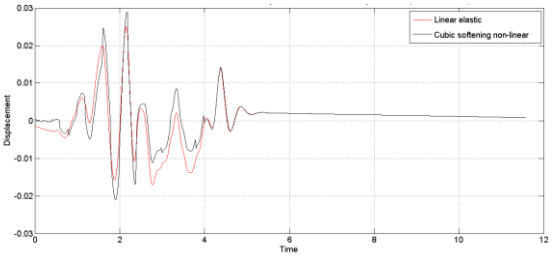
Numerical modeling of dam-reservoir interaction seismic response using the Hybrid Frequency–Time Domain (HFTD) method







Graduation Committee:
Prof. Dr. A.V. Metrikine
Dr. Ir. M.A.N. Hendriks
Dr. Ir. E.M. Lourens
Ir. W.P. Kikstra

Author : Luis Fernando Sirumbal Zapata
Student ID : 4179471

Date : June 2013

TU Delft
Technische Universiteit Delft

TNO DIANA
A TNO Company

Numerical modeling of dam-reservoir interaction seismic response using the Hybrid Frequency–Time Domain (HFTD) method

Master of Science Thesis

for the acquisition of the degree
Master of Science in Civil Engineering
at Delft University of Technology

by
Luis Fernando Sirumbal Zapata

June 2013

Graduation Committee:

Prof. Dr. A.V. Metrikine (chairman)
Section of Structural Mechanics, Faculty of Civil Engineering and Geosciences, Delft University of Technology

Dr. Ir. M.A.N. Hendriks
Section of Structural Mechanics, Faculty of Civil Engineering and Geosciences, Delft University of Technology

Dr. Ir. E.M. Lourens
Section of Offshore Wind, Faculty of Civil Engineering and Geosciences, Delft University of Technology

Ir. W.P. Kikstra
Senior Software Development Engineer, TNO DIANA

PREFACE

This research work constitutes the graduation project of my studies in the Master of Science in Civil Engineering program at Delft University of Technology, Faculty of Civil Engineering and Geosciences. During the last two years in TU Delft, I followed the structural mechanics specialization of the structural engineering track.

This research project about the Hybrid Frequency-Time Domain (HFTD) method for non-linear structural dynamics was sponsored and developed in cooperation with TNO DIANA Finite Element Software Company for Civil Engineering.

I am deeply grateful to Wijtze Pieter Kikstra, Senior Software Development Engineer at TNO DIANA, because of his unconditional commitment with this project. He is my mentor and friend, and without his participation the results obtained in the project would not have been as good as they are.

I would like to thank Dr. Gerd-Jan Schreppers, CEO at TNO DIANA, for giving me the opportunity of work in this project. Special thanks to: Jonna Manie, Software Development Engineer, which collaboration was crucial in the most important stages of the project; and Maziar Partovi, Advanced Application Engineer, who was always friendly and eager to talk about life. In general, many thanks to all the TNO DIANA people with whom I worked and spent enjoyable time during the last eight months.

From TU Delft I would like to thank my supervisor, Prof. Dr. Andrei Metrikine of the Section of Structural Mechanics, for his valuable guidance during the graduation committee and personal meetings we had during the last months. I would also like to acknowledge the important participation of the graduation committee members, Dr. Max Hendriks and Dr. Eliz-Marie Lourens. And finally I want to express my gratitude to Prof. Lambert Houben who guided me through all the graduation process.

I would like to dedicate this work to my home country university, National University of Engineering (UNI), in Lima, Peru. I am particularly grateful to the people of some institutions, like the Civil Engineering Faculty Research Institute (IIFIC-UNI) and the National University of Engineering Foundation (Pro-UNI), who believed in me and supported my studies at TU Delft.

Finally, I proudly dedicate my work to my family, parents and brother, and beloved persons in Peru. Without their love and constant support it would not be possible for me to be here writing these lines.

L. Fernando Sirumbal Z.

Delft, June 2013

ABSTRACT

This research work studies the fundamentals of the Hybrid Frequency-Time Domain (HFTD) method for dynamic analysis of structures, applied to the seismic analysis of dam-reservoir interaction systems.

The theoretical consistency and accuracy of the HFTD method, based on the frequency domain Fourier analysis, is demonstrated through all the parts of the thesis, but particularly in Chapter 5 for linear and highly non-linear single degree of freedom (SDOF) models subject to several types of loading, and in Chapter 7 for a 2D dam-reservoir interaction finite element model subject to an earthquake ground acceleration loading.

On the other hand, a Finite Element Method (FEM) formulation of dam-reservoir interaction problems is presented, being demonstrated the frequency-dependence nature of this type of systems. The fluid compressibility and boundary conditions of the reservoir are analytically identified as the frequency dependent properties of the system.

In this context, the HFTD method is proposed as an ideal method for the dynamic analysis of frequency dependent systems with non-linear behavior, being the dam-reservoir interaction problem taken as a case study to prove this hypothesis.

Therefore, one important part of the research work is the programming and implementation of the HFTD method in MATLAB for SDOF models, and in DIANA Finite Element Software for multi degree of freedom (MDOF) general models. In both cases, the results obtained with the HFTD method are compared with the solutions of analytical or numerical methods in the time domain, like the well-established Newmark method.

Through the entire thesis, special attention is given to the study of the following topics:

- Fluid-structure interaction between the dam and the reservoir.
- Wave differential equation of a fluid fulfilling the following hypothesis: inviscid, compressible, small amplitude motion.
- Influence of fluid compressibility and reservoir boundary conditions assumptions (f.e: bottom absorption) in the seismic response of the system.
- Soil-structure interaction between the dam and the rock foundation.
- Modeling of the far field non-reflecting boundary conditions for the underground and reservoir (radiation effect).
- Modeling of earthquake forces using base acceleration load (earthquake accelerogram).
- Numerical implementation issues of Discrete Fourier Transform (DFT) and HFTD (segmenting approach, time steps, period duration, pseudo force convergence).
- Fluid finite elements.
- Interface finite elements for fluid-structure interaction analysis.

For MDOF systems the HFTD method is compatible with the Mode Superposition method, which reduces considerable the size of the problem and therefore the computational time of analysis. However, the Mode Superposition method also showed to have a great influence in the accuracy of the results, especially when the HFTD method is applied to non-linear MDOF systems.

Additionally, the influence in the HFTD performance of other important implementation issues, like the time segmentation approach, the time step size and the Fourier Transform period, is explained and numerically demonstrated.

One important conclusion of the research work is that the best performance of the HFTD method, in combination with the Mode Superposition method, is obtained when it is applied to MDOF systems with frequency dependent properties and linear or mild non-linear behavior.

From the research work it can also be concluded that these frequency dependent properties actually have an impact in the response of the dam-reservoir interaction system. However, it was not possible to outline any definitive conclusion about the relative importance of their inclusion in the system's response.

Further case studies, especially 3D dam-reservoir interaction models, should be analyzed using the HFTD implementation in DIANA developed in this research work, with the objective of describing more clearly the relative importance of the frequency dependent properties in the linear and non-linear response of this type of structures.

Finally, it is concluded that the HFTD method shows promising perspectives for its application in the academic and professional practice, especially for the seismic analysis of fluid-structure and soil-structure interaction systems, or any other area for which the research of dynamic non-linear behavior of frequency dependent systems becomes every time more important.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A	Interaction surface area
$\mathbf{b}$	Body forces vector
$\mathbf{B}_{\mathrm{F}}$	Generalized fluid matrix multiplying the second time derivate of the pressure vector
$\mathbf{B}_{\mathrm{S}}$	Generalized structure mass matrix
c	Damping coefficient of a single degree of freedom (SDOF) dynamic system
c	Wave or sonic speed of the reservoir fluid
c_{cr}	Critic damping coefficient
c_{fd}	Frequency dependent damping coefficient of a SDOF system
c_{nr}	Viscous coefficient of non-reflective boundary element
c_r	Wave or sonic speed of the reservoir bottom materials
c_S	Structure damping coefficient of a SDOF dynamic systems
$\mathbf{C}$	Damping matrix
$\mathbf{C}_{fd}$	Frequency dependent damping matrix
$\mathbf{C}_{\mathrm{F}}$	Fluid matrix multiplying the first time derivate of the pressure vector
$\mathbf{C}_0$	Constant damping matrix
$\mathbf{C}_{\mathrm{S}}$	Structure damping matrix
$\tilde{\mathbf{C}}_{\mathrm{F}}$	Fluid-structure interaction added damping matrix
$\mathbf{D}_{\mathrm{F}}$	Generalized fluid matrix multiplying the first time derivate of the pressure vector
$\mathbf{D}_{\mathrm{S}}$	Generalized structure damping matrix
E_r	Modulus of elasticity of the reservoir bottom materials
$\mathbf{E}_{\mathrm{S}}$	Generalized structure stiffness matrix
f	Frequency
f_{int}	Non-linear internal force of a SDOF dynamic system
f_p	External force appended decay function
f_q	Pseudo force appended decay function
f_D	Damping force of a SDOF dynamic system
f_S	Stiffness force of a SDOF dynamic system
Δf	Frequency increment or interval
$\boldsymbol{f}_q$	Pseudo force appended decay function vector

$\boldsymbol{f}_{\ddot{u}_g}$	Ground acceleration appended decay function vector
$\mathbf{f}_{\mathrm{b}}$	Base reaction force vector
$\mathbf{f}_{\mathrm{s}}$	Superstructure external load vector
$\mathbf{f}_{\mathrm{S}}^{\mathrm{ext}}$	Structure external load vector
$\mathbf{f}_{\mathrm{S}}^{\mathrm{int}}$	Structure nonlinear internal force vector
F	Maximum sampling frequency
F_q	Vibration frequency of the highest mode included in the analysis
F_{Nyq}	Nyquist frequency
$\mathbf{F}_{\mathrm{g}}$	Effective generalized load vector
g	Gravity constant
H_F	Reservoir height
$\mathbf{i}$	Rigid body displacement vector
$\mathbf{I}$	Rigid body displacement matrix
k	Stiffness of a SDOF dynamic system
k_S	Structure stiffness of a SDOF dynamic systems
$\hat{k}$	Newmark time integration method effective stiffness
$\mathbf{K}$	Stiffness matrix
$\mathbf{K}_{\mathrm{F}}$	Fluid matrix multiplying the pressure vector
$\mathbf{K}_0$	Constant stiffness matrix
$\mathbf{K}_{\mathrm{S}}$	Structure stiffness matrix
l_{Xi}	Participation factor in the X direction of the ith mode of vibration
L_F	Reservoir length
m	Mass of a SDOF dynamic system
$m_{eff,X,i}$	Effective matrix in the X-direction of the ith mode of vibration
m_i	Generalized mass of the ith mode of vibration
m_S	Structure mass of a SDOF dynamic systems
$\mathbf{M}$	Mass matrix
$\mathbf{M}_{\mathrm{F}}$	Fluid matrix multiplying the second time derivate of the pressure vector
$\mathbf{M}_{\mathrm{S}}$	Structure mass matrix
$\widetilde{\mathbf{M}}_{\mathrm{F}}$	Fluid-structure interaction added mass matrix
n	Normal direction
$\mathbf{n}$	Normal vector

N	Number of sampling points for the Fourier Transform
N_S	Number of sampling points per time segment
$\mathbf{N}$	Shape function matrix
p	External load defined in the time domain
p	Hydrodynamic pressure
p_i	Incident hydrodynamic pressure wave
p_0	Harmonic loading amplitude of a SDOF dynamic system
p_r	Reflected hydrodynamic pressure wave
p^k	Modified external load for the time segment k
$\hat{p}$	Complex external load amplitude of a SDOF dynamic system
$\hat{p}_i$	Complex amplitude of the incident hydrodynamic pressure wave
$\hat{p}_r$	Complex amplitude of the reflected hydrodynamic pressure wave
$\Delta\hat{p}$	Newmark time integration method effective load increment
$\hat{\mathbf{p}}$	Complex external load amplitude vector
$\overline{\mathbf{p}}$	Nodal pressure vector
$\hat{P}$	Complex magnitude of the external load defined in the frequency domain
$\hat{P}$	Complex magnitude of the hydrodynamic pressure load
q	Pseudo force of a SDOF dynamic system
q^k	Modified pseudo force for the time segment k
$\hat{q}$	Complex pseudo force amplitude of a SDOF dynamic system
$\mathbf{q}$	Pseudo force vector
$\hat{\mathbf{q}}$	Complex pseudo force amplitude vector
$\overline{\mathbf{q}}^{k,j+1}$	Modified pseudo force vector for the time segment k and iteration j
$\mathbf{R}$	Fluid-structure interaction matrix
t	Time
t_c	Last converged time step
t_0	Load signal duration
Δt	Time increment or time step size
$\mathbf{t}$	Traction surface force
ΔT_d	Decay function duration
ΔT_s	Time segment duration
u	Displacement of a SDOF system

$\mathrm{u_g}$ Base displacement of a SDOF system

u_0 Initial displacement of a SDOF system

u_r Relative displacement of a SDOF system

u_B Bottom reservoir displacement

u_S Structure displacement

$\hat{u}$ Complex displacement amplitude of a SDOF dynamic system

$\hat{u}_r$ Complex relative displacement amplitude of a SDOF system

$\hat{\mathrm{u}}_\mathrm{g}$ Complex base displacement amplitude of a SDOF system

$\mathbf{u}_\mathrm{b}$ Base displacement vector

$\mathbf{u}_\mathrm{g}$ Ground displacement vector

$\mathbf{u}_\mathrm{r}$ Structure displacement vector relative to the base

$\mathbf{u}_\mathrm{s}$ Superstructure displacement vector

$\mathbf{u}_\mathrm{S}$ Structure displacement vector

$\hat{\mathbf{u}}$ Complex displacement amplitude vector

$\hat{\mathbf{u}}_\mathrm{g}$ Complex ground displacement amplitude vector

$\hat{\mathbf{u}}_\mathrm{r}$ Complex displacement amplitude vector relative to the base

$\bar{\mathbf{u}}$ Nodal displacement vector

$\bar{\ddot{\mathbf{u}}}_\mathrm{g}^{\,k}$ Modified ground acceleration vector for the time segment k

U Displacement amplitude of the SDOF steady state response

$\widehat{U}$ Complex displacement amplitude of a SDOF dynamic system

v_F Fluid velocity

v_0 Initial velocity of a SDOF system

$\mathbf{v}$ Velocity vector

$\dot{\mathbf{v}}_\mathrm{F}$ Fluid velocity vector

$\hat{y}_i$ Complex generalized displacement of the i^{th} mode of vibration

$\hat{\mathbf{y}}$ Complex generalized displacement amplitude vector

Z Impedance factor

α	Phase angle (lag) of the response with respect to excitation force
α	Bottom absorption wave reflection coefficient
β	Bulk modulus
β	Newmark time integration method parameter
γ	Newmark time integration method parameter
Γ_{I}	Fluid-structure interface domain
Γ_{b}	Reservoir bottom domain
Γ_{B}	Reservoir bottom absorption domain
Γ_{e}	Reservoir infinite extent domain
Γ_{s}	Reservoir free surface domain
μ	Distributed damping coefficient per volume
ξ	Viscous damping ratio
ξ_H	Hysteretic damping ratio
ρ	Density
ρ_0	Constant hydrostatic density
ρ_r	Density of the bottom reservoir materials
ρ_S	Density of the structure
$\boldsymbol{\sigma}_{\mathrm{S}}$	Structure stress tensor
T_k	Initial time point of segment k
T_p	Fourier period
T_q	Quiet zone duration
T_1	Fundamental period
$\boldsymbol{\phi}_i$	Shape vector of the ith mode of vibration
$\boldsymbol{\Phi}$	Mode shape matrix
ω	Circular frequency of vibration
ω_d	Free vibration frequency of a SDOF under damped system
ω_o	Natural vibration frequency of a SDOF system
Ω	Excitation circular frequency
Ω_1	Fundamental excitation circular frequency

1. INTRODUCTION

1.1. Research project background and motivation

Dynamic analysis of structures can be performed in frequency or in time domain. However both approaches have the following advantages and drawbacks that should be taken into account when facing fluid-structure interaction problems:

- Frequency domain analysis methods are restricted to harmonic loading input. Besides, it is only possible to obtain the steady state response of a linear elastic system.
- Time domain analysis can be applied with any type of loading input. Besides, it is possible to obtain the transient response of a system including several types of non-linearities.
- However, time domain analysis cannot be applied directly in dam-reservoir interaction problems taking into account fluid compressibility or radiation boundary conditions. These kinds of systems are frequency dependent, and therefore, their equation of motion in the time domain is defined in terms of convolution integrals. Due to their complexity, these differential equations with convolution integrals are not directly solved by the classical time domain methods.
- One indirect way of dealing with frequency dependent properties in time domain analysis is following a weakly coupled approach between the solid and fluid equations of motion, by combining both of them in a global iterative procedure. Nevertheless, this also means a large computational effort.
- On the other hand, dam-reservoir interaction problems, including fluid compressibility and radiation boundary conditions, are naturally solved in the frequency domain by means of a decoupled solution procedure of the solid and fluid equations of motion. In general, this is the case for any system with frequency dependent properties.
- Furthermore, frequency domain decoupled solution procedure can be applied in combination with the mode superposition method, attaining a remarkable reduction of the computational effort.

For these reasons, hybrid methods that combine advantages of both approaches and diminish their drawbacks are proposed to solve dynamic dam-reservoir interaction problems, including compressible fluid and radiation boundary conditions.

In this sense, the research work consists of the study and implementation of the Hybrid Frequency Time Domain (HFTD) method, applied to the numerical modeling of the linear and non-linear response of concrete dams under seismic loading, including dynamic dam-reservoir interaction, compressible fluid and radiation boundary conditions.

The thesis includes programming, codification and implementation of the HFTD method in DIANA finite element software. The transient results obtained with the HFTD method for a realistic model of a high concrete dam is studied and compared with the results obtained with other time domain analysis methods available in DIANA.

Through the entire thesis, special attention is given to the study of the following topics:

- Fluid-structure interaction between the dam and the reservoir.
- Wave differential equation of a fluid fulfilling the following hypothesis: inviscid, compressible, small amplitude motion.

- Influence of fluid compressibility and reservoir boundary conditions assumptions (f.e: bottom absorption) in the seismic response of the system.
- Soil-structure interaction between the dam and the rock foundation.
- Modeling of the far field non-reflecting boundary conditions for the underground and reservoir (radiation effect).
- Modeling of earthquake forces using base acceleration load (earthquake accelerogram).
- Numerical implementation issues of Discrete Fourier Transform (DFT) and HFTD (segmenting approach, time steps, period duration, pseudo force convergence).
- Fluid finite elements.
- Interface finite elements for fluid-structure interaction analysis.

1.2. Thesis objectives and research questions

The global objective is to study the HFTD method for the seismic analysis of dam-reservoir interaction problems, determine its performance, and compare the results obtained with the transient response of time domain analysis methods. In this way, the feasibility of an extended and generalized use of the HFTD numerical method for dam's seismic response analysis in engineering professional practice will be evaluated.

A second objective is to study the relevance that frequency dependent properties (fluid compressibility, reservoir boundary conditions) and far field boundary conditions have in the dam-reservoir seismic response.

A third objective is to study simplified and ideal models of dams (for example, one dimensional single degree of freedom systems) which make it possible to have an analytical insight of the results obtained. The results obtained from these simplified models should provide additional information related with the topics mentioned in the second objective.

A fourth objective is to program, codify and implement the HFTD method for linear and nonlinear dynamic analysis of fluid-structure interaction problems in DIANA finite element software.

A fifth objective is to use this implementation for the modeling and analysis of a complex real large scale case study.

Finally, the last objective is to show that HFTD numerical method provides most of the advantages offered separately by time and frequency domain methods, and therefore it is an ideal approach for the solution of dam-reservoir interaction seismic problems.

Based on the five objectives previously presented, the following research questions constitute the guidelines of the thesis:

- What is the influence of the frequency dependent properties (fluid compressibility, radiation boundary conditions) in the seismic response of dam-reservoir interaction systems?
- Which are the most important numerical implementation issues the HFTD method?
- What is the general performance (accuracy, stability and convergence) of the HFTD method for the seismic analysis of dam-reservoir interaction problems?

- Which are the most relevant advantages and drawbacks of the HFTD method for seismic analysis of dam-reservoir interaction systems when compared with time domain analysis methods?
- How feasible is the extended and generalized use of the HFTD numerical method in engineering practice for the computational modeling and seismic analysis of real high dams?

1.3. Methodology of work and organization of the thesis

First, the research methodology starts with the literature review of the theoretical background with special attention to the following topics:

- Theory of wave propagation dynamics
- Frequency domain analysis
- Fourier analysis (DFT and FFT)
- Fluid-structure interaction (Mixed displacement-pressure Finite Element Method formulation)
- Dam-reservoir interaction (boundary conditions)
- HFTD method implementation
- Time domain analysis
- SDOF linear and non-linear dynamic response
- Base acceleration dynamic load
- Soil-structure interaction
- Finite Element modeling of real scale dam-reservoir interaction problems

Second, as a sensitivity exercise, the response of a single degree of freedom model of a dam on rigid foundation is studied. HFTD and Newmark methods are implemented and applied in these models and the results obtained are compared and interpreted. In the case of the HFTD method, compressibility and radiation boundary conditions are taken into account in some cases, whereas in others they are neglected. The effects of these properties in the seismic response of this simple model are analyzed.

Third, the programming, codification and implementation of the HFTD method in DIANA are carried out. A real scale two dimensional dam-reservoir-foundation interaction case study is modeled in DIANA in order to test the HFTD implementation and compare the results obtained with the Newmark time domain method available in the software.

Finally, conclusions about the influence and importance of compressibility, boundary conditions and the method of analysis in the seismic response of dam-reservoir interaction problems are elaborated. The relevance of HFTD method for this type of problems is evaluated.

The case study is a 100 m. height concrete dam, which foundation material is a flexible soil. In DIANA, 2D (plane strain) models of this case study are elaborated, putting special attention to the modeling details and results interpretation. In addition to the seismic time history ground acceleration loading in the horizontal translational direction, hydrostatic water pressure and gravity loads are considered as initial conditions.

The contents of the thesis are organized in six chapters. Chapter 2 presents a succinct theory background review of the frequency domain analysis in single degree of freedom (SDOF) and multi

degrees of freedom (MDOF) systems. The frequency domain analysis theory is developed using complex numbers notation. Topics like mode superposition method, direct method and Fourier analysis are presented.

Chapter 3 develops the Finite Element Method (FEM) formulation of dynamic fluid-structure interaction (FSI) applied to MDOF dam-reservoir interaction systems. The discrete fluid-structure coupled equations of motion are obtained and the influences of the fluid compressibility and the reservoir boundary conditions are analytically demonstrated. In addition, the modified equation of motion for base acceleration loading, together with its underlying assumptions and simplifications, are explained.

Chapter 4 develops the HFTD method formulation applied to the MDOF FEM formulation of the dam-reservoir interaction system presented in Chapter 3. This comprehensive chapter contains the background theory of the method, and a detailed description of the method procedure, implementation issues and input parameters.

In Chapter 5 the dynamic response of SDOF systems is studied. For this purpose, the HFTD and Newmark methods are developed and programmed in MATLAB especially for these SDOF systems. The MATLAB routines are used to test several types of linear and non-linear examples, and compare the results obtained with both methods.

Chapter 6 presents the formulation of a simplified one dimensional dam-reservoir interaction system including considerations like fluid compressibility and radiation boundary condition. The frequency dependent terms introduced by the hydrodynamic pressure are added to the SDOF formulation developed in Chapter 5. Again, the modified HFTD method is used to test a particular example of a simplified dam-reservoir interaction system, and evaluate the influence of compressibility and radiation boundary in its linear and non-linear response.

Finally, Chapter 7 presents the results of the HFTD method implementation in DIANA finite element software. A comprehensive study of the seismic response of 2D foundation-dam-reservoir interaction case study modeled with DIANA is performed. The modeling process, as well as the analysis results, is presented in detail for linear elastic behavior and dam's concrete cracking non-linear behavior. In addition to the HFTD method, different types of dynamic analysis, like eigen-analysis, frequency domain analysis (Direct Method) and time domain transient analysis (Newmark method), are also carried out.

The conclusions and recommendations of the thesis are presented in Chapter 8. Besides, important information about the HFTD implementation in DIANA, the MATLAB routines developed for SDOF systems, and the DIANA HFTD command file for the analysis of the 2D foundation-dam-reservoir interaction case study, is contained in Appendix A, B and C, respectively.

2. FREQUENCY DOMAIN ANALYSIS

2.1. Complex frequency response[1]

The equation of motion of a viscous damped SDOF system subject to a harmonic excitation is expressed in its complex form by Eq. (2.1). The excitation force, defined in Eq. (2.1) in terms of its amplitude (p_0) and excitation frequency (Ω), is a rotating vector force in the complex plane which two components are real (cosine) and imaginary (sine).

$$m\hat{\ddot{u}}(t) + c\hat{\dot{u}}(t) + k\hat{u}(t) = \hat{p}(t) = p_0 e^{i\Omega t} \quad \textbf{(2.1)}$$

The solution of Eq. (2.1) (particular solution) is denominated the steady state response of the system, which in combination with the free vibration solution (complementary solution) and the initial conditions, provides the total response of the system.

The steady state response of the viscous damped SDOF system defined in Eq. (2.1) is also defined in the complex plane as expressed in Eq. (2.2). Furthermore, the complex steady state response of the system can be expressed in terms of the excitation frequency Ω and the complex amplitude $\widehat{U}$, as defined in Eq. (2.3).

$$\hat{u}(t) = u_{\Re} + iu_{\Im} \quad \textbf{(2.2)}$$

$$\hat{u}(t) = \widehat{U}e^{i\Omega t} \quad \textbf{(2.3)}$$

Eq. (2.3) is differentiated with respect to time to obtain the expressions for the velocity and acceleration expressed in Eqs. (2.4) and (2.5), respectively. Both responses can be expressed in terms of the displacement.

$$\hat{\dot{u}}(t) = i\Omega\widehat{U}e^{i\Omega t} = i\Omega\hat{u}(t) \quad \textbf{(2.4)}$$

$$\hat{\ddot{u}}(t) = -\Omega^2\widehat{U}e^{i\Omega t} = -\Omega^2\hat{u}(t) \quad \textbf{(2.5)}$$

The magnitude of the displacement ($\widehat{U}$) is a complex quantity defined in Eq. (2.6) in terms of the real amplitude (U) and the phase angle or phase lag (α) with respect to the excitation force. Different expressions for the displacement, velocity and acceleration are obtained in Eqs. (2.7) to (2.9), respectively, after replacing the complex amplitude given by Eq. (2.6) into Eqs. (2.3) to (2.5).

$$\widehat{U} = Ue^{-i\alpha} \quad \textbf{(2.6)}$$

$$\hat{u}(t) = Ue^{i(\Omega t-\alpha)} \quad \textbf{(2.7)}$$

$$\hat{\dot{u}}(t) = i\Omega Ue^{i(\Omega t-\alpha)} = \Omega Ue^{i\left(\Omega t-\alpha+\frac{\pi}{2}\right)} \quad \textbf{(2.8)}$$

[1] Based on Craig and Kurdilla (2006).

$$\hat{\ddot{u}}(t) = -\Omega^2 U e^{i(\Omega t-\alpha)} = \Omega^2 U e^{i(\Omega t-\alpha+\pi)} \quad \textbf{(2.9)}$$

The system's response and excitation force in the complex plane at time "t" are illustrated in Fig. 2.1, which presents a graphical representation of the phase angle between the force and the response, the magnitude of velocity and acceleration with respect to displacement, and the $\frac{\pi}{2}$ and π rad out-of-phase angles of velocity and acceleration, respectively, with respect to displacement.

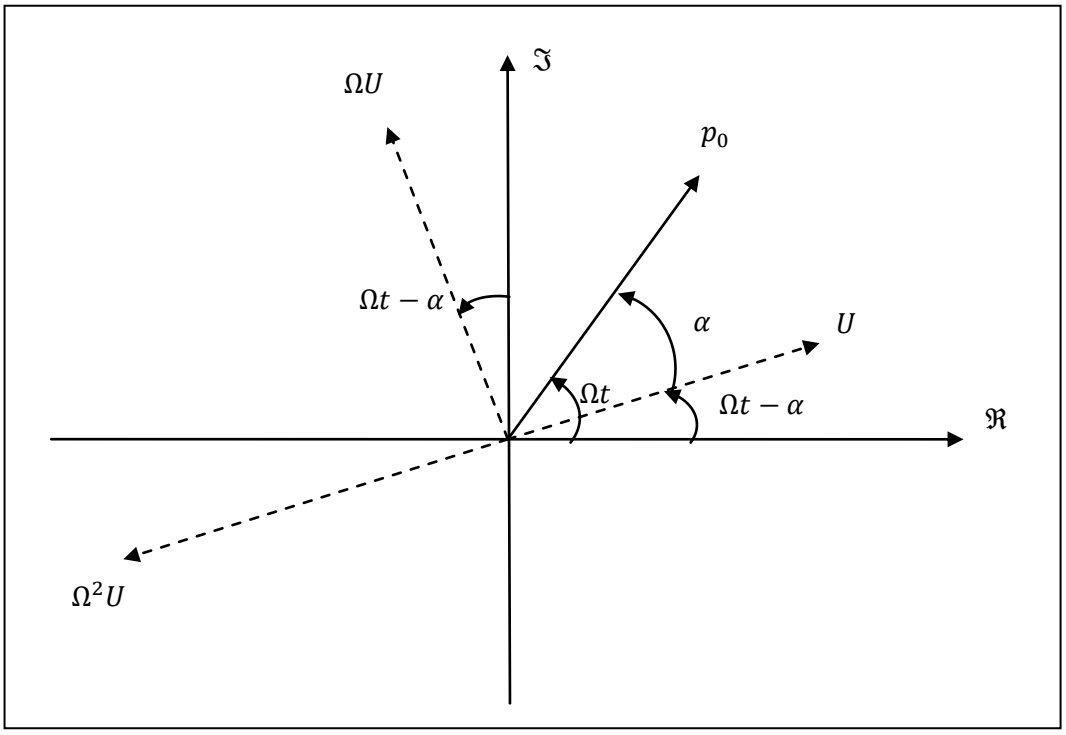


Figure 2.1: Rotating force, displacement, velocity and acceleration vectors in the complex plane.

The substitution of Eqs. (2.3) to (2.5) into Eq. (2.1) results in the expression of the displacement's complex amplitude $\widehat{U}$ given by Eq. (2.10). From this equation, the expressions for the phase angle and the real displacement amplitude are determined in Eqs. (2.11) and (2.12).

$$\widehat{U} = \frac{p_0}{(k-\Omega^2 m)+i\Omega c} \quad \textbf{(2.10)}$$

$$\alpha(\Omega) = \frac{\Omega c}{k-\Omega^2 m} \quad \textbf{(2.11)}$$

$$U(\Omega) = \frac{p_0}{\sqrt{(k-\Omega^2 m)^2+(\Omega c)^2}} \quad \textbf{(2.12)}$$

2.2. Direct method[2]

The equation of motion of a SDOF in the frequency domain defined in Eq. (2.13) is obtained after replacing Eqs. (2.3) to (2.5) into Eq. (2.1). The solution of Eq. (2.13) provides the complex displacement amplitude ($\widehat{U}$) of the steady state response for each excitation frequency (Ω).

$$(k - \Omega^2 m + i\Omega c)\widehat{U} = p_0 \tag{2.13}$$

The extension of Eq. (2.13) to MDOF systems is expressed in Eq. (2.14) in terms of the stiffness matrix ($\mathbf{K}$), the damping matrix ($\mathbf{C}$) and the mass matrix ($\mathbf{M}$). The displacement amplitude and force amplitude are defined by the complex vectors $\widehat{\mathbf{u}}$ and $\widehat{\mathbf{p}}$, respectively. Usually, when the excitation force is defined in the frequency domain, the imaginary part of the force amplitude vector is equal to zero. However, if the excitation force is defined in the time domain, then a direct Fourier Transform is required to be expressed in the frequency domain (See Section 2.4), and therefore the force amplitude vector is complex.

$$(\mathbf{K} - \Omega^2\mathbf{M} + i\Omega\mathbf{C})\widehat{\mathbf{u}} = \widehat{\mathbf{p}} \tag{2.14}$$

One way of solving Eq. (2.14) is by means of the Direct solution method which converts a n x n complex system of equations into the 2n x 2n system with real coefficients given by Eq. (2.15). This double-sized system is obtained as a consequence of splitting the displacement and force amplitude vectors into their real and imaginary components.

$$\begin{bmatrix} \mathbf{K} - \Omega^2\mathbf{M} & \Omega\mathbf{C} \\ -\Omega\mathbf{C} & \mathbf{K} - \Omega^2\mathbf{M} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_{\Im} \\ \mathbf{u}_{\Re} \end{Bmatrix} = \begin{Bmatrix} \mathbf{p}_{\Im} \\ \mathbf{p}_{\Re} \end{Bmatrix} \tag{2.15}$$

The steady state response of the system, given by the real ($\mathbf{u}_{\Re}$) and imaginary ($\mathbf{u}_{\Im}$) parts of the displacement amplitude vector, is obtained from the solution of Eq. (2.15) for each excitation frequency Ω.

2.3. Mode superposition method[3]

For linear systems in which $\mathbf{K}$, $\mathbf{C}$, and $\mathbf{M}$ remain constant, the principle of mode superposition is applicable. This method requires the previous execution of an eigen-analysis from which a number of important modes of vibration are selected. Then, the solution of the system is approximated to the superposition of the solutions corresponding to each selected mode of vibration.

Eq. (2.16) presents the mode shape matrix ($\mathbf{\Phi}$) containing the "m" vibration mode shape vectors ($\boldsymbol{\phi}_i$) selected for the execution of the mode superposition method. The complex displacement amplitude

[2] Based on TNO DIANA (2011).
[3] Based on TNO DIANA (2011).

vector ($\hat{\mathbf{u}}$) is expressed in terms of the complex generalized displacement amplitude ($\hat{\mathbf{y}}$) and the mode shape matrix by means of Eq. (2.17).

$$\mathbf{\Phi}_{\mathrm{n\times m}} = [\boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_i \dots \boldsymbol{\phi}_m] \tag{2.16}$$

$$\hat{\mathbf{u}}_{\mathrm{n}\times 1} = \mathbf{\Phi}_{\mathrm{n\times m}}\hat{\mathbf{y}}_{\mathrm{m}\times 1} \tag{2.17}$$

Eq. (2.17) is substituted in Eq. (2.14) to obtain the equation of motion in terms of the generalized system given by Eq. (2.18). The system of equations is reduced to a m x m system (m < n) by multiplying Eq. (2.18) times the transpose of the mode shape matrix, as expressed in Eq. (2.19) in terms of the generalized stiffness ($\mathbf{K}_{\mathrm{g}}$), mass ($\mathbf{M}_{\mathrm{g}}$) and damping ($\mathbf{C}_{\mathrm{g}}$) matrices defined in Eqs. (2.20) to (2.22), respectively.

$$(\mathbf{K} - \Omega^2\mathbf{M} + i\Omega\mathbf{C})\mathbf{\Phi}\hat{\mathbf{y}} = \hat{\mathbf{p}} \tag{2.18}$$

$$(\mathbf{K}_{\mathrm{g}} - \Omega^2\mathbf{M}_{\mathrm{g}} + i\Omega\mathbf{C}_{\mathrm{g}})\hat{\mathbf{y}} = \mathbf{\Phi}^{\mathrm{T}}\hat{\mathbf{p}} \tag{2.19}$$

$$\mathbf{K}_{\mathrm{g}_{\mathrm{m}\times\mathrm{m}}} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{K}\mathbf{\Phi} \tag{2.20}$$

$$\mathbf{M}_{\mathrm{g}_{\mathrm{m}\times\mathrm{m}}} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{M}\mathbf{\Phi} \tag{2.21}$$

$$\mathbf{C}_{\mathrm{g}_{\mathrm{m}\times\mathrm{m}}} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{C}\mathbf{\Phi} \tag{2.22}$$

The generalized displacement vector ($\hat{\mathbf{y}}$) is determined from the solution of the reduced equation of motion defined in Eq. (2.19), which can be solved for each excitation frequency using the direct method [Eq. (2.15)]. Then $\hat{\mathbf{y}}$ is replaced into Eq. (2.17) to obtain $\hat{\mathbf{u}}$.

A more simplified procedure is possible for a special type of damping matrix $\mathbf{C}$. Due to the orthogonality property of the mass and stiffness matrices with respect to the modal shape matrix, $\mathbf{M}_{\mathrm{g}}$ and $\mathbf{K}_{\mathrm{g}}$ are diagonal matrices. On the other hand, the damping matrix $\mathbf{C}$ is usually not orthogonal with respect to the modal shape matrix and therefore $\mathbf{C}_{\mathrm{g}}$ is not diagonal. However, when $\mathbf{C}$ is proportional to $\mathbf{M}_{\mathrm{g}}$ or $\mathbf{K}_{\mathrm{g}}$ the generalized damping matrix is also diagonal and the system defined by Eq. (2.19) is uncoupled.

For this particular uncoupled system the solution with the mode superposition method for each excitation frequency is given by Eqs. (2.23) and (2.24), which are derived from Eqs. (2.17) and (2.19), respectively.

$$\hat{\mathbf{u}} = \sum_{i=1}^{m} \hat{y}_i\boldsymbol{\phi}_i \tag{2.23}$$

$$\hat{y}_i = \frac{\boldsymbol{\phi}_i^{\mathrm{T}}\hat{\mathbf{p}}}{\boldsymbol{\phi}_i^{\mathrm{T}}\mathbf{K}\boldsymbol{\phi}_i - \Omega^2\boldsymbol{\phi}_i^{\mathrm{T}}\mathbf{M}\boldsymbol{\phi}_i + i\Omega\boldsymbol{\phi}_i^{\mathrm{T}}\mathbf{C}\boldsymbol{\phi}_i} \tag{2.24}$$

2.4. Fourier analysis[4]

2.4.1. Complex Fourier series for periodic excitation

The periodic loading function $p(t)$ of a SDOF system is expressed in terms of its harmonic components by means of the complex Fourier series expansion (also denominated Inverse Fourier Transform) defined in Eq. (2.25).

$$p(t) = \sum_{n=-\infty}^{\infty}\left[\hat{P}_n(\Omega)e^{i(n\Omega_1 t)}\right] \quad \textbf{(2.25)}$$

The fundamental frequency Ω_1 is related to the longest period (T_1) of the harmonic components contained in the loading function $p(t)$ as expressed in Eq. (2.26). The complex magnitude $\hat{P}_n(\Omega)$ of the loading is defined for each harmonic component by the Direct Fourier Transform expressed in Eq. (2.27).

$$\Omega_1 = \frac{2\pi}{T_1} \quad \textbf{(2.26)}$$

$$\hat{P}_n(\Omega) = \frac{1}{T_1}\int_{\tau}^{\tau+\mathrm{T}_1} p(t)e^{-i(n\Omega_1 t)}dt \quad \textbf{(2.27)}$$

2.4.2. Fourier integral for non-periodic excitation

If the loading function $p(t)$ is not periodic, it can be expressed in the frequency domain by means of a Fourier integral. For doing this the fundamental period T_1 is extended to infinity [Eq. (2.28)] and therefore the fundamental frequency Ω_1 becomes a differential term [Eq. (2.29)] and the discrete frequencies become a continuous variable [Eq. (2.30)].

$$\mathrm{T}_1 \to \infty \quad \textbf{(2.28)}$$

$$\Omega_1 = d\Omega \quad \textbf{(2.29)}$$

$$n\Omega_1 = \Omega \quad \textbf{(2.30)}$$

Furthermore, in order to obtain complex loading magnitude $\hat{P}_n(\Omega)$ different than zero when applying the Direct Fourier Transform defined in Eq. (2.27) (T_1) in the denominator tends to infinity), the change of variable expressed in Eq. (2.31) is introduced.

$$\hat{P}(\Omega_n) = \mathrm{T}_1\hat{P}_n(\Omega) \quad \textbf{(2.31)}$$

[4] Based on Craig and Kurdilla (2006).

After replacing Eqs. (2.28) to (2.31) into Eqs. (2.25) and (2.27), and substituting the summatory operators by integrals, the Direct and Inverse Fourier Transform for non-periodic loading functions are defined by Eqs. (2.32) and (2.33), respectively.

$$\hat{P}(\Omega) = \int_{-\infty}^{\infty} p(t) e^{-i(\Omega t)} dt \quad \textbf{(2.32)}$$

$$p(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{P}(\Omega) e^{i(\Omega t)} d\Omega \quad \textbf{(2.33)}$$

2.4.3. Discrete Fourier Transform (DFT)

The Fourier integral expressions given by Eqs. (2.32) and (2.33) are used in the determination of the transient response of a system. However these analytical expressions should be numerically implemented by means of their discrete counterpart, namely, the Discrete Fourier Transform (DFT).

The formulation of the DFT starts with the sampling of the loading function in N points which uniform separation is defined by the time step size Δt. Therefore, the fundamental period (T_1) and the frequency interval (Δf) are defined in Eqs. (2.34) and (2.35), respectively.

$$T_1 = N\,\Delta t \quad \textbf{(2.34)}$$

$$\Delta f = \frac{1}{T_1} \quad \textbf{(2.35)}$$

The continuous time (t) and circular frequency (Ω) variables of Eqs. (2.32) and (2.33) have to be transformed to their equivalents discrete variables for the DFT formulation. In this sense, the discrete time points (t_m) and frequencies (f_n) are defined in Eqs. (2.36) and (2.37), respectively. Similarly, the differential time (dt) and frequency ($d\Omega$) is expressed in terms of the time step and frequency interval by Eqs. (2.38) and (2.39), respectively.

$$t = t_m = m\,\Delta t \quad \textbf{(2.36)}$$

$$\Omega = 2\pi f_n = 2\pi\, n\,\Delta f \quad \textbf{(2.37)}$$

$$dt = \Delta t \quad \textbf{(2.38)}$$

$$d\Omega = 2\pi\,\Delta f \quad \textbf{(2.39)}$$

Finally, the substitution of Eqs. (2.34) to (2.39) into Eq. (2.32) and (2.33), and the replacement of the integrals by summatory operators, provide the expression for the Direct and Inverse DFT defined by Eqs. (2.40) and (2.41), respectively.

$$\hat{P}(f_n) = \Delta t \sum_{m=0}^{N-1} \left[p(t_m) e^{-i\left(\frac{2\pi mn}{N}\right)} \right] \quad \textbf{(2.40)}$$

$$p(t_m) = \frac{1}{N\,\Delta t} \sum_{n=0}^{N-1} \left[\hat{P}(f_n) e^{i\left(\frac{2\pi mn}{N}\right)} \right] \quad \textbf{(2.41)}$$

3. FINITE ELEMENT METHOD (FEM) SEISMIC ANALYSIS OF DAM-RESERVOIR INTERACTION PROBLEMS

3.1. FEM formulation of dynamic fluid-structure interaction (FSI)

3.1.1. Dynamic FSI in dam-reservoir interaction

Dynamic FSI problems consist of coupled systems which usually describe different physical phenomena connected one to the other through a common surface called fluid-structure interface. The coupling of the systems is produced through this interface where unknown interaction forces are produced.

Each physical domain is described by its corresponding differential equations of motion, which in most of the cases are expressed in terms of different dependent variables. These equations are coupled through the boundary conditions imposed in the fluid-structure interface, making the motion of one domain dependent on the motion of the others.

For the seismic analysis of dam-reservoir interaction problems, although the fluid displacement remains small, the interaction with the structure is considerable. The base acceleration load is applied in the rock foundation, and as a consequence, the soil seismic waves propagate and produce the structural vibration of the dam. The structural motion of the dam generates pressures and wave propagation in the reservoir. Consequently, the reservoir waves interact with the dam generating pressure loads on the dam's surface. For this reason, none of the two equations of motion (structure and fluid) can be solved independently of the other.

3.1.2. Eulerian pressure formulation of the fluid wave equation[1]

One way to formulate the coupled FEM equations of motion for FSI problems is using a mixed displacement – scalar potential approach, which defines the solid variables in terms of displacement degrees of freedom (DOF), and the fluid variables in terms of pressure DOF. This is called the Eulerian pressure formulation.

One of the advantages of this type of formulation is the simple description of the fluid domain using a single scalar pressure variable. This reduces considerably the number of variables of the system since only one DOF per node is required to describe the motion of the fluid domain.

Taking into account that for dam-reservoir interaction problems the fluid motion is not substantial but small, considerable simplifications can be made in its equation of motion formulation. Those simplifications are a consequence of the following assumptions only valid for this type of problems:

- Small displacement amplitudes
- Small velocities (convective effects are omitted)
- Inviscid (viscous effects are neglected)
- Small compressibility (variation of density is small)

[1] Based on Zienkiewicz et al. (2005), Rizos and Karabalis (2000), and Zienkiewicz and Bettes (1978).

Therefore, for this type of problems, the linearized Navier-Stokes dynamic equation of the fluid in terms of density (ρ), velocity (**v**), pressure (p) and constant body force of gravity (**b**), is defined in the vectorial Eq. (3.1).

$$\frac{\partial(\rho \mathbf{v})}{\partial t} = -\boldsymbol{\nabla} p + \mathbf{b} \tag{3.1}$$

Considering that a small variation of density is assumed, in Eq. (3.1) ρ can be taken out of the derivative operator and be replaced by the constant hydrostatic density (ρ_0). Furthermore, if the body gravity force is neglected the dynamic equation of fluid is reduced to Eq. (3.2).

$$\rho_0 \dot{\mathbf{v}} = -\boldsymbol{\nabla} p \tag{3.2}$$

On the other hand, the linearized continuity or mass conservation equation of the fluid based on the same assumptions is given by the scalar Eq. (3.3).

$$\rho_0 (\boldsymbol{\nabla} . \mathbf{v}) = \frac{\partial \rho}{\partial t} \tag{3.3}$$

Density variation is a consequence of elastic deformability and therefore it involves a change in pressure. Under the previously stated assumptions, and considering that elastic deformability in fluids depends on the bulk modulus (β), Eq. (3.4) holds.

$$\frac{\partial p}{\partial t} = \frac{\beta}{\rho_0} \frac{\partial \rho}{\partial t} \tag{3.4}$$

If Eq. (3.4) is replaced into Eq. (3.3) and then it is differentiated with respect to time, the continuity equation is expressed in terms of pressure by Eq. (3.5).

$$\boldsymbol{\nabla} . \dot{\mathbf{v}} = -\frac{1}{\beta} \frac{\partial^2 p}{\partial t^2} \tag{3.5}$$

Eq. (3.2) is differentiated with respect to space and transformed into a scalar equation by applying the divergence operator. Then it is combined with Eq. (3.5), eliminating the fluid velocity vector variable (**v**) and obtaining the classical scalar wave equation given by Eq. (3.6), that governs the fluid domain motion. The wave speed (c) is defined in terms of the fluid density and bulk modulus, as shown in Eq. (3.7).

$$\boldsymbol{\nabla}^2 p = \frac{1}{c^2} \ddot{p} \tag{3.6}$$

$$c = \sqrt{\frac{\beta}{\rho}} \tag{3.7}$$

3.1.3. Reservoir boundary conditions[2]

The fluid domain which motion is governed by Eq. (3.6) requires some boundary conditions that complete the system definition. The boundary conditions describe the coupling of the reservoir with the dam at the interface and the radiation condition of infinite extent in the upstream direction. The effect of surface waves in the top of the reservoir and the reflection of pressure waves in the bottom can also be described by the boundary conditions.

In summary, for a reservoir mass of fluid four types of boundary conditions[3] are generally defined as shown in Fig. 3.1.

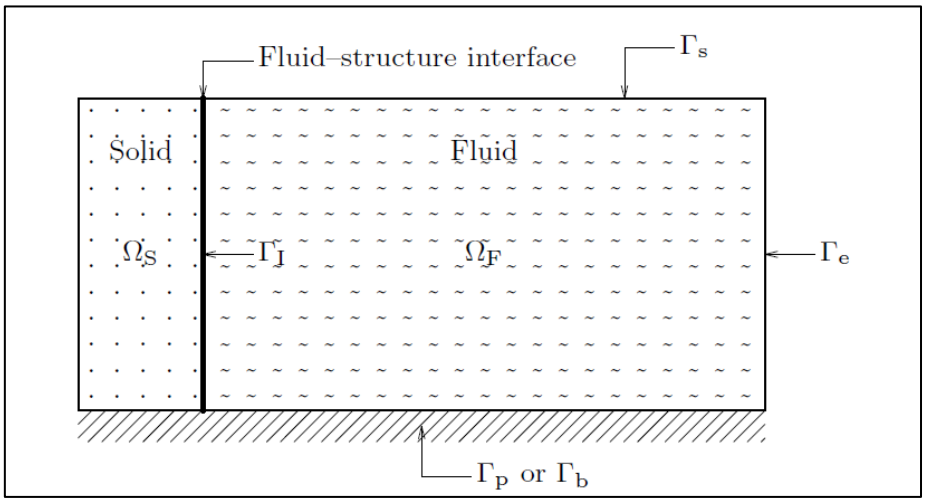


Figure 3.1: Fluid and solid domains, fluid-structure interface and boundary conditions (TNO DIANA BV, 2011).

3.1.3.1. Fluid-structure interface (Γ_I)

The expressions describing the boundary conditions should be expressed in terms of pressure (p) (essential boundary condition) or pressure gradients in the normal direction to the boundary surface ($\frac{\partial p}{\partial n}$) (natural boundary condition).

Taking this into account, the boundary condition for the fluid-structure interface is obtained by multiplying the transpose of the normal vector to the boundary surface ($\mathbf{n}^T$)[4] by Eq. (3.2). Then, the time derivative of the fluid normal velocity ($\dot{v}_{Fn}$) is prescribed in relation with the pressure gradient in the normal direction ($\frac{\partial p}{\partial n}$), as shown in Eqs. (3.8) and (3.9).

$$\mathbf{n}^T \nabla p = -\rho_0 (\mathbf{n}^T \dot{\mathbf{v}}_F) \quad \textbf{(3.8)}$$

$$\frac{\partial p}{\partial n} = -\rho_0 \dot{v}_{Fn} \quad \textbf{(3.9)}$$

[2] Based on Zienkiewicz et al. (2005), Fenves and Chopra (1983), Fenves and Chopra (1984) and TNO DIANA BV (2011).
[3] Generally, for 2D models one of these four boundary conditions belongs to each side of the fluid region.
[4] The normal vector **n** points outward of the fluid domain.

The coupling between the fluid and the structure is achieved through the motion continuity of both physical domains. In this sense, the prescribed fluid normal velocity ($\dot{v}_{Fn}$) is coupled with the structure normal displacement (u_{Sn}) as expressed in Eq. (3.10).

$$v_{Fn} = \dot{u}_{Sn} = \mathbf{n}^{\mathrm{T}}\dot{\mathbf{u}}_{\mathrm{S}} \tag{3.10}$$

The final expression of the fluid-structure interface boundary condition is given by Eq. (3.11), after replacing Eq. (3.10) into Eq. (3.9).

$$\frac{\partial p}{\partial n} + \rho_0(\mathbf{n}^{\mathrm{T}}\ddot{\mathbf{u}}_{\mathrm{S}}) = 0 \qquad \text{on} \qquad \Gamma_{\mathrm{I}} \tag{3.11}$$

3.1.3.2. Free surface waves (Γ_s)

It is possible to specify two types of boundary conditions for the free surface of the reservoir. The first and simplest one is a consequence of neglecting the effect of the surface waves, prescribing a pressure equal to zero in the horizontal top free surface, as expressed in Eq. (3.12). This is an essential or Dirichlet type of boundary condition.

$$p = 0 \qquad \text{on} \qquad \Gamma_{\mathrm{s}} \tag{3.12}$$

The second option for the free surface boundary condition is to use Eq. (3.13) as an approximated expression to determine the pressure caused by the surface gravity waves. In this case, u_{Fz} is the vertical elevation (Z-direction) of the fluid relative to the mean surface.

$$p = \rho_0 g u_{Fz} \tag{3.13}$$

Eq. (3.14) is obtained from Eq. (3.9) after defining z as the normal direction (n).

$$\frac{\partial p}{\partial z} = -\rho_0 \dot{v}_{Fz} = -\rho_0 \ddot{u}_{Fz} \tag{3.14}$$

The combination of Eqs. (3.13) and (3.14) allows the elimination of u_{Fz} obtaining the linearized surface wave condition given by Eq. (3.15)

$$\frac{\partial p}{\partial z} + \frac{1}{g}\ddot{p} = 0 \qquad \text{on} \qquad \Gamma_{\mathrm{s}} \tag{3.15}$$

3.1.3.3. Radiation boundary of infinite extent (Γ_e)

Considering only the fluid wave propagation in X-direction, the general solution of the wave equation given by Eq. (3.6) is expressed in Eq. (3.16).

$$p = F(x - ct) + G(x + ct) \quad \textbf{(3.16)}$$

On the radiation boundary, which is theoretically located at an infinite distance from the dam in the X-direction, no incoming waves are entering into the system (only outgoing waves), and therefore Eqs. (3.17) and (3.18) describe the pressure wave propagation on this boundary.

$$G(x + ct) = 0 \quad \textbf{(3.17)}$$

$$p = F(x - ct) \quad \textbf{(3.18)}$$

Eqs. (3.19) and (3.20) are obtained after differentiate Eq. (3.18) with respect to space (x) and time (t), respectively.

$$\frac{\partial p}{\partial x} = F' \quad \textbf{(3.19)}$$

$$\frac{\partial p}{\partial t} = -cF' \quad \textbf{(3.20)}$$

After eliminating F′ from Eqs. (3.19) and (3.20), the radiation boundary condition of infinite extent given by Eq. (3.21) is obtained.

$$\frac{\partial p}{\partial x} + \frac{1}{c}\dot{p} = 0 \quad \text{on} \quad \Gamma_e \quad \textbf{(3.21)}$$

3.1.3.4. Bottom absorption (Γ_b)

Similarly to Eq. (3.6), the wave equation that describes the movement of the bottom materials in terms of the normal displacement to the bottom surface (u_{Bn}) and wave speed (c_r) is defined by Eqs. (3.22) and (3.23).

$$\boldsymbol{\nabla}^2 u_{Bn} = \frac{1}{c_r^2}\ddot{u}_{Bn} \quad \textbf{(3.22)}$$

$$c_r = \sqrt{\frac{E_r}{\rho_r}} \quad \textbf{(3.23)}$$

Taking into account that in the reservoir bottom only outward propagating waves in the normal direction are considered, and that there is no incoming propagation waves due to the radiation condition assumed for the thick layer of bottom materials, the general solution of Eq. (3.22) is given by Eq. (3.24).

$$u_{Bn} = F(n - c_r t) \quad \textbf{(3.24)}$$

On the other hand, the equilibrium equation is obtained in terms of the fluid pressure (defined positive for compression) and the normal stresses in the bottom materials surface, as expressed by Eq. (3.25).

$$p = -E_r \frac{\partial u_{Bn}}{\partial n} \quad \textbf{(3.25)}$$

Eq. (3.26) is obtained after replacing Eq. (3.24) into Eq. (3.25), whereas its derivative with respect to time is given by Eq. (3.27).

$$p = -E_r F' \quad \textbf{(3.26)}$$

$$\dot{p} = E_r c_r F'' \quad \textbf{(3.27)}$$

The interaction and continuity between the fluid and the bottom materials is ensured by the compatibility of displacements, required to derive Eq. (3.28) from Eq. (3.14). Eq. (3.29) is obtained after replacing Eq. (3.24) into Eq. (3.28).

$$\frac{\partial p}{\partial n} = -\rho_0 \ddot{u}_{Bn} \quad \textbf{(3.28)}$$

$$\frac{\partial p}{\partial n} = -\rho_0 c_r^2 F'' \quad \textbf{(3.29)}$$

After combining Eqs. (3.27) and (3.29), the term F" is eliminated to obtain Eq. (3.30). Finally, the bottom absorption boundary condition defined by Eqs. (3.31) and (3.32) is obtained after eliminating E_r through the combination of Eqs. (3.23) and (3.30).

$$\frac{\partial p}{\partial n} = -\frac{\rho_0 c_r}{E_r} \dot{p} \quad \textbf{(3.30)}$$

$$\frac{\partial p}{\partial n} + q\dot{p} = 0 \quad \text{on} \quad \Gamma_e \quad \textbf{(3.31)}$$

$$q = \frac{\rho_0}{\rho_r c_r} \quad \textbf{(3.32)}$$

To determine the constant q in terms of the wave reflection coefficient (α), the harmonic pressure wave propagating in the normal direction of the bottom surface is defined as the summation of a outward incident wave (p_i) and an inward reflected wave (p_r), as expressed by Eqs. (3.33) to (3.35).

$$p = p_i + p_r \quad \textbf{(3.33)}$$

$$p_i = e^{i\omega(-\frac{n}{c}+t)} \quad \textbf{(3.34)}$$

$$p_r = \alpha e^{i\omega(\frac{n}{c}+t)} \quad \textbf{(3.35)}$$

After the substitution of Eqs. (3.33) to (3.35) into Eq. (3.31), the relation given by Eq. (3.36) is obtained for the bottom boundary (n=0). The expression of q in terms of α is given by Eq. (3.37), obtained after the cancellation of the exponential terms of eq. (3.36).

$$-\frac{i\omega}{c}\left(e^{i\omega t}-\alpha e^{i\omega t}\right)=-qi\omega(e^{i\omega t}+\alpha e^{i\omega t}) \quad \textbf{(3.36)}$$

$$q=\frac{1-\alpha}{c(1+\alpha)} \quad \textbf{(3.37)}$$

Eq. (3.37) is replaced into Eq. (3.31) to obtain the bottom absorption boundary condition expressed by Eq. (3.38) in terms of the wave reflection coefficient (α).

$$\frac{\partial p}{\partial n}+\frac{1-\alpha}{c(1+\alpha)}\dot{p}=0 \qquad \text{on} \qquad \Gamma_b \quad \textbf{(3.38)}$$

3.1.4. FEM formulation: weak form of the coupled system[5]

The equation of motion of the solid part of the system (dam and foundation) is given by Eq. (3.39), whereas the natural or Neumann boundary condition is given by Eq. (3.40)[6].

$$\rho_S\ddot{\mathbf{u}}_S+\mu\dot{\mathbf{u}}_S-\boldsymbol{\nabla}.\boldsymbol{\sigma}_S-\mathbf{b}=\mathbf{0} \quad \textbf{(3.39)}$$

$$\boldsymbol{\sigma}_S\mathbf{n}_S-\mathbf{t}=\mathbf{0} \qquad \text{on} \qquad \Gamma_h \quad \textbf{(3.40)}$$

The weak form of the solid part of the system is expressed in Eq. (3.41). Besides, Eq. (3.42) is obtained after applying integration by parts and the Green's theorem over the term containing the stress tensor ($\boldsymbol{\nabla}.\boldsymbol{\sigma}_S$) in Eq. (3.41).

$$\int_{\Omega_S}\delta\mathbf{u}_S^T[\rho_S\ddot{\mathbf{u}}_S+\mu\dot{\mathbf{u}}_S-\boldsymbol{\nabla}.\boldsymbol{\sigma}_S-\mathbf{b}]\,d\Omega+\int_{\Gamma_h}\delta\mathbf{u}_S^T[\boldsymbol{\sigma}_S\mathbf{n}_S-\mathbf{t}]\,d\Gamma=0 \quad \textbf{(3.41)}$$

$$\int_{\Omega_S}\delta\mathbf{u}_S^T[\rho_S\ddot{\mathbf{u}}_S+\mu\dot{\mathbf{u}}_S-\mathbf{b}]\,d\Omega+\int_{\Omega_S}\boldsymbol{\nabla}\delta\mathbf{u}_S:\boldsymbol{\sigma}_S\,d\Omega-\int_{\Gamma_h}\delta\mathbf{u}_S^T\mathbf{t}\,d\Gamma=0 \quad \textbf{(3.42)}$$

The interaction between the fluid and the structure is introduced in the weak form of the solid equation of motion [Eq. (3.42)] as a Neumann boundary condition in the interface (Γ_I), by relating the surface traction stresses applied in the fluid-structure interface of the solid domain with the hydrodynamic pressure (p) defined positive in compression. This interaction boundary condition for the solid domain is expressed in Eq. (3.43).

$$\mathbf{t}=-p\mathbf{n}_S=p\mathbf{n} \quad \textbf{(3.43)}$$

[5] Based on Zienkiewicz et al. (2005) and Zienkiewicz and Bettes (1978).
[6] The normal vector $\mathbf{n}_S$ points outward of the solid domain.

After replacing Eq. (3.43) into Eq. (3.42) the final expression of the weak form of the solid part of the system is obtained in Eq. (3.44). This expression is valid for linear and nonlinear analysis. The most common nonlinearities, like material or geometric nonlinearities, can be introduced in the terms $\boldsymbol{\nabla}\delta\mathbf{u_S}$ and $\boldsymbol{\sigma_S}$.

$$\int_{\Omega_S} \delta\mathbf{u}_S^T[\rho_S\ddot{\mathbf{u}}_S + \mu\dot{\mathbf{u}}_S - \mathbf{b}]\, d\Omega + \int_{\Omega_S} \boldsymbol{\nabla}\delta\mathbf{u}_S{:}\,\boldsymbol{\sigma}_S\, d\Omega - \int_{\Gamma_I} \delta\mathbf{u}_S^T\mathbf{n}p\, d\Gamma = 0 \qquad \textbf{(3.44)}$$

With the objective to determine the weak form of the fluid part of the system, its wave equation and boundary conditions previously defined are reproduced below.

$$\frac{1}{c^2}\ddot{p} - \boldsymbol{\nabla}^2 p = 0 \qquad \textbf{(3.6)}$$

$$\frac{\partial p}{\partial n} + \rho_0(\mathbf{n}^T\ddot{\mathbf{u}}_S) = 0 \quad \text{on} \quad \Gamma_I \qquad \textbf{(3.11)}$$

$$\frac{\partial p}{\partial z} + \frac{1}{g}\ddot{p} = 0 \quad \text{on} \quad \Gamma_s \qquad \textbf{(3.15)}$$

$$\frac{\partial p}{\partial x} + \frac{1}{c}\dot{p} = 0 \quad \text{on} \quad \Gamma_e \qquad \textbf{(3.21)}$$

$$\frac{\partial p}{\partial n} + \frac{1-\alpha}{c(1+\alpha)}\dot{p} = 0 \quad \text{on} \quad \Gamma_b \qquad \textbf{(3.38)}$$

The weak form of the fluid part of the system is expressed in Eq. (3.45). Eq. (3.46) is obtained after applying integration by parts and the Green's theorem over the term $\boldsymbol{\nabla}^2 p$ of Eq. (3.45).

$$\int_{\Omega_F} \delta p\left[\frac{1}{c^2}\ddot{p} - \boldsymbol{\nabla}^2 p\right] d\Omega + \int_{\Gamma_I} \delta p\left[\frac{\partial p}{\partial n} + \rho_0(\mathbf{n}^T\ddot{\mathbf{u}}_S)\right] d\Gamma + \int_{\Gamma_s} \delta p\left[\frac{\partial p}{\partial z} + \frac{1}{g}\ddot{p}\right] d\Gamma + \int_{\Gamma_e} \delta p\left[\frac{\partial p}{\partial x} + \frac{1}{c}\dot{p}\right] d\Gamma + \int_{\Gamma_b} \delta p\left[\frac{\partial p}{\partial n} + \frac{1-\alpha}{c(1+\alpha)}\dot{p}\right] d\Gamma = 0 \qquad \textbf{(3.45)}$$

$$\int_{\Omega_F} \delta p\,\frac{1}{c^2}\ddot{p}\, d\Omega + \int_{\Omega_F} \boldsymbol{\nabla}\delta p\boldsymbol{\nabla}p\, d\Omega + \int_{\Gamma_I} \delta p[\rho_0(\mathbf{n}^T\ddot{\mathbf{u}}_S)]\, d\Gamma + \int_{\Gamma_s} \delta p\left[\frac{1}{g}\ddot{p}\right] d\Gamma + \int_{\Gamma_e} \delta p\left[\frac{1}{c}\dot{p}\right] d\Gamma + \int_{\Gamma_b} \delta p\left[\frac{1-\alpha}{c(1+\alpha)}\dot{p}\right] d\Gamma = 0 \qquad \textbf{(3.46)}$$

3.1.5. Discrete coupled problem definition: Galerkin method[7]

The displacement and pressure fields are discretized using the approximated expressions given by Eqs. (3.47) and (3.48), respectively. $\overline{\mathbf{u}}$ and $\overline{\mathbf{p}}$ are the vectors containing the nodal values for each degree of freedom of the solid and fluid part of the system, respectively, whereas $\mathbf{N_u}$ and $\mathbf{N_p}$ are the corresponding shape function matrices.

$$\mathbf{u}_S = \mathbf{N}_u\overline{\mathbf{u}} \qquad \textbf{(3.47)}$$

[7] Based on Zienkiewicz et al. (2005) and Zienkiewicz and Bettes (1978).

$$p = \mathbf{N}_{\mathrm{p}}\overline{\mathbf{p}} \tag{3.48}$$

After replacing Eqs. (3.47) and (3.48) into Eq. (3.44), the standard Galerkin discretization applied to the weak form of the solid equation is defined by Eq. (3.49). For the particular case of linear structural systems, the discrete equation of motion is transformed as shown in Eq. (3.50).

$$\mathbf{M}_{\mathrm{S}}\ddot{\overline{\mathbf{u}}} + \mathbf{C}_{\mathrm{S}}\dot{\overline{\mathbf{u}}} + \mathbf{f}_{\mathrm{S}}^{\mathrm{int}}(\overline{\mathbf{u}}) - \mathbf{R}^{\mathrm{T}}\overline{\mathbf{p}} = \mathbf{f}_{\mathrm{S}}^{\mathrm{ext}}(t) \tag{3.49}$$

$$\mathbf{M}_{\mathrm{S}}\ddot{\overline{\mathbf{u}}} + \mathbf{C}_{\mathrm{S}}\dot{\overline{\mathbf{u}}} + \mathbf{K}_{\mathrm{S}}\overline{\mathbf{u}} - \mathbf{R}^{\mathrm{T}}\overline{\mathbf{p}} = \mathbf{f}_{\mathrm{S}}^{\mathrm{ext}}(t) \tag{3.50}$$

The mass ($\mathbf{M_S}$), damping ($\mathbf{C_S}$), stiffness ($\mathbf{K_S}$) and interaction ($\mathbf{R}$) matrices, as well as the external ($\mathbf{f_S^{ext}}$) and internal ($\mathbf{f_S^{int}}$) force vectors appearing in Eqs. (3.49) and (3.50), are directly obtained from Eqs. (3.44), (3.47) and (3.48). For instance, the expression of the interaction matrix is reproduced below in Eq. (3.51).

$$\mathbf{R} = \int_{\Gamma_{\mathrm{I}}} \mathbf{N}_{\mathrm{p}}^{\mathrm{T}}\mathbf{n}^{\mathrm{T}}\mathbf{N}_{\mathrm{u}}\, d\Gamma \tag{3.51}$$

On the other hand, after replacing Eqs. (3.47) and (3.48) into Eq. (3.46), the standard Galerkin discretization applied to the weak form of the fluid equation is defined by Eq. (3.52). The expressions defining the matrices $\mathbf{M_F}$, $\mathbf{C_F}$ and $\mathbf{K_F}$ are given by Eqs. (3.53) to (3.55), respectively.

$$\mathbf{M}_{\mathrm{F}}\ddot{\overline{\mathbf{p}}} + \mathbf{C}_{\mathrm{F}}\dot{\overline{\mathbf{p}}} + \mathbf{K}_{\mathrm{F}}\overline{\mathbf{p}} + \rho_0\mathbf{R}\ddot{\overline{\mathbf{u}}} = \mathbf{0} \tag{3.52}$$

$$\mathbf{M}_{\mathrm{F}} = \frac{1}{c^2}\int_{\Omega_{\mathrm{F}}} \mathbf{N}_{\mathrm{p}}^{\mathrm{T}}\mathbf{N}_{\mathrm{p}}\, d\Omega + \frac{1}{g}\int_{\Gamma_{\mathrm{s}}} \mathbf{N}_{\mathrm{p}}^{\mathrm{T}}\mathbf{N}_{\mathrm{p}}\, d\Gamma \tag{3.53}$$

$$\mathbf{C}_{\mathrm{F}} = \frac{1}{c}\int_{\Gamma_{\mathrm{e}}} \mathbf{N}_{\mathrm{p}}^{\mathrm{T}}\mathbf{N}_{\mathrm{p}}\, d\Gamma + \frac{1-\alpha}{c(1+\alpha)}\int_{\Gamma_{\mathrm{b}}} \mathbf{N}_{\mathrm{p}}^{\mathrm{T}}\mathbf{N}_{\mathrm{p}}\, d\Gamma \tag{3.54}$$

$$\mathbf{K}_{\mathrm{F}} = \int_{\Omega_{\mathrm{F}}} \boldsymbol{\nabla}\mathbf{N}_{\mathrm{p}}^{\mathrm{T}}\boldsymbol{\nabla}\mathbf{N}_{\mathrm{p}}\, d\Omega \tag{3.55}$$

3.1.6. Solution of the coupled system in the time domain[8]

The coupled system of equations defined in the general case by Eqs. (3.49) and (3.52) can be solved in time domain using time stepping procedures. Both equations have to be first discretized in time separately and then combined, with the objective of eliminating one of the variables ($\overline{\mathbf{p}}$) and solve the resulting equation for the remaining variable ($\overline{\mathbf{u}}$). However, the computational effort required to solve large coupled systems with time stepping procedures is high due to the unsymmetrical characteristics of the matrices appearing in the combined equation. Furthermore, unconditional stability of the time integration algorithms is limited even for linear elastic analysis.

[8] Based on Zienkiewicz et al. (2005).

For this reason, staggered solution procedures are used as a more efficient alternative to solve fluid-structure problems in the time domain. In this type of procedure both equation are solved independently by means of an artificial uncoupling. This is achieved introducing an assumed or predicted value of the vector $\overline{\mathbf{p}}$ in Eq. (3.49) and solving this equation independently with time stepping procedures. Consequently, the solution obtained for the vector $\overline{\mathbf{u}}$ is then replaced in Eq. (3.52) and this equation is solved to obtain a new value of $\overline{\mathbf{p}}$. The iterative procedure is repeated until convergence for both vectors $\overline{\mathbf{u}}$ and $\overline{\mathbf{p}}$ is achieved.

The advantage of using staggered process is the reduction of the size of the problem and the computational effort due to the fact that two smaller and uncoupled (instead of one larger and coupled) systems of equations are solved separately. However, it is important to mention that in most of the cases staggered solution procedures are only conditionally stable.

In a staggered solution procedure the modified solid domain equation for nonlinear problems is given by Eq. (3.56) after replacing the predicted nodal pressure vector ($\overline{\mathbf{p}}_{n+1}^{p}$) into Eq. (3.49).

$$\mathbf{M}_{\mathrm{S}}\ddot{\overline{\mathbf{u}}}_{n+1} + \mathbf{C}_{\mathrm{S}}\dot{\overline{\mathbf{u}}}_{n+1} + \mathbf{f}_{\mathrm{S}}^{\mathrm{int}}(\overline{\mathbf{u}}_{n+1}) = \mathbf{f}_{\mathrm{S}}^{\mathrm{ext}}{}_{n+1}(t) + \mathbf{R}^{\mathrm{T}}\overline{\mathbf{p}}_{n+1}^{p} \qquad \textbf{(3.56)}$$

An incremental iterative scheme (f.e. Newton-Raphson method) and a time integration scheme (f.e. Newmark method) have to be chosen in order to solve Eq. (3.56) and find an approximated solution for the nodal acceleration vector in the structure ($\ddot{\overline{\mathbf{u}}}_{n+1}^{p}$), which is introduced in Eq. (3.52) to obtain the modified fluid domain equation for linear behavior given by Eq. (3.57).

$$\mathbf{M}_{\mathrm{F}}\ddot{\overline{\mathbf{p}}}_{n+1} + \mathbf{C}_{\mathrm{F}}\dot{\overline{\mathbf{p}}}_{n+1} + \mathbf{K}_{\mathrm{F}}\overline{\mathbf{p}}_{n+1} = -\rho_0\mathbf{R}\ddot{\overline{\mathbf{u}}}_{n+1}^{p} \qquad \textbf{(3.57)}$$

Due to linear behavior of the fluid, the solution of Eq. (3.57) only requires the election of a time integration method, not being required to follow an iterative procedure. A new vector $\overline{\mathbf{p}}_{n+1}^{p}$ is obtained to be introduced again in Eq. (3.56), and the iterative process is continued until convergence is achieved. Finally, the whole procedure is repeated for the next time step.

3.1.7. Solution of the coupled system in the frequency domain[9]

A solution in the frequency domain is only valid for linear behavior of the structure. The coupled system defined by Eqs. (3.50) and (3.52) can be compactly expressed by Eq. (3.58).

$$\begin{bmatrix}\mathbf{M}_{\mathrm{S}} & \mathbf{0}\\ \rho_0\mathbf{R} & \mathbf{M}_{\mathrm{F}}\end{bmatrix}\begin{Bmatrix}\ddot{\overline{\mathbf{u}}}\\ \ddot{\overline{\mathbf{p}}}\end{Bmatrix} + \begin{bmatrix}\mathbf{C}_{\mathrm{S}} & \mathbf{0}\\ \mathbf{0} & \mathbf{C}_{\mathrm{F}}\end{bmatrix}\begin{Bmatrix}\dot{\overline{\mathbf{u}}}\\ \dot{\overline{\mathbf{p}}}\end{Bmatrix} + \begin{bmatrix}\mathbf{K}_{\mathrm{S}} & -\mathbf{R}^{\mathrm{T}}\\ \mathbf{0} & \mathbf{K}_{\mathrm{F}}\end{bmatrix}\begin{Bmatrix}\overline{\mathbf{u}}\\ \overline{\mathbf{p}}\end{Bmatrix} = \begin{Bmatrix}\mathbf{f}_{\mathrm{S}}^{\mathrm{ext}}(t)\\ \mathbf{0}\end{Bmatrix} \qquad \textbf{(3.58)}$$

It is possible to determine the steady state solution of this coupled system of equations in the frequency domain by expressing the nodal pressure vector ($\overline{\mathbf{p}}$) in terms of the nodal structural

[9] Based on TNO DIANA BV (2011).

displacement vector ($\overline{\mathbf{u}}$). First, the external force vector should be expressed or transformed into the harmonic form shown in Eq. (3.59).

$$\mathbf{f}_{\mathrm{S}}^{\mathrm{ext}}(t) = \hat{\mathbf{f}}_{\mathrm{S}}^{\mathrm{ext}} \mathrm{e}^{i\omega t} \tag{3.59}$$

Considering that both, the structural displacements and the fluid pressures, are produced by the external force, the steady state solution of both responses will exist in the same harmonic form as expressed in Eqs. (3.60) and (3.61).

$$\overline{\mathbf{u}}(t) = \hat{\mathbf{u}} \mathrm{e}^{i\omega t} \tag{3.60}$$

$$\overline{\mathbf{p}}(t) = \hat{\mathbf{p}} \mathrm{e}^{i\omega t} \tag{3.61}$$

After replacing Eqs. (3.59) to (3.61) into Eq. (3.58), the system of equations shown in Eq. (3.62) is obtained, where the complex amplitudes of the nodal structural displacements ($\hat{\mathbf{u}}$) and pressure ($\hat{\mathbf{p}}$) vectors are the unknown variables.

$$\begin{bmatrix} -\omega^2 \mathbf{M}_{\mathrm{S}} + i\omega \mathbf{C}_{\mathrm{S}} + \mathbf{K}_{\mathrm{S}} & -\mathbf{R}^{\mathrm{T}} \\ -\omega^2 \rho_0 \mathbf{R} & -\omega^2 \mathbf{M}_{\mathrm{F}} + i\omega \mathbf{C}_{\mathrm{F}} + \mathbf{K}_{\mathrm{F}} \end{bmatrix} \begin{Bmatrix} \hat{\mathbf{u}} \\ \hat{\mathbf{p}} \end{Bmatrix} = \begin{Bmatrix} \hat{\mathbf{f}}_{\mathrm{S}}^{\mathrm{ext}} \\ \mathbf{0} \end{Bmatrix} \tag{3.62}$$

From the second row of Eq. (3.62) the amplitude of the nodal pressure vector can be expressed in terms of the amplitude of the nodal displacement vector as defined in Eqs. (3.63) and (3.64).

$$\hat{\mathbf{p}} = \omega^2 \rho_0 \mathbf{H}_{\mathrm{F}} \mathbf{R} \hat{\mathbf{u}} \tag{3.63}$$

$$\mathbf{H}_{\mathrm{F}}(\omega) = [-\omega^2 \mathbf{M}_{\mathrm{F}} + i\omega \mathbf{C}_{\mathrm{F}} + \mathbf{K}_{\mathrm{F}}]^{-1} \tag{3.64}$$

After replacing the expression of $\hat{\mathbf{p}}$ given by Eq. (3.63) into the first row of Eq. (3.62), the uncoupled equation to determine the steady state response of the structure in the frequency domain, in terms of the complex amplitudes of the nodal displacement vector ($\hat{\mathbf{u}}$), is given by Eqs. (3.65) to (3.67). The effect of the reservoir interaction in the dam's equation of motion is represented by $\widetilde{\mathbf{M}}_{\mathrm{F}}$ and $\tilde{\mathbf{C}}_{\mathrm{F}}$, called added mass and added damping, respectively.

$$\left[-\omega^2 \left(\mathbf{M}_{\mathrm{S}} + \widetilde{\mathbf{M}}_{\mathrm{F}}\right) + i\omega \left(\mathbf{C}_{\mathrm{S}} + \tilde{\mathbf{C}}_{\mathrm{F}}\right) + \mathbf{K}_{\mathrm{S}}\right] \hat{\mathbf{u}} = \hat{\mathbf{f}}_{\mathrm{S}}^{\mathrm{ext}} \tag{3.65}$$

$$\widetilde{\mathbf{M}}_{\mathrm{F}} = \rho_0 \Re\left(\mathbf{R}^{\mathrm{T}} \mathbf{H}_{\mathrm{F}} \mathbf{R}\right) \tag{3.66}$$

$$\tilde{\mathbf{C}}_{\mathrm{F}} = -\rho_0 \omega \Im\left(\mathbf{R}^{\mathrm{T}} \mathbf{H}_{\mathrm{F}} \mathbf{R}\right) \tag{3.67}$$

Eqs. (3.65) to (3.67) can be solved in the frequency domain using the Direct or the Mode superposition methods (see Chapter 2).

3.1.8. Special case for incompressible fluids[10]

If compressibility and surface gravity waves boundary condition are neglected, a simplified uncoupled equation describing the fluid-structure interaction is obtained instead of the coupled system of equations defined by Eqs. (3.49) and (3.52).

