REAL VALUED ITERATIVE METHODS FOR
SOLVING COMPLEX LINEAR SYSTEMS

Owe Axelsson, Andrey Kucherov

Report No. 9904 (January 1999)
Real Valued Iterative Methods for Solving Complex Linear Systems

Owe Axelsson*, Andrey B. Kucherov†

Abstract

Complex valued systems of equations with a matrix $R + iS$ where $R$ and $S$ are real valued arise in many applications. A preconditioned iterative solution method is presented when $R$ and $S$ are symmetric positive semi-definite and at least one of $R, S$ is positive definite. The condition number of the preconditioned matrix is bounded above by 2, so only few iterations are required. Applications when solving matrix polynomial equation systems, linear systems of ordinary differential equations, and using time-stepping integration schemes based on Padé approximation for parabolic and hyperbolic problems are also discussed.

Key words: complex matrices, matrix polynomial equation, Cauchy problem, parabolic equation, hyperbolic equation, preconditioned iterative methods

1 Introduction

Complex valued systems arise in many applications. A common case is the solution of a matrix polynomial equation

(1.1) \[ Q_m(A)x = b \]

where $A$ is a real square matrix and $Q_m$ is a polynomial of degree $m$ which has no zeroes at the eigenvalues of $A$. With no limitation we may assume that the leading coefficient of $Q_m$ is $(-1)^m$. Letting $z_k$ be the zeroes of $Q_m$ we find

\[ \prod_{k=1}^{m} (A - z_k I)x = b, \]

or assuming that $z_k \neq z_l, k \neq l$.

---

*Faculty of Mathematics and Informatics, University of Nijmegen, Toernooiveld, NL-6525 ED Nijmegen
†Department of Computational Mathematics and Cybernetics, Moscow State University, Moscow, 119899, Russia

2
\[ x = \sum_{k=1}^{m} a_k (A - z_k I)^{-1} b \]

where \( a_k = [Q_m(z_k)]^{-1} \). Hence the solution of (1.1) can be computed by solving \( m \) systems

\[ (A - z_k I)y^{(k)} = b, \quad k = 1, 2, \ldots, m \]

in parallel and then forming \( x = \sum a_k y^{(k)} \). In general, \( z_k \) are complex numbers. To avoid complex arithmetic the systems will be reduced to real valued form. The systems in (1.2) have a special form. Consider more generally

\[ (R + iS)(x + iy) = u + iv, \]

where \( R, S \) are real matrices of order \( N \) and \( x, y, u, v \in \mathbb{R}^N \). Normally \( R \) and \( S \) are large but sparse matrices. We assume that \( R \) and \( R + SR^{-1}S \) are nonsingular. This holds, in particular, if \( R \) is symmetric and positive definite (s.p.d.) and \( S \) is symmetric. Alternatively, if \( S \) and \( S + RS^{-1}R \) are nonsingular (but \( R \) is possibly singular), we multiply (1.1) by the imaginary unit and consider \((S - iR)(x + iy) = v - iu\).

Using the identity,

\[ (I - iSR^{-1})(R + iS) = R + SR^{-1}S, \]

we find the solution to (1.3)

\[ x + iy = (R + SR^{-1}S)^{-1}(I - iSR^{-1})(u + iv) \]

or

\[ x = (R + SR^{-1}S)^{-1}(u + SR^{-1}v) \]  
\[ y = (R + SR^{-1}S)^{-1}(v - SR^{-1}u). \]

Here we have reduced the system (1.3) to real valued form.

The computation of the vectors \( x, y \) using (1.4a,b) straightforwardly require two solutions of systems with both \( R + SR^{-1}S \) and \( R \) and some additional matrix-vector multiplications and vector operations. In order to reduce the computational effort we note first that (1.3) is equivalent to

\[ \begin{cases} 
Rx - Sy = u \\
Sx + Ry = v.
\end{cases} \]

Hence, having computed \( x \) by (1.4a) we can alternatively compute \( y \) from

\[ Ry = v - Sx, \]
instead of using (1.4b). This saves one solution with a system with \( R + SR^{-1}S \).

