Normalized implicit preconditioned methods based on normalized finite element approximate factorization procedures

George A. Gravvanis*, Konstantinos M. Giannoutakis

Department of Electrical and Computer Engineering, School of Engineering, Democritus University of Thrace, GR 67100 Xanthi, Greece

Abstract

Normalized implicit preconditioned conjugate gradient-type schemes based on finite element normalized approximate factorization procedures for solving sparse linear systems, which are derived from the finite element method of partial differential equations in three space variables, are presented. Theoretical estimates on the rate of convergence and computational complexity of the normalized implicit preconditioned conjugate gradient method are also given. The application of the proposed method on a characteristic three-dimensional boundary value problem is discussed and numerical results are given.

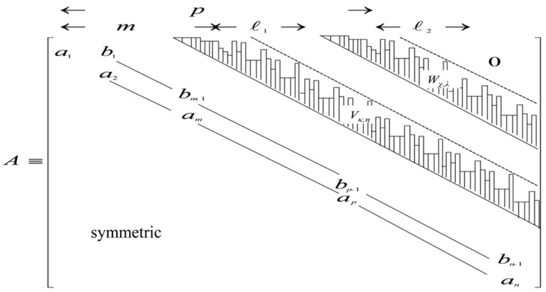
Keywords: Finite element systems; Normalized approximate factorization procedures; Preconditioning; Conjugate gradient schemes; Rate of convergence and computational complexity

1. Introduction

Let us consider the linear system resulting from the finite element (FE) discretization of an elliptic boundary value problem in three dimensions, i.e.

$$Au = s \tag{1}$$

where A is a non-singular large sparse symmetric positive definite, diagonally dominant matrix of irregular structure (where all the off-center band terms are grouped in regular bands of width ℓ_1 and ℓ_2 at semi-bandwidths m and p), viz.



(2)

while u is the FE solution and s is a vector, of which the components result from a combination of source terms and imposed boundary conditions.

Then, by using normalized implicit preconditioned methods based on normalized approximate factorization procedures, the FE solution can be obtained as the limit of a convergent sequence of vectors $\{u_{i+1}\}$ generated by the normalized implicit preconditioned conjugate gradient (NIPCG) method.

Finally, the performance and applicability of the normalized implicit preconditioned methods are illustrated by solving a characteristic elliptic boundary value problem and numerical results are given.

2. Normalized approximate factorization procedures

Let us now assume the normalized approximate factorization such that:

$$A \approx D_{r_1,r_2} T^t_{r_1,r_2} T_{r_1,r_2} D_{r_1,r_2},\ r_1 \in [1,\ldots,m-1),\quad r_2 \in [1,\ldots,p-1) \tag{3}$$

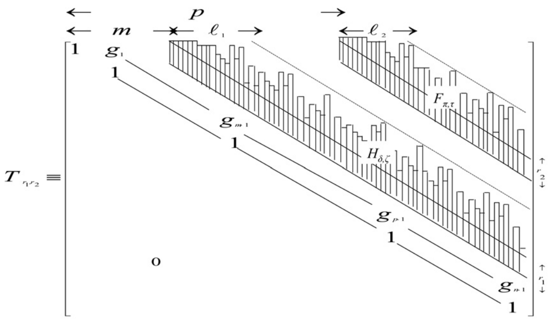
where r_1 and r_2 are the 'fill-in' parameters, i.e. the number of outermost off-diagonal entries retained at semi-bandwidths m and p, D_{r_1,r_2} is a diagonal matrix and T_{r_1,r_2} is a sparse upper (with unit diagonal elements)

* Corresponding author. E-mail: ggravvan@ee.duth.gr

Computational Fluid and Solid Mechanics 2005
K.J. Bathe (Editor)

triangular matrix of the same profile as the coefficient matrix A.

