Rice University

Preconditioner Schemes for Elliptic Saddle-point Matrices Based Upon Jacobi Multi-band Polynomial Matrices.

by
Victor J. Parr

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE
DOCTOR OF PHILOSOPHY

APPROVED, THESIS COMMITTEE:

Mary F. Wheeler

Prof. Mary F. Wheeler
Thesis Advisor, Rice University
Computational & Applied Math Dept.

Richard Tapia

Prof. Richard Tapia
Rice University
Computational & Applied Math Dept.

Clint Dawson

Prof. Clint Dawson
Rice University
Computational & Applied Math Dept.

Kyriacos Zygourakis

Prof. Kyriacos Zygourakis
Rice University
Department of Chemical Engineering

Houston, Texas

September, 1994

ABSTRACT

Preconditioner Schemes for Elliptic Saddle-point Matrices Based Upon Jacobi Multi-band Polynomial Matrices.

by

Victor J. Parr

Simulation of flow in porous media requires the numerical approximation of elliptic partial differential equations. Mixed finite element methods are frequently employed, because of local mass conservation and accurate approximation of both pressure and velocity. Mixed methods give rise to "elliptic" saddle-point (ESP) matrices, which are difficult to solve numerically. In addition, the problems to be modelled in ground water flow require that the hydraulic conductivity or absolute permeability be a tensor, which adds additional complexity to the resulting saddle-point matrices.

This research develops several preconditioners for restarted GMRES solution of the ESP linear systems. These preconditioners are based on a new class of polynomial matrices, which we refer to as Multi-band Jacobi Polynomial (JMP) matrices. Applications of these preconditioners to the numerical solution of two and three spatial dimensional flow equations with tensor coefficients using rectangular lowest order Raviart-Thomas spaces are presented.

Acknowledgments

I thank Dr. Mary Wheeler for continued guidance and friendship in writing this thesis.

I thank Drs. Clint Dawson, Richard Tapia, William Symes, Dan Sorensen, Todd Arbogast, Marcelo Ramé, and graduate students Lawrence Cowsar and Wendy Kinton of Rice University's Computational and Applied Mathematics Department, for encouragement and inspiration.

I thank Dr. Andrew Tompson of LLNL for the turning-bands random field generator software, and Dr. Dan Sorensen for the use of his code ARPACK, which implements the Implicitly Restarted Arnoldi algorithm, to estimate the extreme eigenvalues of the blocks of the matrices studied here.

I am grateful to Cray Research, Inc for the use of computers, which made the programming simple, and to Mick Edwards, George Stephenson, James Lord, Frank Kampe, Janet Shiu, and John Romo of Cray Research for their encouragement.

I dedicate this thesis to the memory of David Bourgin and my father Vital J. Parr. Finally, my wife Bobbie has made it all bearable.

Contents

Chapter 1. Introduction.

Simulation of flow in porous media requires the solution of mass balance equations in which the phase velocities are described by Darcy's Law [4,18]. Here the phase velocities are proportional to the gradient of the phase pressures. Mixed finite element methods or cell centered finite difference methods are frequently employed to model these equations [9,37]. These methods are locally conservative and pressure and velocity can both be approximated to the same degree of accuracy.

In this Thesis we shall restrict our attention to incompressible single phase flow. Here we must solve an elliptic partial differential equation with highly varying discontinuous tensor coefficients, the hydraulic conductivity, and/or absolute permeability. We further restrict our attention to the lowest order Raviart Thomas approximating space for the mixed finite element method. This space is the one most frequently used in practice.

The major computational difficulty in applying the mixed finite element procedure to the Darcy flow problem is that the solution of the linear system which arises involves solving an saddle-point system (SP), which has the form

$$\begin{bmatrix} A & B^t \\ B & -C \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}. \tag{1.1}$$

Here A is a symmetric positive definite submatrix, C is a positive semi-definite submatrix, and B is a rectangular submatrix, and the solution vector is a pair (u,p) in which u represents the Darcy velocity and p a pressure.

When C = 0, we call the saddle-point system an elliptic saddle-point system. On rectangular grids, if the conductivity is a diagonal tensor, the submatrix A is block diagonal. In this case the saddle-point system (1.1) can be reduced to an equation for the scalar variable p:

$$(B\ A^{-1}\ B^t + C)\ p \quad = \quad B\ A^{-1}\ f - g\,. \tag{1.2}$$

If the conductivity is a full tensor, then the block diagonal structure of A is lost, and one is forced to solve the unreduced system (1.1).

The main objective of this Thesis is the development of efficient and robust proconditioners for the system (1.1) with emphasis on the Darcy flow problem. As a basic preconditioning tool we construct a new class of polynomial matrices, which generalize the polynomial matrices described in [24]. We call this new class Jacobi Multi-band Polynomial (JMP) matrices, because they are based locally on the classical Jacobi polynomials. These polynomial matrices can be used to precondition any matrix with a real spectrum. Specifically, we formulate a JMP matrix p(A) to precondition A, a JMP matrix $p(BB^t)$ to precondition BB^t, and a JMP matrix p(S) to precondition the full ESP matrix S. With these flexible tools we construct preconditioners for ESP matrices and use them in restarted preconditioned GMRES solution of Darcy flow problems with highly varying coefficients.

This thesis is organized as follows. In Chapter 2, we develop a theory of JMP matrix preconditioners. In Chapter 3, we establish bounds for the spectrum of an ESP matrix, and construct several preconditioners using the JMP matrix tools mentioned above. Computational results arising from modeling Darcy flow are described in Chapter 4. These calculations were performed on a Cray C-90. In Chapter 5 concluding remarks are made.

Chapter 2. Jacobi Multi-band Polynomial Matrix Preconditioners.

2.1 Introduction.

In [24] Saad developed a theory for polynomial matrix preconditioners, which are used to accelerate the Richardson iteration for symmetric indefinite matrices. The preconditioner polynomials are expanded in a basis of polynomials, which are orthogonal on the union of two intervals. The inner-product employed by Saad, which defines the orthogonality on the union of two intervals is the sum of two inner-products, each defining orthogonality locally on one of the intervals. The local inner-products are chosen such that an orthogonal basis consists of scaled and shifted Chebyshev polynomials of the first kind. The preconditioner polynomial p(x) is then constructed such that the polynomial

$$q(x) = x\,p(x)$$

is the least squares approximation with respect to the global inner-product of the constant function f(x) = 1.

In this chapter Saad's construction is generalized as follows:

(1) the number of intervals is not restricted,

(2) the local basis polynomials are extended to Jacobi polynomials $p^{(\alpha,\beta)}(x)$,

(3) x p(x) approximates a piecewise polynomial on the set of intervals.

We shall refer to these new polynomial preconditioners as Jacobi multi-band polynomials (JMP) , and the corresponding polynomial matrices as JMP matrices.

This Chapter is organized as follows. In Section 2.2 the JMP approximation spaces and inner-product are defined. In Section 2.3 a Gram-Schmidt orthogonalization procedure (which we refer to as the JMP Stieltjes procedure) is outlined for construction of an orthonormal basis. In Section 2.4, quadrature rules for the JMP inner-product calculations are defined. In Section 2.5 we employ orthogonal expansions to define several types of preconditioners, which are used as tools to solve ESP linear systems.

2.2 JMP Approximation.

Before defining the JMP inner-product, we establish some notation.

Definition 2.2.1 Let A be a real symmetric matrix and $\Lambda^+(A)$ its non-zero eigenvalue spectrum. A sequence $\Sigma(A) = \{ [a_k,b_k] \}_{k=1}^{nb}$ of non-overlapping closed intervals is a **spectral cover** for A if 0 is not interior to any interval of $\Sigma(A)$, and $\Lambda^+(A) \subseteq \bigcup_{k=1}^{nb} [a_k,b_k]$. The interval $[a_k,b_k]$ is called an **eigenband** of A. The set theoretic union $\Sigma^*(A) = \bigcup_{k=1}^{nb} [a_k,b_k]$ is called a **spectral support** for A. The spectral cover $\Sigma(A)$ is **minimal** if each endpoint of an eigenband is an eigenvalue of A.

As examples, if {c,d} are the extreme eigenvalues of the SPD matrix A, then { [c,d] } is a minimal spectral cover for A. If {a,b} are the extreme

negative and {c,d} the extreme positive eigenvalues of the real symmetric indefinite matrix A, then { [a,b], [c,d] } is a minimal spectral cover for A.

Definition 2.2.2 Let $\Sigma(A)$ denote a spectral cover for A. Let $P_n(\Sigma^*(A))$ denote the vector space of polynomials restricted to $\Sigma^*(A)$ of degree $\leq n$, and let $Q_n(\Sigma^*(A))$ denote the vector subspace of polynomials of the form

$$q(x) = x\ p(x) \tag{2.2.1}$$

such that $p(x) \in P_{n-1}(\Sigma^*(A))$. Then $Q_n(\Sigma^*(A))$ is called the space of preconditioned polynomials, and $P_{n-1}(\Sigma^*(A))$ the space of preconditioner polynomials.