In this form the method to solve (1.3) using real valued arithmetic is:

Solve, in order,

\[(1.6a) \quad (R + SR^{-1}S)x = u + SR^{-1}v \]

\[(1.6b) \quad Ry = v - Sx.\]

The method can be generalized in the following way introducing a real parameter \( \alpha \).

Rewrite (1.5) in the form

\[ Rx - Sy = u \]
\[ (R + \alpha S)y - (\alpha R - S)x = v - \alpha u. \]

Eliminating \( y \) we find

\[ Rx - S(R + \alpha S)^{-1}(\alpha R - S)x = u + S(R + \alpha S)^{-1}(v - \alpha u). \]

Using the identity

\[ \alpha R - S = \alpha(R + \alpha S) - (1 + \alpha^2)S, \]

we obtain finally

\[(1.7a) \quad G_\alpha x = u + S(R + \alpha S)^{-1}(v - \alpha u), \]

where

\[(1.7b) \quad G_\alpha = R - \alpha S + (1 + \alpha^2)S(R + \alpha S)^{-1}S. \]

We assume here that \( R + \alpha S \) and \( G_\alpha \) are nonsingular.

If \( \alpha = 0 \), (1.7a) reduces to (1.6a). Evidently, (1.7a) can be useful if \( R \) is ill-conditioned (or even singular) but \( S \) is such that \( R + \alpha S \) is better conditioned for some \( \alpha \neq 0 \). The system in the form (1.7) was considered first in [11]. For practical reasons (1.7a) should be solved by iteration. As we will see, when \( R \) is s.p.d. and \( S \) is symmetric solving (1.7a) by iteration is essentially equivalent to solving (1.6a) by some iteration method using \((R + \alpha S)R^{-1}(R + \alpha S)\) as a preconditioner. The latter method was considered in [1], [2].

If we solve (1.6a) or (1.7a) by a direct solution method we must form \( R + SR^{-1}S \) or the matrix in (1.7b). This would frequently be too costly both with respect to computational labor and memory requirements. Iterative solution methods do not require the formation of these matrices and are particularly efficient when \( S \) and \( R \) are sparse.

A basic iteration method has the form

\[(1.8) \quad C(x^{l+1} - x^l) = -\tau r^l, \quad l = 0, 1, \ldots. \]
where $x^0$ is a given initial approximation, $C$ is a preconditioning matrix and $\tau_\ell$ are acceleration parameters, such as in a Chebyshev iteration method. Similar computations occur in conjugate gradient type methods (see, for instance, [3]).

The most efficient form of the computation of the residual $r^\ell$ is the following. We have from (1.7a)

$$r^\ell = (R - \alpha S)x^\ell - u + S(R + \alpha S)^{-1}[(1 + \alpha^2)Sx^\ell - v + \alpha u].$$

Using the identity

$$(1 + \alpha^2)S = \alpha(R + \alpha S) + S - \alpha R$$

we obtain

$$r^\ell = Rx^\ell - u - S(R + \alpha S)^{-1}[(\alpha R - S)x^\ell + v - \alpha u].$$

This shows that we can compute $r^\ell$ in the following way:

\begin{align}
\text{(1.9a)} & \quad \text{Solve} \quad (R + \alpha S)y^\ell = (\alpha R - S)x^\ell + v - \alpha u \\
\text{(1.9b)} & \quad r^\ell = Rx^\ell - S y^\ell - u
\end{align}

In this form we avoid the initial computation of the right hand side vector $(R + \alpha S)^{-1}(v - \alpha u)$. In addition, the vector $y^\ell$ is found as a part of the computation of $r^\ell$ so there is no need to solve (1.6b) when $x$ has been computed. These savings can be significant because normally the systems with $R$ or with $R + \alpha S$ are the most costly parts of the whole computation. Also as we shall see, in many cases one can choose the parameter $\alpha$ or a preconditioning matrix $C$ efficiently so that in total there will be few iteration. A savings of two solutions with $R + \alpha S$ can therefore be important.

