Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Journal of Computational Physics 211 (2006) 91–98

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

On the discrete dynamic nature of the conjugate gradient method

Y.T. Feng *

Civil and Computational Engineering Centre, School of Engineering, University of Wales Swansea, Singleton Park, Swansea Wales SA2 8PP, UK

Received 15 February 2005; received in revised form 16 April 2005; accepted 17 May 2005
Available online 6 July 2005

Abstract

The primary objective of this technical note is to establish an equivalence between the (preconditioned) conjugate gradient (PCG) algorithm and a special central difference based DR procedure, thereby revealing a discrete dynamic nature of the CG iterative procedure. This may therefore provide an alterative viewpoint to gain a further understanding of the CG method and its variants.

Keywords: Preconditioned conjugate gradient method; Dynamic relaxation; Central difference time integration; Discrete dynamics; Linear system of equations

1. Introduction

Consider a linear algebraic system of equations

$$\mathbf{Ku} = \mathbf{b} \tag{1}$$

in which $\mathbf{K}$ is an $n \times n$ symmetric positive-definite (SPD) matrix (where n is the order of the equations); and $\mathbf{b}$ and $\mathbf{u}$ are, respectively, the known right hand side and the solution to be sought.

The conjugate gradient (CG) method [1] is the most prominent and well-understood iterative procedure for solving the above equations. It possesses several unique properties including the guaranteed convergence in at most n iterations in exact arithmetic. In addition, the CG method has served as the basis for further developments of several CG-type iterative solvers (such as CGS [2] and BiCGStab [3]) for undefinite

* Tel.: +44 1792 295161; fax: +44 1792 295676.
E-mail address: y.feng@swansea.ac.uk.

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.05.006

and unsymmetric linear systems, and the basis of many algorithms widely used in general optimization and eigenvalue problems [4–6]. In practical terms, the CG method is free from any user-specified parameters and very simple to implement. With a proper preconditioning scheme [7,8], the preconditioned conjugate gradient (PCG) method is often found to be the most effective iterative solver for many applications, particularly for large scale problems.

On the other hand, the dynamic relaxation (DR) technique [9,10] can also be used to obtain a solution of Eq. (1). By viewing (1) as a static problem, the basic idea of the DR technique is to transform the problem into a damped second order dynamic system and the steady-state, often obtained by employing a central difference (CD) based time integration scheme, will be the solution to be sought.

The primary objective of this note is to establish an equivalence between the CG algorithm and a special central difference based DR procedure, thereby revealing a discrete dynamic nature of the CG iterative procedure. This may therefore provide an alterative viewpoint to gain a further understanding of the CG method. Note that the interpretation of stationary iterative solvers, including SOR and ADI, as time integration schemes for first order ODE systems of diffusion type is well-established [11].

As the CG method is well-documented, its algorithmic detail will be very briefly described. Slightly more detail will be given to the description of the standard DR approach, particularly a general central difference time integration scheme from which the discrete dynamic nature of the CG method is derived. Simple numerical examples will then be employed to highlight some dynamic features of the CG algorithm.

2. Theoretical aspects

2.1. Preconditioned conjugate gradient method

Suppose that the preconditioning matrix is $\mathbf{M}_\mathrm{P}$ (also symmetric and positive-definite). Without loss of generality a zero initial guess, $\mathbf{u}_0 = 0$, is assumed. The PCG algorithm can be summarized as follows:

- Set $\mathbf{u}_0 = 0$; $\mathbf{r}_0 = \mathbf{b}$. Compute $\mathbf{p}_0 = \mathbf{M}_\mathrm{P}^{-1}\mathbf{r}_0$
- For $i = 0, 1, 2, \ldots$ until convergence:

