Research Signpost
37/661 (2), Fort P.O., Trivandrum-695 023, Kerala, India

Recent Research Developments in Structural Dynamics, 2003 : 33-54 ISBN : 81-7736-186-4
Editor: Angelo Luongo

2 Analytical and numerical techniques in frequency domain response computation

Federico Perotti[1] and Valeria Simoncini[2]
[1]Dipartimento di Ingegneria Strutturale, Politecnico di Milano, Piazza Leonardo da Vinci, 32 20133 Milano, Italy; [2]Dipartimento di Matematica, Università di Bologna and IMATI-CNR Pavia, Piazza di Porta S. Donato, 5, 40127 Bologna, Italy. E-mail: val@imati.cnr.it

Abstract

In this paper the analytical formulation of frequency domain analysis is reviewed, under a unifying approach, for deterministic and random dynamic loading. In the random case both stationary and nonstationary excitations are considered. It is shown how, in all cases, a complex linear system must be solved at each frequency step; if the solutions are carried on independently, the solution cost grows almost linearly with the number of steps. Aiming at a computationally more efficient approach, iterative procedures are here proposed and developed, allowing for the simultaneous solution for a large number of frequencies. Some of the prolems arising in applications involving "large" systems are then discussed and solved. Examples are given with particular reference to 3D soil-structure finite-element models.

Correspondence/Reprint request: Dr. Federico Perotti, Dipartimento di Ingegneria Strutturale, Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20133 Milano, Italy. E-mail: perotti@stru.polimi.it

1. Introduction

A significant portion of the "cutting-edge" research in structural dynamics is related to non-linear problems; nevertheless, linearized solutions are still of utmost importance in the field. In fact there exist a number of practical cases in which such solutions are motivated not only by simplicity but also by the problem inherent features. We refer to situations where anticipated deformations in structural materials are extremely small and where all other hypotheses leading to linearization (small oscillations etc.) are actually justified on mechanical grounds. This is typical of vibration analyses under "serviceability" loads, among which we can quote the following:

- vibration of machine foundations due to unbalanced rotating parts;
- propagation of dynamic disturbances induced by traffic (road or rail) loading;
- propagation of dynamic disturbances due to construction activities (pile driving, demolitions, use of explosives, etc);
- structural vibration induced by human activities (walking, running, etc);
- structural vibration due to normal wind or wave loading.

Within this context, direct frequency domain analysis (DFDA) can be an efficient alternative to modal analysis and to direct time domain techniques. In fact, while direct step-by-step time integration offers unique opportunities to treat non-linear systems, and while modal analysis seems to be a very reliable approach to "standard" linear problems, DFDA appears as a more powerful tool for handling complex linear systems. This is because of its inherent capability to deal with frequency dependent mechanical properties of the dynamical model which is set for the analysis, where such dependence can be due to different causes of both physical and analytical nature.

On physical grounds, frequency dependent properties (typically stiffness) of a mechanical system or component can be related to the influence of strain rate on the material properties; this is typical, for example, of rubber devices which are widely used as absorbing bearing pads (see [1] and references herein).

In most cases, however, frequency dependence is related to modeling criteria and assumptions, which often lead to drastic simplification of complex phenomena or, in other cases, to a reduction of the problem size. The first situation is typical of the simplified treatment of interaction problems; as an example we can quote wind-structure interaction studies, where aerodynamic forces on elongated bodies are expressed in terms of frequency dependent aerodynamic coefficients, which are obtained from wind tunnel tests on harmonically oscillating specimens [2].

Reduction (or condensation) of the problem size, on the other hand, is often related to system substructuring. A typical example can be found in linearized soil-structure analysis [3] where, in many situations, the behavior of the ground subsystem can be "condensed" into a limited number of functions of frequency (impedances) expressing the linear relation between forces and displacement components at interface nodes.

A simpler case of frequency dependence is represented by the linear hysteretic damping (LHD) model, in which the damping coefficients are assumed to be inversely proportional to the circular frequency. The model has been introduced with reference to steady state harmonic analysis; its use within the more general context of frequency analysis for arbitrary loading has been questioned, since LHD violates, in transient analysis, the causality condition (see [4] and references herein); in spite of this deficiency, the model will be addressed in this work, in view of the above "very small deformation" hypothesis. Under such hypothesis, limited material hysteresis is anticipated, leading to lack-of-causality errors which can be accepted in front of the significant advantages of the mixed (viscous plus hysteretic) damping model in treating complex dissipation properties.

In the first part of this paper the analytical formulation of DFDA analysis will be reviewed, under a unified approach, both for deterministic and for stochastic dynamic loading; for the latter case, stationary and nonstationary excitations will be considered. Nonstationary random vibration will be treated within the framework of evolutionary processes, according to the well-known Priestley's theory [5].

By the computational standpoint it will be shown how, in all cases, the most critical aspect of frequency domain analysis is represented by the linear system solution which is necessary, at all frequency steps, for evaluating vibration amplitudes and phases. The coefficient matrix of the system (mechanical impedance matrix) is complex, symmetric and indefinite and shows a polynomial (quadratic) variation with frequency, while the right-hand side remains constant. In such situation a question naturally arises, i.e. whether all solutions must be carried on independently to each other, or the peculiarities of the algebraic problem can be exploited for developing efficient strategies aiming at the simultaneous system solution for a large number of frequency values.

A survey of existing literature [6] indicates that strategies of this type have been developed with reference to the "standard shifted form" of a linear system, i.e. to the form $[\,\boldsymbol{T}+\lambda \boldsymbol{I}\,]\boldsymbol{z}=\boldsymbol{d}$, where λ is the parameter, which in our case must be related to frequency, and $\boldsymbol{I}$ is the identity matrix. These strategies are based on the properties of a class of iterative solvers (Krylov subspace solvers, see [7] for a comprehensive treatment) for which all linear systems corresponding to different λ values can be projected onto the same vector basis. Since the generation of the vector space (Krylov subspace) is the most expensive step of the iterative solution, large computational savings can be obtained from the simultaneous approach.

A preliminary problem to be solved for applying Krylov subspace methods is the transformation of the original complex system (which is quadratic in the frequency parameter) into a convenient form, linear in λ. This transformation can be performed in different ways, leading to different properties of the iterative solutions of the standard shifted system. In previous research work [6] it was shown that the transformation which provides the best convergence properties to the iterative schemes is such that, unfortunately, a system solution with the stiffness matrix is required. In spite of this computational overhead cost, the procedures for simultaneous solution which were subsequently developed (based upon the GMRES solver [8]) demonstrated excellent performance in comparison to independent direct solutions, at least for "medium-size" systems (up to 10-20 x 10^3 degrees of freedom).

While GMRES appears to be the method of choice on small size problems for its favorable convergence properties, different strategies need to be devised for large problems, in order to limit memory requirements (see [6]). We showed in [9] that an *ad-hoc* implementation of the Lanczos procedure (SSL - Simplified Shifted Lanczos method) can conveniently exploit the symmetry properties of the problem so as to comply with the mentioned limits. In this paper we review the SSL algorithm and linge over some practical implementation aspects that make the method amenable to the treatment of very large application problems.

Another aspect concerns the system solution with the stiffness matrix, which is necessary for the transformation into the standard form; for small and medium size systems the matrix can be factored once and stored for the remaining of the analysis. For systems having a larger number of degrees of freedom, this becomes impossible and an iterative scheme must be again devised, leading to a so-called “inner-outer” iteration, where “inner” denotes the solution with the stiffness matrix and “outer” the overall procedure for the simultaneous solution of the frequency steps. A procedure of this type has been developed (ISSL - Inexact Simplified Shifted Lanczos) and will be here described with particular reference to some critical aspects of the inner iteration, namely the choice of the stopping criterion and the treatment of ill-conditioned stiffness matrices.

In the last section of this paper some numerical experiments will be reported, mainly focused on 3D soil-structure interaction problems. The performance of the SSL and ISSL procedures will be first compared to that of an efficient iterative solver, applied separately at each frequency. Further results will be finally given, specifically addressing the case of ill-conditioned stiffness matrices and the influence of the tolerance of the inner procedure.