Incompressible fluid assumption ($c = \infty$) combined with a prescribed pressure equal to zero in the horizontal top free surface [Eq. (3.12)], produce null matrices $\mathbf{M_F}$ and $\mathbf{C_F}$, as expressed by Eqs. (3.68) and (3.69), respectively.

$$\mathbf{M}_{\mathrm{F}} = \mathbf{0} \tag{3.68}$$

$$\mathbf{C}_{\mathrm{F}} = \mathbf{0} \tag{3.69}$$

Replacing Eqs. (3.68) and (3.69) into Eq. (3.52) results in the simplified explicit discrete equation given by Eq. (3.70), which describes the incompressible fluid motion.

$$\overline{\mathbf{p}} = -\rho_0 \mathbf{K}_F^{-1} \mathbf{R} \ddot{\mathbf{u}} \tag{3.70}$$

In turn, the substitution of Eq. (3.70) into Eq. (3.49) generates a fully uncoupled equation for the structural part of the system. This simplified structural equation of motion, based on the disregard of compressible fluid and surface gravity waves, is expressed in Eq. (3.71). It is interesting to note that the only difference between Eq. (3.71) and the isolated structural system [Eq. (3.49)] is the presence of the added mass matrix ($\widetilde{\mathbf{M}}_{\mathrm{F}}$), which expression is given by Eq. (3.72). The simplified Eq. (3.71) for incompressible fluids can be solved directly in time domain using direct integration or mode superposition methods, without the necessity of staggered solution procedures.

$$\left(\mathbf{M}_{\mathrm{S}} + \widetilde{\mathbf{M}}_{\mathrm{F}}\right)\ddot{\mathbf{u}} + \mathbf{C}_{\mathrm{S}}\dot{\mathbf{u}} + \mathbf{f}_{\mathrm{S}}^{\mathrm{int}}(\overline{\mathbf{u}}) = \mathbf{f}_{\mathrm{S}}^{\mathrm{ext}}(t) \tag{3.71}$$

$$\widetilde{\mathbf{M}}_{\mathrm{F}} = \rho_0 \mathbf{R}^{\mathbf{T}} \mathbf{K}_F^{-1} \mathbf{R} \tag{3.72}$$

It is not completely clear when it is recommendable to neglect compressibility effects and assume this simplified special case. One criterion for taking a decision is to estimate the order of magnitude of the fundamental compressible period of the structure in terms of the dam's height (H) and the fluid wave speed velocity (c), using the expression H/c. If this period is close to the magnitude of the earthquake acceleration period, then the compressibility effects play an important role and therefore should not be neglected. This is generally the case for high flexible dams containing a long range of modal vibration periods. On the other hand, compressibility can be less important in the case of low height stiffer dams with a compressible fundamental period far smaller than the excitation period.

[10] Based on TNO DIANA BV (2011) and Zienkiewicz et al. (2005).

3.1.9. Free vibration of FSI systems[11]

If a mode superposition method is used to solve Eqs. (3.65) or (3.71), then first it is required a free vibration eigen-analysis. Numerous procedures and assumptions can be followed in order to find the eigen-vectors (or mode shape vectors) and eigen-values (or modal frequencies) of the fluid-structure interaction system. Nevertheless, several authors, like Fenves and Chopra (1983) and Darbre (1996), coincide in disregarding the influence of the fluid interaction to determine the mode shapes of the structure.

It is understood that the shape modes of a structure, like an arch dam, is almost the same no matter if it is with a full or an empty reservoir. For this reason Eq. (3.50) is used for deriving the free vibration problem described by Eq. (3.73) and calculating the mode shape vectors of the structure in a fluid-structure interaction system.

$$\mathbf{M}_{\mathrm{S}}\ddot{\overline{\mathbf{u}}} + \mathbf{K}_{\mathrm{S}}\overline{\mathbf{u}} = \mathbf{0} \tag{3.73}$$

This approach, known as "dry dam" eigen-analysis, is accurate enough for the calculation of the mode shape vectors, but not for the calculation of the modal frequencies, which require the consideration of the fluid effects. One way of doing this is using the full coupled equation defined in Eq. (3.58) to obtain the free vibration problem expressed in Eq. (3.74).

$$\begin{bmatrix} \mathbf{M}_{\mathrm{S}} & \mathbf{0} \\ \rho_0\mathbf{R} & \mathbf{M}_{\mathrm{F}} \end{bmatrix}\begin{Bmatrix} \ddot{\overline{\mathbf{u}}} \\ \ddot{\overline{\mathbf{p}}} \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{\mathrm{S}} & -\mathbf{R}^{\mathrm{T}} \\ \mathbf{0} & \mathbf{K}_{\mathrm{F}} \end{bmatrix}\begin{Bmatrix} \overline{\mathbf{u}} \\ \overline{\mathbf{p}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \end{Bmatrix} \tag{3.74}$$

However, the matrices of Eq. (3.74) are neither symmetric nor positive definite, which make the classical computational methods for eigenvalue analysis not directly applicable. Some manipulations and variable symmetrizing procedures can be followed to solve the eigen-value problem defined by Eq. (3.74), yet making the process too complicated.

Another approach known as "wet dam" eigen-analysis consists in assuming that the interacting fluid is incompressible. Then, the simplified equation of motion given by Eq. (3.71) can be used to determine the modal frequencies of the fluid-structure interaction system through the solution of Eq. (3.75). The influence of the fluid is concentrated in the added mass term ($\widetilde{\mathbf{M}}_{\mathrm{F}}$) given by Eq. (3.72).

$$\left(\mathbf{M}_{\mathrm{S}} + \widetilde{\mathbf{M}}_{\mathrm{F}}\right)\ddot{\overline{\mathbf{u}}} + \mathbf{K}_{\mathrm{S}}\overline{\mathbf{u}} = \mathbf{0} \tag{3.75}$$

[11] Based TNO DIANA BV (2011), Zienkiewicz et al. (2005), Fenves and Chopra (1983) and Darbre (1996).

3.2. Modified equation of motion for base acceleration loading[12]

3.2.1. Partitioned discrete equation of motion

In earthquake analysis the input load is defined as a base acceleration which is multiplied by the mass of the structure, generating an inertial load applied through all the structure. Due to the prescribed movement applied to the base, the response of the other part of the structure (superstructure) can be classified as absolute or relative to the base motion.

Generally, earthquake analysis requires the relative displacements, velocities and accelerations for the design and evaluation of the structure. Therefore, in this section the classical equation of motion for seismic analysis, in terms of the relative displacement, velocity and acceleration vectors, is derived and the assumptions relying on this formulation are explained.

When an acceleration load is applied in the base of the structure, the dynamic equation of motion of this system can be partitioned into known DOF located in the base ($\mathbf{u}_\mathrm{b}$) (where accelerations are prescribed) and unknown DOF located in the superstructure ($\mathbf{u}_\mathrm{s}$). The partitioned equation of motion is expressed in Eq. (3.76), which right-hand side has no additional external loads in the superstructure, taking into account that the load is generated by the base motion ($\mathbf{u}_\mathrm{b}$), and that $\mathbf{f}_\mathrm{b}$ is the associated unknown reaction force vector in the base.

$$\begin{bmatrix}\mathbf{M}_{\mathrm{ss}} & \mathbf{M}_{\mathrm{sb}}\\ \mathbf{M}_{\mathrm{bs}} & \mathbf{M}_{\mathrm{bb}}\end{bmatrix}\begin{Bmatrix}\ddot{\mathbf{u}}_\mathrm{s}\\ \ddot{\mathbf{u}}_\mathrm{b}\end{Bmatrix}+\begin{bmatrix}\mathbf{C}_{\mathrm{ss}} & \mathbf{C}_{\mathrm{sb}}\\ \mathbf{C}_{\mathrm{bs}} & \mathbf{C}_{\mathrm{bb}}\end{bmatrix}\begin{Bmatrix}\dot{\mathbf{u}}_\mathrm{s}\\ \dot{\mathbf{u}}_\mathrm{b}\end{Bmatrix}+\begin{bmatrix}\mathbf{K}_{\mathrm{ss}} & \mathbf{K}_{\mathrm{sb}}\\ \mathbf{K}_{\mathrm{bs}} & \mathbf{K}_{\mathrm{bb}}\end{bmatrix}\begin{Bmatrix}\mathbf{u}_\mathrm{s}\\ \mathbf{u}_\mathrm{b}\end{Bmatrix}=\begin{Bmatrix}0\\ \mathbf{f}_\mathrm{b}\end{Bmatrix} \tag{3.76}$$

3.2.2. Superstructure absolute equation of motion

After taking from Eq. (3.76) the first partition belonging to the superstructure, the equation of motion in terms of the absolute response is defined by Eqs. (3.77) and (3.78).

$$\mathbf{M}_{\mathrm{ss}}\ddot{\mathbf{u}}_\mathrm{s}+\mathbf{C}_{\mathrm{ss}}\dot{\mathbf{u}}_\mathrm{s}+\mathbf{K}_{\mathrm{ss}}\mathbf{u}_\mathrm{s}=\mathbf{f}_\mathrm{s} \tag{3.77}$$

$$\mathbf{f}_\mathrm{s}=-\mathbf{M}_{\mathrm{sb}}\ddot{\mathbf{u}}_\mathrm{b}-\mathbf{C}_{\mathrm{sb}}\dot{\mathbf{u}}_\mathrm{b}-\mathbf{K}_{\mathrm{sb}}\mathbf{u}_\mathrm{b} \tag{3.78}$$

It can be noticed from Eq. (3.78) that the "external force" in the absolute equilibrium equation is produced by the base displacement, velocity and acceleration. However, there are two assumptions that simplify the force vector given by Eq. (3.78). The first assumption is that in the mass matrix there is no coupling between the base and superstructure DOF, and therefore Eq. (3.79) holds. The second assumption, expressed in Eq. (3.80), is to disregard the damping term contribution due to its not completely clear physical meaning (Wilson, 2002).

$$\mathbf{M}_{\mathrm{sb}}=\mathbf{0} \tag{3.79}$$

$$\mathbf{C}_{\mathrm{sb}}=\mathbf{0} \tag{3.80}$$

[12] Based TNO DIANA BV (2011) and Wilson (2002).

After replacing the assumptions expressed in Eqs. (3.79) and (3.80) into Eq. (3.78), the force vector of the absolute equilibrium equation is expressed by Eq. (3.81) in terms of base displacement only.

$$\mathbf{f}_{\mathrm{s}} = -\mathbf{K}_{\mathrm{sb}}\mathbf{u}_{\mathrm{b}} \tag{3.81}$$

3.2.3. Rigid base movement

If the base of the structure is rigid, then the displacements, velocities and accelerations are uniform through all the base points in the three directions of analysis. This uniform motion is defined by the ground displacement vector ($\mathbf{u}_{\mathrm{g}}$) as expressed in Eq. (3.82).

$$\mathbf{u}_{\mathrm{g}} = [u_{gx} \quad u_{gy} \quad u_{gz}]^{\mathrm{T}} \tag{3.82}$$

The matrices $\mathbf{I}^{(\mathrm{s})}$ and $\mathbf{I}^{(\mathrm{b})}$ represent the unitary rigid body displacements of the superstructure and the base, respectively, in the three translational directions (X, Y, Z). Eqs. (3.83) and (3.84) show that each of these rigid body motion matrices is composed by three vectors (one per translational direction) which values vary between 0 and 1, depending on the cosine direction of each DOF with respect to the XYZ coordinate system.

$$\mathbf{I}^{(\mathrm{s})} = [\mathbf{i}_{\mathrm{x}}^{(\mathrm{s})} \quad \mathbf{i}_{\mathrm{y}}^{(\mathrm{s})} \quad \mathbf{i}_{\mathrm{z}}^{(\mathrm{s})}] \tag{3.83}$$

$$\mathbf{I}^{(\mathrm{b})} = [\mathbf{i}_{\mathrm{x}}^{(\mathrm{b})} \quad \mathbf{i}_{\mathrm{y}}^{(\mathrm{b})} \quad \mathbf{i}_{\mathrm{z}}^{(\mathrm{b})}] \tag{3.84}$$

Eq. (3.85) shows the superstructure absolute displacement expressed in terms of the rigid body motion matrix, the ground displacement and the relative displacement ($\mathbf{u}_{\mathrm{r}}$). Similarly, Eq. (3.86) shows a similar expression for the base, with the obvious difference that in this case there is no relative displacement.

$$\mathbf{u}_{\mathrm{s}} = \mathbf{u}_{\mathrm{r}} + \mathbf{I}^{(\mathrm{s})}\mathbf{u}_{\mathrm{g}} \tag{3.85}$$

$$\mathbf{u}_{\mathrm{b}} = \mathbf{I}^{(\mathrm{b})}\mathbf{u}_{\mathrm{g}} \tag{3.86}$$

3.2.4. Superstructure relative equation of motion

The relative equation of motion is obtained by rewriting the absolute equation of the superstructure in terms of the relative displacements. Therefore, Eqs. (3.85) and (3.86) are replaced into Eqs. (3.77) and (3.78) to obtain the full relative equilibrium equation of motion given by Eq. (3.87).

$$\mathbf{M}_{\mathrm{ss}}(\ddot{\mathbf{u}}_{\mathrm{r}} + \mathbf{I}^{(\mathrm{s})}\ddot{\mathbf{u}}_{\mathrm{g}}) + \mathbf{C}_{\mathrm{ss}}\left(\dot{\mathbf{u}}_{\mathrm{r}} + \mathbf{I}^{(\mathrm{s})}\dot{\mathbf{u}}_{\mathrm{g}}\right) + \mathbf{K}_{\mathrm{ss}}(\mathbf{u}_{\mathrm{r}} + \mathbf{I}^{(\mathrm{s})}\mathbf{u}_{\mathrm{g}}) = -\mathbf{M}_{\mathrm{sb}}\mathbf{I}^{(\mathrm{b})}\ddot{\mathbf{u}}_{\mathrm{g}} - \mathbf{C}_{\mathrm{sb}}\mathbf{I}^{(\mathrm{b})}\dot{\mathbf{u}}_{\mathrm{g}} - \mathbf{K}_{\mathrm{sb}}\mathbf{I}^{(\mathrm{b})}\mathbf{u}_{\mathrm{g}} \tag{3.87}$$

Taking into account that internal forces are only proportional to relative displacements (stiffness force) and velocities (viscous damping force), the multiplication of the stiffness (**K**) and damping (**C**) matrices by the rigid body displacements ($\mathbf{I}$) is equal to zero[13], as expressed in Eqs. (3.88) and (3.89).

$$\mathbf{K}_{\mathrm{ss}}\mathbf{I}^{(\mathrm{s})} = \mathbf{K}_{\mathrm{sb}}\mathbf{I}^{(\mathrm{b})} = \mathbf{0} \tag{3.88}$$

$$\mathbf{C}_{\mathrm{ss}}\mathbf{I}^{(\mathrm{s})} = \mathbf{C}_{\mathrm{sb}}\mathbf{I}^{(\mathrm{b})} = \mathbf{0} \tag{3.89}$$

After inserting Eqs. (3.79), (3.88) and (3.89) into Eq. (3.87), the dynamic equilibrium equation given by Eq. (3.90) is obtained in terms of the superstructure displacements relative to the base motion ($\mathbf{u}_{\mathrm{r}}$). This approximate form of the relative equation of motion is the classic equation used in seismic analysis.

$$\mathbf{M}_{\mathrm{ss}}\ddot{\mathbf{u}}_{\mathrm{r}} + \mathbf{C}_{\mathrm{ss}}\dot{\mathbf{u}}_{\mathrm{r}} + \mathbf{K}_{\mathrm{ss}}\mathbf{u}_{\mathrm{r}} = -\mathbf{M}_{\mathrm{ss}}\mathbf{I}^{(\mathrm{s})}\ddot{\mathbf{u}}_{\mathrm{g}} \tag{3.90}$$

The base acceleration load can be expressed for each of the three translational directions as shown in Eq. (3.91), after replacing Eqs. (3.82) and (3.83) into Eq. (3.90).

$$\mathbf{M}_{\mathrm{ss}}\ddot{\mathbf{u}}_{\mathrm{r}} + \mathbf{C}_{\mathrm{ss}}\dot{\mathbf{u}}_{\mathrm{r}} + \mathbf{K}_{\mathrm{ss}}\mathbf{u}_{\mathrm{r}} = -\mathbf{M}_{\mathrm{ss}}(\ddot{u}_{gx}\mathbf{i}_{\mathrm{x}}^{(\mathrm{s})} + \ddot{u}_{gy}\mathbf{i}_{\mathrm{y}}^{(\mathrm{s})} + \ddot{u}_{gz}\mathbf{i}_{\mathrm{z}}^{(\mathrm{s})}) \tag{3.91}$$

It is interesting to notice that in the approximate absolute equation of motion [Eqs. (3.77) and (3.81)], the force vector is proportional to the prescribed base displacements, and it is applied only in the superstructure nodes close to the base ($-\mathbf{K}_{\mathrm{sb}}\mathbf{u}_{\mathrm{b}}$). On the other hand, the force vector of the approximate relative equation of motion [Eq. (3.90)] is proportional to the prescribed base accelerations, and it is applied in the superstructure nodes not linked with the base ($-\mathbf{M}_{\mathrm{ss}}\mathbf{I}^{(\mathrm{s})}\ddot{\mathbf{u}}_{\mathrm{g}}$).

[13] For the stiffness matrix this statement is always correct. However, in the case of the damping matrix the statement does not fulfill if Rayleigh damping definition (a**M**+b**K**) is used. The multiplication of the mass proportional term (a≠0) by the rigid body motion matrix results in a damping forces different than zero. Nevertheless, this artificial mass proportional damping energy added to the system is usually ignored (TNO DIANA BV, 2011).

4. HYBRID FREQUENCY-TIME DOMAIN (HFTD) METHOD SOLUTION PROCEDURE

4.1. Description of the HFTD method[1]

The HFTD method is generally applied in the solution of dynamic problems of systems with frequency dependent properties and non-linear behavior. Frequency dependent properties, which are present in soil-structure and fluid-structure interaction systems, are properly taken into account by frequency domain solution methods.

Nevertheless, this type of analysis methods is only applicable in linear systems. When some kinds of non-linearities (f.e.: material, geometrical, contact, friction, sliding) are required to be included in the analysis, frequency domain methods are not applicable anymore and a time-domain solution (generally by means of time-stepping algorithms) should be used.

Time domain analysis methods allow tracking the development of the non-linear response and evaluating its magnitude in each time step. However, it is not possible to consider frequency dependent properties directly and consistently.

It is under these circumstances that HFTD method is proposed as an alternative to study the non-linear dynamic response of frequency-dependent systems, by means of a hybrid formulation that combines the advantages of both, time and frequency domain analysis methods, and diminish their corresponding drawbacks.

In general terms, the HFTD method transforms the non-linear equation of motion defined in the time domain into a pseudo-linear equation of motion defined in the frequency domain. For this to be achieved, the internal force vector is divided into linear and non-linear components.

The linear component of the internal force is the summation of the linear elastic stiffness and damping forces. The non-linear component is an unknown vector denominated pseudo force. Besides, the linear component of the internal force is kept in the left hand side of the equation of motion, whereas the pseudo force vector is moved to the right hand side. In this way, the left-hand side is composed exclusively by linear terms, whereas the right-hand side contains the applied external forces and the pseudo force vector which unknown value is determined iteratively.

As mentioned before, the HFTD method solves iteratively the pseudo-linear (or linearized) equation of motion in the frequency domain. However, the dynamic external force applied to the system is defined by a signal in the time domain. In the same way, the pseudo force vector representing the nonlinear behavior of the system can only be evaluated in the time domain. Therefore, the HFTD method requires the transformation of both time dependent components of the right-hand side of the equation (time-dependent loading and pseudo force) into a series of harmonic forces by means of a Direct Fourier Transform (transformation to the frequency domain).

As a consequence the linearized equation of motion in the frequency domain can be solved for a range of loading frequencies, using the Direct or Mode superposition methods (see Chapter 2). The next step is to transform the solution obtained (displacements, velocities and accelerations) back to the time domain by means of the Inverse Fourier Transform.

[1] Based on Darbre (1996) and Fenves and Chavez (1990).

With the response in the time domain it is possible to use the constitutive laws ruling the non-linear behavior of the system and evaluate the internal forces. Then, a new value of the pseudo forces is computed as the difference between the linear elastic force and the nonlinear internal force.

Finally, the new value of the pseudo force vector is transformed again to the frequency domain to solve one more time the linearized equation in the frequency domain. The process is repeated until convergence is achieved.

4.2. Formulation of the HFTD method for the seismic analysis of a dam-reservoir dynamic interaction system[2]

A general non-linear equation of motion of the solid part of a fluid-structure interaction problem can be derived from Eq. (3.49). This equation of motion in absolute coordinates is shown in Eq. (4.1).

$$\mathbf{M}_{\mathrm{S}}\ddot{\mathbf{u}} + \mathbf{f}_{\mathrm{S}}^{\mathrm{int}}(t, \mathbf{u}, \dot{\mathbf{u}}) - \mathbf{R}^{\mathrm{T}}\mathbf{p} = \mathbf{f}_{\mathrm{S}}^{\mathrm{ext}}(t) \tag{4.1}$$

For the particular case of earthquake analysis of dam-reservoir interaction systems, the seismic load can be applied as a base acceleration loading in the three translational directions, defined by the ground acceleration vector ($\ddot{\mathbf{u}}_{\mathrm{g}}$). This approach is conveniently formulated in relative coordinates without necessity of the application of an external force ($\mathbf{f}_{\mathrm{S}}^{\mathrm{ext}}(t) = \mathbf{0}$). Based on Eqs. (3.90) and (4.1), the non-linear equation of motion in relative coordinates for the seismic analysis of a dam-reservoir interaction system is given by Eq. (4.2).

$$\mathbf{M}_{\mathrm{S}}\ddot{\mathbf{u}}_{\mathrm{r}} + \mathbf{f}_{\mathrm{S}}^{\mathrm{int}}(t, \mathbf{u}_{\mathrm{r}}, \dot{\mathbf{u}}_{\mathrm{r}}) = -\mathbf{M}_{\mathrm{S}}\mathbf{I}^{(\mathrm{s})}\ddot{\mathbf{u}}_{\mathrm{g}}(t) + \mathbf{R}^{\mathrm{T}}\mathbf{p} \tag{4.2}$$

In the HFTD method, the internal force ($\mathbf{f}_{\mathrm{S}}^{\mathrm{int}}$) of the dam (structural part of the coupled system) is decomposed into a linear and a nonlinear components, as expressed in Eq. (4.3). The linear component is expressed in terms of the initial tangent stiffness and damping matrices of the dam ($\mathbf{K}_{\mathrm{S}}, \mathbf{C}_{\mathrm{S}}$), and the displacement and velocity vectors relative to the base ($\mathbf{u}_{\mathrm{r}}, \dot{\mathbf{u}}_{\mathrm{r}}$). On the other hand, the nonlinear component is treated as a general remaining force depending on time ($\mathbf{q}$).

$$\mathbf{f}_{\mathrm{S}}^{\mathrm{int}}(t, \mathbf{u}_{\mathrm{r}}, \dot{\mathbf{u}}_{\mathrm{r}}) = \mathbf{K}_{\mathrm{S}}\mathbf{u}_{\mathrm{r}} + \mathbf{C}_{\mathrm{S}}\dot{\mathbf{u}}_{\mathrm{r}} - \mathbf{q}(t) \tag{4.3}$$

After replacing Eq. (4.3) into Eq. (4.2), and bringing the nonlinear component ($\mathbf{q}$) to the right-hand side of the equation, a pseudo linear system described by Eq. (4.4) is obtained. The left-hand side of Eq. (4.4) is completely defined by linear terms, whereas the nonlinearities are concentrated in the pseudo force vector ($\mathbf{q}$) located in the right-hand side.

$$\mathbf{M}_{\mathrm{S}}\ddot{\mathbf{u}}_{\mathrm{r}} + \mathbf{C}_{\mathrm{S}}\dot{\mathbf{u}}_{\mathrm{r}} + \mathbf{K}_{\mathrm{S}}\mathbf{u}_{\mathrm{r}} = -\mathbf{M}_{\mathrm{S}}\mathbf{I}^{(\mathrm{s})}\ddot{\mathbf{u}}_{\mathrm{g}}(t) + \mathbf{R}^{\mathrm{T}}\mathbf{p} + \mathbf{q}(t) \tag{4.4}$$

[2] Based on Darbre (1996), Fenves and Chavez (1990), Mansur et al. (2000) and Aprile et al. (1994).

The most consistent way of taking into account the frequency-dependent properties of the reservoir's compressible fluid, defined by the hydrodynamic force vector ($\mathbf{R}^{\mathrm{T}}\mathbf{p}$), is by means of solving the pseudo linear system described by Eq. (4.4) in the frequency domain. By doing this, in the more general case, frequency-dependent damping can also be included in the system as shown by Eqs. (4.5) and (4.6). $\mathbf{K}_0$ and $\mathbf{C}_0$ indicate the constant stiffness and damping matrices, respectively, whereas $\mathbf{C}_{fd}(\omega)$ is the frequency dependent damping matrix, $\mathbf{F}(\omega)$ is a general frequency dependent matrix, and ξ_H is the hysteretic damping ratio.

$$\mathbf{C}_{\mathrm{S}} = \mathbf{C}_0 + \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbf{C}_{fd}(\omega) e^{i(\omega t)} d\omega \tag{4.5}$$

$$\mathbf{C}_{fd}(\omega) = \mathbf{F}(\omega) + i\xi_H \mathbf{K}_0 \tag{4.6}$$

In order to solve Eq. (4.4) in the frequency domain, the excitation force given by the ground acceleration vector is expressed in the harmonic form shown in Eq. (4.7). Consequently, the harmonic form of the relative displacement, the hydrodynamic pressure and the pseudo force vectors are expressed in Eqs. (4.8) to (4.10), respectively.

$$\ddot{\mathbf{u}}_{\mathrm{g}}(t) = \hat{\ddot{\mathbf{u}}}_{\mathrm{g}}(\omega)\mathrm{e}^{i\omega t} \tag{4.7}$$

$$\mathbf{u}_{\mathrm{r}} = \hat{\mathbf{u}}_{\mathrm{r}}(\omega)\mathrm{e}^{i\omega t} \tag{4.8}$$

$$\mathbf{p} = \hat{\mathbf{p}}(\omega)\mathrm{e}^{i\omega t} \tag{4.9}$$

$$\mathbf{q}(t) = \hat{\mathbf{q}}(\omega)\mathrm{e}^{i\omega t} \tag{4.10}$$

After replacing Eqs. (4.7) to (4.10) into Eq. (4.4), the exponential common terms are eliminated, and as a result, the equation of motion of the dam-reservoir system in the frequency domain is defined by Eq. (4.11) in terms of the complex-valued amplitudes of the ground acceleration [$\hat{\ddot{\mathbf{u}}}_{\mathrm{g}}(\omega)$], the relative dam's displacement [$\hat{\mathbf{u}}(\omega)$], the hydrodynamic pressure [$\hat{\mathbf{p}}(\omega)$] and the pseudo force vector [$\hat{\mathbf{q}}(\omega)$].

$$[-\omega^2\mathbf{M}_{\mathrm{S}} + i\omega\mathbf{C}_{\mathrm{S}} + \mathbf{K}_{\mathrm{S}}]\hat{\mathbf{u}}_{\mathrm{r}}(\omega) = -\mathbf{M}_{\mathrm{S}}\mathbf{I}^{(\mathrm{s})}\hat{\ddot{\mathbf{u}}}_{\mathrm{g}}(\omega) + \mathbf{R}^{\mathrm{T}}\hat{\mathbf{p}}(\omega) + \hat{\mathbf{q}}(\omega) \tag{4.11}$$

Based on Eq. (3.52), the complex-valued frequency dependent expression of the hydrodynamic pressure amplitude is determined from the fluid equation of motion given by Eq. (4.12).

$$\mathbf{M}_{\mathrm{F}}\ddot{\mathbf{p}} + \mathbf{C}_{\mathrm{F}}\dot{\mathbf{p}} + \mathbf{K}_{\mathrm{F}}\mathbf{p} + \rho_0\mathbf{R}\ddot{\mathbf{u}} = \mathbf{0} \tag{4.12}$$

Eq. (4.12) is transformed into Eq. (4.13) after expressing the dam absolute acceleration vector ($\ddot{\mathbf{u}}$) in terms of the ground ($\ddot{\mathbf{u}}_{\mathrm{g}}$) and relative ($\ddot{\mathbf{u}}_{\mathrm{r}}$) acceleration vectors.

$$\mathbf{M}_{\mathrm{F}}\ddot{\mathbf{p}} + \mathbf{C}_{\mathrm{F}}\dot{\mathbf{p}} + \mathbf{K}_{\mathrm{F}}\mathbf{p} + \rho_0\mathbf{R}(\mathbf{I}^{(\mathrm{s})}\ddot{\mathbf{u}}_{\mathrm{g}} + \ddot{\mathbf{u}}_{\mathrm{r}}) = \mathbf{0} \tag{4.13}$$

After replacing Eqs. (4.7) to (4.9) into Eq. (4.13), the complex-valued frequency dependent expression of the hydrodynamic pressure amplitude is defined by Eqs. (4.14) and (4.15).

$$\widehat{\mathbf{p}}(\omega) = -\rho_0 \mathbf{H}_\mathrm{F} \mathbf{R} \left[\mathbf{I}^{(\mathrm{s})} \widehat{\ddot{\mathbf{u}}}_\mathrm{g}(\omega) - \omega^2 \widehat{\mathbf{u}}_\mathrm{r}(\omega) \right] \tag{4.14}$$

$$\mathbf{H}_\mathrm{F}(\omega) = \left[-\omega^2 \mathbf{M}_\mathrm{F} + i\omega \mathbf{C}_\mathrm{F} + \mathbf{K}_\mathrm{F} \right]^{-1} \tag{4.15}$$

Eq. (4.16) is obtained by inserting the expression of the hydrodynamic pressure amplitude given by Eq. (4.14) into the equation of motion of the coupled dam-reservoir system defined by Eq. (4.11). The hydrodynamic mass matrix ($\mathbf{L}_\mathrm{F}$) that appears in both sides of Eq. (4.16) is a complex-valued frequency-dependent term defined by Eq. (4.17), which represents the "added mass" effect in the dam provided by the reservoir.

$$\left[-\omega^2 (\mathbf{M}_\mathrm{S} + \mathbf{L}_\mathrm{F}) + i\omega \mathbf{C}_\mathrm{S} + \mathbf{K}_\mathrm{S} \right] \widehat{\mathbf{u}}_\mathrm{r}(\omega) = -[\mathbf{M}_\mathrm{S} + \mathbf{L}_\mathrm{F}] \mathbf{I}^{(\mathrm{s})} \widehat{\ddot{\mathbf{u}}}_\mathrm{g}(\omega) + \widehat{\mathbf{q}}(\omega) \tag{4.16}$$

$$\mathbf{L}_\mathrm{F}(\omega) = \rho_0 \mathbf{R}^\mathrm{T} \mathbf{H}_\mathrm{F} \mathbf{R} \tag{4.17}$$

Another possibility of expressing the effect of the reservoir on the dam is to split $\mathbf{L}_\mathrm{F}$ in its real ("added mass") and imaginary ("added damping") components, as shown in Eq. (4.18). The expressions for the added mass ($\widetilde{\mathbf{M}}_\mathrm{F}$) and damping ($\tilde{\mathbf{C}}_\mathrm{F}$) are given by Eqs. (4.19) and (4.20), respectively.

$$\left[-\omega^2 \left(\mathbf{M}_\mathrm{S} + \widetilde{\mathbf{M}}_\mathrm{F} \right) + i\omega \left(\mathbf{C}_\mathrm{S} + \tilde{\mathbf{C}}_\mathrm{F} \right) + \mathbf{K}_\mathrm{S} \right] \widehat{\mathbf{u}}_\mathrm{r}(\omega) = -\left[\mathbf{M}_\mathrm{S} + \widetilde{\mathbf{M}}_\mathrm{F} - i\frac{1}{\omega} \tilde{\mathbf{C}}_\mathrm{F} \right] \mathbf{I}^{(\mathrm{s})} \widehat{\ddot{\mathbf{u}}}_\mathrm{g}(\omega) + \widehat{\mathbf{q}}(\omega) \tag{4.18}$$

$$\widetilde{\mathbf{M}}_\mathrm{F} = \Re(\mathbf{L}_\mathrm{F}) \tag{4.19}$$

$$\tilde{\mathbf{C}}_\mathrm{F} = -\omega \Im(\mathbf{L}_\mathrm{F}) \tag{4.20}$$

The mode superposition method can be employed by introducing the variable defined by Eq. (4.21). Consequently, the pre-multiplication of Eq. (4.18) by the matrix of modal vectors transposed ($\mathbf{\Phi}^\mathrm{T}$) results in the equation of motion, defined by Eqs. (4.22) to (4.28), in terms of the generalized modal coordinates vector ($\hat{\mathbf{y}}$).

$$\widehat{\mathbf{u}}_\mathrm{r}(\omega) = \mathbf{\Phi} \hat{\mathbf{y}}(\omega) \tag{4.21}$$

$$\left[-\omega^2 (\mathbf{B}_\mathrm{S} + \mathbf{B}_\mathrm{F}) + i\omega (\mathbf{D}_\mathrm{S} + \mathbf{D}_\mathrm{F}) + \mathbf{E}_\mathrm{S} \right] \hat{\mathbf{y}}(\omega) = -\mathbf{\Phi}^\mathrm{T} \mathbf{F}_\mathrm{g} \mathbf{I}^{(\mathrm{s})} \widehat{\ddot{\mathbf{u}}}_\mathrm{g}(\omega) + \mathbf{\Phi}^\mathrm{T} \widehat{\mathbf{q}}(\omega) \tag{4.22}$$

$$\mathbf{B}_\mathrm{S} = \mathbf{\Phi}^\mathrm{T} \mathbf{M}_\mathrm{S} \mathbf{\Phi} \tag{4.23}$$

$$\mathbf{B}_\mathrm{F}(\omega) = \mathbf{\Phi}^\mathrm{T} \widetilde{\mathbf{M}}_\mathrm{F} \mathbf{\Phi} \tag{4.24}$$

$$\mathbf{D}_\mathrm{S}(\omega) = \mathbf{\Phi}^\mathrm{T} \mathbf{C}_\mathrm{S} \mathbf{\Phi} \tag{4.25}$$

$$\mathbf{D}_{\mathrm{F}}(\omega) = \mathbf{\Phi}^{\mathrm{T}}\tilde{\mathbf{C}}_{\mathrm{F}}\mathbf{\Phi} \tag{4.26}$$

$$\mathbf{E}_{\mathrm{S}} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{K}_{\mathrm{S}}\mathbf{\Phi} \tag{4.27}$$

$$\mathbf{F}_{\mathrm{g}}(\omega) = \mathbf{M}_{\mathrm{S}} + \widetilde{\mathbf{M}}_{\mathrm{F}} - i\frac{1}{\omega}\tilde{\mathbf{C}}_{\mathrm{F}} \tag{4.28}$$

The equation of motion in the generalized modal coordinates constitutes a condensed system of equations in which the unknowns are reduced to a number of chosen modes of vibration. In the general case only the frequency independent matrices $\mathbf{B}_{\mathrm{S}}$ and $\mathbf{E}_{\mathrm{S}}$ are diagonal, which means that the system is not uncoupled.

One way of uncouple numerically Eq. (4.22) is to express the full matrices $\mathbf{B}_{\mathrm{F}}$, $\mathbf{D}_{\mathrm{S}}$ and $\mathbf{D}_{\mathrm{F}}$ as the summation of a diagonal (d) and an off-diagonal (od) terms as shown in Eqs. (4.29) to (4.31).

$$\mathbf{B}_{\mathrm{F}} = \mathbf{B}_{\mathrm{F}}^{d} + \mathbf{B}_{\mathrm{F}}^{od} \tag{4.29}$$

$$\mathbf{D}_{\mathrm{S}} = \mathbf{D}_{\mathrm{S}}^{d} + \mathbf{D}_{\mathrm{S}}^{od} \tag{4.30}$$

$$\mathbf{D}_{\mathrm{F}} = \mathbf{D}_{\mathrm{F}}^{d} + \mathbf{D}_{\mathrm{F}}^{od} \tag{4.31}$$

Replacing Eqs. (4.29) to (4.31) into Eq. (4.22), and bringing the off-diagonal terms to the right-hand side, results in an uncoupled system of equations defined by Eq. (4.32) which matrices in the left-hand side are diagonal. The uncoupled system can be solved iteratively by treating the off-diagonal terms as a second set of pseudo forces dependent of the generalized modal coordinates vector and defined by Eq. (4.33).

$$\left[-\omega^2\left(\mathbf{B}_{\mathrm{S}} + \mathbf{B}_{\mathrm{F}}^{d}\right) + i\omega\left(\mathbf{D}_{\mathrm{S}}^{d} + \mathbf{D}_{\mathrm{F}}^{d}\right) + \mathbf{E}_{\mathrm{S}}\right]\hat{\mathbf{y}}^{j+1}(\omega) = -\mathbf{\Phi}^{\mathrm{T}}\mathbf{F}_{\mathrm{g}}\mathbf{I}^{(\mathrm{s})}\hat{\ddot{\mathbf{u}}}_{\mathrm{g}}(\omega) - \mathbf{Q}^{od}\hat{\mathbf{y}}^{j}(\omega) + \mathbf{\Phi}^{\mathrm{T}}\hat{\mathbf{q}}^{j}(\omega) \tag{4.32}$$

$$\mathbf{Q}^{od}(\omega) = -\omega^2\mathbf{B}_{\mathrm{F}}^{od} + i\omega\left(\mathbf{D}_{\mathrm{S}}^{od} + \mathbf{D}_{\mathrm{F}}^{od}\right) \tag{4.33}$$

Nevertheless, the introduction of additional pseudo-force vectors to impose the uncoupling of the system of equations could lead to a cumbersome numerical process, especially in large models with high amount of DOF.

4.3. Implementation issues[3]

4.3.1. Solution procedure sequence of the HFTD method

The HFTD method can be summarized in the following general steps, graphically schematized in the flowchart of Fig. 4.1.

1. Select the number of modes to be included in the response calculation.
2. Do the eigenvalue analysis of the dam with empty reservoir and calculate the generalized mass ($\mathbf{B}_S$), damping ($\mathbf{D}_S$) and stiffness ($\mathbf{E}_S$) matrices.
3. Determine the range of excitation frequencies to be used in the solution of the linearized equation [Eq. (4.22)].
4. Determine for each excitation frequency the generalized added mass ($\mathbf{B}_F$), added damping ($\mathbf{D}_F$) and load ($\mathbf{F}_g$). If applicable, the frequency dependent part of the generalized structural damping matrix ($\mathbf{D}_S$) should be calculated for each excitation frequency.
5. Initialize the pseudo force vector in time domain $[\mathbf{q}(t)]$. Only for the first iteration, the pseudo force vector should be set equal to zero.
6. Perform the Fourier transformation of the ground acceleration signal from the time domain $[\ddot{\mathbf{u}}_g(t)]$ to the frequency domain $[\widehat{\ddot{\mathbf{u}}}_g(\omega)]$.
7. Set up the current generalized pseudo force vector in the time domain $[\mathbf{\Phi}^T\mathbf{q}(t)]$.
8. Perform the Fourier transform of the generalized pseudo force vector to the frequency domain $[\mathbf{\Phi}^T\hat{\mathbf{q}}(\omega)]$.
9. Solve the linearized Eq. (4.22) for each excitation frequency and determine the generalized displacement in the frequency domain $[\hat{\mathbf{y}}(\omega)]$.
10. Perform the inverse Fourier transform of the generalized displacement ($\mathbf{y}$), velocity ($\dot{\mathbf{y}}$) and acceleration ($\ddot{\mathbf{y}}$) to the time domain.
11. Expand the generalized response in the time domain to the natural system of coordinates ($\mathbf{u}_r$, $\dot{\mathbf{u}}_r$ and $\ddot{\mathbf{u}}_r$) using Eq. (4.21).
12. Perform the state determination in the time domain: Calculation of the nonlinear internal force of the system using the appropriate constitutive law and the response obtained in step 11.
13. Determine the linear restoring force ($\mathbf{K}_S\mathbf{u}_r + \mathbf{C}_S\dot{\mathbf{u}}_r$) based on the response calculated in step 11.
14. Evaluate the new pseudo force vector using Eq. (4.3) and the results obtained in steps 12 and 13.
15. Check the convergence of the displacement and pseudo force. If convergence is not achieved, go to step 7 and perform a new iteration.

[3] Based on Darbre and Wolf (1988), Fenves and Chavez (1990), Darbre (1990), Chavez and Fenves (1993), Chavez and Fenves (1994) and Veletsos and Ventura (1985).

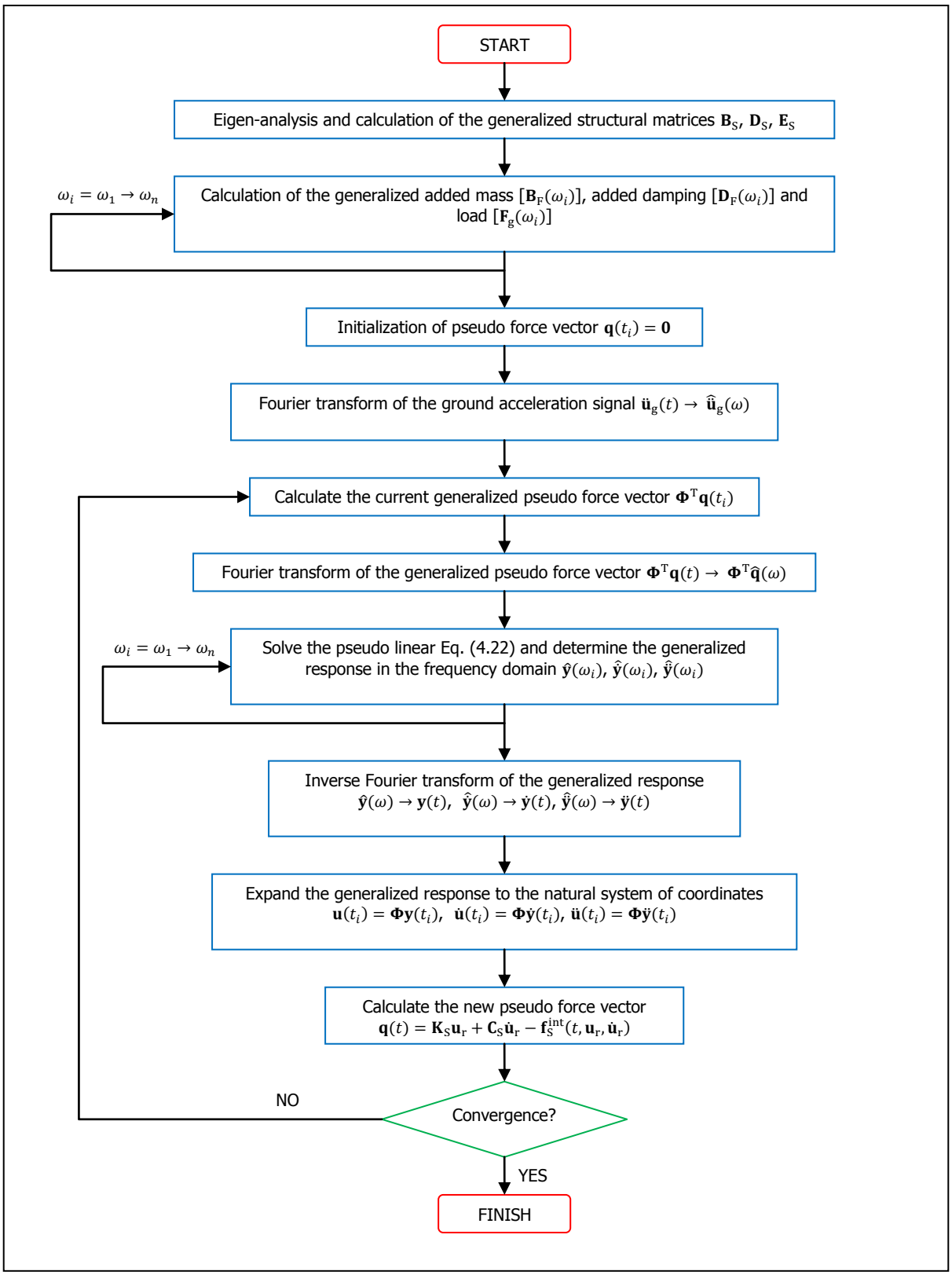


Figure 4.1: Flowchart of the HFTD method general solution procedure

4.3.2. HFTD method time segmentation approach

The HFTD procedure converges in a time progressive manner, which means that "convergence at any time is reached only after the solution has converged at all previous times" (Darbre and Wolf, 1988). If the procedure described above is executed directly along the whole time span of interest, convergence and stability problems are likely to occur.

For this reason, the HFTD method is performed following a time segmentation approach in which the displacements and pseudo forces are evaluated for a limited number of consecutive time steps contained in time segments. Once convergence is achieved in all the time steps within a segment, the iterative procedure is repeated for the next time segment, until the total time span of interest is covered.

The minimum number of time segments required depends on stability conditions[4]. When a system is highly stable, the number of time segments and iterations required to obtain convergence diminish. Nevertheless, the minimum number of iterations required to obtain convergence is more sensitive to the degree of stability than the minimum number of time segments. This means that two different systems can converge with the same number of time segments, but the system which is less stable will always require a higher number of iterations.

In conclusion, the time span of interest has to be divided into time segments containing one or several time steps. It is important to mention that the length of time segments does not affect the accuracy of the results obtained but leads to convergent (small length) or divergent (very large length) solution. When stability or convergence problems occur, then the number of time segments should be increased by reducing their size (less number of time steps contained in one segment). The extreme case is given by segments consisting in just one time step.

On the other hand, the number of iterations and the computational effort are proportional to the number of segments used. If many segments consisting of a few time step each are defined for the iterative procedure, then the computational effort and the total number of iterations required to achieve convergence is high. However, this proportional dependence is not critic unless a considerably higher number of segments than the optimal are used.

4.3.3. Fourier transform period

The period of time used in the Discrete Fourier Transform (Section 2.4) indicates the period of the periodic loading function (ground acceleration or pseudo force) acting on the system. Therefore, if the loading (earthquake) duration (t_0) is used to transform the ground acceleration and pseudo force to the frequency domain, the results obtained from the HFTD method would be the steady-state response of the system subject to a periodic loading with a period equal to t_o, which harmonic components are given by its Fourier transform.

The purpose of the HFTD method is to determine the actual transient response of the system subject to an arbitrary ground acceleration loading, and not the steady-state response of the system subject to a periodic loading. Therefore, to obtain an accurate approximation of the transient response, the period used in the Fourier transform should be extended sufficiently long that the effects of the steady-state response and the loading periodicity become negligible (Veletsos and Ventura, 1985).

[4] For a detailed discussion of HFTD method stability criteria see Darbre and Wolf (1988).

The Fourier period (T_p) should be defined as the summation of the ground acceleration duration (t_o) plus an additional extended period called quiet zone (T_q). The quiet zone is a band of zeros added at the end of the ground acceleration signal, which length depends on the free vibration properties of the system (fundamental period of vibration and viscous damping ratio of the structure).

Steps 6, 8 and 10 of the HFTD solution procedure outlined in Section 4.3.1 indicate that the Fourier transform is applied to the earthquake ground acceleration (direct), the pseudo force (direct) and the response of the system (inverse), respectively. All these transformations should always be performed over the same Fourier period (T_p), which is calculated using Eqs. (4.36) and (4.37), in terms of the fundamental vibration period (T_1) and the viscous damping ratio of the system (ξ) (Chavez and Fenves, 1984).

$$T_p = t_0 + T_q \tag{4.36}$$

$$T_q \geq 0.75 \frac{T_1}{\xi} \tag{4.37}$$

Eq. (4.37) shows that for long oscillation periods (low frequency systems) and low damping ratio, the quiet zone (T_q) is longer. In the extreme case of a damping ratio equal to zero T_q tends to infinity, which means that the HFTD method is not applicable for undamped systems (Veletsos and Ventura, 1985).

As mentioned before, the quiet zone is necessary to satisfy the initial conditions and consequently obtain a better approximation of the transient response instead of the steady state response. On the contrary, "the use of Fourier transformation period which is too short provides inaccurate results, no matter that they be stable and apparently reasonable" (Darbre and Wolf, 1988). In conclusion, the insertion of a quiet zone should be understood as a numerical representation of what theoretically should be an infinity period in the analytic Fourier integral (See Section 2.4).

Fig. 4.2 shows a ground acceleration signal which Fourier Transform is calculated twice, one using a Fourier period equal to the ground acceleration duration ($T_p = t_0 = 4.52\ s$), and the other using an extended Fourier period that includes the quiet zone ($T_p = t_0 + T_q = 45.2\ s$).

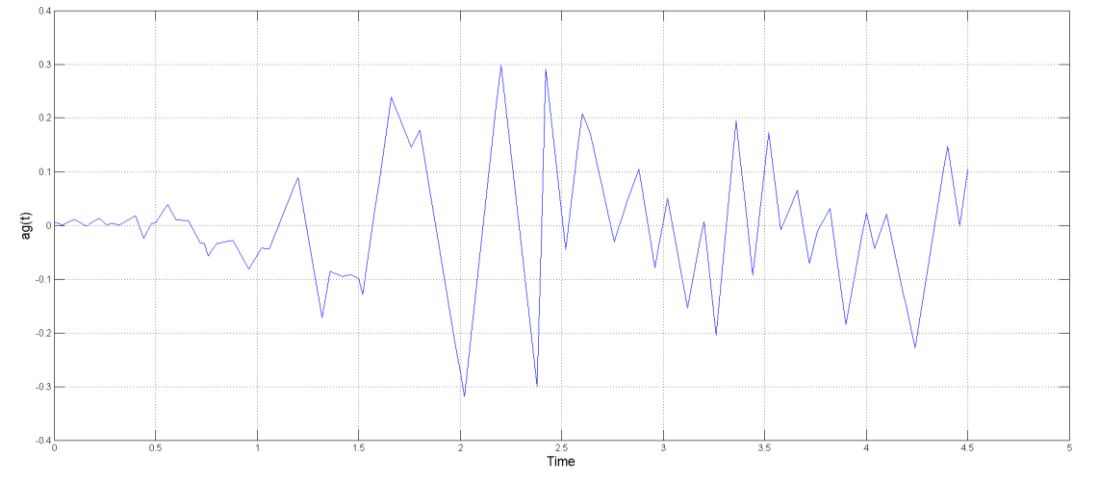


Figure 4.2: El Centro N-S earthquake ground acceleration signal.

The spectrums of the ground acceleration amplitudes obtained in both cases are shown in Figs. 4.3 and 4.4, respectively. It can be noticed that both spectrums have similar shapes, with the difference that the one obtained with the extended Fourier period (Fig. 4.4) is smoother due the higher number of points used in its definition.

The extended period is ten times the loading duration, and therefore, the number of points used in both cases holds the same relation. This is the reason why the magnitudes of the ground acceleration amplitudes of Fig. 4.4 are around ten times the amplitudes of Fig. 4.3.

In order to obtain the transient response of the system, the HFTD method uses the ground acceleration spectrum shown in Fig. 4.4.

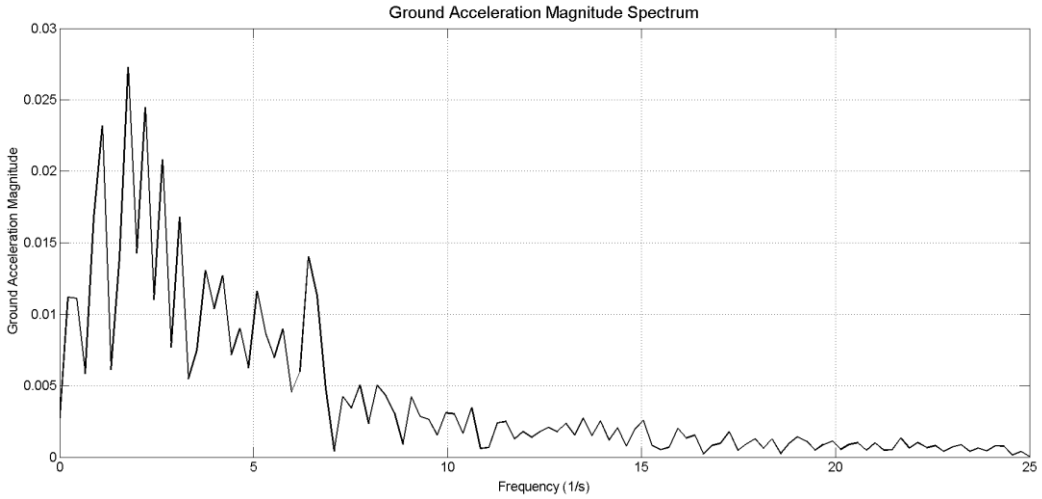


Figure 4.3: Spectrum of the ground acceleration amplitude for Fourier period equal to the loading duration ($T_p = t_0 = 4.52\ s$)

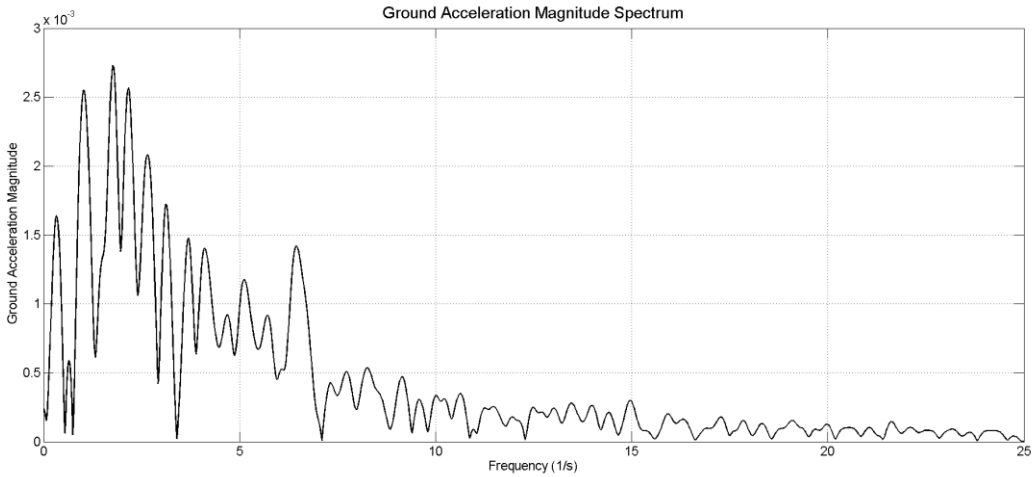


Figure 4.4: Spectrum of the ground acceleration amplitude for extended Fourier period including quiet zone ($T_p = t_0 + T_q = 45.2\ s$)

4.3.4. Details of the HFTD method solution procedure using the time segmentation approach

Using a time segmentation approach implies the repetition of the HFTD solution outlined in Section 4.3.1 (from step 6 to step 15) for all the segments defined in the time span of interest. It is important to realize that the time span of interest (T_r) is not the same as the Fourier period (T_p), but the time for which the calculation of the system response is desired.

Logically, the time span of interest (T_r) is shorter than the period used for the Fourier transform (T_p). As shown in Eq. (4.38), the minimum value of T_r includes the ground acceleration duration (t_o) plus and additional free vibration time equal to one half of the fundamental period of vibration, during which a maximum peak response could still be attained. The time segmentation should be performed only over T_r, and the HFTD solution procedure is concluded when the last segment of T_r has converged.

$$T_r \geq t_o + \frac{1}{2}T_1 \qquad \textbf{(4.38)}$$

Fig. 4.5 shows the k^{th} time segment and the time steps contained inside. The initial points of time segments "k" (T_k) and "k+1" (T_{k+1}), the last converged time step (t_c) of the previous iteration, the time step size (Δt), and the time segment size (ΔT_s) can also be identified from Fig. 4.5. The initial time step of segment "k" is calculated with Eq. (4.39).

$$T_k = (k-1)\Delta T_s \qquad \textbf{(4.39)}$$

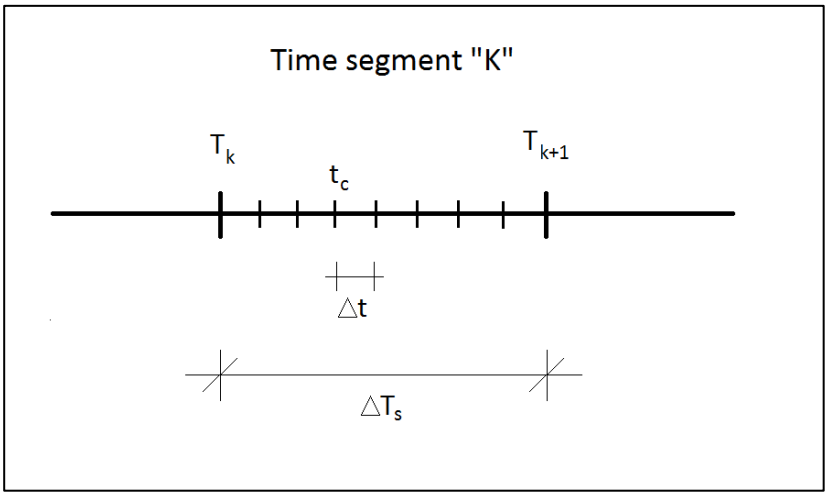


Figure 4.5: Graphic representation of the k^{th} time segment.

The HFTD procedure outlined in Section 4.3.1 indicates that, for the first iteration of the first time segment only, the pseudo force vector is set equal to zero (step 5) and transformed to the frequency domain (step 8) using the Fourier transformation period (T_p). Once the pseudo linear system has been solved in the frequency domain (step 9), the generalized response is transformed back to the time domain (step 10).

For the subsequent iterations, the segmentation approach requires that the pseudo forces and displacements in the time domain are calculated and updated (step 11 to step 14) only for the not converged time steps belonging to the current time segment ($t_c < t \leq T_{k+1} - \Delta t$). The iterative process (step 7 to step 14) is repeated until the convergence criteria (step 15) are fulfilled for all the time steps inside the current segment of analysis ($t_c = T_{k+1} - \Delta t$).

When all the time steps inside the current segment have converged ($t_c = T_{k+1} - \Delta t$), the next segment is investigated. However, in the first iteration of the new segment (step 6) the values of the pseudo forces belonging to the previous converged time segments are maintained (the pseudo force values of the converged previous time steps are not modified nor updated anymore), whereas for the current and subsequent segments, the values of the pseudo forces are set to zero (the definition of a decaying appended function is also possible as explained in the next section).

In this way, the iterative procedure is continued until the last segment of the time span of interest is covered, doing always the Fourier transformation through all the period T_p, but updating progressively the displacement and pseudo force vectors only for the non-converged time steps inside the current segment of analysis. The modified version of the HFTD method flowchart, including time segmentation approach in the solution procedure, is shown in Fig. 4.6

As a final point, special attention should be given to the fact that the equation of motion of the pseudo linear system (not the real non-linear system) is the one solved in the frequency domain. Therefore, the eigenvalues, eigenvectors and critical damping ratios, required to perform the mode superposition method and to determine the length of the Fourier transformation period (T_p) [Eqs. (4.36) and (4.37)], should be based on the properties of the pseudo linear system defined in Eq. (4.22).

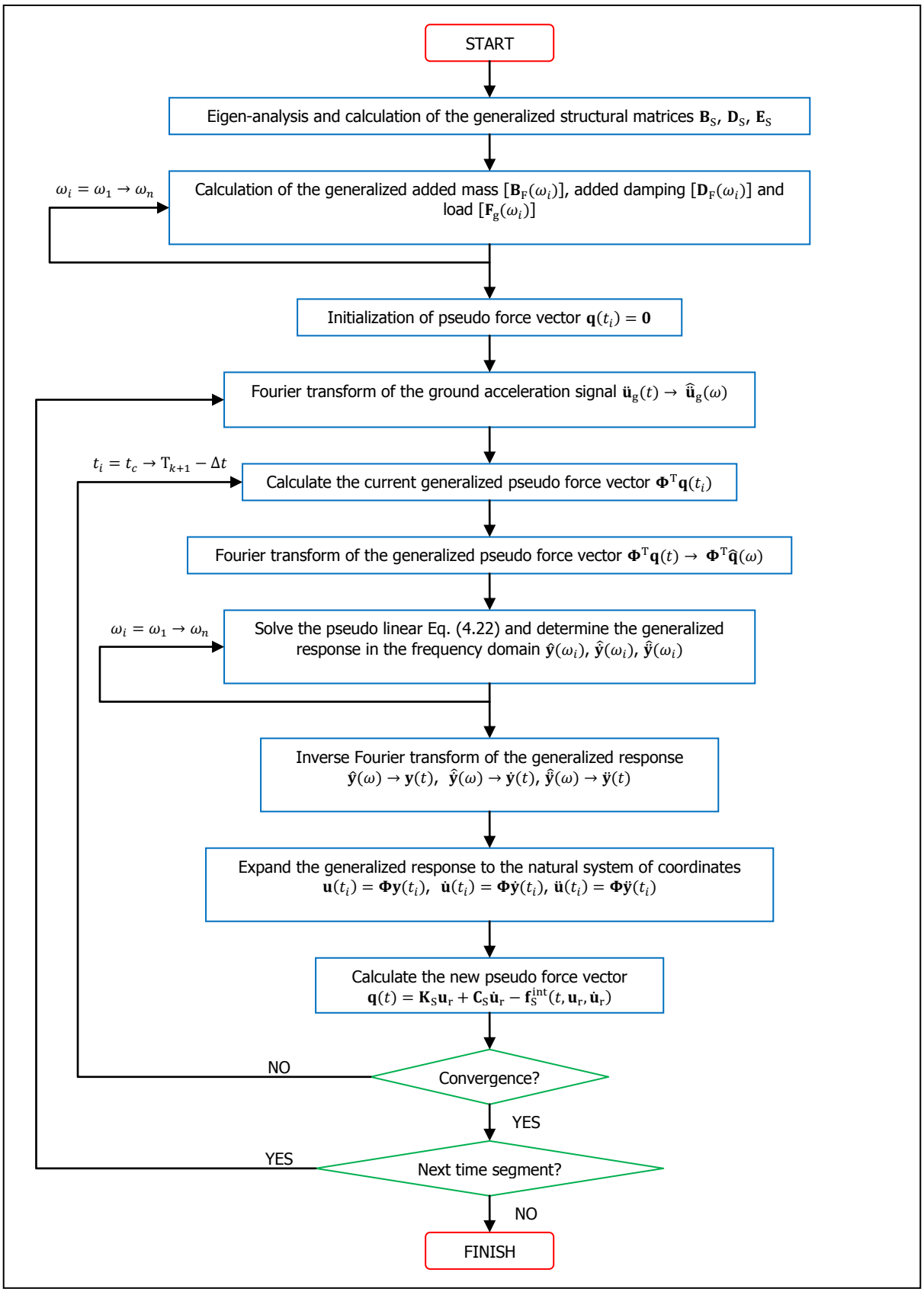


Figure 4.6: Flowchart of the modified HFTD method solution procedure including the time segmentation approach

4.3.5. Definition of the appended decaying functions for the ground acceleration loading and the pseudo force

Due to the segmentation approach for solving the linearized equation in the frequency domain, both load vectors in the right-hand side of Eq. (4.22), ground acceleration $[\ddot{\mathbf{u}}_{\mathrm{g}}(t)]$ and pseudo force $[\mathbf{q}(t)]$, are filled with zeros for all the time steps belonging to the time segments coming after the current. Therefore, the expressions defining $\ddot{\mathbf{u}}_{\mathrm{g}}(t)$ and $\mathbf{q}(t)$, for the time segment "k" and iteration "j", are given by Eqs. (4.40) and (4.41), respectively, in terms of the Heaviside function (H).

$$\ddot{\mathbf{u}}_{\mathrm{g}}{}^{k}(t) = \ddot{\mathbf{u}}_{\mathrm{g}}(t)[H(t) - H(t - \mathrm{T}_{k+1})], \qquad 0 < t < \mathrm{T}_p \tag{4.40}$$

$$\mathbf{q}^{k,j}(t) = \mathbf{q}^{k,j-1}(t)[H(t) - H(t - \mathrm{T}_{k+1})], \qquad 0 < t < \mathrm{T}_p \tag{4.41}$$

This means that both time load vectors die out suddenly between the time steps "T_{k+1} – Δt" and "T_{k+1}". In order to avoid this sudden unloading and consequent inaccurate results, a decaying function may be appended to both load components. The decaying function must start in the last point of the current time segment (T_{k+1} – Δt) and its extension ($\Delta\mathrm{T}_d$) is recommended not to be longer than one segment length ($\Delta\mathrm{T}_s$), due to implementation more than performance or accuracy reasons (Darbre, 1990).

One type of decaying function that can be used for this purpose is the sinusoidal function proposed by Darbre (1990), which is defined by Eqs. (4.42) and (4.43). This decaying function matches the value, slope and curvature of the loading signal in the time step "T_{k+1} – Δt".

$$f(\tau) = \left[f(0) + \tau\dot{f}(0) + \frac{\tau^2}{2}\ddot{f}(0)\right]\left[1 - \frac{\tau}{\Delta\mathrm{T}_d} + \frac{\sin\left(2\pi\tau/\Delta\mathrm{T}_d\right)}{2\pi}\right], \quad 0 < \tau < \Delta\mathrm{T}_d \tag{4.42}$$

$$\Delta\mathrm{T}_d \leq \Delta\mathrm{T}_s \tag{4.43}$$

As expressed in Eq. (4.40), the ground acceleration signal is taken from 0 to "T_{k+1} – Δt", neglecting the subsequent part. Instead, the decaying function [Eq. (4.42)] is appended to the signal in the time point "T_{k+1} – Δt", followed by a band of zeros until the Fourier period (T_p) is completed.

Therefore, the modified ground acceleration signal for the time segment "k" is expressed in Eq. (4.44), where the definition of τ_k is given by Eq. (4.45). Fig. 4.7 is a graphical representation of Eq. (4.44) applied to the ground acceleration loading shown in Fig. 4.2.

$$\overline{\ddot{\mathbf{u}}}_{\mathrm{g}}{}^{k}(t) = \ddot{\mathbf{u}}_{\mathrm{g}}(t)[H(t) - H(t - \mathrm{T}_{k+1})] + \boldsymbol{f}_{\ddot{u}_g}(\tau_k), \qquad 0 < t < \mathrm{T}_p \tag{4.44}$$

$$\tau_k = t - T_{k+1} + \Delta t \tag{4.45}$$

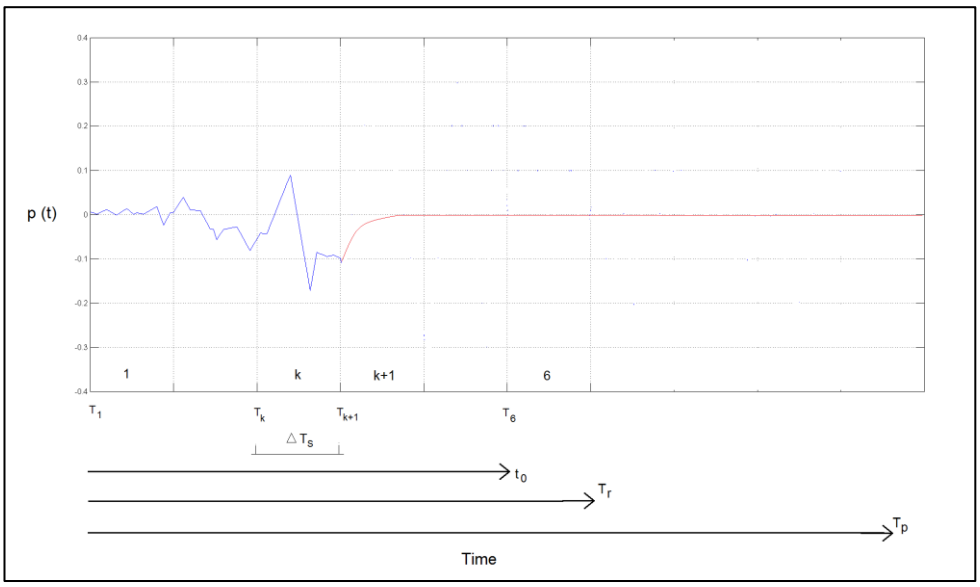


Figure 4.7: Modified excitation load with appended decaying function for the HFTD analysis of the kth time segment.

On the other hand, for the first iteration the pseudo force vector is set equal to zero for all the time steps between 0 and T_p. For the next iterations, the values of the pseudo force vector corresponding to the converged time steps (from 0 to t_c) are kept fixed, whereas for the other not converged time steps inside the current segment (from t_c to $T_{k+1} - \Delta t$), the pseudo force values corresponding to the last iteration are assigned.

Similarly as in the case of the earthquake ground acceleration, the pseudo force values of the following time segments correspond to the appended decaying function, continuing with a band of zeros until completing the Fourier period (T_p). The modified pseudo force vector for the time segment "k" and iteration "j" is expressed in Eq. (4.46).

$$\bar{\mathbf{q}}^{k,j+1}(t) = \bar{\mathbf{q}}^{j}(t)[H(t) - H(t - T_{k+1})] + \boldsymbol{f}_q(\tau), \qquad 0 < t < T_p \tag{4.46}$$

4.3.6. Convergence criteria

The convergence criteria check is performed for both, the pseudo force vector and the displacement vector in natural or generalized coordinates. For each time step inside the current segment, the convergence criteria verification after iteration "j" is given by Eqs. (4.47) and (4.48).

$$\frac{|\mathbf{u}_r{}^{j}(t) - \mathbf{u}_r{}^{j-1}(t)|}{|\mathbf{u}_r{}^{j}(t)|} \leq tol, \qquad t_c + \Delta t \leq t \leq T_{k+1} - \Delta t \tag{4.47}$$

$$\frac{|\mathbf{q}^{j}(t) - \mathbf{q}^{j-1}(t)|}{|\mathbf{q}^{j}(t)|} \leq tol, \qquad t_c + \Delta t \leq t \leq T_{k+1} - \Delta t \tag{4.48}$$

4.4. HFTD method input parameters[5]

4.4.1. Number of modes included in the mode superposition method

The number of modes to be included in the analysis can be determined from the percentage of effective mass in the direction of analysis, which is obtained from the eigen-analysis of the system (Section 3.1.9). The general criterion is to include the minimum number of modes which effective masses in the direction of analysis provide 90% of the total mass of the structure[6].

The effective masses in the three translational directions of analysis corresponding to each mode of vibration are part of the output results of the eigen-analysis. For example, the effective mass in the X-direction for the mode of vibration "i" is defined by Eqs. (4.49), (4.50) and (4.51).

$$m_{eff,X,i} = \frac{{l_{Xi}}^2}{m_i} \tag{4.49}$$

$$l_{Xi} = \boldsymbol{\phi}_{\mathrm{i}}^{\mathrm{T}} \mathbf{M}_{\mathrm{S}} \mathbf{i}_{\mathrm{x}}^{(\mathrm{s})} \tag{4.50}$$

$$m_i = \boldsymbol{\phi}_{\mathrm{i}}^{\mathrm{T}} \mathbf{M}_{\mathrm{S}} \boldsymbol{\phi}_{\mathrm{i}} \tag{4.51}$$

4.4.2. Range of excitation frequencies for the frequency domain solution

The range of frequency samples used in the Fourier analysis goes from 0 to F, which is the maximum sampling frequency. Taking into account that the Discrete Fourier Transformation (DFT) considers only positive frequency samples (one-sided expansion), the maximum signal frequency effectively processed is one half of the maximum sampling frequency (Section 2.4). This relation between the highest harmonic component of the signal included in the analysis (Nyquist frequency) and the maximum sampling frequency (F), expressed in Eq. (4.52), is called the Nyquist sampling theorem (Craig and Kurdila, 2006).

$$F_{Nyq} = \frac{F}{2} \tag{4.52}$$

The value of the maximum sampling frequency (F) is inversely proportional to the time step size (Δt). This means that once Δt is defined, F is determined using Eq. (4.53).

$$F = {}^{1}/_{\Delta t} \tag{4.53}$$

[5] Based on Chavez and Fenves (1994), Veletsos and Ventura (1985), Craig and Kurdila (2006) and TNO DIANA BV (2011).

[6] In case of analysis including dam-foundation interaction, the mass of the soil should not be taken into account in the calculation of the effective mass.

4.4.3. Time step size

The time step size should be small enough when compared to the more significant oscillation periods of the loading harmonic components and to the fundamental free vibration period of the structural system. The first condition ensures the correct representation of the forced or steady-state component of the system's total response, whereas the second condition allows obtaining a correct representation of the transient free vibration component (Veletsos and Ventura, 1985).

If high frequency systems subject to long duration loading are analyzed, then the first condition based on the loading period of oscillation rules the size of the time step. On the other hand, in the case of low frequency systems subject to short duration loading (f.e. earthquake analysis of high dams), the second condition based on the fundamental free vibration period is the most relevant criterion.

The time step size which takes into account both criteria fulfills the conditions stated in Eq. (4.54), in terms of the fundamental vibration frequency (F_1), the vibration frequency of the highest mode included in the analysis (F_q), and the greater frequency usually contained in earthquake ground acceleration records (25 Hz) (Chavez and Fenves, 1994).

$$\Delta t \leq min \begin{cases} 0.01/F_1 \\ 0.5/F_q \\ 0.02\ s \end{cases} \quad \textbf{(4.54)}$$

When the problem to be solved with the HFTD method is highly nonlinear (f.e. problems including contact or base sliding nonlinearities), it is recommended that in addition the time step size fulfills Eq. (4.55) (Chavez and Fenves, 1994).

$$\Delta t \leq 0.01\ s \quad \textbf{(4.55)}$$

4.4.4. Parameters for the Discrete Fourier Transform (DFT)

As explained before, the total period for the Fourier Transform (T_p) is calculated using Eqs. (4.36) and (4.37). Additionally, it is recommended that T_p fulfills the condition stated by Eq. (4.56) (Chavez and Fenves, 1994). However the very long value of T_p obtained with Eq. (4.56) may not be manageable, in terms of computational time and storage memory capacity, for large multi degree of freedom (MDOF) systems. Then in those cases the use of a reduced Fourier period should be evaluated without prejudice of the accuracy of the results obtained.

$$T_p \geq 50\ T_1 \quad \textbf{(4.56)}$$

The number of time point samples (N) given by Eq. (4.57) is obtained from the Fourier period (T_p) and the time step size (Δt). The frequency increment (Δf) is calculated using Eq. (4.58), whereas the maximum sampling frequency (F) is calculated with Eq. (4.59), which is equivalent to Eq. (4.53).

$$N = \frac{T_p}{\Delta t} \quad \textbf{(4.57)}$$

$$\Delta f = \frac{1}{T_p} \quad \textbf{(4.58)}$$

$$F = N\Delta f \quad \textbf{(4.59)}$$

5. SINGLE DEGREE OF FREEDOM (SDOF) DYNAMIC SYSTEMS

5.1 Solution procedure of the HFTD method for SDOF non-linear systems

The HFTD method, which was previously explained in detail for seismic analysis of dam-reservoir dynamic interaction multi degree of freedom (MDOF) systems (Chapter 4), is developed here for the particular case of SDOF dynamic linear and nonlinear systems.

It was previously stated that the HFTD method was developed to analyze systems with frequency dependent properties and nonlinear behavior. Therefore, the formulation of the HFTD method for SDOF systems also considers the inclusion of frequency dependent terms, no matter that in most of the SDOF numerical examples developed in Section 5.3, these frequency dependent systems are not taken into account. This is because to evaluate the soundness, precision and stability of the HFTD method, the results obtained for the SDOF systems have to be compared with other analytical solutions or established numerical methods (f.e.: Newmark method) in the time domain, which are naturally capable of dealing with nonlinearities, but not with frequency dependence.

The SDOF system considered is expressed by Eq. (5.1). As it can be noticed, this equation of motion constitutes a general case with nonlinear behavior $[f_{int}(u,\dot{u})]$ and frequency dependent terms $[c_{fd}(\omega)]$. Therefore, it is not possible to directly solve Eq. (5.1) in the time domain.

$$m\ddot{u} + \left[\frac{1}{2\pi}\int_{-\infty}^{\infty} c_{fd}(\omega)e^{i(\omega t)}d\omega\right]\dot{u} + f_{int}(u,\dot{u}) = p(t) \qquad \textbf{(5.1)}$$

Assuming that additive decomposition between linear and nonlinear terms is possible, the internal nonlinear restoring force is expressed by Eq. (5.2) in terms of the pseudo force $[q(t)]$, the constant stiffness (k) and the constant damping (c) (Aprile et al., 1994).

$$f_{int}(u,\dot{u}) = c\dot{u} + ku - q(t) \qquad \textbf{(5.2)}$$

After replacing Eq. (5.2) into Eq. (5.1) and passing the pseudo force to the right hand side, Eq. (5.3) is obtained. This equation of motion contains only linear terms in the left hand side, whereas all sources of nonlinearities are concentrated in the pseudo force term located in the right hand side of Eq. (5.3).

$$m\ddot{u} + \left[\frac{1}{2\pi}\int_{-\infty}^{\infty} c_{fd}(\omega)e^{i(\omega t)}d\omega + c\right]\dot{u} + ku = p(t) + q(t) \qquad \textbf{(5.3)}$$

Obviously, the value of the pseudo force is not known a priori, and for that reason Eq. (5.3) (known as pseudo linear or linearized equation) has to be solved iteratively until convergence is achieved for both, the displacement solution (u) and the pseudo force (q). Based on Eq. (5.2), Eq. (5.4) shows the expression to calculate the value of the pseudo force in iteration j.

$$q^j(t) = c\dot{u}^j + ku^j - f_{int}\left(u^j,\dot{u}^j\right) \qquad \textbf{(5.4)}$$

If there were no frequency dependent terms, the pseudo linear equation could be solved in both, frequency and time domains. However, frequency dependent terms are considered in Eq. (5.3) and therefore, the HFTD method solves the pseudo linear system in the frequency domain.

Accordingly, the excitation force is expressed in terms of its harmonic components defined in Eq. (5.5). As a consequence, the displacement and the pseudo force are defined in the same terms by Eqs. (5.6) and (5.7), respectively. N is the number of excitation frequencies (ω_n) contained in the force function (or signal) $p(t)$.

$$p = \sum_{n=0}^{N-1}\left[\hat{p}(\omega_n)e^{i\omega_n t}\right] \quad \textbf{(5.5)}$$

$$u = \sum_{n=0}^{N-1}\left[\hat{u}(\omega_n)e^{i\omega_n t}\right] \quad \textbf{(5.6)}$$

$$q = \sum_{n=0}^{N-1}\left[\hat{q}(\omega_n)e^{i\omega_n t}\right] \quad \textbf{(5.7)}$$

Eq. (5.8) is the pseudo linear equation of motion in the frequency domain, which is solved for each excitation frequency (ω_n). Eq. (5.8) is obtained in terms of the complex valued amplitudes of the excitation force [$\hat{p}(\omega_n)$], displacement [$\hat{u}(\omega_n)$] and pseudo force [$\hat{q}(\omega_n)$], after replacing Eqs. (5.5) to (5.7) into Eq. (5.3).

$$\left[-{\omega_n}^2\, m + i\omega_n\left(c_{fd}(\omega_n) + c\right) + k\right]\hat{u}(\omega_n) = \hat{p}(\omega_n) + \hat{q}(\omega_n) \quad \textbf{(5.8)}$$

Prior to solve Eq. (5.8), the complex right hand side amplitudes [$\hat{p}(\omega_n)$, $\hat{q}(\omega_n)$] have to be determined. Then, the equation is solved and the complex amplitude of the displacement [$\hat{u}(\omega_n)$] is obtained. The HFTD method uses the Discrete Fourier Transform (DFT) defined in Eqs. (5.9) and (5.10) to determine $\hat{p}(\omega_n)$ and $\hat{q}(\omega_n)$, respectively.

$$\hat{p}(\omega_n) = \frac{1}{N}\sum_{m=0}^{N-1}\left[p(t_m)\, e^{-i(\frac{2\pi mn}{N})}\right] \quad \textbf{(5.9)}$$

$$\hat{q}(\omega_n) = \frac{1}{N}\sum_{m=0}^{N-1}\left[q(t_m)\, e^{-i(\frac{2\pi mn}{N})}\right] \quad \textbf{(5.10)}$$

The number of excitation frequencies (N) considered in the analysis depends on the period used for the Fourier transform (T_p) and the time step size (Δt) that determines the separation of the discrete time points in which the response of the system is computed. Eq. (5.11) presents the expression to determine N, whereas Eq. (5.12) the expression to calculate ω_n.

$$N = {T_p}/{\Delta t} \quad \textbf{(5.11)}$$

$$\omega_n = \frac{2\pi n}{T_p} \quad \textbf{(5.12)}$$

In Chapter 4 it was explained that in the HFTD method the Fourier period (T_p) is not equal to but longer than the duration of the excitation force (t_0). The reason for this is to obtain the transient response of the system instead of the steady state response (Section 4.3.3). The expression to calculate the ideal value of the Fourier period is given by Eq. (5.13), where the natural period of the undamped system (T_1) and the viscous damping ratio (ξ) are defined by Eqs. (5.14) and (5.15), respectively. For the SDOF systems treated in this chapter, the very long value of T_p obtained with Eq. (5.13) is totally manageable in terms of computational time and memory capacity.

$$T_p = max\left\{t_0 + 0.75\frac{T_1}{\xi}\ ;\ 50\ T_1\right\} \quad \textbf{(5.13)}$$

$$T_1 = 2\pi\sqrt{m/k} \quad \textbf{(5.14)}$$

$$\xi = c/c_{cr} = \frac{c}{2\sqrt{k\ m}} \quad \textbf{(5.15)}$$

The fundaments and procedure of the time segmentation approach followed by the HFTD method to achieve convergence when solving nonlinear systems have already been explained in Section 4.3. This procedure consists in dividing the time span of interest (not the total Fourier period) in time segments over which the iterative solution is found sequentially for each of these segments.

