

1993

## Preconditioner Construction with Rational Approximation

Mo Mu

John R. Rice  
*Purdue University, jrr@cs.purdue.edu*

Report Number:  
93-001

---

Mu, Mo and Rice, John R., "Preconditioner Construction with Rational Approximation" (1993). *Department of Computer Science Technical Reports*. Paper 1021.  
<https://docs.lib.purdue.edu/cstech/1021>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.  
Please contact [epubs@purdue.edu](mailto:epubs@purdue.edu) for additional information.

**Preconditioner Construction with  
Rational Approximation**

**Mo Mu  
John R. Rice**

**CSD-TR-93-001  
January 1993**

## Preconditioner Construction with Rational Approximation \*

Mo Mu<sup>†</sup>      John R. Rice<sup>†</sup>

### Abstract

This paper deals with the domain decomposition-based preconditioned conjugate gradient method. The Schur complement is expressed as a function of a simple interface matrix. This function is approximated by a simple rational function to generate a simple matrix that is then used as a preconditioner for the Schur complement. Extensive experiments are performed to examine the effectiveness and efficiency of this approach.

### 1 Introduction

A new approach of constructing preconditioners for the domain decomposition-based preconditioned conjugate gradient (PCG) method through a function approximation was recently proposed in [1] and further extended in [2] based on the following theory of the Schur complement expression.

**PROPOSITION 1.1.** [2] *Let a rectangular domain  $\Omega$  be decomposed into two subdomains  $\Omega_1$  and  $\Omega_2$  by a horizontal interface  $\Gamma$ ; a separable and self-adjoint second order elliptic Dirichlet boundary value problem be discretized by the 5-point-star stencil on a tensor product grid. Denote  $T_x \equiv \Theta_x^{-1/2} A_x \Theta_x^{-1/2}$ ,  $T_y^i(t) \equiv t\Theta_y^i + A_y^i$ ,  $i = 1, 2$ , where  $A_x$  is the discrete operator in the  $x$  direction,  $A_y^i$  is the discrete operator in the  $y$  direction in  $\Omega_i$ ;  $\Theta_x \equiv \text{diag} \left( \frac{h_x^k + h_x^{k+1}}{2} \right)$  with  $\{h_x^k\}$  being the spacings in the  $x$  direction, and similarly for  $\Theta_y^i$  in the  $y$  direction. Then the interface Schur complement matrix can be expressed as*

$$(1) \quad S = \Theta_x^{1/2} f(T_x) \Theta_x^{1/2},$$

where

$$(2) \quad f(t) = (\theta_y^0 t + \alpha_y^0) - \sum_{i=1}^2 \left( \frac{\beta_y^{i0}}{l_{m_i m_i}^i} \right)^2,$$

and  $l_{m_i m_i}^i$  is the last diagonal element of the Cholesky factor of  $T_y^i(t)$  with  $m_i$  being the number of horizontal grid lines in  $\Omega_i$ ; the triple  $(\beta_y^{10}, t\theta_y^0 + \alpha_y^0, \beta_y^{20})$  is similar to rows in  $T_y^i(t)$  but corresponding to the interface.

Our preconditioner construction approach is to seek an approximation  $r(t)$  to  $f(t)$ , such that, with  $q(t) \equiv f(t)/r(t)$ , (a)  $R \equiv r(T_x)$  is easily invertible; (b)  $\frac{\max_i |q(t_i)|}{\min_i |q(t_i)|} \sim 1$ , or  $\{q(t_i)\}$  are clustered, where  $\{t_i\} = \sigma(T_x)$  is the spectrum of  $T_x$ . Then  $M \equiv \Theta_x^{1/2} R \Theta_x^{1/2}$  is a good

\*The work of the first author was supported in part by National Science Foundation grant CCR-8619817. The work of the second author was supported in part by the Air Force Office of Scientific Research grants, 88-0243, F49620-92-J-0069 and the Strategic Defense Initiative through Army Research Office contract DAAL03-86-K-0106.

<sup>†</sup>Computer Sciences Department, Purdue University, West Lafayette, IN 47907.

preconditioner for  $S$  for the *PCG* method. Thus, the preconditioning problem is reduced to a function approximation.

A rational function is a natural and successful choice for  $r(t)$ . An intuitive strategy is suggested in [1] to construct  $r(t)$  by utilizing the special properties of  $f(t)$  and [2] describes a more general way using *weighted rational Chebyshev approximation*. Concrete examples in [2] illustrate the effectiveness of this new preconditioning approach. We only report on selected results from this extensive experimental study.