$$D_{r_1,r_2} \equiv diag\left\{d_1,\ldots,d_{m-1} \vdots d_m,\ldots,d_{p-1} \vdots d_p,\ldots,d_n\right\}, \tag{4}$$



(5)

Then, the elements of the decomposition factors D_{r_1,r_2} and T_{r_1,r_2}, were computed by the finite element approximate normalized factorization 3D algorithm (henceforth called the FEANOF-3D algorithm), and can be expressed by the following compact algorithmic scheme:

$$d_1 = \sqrt{a_1} \tag{6}$$

for $i = 2$ to $m-1$

$$d_i = \left(a_i - (b_i - 1/d_{i-1})^2\right)^{1/2} \tag{7}$$

$$g_{i-1} = b_{i-1}/(d_{i-1}d_i) \tag{8}$$

By applying the FEANOF-2D algorithm [1] the elements $h_{\delta,\zeta}$ of the lower submatrix at semi-bandwidth m retaining r_1 outermost-off diagonal entries, the co-diagonal elements g_i and also the diagonal elements $d_m, \ldots, d_{p-1}$ of the diagonal matrix were obtained.

The equations to determine the elements of T_{r_1,r_2} proved to be non-linear and a simple iterative Picard-type scheme was used in an inner loop to determine the values of $d_p, \ldots, d_n$, as the direct solution of these equations proved to be intractable [1].

Find the first non-zero element of submatrix $W_{\chi,\lambda}$ of the $(p+j-1)$th column of A (the procedure IXNOS can be used [1]).

for $j = 1$ to $n - m + 1$

if $j - p + m < \ell_2$ then

$$f_{\chi j} = w_{\chi\lambda}/\left(d_k d_{m+j-1}\right) \tag{9}$$

else

$$f_{\chi+\ell_2-j+p-m,j-p+m} = w_{\chi\lambda}/\left(d_k d_{m+j-1}\right) \tag{10}$$

if $j - p + m \le r_2 + \ell_2 - 2$ then

if $j - p + m \le \ell_2$ then

$$f_{i,j-p+m} = -d_{i-1}f_{i-1,j-p+m} \tag{11}$$

if $i < j - p + m + 1$ then

$$f_{i,j-p+m} = f_{i,j-p+m} - w_{i,j-p+m-i+1}/\left(d_i d_{m+j-1}\right) \tag{12}$$

for $i = \chi + 1, \ldots, r_2 + j - p + m - 1$

else

$$f_{i,j-p+m} = -g_{i+j-p+m-\ell_2-1}f_{i-1,j-p+m} \tag{13}$$

if $i \le \ell_2$ then

$$f_{i,j-p+m} = f_{i,j-p+m} - w_{i+j-p+m-\ell_2,\ell_2-i+1}/\left(d_i d_{m+j-1}\right) \tag{14}$$

for $i = \chi + \ell_2 - j + p - m + 1, \ldots, r_2 + \ell_2 - j + p - m + 1$

if $j - p + m > \ell_2$ then

if $i \ge p + 2\ell_2 - j + p - m$ and $j - p + m \ge \ell_2 + 2$ then

$$f_{i,j-p+m} = -g_{i+j-p+m-\ell_2-1}f_{i-1,j-p+m} - \sum_{k=1}^{i-1} h_{k-i+r_2+\ell_2,i+j-p+m-r_2-\ell_2}f_{k,j-p+m} \tag{15}$$

else

$$f_{i,j-p+m} = -g_{i+j-p+m-\ell_2-1}f_{i-1,j-p+m} - \sum_{k=1}^{i-1} h_{k+j-p+m-\ell_2,i+j-p+m-r_2-\ell_2}f_{k,j-p+m} \tag{16}$$

then, if $i \le \ell_2$

$$f_{i,j-p+m} = f_{i,j-p+m} + w_{i+j-p+m-\ell_2,\ell_2-i+1}/\left(d_{i+j-p+m-\ell_2}d_{m+j-1}\right) \tag{17}$$

for either $i = r_2 + \ell_2 - j + 1 + p - m, \ldots, r_2 + \ell_2 - 1$ and $\chi < p$

or $i = \chi + \ell_2 - j + p - m + 1, \ldots, r_2 + \ell_2 - 1$ and $\chi \ge p$, with $j < r_2 + \ell_2 - 1$

or $i = \chi + \ell_2 - j + p - m + 1, \ldots, r_2 + \ell_2 - 1$ for all $j - p + m \ge r_2 + \ell_2 - 1$.