The time span of interest (T_r) is the time for which the user desires to obtain the response of the system. Usually, T_r is equal or longer than the load duration but much smaller than the Fourier period ($t_0 \leq T_r << T_p$).

On the other hand, all the time segments are generally composed of a constant number of time steps. Therefore, the size of each time segment (ΔT_S) is calculated with Eq. (5.16), in terms of the number of samples (or time points) per segment (N_S) and the time step size (Δt), whereas the starting time point of segment k (T_k) is calculated using Eq. (5.17).

$$\Delta T_S = N_S\ \Delta t \quad \textbf{(5.16)}$$

$$T_k = (k-1)\Delta T_S \quad \textbf{(5.17)}$$

Fig. 5.1 shows an example of the excitation load $p(t)$ with duration equal to t_0 and Fourier period equal to T_p. In this particular case the time span of interest (T_r) is divided in six time segments. For each time segment, $\hat{p}(\omega_n)$ and $\hat{q}(\omega_n)$ should be calculated by means of the DFT expressed in Eqs. (5.9) and (5.10), respectively.

If the k^{th} time segment of a system subjected to the load shown in Fig. 5.1 is being analyzed, the segmentation approach of the HFTD method requires modifying the excitation load $p(t)$ before doing the DFT. The required modifications of load $p(t)$ are shown in Fig. 5.2. The original force signal is replicated until the k^{th} segment currently analyzed. Subsequently, an appended decaying function that develops a smooth transition to load values equal to zero is added. Finally, the zero load values are extended until covering all the time points contained by the Fourier period (T_p).

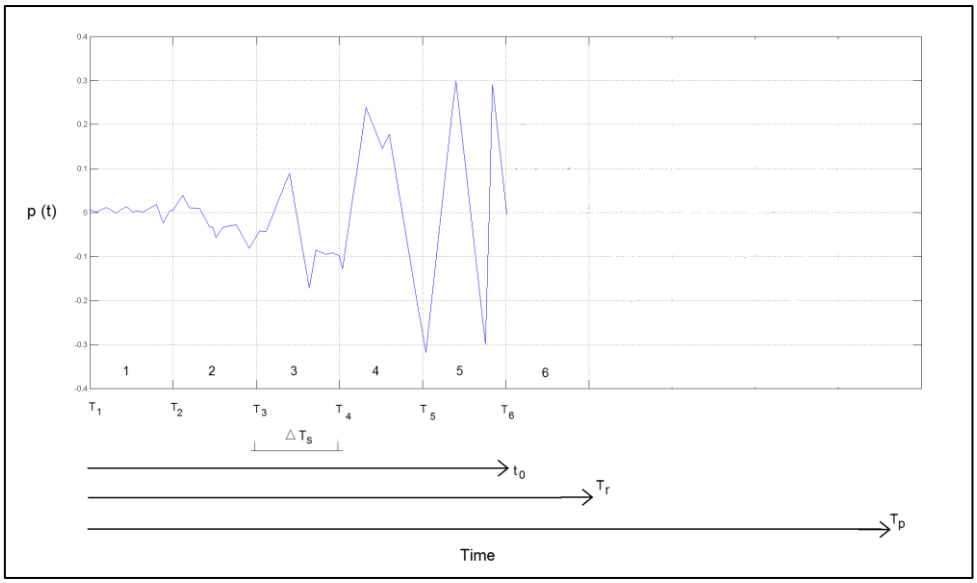


Figure 5.1: Excitation load, segmentation approach, time span of interest and Fourier period.

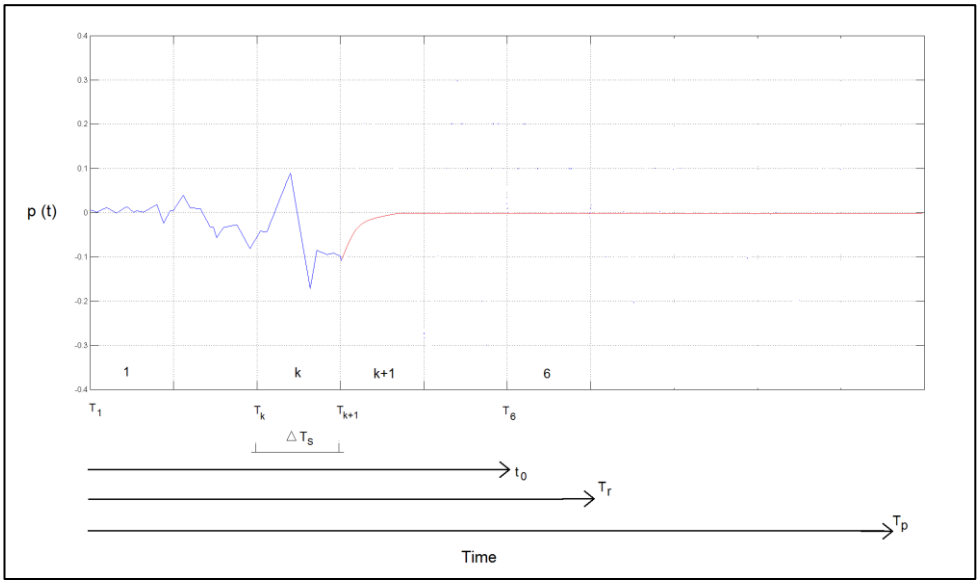


Figure 5.2: Modified excitation load with appended decaying function for the HFTD analysis of the k^{th} time segment.

The use of a decaying function avoids a sudden unloading of the system, and consequent inaccuracy and artificial oscillations in the obtained response. Taking into account that in the pseudo linear equation of motion solved by the HFTD method, the frequency dependent complex amplitude terms located in the right hand side of Eq. (5.8) do not only consist in the Fourier transform of the load

$[p(t)]$, but of the pseudo force $[q(t)]$ as well, a similar modification shown in Fig. 5.2 for $p(t)$ has to be accomplished for $q(t)$ before the DFT defined in Eq. (5.10) is applied.

The modifications of the excitation load and the pseudo force for the analysis of the k[th] time segment are analytically expressed in Eqs. (5.18) and (5.19), respectively, by means of the Heaviside function $[H(t)]$.

$$p^k(t) = p(t)[H(t) - H(t - \mathrm{T}_{k+1})] + f_p(\tau_k), \qquad 0 < t < \mathrm{T}_p \tag{5.18}$$

$$q^k(t) = q(t)[H(t) - H(t - \mathrm{T}_{k+1})] + f_q(\tau_k), \qquad 0 < t < \mathrm{T}_p \tag{5.19}$$

Several types of decaying functions can be used for the load $[f_p(\tau_k)]$ and pseudo force $[f_q(\tau_k)]$. Nevertheless, the simplest decaying function defined by Eqs. (5.20) and (5.21) is selected for both of them. This is a linear decaying function defined in terms of its time variable (τ_k) and its duration (ΔT_d). τ_k starts in the last point of the time segment k, whereas by definition ΔT_d should be smaller than the time segment length (ΔT_S).

$$f(\tau_k) = f(0)\left(1 - \frac{\tau_k}{\Delta T_d}\right), \quad 0 < \tau_k < \Delta T_d \tag{5.20}$$

$$\tau_k = t - T_{k+1} + \Delta t \tag{5.21}$$

Going back to the solution of Eq. (5.8), the complex amplitude of the displacement $(\hat{u})$ is determined in the frequency domain for each excitation frequency (ω_n). The complex amplitudes of the system's velocity and acceleration in the frequency domain are determined for each excitation frequency using Eqs. (5.22) and (5.23).

$$\hat{\dot{u}}(\omega_n) = i\omega_n\, \hat{u}(\omega_n) \tag{5.22}$$

$$\hat{\ddot{u}}(\omega_n) = -{\omega_n}^2\, \hat{u}(\omega_n) \tag{5.23}$$

From the amplitudes of the system response in the frequency domain, the response in the time domain is obtained by means of the inverse DFT, expressed in Eqs. (5.24) to (5.26) for the displacement, velocity and acceleration, respectively.

$$u(t_m) = \sum_{n=0}^{N-1}\left[\hat{u}(\omega_n)\, e^{i(\frac{2\pi mn}{N})}\right] \tag{5.24}$$

$$\dot{u}(t_m) = \sum_{n=0}^{N-1}\left[\hat{\dot{u}}(\omega_n)\, e^{i(\frac{2\pi mn}{N})}\right] \tag{5.25}$$

$$\ddot{u}(t_m) = \sum_{n=0}^{N-1}\left[\hat{\ddot{u}}(\omega_n)\, e^{i(\frac{2\pi mn}{N})}\right] \tag{5.26}$$

Due to the segmentation approach followed in the HFTD method, the response in the time domain is only required to be calculated for those time points located inside the current segment being analyzed, for which convergence has not been achieved yet. This means that if the last time point in the segment k for which convergence has been achieved in the previous iteration is denominated t_c, then the time points t_m, for which Eqs. (5.24) to (5.26) have to be evaluated, fulfill the condition expressed in Eq. (5.27).

$$t_c + \Delta t \leq t_m \leq T_{k+1} - \Delta t \quad \textbf{(5.27)}$$

Finally, with the response obtained for the time points t_m, the pseudo force for iteration j is obtained after replacing $u^j(t_m)$ and $\dot{u}^j(t_m)$ in Eq. (5.4). Then, the convergence criteria check is performed for both, the pseudo force and the displacement. The convergence criteria verification for iteration j is presented in Eqs. (5.28) and (5.29).

$$\frac{|u^j(t_m) - u^{j-1}(t_m)|}{|u^j(t_m)|} \leq tol \quad \textbf{(5.28)}$$

$$\frac{|q^j(t_m) - q^{j-1}(t_m)|}{|q^j(t_m)|} \leq tol \quad \textbf{(5.29)}$$

If one of the time points t_m does not fulfill the convergence criteria defined by Eqs. (5.28) and (5.29), then a new iteration is initiated with the DFT of the pseudo force obtained in the previous iteration. The last time point for which convergence has been achieved (t_c) should be updated, being the process repeated until t_c coincides with the time span of interest ($t_c = T_r$). Tables 5.1 and 5.2 show a schematic summary of the HFTD method for SDOF non-linear systems.

Table 5.1: Summary of the HFTD method for SDOF non-linear systems

1. Initial calculations
   - 1.1 Select the time step size (Δt)
   - 1.2 Determine the duration of the excitation load (t_0)
   - 1.3 Calculate the fundamental period of the system (T_1)

$$T_1 = 2\pi\sqrt{m/k}$$

   - 1.4 Calculate the damping ratio (ξ)

$$\xi = \frac{c}{2\sqrt{k\,m}}$$

2. Fourier period
   - 2.1 Calculate the Fourier period (T_p)

$$T_p = max\left\{t_0 + 0.75\frac{T_1}{\xi}\ ;\ 50\,T_1\right\}$$

   - 2.2 Calculate the number of time points (N) for the Fourier transform

$$N = T_p/\Delta t$$

3. Frequency dependent properties
   - 3.1 Calculate the excitation frequencies (ω_n)

$$\omega_n = \frac{2\pi n}{T_p}$$

   - 3.2 Evaluate the frequency dependent damping $[c_{fd}(\omega_n)]$ for each excitation frequency ω_n

4. Number of time segments
   - 4.1 Select the time span of interest (T_r)
   - 4.2 Select the number of samples (time points) per segment (N_S)
   - 4.3 Calculate the size of the time segment (ΔT_S)

$$\Delta T_S = N_S\,\Delta t$$

   - 4.4 Calculate the number of time segments (N_T)

$$N_T = T_r/\Delta T_S$$

5. Set up the initial the value of the pseudo force equal to zero

$$q^0(t) = 0$$

6. Calculations for each time segment k
   - 6.1 Determine the modified excitation load for time segment k (p^k)

$$p^k(t) = p(t)[H(t) - H(t - \mathrm{T}_{k+1})] + f_p(\tau_k)$$

   - 6.2 Calculate the DFT of the excitation load $[\hat{p}(\omega_n)]$

$$\hat{p}(\omega_n) = \frac{1}{N}\sum_{m=0}^{N-1}\left[p(t_m)\,e^{-i\left(\frac{2\pi mn}{N}\right)}\right]$$

   - 6.3 Follow the HFTD iterative procedure to determine the system response and pseudo forces of the time points of segment k (See Table 5.2)

Table 5.2: Iterative procedure of the HFTD method for SDOF nonlinear systems

1. Calculations for each iteration j
   - 1.1 Determine the modified pseudo force for time segment k and iteration j-1 $[\mathrm{q}^{\mathrm{k,j-1}}(\mathrm{t})]$

$$q^{k,j-1}(t) = q^{j-1}(t)[H(t) - H(t - T_{k+1})] + f_q(\tau_k)$$

   - 1.2 Calculate the DFT of the pseudo force $[\hat{\mathrm{q}}(\omega_{\mathrm{n}})]$

$$\hat{q}(\omega_n) = \frac{1}{N}\sum_{m=0}^{N-1}\left[q^{k,j-1}(t_m)\, e^{-i(\frac{2\pi mn}{N})}\right]$$

   - 1.3 Determine the modified pseudo force for time segment k and iteration j-1 $[\mathrm{q}^{\mathrm{k,j-1}}(\mathrm{t})]$

$$q^{k,j-1}(t) = q^{j-1}(t)[H(t) - H(t - T_{k+1})] + f_q(\tau_k)$$

   - 1.4 Calculate the DFT of the pseudo force $[\hat{\mathrm{q}}(\omega_{\mathrm{n}})]$

$$\hat{q}(\omega_n) = \frac{1}{N}\sum_{m=0}^{N-1}\left[q^{k,j-1}(t_m)\, e^{-i(\frac{2\pi mn}{N})}\right]$$

   - 1.5 Calculation for each excitation frequency ω_{n}
     - 1.5.1 Solve the pseudo linear equation to determine displacement amplitude in the frequency domain $\hat{\mathrm{u}}(\omega_{\mathrm{n}})$

$$\left[-{\omega_n}^2\, m + i\omega_n\big(c_{fd}(\omega_n) + c\big) + k\right]\hat{u}(\omega_n) = \hat{p}(\omega_n) + \hat{q}(\omega_n)$$

     - 1.5.2 Calculate the velocity $[\hat{\mathrm{u}}(\omega_{\mathrm{n}})]$ and acceleration $[\hat{\mathrm{u}}(\omega_{\mathrm{n}})]$ amplitudes in the frequency domain

$$\hat{u}(\omega_n) = i\omega_n\, \hat{u}(\omega_n) \qquad \hat{\ddot{u}}(\omega_n) = -{\omega_n}^2\, \hat{u}(\omega_n)$$

   - 1.6 Calculate the inverse DFT of the displacement, velocity and acceleration

$$u(t_m) = \sum_{n=0}^{N-1}\left[\hat{u}(\omega_n)\, e^{i(\frac{2\pi mn}{N})}\right]$$

$$\dot{u}(t_m) = \sum_{n=0}^{N-1}\left[\hat{u}(\omega_n)\, e^{i(\frac{2\pi mn}{N})}\right]$$

$$\ddot{u}(t_m) = \sum_{n=0}^{N-1}\left[\hat{\ddot{u}}(\omega_n)\, e^{i(\frac{2\pi mn}{N})}\right]$$

   - 1.7 Calculation of the internal force $[f_{int}(u^j, \dot{u}^j)]$ for the time points $\mathrm{t_m}$ between $\mathrm{t_c}$ and $\mathrm{T_{k+1}}$
   - 1.8 Calculation of the pseudo force for the time points $\mathrm{t_m}$ between $\mathrm{t_c}$ and $\mathrm{T_{k+1}}$

$$q^j(t_m) = c\dot{u}^j + ku^j - f_{int}(u^j, \dot{u}^j)$$

   - 1.9 Convergence criteria verification for the time points $\mathrm{t_m}$ between $\mathrm{t_c}$ and $\mathrm{T_{k+1}}$

$$\frac{|u^j(t_m) - u^{j-1}(t_m)|}{|u^j(t_m)|} \le tol \qquad \frac{|q^j(t_m) - q^{j-1}(t_m)|}{|q^j(t_m)|} \le tol$$

   - 1.10 Update $\mathrm{t_c}$
   - 1.11 Repeat the iterative process until $\mathrm{t_c} = \mathrm{T_{k+1}} - \Delta\mathrm{t}$

5.2 Time domain analysis of SDOF systems

The accuracy of the results obtained with the HFTD method previously described for SDOF systems has to be compared with analytical solutions or other well established methods of analysis in the time

domain. This permits to evaluate the HFTD method performance and generate confidence about the results obtained when frequency dependent properties are actually required to be included. Nevertheless, it is obvious that the numerical examples tested in the section 5.3 cannot have frequency dependent properties, due to fact that the benchmark solutions are obtained from time domain analysis methods.

This section is divided in two parts. The first part covers the analytic solution in the time domain of SDOF linear systems. Two particular types of loading are considered: harmonic and general. The second part develops the well-established Newmark method in the time domain for SDOF non-linear systems.

5.2.1 Analytic solution in the time domain for SDOF linear systems[1]

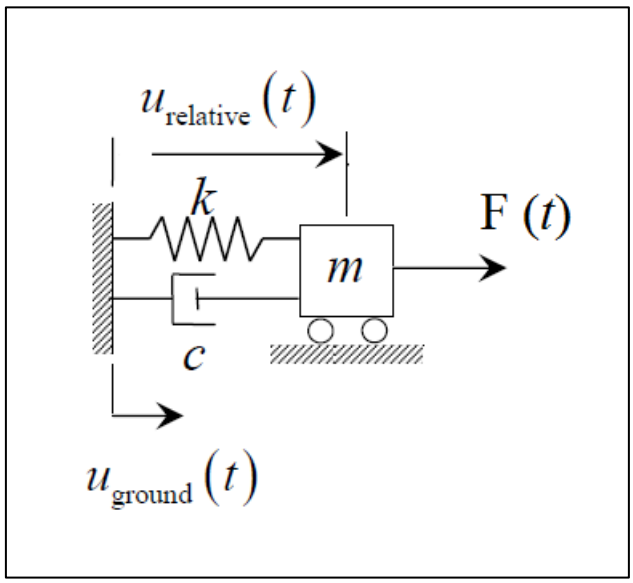


Figure 5.3: Single degree of freedom damped system

The equation of motion of the damped SDOF linear system shown in Fig. 5.3 is expressed by Eq. (5.30) in terms of the relative displacement (u_r). The initial conditions expressed in Eqs. (5.31) and (5.32) complete the definition of the governing equations.

$$m\ddot{u}_r + c\dot{u}_r + ku_r = -m\ddot{u}_g(t) + F(t) \quad \textbf{(5.30)}$$

$$u_r(0) = u_0 \quad \textbf{(5.31)}$$

$$\dot{u}_r(0) = v_0 \quad \textbf{(5.32)}$$

[1] Based on Craig and Kurdila (2006) and Paz and Leigh (2004).

Eq. (5.30) can be expressed as shown in Eq. (5.33). The definition of the natural frequency of vibration (ω_o) and the viscous damping ratio (ξ) is presented in Eqs. (5.34) and (5.15), respectively.

$$\ddot{u}_r + 2\xi\omega_o\,\dot{u}_r + \omega_o^2\,u_r = -\ddot{u}_g(t) + \frac{F(t)}{m} \tag{5.33}$$

$$\omega_o = \sqrt{k/m} \tag{5.34}$$

If the right hand side of Eq. (5.33) is denominated $\frac{p(t)}{m}$ and the subscript "r" of relative movement is omitted, a general equation valid for relative and absolute coordinates is expressed in Eq. (5.35).

$$\ddot{u} + 2\xi\omega_o\,\dot{u} + \omega_o^2 u = \frac{p(t)}{m} \tag{5.35}$$

5.2.1.1 Response of SDOF systems subjected to harmonic loading

When the excitation force acting in the right-hand side of Eq. (5.35) is harmonic in the form $p(t) = p_0\cos(\Omega t)$, then Eq. (5.35) can be expressed as shown in Eq. (5.36).

$$\ddot{u} + 2\xi\omega_o\,\dot{u} + \omega_o^2 u = \frac{p_0}{m}\cos(\Omega t) \tag{5.36}$$

The solution of Eq. (5.7) is given by the summation of the general homogeneous solution and the particular solution. For under-damped systems ($\xi < 1$), the general solutions is defined by Eqs. (5.37) and (5.38).

$$u_{general} = e^{-\xi\omega_o t}(A_1\cos(\omega_d t) + A_2\sin(\omega_d t)) \tag{5.37}$$

$$\omega_d = \omega_o\sqrt{1-\xi^2} \tag{5.38}$$

On the other hand, the particular solution or steady state response is given by Eq. (5.39) in terms of the displacement amplitude (U) and the phase angle (α), defined by Eqs. (5.40) and (5.41), respectively.

$$u_{particular} = U\cos(\Omega t - \alpha) \tag{5.39}$$

$$U = p_0/k\;\frac{1}{\sqrt{\left[1-\left(\Omega/\omega_o\right)^2\right]^2+\left[2\xi\left(\Omega/\omega_o\right)\right]^2}} \tag{5.40}$$

$$\tan\alpha = \frac{2\xi\left(\Omega/\omega_o\right)}{1-\left(\Omega/\omega_o\right)^2} \tag{5.41}$$

Therefore, the total solution of the SDOF under-damped linear system subjected to the harmonic load $p_0 \cos(\Omega t)$ is given by the sum of Eqs. (5.37) and (5.39), as expressed in Eq. (5.42). The derivative of Eq. (5.42) provides the expression of the velocity of the system, shown in Eq. (5.43). The acceleration can be obtained from the equation of motion, after the substitution of Eqs. (5.42) and (5.43) into Eq. (5.36). The values of A_1 and A_2 are obtained from the initial conditions defined in Eqs. (5.31) and (5.32).

$$u(t) = U\cos(\Omega t - \alpha) + e^{-\xi\omega_o t}(A_1\cos(\omega_d t) + A_2\sin(\omega_d t)) \quad \textbf{(5.42)}$$

$$\dot{u}(t) = -\Omega U\sin(\Omega t - \alpha) + e^{-\xi\omega_o t}[(A_2\omega_d - A_1\xi\omega_o)\cos(\omega_d t) - (A_1\omega_d + A_2\xi\omega_o)\sin(\omega_d t)] \quad \textbf{(5.43)}$$

5.2.1.2 Response of SDOF systems subjected to general loading by the Convolution Integral method

If the loading is not harmonic but a general dynamic excitation $p(t)$, the particular solution is obtained by the Convolution (or Duhamel) integral method, which is expressed in Eq. (5.44). The total solution is given therefore by Eq. (5.45).

$$u_{particular} = \frac{1}{m\,\omega_d}\int_0^t p(\tau)e^{-\xi\omega_o(t-\tau)}\sin\big(\omega_d(t-\tau)\big)\,d\tau \quad \textbf{(5.44)}$$

$$u(t) = \frac{1}{m\,\omega_d}\int_0^t p(\tau)e^{-\xi\omega_o(t-\tau)}\sin(\omega_d(t-\tau))d\,\tau + e^{-\xi\omega_o t}(A_1\cos(\omega_d t) + A_2\sin(\omega_d t)) \quad \textbf{(5.45)}$$

5.2.1.3 Response of SDOF systems subjected to general loading by the Direct Integration method

In many applications, the excitation force $p(t)$ cannot be expressed by a mathematical function (f.e: ground acceleration signal records) and therefore, the convolution integral of Eq. (5.45) cannot be obtained in a continuous closed form.

One way to deal with these kind of problems without losing accuracy and theoretical consistency is to evaluate the convolution integral in a discrete form by dividing the loading in small linear segments, as shown in Fig. 5.4.

In this way, if the time step size (Δt) that determine the position of the discrete points in the force function is small enough, the solution of the system can be calculated for the time step i, using as initial conditions the solution obtained for the time step i-1. This approach is known as the Direct Integration Method and provides the "exact" analytical solution of the system, provided that the time step size is small enough to describe accurately the force function.

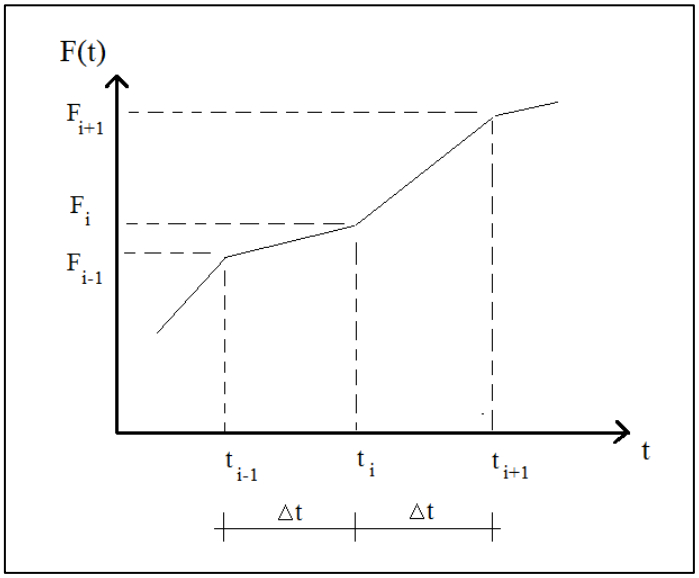


Figure 5.4: Discretization of the force the function

The force for the time interval between t_i and t_{i+1} is approximated to a linear function by Eqs. (5.46) and (5.47). Therefore, the equation of motion (only valid for the time interval between t_i and t_{i+1}) is expressed in Eq. (5.48).

$$p(\tau) = \left(1 - \frac{\tau}{\Delta t}\right) p_i + \left(\frac{\tau}{\Delta t}\right) p_{i+1} \quad \textbf{(5.46)}$$

$$\tau = t - t_i \quad \textbf{(5.47)}$$

$$\ddot{u} + 2\xi\omega_o\,\dot{u} + \omega_o^2 u = \frac{1}{m}\left[\left(1 - \frac{\tau}{\Delta t}\right) p_i + \left(\frac{\tau}{\Delta t}\right) p_{i+1}\right] \quad \textbf{(5.48)}$$

The general and particular solutions of the system for the interval i are given by Eqs. (5.49) and (5.50), respectively. The constants C_i and D_i of the general solution are obtained based on the initial conditions of the interval (i.e. u_i and $\dot{u}_i$). On the other hand, the constants A_i and B_i are obtained after replacing the particular solution in the equation of motion defined in Eq. (5.48).

$$u_{general} = e^{-\xi\omega_o\tau}(C_i\cos(\omega_d\tau) + D_i\sin(\omega_d\tau)) \quad \textbf{(5.49)}$$

$$u_{particular} = B_i + A_i\tau \quad \textbf{(5.50)}$$

The total solution for displacement and velocity is given respectively by Eqs. (5.51) and (5.52), respectively.

$$u(\tau) = B_i + A_i\tau + e^{-\xi\omega_o\tau}(C_i\cos(\omega_d\tau) + D_i\sin(\omega_d\tau)) \quad \textbf{(5.51)}$$

$$\dot{u}(\tau) = A_i + e^{-\xi\omega_o\tau}[(D_i\omega_d - C_i\xi\omega_o)\cos(\omega_d\tau) - (C_i\omega_d + D_i\xi\omega_o)\sin(\omega_d\tau)] \quad \textbf{(5.52)}$$

After replacing Eq. (5.50) into Eq. (5.48), the expressions defined in Eqs. (5.53) and (5.54) are the solution for A_i and B_i, respectively.

$$A_i = \frac{p_{i+1} - p_i}{m\,\omega_o^2\,\Delta t} \tag{5.53}$$

$$B_i = \frac{p_i - 2m\,\xi\omega_o A_i}{m\,\omega_o^2} \tag{5.54}$$

At the beginning of interval i, the initial conditions are u_i and $\dot{u}_i$. After replacing both conditions in Eqs. (5.51) and (5.52), respectively, the constants C_i and D_i are obtained with the expressions given by Eqs. (5.55) and (5.56).

$$C_i = u_i - B_i \tag{5.55}$$

$$D_i = \frac{\dot{u}_i - A_i - \xi\omega_o C_i}{\omega_d} \tag{5.56}$$

Finally, the solution of interval i is completely defined and therefore it is possible to determine the response of the next time step i+1, by replacing the time $\tau = \Delta t$ in Eqs. (5.51) and (5.52), to obtain u_{i+1} and $\dot{u}_{i+1}$ respectively. The acceleration ($\ddot{u}_{i+1}$) is obtained from Eq. (5.48).

After substitution and simplification, the following final recurrence formulas for displacement, velocity and acceleration of the time step i+1 are presented in Eqs. (5.57) to (5.59). The expressions to calculate the coefficients of the recurrence formulas are summarized in Table 5.3.

$$u_{i+1}(\tau) = Ap_i + Bp_{i+1} + Cu_i + D\dot{u}_i \tag{5.57}$$

$$\dot{u}_{i+1}(\tau) = A'p_i + B'p_{i+1} + C'u_i + D'\dot{u}_i \tag{5.58}$$

$$\ddot{u}_{i+1}(\tau) = -2\xi\omega_o\,\dot{u}_{i+1} - \omega_o^2\,u_{i+1} + \frac{p_{i+1}}{m} \tag{5.59}$$

Table 5.3: Coefficients for the recurrence formulas of the Direct Integration Method

$$\beta = \xi\omega_o$$

$$A = \frac{1}{k\omega_d\Delta t}\left\{e^{-\beta\Delta t}\left[\left(\frac{\omega_d^2-\beta^2}{\omega_o^2}-\beta\Delta t\right)\sin(\omega_d\Delta t)-\left(\frac{2\omega_d\beta}{\omega_o^2}+\omega_d\Delta t\right)\cos(\omega_d\Delta t)\right]+\frac{2\omega_d\beta}{\omega_o^2}\right\}$$

$$B = \frac{1}{k\omega_d\Delta t}\left\{e^{-\beta\Delta t}\left[-\frac{\omega_d^2-\beta^2}{\omega_o^2}\sin(\omega_d\Delta t)+\frac{2\omega_d\beta}{\omega_o^2}\cos(\omega_d\Delta t)\right]+\omega_d\Delta t-\frac{2\omega_d\beta}{\omega_o^2}\right\}$$

$$C = e^{-\beta\Delta t}\left[\cos(\omega_d\Delta t)+\frac{\beta}{\omega_d}\sin(\omega_d\Delta t)\right]$$

$$D = \frac{1}{\omega_d}e^{-\beta\Delta t}\sin(\omega_d\Delta t)$$

$$A' = \frac{1}{k\omega_d\Delta t}\left\{e^{-\beta\Delta t}[(\omega_o^2\Delta t+\beta)\sin(\omega_d\Delta t)+\omega_d\cos(\omega_d\Delta t)]-\omega_d\right\}$$

$$B' = \frac{1}{k\omega_d\Delta t}\left\{-e^{-\beta\Delta t}[\beta\sin(\omega_d\Delta t)+\omega_d\cos(\omega_d\Delta t)]+\omega_d\right\}$$

$$C' = -\frac{\omega_o^2}{\omega_d}e^{-\beta\Delta t}\sin(\omega_d\Delta t)$$

$$D' = e^{-\beta\Delta t}\left[\cos(\omega_d\Delta t)-\frac{\beta}{\omega_d}\sin(\omega_d\Delta t)\right]$$

5.2.2 Solution procedure of the Newmark method for SDOF non-linear systems[2]

The equation of motion for nonlinear systems solved by the HFTD method [Eq. (5.1)] is modified in such a way that no frequency dependent terms are taken into account, as it is expressed in Eq. (5.60). This general equation for nonlinear SDOF systems is valid for absolute and relative coordinates, and therefore, the force vector $p(t)$ can include a ground acceleration excitation.

$$m\ddot{u} + f_{int}(u,\dot{u}) = p(t) \qquad \textbf{(5.60)}$$

One of the most stable and accurate time domain methods for nonlinear analysis of structural dynamics systems is the Newmark method. Due to implementation purposes of Newmark method, it is desirable to split the nonlinear internal force $[f_{int}(u,\dot{u})]$ in stiffness $[f_S(u,\dot{u})]$ and damping $[f_D(u,\dot{u})]$ nonlinear forces. This is expressed in Eq. (5.61).

$$m\ddot{u} + f_D(u,\dot{u}) + f_S(u,\dot{u}) = p(t) \qquad \textbf{(5.61)}$$

Eq. (5.61) can be discretized so it is valid at discrete time instants t_i. Therefore, the nonlinear equation of motion is satisfied in the time instants t_i and t_{i+1}, as it is expressed in Eqs. (5.62) and (5.63).

$$m\ddot{u}_i + f_D(u_i,\dot{u}_i) + f_S(u_i,\dot{u}_i) = p_i(t) \qquad \textbf{(5.62)}$$

$$m\ddot{u}_{i+1} + f_D(u_{i+1},\dot{u}_{i+1}) + f_S(u_{i+1},\dot{u}_{i+1}) = p_{i+1}(t) \qquad \textbf{(5.63)}$$

[2] Based on Chopra (2007).

The difference between Eqs. (5.63) and (5.62) gives the incremental equilibrium equation expressed in Eq. (5.64), for which the time step is defined by Eq. (5.65)

$$m\Delta\ddot{u}_i + \Delta f_{D\,i} + \Delta f_{S_i} = \Delta p_i \quad \textbf{(5.64)}$$

$$\Delta t = t_{i+1} - t_i \quad \textbf{(5.65)}$$

The incremental stiffness and damping internal restoring forces can be expressed as shown in Eqs. (5.66) and (5.67), respectively. This approximation is valid for small step sizes (Δt) and for tangent stiffness and damping coefficients defined by Eqs. (5.68) and (5.69), respectively.

$$\Delta f_{D\,i} = c_i \Delta\dot{u}_i \quad \textbf{(5.66)}$$

$$\Delta f_{S_i} = k_i \Delta u_i \quad \textbf{(5.67)}$$

$$c_i = \frac{\partial f_D}{\partial \dot{u}}(u_i, \dot{u}_i) \quad \textbf{(5.68)}$$

$$k_i = \frac{\partial f_S}{\partial u}(u_i, \dot{u}_i) \quad \textbf{(5.69)}$$

After replacing Eqs. (5.66) and (5.67) into Eq. (5.64), the incremental equilibrium equation is reformulated and presented in Eq. (5.70). This equation has the same form of the equation for linear systems, and this is the reason for splitting the restoring force in Eq. (5.61).

$$m\Delta\ddot{u}_i + c_i\Delta\dot{u}_i + k_i\Delta u_i = \Delta p_i \quad \textbf{(5.70)}$$

Now, the Newmark method can be applied in the same fashion as it is done for linear systems. The only differences are that the tangent stiffness and damping have to be updated for each time step, and that an iterative procedure has to be included to avoid the error accumulation of pure incremental procedures (modified Newton-Raphson iterative method)[3].

The Newmark method approximates the displacement and velocity of the system in the time instant t_{i+1} via the recurrence algorithms presented in Eqs. (5.71) and (5.72), respectively. The values of the parameters γ and β determine the stability and accuracy of the method. The two most used cases due to their high accuracy are the linear (γ=1/2, β=1/6) and average (γ=1/2, β=1/4) acceleration methods.

$$u_{i+1} = u_i + \Delta t\ \dot{u}_i + \left(\frac{1}{2} - \beta\right)(\Delta t)^2\ddot{u}_i + \beta(\Delta t)^2\ddot{u}_{i+1} \quad \textbf{(5.71)}$$

$$\dot{u}_{i+1} = \dot{u}_i + (1-\gamma)\Delta t\ \ddot{u}_i + \gamma\ \Delta t\ \ddot{u}_{i+1} \quad \textbf{(5.72)}$$

[3] See section 5.7 of Chopra (2007), for a detailed explanation of the modified Newton-Raphson iterative procedure applied to nonlinear dynamic analysis of SDOF systems.

After rearranging Eqs. (5.71) and (5.72), the incremental velocity and acceleration are expressed in Eqs. (5.73) and (5.74), respectively, in terms of the incremental displacement and the response of the time instant t_i.

$$\Delta\dot{u}_i = \frac{\gamma}{\beta\,\Delta t}\Delta u_i - \frac{\gamma}{\beta}\dot{u}_i + \left(1 - \frac{\gamma}{2\beta}\right)\Delta t\,\ddot{u}_i \tag{5.73}$$

$$\Delta\ddot{u}_i = \frac{1}{\beta\,(\Delta t)^2}\Delta u_i - \frac{1}{\beta\,\Delta t}\dot{u}_i - \frac{1}{2\beta}\,\ddot{u}_i \tag{5.74}$$

Eqs. (5.73) and (5.74) are substituted into Eq. (5.70) to obtain the incremental equilibrium equation, defined by Eqs. (5.75) to (5.77). The only unknown of Eq. (5.75) is the incremental displacement (Δu_i), which is calculated based on the solution of the previous time instant t_i, and following the modified Newton-Raphson iterative procedure in order to obtain accurate results.

$$\hat{k}_i\,\Delta u_i = \Delta\hat{p}_i \tag{5.75}$$

$$\hat{k}_i = k_i + \frac{\gamma}{\beta\,\Delta t}c_i + \frac{1}{\beta\,(\Delta t)^2}m \tag{5.76}$$

$$\Delta\hat{p}_i = \Delta p_i + \left(\frac{1}{\beta\,\Delta t}m + \frac{\gamma}{\beta}c_i\right)\dot{u}_i + \left[\frac{1}{2\beta}m + \left(\frac{\gamma}{2\beta} - 1\right)\Delta t\,c_i\right]\ddot{u}_i \tag{5.77}$$

The incremental displacement found after solving Eq. (5.75) can be replaced into Eq. (5.73) to find $\Delta\dot{u}_i$. Then, the displacement, velocity and acceleration of the time instant t_{i+1} are calculated using Eqs. (5.78) to (5.80), respectively.

$$u_{i+1} = u_i + \Delta u_i \tag{5.78}$$

$$\dot{u}_{i+1} = \dot{u}_i + \Delta\dot{u}_i \tag{5.79}$$

$$\ddot{u}_{i+1} = \frac{p_{i+1} - f_{D\,i+1} - f_{S\,i+1}}{m} \tag{5.80}$$

The internal restoring forces in the time instant t_{i+1} are calculated incrementally, as it is expressed in Eqs. (5.81) and (5.82). The definition of the restoring forces increment is given by Eqs. (5.66) and (5.67).

$$f_{D\,i+1} = f_{D\,i} + \Delta f_{D\,i} = f_{D\,i} + c_i\Delta\dot{u}_i \tag{5.81}$$

$$f_{S\,i+1} = f_{S\,i} + \Delta f_{S\,i} = f_{S\,i} + k_i\Delta u_i \tag{5.82}$$

Table 5.4 shows a schematic summary of the Newmark method for SDOF nonlinear systems, whereas Table 5.5 shows the procedure of the Newton-Raphson iterative procedure.

Table 5.4: Summary of the Newmark method for SDOF nonlinear systems[4]

Special cases:

(1) Average acceleration method ($\gamma = \frac{1}{2},\ \beta = \frac{1}{4}$)

(2) Linear acceleration method ($\gamma = \frac{1}{2},\ \beta = \frac{1}{6}$)

1. Initial calculations

1.1 Select the time step size (Δt)

1.2 Determine the initial acceleration ($\ddot{u}_0$)

$$\ddot{u}_0 = \frac{p_0 - f_{D_0} - f_{S_0}}{m}$$

2. Calculations for each time step i = 1,2,3, …

2.1 Determine the load increment (Δp_i)

$$\Delta p_i = p_{i+1} - p_i$$

2.2 Determine the tangent stiffness (k_i) and damping (c_i)

$$k_i = \frac{\partial f_S}{\partial u}(u_i, \dot{u}_i)\ ; \qquad c_i = \frac{\partial f_D}{\partial \dot{u}}(u_i, \dot{u}_i)$$

2.3 Determine the effective stiffness coefficient ($\hat{k}_i$)

$$\hat{k}_i = k_i + \frac{\gamma}{\beta\ \Delta t} c_i + \frac{1}{\beta\ (\Delta t)^2} m$$

2.4 Determine the effective load increment ($\Delta \hat{p}_i$)

$$\Delta \hat{p}_i = \Delta p_i + \left(\frac{1}{\beta\ \Delta t} m + \frac{\gamma}{\beta} c_i\right) \dot{u}_i + \left[\frac{1}{2\beta} m + \left(\frac{\gamma}{2\beta} - 1\right) \Delta t\ c_i\right] \ddot{u}_i$$

2.5 Determine the displacement increment (Δu_i) using modified Newton-Raphson iterative procedure to solve the following equation (Table 5.5)

$$\hat{k}_i\ \Delta u_i = \Delta \hat{p}_i$$

2.6 Determine the velocity increment ($\Delta \dot{u}_i$)

$$\Delta \dot{u}_i = \frac{\gamma}{\beta\ \Delta t} \Delta u_i - \frac{\gamma}{\beta} \dot{u}_i + \left(1 - \frac{\gamma}{2\beta}\right) \Delta t\ \ddot{u}_i$$

2.7 Determine the displacement (u_{i+1}) and velocity ($\dot{u}_{i+1}$) of time step i+1

$$u_{i+1} = u_i + \Delta u_i\ ; \qquad \dot{u}_{i+1} = \dot{u}_i + \Delta \dot{u}_i$$

2.8 Determine the internal restoring forces ($f_{D_{i+1}},\ f_{S_{i+1}}$) of time step i+1

$$f_{D_{i+1}} = f_{D_i} + c_i \Delta \dot{u}_i\ ; \qquad f_{S_{i+1}} = f_{S_i} + k_i \Delta u_i$$

2.9 Determine the acceleration of time step i+1 ($\ddot{u}_{i+1}$)

$$\ddot{u}_{i+1} = \frac{p_{i+1} - f_{D_{i+1}} - f_{S_{i+1}}}{m}$$

[4] Based on Table 5.7.2 of Chopra (2007)

Table 5.5: Summary of the Modified Newton-Raphson iterative procedure

1. Initial calculations

   1.1 Initial estimation of the displacement for the time step i+1 ($u_{i+1}^{(0)}$)

   $$u_{i+1}^{(0)} = u_i$$

   1.2 Initial estimation of the nonlinear stiffness ($f_{S_{i+1}}^{(0)}$) and damping ($f_{D_{i+1}}^{(0)}$) force for time step i+1

   $$f_{S_{i+1}}^{(0)} = f_{S_i} \qquad f_{D_{i+1}}^{(0)} = f_{D_i}$$

   1.3 Initial residual force for iteration 1 ($\Delta R^{(1)}$)

   $$\Delta R^{(1)} = \Delta \hat{p}_i$$

2. Calculations for each iteration j = 1, 2, 3, …

   2.1 Solve the equation to determine the displacement increment ($du^{(j)}$) between iteration j and iteration j-1

   $$\hat{k}_i \, du^{(j)} = \Delta R^{(j)}$$

   2.2 Determine the total displacement of iteration j for time step i+1 ($u_{i+1}^{(j)}$)

   $$u_{i+1}^{(j)} = u_{i+1}^{(j-1)} + \, du^{(j)}$$

   2.3 Determine the total displacement increment of iteration j between time steps i and i+1 ($\Delta u_i^{(j)}$)

   $$\Delta u_i^{(j)} = u_{i+1}^{(j)} - u_i$$

   2.4 Determine the total velocity increment of iteration j between time steps i and i+1 ($\Delta \dot{u}_i^{(j)}$)

   $$\Delta \dot{u}_i^{(j)} = \frac{\gamma}{\beta \, \Delta t} \Delta u_i^{(j)} - \frac{\gamma}{\beta} \dot{u}_i + \left(1 - \frac{\gamma}{2\beta}\right) \Delta t \, \ddot{u}_i$$

   2.5 Determine the total velocity of iteration j for time step i+1 ($\dot{u}_{i+1}^{(j)}$)

   $$\dot{u}_{i+1}^{(j)} = \dot{u}_i + \Delta \dot{u}_i^{(j)}$$

   2.6 Determine the internal restoring forces of iteration j for time step i+1 ($f_{D_{i+1}}^{(j)}$, $f_{S_{i+1}}^{(j)}$) based on the nonlinear relations force, displacement and velocity

   2.7 Determine the variation of the external force after iteration j ($\Delta f_{ext}^{(j)}$)

   $$\Delta f_{ext}^{(j)} = \begin{cases} \Delta p_i + \dfrac{1}{\beta \, \Delta t} m \dot{u}_i + \dfrac{1}{2\beta} m \ddot{u}_i \,, & j = 1 \\ \Delta R^{(j)} \,, & j \geq 2 \end{cases}$$

   2.8 Determine the variation of the internal force after iteration j ($\Delta f_{int}^{(j)}$)

   $$\Delta f_{int}^{(j)} = \left(f_{S_{i+1}}^{(j)} - f_{S_{i+1}}^{(j-1)}\right) + \left(f_{D_{i+1}}^{(j)} - f_{D_{i+1}}^{(j-1)}\right) + \frac{1}{\beta \, (\Delta t)^2} m \, du^{(j)}$$

   2.9 Determine the residual force for the next iteration j+1 ($\Delta R^{(j+1)}$)

   $$\Delta R^{(j+1)} = \Delta f_{ext}^{(j)} - \Delta f_{int}^{(j)}$$

5.3 HFTD method numerical test examples for SDOF systems

The HFTD (Tables 5.1 and 5.2), the Direct Integration (Table 5.3) and the Newmark (Tables 5.4 and 5.5) methods presented in Sections 5.1 and 5.2, were programmed in MATLAB[5] (The Mathworks, Inc., 2012) to be used in the solution of the examples developed in this section.

5.3.1 Test examples for SDOF linear systems

Four examples of SDOF linear systems are solved with the HFTD method and compared with the analytic solution for harmonic loading or the Direct Integration method for general loading. Each of these examples represents different types of loading that cover a wide range of possibilities.

5.3.1.1 Harmonic loading[6]

The SDOF system is subject to the harmonic load $p(t) = 10\cos(10t)$. The numerical data defining the characteristics of the system is shown in Table 5.6.

Table 5.6: Numerical data of Example 4.2 (Craig and Kurdila, 2006)

k	m	ξ	p_0	Ω	t_0	u_0	v_0
40	0.1	0.2	10	10	2	0	0

Based on the information given in Table 5.6, the parameters of the SDOF system shown in Table 5.7 are determined. The analytical solution for harmonic loading developed in section 2.1 is used to solve this example. Table 5.8 shows the displacement amplitude (U), phase angle (α) and constants A_1 and A_2, which are obtained using Eqs. (5.40) to (5.43), as well as the information shown in Tables 5.6 and 5.7.

Table 5.7: Parameters of the SDOF linear system subject to harmonic loading

T_r	T_1	ω_o	ω_d
2	0.314	20	19.60

Table 5.8: Analytic solution of SDOF linear system subject to harmonic loading

U	α	A_1	A_2
0.322	0.261	-0.311	-0.1059

Finally, Eqs. (5.83) and (5.84) present the analytic displacement and velocity, obtained from Eqs. (5.42) and (5.43), respectively. The acceleration response is indirectly determined by replacing the displacement and velocity values into Eq. (5.36).

$$u(t) = 0.322\cos(10t - 0.261) - e^{-4t}(0.311\cos(19.60t) + 0.1059\sin(19.60t)) \quad \textbf{(5.83)}$$

[5] The MATLAB routines can be found in Appendix B.
[6] Example 4.2 of Craig and Kurdila (2006).

$$\dot{u}(t) = -3.22\sin(10t - 0.261) + e^{-4t}[(-2.0756 + 1.244)\cos(19.60t) - (-6.0956 - 0.4236)\sin(19.60t)] \quad \textbf{(5.84)}$$

On the other hand, Table 5.9 shows the values of the parameters required to perform the HFTD method schematized in Tables 5.1 and 5.2. It is important to mention that for linear systems the HFTD method does not require segmentation approach or iterative procedure. For that reason only one time segment (N_T =1) is used in all the examples corresponding to linear systems.

Table 5.9: HFTD parameters for the SDOF linear system subject to harmonic loading

Δt	T_p	N	$\omega_n = \frac{2\pi n}{T_p}$	N_T
0.02	15.72	786	0.0636 n	1

The comparison between the analytic and the HFTD solutions for the displacement, velocity and acceleration is presented in Figs. 5.5 to 5.7.

For the displacement and velocity responses, the major differences of the HFTD solution with respect to the analytical solution are found during the first 0.5 s. For the acceleration, this range is extended until the time 0.75 s. After this initial period, the HFTD response gets stable and matches almost exactly the analytical solution thenceforth.

The reason for this initial inaccurate range is that the magnitude of the cosine harmonic load does not start in zero, but in p_0 equal to 10. The HFTD is not capable to exactly trace the response generated by this sudden loading at the initial time t=0. Once the effect of the sudden initial loading is diminished, the response obtained with the HFTD method gradually matches the analytical solution.

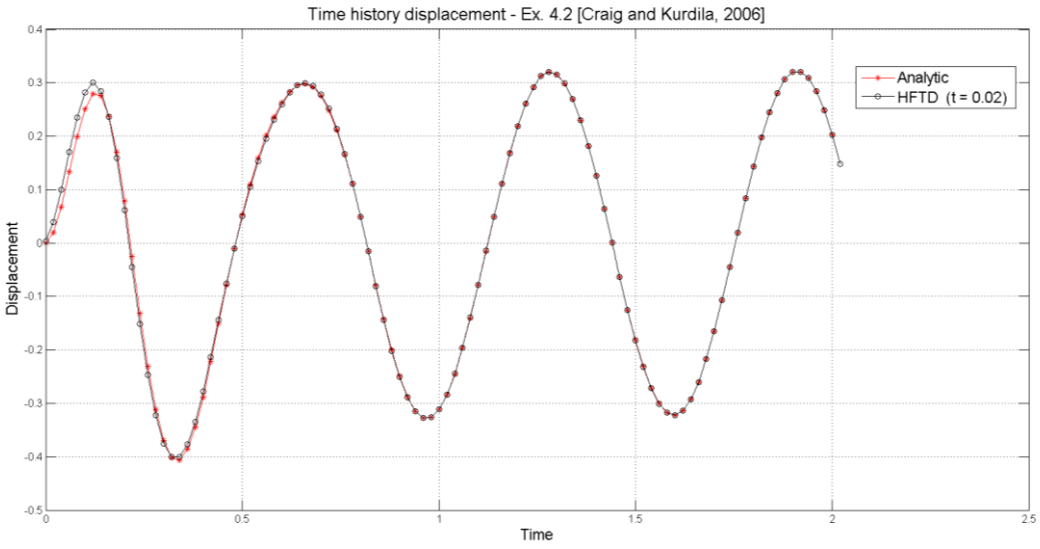


Figure 5.5: Harmonic loading example. Displacement response for Analytic and HFTD solutions.

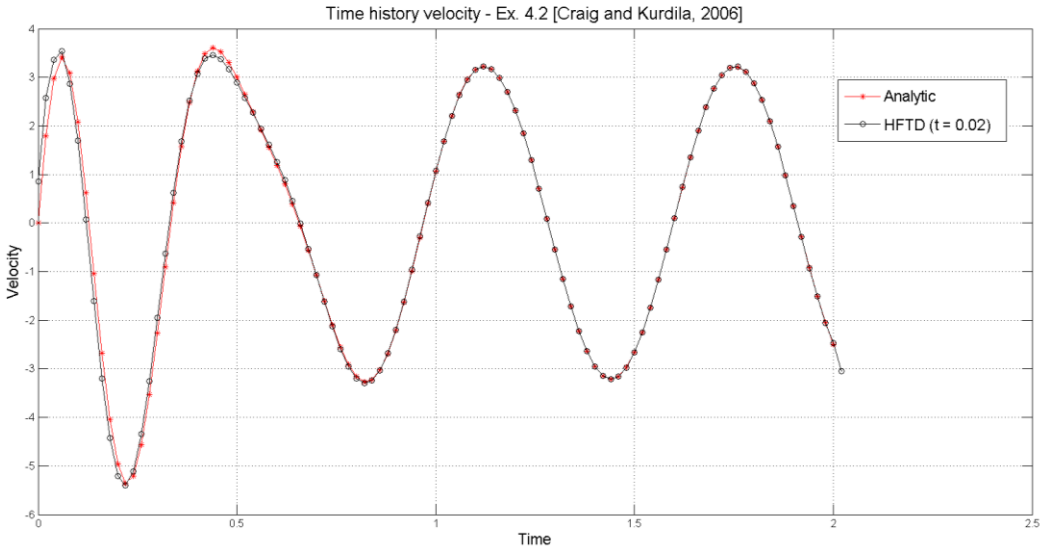


Figure 5.6: Harmonic loading example. Velocity response for Analytic and HFTD solutions.

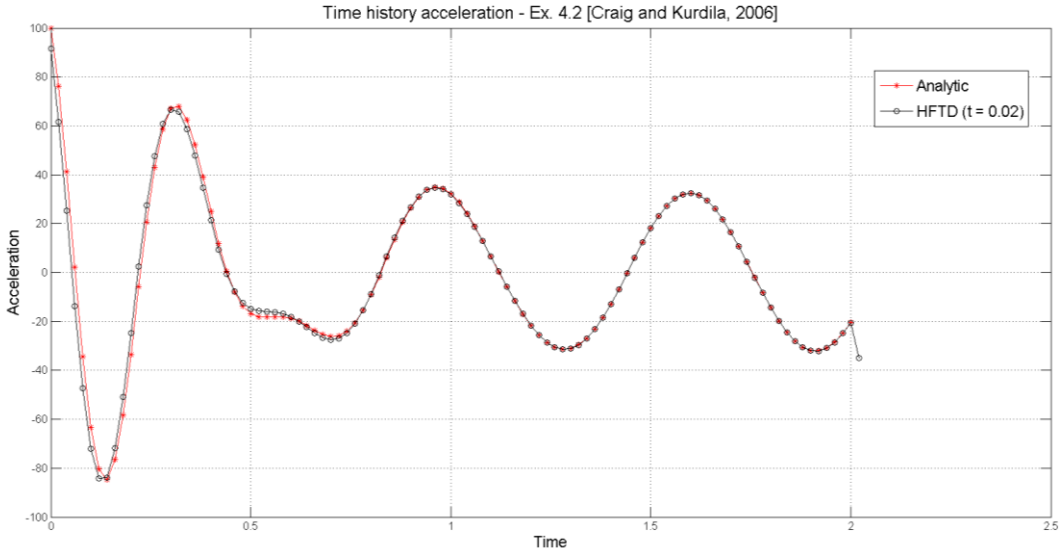


Figure 5.7: Harmonic loading example. Acceleration response for Analytic and HFTD solutions.

5.3.1.2 Blast loading[7]

The SDOF system is subject to the blast load schematized in Fig. 5.8. The numerical data defining the characteristics of the system is shown in Table 5.10.

[7] Example 4.2 of Paz and Leigh (2004).

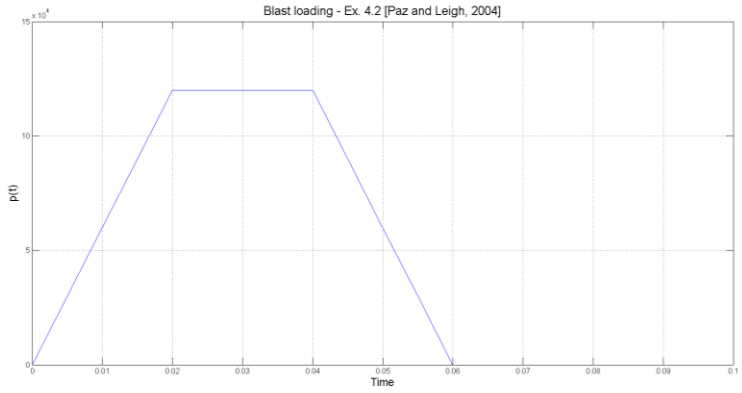


Figure 5.8: Blast loading.

Table 5.10: Numerical data of Example 4.2 (Paz and Leigh, 2004)

k	m	ξ	t_0	u_0	v_0
100 000	100	0.2	0.06	0	0

Based on the information given in Table 5.10, the parameters of the SDOF system shown in Table 5.11 are determined. The recurrence formulas of the Direct Integration method developed in Section 5.2.1.3 are used to calculate the exact analytic response of the SDOF linear system. Table 5.12 shows the values of the coefficients for the recurrence formulas of the Direct Integration method. They were calculated based on the information given in Tables 5.10 and 5.11, as well as the expressions of Table 5.3.

Table 5.11: Parameters of the SDOF linear system subject to blast loading

T_r	T_1	ω_o	ω_d
0.20	0.20	31.62	30.92

Table 5.12: Direct Integration method coefficients for the SDOF linear system subject to blast loading

Δt	β	A	B	C	D	A′	B′	C′	D′
0.005	6.32	$8.1\text{x}10^{-8}$	$4.1\text{x}10^{-8}$	0.9878	0.0048	$2.4\text{x}10^{-5}$	$2.4\text{x}10^{-5}$	-4.825	0.9268

On the other hand, Table 5.13 shows the values of the parameters required to perform the HFTD method (Tables 5.1 and 5.2). Similarly as in the previous example, for linear analysis the segmentation approach is not required (only one time segment is required to be defined).

Table 5.13: HFTD parameters for the SDOF linear system subject to blast loading

Δt	T_p	N	$\omega_n = \frac{2\pi n}{T_p}$	N_T
0.005	9.935	1987	$0.1007\ n$	1

The comparison between the analytic Direct Integration and the HFTD solutions for the displacement, velocity and acceleration is presented in Figs. 5.9 to 5.11. The accuracy obtained in the three responses is remarkable. The variations are so small that are almost no perceptible. This means that when the loading starts in zero (see Fig. 5.8) the HFTD method shows a great performance.

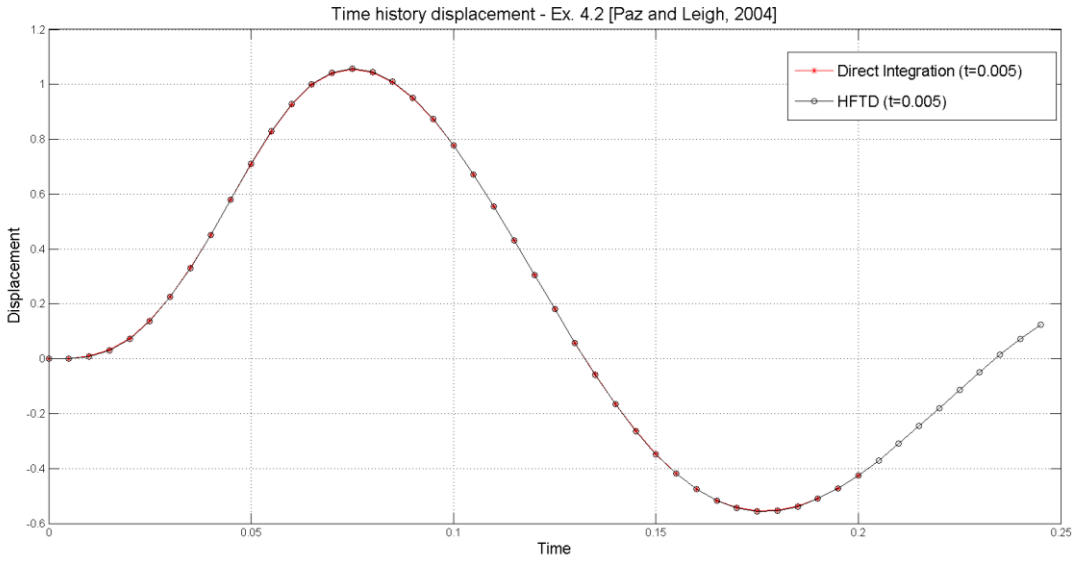


Figure 5.9: Blast loading example. Displacement response for Direct Integration and HFTD solutions.

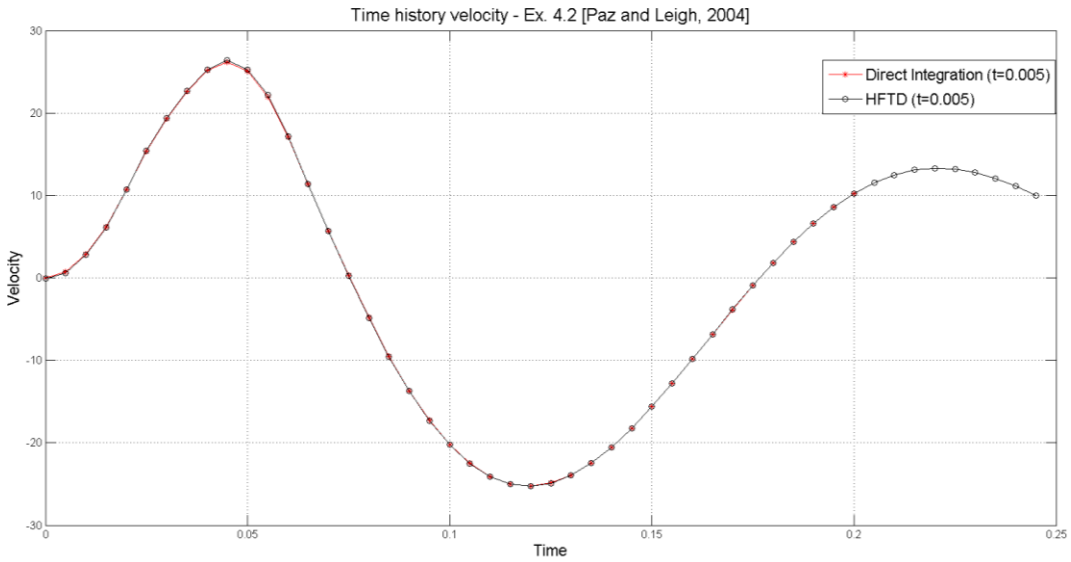


Figure 5.10: Blast loading example. Velocity response for Direct Integration and HFTD solutions.

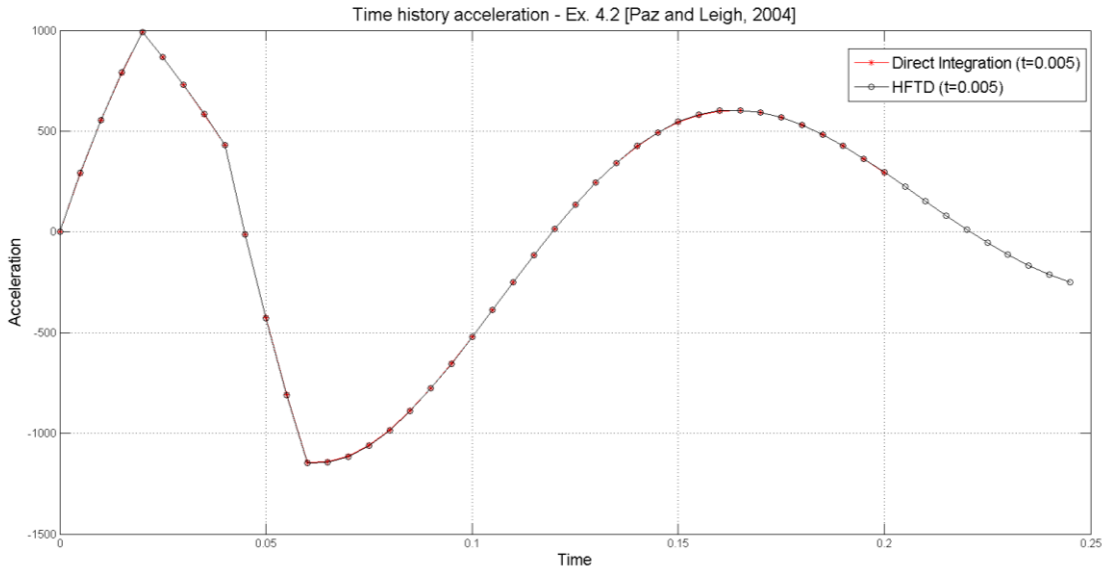


Figure 5.11: Blast loading example. Acceleration response for Direct Integration and HFTD solutions.

5.3.1.3 Impulsive loading[8]

The SDOF system is subject to the impulse load schematized in Fig. 5.12. The numerical data defining the characteristics of the system is shown in Table 5.14.

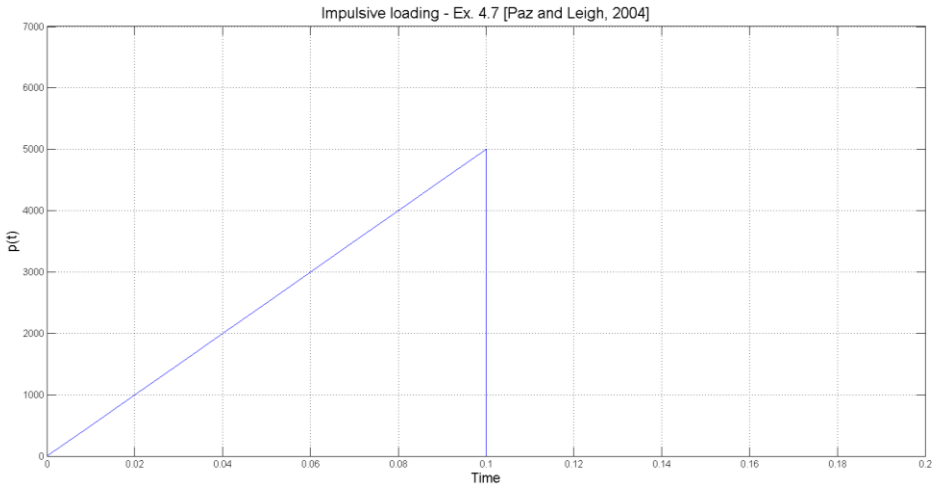


Figure 5.12: Impulsive loading.

[8] Example 4.7 of Paz and Leigh (2004).

Table 5.14: Numerical data of Example 4.7 (Paz and Leigh, 2004)

k	m	ξ	t_0	u_0	v_0
10 000	10	0.1	0.10	0	0

Based on the information given in Table 5.14, the parameters of the SDOF system shown in Table 5.15 are determined. Table 5.16 shows the values of the coefficients for the recurrence formulas of the Direct Integration method [Eqs. (5.57) to (5.59)].

Table 5.15: Parameters of the SDOF linear system subject to impulsive loading

T_r	T_1	ω_o	ω_d
0.22	0.20	31.62	31.46

Table 5.16: Direct Integration method coefficients for the SDOF linear system subject to impulsive loading

Δt	β	A	B	C	D	A′	B′	C′	D′
0.01	3.16	$3.2\text{x}10^{-6}$	$1.6\text{x}10^{-6}$	0.9514	0.0095	$4.7\text{x}10^{-4}$	$4.9\text{x}10^{-4}$	-9.53	0.8912

On the other hand, Table 5.17 shows the values of the parameters required to perform the HFTD method. Similarly as in the previous example, for linear analysis only one time segment is required.

Table 5.17: HFTD parameters for the SDOF linear system subject to impulsive loading

Δt	T_p	N	$\omega_n = \frac{2\pi n}{T_p}$	N_T
0.01	9.94	994	0.1006 n	1

The comparison between the analytic Direct Integration and the HFTD solutions for the displacement, velocity and acceleration is presented in Figs. 5.13 to 5.15. The accuracy obtained in the three responses is as good as in the previous example.

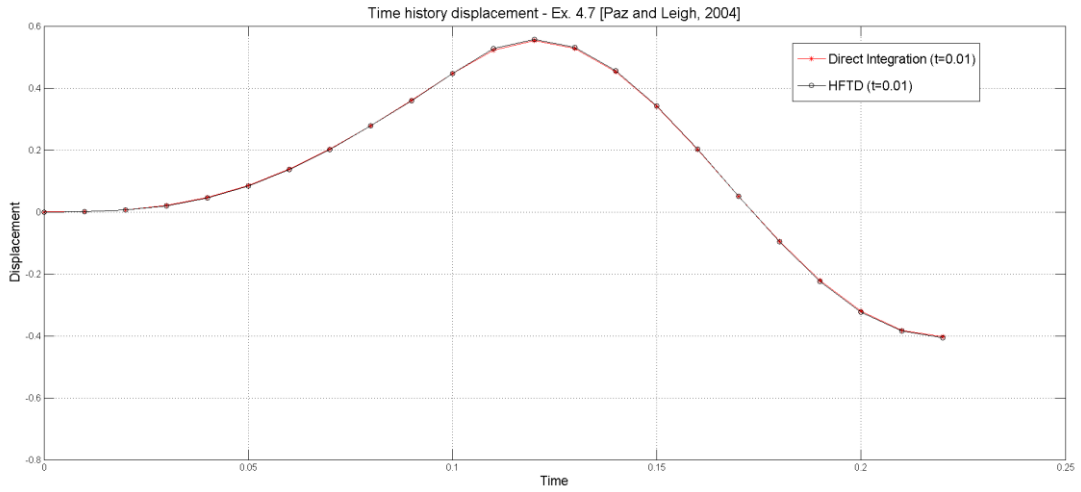


Figure 5.13: Impulsive loading example. Displacement response for Direct Integration and HFTD solutions.

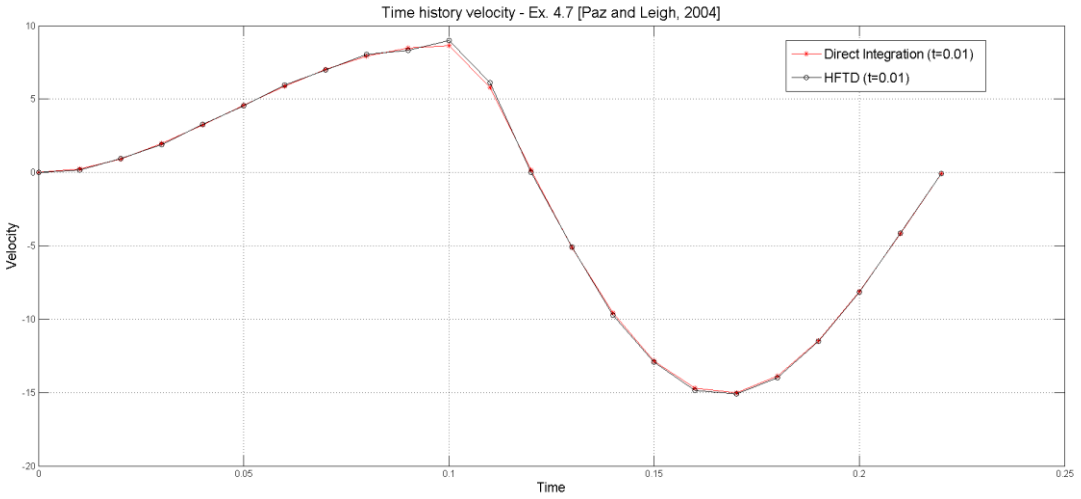


Figure 5.14: Impulsive loading example. Velocity response for Direct Integration and HFTD solutions.

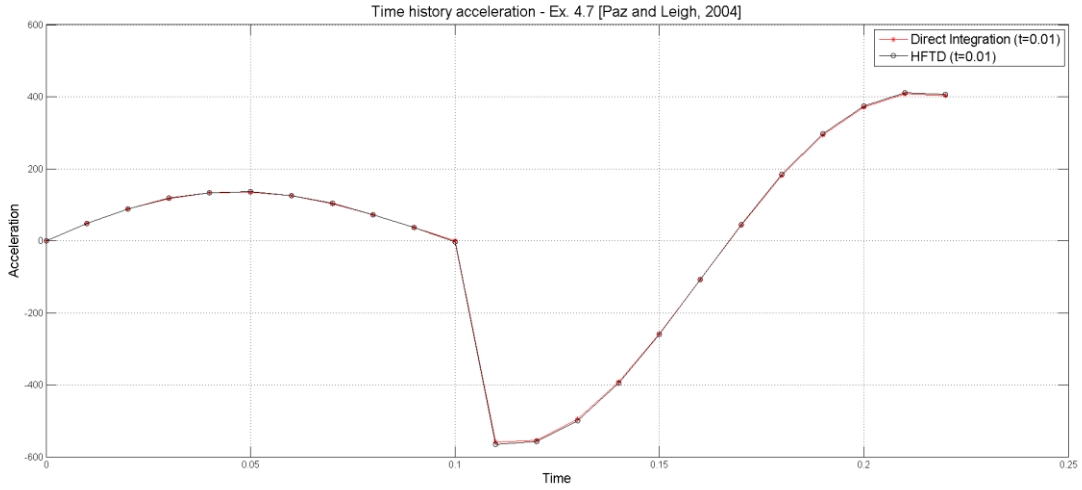


Figure 5.15: Impulsive loading example. Acceleration response for Direct Integration and HFTD solutions.

5.3.1.4 Ground acceleration record loading[9]

The SDOF system is subject to the ground acceleration correspondent to the North-South component of the El Centro earthquake shown in Fig. 5.16. The numerical data defining the characteristics of the system is shown in Table 5.18.

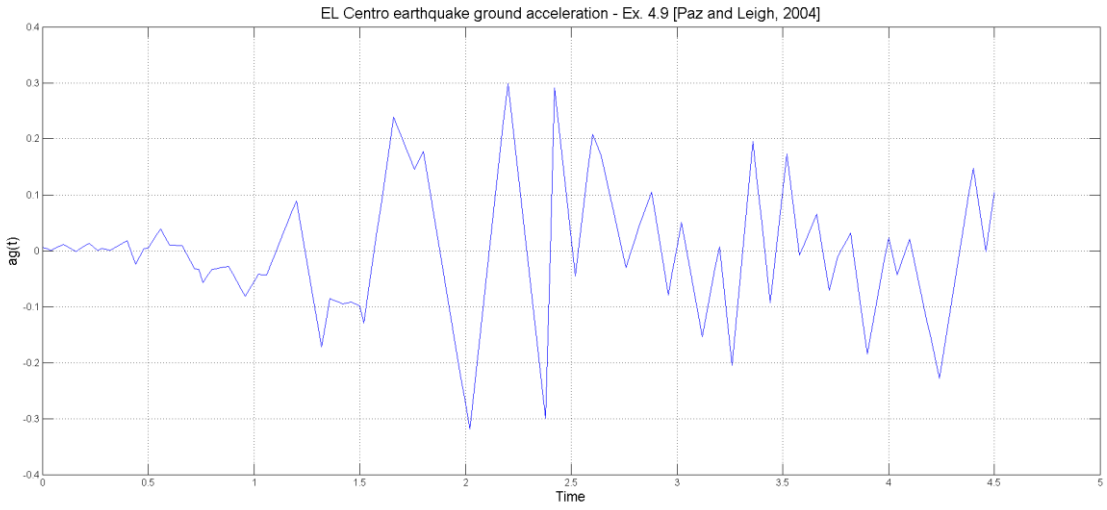


Figure 5.16: El Centro N-S earthquake ground acceleration.

[9] Example 4.9 of Paz and Leigh (2004).

Table 5.18: Numerical data of Example 4.9 (Paz and Leigh, 2004)

K	m	ξ	t_0	u_0	v_0
2136	38.86	0.05	4.5	0	0

Based on the information given in Table 5.18, the parameters of the SDOF system shown in Table 5.19 are determined. Table 5.20 shows the values of the coefficients for the recurrence formulas of the Direct Integration method. On the other hand, Table 5.21 shows the values of the parameters required to perform the HFTD method.

Table 5.19: Parameters of the SDOF linear system subject to earthquake ground acceleration

T_r	T_1	ω_o	ω_d
4.38	0.85	7.414	7.405

Table 5.20: Direct Integration method coefficients for the SDOF linear system subject to earthquake ground acceleration

Δt	β	A	B	C	D	A′	B′	C′	D′
0.02	0.37	$3.4\text{x}10^{-6}$	$1.7\text{x}10^{-6}$	0.9891	0.0198	$2.5\text{x}10^{-4}$	$2.5\text{x}10^{-4}$	-1.09	0.9744

Table 5.21: HFTD parameters for the SDOF linear system subject to earthquake ground acceleration

Δt	T_p	N	$\omega_n = \frac{2\pi n}{T_p}$	N_T
0.02	42.38	2119	$0.0236\ n$	1

The comparison between the analytic Direct Integration and the HFTD solutions for the displacement, velocity and acceleration is presented in Figs. 5.17 to 5.19. The accuracy obtained in the three responses is as good as in the previous example.

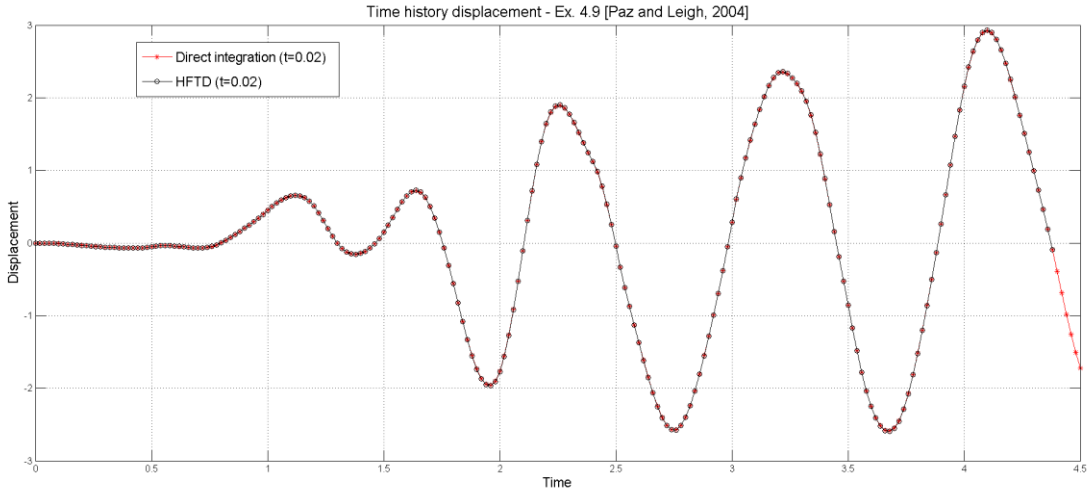


Figure 5.17: Ground acceleration loading example. Displacement response for Direct Integration and HFTD solutions.