We now define the JMP inner-product.

Definition 2.2.3 For each eigenband $[a_k,b_k]$, let $w^{(\alpha_k,\beta_k)}: [a_k,b_k] \rightarrow \Re^1$ be the non-negative real-valued Jacobi weight function defined by

$$w^{(\alpha_k,\beta_k)}(x) = \begin{cases} (x-a_k)^{\beta_k}(b_k-x)^{\alpha_k}, & x \in [a_k,b_k], \\ 0 & x \in \Sigma^*(A)/[a_k,b_k], \end{cases} \tag{2.2.2}$$

which induces the inner-product,

$$\langle f, g \rangle_{(\alpha_k,\beta_k)} = \int_{a_k}^{b_k} f(x)\, g(x)\, w^{(\alpha_k,\beta_k)}(x)\, dx. \tag{2.2.3}$$

Definition 2.2.4 Let the vectors $\underline{\alpha} = (\alpha_1,..,\alpha_{nb})^t$ and $\underline{\beta} = (\beta_1,..,\beta_{nb})^t$ be given. If the **JMP weight function** $w^{(\underline{\alpha},\underline{\beta})}$ with respect to the spectral cover $\Sigma(A)$ is given by $w^{(\underline{\alpha},\underline{\beta})}(x) = w^{(\alpha_k,\beta_k)}(x)$, for $x \in [a_k,b_k]$, then the

JMP-inner-product is defined by

$$\langle f, g \rangle_{(\underline{\alpha},\underline{\beta})} = \sum_{k=1}^{nb} \langle f, g \rangle_{(\alpha_k,\beta_k)} . \qquad (2.2.4)$$

Definition 2.2.5 Let $P_n^{(\underline{\alpha},\underline{\beta})}$ and $Q_n^{(\underline{\alpha},\underline{\beta})}$ denote the Hilbert spaces defined by the pairs $\{ P_n(\Sigma^*(A)), \langle *, * \rangle_{(\underline{\alpha},\underline{\beta})} \}$, and $\{ Q_n(\Sigma^*(A)), \langle *, * \rangle_{(\underline{\alpha},\underline{\beta})} \}$, respectively. Then $P_{n-1}^{(\underline{\alpha},\underline{\beta})}$ is the space of JMP precondtioner polynomials and $Q_n^{(\underline{\alpha},\underline{\beta})}$ the space of JMP preconditioned polynomials.

In the polynomial preconditioner literature [2,3,17,24,25,27,30,31] the ideal preconditioned function f(x) is assumed to be the constant one restricted to a spectral cover of the matrix to be preconditioned. Here we generalize f(x) to be a piecewise-polynomial with respect to the spectral cover as follows.

Definition 2.2.6 (**Ideal Preconditioned Functions**) If n is a positive integer, and $\Sigma(A)$ a spectral cover for A, then $PP_{2*n+1}(\Sigma^*(A))$ denotes the set of functions f(x) such that f(x) restricted to each eigenband of $\Sigma(A)$ is a polynomial of degree $\leq 2*n+1$.

It is clear that the set $PP_{2*n+1}(\Sigma^*(A))$ is a vector space.

We next define several linear maps, which pertain to JMP approximation of functions in $PP_{2*n+1}(\Sigma^*(A))$.

Definition 2.2.7 If $\{p_j^{(\underline{\alpha},\underline{\beta})}(x)\}_{j=0}^{n}$ is an orthonormal basis for the Hilbert space $P_n^{(\underline{\alpha},\underline{\beta})}$, then let $\pi_n^{(\underline{\alpha},\underline{\beta})}: PP_{2*n+1}(\Sigma^*(A)) \rightarrow P_n^{(\underline{\alpha},\underline{\beta})}$ be the linear map defined by

$$\pi_n^{(\underline{\alpha},\underline{\beta})}(f)(x) = \sum_{j=0}^{n} \langle f, p_j^{(\underline{\alpha},\underline{\beta})} \rangle_{(\underline{\alpha},\underline{\beta})} p_j^{(\underline{\alpha},\underline{\beta})}(x). \qquad (2.2.5)$$

Definition 2.2.8 Let the vector space isomorphism $\varphi_n : P_{n-1}^{(\underline{\alpha},\underline{\beta})} \rightarrow Q_n^{(\underline{\alpha},\underline{\beta})}$ be defined by

$$\varphi_n(p(x)) = x\ p(x). \qquad (2.2.6)$$

Note that φ_n associates to each preconditioner polynomial a preconditioned polynomial.

Definition 2.2.9 Let $\iota_n : Q_n^{(\vec{\alpha},\vec{\beta})} \rightarrow P_n^{(\vec{\alpha},\vec{\beta})}$ denote the linear inclusion map , which maps the n-dimensional subspace $Q_n^{(\underline{\alpha},\underline{\beta})}$ into the n+1-dimensional space $P_n^{(\underline{\alpha},\underline{\beta})}$.

Using the three linear maps φ_n, π_n, and ι_n , we get the following JMP approximation diagram.

$$\begin{array}{ccccc} & & & & PP_{2*n+1} \\ & \varphi_n & & \iota_n & \downarrow \pi_n \\ P_{n-1}^{(\underline{\alpha},\underline{\beta})} & \rightarrow & Q_n^{(\underline{\alpha},\underline{\beta})} & \rightarrow & P_n^{(\underline{\alpha},\underline{\beta})} \end{array}$$

JMP Approximation Diagram

JMP polynomial preconditioners are constructed as least-squares approximations of polynomials in $P_n^{(\underline{\alpha},\underline{\beta})}$ from the subspace $Q_n^{(\underline{\alpha},\underline{\beta})}$. The following facts are immediate from least-squares approximation theory.

(1) The least-squares approximation problem

$$q_n^{(\underline{\alpha},\underline{\beta})}(f)(x) = \min_{q \in Q_n^{(\underline{\alpha},\underline{\beta})}} \| \pi_n^{(\underline{\alpha},\underline{\beta})}(f) - q \|_{(\underline{\alpha},\underline{\beta})} \qquad (2.2.7)$$

has a unique solution.

(2) $q_n^{(\underline{\alpha},\underline{\beta})}(f)(x)$ is the orthogonal projection of $\pi_n^{(\underline{\alpha},\underline{\beta})}(f)$ onto $Q_n^{(\underline{\alpha},\underline{\beta})}$ and the approximation error is orthogonal to $Q_n^{(\underline{\alpha},\underline{\beta})}$.

(3) If $\{q_j^{(\underline{\alpha},\underline{\beta})}(x)\}_{j=1}^n$ is an orthonormal basis for $Q_n^{(\underline{\alpha},\underline{\beta})}$, then the finite expansion

$$q_n^{(\underline{\alpha},\underline{\beta})}(f)(x) = \sum_{j=1}^{n} \langle \pi_n^{(\underline{\alpha},\underline{\beta})}(f), q_j^{(\underline{\alpha},\underline{\beta})} \rangle_{(\underline{\alpha},\underline{\beta})} q_j^{(\underline{\alpha},\underline{\beta})}(x) \qquad (2.2.8)$$

is unique.

2.3 JMP Stieltjes Procedure.

We now define a Gram-Schmidt process for constructing an orthonormal basis for the polynomial space $Q_n^{(\underline{\alpha},\underline{\beta})}$. From the theory of orthogonal polynomials, it is known that every system of orthogonal polynomials satisfies a recurrence of the form

$$q_{j+1}(x) = (x - \sigma_j) * q_j(x) - \tau_j * q_{j-1}(x) \qquad (2.3.1)$$

such that the recurrence coefficients σ_j and τ_j are real constants with $\tau_j > 0$.

One classical algorithm for generating the recursion coefficients of orthogonal polynomials is the Stieltjes procedure [14,32]. The Classical Stieltjes Procedure can be described as follows:

1. (Start) $\hat{q}_0(x) = 1$, $q_0(x) = \hat{q}_0(x) / \| \hat{q}_0 \|$, $\tau_0 = 0$, and $q_{-1}(x) = 0$.

2. (Iterate) For $n = 0$ to maxits set
   $\sigma_n = \langle x q_n, q_n \rangle$
   $\hat{q}_{n+1}(x) = (x - \sigma_n)\, q_n(x) - \tau_n\, q_{n-1}(x)$
   $\tau_{n+1} = \| \hat{q}_{n+1}(x) \|$
   $q_{n+1}(x) = (1/\tau_{n+1})\, \hat{q}_{n+1}(x)$.

The numerical stability of the Stieltjes procedure depends on the accuracy of the parameters σ_n and τ_n. Our method generalizes a procedure given in [24]. It consists of constructing an orthogonal expansion of each JMP basis polynomial in the local basis of scaled and shifted Jacobi polynomials, which allows exact computation of σ_n and τ_n using classical Gauss-Jacobi quadratures.