Iterative solutions of complex systems have been considered for instance, in [1] - [5], [7] - [11]. Real valued forms have been considered in [1], [2], [10] and [11]. The remainder of the present paper is organized as follows.

In the next section preconditioning matrices for (1.6a) and (1.7a) are analysed, condition number estimates and the optimal value of parameter $\alpha$ are obtained, and a two-parametric extension of the system (1.7) is presented. An efficient algorithm for solving matrix polynomial completes section 2. In section 3 various applications of the methods are considered when solving the Cauchy problems including parabolic and hyperbolic equations.

2 Efficient preconditioning methods for complex systems reduced to real valued form

Consider (1.7a) where we assume that $R$ is symmetric and positive definite and $S$ is symmetric and positive semidefinite. Let $R + \alpha S$ be a preconditioner to $G_\alpha$ in (1.7b).
To analyse the corresponding condition number we let $H = R^{-\frac{1}{2}}SR^{-\frac{1}{2}}$ and consider the generalized eigenvalue problem

$$\mu(R + \alpha S)x = G_\alpha x,$$

where $\mu$ is the eigenvalue corresponding to an eigenvector $x$. Equivalently

$$(2.1) \quad \mu(I + \alpha H)y = [I - \alpha H + (1 + \alpha^2)H(I + \alpha H)^{-1}H]y,$$

where $y = R^\frac{1}{2}x$.

If $\lambda Rx = Sx, x \neq 0$ or, equivalently, $Hy = \lambda y, y \neq 0$ then (2.1) shows that

$$\mu = \frac{1 - \alpha \lambda + (1 + \alpha^2)\lambda^2/(1 + \alpha \lambda)}{1 + \alpha \lambda}$$

or

$$\mu = \frac{1 + \lambda^2}{(1 + \alpha \lambda)^2}.$$

Following up a remark made on the equivalence between the methods based on (1.6a) and (1.7b), this shows that, under the assumption made on $R$ and $S$, the preconditioning matrix $R + \alpha S$ for $G_\alpha$ will have the same rate of convergence as the preconditioning matrix $C = (R + \alpha S)R^{-1}(R + \alpha S)$ in (1.8) when $r^l$ is defined by (1.6a), i.e.

$$r^l = (R + \alpha S)x^l - (u + SR^{-1}v), \quad l = 0, 1, \ldots$$

The latter method can be rewritten in the form

$$(2.2) \quad \alpha^2 C^{-1}r^l = x^l - C^{-1}[((\alpha^2 - 1)R - 2\alpha S)x^l + \alpha^2(u + SR^{-1}v)], l = 0, 1, \ldots$$

Hence apart from computing $R^{-1}v$ once, in (2.2), there is no need to solve systems with matrix $R$ in any of the two methods.

Both methods involve two solutions with the matrix $R + \alpha S$ at each iteration step: In method (1.6a) to solve a system with preconditioner $C$ and in method (1.7a), implemented as in (1.9a,b), one solution of a system where this matrix is needed to compute $r^l$ and one for the preconditioner.

### 2.1 Condition number estimate

We want to choose $\alpha$ to minimize the spectral condition number $\mu_{max}/\mu_{min}$.

**Theorem 2.1.** Assume that $R$ is s.p.d. and $S$ is symmetric and positive semidefinite. Then the extreme eigenvalues of $(R + \alpha S)^{-1}G_\alpha$, where $G_\alpha$ is defined in (1.7b), satisfy
\[ \mu_{\text{min}} = \begin{cases} 
\frac{1}{1+\alpha^2}, & \text{if } 0 \leq \alpha \leq \hat{\lambda} \\
\frac{1+\alpha^2}{(1+\alpha\hat{\lambda})^2}, & \text{if } \hat{\lambda} \leq \alpha 
\end{cases} \]

\[ \mu_{\text{max}} = \begin{cases} 
1, & \text{if } \hat{\alpha} \leq \alpha \\
\frac{1+\hat{\lambda}^2}{(1+\alpha\hat{\lambda})^2}, & \text{if } 0 \leq \alpha \leq \hat{\alpha} 
\end{cases} \]

where \( \hat{\lambda} \) is the maximum eigenvalue of \( R^{-1}S \) and \( \hat{\alpha} = \frac{\hat{\lambda}}{1+\sqrt{1+\hat{\lambda}^2}} \). The spectral condition number is minimized when \( \alpha = \hat{\alpha} \), in which case it takes the value

\[ \frac{\mu_{\text{max}}}{\mu_{\text{min}}} = 1 + \hat{\alpha}^2 = \frac{2\sqrt{1+\hat{\lambda}^2}}{1 + \sqrt{1+\hat{\lambda}^2}}. \]