2. Frequency analysis of discretized linear systems

We are interested in the dynamic response of a discretized *n-DOF* linear, time-invariant system, whose motion is described by the following matrix equation

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{F}\mathbf{g}(t) \tag{1}$$

where **M**, **C**, and **K** are the mass, viscous damping and stiffness (n, n) matrices respectively, while **q** is the configuration vector, **F** is a (n, m) load amplitudes matrix and **g** is a (m,1) load histories vector. In most practical situations involving large systems the number of independent load histories m is at least one order of magnitude smaller than n.

In the most general case we can suppose $\mathbf{g}(t)$ to be a realization of an m-dimensional non stationary stochastic process; focusing our attention to the case of evolutionary processes [5], the loading histories can be expressed in the Fourier-Stieltjes integral form:

$$\mathbf{g}(t) = \int_{-\infty}^{\infty} \mathbf{\Psi}(t,f)\, e^{i2\pi ft} d\mathbf{G}(f) \tag{2}$$

where $\mathbf{\Psi}(t, f)$ is a diagonal matrix containing m deterministic, slowly varying modulating functions. The $d\mathbf{G}(f)$ vector, whose components must be of the order of $\sqrt{df}$, lists the realizations of an m dimensional process having the following orthogonality property:

$$E\left[\, d\mathbf{G}(f_1)\, d\mathbf{G}^*(f_2)\,\right] = \delta(f_2 - f_1)\mathbf{S}_g(f_1)\, df_1\, df_2 \tag{3}$$

In Eq. (3) the asterisk denotes conjugation and transposition, E[ ] is the expectation operator, δ is a Dirac delta and $\mathbf{S}_g(f)$ is the (m, m) power spectral density matrix of the "underlying" m-dimensional stationary process whose realizations can be expressed as

$$\overline{\mathbf{g}}(t) = \int_{-\infty}^{\infty} e^{i2\pi ft} d\mathbf{G}(f) \tag{4}$$

The system response to the above-described dynamic excitation can be computed via the usual time domain convolution:

$$\mathbf{q}(t) = \int_{t_0}^{t} \mathbf{h}(t-\tau)\,\mathbf{F}\,\mathbf{g}(\tau)d\tau \tag{5}$$

where $\mathbf{h}(t)$ is the (n, n) matrix of unit-impulse response functions and t_0 the time in which initial conditions are imposed. Substituting Eq. (2), interchanging the order of integration and applying the change of variable $\boldsymbol{u} = \boldsymbol{t} - \boldsymbol{\tau}$ leads to the following frequency domain integral:

$$\mathbf{q}(t) = \int_{t_0}^{t} \mathbf{h}(t-\tau)\mathbf{F}\int_{-\infty}^{\infty} \mathbf{\Psi}(\tau, f)e^{i2\pi f\tau}d\mathbf{G}(f)d\tau = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathbf{h}(u)\mathbf{F}\,\mathbf{\Psi}(t-u, f)e^{-i2\pi fu}du\; e^{i2\pi ft}d\mathbf{G}(f) =$$

$$= \int_{-\infty}^{\infty} \mathbf{\Gamma}(t, f)e^{i2\pi ft}d\mathbf{G}(f) \tag{6}$$

Note that the upper limit of the convolution integral (5) can always be extended to infinity for causal systems, since $\mathbf{h}(t - \tau) = \mathbf{0}$ for $\tau > t$: the lower limit can go to minus infinity whenever the envelope functions are null for $t < t_0$ and in all situations in which the effect of initial condition has vanished.

In Equation 6 a time varying frequency response function (n, m) matrix was defined as:

$$\mathbf{\Gamma}(t, f) = \int_{-\infty}^{\infty} \mathbf{h}(u)\mathbf{F}\,\mathbf{\Psi}(t-u, f)e^{-i2\pi fu}du \tag{7}$$

It's worth noting that for a deterministic transient excitation, for which $\mathbf{G}(f)$ is differentiable and $\mathbf{\Psi}(t, f)$ can be omitted, we get the following standard relations

$$\mathbf{\Gamma}(t, f) = \int_{-\infty}^{\infty} \mathbf{h}(u)\; e^{-i2\pi fu}du\,\mathbf{F} = \mathbf{H}(f)\,\mathbf{F} \tag{8}$$

$$\mathbf{q}(t) = \int_{-\infty}^{\infty} \mathbf{H}(f)\,\mathbf{F}\,\widetilde{\mathbf{g}}(f)e^{i2\pi ft}df \tag{9}$$

where $\mathbf{H}(f)$ is the (n, n) matrix of frequency response functions and the $\widetilde{\mathbf{g}}(f)$ vector is the Fourier Transform of $\mathbf{g}(t)$.

In the stochastic case Eq. (6) points out that $\mathbf{q}(t)$ is again a multi-dimensional nonstationary process of evolutionary type, having $\mathbf{\Gamma}(t, f)$ as modulating function matrix.

Since this matrix is no longer diagonal, each ealization of $\mathbf{q}(t)$ is actually the sum of m evolutionary process realizations.

To define the evolutionary spectral density matrix of the output process $\mathbf{q}(t)$, we can express the time-varying covariance matrix in terms of the response equation (6) derived above, i.e.:

$$\boldsymbol{\mu}_q(t) = E\left[\mathbf{q}(t)\mathbf{q}^*(t)\right] = \int_{-\infty}^{\infty} \mathbf{S}_q(t,f)df = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \boldsymbol{\Gamma}(t,f_1)E\left[d\mathbf{G}(f_1)d\mathbf{G}^*(f_2)\right]\boldsymbol{\Gamma}^*(t,f_2)$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \boldsymbol{\Gamma}(t,f_1)\mathbf{S}_g(f_1)\delta(f_2-f_1)\boldsymbol{\Gamma}^*(t,f_2)\,df_2 df_1 = \int_{-\infty}^{\infty} \boldsymbol{\Gamma}(t,f)\mathbf{S}_g(f)\boldsymbol{\Gamma}^*(t,f)\,df \qquad (10)$$

Eq. (10) implicitly defines the (n, n) evolutionary spectral density matrix of $\mathbf{q}(t)$ as:

$$\mathbf{S}_q(t,f) = \boldsymbol{\Gamma}(t,f)\mathbf{S}_g(f)\boldsymbol{\Gamma}^*(t,f) \qquad (11)$$

In the stationary case (unit envelope functions, i.e. $\boldsymbol{\Psi} = \mathbf{I}$) Eq. (8) holds and the spectral density matrix of the response process takes the usual form

$$\mathbf{S}_q(f) = \mathbf{H}(f)\mathbf{F}\mathbf{S}_g(f)\mathbf{F}^T\mathbf{H}^*(f) \qquad (12)$$

When the computation of $\boldsymbol{\Gamma}(f, t)$ is considered, we can note that Eq. (7) is the time convolution of two matrix functions, namely $\mathbf{F}\boldsymbol{\Psi}\ (t,f)$ and $\mathbf{h}(t)\exp(-i2\pi\, f\, t)$. According to the frequency domain convolution theorem, the Fourier Transform of $\boldsymbol{\Gamma}(t, f)$ can thus be computed as the following product of the transforms of the two functions

$$\widetilde{\boldsymbol{\Gamma}}(\alpha,f) = \int_{-\infty}^{\infty} \mathbf{h}(u)e^{-i2\pi(f+\alpha)u}du\,\mathbf{F}\,\widetilde{\boldsymbol{\Psi}}(\alpha,f) = \mathbf{H}(\alpha+f)\mathbf{F}\,\widetilde{\boldsymbol{\Psi}}(\alpha,f) \qquad (13)$$

where $\widetilde{\boldsymbol{\Psi}}(\alpha,f)$ is the Fourier Transform of the envelope functions matrix and $\mathbf{H}(f)$ is again the frequency response matrix. The time varying frequency response matrix can be thus computed as:

$$\boldsymbol{\Gamma}(t,f) = \int_{-\infty}^{\infty} \mathbf{H}(\alpha+f)\mathbf{F}\,\widetilde{\boldsymbol{\Psi}}(\alpha,f)e^{i2\pi\alpha t}d\alpha \qquad (14)$$

For the evolutionary case, we can note that the envelopes are slowly varying so that their transforms are narrow-banded around the frequency origin. Therefore, for each frequency f the envelope transforms act, in the integral (14), as bandpass filters centered on f and applied to the frequency response functions.