## 2 The Behavior of $f(t)$

It is easy to see that  $f(t)$  is independent of the information in the  $x$  direction; but it may depend on the  $y$  direction information, such as the location of the interface, the grid nonuniformity, coefficients of the elliptic operator, and so on. It is shown in [2] that  $f(t)$  can be expressed as a rational function in terms of certain orthogonal polynomials. For a Laplace operator and a uniform  $y$  direction grid, these polynomials further reduce to the Chebyshev polynomials. In this case, the major concern is the effect of the location of  $\Gamma$ . As shown in [1], the behavior of  $f(t)$  is then almost identical when moving  $\Gamma$  up and down, the effect is only seen near the minimum eigenvalue  $\lambda_{\min}(T_x)$ . Thus  $f(t)$  is insensitive to the location of  $\Gamma$ . However, the behavior of  $f(t)$  does vary for more complicated cases. To illustrate it, we use a  $61 \times 33$  grid which is nonuniform in both  $x$  and  $y$  directions. The spacings in each direction are of an exponential distribution to account for an exponential type of singularities in the *PDE* solution so that the grid lines are very clustered near the  $x$  and  $y$  axes. See Figure 3.1 in [2] for more details. We denote  $\Omega_1$  as the subdomain below  $\Gamma$ .

We consider the dependence of  $f(t)$  on the location of  $\Gamma$  for this nonuniform grid and the Laplacian. Fig. 1 shows curves of  $f(t)$  for various locations of  $\Gamma$ . The left figure shows the entire plot when  $t$  varies in the spectrum of  $T_x$  and the right figure is an enlarged view of the lower left part. The closer  $\Gamma$  is to the region where grid lines are more clustered, the more the right end of  $f(t)$  bends towards, and the left end goes away from, the  $t$  axis; and two top curves almost coincide. We conclude that remote parts of the domain have weak effects on the interface.

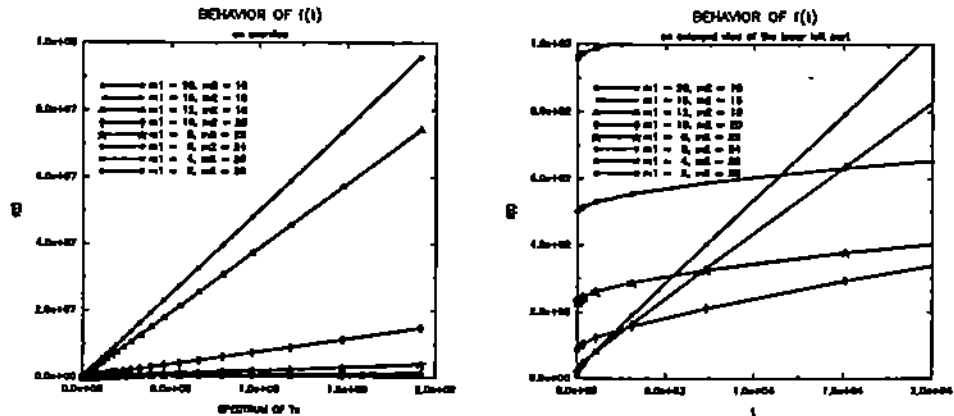


FIG. 1. Curves of  $f(t)$  for various pairs of  $(m_1, m_2)$ : the left figure is an overview when  $t$  varies in the spectrum of  $T_x$ ; the right figure is an enlarged view of the lower left part.

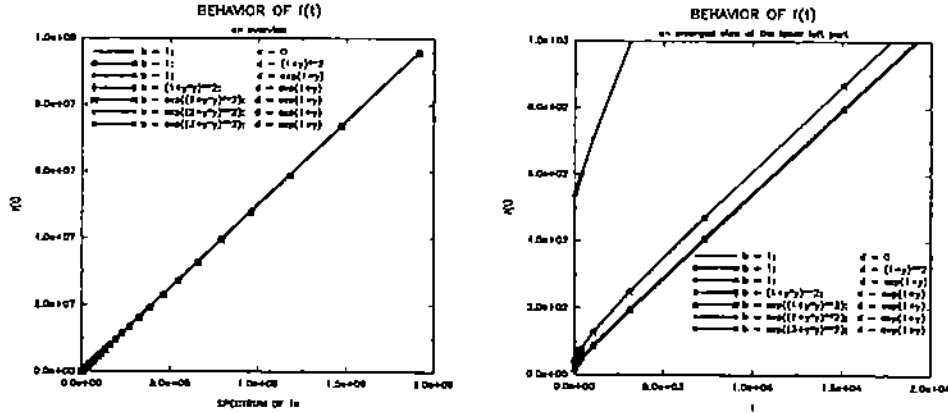


FIG. 2. Curves of  $f(t)$  for various coefficients  $b(y)$  and  $d(y)$ : the left figure is an overview when  $t$  varies in the spectrum of  $T_x$ ; the right figure is an enlarged view of the lower left part.

The second example uses fixed parameters  $m_1 = m_2 = 15$ . The  $y$  direction operator  $L_y$  is extended to  $-\partial/\partial y(b(y)\partial/\partial y) + d(y)$ . Fig. 2 shows curves of  $f(t)$  for various coefficients  $b(y)$  and  $d(y)$ . These coefficient functions vary from a constant, to a polynomial, and to an exponential. Overall, all curves almost coincide as the coefficients change. The effect is only seen near  $\lambda_{\min}(T_x)$  when the lower left part of the plot is enlarged.