\textbf{Proof.} The bounds of the extreme eigenvalues follow by elementary computations of \( \mu = (1+\lambda^2)/(1+\alpha\lambda)^2, \quad 0 \leq \lambda \leq \hat{\lambda} \). Similarly, it is readily seen that \( \mu_{\text{max}}/\mu_{\text{min}} \) is minimized for some \( \alpha \) in the interval \( \hat{\alpha} \leq \alpha \leq \hat{\lambda} \), where \( \mu_{\text{max}} = 1 \). Hence, it is minimized for \( \alpha = \arg\max_{\alpha \leq \hat{\alpha}} (1+\alpha^2)^{-1} \), i.e. for \( \alpha = \hat{\alpha} \). \( \square \)

Note that condition number is bounded above by \( 2 \) for any choice of \( \alpha, \hat{\alpha} \leq \alpha \leq 1 \). In practice \( \hat{\lambda} \) is often large, so \( \hat{\alpha} = 1 - 1/\hat{\lambda} + O(1/\hat{\lambda}^2), \quad \hat{\lambda} \to \infty \). Therefore, if \( \hat{\lambda} \) is not known we let \( \alpha = 1 \), in which case the smallest eigenvalue is \( \frac{1}{2} \), the largest is \( 1 \) and the condition number takes the value \( 2 \).

It is well known that the preconditioned Chebychev or the conjugate gradient methods to solve (1.7a) converge with a convergence factor

\[ \left(\frac{2}{\sigma^n + \sigma^{-1}}\right)^{1/2}, \quad \sigma = (\sqrt{\frac{\mu_{\text{max}}}{\mu_{\text{min}}}} - 1)/(\sqrt{\frac{\mu_{\text{max}}}{\mu_{\text{min}}}} + 1) \]

and \( l \) is the iteration number.

\subsection*{2.2 A two-parametric real valued method}

As has been shown in \([11]\), the method in (1.7a) can be extended using two parameters. Let then \( 0 < \beta < \alpha \) and consider

\[ (R + \frac{1}{\alpha}S)^{-1}G_\alpha x = (R + \frac{1}{\alpha}S)^{-1}b_\alpha \]

where \( G_\alpha \) is defined in (1.7b) and

\[ b_\alpha = u + S(R + \alpha S)^{-1}(v - \alpha u). \]

Similarly

\[ (R + \frac{1}{\beta}S)^{-1}G_\beta x = (R + \frac{1}{\beta}S)^{-1}b_\beta. \]
Then, substracting (2.3), multiplied by \(\frac{(\alpha+1)^2}{2(\alpha^2+1)}\), from (2.4), multiplied by \(\frac{(\beta+1)^2}{2(\beta^2+1)}\), we obtain

\[ G_{\alpha,\beta} x = b_{\alpha,\beta} \]

where

\[ G_{\alpha,\beta} = \frac{(\beta+1)^2}{2(\beta^2+1)} (R + \frac{1}{\beta} S)^{-1} G_{\beta} - \frac{(\alpha+1)^2}{2(\alpha^2+1)} (R + \frac{1}{\alpha} S)^{-1} G_{\alpha} \]

and \(b_{\alpha,\beta}\) is defined similarly. If \(H = R^{-\frac{1}{2}} S R^{-\frac{1}{2}}\) has eigenvalue \(\lambda\), a computation shows that \(G_{\alpha,\beta}\) has eigenvalues