Our modification of the clasical Stieltjes procedure leads to the formulation of effective polynomial matrix preconditioners. In particular, an ideal preconditioned function f(x) is first expanded in the local Jacobi polynomial bases. Using a formula developed in the next section, we then obtain expansion coefficients of f(x) with repect to the global JMP polynomial basis constructed by our Stieltjes procedure.

To establish notation, we include here a characterization of the classical Jacobi polynomials defined on [-1,1] [33].

Lemma 2.3.1 (Classical Jacobi Polynomials) If $\alpha > -1$, and $\beta > -1$, and m is a positive integer, then let $P_n^{(\alpha,\beta)}$ denote the classical Jacobi polynomial of degree not exeeding m on the interval [-1,1]. Then if $n \leq m$, the three-term linear recurrence

$$P_n^{(\alpha,\beta)} = (A_n x + B_n) \; P_{n-1}^{(\alpha,\beta)} - C_n \; P_{n-2}^{(\alpha,\beta)}, \qquad (2.3.4)$$

with $P_0^{(\alpha,\beta)}(x) = 1$, $P_1^{(\alpha,\beta)}(x) = .5(\alpha+\beta)+2)x+.5(\alpha-\beta)$,

$$A_n = \frac{(2n+\alpha+\beta-1)(2n+\alpha+\beta)}{2n(n+\alpha+\beta)}, \qquad (2.3.5)$$

$$B_n = \frac{(\alpha^2-\beta^2)(2n+\alpha+\beta-1)}{2n(n+\alpha+\beta)(2n+\alpha+\beta-2)}, \qquad (2.3.6)$$

$$C_n = \frac{(n+\alpha-1)(n+\beta-1)(2n+\alpha+\beta)}{2n(n+\alpha+\beta)(2n+\alpha+\beta-2)} \qquad (2.3.7)$$

holds.

Using standard notation for Jacobi polynomials [33] , our basis polynomials on the eigenband $[a_k,b_k]$ are denoted by

$$P_n^{(\alpha_k,\beta_k)}(z_k), \qquad (2.3.8)$$

where

$$z_k = \frac{x - c_k}{d_k} \qquad (2.3.9)$$

$$c_k \equiv (a_k+b_k)/2, \qquad (2.3.10)$$

and

$$d_k \equiv (b_k-a_k)/2 . \qquad (2.3.11)$$

An orthornormal set of Jacobi polynomials is derived from the basis set $\{ P_n^{(\alpha,\beta)} \}$ using the scaling

$$p_n^{(\alpha_k,\beta_k)}(x) \equiv \frac{s_n}{\sqrt{d}} P_n^{(\alpha_k,\beta_k)}(z_k) \qquad (2.3.12)$$

where the scale factors are defined for $n \geq 0$ as

$$s_n = \sqrt{\frac{2n+\alpha+\beta+1 \;\; \Gamma(n+1)\,\Gamma(n+\alpha+\beta+1)}{2^{\alpha+\beta+1}\,\Gamma(n+\alpha+1)\,\Gamma(n+\beta+1)}}.$$

Rearranging the recurrence (2.3.4) for $p_n^{(\alpha_k,\beta_k)}(x)$ and setting

$$E_j^k \equiv \frac{Sj}{Aj+1\;Sj+1}, \quad F_j^k \equiv \frac{-Bj+1}{Aj+1}, \quad G_j^k \equiv \frac{Cj+1\;Sj}{Aj+1\;Sj-1} \qquad (2.3.13)$$

we obtain the **expansion-recurrence** formulas.

$$z_k\, p_0^{(\alpha_k,\beta_k)} = F_0^k\, p_0^{(\alpha_k,\beta_k)} + E_0^k\, p_1^{(\alpha_k,\beta_k)} \qquad (2.3.14)$$

$$z_k\, p_j^{(\alpha_k,\beta_k)} = E_{j-1}^k\, p_{j-1}^{(\alpha_k,\beta_k)} + F_j^k\, p_j^{(\alpha_k,\beta_k)} + E_j^k\, p_{j+1}^{(\alpha_k,\beta_k)} \qquad (2.3.15)$$

Using (2.3.14) and (2.3.15) in Lemma 2.3.1 gives the expansion-recurrence coefficients:

$$E_0^k = \frac{2}{\alpha_k+\beta_k+2}\sqrt{\frac{(\alpha_k+1)(\beta_k+1)}{\alpha_k+\beta_k+3}} \qquad F_0^k = \frac{\beta_k-\alpha_k}{\alpha_k+\beta_k+2}, \qquad (2.3.16)$$

$$E_j^k = \frac{2}{2j+\alpha_k+\beta_k+2}\sqrt{\frac{(j+1)(j+\alpha_k+1)(j+\beta_k+1)(j+\alpha_k+\beta_k+1)}{(2j+\alpha_k+\beta_k+1)(2j+\alpha_k+\beta_k+3)}}, \qquad (2.3.17)$$

$$F_j^k = \frac{(\beta_k^2-\alpha_k^2)}{(2j+\alpha_k+\beta_k)\,(2j+\alpha_k+\beta_k+2)}. \qquad (2.3.18)$$

To start the JMP Stieltjes procedure for constructing an orthonormal basis for $Q_n^{(\underline{\alpha},\underline{\beta})}$, we must expand x in the local Jacobi polynomial basis on each eigenband. To expand x with respect to the local basis on the eigenband $[a_k,b_k]$

$$p_0^{(\alpha_k,\beta_k)} = \frac{s0}{\sqrt{d}} \text{ and } p_1^{(\alpha_k,\beta_k)} = \frac{s1}{\sqrt{d}}\{\frac{\alpha_k+\beta_k+2}{2} z_k + \frac{\alpha_k-\beta_k}{2}\},$$

use the inverse of the linear map (2.3.9) to obtain

$$\begin{aligned} x &= c + dz_k = \overset{\wedge k}{\gamma_0}\, p0 + \overset{\wedge k}{\gamma_1}\, p1 \\ &= \overset{\wedge k}{\gamma_0}\, \frac{s0}{\sqrt{d}} + \overset{\wedge k}{\gamma_1}\, \frac{s1}{\sqrt{d}}\, \frac{\alpha_k-\beta_k}{2} + \overset{\wedge k}{\gamma_1}\, \frac{s1}{\sqrt{d}}\, \frac{\alpha_k+\beta_k+2}{2}\, z_k . \end{aligned}$$

Thus we obtain

$$\overset{\wedge k}{\gamma_0} = \{ c_k + \frac{d_k\,(\beta_k-\alpha_k)}{\alpha_k+\beta_k+2} \}\, \frac{\sqrt{d}}{s0} \quad \text{and} \quad \overset{\wedge k}{\gamma_1} = \{ \frac{2}{\alpha_k+\beta_k+2} \}\, \frac{3/2\sqrt{d}}{s1} . \qquad (2.3.19)$$

We next state the basic JMP expansion Theorem.

Lemma 2.3.1: (JMP Expansion Theorem) Suppose $q(x) \in Q_n^{(\underline{\alpha},\underline{\beta})}$, and for each eigenband $[a_k,b_k]$ assume the existence of the orthonormal expansion

$$q(x) = \sum_{j=0}^{n} \overset{k}{\gamma_j}\, p_n^{(\alpha_k,\beta_k)}(x). \qquad (2.3.20)$$

Set $$s_k \equiv \sum_{j=0}^{n} \overset{k}{\gamma_j}{}^2 , \qquad (2.3.21)$$

and $$t_k \equiv \sum_{j=1}^{n} F_j^k\, \overset{k}{\gamma_j}{}^2 + \sum_{j=1}^{n-1} 2 * E_j^k\, \overset{k}{\gamma_j}\, \overset{k}{\gamma_{j+1}}. \qquad (2.3.22)$$

Then
$$\langle q, q \rangle_{(\underline{\alpha},\underline{\beta})} = \sum_{k=1}^{nb} s_k \tag{2.3.23}$$

and
$$\langle x\,q, q \rangle_{(\underline{\alpha},\underline{\beta})} = \sum_{k=1}^{nb} c_k s_k + d_k t_k. \tag{2.3.24}$$

Proof: From Parseval's Theorem $s_k = \langle q, q \rangle_{(\alpha k,\beta k)}$ on each eigenband. Formula (2.3.23) follows from summing over the eigenbands. The expansion (2.3.20) for q(x) implies that

$$z_k\, q(x) = \sum_{j=0}^{n} {}^{k}\gamma_j\, z_k\, p_n^{(\alpha_k,\beta_k)}(x).$$

Using the expansion recurrence (2.2.34) with the above equation yields

$$t_k = \langle z_k\, q, q \rangle_{(\alpha_k,\beta_k)}.$$

Finally, the equation $x = d\, z_k + c$, and the equation

$$\langle x\,q, q \rangle_{(\alpha_k,\beta_k)} = d_k \langle z_k\, q, q \rangle_{(\alpha_k,\beta_k)} + c_k \langle q, q \rangle_{(\alpha_k,\beta_k)}$$

imply (2.3.24). □

Immediately, we have a method for computing the JMP Stieltjes recurrence cooefficients.