  (1) Update solution:

$$\mathbf{u}_{i+1} = \mathbf{u}_i + \alpha_i \mathbf{p}_i, \tag{2}$$

  where

$$\alpha_i = \frac{\mathbf{p}_i^\mathrm{T}\mathbf{r}_i}{\mathbf{p}_i^\mathrm{T}\mathbf{K}\mathbf{p}_i} = \frac{\mathbf{r}_i^\mathrm{T}\mathbf{M}_\mathrm{P}^{-1}\mathbf{r}_i}{\mathbf{p}_i^\mathrm{T}\mathbf{K}\mathbf{p}_i}. \tag{3}$$

  (2) Update residual:

$$\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{K}\mathbf{p}_i. \tag{4}$$

  (3) Compute new search direction:

$$\mathbf{p}_{i+1} = \mathbf{M}_\mathrm{P}^{-1}\mathbf{r}_{i+1} + \beta_i \mathbf{p}_i, \tag{5}$$

  where

$$\beta_i = -\frac{\mathbf{p}_i^\mathrm{T}\mathbf{K}\mathbf{M}_\mathrm{P}^{-1}\mathbf{r}_{i+1}}{\mathbf{p}_i^\mathrm{T}\mathbf{K}\mathbf{p}_i} = \frac{\mathbf{r}_{i+1}^\mathrm{T}\mathbf{M}_\mathrm{P}^{-1}\mathbf{r}_{i+1}}{\mathbf{r}_i^\mathrm{T}\mathbf{M}_\mathrm{P}^{-1}\mathbf{r}_i}. \tag{6}$$

Detailed discussions about the properties and convergence rate of the above PCG algorithm can be found elsewhere (see for instance [4,5]).

2.2. Dynamic relaxation

In structural terms, Eq. (1) represents a static problem. By artificially introducing both a mass matrix **M** and a damping matrix **C**, Eq. (1) can be transformed into a second order dynamic problem:

$$\mathbf{Ma} + \mathbf{Cv} + \mathbf{Ku} = \mathbf{b}, \tag{7}$$

where **a** and **v** are, respectively, the acceleration and velocity of the system; and is subject to (arbitrary) initial conditions:

$$\mathbf{u}(0) = 0; \quad \mathbf{v}(0) = 0.$$

Due to the nature of damping and a constant force applied, the dynamic/transient response of the above system under any given initial conditions will eventually disappear, resulting in only the static solution being retained. This constitutes the main idea of the dynamic relaxation. Numerically an explicit time stepping scheme, mainly the central difference algorithm, is employed to obtain the steady state solution of the system. In order to maximize the convergent rate, several numerical parameters need to be carefully selected.

2.2.1. Standard central difference algorithm with constant time step

In the standard central difference time integration algorithm with a constant time step Δt, both velocity and displacement at each discrete time instance are updated by

$$\mathbf{v}_{i+\frac{1}{2}} = [2\mathbf{M} + \Delta t\mathbf{C}]^{-1}\left[(2\mathbf{M} - \Delta t\mathbf{C})\mathbf{v}_{i-\frac{1}{2}} + 2\Delta t(\mathbf{b} - \mathbf{Ku}_i)\right], \tag{8}$$

$$\mathbf{u}_{i+1} = \mathbf{u}_i + \Delta t\mathbf{v}_{i+\frac{1}{2}}, \tag{9}$$

where $\mathbf{u}_i$ and $\mathbf{u}_{i+1}$ are, respectively, the displacements at times t_i and t_{i+1}, and $\mathbf{v}_{i+\frac{1}{2}}$ is the velocity at the half-time step $t_{i+1/2} = (t_i+t_{i+1})/2$. A common choice in DR is

$$\mathbf{M} = \mathrm{diag}\{\mathbf{K}\}; \quad \mathbf{C} = 2c\mathbf{M}, \tag{10}$$

where c is termed the damping parameter. Thus, (8) can be simplified as

$$\mathbf{v}_{i+\frac{1}{2}} = \frac{1 - c\Delta t}{1 + c\Delta t}\mathbf{v}_{i-\frac{1}{2}} + \frac{\Delta t}{1 + c\Delta t}\mathbf{M}^{-1}\mathbf{r}_i \quad (\mathbf{r}_i = \mathbf{b} - \mathbf{Ku}_i). \tag{11}$$