Then, for $i = r_2$

if $j - p + m \le \ell_2$ then

$$d_{m+j-1} = \sqrt{w_{m+j-1}} \Bigg/ \sqrt{1 + \sum_{k=1}^{r_1+\ell_1-1} h_{k,j}^2 + \sum_{k=1}^{r_2+j-p+m-1} f_{k,j-p+m}^2} \tag{18}$$

else

$$d_{m+j-1} = \sqrt{w_{m+j-1}} \Bigg/ \sqrt{1 + \sum_{k=1}^{r_1+\ell_1-1} h_{k,j}^2 + \sum_{k=1}^{r_2+\ell_2-1} f_{k,j-p+m}^2} \tag{19}$$

The memory requirements of the FEANOF-3D algorithm is $\approx (r_1+r_2+2\ell_1+2\ell_2+4)n$ words, while the computational work required is $\approx O\left[(r_1+\ell_1)^2+(r_2+\ell_2)^2\right]n$ multiplicative operations and n square roots.

It should be noted that if the width parameter $\ell_2 = 0$, then the algorithm reduces to one for solving sparse linear systems which are encountered in solving 2D boundary value problems by the finite element method [1]. If the width-parameters $\ell_1 = 1$ and $\ell_2 = 1$, then the algorithm reduces to one for solving sparse linear systems which are encountered in solving 3D boundary value problems by the finite difference method [2]. If the width parameters $\ell_1 = 1$ and $\ell_2 = 0$, then the algorithm reduces to one for solving sparse linear systems which are encountered in solving 2D boundary value problems by the finite difference method [3].

3. Normalized implicit preconditioned conjugate gradient methods

In this section we present normalized implicit preconditioned conjugate gradient methods, based on normalized approximate factorization procedures and theoretical estimates on the rate of convergence and the computational complexity are derived [2,4].

The normalized implicit preconditioned conjugate gradient (NIPCG) method for solving linear systems can be stated as follows:

Let u_0 be an arbitrary initial approximation to the solution vector u. Then,

$$\text{form} \quad \tilde{r}_0 = s - Au_0 \tag{20}$$

$$\text{solve} \quad \left(T_{r_1,r_2}^t\, T_{r_1,r_2}\right) D_{r_1,r_2}\, r_0^* = D_{r_1,r_2}^{-1}\, \tilde{r}_0 \tag{21}$$

$$\text{set} \quad \tilde{\sigma}_0 = r_0^* \tag{22}$$

Then, for $i = 0, 1, \ldots$, (until convergence) compute the vectors u_{i+1}, $\tilde{r}_{i+1}$, $\tilde{\sigma}_{i+1}$ and the scalar quantities $\tilde{a}_i$, $\tilde{\beta}_{i+1}$ as follows:

$$\text{form} \quad q_i = A\tilde{\sigma}_i, \quad \text{and} \quad p_i = \left(\tilde{r}_i,\ r_i^*\right), \quad \text{when } (i=0) \tag{23}$$

$$\text{evaluate} \quad \tilde{a}_i = p_i/(\tilde{\sigma}_i, q_i),\ u_{i+1} = u_i + \tilde{a}_i\tilde{\sigma}_i, \text{ and}$$
$$\tilde{r}_{i+1} = \tilde{r}_i - \tilde{a}_i\tilde{q}_i \tag{24}$$