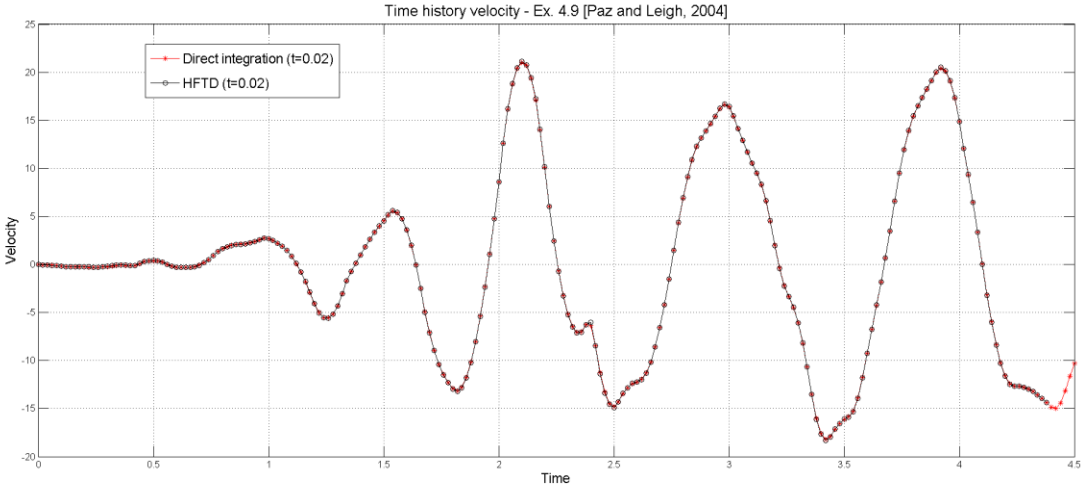


Figure 5.18: Ground acceleration loading example. Velocity response for Direct Integration and HFTD solutions.

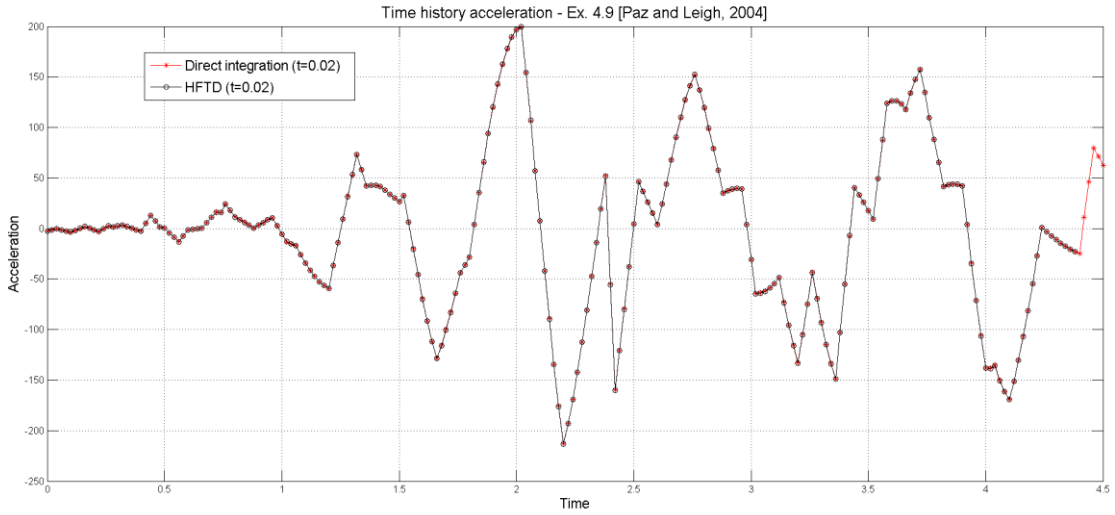


Figure 5.19: Ground acceleration loading example. Acceleration response for Direct Integration and HFTD solutions.

5.3.2 Resonance case study: SDOF linear systems subject to harmonic loading with exponential amplitude

The SDOF linear system is subject to a harmonic sinusoidal load which amplitude varies exponentially, as expressed by Eq. (5.85). In order to test the stability of the HFTD method, the influence of load excitation frequencies close to the resonance condition ($\Omega = \omega_o$) is studied for systems with low damping ratios ($\xi \leq 0.05$)[10]. In addition, the loading duration (t_0) is expressed in Eq. (5.86) as a number "n" of load excitation periods (T) or peaks.

$$p(t) = P_o\, e^{-(t/t_0 - 0.5)^2} \sin(\Omega t) \qquad \textbf{(5.85)}$$

$$t_0 = nT = n\left(\frac{2\pi}{\Omega}\right) \qquad \textbf{(5.86)}$$

The numerical data of the SDOF system to be studied is shown in Table 5.22. As it can be noticed, three different low damping ratios are taken into account.

Table 5.22: Numerical data of the SDOF system for resonance case studies

k	m	ω_o	ξ	u_0	v_0
2136	38.86	7.414	(0.01 ; 0.03 ; 0.05)	0	0

First, four load excitation frequencies close to the natural vibration frequency of the system are considered with a damping ratio equal to 0.05. The characteristics of these four case studies are summarized in Table 5.23.

[10] As explained in Section 4.3, the HFTD method is not applicable to undamped systems.

Table 5.23: Numerical data of case study A: Excitation frequencies close to resonance

Case	ξ	Ω	T	P_o	n	t_0
A-1	0.05	$0.80\omega_o$	1.06	2500	25	26.49
A-2	0.05	$0.95\omega_o$	0.89	2500	25	22.31
A-3	0.05	$1.05\omega_o$	0.81	2500	25	20.19
A-4	0.05	$1.20\omega_o$	0.71	2500	25	17.65

The harmonic load $p(t)$ given by Eq. (5.85) is shown in Fig. 5.20 as example for the particular case A-4. Fig. 5.20 shows the maximum amplitude P_o, the number of peaks or load periods n and the load duration t_0. Similar graphics can be presented for the other three cases.

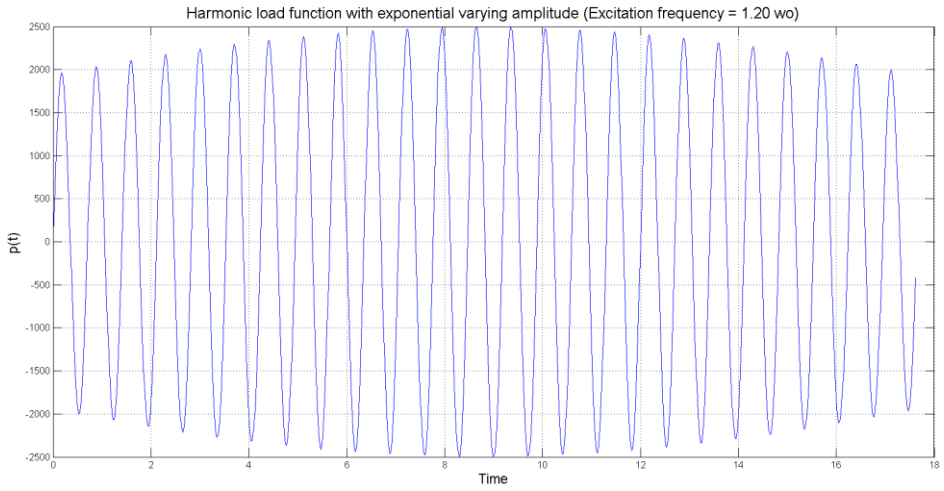


Figure 5.20: Harmonic sinusoidal load with exponential amplitude for the case study A-4.

Like in the previous examples, the results obtained with the HFTD method (Tables 5.1 and 5.2) are compared with the analytical solution provided by the Direct Integration method described in Section 5.2.1.3, and summarized by Eqs. (5.57) to (5.59) and Table 5.3.

The HFTD parameters for each of the four case studies are summarized in Table 5.24. Due to lack of space, only the results obtained for the displacement response of the SDOF linear system are presented and compared with the Direct Integration method. The analysis of velocity and acceleration results revealed as good performance as the displacement results, presented in Figs. 5.21 to 5.24 for the cases A-1, A-2, A-3 and A-4, respectively.

Table 5.24: HFTD parameters for case study A

Case	Δt	T_p	N	$\omega_n = \frac{2\pi n}{T_p}$	N_T
A-1	0.025	53.00	2120	$0.0189\ n$	1
A-2	0.025	44.68	1787	$0.0224\ n$	1
A-3	0.025	42.38	1695	$0.0236\ n$	1
A-4	0.025	42.38	1695	$0.0236\ n$	1

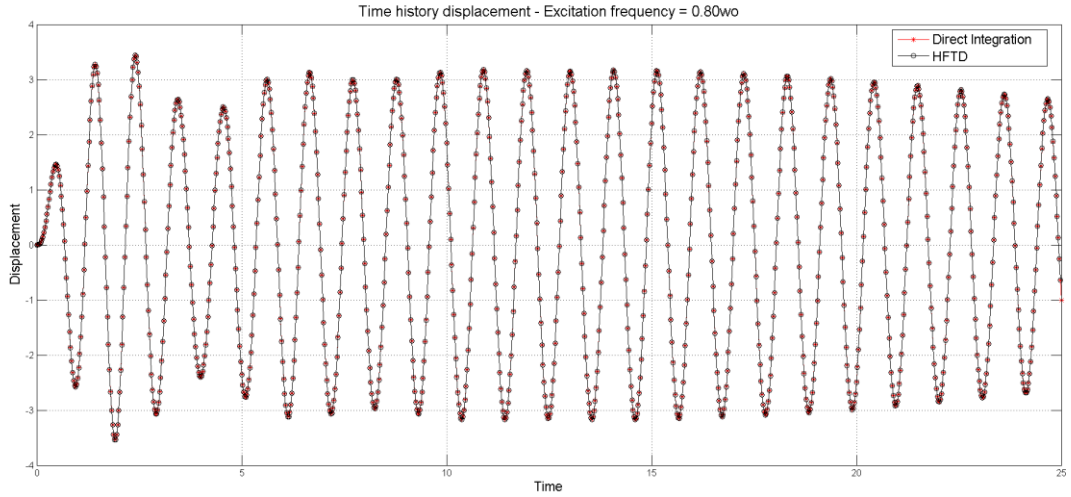


Figure 5.21: Resonance case study A-1. Displacement response for Direct Integration and HFTD solutions.

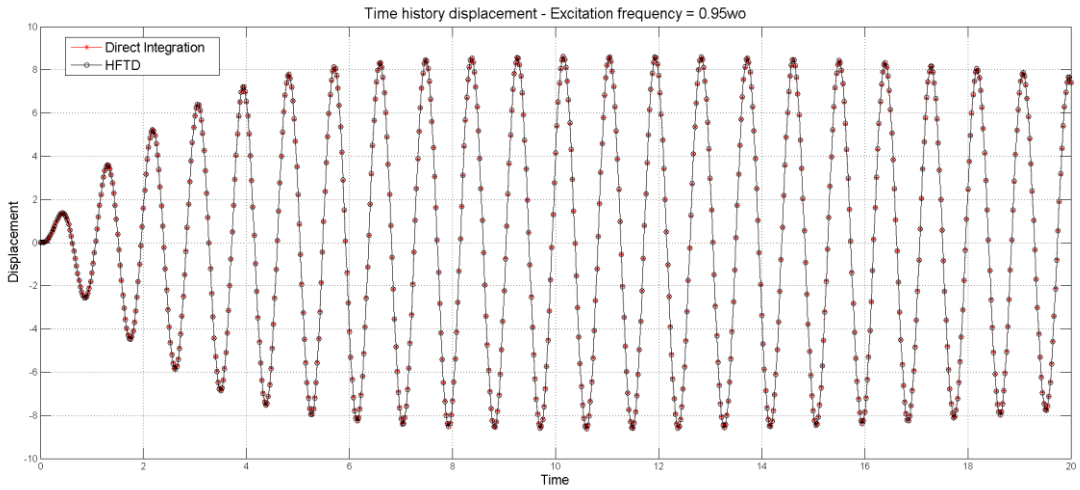


Figure 5.22: Resonance case study A-2. Displacement response for Direct Integration and HFTD solutions.

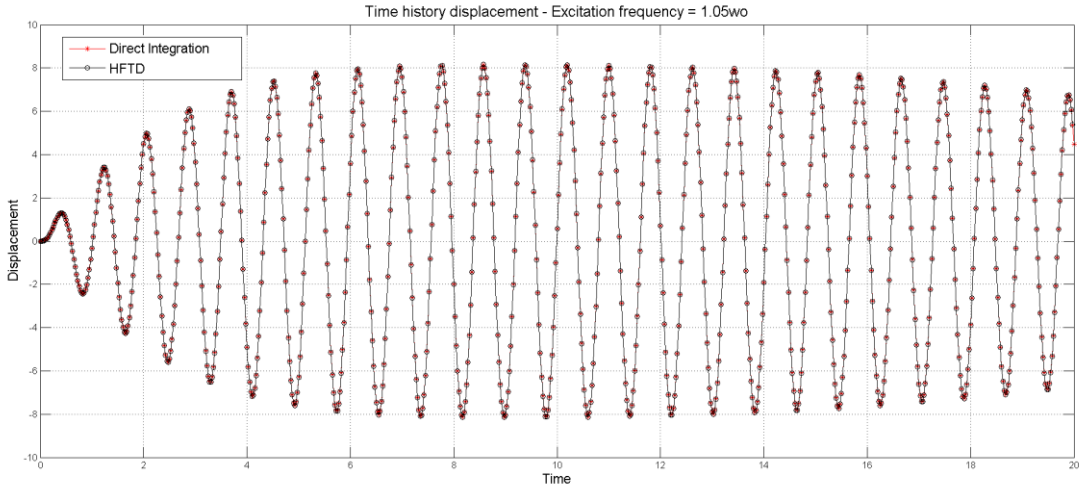


Figure 5.23: Resonance case study A-3. Displacement response for Direct Integration and HFTD solutions.

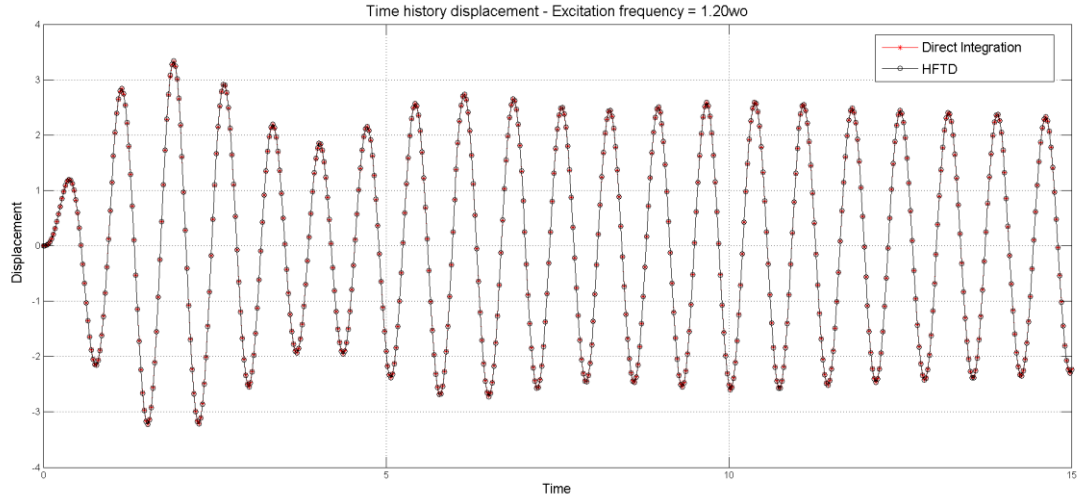


Figure 5.24: Resonance case study A-4. Displacement response for Direct Integration and HFTD solutions.

From the analysis of Figs. 5.21 to 5.24, the first conclusion is that the accuracy of the HFTD method is not affected due to load excitation frequencies close to natural vibration frequency of the system (resonance condition).

Second, there is a remarkable similarity between the responses of the case studies with excitation frequencies which are symmetric with respect to the resonance frequency. This is the situation of case studies A-1 and A-4 on the one hand, and case studies A-2 and A-3 on the other hand, which time history displacements have similar forms and amplitudes. The only small difference, besides the obvious fact that the responses of the cases with higher load excitation frequencies have a higher

number of peaks too, is that the amplitudes of the cases with excitation frequencies below resonance are slightly greater than its correspondent symmetric cases above resonance.

Finally, it is remarkable the increase of the displacement magnitude for the case studies closer to the resonance frequency. The maximum displacement magnitude increases from 3 (A-1, A-4) to 8 (A-2, A-3), approximately.

Then, based on the good performance shown by the HFTD method in the previous case study, more critical situations of very lightly damped SDOF linear systems close to resonance condition are considered. Table 5.25 shows three additional case studies with load excitation frequencies equal to the resonance frequency and three low damping ratios equal or smaller than 0.05.

Table 5.25: Numerical data of case study B: Resonance excitation frequency with very low damping ratios

Case	ξ	Ω	T	P_o	n	t_0
B-1	0.05	ω_o	0.85	2500	25	21.19
B-2	0.03	ω_o	0.85	2500	25	21.19
B-3	0.01	ω_o	0.85	2500	25	21.19

The HFTD parameters for each of the three case studies are summarized in Table 5.26. Similarly than in case study A, Figs. 5.25 to 5.27 only present the results obtained for displacements, which are compared with the Direct Integration method response. The analysis of velocity and acceleration results revealed as good performance as displacement results.

Table 5.26: HFTD parameters for case study B

Case	Δt	T_p	N	$\omega_n = \frac{2\pi n}{T_p}$	N_T
B-1	0.025	42.43	1697	0.0236 n	1
B-2	0.025	42.43	1697	0.0236 n	1
B-3	0.025	84.78	3391	0.0118 n	1

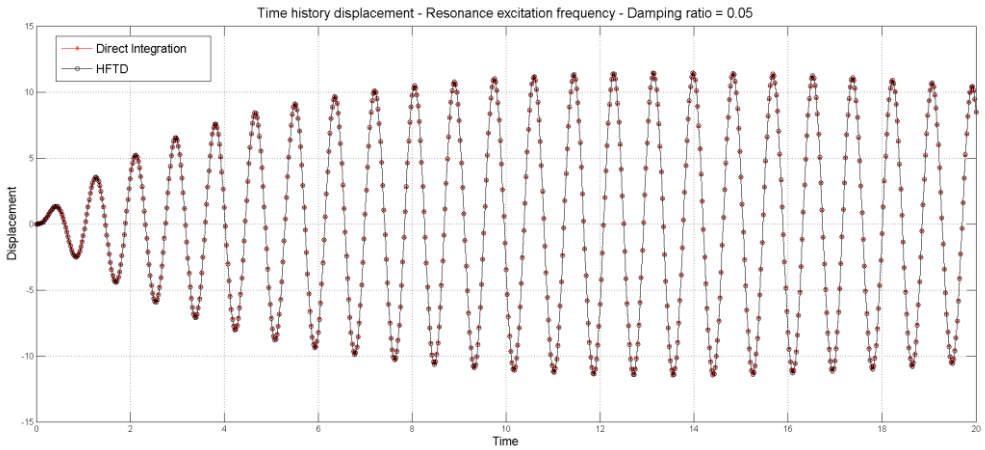


Figure 5.25: Resonance case study B-1. Displacement response for Direct Integration and HFTD solutions.

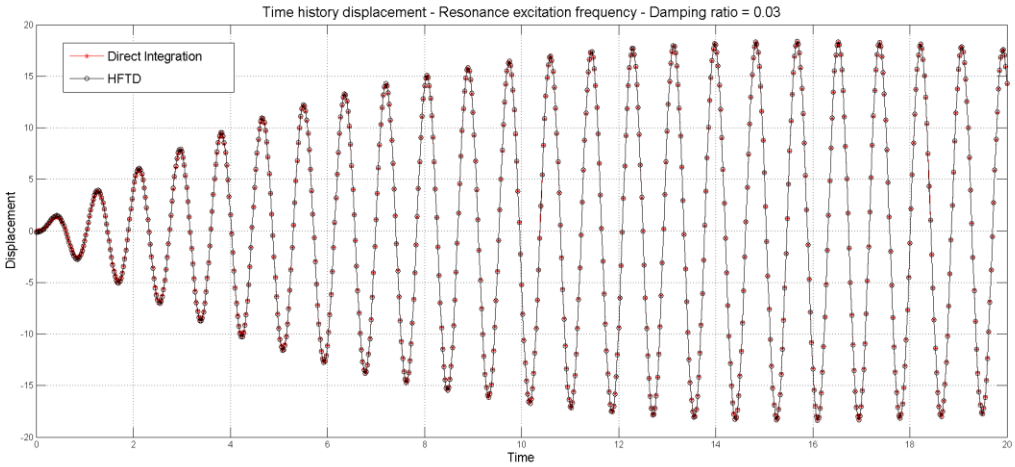


Figure 5.26: Resonance case study B-2. Displacement response for Direct Integration and HFTD solutions.

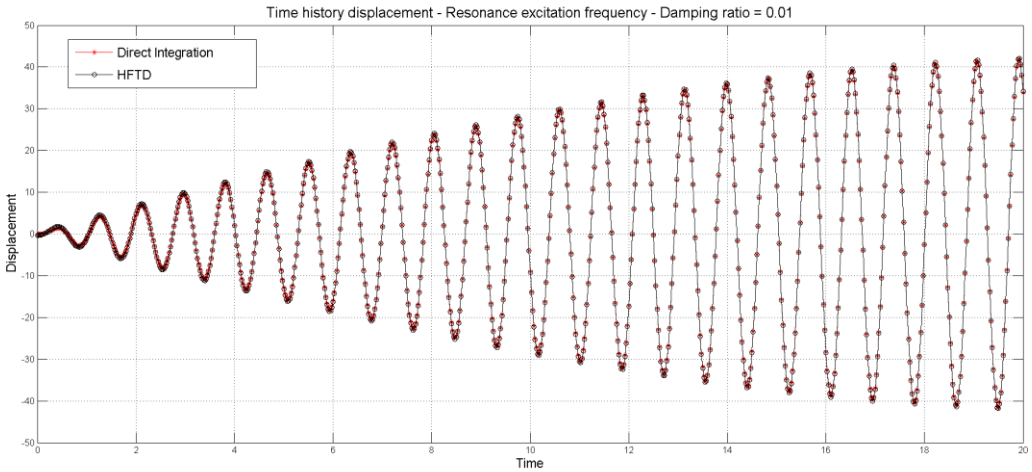


Figure 5.27: Resonance case study B-3. Displacement response for Direct Integration and HFTD solutions.

From the analysis of Figs. 5.25 to 5.27, it is clear that the HFTD method remains accurate with very low damped systems in resonance conditions. In addition, the comparison between Fig. 5.25 (ξ = 0.05) and Fig. 5.27 (ξ = 0.01) shows the notorious effect of damping in the amplitude and stability of motion. The displacement of case study B-3 (system with extremely low damping) growths in time without bound, until reaching a maximum amplitude equivalent to two and four times the maximum amplitude of case studies B-2 and B-1, respectively.

5.3.3 Test examples for SDOF non-linear systems

Three examples of SDOF non-linear systems are solved with the HFTD method (Section 5.1) and compared with the time domain solution of the Newmark method (Section 5.2.2). The first two examples present the response of SDOF systems with two different types of hysteretic stiffness non-linearity. The third test example is equal to the second example with the addition of damping non-linearity.

5.3.3.1 Linear elasto-plastic non-linear stiffness behavior subject to a half cycle sine pulse[11]

The SDOF system described in Table 5.27 is subject to the half cycle sine pulse $p(t) = 10 \sin ({}^{\pi t}/_{0.6})$ shown in Fig. 5.28. The linear elasto-plastic non-linear stiffness force $[f_S(u, \dot{u})]$ of the system is shown in Fig. 5.29. Based on the information given in Table 5.27, the parameters of the SDOF system shown in Table 5.28 are determined.

Table 5.27: Numerical data of Example 5.5 (Chopra, 2007)

k	m	ζ	u_0	v_0	R_t	R_c
10	0.2533	0.05	0	0	7.5	-7.5

Table 5.28: Parameters of the SDOF system with stiffness non-linearity subject to sine pulse loading

T_r	T_1	ω_o	ω_d
0.90	1.0	6.283	6.275

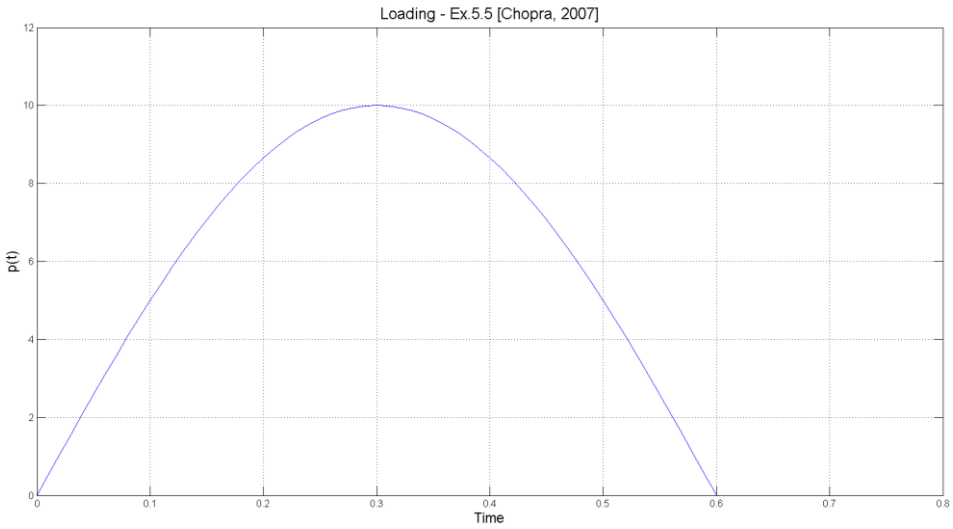


Figure 5.28: Half cycle sine pulse loading.

[11] Example 5.5 of Chopra (2007)

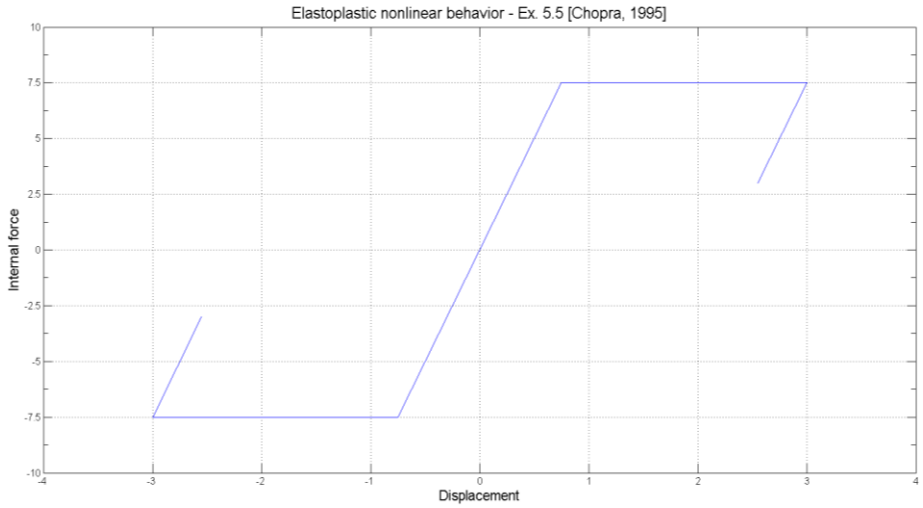


Figure 5.29: Linear elasto-plastic stiffness non-linearity.

Newmark and HFTD methods are applied in the system using two different time steps in order to evaluate the improvement of the HFTD accuracy, when reducing the time step size. For the Newmark method, the Average Acceleration special case was performed, according to the procedure outlined in Tables 5.4 and 5.5. The parameters required for the execution of Newmark method are shown in Table 5.29.

Table 5.29: Newmark method (average acceleration) parameters for the SDOF system with stiffness non-linearity subject to sine pulse loading

γ	β	Δt_1	Δt_2
$^1/_2$	$^1/_4$	0.1	0.01

On the other hand, Table 5.30 shows the values of the parameters required to perform the HFTD method, outlined in Tables 5.1 and 5.2, for each of the two time step sizes taken into account. In this case more than one time segment (N_T) is required in order to achieve convergence. The number of time segments required increases for smaller time steps due to the higher amount of time points and excitation frequencies (N) included in the analysis.

Table 5.30: HFTD parameters for the SDOF system with stiffness non-linearity subject to sine pulse loading

Δt	T_p	N	$\omega_n = \frac{2\pi n}{T_p}$	N_T
0.1	50.00	500	0.02 n	10
0.01	50.00	5000	0.02 n	13

The comparison between the Newmark time domain method and the HFTD method solutions for the displacement, velocity and acceleration is presented in Figs. 5.30 to 5.32. In addition, the non-linear hysteretic relation between the stiffness force and the displacement is presented in Fig. 5.33.

The most remarkable observation is that all the responses obtained using both methods are almost identical when a time step (Δt) equal to 0.01 s is used. On the other hand, when the time step is equal to 0.1 s, the accuracy of both methods diminishes, being possible to identify small drifts with respect to the more accurate solution (Δt = 0.01 s). However this is a shortcoming of the selected time step size, and not of the particular characteristics of the methods applied. This can be clearly appreciated in Fig. 5.33, where the non-linear hysteretic elasto-plastic relation between the stiffness force and the displacement is shown. When an abrupt change of the tangent stiffness is produced, the proper description of the nonlinear relation is only possible when small time steps are used.

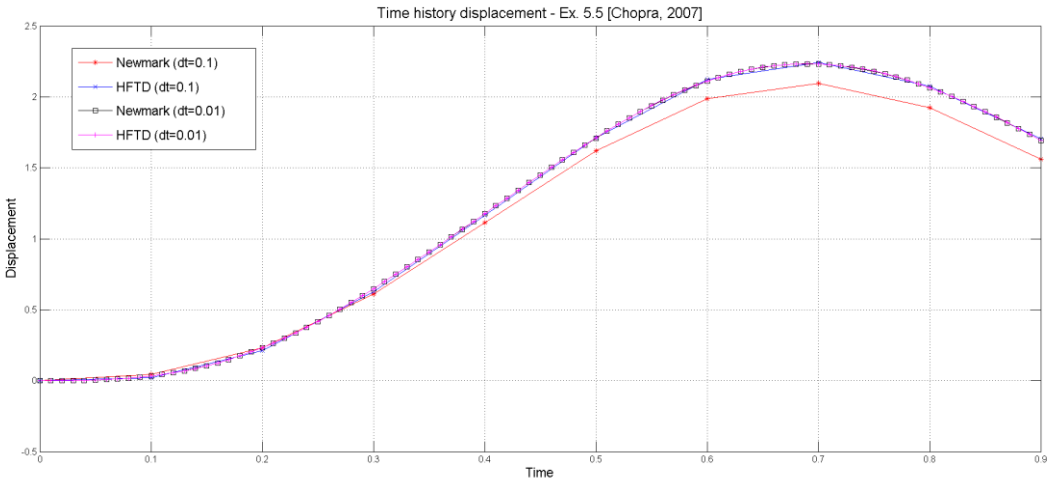


Figure 5.30: Linear elasto-plastic stiffness force non-linearity. Displacement response for Newmark and HFTD solutions.

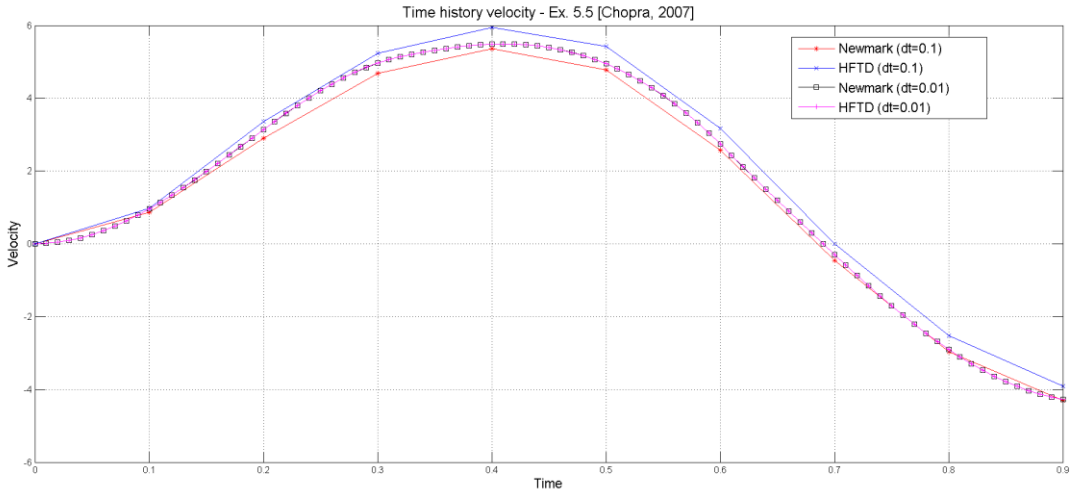


Figure 5.31: Linear elasto-plastic stiffness force non-linearity. Velocity response for Newmark and HFTD solutions.

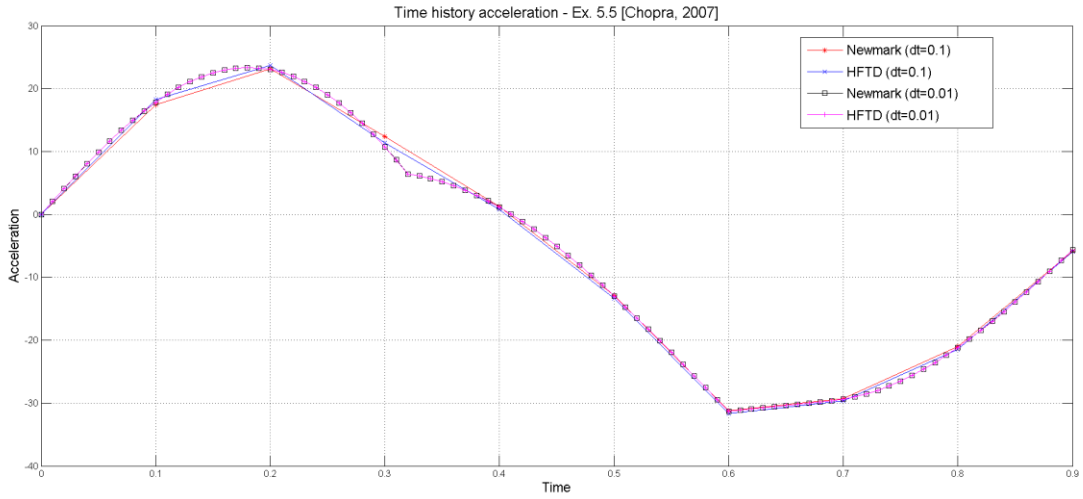


Figure 5.32: Linear elasto-plastic stiffness force non-linearity. Acceleration response for Newmark and HFTD solutions.

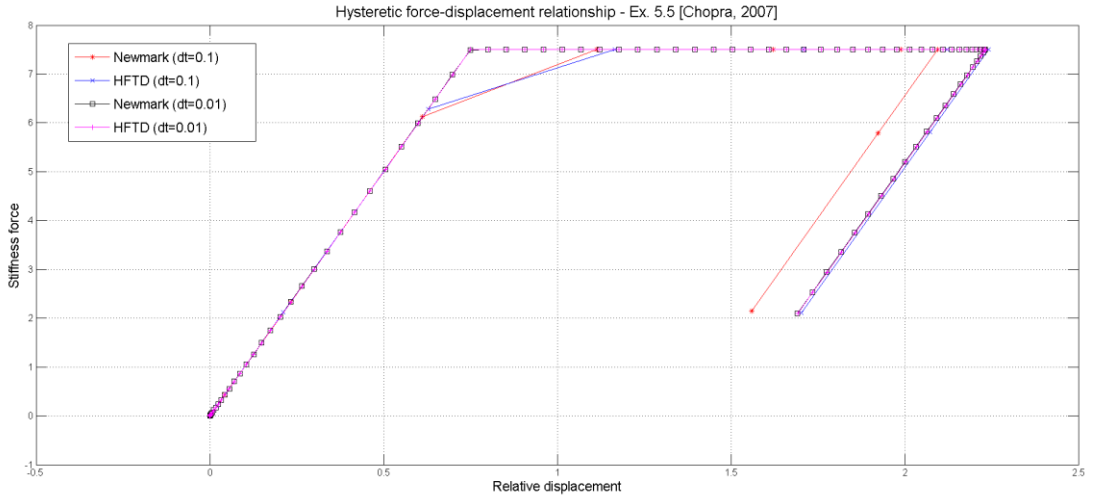


Figure 5.33: Linear elasto-plastic stiffness force non-linearity. Hysteretic stiffness force-displacement relation for Newmark and HFTD solutions.

5.3.3.2 Cubic elasto-plastic non-linear stiffness behavior subject to a blast loading[12]

The SDOF system described in Table 5.31 is subjected to the blast loading shown in Fig. 5.34.

Table 5.31: Numerical data of Example 8.15 (Humar, 2002)

k	m	ζ	u_0	v_0	R_t	R_c
8	0.10	0.1118	0	0	8	-8

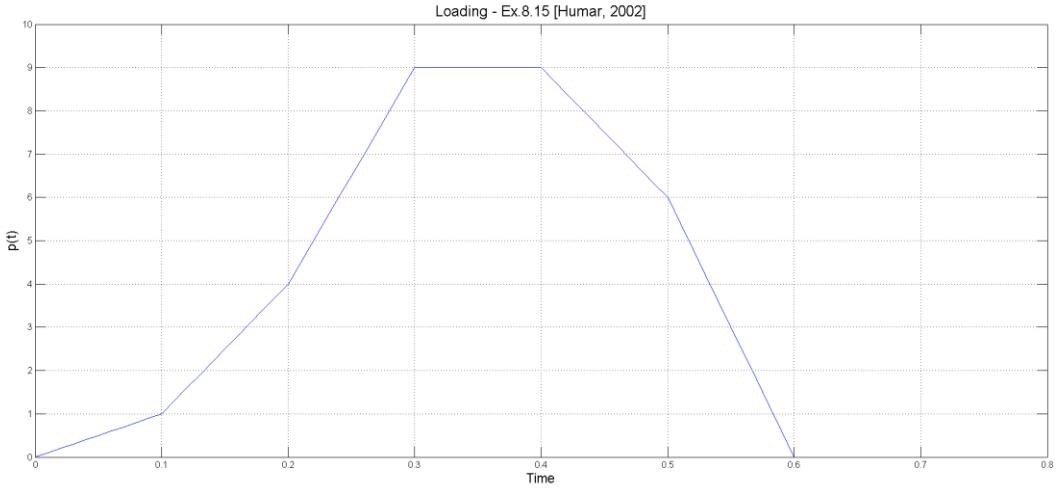


Figure 5.34: Blast loading.

The non-linear stiffness force-displacement relation $[f_S(u, \dot{u})]$ of the system is expressed and shown in Eq. (5.87) and Fig. 5.35, respectively. Eq. (5.88) shows the initial tangent stiffness equal to 8 which is obtained through the evaluation of the initial displacement (u_0) in the derivative of the force-displacement relation.

$$f_S(u) = 8u - \frac{32}{27}u^3 \quad \textbf{(5.87)}$$

$$k = \frac{df_S}{du}(u_0 = 0) = 8 \quad \textbf{(5.88)}$$

The nonlinear stiffness diminishes from 8 to 0, which is the point where the stiffness force reaches its maximum tensile strength equal to 8. After this, the plastic part of the nonlinear relation starts, the displacement increases and the velocity diminishes until zero. The negative velocity makes the system going back in the direction of the equilibrium position following a straight line path with a slope equal to the initial stiffness (see Fig. 5.35). After the stiffness force is reduced to zero, the system enters in the compression zone under the same cubic elastic relation given by Eq. (5.87). The hysteretic cycle is repeated until the whole time span of interest is covered.

[12] Example 8.15 of Humar (2002).

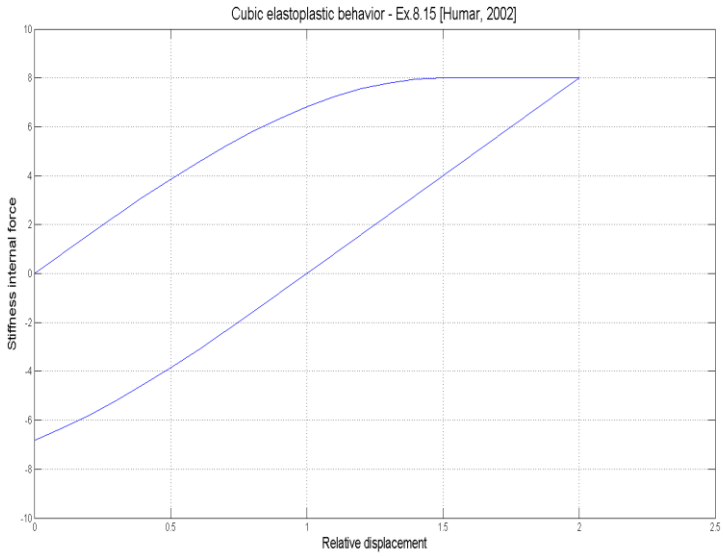


Figure 5.35: Cubic elasto-plastic stiffness non-linearity.

Based on the information given in Table 5.31, the parameters of the SDOF system shown in Table 5.32 are determined.

Table 5.32: Parameters of the SDOF system with stiffness non-linearity subject to blast loading

T_r	T_1	ω_o	ω_d
0.90	0.702	8.950	8.894

Two different time steps are also used in this test example. For the Newmark method, the Average Acceleration special case was performed using the parameters shown in Table 5.33. On the other hand, Table 5.34 shows the values of the parameters required to perform the HFTD method for each of the two time step sizes taken into account.

Table 5.33: Newmark method (average acceleration) parameters for the SDOF system with stiffness non-linearity subject to blast loading

γ	β	Δt_1	Δt_2
$1/2$	$1/4$	0.1	0.01

Table 5.34: HFTD parameters for the SDOF system with stiffness nonlinearity subject to blast loading

Δt	T_p	N	$\omega_n = \frac{2\pi n}{T_p}$	N_T
0.1	35.2	352	0.0284 n	9
0.01	35.13	3513	0.0285 n	9

The comparison between the solutions of the Newmark and the HFTD methods for the displacement, velocity and acceleration responses is presented in Figs. 5.36 to 5.38. In addition, the nonlinear hysteretic relation between the stiffness force and the displacement is presented in Fig. 5.39.

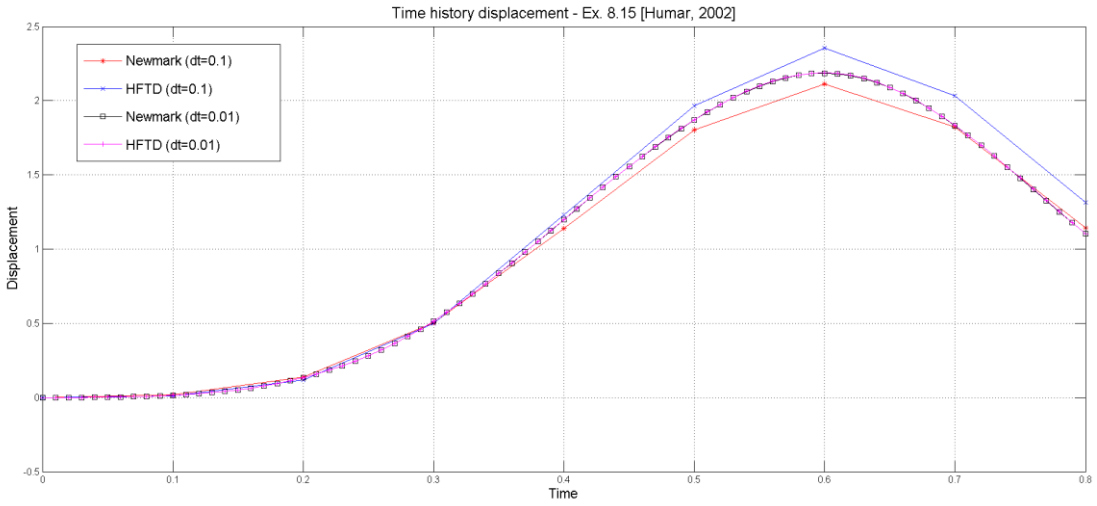


Figure 5.36: Cubic elasto-plastic stiffness force non-linearity. Displacement response for Newmark and HFTD solutions.

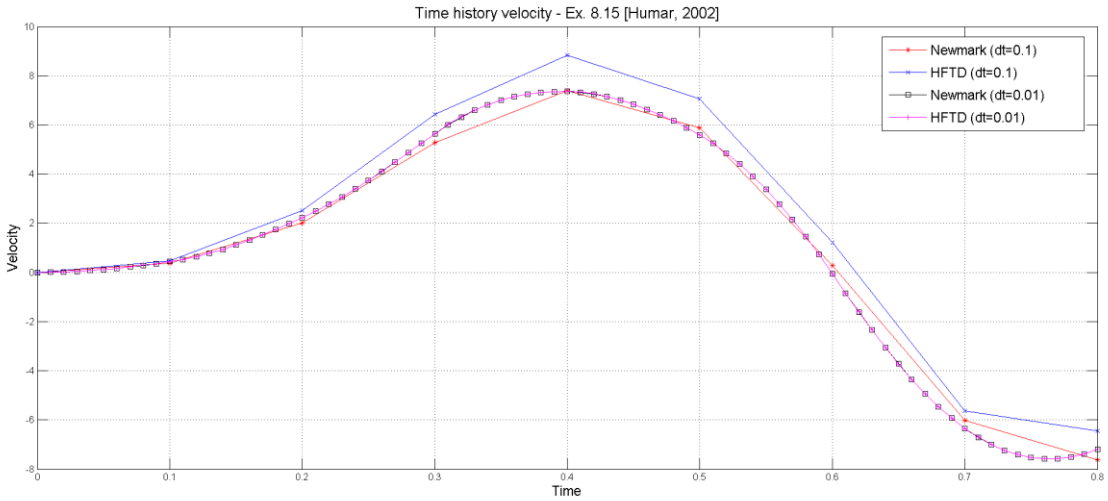


Figure 5.37: Cubic elasto-plastic stiffness force non-linearity. Velocity response for Newmark and HFTD solutions.

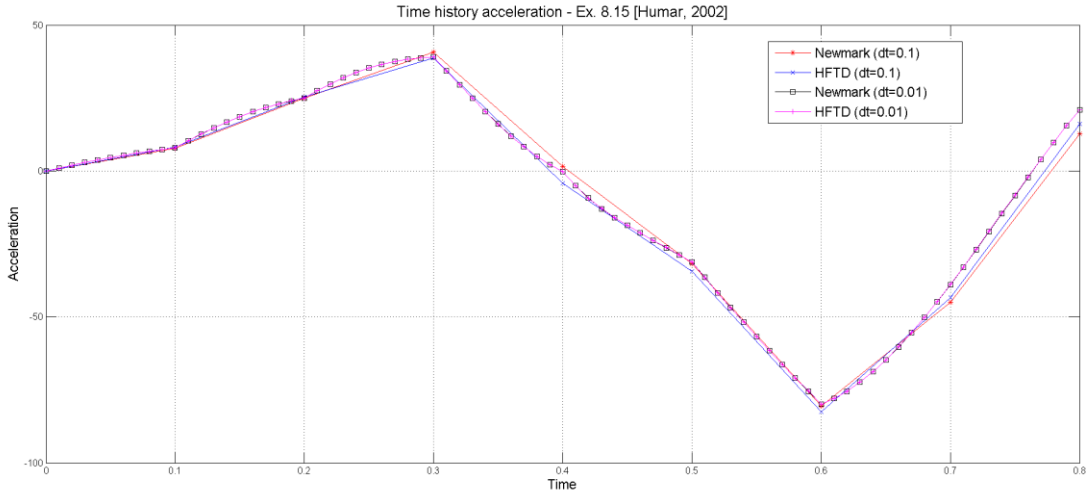


Figure 5.38: Cubic elasto-plastic stiffness force non-linearity. Acceleration response for Newmark and HFTD solutions.

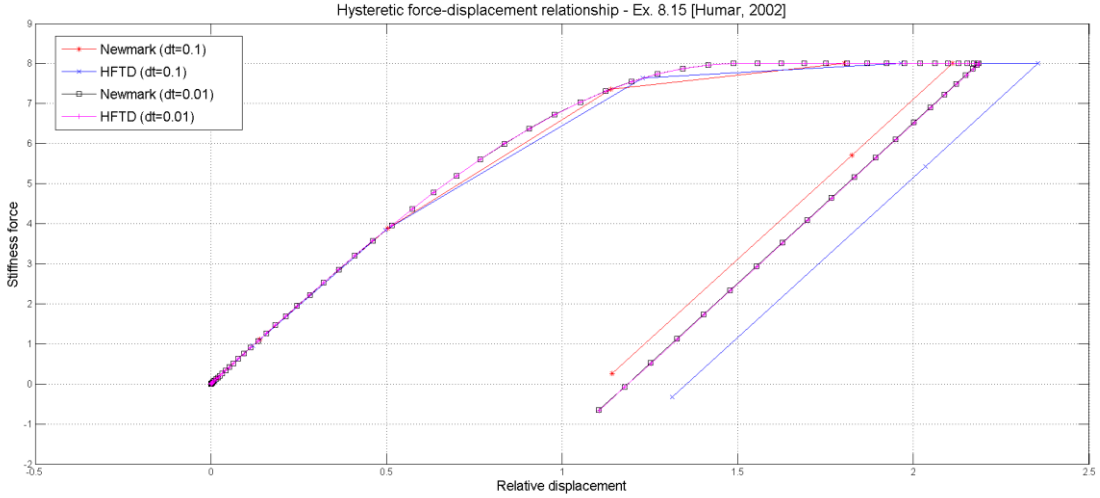


Figure 5.39: Cubic elasto-plastic stiffness force non-linearity. Hysteretic stiffness force-displacement relation for Newmark and HFTD solutions.

From the analysis of Figs. 5.36 to 5.39, it is clear that using a time step equal to 0.1 s does not generate accurate results. The accuracy of the HFTD method is more affected by the time step size due to the relation that this parameter has with the range of frequencies included in the Fourier analysis. When the time step is large, the maximum excitation frequency included in the HFTD analysis is small, and as a consequence, the results obtained are less accurate.

However, similarly than in the previous example, a sufficiently small time step (Δt = 0.01 s) ensures the high precision of the results obtained, not only for the HFTD method, but also for the time domain Newmark method.

Finally, Fig. 5.39 shows that the use of a time step equal to 0.01 s makes it possible to trace the cubic elastic stiffness force-displacement relation with good precision, which is not possible when using a larger time step. Furthermore, both Newmark and HFTD methods show that the last two time points calculated are located in the cubic elastic compression zone. This information is not identified with the larger time step.

5.3.3.3 Cubic elasto-plastic non-linear stiffness and displacement dependent non-linear damping behavior subject to a blast loading

The SDOF system is exactly the same of the previous example (Table 5.31) with the only difference that the damping force is neither linear nor exclusively proportional to velocity, but also displacement dependent, as expressed in Eq. (5.89). The initial tangent damping is defined by Eq. (5.90), whereas the value of the constant damping coefficient (c) is calculated with Eq. (5.15). The blast loading used in this example is also the same as the previous example (Fig. 5.34).

$$f_D(u,\dot{u}) = c\dot{u}(\alpha u^2 \pm 1) \quad \textbf{(5.89)}$$

$$\frac{df_D}{d\dot{u}}(u_0 = 0) = \pm c \quad \textbf{(5.90)}$$

Eq. (5.90) shows that the initial tangent damping can be positive or negative. Both cases are calculated using Newmark and HFTD methods, but considering just one time step equal to 0.01 s, for which it has been shown in the previous examples that the more accurate results are obtained. The values of the parameters c and α for the definition of the damping force in Eq. (5.89) are shown in Table 5.35.

Table 5.35: Parameters for the definition of the nonlinear damping force

c	α
0.20	1

On the other hand, the hysteretic nonlinear stiffness force-displacement relation $[f_S(u,\dot{u})]$ of the system is the same of the previous example, which is defined by Eq. (5.87) and Fig. 5.35. The parameters used with the Newmark (average acceleration) method are shown in Table 5.36, whereas the HFTD parameters are shown in Table 5.37.

Table 5.36: Newmark method (average acceleration) parameters for the SDOF system with stiffness and damping nonlinearity subject to blast loading

γ	β	Δt
$1/2$	$1/4$	0.01

Table 5.37: HFTD parameters for the SDOF system with stiffness and damping nonlinearity subject to blast loading

Δt	T_p	N	$\omega_n = \frac{2\pi n}{T_p}$	N_T
0.01	35.13	3513	0.0285 n	9

It is important to state that for both cases (negative and positive initial tangent damping) studied in this example, the parameters of both methods are the same. What is more, the pseudo linear equation in the frequency domain of the HFTD method (step 1.5.1 of Table 5.2) is always solved using a positive value of c, no matter the sign of the initial tangent damping given in Eq. (5.90). Therefore, the difference between one case and the other is concentrated in the expression for calculating the nonlinear internal damping force in step 1.7 of Table 5.2 (HFTD) and step 2.6 of Table 5.5 (Newmark).

For both cases (positive and negative initial tangent damping), the comparison between the Newmark time domain method and the HFTD method solutions is presented in Figs. 5.40 to 5.42, for the displacement, velocity and acceleration, respectively. As in the previous cases, it is shown that using an adequate time step size (Δt = 0.01) makes the HFTD method as precise as the Newmark method, even for this type of highly non-linear systems defined by Eqs. (5.87) and (5.89).

Moreover, the nonlinear hysteretic relations stiffness force-displacement and damping force-velocity are presented in Figs. 5.43 and 5.44. Both figures confirm that the precision of the HFTD method is not affected by the type (stiffness or damping non-linearity) and degree (quadratic or cubic) of non-linearity, neither by the sign of the initial tangent stiffness or damping.

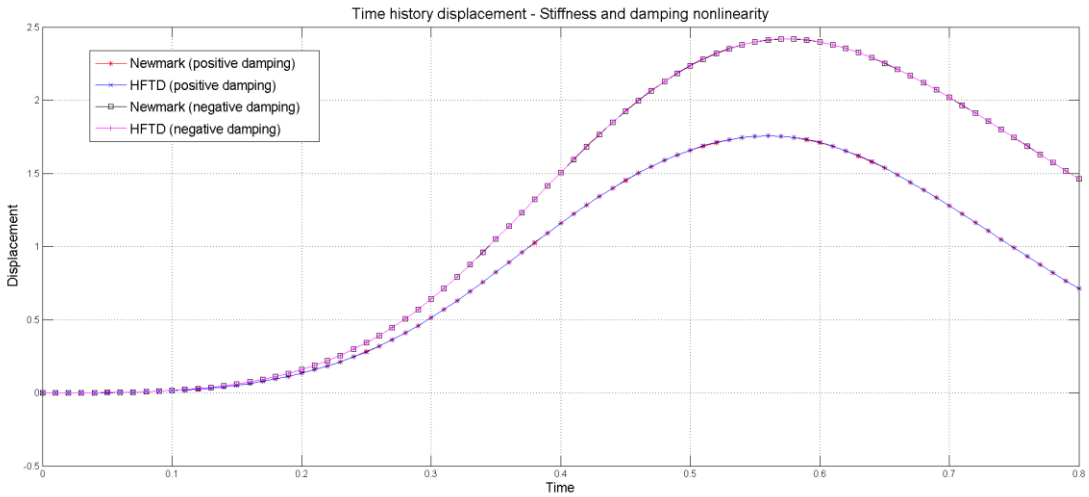


Figure 5.40: Cubic elasto-plastic stiffness force and displacement-dependent damping non-linearity. Displacement response for Newmark and HFTD solutions.

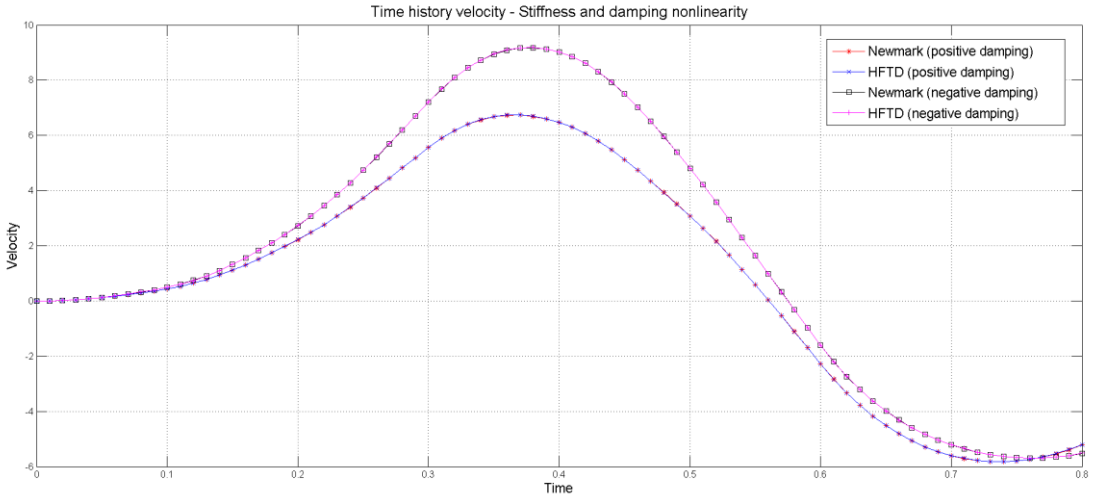


Figure 5.41: Cubic elasto-plastic stiffness force and displacement-dependent damping non-linearity. Velocity response for Newmark and HFTD solutions.

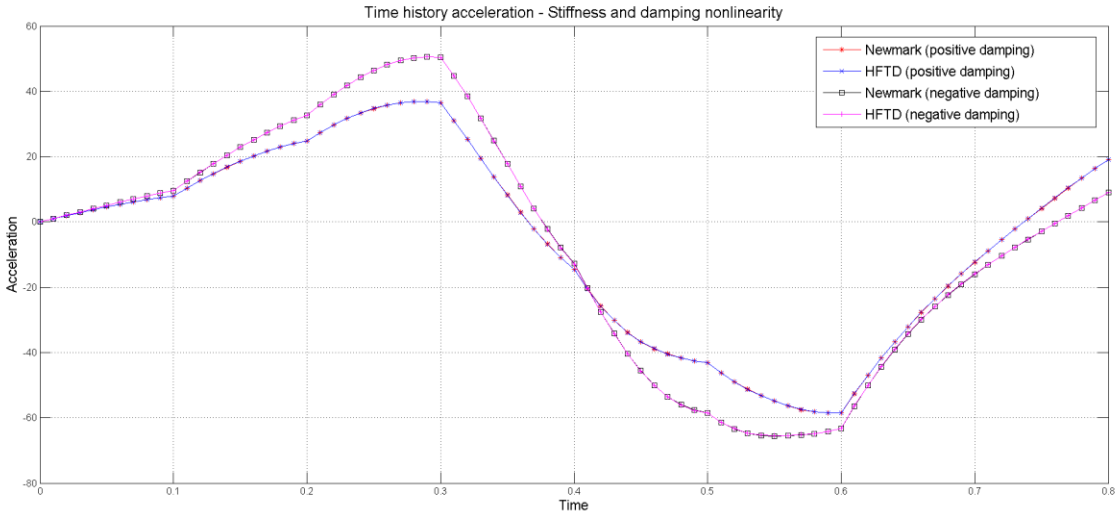


Figure 5.42: Cubic elasto-plastic stiffness force and displacement-dependent damping non-linearity. Acceleration response for Newmark and HFTD solutions.

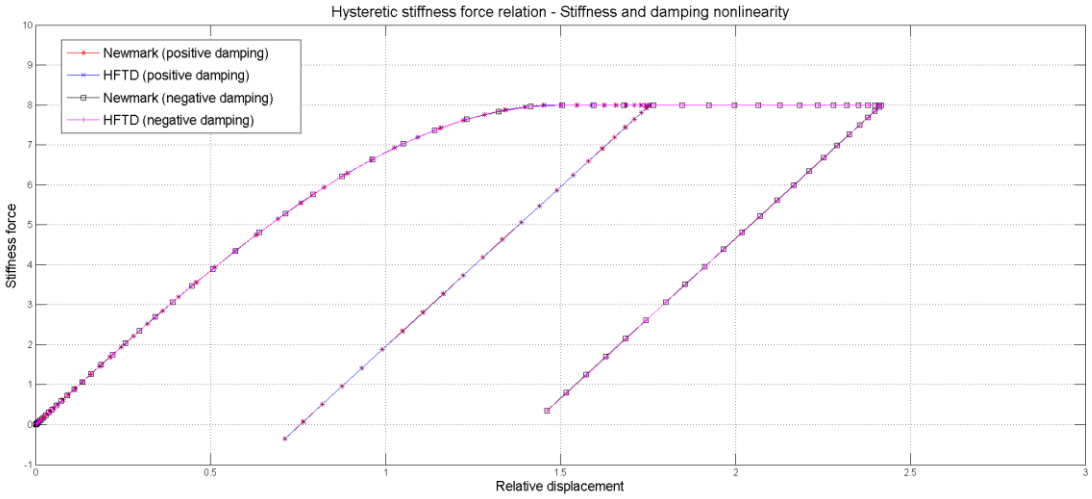


Figure 5.43: Cubic elasto-plastic stiffness force and displacement-dependent damping non-linearity. Hysteretic stiffness force-displacement relation for Newmark and HFTD solutions.

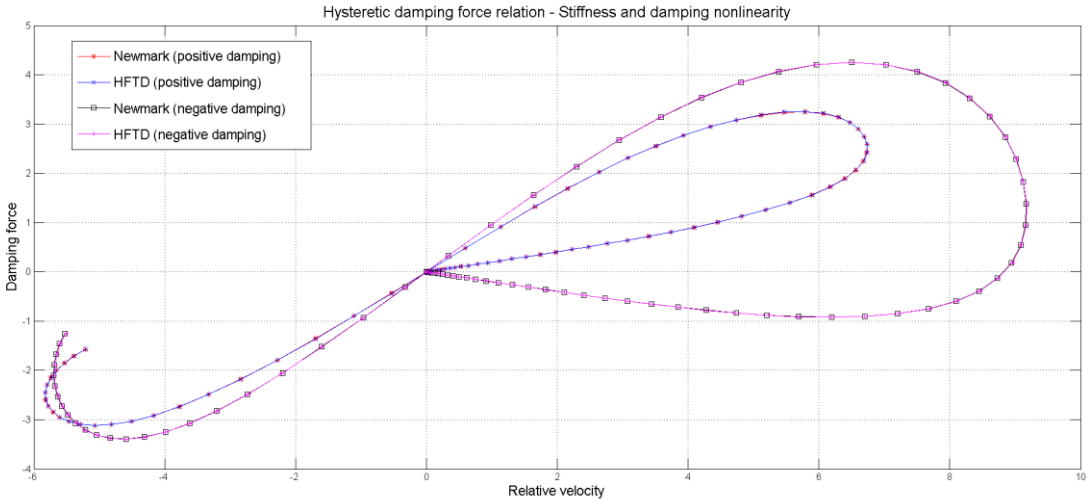


Figure 5.44: Cubic elasto-plastic stiffness force and displacement-dependent damping non-linearity. Hysteretic damping force-velocity relation for Newmark and HFTD solutions.

5.4 Effect of non-linearity in the response of SDOF systems

The example 8.15 of Humar (2002) was selected to show a comparison in the responses of a SDOF system with linear elastic, non-linear elasto-plastic stiffness and nonlinear displacement dependent damping behaviors. The linear response of the system is presented together with the non-linear

response (previously shown) for the displacement, velocity and acceleration in Figs. 5.45 to 5.47, respectively. In this section only the HFTD solutions are displayed, taking into account that in the previous section it was demonstrated that HFTD provides the same results as Newmark method.

In general terms, Fig. 5.45 shows that non-linear behavior increases the magnitude of the displacement response. The effect of damping non-linearity described by Eq. (5.89) depends on the sign of the initial tangent damping. If it is positive, the damping force increases since the beginning and therefore the magnitude of the displacement response is reduced. On the other hand, if the initial damping is negative, the initial damping force has the same direction of the movement and the displacement magnitude is amplified.

Figs. 5.46 and 5.47 show that for velocity and acceleration responses, the first positive peak value is higher for damping non-linearity with negative initial tangent damping. Nevertheless, once the effect of the initial damping is vanished, the linear elastic behavior provides the higher peaks.

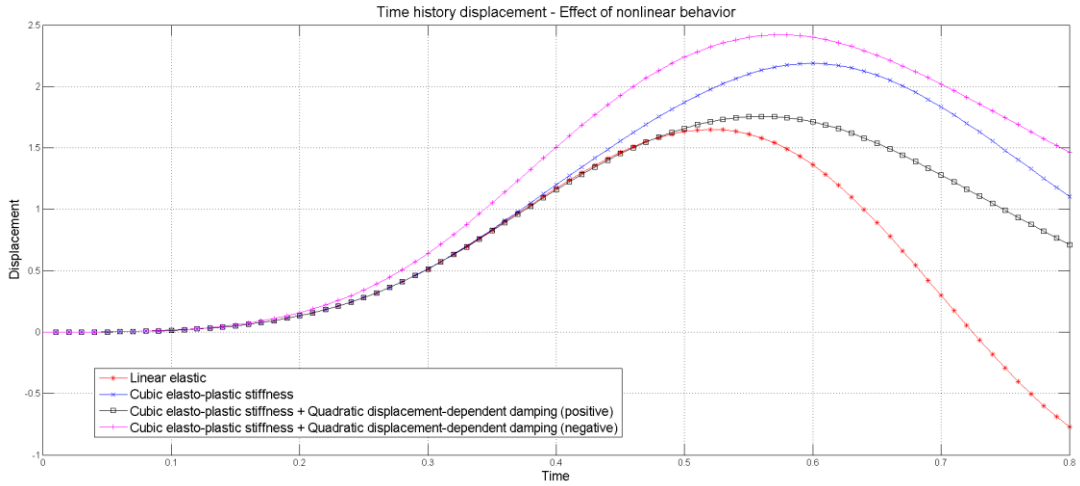


Figure 5.45: Effect of non-linear behavior. Displacement response for HFTD solution.

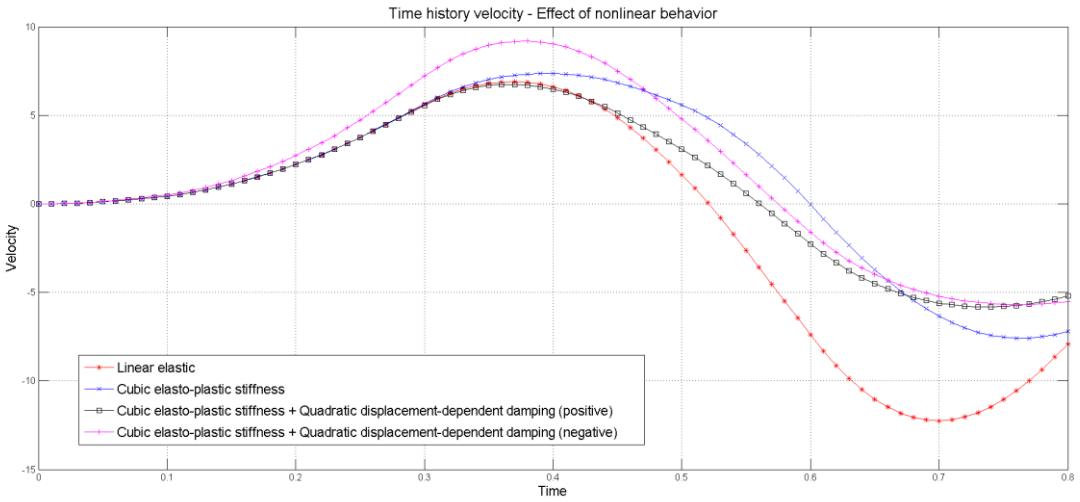


Figure 5.46: Effect of non-linear behavior. Velocity response for HFTD solution.

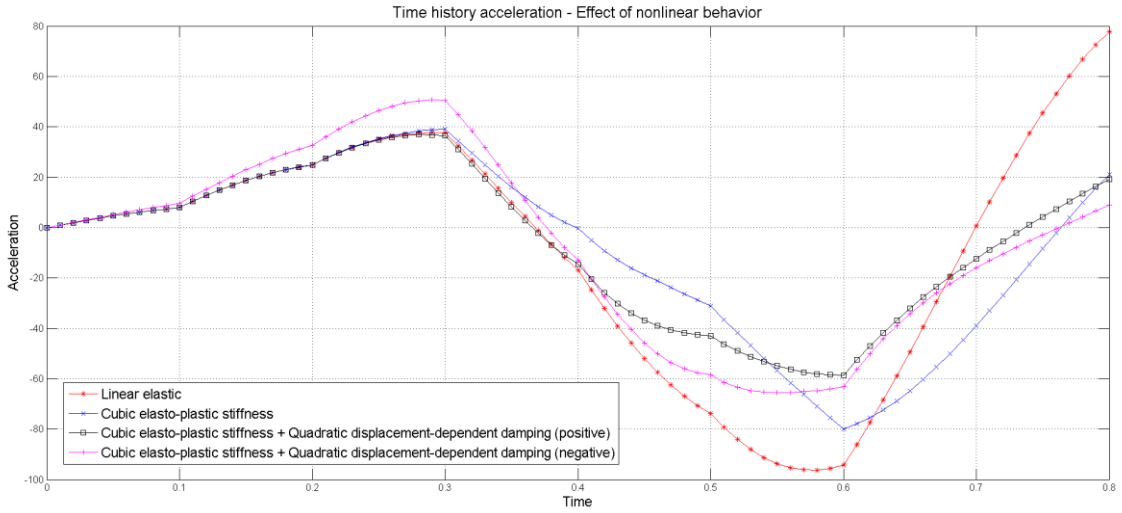


Figure 5.47: Effect of non-linear behavior. Acceleration response for HFTD solution.

6. SIMPLIFIED ONE-DIMENSIONAL DYNAMIC MODELS FOR DAM-RESERVOIR INTERACTION SYSTEMS

6.1. Description of the one-dimensional system

The dam-reservoir interaction system shown in Fig. 6.1 can be modeled in a very simplified way by just taking into account the one dimensional motion in the horizontal direction.

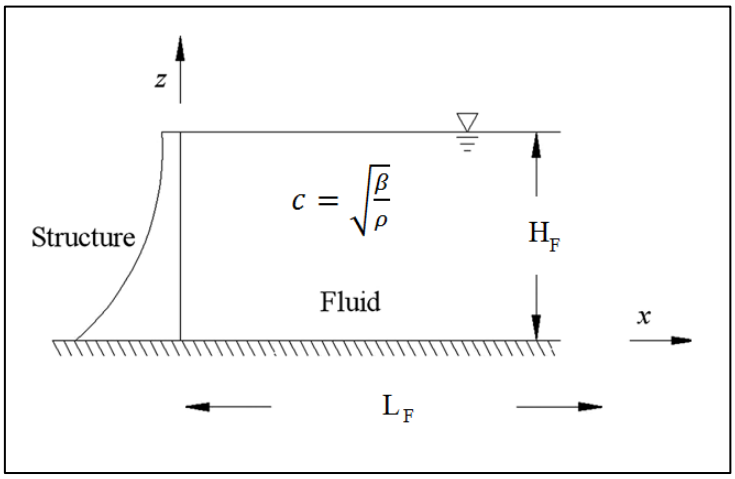


Figure 6.1: Scheme of a dam-reservoir interaction system.

The structural part of such a one-dimensional model can be represented by a one DOF mass-spring-dashpot system which reflects its inertial, stiffness and damping properties. On the other hand, the fluid part can be modeled as a rod which longitudinal motion is described by the one-dimensional wave equation, the boundary conditions in both ends of the rod, and the physical and geometric properties of the fluid domain.

This simplified one-dimensional model of a dam-reservoir interaction system excited by a ground acceleration loading is shown in Fig. 6.2.

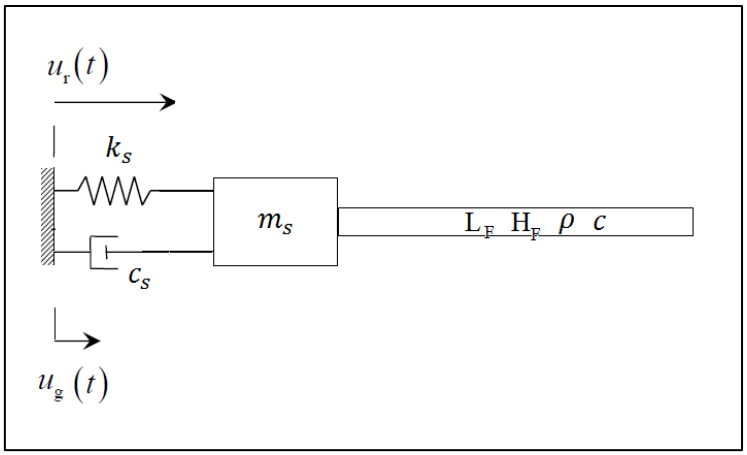


Figure 6.2: Simplified one-dimensional model for dam-reservoir interaction systems

The system shown in Fig. 6.2 is solved using two approaches. The first one is the displacement approach in the time domain (Section 6.2), whereas the second one is the hydrodynamic pressure in the frequency domain (Section 6.3), which follows the same formulation explained in Chapter 3 for 3D MDOF dam-reservoir interaction systems.

In Section 6.4 some dam-reservoir interaction examples are modeled and analyzed with both approaches. Due to the fact that the accuracy of HFTD method for SDOF systems is proven in Chapter 5, all the simulations developed in Section 6.4 are carried out using this method.

6.2. Displacement approach: non-reflective boundary element

In Fig. 6.2 the fluid part of the system is represented as a one-dimensional continuum medium (rod) attached to the structure. The movement of the structure interacts with this continuum medium generating waves that travel longitudinally through all its extension.

One common assumption for the infinite extent boundary condition of the fluid continuum medium is a radiation or non-reflective condition (See Section 3.1.3.3), which absorb all the energy produced by the incident waves. This means that at the boundary location no reflective wave traveling in direction to the structure is generated.

It has been demonstrated that this boundary condition for longitudinal waves can be equivalently represented by a dashpot (non-reflective boundary element) which viscous damping coefficient (c_{nr}) depends on the geometric and physical characteristics of the continuum medium, specifically, on the rod impedance (Z) (Metrikine and Vrouwenvelder, 2006).

The expression to determine the damping coefficient of the non-reflective boundary element is given by Eq. (6.1). Therefore, using this approach, the interaction between the fluid and the structure can be reduced to the effect that the non-reflective boundary condition has in the structural response. This is achieved by replacing the fluid rod shown in Fig. 6.2 for the dashpot shown in Fig. 6.3, which damping coefficient is calculated as expressed in Eq. (6.1), in terms of the wave velocity (c), the fluid density (ρ) and the interaction surface area (A).

$$c_{nr} = Z = c\rho A \qquad \textbf{(6.1)}$$

Finally, the HFTD equation of motion of the SDOF model shown in Fig. 6.3 is given by Eq. (6.2) in terms of the relative displacement amplitude of the dam ($\hat{u}_r$), the ground displacement amplitude ($\hat{\ddot{u}}_g$), the pseudo force ($\hat{q}$) containing the system's non-linearity, and the non-reflective viscous damping (c_{nr}) due to the interaction with the reservoir[1].

$$[-\omega^2 m_S + i\omega(c_S + c_{nr}) + k]\hat{u}_r(\omega) = -(m_S - i\tfrac{c_{nr}}{\omega})\hat{\ddot{u}}_g(\omega) + \hat{q}(\omega) \qquad \textbf{(6.2)}$$

[1] It is important to notice that in Eq. (6.2) the damping force generated by the non-reflective boundary is proportional to the absolute velocity, whereas the damping force of the structure is proportional only to the velocity relative to the ground.

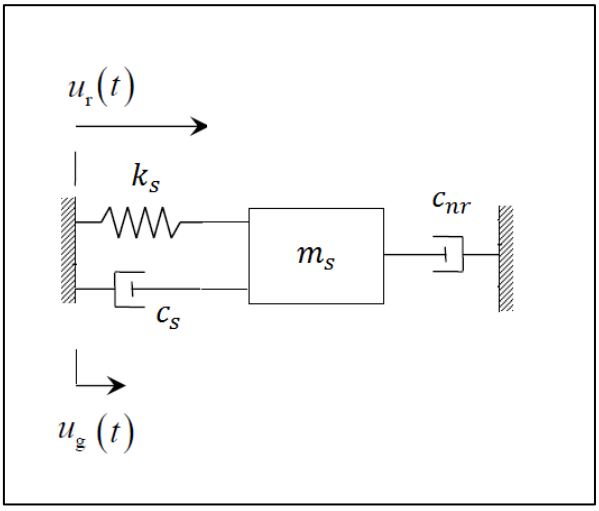


Figure 6.3: Simplified one-dimensional model. Displacement approach with radiation boundary.

6.3. Pressure approach: Frequency dependent hydrodynamic pressure

The one-dimensional wave equation governing the behavior of the continuum fluid medium shown in Fig. 6.2 is defined by Eq. (6.3) in terms of the hydrodynamic pressure.

$$\frac{\partial^2 p}{\partial x^2} = \frac{1}{c^2}\ddot{p} \qquad \textbf{(6.3)}$$

In order to solve this equation and find the analytic expression of the pressure wave propagation, two boundary conditions (one per each end of the rod) have to be defined. The first boundary condition is given by the interaction between the structural mass and the fluid, whereas the second one is given by the hypothesis of the reservoir length (finite or infinite) extension.

The pressure wave equation approach to model the one-dimensional dam-reservoir interaction system is graphically described in Fig. 6.4. Depending on the assumption for the far extent boundary condition two different cases can be developed (Sections 6.3.1 and 6.3.2), whereas, if the fluid is assumed to be incompressible, a third case (Section 6.3.3) is obtained for this approach.

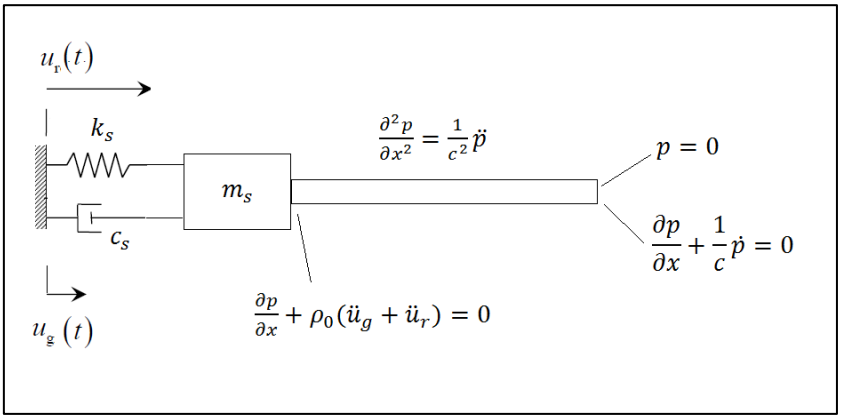


Figure 6.4: Simplified one-dimensional model. Hydrodynamic pressure approach.

In any of the three particular cases the interaction between the fluid and the structure is reduced to a frequency dependent pressure or surface load $[\hat{P}(\omega)]$ applied to the structural mass-spring-dashpot system[2], as it is shown in Fig. 6.5. For this reason, the one-dimensional model under the pressure approach is ideal to be solved with the HFTD method.

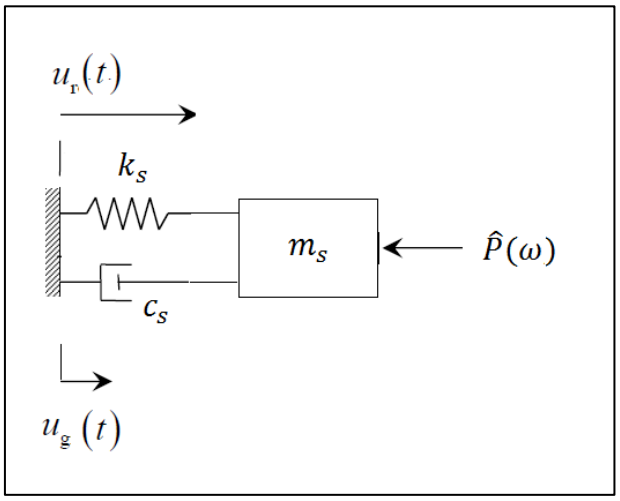


Figure 6.5: Simplified one-dimensional model. Frequency dependent hydrodynamic pressure load.