There are two parameters, Δt and c, involved in the above procedure. In order to achieve best possible convergence rate, the optimum values of these parameters have been established [9,10] as

$$\Delta t_{\mathrm{op}} = \frac{2}{\sqrt{\lambda_{\min} + \lambda_{\max}}}; \qquad c_{\mathrm{op}} = 2\sqrt{\frac{\lambda_{\min}\lambda_{\max}}{\lambda_{\min} + \lambda_{\max}}}, \tag{12}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and maximum eigenvalues of $\mathbf{M}^{-1}\mathbf{K}$. Also note that the above optimum time step is (slightly) less than the critical time step, Δt_{cr}, (the maximum step to ensure a stable time integration):

$$\Delta t_{\mathrm{op}} < \Delta t_{\mathrm{cr}} = 2/\sqrt{\lambda_{\max}}. \tag{13}$$

2.3. Alternative central difference algorithms

Now consider a more general central difference based DR algorithm: (1) time steps $\Delta t_i = t_{i+1} - t_i$ are not constant; and (2) the damping parameter c can be varied. In addition, the following two slightly different discrete dynamic equations, namely the standard and half-step versions, are also considered

$$\text{(Standard)}\qquad \mathbf{M}\mathbf{a}_i + 2c_i\mathbf{M}\mathbf{v}_i + \mathbf{K}\mathbf{u}_i = \mathbf{b}, \tag{14}$$

$$\text{(Half-step)}\qquad \mathbf{M}\mathbf{a}_i + 2c_i\mathbf{M}\mathbf{v}_{i-1/2} + \mathbf{K}\mathbf{u}_i = \mathbf{b}. \tag{15}$$

The main difference between them is that the velocity in the damping term in the half-step version is lagged behind by half a time step. The corresponding formulas for updating the velocity are, respectively, [12]:

$$\text{(Standard)}\qquad \mathbf{v}_{i+\frac{1}{2}} = \frac{\Delta\bar{t}_i}{1+c_i\Delta t_i}\mathbf{M}^{-1}\mathbf{r}_i + \frac{1-c_i\Delta t_{i-1}}{1+c_i\Delta t_i}\mathbf{v}_{i-\frac{1}{2}}, \tag{16}$$

$$\text{(Half-step)}\qquad \mathbf{v}_{i+\frac{1}{2}} = \Delta\bar{t}_i\mathbf{M}^{-1}\mathbf{r}_i + (1-2c_i\Delta\bar{t}_i)\mathbf{v}_{i-\frac{1}{2}}, \tag{17}$$

where $\Delta\bar{t}_i$ is the half-time step, defined as $\Delta\bar{t}_i = t_{i+1/2} - t_{i-1/2} = \frac{1}{2}(\Delta t_i + \Delta t_{i-1})$. The displacement is updated in both versions by

$$\mathbf{u}_{i+1} = \mathbf{u}_i + \Delta t_i\mathbf{v}_{i+\frac{1}{2}}. \tag{18}$$

In the above two algorithms, there are two parameters, Δt_i and c_i, to be determined at each time step. Without a practical means to determine the parameters, these two algorithms would have only theoretical values but without practical significance. It is shown in what follows, however, that Δt_i and c_i can be specially chosen so that the two versions of the variable time step central difference algorithm become identical to the CG algorithm.