Theorem 2.3.1: (JMP Stieltjes Recurrence Coefficients) If, for the n-th step of the JMP Stieltjes procedure, one has the expansions

$$q_n^{(\underline{\alpha},\underline{\beta})}(x) = \sum_{j=0}^{n} {}^{k}\gamma_j(n)\, p_n^{(\alpha_k,\beta_k)}(x),$$

$$\hat{q}_{n+1}^{(\underline{\alpha},\underline{\beta})}(x) = \sum_{j=0}^{n+1} {}^{\wedge k}\gamma_j(n+1)\, p_n^{(\alpha_k,\beta_k)}(x),$$

then setting $\hat{s}_k \equiv \sum_{j=0}^{n} {}^{\wedge k}\gamma_j(n+1)^2$, the JMP Stieltjes Procedure recurrence coefficients may be calculated by

$$\sigma_n = \sum_{k=1}^{nb} c_k s_k + d_k t_k \quad \text{and} \quad \tau_{n+1} = \sqrt{\sum_{k=1}^{nb} \hat{s}_k}. \qquad (2.3.25)$$

We next give a recurrence, which provides orthogonal expansions for the JMP polynomials $\hat{q}_{n+1}^{(\underline{\alpha},\underline{\beta})}(x)$ given orthogonal expansions for $q_n^{(\underline{\alpha},\underline{\beta})}(x)$ and $q_{n-1}^{(\underline{\alpha},\underline{\beta})}(x)$. The proof is obvious, but tedious.

Theorem 2.3.2 (Expansion Coefficient Recurrence) Suppose, for each $[a_k,b_k]$, we have the expansions

$$q_{n-1}^{(\underline{\alpha},\underline{\beta})}(x) = \sum_{j=0}^{n-1} \gamma_j^k(n-1)\, p_j^{(\alpha_k,\beta_k)}(x),$$

and

$$q_n^{(\underline{\alpha},\underline{\beta})}(x) = \sum_{j=0}^{n} \gamma_j^k(n)\, p_j^{(\alpha_k,\beta_k)}(x).$$

Then

$$\hat{q}_{n+1}^{(\underline{\alpha},\underline{\beta})}(x) = \sum_{j=0}^{n+1} \hat{\gamma}_j^k(n+1)\, p_j^{(\alpha_k,\beta_k)}(x),$$

where

$$\hat{\gamma}_0^k(n+1) = (c_k - \sigma_n)\,\gamma_0^k(n) + d_k\,(\gamma_0^k(n)\,F_0^k + \gamma_1^k(n)\,E_0^k) - \tau_n\,\gamma_0^k(n-1), \qquad (2.3.26)$$

and for $1 \le j \le n-1$,

$$\hat{\gamma}_j^k(n+1) = (c_k - \sigma_n)\,\gamma_j^k(n) + d_k\,(\gamma_{j-1}^k(n)\,E_{j-1}^k + \gamma_j^k(n)\,F_j^k + \gamma_{j+1}^k(n)\,E_j^k) - \tau_n\,\gamma_j^k(n-1) \qquad (2.3.27)$$

$$\hat{\gamma}_n^k(n+1) = (c_k - \sigma_n)\,\gamma_n^k(n) + d_k\,(\gamma_{n-1}^k(n)\,E_{n-1}^k + \gamma_n^k(n)\,F_n^k) \qquad (2.3.28)$$

$$\hat{\gamma}_{n+1}^k(n+1) = d_k\,(\gamma_n^k(n)\,E_n^k). \qquad (2.3.29)$$

Our Stieltjes procedure is constructed as follows.

JMP Stieltjes Procedure

Inputs:

(1) a spectral cover $\{ [a_k,b_k] \}_{k=1}^{nb}$

(2) the vectors $\underline{\alpha} = (\alpha_1,..,\alpha_{nb})^t$ and $\underline{\beta} = (\beta_1,..,\beta_{nb})^t$ which define the Jacobi polynomials used on $[a_k,b_k]$

(3) n = dimension of the JMP approximation space $Q_n^{(\underline{\alpha},\underline{\beta})}$

Output:

(1) the recurrence coefficients $\{ \sigma_j \}_{j=0}^{n-1}$, $\{ \tau_j \}_{j=1}^{n-1}$

(2) expansion coefficients $\gamma_j^k(m)$ for each orthonormal JMP basis polynomial $q_m^{(\underline{\alpha},\underline{\beta})}(x)$, and for each eigenband $[a_k,b_k]$

$$q_m^{(\underline{\alpha},\underline{\beta})}(x) = \sum_{j=0}^{m+1} \gamma_j^k(m)\, p_j^{(\alpha_k,\beta_k)}(x).$$

Initialization:

(1) Calculate eigenband centers and half-bandwiths
$c_k = .5 (a_k + b_k)$ and $d_k = .5 (b_k - a_k)$;

(2) Calculate expansion-recurrence coefficients E_j^k, and F_j^k;

(3) Calculate expansion coefficients of x in each band

$$\hat{\gamma}_0^k(0) = \{ c_k + \frac{d_k\ (\beta_k-\alpha_k)}{\alpha_k+\beta_k+2} \} \frac{\sqrt{d}}{s0} \quad \text{and} \quad \hat{\gamma}_1^k(0) = \{ \frac{2}{\alpha_k+\beta_k+2} \} \frac{3/2\sqrt{d}}{s1};$$

(4) Calculate $\langle x, x \rangle_{(\underline{\alpha},\underline{\beta})} = \sum_{k=1}^{nb} \hat{s}_k$ with $\hat{s}_k = \sum_{j=0}^{1} \hat{\gamma}_1^k(0)^2$;

(5) Calculate expansion coefficients of normalized x

$\gamma_0^k(0) = \hat{\gamma}_0^k(0) / \langle x, x \rangle_{(\underline{\alpha},\underline{\beta})}$ and $\gamma_1^k(0) = \hat{\gamma}_1^k(0) / \langle x, x \rangle_{(\underline{\alpha},\underline{\beta})}$;

(6) Calculate $t_k = \langle z_k q_0, q_0 \rangle_{(\alpha_k,\beta_k)}$ using (2.3.35) ;

(7) Set $\sigma_0 = 1./ \| x \|_{(\underline{\alpha},\underline{\beta})}$.

Iteration: For n = 0 to maxits (= degree of preconditioned polynomial)

(1) Compute Stieltjes recurrence coefficient σ_{n+1} using

$$\sigma_{n+1} = \sum_{k=1}^{nb} c_k s_k + d_k t_k \ ;$$

(2) Compute expansion coefficients $\hat{\gamma}_j^k(n)$, j=0, n+1 for $\hat{q}_{n+1}^{(\underline{\alpha},\underline{\beta})}(x)$;

(3) Compute new $\hat{s}_k$ for each eigenband ;

(4) Compute Stieltjes recurrence coefficient τ_{n+1} using

$$\tau_{n+1} = \sqrt{\sum_{k=1}^{nb} \hat{s}_k} \ ;$$

(5) Compute expansion coefficients $\gamma_j^k(n)$, j=0, n+1 for $q_{n+1}^{(\underline{\alpha},\underline{\beta})}(x)$ using

$$\gamma_j^k(n) = \hat{\gamma}_j^k(n) \ / \ \tau_{n+1}^2 \ ;$$

(6) Compute s_k for each band using $s_k = \hat{s}_k \ / \ \tau_{n+1}^2$;

(7) Compute t_k for each eigenband using

$$t_k \equiv \sum_{j=0}^{n+1} F_j^k \ \gamma_j^{k2} + \sum_{j=0}^{n} 2 * E_j^k \ \gamma_j^k \ \gamma_{j+1}^k \ .$$

2.4 JMP Orthogonal Expansions.

Having constructed an orthonormal basis for the JMP approximation space $Q_n^{(\underline{\alpha},\underline{\beta})}$, we now wish to construct an orthogonal expansion for an ideal preconditioned function $f(x) \in PP_{2*n+1}(\Sigma^*(A))$. This necessitates defining a method for computing the local inner-products (2.4.1).

$$\langle f, p_j^{(\alpha_k,\beta_k)} \rangle_{(\alpha_k,\beta_k)} = \int_{a_k}^{b_k} f(x)\, p_j^{(\alpha_k,\beta_k)}(x)\, w^{(\alpha_k,\beta_k)}(x)\, dx \qquad (2.4.1)$$

These inner-products are defined by classical Gauss-Jacobi quadrature, which is summarized in the following theorem [33].