For the limiting case of a stationary excitation process, i.e. for a unit envelope function, we have $\widetilde{\boldsymbol{\Psi}}(\alpha,f) = diag\left[\delta(\alpha)\right]$ and Eq. (14) obviously yields to the same result as in Eq. (8).

3. The formulation of the algebraic problem in frequency analysis

A key issue in the evaluation of the numerical performance of frequency domain analysis is represented, as it can be inferred by Eq. (9) (deterministic case), Eq. (12) (random stationary) and Eqs. (11-14) (random nonstationary), by the computation of the product **H**(*f*)**F**. This amounts to solving, for a large number of frequencies, the following complex symmetric linear system with *m* right-hand sides:

$$\mathbf{E}(f)\,\overline{\mathbf{q}}(f)=\mathbf{F} \tag{15}$$

where **E**(*f*) is the well-known mechanical impedance matrix

$$\mathbf{E}(f)=\mathbf{H}^{-1}(f)=\left[\ -(2\pi f)^2\mathbf{M}+i(2\pi f)\mathbf{C}+\mathbf{K}\ \right] \tag{16}$$

When mixed damping (hysteretic+viscous) is introduced, the impedance matrix takes the complex and symmetric form:

$$\mathbf{E}(f)=\left[\ -(2\pi f)^2\mathbf{M}+i(2\pi f)\mathbf{C}_V+\mathbf{K}+i\mathbf{C}_H\ \right] \tag{17}$$

where $\mathbf{C}_V$ and $\mathbf{C}_H$ are the viscous and hysteretic damping matrices respectively.

If a direct solver is employed in (15) the computing time grows almost linearly with the number of frequencies, since the factorization must be recomputed for each value of *f*. On the other hand, to exploit the properties of shifted systems, Eqs. 15-17 must be transformed into the standard form. As demonstrated in previous research [6], the performance of the iterative procedures which will be used for the system solution is significantly improved if the system is first rewritten in terms of acceleration amplitudes as unknowns, namely

$$\left[\mathbf{M}+\frac{1}{i(2\pi f)}\mathbf{C}_V+\frac{1}{-(2\pi f)^2}\mathbf{K}_*\right]\overline{\mathbf{a}}=\mathbf{F} \tag{18}$$

where $\mathbf{K}_*=\mathbf{K}+i\,\mathbf{C}_H$, $\overline{\mathbf{a}}=-(2\pi f)^2\overline{\mathbf{q}}$ and a "single-input" system (*m=1*) has been first considered.

To obtain the standard form, a two-step approach can be adopted. In the first step the system is transformed into a new one, of twice the size, of the type

$$\left[\mathbf{A}+\lambda\mathbf{B}\right]\mathbf{x}=\mathbf{d} \tag{19}$$

where $\lambda=(i2\pi f)^{-1}$, $\mathbf{d}=\left[\mathbf{0}\,;\mathbf{F}\right]$ and where **A** and **B** are the following (2*n*, 2*n*) symmetric matrices:

$$\mathbf{A}=\begin{bmatrix}0 & \mathbf{M}\\ \mathbf{M} & \mathbf{C}_V\end{bmatrix}\ ;\ \mathbf{B}=\begin{bmatrix}-\mathbf{M} & 0\\ 0 & \mathbf{K}_*\end{bmatrix}\ ;\ \mathbf{x}=\begin{bmatrix}\overline{\mathbf{a}}\\ (i2\pi f)^{-1}\overline{\mathbf{a}}\end{bmatrix} \tag{20}$$

As a second step $\mathbf{B}^{-1}$ can be collected on the right, to get the standard shifted system in the form

$$[\ \mathbf{T} + \lambda \mathbf{I}_{2n}]\,\mathbf{z} = \mathbf{d} \quad ; \quad \mathbf{T} = \mathbf{A}\,\mathbf{B}^{-1} \quad ; \quad \mathbf{z} = \mathbf{B}\,\mathbf{x} \tag{21}$$

in which $\mathbf{I}_{2n}$ is the unit matrix of dimension *2n*.

The formulation obtained, though suitable for the efficiency of iterative solvers, has the obvious disadvantage of requiring solutions with matrix **B**. For "medium size" systems the matrix can be factored once and stored. For larger systems different strategies must be adopted. We also emphasize that the case of **M** singular can be dealt with easily; indeed, since **M** is diagonal, its zero rows and columns can be eliminated. In such situation, the resulting augmented system is of size (n + n_m)**,** where n_m is the number of nonzero diagonal elements in **M** [10].

It can also be noted that, though **A** and **B** in (20) are symmetric, symmetry is lost in (21). We recall, however that a square matrix **X** is **J**-symmetric if $\mathbf{JX} = \mathbf{X}^T\mathbf{J}$, where **J** is a symmetric nonsingular matrix; for our case it is easy to verify that the matrix **T** is both $\mathbf{A}^{-1}$ and $\mathbf{B}^{-1}$ symmetric. This property will be exploited in the solution phase.

4. The iterative solvers

4.1 Krylov subspace methods

The special structure of a standard shifted system can be efficiently exploited by Krylov subspace methods [7]; these methods approximate the system solution by projecting the problem on the subspace

$$\mathbf{K}_k(\mathbf{T}, \mathbf{v}) = span\{\mathbf{v}, \mathbf{Tv}, ..., \mathbf{T}^{k-1}\mathbf{v}\} \tag{22}$$

where **T** is the coefficient matrix and **v** is a starting vector. The latter is usually taken as a normalized residual:

$$\mathbf{v} = \mathbf{r}_0 / \beta = (\mathbf{d} - \mathbf{T}\mathbf{z}_0)/\beta \tag{23}$$

where $\mathbf{z}_0$ is a starting approximate solution and $\beta^2 = \|\mathbf{r}_0\|^2 = \sum_{i=1}^{n} |r_{0,i}|^2$ is the square of the norm of $\mathbf{r}_0$.

The subspace (22) is invariant under shift, so that the system (21) can be projected onto the same subspace for all values of the parameter λ, as long as the generating vector **v** does not depend on the parameter.

4.2 The GMRES method

In previous research work [6] the performance of the GMRES solver [8] was tested. Given a starting vector $\mathbf{v}_1 \neq \mathbf{0}$, at each iteration k the method generates a new vector $\mathbf{v}_{k+1}$ of the Krylov subspace by multiplying the previous vector $\mathbf{v}_k$ by the matrix **T** and by orthogonalizing $\mathbf{v}_{k+1}$, according to the Gram-Schmidt technique, with respect to all previous vectors $\mathbf{v}_1,\ldots\mathbf{v}_k$. By defining the matrix $\mathbf{V}_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$, whose columns span $\mathbf{K}_{\boldsymbol{k}}$ (**T**, **v**), the following relation holds

$$\mathbf{T}\,\mathbf{V}_k = \mathbf{V}_{k+1}\mathbf{N}_k \tag{24}$$

where $\mathbf{N}_\mathrm{k}$ is a (k + 1, k) upper Hessenberg matrix whose coefficients are computed according to the orthonormalization procedure.

In GMRES, the approximation $\mathbf{z}_k = \mathbf{z}_0 + \mathbf{V}_k\ \mathbf{y}$ is obtained by minimizing the norm of the residual $\mathbf{r}_k = \mathbf{d} - \mathbf{T}\mathbf{z}_k$. Denote by $\mathbf{e}_k$ the kth column of the unit matrix. By using (23) and the orthonormality condition, the residual can be computed as follows:

$$\mathbf{d} - \mathbf{T}\mathbf{z}_k = \mathbf{d} - \mathbf{T}(\mathbf{z}_0 + \mathbf{V}_k\,\mathbf{y}) = \mathbf{r}_0 - \mathbf{T}\mathbf{V}_k\,\mathbf{y} = \beta\ \mathbf{v}_1 - \mathbf{V}_{k+1}\mathbf{N}_k\mathbf{y} = \mathbf{V}_{k+1}(\beta\mathbf{e}_1 - \mathbf{N}_k\mathbf{y}) \tag{25}$$

Orthonormality of the columns of $\mathbf{V}_{k+1}$ yields the simplified expression

$$\|\mathbf{d} - \mathbf{T}(\mathbf{z}_0 + \mathbf{V}_k\,\mathbf{y})\| = \|\beta\mathbf{e}_1 - \mathbf{N}_k\,\mathbf{y}\| \tag{26}$$

The minimization of (26) is simple and inexpensive, since it requires the solution of a least squares problem of size (k + 1, k), where k is the subspace dimension.