2.4. Equivalence between the CG method and the CD algorithms

The time marching formulas for the above two central difference algorithms, the standard and half-time, can be re-arranged as:

$$\text{(Standard)}\qquad \begin{cases} \mathbf{v}^*_{i+\frac{1}{2}} = \mathbf{M}^{-1}\mathbf{r}_i + \frac{1-c_i\Delta t_{i-1}}{\gamma_{i-1}\Delta\bar{t}_i}\mathbf{v}^*_{i-\frac{1}{2}}, \\ \mathbf{u}_{i+1} = \mathbf{u}_i + \frac{\Delta\bar{t}_i\Delta t_i}{1+c_i\Delta t_i}\mathbf{v}^*_{i+\frac{1}{2}}, \end{cases} \tag{19}$$

where $\mathbf{v}^*_{i+\frac{1}{2}} = \gamma_i\mathbf{v}_{i+\frac{1}{2}}$ and $\gamma_i = (1+c_i\Delta t_i)/\Delta\bar{t}_i$; and

$$\text{(Half-step)}\qquad \begin{cases} \mathbf{v}^*_{i+\frac{1}{2}} = \mathbf{M}^{-1}\mathbf{r}_i + \frac{1-2c_i\Delta\bar{t}_i}{\Delta\bar{t}_i}\mathbf{v}^*_{i-\frac{1}{2}}, \\ \mathbf{u}_{i+1} = \mathbf{u}_i + \Delta\bar{t}_i\Delta t_i\mathbf{v}^*_{i+\frac{1}{2}}, \end{cases} \tag{20}$$

where $\mathbf{v}^*_{i+\frac{1}{2}} = \mathbf{v}_{i+\frac{1}{2}}/\Delta\bar{t}_i$.

By comparing (19) and (20) with (2) and (4) of the PCG method, it is obvious that the three algorithms will be identical if the following conditions are satisfied:

$$\mathbf{M}_\mathrm{P} = \mathbf{M}, \tag{21}$$

$$\mathbf{p}_i = \mathbf{v}^*_{i+\frac{1}{2}}, \tag{22}$$

$$\text{(Standard)}\qquad \beta_{i-1} = \frac{1-c_i\Delta t_{i-1}}{\gamma_{i-1}\Delta\bar{t}_i};\quad \alpha_i = \frac{\Delta\bar{t}_i\Delta t_i}{1+c_i\Delta t_i}, \tag{23}$$

$$\text{(Half-step)}\qquad \beta_{i-1} = \left(\frac{1}{\Delta\bar{t}_i} - 2c_i\right)\Delta\bar{t}_{i-1};\quad \alpha_i = \Delta\bar{t}_i\Delta t_i. \tag{24}$$

It follows from (23) and (24) that Δt_i and c_i can be expressed by β_{i-1} and α_i in a recursive form as

$$\text{(Standard)} \qquad \begin{cases} \Delta t_i = \frac{2\alpha_i}{\Delta t_{i-1}+\alpha_i\beta_{i-1}\gamma_{i-1}}, \\ c_i = \frac{1-\beta_{i-1}\gamma_{i-1}\Delta\bar{t}_i}{\Delta t_{i-1}}, \end{cases} \quad (i = 1, 2, \ldots), \tag{25}$$

$$\text{(Half-step)} \qquad \begin{cases} \Delta t_i = -\frac{\Delta t_{i-1}}{2} + \sqrt{\left(\frac{\Delta t_{i-1}}{2}\right)^2 + 2\alpha_i}, \\ 2c_i = \frac{1}{\Delta\bar{t}_i} - \frac{\beta_{i-1}}{\Delta\bar{t}_{i-1}}, \end{cases} \quad (i = 1, 2, \ldots). \tag{26}$$

In both cases, the values for Δt_0 and c_0 are

$$\Delta t_0 = \sqrt{2\alpha_0}; \qquad c_0 = 0.$$

It is evident that for the half-step version, Δt_i are determined by α_i, while in the standard version, they are also affected by β_{i-1}.