The HFTD equation of motion of the SDOF model shown in Fig. 6.5 is given by Eq. (6.4) in terms of the relative displacement amplitude of the dam ($\hat{u}_r$), the ground displacement amplitude ($\hat{\ddot{u}}_g$), the pseudo force ($\hat{q}$) containing the system's non-linearity, and the hydrodynamic pressure load ($\hat{P}$) due to the interaction with the reservoir.

$$[-\omega^2 m_S + i\omega c_S + k]\hat{u}_r(\omega) = -m_S\hat{\ddot{u}}_g(\omega) - \hat{P}(\omega) + \hat{q}(\omega) \qquad \textbf{(6.4)}$$

[2] This is the same approach used in Chapter 3 for MDOF systems.

The analytic expressions of $\hat{P}(\omega)$ for each of the three particular cases previously mentioned are derived in the next sections.

6.3.1. Case I: Compressible fluid with radiation boundary of infinite extent

The first boundary condition of Case I is expressed by Eq. (6.5). This equation describes the fluid-structure interaction at the interface ($x = 0$), being derived from Eq. (3.11) for one-dimensional motion. Eq. (6.5) denotes the horizontal force equilibrium between the fluid pressure gradient and the structure inertial force[3] per volume.

$$\frac{\partial p}{\partial x} + \rho_0(\ddot{u}_g + \ddot{u}_r) = 0 \qquad \text{on} \qquad x = 0 \qquad \textbf{(6.5)}$$

The second boundary condition is expressed by Eq. (6.6) which defines the radiation boundary at infinite extent ($x \to \infty$). This boundary condition, known as the Sommerfeld's radiation condition, was previously presented in Eq. (3.21).

$$\frac{\partial p}{\partial x} + \frac{1}{c}\dot{p} = 0 \qquad \text{on} \qquad x \to \infty \qquad \textbf{(6.6)}$$

The HFTD method solves in the frequency domain the system shown in Fig. 6.4. Consequently, it is convenient to express the excitation force (ground acceleration) and the response (relative displacement) of the system in their harmonic form, as shown in Eqs. (6.7) and (6.8), respectively.

$$\ddot{u}_g = \hat{\ddot{u}}_g(\omega)e^{i\omega t} \qquad \textbf{(6.7)}$$

$$u_r = \hat{u}_r(\omega)e^{i\omega t} \qquad \textbf{(6.8)}$$

In the same way, the pressure wave generated in the fluid domain due to the excitation force in the structural domain is given by the summation of the incident and reflected harmonic waves. Eq. (6.9) expresses both harmonic waves in terms of their frequency dependent complex amplitudes and the wave velocity of propagation.

$$p = p_i + p_r = \hat{p}_i(\omega)e^{i\omega\left(t-\frac{x}{c}\right)} + \hat{p}_r(\omega)e^{i\omega\left(t+\frac{x}{c}\right)} \qquad \textbf{(6.9)}$$

The expression of the harmonic wave defined in Eq. (6.9) fulfills the wave equation of motion defined in Eq. (6.3). Therefore, in order to calculate the unknown amplitudes of the incident and reflected waves, it is required to use both boundary conditions previously defined. After replacing Eq. (6.9) into the infinite extent radiation boundary condition given by Eq. (6.6), the relations expressed in Eqs. (6.10) and (6.11) are obtained.

[3] Notice that in Eq. (6.5) the total acceleration is expressed as the sum of ground and relative acceleration.

$$i\frac{2\omega}{c}\,\hat{p}_r(\omega)e^{i\omega\left(t+\frac{L_F}{c}\right)}=0 \qquad \textbf{(6.10)}$$

$$L_F \to \infty \qquad \textbf{(6.11)}$$

From Eq. (6.10) it can be concluded that the amplitude of the reflected pressure is equal to zero $[\hat{p}_r(\omega)=0]$, which is expected due to the non-reflective nature of Sommerfeld's radiation boundary condition. Therefore, the pressure wave is only composed by the incident wave, and Eq. (6.9) is reformulated as expressed in Eq. (6.12).

$$p=p_i=\hat{p}_i(\omega)e^{i\omega\left(t-\frac{x}{c}\right)} \qquad \textbf{(6.12)}$$

After replacing Eqs. (6.7), (6.8) and (6.12) into the second boundary condition at $x=0$ given by Eq. (6.5), the expression of the incident pressure wave amplitude is obtained in Eq. (6.13). The frequency dependent amplitude of the hydrodynamic force $[\hat{P}(\omega)]$ is obtained in Eq. (6.14) after multiplying the pressure times the interaction surface area (A).

$$\hat{p}_i(\omega)=-i\frac{c\rho}{\omega}[\hat{\ddot{u}}_g(\omega)-\omega^2\hat{u}_r(\omega)] \qquad \textbf{(6.13)}$$

$$\hat{P}(\omega)=-i\frac{c\rho A}{\omega}[\hat{\ddot{u}}_g(\omega)-\omega^2\hat{u}_r(\omega)] \qquad \textbf{(6.14)}$$

The HFTD equation for the particular Case I is obtained in Eq. (6.15) by replacing the hydrodynamic force amplitude given by Eq. (6.14) into Eq. (6.4). It is important to notice that Eq. (6.15) is identical to Eq. (6.2). This is because that, in spite of being derived by different formulations, both equations responds to the same hypothesis for the fluid domain, namely, compressible with radiation boundary of infinite extent.

$$[-\omega^2 m_S+i\omega(c_S+c\rho A)+k]\hat{u}_r(\omega)=-(m_S-i\frac{c\rho A}{\omega})\hat{\ddot{u}}_g(\omega)+\hat{q}(\omega) \qquad \textbf{(6.15)}$$

Eq. (6.15) shows that for Case I the reservoir interaction with the structure provides additional damping and frequency dependent loading terms in the left and right-hand sides of the equation, respectively.

6.3.2. Case II: Compressible fluid with zero pressure boundary of finite extent

The zero pressure boundary condition of finite extent is defined in Eq. (6.16) for a reservoir with finite length equal to L_F. After replacing Eq. (6.9) into Eq. (6.16) the relation between the incident and reflected pressure amplitudes is obtained in Eq. (6.17).

$$p=0 \qquad \text{on} \qquad x=L_F \qquad \textbf{(6.16)}$$

$$\frac{p_r}{p_i} = -e^{i\left(-\frac{2L_F}{c}\omega\right)} \tag{6.17}$$

The substitution of Eqs. (6.7) to (6.9) and Eq. (6.17) into the fluid-structure interface boundary condition, given by Eq. (6.5), provides the expression of the incident pressure amplitude in Eq. (6.18). The reflected pressure amplitude is obtained in Eq. (6.19) after replacing Eq. (6.17) into Eq. (6.18).

$$\hat{p}_i(\omega) = -i\frac{c\rho}{\omega}\frac{1}{1+e^{i\left(-\frac{2L_F}{c}\omega\right)}}[\hat{\ddot{u}}_g(\omega) - \omega^2\hat{u}_r(\omega)] \tag{6.18}$$

$$\hat{p}_r(\omega) = i\frac{c\rho}{\omega}\frac{1}{1+e^{i\left(\frac{2L_F}{c}\omega\right)}}[\hat{\ddot{u}}_g(\omega) - \omega^2\hat{u}_r(\omega)] \tag{6.19}$$

The summation of both amplitudes given by Eq. (6.20) provides the total amplitude of the pressure wave acting at the fluid-structure interface. The frequency dependent hydrodynamic force amplitude $[\hat{P}(\omega)]$ is obtained in Eq. (6.21) after multiplying the pressure times the interaction surface area (A) and simplify the Euler notation complex numbers division.

$$\hat{p}(\omega) = -i\frac{c\rho}{\omega}\frac{1-e^{i\left(-\frac{2L_F}{c}\omega\right)}}{1+e^{i\left(-\frac{2L_F}{c}\omega\right)}}[\hat{\ddot{u}}_g(\omega) - \omega^2\hat{u}_r(\omega)] \tag{6.20}$$

$$\hat{P}(\omega) = \frac{c\rho A}{\omega}\frac{\sin(\frac{2L_F}{c}\omega)}{1+\cos(\frac{2L_F}{c}\omega)}[\hat{\ddot{u}}_g(\omega) - \omega^2\hat{u}_r(\omega)] \tag{6.21}$$

The HFTD equation for the particular Case II is obtained in Eq. (6.22) by replacing the hydrodynamic force amplitude given by Eq. (6.21) into Eq. (6.4). Due to the finite extent reflexive boundary condition, the fluid interaction with the structure provides exclusively frequency dependent additional mass to the left and right-hand side of Eq. (6.22).

$$\left[-\omega^2\left(m_S + \frac{c\rho A}{\omega}\frac{\sin(\frac{2L_F}{c}\omega)}{1+\cos(\frac{2L_F}{c}\omega)}\right) + i\omega c_S + k\right]\hat{u}_r(\omega) = -\left(m_S + \frac{c\rho A}{\omega}\frac{\sin(\frac{2L_F}{c}\omega)}{1+\cos(\frac{2L_F}{c}\omega)}\right)\hat{\ddot{u}}_g(\omega) + \hat{q}(\omega) \tag{6.22}$$

6.3.3. Case III: Incompressible fluid

For the incompressible fluid case ($c \to \infty$), Eq. (6.3) is transformed into Eq. (6.23). This means that the fluid is not anymore a medium where harmonic pressure waves propagate, but a vibrating body which amplitudes varies in space. The harmonic form of the hydrodynamic pressure for the incompressible fluid case is given by Eqs. (6.24) and (6.25).

$$\frac{\partial^2 p}{\partial x^2} = 0 \tag{6.23}$$

$$p = F(x,\omega)e^{i\omega t} \tag{6.24}$$

$$F(x,\omega) = C_1 x + C_2 \quad \textbf{(6.25)}$$

Due to the lack of fluid's compressibility, radiation boundary condition of infinite extent is not possible to be considered. Therefore, Eqs. (6.24) and (6.25) are introduced in the boundary condition of finite extent at $x = L_F$, given by Eq. (6.16). The relation between the constants C_1 and C_2 is obtained in Eq. (6.26).

$$-\frac{C_2}{C_1} = L_F \quad \textbf{(6.26)}$$

Replacing Eqs. (6.7), (6.8), (6.24) and (6.25) into the fluid-structure interface boundary condition given by Eq. (6.5), and then using the relation of Eq. (6.26), provide the expressions of C_1 and C_2, given by Eqs. (6.27) and (6.28), respectively.

$$C_1 = -\rho[\hat{\ddot{u}}_g(\omega) - \omega^2 \hat{u}_r(\omega)] \quad \textbf{(6.27)}$$

$$C_2 = \rho L_F[\hat{\ddot{u}}_g(\omega) - \omega^2 \hat{u}_r(\omega)] \quad \textbf{(6.28)}$$

After replacing Eqs. (6.27) and (6.28) into Eq. (6.25), the space and frequency dependent amplitude of the hydrodynamic pressure is obtained in Eq. (6.29). However, it is of interest to know what is the pressure amplitude at $x = 0$, where the mass of the structure is located. This pressure acting in the fluid-structure interface $[\hat{p}(\omega)]$, as well as its associated force $[\hat{P}(\omega)]$, is defined in Eqs. (6.30) and (6.31), respectively.

$$F(x,\omega) = -\rho(x - L_F)[\hat{\ddot{u}}_g(\omega) - \omega^2 \hat{u}_r(\omega)] \quad \textbf{(6.29)}$$

$$\hat{p}(\omega) = \rho L_F[\hat{\ddot{u}}_g(\omega) - \omega^2 \hat{u}_r(\omega)] \quad \textbf{(6.30)}$$

$$\hat{P}(\omega) = \rho A L_F[\hat{\ddot{u}}_g(\omega) - \omega^2 \hat{u}_r(\omega)] \quad \textbf{(6.31)}$$

The HFTD equation for the particular Case III is obtained in Eq. (6.32) by replacing the hydrodynamic force given by Eq. (6.31) into Eq. (6.4). Eq. (6.32) shows that the hydrodynamic force for the incompressible fluid case is an additional inertial force provided by the mass of the fluid body ($\rho A L_F$). This means that under this assumption, the mass of the fluid behaves like a rigid body oscillating attached to the structure and introducing no damping to the system, like in the previous particular cases.

$$[-\omega^2 (m_S + \rho A L_F) + i\omega c_S + k]\hat{u}_r(\omega) = -(m_S + \rho A L_F)\hat{\ddot{u}}_g(\omega) + \hat{q}(\omega) \quad \textbf{(6.32)}$$

6.4. HFTD method numerical examples

A modified version of the HFTD method programmed in MATLAB (The Mathworks, Inc., 2012)[4] for the solution of the SDOF system examples developed in Chapter 5 is used in this section to obtain the response of the simplified dam-reservoir interaction system.

6.4.1. Description of the simplified dam-reservoir interaction system

The numerical data describing the characteristics of the one-dimensional dam model shown in Fig. 6.2 is given in Table 6.1. The dam's damping coefficient (c_S) is calculated with Eq. (6.33), the reservoir infinite extent non-reflective damping coefficient ($c\rho A$) is calculated with Eq. (6.34), the time that the incident and reflected pressure wave takes to reach the back of the reservoir and go back to the dam-reservoir interface ($\frac{2L_F}{c}$) is calculated with Eq. (6.35), and the mass of water contained in the reservoir ($\rho A L_F$) is calculated in Eq. (6.36).

The ground acceleration loading applied to the base of the dam corresponds to the first 4.5 s of the "El Centro" earthquake North-South component shown in Fig. 6.6.

Table 6.1: Numerical data of the one-dimensional dam-reservoir system

Dam			Reservoir			
k_S (N/m)	ξ	m_S (kg)	L_F (m)	H_F (m)	ρ (kg/m^3)	c (m/s)
6.6 x 10^9	0.05	3.6 x 10^7	2500	200	1000	1440

$$c_S = 2\xi\sqrt{k_S m_S} = 4.87 \times 10^7 \text{ Ns/m} \quad \textbf{(6.33)}$$

$$c\rho A = c\rho H_F = 2.88 \times 10^8 \text{ Ns/m} \quad \textbf{(6.34)}$$

$$\frac{2L_F}{c} = 3.47 \text{ s} \quad \textbf{(6.35)}$$

$$\rho A L_F = \rho H_F L_F = 5.00 \times 10^8 \, kg \quad \textbf{(6.36)}$$

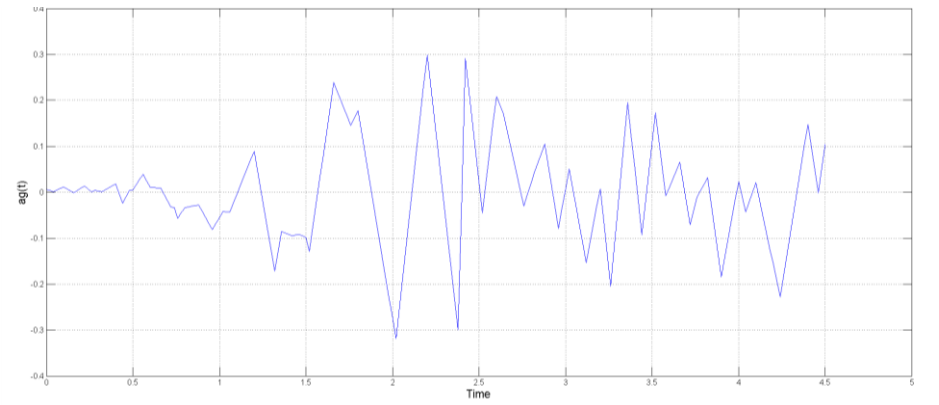


Figure 6.6: El Centro N-S earthquake ground acceleration.

[4] The MATLAB routines can be found in Appendix B.

6.4.2. Dam's linear behavior: Influence of dam-reservoir interaction frequency dependent properties

The first numerical examples correspond to a linear-elastic behavior of the dam, which is generally the case for frequency-dependent systems. The information given in Table 6.1 and Eqs. (6.33) to (6.36) is used by the HFTD method to solve the pseudo-linear equations defined in Eqs. (6.15), (6.22) and (6.32) for the cases I, II and III, respectively. Additionally, the corresponding SDOF system without taking into account the hydrodynamic force [$P(\omega)$] is also solved as a fourth case which simulates the analysis of a dam with empty reservoir ("dry dam").

Figs. 6.7 to 6.9 show the time history responses for the displacement, velocity and acceleration, respectively, relative to the base motion. Fig 6.10 presents the variation in time of the base reaction force, defined as the summation of the dam's stiffness and damping forces ($f_S + f_D$).

Considering that these one-dimensional models constitute a substantial simplification of reality, it is not possible to obtain general conclusions of the linear behavior of dam-reservoir interaction systems from Figs. 6.7 to 6.10. However, this information is very illustrating to have an insight of the influence that frequency dependent properties have in the response of dynamic equations of motion like Eqs. (6.15), (6.22) and (6.32).

As explained in Section 6.2 and demonstrated in Section 6.3.1, the radiation boundary of infinite extent (Case I) acts like a dashpot in the system. For that reason, the peak values of all the responses of Case I are smaller than the corresponding peak values of the dry dam case. For this particular example, the magnitude of the non-reflective dashpot coefficient [Eq. (6.34)] is so high that one the earthquake time is finished (4.5 s), the system goes back to the equilibrium position without oscillations and extremely slow.

The incompressible fluid case (Case III) presents very high peaks for displacement and base reaction responses. The reason for that is the higher flexibility of the system compared to the dry dam due to the additional mass provided by the fluid [Eq. (6.32)]. This produces an increment of the length of the natural vibration period of the system.

One interesting feature of the zero pressure reflexive boundary (case II) is that for all the responses presented its peak values occur after the earthquake loading has finished. This fact confirms the reflective characteristic of this boundary condition which does not takes out but introduces energy to the system. This is the only case which responses do not damp out in time when the loading is not present anymore.

These observations based on the response of frequency dependent SDOF systems (Figs. 6.7 to 6.10) provide a general overview of the importance that the reservoir compressibility and boundary conditions assumptions can have in the seismic analysis of dams.

In the next section the effect that non-linear behavior has in this type of frequency dependent systems is assessed.

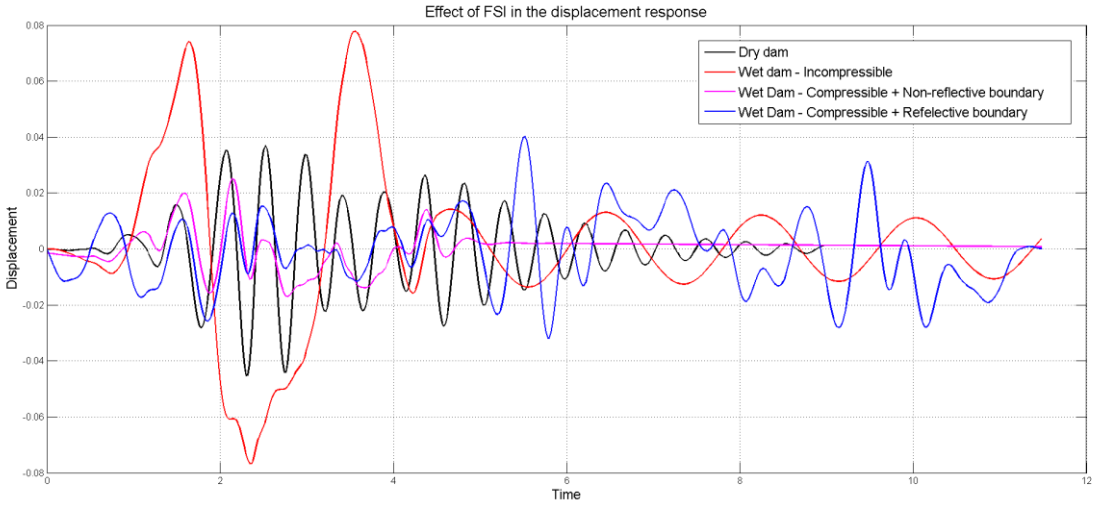


Figure 6.7: Dam-reservoir interaction SDOF systems. Displacement response for linear-elastic behavior.

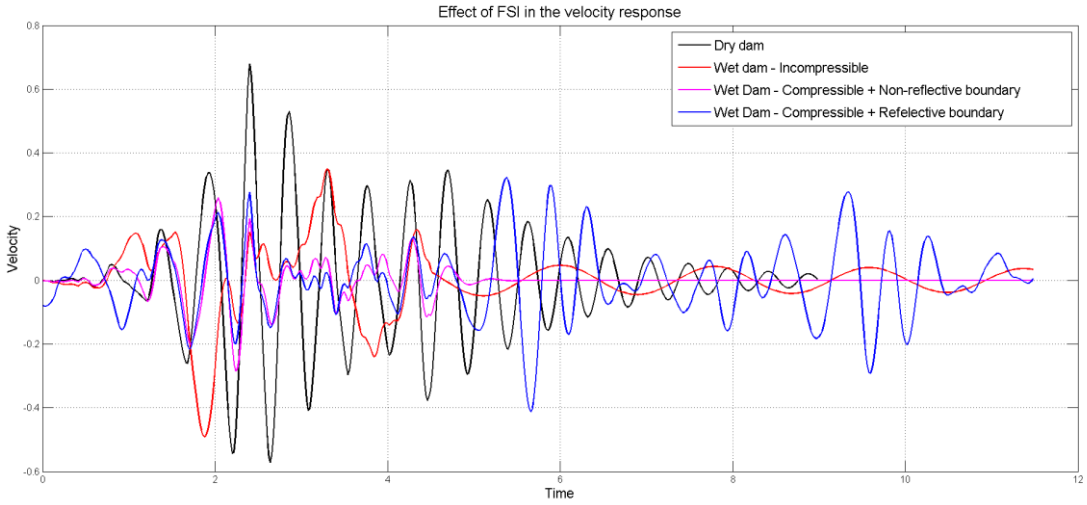


Figure 6.8: Dam-reservoir interaction SDOF systems. Velocity response for linear-elastic behavior.

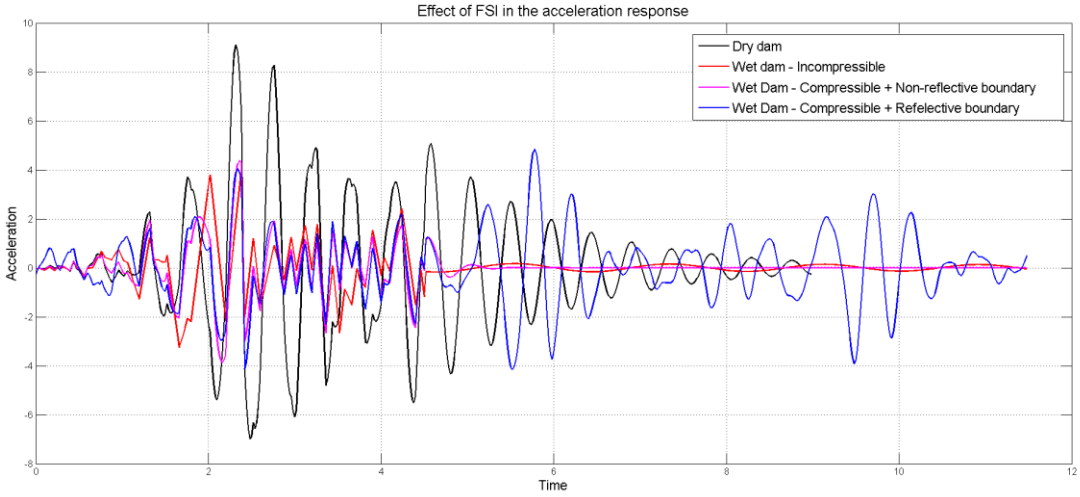


Figure 6.9: Dam-reservoir interaction SDOF systems. Acceleration response for linear-elastic behavior.

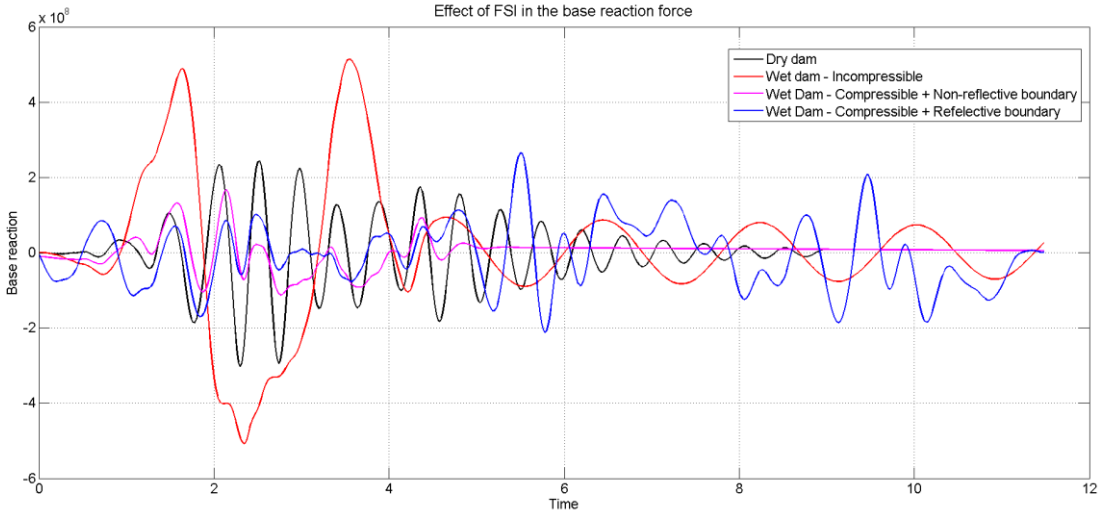


Figure 6.10: Dam-reservoir interaction SDOF systems. Base reaction response for linear-elastic behavior.

6.4.3. Dam's cracking non-linearity: Stiffness softening

The dam's concrete cracking non-linearity is represented by the softening cubic relation given by Eq. (6.37) and schematically shown in Fig. 6.11, between the stiffness force (f_S) and the relative displacement (u_r).

$$f_S(u_r) = k_1 u_r - k_3 {u_r}^3 \qquad \textbf{(6.37)}$$

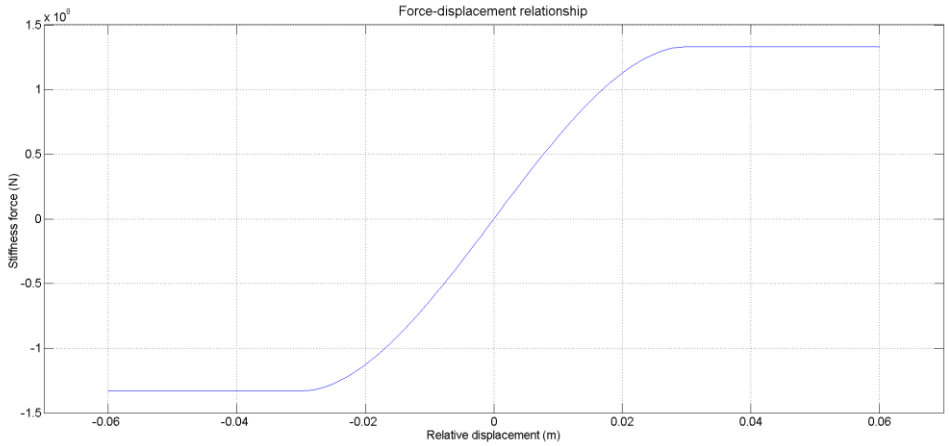


Figure 6.11: Cubic stiffness softening force-displacement relationship for concrete cracking non-linear behavior

As explained before, Cases I and II are frequency dependent systems which take into account the reservoir compressibility. Therefore, it is of interest to study the effect of the non-linear behavior in these two frequency dependent systems. Additionally, the non-linear response of the "dry dam" case (no frequency-dependence) is also presented as a reference.

For the three cases the initial tangent stiffness is assumed to be equal to the linear elastic stiffness of the dam (k_S), as expressed in Eq. (6.38). However, the values of the cubic coefficient (k_3) in Eq. (6.37) are assigned for each case as shown in Table 6.2.

$$\frac{df_S}{du_r}(0) = k_1 = k_S = 6.6 \times 10^9\ N/m \qquad \textbf{(6.38)}$$

Table 6.2: Values assigned to the cubic coefficient k_3 of Eq. (6.37) for each analysis case

Dry dam	Compressible water with radiation boundary of infinite extent (Case I)	Compressible water with reflective boundary of finite extent (Case II)
2.40 x 10^{12}	2.40 x 10^{12}	1.05 x 10^{12}

The previous analysis of Figs. 6.7 to 6.10 demonstrates that the reflective boundary condition (Case II) generates higher linear-elastic responses, and therefore a stiffer and more resistant non-linear behavior is required for this particular system. This is the reason why a smaller value of k_3 is assigned to this case in Table 6.2.

Finally, the pseudo force in the time domain for the cubic stiffness softening non-linearity is calculated using Eq. (6.39). Then, the non-linear behavior is completely defined and the HFTD method can be applied in the solution of the SDOF dam-reservoir interaction models.

$$q(t) = k_S u_r - f_S(u_r) \qquad \textbf{(6.39)}$$

First, the nonlinear behavior of the dry dam case is studied as a reference of a non frequency dependent system. Figs. 6.12 and 6.13 present the effect of non-linear behavior in the displacement and stiffness internal force of the SDOF system, respectively.

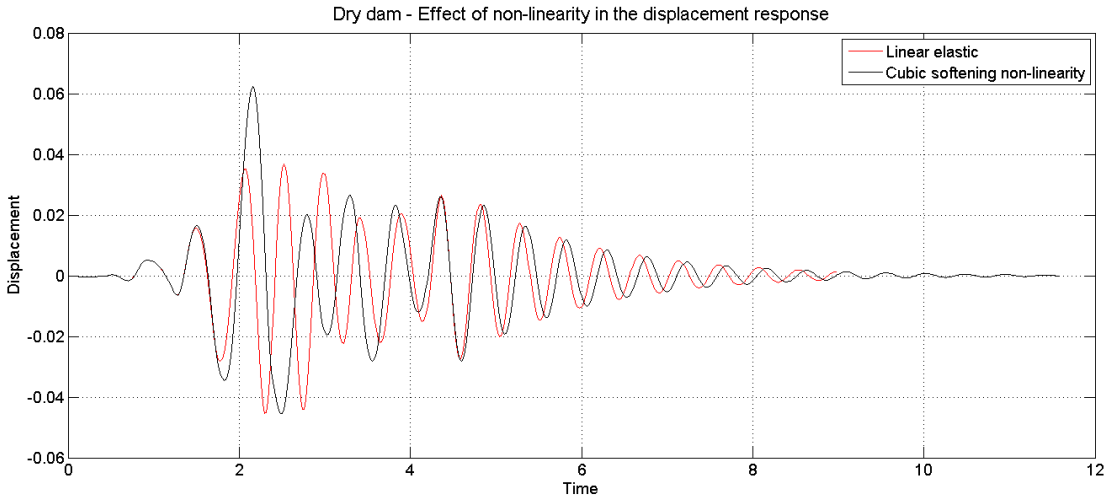


Figure 6.12: "Dry dam" case. Effect of non-linearity in the displacement response.

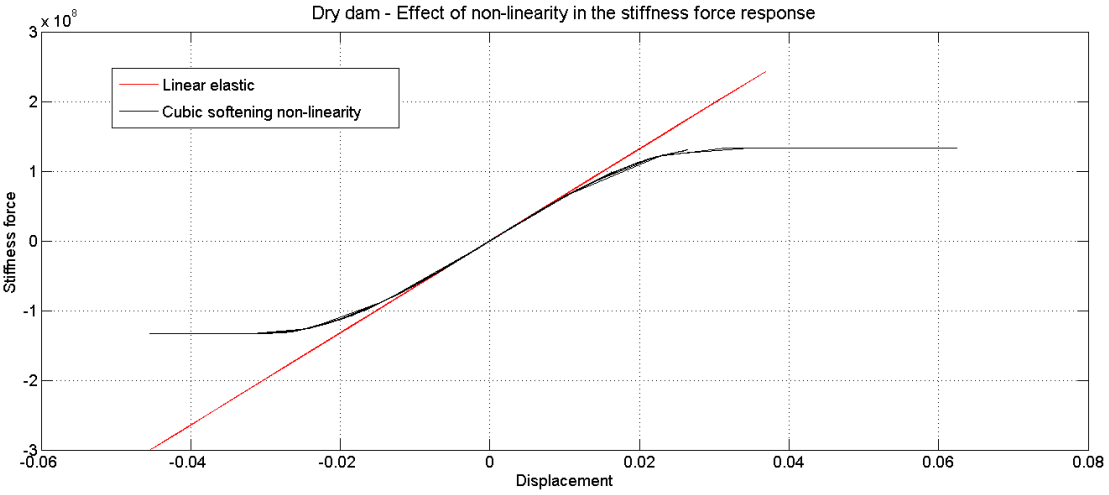


Figure 6.13: "Dry dam" case. Effect of non-linearity in the stiffness force response.

Fig 6.12 shows that the major effects of non-linear behavior occur between 2 s and 4 s. As expected, the system including softening non-linearity generates the higher displacement peak value after the time 2 s. Fig 6.13 shows that this is the most far-away point located in the plastic regime (horizontal branch), where the displacement of the system increases with no additional resisting stiffness force. After the time 4.5 s the ground acceleration loading ends and the system comes back in free vibration to the linear elastic regime, where both linear and non-linear responses damp-out with a small phase and the same magnitude.

Second, the nonlinear behavior of the frequency dependent Case I, representing a dam-reservoir interaction system with a radiation boundary of infinite extent, is studied. Figs. 6.14 and 6.15 present the effect of non-linear behavior in the displacement and stiffness internal force, respectively, of this SDOF frequency dependent system.

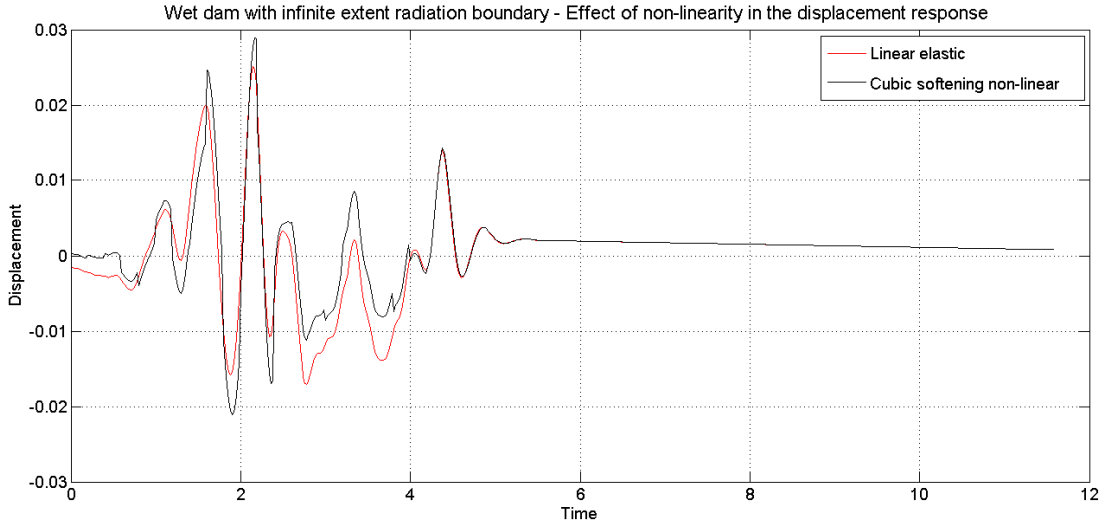


Figure 6.14: Dam-reservoir interaction with infinite extent radiation boundary (Case I). Effect of non-linearity in the displacement response.

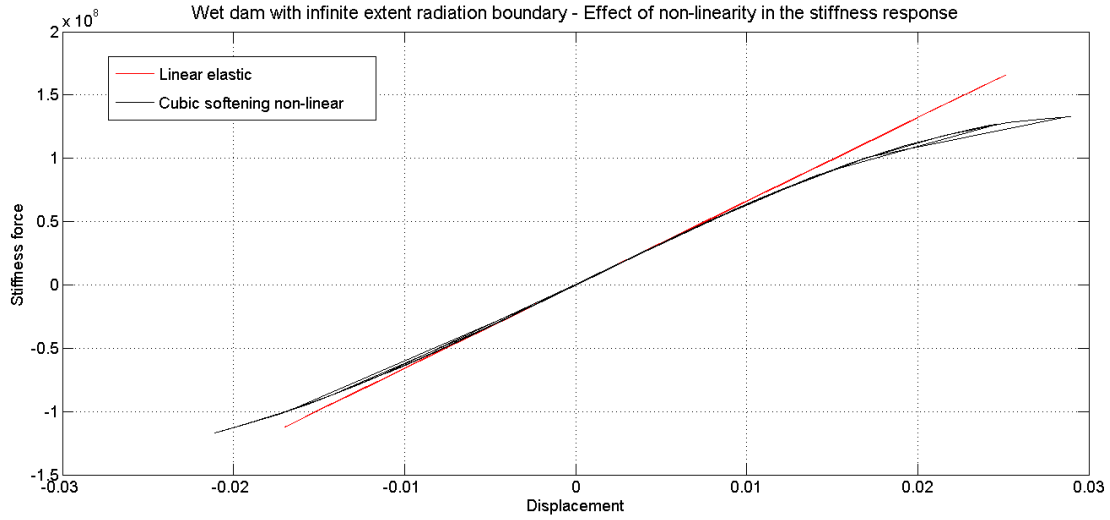


Figure 6.15: Dam-reservoir interaction with infinite extent radiation boundary (Case I). Effect of non-linearity in the stiffness force response.

Due to the additional damping introduced to the system by the infinite extent boundary condition [Eq. (6.15)], the influence of non-linear behavior is less important in this case than in the dry dam case. Certainly, Fig. 6.14 shows that there are increments of the displacement response peak values around time 2 s, however these are not as significant as in Fig. 6.12. The same can be concluded from Fig.

6.15 which shows that the displacements produced are not large enough to reach the horizontal yielding branch and develop a more important non-linear behavior.

Finally, the nonlinear behavior of the frequency dependent Case II, representing a dam-reservoir interaction system with a reflective boundary of finite extent, is studied in Figs. 6.16 and 6.17 which present the effect of non-linear behavior in the displacement and stiffness internal force, respectively.

Figure 6.16: Dam-reservoir interaction with finite extent reflective boundary (Case II). Effect of non-linearity in the displacement response.

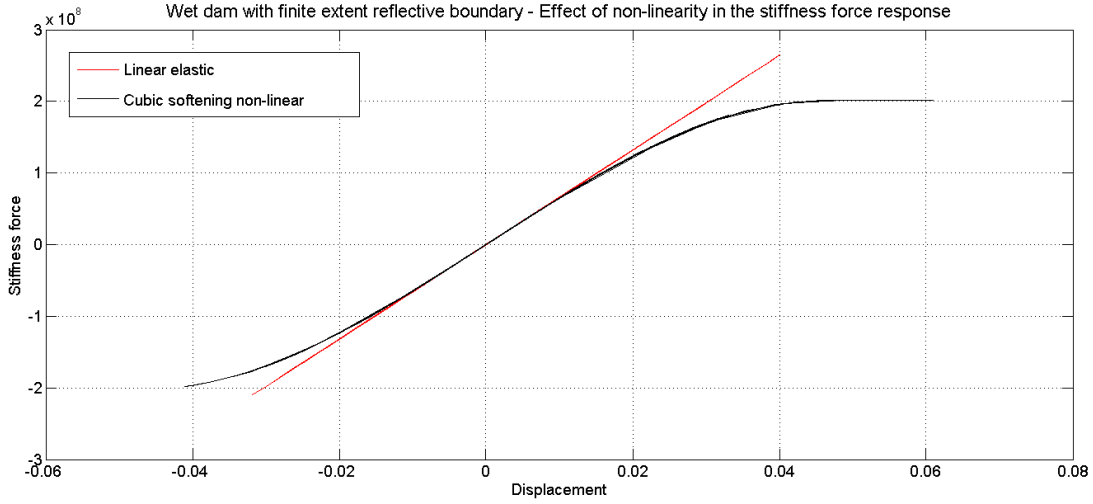


Figure 6.17: Dam-reservoir interaction with finite extent reflective boundary (Case II). Effect of non-linearity in the stiffness force response.

In this case the non-linear behavior is more important than in the radiation boundary case. As explained before, the reflective boundary condition introduces energy in the system generating large displacements even after the loading duration. Fig. 6.16 shows that all the peak values of the displacement are considerable higher for the non-linear behavior. It is important to notice that the

yielding branch of the stiffness force is reached only after time 5 s where the ground acceleration loading is not present anymore.

One final observation valid for the three cases presented is that for this type of loading the non-linear behavior of the system influences only the magnitude but not the profile of the time history response. Figs. 6.12, 6.14 and 6.16 prove that this is true no matter if the system is frequency dependent or not.

7. IMPLEMENTATION OF THE HFTD METHOD IN DIANA FOR THE SEISMIC ANALYSIS OF FOUNDATION-DAM-RESERVOIR INTERACTION PROBLEMS

7.1. Definition of foundation-dam-reservoir interaction models in DIANA[1]

7.1.1. Flow finite elements

Flow finite elements are used in DIANA for solving all types of convection-diffusion problems. For this particular case, general flow elements are used to model the water of the reservoir.

The basic variable of a general flow element are scalar potentials in their nodes, which for fluid-structure interaction models are pressure potentials (p). The derivative of these pressure potentials in each direction of analysis produces fluxes (q_x, q_y, q_z) which are located in the integration points.

Different types of 2D, 3D and axisymmetric flow elements can be defined depending on the characteristics of the model. For each type of fluid element a particular boundary element should be assigned to the fluid domain in order to define the boundary conditions.

Quadrilateral flow elements Q4HT and CQ8HT shown in Fig. 7.1 are presented and used in the modeling of two-dimensional dam-reservoir interaction modeling. The pressure field inside the flow presents a linear variation for Q4HT and quadratic for CQ8HT.

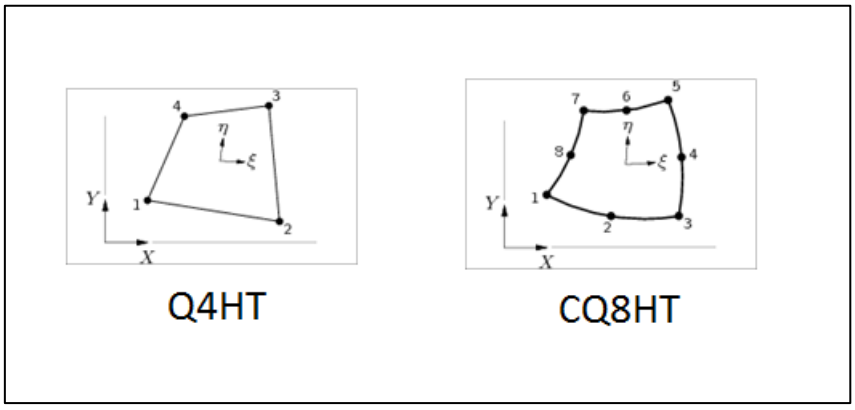


Figure 7.1: DIANA 2D flow finite elements.[2]

For each type of 2D flow element a specific flow boundary element should be attached in the edges of the fluid domain. A particular flux (or pressure derivative) in the direction perpendicular to the reservoir edge is assigned to the flow boundary element, depending on the type of boundary condition required to be modeled. As explained in Chapter 3, for the dam-reservoir interaction problem these boundary conditions are bottom absorption, radiation boundary of infinite extent and surface waves (See Section 3.1.3).

[1] Based on TNO DIANA (2011).
[2] Figures taken from TNO DIANA (2011).

Straight and curved 2D flow boundary elements, B2HT and BC3HT, respectively, are shown in Fig. 7.2. The flux q_y perpendicular to the boundary presents a linear variation along its length for B2HT and a quadratic variation for BC3HT.

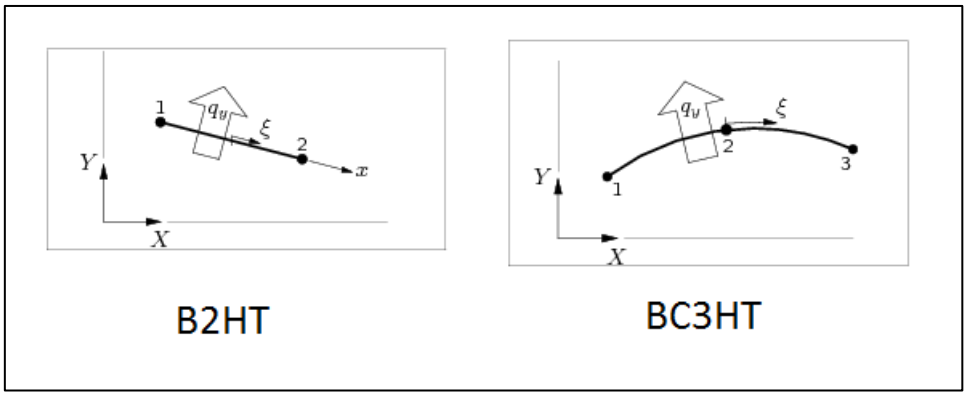


Figure 7.2: DIANA 2D boundary flow finite elements.[3]

Similarly than for the two-dimensional models, Fig. 7.3 shows two types of flow finite elements for three-dimensional models, namely HX8HT and CHX20H, respectively. BQ4HT and BCQ8HT are their corresponding flow boundary elements which are shown in Fig. 7.4.

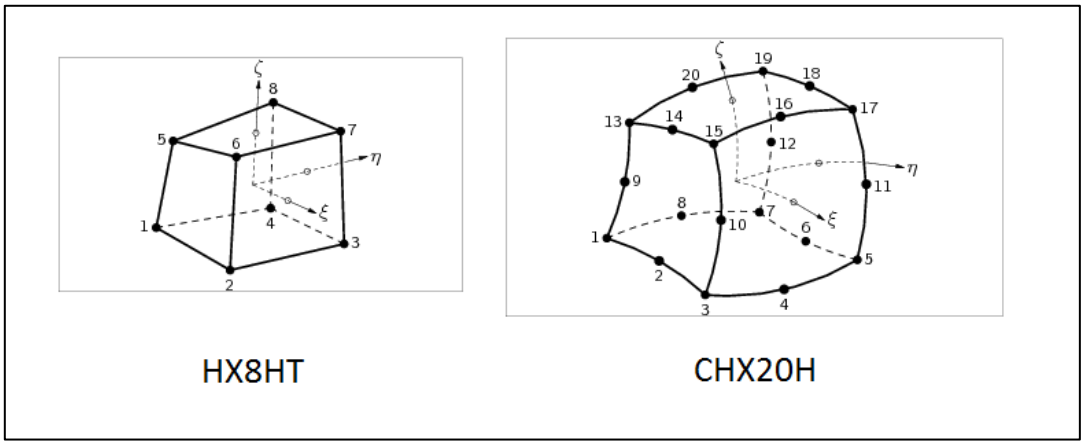


Figure 7.3: DIANA 3D flow finite elements.[4]

[3] Figures taken from TNO DIANA (2011).
[4] Figures taken from TNO DIANA (2011).

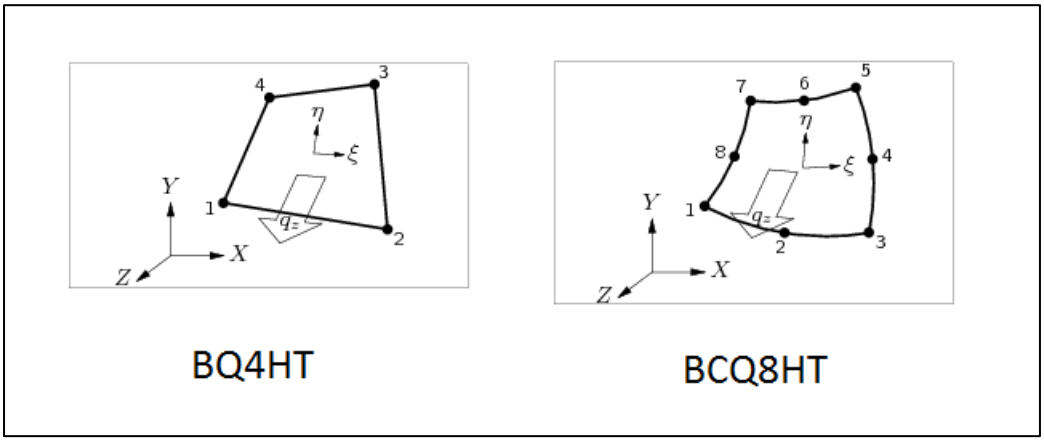


Figure 7.4: DIANA 3D boundary flow finite elements.[5]

Two material parameters should be defined for the flow elements in a fluid-structure interaction model. The first one is the conductivity (c_F) of the fluid which in this type of analysis is a "dummy" parameter which definition allows the calculation of the fluid domain conductivity matrix $\mathbf{K}_F$. If in addition the compressible effects are desired to be taken into account, then the sonic speed (c) should be defined for the compressibility matrix $\mathbf{M}_F$ to be implemented (See Section 3.1.5).

7.1.2. Fluid-structure interface

Fluid-structure interface elements are located in the common contact surface (or line) between the fluid and the structure domain. This element couples both domains by ensuring the equilibrium and continuity between the fluid pressure and the solid displacements normal to the interface.

In DIANA there are two types of fluid-structure interface elements: 1) Line interface elements for 2D or axisymmetric models; 2) Plane interface elements for 3D models. Schematic examples of both types of fluid-structure interface connecting structural and flow elements are shown in Fig. 7.5.

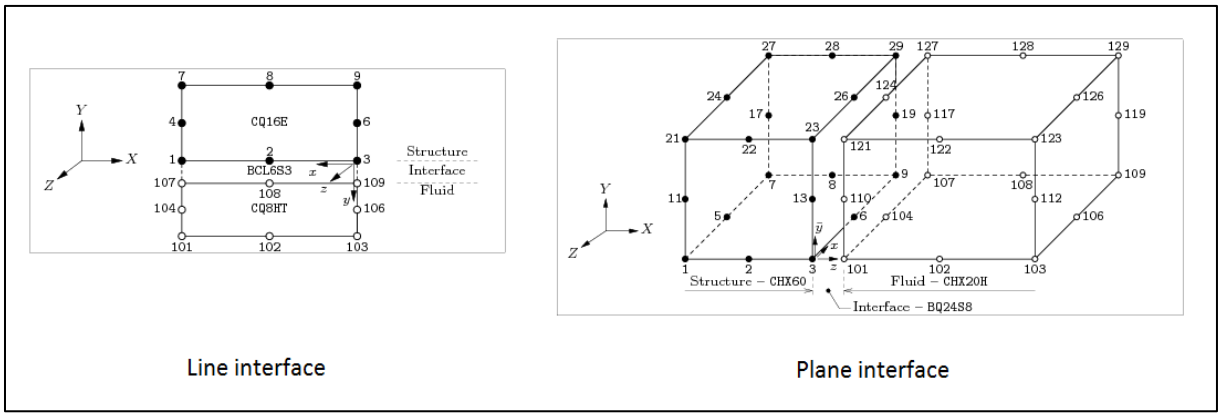


Figure 7.5: DIANA fluid-structure interface elements.[6]

[5] Figures taken from TNO DIANA (2011).
[6] Figures taken from TNO DIANA (2011).

One side of the fluid-structure interface elements are defined in terms of fluid pressure variables in a direction perpendicular to the interaction surface, whereas the other side is defined in terms of structure displacements variables in the three (or two) dimensions of analysis. For example, the hybrid variables definition of the fluid-structure interface elements is shown in Fig. 7.6 for the line interface BL4S2 and the plane interface BQ12S4.

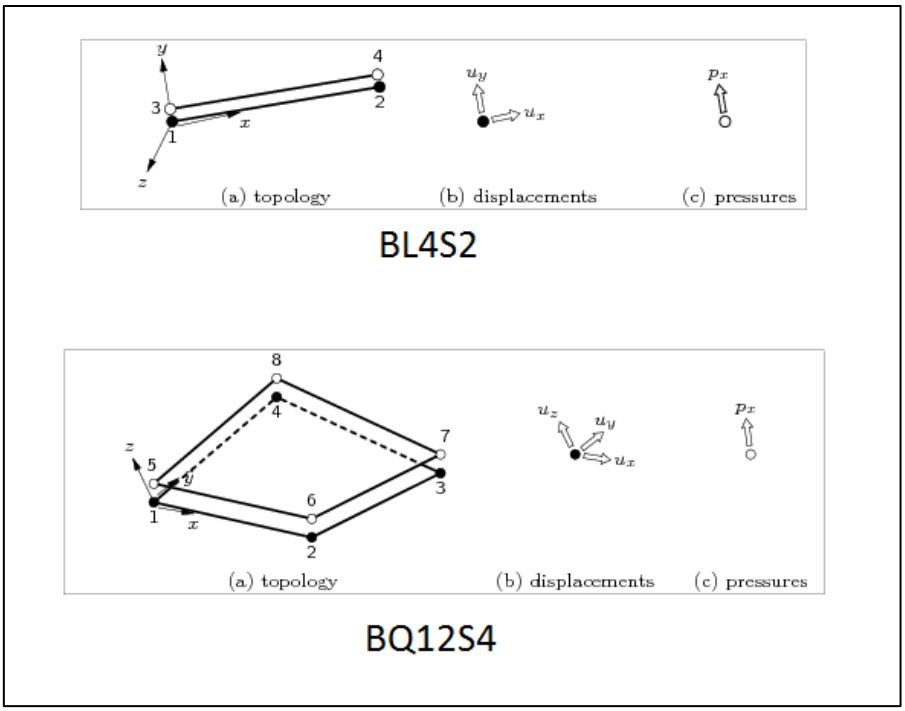


Figure 7.6: DIANA fluid-structure interface elements variables.[7]

One material parameter is required by DIANA for the definition of a fluid-structure interface property, which is the fluid density (ρ).

7.1.3. Reservoir boundary conditions

Specific material parameters should be assigned to the DIANA boundary flow elements (Section 7.1.1) in order to model the different types of reservoir boundary conditions explained in detail in Section 3.1.3.

Two types of boundary conditions were used in the present work: bottom absorption boundary and radiation boundary of infinite extent. The two material parameters required to be assigned to the boundary flow elements in order to model these two types of boundary conditions are the sonic speed (c) and the wave reflection coefficient (α), which value varies between -1 (fully absorptive) and 1 (fully reflective). For the infinite extent radiation boundary, the wave reflection coefficient is always equal to zero ($\alpha = 0$).

[7] Figures taken from TNO DIANA (2011).

The definition of these parameters is used in the calculation of the radiation matrix $\mathbf{C}_{\mathrm{F}}$ (See Section 3.1.5).

7.1.4. Foundation modeling

The soil which acts as the dam foundation is modeled with the same type of structural finite elements used to model the dam. Generally, plain strain elements are suitable for 2D models, whereas solid elements for 3D models.

In DIANA the foundation elements can be directly connected to the base of the dam if no base sliding analysis is considered. The flexibility of the foundation is taken into account by defining its linear elastic behavior parameters (f.e.: Young modulus, poison ratio), which values depend on the type of foundation (f.e.: sand, clay, rock).

The mass of the foundation is not necessarily taken into account. In structural dynamics the massless foundation is a well-established approach in which the stiffness of the foundation is included in the analysis but keeping out its mass. In this way, the foundation only plays the role of a boundary condition for the structure, and not a vibrating mass inside the model.

In DIANA the radiation far field boundary of the soil can be modeled using boundary surface elements. This is a special element to model the free field medium in dynamic analysis[8]. Another way to do this is by means of viscous boundaries consisting on concentrated dashpots located in the foundation edges. The damping coefficients are calculated for both directions, normal and tangential to the boundary surface.

In the present work the radiation far field boundary of the foundation is not taken into account. In turn the size of the soil is considerably extended in such a way that the soil boundary reflective effects have no influence in the dam's vibration.

7.1.5. Dam's concrete cracking non-linear behavior

The total strain rotating crack model was used in DIANA to model the non-linear concrete cracking behavior of the dam. Under this approach both the tensile and compressive behavior of the concrete is defined by means of a stress-strain curve which is used for determining the non-linear internal force vector, and therefore the pseudo force vector required by the HFTD method (See Chapter 4), for each time step.

For the concrete tensile behavior different types of predefined tension softening functions shown in Fig. 7.7 can be chosen in DIANA. Similarly, for the concrete crushing behavior the predefined compression functions shown in Fig. 7.8 are available. Depending on the type of function, different parameters are required to be defined[9].

[8] For detailed information about boundary surface elements see TNO DIANA (2011).
[9] For detailed information about total strain crack models see TNO DIANA (2011).

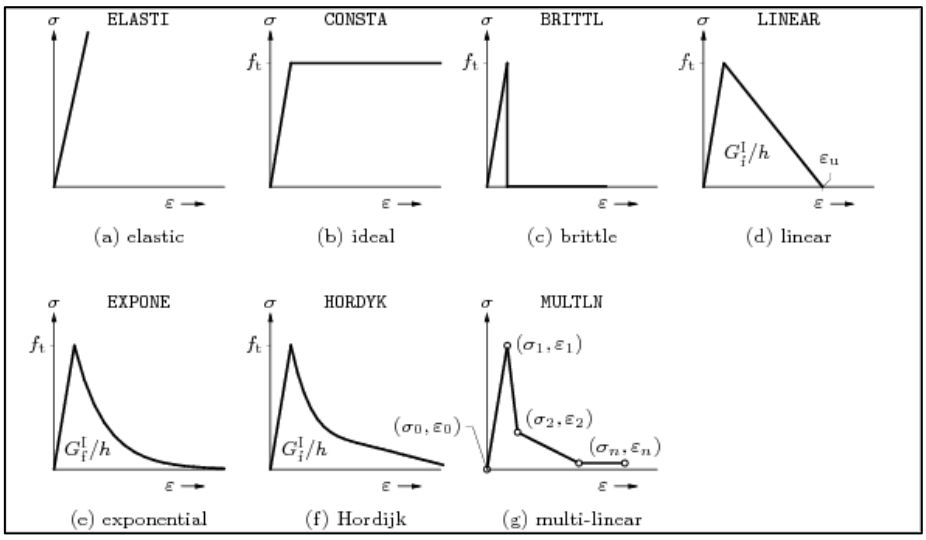


Figure 7.7: DIANA predefined tension softening functions for total strain crack model.[10]

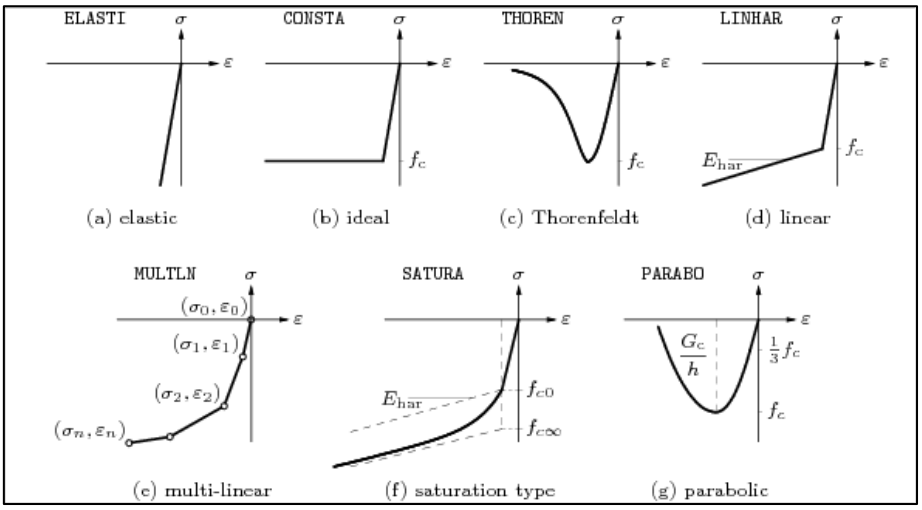


Figure 7.8: DIANA predefined compression functions for total strain crack model.[11]

7.2. HFTD implementation in DIANA

Hybrid Frequency-Time Domain method was programmed, codified and implement in DIANA based on the theoretical background of Chapter 4. The flowcharts of the HFTD implementation in DIANA are reproduced in Appendix A. In addition, the command file for the execution of the HFTD analysis in DIANA is shown in Appendix C.

[10] Figures taken from TNO DIANA (2011).
[11] Figures taken from TNO DIANA (2011).

7.3. Case study: 2D foundation-dam-reservoir interaction model

7.3.1. Case study description

The finite element mesh of the 2D case study is shown in Fig. 7.9. The model is composed of three parts which are the dam (cyan), the reservoir (yellow) and the foundation (green). The gravity dam's height is 100 m, whereas its base length is 80 m. The reservoir's length is considered to be 390 m, whereas the length and depth of the soil foundation block are 720 m and 200 m, respectively.

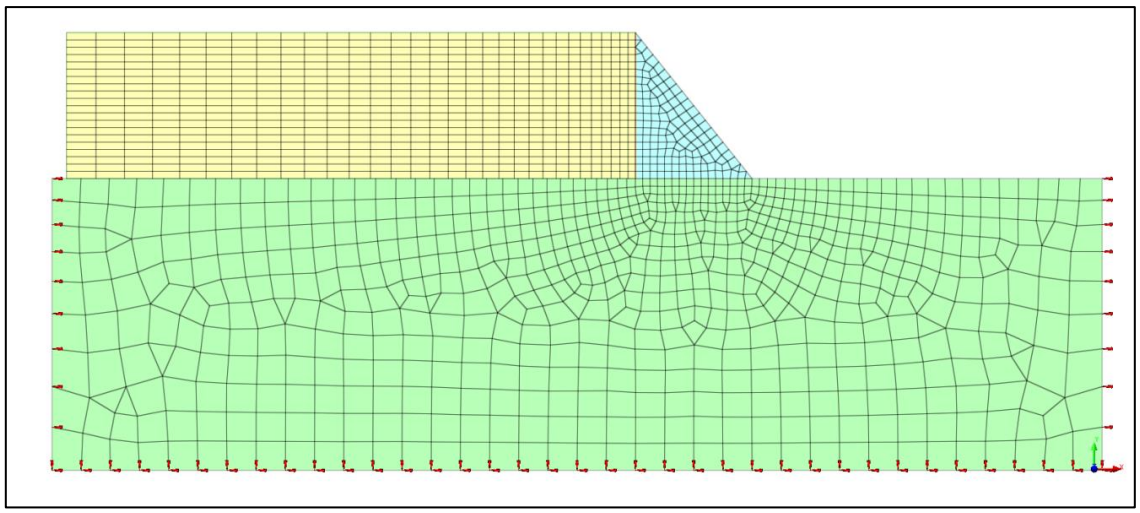


Figure 7.9: DIANA model of the 2D foundation-dam-reservoir interaction case study.

The types of finite elements used for each component of the model are indicated in Table 7.1. The dam and foundation elements are the typical quadrilateral (Q8EPS) and triangular (T6EPS) plane strain elements with linear interpolation. The characteristics of the reservoir flow and fluid-structure interface elements are explained in Section 7.1.

The values of the linear and non-linear material parameters used in this case study for the dam, the foundation and the reservoir (fluid domain and boundary parameters) are listed in Tables 7.2, 7.3 and 7.4, respectively.

Table 7.1: Type of DIANA finite elements used for the case study modeling

Component	Type	Element
Dam	Plane strain 2D	Q8EPS T6EPS
Foundation	Plane strain 2D	Q8EPS T6EPS
Reservoir	General flow 2D	Q4HT
Reservoir boundaries	Flow boundary 2D	B2HT
Reservoir-dam interface	Fluid-structure interface 1D	BL4S2

Table 7.2: Material parameters for the concrete dam

Parameter	DIANA variable name	Value / Type	Units
Modulus of elasticity	YOUNG	2.7×10^{10}	N / m^2
Poison modulus	POISON	1.67×10^{-1}	-
Density	DENSIT	2.4×10^{3}	kg / m^3
Rayleigh damping	RAYLEI	8.19456×10^{-1} 1.38396×10^{-3}	- -
Softening tension curve	TENCRV	LINEAR	-
Tensile strength	TENSTR	3.8×10^{6}	N / m^2
Ultimate tensile strain	EPSULT	5.0×10^{-4}	-
Compression curve	COMCRV	CONSTA	-
Compressive strength	COMSTR	3.5×10^{7}	N / m^2

Table 7.3: Material parameters for the foundation soil

Parameter	DIANA variable name	Value / Type	Units
Modulus of elasticity	YOUNG	2.5×10^{10}	N / m^2
Poison modulus	POISON	2.0×10^{-1}	-
Density	DENSIT	0	kg / m^3

Table 7.4: Material parameters for the reservoir fluid

Parameter	DIANA variable name	Value / Type	Units
Conductivity	CONDUC	2.5×10^{10}	N / m^2
Sonic speed	CSOUND	1.44×10^{3}	m / s
Density	DENSIT	1.0×10^{3}	kg / m^3
Wave reflection coefficient for infinite extent boundary	ALPHAB	0	-
Wave reflection coefficient for bottom absorption boundary	ALPHAB	5.0×10^{-1}	-

Finally, the foundation-dam-reservoir interaction system is subject to a base acceleration loading in the horizontal direction which is applied in the supports located in the foundation edges (See Fig. 7.9).

For transient and HFTD analysis the El Centro earthquake ground acceleration signal shown in Fig. 7.10 is defined as the loading input. On the other hand, for direct frequency analysis the corresponding ground acceleration spectrum shown in Fig. 7.11 is taken into account.

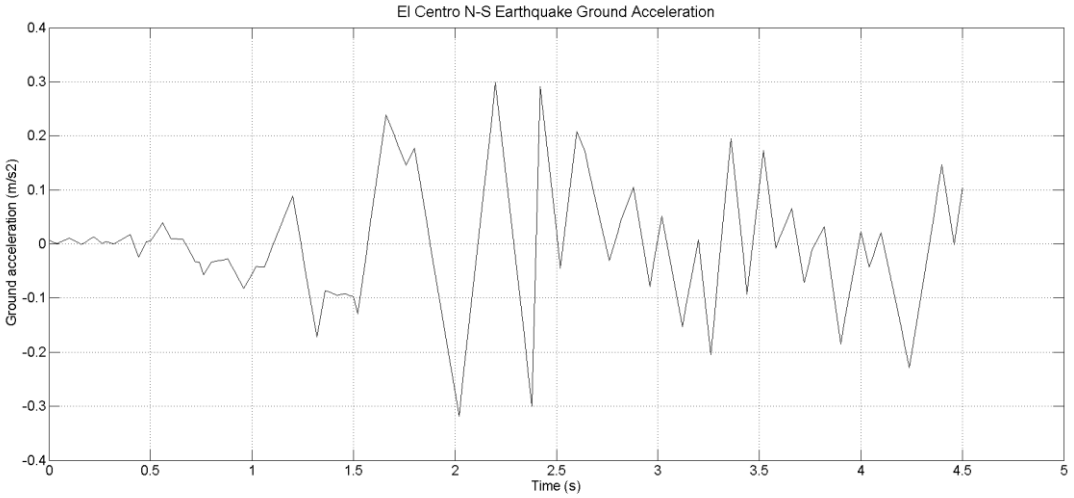


Figure 7.10: El Centro N-S earthquake ground acceleration signal.

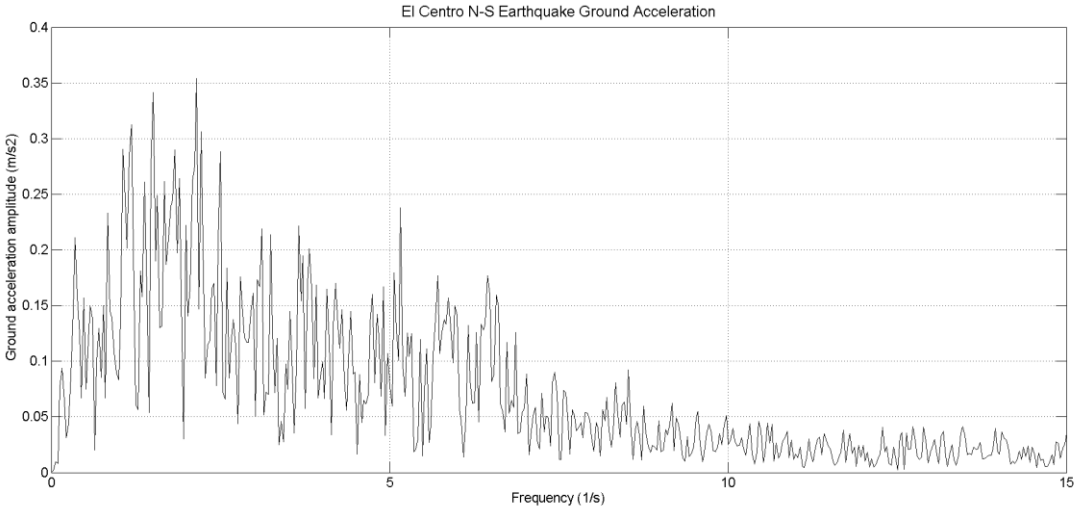


Figure 7.11: El Centro N-S earthquake ground acceleration spectrum.

7.3.2. Eigen-analysis

The eigen-analysis of the dam was performed following the two procedures presented and explained in Section 3.1.9. The first procedure, known as "dry dam" eigen-analysis, does not take into account the influence of the reservoir in the calculations, and therefore, it solves the free vibration equation of motion of the structures given by Eq. (3.73) in terms of the mass ($\mathbf{M}_S$) and stiffness ($\mathbf{K}_S$) matrices.

$$\mathbf{M}_S\ddot{\mathbf{u}} + \mathbf{K}_S\mathbf{u} = \mathbf{0} \qquad \textbf{(3.73)}$$

On the other hand, the second procedure, known as "wet dam" eigen-analysis, does take into account the influence of the fluid reservoir but assuming that it is incompressible and therefore, without radiation but reflective boundary conditions. In this approach the reservoir introduces in the free vibration equation of motion, given by Eq. (3.75), a fluid added mass term ($\widetilde{\mathbf{M}}_{\mathrm{F}}$) defined by Eq. (3.72) (See Chapter 3).

$$\widetilde{\mathbf{M}}_{\mathrm{F}} = \rho_0 \mathbf{R}^{\mathrm{T}} \mathbf{K}_F^{-1} \mathbf{R} \tag{3.72}$$

$$\left(\mathbf{M}_{\mathrm{S}} + \widetilde{\mathbf{M}}_{\mathrm{F}}\right)\ddot{\mathbf{u}} + \mathbf{K}_{\mathrm{S}}\mathbf{u} = \mathbf{0} \tag{3.75}$$

The dry dam eigen-frequencies and effective mass participation in both directions of analysis are presented in Fig. 7.12 for the first 15 modes of vibration. The fundamental vibration frequency for the dam without reservoir interaction is 3.33 Hz. Besides, it can be noticed that the first modes of vibration concentrate more than 90% of the vibration mass in both directions, being the first and third modes the most important for X and Y direction, respectively.

MODE	FREQUENCY	EFF.MASS TX	PERCENTAGE	CUM.PERCENT.
1	0.33345E+01	0.62435E+07	0.65036E+02	0.65036E+02
2	0.65239E+01	0.98313E+06	0.10241E+02	0.75277E+02
3	0.78315E+01	0.18548E+07	0.19320E+02	0.94598E+02
4	0.13771E+02	0.40431E+06	0.42116E+01	0.98809E+02
5	0.19386E+02	0.44069E+02	0.45905E-03	0.98810E+02
6	0.21650E+02	0.78850E+05	0.82135E+00	0.99631E+02
7	0.26757E+02	0.44294E+03	0.46139E-02	0.99636E+02
8	0.29083E+02	0.43391E+04	0.45199E-01	0.99681E+02
9	0.32387E+02	0.11596E+05	0.12079E+00	0.99802E+02
10	0.34115E+02	0.24625E+04	0.25651E-01	0.99827E+02
11	0.37617E+02	0.90609E+02	0.94384E-03	0.99828E+02
12	0.39882E+02	0.69189E+04	0.72071E-01	0.99900E+02
13	0.43091E+02	0.33208E+03	0.34592E-02	0.99904E+02
14	0.46134E+02	0.37105E+01	0.38651E-04	0.99904E+02
15	0.47507E+02	0.12853E+04	0.13389E-01	0.99917E+02

MODE	FREQUENCY	EFF.MASS TY	PERCENTAGE	CUM.PERCENT.
1	0.33345E+01	0.34599E+06	0.36040E+01	0.36040E+01
2	0.65239E+01	0.81849E+07	0.85260E+02	0.88864E+02
3	0.78315E+01	0.79899E+06	0.83228E+01	0.97186E+02
4	0.13771E+02	0.65815E+05	0.68557E+00	0.97872E+02
5	0.19386E+02	0.16294E+06	0.16973E+01	0.99569E+02
6	0.21650E+02	0.28339E+04	0.29520E-01	0.99599E+02
7	0.26757E+02	0.30375E+03	0.31640E-02	0.99602E+02
8	0.29083E+02	0.15947E+04	0.16612E-01	0.99619E+02
9	0.32387E+02	0.19892E+04	0.20721E-01	0.99639E+02
10	0.34115E+02	0.17227E+05	0.17945E+00	0.99819E+02
11	0.37617E+02	0.55119E+04	0.57416E-01	0.99876E+02
12	0.39882E+02	0.16048E+04	0.16717E-01	0.99893E+02
13	0.43091E+02	0.16600E+03	0.17292E-02	0.99895E+02
14	0.46134E+02	0.39263E+03	0.40899E-02	0.99899E+02
15	0.47507E+02	0.90613E+01	0.94388E-04	0.99899E+02

Figure 7.12: Eigen-analysis results for the empty reservoir or dry dam case

The total mass of the dam is equal to 9.6 x 10^6 kg. When the wet dam eigen-analysis is performed, the calculated fluid added mass is equal to 5.42 x 10^6, which means a mass increment of more than

50%. This considerable additional mass makes the system more flexible and therefore the eigen-frequencies obtained are smaller than in the dry dam case.

Indeed, this is confirmed by the fundamental eigen-frequency for the wet dam case shown in Fig. 7.13, which is reduced to 2.62 Hz. However, even since the first mode, the effective mass participating in the X-direction is higher than 115%, which means that the fluid added mass actively contributes in the dam's horizontal vibration. This is not the case for the vertical vibration (Y-direction) where the fluid mass has no participation at all, due to the vertical orientation of the dam-reservoir interface.

MODE	FREQUENCY	EFF.MASS TX	PERCENTAGE	CUM.PERCENT.
1	0.26204E+01	0.11052E+08	0.11513E+03	0.11513E+03
2	0.63200E+01	0.22440E+07	0.23375E+02	0.13850E+03
3	0.69574E+01	0.10972E+07	0.11429E+02	0.14993E+03
4	0.12652E+02	0.48667E+06	0.50695E+01	0.15500E+03
5	0.18754E+02	0.86948E+04	0.90571E-01	0.15509E+03
6	0.20567E+02	0.84918E+05	0.88457E+00	0.15598E+03
7	0.22823E+02	0.34150E+04	0.35573E-01	0.15601E+03
8	0.26900E+02	0.73169E+04	0.76217E-01	0.15609E+03
9	0.30597E+02	0.11594E+05	0.12077E+00	0.15621E+03
10	0.32791E+02	0.29189E+04	0.30405E-01	0.15624E+03
11	0.36107E+02	0.14004E+04	0.14587E-01	0.15626E+03
12	0.37106E+02	0.42105E+04	0.43860E-01	0.15630E+03
13	0.41099E+02	0.36538E+04	0.38060E-01	0.15634E+03
14	0.43149E+02	0.72955E+02	0.75994E-03	0.15634E+03
15	0.44959E+02	0.30529E+03	0.31801E-02	0.15634E+03

MODE	FREQUENCY	EFF.MASS TY	PERCENTAGE	CUM.PERCENT.
1	0.26204E+01	0.15964E+06	0.16629E+01	0.16629E+01
2	0.63200E+01	0.55293E+07	0.57597E+02	0.59260E+02
3	0.69574E+01	0.36087E+07	0.37591E+02	0.96851E+02
4	0.12652E+02	0.80358E+05	0.83706E+00	0.97688E+02
5	0.18754E+02	0.16388E+06	0.17071E+01	0.99395E+02
6	0.20567E+02	0.10181E+05	0.10605E+00	0.99501E+02
7	0.22823E+02	0.81938E+04	0.85352E-01	0.99586E+02
8	0.26900E+02	0.62153E+02	0.64742E-03	0.99587E+02
9	0.30597E+02	0.46919E+04	0.48874E-01	0.99636E+02
10	0.32791E+02	0.11823E+05	0.12316E+00	0.99759E+02
11	0.36107E+02	0.89186E+04	0.92902E-01	0.99852E+02
12	0.37106E+02	0.24928E+04	0.25966E-01	0.99878E+02
13	0.41099E+02	0.33110E+03	0.34489E-02	0.99881E+02
14	0.43149E+02	0.12835E+04	0.13370E-01	0.99894E+02
15	0.44959E+02	0.53183E+02	0.55399E-03	0.99895E+02

Figure 7.13: Eigen-analysis results for the full reservoir or wet dam case

The remarkable influence of the fluid added mass in the eigen-frequency values is not replicated in the mode shapes. Fig. 7.14 shows the modes shapes of the 1st, 2nd and 4th vibration modes in the X-direction. It can be noticed that the first mode shape is not influenced by the added fluid mass, no matter that the eigen-frequency experiences a considerable reduction. Small magnitude differences can be appreciated in the second and fourth mode shapes.

It is due to these negligible differences that the HFTD method uses the mode shapes of the empty reservoir eigen-analysis for the mode-superposition method.

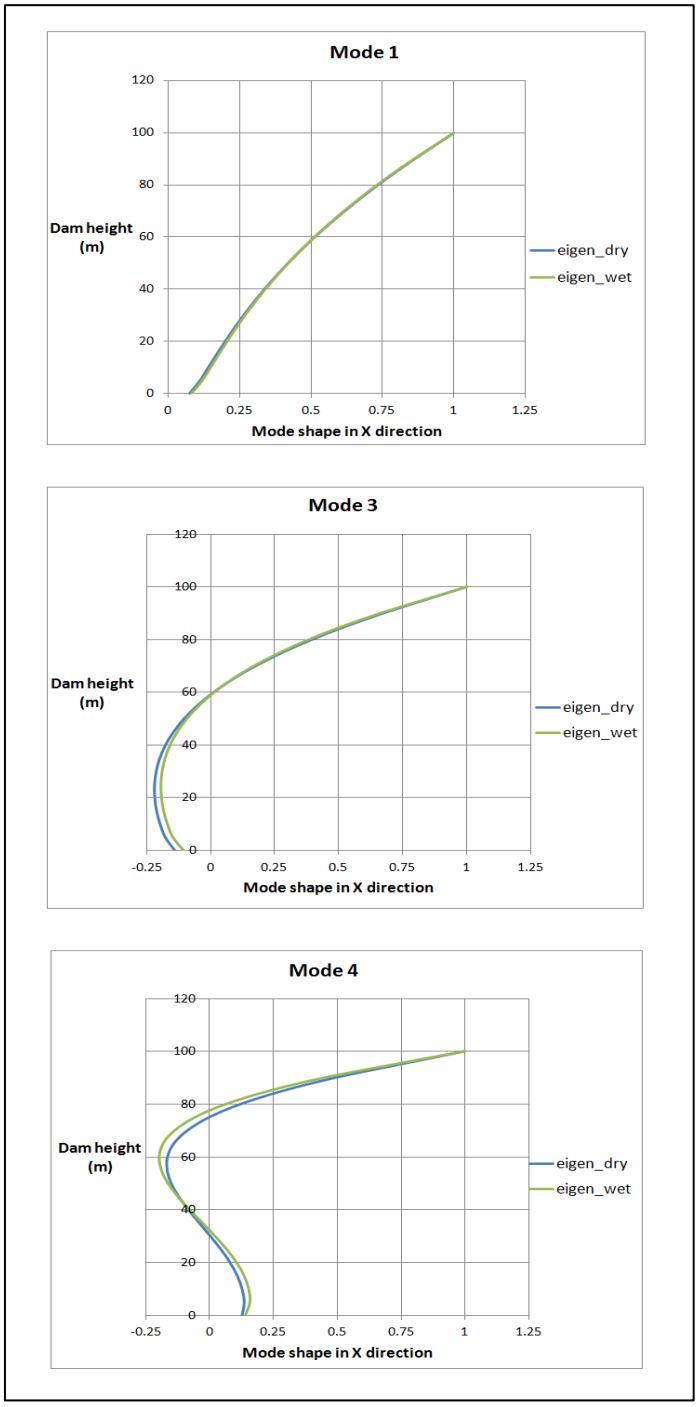


Figure 7.14: First three mode shapes in the X-direction for empty (eigen-dry) and full (eigen-wet) reservoir cases

Finally, the influence of the contribution of the initial conditions in the geometric stiffness matrix, and therefore in the eigen-analysis results, was investigated. For this purpose, the gravity load of the dam and the hydrostatic pressure applied in the dam's face and in the soil below the reservoir were included as initial conditions.