\[ (2.5) \mu = \frac{1}{2} (1 + \frac{1}{\beta}) + \frac{1}{1 + \eta} - \frac{1}{2} (1 + \frac{1}{\alpha}) - \frac{1}{1 + \alpha \eta} = \frac{a - b}{2ab} \left[ 1 + \frac{\alpha \eta(1 - \eta)}{(1 - a)(1 - \alpha \eta)} \right] \]

where

\[
a = \frac{1}{2} (\alpha + \frac{1}{\alpha}) > 1, \quad b = \frac{1}{2} (\beta + \frac{1}{\beta}) \geq 1, \quad \eta = \frac{2\lambda}{1 + \lambda^2} \]

and \(0 \leq \eta \leq 1\), since \(0 \leq \lambda < \infty\).

Using (2.5) it can be shown that

\[ (2.6) \frac{\mu_{\text{max}}}{\mu_{\text{min}}} = \left( \frac{2a + 1}{a \sqrt{2} + \sqrt{a + 1}} \right)^2 \]

where we have let \(\beta = 1\), which is optimal value of \(\beta\) to minimize the condition number. Since the function in the right hand side of (2.6) is monotonically increasing in \(a\), we let \(a\) be a number close to 1 (but \(a > 1\), because \(\alpha > \beta\)). The lower bound of the condition number, which is taken for \(a = 1\) is \((\frac{3}{\sqrt{2}})^2 = \frac{9}{8} = 1.125\).

Each iteration of the above two-parametric method involves four solutions of systems with \(R + \theta S\), where \(\theta = \alpha, \ 1/\alpha, \ \beta, \ \text{and} \ 1/\beta\). Alternatively, we can use two steps of the Chebyshev iteration method with one parameter, the cost of which will be about the same as one step of the two-parametric method. Since both methods involve rational functions in \(H\) of the same order, and because of the optimality property of the Chebyshev iteration method, it can be seen that the latter does never converge slower than the first method. Hence, since both methods involve about the same amount of computations, there is actually no advantage in using a two-parametric preconditioning method compared to a one parametric when a Chebyshev or conjugate gradient acceleration method is used. However, a two-parametric method is better tailored for parallel computers, because a twice bigger number of parallel processors can be utilized than in the one-parametric method.
2.3 Solving matrix polynomial equation

Consider now the matrix polynomial equation (1.1). We assume that \( Q_m \) has real coefficients. We want to reduce the solution of (1.1) to a number of first order equations. To this end, we note that \( Q_m \) can be factored in first and second order factors with real coefficients, where each second factor corresponds to a pair of complex conjugate roots of \( Q_m \). Therefore, it suffices to consider a second order polynomial

\[
P_2(A) = I + 2Re(a)A + a\bar{a}A^2,
\]

where \(-a^{-1}, -\bar{a}^{-1}\) are the roots of \( P_2 \). An elementary computation shows that

\[
P_2^{-1}(A) = b(I + aA)^{-1} + \bar{b}(I + \bar{a}A)^{-1}
\]

where

\[
b = \frac{a}{a - \bar{a}} = \frac{1}{2} - \frac{1}{2} \frac{Re(a)}{Im(a)}.
\]

To solve \( P_2(A)y = f \), we then compute

\[
(I + aA)v = f
\]
\[
(I + \bar{a}A)w = f
\]
\[
y = bv + \bar{b}w,
\]

where \( y = P_2(A)^{-1}f \). If \( f \) is real, then

\[
w = (I + \bar{a}A)^{-1}f = \bar{v},
\]

so, in this case, it suffices to solve one and and only one such system with matrix \((I + aA)\). We compute then a solution of \( P_2(A)y = f \), in order

\[
(I + aA)v = f
\]
\[
y = bv + \bar{b}v = 2Re(bv).
\]

3 Applications for differential equations

Let \( u \) and \( f \) be \( N \)-dimensional vector functions and consider the system of ordinary differential equations

\[
\frac{du(t)}{dt} + Lu(t) = f(t), \quad t > 0, \quad u(0) = u_0
\]

where \( L \) is an \( N \times N \) real matrix which does not depend on \( t \) and with eigenvalues with positive real parts.
### 3.1 Periodic source function