With the equivalence between the procedures established above, the following remarks can thus be made:

- The preconditioned CG method can be viewed as a special central difference time integration scheme for a damped dynamic system;
- The preconditioning matrix $\mathbf{M}_\mathrm{P}$ in PCG serves as the mass matrix in the dynamic system, while for the standard CG method, $\mathbf{M} = \mathbf{I}$ (the identity matrix);
- The search directions $\mathbf{p}_i$ in the (P)CG algorithm are closely associated with the velocities in the central difference procedure;
- The variable nature of α_i and β_i ensures that both Δt_i and c_i will be different at each step. Additional features of Δt_i and c_i are further illustrated by two numerical examples below.

3. Numerical illustrations

Example 1. First consider $\mathbf{K}$ to be a diagonal matrix of order $n = 100$:

$$\mathbf{K} = \mathrm{diag}\{a_1, \ldots, a_n\},$$

where $a_1 = 1$, $a_n = 10^3$, and the rest terms are randomly chosen with a uniform distribution in the range $[a_1, a_n]$. A higher condition number can be achieved by giving a larger value of a_n. The right hand side $\mathbf{b}$ is also generated randomly within the range $[-1, 1]$. No preconditioning is employed for the CG algorithm, i.e., $\mathbf{M} = M_\mathrm{P} = \mathbf{I}$. Although this simple example cannot comprehensively represent a wide range of problems encountered in practice, it can highlight certain dynamic features of the CG algorithm.

The convergence history of the CG iterations, in terms of relative residual norm

$$e_i = \sqrt{\mathbf{r}_i^\mathrm{T}\mathbf{r}_i}\Big/\sqrt{\mathbf{b}^\mathrm{T}\mathbf{b}}$$

is depicted in Fig. 1(a). For comparison, the convergence of the standard DR procedure with the optimum parameters Δt_op and c_op (referring to (12)) is also given in the figure. The superiority of the CG algorithm, in terms of convergence, to the DR approach is clearly demonstrated at least in this case.

Define, respectively, two types of energy, 'kinetic' and 'elastic', of the system at any iteration as

$$E_\mathrm{k} = \frac{1}{2}\mathbf{v}_{i+1/2}^\mathrm{T}\mathbf{M}\mathbf{v}_{i+1/2}; \quad E_\mathrm{e} = \frac{1}{2}\mathbf{u}_i^\mathrm{T}\mathbf{K}\mathbf{u}_i.$$

The evolution of these two energy functions for both the CG and DR iterations is given in Fig. 1(b).