Theorem 2.4.1 (Gauss-Jacobi Quadrature) If f(x) restricted to [a,b] is a polynomial of degree $\leq 2n+1$, and $\{ x_n^{(\alpha_k,\beta_k)} \}_{j=1}^{n}$ denotes the n real-zeroes (the Gauss-points) of the polynomial $p_n^{(\alpha_k,\beta_k)}(x)$, then there exists a real number sequence $\{ c_i \}_{i=1}^{n}$ (the Christoffel numbers) such that the Gauss-Jacobi quadrature formula

$$\int_a^b w_k(x)\, q(x)\, p_n^{(\alpha_k,\beta_k)}(x)dx = \sum_{i=1}^{n} c_i\, p_n^{(\alpha_k,\beta_k)}(x_n^{(\alpha_k,\beta_k)}) \qquad (2.4.2)$$

is exact, and the Christoffel numbers are positive and are computed by

$$c_i = \frac{1.}{\sum_{j=0}^{n} p_j^{(\alpha_k,\beta_k)}(x_i^{(\alpha_k,\beta_k)})^2}. \qquad (2.4.3)$$

We use the method of Golub-Welsch [16] for computing the Gauss-points $\{ x_n^{(\alpha_k,\beta_k)} \}_{j=1}^{n}$ and the Christoffel numbers $\{ c_i \}_{i=1}^{n}$. This method expresses the expansion recurrence (2.3.14)-(2.3.15) as a matrix equation, which we illustrate for the case n = 3 :

$$\begin{bmatrix} F0 & E0 & 0 \\ E0 & F1 & E1 \\ 0 & E1 & F2 \end{bmatrix} \begin{bmatrix} p_0^{(\alpha_k,\beta_k)}(t) \\ p_1^{(\alpha_k,\beta_k)}(t) \\ p_2^{(\alpha_k,\beta_k)}(t) \end{bmatrix} = t \begin{bmatrix} p_0^{(\alpha_k,\beta_k)}(t) \\ p_1^{(\alpha_k,\beta_k)}(t) \\ p_2^{(\alpha_k,\beta_k)}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ p_3^{(\alpha_k,\beta_k)}(t) \end{bmatrix} . \qquad (2.4.4)$$

If the number t is a zero of $p_3^{(\alpha_k,\beta_k)}$, then the scalar t and the vector (2.4.5) are an eigenpair of the symmetric tridiagonal matrix in (2.4.4). Therefore, computing the eigenvalues and eigenvectors of this matrix, and using the fact that $p_0^{(\alpha_k,\beta_k)}(t) = s_0$, permits scaling the eigenvector $(v_0,v_1,v_2)^t$ to have the form

$$\begin{bmatrix} p_0^{(\alpha_k,\beta_k)}(t) \\ p_1^{(\alpha_k,\beta_k)}(t) \\ p_2^{(\alpha_k,\beta_k)}(t) \end{bmatrix} . \qquad (2.4.5)$$

Using the scale factor s_0/v_0, which yields the formula

$$c_i = \frac{1.}{\frac{s_0^2}{v_0^2} \sum_{j=0}^{n} v_j^2} \qquad (2.4.6)$$

for computing the Christoffel-number c_i .

With the Golub-Welsch method for computing the Gauss-points and Christoffel numbers, we can now outline a four-step JMP approximation procedure for a function $f(x) \in PP_{2*n+1}(\Sigma^*(A))$ from the JMP approximation space $Q_n^{(\underline{\alpha},\underline{\beta})}$.

JMP-approximation Algorithm:

(1) Use JMP Stieltjes procedure to obtain the expansions, $m = 0$, maxits for each eigenband $[a_k,b_k]$

$$q_n^{(\underline{\alpha},\underline{\beta})}(x) = \sum_{j=0}^{m+1} \gamma_j^k(m)\, p_j^{(\alpha_k,\beta_k)}(x);$$

(2) Use Golub-Welsch method to compute the Gauss-points $\{ x_j^{(\alpha_k,\beta_k)} \}_{j=1}^{n}$ and Christoffel-numbers $\{ c_i^k \}_{i=1}^{n}$ for each $[a_k,b_k]$;

(3) Use Gauss-Jacobi quadrature formula to obtain the expansion coefficients for the polynomial $\pi_n(f(x))$, for each eigenband $[a_k,b_k]$

$$\pi_n(f(x)) = \sum_{j=0}^{n} \gamma_j^k(f(x))\, p_j^{(\alpha_k,\beta_k)}(x) ;$$

(4) Compute expansion coefficients for the JMP polynomial

$$q_n^{(\underline{\alpha},\underline{\beta})}(f)(x) = \sum_{j=1}^{n} \langle \pi_n^{(\underline{\alpha},\underline{\beta})}(f) , q_j^{(\underline{\alpha},\underline{\beta})} \rangle_{(\underline{\alpha},\underline{\beta})}\, q_j^{(\underline{\alpha},\underline{\beta})}(x),$$

using the formula

$$\langle \pi_n^{(\underline{\alpha},\underline{\beta})}(f) , q_j^{(\underline{\alpha},\underline{\beta})} \rangle_{(\underline{\alpha},\underline{\beta})} = \sum_{k=1}^{nb} \sum_{i=0}^{j+1} \gamma_i^k(j)\, \gamma_i^k(f(x)).$$

2.5 JMP Preconditioner Matrices.

We now give the ingredients for constructing JMP preconditioner matrices.

Definition 2.5.1 A **JMP preconditioner-scheme** for a real symmetric matrix A consists of the 6-tuple

$$(\Sigma(A) , nb, f(x), \underline{\alpha} ,\underline{\beta} ,n) \tag{2.5.1}$$

such that

(1) $\Sigma(A)$ is a spectral cover for A consisting of nb intervals,

(2) $f(x) \in PP_{2*n+1} (\Sigma^*(A))$ is the ideal preconditioned function to be approximated,

(3) the vectors $\underline{\alpha} = (\alpha_1,..,\alpha_{nb})^t$ and $\underline{\beta} = (\beta_1,..,\beta_{nb})^t$ which define the Jacobi polynomials used on $[a_k,b_k]$

(4) n is the degree of the preconditioned polynomial $q_n^{(\underline{\alpha},\underline{\beta})}(f)(x) = x\, p_{n-1}^{(\underline{\alpha},\underline{\beta})}(f)(x)$.

The polynomial matrix $p_{n-1}^{(\underline{\alpha},\underline{\beta})}(f)(A)$ is called a JMP preconditioner matrix for A, and the polynomial matrix $q_n^{(\underline{\alpha},\underline{\beta})}(f)(A)$ is called a JMP preconditioned matrix for A. We next describe three JMP preconditioner schemes, which are used to solve ESP linear sytems. The first scheme constructs an approximation to the inverse of a SPD matrix, the second scheme constructs an approximation to the inverse of a Choleski factor of a SPD matrix, and the third scheme constructs an approximation to the inverse of an indefinite matrix on its positive eigenspace and to minus its inverse on its negative eigenspace.

The Single-Function Scheme for the SPD Matrix A

This Scheme consists of the following:

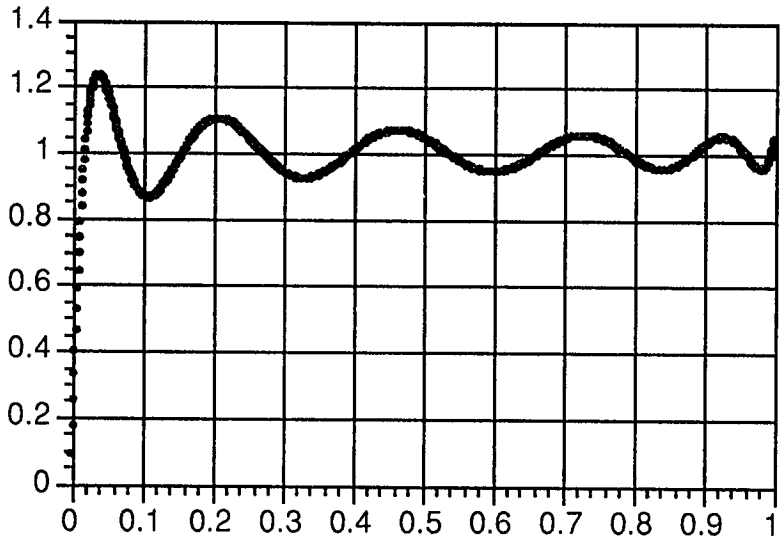
(1) { [a,b] } is a spectral cover for the SPD matrix A

(2) $f(x) = 1, \quad x \in [a,b]$

(3) α and β based upon the desired kind of JMP approximation to f(x).