In both examples we see that  $f(t)$  has a *two-part* property as observed in [1] [2]. That is,  $f(t)$  looks mostly like a linear function (the *easy part*), and at the left end region of  $\sigma(T_x)$  the few smallest eigenvalues make  $f(t)$  behave like  $t^{1/2}$  (the *hard part*). A *two-phase* approximation strategy is used [1] [2] with

$$(3) \quad r(t) = r_1(t)r_2(t),$$

where  $r_1(t)$  and  $r_2(t)$  approximate the *hard* and *easy* parts, respectively.

### 3 The Computation of $r(t)$

As in [1] [2] we choose  $r_1(t)$  and  $r_2(t)$  as first-degree rational functions determined by three interpolating points; The preconditioning computation then uses two matrix-vector multiplies and two solves with tridiagonal matrices because  $T_x$  is tridiagonal. It is natural to compute  $r_1(t)$  by interpolating the first three smallest eigenvalues of  $T_x$ . Similarly, we use the minimum and the two largest eigenvalues of  $T_x$  for computing  $r_2(t)$  since  $f(t)/r_1(t)$  is almost linear for  $t \in \sigma(T_x)$ . The spectrum  $\sigma(T_x)$ , over which  $f(t)$  is approximated, depends on the  $x$  direction information. Only rough estimates are required for picking these interpolating points and it is very easy to compute the eigenvalues of  $T_x^*$ , a perturbation of  $T_x$  with the  $x$  direction grid uniform and the operator coefficients constant with average values. In general, the small eigenvalues of  $T_x^*$  are fairly good estimates for those of  $T_x$ , but the large ones may differ significantly. For the case of the Laplacian operator, a uniform  $y$  direction grid, and moderately nonuniform  $x$  direction grids, this works fine by using all interpolating points computed from  $T_x^*$  [1]; for some extremely nonuniform  $x$  direction grids, it is necessary to change  $\lambda_{\max}(T_x^*)$  to a better estimate of  $\lambda_{\max}(T_x)$ , which is easily done. With this remedy, this strategy usually also works very well for general cases. However,

TABLE 1

Convergence performance of the preconditioner using the rational approximation for the  $f(t)$  in Fig. 1,  $\kappa$  is the condition number of the preconditioned system,  $k$  is the number of iterations.

$f(t)$	1	2	3	4	5	6	7	8
$\kappa$	1.111	1.106	1.111	1.337	1.591	1.767	1.895	1.984
$k$	4	4	4	6	7	7	8	12

TABLE 2

Convergence performance of the preconditioner using the rational approximation for the  $f(t)$  in Fig. 2,  $\kappa$  is the condition number of the preconditioned system,  $k$  is the number of iterations.

$f(t)$	1	2	3	4	5	6	7
$\kappa$	1.106	1.125	1.136	1.133	1.083	1.075	1.072
$k$	4	6	7	7	6	4	3

there are cases where the interface  $\Gamma$  is too close to the clustered grid lines, or the operator coefficients vary too much (such as  $b(y) = e^{(3+y^2)^2}$ ,  $d(y) = e^{1+y}$ ), so that the second largest eigenvalue of  $(T_x^*)$  is still within the *hard* part. In these cases, one adjusts the second largest interpolating point to, say,  $c$  times the largest interpolating point, where  $c$  is a factor such as 0.5. We adopt this rational approximation (with the two remedies) in the previous two examples. The numerical results of this approximation are listed in Tables 1 and 2 for the two cases of Fig. 1 and Fig. 2, respectively, where  $\kappa \equiv \frac{\max_i |g(t_i)|}{\min_i |g(t_i)|}$  governs the convergence rate of the *PCG* method and  $k$  is the corresponding number of iterations to achieve accuracy of  $10^{-5}$  (single precision 32 bits is used in the computation). The curves of  $f(t)$  are indexed corresponding to their data sets (or see the legends) in the figures. These results show that our approach is very effective and also fairly stable. We do see in Table 1 that the approximation becomes poorer when the interface  $\Gamma$  moves towards the clustered grid lines, so one might further adjust the interpolating points to get a better approximation. Table 2 suggests that the more dominant the second order term (the coefficient  $b(y)$ ) is, the better the approximation is. We also note for case 7 in Table 2 that if the second largest interpolating point is simply computed using  $T_x^*$ , then  $\kappa$  becomes 114.19 and the approximation is totally out of the range. Finally, for comparison [1], the standard *CG* method and other well known preconditioners, such as the  $K^{1/2}$ -family, all fail to converge in 100 iterations for these difficult examples.

## References

- [1] M. Mu, *A new family of preconditioners for domain decomposition*, CSD-TR-92-064 and CER-92-27, Department of Computer Sciences, Purdue University, West Lafayette, IN47907, September, (1992), Submitted to SIAM J. Sci. Stat. Comput..
- [2] M. Mu and J.R. Rice, *Preconditioning for domain decomposition through function approximation*, CSD-TR-92-091 and CER-92-42, Department of Computer Sciences, Purdue University, West Lafayette, IN47907, December, (1992), Submitted to SIAM J. Sci. Stat. Comput..