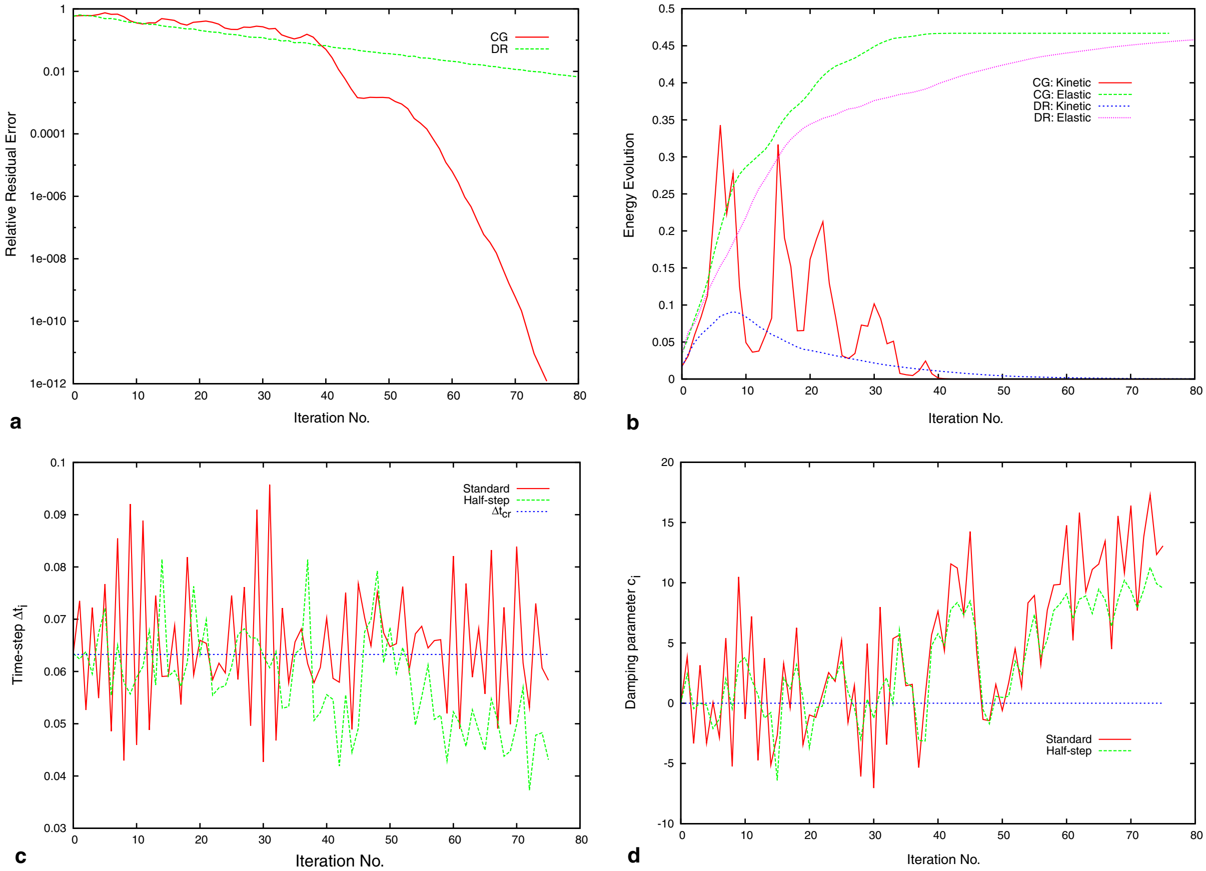

Fig. 1. Example 1: (a) convergence histories; (b) energy evolution; (c) time-step Δt_i; and (d) damping coefficient c_i.

Δt_i and c_i computed on the basis of (25) and (26) are shown, respectively, in Figs. 1(c) and (d). It is evident that: (1) Δt_i are different at each step, and sometimes exceed the time step limit Δt_{cr}; (2) The damping parameter c_i are varied and become negative at some iterations; (3) The oscillation of both Δt_i and c_i is stronger in the standard version than the half-step version, particularly for Δt_i, while the evolution patterns of c_i in the two versions are fairly similar; (4) When the CG iterative procedure starts to converge, Δt_i and c_i tend to oscillate less in the half-time version. However, it is found that the oscillation of both Δt_i and c_i in the standard version is much stronger and also persistent for larger scale problems; while the evolution patterns of Δt_i and c_i in the half-step version appear to be consistent for all cases. In particular c_i in the half-step version would be negative at iterations where the residual errors are larger than at the previous steps and become positive when a monotonic decrease of the residual error is established.

In the context of a dynamic system, the behaviour of the CG procedure may be revealed by the evolution of E_{k} and E_{e} and how these two types of energy exchange and transfer during the whole course of the iterations. From the convergence point of view, as both energies are initially zero (since $\mathbf{u}_0 = 0$ and $\mathbf{v}_0 = 0$), the kinetic energy should be increased sufficiently fast at the initial stage so that a required level of elastic energy can be rapidly achieved, while at the later stage the kinetic energy should be effectively damped out. The mechanism that governs the evolution and exchange of these two types of energy may determine the convergence of the CG algorithm.

As illustrated in Fig. 1(b), the CG iterative procedure is characterized by a strong oscillation of the kinetic energy prior to the establishment of convergence, in contrast to a fairly smooth convergence of the elastic energy. It appears that significant amount of kinetic energy is gained mainly at those iterations where the damping is negative, and may also be attributed, to a lesser degree, to those time steps larger than Δt_{cr}. At these iterations a high level of the residual error is also observed (referring to Fig. 1(a)). Note that the peak values of the kinetic energy can be much more pronounced for higher condition numbers. On the contrary, the evolution of the kinetic energy in the DR procedure, as also given in the figure, is very smooth, but its decay (and the convergence of the elastic energy) is much slower.