$$\text{Then, solve} \quad \left(T_{r_1,r_2}^t\, T_{r_1,r_2}\right) D_{r_1,r_2}\, r_{i+1}^* = D_{r_1,r_2}^{-1}\, \tilde{r}_{i+1} \tag{25}$$

$$\text{compute} \quad p_{i+1} = \left(\tilde{r}_{i+1},\ r_{i+1}^*\right),\ \beta_{i+1} = p_{i+1} \Big/ p_i \text{ and}$$
$$\tilde{\sigma}_{i+1} = r_{i+1}^* + \tilde{\beta}_{i+1}\tilde{\sigma}_i \tag{26}$$

The normalized implicit preconditioned conjugate gradient square (NIPCGS) method and the normalized implicit preconditioned biconjugate conjugate gradient-STAB (NIPBICG-STAB) method can be similarly derived [2].

The basic properties of the NIPCG method, Notay [5], can be stated by the following theorem:

Theorem 3.1: Let A be a positive definite matrix, s some given vector and $u = A^{-1}s$ the solution to the linear system (1). Let us also consider that $u^{(k)}$ is the solution vector after k iterations of the NIPCG method [5] then we have

$$\left\|u - u^{(k)}\right\|_B = \min_{P_k} \left\| P_k\left(\Phi_{r_1,r_2}\right)\left(u - u^{(0)}\right)\right\|_B \tag{27}$$

$$\text{and} \quad \frac{\left\|u - u^{(k)}\right\|_B}{\left\|u - u^{(0)}\right\|_B} = \min_{P_k} \max_{x\in[\lambda_{min},\lambda_{max}]} |P_k(x)|, \tag{28}$$

$$\text{where } \Phi_{r_1,r_2} = \left(D_{r_1,r_2}\, T_{r_1,r_2}^t\, T_{r_1,r_2}\, D_{r_1,r_2}\right)^{-1} A \tag{29}$$

while P_k is the set of polynomials of degree k such that $P_k(0) = 1$, and $\|x\|_B = (x, B)^{1/2}x$.

Since $e^{(k)} = u - u^{(k)}$, it can be proved [5] that:

$$\left\|e^{(k)}\right\|_B \leq \min_{P_k} \max_{x\in[\lambda_{min},\lambda_{max}]} |P_k(x)| \left\|e^{(0)}\right\|_B \tag{30}$$

where the polynomial $P_k(x)$ can be defined by the Chebyshev polynomials in order to derive an estimate of the number of iterations required for the interval $[\lambda_{\min}, \lambda_{\max}]$. Let us assume through this section that:

$$\left\|\Phi_{r_1,r_2}\right\| \equiv \left\|\Phi_{r_1,r_2}\right\|_1 = \max_{j\in[1,n]} \left\{ \sum_{i=1}^{n} \left| \left(\Phi_{r_1,r_2}\right)_{i,j} \right| \right\} \tag{31}$$

The following Lemma allows us to establish bounds for the preconditioning matrix Φ_{r_1,r_2}.

Lemma 3.1: Let $\Phi_{r_1,r_2} = (D_{r_1,r_2}\, T_{r_1,r_2}^t\, T_{r_1,r_2}\, D_{r_1,r_2})^{-1} A$ be the preconditioning matrix of the NIPCG iterative scheme. Let m and p be the semi-bandwidths of the coefficient matrix A, ℓ_1 and ℓ_2 be the width-parameters at semi-bandwidth m and p respectively and r_1, r_2 be the 'fill-in' parameters. Then

$$\frac{1}{\Lambda_0}\frac{1}{1 + C_1^{(r_1,r_2,\ell_1,\ell_2)}} \leq \left\|\Phi_{r_1,r_2}\right\| \leq -1 + 2C_2^{(\Lambda,m,p,r_1,r_2,\ell_1,\ell_2)} \tag{32}$$

where $C_2^{(\Lambda,m,p,r_1,r_2,\ell_1,\ell_2)}$ is a constant depending on Λ, m, p, r_1, r_2, ℓ_1, ℓ_2 and strictly greater than unity, $C_1^{(r_1,r_2,\ell_1,\ell_2)}$ is a constant depending on r_1, r_2 and independent of the mesh-size and $\Lambda_0 = M_0^2/n^2$, with M_0 the M-condition number of the matrix $A = DT^tTD$.