The main advantage of the GMRES method is that monotonic convergence, in terms of residual norm, is assured. On the other hand, the subspace grows during the iteration and all basis vectors must be kept for the orthogonalization process. To face the problem a restart option can be adopted: when the Krylov subspace reaches a maximum allowable size, a new subspace is generated assuming the last computed residual as a new starting residual.

When the simultaneous solution of the system (21) for several, say s, values of λ is sought, the subspace generation must be performed only once for all parameters; each solution vector can be subsequently obtained by the residual minimization. Some problems arise when a restart must be performed; to solve them the true minimization must be sacrificed, so that monotonic convergence is no longer guaranteed. We also point to the more recent work in [11] for further investigations on this topic.

In [6] the performance of the simultaneous solution procedure based upon GMRES and restarted GMRES was compared to the one of the sparse symmetric solver ME47 [12] for six prototype systems stemming from civil engineering dynamics problems. The number of degrees of freedom ranged from 327 to 11957; in almost all cases, and especially for the largest systems, the simultaneous iterative procedures based on GMRES outperformed the direct solver. In addition, the computing time growth as a function of the number of frequencies was almost linear for the direct solver, whereas it was very limited for the proposed procedures (only +6% passing from 11 to 101 solutions for the largest model). Note that ME47 was also used to factorize **B**, that is **K,** to obtain $\mathbf{B}^{-1}\mathbf{v}_j$ in the iterative procedures.

4.3 Procedures based on the Lanczos method

We first refer to the non-shifted solution **T z** = **d**. For a non-symmetric matrix **T** the Lanczos algorithm generates the two Krylov subspaces

$$\mathbf{K}_k(\mathbf{T},\mathbf{v}) = span\left\{\mathbf{v},\mathbf{T}\mathbf{v},...,\mathbf{T}^{k-1}\mathbf{v}\right\} \qquad (27)$$

$$\mathbf{K}_k(\mathbf{T}^T,\tilde{\mathbf{v}}) = span\left\{\tilde{\mathbf{v}},\mathbf{T}^T\tilde{\mathbf{v}},...,\left(\mathbf{T}^T\right)^{k-1}\tilde{\mathbf{v}}\right\} \qquad (28)$$

The vector sequences $\mathbf{v}_j$ and $\tilde{\mathbf{v}}_j$ are biorthogonal, i.e. $\tilde{\mathbf{v}}_j^T\mathbf{v}_i = 0$ for $j \neq 1$ and they satisfy a "three-term recurrence" algorithm, based on the matrix relation:

$$\mathbf{T}\mathbf{V}_k = \mathbf{V}_{k+1}\mathbf{S}_k \qquad (29)$$

where $\mathbf{S}_k$ is a tridiagonal (k+1, k) matrix. The first basis vector $\mathbf{v}_1$ is again the normalized starting residual, while $\tilde{\mathbf{v}}_1$ is arbitrary. To compute an approximate solution as $\mathbf{z}_k = \mathbf{z}_0 + \mathbf{V}_k\mathbf{y}$, similar reasoning as in GMRES can be applied, leading to computation of the residual according to:

$$\mathbf{d} - \mathbf{T}\mathbf{z}_k = \mathbf{V}_{k+1}\left(\beta\mathbf{e}_1 - \mathbf{S}_k\mathbf{y}\right) \qquad (30)$$

Since the columns of $\mathbf{V}_{k+1}$ are no longer orthonormal, a relation of the type (26) does not hold true. However it seems still reasonable to minimize the function $\|\beta\mathbf{e}_1 - \mathbf{S}_k\mathbf{y}\|$ to compute $\mathbf{y}$ and the approximate solution; this procedure is called Quasi-Minimal Residual (QMR) approximation. In a practical implementation, the three-term recurrence above is usually replaced by a more stable coupled two-term recurrence. We refer to [13] for a detailed description of this procedure, which is the one used in the experiments reported in section 5.

Whenever **T** is symmetric, the Lanczos algorithm builds an orthogonal basis of the Krylov subspace $\mathbf{K}_k$; in the popular "Conjugate Gradient" (CG) method an approximate solution is subsequently computed by imposing that the system residual is orthogonal (Galerkin condition) to the subspace $\mathbf{K}_k$.

4.3.1 The Simplified Shifted Lanczos (SSL) method

In our setting the system to be solved is $\left[\mathbf{T} + \lambda\,\mathbf{I}_{2n}\right]\mathbf{z} = \mathbf{d}$. Let us for the moment omit the shift $\lambda\mathbf{I}$. We have observed that the matrix $\mathbf{T} = \mathbf{A}\mathbf{B}^{-1}$ is $\mathbf{B}^{-1}$-symmetric. Moreover, by choosing $\tilde{\mathbf{v}}_1 = \gamma_1\mathbf{B}^{-1}\mathbf{v}_1$, for some complex γ_1, we can compute the next vector of the left subspace (28) by first performing the following operation

$$\mathbf{T}^T\tilde{\mathbf{v}}_1 = \gamma_1\mathbf{T}^T\mathbf{B}^{-1}\mathbf{v}_1 = \gamma_1\mathbf{B}^{-1}\mathbf{T}\mathbf{v}_1 \qquad (32)$$

The next vector $\tilde{\mathbf{v}}_2$ is then obtained upon normalizaton and orthogonalization with respect to $\tilde{\mathbf{v}}_1$. In the same way we can show that all subsequent vectors satisfy the relation $\tilde{\mathbf{v}}_k = \gamma_k\mathbf{B}^{-1}\mathbf{v}_k$, where γ_k is a normalization factor and can be determined from γ_{k-1}; therefore the left Krylov subspace in (28) is simply $\mathbf{B}^{-1}$ times the right Krylov subspace. Note that a solution with **B** is anyway necessary, at each iteration, for the computation of the next vector in the right Krylov subspace. This means that, in practice, only the set of the $\mathbf{v}_k$ basis vectors need to be recursively computed, while the vectors $\widetilde{\mathbf{v}}_k$ can be recovered at no additional cost [9].

When the shifted system is considered, due to the property $\mathbf{K}_k(\mathbf{T}+\lambda\mathbf{I}_{2n},\mathbf{v}_1)=\mathbf{K}_k(\mathbf{T},\mathbf{v}_1)$ (shift-invariance of Krylov subspaces), the following relation holds:

$$\left[\mathbf{T}+\lambda\mathbf{I}_{2n}\right]\mathbf{V}_k=\mathbf{V}_{k+1}\left[\mathbf{S}_k+\lambda\bar{\mathbf{I}}_k\right] \tag{33}$$

where $\bar{\mathbf{I}}_k=\left[\mathbf{I}_k;\,\mathbf{0}\right]$. Note that the generating vector must be the same for all value of λ. Since we take $\mathbf{v}_1$ to be equal to the normalized starting residual (23), $\mathbf{z}_0$ must be the zero vector in order to have $\mathbf{v}_1 = \mathbf{d}$ for all shifted systems. Once the space $\mathbf{K_k}$ ($\mathbf{T}$, $\mathbf{v}_1$) is built, an approximate solution $\mathbf{z}_k(\lambda_j) = \mathbf{V}_k\ \mathbf{y}\ (\lambda_j)$ can be obtained for each $\lambda_j, j=1,\dots s$, by using a QMR procedure, i.e. by minimizing

$$\left\|\beta\mathbf{e}_1-\left[\mathbf{S}_k+\lambda_j\bar{\mathbf{I}}_k\right]\mathbf{y}\right\| \tag{34}$$

Finally, when $\mathbf{z}_k$ is obtained the approximate solution to the original system (19) is determined as $\mathbf{x}_k=\mathbf{B}^{-1}\mathbf{z}_k$. However, since $\mathbf{z}_\mathrm{k}$ is obtained as a linear combination of the columns of $\mathbf{V}_k$, from (32) it follows that the solution $\mathbf{x}_\mathrm{k}$ can directly be updated using the (left) basis $\widetilde{\mathbf{V}}_k$.