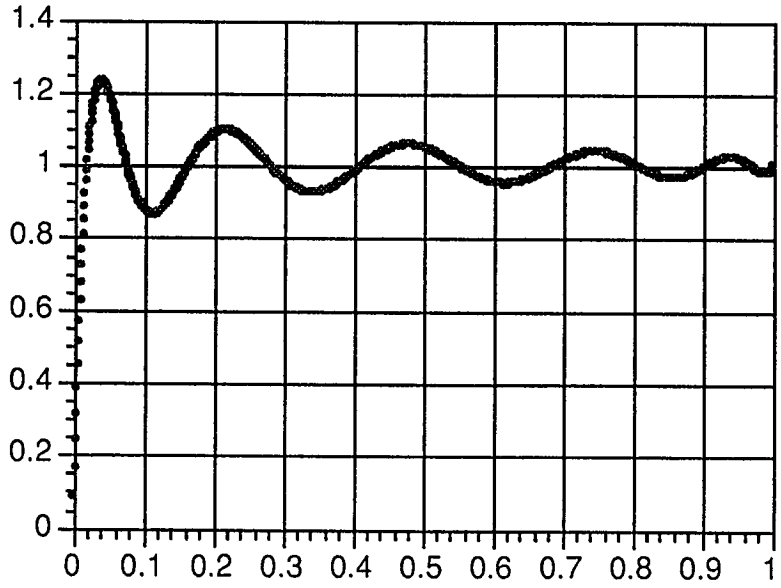
This polynomial preconditioner has been discussed by a variety of authors [3], [17,25,31]. We include the graphs of two JMP preconditioned polynomials for the spectral cover { [.001,1] } to illustrate a biased approximation of the function f(x) =1.

The first polynomial is based on Jacobi polynomials $p^{(\alpha,\beta)}(x)$ with $\alpha = -.5$ and $\beta = -.5$, and the second polynomial uses $\alpha = -.95$, and $\beta = -5$.



Single-Function Preconditioned Polynomial

degree = 11, $\alpha = -.5$, $\beta = -.5$.



Single-Function Preconditioned Polynomial

degree = 11, α = -.95, β = -.5 .

The JMP Choleski Scheme for the SPD Matrix A consists of the following:

(1) { [a,b] } is a spectral cover for the SPD matrix A ;

(2) $f(x) = \sqrt{x}$, $x \in [a,b]$;

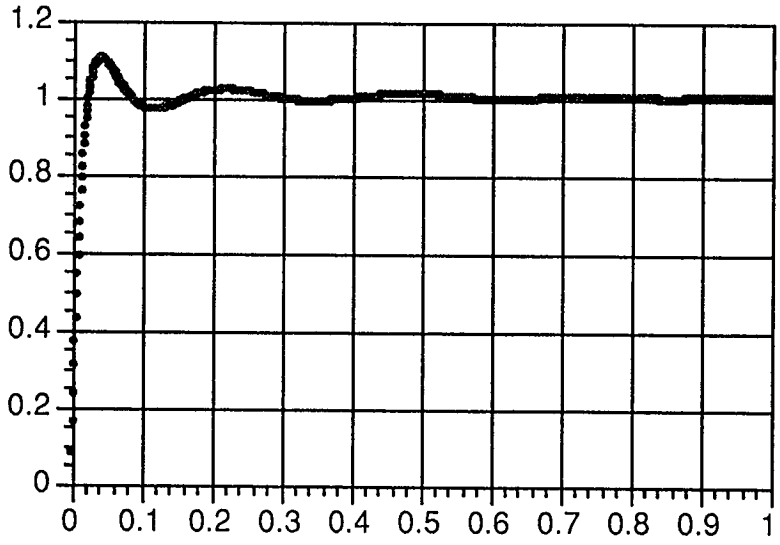
(3) α and β based upon the desired kind of approximation to f(x).

The JMP preconditioned polynomial $q_n^{(\alpha,\beta)}(f)(x)$ approximates the function $\sqrt{x}$ on [a,b], and the JMP preconditioner polynomial $p_{n-1}^{(\alpha,\beta)}(f)(x)$ approximates $\frac{1}{\sqrt{x}}$ on [a,b].

This JMP Choleski preconditioner matrix $p_{n-1}^{(\underline{\alpha},\underline{\beta})}(f)(A)$ is used for split preconditioning of an SPD matrix A, and is applied symmetrically to the matrix as

$$p_{n-1}^{(\underline{\alpha},\underline{\beta})}(f)(A) \quad A \quad p_{n-1}^{(\underline{\alpha},\underline{\beta})}(f)(A). \tag{2.5.2}$$

To our knowledge the Choleski polynomial preconditioner has not appeared in the polynomial preconditioner literature. We include a graph of the JMP Choleski symmetrically preconditioned polynomial with $\alpha = 0.$ and $\beta = 0.$ Notice that the oscillations are much damped compared to the single function preconditioned polynomial. This is due to the well studied phenomena associated with approximating a function near a simple discontinuity.



Choleski Symmetrically Preconditioned Polynomial

degree = 11, $\alpha = 0.$, $\beta = 0.$

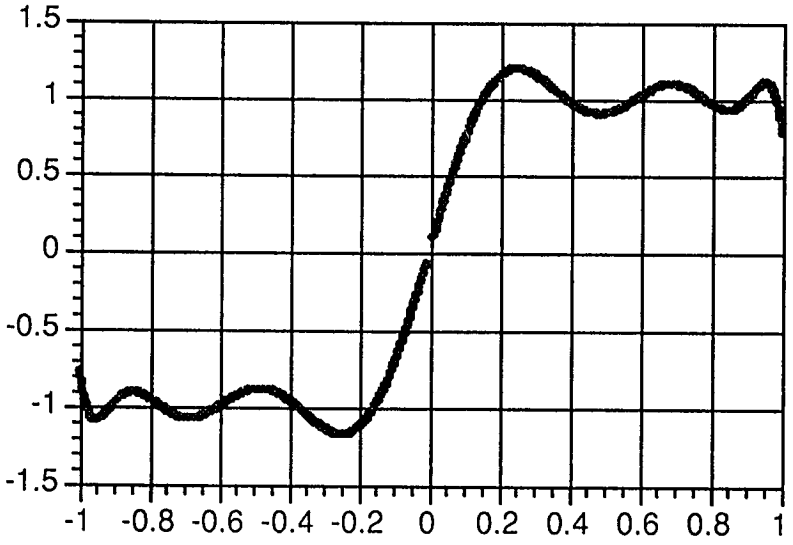
The Two-Function Scheme for the Symmetric Indefinite Matrix A consists of the following:

(1) $\Sigma(A) = \{ [a,b] , [c,d] \}$ is a spectral cover for A ;

(2) $f(x) = -1,\ x \in [a,b]$, and $f(x) = 1, x \in [c,d]$;

(3) α and β based upon the desired kind of approximation to f(x).

This scheme approximates -1 on the negative eigenbands and +1 on the positive eigenbands. The JMP preconditioner polynomial matrix is SPD, but the JMP preconditioned polynomial matrix is indefinite just as the un-preconditioned matrix. To our knowledge the Two-Function polynomial preconditioner has not been discussed in the literature. Here is a graph of a JMP Two-Function preconditioned polynomial defined on the spectral cover { [-1.,-.01], [.01,1] } using Legendre polynomials on both eigenbands.



Two-Function Preconditioned Polynomial

degree = 11, $\alpha 1 = 0.$, $\beta 1 = 0.$,

$\alpha 2 = 0.$, $\beta 2 = 0.$

Chapter 3. Preconditioned GMRES Algorithms for ESP Matrices.

In this Chapter we develop several preconditioners for elliptic saddle-point (ESP) matrices. Each preconditioner, ultimately based upon the polynomial matrices described in Chapter 2, requires an estimate for a spectral cover for the submatrix A and the matrix BB^t.

In Section 3.1 we develop a formula for computing a spectral cover for the general saddle-point matrix (1.1) using spectral covers for A, C, and BB^t. In Section 3.2 we describe an efficient method for estimation of spectral covers for A, C, and BB^t, using an Arnoldi iteration. In Section 3.3, we describe preconditioners for A, C, and BB^t based upon JMP polynomial matrices. In Section 3.4 we describe three preconditioners for the Schur complements of ESP matrices, which shall refer to as sandwich matrices. In Section 3.5 we describe three preconditioners for ESP matrices.

3.1 Spectra of ESP Matrices.

We now define more precisely the linear systems we shall investigate.

Definition 3.1.1 The 2 x 2 block-partitioned symmetric (n+m) x (n+m) matrix

$$S = \begin{bmatrix} A & B^t \\ B & -C \end{bmatrix} \tag{3.1.1}$$

is a saddle-point **(SP)** matrix if

(1) A is an n x n symmetric positive-definite (SPD) matrix,

(2) C is an m x m symmetric positive-semi-definite matrix.

We shall focus on the special case (C = 0) of elliptic saddle-point (ESP) matrices with the form

$$\begin{bmatrix} A & B^t \\ B & 0 \end{bmatrix}. \tag{3.1.2}$$

It is well known [13] that for an ESP system

$$\begin{bmatrix} A & B^t \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \tag{3.1.3}$$

there exits a unique vector $u \in \Re^n$ and unique $p \in \Re^m$ such that if $z \in \text{null}(B^t)$, then the pair (u,p+z) is a solution of the system. Moreover, (u,p) is the unique solution iff rank(B) = m.