Assume \( f(t) \) to be periodic in \( t \), i.e.

\[
f(t) = f_0 e^{i\omega t}
\]

where \( f_0 \) is a real constant vector. As is well known, the solution of (3.1) takes the form

\[
u(t) = \exp(-Lt)u_0 + \int_0^t \exp(L(s-t))f(s)ds
\]

and an elementary computation shows that for the periodic function \( f \),

\[
(3.2) \quad u(t) = \exp(-Lt)(u_0 - v_0) + e^{i\omega t}v_0
\]

where \( v_0 \) is the solution of

\[
(3.3) \quad (L + i\omega I)v_0 = f_0.
\]

By assumption, \( \exp(-Lt) \to 0 \) so the stationary solution of (3.1) is periodic, \( e^{i\omega t}v_0 \).

Note that (3.3) has the form (1.2). To compute \( v_0 \) we proceed as follows. Let \( v_0 = x + iy \), where \( x, y \) are real. To apply (1.7a,b) and Theorem 2.1 we let \( R = L, S = \omega I \), and find

\[
\hat{\lambda} = \frac{\omega}{\lambda_{\min}(L)}, \quad \alpha = \alpha = \frac{1}{\sqrt{1 + \left(\frac{\lambda_{\min}(L)}{\omega}\right)^2 + \frac{\lambda_{\min}(L)}{\omega}}}.
\]

If \( \lambda_{\min}(L) \) is not known, we let \( \alpha = 1 \). The spectral condition number of the preconditioned matrix is bounded above by 2 for \( 0 < \omega < \infty \).

We can compare the obtained result with the efficiency of complex valued iterative methods for solving (3.3). For example, one could choose \( L \) as an preconditioner for (3.3) and solve such preconditioned system with a matrix \( I + i\omega L^{-1} \) by complex valued iterative schemes with Chebyshev or conjugate gradient-like parameters [9]. Eigenvalues of preconditioned matrix \( I + i\omega L^{-1} \) are complex and located in the segment \([c_1, c_2] \), where 

\[
c_1 = 1 + i\omega/\lambda_{\max}(L), \quad c_2 = 1 + i\omega/\lambda_{\min}(L).
\]

In the latter case the number of iterations will be proportional to the square root of the condition number (see [9])

\[
\kappa(I + i\omega L^{-1}) = \frac{\lambda_{\max}(L)(\omega^2 + \lambda_{\min}^2(L))^{\frac{1}{2}}}{\lambda_{\min}(L)(\omega^2 + \lambda_{\max}^2(L))^{\frac{1}{2}}}
\]

For large \( \omega \) we have \( \kappa \approx \frac{\lambda_{\max}(L)}{\lambda_{\min}(L)} \), so complex iterative methods will converge very slowly, while for the proposed method a number of iterations \( l(\epsilon) \), where \( \epsilon \) is a required tolerance, is bounded by
\[
\ln(\frac{2}{\epsilon}) \leq \frac{\ln \frac{2}{\epsilon}}{ln \frac{\sqrt{2}+1}{\sqrt{2}-1}} \approx 0.5673 \ln \frac{2}{\epsilon}.
\]

Summarizing the above algorithmic improvements and taking into account that \(f_0\) is real the real valued iterative method for solving the system (3.3) takes the following form.

\[
x^0, y^0 \text{ - initial guess}
\]

For \(l = 0, 1, \ldots\), do

\[
\text{Solve } (L + \alpha L)y^{l+1} = (\alpha L - \omega I)x^l - \alpha f_0
\]

\[
r^l = Lx^l - \omega y^l - f_0
\]

\[
\text{Solve } (L + \alpha L)w = -\gamma r^l
\]

\[
x^{l+1} = x^l + w
\]

\[
\text{enddo}
\]

Clearly, the method (3.4) is applicable also when \(f\) is a sum of periodic functions, \(f = \sum f_k e^{i\omega_k t}\), in which case we seek \(u\) in the form \(u = \exp(-Lt)(u_0 - \sum v_k) + \sum v_k e^{i\omega_k t}\), where each \(v_k\) satisfies \((L + i\omega_k) v_k = f_k\). Note that the vectors \(v_k\) can be computed in parallel, and each matrix to be solved in (3.4) is better conditioned the larger the frequency \(\omega_k\). For small frequencies, it can be efficient to consider the system \((I - i\omega_k^{-1} L)v_k = -i\omega_k^{-1} f_k\), instead.