Convergence of the method can be monitored by computing the norm of the residual of the system (19), i.e. $\mathbf{r}_k = \mathbf{d}-[\mathbf{A} + \lambda\mathbf{B}]\mathbf{x}_k$. Upper bounds of the residual norm can be easily obtained as shown in [9]. Let $\mathbf{x}_k=\left[\mathbf{x}_k^{(1)};\mathbf{x}_k^{(2)}\right]$. If after k iterations the exact solution were determined, we would have that $\mathbf{x}_k^{(2)}=\lambda\,\mathbf{x}_k^{(1)}$. In general, $\mathbf{x}_k^{(1)}$ provides an approximate solution $\bar{\mathbf{a}}_k$ to the original problem (18), whereas the residual norm $\|\mathbf{r}_k\|$ can substantially differ from that of the original system (18), $\mathbf{R}_k=\mathbf{F}-\left[\mathbf{M}+\lambda\mathbf{C}_v+\lambda^2\mathbf{K}_*\right]\bar{\mathbf{a}}_k$. However, if the approximate solution of (18) is estimated as $\bar{\mathbf{a}}_k=\lambda^{-1}\mathbf{x}_k^{(2)}$, then the following relation holds [10]

$$\mathbf{R}_k=\left[\mathbf{I}_n,\frac{1}{\lambda}\mathbf{I}_n\right]\mathbf{r}_k$$

and $\|\mathbf{R}_k\|$ can easily be bounded, at iteration k, by using the estimates of $\|\mathbf{r}_k\|$. At termination the choice between $\mathbf{x}_k^{(1)}$ and $\lambda^{-1}\mathbf{x}_k^{(2)}$ can be done “a posteriori”.

In terms of memory requirements the overall procedure is very efficient. In fact the Krylov subspaces are obtained by means of the Lanczos short-term recurrence. The basis vectors do not need to be stored, since each solution $\mathbf{z}_k(\lambda_j) = \mathbf{V}_k\ \mathbf{y}(\lambda_j)$ can be updated, at each iteration, by only employing the last computed vectors. The same consideration applies to the minimization (34), which can be carried out without explicitly assembling the whole matrix $\mathbf{S}_k$. Globally SSL requires six vectors of *2n* components and two (2*n*, *s*) matrices, where s is the number of frequencies; the size of the matrices can be drastically reduced when, as often happens, only a limited number of components of the solution vector is of interest.

4.3.2 Acceleration procedures

The velocity of convergence of the iterative procedure depends on the spectral properties of the coefficient matrix. In typical iterative strategies, preconditioning (see again [7]) of the matrix is usually applied to enhance these properties.

We recall that a preconditioner of an iterative technique can be roughly defined as a matrix operator **P**, which is applied to the system matrix **T** at each iteration in order to achieve better efficiency.

A common choice is to perform an "incomplete" factorization of **T**; this is defined as $\mathbf{P} \approx \mathbf{LU}$, where **L** and **U** are the complete factors of **T** and **P** is determined by dropping some of the entries of **L** and **U,** according to a specified criterion. The simplest and most economical criterion is to impose to **P** the same sparsity pattern of **T**. The resulting preconditioner is often termed as incomplete factorization with "zero fill-in". More sophisticated techniques entail the definition of strategies for performing some degree of fill-in, i.e. for adding coefficients to **P**; this results in a more demanding preconditioning procedure, both in terms of computing time and storage, which can be compensated by acceleration of convergence. A fill-in parameter p is generally defined as the number of the coefficients per column which are null in **T** and non-zero in the preconditioner.

Preconditioning of the SSL method, however, is complicated by two problems; the first is that the coefficient matrix $\mathbf{AB}^{-1}$ is not formed explicitly, so that incomplete factorization schemes are not applicable. Secondly, preconditioning a system while maintaining the shifted structure is by itself a challenging problem.

An alternative acceleration strategy consists in shifting, i.e. choosing a reference value λ_0 and then solving the problem, either the original (18) or the linearized (19), with respect to the parameter $\overline{\lambda} = \lambda - \lambda_0$. Implications and problems related to shifting are thoroughly discussed in [9]; here we recall that this technique could be also employed to face breakdown problems which may be encountered using the Lanczos procedure. In the numerical experiments we shall report on, no particular acceleration strategy has been employed and no breakdown occurred.

4.3.3 Application to "large size" systems

As already pointed out, the SSL method solves one of the problems related to the application of Krylov subspace solvers to the analysis of "large" shifted systems, i.e. the generation of the subspace on the basis of a short-term recurrence capable of exploiting the **J**-symmetric structure of matrix **T**.

A second problem is represented by the solution with matrix **B**, i.e. with the complex stiffness matrix $\mathbf{K}_*$, which is necessary at each iteration. Though such matrix is sparse, its factor can be quite dense and memory consuming.

In such situations an iterative scheme for the solution with **B** can be adopted for the larger cases, leading to an inner-outer iterative procedure; within this context, a problem that must be faced is the influence of the inner system solution accuracy on the overall performance of the method. Some very recent research results [21] have shown how to *dinamically* change the inner stopping tolerance without affecting the overall convergence history of the method. As a result, significant computation cost savings can be observed at no implementation overhead; we shall show that this technique can be successfully applied in our context.

Another important aspect is represented by the choice of the inner procedure. In the experiments reported here the Conjugate Gradient method for complex symmetric matrices was used; preconditioning was performed according to the ICT algorithm by Chow & Saad [14], adapted for complex systems. The ICT method produces an incomplete Choleski factorization, governed by a fill-in parameter and a threshold value; the latter ensures that small elements are not taken into consideration in the fill-in procedure. The resulting inner-outer iterative procedure has been termed as ISSL (Inexact Simplified Shifted Lanczos).

A particular situation arises when the stiffness matrix $\mathbf{K}_*$ is singular. A typical case is represented by unrestrained systems. In infinite precision this implies that the matrix has six zero eigenvalues associated with the rigid-body modes. In practice, these eigenvalues are no longer zero, but some orders of magnitude smaller than the others and the rigid-body modes are close (known) approximations to the associated eigenvectors. The resulting matrix is in general severely ill-conditioned; numerically speaking the system behaves as if highly deformable springs were supporting it. In the following we shall call quasi-rigid-body (QRB) modes those associated with the six small natural frequencies.

To avoid the numerical problems due to ill-conditioning, we can exploit the fact that the vector basis $\mathbf{W}$ of size (n, 6) listing the model rigid-body modes can be easily formed, and that these vectors are good approximations to the eigenvectors associated to the QRB modes. Hence, a recent projection-type algorithm can be exploited that efficiently employs the computed basis $\mathbf{W}$ [15]. The idea is to generate an approximation space that adds $\mathbf{W}$ to the currently generated Krylov subspace. In [15] it is shown that for CG-type methods this can be done in an efficient manner, so that each iteration only requires a few additional operations on vectors of size n. We shall call this approach the *Augmented CG* (ACG) method, as it augments (enriches) the approximation space with the basis $\mathbf{W}$. It can be proved that the method is particularly effective when $\mathbf{W}$ is a good approximation to eigenvectors corresponding to the smallest eigenvalues (in modulo) of the coefficient matrix, which is exactly our setting.

4.4 Multiple-input systems

So far we have considered the case of a single right-hand side, i.e. the situation of a dynamic loading which is characterized by a single time history. In the multiple-input case the linear system (18) must be solved for all right-hand sides represented by the m columns of $\mathbf{F}$. In such situation the linearized system (19) takes the form

$$\left\lfloor \mathbf{A}+\lambda_j \mathbf{B} \right\rfloor \mathbf{X}=\mathbf{D} \tag{35}$$

where the r.h.s. $\mathbf{D} = [\mathbf{0}; \mathbf{F}]$ and the solution $\mathbf{X} = \mathbf{X}(\lambda_j)$ are now ($2n$, m) matrices.