Example 2. The dynamic features of PCG observed above and the effects of different preconditioning techniques are further examined by a practical structural example. It consists of finite element modelling of a 3D elastic bridge with 10-noded tetrahedral elements. The geometric and material details of the example can be found in [13]. The resulting linear system of equations with an order of 88,622 is solved by PCG using two preconditioners: the no-fill Incomplete Cholesky (IC(0)) decomposition and the diagonal (Jacobi) preconditioning (DP).

Again, the convergence history of the relative residual error, the evolution of both kinetic and elastic energy, and the corresponding Δt_i and c_i at each iteration version are illustrated in Fig. 2, where the dynamic interpolation of the PCG iterations is based on the half-step version. A similarity to Fig. 1 (Example

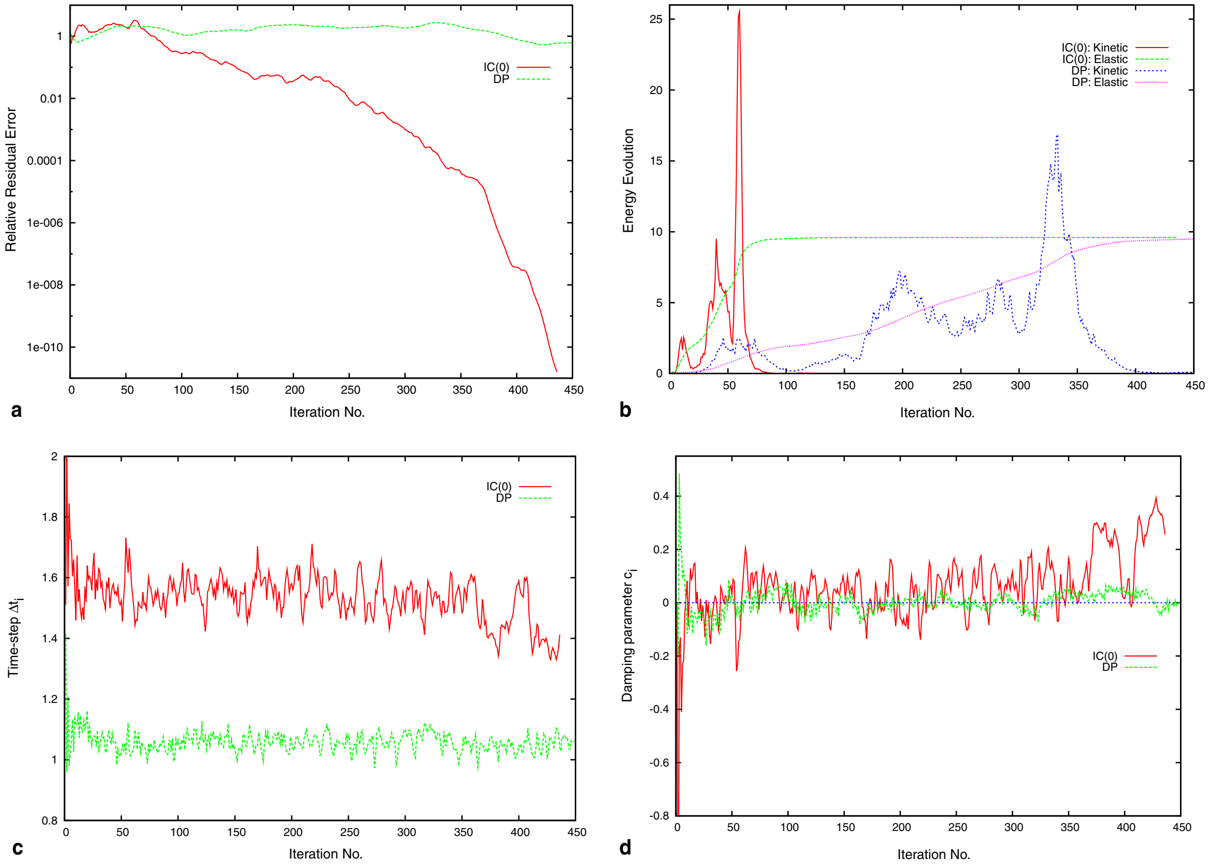

Fig. 2. Example 2: (a) convergence histories; (b) energy evolution; (c) time-step Δt_i; and (d) damping coefficient c_i.

1) in terms of the dynamic features for both preconditioning schemes is clearly demonstrated. It is also illustrated that compared to the DP scheme, the IC(0) preconditioner results in larger time steps Δt_i, generally larger and a wider range of damping parameters c_i, and a faster and stronger evolution of the kinetic energy, leading to a better convergence.

4. Concluding remarks

The current work has established the fact that the preconditioned CG method can be viewed as a central difference time integration scheme with a variable time step for a damped dynamic system, and therefore possesses a discrete dynamic nature. Although its immediate practical benefit is not clear at present, this new connection may provide an additional viewpoint to gain a further understanding of the behaviour of the PCG method, and might also be possible to extend to other Krylov based iterative solvers in general. In addition, this 'dynamic' interpretation might shed new light on problems such as the understanding of the short time convergence behaviour of most Krylov methods and the development of an optimal evolution scheme for a 'flexible' or dynamically changing preconditioning technique. On the other hand, it may also offer a possibility of developing a new scheme for the selection of the optimal parameters in the DR technique. These issues are clearly of both theoretical and practical importance and thus deserve further investigations.