### 3.2 Parabolic problems, general source function

Consider now (3.1) for a general function \(f\). The exponential function in (3.2) can be approximated using a Padé approximation

\[
R_{km}(z) = \sum_{j=0}^{k} \frac{k!}{(k-j)!} \frac{(k+m-j)!}{(k+m)!} \frac{z^j}{j!} \sum_{j=0}^{m} \frac{m!}{(m-j)!} \frac{(k+m-j)!}{(k+m)!} \frac{(-z)^j}{j!}
\]

which has error (see, for instance, [6])

\[
e^z - R_{km}(z) = (-1)^m \frac{k!m!}{(k+m)!((k+m)+1)!} z^{k+m+1} + O(z^{k+m+2}).
\]

It is known that if \(m \geq 2\), then the denominator polynomial has complex zeroes, except when \(m\) is odd in which case one zero is real. Hence the arising matrix polynomial equation can be solved using the methods in Sections 1 and 2. This particular application has been discussed previously in [1], [2].

Consider now the following step by step approximation of type \(R_{22}\) of (3.1),

\[
(I + \frac{1}{2} \tau L + \frac{1}{12} \tau^2 L^2) u_{n+1} = (I - \frac{1}{2} \tau L + \frac{1}{12} \tau^2 L^2) u_n + \tau \phi_n,
\]

\[\text{enddo}\]
where
\[
\phi_n = \frac{1}{2}(f_{n+1} + f_n) + \frac{1}{12}\tau L(f_{n+1} - f_n) - \frac{1}{12}\tau(f'_{n+1} - f'_n), \quad n = 0, \ldots,
\]
and \(\tau\) is the step-size. This approximation has an optimal fourth order of accuracy.

It is computationally efficient to rewrite (3.5) in the form
\[
(I + \frac{1}{2}\tau L + \frac{1}{12}\tau^2 L^2)(u_{n+1} - u_n) = \tau(-Lu_n + \phi_n), \quad n = 0, 1, \ldots
\]
Applying (2.7), (2.8) we obtain the following algorithm:

\[
\text{For } n = 0, 1, \ldots, \text{ do}
\]
\[
\quad g_n = -Lu_n + \phi_n,
\]
\[
\quad \text{Solve } (I + (1 + \frac{1}{\sqrt{3}i})\frac{\tau}{4}L)v = \tau g_n
\]
\[
\quad u_{n+1} = u_n + v_{\text{Re}} + \sqrt{3}v_{\text{Im}}
\]
\[
\text{endo,}
\]
where \(v_{\text{Re}}\) and \(v_{\text{Im}}\) denote the real and imaginary parts of \(v\), respectively.

Assuming that \(L\) is positive semi-definite, which is typical for parabolic second order problems where the differential operator in space has been discretized using finite difference or finite element methods, and using the previously discussed method to solve the complex system with preconditioner \((I + \frac{\tau}{4}L + \alpha\frac{\tau}{4\sqrt{3}}L)\) where in Theorem 2.1, \(R = I + \frac{\tau}{4}L, S = \frac{\tau}{4\sqrt{3}}L, A = \max \lambda((I + \frac{\tau}{4}L)^{-1}\frac{\tau}{4\sqrt{3}}L) \leq \frac{1}{\sqrt{3}}\),

we find
\[
\hat{\alpha} = \frac{1/\sqrt{3}}{1 + \sqrt{1 + \frac{1}{3}}} = \frac{1}{2 + \sqrt{3}}.
\]