The system (35) can be transformed into the standard form as $\left[\mathbf{T}+\lambda\,\mathbf{I}_{2n}\right]\mathbf{Z}=\mathbf{D}$. The latter problem can be solved by using a generalization of the Simplified Shifted

Lanczos algorithm for multiple right-hand sides. The methods builds the following *block* Krylov subspace:

$$\mathbf{K}_{mk}(\mathbf{T},\mathbf{D}) = span\left\{\mathbf{D},\mathbf{TD},...,\mathbf{T}^{k-1}\mathbf{D}\right\} \tag{36}$$

and then approximates all shifted systems using the shift-invariance of $\mathbf{K}_{mk}$ (cf. eg. [16]). The block version of the two-term Lanczos recurrence can be used for efficiently generating the subspace (36). Again, the approximate solutions are obtained as $\mathbf{Z}_k(\lambda_j) = \mathbf{V}_k\,\mathbf{Y}_k(\lambda_j)$, where the least squares solution $\mathbf{Y}_k$ is updated at each iteration at a cost which is of the order of m^2.

It must be observed that, in some settings, there exists an alternative strategy for facing the multiple-input case. Looking at the deterministic response computation as given in Eq. (9), we note that if the right-hand side in Eq. (1) is expressed in term of the (n, 1) time varying vector $\mathbf{P}(t) = \mathbf{Fg}(t)$, the frequency domain convolution takes the form

$$\mathbf{q}(t) = \int_{-\infty}^{\infty} \mathbf{H}(f)\,\tilde{\mathbf{p}}(f)e^{i2\pi ft}df \tag{37}$$

where $\tilde{\mathbf{p}}(f)$ is the Fourier Transform of $\mathbf{p}(t)$. Therefore, to compute the Fourier Transform $\tilde{\mathbf{q}}(f)$ of the response vector the following linear system must be solved:

$$\mathbf{E}(f)\tilde{\mathbf{q}}(f) = \tilde{\mathbf{p}}(f) \tag{38}$$

showing a frequency dependent single right-hand side. The linearized version of (38) can be still cast in the form (19), where, however, $\mathbf{d} = \mathbf{d}(\lambda)$. To solve the problem all systems can be treated simultaneously, yielding a Sylvester equation whose solution can be again approximated upon projection on a block Krylov subspace. The implementation is analogous to that for constant multiple right-hand sides. The computational cost for updating the least squares solution $\mathbf{y}_k$ at each iteration, however, is proportional to the square s^2 of the number of frequencies. The preference between the two approaches here described, therefore, depends on the magnitude of s (number of frequencies) and m (number of independent loading histories).

Finally, we observe that in [17] a method is proposed, for real symmetric positive definite matrix $\mathbf{AB}^{-1}$, that deals with the multiple right-hand side and associated shift. It would be interesting to see whether a generalization to our setting would be possible.

5. Numerical experiments

5.1 Description of test cases

In previous research activity it was demonstrated how simultaneous solution strategies based on Krylov subspace methods are able to outperform direct solvers for what we called "medium size" systems, i.e. systems for which it is possible to store the triangularized stiffness matrix.

The Inexact SSL method was subsequently developed for targeting applications to larger systems. In this light, its performance was compared to the one of an efficient iterative solver, applied to the original system (18) separately for each frequency;

numerical experiments are reported in section 5.2. Further experiments were run, dealing with the aspects of the ISSL implementation which we mentioned in the previous section, i.e. the problem of treating ill-conditioned **B** matrices (section 5.3) and the influence of the inner tolerance on the overall procedure (section 5.4).

The experiments were performed on the test cases whose essential features are gathered in Table 1; cases B, C, F and F1 represent a subset of the examples considered in [6]. Table 1 lists the number of DOFs, the number of nonzero coefficients in the stiffness matrix and an estimate of its condition number in the 1-norm. The frequency interval employed for the analysis is also given. In all cases a single dynamic load was applied to the dynamic model.

Table 1. Numerical properties of the test cases.

Test case	Problem size	# elements in **K**	Cond(**K**)	Freq. int [Hz]
B	3627	102378	9.7×10^4	0.1 - 60.1
C	2472	23340	3.6×10^7	0.1 - 60.1
F	11957	419160	2.6×10^{12}	10. – 50.
F1	11907	416855	2.0×10^7	10. – 50.
G	42941	1601793	-	10. – 37.

In case B a reinforced concrete foundation is modeled by means of 8-node brick elements resting on a Winkler-type spring bed. In case C a reinforced concrete structure is considered, composed of an upper deck, which is modeled with shell elements and is connected to a lower massive foundation beam by means of column elements. Soil-structure interaction is considered by a lumped parameter approach; accordingly, the foundation beam is restrained by springs and dashpots modeling ground deformability and damping.

In cases F and F1 a simple 3-D FE soil-structure model is analyzed, encompassing a rigid slab (2 x 2 m) centered on a ground parallelepiped (10 x 10 x 4 m) which is modeled via 8-node brick elements (see Fig. 1). Ground shear modulus and Poisson coefficient are taken equal to 100 Mpa and 0.3 respectively. According to Lysmer and Kuhlemeyer [18], viscous dampers are used as absorbing boundaries at nodes located on the outer faces of the ground mesh (see Fig. 2), thus leading to a singular complex stiffness matrix.

To avoid strict singularity, highly deformable springs (stiffness k_s = 1 kN/m) were put, in case F, in parallel to the quoted dampers, this leading to the very unfavorable condition number (2.6×10^{12}) listed in Table 1; in F1 the springs stiffness was taken equal to 1000 kN/m to achieve a condition number of 2×10^7. In judging the magnitude of k_s it can be noted that, for a soil of the type here considered (see elastic modulus) a reasonable value of the modulus of subgrade reaction should be of the order of 10-50 x 10^3 kN/m^3 leading to a spring constant of 2500-12500 kN/m.

In case G the same problem is analyzed by using a larger FE mesh (14 x 14 x 8 m) with the same element size (0.5 m). In order to test the influence of the condition properties of K and the performance of the ACG method (see 4.3.3), three case were analyzed with different boundary conditions, corresponding to k_s equal to zero, 100 and 1000 kN/m.

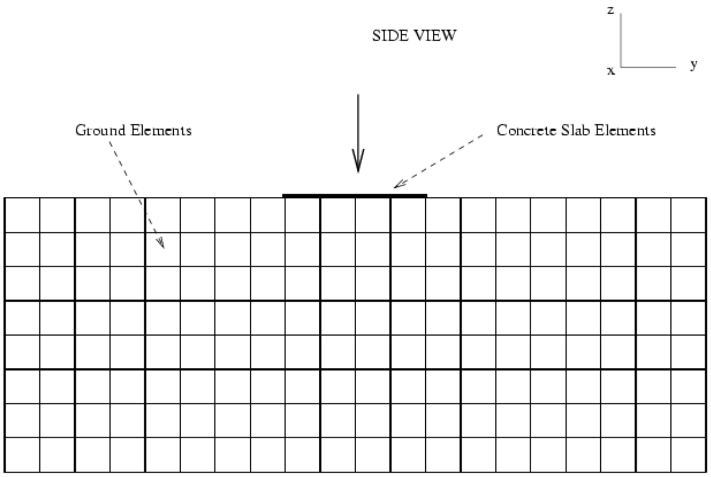

Figure 1. Test case F/F1: side view of the FE discretization.

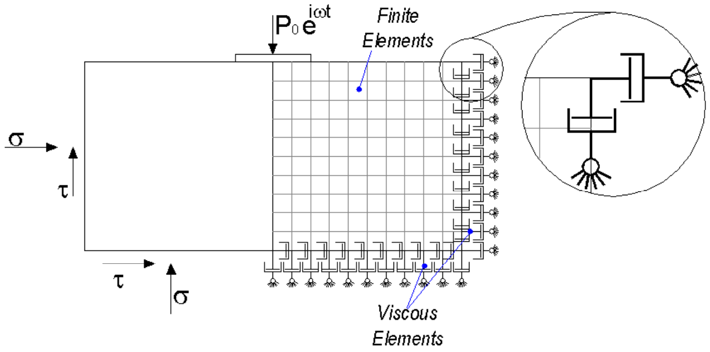

Figure 2. Viscous absorbing boundaries.

Note that the condition number was not estimated for the largest case G. Nonetheless, the same considerations as for cases F/F1 hold.