For an ESP matrix S with Schur complement $B\,A^{-1}\,B^t$, let $k = \dim \text{null}(B^t)$. Then applying Sylvester's Law of Inertia Theorem [15],

and the block L-D-L^t matrix factorization

$$\begin{bmatrix} A & B^t \\ B & 0 \end{bmatrix} = \begin{bmatrix} A & 0 \\ B & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & -B\,A^{-1}\,B^t \end{bmatrix} \begin{bmatrix} A & B^t \\ 0 & I \end{bmatrix} \tag{3.1.8}$$

we obtain

(1) S has exactly n positive eigenvalues (since A does),

(2) S has exactly k zero eigenvalues (since B^t does),

(3) S has exactly m-k negative eigenvalues (since $-B\,A^{-1}\,B^t$ does).

An ESP matrix is, therefore, a symmetric indefinite matrix, whose non-zero eigenvalue spectrum can be bounded as $\{\lambda_j\}_{j=1}^{n+m-k} \subseteq [a,b] \cup [c,d]$ where {a,b,c,d} denote the four extreme eigenvalues.

We now estimate a spectral cover for a general SP matrix, which generalizes a result in [21,22] for ESP matrices.

Theorem 3.1 (Saddle-Point Matrix Spectral Cover Theorem) If the intervals $[\lambda_n, \lambda_1]$, $[\gamma_m, \gamma_1]$, and $[\sigma_m^2, \sigma_1^2]$, are spectral covers for the matrices A, C, and BB^t, respectively, of the non-singular saddle-point matrix S, then

$$a = 1/2\ [\ (\lambda_n - \gamma_1) - \sqrt[2]{(\lambda_n + \gamma_1)^2 + 4\ \sigma_1^2}\ ], \tag{3.1.9a}$$

$$b = 1/2\ [\ (\lambda_1 - \gamma_m) - \sqrt[2]{(\lambda_1 + \gamma_m)^2 + 4\ \sigma_m^2}\ ], \tag{3.1.9b}$$

$$c = \lambda_n, \tag{3.1.9c}$$

$$d = 1/2\ [\ (\lambda_1 - \gamma_m) + \sqrt[2]{(\lambda_1 + \gamma_m)^2 + 4\ \sigma_1^2}\ ]. \tag{3.1.9d}$$

Proof: (See Appendix A).

Immediately, we have the result described in [21].

Corollary 3.1 (ESP Spectral Cover) If S is an ESP matrix, then

$$a = 1/2\ [\lambda_n - \sqrt[2]{\lambda_n^2 + 4\sigma_1^2}\ ], \tag{3.1.10a}$$

$$b = 1/2\ [\lambda_1 - \sqrt[2]{\lambda_1^2 + 4\sigma_m^2}\ ], \tag{3.1.10b}$$

$$c = \lambda_n , \tag{3.1.10c}$$

$$d = 1/2\ [\lambda_1 + \sqrt[2]{\lambda_1^2 + 4\sigma_1^2}\ ]. \tag{3.1.10d}$$

Since $k_2(A) = \lambda_1/\lambda_n$ and $k_2(B) = \sigma_1/\sigma_{m-k}$, the formulas in Corollary 3.1 imply that the interval [a,b] converges to a point and the length of the interval [c,d] converges to σ_1 as $k_2(A)$ and $k_2(B)$ converge to 1.

3.3 Spectral Covers for ESP Matrices.

Corollary 3.1 implies that a spectral cover for an ESP matrix S can be constructed using spectral covers for the SPD matrix A, and the positive semi-definite matrix BB^t.

To estimate a spectral cover for an arbitrary positive semi-definite matrix M, we apply an Arnoldi iterative method to find the matrix's extreme eigenvalues [a,b]. First, we estimate the largest eigenvalue b of M using an Arnoldi iteration. To estimate a is more complicated. First, we construct a low degree (3-5) JMP preconditioner polynomial p(x) defined on the interval [0,b]. Then we apply Arnoldi's iteration to the JMP matrix p(M).

Here the Arnoldi iteration converges to a dominant eigen-vector of p(M). Using Raleigh quotients with respect to M instead of p(M), we estimate the smallest eigenvalue of M.

Experimentation with the matrices in our numerical work in Chapter 4, shows that this method converges in 20 to 30 Arnoldi iterations.

If the matrix M is singular, it is necessary to start with a vector orthogonal to null(M), and to orthogonalize the vector Mv with respect to a basis of null vectors after each matrix-vector multiply, because the JMP polynomial p(x) has large values for x less than or equal to zero.

A practical concern is that if Arnoldi's iteration requires a significant number of iterations for convergence for a large matrix, it will require excessive storage. It is preferable, when solving large systems, to use a scheme such as the Implicitly Restarted Arnoldi algorithm [28,29]. This scheme maintains a fixed sized Arnoldi matrix, and a fixed number of Arnoldi vectors, which conserves central memory, and accelerates convergence, by implicitly applying polynomials to the Arnoldi iteration.

3.3. Preconditioners for Definite Matrices.

Let M denote a symmetric positive semi-definite matrix, and assume an orthonomal basis for null(M) is given. Let $\Sigma(M)$ denote a minimal spectral cover for M, and let p(x) be a JMP preconditioner polynomial defined on $\Sigma(M)$. We now define two preconditioners for M. The first preconditioner is the JMP matrix p(M).

The second precondtioner for M is to iteratively solve $M\ x = b$ using the algebraic splitting

$$M = p(M)^{-1} - (p(M)^{-1} - M). \tag{3.3.1}$$

This splitting produces the linear stationary iteration

$$x^{(n+1)} = \{ I - p(M)\ M \}\ x^{(n)} + p(M)\ b \tag{3.3.2}$$

whose iteration matrix is

$$r(M) = I - p(M)\ M. \tag{3.3.3}$$

We shall use the basic iteration (3.3.2), or Chebyshev acceleration of (3.3.2), if the spectral radius of r(A) is large. In practice reducing the residual to some prescribed tolerance requires only a few iterations.

3.4 Preconditioners for Sandwich Matrices.

A sandwich matrix is defined as follows.

Definition 3.4.1 A matrix with the form

$$B\ M\ B^t \tag{3.4.1}$$

is called a sandwich matrix if the matrix M is an SPD matrix, and a simple sandwich matrix if M is an identity matrix.

It is easily seen that a sandwich matrix is symmetric positive semi-definite.

We shall have interest in three types of sandwich matrix:

(Type-1) $M = \mathrm{diag}(A)$;

(Type-2) $M = A^{-1}$, i.e., M is the inverse of a known matrix ;

(Type-3) $M = p(A)^2$, i.e., M is the square of a polynomial matrix.

Each of these preconditioners for sandwich matrices relies on the fact that the matrix "inside the sandwich" is not terribly ill-conditioned: the ill-conditioning of the sandwich matrix is due to the ill-conditioning of the matrix $B\,B^t$.

Definition 3.4.2 (Type-1 Sandwich preconditioner) Let $\varepsilon > 0$. Assume that D is a diagonal matrix, that BDB^t is a type-1 sandwich matrix. Let p(x) denote a JMP polynomial defined on a spectral cover $\Sigma(BDB^t)$ such that $p(BDB^t)$ is a Choleski JMP matrix preconditioner. Then a preconditioner for the sandwich matrix BDB^t consists of using GMRES to reduce the residual of the split preconditioned equation

$$p(BDB^t)\, BDB^t\, p(BDB^t)\;\; x \;\; = \;\; p(BDB^t)\, b \qquad (3.4.4)$$

less than the tolerance ε.

We next define a preconditioner for sandwich matrices of type-2, which is useful if the matrix $M = A^{-1}$ is not ill-conditioned. This method successively approximately solves two linear systems analogous to the solution of a non-singular square matrix using its LU factorization.

Definition 3.4.3 (Type-2 Sandwich precondtioner) Let $\varepsilon_1 > 0$, and $\varepsilon_2 > 0$. Then a preconditioner for the sandwich matrix $BA^{-1}B^t$ consists of the following 2-step algorithm, which solves the linear systems

$$(1) \qquad B\,A^{-1}\;\; v \;\; = \;\; g, \qquad (3.4.5a)$$

$$(2) \qquad B^t\; p \;\; = \;\; v. \qquad (3.4.5b)$$

by reducing their residuals to ε_1 and ε_2, respectively.

An efficient method for solving (3.4.5a) and (3.4.5b) is to construct a JMP precondtioner matrix $p_1(B\,B^t)$, and use GMRES to solve equations

$$(1) \quad B^t\, p_1(B\,B^t)\, B\, A^{-1}\ v \ = \ B^t\, p_1(B\,B^t)\, g, \tag{3.4.5c}$$

$$(2) \quad B\,B^t\ p_1(B\,B^t)\ p \ = \ B\ v. \tag{3.4.5d}$$

to the tolerances ε_1 and ε_2, respectively.