The condition number \(\kappa\) of the corresponding preconditioned matrix is (see Theorem 2.1)
\[
\kappa = 1 + \hat{\alpha}^2 = \frac{4}{2 + \sqrt{3}} \simeq 1.072.
\]
This corresponds to a reduction factor
\[
\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \simeq 0.018
\]
and with such a small reduction factor it suffices in practice with two or three iterations to solve (3.7).
3.3 Hyperbolic problems

Consider now the problem (3.1) in the form

\[ \frac{dU}{dt} + \tilde{L}U = F(t), \quad t > 0, \quad U(0) = U_0 \]

where

\[ U = \begin{bmatrix} U^{(1)} \\ U^{(2)} \end{bmatrix}, \quad \tilde{L} = \begin{bmatrix} 0 & -I \\ L & 0 \end{bmatrix}, \quad F = \begin{bmatrix} 0 \\ f(t) \end{bmatrix} \]

and \( L \) is symmetric positive semi-definite, typically a discrete analog of the negative Laplacian operator. Systems of the form (3.8) arise when solving second order wave equations \( \frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f(x,y), t > 0 \) for instance. To solve (3.8) we apply the previously presented \( R_{22} \)-Padé type approximation and the algorithm in (3.7) takes now the form:

\[
\begin{align*}
\text{For } n = 0, \ldots, & \text{ do} \\
& g_n = \begin{bmatrix} U_n^{(2)} - \frac{1}{12} \tau (f_{n+1} - f_n) \\
-LU_n^{(1)} + \frac{1}{2} (f_{n+1} + f_n) - \frac{1}{12} \tau (f'_{n+1} - f'_n) \end{bmatrix} \\
& \text{Solve } \begin{bmatrix}
I & (1 + \frac{1}{\sqrt{3}} i) \frac{\tau}{4} L \\
-(1 + \frac{1}{\sqrt{3}} i) \frac{\tau}{4} I & I
\end{bmatrix} V = \tau g_n, \\
& U_{n+1} = U_n + V_{\text{Re}} + \sqrt{3} V_{\text{Im}}, \quad n = 0, 1, \ldots \\
\text{Enddo}
\end{align*}
\]

To find components \( V^{(1)}, V^{(2)} \) of \( V \) (3.9) shows that we must solve the equation

\[ [I + (1 + \sqrt{3} i) \frac{\tau^2}{24} L]w = \psi, \]

where \( \psi = g_n^{(1)} + (1 + \frac{1}{\sqrt{3}} i) \frac{\tau}{4} g_n^{(2)} \), and then compute

\[ V^{(1)} = \tau w, \]
\[ V^{(2)} = (\sqrt{3} i - 3)(g_n^{(1)} - w). \]

The preconditioning matrix for (3.10) is

\[ I + \frac{1 + \alpha \sqrt{3}}{24} \tau^2 L \]

and we find

\[ \hat{\lambda} = \sqrt{3}/(1 + \frac{24}{\tau^2 \lambda_{\text{max}}(L)}) \leq \sqrt{3}. \]
If $L$ is the standard difference approximation of the Laplacian operator on a unit square and $\tau = h$, where $h$ is the space discretization parameter, then

$$\hat{\lambda} = \frac{\sqrt{3}}{4}, \quad \alpha = \hat{\alpha} = \frac{\sqrt{3}}{4 + \sqrt{19}},$$

and the condition number of the preconditioned matrix (cf. Theorem 2.1) becomes

$$\kappa \leq 1 + \frac{3}{(4 + \sqrt{19})^2} \simeq 1.04$$

which results in a reduction factor

$$\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} - 1} \simeq 0.01.$$

Note that in this case, when $\tau = O(h)$, the preconditioned matrix (3.11) has a condition number $O(1)$. Therefore, each such system can be solved with a computational cost $O(N)$, where $N$ is the degree of freedoms. Hence, the above algorithm works essentially as an explicit time-stepping method, but is still unconditionally stable.

4 Conclusions

It has been shown how certain complex valued linear systems of equations can be rewritten to involve only real arithmetic and how the resulting system can be solved efficiently by iteration involving a positive definite system as preconditioner and a small condition number which is not greater than 2.

Various applications of the method when solving matrix polynomial equations and evolution equations are also demonstrated.

References