Acknowledgements

The author thank the two reviewers for their constructive comments to improve this paper, and particularly for their suggestions for possible further work.

References

[1] M.R. Hestenes, E.L. Stiefel, Methods of conjugate gradients for solving linear systems, J. Natl. Bur. Stand 49 (1952) 409–436.

[2] P. Sonneveld, CGS: a fast Lanczos-type solver for nonsymmetric linear systems, SIAM J. Sci. Stat. Comp. 10 (1989) 36–52.

[3] H. van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear equations, SIAM J. Sci. Stat. Comp. 13 (1992) 631–644.

[4] E. Polak, Computational Methods in Optimization: A Unified Approach, Academic Press, New York, 1971.

[5] M.A. Wolfe, Numerical Methods for Unconstrained Optimization, An Introduction, Von Nostrand Reinhold Company, Berkshire, England, 1978.

[6] Y.T. Feng, D.R.J. Owen, Conjugate gradient methods for solving the smallest eigenpair of large symmetric eigenvalue problems, Int. J. Numer. Eng. 39 (13) (1996) 2209–2229.

[7] J. Meijerink, H. van der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, Math. Comp. 31 (1977) 148–162.

[8] M. Benzi, Preconditioning techniques for large linear systems: a survey, J. Comput. Phys. 182 (2002) 418–477.

[9] R.D. Lynch, S. Kelsey, H.C. Saxe, The application of DR to the finite element method of structural analysis. Technical Report No. THEMIS-UND-68-1, Univ. Notre Dame, 1968.

[10] M. Papadrakakis, A method for the automatic evaluation of the dynamic relaxation parameters, Comp. Meth. Appl. Mech. Eng. 25 (1981) 35–48.

[11] R.S. Varga, Matrix Iterative Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1962.

[12] T. Belytschko, W.K. Liu, B. Moran, Nonlinear Finite Elements for Continua and Structures, Wiley, Chichester, England, 2000.

[13] Y.T. Feng, D. Perić, D.R.J. Owen, A non-nested Galerkin multi-grid method for solving linear and nonlinear solid mechanics problems, Comp. Meth. Appl. Mech. Eng. 144 (1997) 307–325.