Let us consider the positive numbers ξ_1 and ξ_2, where $[\xi_1, \xi_2] \supset [\lambda_{\min}, \lambda_{\max}]$, such that:

$$\xi_2 \leq -1 + 2C_2^{(\Lambda,m,p,r_1,r_2,\ell_1,\ell_2)} \quad \text{and} \quad \xi_1 \geq \frac{1}{\Lambda_\Psi \Lambda_0} \frac{1}{1 + C_1^{(r_1,r_2,\ell_1,\ell_2)}} \tag{33}$$

Then assuming that a uniform volumetric network of mesh size h is superimposed over a cube, similarly it can be proved that [2]:

$$0\langle \ \|\Psi_{r_1,r_2}\| \leq -1 + 2\Lambda\sigma\left\{\frac{m+\ell_1-2}{r_1+\ell_1-1} + \frac{p+\ell_2-2}{r_2+\ell_2-1} - 1\right\} + q \tag{34}$$

hence

$$\xi_2 = \lambda_{max}\ (\Phi_{r_1,r_2}) \leq \ \|\Phi_{r_1,r_2}\| \leq -1 + 2\Lambda\left\{\frac{m+\ell_1-2}{r_1+\ell_1-1} + \frac{p+\ell_2-2}{r_2+\ell_2-1}\right\} + q \tag{35}$$

which can be equivalently written as:

$$\xi_2 = O\ \lfloor (r_1+\ell_1-1)^{-1}(m+\ell_1-2) + (r_2+\ell_2-1)^{-1}(p+\ell_2-2) \rfloor \tag{36}$$

It should be noted that the behavior of the values of $C_1^{(r_1,r_2,\ell_1,\ell_2)}$ and $C_2^{(\Lambda,m,p,r_1,r_2,\ell_1,\ell_2)}$ are closely related to the eigenvalue distribution of the 3D-model problem (the Laplace equation in the unit cube with zero boundary values) as $r_1 \to m-1$ and $r_2 \to p-1$.

Then the following Theorem on the rate of convergence and computational complexity of the NIPCG method can be stated:

Theorem 3.2: Let be the $\Phi_{r_1,r_2} = (D_{r_1,r_2}\ T^t_{r_1,r_2}\ T_{r_1,r_2}\ D_{r_1,r_2})^{-1} A$ preconditioning matrix of the NIPCG iterative scheme, where r_1 and r_2 are the 'fill-in' parameters. Suppose there exist positive numbers ξ_1 and ξ_2, where ξ_1 is independent of the mesh size and $\xi_2 = O[(r_1 + \ell_1 - 1)^{-1}(m + \ell_1 - 2) + (r_2 + \ell_2 - 1)^{-1}(p + \ell_2 - 2)]$. Then the number of iterations of the NIPCG method required to reduce the L_∞-norm of the error by a factor $\varepsilon > 0$ is given by:

$$v = O\left[\left\{(r_1+\ell_1-1)^{-1}(m+\ell_1-2) + (r_2+\ell_2-1)^{-1}(p+\ell_2-2)\right\}^{1/2} \log\varepsilon^{-1}\right] \tag{37}$$

Furthermore, the computational complexity for the computation of the solution u_ν is given by:

$$O\left[h^{-3}(r_1+\ell_1)(r_2+\ell_2)\left\{(r_1+\ell_1-1)^{-1}(m+\ell_1-2) + (r_2+\ell_2-1)^{-1}(p+\ell_2-2)\right\}^{1/2} \log\varepsilon^{-1}\right] \tag{38}$$

Additionally, it has been observed that the NIPCGS and NIPBICG-STAB method converges roughly twice as fast as the NIPCG method.