5.2 Comparison with other techniques

The iterative algorithm chosen for the comparison with ISSL is the complex symmetric QMR method [19]. Two strategies were considerated in the solution of (18); in the first one (PQMR-ME47) the exact factorization of the impedance matrix at *f=30Hz* was computed and applied, as a preconditioner, to all solutions. In the second (PQMR-ICT), the system was preconditioned by the ICT method with zero fill-in, performed at each frequency. Note that fill-in in the PQMR-ICT preconditioner provided an overall worse performance of the method than no fill-in.

The performance of ISSL was also compared to that of SSL, where the ME47 routine was used to solve for **B** at each iteration.

In the ISSL tests, the inner procedure was run with high preconditioning fill-in (*p*=100 for cases B and C and *p*=30 for cases F and F1) and strict tolerances (10^{-6} for case B and 10^{-8} for the rest). For the outer iteration the relative residual norm was tested, in all cases, against a tolerance of 10^{-4} (see [9] for further details).

In Figs 3 to 6 the results of the comparison are given in terms of total elapsed time vs. number of solutions, i.e. number of frequency values.

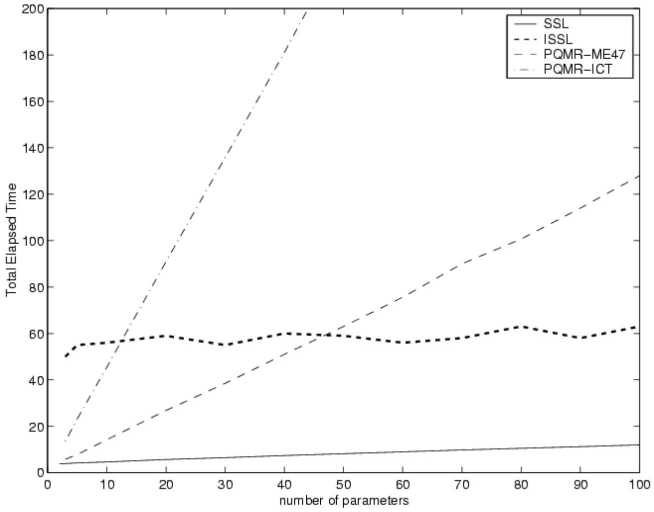

Figure 3. Case B. Performance comparison: simultaneous vs independent solutions.

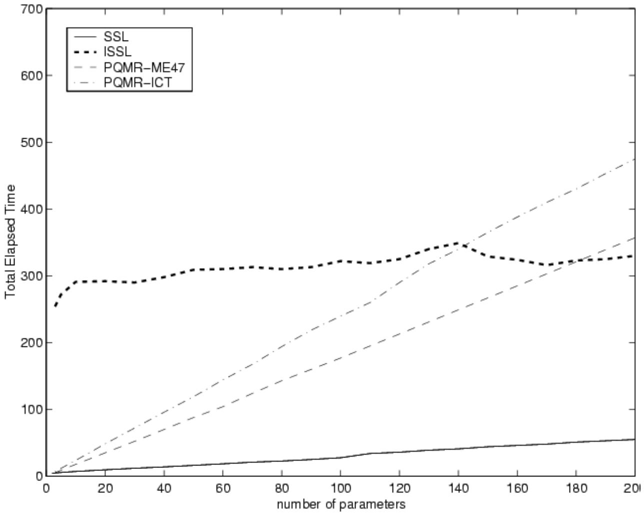

Figure 4. Case C. Performance comparison: simultaneous vs independent solutions.

Note that, in order to make a fair performance evaluation, SSL should be compared to PQMR-ME47, since they both perform a direct solution, while ISSL should be compared to PQMR-ICT. In the latter comparison we can notice that the sparsity pattern of the coefficient matrix in (18) is, in our application, the same as that of $\mathbf{K}_*$. Solving (18) is thus not much worse that solving with $\mathbf{K}_*$, so that the inner-outer solver cannot do better if the number of iterations to solve the linearized problem (21) is larger than the number of frequencies. On the other hand, unless $\mathbf{K}_*$ is severely ill-conditioned as in

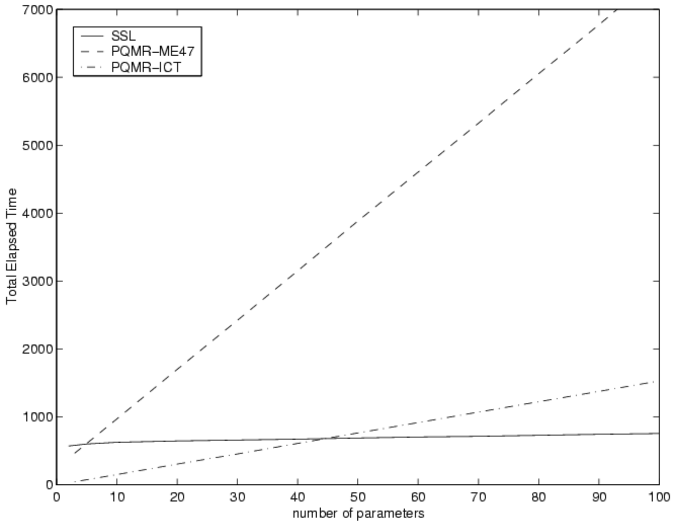

Figure 5. Case F. Performance comparison: simultaneous vs independent solutions.

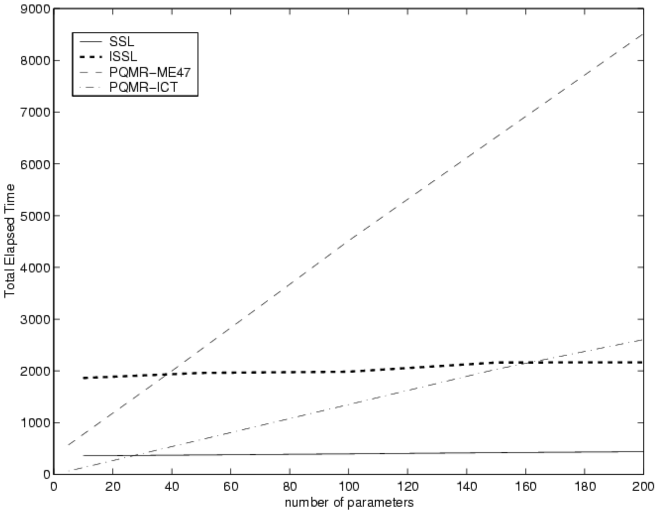

Figure 6. Case F1. Performance comparison: simultaneous vs independent solutions.

case F, ISSL outperforms PQMR-ICT for a large enough number of frequency values.

Examination of the curves in all figures suggests the following considerations.

- Severe ill-conditioning of $\mathbf{K}_*$ results in very poor performance of the inner-outer method, as can be seen from case F (Fig. 5) where the ISSL curve is out of the plot.
- Computing time grows very little with the number of frequencies for both SSL and ISSL.
- The SSL method is the most competitive in all cases, when a very limited number of solutions (45 for the most unfavorable case F) is exceeded.
- The ISSL method becomes competitive, with respect to independent solutions via PQMR-ICT, for a number of frequency steps ranging from about 12 (case B) to about 180 (case C).

5.3 Dealing with ill—conditioned stiffness matrices

Given the influence of the condition properties of $\mathbf{K}_*$ on the performance of the inner-outer method, some tests have been performed in order to assess the efficiency of the deflation procedure (ACG method) described in section 4.3.3. The tests were run on the three versions of model G, differing for the stiffness k_s of the springs at ground boundaries. Note that for $k_s = 0$ the system can undergo, in principle, rigid-body modes; actually, we have that, with a single-precision computation of $\mathbf{K}_*$, the first six eigenvalues of the matrix are of the order of 10^{-1}, while all others are between 10^3 and 10^7. In Table 2 we report the experimental results when using the Augmented CG method as inner solver and we compare the overall ISSL performance with that of ISSL applied to the model G when springs are put in parallel to the viscous boundaries. In the tests, ACG was preconditioned with ICT, by using the implementation described in [15].