As shown in Figs. 7.15 and 7.16 for the dry and wet dam cases, respectively, the influence of the initial loading conditions in the stiffness matrix of the structure is not significant. For this reason, the eigen-analysis results presented in Figs. 7.15 and 7.16 are almost the same as those presented in Figs. 7.12 and 7.13.

MODE	FREQUENCY	EFF.MASS TX	PERCENTAGE	CUM.PERCENT.
1	0.33338E+01	0.62429E+07	0.65031E+02	0.65031E+02
2	0.65237E+01	0.98345E+06	0.10244E+02	0.75275E+02
3	0.78308E+01	0.18549E+07	0.19322E+02	0.94597E+02
4	0.13770E+02	0.40440E+06	0.42125E+01	0.98809E+02
5	0.19386E+02	0.43280E+02	0.45083E-03	0.98810E+02
6	0.21649E+02	0.78861E+05	0.82147E+00	0.99631E+02
7	0.26757E+02	0.44436E+03	0.46288E-02	0.99636E+02
8	0.29082E+02	0.43410E+04	0.45219E-01	0.99681E+02
9	0.32386E+02	0.11595E+05	0.12078E+00	0.99802E+02
10	0.34115E+02	0.24620E+04	0.25646E-01	0.99827E+02
11	0.37616E+02	0.90945E+02	0.94734E-03	0.99828E+02
12	0.39880E+02	0.69199E+04	0.72082E-01	0.99900E+02
13	0.43090E+02	0.33164E+03	0.34546E-02	0.99904E+02
14	0.46132E+02	0.36816E+01	0.38350E-04	0.99904E+02
15	0.47506E+02	0.12860E+04	0.13396E-01	0.99917E+02

MODE	FREQUENCY	EFF.MASS TY	PERCENTAGE	CUM.PERCENT.
1	0.33338E+01	0.34598E+06	0.36039E+01	0.36039E+01
2	0.65237E+01	0.81844E+07	0.85254E+02	0.88858E+02
3	0.78308E+01	0.79949E+06	0.83280E+01	0.97186E+02
4	0.13770E+02	0.65826E+05	0.68569E+00	0.97872E+02
5	0.19386E+02	0.16293E+06	0.16972E+01	0.99569E+02
6	0.21649E+02	0.28398E+04	0.29582E-01	0.99599E+02
7	0.26757E+02	0.30407E+03	0.31674E-02	0.99602E+02
8	0.29082E+02	0.15945E+04	0.16609E-01	0.99619E+02
9	0.32386E+02	0.19888E+04	0.20716E-01	0.99639E+02
10	0.34115E+02	0.17225E+05	0.17943E+00	0.99819E+02
11	0.37616E+02	0.55136E+04	0.57433E-01	0.99876E+02
12	0.39880E+02	0.16057E+04	0.16726E-01	0.99893E+02
13	0.43090E+02	0.16596E+03	0.17287E-02	0.99895E+02
14	0.46132E+02	0.39234E+03	0.40868E-02	0.99899E+02
15	0.47506E+02	0.90384E+01	0.94150E-04	0.99899E+02

Figure 7.15: Eigen-analysis results for the empty reservoir or dry dam case including initial loading conditions

MODE	FREQUENCY	EFF.MASS TX	PERCENTAGE	CUM.PERCENT.
1	0.26199E+01	0.11052E+08	0.11512E+03	0.11512E+03
2	0.63197E+01	0.22457E+07	0.23393E+02	0.13851E+03
3	0.69569E+01	0.10962E+07	0.11418E+02	0.14993E+03
4	0.12651E+02	0.48677E+06	0.50705E+01	0.15500E+03
5	0.18753E+02	0.87094E+04	0.90723E-01	0.15509E+03
6	0.20566E+02	0.84919E+05	0.88458E+00	0.15598E+03
7	0.22823E+02	0.34115E+04	0.35537E-01	0.15601E+03
8	0.26899E+02	0.73213E+04	0.76263E-01	0.15609E+03
9	0.30595E+02	0.11592E+05	0.12075E+00	0.15621E+03
10	0.32790E+02	0.29181E+04	0.30397E-01	0.15624E+03
11	0.36106E+02	0.14023E+04	0.14607E-01	0.15626E+03
12	0.37104E+02	0.42114E+04	0.43869E-01	0.15630E+03
13	0.41098E+02	0.36526E+04	0.38048E-01	0.15634E+03
14	0.43148E+02	0.72933E+02	0.75972E-03	0.15634E+03
15	0.44958E+02	0.30531E+03	0.31803E-02	0.15634E+03

MODE	FREQUENCY	EFF.MASS TY	PERCENTAGE	CUM.PERCENT.
1	0.26199E+01	0.15965E+06	0.16630E+01	0.16630E+01
2	0.63197E+01	0.55257E+07	0.57559E+02	0.59222E+02
3	0.69569E+01	0.36123E+07	0.37628E+02	0.96850E+02
4	0.12651E+02	0.80375E+05	0.83724E+00	0.97688E+02
5	0.18753E+02	0.16386E+06	0.17069E+01	0.99394E+02
6	0.20566E+02	0.10197E+05	0.10622E+00	0.99501E+02
7	0.22823E+02	0.81949E+04	0.85364E-01	0.99586E+02
8	0.26899E+02	0.62215E+02	0.64808E-03	0.99587E+02
9	0.30595E+02	0.46912E+04	0.48867E-01	0.99636E+02
10	0.32790E+02	0.11822E+05	0.12314E+00	0.99759E+02
11	0.36106E+02	0.89181E+04	0.92897E-01	0.99852E+02
12	0.37104E+02	0.24956E+04	0.25996E-01	0.99878E+02
13	0.41098E+02	0.33112E+03	0.34492E-02	0.99881E+02
14	0.43148E+02	0.12832E+04	0.13366E-01	0.99894E+02
15	0.44958E+02	0.53362E+02	0.55585E-03	0.99895E+02

Figure 7.16: Eigen-analysis results for the full reservoir or wet dam case including initial loading conditions

7.3.3. Frequency domain linear elastic analysis

With the objective of evaluate the effect of the frequency dependent properties introduced to the dam-reservoir interaction system by fluid compressibility and reservoir boundary conditions, like infinite radiation extent and bottom absorption, a linear elastic frequency domain analysis using the Direct Method[12] is performed for the following five cases which are illustrated in Fig. 7.17.

- Case I: Incompressible fluid.
- Case II: Compressible fluid with reflective reservoir boundaries.
- Case III: Compressible fluid with bottom absorption boundary.
- Case IV: Compressible fluid with infinite extent radiation boundary.
- Case V: Compressible fluid with bottom absorption and infinite extent radiation boundaries.

The ground acceleration spectrum shown in Fig. 7.11 is applied to the five systems shown Fig. 7.17, and as a consequence, the Amplitude-Frequency response plots given by Figs. 7.18 to 7.21 give an

[12] For theoretical background of the frequency domain analysis see Chapter 2.

illustrative insight of the effect that compressibility and reservoir boundary conditions have in the dynamic behavior of dam-reservoir interaction systems.

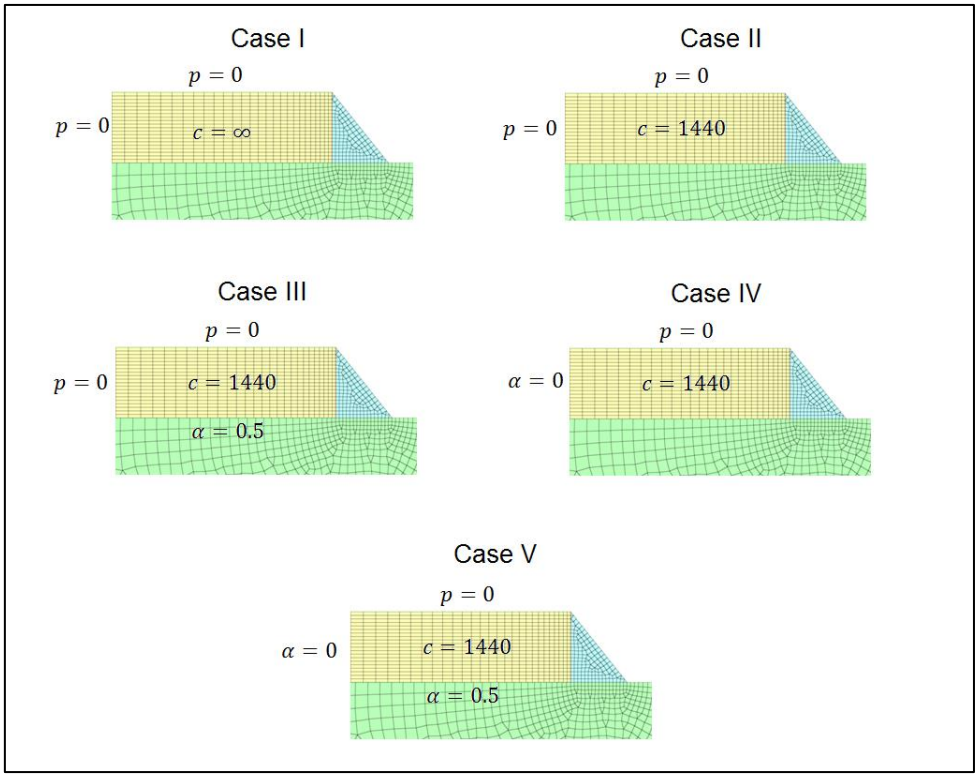


Figure 7.17: Direct frequency linear elastic analysis cases.

Figs. 18 and 19 show the relative Displacement Amplitude-Frequency plot, in the X and Y directions respectively, of the dam's crest node (higher point of the dam).

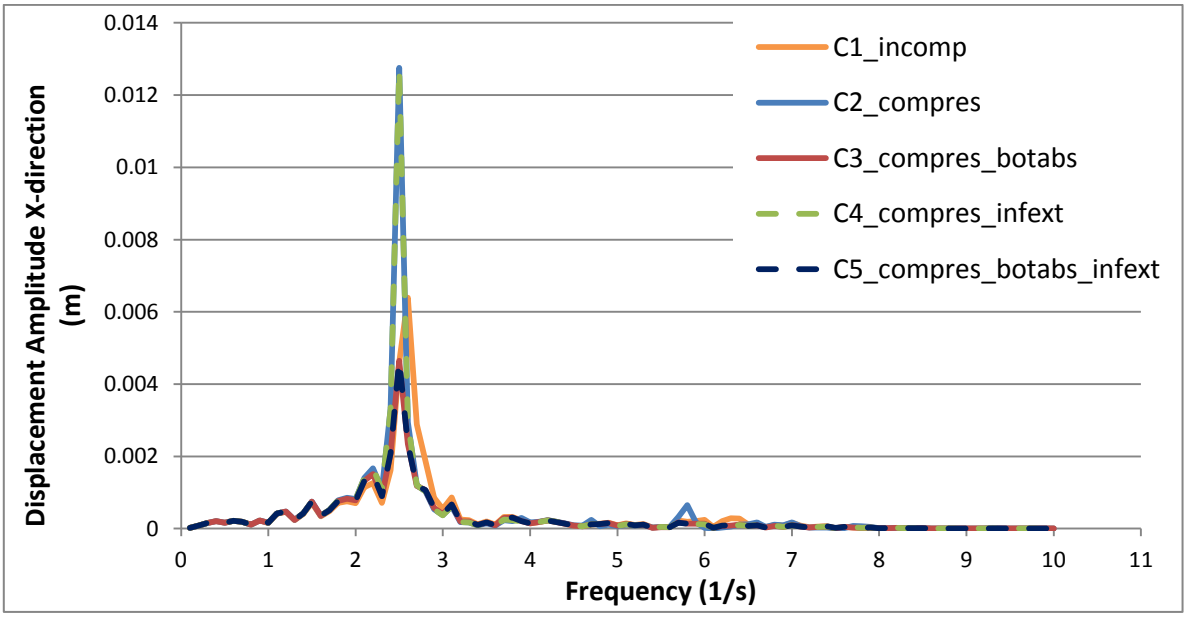


Figure 7.18: X-direction Displacement Amplitude-Frequency plot of the dam's crest.

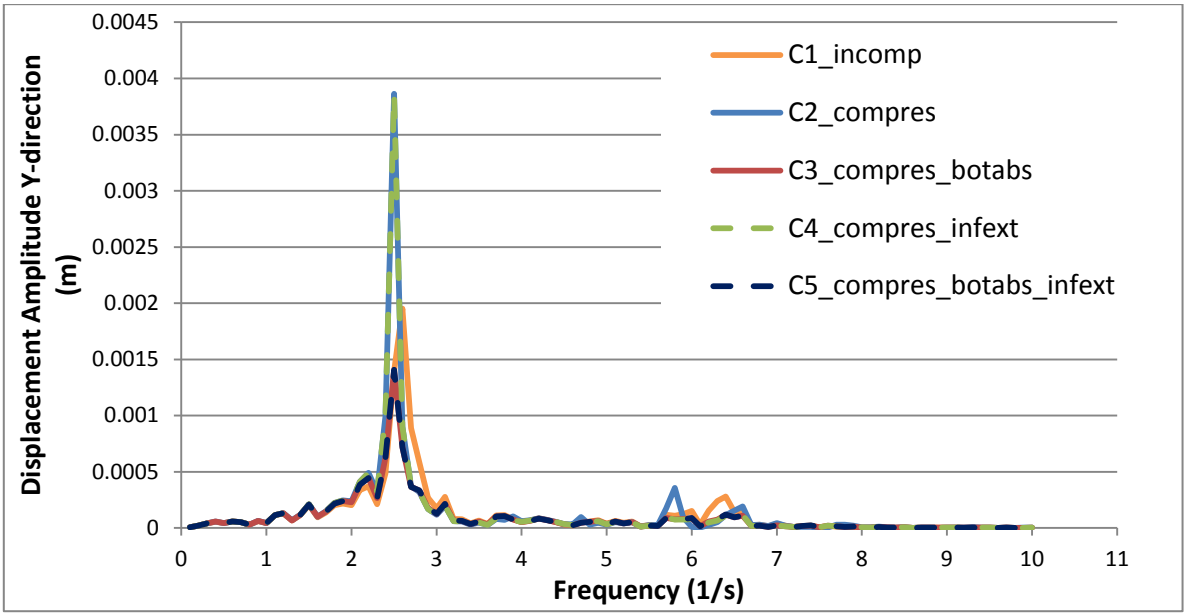


Figure 7.19: Y-direction Displacement Amplitude-Frequency plot of the dam's crest.

For all the case studies the higher amplitudes are obtained around the fundamental vibration frequency of the wet-dam system equal to 2.62 Hz (See Figs. 7.13 and 7.16).

Comparing the cases C1 and C2 it can be concluded that assuming that the fluid of the reservoir is compressible produces an increment of the displacement amplitudes. The opposite occurs in case C3 for which a bottom absorption boundary condition is introduced. This is due to the additional damping introduced to the system by the bottom absorption (See Chapter 3).

On the other hand, in this case study the introduction of infinite extent radiation boundary condition (cases C4 and C5) does not influences decisively the magnitude of the responses around the fundamental frequency of vibration. Only for the higher vibration frequencies around 6 Hz the introduction of the infinite extent radiation boundary damps out the peaks of Cases C1 and C2. The main reason for the negligible contribution of the infinite extent boundary in this case study is the long extension of the reservoir, defined in the model and equal to 390 m.

Finally, Figs. 20 and 21 present the relative velocity and acceleration, respectively, of the dam's crest node in the X-direction. The same observations made for the displacement response are applicable to both velocity and acceleration amplitudes, with the only difference that the importance of compressibility and infinite extent radiation boundary is considerably higher for acceleration (Fig. 21) than for any other response.

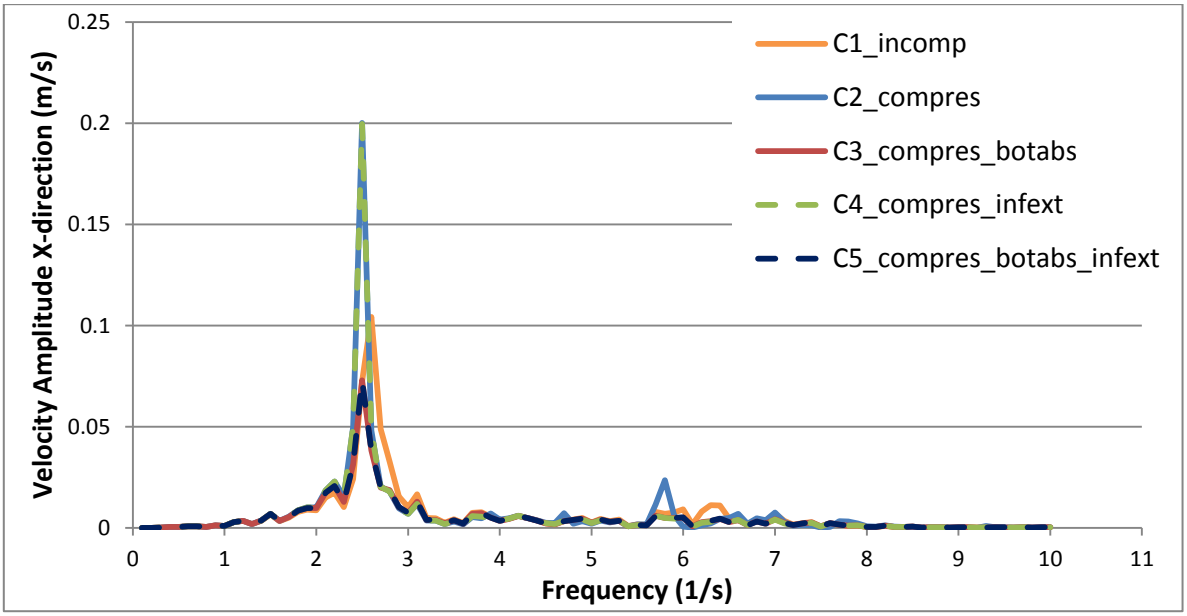


Figure 7.20: X-direction Velocity Amplitude-Frequency plot of the dam's crest.

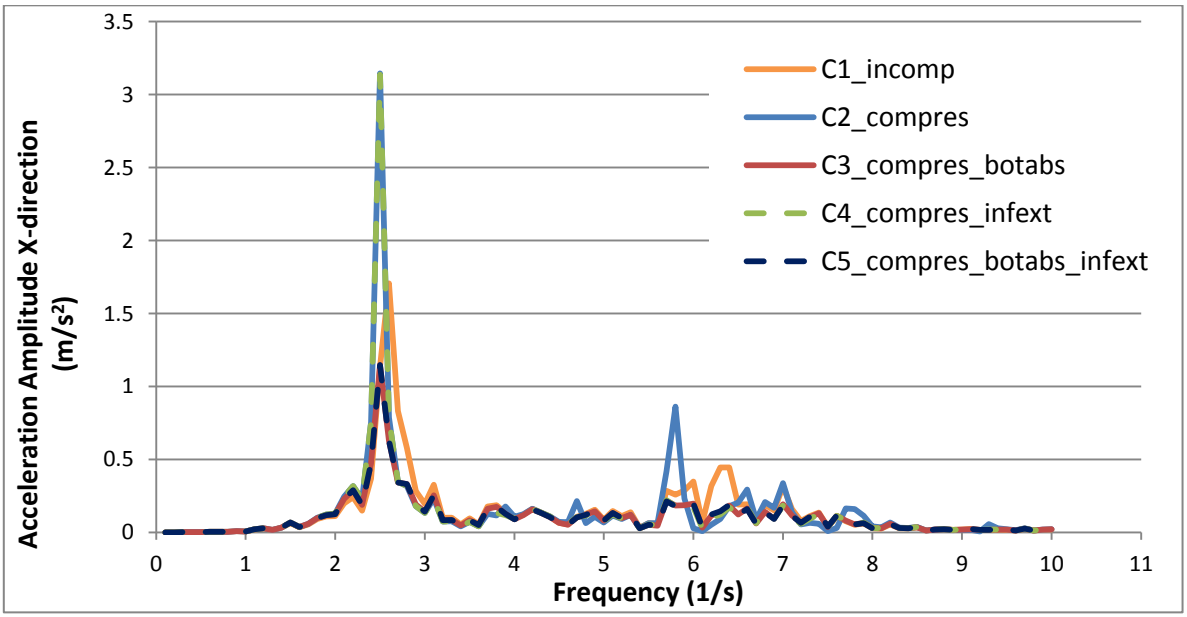


Figure 7.21: X-direction Acceleration Amplitude-Frequency plot of the dam's crest.

7.3.4. Transient analysis: Linear and non-linear seismic behavior

Transient linear and non-linear analysis of the 2D foundation-dam-reservoir interaction system is performed for the following three cases which are described in Fig. 7.22.

- Case I: Empty reservoir (influence fluid-structure interaction not considered).
- Case II: Incompressible fluid (no frequency dependent properties)
- Case III: Compressible fluid with reflective reservoir boundaries.

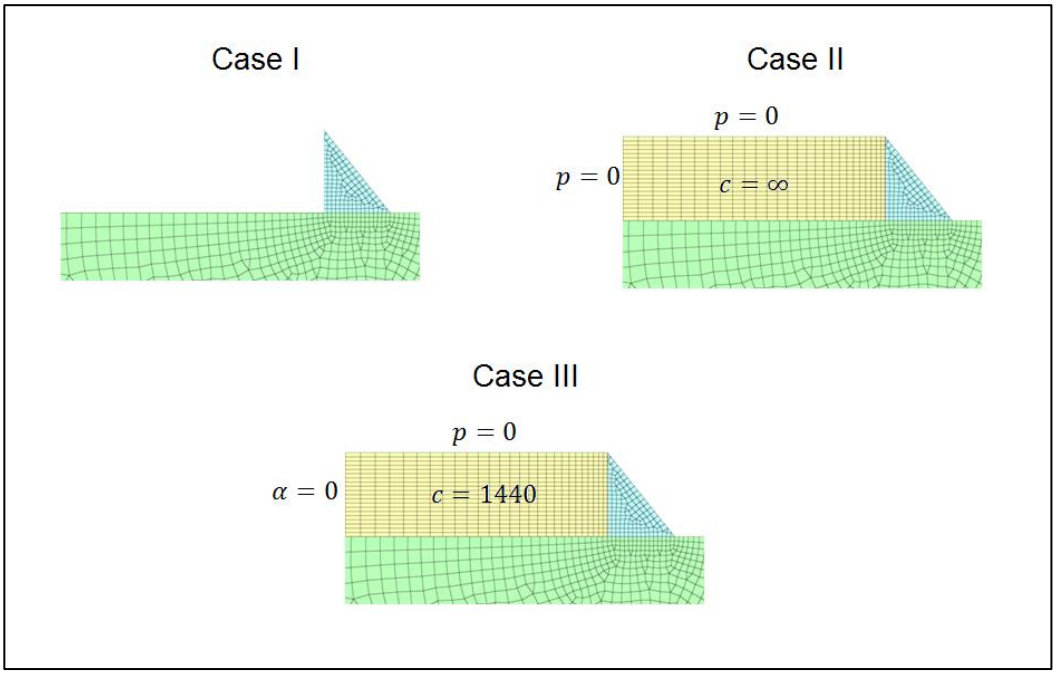


Figure 7.22: Transient linear and non-linear analysis cases.

The ground acceleration loading shown in Fig. 7.10 is applied to the three case studies and their responses are determined using the Newmark (average acceleration) and HFTD methods.

The type of cracking non-linearity, considered exclusively for the dam's concrete, is described in Section 7.1.5, whereas the values of the concrete non-linear parameters are presented in Table 7.2.

7.3.4.1. Case I: Empty reservoir

A comparison of the structural concrete cracking non-linear responses obtained in the dam's crest employing both Newmark and HFTD methods is presented in Figs. 7.23 to 7.25 for the relative displacement, velocity and acceleration responses in the X-direction, respectively.

The empty reservoir is an ideal case in which the influence of the dynamic interaction of the reservoir with the dam is neglected. For this reason, the magnitudes of the responses obtained are small and the system barely reaches the non-linear behavior. Actually, most of the behavior shown in Figs. 7.23 to 7.25 corresponds to a linear elastic behavior.

The HFTD and Newmark results coincide very well until the time 2.5 s. After this time step, the slight cracking of concrete begins and the difference between both responses starts to be noticeable. The reason for this difference other is that, in spite of being small in both methods, the extension and magnitude of concrete cracking in Newmark is greater than in HFTD.

Fig. 7.26 shows the cracking pattern of HFTD and Newmark methods for the last time step of the simulation equal to 4.49 s. It can be noticed that as expected the small cracking zone is located for both methods in the foot of the dam in the side of the reservoir. However, Newmark solution develops a more extended cracking zone for which the cracking strains are also higher in comparison with the

HFTD solution. An explanation for the inherent different non-linear performance of both methods is developed in the full reservoir with incompressible fluid case (Section 7.3.4.2).

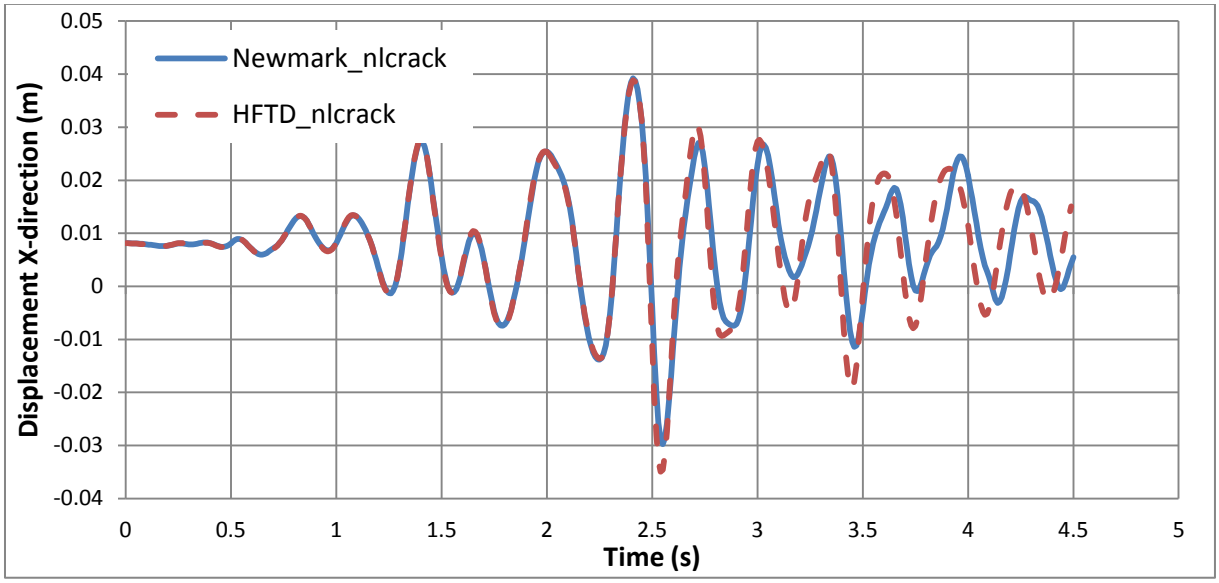


Figure 7.23: Empty reservoir concrete cracking non-linear response. Dam's crest displacement X-direction.

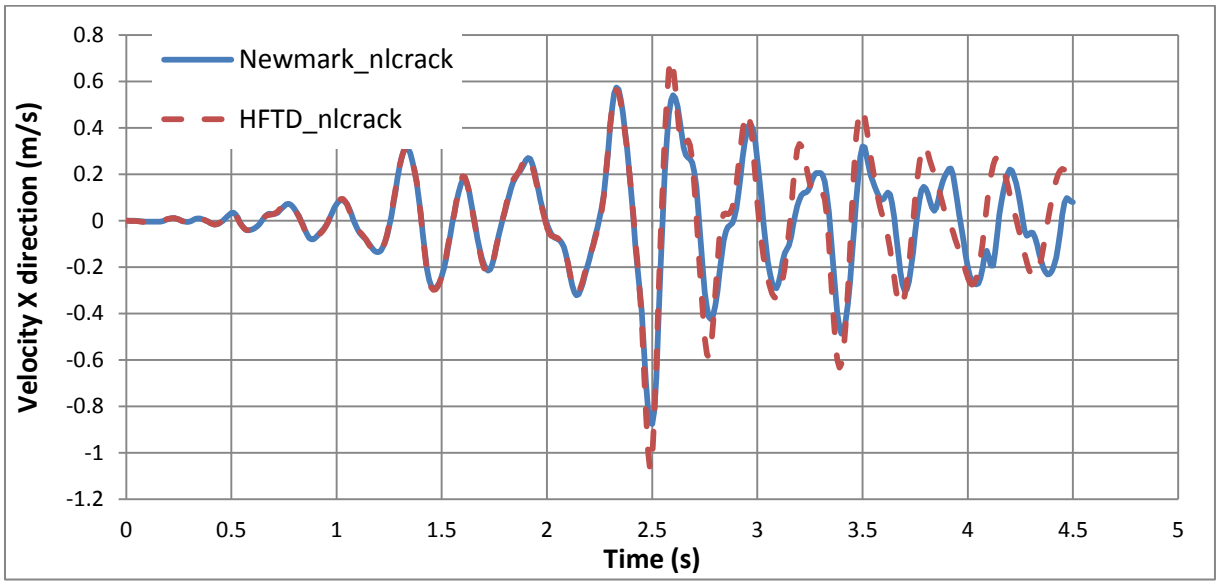


Figure 7.24: Empty reservoir concrete cracking non-linear response. Dam's crest velocity X-direction.

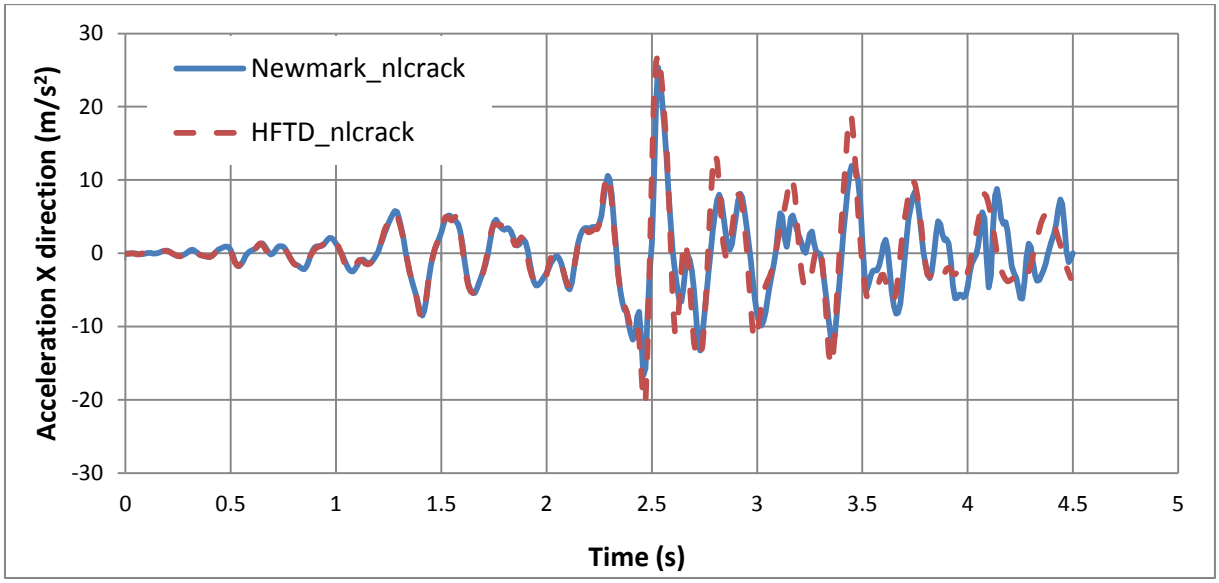


Figure 7.25: Empty reservoir concrete cracking non-linear response. Dam's crest acceleration X-direction.

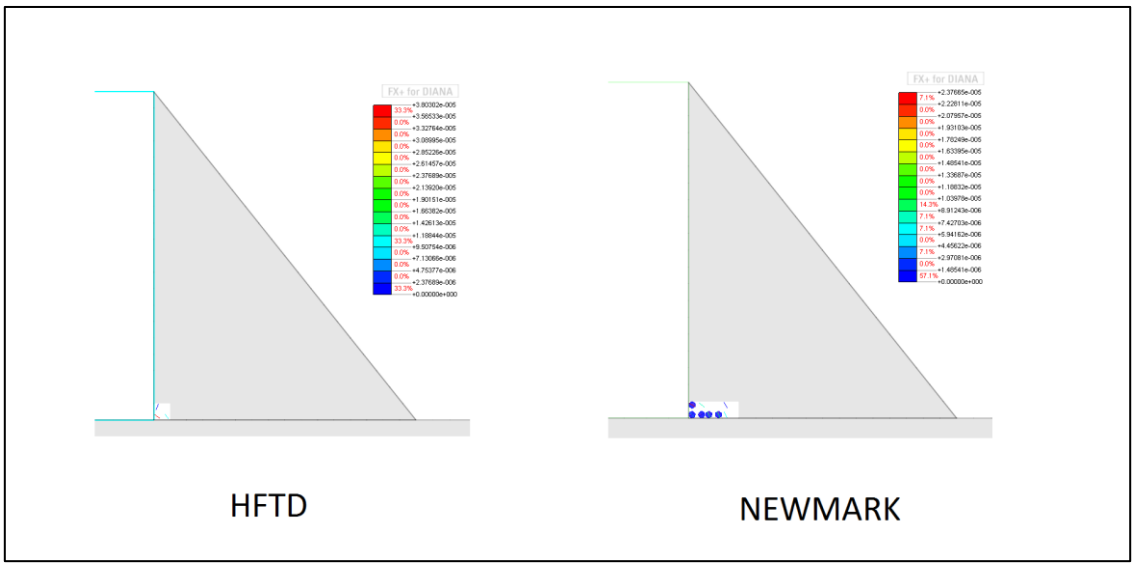


Figure 7.26: Empty reservoir concrete cracking non-linear response. Cracking pattern at time 4.49 s.

7.3.4.2. Case II: Full reservoir with incompressible fluid

When the reservoir interaction assuming incompressible fluid is considered, the responses of the dam become greater[13]. First a comparison of the structural linear elastic responses obtained in the dam's

[13] The influence of the reservoir interaction in the response of the dam was analytically presented for the one-dimensional models of Chapter 6.

crest employing both Newmark and HFTD methods is presented in Figs. 7.27 to 7.29 for the relative displacement, velocity and acceleration in the X-direction, respectively.

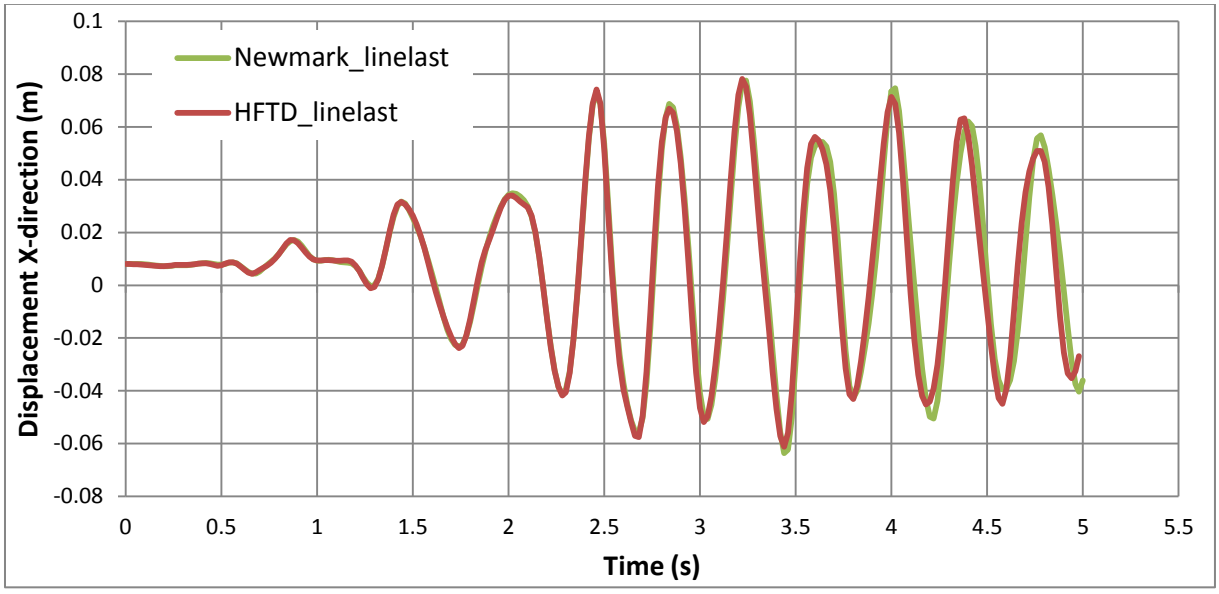


Figure 7.27: Incompressible fluid linear elastic response. Dam's crest displacement X-direction.

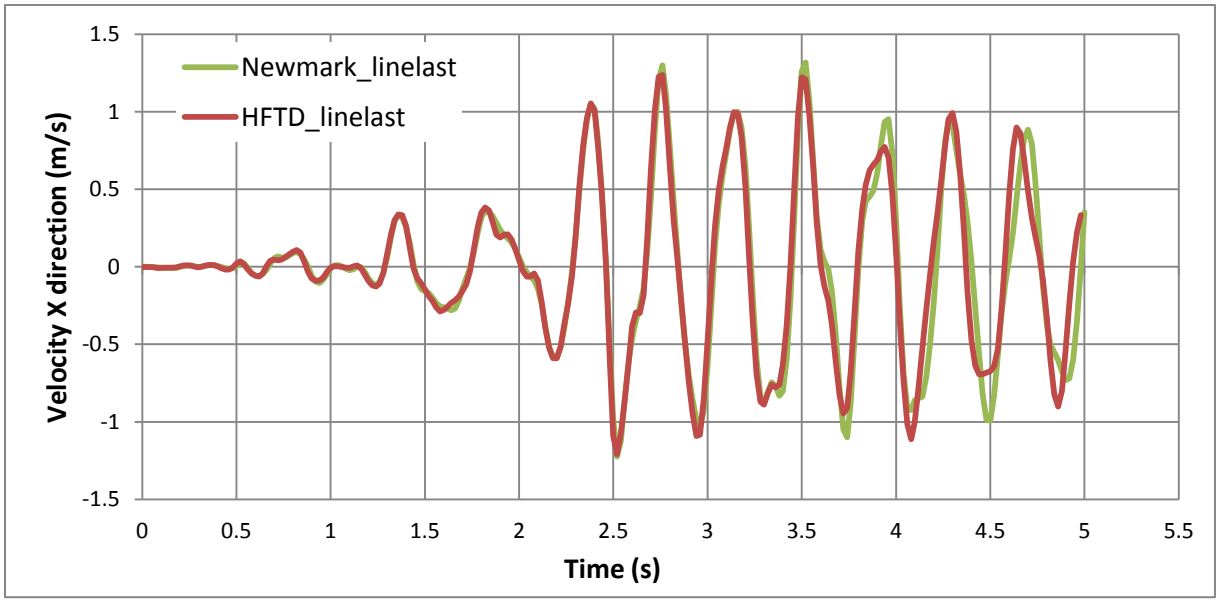


Figure 7.28: Incompressible fluid linear elastic response. Dam's crest velocity X-direction.

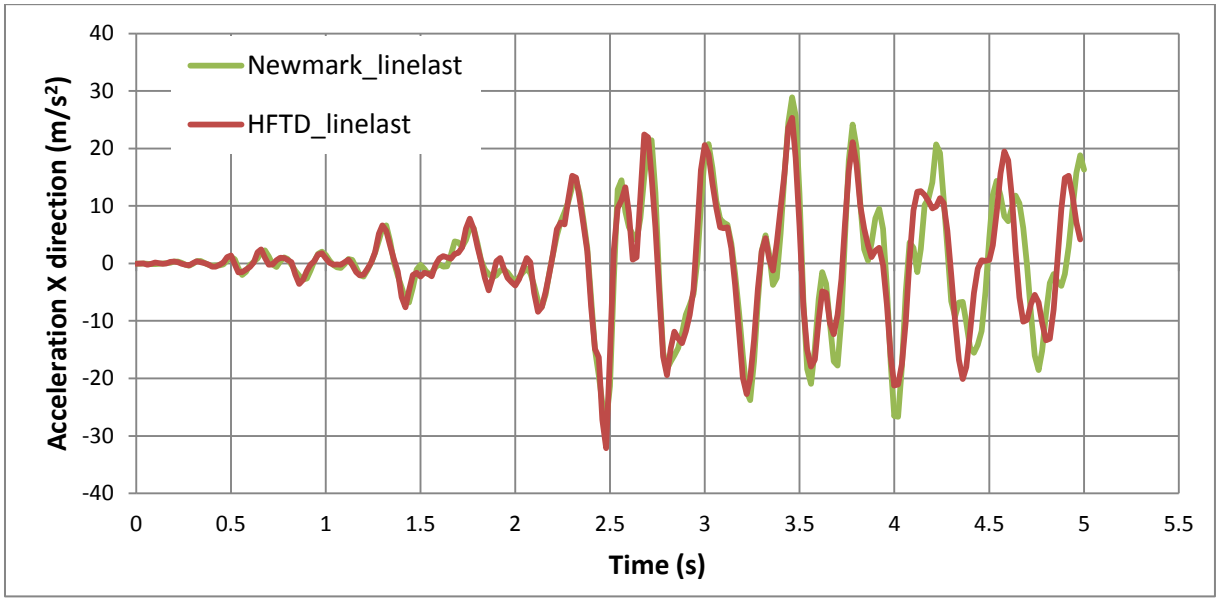


Figure 7.29: Incompressible fluid linear elastic response. Dam's crest acceleration X-direction.

Figs. 7.27 to 7.29 show a very good agreement in the linear elastic solutions obtained with both methods. However, small differences are appreciated in the peak values since time 3.5 s thereafter, especially for the acceleration response.

As it is explained in the following lines for the non-linear response, the reason for these differences is related with the choice of implementing the HFTD method in combination with the mode superposition method. Not all the modes of vibration but only fifteen are included in the HFTD analysis, and therefore part of the solution contributing with the peaks response is excluded.

Furthermore, the loading input signal shown in Fig. 7.10 has a maximum sampling frequency equal to 25 Hz. This means that any excitation frequency contained in the loading, or any modal vibration frequency of the system, which is greater to 25 Hz is not included in the HFTD response.

Before showing the results obtained for the concrete cracking non-linear behavior of the dam, the hydrodynamic pressures generated by the linear elastic dam-reservoir interaction is shown for different time steps in Fig. 7.30. It is clear that the hydrodynamic pressures can assume several types of height distribution depending of the acceleration in the interface nodes for each time step of the analysis.

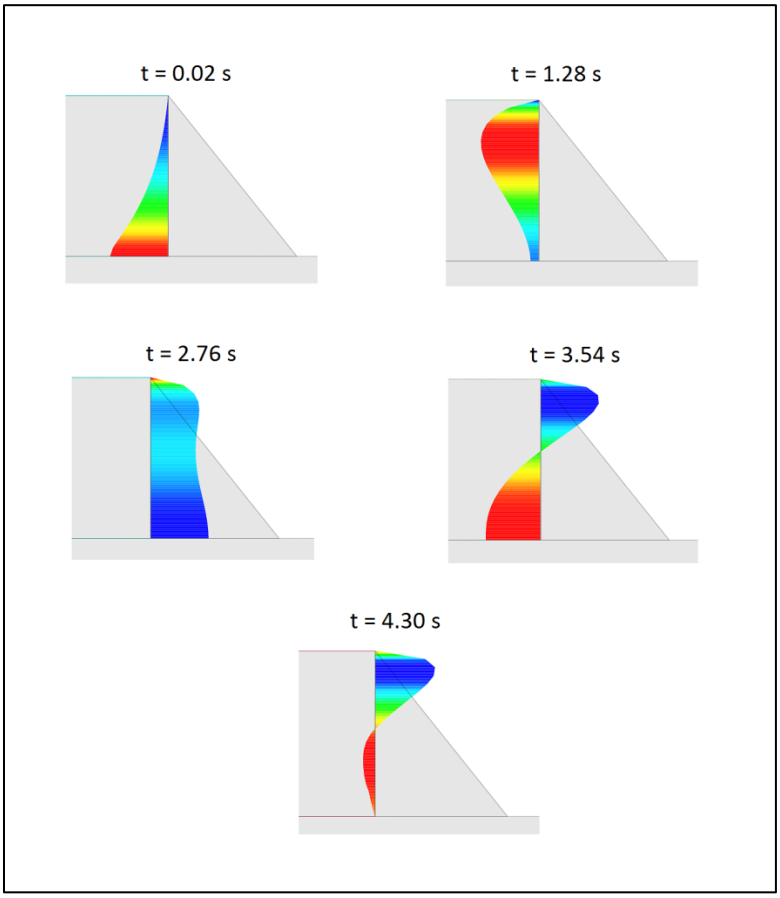


Figure 7.30: Incompressible fluid linear elastic response. Hydrodynamic pressures at different time steps.

Now, a comparison of the structural concrete cracking non-linear responses obtained in the dam's crest, employing both Newmark and HFTD methods, is presented in Figs. 7.31 to 7.33 for the relative displacement, velocity and acceleration in the X-direction, respectively.

As mentioned before, the consideration of a full reservoir with incompressible fluid in the seismic analysis produces a significant increment of the responses amplitudes. For example, Fig. 7.31 shows that the maximum displacement obtained in this case study is close to between 0.06 m and 0.08 m.

On the other hand, Fig. 7.23 shows that when the reservoir interaction with the dam is neglected, the maximum non-linear displacement is less than 0.04 m. This means that the level of cracking is also intensified when the incompressible fluid is taken into account, and as a consequence the difference between the cracking non-linear behaviors predicted by HFTD and Newmark methods is more obvious than in the empty reservoir case.

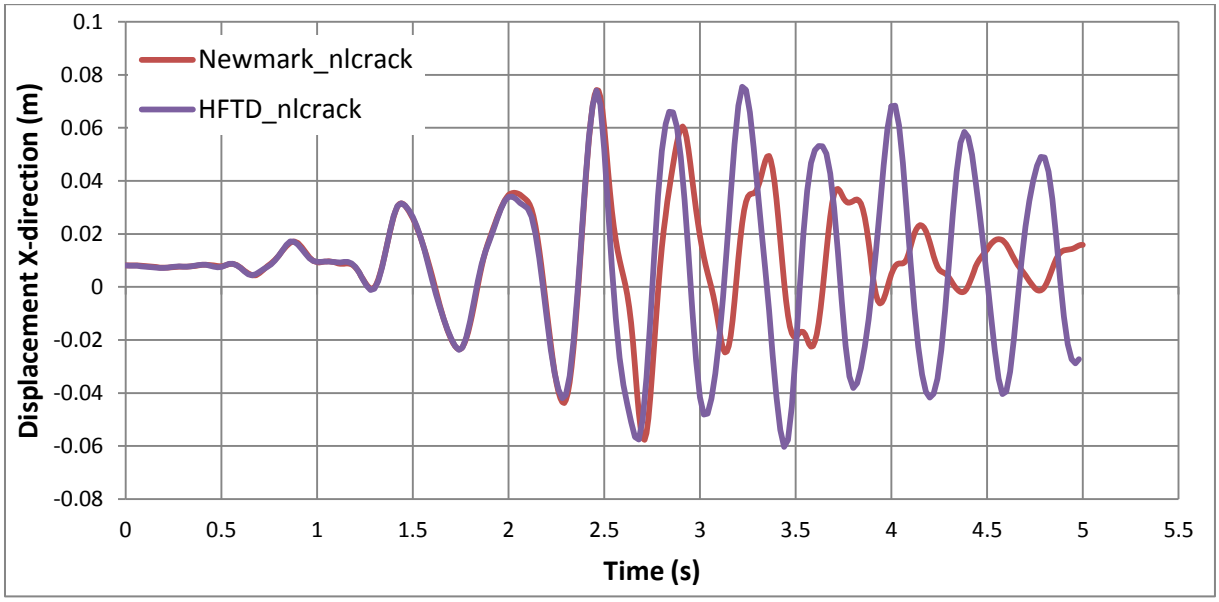


Figure 7.31: Incompressible fluid concrete cracking non-linear response. Dam's crest displacement X-direction.

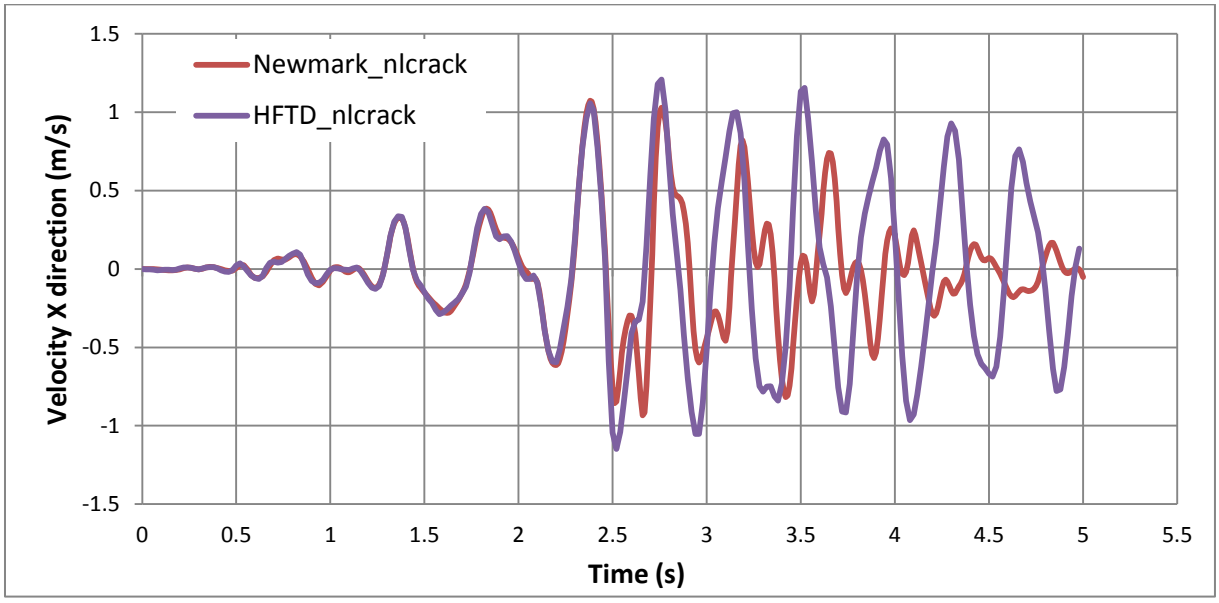


Figure 7.32: Incompressible fluid concrete cracking non-linear response. Dam's crest velocity X-direction.

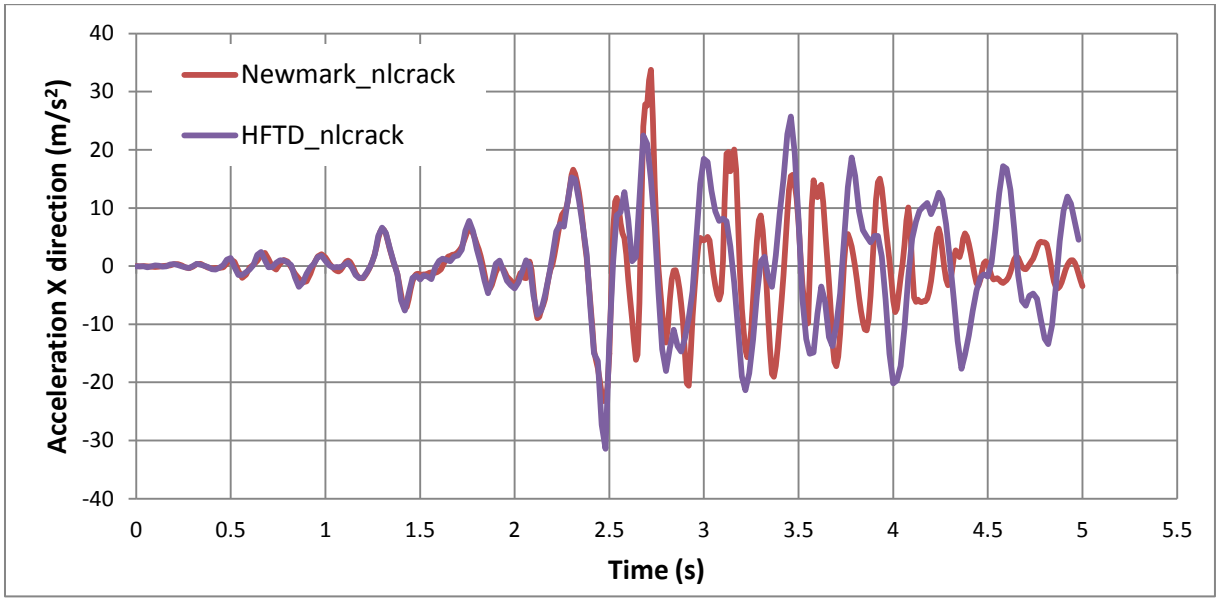


Figure 7.33: Incompressible fluid concrete cracking non-linear response. Dam's crest acceleration X-direction.

Figs. 7.31 to 7.33 indicate that the non-linear behavior described by the HFTD method is moderate, resembling what is a known as a mild type of non-linearity. For this reason, the shape of the time-history plots is similar to the ones of the linear elastic behavior (Figs. 7.27 to 7.29), but with slightly smaller amplitudes.

On the other hand, the non-linear behavior described by the Newmark method is more severe, diminishing considerably the response amplitudes and altering the periods of oscillations. The substantial difference between the non-linear cracking behaviors predicted by each method at different time steps is illustrated in Fig. 7.34.

It can be noticed that in the time step 1.42 s, where the first crack appears, there is a good agreement between both methods about which is the zone with higher tensile stresses. However, during the following time steps, and especially around 2.5 s, the Newmark method develops a very pronounced crack in the base due to the concentration of high tensile strains nearly that zone.

In spite of calculating a similar displacement field as Newmark does, HFTD method does not predict such a tensile strain concentration in the base of the dam. Therefore, the HFTD tensile strain field is uniformly distributed through the entire dam's body and has considerably smaller magnitudes.

This observation about the displacement and strain fields obtained with both methods is graphically demonstrated in Fig. 7.35 for the time step 2.40 s, when the big cracks shown in Fig. 7.34 for the time step 2.61 s have not appeared yet.

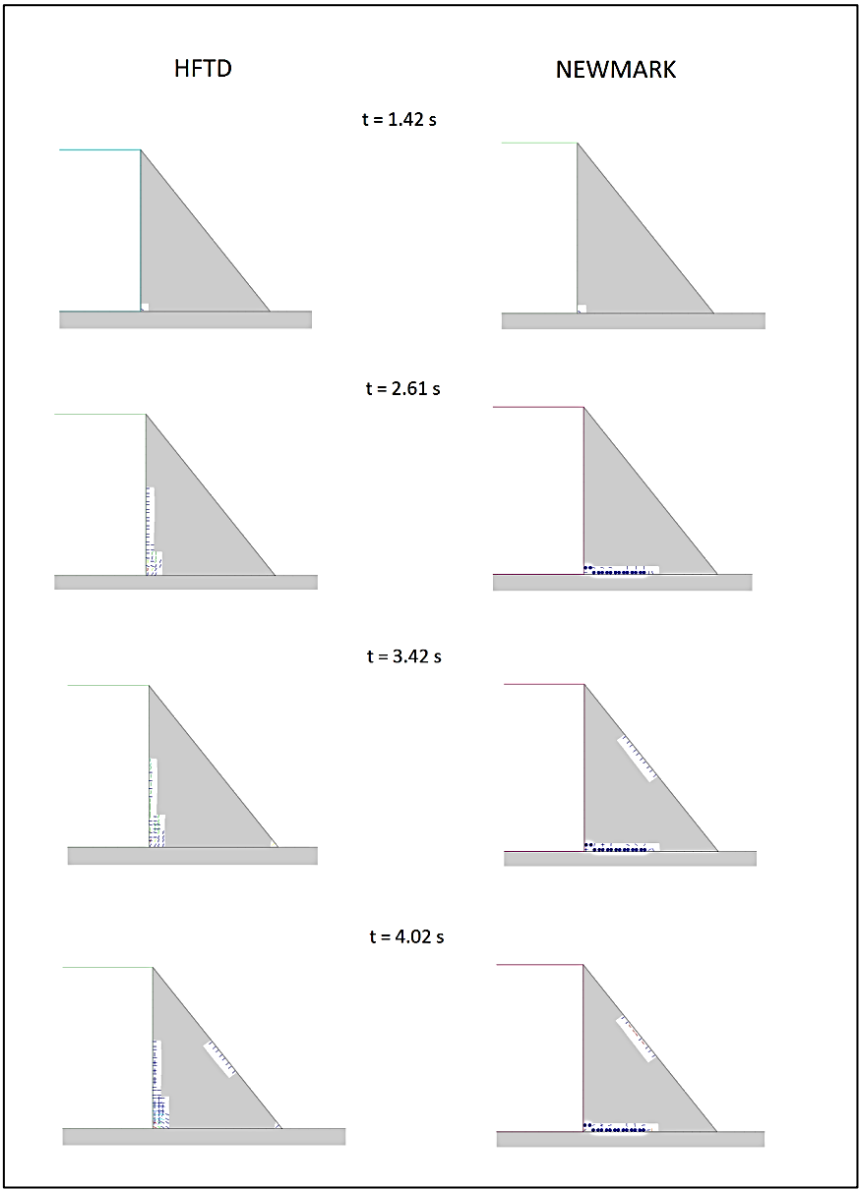


Figure 7.34: Incompressible fluid reservoir non-linear response. Dam's concrete cracking pattern at different time steps.

The large crack in the base of the dam calculated by the Newmark method (Fig. 7.34) is the reason why the responses shown in Figs. 7.31 to 7.33 are highly non-linear for this method. What is more, due to the position and magnitude of the concrete cracking, the dam acts like an isolated system in which the ground acceleration loading is not fully transmitted to the structure. For this reason, since the time step 2.5 s the magnitudes of the responses in the crest of the dam calculated with the Newmark method diminish substantially.

On the other hand, due to the fact that in the HFTD method the non-linear response of the dam is calculated based on the superposition of the linear elastic modes of vibration, the higher tensile strains are predicted to be located in the vertical face of the dam interacting with the reservoir and not in the base. In this case, the cracking is more distributed and therefore less severe. Furthermore,

the location of the higher tensile zones seems to be ruled by the first and the third mode shapes of vibration in the horizontal direction (See Fig. 7.14).

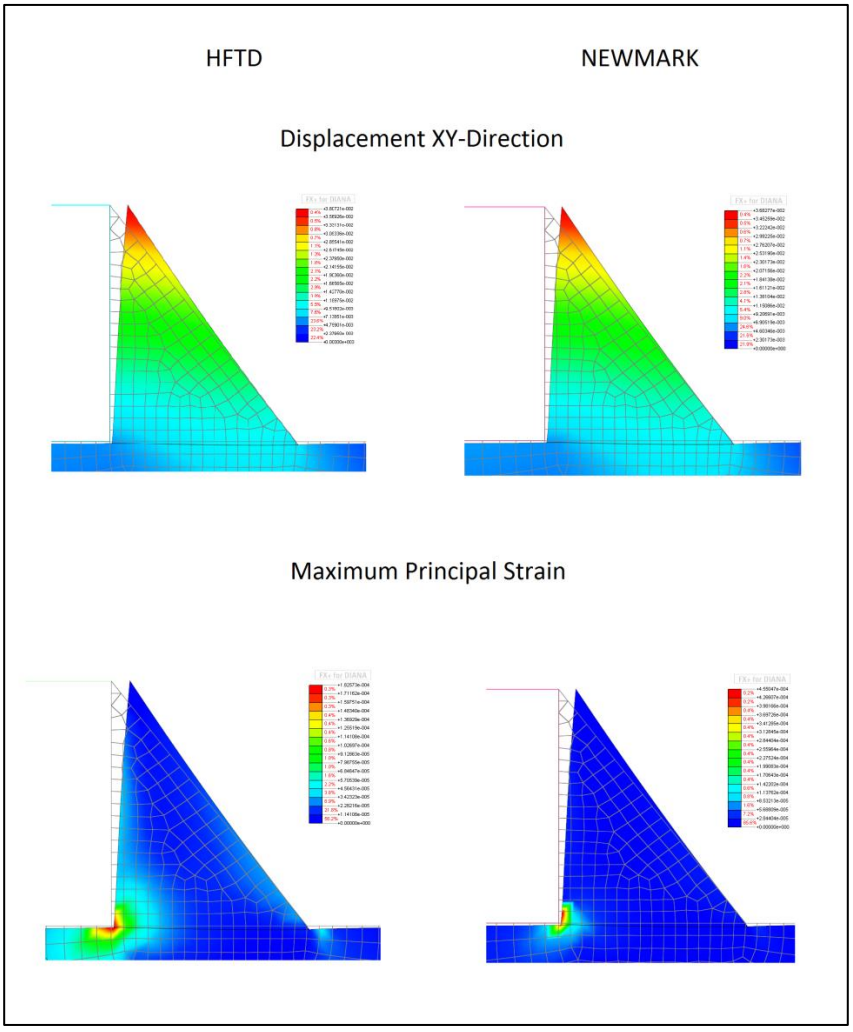


Figure 7.35: Incompressible fluid reservoir non-linear response. Displacement and principal strain fields at time step 2.40 s.

7.3.4.3. Case III: Full reservoir with compressible fluid

The frequency dependent dam-reservoir interaction system with compressible fluid and infinite extent radiation boundary condition (Case III in Fig. 7.22) is directly solved in DIANA only by the Direct Method (linear frequency domain solution) and the HFTD method (non-linear transient solution). Due to the frequency dependence of the system, Newmark method is not applicable for this case.

Therefore, non-linear concrete cracking HFTD solution is presented in Figs. 7.36 to 7.38 for the dam's crest relative displacement, velocity and acceleration, respectively, in the X-direction. A general observation of the time history graphics allows concluding that for this particular model the inclusion of compressibility in combination with the radiation boundary of infinite extent does not change substantially the peak values of the non-linear responses of the system. This analyzed in detail in the next section.

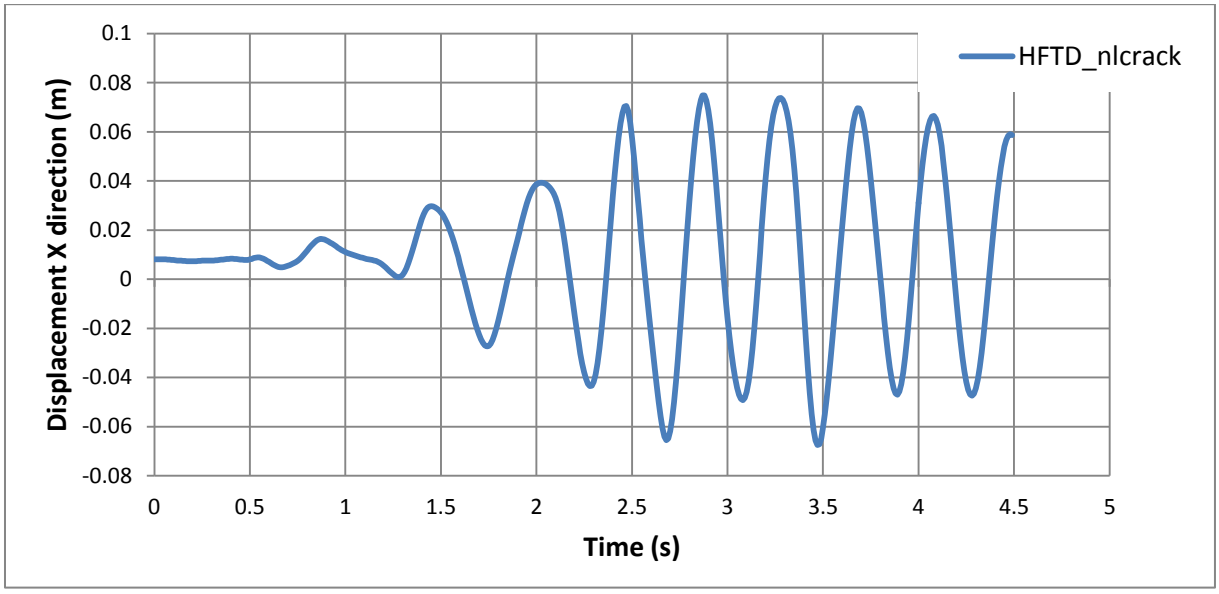


Figure 7.36: Compressible fluid with radiation boundary. HFTD non-linear solution for the dam's crest displacement in X-direction.

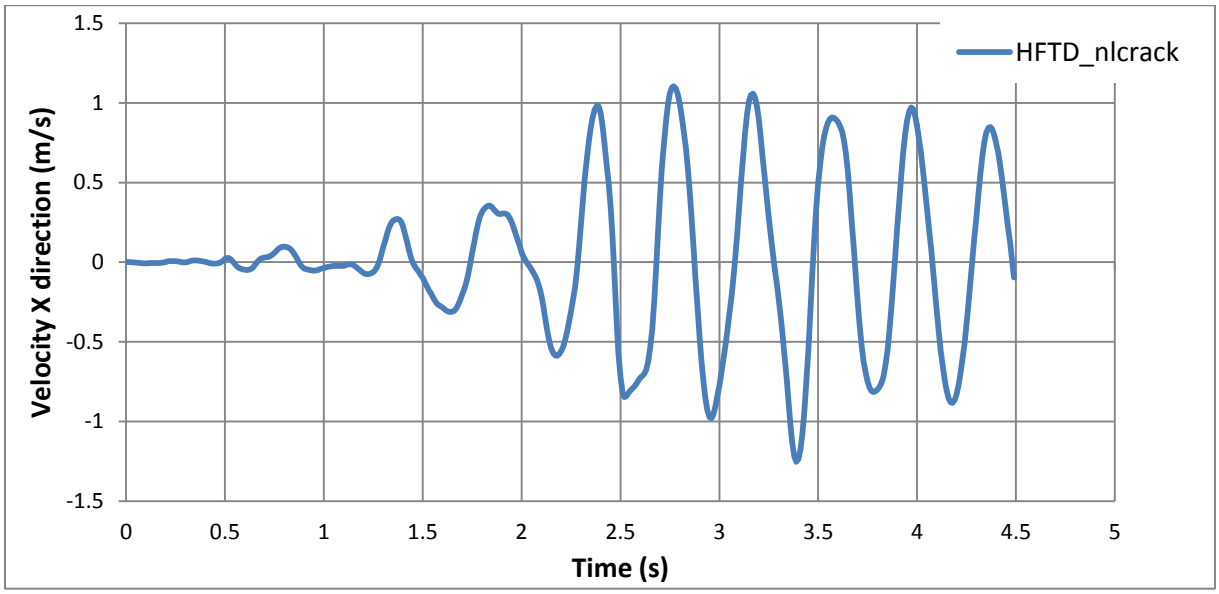


Figure 7.37: Compressible fluid with radiation boundary. HFTD non-linear solution for the dam's crest velocity in X-direction.

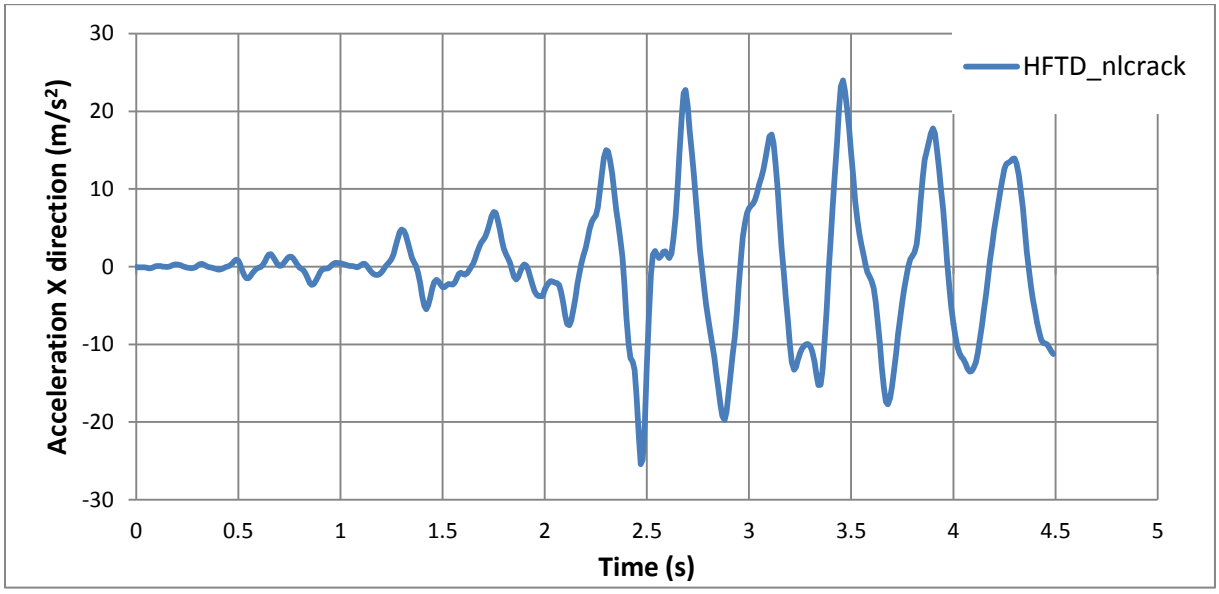


Figure 7.38: Compressible fluid with radiation boundary. HFTD non-linear solution for the dam's crest acceleration in X-direction.

7.3.4.4. HFTD results analysis: Influence of the dam-reservoir interaction in the non-linear structural response

The influence of dam-reservoir interaction and fluid compressibility in the dam's concrete cracking non-linear response is analyzed based on the HFTD solutions of the three cases previously studied (Fig. 7.22). The dam's crest relative displacement, velocity and acceleration in the X-direction are presented in Figs. 7.39 to 7.41, respectively.

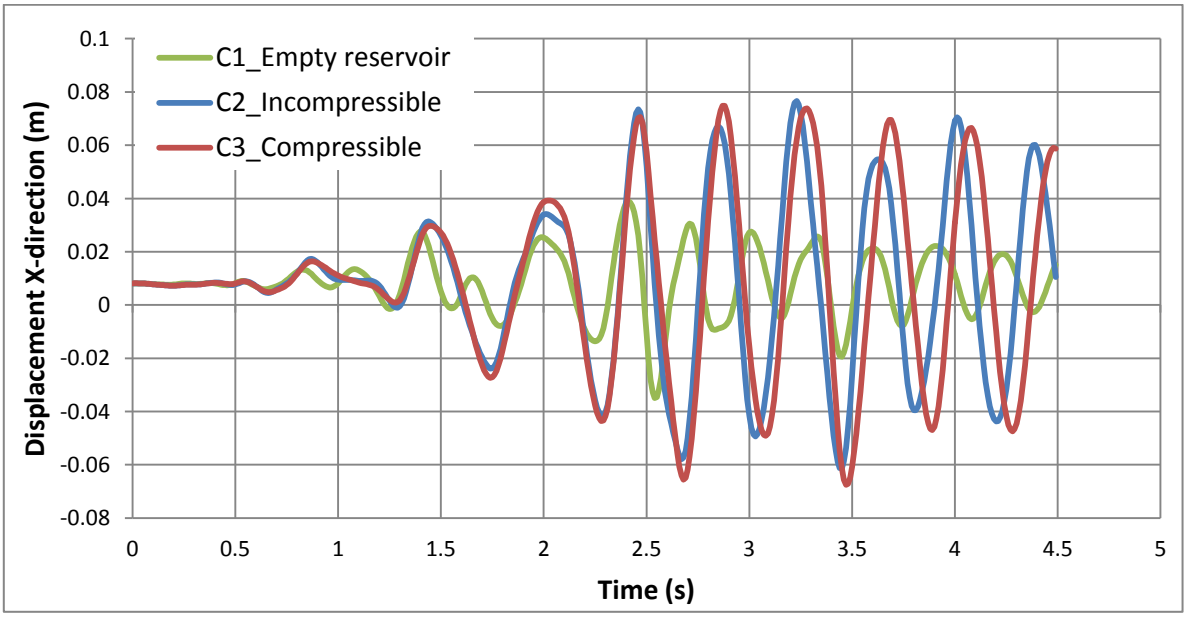


Figure 7.39: Influence of the dam-reservoir interaction and fluid compressibility in the concrete cracking non-linear response. Dam's crest displacement X-direction.

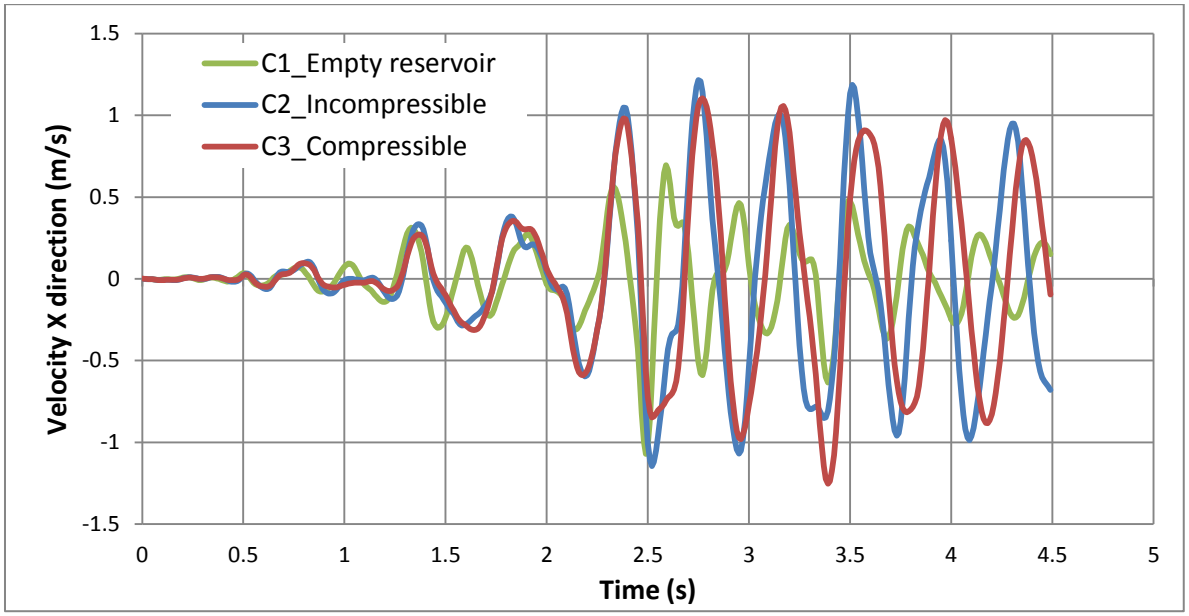


Figure 7.40: Influence of the dam-reservoir interaction and fluid compressibility in the concrete cracking non-linear response. Dam's crest velocity X-direction.

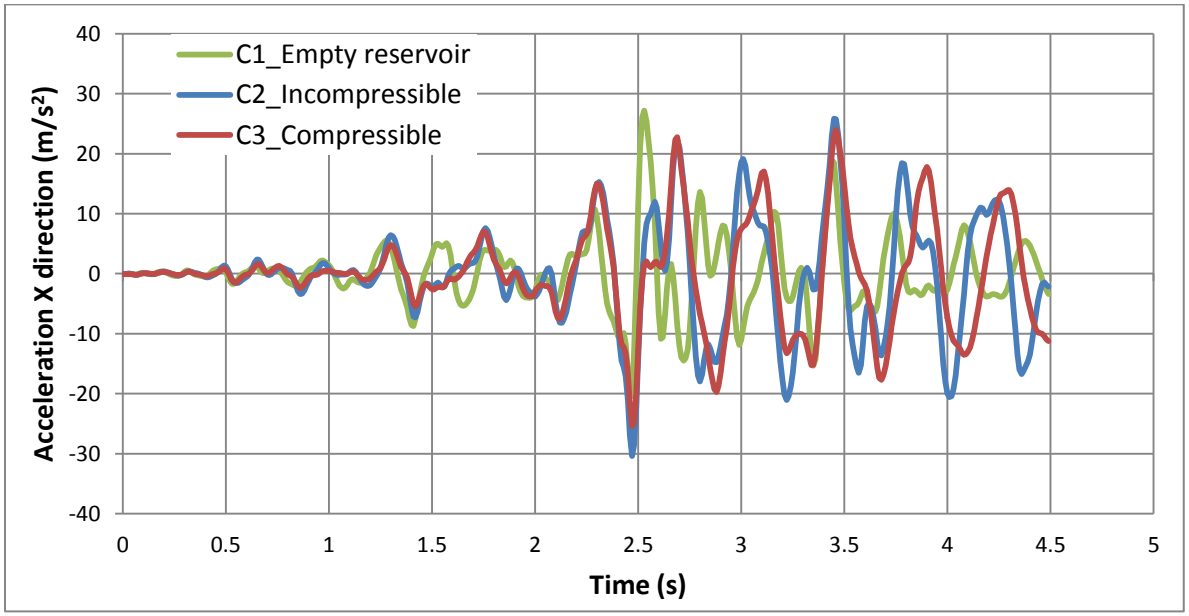


Figure 7.41: Influence of the dam-reservoir interaction and fluid compressibility in the concrete cracking non-linear response. Dam's crest acceleration X-direction.

The consideration of the reservoir interaction in the seismic analysis is demonstrated to be important. Fig. 7.39 shows that the maximum displacement of the full reservoir analysis (Cases C2 and C3) is twice the maximum displacement of the empty reservoir analysis (case C1). Additionally, the vibration period of the full reservoir cases is longer than the oscillation period of the empty reservoir case.

On the other hand, for this particular case compressibility does not play a fundamental role. Even though higher displacement amplitudes are indeed obtained for the compressible fluid case, the difference with the amplitudes of the incompressible fluid case is very small. Fig. 7.39 also shows that the oscillation period of the compressible fluid case becomes progressively longer than the oscillation period of the incompressible fluid case.