Table 1
The convergence behavior of the NIPCGS and NIPBICG-STAB method

Method	n	m	p	r_1,r_2		
				1	**2**	**4**
NIPCGS	729	10	82	5	5	5
	2744	15	197	6	6	6
	6859	20	362	6	6	6
	13824	25	577	6	6	6
	24389	30	842	6	6	6
	59319	40	1522	6	6	6
	205379	60	3482	6	6	6
NIPBICG-STAB	729	10	82	5	5	5
	2744	15	197	6	6	6
	6859	20	362	6	6	6
	13824	25	577	6	6	6
	24389	30	842	6	6	6
	59319	40	1522	6	6	6
	205379	60	3482	6	6	6

4. Numerical results

In this section we examine the applicability and effectiveness of the proposed schemes for solving characteristic 3D-boundary value problems.

Let us consider a 3D-boundary value problem with Dirichlet boundary conditions:

$$u_{xx} + u_{yy} + u_{zz} + u = F, \quad (x,y,z) \in R \tag{39}$$

where R is the unit cube. The domain is covered by a non-overlapping triangular network resulting in a hexagonal mesh. The right-hand side vector of the system (1) was computed as the product of the matrix A by the solution vector, with its components equal to unity. The iterative process was terminated when $\|r_i\|_\infty \langle 10^{-6}$.

The convergence behavior of the NIPCGS and NIPBICG-STAB methods for several values of the order n, semi-bandwidths m, p and the 'fill-in' parameters r_1, r_2 is presented in Table 1. The performance of the FEANOF-3D algorithm and of the NIPCGS and NIPBICG-STAB method (given in h:m:s.h) for various values of the 'fill-in' parameters r_1, r_2 is presented in Table 2.

It should be mentioned that the theoretical results

Table 2
The performance of the FEANOF-3D algorithm and of the NIPCGS and NIPBICG-STAB schemes

n	**Method**	r_1,r_2		
		1	**2**	**4**
59319	**FEANOF-3D**	0:0:00.18	0:0:00.22	0:0:00.31
	NIPCGS	0:6:38.31	0:6:39.43	0:6:42.25
	NIPBICG-STAB	0:9:04.92	0:9:05.43	0:9:15.22
205379	**FEANOF-3D**	0:0:00.56	0:0:00.69	0:0:00.98
	NIPCGS	1:13:10.64	1:14:02.60	1:14:41.77
	NIPBICG-STAB	1:50:29.93	1:50:46.27	1:50:55.14

obtained were found to be in qualitative agreement with the numerical results presented.

It is evident that the forward–backward substitution is responsible for such performance of the normalized implicit preconditioned conjugate gradient-type methods. In order to overcome such inefficiencies in terms of performance, the finite element normalized explicit approximate inverse preconditioning should be exploited.

References

[1] Lipitakis EA, Evans DJ. Solving linear finite element systems by normalized approximate matrix factorization semi-direct methods. Comp Meth App Mech Eng 1984;43:1–19.

[2] Gravvanis GA, Giannoutakis KM. On the rate of convergence and complexity of normalized implicit preconditioning for solving finite difference equations in three space variables. Int J Comput Meth 2004;1(2):367–386.

[3] Evans DJ, Lipitakis EA. A normalized implicit conjugate gradient method for the solution of large sparse systems of linear equations. Comp Meth Appl Mech Eng 1980;23:1–19.

[4] Lipitakis EA, Gravvanis GA. The convergence rate and complexity of preconditioned iterative methods based on approximate finite element matrix factorization. Bull Greek Math. Society 1992;33:11–23.

[5] Notay Y. On the convergence rate of the conjugate gradients in presence of rounding errors. Numer Math 1993;65:301–317.