The results in Table 2 are given in terms of computing time and number of iterations (average value for the inner procedure). The tests encompassed the solution for three frequency values (10, 23.3 and 36.7 Hz); inner tolerance was 10^{-6}, outer tolerance was 10^{-3}. Storage requirements for the preconditioner were dependent on the fill-in parameter; for the three values here considered (p equal to 5, 10 and 15) the memory allocation was respectively 1.8, 2 and 2.3 Mb for real variables and 1.9, 2.2 and 2.4 Mb for integer variables.

Table 2. Performance of the Augmented CG method and influence of stiffness conditioning.

	Fill-in p=5		Fill-in p=10		Fill-in p=15	
	E. Time [s]	# its (outer/ avg. inner)	E. Time [s]	# its (outer/ avg. inner)	E. Time [s]	# its (outer/ avg. inner)
$k_s = 0$ ACG	14066	296/38	13344	289/36	12790	281/33
$k_s = 100$	14072	117/120	14308	122/108	13739	121/102
$k_s = 1000$	8694	88/96	8383	89/88	8724	89/83

It can be observed that elapsed time is about 60% higher for the deflated model than for the "stiff-spring" model, which is obviously the best conditioned one. When k_s is actually small, however, computing times are practically the same as those obtained with ACG, even though the behavior is very different; in fact ACG is very efficient in solving the inner system, but this is compensated by slower convergence of the outer Lanczos procedure.

Higher fill-in in the preconditioner does not significantly change the performance; a 10% improvement, however, is detected for the ACG performance. In fact, since increase in memory requirements is limited it can be said that higher fill-in should be preferred. It is important to emphasize that the ACG method allows for a competitive solution of the original problem, without mechanical modifications.

5.4 Influence of the inner stopping criterion

In this section we report on numerical experiments we carried out on model G when using a dynamic inner tolerance. It is common belief that the inner system should be solved more accurately than the outer system throughout the iterative solution process. It was experimentally observed in [20] and then theoretically justified in [21] and [22] that the inner tolerance can indeed be relaxed (made looser) as the outer process converges. Using the notation of section 4.3.1, if $\rho_{k\text{-}1}$ represents a bound (see [9]) for the norm of the (outer) residual $\mathbf{R}_{k\text{-}1}$, then it can be shown that it is in general sufficient to impose the following inner stopping tolerance for solving the inner system $\mathbf{K}\,\mathbf{u} = \mathbf{d}$

$$\frac{\left\|\mathbf{d}-\mathbf{K}\mathbf{u}_{\mathbf{k}}\right\|}{||\,\mathbf{d}\,||} < \frac{l_{\bar{k}}\,\varepsilon}{\rho_{k-1}} \tag{39}$$

where $l_{\bar{k}}$ depends, in general, on the condition number of $\mathbf{K}$ and on the maximum number $\bar{k}$ of iterations allowed. Note that as ρ_{k-1} gets smaller, the inner stopping tolerance (the right-hand side in the formula above) becomes larger, thus allowing the inner method to stop after fewer iterations. The parameter $l_{\bar{k}}$ can be very small in some particular cases. However, $l_{\bar{k}} = 1$ works well in many circumstances; such value is what we used in our experiments, along with $\varepsilon = 10^{-6}$.

In Table 3 the results obtained running ISSL with different inner stopping strategies are summarized with the same criteria as in Table 2. In the tests, case G with $k_s = 0$ was considered and the preconditioned Augmented CG method was used as inner solver.

Table 3. Influence of inner stopping strategy on ISSL performance.

	Fill-in p=5		Fill-in p=10		Fill-in p=15	
	E. Time [s]	# outer its	E. Time [s]	# outer its	E. Time [s]	# outer its
Tol. 10^{-6}	14066	296	13344	289	12790	281
Tol. (39)	11579	301	11365	293	10775	299

It can be noted that the dynamic stopping criterion (35) leads to a speed-up of almost 20%, due to the fact that the inner tolerance grows during the iteration, without significantly affecting the performance of the outer procedure. In the case p=5, for example, if the dynamic stopping criterion is adopted, the inner residual at convergence grows from 4.7 x 10^{-7} at the first outer iteration to 8.7 x 10^{-4} at the last (301th), while the number of inner iterations, consequently, drops from 38 to 18. Note that 38 is the average number of inner iterations for the fixed tolerance case.

6. Conclusions

In this paper the performance of direct frequency domain analysis (DFDA) has been addressed both from the analytical and from the computational standpoint. The analytical formulation of DFDA has been first reviewed for deterministic and stochastic loads, showing how, in all situations, a linear system solution with the complex impedance matrix is necessary for each frequency value in the range of interest.

The numerical solution of such system has been addressed; a simultaneous solution strategy has been developed which can deliver the vibration amplitudes at a computational cost which grows sublinearly with the number of frequency values.

The procedure is based on Krylov subspace iterative solvers with Lanczos recurrence. At each iteration the procedure requires a solution with the stiffness matrix. For “medium-size” models this can be done with a direct solver (SSL method). For “large” models an inner-outer iterative method (ISSL) was devised. Particular attention was devoted, in its development and testing, to the influence of the inner system stopping criterion and to the treatment of ill-conditioned stiffness matrices.

Examples were given with particular reference to 3D finite-element soil-structure systems; the performance of the ISSL and SSL procedures were compared to that of efficient iterative solvers, applied separately for each frequency. The results of the comparison show that the simultaneous solution strategies can be very effective when, as usually happens, a significant number of frequency values must be analyzed.

Acknowledgment

The research of the first author was done with the contribution of the Italian Ministry for University and Scientific and Technological Research.

References

1. Mallik, A.K., Kher, V., Puri, M. and Hatwal, H. 1999, J. of Sound and Vibration, 219(2), 239-253.
2. Simiu, E. and Scanlan, R.H. 1996, Wind Effects on Structures, John Wiley & Sons, Inc. New York.
3. Wolf, J.P. 1985, Dynamic Soil-Structure Interaction, Prentice-Hall, Englewood Cliffs.
4. Feriani, A. and Perotti, F. 1996, Earthquake Engineering and Structural Dynamics, 25, 689-709.
5. Priestley, M.B. 1967, J. of Sound and Vibration, 6, 86-97.
6. Feriani, A., Perotti, F. and Simoncini, V. 2000, Computer Methods in Applied Mechanics and Engineering, 190, 1719-1739.
7. Saad, Y. 1996, Iterative Methods for Sparse Linear Systems, PWS Publishing Company, Boston.
8. Saad, Y. and Schultz, M.H. 1986, SIAM J. Sci. Stat. Comput.,7, 856-869.
9. Simoncini, V. and Perotti, F. 2002, SIAM J. Sci. Comput. 23, 1876-1898.
10. Simoncini, V. 1999, Iterative Methods in Scientific Computation II, D.Kincaid and A. Elster (Eds.) IMACS Series in Computational and Applied Mathematics, 5, 1999, 451-461.

11. Simoncini, V. 2002, to appear in BIT, Numerical Mathematics, also available at http://www.imati.cnr.it/~val.
12. AEA TECHNOLOGY 1995, Harwell Subroutine Library, Harwell Laboratory, Oxfordshire; England.
13. Freund, R.W. and Nachtigal, N.M. 1994, SIAM J. Sci. Comput., 15.
14. Chow, E. and Saad, Y. 1998, Private communication.
15. Saad, Y., Yeung, M., Ehrel, J. and Guyomarc'h, F. 2000, SIAM J. Sci. Comput., 21, 1909-1926.
16. Simoncini, V. and Gallopoulos, E. 1996, J. Comp. and Applied Math., 66, 457-469.
17. Chan, T.F. and Ng, M.K. 2001, SIAM J. Sci. Comput., 21, 836-850.
18. Lysmer, J. and Kuhlemeyer, R.L. 1969, ASCE J. Eng. Mech, 95, 859-876.
19. Freund, R.W. 1992, SIAM J. Sci. Stat. Comput., 13, 425-448.
20. Bouras, A., Fraissé, V. and Giraud L. 2000, Tech. Rep. TR/PA/00/17, CERFACS, Toulouse, France.
21. Simoncini, V. and Szyld, D.B. 2002, to appear in SIAM J. Scient. Computing, also available at http://www.imati.cnr.it/~val.
22. Sleijpen G.L. and van den Eshof, J. 2002, Preprint nr. 1224, Department of Mathematics, Universiteit Utrecht.