Anyhow, the time history acceleration plot shown in Fig. 7.41 demonstrates that the consideration of the reservoir interaction and the fluid compressibility in the seismic analysis of dams produces very different non-linear dynamic responses. This is confirmed by the cracking pattern at different time steps shown in Fig.7.42 for each of the three case studies.

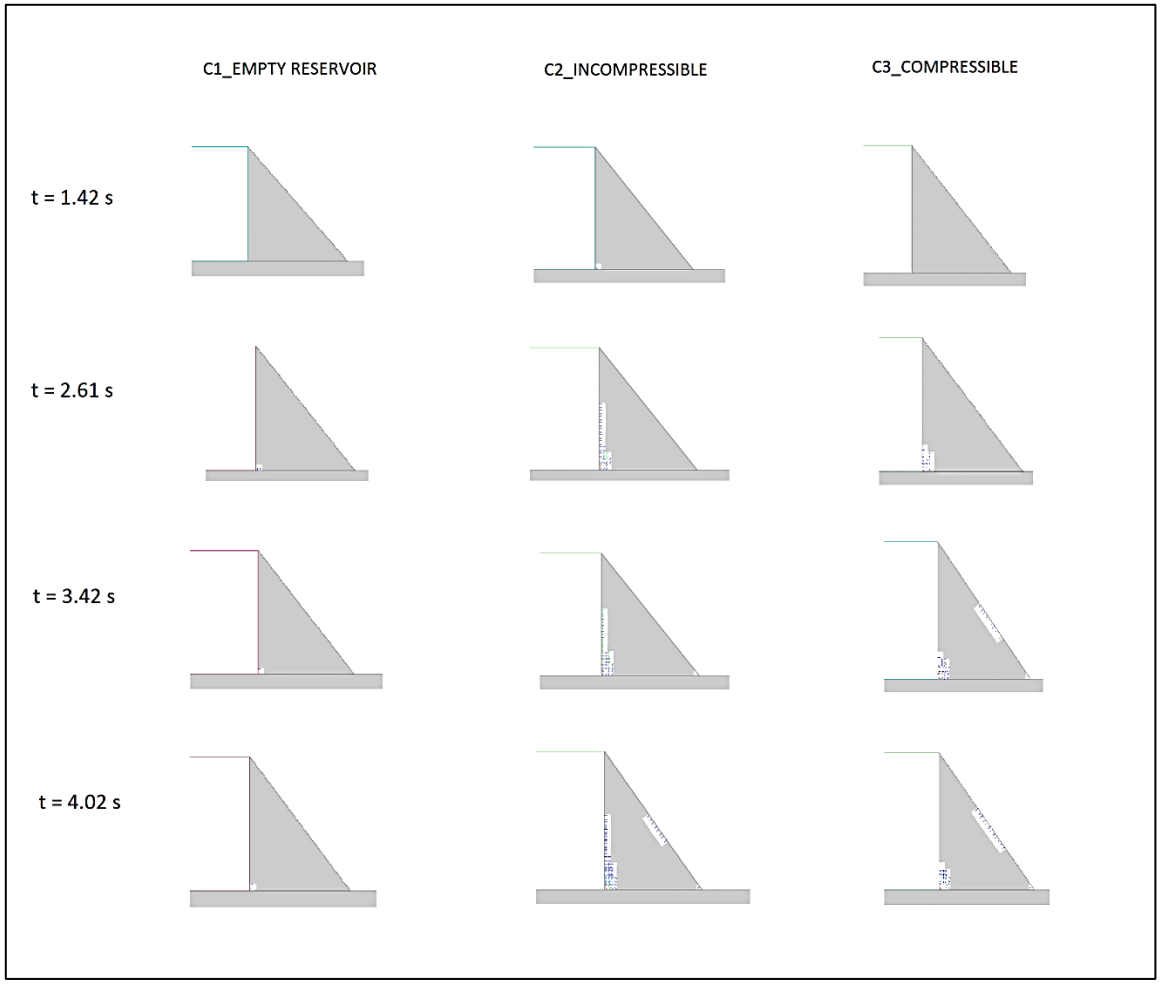


Figure 7.42: Influence of the dam-reservoir interaction and fluid compressibility in the non-linear response. Dam's concrete cracking pattern at different time steps.

Fig. 7.42 shows that when the reservoir interaction is not taken into account, the dam barely cracks; there is only a very small cracking zone located in the dam foot. On the other hand, the first crack appears at time step 1.42 s for the full reservoir with incompressible fluid case study. This is the case that suffers the higher levels of cracking in the zone of the vertical dam-reservoir interface. Finally, the higher levels of cracking in the inclined back face of the dam correspond to the compressible fluid case study.

7. Implementation of the HFTD Method in DIANA for the Seismic Analysis of Foundation-Dam-Reservoir Interaction Problems

8. CONCLUSIONS AND RECOMMENDATIONS

8.1. Conclusions

Through the different parts of this thesis it has been demonstrated that the HFTD method is founded on a strong theoretical basis which fundamentals is the frequency domain analysis, particularly, the Fourier analysis.

The results of the SDOF numerical examples performed in Chapter 5 validate the high accuracy of the HFTD method for linear and even for highly non-linear systems, including simultaneous hysteretic stiffness and damping non-linearity, subject to several types of loading.

For linear elastic SDOF systems, the HFTD method results are almost exact for all types of loading, with exception of sudden loading functions which starts at an initial value different than zero (f.e.: cosine loading function). In these latter cases the results obtained are very accurate but not exact.

For non-linear SDOF systems, if the size of the time step is appropriate, the HFTD method also shows a great precision, similar to the accuracy of the Newmark method. In order to achieve very accurate results, both methods require that the time step size is in the order of the natural period of vibration of the system divided over one hundred.

However, in addition to the size of the time step, the research work has shown that the accuracy of the HFTD method depends on two numerical implementation issues: 1) Time segmentation approach; and 2) Extended Fourier Transform period.

The division of the response time span of interest in a determined number of segments is a procedure implemented in the HFTD method to achieve convergence in non-linear problems. This implementation issue, denominated time segmentation approach, has proved to be very useful especially when the system is highly non-linear, the time span of interest is long, or the system includes frequency dependent properties. Besides, the division of the time span of interest in a higher number of time segments improves the accuracy of the results pertaining to the initial time steps.

The extension of the Fourier Transform period by adding a band of zeros, called quiet zone, to the loading signal is a fundamental implementation issue of the HFTD method, which makes possible to satisfy the initial conditions and determine the transient response of the system, instead of the steady-state response. The properly definition of the HFTD method must perform all the Fourier Transforms using this extended period which resembles the infinite period of the analytical Fourier Integral.

The HFTD method for MDOF systems was implemented in DIANA in combination with the Mode Superposition method. This implementation option, explained in detail in Chapter 4, is theoretically consistent due to the fact that the HFTD method solves a pseudo-linear equation in the frequency domain.

The programming of the HFTD method for SDOF systems in MATLAB was essential for the understanding, implementation and validation of the HFTD implementation for general applications in DIANA.

The application of the HFTD method is mainly interesting in MDOF systems with frequency dependent properties, like the dam-reservoir interaction system subject to a seismic ground acceleration loading. In Chapter 7 a case study of this system was tested in DIANA using the HFTD and Newmark methods.

For the linear elastic analysis of the dam-reservoir interaction system, the results obtained with the HFTD and the Newmark methods show a very good agreement, in terms of accuracy and performance. This is not the case for the concrete cracking non-linear behavior of the dam, which HFTD and Newmark results show considerable differences.

The main reason for the differences in the results when dealing with the concrete cracking non-linearity of this particular case study is the great influence that the Mode Superposition method has in the description of the non-linear behavior of the structure determined with the HFTD method. Due to the combination these two methods, the HFTD non-linear response of the structure is calculated based on the superposition of the linear elastic modes of vibration, and therefore, the distribution of higher stresses and cracking zones is ruled by their associated mode shapes.

A totally different situation is defined by the Newmark method, which solution is not based on the Mode Superposition method, but on the solution of the complete system. Logically, the consequences of this inherent difference between both methods cannot be evaluated with the SDOF systems studied in Chapter 5, which total response is exactly the same as the modal response.

Therefore, one important conclusion of the research work is that the best performance of the HFTD method, in combination with the Mode Superposition method, is obtained when it is applied to MDOF systems with frequency dependent properties and linear or mild non-linear behavior.

Mild non-linear behavior is understood as a type of slight non-linearity which is not concentrated in particular zones but spread in the whole structure. It is assumed that mild non-linearity does not modify drastically the structural configuration of the linear elastic system, and consequently, the superposition of the modal responses system can still provide a good estimation of the non-linear response.

In the particular application of HFTD in dam-reservoir interaction systems, the frequency dependent properties are composed by the fluid compressibility and the radiation boundary conditions of the reservoir, like for example, the bottom absorption and the infinite extent.

From the thesis it can be concluded that these frequency dependent properties actually have an impact in the response of the dam-reservoir interaction system. However, it was not possible to outline any definitive conclusion about the relative importance of their inclusion in the system's response.

Nevertheless, it is clear that the radiation boundary conditions, especially the bottom absorption, act like additional damping in the system, and therefore, they reduce the magnitude of the system's response.

The effect of fluid compressibility, on the other hand, is less predictable. For linear elastic frequency domain analysis, the assumption of a compressible fluid clearly increases the amplitudes of the response. However, for this particular case study, the time domain response obtained with the HFTD method shows that, even though fluid compressibility indeed produces higher peaks, this increment is small and constitutes a minor change in the time-history response of the system.

One important advantage of the HFTD method is its numerical stability. Accurate solutions are obtained without convergence problems mainly because the solution of the pseudo-linear equation in

the frequency domain does not require the calculation of a tangent stiffness matrix for each iteration or each time step, as it is the case of the Newton-Raphson method. On the contrary, the HFTD method uses the same linear elastic matrices through all the process, changing only the values of the excitation frequencies, the excitation forces and the pseudo force vector. This allows the HFTD method to solve large 3D models achieving convergence in an efficient way.

As a final conclusion, the HFTD method shows promising perspectives for its application in the academic and professional practice, especially for the seismic analysis of fluid-structure and soil-structure interaction systems, or any other area for which the research of dynamic non-linear behavior of frequency dependent systems becomes every time more important.

8.2. Recommendations

It is recommended that the current implementation of the HFTD method in DIANA be used in the investigation of different types of frequency dependent systems.

Three-dimensional models of dam-reservoir interaction systems could be elaborated not only to keep evaluating the performance of the HFTD method, but to study more deeply the importance of fluid compressibility and radiation boundary conditions in the time response of this kind of systems.

It would be important to test the HFTD method using different types of non-linear behavior (concrete crushing, steel yielding, base sliding), and evaluate the results obtained in the context of the mild-non-linearity concept and the influence that the Mode Superposition method has on the HFTD method.

The effect that different types of structural geometries have in the HFTD non-linear responses constitutes an interest research field which has not been studied yet. For some types of structural geometries, like for example slender structures, the principal mode shapes describe more precisely the actual deformation and stress distribution of the system, even after the incursion in the non-linear behavior stage.

Finally, for future research it is recommended to study the feasibility, in terms of computational efficiency, of implementing the HFTD method without including the Mode Superposition method. This would allow overcoming the restriction to mild non-linear behavior, but assuming the cost of increasing drastically the size of the system of equations to be solved.

REFERENCES

1. Aprile, A., Benedetti, A., and Trombetti, T. (1994). *On non-linear dynamic analysis in the frequency domain: algorithms and applications*. Earthquake Engineering and Structural Dynamics, Vol.23, pp. 363-388.
2. Clough, R.W., and Penzien, J. (2003). *Structural dynamics*. Computers and Structures, California, Berkeley.
3. Craig, R.R., and Kurdila, A.J. (2006). *Fundamentals of structural dynamics.* John Wiley and Sons, New Jersey.
4. Chavez, J.W., and Fenves, G.L. (1993). *Earthquake analysis and response of concrete gravity dams including base sliding*. Report UBC/EERC-93/07, Earthquake Eng. Research Center, Univ. of California, Berkeley.
5. Chavez, J.W., and Fenves, G.L. (1994). *EAG-SLIDE: A computer program for the earthquake analysis of concrete gravity damsincluding base sliding*. Report UBC/SEMM-94/02, Structural Engineering Mechanics and Materials, Department of Civil Engineering, Univ. of California, Berkeley.
6. Chopra, A.K. (2007). *Dynamics of structures: theory and applications to earthquake engineering.* Third edition. Pearson Education Inc., New Jersey.
7. Darbre, G.R., and Wolf, J.P. (1988). *Criterion of stability and implementation issues of hybrid frequency-time domain procedure for nonlinear dynamic analysis*. Earthquake Engineering and Structural Dynamics, Vol.16, pp. 569-581.
8. Darbre, G.R. (1990). *Seismic analysis of a non-linearly base-isolated soil-structure interacting reactor building by way of the hybrid frequency-time domain procedure*. Earthq. Eng. and Struct. Dyn., Vol.19, pp. 725-738.
9. Darbre, G.R. (1996). *Nonlinear dam-reservoir interaction analysis*. Proc. Eleven World Conf. on Earthquake Engineering, Acapulco, Paper No. 760.
10. Fenves, G.L., and Chavez, J.W. (1990). *Hybrid frequency-time domain analysis of nonlinear fluid-structure systems*. Proc. Fourth U.S. Nat. Conf. on Earthquake Engineering, California, pp. 97-104.
11. Fenves, G.L., and Chopra, A.K. (1983). *Effects of reservoir bottom absorption on earthquake response of concrete gravity dams*. Earthquake Engineering and Structural Dynamics, Vol.11, pp. 809-829.
12. Fenves, G.L., and Chopra, A.K. (1984). *Earthquake analysis and response of concrete gravity dams*. Report UBC/EERC-84/10, Earthquake Eng. Research Center, Univ. of California, Berkeley.
13. Humar, J.L. (2002). *Dynamics of structures.* Second edition. Balkema Publishers, Lisse.
14. Mansur, W.J. et al. (2000). *Time-segmented frequency-domain analysis for non-linear multi-degree-of-freedom structural systems*. Journal of Sound and Vibration, Vol. 237(3), 457-475.
15. Metrikine, A.V., and Vrouwenvelder, A.C. (2006). *Dynamics of structures – Part 2: Wave dynamics.* Lecture Notes of Course CIE4140, Section of Structural Mechanics, Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft.
16. Paz, M., and Leigh, W. (2004). *Structural dynamics: theory and computation*. Fifth edition. Springer Science+Bussiness media, New York.
17. Rizos, D.C., and Karabalis, D.L. (2000). *Fluid-soil-structure interaction.* International series on advances in earthquake engineering. WIT Press, UK.
18. The Mathworks, Inc. (2012). *MATLAB and Statistics Toolbox Release 2012b*, Natick, Massachusetts.
19. TNO DIANA BV (2011). *DIANA User's Manual – Release 9.4.4*, Delft.

20. Veletsos, A.S., and Ventura, C.E. (1985). *Dynamic analysis of structures by the DFT method.* J. Struct. Eng. ASCE, Vol.111, 2625-2642.
21. Wilson, E.L. (2002). *Three-dimensional static and dynamic analysis of structures*. Third Edition. Computers and Structures Inc., California.
22. Zienkiewicz, O.C., and Bettes, P. (1978). *Fluid-structure dynamic interaction and wave forces. An introduction to numerical treatment*. Int J Numer Meth Eng, Vol. 13, 1-16.
23. Zienkiewicz, O.C., Taylor, R.L., and Zhu, J.Z. (2005). *The finite element method – Its basis and fundamentals*. Sixth Edition. Elsevier Butterworth-Heinemann, London.

APPENDIX A: FLOWCHART OF THE HFTD METHOD IMPLEMENTATION IN DIANA

A.1. Main HFTD flowchart

START

Set up of fluid element matrices
$\mathbf{M}_F, \mathbf{C}_F, \mathbf{K}_F$

Set up of structural element matrices
$\mathbf{M}_S, \mathbf{C}_S, \mathbf{K}_S$

Eigenvalue analysis of the pseudo-linear system

$$[\mathbf{K}_S - \omega^2 \mathbf{M}_S]\boldsymbol{\phi} = \mathbf{0}$$

NSDOF = Number of structural degrees of freedom
NMODES = Number of modes included in the analysis

$\boldsymbol{\Phi}$ % Eigenvectors matrix (NSDOF x NMODES)

$\mathbf{B}_S = \boldsymbol{\Phi}^T \mathbf{M}_S \boldsymbol{\Phi}$ % Generalized structural mass
$\mathbf{D}_S = \boldsymbol{\Phi}^T \mathbf{C}_S \boldsymbol{\Phi}$ % Generalized structural damping
$\mathbf{E}_S = \boldsymbol{\Phi}^T \mathbf{K}_S \boldsymbol{\Phi}$ % Generalized structural stiffness

$\Upsilon_p = \frac{\boldsymbol{\phi}_p{}^T \mathbf{M}_S \boldsymbol{i}}{\boldsymbol{\phi}_p{}^T \mathbf{M}_S \boldsymbol{\phi}_p}$ % Participation factor of mode p

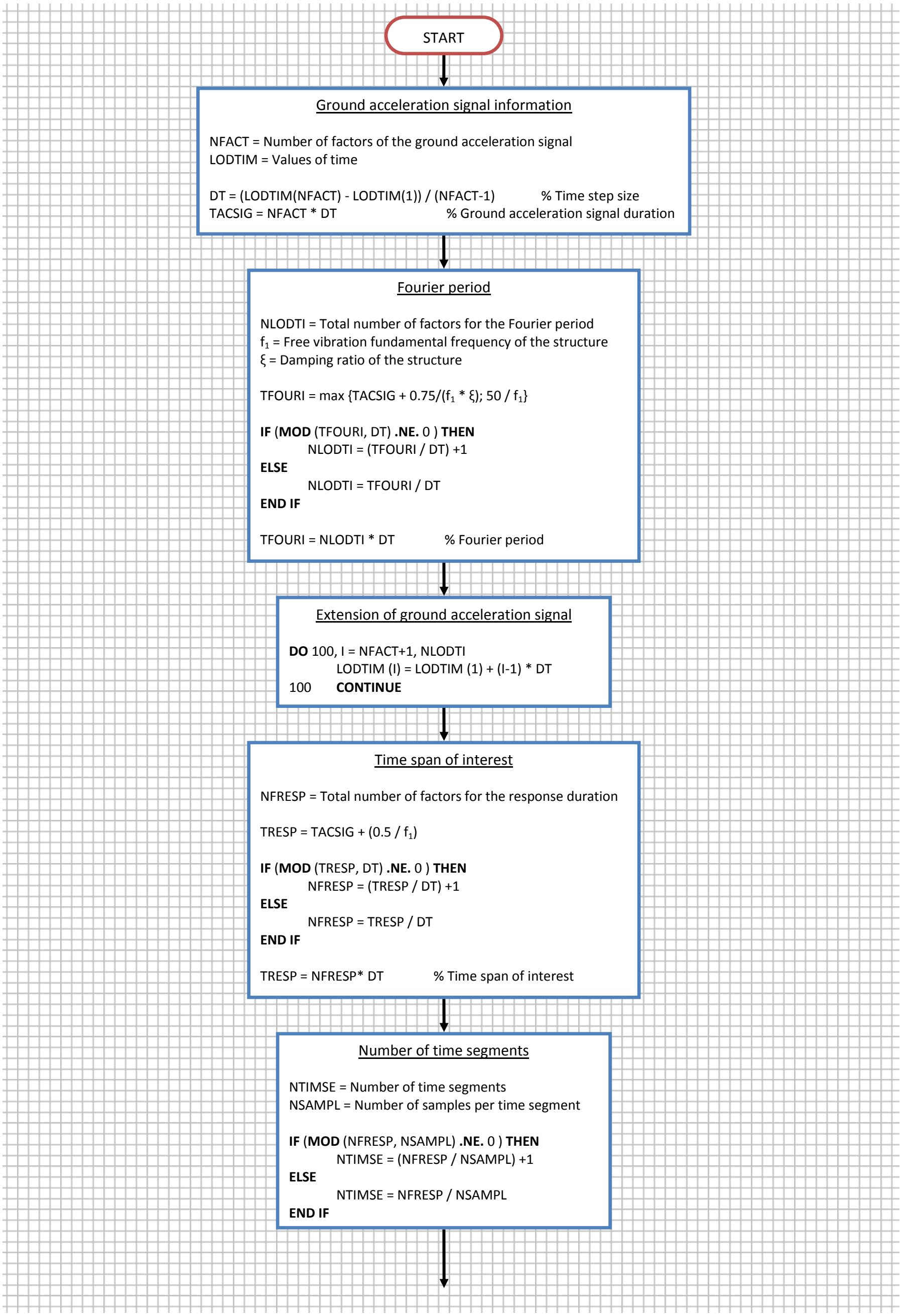
$\Gamma_{TX,p} = \frac{\boldsymbol{\phi}_p{}^T \mathbf{M}_S \boldsymbol{i}_{TX}}{\boldsymbol{\phi}_p{}^T \mathbf{M}_S \boldsymbol{\phi}_p}$ % Translational X-direction participation factor of mode p

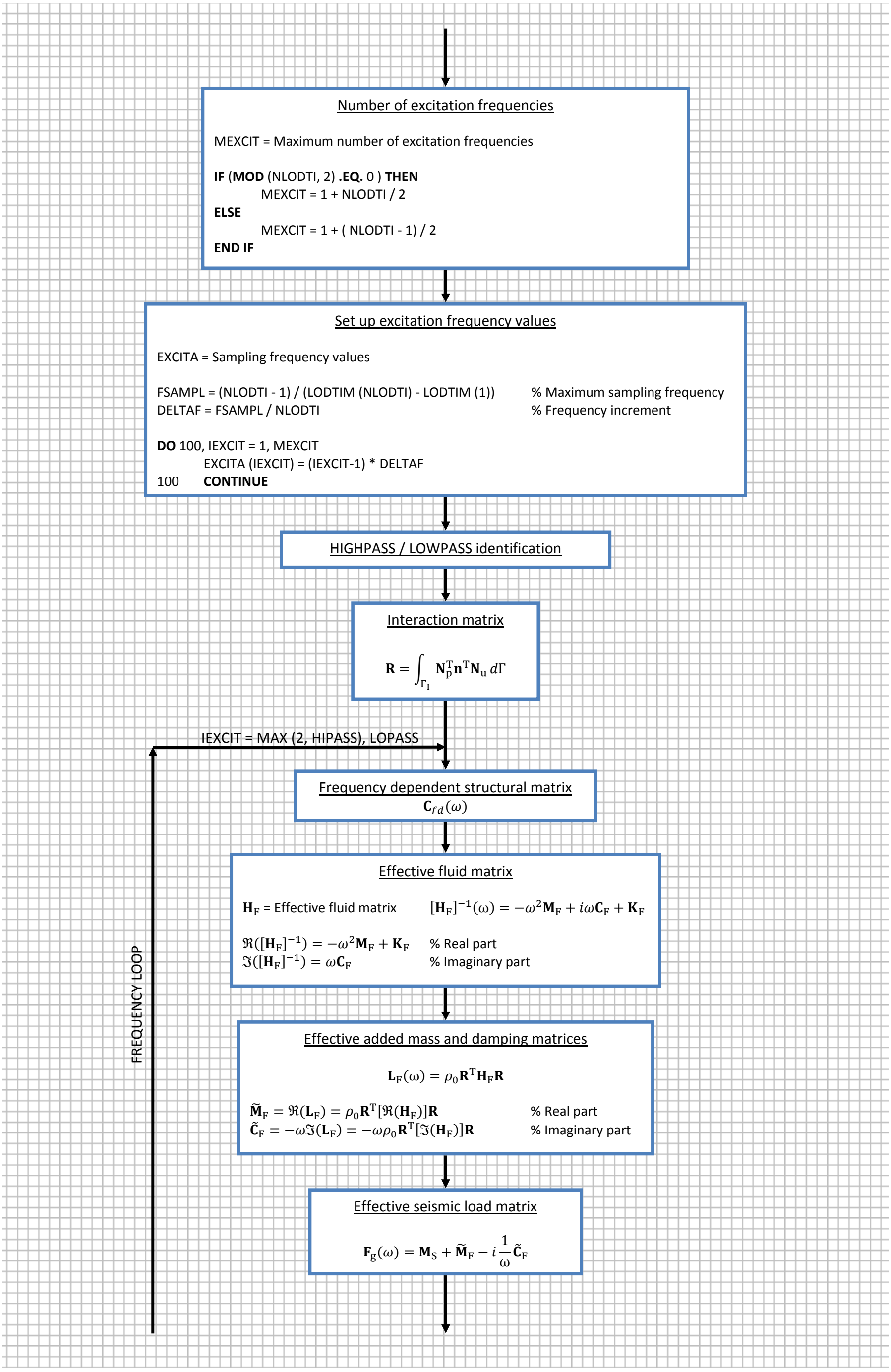
$m_{eff,TX,p} = \frac{(\boldsymbol{\phi}_p{}^T \mathbf{M}_S \boldsymbol{i}_{TX})^2}{\boldsymbol{\phi}_p{}^T \mathbf{M}_S \boldsymbol{\phi}_p}$ % Effective mass of mode p in the translational X-direction

HFRESP

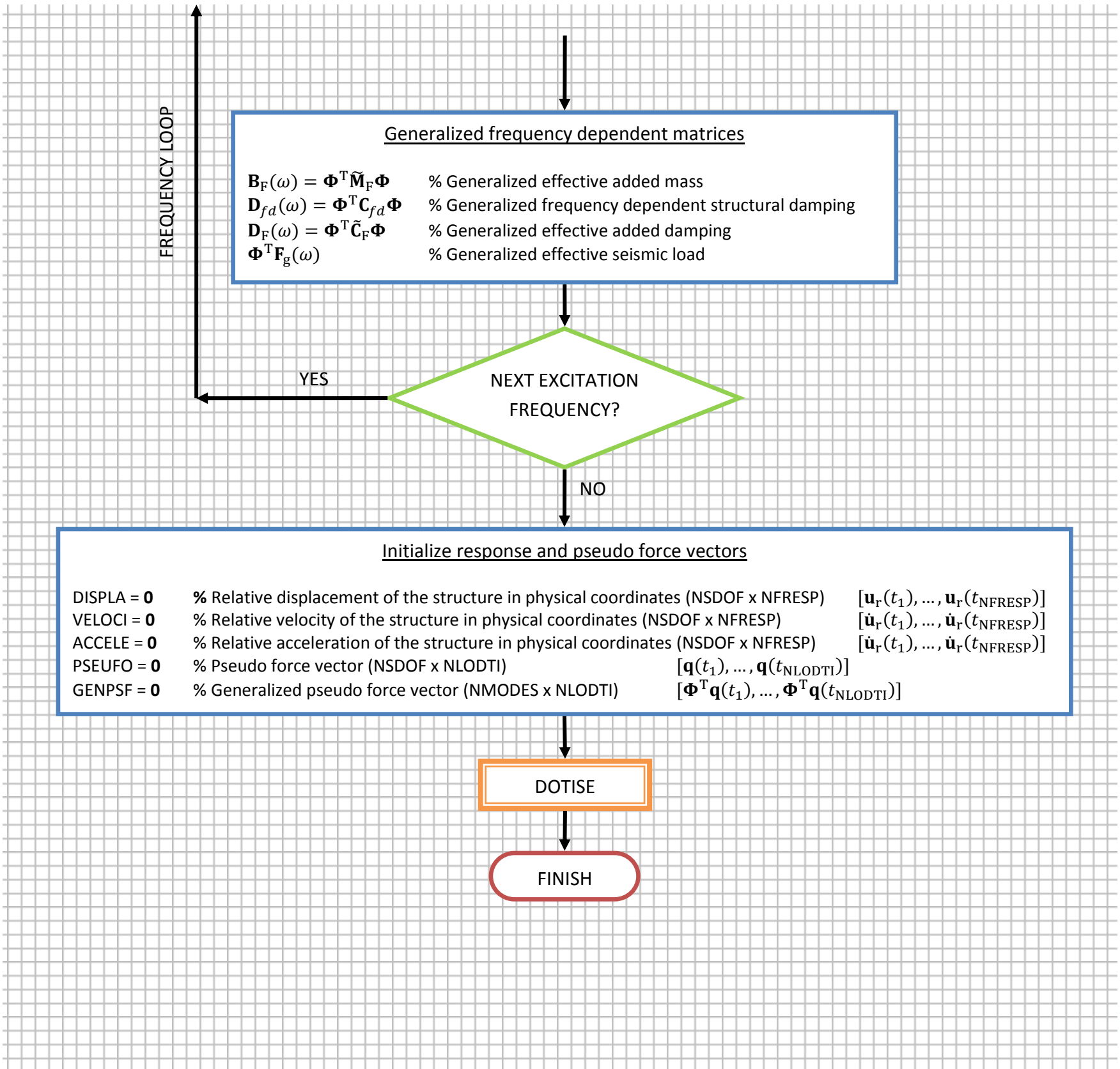
FINISH

A.2. HRESP function flowchart

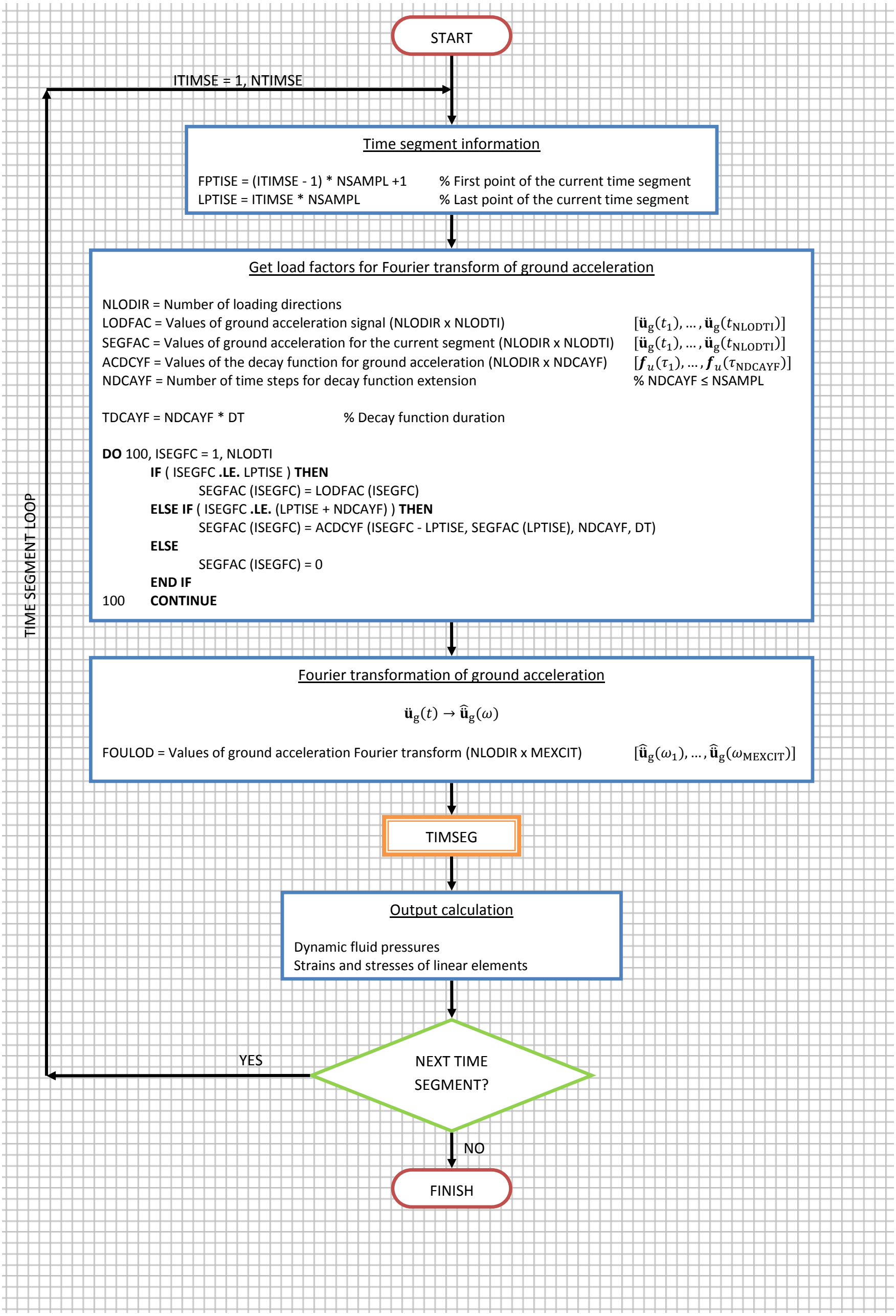



Number of excitation frequencies
MEXCIT = Maximum number of excitation frequencies
IF (MOD (NLODTI, 2) .EQ. 0) THEN
MEXCIT = 1 + NLODTI / 2
ELSE
MEXCIT = 1 + (NLODTI - 1) / 2
END IF
Set up excitation frequency values
EXCITA = Sampling frequency values
FSAMPL = (NLODTI - 1) / (LODTIM (NLODTI) - LODTIM (1)) % Maximum sampling frequency
DELTAF = FSAMPL / NLODTI % Frequency increment
DO 100, IEXCIT = 1, MEXCIT
EXCITA (IEXCIT) = (IEXCIT-1) * DELTAF
100 CONTINUE
HIGHPASS / LOWPASS identification
Interaction matrix
$\mathbf{R} = \int_{\Gamma_I} \mathbf{N}_p^T \mathbf{n}^T \mathbf{N}_u \, d\Gamma$
IEXCIT = MAX (2, HIPASS), LOPASS
Frequency dependent structural matrix
$\mathbf{C}_{fd}(\omega)$
Effective fluid matrix
$\mathbf{H}_F$ = Effective fluid matrix
$[\mathbf{H}_F]^{-1}(\omega) = -\omega^2 \mathbf{M}_F + i\omega \mathbf{C}_F + \mathbf{K}_F$
$\Re([\mathbf{H}_F]^{-1}) = -\omega^2 \mathbf{M}_F + \mathbf{K}_F$ % Real part
$\Im([\mathbf{H}_F]^{-1}) = \omega \mathbf{C}_F$ % Imaginary part
FREQUENCY LOOP
Effective added mass and damping matrices
$\mathbf{L}_F(\omega) = \rho_0 \mathbf{R}^T \mathbf{H}_F \mathbf{R}$
$\tilde{\mathbf{M}}_F = \Re(\mathbf{L}_F) = \rho_0 \mathbf{R}^T [\Re(\mathbf{H}_F)] \mathbf{R}$ % Real part
$\tilde{\mathbf{C}}_F = -\omega \Im(\mathbf{L}_F) = -\omega \rho_0 \mathbf{R}^T [\Im(\mathbf{H}_F)] \mathbf{R}$ % Imaginary part
Effective seismic load matrix
$\mathbf{F}_g(\omega) = \mathbf{M}_S + \tilde{\mathbf{M}}_F - i \frac{1}{\omega} \tilde{\mathbf{C}}_F$

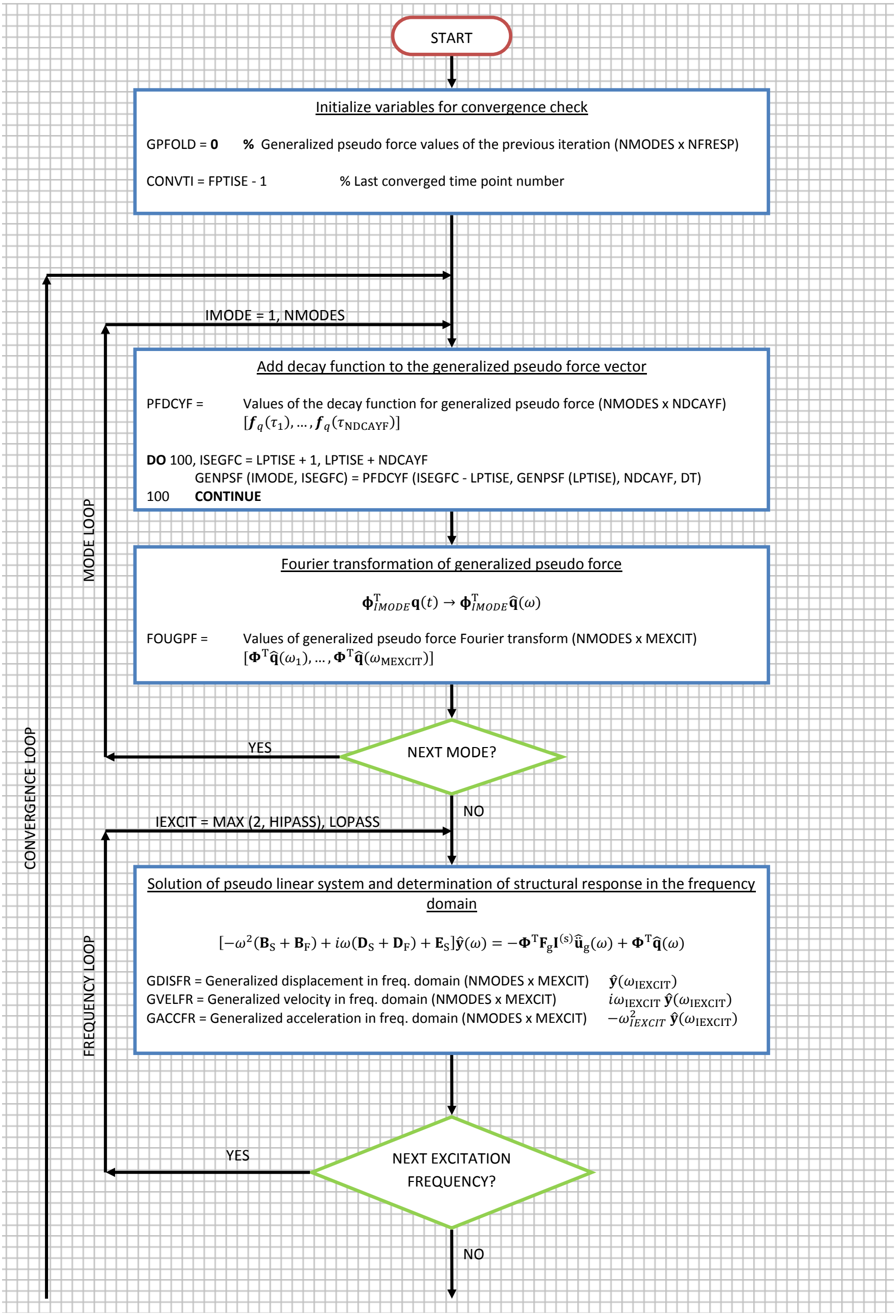
Appendix A: Flowchart of the HFTD Method Implementation in Diana



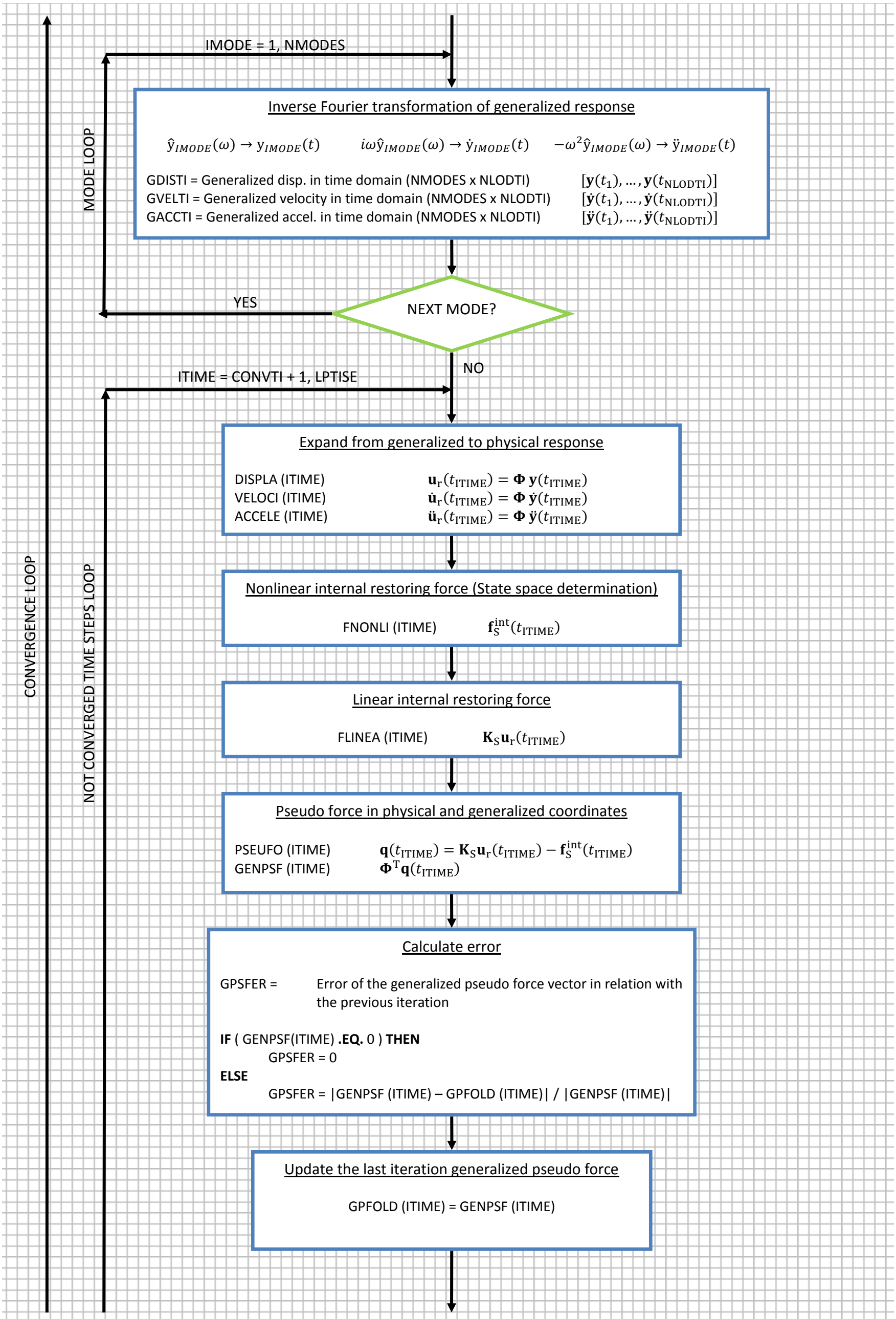
A.3. DOTISE function flowchart



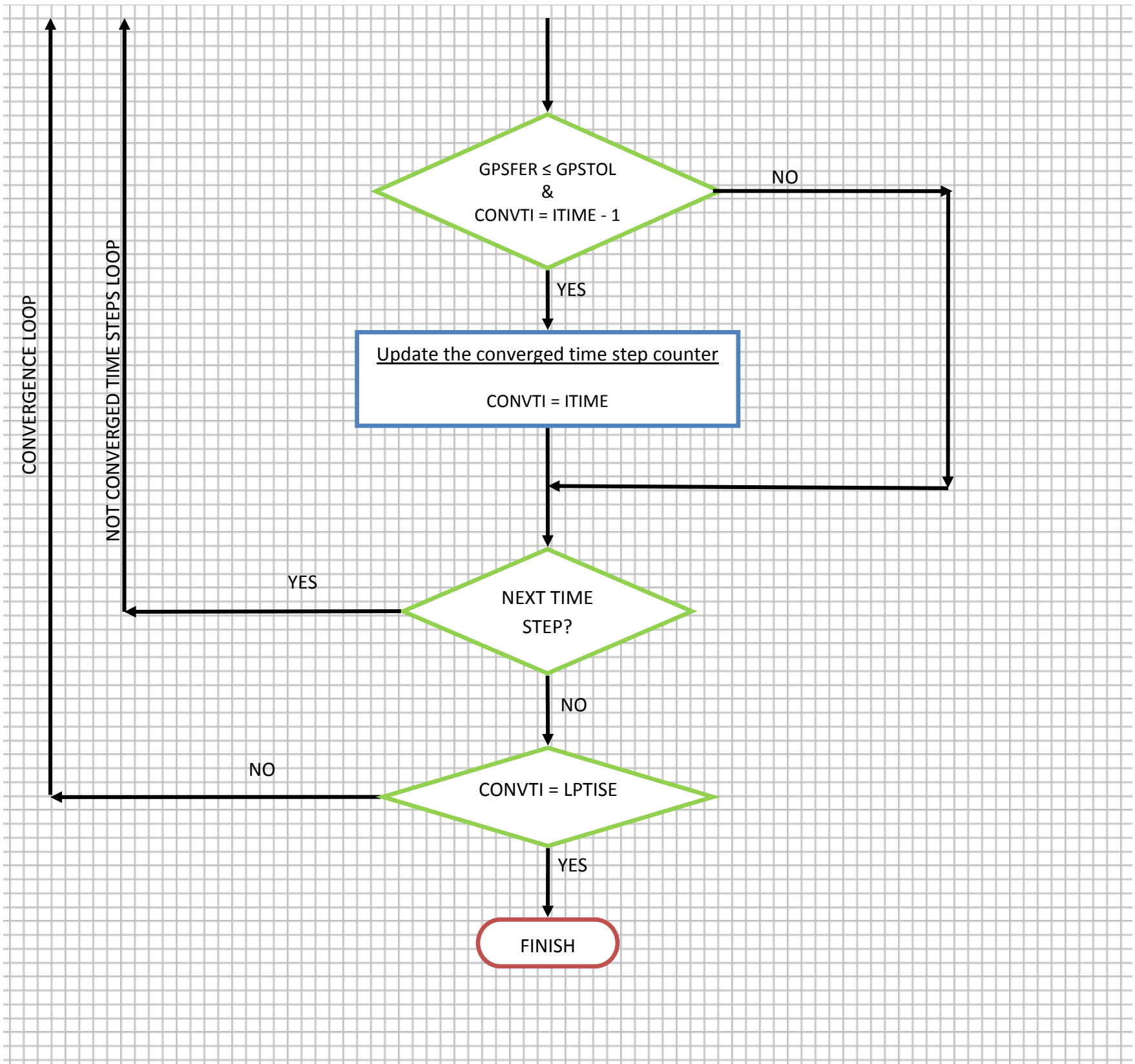
A.4. TIMSEG function flowchart



Appendix A: Flowchart of the HFTD Method Implementation in Diana



Appendix A: Flowchart of the HFTD Method Implementation in Diana



APPENDIX B: MATLAB ROUTINES

B.1. HFTD method code

B.1.1. Main HFTD script

```
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   HFTD for simplified MDOF systems
%       L. Fernando Sirumbal Z.
%       April, 2013
%
%   This script execute the HFTD method for linear and nonlinear simplfied
%   dynamic models of one or more degrees of freedom (DOF) subjected to
%   base (ground) acceleration loading.
%
%   After defining the system properties, the load signal and the HFTD
%   method parameters, the following results are obtained: time history
%   displacement, velocity, acceleration and nonlinear internal forces of
%   the system. Besides, the frequency response of the pseudo linear system
%   is also obtained.
%
%   The ground acceleration signal should be saved with the name "signal"
%   in the work folder as a matlab variable file called "signal.mat". The
%   values of the ground acceleration stored in the variable signal should
%   be normalized with respect to the gravity (g).
%
%   A Frequency dependent damping relation for the dashpots coefficients
%   can be specified in the function "f2_freq_dep_damp1".
%
%   The desired type of decaying function should be defined in the
%   function file "f5_decay_func".
%
%   The type of stiffness and damping non-linearity to be used in the
%   problem should be selected in Section 16.6.6 of this script.
%   The selected function file is used to determine FNONLI.
%   The nonlinear parameters of each type of non-linear function should be
%   defined inside the corresponding function file.
%
%   The output results are saved as excel tables, figures and matlab data
%   files. The file path where these output results files are saved should
%   be indicated Secttion 17 of this script and in the function files
%   "f15_excel_tables_v2" and "f16_plot_resp_v2".
%
%
```

Section 1: File name

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      All the output results of the program will be saved in the folder
%      "Output results" located in the work folder. The name specified by
%      the user is used to save the output files.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
clear;
```

```
disp('==========================================================================')
disp('   Problem name                  ')
disp('==========================================================================')
name=input('\nEnter the name of the file (this name is used to save the output data)... ','s');
disp('   ')
pause(1);
```

Section 2: Exercise data definition

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Input the mass, damping and stiffness matrices of the simplified MDOF
%   system. The gravity constant and the tolerance for non-linear analysis
%   are also input by the user.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
disp('==========================================================================')
disp('   Definition of the mass matrix                  ')
disp('==========================================================================')
pause(1);
lock=0;
while lock==0
    MCOF=input('\nEnter the value of the translational masses for each degree \nof freedom m (in
a row vector enter one mass per DOF).... ');
    sizem=size(MCOF);
    if sizem(1)~=1
        disp(' ')
        disp('Error message: The masses should be arranged in a row vector')
        pause(2);
    else
        lock=1;
    end
end
NDOF=sizem(2);
pause(1);
disp(' ')
disp('The number of degrees of freedom (NDOF) of the system is defined to be ')
NDOF
pause(1)
disp(' ')
disp('The masses of each degree of freedom are defined to be ')
MCOF
pause(1);
disp('  ')
disp('==========================================================================')
disp('   Definition of the elastic stiffness matrix                ')
disp('==========================================================================')
pause(1);
lock=0;
while lock==0
```

```
    KELAST=input('\nEnter the linear elastic stiffness coefficients k of the springs \n(in a row
vector enter NDOF+1 stiffness coefficients) .... ');
    sizek=size(KELAST);
    if sizek(1)~=1
        disp('  ')
        disp('Error message: The spring coefficients should be arranged in a row vector')
        pause(2);
    elseif sizek(2)~=NDOF+1
        disp(' ')
        disp('Error message: The row vectors defining the coefficientes of K should have NDOF+1
components ')
        pause(2);
    else
        lock=1;
    end
end
disp(' ')
disp('The linear elastic stiffness coefficients are defined to be')
KELAST
disp(' ')
pause(1);
disp('==========================================================================')
disp('   Definition of the damping matrices                  ')
disp('==========================================================================')
pause(1);
lock=0;
while lock==0
    CELAST=input('\nEnter the constant (not frequency dependent) linear elastic damping
coefficients c \nof the dashpots (in a row vector enter NDOF+1 damping coefficients)... ');
    sizec=size(CELAST);
    if sizec(1)~=1
        disp(' ')
        disp('Error message: The dashpots coefficients should be arranged in a row vector')
        pause(2);
    elseif sizec(2)~=NDOF+1
        disp(' ')
        disp('Error message: The row vectors defining the constant dashpot coefficients should
have NDOF+1 components ')
        pause(2);
    else
        lock=1;
    end
end
disp(' ')
disp('The constant (not frequency dependent) linear elastic damping coefficients are defined to
be ')
CELAST
pause(1);
lock=0;
while lock==0
```

```
    RAYLEI=input('\nEnter the coeficients for the definition of Rayleigh damping matrix aM+bK
\nin a row vector [a b]. If there is no Rayleigh damping enter [0 0]... ');
    sizer=size(RAYLEI);
    if sizer(1)~=1
        disp(' ')
        disp('Error message: The Rayleigh coefficients should be arranged in a row vector')
        pause(2);
    elseif sizer(2)~=2
        disp(' ')
        disp('Error message: The row vectors defining the Rayleigh coefficientes a and b should
have 2 components ')
        pause(2);
    else
        lock=1;
    end
end
a=RAYLEI(1);
b=RAYLEI(2);
disp(' ')
disp('The Rayleigh damping coefficients are defined to be ')
a
b
pause(1);
lock=0;
while lock==0
    HYSTER=input('\nEnter the hysteretic damping ratio. If there is no hysteretic damping enter
0... ');
    sizeh=size(HYSTER);
    if sizeh(1)==1 && sizeh(2)==1
        lock=1;
    else
        disp(' ')
        disp('Error message: The input value must be scalar')
    end
end
disp(' ')
disp('The hysteretic damping ratio is defined to be ')
HYSTER
disp(' ')
pause(1);
disp('=============================================================================')
disp('   Gravity constant                  ')
disp('=============================================================================')
pause(1);
lock=0;
while lock==0
    g=input('\nEnter the value of the gravity constant g .... ');
    sizeg=size(g);
    if sizeg(1)~= 1 || sizeg(2)~=1
        disp(' ')
```

```
        disp('Error message: g must be a scalar value')
        pause(2);
    else
        lock=1;
    end
end
disp(' ')
disp('The gravity constant is defined to be ')
g
disp('   ')
pause(1);
disp('=========================================================================')
disp('   Tolerance for convergence of non-linear analysis                  ')
disp('=========================================================================')
pause(1);
lock=0;
while lock==0
    tol=input('\nEnter the convergence tolerance tol for non-linear analysis.... ');
    sizet=size(tol);
    if sizet(1)~= 1 || sizet(2)~=1
        disp(' ')
        disp('Error message: tol must be a scalar value')
        pause(2);
    elseif tol>0.01 || tol<=0
        disp(' ')
        disp('Error message: tol must be positive and smaller than 10^-2')
        pause(2);
    else
        lock=1;
    end
end
disp(' ')
disp('The tolerance for convergence check is defined to be ')
tol
pause(2);
disp(' ')
```

Section 3: Construction of the linear elastic matrices of the system

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Based on the coefficients values defined in the previous section for
%   each mass, spring and dashpot, the linear elastic matrices of the
%   system are constructed using the function file "f1_linelast_mat".
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
disp('=========================================================================')
disp('   Linear elastic matrices                  ')
disp('=========================================================================')
%Mass matrix
```

```
M=zeros(NDOF);
for IDOF=1:NDOF
    M(IDOF,IDOF)=MCOF(IDOF);
end
disp(' ')
disp('The mass matrix M is')
M
pause(1);
%Stiffness matrix
K=f1_linelast_mat(NDOF,KELAST);
disp(' ')
disp('The linear elastic stiffness matrix K is')
K
pause(1);
%Damping matrix
Cnp=f1_linelast_mat(NDOF,CELAST);
disp(' ')
disp('The constant (not frequency dependent) linear elastic non proportional (dashpots) damping
matrix Cnp is')
Cnp
pause(1)
Cray=a*M+b*K;
disp(' ')
disp('The constant (not frequency dependent) linear elastic Rayleigh damping matrix Cray is')
Cray
pause(1)
C=Cnp+Cray;
disp(' ')
disp('The total constant (not frequency dependent) linear elastic damping matrix C is')
C
pause(1)
Ch=HYSTER*K;
disp(' ')
disp('The hysteretic linear elastic damping matrix Ch is')
Ch
pause(2);
disp(' ')
```

Section 4: Free vibration Eigen-analysis

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Eigen-analysis of the simplified system. Mode shape vectors and
%   eigen-frequencies of the system are calculated.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
disp('=========================================================================')
disp('   Free vibration natural frequencies and mode shape vectors                  ')
disp('=========================================================================')
[Phi,Natfreq]=eig(K,M);
```

```
Mgen=transpose(Phi)*M*Phi;
w1=sqrt(Natfreq(1,1));
T1=2*pi/w1;
pause(1);
disp('   ')
disp('The eigenvectors matrix is ')
disp('   ')
Phi
pause(1);
disp('   ')
disp('The eigenvalues (squared natural circular frequencies) matrix is')
disp('   ')
Natfreq
pause(1);
disp('   ')
disp('The fundamental period of the system is defined to be')
disp('   ')
T1
pause(1);
disp('   ')
disp('The fundamental circular frequency of the system is defined to be')
disp('   ')
w1
pause(1);
```

Section 5: Fundamental mode of vibration damping ratio

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Damping ratio of the fundamental mode of vibration taking into account
%   frecuency dependent and hysteretic damping. The frequency dependent
%   damping should be defined by the user in the function file
%   "f2_freq_dep_damp1".
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
disp('=========================================================================')
disp('   Damping ratio of the fundamental mode of vibration considering                  ')
disp('   hysteretic and frequency dependent damping                    ')
disp('=========================================================================')
[CFREDE1,EM]=f2_freq_dep_damp1(NDOF,w1);% Function defining the frequency dependent damping
coefficients
if EM==1
    return
end
Cfd1=f1_linelast_mat(NDOF,CFREDE1);
Ch1=(1/w1)*Ch;
pause(1);
disp('   ')
disp('The hysteretic damping matrix (Ch1) and the frequency dependent damping ')
disp('matrix (Cfd1) corresponding to the fundamental mode of vibration are defined to be ')
```

```
Ch1
Cfd1
pause(1);
C1=C+Ch1+Cfd1;
disp('  ')
disp('The total damping matrix (constant + hysteretic + frequency dependent) ')
disp('for the fundamental mode of vibration is determined to be ')
C1
pause(1);
C1gen=transpose(Phi(:,1))*C1*Phi(:,1);
xi1=(1/(2*w1))*(C1gen/Mgen(1,1));
disp('   ')
disp('The damping ratio of the fundamental mode of vibration of the system is defined to be')
disp('   ')
xi1
disp('   ')
pause(2);
```

Section 6: Ground acceleration signal information

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The data of the ground acceleration signal must be stored as a
%       matlab variable called "signal". This variable should be
%       defined as a matrix with two columns and as many rows as time
%       points available in the signal data. The first column contains the
%       time steps values, which should start in time = 0 seconds, and
%       should be equally spaced (constant time step size). The second
%       colum contains the ground acceleration values normalized with the
%       gravity acceleration g.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
disp('========================================================================')
disp('   Ground acceleration signal information                 ')
disp('========================================================================')
disp('   ')
disp('Loading the ground acceleration variable "signal" from file signal.mat saved in the work
folder...')
disp('   ')
pause(1);
load signal;
lock=1;
while lock~=0
    sizesig=size(signal);
    [SIGTIM,NFACT,SIGNDT,lock,k]=f3_ground_accel_signal_check(signal,sizesig);% Function that
checks the correct definition of the signal variable
    if lock==1
        disp('Error message: The first time step must be zero')
        return
    elseif lock==2
```

```
        disp('Error message: Signal data variable must have two columns')
        return
    elseif lock==3
        disp('Error message: Time step size of signal data is not constant')
        disp('   ')
        pause(2);
        fprintf('The problem was found between the time points %d and %d\n',k,k+1)
        return
    end
end
SIGFAC=g*signal(:,2);
TACSIG=NFACT*SIGNDT;
pause(2);
disp('   ')
disp('The number of time points contained in the ground acceleration signal is ')
disp('   ')
NFACT
pause(1);
disp('   ')
disp('The time step size of the ground acceleration signal is ')
disp('   ')
SIGNDT
pause(1);
disp('   ')
disp('The duration of the ground acceleration signal is ')
disp('   ')
TACSIG
pause(1);
disp('   ')
disp('The maximum ground acceleration magnitude contained in the signal is ')
disp('   ')
disp(max(abs(SIGFAC)))
disp('   ')
pause(2);
```

Section 7: Time step size and Fourier Period

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The time step size (DT) is calculated base on the signal
%       discretization.
%       The Fourier period length (TFOURI)is calculated based on the
%       fundamental mode of vibration period (T1) and damping ratio (xi1).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('=========================================================================')
disp('   Period for the Fourier transform                ')
disp('=========================================================================')
lock=0;
while lock==0
    pause(2);
```

```
    opt=input('\nSelect the option for the definition of the HFTD time step size (DT). Enter "0"
for DT to be \nequal to the signal step size (SIGNDT) or enter "1" for DT to be a multiple of
SIGNDT .... ');
    if opt==0
        TIMES=1;
        lock=1;
    elseif opt==1
        pause(2);
        TIMES=input('\nEnter the number of times that SIGNDT should be multiplied to obtain
DT.... ');
        if mod(TIMES,1)~=0 || TIMES<=0
            disp('   ')
            disp('Error message: The input value should be a positive integer')
        else
            lock=1;
        end
    else
        pause(2);
        disp('   ')
        disp('Error message: The input value must be 0 or 1')
    end
end
DT=TIMES*SIGNDT;
prob=[TACSIG+0.75*T1/abs(xi1),50*T1,2*TACSIG];
TFOURI=max(prob);
NLODTI=ceil(TFOURI/DT);
TFOURI=DT*NLODTI;
pause(1);
disp('    ')
disp('The number of time points (NLODTI) contained in the Fourier period is calculated to be ')
disp('    ')
NLODTI
pause(1);
disp('    ')
disp('The HFTD time step size (DT) is defined to be')
disp('    ')
DT
pause(1);
disp('    ')
disp('The Fourier period (TFOURI) is calculated to be')
disp('    ')
TFOURI
disp('    ')
pause(2)
```

Section 8: Extension of ground acceleration signal

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The ground acceleration data is extended adding zeros (quiet zone)
```

```
%       until the Fourier period is covered.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
LODTIM=zeros(NLODTI,1);
LODFAC=zeros(NLODTI,1);
for k=1:NLODTI
    LODTIM(k)=(k-1)*DT;
    j=1+(k-1)*TIMES;
    if j<=NFACT
        LODFAC(k)=SIGFAC(j);
    else
        LODFAC(k)=0;
    end
end
```

Section 9: Time span of interest

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       Time span for which the reponses are calculated.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('==========================================================================')
disp('   Time span of interest                ')
disp('==========================================================================')
lock=0;
while lock==0
    pause(2);
    opt=input('\nSelect the option for the definition of the time span of interest: Enter "1" for
the time span of interest to be \ndefined by the user or enter "0" for it to be calculated by te
program .... ');
    if opt==0
        TRESP=TACSIG+0.5*T1;
        NFRESP=ceil(TRESP/DT);
        TRESP=DT*NFRESP;
        lock=1;
    elseif opt==1
        pause(1);
        TRESP=input('\nEnter the time span of interest in seconds .... ');
        NFRESP=ceil(TRESP/DT);
        TRESP=DT*NFRESP;
        if TRESP>=TFOURI/2 || TRESP<=0
            disp('    ')
            disp('Error message: The time span of interest should be posistive and not greater
than half the Fourier period (TFOURI)')
        else
            lock=1;
        end
    else
        pause(1);
        disp('    ')
        disp('Error message: The input value must be 0 or 1')
```

```
    end
end
disp('    ')
disp('The number of time steps defining the time span of interest length (NFRESP) is defined to
be')
disp('    ')
NFRESP
pause(1);
disp('    ')
disp('The response time span of interest (TRESP) is defined to be')
disp('    ')
TRESP
disp('    ')
pause(2);
```

Section 10: Number of time segments

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       In order to achieve accurate results, HFTD requires that the time
%       span of interest be divided in segments containing a certain number
%       (NSAMPL) of time steps (DT). The convergence and iterative process
%       is carried out sequentially, segment by segment, one after the
%       other.
%       One extreme case (not recommended) is to have only one time segment
%       (NSAMPL*DT=TRESP). On the other hand, the other extreme case is to
%       have as many segements as time steps (NSAMPL=1). Intermediate case
%       is reccommended taking into account that increasing the number of
%       time segments provides better accuracy and stability to the HFTD
%       method.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('=========================================================================')
disp('   Division of the time span of interest in time segments               ')
disp('=========================================================================')
lock=0;
while lock==0
    pause(2);
    NSAMPL=input('\nEnter the number of time steps (DT) per time segment .... ');
    if mod(NSAMPL,1)==0 && NSAMPL>=1 && NSAMPL<=NFRESP
        lock=1;
    else
        pause(1);
        fprintf('\nError message: The number of samples per time segment (NSAMPL) should be a
positive \ninteger such that NSAMPL*DT is not greater than TRESP')
    end
end
disp('    ')
disp('The number of samples per time segment (NSAMPL) is defined to be')
disp('    ')
NSAMPL
```

```
disp('    ')
NTIMSE=ceil(NFRESP/NSAMPL);
pause(1);
disp('The number of time segments (NTIMSE) through all the interest time span is defined to be')
disp('    ')
NTIMSE
disp('    ')
pause(2);
```

Section 11: Decay function length

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       For the Fourier transformation of the ground acceleration and the
%       pseudo force vector, an decaying function is appended at the end of
%       the current time segment being analyzed.
%       The number of time steps (DT) that define the decay function length
%       is stored in the variable NDCAYF, which should be smaller or equal
%       than NSAMPL (the decay function should not extend more than the
%       time segment length).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('    ')
disp('=============================================================================')
disp('   Number of time steps defining the decaying function length              ')
disp('=============================================================================')
lock=0;
while lock==0
    pause(2);
    NDCAYF=input('\nEnter the number of time steps (DT) for the definition of the decaying
function length .... ');
    if NDCAYF>=0 && mod(NDCAYF,1)==0 && NDCAYF<=NSAMPL
        lock=1;
    else
        disp('    ')
        fprintf('Error message: The number of time steps defining the decaying function length
(NDCAYF) should be \nzero or a positive integer not greater than NSAMPL\n')
    end
end
TDCAYF=NDCAYF*DT;
disp('    ')
disp('The number of time steps defining the length of the decaying function (NDCAYF) is defined
to be')
NDCAYF
pause(1)
disp('    ')
disp('The lentgh of the decaying function (TDCAYF) is defined to be')
TDCAYF
disp('    ')
pause(2);
```

Section 12: Set up excitation frequencies

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The number of excitation frequencies (MEXCITA), the frequency
%       increment (DELTAF) and the maximum sampling frequency (FSAMPL) are
%       calculated based on the Fourier Period (TFOURI) and its number of
%       time points (NLODTI).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('===========================================================================================
=========================================')
fprintf('   Number of excitation frequencies, excitation frequency values, frequency increment \n
and maximum sampling frequency.               \n')
disp('===========================================================================================
=========================================')
if mod(NLODTI,2)==0
    MEXCIT=1+NLODTI/2;
else
    MEXCIT=1+(NLODTI-1)/2;
end
FSAMPL=(NLODTI-1)/(LODTIM(NLODTI)-LODTIM(1));
DELTAF=FSAMPL/NLODTI;
EXCITA=zeros(MEXCIT,1);
for IEXCIT=1:MEXCIT
    EXCITA(IEXCIT)=(IEXCIT-1)*DELTAF;
end
pause(2);
disp('    ')
disp('The maximum number of frequencies (MEXCIT) is')
disp('    ')
MEXCIT
disp('    ')
pause(1);
disp('The frequency increment (DELTAF) is')
disp('    ')
DELTAF
disp('    ')
pause(1);
disp('The maximum sampling frequency (FSAMPL) is')
disp('    ')
FSAMPL
disp('    ')
pause(1);
disp('The maximum excitation frequency that can be accurately included in the analysis is')
disp('    ')
disp(FSAMPL/2)
disp('    ')
pause(2);
```

Section 13: Highpass/Lowpass frequency identification

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The frequency response can be determined for a restricted range of
%       frequency instead of the frequency range from 0 to EXCITA(MEXCIT).
%       This restriction is (optionally) defined by the user after
%       specifying a minimum (Highpass) and a maximum (Lowpass) response
%       frequency.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('=============================================================================')
disp('   Highpass/Lowpass frequency identification              ')
disp('=============================================================================')
lock=0;
while lock==0
    pause(2);
    MINFRE=input('\nEnter the minimum frequency to be included in the analysis .... ');
    pause(1);
    if MINFRE>=0 && MINFRE<=EXCITA(MEXCIT)
        lock=1;
    else
        disp('    ')
        fprintf('Error message: The minimum frequency (MINFRE) should be zero or a positive value
\nnot greater than (MEXCIT-1)*DELTAF\n')
    end
end
lock=0;
while lock==0
    MAXFRE=input('\nEnter the maximum frequency to be included in the analysis .... ');
    pause(1);
    disp('    ')
    if MAXFRE>=MINFRE && MAXFRE<=EXCITA(MEXCIT)
        lock=1;
    else
        fprintf('Error message: The maximum frequency (MAXFRE) should higher or equal than MINFRE
\nand not greater than (MEXCIT-1)*DELTAF\n')
    end
end
[LOPASS,HIPASS]=f4_high_low_pass_freq(EXCITA,MEXCIT,MAXFRE,MINFRE);
disp('The HIGHPASS frequency is found to be')
disp('    ')
disp(EXCITA(HIPASS))
disp('    ')
pause(1);
disp('The LOWPASS frequency is found to be')
disp('    ')
disp(EXCITA(LOPASS))
```

Section 14: Hysteretic and frequency dependent damping for each excitation frequency

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The hysteretic and frequency dependent damping matrices are
%       calculated for each excitattion frequency between HIPASS and
%       LOPASS. The frequency dependent damping should be defined by the
%       user in the function file "f2_freq_dep_damp1".
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
CFREDE=zeros(LOPASS-HIPASS+1,NDOF+1);
Cfd=zeros(NDOF,NDOF,LOPASS-HIPASS+1);
Chy=zeros(NDOF,NDOF,LOPASS-HIPASS+1);
for IEXCIT=HIPASS:LOPASS
    w=2*pi*EXCITA(IEXCIT);
    [CFREDE(IEXCIT-HIPASS+1,:),EM]=f2_freq_dep_damp1(NDOF,w); %Function defining the frequency
dependent damping
    Cfd(:,:,IEXCIT-HIPASS+1)=f1_linelast_mat(NDOF,CFREDE(IEXCIT-HIPASS+1,:));
    if w==0
        Chy(:,:,IEXCIT-HIPASS+1)=0*Ch;
    else
        Chy(:,:,IEXCIT-HIPASS+1)=(1/w)*Ch;
    end
end
```

Section 15: Initialization of variables

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       Set up to zero the response, pseudo force and internal restoring
%       force variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
NRESIT=NTIMSE*NSAMPL;
DISPLA=zeros(NDOF,NRESIT);
VELOCI=zeros(NDOF,NRESIT);
ACCELE=zeros(NDOF,NRESIT);
PSEUFO=zeros(NDOF,NLODTI);
FNONLI=zeros(NDOF,NRESIT);
FLINEA=zeros(NDOF,NRESIT);
KEY=zeros(NDOF+1,NRESIT);
FNLSTI=zeros(NDOF,NRESIT);
SPRFOR=zeros(NDOF+1,NRESIT);
FNLDAM=zeros(NDOF,NRESIT);
DASHFO=zeros(NDOF+1,NRESIT);
UT=zeros(NDOF+1,NRESIT);
UC=zeros(NDOF+1,NRESIT);
UO=zeros(NDOF+1,NRESIT);
disp('    ')
pause(1);
```

Section 16: HFTD method execution for each time segment

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The HFTD calculation process is executed segment by segment. A time
%       segment loop is defined in order to repeat the process for the
%       NTIMSE segments previously determined.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('========================================================================')
disp('   HFTD method execution for each time segment             ')
disp('========================================================================')
for ITIMSE=1:NTIMSE

    pause(1);
    disp('    ')
    disp('Starting the HFTD calculation process: Time segment number ... ')
    disp(ITIMSE)
```

Section 16.1: Time segment information

```
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %       The index of the first (FPTISE) and last (LPTISE) point of the
    %       current time segment (ITIMSE) are identified.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    FPTISE=(ITIMSE-1)*NSAMPL+1;
    LPTISE=ITIMSE*NSAMPL;
```

Section 16.2: Ground acceleration decay function definition for the current segment

```
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %       The function "f5_decay_func" is called in order to determine
    %       the ground acceleration factors pertaining to the decay
    %       function appended at the end of the current time segment.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    [GADCYF]=f5_decay_func(LODFAC,LPTISE,NDCAYF,DT,TDCAYF);
```

Section 16.3: Load factors for the Fourier transform of the ground acceleration

```
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %       The load factors used for the Fourier transform of the ground
    %       acceleration vary in each time segment. The ground acceleration
    %       load factors including the decaying function used for the
    %       current time segment are stored in the variable SEGFAC.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    SEGFAC=LODFAC;
    SEGFAC(LPTISE+1:LPTISE+NDCAYF)=GADCYF;
    SEGFAC(LPTISE+NDCAYF+1:NLODTI)=zeros;
```

Section 16.4: Fast Fourier transform of the ground acceleration loading

```
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %       The FFT function is executed in the variable SEGFAC by
    %       means of the function "f6_fast_fourier_transf".
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    disp('Fourier transform of the ground acceleration signal ... ')
    [FOUACR,FOUACI]=f6_fast_fourier_transf(SEGFAC,NLODTI,MEXCIT);
    FOUACC=FOUACR+1i*FOUACI;
```

Section 16.5: Initialize variables for convergence check

```
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %       Set up to zero the response, pseudo force and internal
    %       restoring force variables
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    PSFOLD=zeros(NDOF,NSAMPL); %Pseudo force values of the last iteration within the current
segment.
    DISOLD=zeros(NDOF,NSAMPL); %Displacement values of the last iteration within the current
segment.
    PFDCYF=zeros(NDOF,NDCAYF); %Decay function for the pseudo force vector.
    FOUPFR=zeros(NDOF,MEXCIT); %Direct Fourier transform of the pseudo force vector (real part).
    FOUPFI=zeros(NDOF,MEXCIT); %Direct Fourier transform of the pseudo force vector (imaginary
part).
    IFOUDR=zeros(NDOF,NSAMPL); %Inverse Fourier transform of the displacement vector(real part)
for the time points inside the current time segment.
    IFOUDI=zeros(NDOF,NSAMPL); %Inverse Fourier transform of the displacement vector(imaginary
part) for the time points inside the current time segment.
    IFOUVR=zeros(NDOF,NSAMPL); %Inverse Fourier transform of the velocity vector(real part) for
the time points inside the current time segment.
    IFOUVI=zeros(NDOF,NSAMPL); %Inverse Fourier transform of the velocity vector(imaginary part)
for the time points inside the current time segment.
    IFOUAR=zeros(NDOF,NSAMPL); %Inverse Fourier transform of the acceleration vector(real part)
for the time points inside the current time segment.
    IFOUAI=zeros(NDOF,NSAMPL); %Inverse Fourier transform of the acceleration vector(imaginary
part) for the time points inside the current time segment.
    CONVTI=FPTISE-1;           %Counter of the last time steps already converged.
```

Section 16.6: Convergence loop

```
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %       The procedure below is repeated until achieving convergence in
    %       all the time steps within the current segment.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    iter=0;
    convlock=0;
    pause(1);
    disp('    ')
```

```
    fprintf('Starting the iteration process untill convergence is achieved for all the time steps
\ninside segment number... %d\n',ITIMSE)
    while convlock==0
```

```
        iter=iter+1;
        disp('    ')
        disp('Starting iteration number... ')
        disp(iter)
        disp('    ')
        disp('Fourier transform of the pseudo force vector ... ')
        for IDOF=1:NDOF
```

Section 16.6.1: Pseudo force decay function definition

```
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %       The function "f5_decay_func is called in order to
            %       determine the pseudo force factors pertaining to the
            %       decay function appended at the end of the current time
            %       segment.
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            [PFDCYF(IDOF,:)]=f5_decay_func(PSEUFO(IDOF,:),LPTISE,NDCAYF,DT,TDCAYF);
```

Section 16.6.2: Load factors for the Fourier transform of the pseudo force

```
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %       The load factors used for the Fourier transform of the
            %       pseudo force vary in each iteration in accordance with
            %       the results obtained in the previous iteration. The
            %       pseudo force load factors including the decaying
            %       function used for the current time segment and
            %       iteration are stored in the variable PSEUFO.
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            PSEUFO(IDOF,LPTISE+1:LPTISE+NDCAYF)=PFDCYF(IDOF,:);
```

Section 16.6.3: Fast Fourier transform of the pseudo force

```
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %       The FFT function is executed in the variable PSEUFO by
            %       means of the function "f6_fast_fourier_transf".
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            [FOUPFR(IDOF,:),FOUPFI(IDOF,:)]=f6_fast_fourier_transf(PSEUFO(IDOF,:),NLODTI,MEXCIT);
```

```
        end
```

Section 16.6.4: Pseudo linear equation of motion in the frequency domain

```
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %    Solution of the pseudo linear system for each excitation
        %    frequency.
        %
        %       ${-(w^2)*M + i*w*C + K} u(w) = -M*I*ug(w) + q(w)$
        %       [-(w^2)*M + i*w*C + K] u(w) = -M*I*ug(w) + q(w)
        %   The equation is solved using the direct method defined in the
        %   function "f7_solver_pseudo_lin_direct_freq_v2".
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        disp('    ')
        disp('Starting the solution of the pseudo linear system in the frequency domain ... ')
        DISFRE=zeros(NDOF,NLODTI);
        VELFRE=zeros(NDOF,NLODTI);
        ACCFRE=zeros(NDOF,NLODTI);
        DIFREA=zeros(NDOF,MEXCIT);
        DIFRIM=zeros(NDOF,MEXCIT);
        for IEXCIT=HIPASS:LOPASS
            w=2*pi*EXCITA(IEXCIT);

[DIFREA(:,IEXCIT),DIFRIM(:,IEXCIT)]=f7_solver_pseudo_lin_direct_freq_v2(K,C,Cfd(:,:,IEXCIT-
HIPASS+1),Chy(:,:,IEXCIT-
HIPASS+1),M,w,NDOF,FOUACR(IEXCIT),FOUACI(IEXCIT),FOUPFR(:,IEXCIT),FOUPFI(:,IEXCIT));
            DISFRE(:,IEXCIT)=DIFREA(:,IEXCIT)+1i*DIFRIM(:,IEXCIT);  % Displacement in the
frequency domain
            VELFRE(:,IEXCIT)=1i*w*DISFRE(:,IEXCIT);    % Velocity in the frequency domain
            ACCFRE(:,IEXCIT)=-(w^2)*DISFRE(:,IEXCIT);  % Acceleration in the frequency domain
        end
```

Section 16.6.5: Inverse Fourier transform of the response

```
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %       The complex displacement (DISFRE), velocity (VELFRE) and
        %       acceleration (ACCFRE) vectors in the frequency domain are
        %       translated to the time domain by means of the function
        %       "f8_inv_fast_fourier_transf".
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        disp('    ')
        disp('Inverse Fourier transform of the displacement, velocity and acceleration vectors
... ')
        for IDOF=1:NDOF

[IFOUDR(IDOF,:),IFOUDI(IDOF,:)]=f8_inv_fast_fourier_transf(DISFRE(IDOF,:),NLODTI,FPTISE,LPTISE);
% Inverse Fourier transform  of the displacement vector for the time steps within the current
segment.

[IFOUVR(IDOF,:),IFOUVI(IDOF,:)]=f8_inv_fast_fourier_transf(VELFRE(IDOF,:),NLODTI,FPTISE,LPTISE);
% Inverse Fourier transform  of the velocity vector for the time steps within the current
```

```
segment.

[IFOUAR(IDOF,:),IFOUAI(IDOF,:)]=f8_inv_fast_fourier_transf(ACCFRE(IDOF,:),NLODTI,FPTISE,LPTISE);
% Inverse Fourier transform  of the acceleration vector for the time steps within the current
segment.
        end
```

Section 16.6.6: Not converged time steps loop

```
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %       In this loop, the new iteration values of the displacement,
        %       velocity, acceleration and pseudo force vectors are
        %       calculated and updated only for the time steps inside the
        %       current segment which have not converged in the previous
        %       iterations.
        %       The functions defining the stiffness and damping
        %       non-linearities are indicated below.
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        disp('    ')
        disp('Calculation of the response, internal restoring forces and pseudo force for the non
converged time steps ... ')
        for ITIME=CONVTI+1:LPTISE

            DISPLA(:,ITIME)=IFOUDR(:,ITIME-FPTISE+1); % Displacement vector
            VELOCI(:,ITIME)=IFOUVR(:,ITIME-FPTISE+1); % Velocity vector
            ACCELE(:,ITIME)=IFOUAR(:,ITIME-FPTISE+1); % Acceleration vector
            if ITIME==1

                CEROS1=zeros(NDOF,1);
                CEROS2=zeros(NDOF+1,1);
```

Section 16.6.6.1: Non-linear or linear stiffness force, internal spring force and state variable of time step 1

```
%[KEY(:,ITIME),SPRFOR(:,ITIME),FNLSTI(:,ITIME),UT(:,ITIME),UC(:,ITIME),UO(:,ITIME),EMKEY]=f9_lin_
stiff_int_force(NDOF,CEROS2,CEROS1,DISPLA(:,ITIME),CEROS2,KELAST);

%[KEY(:,ITIME),SPRFOR(:,ITIME),FNLSTI(:,ITIME),UT(:,ITIME),UC(:,ITIME),UO(:,ITIME),EMKEY]=f10_non
lin_hyster_stiff_int_force_v3(VELOCI(:,ITIME),NDOF,CEROS2,CEROS1,DISPLA(:,ITIME),CEROS2,CEROS2,CE
ROS2,CEROS2,ITIME,KELAST);

[KEY(:,ITIME),SPRFOR(:,ITIME),FNLSTI(:,ITIME),UT(:,ITIME),UC(:,ITIME),UO(:,ITIME),EMKEY]=f11_nonl
in_hyster_stiff_int_force_cubic(VELOCI(:,ITIME),NDOF,CEROS2,CEROS1,DISPLA(:,ITIME),CEROS2,CEROS2,
CEROS2,ITIME,KELAST);
```

Section 16.6.6.2: Non-linear or linear damping force, internal dashpot force and state variable of time step 1

```
%[FNLDAM(:,ITIME),DASHFO(:,ITIME)]=f12_lin_damp_int_force_v2(NDOF,VELOCI(:,ITIME),CEROS2,CELAST);
%Nonlinear damping force and internal dashpot force of time step i+1

[FNLDAM(:,ITIME),DASHFO(:,ITIME)]=f13_nonlinmet_damp_int_force(NDOF,DISPLA(:,ITIME),VELOCI(:,ITIM
E),CELAST);
```

```
            else
```

Section 16.6.6.3: Non-linear or linear stiffness force, internal spring force and state variable of time step i+1

```
%[KEY(:,ITIME),SPRFOR(:,ITIME),FNLSTI(:,ITIME),UT(:,ITIME),UC(:,ITIME),UO(:,ITIME),EMKEY]=f9_lin_
stiff_int_force(NDOF,KEY(:,ITIME-1),DISPLA(:,ITIME-1),DISPLA(:,ITIME),SPRFOR(:,ITIME-1),KELAST);

%[KEY(:,ITIME),SPRFOR(:,ITIME),FNLSTI(:,ITIME),UT(:,ITIME),UC(:,ITIME),UO(:,ITIME),EMKEY]=f10_non
lin_hyster_stiff_int_force_v3(VELOCI(:,ITIME),NDOF,KEY(:,ITIME-1),DISPLA(:,ITIME-
1),DISPLA(:,ITIME),SPRFOR(:,ITIME-1),UT(:,ITIME-1),UC(:,ITIME-1),UO(:,ITIME-1),ITIME,KELAST);

[KEY(:,ITIME),SPRFOR(:,ITIME),FNLSTI(:,ITIME),UT(:,ITIME),UC(:,ITIME),UO(:,ITIME),EMKEY]=f11_nonl
in_hyster_stiff_int_force_cubic(VELOCI(:,ITIME),NDOF,KEY(:,ITIME-1),DISPLA(:,ITIME-
1),DISPLA(:,ITIME),UT(:,ITIME-1),UC(:,ITIME-1),UO(:,ITIME-1),ITIME,KELAST);
```

Section 16.6.6.4: Non-linear or linear damping force, internal dashpot force and state variable of time step i+1

```
%[FNLDAM(:,ITIME),DASHFO(:,ITIME)]=f12_lin_damp_int_force_v2(NDOF,VELOCI(:,ITIME)-VELOCI(:,ITIME-
1),DASHFO(:,ITIME-1),CELAST); %Nonlinear damping force and internal dashpot force of time step
i+1

[FNLDAM(:,ITIME),DASHFO(:,ITIME)]=f13_nonlinmet_damp_int_force(NDOF,DISPLA(:,ITIME),VELOCI(:,ITIM
E),CELAST);
```

```
            end
            if EMKEY~=100
                return
            end
```

Section 16.6.6.5: Total non-linear force, linear force and pseudo force

```
            FNONLI(:,ITIME)=FNLSTI(:,ITIME)+FNLDAM(:,ITIME); % Nonlinear internal restoring force
            FLINEA(:,ITIME)=f14_linear_int_rest_force(K,DISPLA(:,ITIME),Cnp,VELOCI(:,ITIME)); %
Linear internal restoring force vector
            PSEUFO(:,ITIME)=FLINEA(:,ITIME)-FNONLI(:,ITIME); % Pseudo force vector
```

```
        end
```

Section 16.6.7: Convergence check

```
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %       The convergence criteria is checked for the time steps from
        %       CONVTI+1 to LPTISE. If after checking these points some of
        %       them have not converged yet, the convergence loop is
        %       repeated all over again for those new not converged tme
        %       steps. On the other hand, if all the time steps of the
        %       current segment (until LPTISE) have converged the
        %       convergence loop is finished and a new time segment
        %       calculation is started.
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        pause(1);
        disp('  ')
        disp('Convergence check ... ')
        OLDCOT=CONVTI;
        convlock=1;
        while CONVTI~=LPTISE
            TT=CONVTI+1;
            ST=CONVTI-FPTISE+2;
            if norm(PSEUFO(:,TT))==0
                PSFERR=norm(PSEUFO(:,TT)-PSFOLD(:,ST));
            else
                PSFERR=norm(PSEUFO(:,TT)-PSFOLD(:,ST))/norm(PSEUFO(:,TT));
            end
            if norm(DISPLA(:,TT))==0
                DISERR=norm(DISPLA(:,TT)-DISOLD(:,ST));
            else
                DISERR=norm(DISPLA(:,TT)-DISOLD(:,ST))/norm(DISPLA(:,TT));
            end
            if PSFERR<=tol && DISERR<=tol
                CONVTI=CONVTI+1;
                disp('    ')
                disp('Convergence achieved for the time step number ... ')
                disp(CONVTI)
            else
                PSFOLD(:,OLDCOT-FPTISE+2:NSAMPL)=PSEUFO(:,OLDCOT+1:LPTISE);
                DISOLD(:,OLDCOT-FPTISE+2:NSAMPL)=DISPLA(:,OLDCOT+1:LPTISE);
                convlock=0;
                break
            end
        end
        disp('    ')
        if convlock==0
            NOTCON=CONVTI+1;
            disp('New iteration required. Not convergence achieved for the time step number ...
')
```

```
            disp(NOTCON)
        elseif ITIMSE~=NTIMSE
            disp('Convergence achieved for all the time steps within the current segment. Next
segment will be evaluated ... ')
        else
            disp('Finish: Convergence achieved for all the time steps of the interest span  ')
        end

    end

end
```

Section 17: Output results post processing

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The output results are postprocessed for each DOF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pause(1)
disp('   ')
disp('Starting output results post processing...  ')
t=LODTIM(1:NRESIT);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       Creation of plots tables in excel
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pause(1)
% Calculation the relative displacement and velocity
DISREL=zeros(NDOF+1,NRESIT);
VELREL=zeros(NDOF+1,NRESIT);
DISREL(1,:)=DISPLA(1,:);
DISREL(NDOF+1,:)=-DISPLA(NDOF,:);
VELREL(1,:)=VELOCI(1,:);
VELREL(NDOF+1,:)=-VELOCI(NDOF,:);
if NDOF>=2
    for IDOF=2:NDOF
        DISREL(IDOF)=DISPLA(IDOF)-DISPLA(IDOF-1);
        VELREL(IDOF)=VELOCI(IDOF)-VELOCI(IDOF-1);
    end
end
% Tables and plot generation
nullrow1=zeros(1,NRESIT);
nullrow2=zeros(1,MEXCIT);
for IDOF=1:NDOF+1
    disp('  ')
    disp('Generating tables and figures of the response of degree of freedom number...')
    disp(IDOF)
    pause(1)
    if IDOF<NDOF+1

f15_excel_tables_hftd_v2(IDOF,name,DISPLA(IDOF,:),VELOCI(IDOF,:),ACCELE(IDOF,:),PSEUFO(IDOF,1:NRE
SIT),FNLSTI(IDOF,:),FNLDAM(IDOF,:),FNONLI(IDOF,:),FLINEA(IDOF,:),KEY(IDOF,:),SPRFOR(IDOF,:),DASHF
```

```
O(IDOF,:),t,UT(IDOF,:),UC(IDOF,:),UO(IDOF,:),EXCITA,DISFRE(IDOF,1:MEXCIT),VELFRE(IDOF,1:MEXCIT),A
CCFRE(IDOF,1:MEXCIT));

f16_plot_resp_hftd_v2(IDOF,name,DISPLA(IDOF,:),VELOCI(IDOF,:),ACCELE(IDOF,:),SPRFOR(IDOF,:),DASHF
O(IDOF,:),DISREL(IDOF,:),VELREL(IDOF,:),t,EXCITA,DISFRE(IDOF,1:MEXCIT),VELFRE(IDOF,1:MEXCIT),ACCF
RE(IDOF,1:MEXCIT),FOUACC(1:MEXCIT));
    else

f15_excel_tables_hftd_v2(IDOF,name,nullrow1,nullrow1,nullrow1,nullrow1,nullrow1,nullrow1,nullrow1
,nullrow1,KEY(IDOF,:),SPRFOR(IDOF,:),DASHFO(IDOF,:),t,UT(IDOF,:),UC(IDOF,:),UO(IDOF,:),EXCITA,nul
lrow2,nullrow2,nullrow2);

f16_plot_resp_hftd_v2(IDOF,name,nullrow1,nullrow1,nullrow1,SPRFOR(IDOF,:),DASHFO(IDOF,:),DISREL(I
DOF,:),VELREL(IDOF,:),t,EXCITA,nullrow2,nullrow2,nullrow2,nullrow2);
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       Change the name of variables to be saved
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('  ')
disp('Changing the name of the selected variables to be saved...')
%   The new names of the variables are created and stored using the
%   function "f17_new_name_var"
[vk,vc,vm,vtol,vg,vphi,vnatfreq,vt1,vw1,vxi1,vsignal,vdt,vnsampl,vntimse,vmexcit,vdeltaf,vexcita,
vlopass,vhipass,vnresit,vdispla,vveloci,vaccele,vpseufo,vnlstiff,vnldamp,vfnonli,vflinea,vkey,vsp
rinfor,vdashfor,vdisfre,vvelfre,vaccfre,vt,vut,vuc,vdisrel,vvelrel,vfouacc]=f17_new_name_var_hftd
_v2(name);
%   The new names are assigned to all the variables to be saved.
eval([vk ' = K;']);
eval([vc ' = C;']);
eval([vm ' = M;']);
eval([vtol ' = tol;']);
eval([vg ' = g;']);
eval([vphi ' = Phi;']);
eval([vnatfreq ' = Natfreq;']);
eval([vt1 ' = T1;']);
eval([vw1 ' = w1;']);
eval([vxi1 ' = xi1;']);
eval([vsignal ' = signal;']);
eval([vdt ' = DT;']);
eval([vnsampl ' = NSAMPL;']);
eval([vntimse ' = NTIMSE;']);
eval([vmexcit ' = MEXCIT;']);
eval([vdeltaf ' = DELTAF;']);
eval([vexcita ' = EXCITA;']);
eval([vlopass ' = LOPASS;']);
eval([vhipass ' = HIPASS;']);
eval([vnresit ' = NRESIT;']);
eval([vdispla ' = DISPLA;']);
eval([vveloci ' = VELOCI;']);
```

```
eval([vaccele ' = ACCELE;']);
eval([vpseufo ' = PSEUFO;']);
eval([vnlstiff ' = FNLSTI;']);
eval([vnldamp ' = FNLDAM;']);
eval([vfnonli ' = FNONLI;']);
eval([vflinea ' = FLINEA;']);
eval([vkey ' = KEY;']);
eval([vsprinfor ' = SPRFOR;']);
eval([vdashfor ' = DASHFO;']);
eval([vdisfre ' = DISFRE;']);
eval([vvelfre ' = VELFRE;']);
eval([vaccfre ' = ACCFRE;']);
eval([vt ' = t;']);
eval([vut ' = UT;']);
eval([vuc ' = UC;']);
eval([vdisrel ' = DISREL;']);
eval([vvelrel ' = VELREL;']);
eval([vfouacc ' = FOUACC;']);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The selected variables are saved
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pause(1)
disp('  ')
disp('Saving selected set of variables...')
%File path where the data is store. The file path can be changed to another
%desired location.
path='C:\Users\slfx\Documents\Fernando\Graduation project\Master thesis\Analysis files\Matlab
models\HFTD_freq dependence\HFTD Output results\Variables\';
%path='C:\Users\Fernando Sirumbal\Documents\Documents FERNANDO\TU Delft\Academic\MSc
thesis\Master thesis\Analysis files\Matlab models\HFTD_freq dependence\HFTD Output
results\Variables\';
%Name of the matlab data files
filename1=[path 'results_' name]; %Name of the matlab file containing the results variables
filename2=[path 'data_' name]; %Name of the matlab file conatining the data variables
%Saving the variables in both files
save(filename1,vdispla,vveloci,vaccele,vpseufo,vnlstiff,vnldamp,vfnonli,vflinea,vkey,vsprinfor,vd
ashfor,vdisfre,vvelfre,vaccfre,vt,vfouacc,vut,vuc,vdisrel,vvelrel);
save(filename2,vk,vc,vm,vtol,vg,vphi,vnatfreq,vt1,vw1,vxi1,vsignal,vdt,vnsampl,vntimse,vmexcit,vd
eltaf,vexcita,vlopass,vhipass,vnresit);
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       Delete the variables from the workspace
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('  ')
disp('Deleting all variables from the workspace...')
clear;
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       End of the program
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
disp('  ')
disp('End of the HFTD program... Check the results files saved in the specified folder')
```

Published with MATLAB® R2012b

B.1.2. HFTD selected functions

```
function [DCYFAC]=f5_decay_func(INPFAC,LPTISE,NDCAYF,DT,TDCAYF)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   HFTD for simplified MDOF systems
%       L. Fernando Sirumbal Z.
%       April, 2013
%
%   Function to determine the factors of the decaying function appended to
%   the ground acceleration and pseudo force.
%
%   inputs
%
%       INPFAC  Vector containing the input (ground acceleration or pseudo
%               force)factors for each time step.
%       LPTISE  Index indicating the last time point of the current
%               segment.
%       NDCAYF  Number of time steps defining the length of the decaying
%               function.
%       DT      Time step size.
%       TDCAYF  Decaying function length.
%
%   outputs
%
%       DCYFAC  Decaying function factors for the input (ground
%               acceleration or pseudo force). Different options for
%               decaying function expressions can be used by uncomment the
%               chosen one.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
f0=INPFAC(LPTISE);
DCYFAC=zeros(1,NDCAYF);
for k=1:NDCAYF
DCYFAC(k)=f0*(1-k*DT/TDCAYF);
%DCYFAC(k)=f0*(1-k*DT/TDCAYF+sin(2*pi*k*DT/TDCAYF)/(2*pi));
end
```

Published with MATLAB® R2012b

```
function [FOUREA,FOUIMA]=f6_fast_fourier_transf(x,N,MEXCIT)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   HFTD for simplified MDOF systems
%       L. Fernando Sirumbal Z.
%       April, 2013
%
%   Function to apply the direct discrete Fourier transform to a time
%   dependent variable
%
%   inputs
%
%       x       Time dependent variable (signal) to which direct Fourier
%               tranform is applied.
%       N       Number of time points contained in x.
%       MEXCIT  Maximum number of frequency samples to be analyzed.
%
%
%   outputs
%
%       FOUREA  Real part of the direct Fourier transform of signal x.
%       FOUIMA  Imaginary part of the direct Fourier transform of signal x.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% FFT computation
y = fft(x,N);
rey=real(y);
imy=imag(y);
% Scale FFT output
reh=rey/N;
imh=imy/N;
% Copy only the frequency responses that are going to be used in the
% frequency domain analysis. The conjugates are not required to be copied
% since they do not provide additional information.
FOUREA(1,:)=reh(1:MEXCIT);
FOUIMA(1,:)=imh(1:MEXCIT);
```

Published with MATLAB® R2012b

```
function
[DIFREA,DIFRIM]=f7_solver_pseudo_lin_direct_freq_v2(K,C,Cfd,Chy,M,w,NDOF,FOUACR,FOUACI,FOUPFR,FOU
PFI)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   HFTD for simplified MDOF systems
%       L. Fernando Sirumbal Z.
%       April, 2013
%
%   Function to solve a linear system in the frequency domain using the
%   direct method
%
%              [-(w^2)*M + i*w*C + K] u(w) = -M*I*ug(w) + q(w)
%
%   inputs
%
%       K       Linear elastic stiffness matrix of the system.
%       C       Linear elastic constant damping matrix of the system.
%       Cfd     Linear elastic frequency dependent damping matrix of the
%               system.
%       Chy     Linear elastic hysteretic damping matrix of the system.
%       M       Mass matrix of the system.
%       w       Excitation frequency.
%       NDOF    Number of degrees of freedom.
%       FOUACR  Fourier transform of the ground acceleration (real part)
%               corresponding to the excitation frequency w
%       FOUACI  Fourier transform of the ground acceleration (imaginary part)
%               corresponding to the excitation frequency w
%       FOUPFR  Fourier transform of the pseudo force vector (real part)
%               corresponding to the excitation frequency w
%       FOUPFI  Fourier transform of the pseudo force vector (imaginary part)
%               corresponding to the excitation frequency w
%
%   outputs
%
%       DIFREA  Displacement vector in the frequency domain (real part)
%               obtained from the solution of the linear system of
%               equations.
%
%       DIFRIM  Displacement vector in the frequency domain (imaginary
%               part) obtained from the solution of the linear system of
%               equations.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Components of the modified stiffness matrix in the frequency domain
A=K-(w^2)*M;
B=w*(C+Cfd+Chy);
% Assembling the modified stiffness matrix
```

```
KMODF=[A B;-B A];
% Rigig body motion vector
l=ones(NDOF,1);
% Real part of the modified force vector
FMODRE=(-FOUACR*M*l)+FOUPFR;
% Imaginary part of the modified force vector
FMODIM=(-FOUACI*M*l)+FOUPFI;
% Assembling the modified force vector
FORMDF=[FMODIM;FMODRE];
% Solution of the system of equations
SOL=KMODF\FORMDF;
% Spliting the solution between real and imaginary terms
DIFRIM=SOL(1:NDOF);
DIFREA=SOL(NDOF+1:2*NDOF);
```

Published with MATLAB® R2012b

```
function [IFOURE,IFOUIM]=f8_inv_fast_fourier_transf(x,N,FPTISE,LPTISE)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   HFTD for simplified MDOF systems
%       L. Fernando Sirumbal Z.
%       April, 2013
%
%   Function to apply the inverse discrete Fourier transform to a time
%   dependent variable
%
%   inputs
%
%       x       Frequency dependent complex variable (spectrum) to which inverse Fourier
%               tranform is applied.
%       N       Number of frequency points contained in x.
%       FPTISE  First time point of the signal that is copied as output
%       LPTISE  Last time point of the signal that is copied as output
%
%   outputs
%
%       IFOURE  Real part of the inverse Fourier transform of spectrum x.
%       IFOUIM  Imaginary part of the inverse Fourier transform of spectrum x.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% IFFT computation
y = ifft(N*x,N,'symmetric');
rey=real(y);
imy=imag(y);
% Copy only the results of the time steps within the current time segment
IFOURE=rey(FPTISE:LPTISE);
IFOUIM=imy(FPTISE:LPTISE);
```

Published with MATLAB® R2012b

```
function FLINEA=f14_linear_int_rest_force(K,DISPLA,C,VELOCI)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   HFTD for simplified MDOF systems
%       L. Fernando Sirumbal Z.
%       April, 2013
%
%   Function to calculate the internal linear restoring force
%
%
%   inputs
%
%       K       Stiffness matrix of the system
%       DISPLA  Displacement vector
%       C       Damping matrix of the system
%       VELOCI  Velocity vector
%
%
%   outputs
%
%       FLINEA  Vector containing the linear restoring force
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
FLINEA=K*DISPLA+C*VELOCI;
```

Published with MATLAB® R2012b

B.2. Newmark method code

B.2.1. Main Newmark script

```
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Newmark method for simplified MDOF systems
%       L. Fernando Sirumbal Z.
%       April, 2013
%
%   This script execute the Newmark method for linear or nonlinear simplfied
%   dynamic models of one or more degrees of freedom (DOF's) subjected to
%   base (ground) acceleration.
%
%   After defining the system properties and the load signal, the following
%   results are obtained: time history displacement, velocity, acceleration
%   and nonlinear internal forces of the system.
%
%   The name of the function indicating the type of nonlinearity to be used
%   in the problem is indicated in Section 11. The functions describing
%   type of tangential stiffnes and damping matrix should be selected.
%   Besides, the non-linear stiffness and damping models are defined inside
%   the function "f18_Newt_Raph_modf_v6".
%
%   The output results are saved as excel tables, figures and matlab data
%   files. The file path where these output results files are saved should
%   be indicated Section 12 of this script and in the function files
%   "f19_excel_tables_newm" and "f20_plot_resp_newm".
%
%
```

Section 1: File name

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      All the output results of the program will be saved in the folder
%      "Output results" located in the work folder. The name specified by
%      the user is used to save the output files.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
clear;
disp('========================================================================')
disp('   Problem data                  ')
disp('========================================================================')
name=input('\nEnter the name of the file (this name will be used to save the output data)...
','s');
disp('   ')
pause(2);
```

Section 2: Newmark method parameters

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The Newmark parameters gamma and beta should be selected. Average
```

```
%       and linear acceleration cases are the two default options.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
disp('=========================================================================')
disp('   Define the Newmark method to be executed                 ')
disp('=========================================================================')
lock=0;
while lock==0
    pause(2);
    opt=input('\nSelect a Newmark special case or the numerical values of the parameters gamma
and beta. \nEnter "AA" for Average Acceleration or enter "LA" for Linear Acceleration, \nor "OC"
for other case .... ','s');
    h=strcmp(opt,{'AA','LA','OC'});
    if h(1)==1
        gama=0.5;
        beta=0.25;
        lock=1;
    elseif h(2)==1
        gama=0.5;
        beta=1/6;
        lock=1;
    elseif h(3)==1
        pause(1);
        gama=input('\nEnter the value of gamma) .... ');
        if gama<0 || gama>1
            disp('   ')
            disp('Error message: Not valid specified value for gamma')
        else
            pause(1);
            beta=input('\nEnter the value of beta) .... ');
            if beta<0 || beta>1
            disp('   ')
            disp('Error message: Not valid specified value for beta')
            else
            lock=1;
            end
        end
    else
        pause(2);
        disp('   ')
        disp('Error message: Option not correctly chosen. Choose one of the three available
options')
    end
end
```

Section 3: Exercise data definition

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Input the mass, damping and stiffness matrices of the simplified MDOF
```

```
%   system. The gravity constant and the tolerance and maximum number of
%   iterations for non-linear analysis are also input by the user.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
disp('==========================================================================')
disp('   Define the elastic stiffness, elastic damping and mass matrices                  ')
disp('==========================================================================')
pause(1);
lock=0;
while lock==0
    MCOF=input('\nEnter the masses of each degree of freedom m_i \n(in a row vector enter one
mass per DOF).... ');
    sizem=size(MCOF);
    if sizem(1)~=1
        disp(' ')
        disp('Error message: The masses should be arranged in a row vector')
        pause(2);
    else
        lock=1;
    end
end
NDOF=sizem(2);
pause(1);
disp(' ')
disp('The number of degrees of freedom (NDOF) of the system is defined to be ')
NDOF
pause(1)
disp(' ')
disp('The masses of each degree of freedom are defined to be ')
MCOF
pause(1);
lock=0;
while lock==0
    KELAST=input('\nEnter the linear elastic stiffness coefficients k_i of the springs \n(in a
row vector enter NDOF+1 stiffness coefficients) .... ');
    sizek=size(KELAST);
    if sizek(1)~=1
        disp('         ')
        disp('Error message: The spring coefficients should be arranged in a row vector')
        pause(2);
    elseif sizek(2)~=NDOF+1
        disp(' ')
        disp('Error message: The row vectors defining the coefficientes of K should have NDOF+1
components ')
        pause(2);
    else
        lock=1;
    end
end
disp(' ')
```

```
disp('The linear elastic stiffness coefficients are defined to be')
KELAST
pause(1);
lock=0;
while lock==0
    CELAST=input('\nEnter the linear elastic damping coefficients c_i of the dashpots \n(in a row
vector enter NDOF+1 damping coefficients)... ');
    sizec=size(CELAST);
    if sizec(1)~=1
        disp(' ')
        disp('Error message: The dashpots coefficients should be arranged in a row vector')
        pause(2);
    elseif sizec~=NDOF+1
        disp(' ')
        disp('Error message: The row vectors defining the coefficientes of C should have NDOF+1
components ')
        pause(2);
    else
        lock=1;
    end
end
disp(' ')
disp('The linear elastic damping coefficients are defined to be ')
CELAST
pause(1);
lock=0;
while lock==0
    g=input('\nEnter the value of the gravity constant g .... ');
    sizeg=size(g);
    if sizeg(1)~= 1 || sizeg(2)~=1
        disp(' ')
        disp('Error message: g must be a scalar value')
        pause(2);
    elseif g<=0
        disp(' ')
        disp('Error message: g must be defined as a positive value')
        pause(2);
    else
        lock=1;
    end
end
disp(' ')
disp('The gravity constant is defined to be ')
g
disp('   ')
pause(1);
lock=0;
while lock==0
    tol=input('\nEnter the convergence tolerance tol for noninear analysis.... ');
    sizet=size(tol);
```

```
    if sizet(1)~= 1 || sizet(2)~=1
        disp(' ')
        disp('Error message: tol must be a scalar value')
        pause(2);
    elseif tol>0.01 || tol<=0
        disp(' ')
        disp('Error message: tol must be positive and smaller than 10^-2')
        pause(2);
    else
        lock=1;
    end
end
disp(' ')
disp('The tolerance for convergence check is defined to be ')
tol
pause(1);
lock=0;
while lock==0
    maxite=input('\nEnter the maximum number of iterations for convergence check in nonlinear
analysis .... ');
    sizemax=size(maxite);
    if sizemax(1)~= 1 || sizemax(2)~=1
        disp(' ')
        disp('Error message: maxite must be a scalar value')
        pause(2);
    elseif mod(maxite,1)~=0 || maxite<=0
        disp(' ')
        disp('Error message: maxite must be positive and integer')
        pause(2);
    else
        lock=1;
    end
end
disp(' ')
disp('The maximum number of iterations for convergence check is defined to be ')
maxite
pause(2);
```

Section 4: Construction of the linear elastic matrices of the system

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Based on the coefficients values defined in the previous section for
%   each mass, spring and dashpot, the linear elastic matrices of the
%   system are constructed using the function file "f1_linelast_mat".
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
disp('=========================================================================')
disp('   Linear elastic matrices                  ')
disp('=========================================================================')
```

```
%Mass matrix
M=zeros(NDOF);
for IDOF=1:NDOF
    M(IDOF,IDOF)=MCOF(IDOF);
end
l=ones(NDOF,1);
P=-M*l;
disp(' ')
disp('The mass matrix M is')
M
pause(1);
%Stiffness matrix
K=f1_linelast_mat(NDOF,KELAST);
disp(' ')
disp('The linear elastic stiffness matrix K is')
K
pause(1);
%Damping matrix
C=f1_linelast_mat(NDOF,CELAST);
disp(' ')
disp('The linear elastic damping matrix C is')
C
pause(2);
```

Section 5: Free vibration Eigen-analysis

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Eigen-analysis of the simplified system. Mode shape vectors,
%   eigen-frequencies, damping ratio and generalized matrices of the system
%   are calculated.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
disp('========================================================================')
disp('   Free vibration natural frequencies and mode shape vectors                  ')
disp('========================================================================')
[Phi,Natfreq]=eig(K,M);
Mgen=transpose(Phi)*M*Phi;
Kgen=transpose(Phi)*K*Phi;
Cgen=transpose(Phi)*C*Phi;
w1=sqrt(Natfreq(1,1));
T1=2*pi/w1;
f1=1/T1;
xi1=(1/(2*w1))*(Cgen(1,1)/Mgen(1,1));
pause(2);
disp('   ')
disp('The eigenvectors matrix is ')
disp('   ')
Phi
pause(1);
```

```
disp('   ')
disp('The eigenvalues (squared natural circular frequencies) matrix is')
disp('   ')
Natfreq
pause(1);
disp('   ')
disp('The fundamental period of the system is defined to be')
disp('   ')
T1
pause(1);
disp('   ')
disp('The fundamental circular frequency of the system is defined to be')
disp('   ')
w1
pause(1);
disp('   ')
disp('The damping ratio of the fundamental mode of vibration of the system is defined to be')
disp('   ')
xi1
disp('   ')
pause(2);
```

Section 6: Ground acceleration signal information

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The data of the ground acceleration signal must be stored as a
%       matlab variable called "signal". This variable should be
%       defined as a matrix with two columns and as many rows as time
%       points available in the signal data. The first column contains the
%       time steps values, which should start in time = 0 seconds, and
%       should be equally spaced (constant time step size). The second
%       colum contains the ground acceleration values normalized with the
%       gravity acceleration g.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
disp('=========================================================================')
disp('   Ground acceleration signal information                  ')
disp('=========================================================================')
disp('   ')
disp('Loading the ground acceleration variable "signal" from file signal.mat saved in the work
folder...')
disp('   ')
pause(2);
load signal;
lock=1;
while lock~=0
    sizesig=size(signal);
    [SIGTIM,NFACT,SIGNDT,lock,k]=f3_ground_accel_signal_check(signal,sizesig);% Function that
checks the correct definition of the signal variable
```

```
    if lock==1
        disp('Error message: The first time step must be zero')
        return
    elseif lock==2
        disp('Error message: Signal data variable must have two columns')
        return
    elseif lock==3
        disp('Error message: Time step size of signal data is not constant')
        disp('   ')
        pause(2);
        fprintf('The problem was found between the time points %d and %d\n',k,k+1)
        return
    end
end
SIGFAC=g*signal(:,2);
TACSIG=NFACT*SIGNDT;
pause(2);
disp('   ')
disp('The number of time points contained in the ground acceleration signal is ')
disp('   ')
NFACT
pause(1);
disp('   ')
disp('The time step size of the ground acceleration signal is ')
disp('   ')
SIGNDT
pause(1);
disp('   ')
disp('The duration of the ground acceleration signal is ')
disp('   ')
TACSIG
pause(1);
disp('   ')
disp('The maximum ground acceleration magnitude contained in the signal is ')
disp('   ')
disp(max(abs(SIGFAC)))
disp('   ')
pause(2);
```

Section 7: Time step size

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The time step size (DT) is calculated base on the signal
%       discretization.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
disp('==========================================================================')
disp('   Time step size              ')
disp('==========================================================================')
```

```
lock=0;
while lock==0
    pause(2);
    opt=input('\nSelect the option for the definition of the HFTD time step size (DT). Enter "0" for DT to be equal to the signal step \nsize (SIGNDT) or enter "1" for DT to be a multiple of SIGNDT .... ');
    if opt==0
        TIMES=1;
        lock=1;
    elseif opt==1
        pause(2);
        TIMES=input('\nEnter the number of times that SIGNDT should be multiplied to obtain (DT) .... ');
        if mod(TIMES,1)~=0 || TIMES<=0
            disp('    ')
            disp('Error message: The input value should be a positive integer')
        else
            lock=1;
        end
    else
        pause(2);
        disp('    ')
        disp('Error message: The input value must be 0 or 1')
    end
end
DT=TIMES*SIGNDT;
pause(1);
disp('    ')
disp('The Newmark time step size (DT) is defined to be')
disp('    ')
DT
pause(2)
```

Section 8: Time span of interest

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       Time span for which the reponses are calculated.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('==========================================================================')
disp('   Time span of interest                ')
disp('==========================================================================')
lock=0;
while lock==0
    pause(1);
    TRESP=input('\nEnter the time span of interest in seconds .... ');
    NFRESP=ceil(TRESP/DT);
    TRESP=DT*NFRESP;
    if TRESP<=0
        disp('    ')
```

```
        disp('Error message: The time span of interest should be positive')
    else
        lock=1;
    end
end
pause(1);
disp('    ')
disp('The number of time steps defining the time span of interest length (NFRESP) is defined to
be')
disp('    ')
NFRESP
pause(1);
disp('    ')
disp('The time span of interest for determining the response (TRESP) is defined to be')
disp('    ')
TRESP
disp('    ')
pause(2);
```

Section 9: Ground acceleration signal modification

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The ground acceleration data is cutted or extended adding zeros
%       until the response time span is covered.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
LODTIM=zeros(1,NFRESP);
LODFAC=zeros(1,NFRESP);
for k=1:NFRESP
    LODTIM(k)=(k-1)*DT;
    j=1+(k-1)*TIMES;
    if j<=NFACT
        LODFAC(k)=SIGFAC(j);
    else
        LODFAC(k)=0;
    end
end
```

Section 10: Initialization of variables

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    Initialize response and internal restoring force variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DISPLA=zeros(NDOF,NFRESP);
VELOCI=zeros(NDOF,NFRESP);
ACCELE=zeros(NDOF,NFRESP);
FNLSTI=zeros(NDOF,NFRESP);
SPRFOR=zeros(NDOF+1,NFRESP);
FNLDAM=zeros(NDOF,NFRESP);
DASHFO=zeros(NDOF+1,NFRESP);
```

```
KEYSTI=zeros(NDOF+1,NFRESP);
KEYDAM=zeros(NDOF+1,NFRESP);
UT=zeros(NDOF+1,NFRESP);
UC=zeros(NDOF+1,NFRESP);
UO=zeros(NDOF+1,NFRESP);
NITERA=zeros(1,NFRESP);
```

Section 11: Newmark method execution for each time step

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The Newmark calculation process is executed step by step. A time
%       step loop is defined in order to repeat the process for the
%       NFRESP time steps previously determined.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('=========================================================================')
disp('   Starting Newmark method execution for each time step              ')
disp('=========================================================================')
ACCELE(:,1)=M\(LODFAC(1)*P-FNLDAM(:,1)-FNLSTI(:,1)); % Initial acceleration
for ITIME=1:NFRESP-1
    VARLOD=(LODFAC(ITIME+1)-LODFAC(ITIME))*P; %Load increment
    %
    %Tangent stiffness matrix
    %STIMAT=f15_tang_stiff_mat_elastoplast_hyster(NDOF,KEYSTI(:,ITIME),KELAST);

STIMAT=f16_tang_stiff_mat_cubic_hyster_v2(NDOF,KEYSTI(:,ITIME),KELAST,DISPLA(:,ITIME),UO(:,ITIME)
);
    %STIMAT=K;
    %
    %Tangent damping matrix
    %DAMMAT=C
    DAMMAT=f17_tang_damp_mat_nonlimet(NDOF,KEYDAM(:,ITIME),CELAST,DISPLA(:,ITIME));
    %
    EFFSTI=STIMAT+((gama/beta/DT)*DAMMAT)+((1/beta/(DT^2))*M); % Effective stiffness matrix

EFVALO=VARLOD+(((1/beta/DT)*M+(gama/beta)*DAMMAT)*VELOCI(:,ITIME))+(((1/2/beta)*M+((gama/2/beta)-
1)*DT*DAMMAT)*ACCELE(:,ITIME)); %Effective load increment

[DISPLA(:,ITIME+1),VELOCI(:,ITIME+1),ACCELE(:,ITIME+1),FNLSTI(:,ITIME+1),FNLDAM(:,ITIME+1),NITERA
(ITIME),KEYSTI(:,ITIME+1),SPRFOR(:,ITIME+1),DASHFO(:,ITIME+1),UT(:,ITIME+1),UC(:,ITIME+1),UO(:,IT
IME+1),EMKEY]=f18_Newt_Raph_modf_v6(NDOF,DISPLA(:,ITIME),VELOCI(:,ITIME),ACCELE(:,ITIME),FNLSTI(:
,ITIME),FNLDAM(:,ITIME),EFVALO,EFFSTI,M,tol,KEYSTI(:,ITIME),SPRFOR(:,ITIME),DASHFO(:,ITIME),UT(:,
ITIME),UC(:,ITIME),UO(:,ITIME),ITIME,maxite,gama,beta,DT,LODFAC(ITIME+1),P,VARLOD,KELAST,CELAST);
    if EMKEY~=100
        return
    end
    if NITERA(ITIME)>maxite
        return
    else
        fprintf('\nConvergence achieved for time step number %d',ITIME)
```

```
        disp('   ')
    end
end
pause(2)
```

Section 12: Output results post processing

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The output results are postprocessed for each DOF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('   ')
disp('Starting output results post processing...  ')
t=LODTIM;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       Creation of plots and tables in excel
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculation of the relative displacement and velocity
DISREL=zeros(NDOF+1,NFRESP);
VELREL=zeros(NDOF+1,NFRESP);
DISREL(1,:)=DISPLA(1,:);
DISREL(NDOF+1,:)=-DISPLA(NDOF,:);
VELREL(1,:)=VELOCI(1,:);
VELREL(NDOF+1,:)=-VELOCI(NDOF,:);
if NDOF>=2
    for IDOF=2:NDOF
        DISREL(IDOF)=DISPLA(IDOF)-DISPLA(IDOF-1);
        VELREL(IDOF)=VELOCI(IDOF)-VELOCI(IDOF-1);
    end
end
% Tables and plots generation
nullrow=zeros(1,NFRESP);
for IDOF=1:NDOF+1
    disp('  ')
    disp('Generating tables and figures of the response of degree of freedom number...')
    disp(IDOF)
    pause(1)
    if IDOF<NDOF+1

f19_excel_tables_newm(IDOF,name,DISPLA(IDOF,:),VELOCI(IDOF,:),ACCELE(IDOF,:),FNLSTI(IDOF,:),FNLDA
M(IDOF,:),KEYSTI(IDOF,:),SPRFOR(IDOF,:),DASHFO(IDOF,:),t,UT(IDOF,:),UC(IDOF,:),UO(IDOF,:))

f20_plot_resp_newm(IDOF,name,DISPLA(IDOF,:),VELOCI(IDOF,:),ACCELE(IDOF,:),SPRFOR(IDOF,:),DASHFO(I
DOF,:),DISREL(IDOF,:),VELREL(IDOF,:),t)
    else

f19_excel_tables_newm(IDOF,name,nullrow,nullrow,nullrow,nullrow,nullrow,KEYSTI(IDOF,:),SPRFOR(IDO
F,:),DASHFO(IDOF,:),t,UT(IDOF,:),UC(IDOF,:),UO(IDOF,:))

f20_plot_resp_newm(IDOF,name,nullrow,nullrow,nullrow,SPRFOR(IDOF,:),DASHFO(IDOF,:),DISREL(IDOF,:)
```

```
,VELREL(IDOF,:),t)
    end
end
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       Changing the name of variables to be saved
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('  ')
disp('Changing the name of the selected variables to be saved...')
%   The new names of the variables are created and stored using the
%   function f21_new_name_var_newm
[vk,vc,vm,vtol,vg,vphi,vnatfreq,vt1,vw1,vxi1,vsignal,vdt,vdispla,vveloci,vaccele,vnlstiff,vnldamp
,vkeystiff,vkeydamp,vsprinfor,vdashfor,vt,vnitera,vut,vuc,vdisrel,vvelrel]=new_name_var_newm(name
);
%   The new names are assigned to all the variables to be saved.
eval([vk ' = K;']);
eval([vc ' = C;']);
eval([vm ' = M;']);
eval([vtol ' = tol;']);
eval([vg ' = g;']);
eval([vphi ' = Phi;']);
eval([vnatfreq ' = Natfreq;']);
eval([vt1 ' = T1;']);
eval([vw1 ' = w1;']);
eval([vxi1 ' = xi1;']);
eval([vsignal ' = signal;']);
eval([vdt ' = DT;']);
eval([vdispla ' = DISPLA;']);
eval([vveloci ' = VELOCI;']);
eval([vaccele ' = ACCELE;']);
eval([vnlstiff ' = FNLSTI;']);
eval([vnldamp ' = FNLDAM;']);
eval([vkeystiff ' = KEYSTI;']);
eval([vkeydamp ' = KEYDAM;']);
eval([vsprinfor ' = SPRFOR;']);
eval([vdashfor ' = DASHFO;']);
eval([vt ' = t;']);
eval([vnitera ' = NITERA;']);
eval([vut ' = UT;']);
eval([vuc ' = UC;']);
eval([vdisrel ' = DISREL;']);
eval([vvelrel ' = VELREL;']);
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       The selected variables are saved
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pause(1)
disp('  ')
disp('Saving selected set of variables...')
%File path where the data is store. The file path can be changed to another
```

```
%desired location.
path='C:\Users\slfx\Documents\Fernando\Graduation project\Master thesis\Analysis files\Matlab
models\Newmark_nonli_MDOF\Newm Output results\';
%path='C:\Users\Fernando Sirumbal\Documents\Documents FERNANDO\TU Delft\Academic\MSc
thesis\Master thesis\Analysis files\Matlab models\Newmark_nonli_MDOF\Newm Output results\';
%Name of the matlab data files
filename1=[path 'results_' name]; %Name of the matlab file containing the results variables
filename2=[path 'data_' name]; %Name of the matlab file conatining the data variables
%Saving the variables in both files
save(filename1,vdispla,vveloci,vaccele,vkeystiff,vkeydamp,vsprinfor,vdashfor,vt,vnlstiff,vnldamp,
vnitera,vut,vuc,vdisrel,vvelrel);
save(filename2,vk,vc,vm,vtol,vg,vphi,vnatfreq,vt1,vw1,vxi1,vsignal,vdt);
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       Deleting all the variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pause(1)
disp('  ')
disp('Deleting all variables from the workspace...')
clear;
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       End of the program
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pause(1)
disp('  ')
disp('End of Newmark program... Check the results files saved in the specified folder')
```

Published with MATLAB® R2012b

B.2.2. Newmark selected functions

```
function STIMAT=f15_tang_stiff_mat_elastoplast_hyster(NDOF,KEY,KELAST)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     Newmark method for simplified MDOF systems
%       L. Fernando Sirumbal Z.
%       April, 2013
%
%   Function to calculate the nonlinear tangential stiffness matrix of the
%   current time step for the linear elastoplastic model
%
%   inputs
%
%       NDOF        Number of degrees of freedom of the system
%       KEY         Vector specifying the nonlinear state of each degree of
%                   freedom in the previous time step
%                   KEY=0   Elastic behavior
%                   KEY=1   Plastic behavior in tension
%                   KEY=-1  Plastic behavior in compression
%       KELAST      Vector containing the elastic stiffness coefficients of
%                   the springs
%
%
%   outputs
%
%       STIMAT      Tangential stiffness matrix of the current time step
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
F=zeros(NDOF+1,NDOF);
STIMAT=zeros(NDOF);
if KEY(1)==0
    F(1,1)=KELAST(1);
end
if KEY(NDOF+1)==0
    F(NDOF+1,NDOF)=-KELAST(NDOF+1);
end
if NDOF>1
    for IDOF=2:NDOF
        if KEY(IDOF)==0
            F(IDOF,IDOF-1)=-KELAST(IDOF);
            F(IDOF,IDOF)=KELAST(IDOF);
        end
    end
end
```

```
for IDOF=1:NDOF
    STIMAT(IDOF,:)=F(IDOF,:)-F(IDOF+1,:);
end
```

Published with MATLAB® R2012b

```
function STIMAT=f16_tang_stiff_mat_cubic_hyster_v2(NDOF,KEY,KELAST,DISPLA,UO)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     Newmark method for simplified MDOF systems
%       L. Fernando Sirumbal Z.
%       April, 2013
%
%   Function to calculate the nonlinear tangential stiffness matrix of the
%   current time step for the cubic elastoplastic model
%
%   inputs
%
%       NDOF        Number of degrees of freedom of the system.
%       KEY         Vector specifying the nonlinear state of each degree of
%                   freedom in the previous time step.
%                   KEY=0   Elastic behavior
%                   KEY=1   Plastic behavior in tension
%                   KEY=-1  Plastic behavior in compression
%       DISPLA      Displacement vector of the previous time step
%       UO          Zero stress displacement of the previous time step
%       KELAST      Vector containing the elastic stiffness coefficients of
%                   the springs
%
%
%   outputs
%
%       STIMAT      Tangential stiffness matrix of the current time step.
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%Relative displacement
DISREL=zeros(NDOF+1,1);
DISREL(1)=DISPLA(1);
DISREL(NDOF+1)=-DISPLA(NDOF);
if NDOF>=2
    for IDOF=2:NDOF
        DISREL(IDOF)=DISPLA(IDOF)-DISPLA(IDOF-1);
    end
end
% Stiffness matrix coefficients
KTANG=8-(32/9)*((DISREL-UO).*(DISREL-UO));
F=zeros(NDOF+1,NDOF);
STIMAT=zeros(NDOF);
if KELAST(1)==0
    F(1,1)=0;
elseif KEY(1)==0 || KEY(1)==3
    F(1,1)=KTANG(1);
elseif KEY(1)==2
```

```
    F(1,1)=KELAST(1);
end
if KELAST(NDOF+1)==0
    F(NDOF+1,NDOF)=0;
elseif KEY(NDOF+1)==0 || KEY(NDOF+1)==3
    F(NDOF+1,NDOF)=-KTANG(NDOF+1);
elseif KEY(NDOF+1)==2
    F(NDOF+1,NDOF)=-KELAST(NDOF+1);
end
if NDOF>1
    for IDOF=2:NDOF
        if KELAST(IDOF)==0
            F(IDOF,IDOF-1)=0;
            F(IDOF,IDOF)=0;
        elseif KEY(IDOF)==0 || KEY(IDOF)==3
            F(IDOF,IDOF-1)=-KTANG(IDOF);
            F(IDOF,IDOF)=KTANG(IDOF);
        elseif KEY(IDOF)==2
            F(IDOF,IDOF-1)=-KELAST(IDOF);
            F(IDOF,IDOF)=KELAST(IDOF);
        end
    end
end
for IDOF=1:NDOF
    STIMAT(IDOF,:)=F(IDOF,:)-F(IDOF+1,:);
end
```

Published with MATLAB® R2012b

```
function DAMMAT=F17_tang_damp_mat_nonlimet(NDOF,KEY,CELAST,DISPLA)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     Newmark method for simplified MDOF systems
%       L. Fernando Sirumbal Z.
%       April, 2013
%
%   Function to calculate the nonlinear tangential damping matrix of the
%   current time step for the quadractic displacement dependent viscous
%   model
%
%   inputs
%
%       NDOF        Number of degrees of freedom of the system.
%       KEY         Vector specifying the nonlinear state of each degree of
%                   freedom in the previous time step.
%                   KEY=0   Elastic behavior
%                   KEY=1   Plastic behavior in tension
%                   KEY=-1  Plastic behavior in compression
%       DISPLA      Displacement vector of the previous time step
%       UO          Zero stress displacement of the previous time step
%       CELAST      Vector containing the elastic damping coefficients of
%                   the dashpots
%
%
%   outputs
%
%       DAMMAT      Tangential damping matrix of the current time step.
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%Nonlinear damping parameters
alfa=1;
a=zeros(NDOF+1,1);
for IDOF=1:NDOF+1
    if CELAST(IDOF)>0
        a(IDOF)=alfa;
    elseif CELAST(IDOF)<0
        a(IDOF)=-alfa;
    end
end
%Relative displacement
DISREL=zeros(NDOF+1,1);
DISREL(1)=DISPLA(1);
DISREL(NDOF+1)=-DISPLA(NDOF);
if NDOF>=2
    for IDOF=2:NDOF
        DISREL(IDOF)=DISPLA(IDOF)-DISPLA(IDOF-1);
```

```
    end
end
% Damping matrix coefficients
F=zeros(NDOF+1,NDOF);
DAMMAT=zeros(NDOF);
CTANG=transp(CELAST).*(1+a.*(DISREL.^2));
if KEY(1)==0
    F(1,1)=CTANG(1);
end
if KEY(NDOF+1)==0
    F(NDOF+1,NDOF)=-CTANG(NDOF+1);
end
if NDOF>1
    for IDOF=2:NDOF
        if KEY(IDOF)==0
            F(IDOF,IDOF-1)=-CTANG(IDOF);
            F(IDOF,IDOF)=CTANG(IDOF);
        end
    end
end
for IDOF=1:NDOF
    DAMMAT(IDOF,:)=F(IDOF,:)-F(IDOF+1,:);
end
```

Published with MATLAB® R2012b

```
function
[DISPLAOUT,VELOCOUT,ACCELEOUT,FNLSTIOUT,FNLDAMOUT,NITERA,KEYSTIOUT,SPRFOROUT,DASHFOUT,UTOUT,UCOUT
,UOOUT,EMKEY]=f18_Newt_Raph_modf_v6(NDOF,DISPLAIN,VELOCIN,ACCELEIN,FNLSTIN,FNLDAMIN,EFVALO,EFFSTI
,M,tol,KEYSTIN,SPRFORIN,DASHFOIN,UTIN,UCIN,UOIN,ITIME,maxite,gama,beta,DT,LODFAC,P,VARLOD,KELAST,
CELAST)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     Newmark method for simplified MDOF systems
%       L. Fernando Sirumbal Z.
%       April, 2013
%
%   Modified Newton Raphson iterative procedure to calculate the non-linear
%   response of the system. The file containing the non-linear models for
%   stiffness and damping should be selected below
%
%   inputs
%
%
%       NDOF        Number of degrees of freedom of the system.
%       DISPLAIN    Displacement vector of the previous time step
%       VELOCIN     Velocity vector of the previous time step
%       ACCELEIN    Acceleration vector of the previous time step
%       FNLSTIN     Non-linear restoring stiffness force vector of the
%                   previous time step
%       FNLDAMIN    Non-linear restoring damping force vector of the
%                   previous time step
%       EFVALO      Effective load increment of the current time step
%       EFFSTI      Effective stiffness matrix of the current time step
%       M           Mass matrix of the system
%       tol         Tolerance for convergence criteria
%       KEYSTIN     Vector specifying the stiffness nonlinear state of each
%                   degree of freedom in the previous time step.
%                   KEY=0   Elastic behavior
%                   KEY=1   Plastic behavior in tension
%                   KEY=-1  Plastic behavior in compression
%       SPRFORIN    Spring internal stiffness force of the previous time
%                   step
%       DASHFOIN    Dashpot internal damping force of the previous time
%                   step
%       UTIN        Maximum tensile displacement in the elastic range of
%                   the previous time step
%       UCIN        Maximum compression displacement in the elastic range
%                   of the previous time step
%       UOIN        Zero stress displacement of the previous time step
%       ITIME       Current time step index
%       maxite      Maximum number of iterations
%       gamma       gamma parameter of the Newmark method
%       beta        beta parameter of the Newmark method
%       DT          Time step size
```

```
%       LODFAC      Load factor of the current time step
%       P           Load vector
%       VARLOD      Load increment of the current time step
%       KELAST      Vector containing the elastic stiffness coefficients of
%                   the springs
%       CELAST      Vector containing the elastic damping coefficients of
%                   the dashpots
%
%
%   outputs
%
%       DISPLAOUT   Displacement vector of the current time step
%       VELOCOUT    Velocity vector of the current time step
%       ACCELEOUT   Acceleration vector of the current time step
%       FNLSTIOUT   Non-linear restoring stiffness force vector of the
%                   current time step
%       FNLDAMOUT   Non-linear restoring damping force vector of the
%                   current time step
%       NITERA      Number of iterations required to achieve convergence
%       KEYSTIOUT   Vector specifying the stiffness nonlinear state of each
%                   degree of freedom in the current time step.
%                   KEY=0   Elastic behavior
%                   KEY=1   Plastic behavior in tension
%                   KEY=-1  Plastic behavior in compression
%       SPRFOROUT   Spring internal stiffness force of the current time
%                   step
%       DASHFOUT    Dashpot internal damping force of the current time
%                   step
%       UTOUT       Maximum tensile displacement in the elastic range of
%                   the current time step
%       UCOUT       Maximum compression displacement in the elastic range
%                   of the current time step
%       UOOUT       Zero stress displacement of the current time step
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Initialization of variables
u=zeros(NDOF,maxite+1);
fs=zeros(NDOF,maxite+1);
fd=zeros(NDOF,maxite+1);
dr=zeros(NDOF,maxite+1);
du=zeros(NDOF,maxite+1);
u(:,1)=DISPLAIN;
fs(:,1)=FNLSTIN;
fd(:,1)=FNLDAMIN;
dr(:,1)=EFVALO;
kt=EFFSTI;
error=1;
j=0;
```

```
%Start of iterative procedure
while error>tol
    j=j+1;
    if j==maxite+1
        NITERA=j;
        fprintf('Error message: Maximum number of iterations (maxite) equal to %d has been
reached without convergence\nin the time step number %d',maxite,ITIME+1)
        return
    end
    du(:,j)=kt\dr(:,j);
    u(:,j+1)=u(:,j)+du(:,j);
    VARDIS=u(:,j+1)-DISPLAIN; %Displacement increment.
    VARVEL=((gama/beta/DT)*VARDIS)-((gama/beta)*VELOCIN)+((1-gama/2/beta)*DT*ACCELEIN); %Velocity
increment
    VELOCOUT=VELOCIN+VARVEL; %Velocity of time step i+1
    %
    % Non-linear or linear stiffness force, internal spring force and state variable of time step
i+1

%[KEYSTIOUT,SPRFOROUT,fs(:,j+1),UTOUT,UCOUT,UOOUT,EMKEY]=f9_lin_stiff_int_force(NDOF,KEYSTIN,DISP
LAIN,u(:,j+1),SPRFORIN,KELAST);

%[KEYSTIOUT,SPRFOROUT,fs(:,j+1),UTOUT,UCOUT,UOOUT,EMKEY]=f10_nonlin_hyster_stiff_int_force_v3(VEL
OCOUT,NDOF,KEYSTIN,DISPLAIN,u(:,j+1),SPRFORIN,UTIN,UCIN,UOIN,ITIME,KELAST);

[KEYSTIOUT,SPRFOROUT,fs(:,j+1),UTOUT,UCOUT,UOOUT,EMKEY]=f11_nonlin_hyster_stiff_int_force_cubic(V
ELOCOUT,NDOF,KEYSTIN,DISPLAIN,u(:,j+1),UTIN,UCIN,UOIN,ITIME,KELAST);
    if EMKEY~=100
        return
    end
    %
    %Nonlinear damping force and internal dashpot force of time step i+1
    %[fd(:,j+1),DASHFOUT]=f12_lin_damp_int_force_v2(NDOF,VARVEL,DASHFOIN,CELAST);
    [fd(:,j+1),DASHFOUT]=f13_nonlinmet_damp_int_force(NDOF,u(:,j+1),VELOCOUT,CELAST);
    %
    %Update the residual load increment for the next iteration
    if j==1
        dr(:,1)=VARLOD+(1/beta/DT)*M*VELOCIN+(1/2/beta)*M*ACCELEIN;
    end
    dr(:,j+1)=dr(:,j)-(fs(:,j+1)-fs(:,j)+fd(:,j+1)-fd(:,j)+(1/beta/(DT^2))*M*du(:,j));
    %
    %Error calculation
    error=norm(du(:,j))/norm(VARDIS);
end
%
%Calculation of the system response for the time step i+1
DISPLAOUT=u(:,j+1);
FNLSTIOUT=fs(:,j+1);
FNLDAMOUT=fd(:,j+1);
```

```
ACCELEOUT=M\(LODFAC*P-FNLDAMOUT-FNLSTIOUT);
NITERA=j;
```

Published with MATLAB® R2012b

B.3. Internal force models

B.3.1. Stiffness force functions

```
function
[KEYOUT,INTFOROUT,FNLSTIOUT,UTOUT,UCOUT,UOOUT,EMKEY]=f9_lin_stiff_int_force(NDOF,KEYIN,DISPLA1,DI
SPLA2,INTFORIN,KELAST)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     Stiffness force models
%       L. Fernando Sirumbal Z.
%       April, 2013
%
%   Function to calculate the internal stiffness force corresponding to
%   the linear elastic model
%
%   inputs
%
%       NDOF        Number of degrees of freedom of the system.
%       KEYIN       Vector specifying the nonlinear state of each degree of
%                   freedom in the previous time step.
%                   KEY=0   Elastic behavior
%                   KEY=1   Plastic behavior in tension
%                   KEY=-1  Plastic behavior in compression
%       DISPLA1     Displacement vector of the previous time step
%       DISPLA2     Displacement vector of the current time step
%       INTFORIN    Spring internal stiffness force of the previous time
%                   step
%       KELAST      Vector containing the elastic stiffness coefficients of
%                   the springs
%
%
%   outputs
%
%       KEYOUT      Vector speciying the nonlinear state of each degree of
%                   freedom in the current time step.
%                   KEY=0   Elastic behavior
%                   KEY=1   Plastic behavior in tension
%                   KEY=-1  Plastic behavior in compression
%       INTFOROUT   Spring internal stiffness force of the current time
%                   step
%       FNLSIOUT    Non-linear restoring stiffness force vector of the
%                   current time step
%
%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Calculation of relative displacements increments
DISREL1=zeros(NDOF+1,1);
DISREL2=zeros(NDOF+1,1);
DISREL1(1)=DISPLA1(1);
DISREL1(NDOF+1)=-DISPLA1(NDOF);
DISREL2(1)=DISPLA2(1);
DISREL2(NDOF+1)=-DISPLA2(NDOF);
if NDOF>=2
    for IDOF=2:NDOF
        DISREL1(IDOF)=DISPLA1(IDOF)-DISPLA1(IDOF-1);
        DISREL2(IDOF)=DISPLA2(IDOF)-DISPLA2(IDOF-1);
    end
end
VARDISREL=DISREL2-DISREL1;
%
% Defining the nonlinear state and non-linear internal spring force for
% each degree of freedom in the current time step
EMKEY=100;
KEYOUT=zeros(NDOF+1,1);
INTFOROUT=zeros(NDOF+1,1);
UTOUT=zeros(NDOF+1,1);
UCOUT=zeros(NDOF+1,1);
UOOUT=zeros(NDOF+1,1);
for IDOF=1:NDOF+1
    if KELAST(IDOF)==0
        INTFOROUT(IDOF)=0;
    elseif KEYIN(IDOF)==0
        INTFOROUT(IDOF)=INTFORIN(IDOF)+KELAST(IDOF)*VARDISREL(IDOF);
    end
end
%
% Recalculation of the nonlinear restoring force vector for the current
% time step
FNLSTIOUT=zeros(NDOF,1);
for IDOF=1:NDOF
    FNLSTIOUT(IDOF)=INTFOROUT(IDOF)-INTFOROUT(IDOF+1);
end
```

Published with MATLAB® R2012b

```
function
[KEYOUT,INTFOROUT,FNLSTIOUT,UTOUT,UCOUT,UOOUT,EMKEY]=f10_nonlin_hyster_stiff_int_force_v3(VELOCI,
NDOF,KEYIN,DISPLA1,DISPLA2,INTFORIN,UTIN,UCIN,UOIN,ITIME,KELAST)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     Stiffness force models
%       L. Fernando Sirumbal Z.
%       April, 2013
%
%   Function to calculate the internal stiffness force corresponding to
%   the elastoplastic hysteretic model
%
%   inputs
%
%
%       VELOCI      Velocity vector of the current time step.
%       NDOF        Number of degrees of freedom of the system.
%       KEYIN       Vector specifying the nonlinear state of each degree of
%                   freedom in the previous time step.
%                   KEY=0   Elastic behavior
%                   KEY=1   Plastic behavior in tension
%                   KEY=-1  Plastic behavior in compression
%       DISPLA1     Displacement vector of the previous time step
%       DISPLA2     Displacement vector of the current time step
%       INTFORIN    Spring internal stiffness force of the previous time
%                   step
%       UTIN        Maximum tensile displacement in the elastic range of
%                   the previous time step
%       UCIN        Maximum compression displacement in the elastic range
%                   of the previous time step
%       UOIN        Zero stress displacement of the previous time step
%       ITIME       Current time step index
%       KELAST      Vector containing the elastic stiffness coefficients of
%                   the springs
%
%
%   outputs
%
%       KEYOUT      Vector speciying the nonlinear state of each degree of
%                   freedom in the current time step.
%                   KEY=0   Elastic behavior
%                   KEY=1   Plastic behavior in tension
%                   KEY=-1  Plastic behavior in compression
%       INTFOROUT   Spring internal stiffness force of the current time
%                   step
%       FNLSIOUT    Non-linear restoring stiffness force vector of the
%                   current time step
%       UTOUT       Maximum tensile displacement in the elastic range of
%                   the current time step
```

```
%       UCOUT       Maximum compression displacement in the elastic range
%                   of the current time step
%       UOOUT       Zero stress displacement of the current time step
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Define the nonlinear elasto-plastic parameters for each degree of
% freedom
% This data has to be modified by the user depending of the problem data
%
TENSTR=[7.5 7.5 12.35 12.35 0 0 0 0 0 0];% Tensile strength for each degree of freedom
COMSTR=[-7.5 -7.5 12.35 12.35 0 0 0 0 0 0];% Compressive strength for each degree of freedom
if ITIME==1
    for IDOF=1:NDOF+1
        UTIN(IDOF)=TENSTR(IDOF)/KELAST(IDOF);
        UCIN(IDOF)=COMSTR(IDOF)/KELAST(IDOF);
    end
    sizet=size(TENSTR);
    sizec=size(COMSTR);
    sizeut=size(UTIN);
    sizeuc=size(UCIN);
    sizeuo=size(UOIN);
    %
    % Checking if the number of parameters provided are enough
    %
    if sizet(2)<NDOF+1 || sizec(2)<NDOF+1
        disp('Error message: the number coefficients of the vectors "TENSTR" and COMSTR, defined
in function "nonlin_hyster_stiff_int_force...", should be equal or higher to NDOF +1')
    elseif sizeut(2)<NDOF+1 || sizeuc(2)<NDOF+1 || sizeuo(2)<NDOF+1
        disp('Error message: the number coefficients of the vectors "UTIN", "UCIN" AND "UOIN",
defined in function "nonlin_hyster_stiff_int_force", should be equal or higher to NDOF +1')
    end
end
%
% Calculation of relative displacements increments and relative velocities
%
DISREL1=zeros(NDOF+1,1);
DISREL2=zeros(NDOF+1,1);
VELREL=zeros(NDOF+1,1);
DISREL1(1)=DISPLA1(1);
DISREL1(NDOF+1)=-DISPLA1(NDOF);
DISREL2(1)=DISPLA2(1);
DISREL2(NDOF+1)=-DISPLA2(NDOF);
VELREL(1)=VELOCI(1);
VELREL(NDOF+1)=-VELOCI(NDOF);
if NDOF>=2
    for IDOF=2:NDOF
        DISREL1(IDOF)=DISPLA1(IDOF)-DISPLA1(IDOF-1);
        DISREL2(IDOF)=DISPLA2(IDOF)-DISPLA2(IDOF-1);
```

```
        VELREL(IDOF)=VELOCI(IDOF)-VELOCI(IDOF-1);
    end
end
VARDISREL=DISREL2-DISREL1;
%
% Defining the nonlinear state and nonlinear internal force for each degree
% of freedom in the current time step
EMKEY=100;
KEYOUT=zeros(NDOF+1,1);
INTFOROUT=zeros(NDOF+1,1);
UTOUT=UTIN;
UCOUT=UCIN;
UOOUT=UOIN;
for IDOF=1:NDOF+1
    if KELAST(IDOF)==0
        INTFOROUT(IDOF)=0;
    elseif KEYIN(IDOF)==0
        INTFOROUT(IDOF)=INTFORIN(IDOF)+KELAST(IDOF)*VARDISREL(IDOF);
        if INTFOROUT(IDOF)>=TENSTR(IDOF)
            KEYOUT(IDOF)=1;
            INTFOROUT(IDOF)=TENSTR(IDOF);
        elseif INTFOROUT(IDOF)<=COMSTR(IDOF)
            KEYOUT(IDOF)=-1;
            INTFOROUT(IDOF)=COMSTR(IDOF);
        else
            KEYOUT(IDOF)=0;
        end
    elseif KEYIN(IDOF)==1
        INTFOROUT(IDOF)=TENSTR(IDOF);
        if VELREL(IDOF)>0
            KEYOUT(IDOF)=1;
        else
            KEYOUT(IDOF)=0;
            if VARDISREL(IDOF)<0
                INTFOROUT(IDOF)=TENSTR(IDOF)+KELAST(IDOF)*VARDISREL(IDOF);
                UTOUT(IDOF)=DISREL1(IDOF);
            else
                UTOUT(IDOF)=DISREL2(IDOF);
            end
            UCOUT(IDOF)=UTOUT(IDOF)-(TENSTR(IDOF)-COMSTR(IDOF))/KELAST(IDOF);
            UOOUT(IDOF)=UTOUT(IDOF)-TENSTR(IDOF)/KELAST(IDOF);
        end
    else
        INTFOROUT(IDOF)=COMSTR(IDOF);
        if VELREL(IDOF)<0
            KEYOUT(IDOF)=-1;
        else
            KEYOUT(IDOF)=0;
            if VARDISREL(IDOF)>0
                INTFOROUT(IDOF)=COMSTR(IDOF)+KELAST(IDOF)*VARDISREL(IDOF);
```

```
                UCOUT(IDOF)=DISREL1(IDOF);
            else
                UCOUT(IDOF)=DISREL2(IDOF);
            end
            UTOUT(IDOF)=UCOUT(IDOF)+(TENSTR(IDOF)-COMSTR(IDOF))/KELAST(IDOF);
            UOOUT(IDOF)=UTOUT(IDOF)-TENSTR(IDOF)/KELAST(IDOF);
        end
    end
end
%
% Recalculation of the nonlinear restoring force vector for the current time
% step
FNLSTIOUT=zeros(NDOF,1);
for IDOF=1:NDOF
    FNLSTIOUT(IDOF)=INTFOROUT(IDOF)-INTFOROUT(IDOF+1);
end
```

Published with MATLAB® R2012b

```
function
[KEYOUT,INTFOROUT,FNLSTIOUT,UTOUT,UCOUT,UOOUT,EMKEY]=f11_nonlin_hyster_stiff_int_force_cubic(VELO
CI,NDOF,KEYIN,DISPLA1,DISPLA2,UTIN,UCIN,UOIN,ITIME,KELAST)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     Stiffness force models
%       L. Fernando Sirumbal Z.
%       April, 2013
%
%   Function to calculate the internal stiffness force corresponding to
%   the cubic elastoplastic hysteretic model
%
%
%   inputs
%
%       VELOCI      Velocity vector of the current time step.
%       NDOF        Number of degrees of freedom of the system.
%       KEYIN       Vector specifying the nonlinear state of each degree of
%                   freedom in the previous time step.
%                   KEY=0   Elastic behavior
%                   KEY=1   Plastic behavior in tension
%                   KEY=-1  Plastic behavior in compression
%       DISPLA1     Displacement vector of the previous time step
%       DISPLA2     Displacement vector of the current time step
%       UTIN        Maximum tensile displacement in the elastic range of
%                   the previous time step
%       UCIN        Maximum compression displacement in the elastic range
%                   of the previous time step
%       UOIN        Zero stress displacement of the previous time step
%       ITIME       Current time step index
%       KELAST      Vector containing the elastic stiffness coefficients of
%                   the springs
%
%
%   outputs
%
%       KEYOUT      Vector speciying the nonlinear state of each degree of
%                   freedom in the current time step.
%                   KEY=0   Elastic behavior
%                   KEY=1   Plastic behavior in tension
%                   KEY=-1  Plastic behavior in compression
%       INTFOROUT   Spring internal stiffness force of the current time
%                   step
%       FNLSIOUT    Non-linear restoring stiffness force vector of the
%                   current time step
%       UTOUT       Maximum tensile displacement in the elastic range of
%                   the current time step
%       UCOUT       Maximum compression displacement in the elastic range
%                   of the current time step
```

```
%       UOOUT       Zero stress displacement of the current time step
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Define the nonlinear cubic elasto-plastic parameters for each degree of
% freedom
% Cubic non-linearity    fs=k1*u-k3*u^3
% This data has to be modified by the user depending of the problem data
%
k1=[8 8 8 8 8 8 8 8 8 8];
k3=[32/27 32/27 32/27 32/27 32/27 32/27 32/27 32/27 32/27 32/27];
d=sqrt(k1./(3*k3));
TENSTR=(k1.*d)-k3.*(d.^3);% Tensile strength for each degree of freedom
COMSTR=-TENSTR;% Compressive strength for each degree of freedom
if ITIME==1
    UTIN=transp(d);
    UCIN=-UTIN;
    sizet=size(TENSTR);
    sizec=size(COMSTR);
    sizeut=size(UTIN);
    sizeuc=size(UCIN);
    sizeuo=size(UOIN);
    %
    % Checking if the number of parameters provided are enough
    %
    if sizet(2)<NDOF+1 || sizec(2)<NDOF+1
        disp('Error message: the number coefficients of the vectors "TENSTR" and COMSTR, defined
in function "nonlin_hyster_stiff_int_force...", should be equal or higher to NDOF +1')
    elseif sizeut(2)<NDOF+1 || sizeuc(2)<NDOF+1 || sizeuo(2)<NDOF+1
        disp('Error message: the number coefficients of the vectors "UTIN", "UCIN" AND "UOIN",
defined in function "nonlin_hyster_stiff_int_force", should be equal or higher to NDOF +1')
    end
end
%
% Calculation of relative displacements increments and relative velocities
DISREL1=zeros(NDOF+1,1);
DISREL2=zeros(NDOF+1,1);
VELREL=zeros(NDOF+1,1);
DISREL1(1)=DISPLA1(1);
DISREL1(NDOF+1)=-DISPLA1(NDOF);
DISREL2(1)=DISPLA2(1);
DISREL2(NDOF+1)=-DISPLA2(NDOF);
VELREL(1)=VELOCI(1);
VELREL(NDOF+1)=-VELOCI(NDOF);
if NDOF>=2
    for IDOF=2:NDOF
        DISREL1(IDOF)=DISPLA1(IDOF)-DISPLA1(IDOF-1);
        DISREL2(IDOF)=DISPLA2(IDOF)-DISPLA2(IDOF-1);
```

```
        VELREL(IDOF)=VELOCI(IDOF)-VELOCI(IDOF-1);
    end
end
VARDISREL=DISREL2-DISREL1;
%
% Defining the nonlinear state and nonlinear internal force for each degree
% of freedom in the current time step
EMKEY=100; %Value of state variable for error message
KEYOUT=zeros(NDOF+1,1);
INTFOROUT=zeros(NDOF+1,1);
UTOUT=UTIN;
UCOUT=UCIN;
UOOUT=UOIN;
for IDOF=1:NDOF+1
    if KELAST(IDOF)==0
        INTFOROUT(IDOF)=0;
    elseif KEYIN(IDOF)==0
        if DISREL2(IDOF)>=UTIN(IDOF)
            KEYOUT(IDOF)=1;
            INTFOROUT(IDOF)=TENSTR(IDOF);
        elseif DISREL2(IDOF)<=UCIN(IDOF)
            KEYOUT(IDOF)=-1;
            INTFOROUT(IDOF)=COMSTR(IDOF);
        else
            KEYOUT(IDOF)=0;
            INTFOROUT(IDOF)=k1(IDOF)*(DISREL2(IDOF)-UOIN(IDOF))-k3(IDOF)*((DISREL2(IDOF)-
UOIN(IDOF))^3);
        end
    elseif KEYIN(IDOF)==1
        INTFOROUT(IDOF)=TENSTR(IDOF);
        if VELREL(IDOF)>0
            KEYOUT(IDOF)=1;
        else
            KEYOUT(IDOF)=2;
            if VARDISREL(IDOF)<0
                INTFOROUT(IDOF)=TENSTR(IDOF)+KELAST(IDOF)*VARDISREL(IDOF);
                UTOUT(IDOF)=DISREL1(IDOF);
            else
                UTOUT(IDOF)=DISREL2(IDOF);
            end
            UOOUT(IDOF)=UTOUT(IDOF)-TENSTR(IDOF)/KELAST(IDOF);
            UCOUT(IDOF)=UOOUT(IDOF)-d(IDOF);
        end
    elseif KEYIN(IDOF)==2
        if DISREL2(IDOF)<=UOIN(IDOF)
            KEYOUT(IDOF)=3;
            INTFOROUT(IDOF)=k1(IDOF)*(DISREL2(IDOF)-UOIN(IDOF))-k3(IDOF)*((DISREL2(IDOF)-
UOIN(IDOF))^3);
        elseif DISREL2(IDOF)>= UTIN(IDOF)
            KEYOUT(IDOF)=1;
```

```
            INTFOROUT(IDOF)=TENSTR(IDOF);
        else
            KEYOUT(IDOF)=2;
            INTFOROUT(IDOF)=KELAST(IDOF)*(DISREL2(IDOF)-UOIN(IDOF));
        end
    elseif KEYIN(IDOF)==3
        if DISREL2(IDOF)<=UCIN(IDOF)
            KEYOUT(IDOF)=-1;
            INTFOROUT(IDOF)=COMSTR(IDOF);
        elseif DISREL2(IDOF)>= UOIN(IDOF)
            KEYOUT(IDOF)=2;
            INTFOROUT(IDOF)=KELAST(IDOF)*(DISREL2(IDOF)-UOIN(IDOF));
        else
            KEYOUT(IDOF)=3;
            INTFOROUT(IDOF)=k1(IDOF)*(DISREL2(IDOF)-UOIN(IDOF))-k3(IDOF)*((DISREL2(IDOF)-
UOIN(IDOF))^3);
        end
    else
        EMKEY=KEYIN(IDOF);
        SPRINGNUM=IDOF;
        fprintf('\nNot defined state variable KEYIN=%d of spring number %d for the calculation of
time step %d\n',EMKEY,SPRINGNUM,ITIME+1)
        return
    end
end
%
% Recalculation of the nonlinear restoring force vector for the current time
% step
FNLSTIOUT=zeros(NDOF,1);
for IDOF=1:NDOF
    FNLSTIOUT(IDOF)=INTFOROUT(IDOF)-INTFOROUT(IDOF+1);
end
```

Published with MATLAB® R2012b

B.3.2. Damping force functions

```
function [FNLDAMOUT,INTFOROUT]=f12_lin_damp_int_force_v2(NDOF,VARVEL,INTFORIN,CELAST)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     Damping force models
%       L. Fernando Sirumbal Z.
%       April, 2013
%
%   Function to calculate the internal damping force corresponding to
%   the linear model
%
%   inputs
%
%       NDOF        Number of degrees of freedom of the system
%       VARVEL      Velocity increment vector from the previous to the
%                   current time step
%       INTFORIN    Dashpot internal damping force of the previous time
%                   step
%       CELAST      Vector containing the linear damping coefficients of
%                   the dashpots
%
%
%   outputs
%
%       INTFOROUT   Dashpot internal damping force of the current time
%                   step
%       FNLDAMOUT   Non-linear restoring damping force vector of the
%                   current time step
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Calculating the relative variation of velocity
VARVELREL=zeros(NDOF+1,1);
VARVELREL(1)=VARVEL(1);
VARVELREL(NDOF+1)=-VARVEL(NDOF);
if NDOF>=2
    for IDOF=2:NDOF
        VARVELREL(IDOF)=VARVEL(IDOF)-VARVEL(IDOF-1);
    end
end
%
% Calculation of dashpot force
INTFOROUT=zeros(NDOF+1,1);
for IDOF=1:NDOF+1
    INTFOROUT(IDOF)=INTFORIN(IDOF)+CELAST(IDOF)*VARVELREL(IDOF);
```

```
end
%
% Calculation of nonlinear internal damping force
FNLDAMOUT=zeros(NDOF,1);
for IDOF=1:NDOF
    FNLDAMOUT(IDOF)=INTFOROUT(IDOF)-INTFOROUT(IDOF+1);
end
```

Published with MATLAB® R2012b

```
function [FNLDAMOUT,INTFOROUT]=f13_nonlinmet_damp_int_force(NDOF,DISPLA,VELOCI,CELAST)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     Damping force models
%       L. Fernando Sirumbal Z.
%       April, 2013
%
%   Function to calculate the internal damping force corresponding to
%   the non-linear displacement dependent model
%
%   inputs
%
%       NDOF        Number of degrees of freedom of the system
%       DISPLA      Displacement vector of the current time step
%       VELOCI      Velocity vector of the current time step
%       CELAST      Vector containing the linear damping coefficients of
%                   the dashpots
%
%
%   outputs
%
%       INTFOROUT   Dashpot internal damping force of the current time
%                   step
%       FNLDAMOUT   Non-linear restoring damping force vector of the
%                   current time step
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Quadratic displacement dependent non-linearity
%   fd=CELAST*VELOCI(ALFA*DISPLA^2+1)
%
%Nonlinear damping parameters
alfa=1;
%
%Relative displacement
DISREL=zeros(NDOF+1,1);
DISREL(1)=DISPLA(1);
DISREL(NDOF+1)=-DISPLA(NDOF);
if NDOF>=2
    for IDOF=2:NDOF
        DISREL(IDOF)=DISPLA(IDOF)-DISPLA(IDOF-1);
    end
end
%
%Relative velocity
VELREL=zeros(NDOF+1,1);
VELREL(1)=VELOCI(1);
VELREL(NDOF+1)=-VELOCI(NDOF);
```

```
if NDOF>=2
    for IDOF=2:NDOF
        VELREL(IDOF)=VELOCI(IDOF)-VELOCI(IDOF-1);
    end
end
%
% Calculation of dashpot force
INTFOROUT=zeros(NDOF+1,1);
for IDOF=1:NDOF+1
    INTFOROUT(IDOF)=VELREL(IDOF)*CELAST(IDOF)*(alfa*((DISREL(IDOF))^2)+1);
    %INTFOROUT(IDOF)=VELREL(IDOF)*CELAST(IDOF)*(alfa*((DISREL(IDOF))^2)-1);
end
%
% Calculation of nonlinear internal damping force
FNLDAMOUT=zeros(NDOF,1);
for IDOF=1:NDOF
    FNLDAMOUT(IDOF)=INTFOROUT(IDOF)-INTFOROUT(IDOF+1);
end
```

Published with MATLAB® R2012b

APPENDIX C: DIANA HFTD COMMAND FILE

```
*FILOS
 INITIA
*INPUT
*PHASE
 BEGIN ACTIVE
   ELEMEN "Dam" "Found" /
   REINFO
 END ACTIVE
*NONLIN
 BEGIN EXECUT
   BEGIN LOAD
     LOADNR 1
     STEPS  EXPLIC  SIZES 0.25(4)
   END LOAD
   ITERAT  CONVER  SIMULT
 END EXECUT
*PHASE
*HFTD
 BEGIN EIGEN
   EXECUT  NMODES 15
   BEGIN OUTPUT
     FXPLUS
     FILE "eigenHFTD_dry"
   END OUTPUT
 END EIGEN
 BEGIN MODEL
   BEGIN MATRIX
     DAMPIN
     STRESS  PHASE
   END MATRIX
 END MODEL
 BEGIN RESPON
   BEGIN EXECUT
     ITERAT  MAXITE 50
     MAXFRE 25
     TIMESE  STEPS  EXPLIC  NUMBER 50
   END EXECUT
   BEGIN OUTPUT
     FXPLUS
     FILE "respHFTD_wet"
     ACCELE  TOTAL  TRANSL  GLOBAL
     DISPLA  TOTAL  TRANSL  GLOBAL
     FORCE  REACTI  TRANSL  GLOBAL
     FSPRES  TOTAL
     STATUS  CRACK
     STRAIN  TOTAL  GREEN  GLOBAL
     STRAIN  CRACK  GREEN
     STRESS  TOTAL  CAUCHY  GLOBAL
     VELOCI  TOTAL  TRANSL  GLOBAL
   END OUTPUT
   REDUCE  SELECT  MODES 1-15 /
 END RESPON
*END
```