Finally, we define a preconditioner for sandwich matrices of type-3. This method is a 3-step method, which successively approximately solves three linear systems analogous to the solution of a non-singular square symmetric matrix using its LDL^t factorization.

Definition 3.4.4 (Type-3 Sandwich preconditioner) Let $\varepsilon_1 > 0$, $\varepsilon_1 > 0$, and $\varepsilon_3 > 0$. Then a preconditioner for the sandwich matrix $Bp(A)^2B^t$ consists of the following 3-step algorithm, which solves the linears systems

$$(1) \quad B\,p(A)\ v \ = \ g, \tag{3.4.6a}$$

$$(2) \quad p(A)\ z \ = \ v, \tag{3.4.6b}$$

$$(3) \quad p(A)\ B^t\ p \ = \ z. \tag{3.4.6c}$$

by reducing their residuals to ε_1, ε_2, and ε_3, respectively.

An efficient method for solving (3.4.6d)-(3.4.6f) is to construct a JMP precondtioner matrix $p_1(B\,B^t)$, and use GMRES to solve equations

$$(1) \quad p(A)\ B^t\, p_1(B\,B^t)\, B\, p(A)\ v \ = \ p(A)\ B^t\, p_1(B\,B^t)\, g \tag{3.4.6d}$$

$$(2) \quad p(A)\ z \ = \ v \tag{3.4.6e}$$

$$(3) \quad B\,B^t\ p_1(B\,B^t)\ p \ = \ B\ z. \tag{3.4.6f}$$

to the tolerances ε_1, ε_2, and ε_3, respectively.

If orthogonality is maintained in the Krylov space built in the solution of the vector v in the first step of the 2-step (3-step) sandwich matrix preconditioner, the vector v will be of the form

$$v = B^t p \quad (\; v = p(A)\, B^t p \;). \tag{3.4.7}$$

Consequently, the solution of the final step will approximate the vector p of equation (3.4.7).

We note that the coefficient matrix in the first step of the 2-step sandwich preconditioner is not symmetric, but is symmetric for the3-step preconditioner. If Householder orthogonalization is used in implementing the GMRES algorithm [36], we have more control on enforcing (3.4.7). Experimentation has shown that a robust preconditioner results from solving each step with modest accuracy.

3.5 Preconditioners for ESP Matrices.

We describe here three preconditioners for ESP matrices. Before formulating these preconditioners, we make the following remark, which was observed computationally.

Remark 3.5.1 (**Property-D**) For the lowest order Raviart-Thomas spaces [19], if D denotes the diagonal matrix with entries $D_i = 1./\mathrm{sqrt}(A_{ii})$, then the symmetric Jacobi scaled matrix DAD has a significantly reduced condition number.

The three preconditioners we now define take advantage of this property. Each ESP matrix preconditioner uses a diagonal matrix to symmetricaly precondition the block A, and a Choleski JMP matrix preconditioner for BB^t.

In Method-I we precondition the resulting ESP matrix with a 2-band JMP preconditioner. In Method-II and Method-III we use a block-LU factorization of the resulting ESP matrix, and handle the sandwich matrix, which corresponds to the Schur complement of the ESP matrix. In Method-II we use the Type-1 or Type-2 preconditioner, and in Method-III we use the Type-1 or Type-3 preconditioner.

3.5.1 Method-1.

Let D_a denote the diagonal matrix such that $D_{a\,ii} = \frac{1.}{\sqrt{A_{ii}}}$.

In this method we begin by pre-multiplying and post-multiplying the ESP matrix by a block diagonal matrix to produce the preconditioned ESP matrix

$$\begin{bmatrix} \hat{A} & \hat{B}^t \\ \hat{B} & 0 \end{bmatrix} = \begin{bmatrix} D_a & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A & B^t \\ B & 0 \end{bmatrix} \begin{bmatrix} D_a & 0 \\ 0 & I \end{bmatrix} \qquad (3.5.1.2)$$

where

$$\hat{A} = D_a\, A\, D_a,$$
$$\hat{B} = B\, D_a,$$
$$\hat{B}^t = D_a\, B^t .$$

The second step is to define a JMP preconditioner polynomial p(x) for the type-1 sandwich matrix $\hat{B}\,\hat{B}^t$, and to pre-multiply and post-multiply the ESP matrix (3.5.1.2) by a block diagonal matrix to produce the preconditioned ESP matrix

$$\begin{bmatrix} \hat{A} & \hat{B}^t\, p(\hat{B}\,\hat{B}^t) \\ p(\hat{B}\,\hat{B}^t)\,\hat{B} & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & p(\hat{B}\,B^t) \end{bmatrix} \begin{bmatrix} \hat{A} & \hat{B}^t \\ \hat{B} & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & p(\hat{B}\,\hat{B}^t) \end{bmatrix}. \quad (3.5.1.3)$$

Thus we obtain the preconditioned ESP linear system

$$\hat{S} \begin{bmatrix} \hat{u} \\ \hat{p} \end{bmatrix} = \begin{bmatrix} \hat{A} & \hat{B}^t\, p(\hat{B}\,\hat{B}^t) \\ p(\hat{B}\,\hat{B}^t)\,\hat{B} & 0 \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{p} \end{bmatrix} = \begin{bmatrix} \hat{f} \\ \hat{g} \end{bmatrix} \quad (3.5.1.4)$$

where

$$\begin{aligned} u &= D_a\,\hat{u}, \\ p &= p(\hat{B}\hat{B}^t)\;\hat{p}, \\ \hat{f} &= D_a\, f, \\ \hat{g} &= p(\hat{B}\hat{B}^t)\;\; g. \end{aligned}$$

Let $p_2(x)$ denote a 2-band JMP preconditioner polynomial as defined in Chapter 2 with respect to the matrix $\hat{S}$. Then the Method-1 preconditioner uses the polynomial matrix $p_2(\hat{S})$ as either a left or right preconditioner of the ESP linear system (3.5.1.4). The solution to the original problem (3.5.1.1) is found with the subsitutions:

$$\begin{aligned} u &= D_a\,\hat{u}, \\ p &= p(B\,D_a^2\,B^t)\;\hat{p}\,. \end{aligned}$$

3.5.2 Method II.

In this method we begin by pre-multiplying and post-multiplying the ESP matrix by a block diagonal matrix to produce the preconditioned ESP matrix

$$\begin{bmatrix} \hat{A} & \hat{B}^t \\ \hat{B} & 0 \end{bmatrix} = \begin{bmatrix} D_a & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A & B^t \\ B & 0 \end{bmatrix} \begin{bmatrix} D_a & 0 \\ 0 & I \end{bmatrix}. \tag{3.5.2.1}$$

The result is the preconditioned ESP linear system

$$\begin{bmatrix} \hat{A} & \hat{B}^t \\ \hat{B} & 0 \end{bmatrix} \begin{bmatrix} \hat{u} \\ p \end{bmatrix} = \begin{bmatrix} \hat{f} \\ g \end{bmatrix} \tag{3.5.2.2}$$

where

$$\begin{aligned} \hat{A} &= D_a A D_a, \\ \hat{B} &= B D_a, \\ \hat{B}^t &= D_a B^t, \\ u &= D_a \hat{u}, \\ \hat{f} &= D_a f. \end{aligned}$$

In this method we assume that an efficient method for solving the linear system $\hat{A} x = b$ exists, such as the iterative method outlined in Section 3.3.

Block LU-factorization of (3.5.2.2) gives the equation

$$\begin{bmatrix} \hat{A} & \hat{B}^t \\ \hat{B} & 0 \end{bmatrix} = \begin{bmatrix} \hat{A} & 0 \\ \hat{B} & -\hat{B}\,\hat{A}^{-1} B^t \end{bmatrix} \begin{bmatrix} I & \hat{A}^{-1}\hat{B}^t \\ 0 & I \end{bmatrix}. \tag{3.5.2.3}$$

In Method-2 the factorization (3.5.2.3) is employed in the following algorithm.

Step-1: Approximately solve (3.5.2.2) as follows:

$$\begin{bmatrix} \hat{A} & 0 \\ \hat{B} & -\hat{B}\,\hat{A}^{-1}B^t \end{bmatrix} \begin{bmatrix} \hat{w} \\ p \end{bmatrix} = \begin{bmatrix} \hat{f} \\ g \end{bmatrix} \qquad (3.5.2.4)$$

(Step-1a) $\hat{w} = \hat{A}^{-1}\hat{f}$,

(Step-1b) approximately solve sandwich system $\hat{B}\,\hat{A}^{-1}B^t p = \hat{B}\,\hat{w} - g$

by applying type-1 (type-2) sandwich preconditioner to $\hat{B}\,\hat{w} - g$.

Step-2: Solve (3.5.2.3) using $\hat{u} = \hat{w} - \hat{A}^{-1}B^t p$.

$$\begin{bmatrix} I & \hat{A}^{-1}\hat{B}^t \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{u} \\ p \end{bmatrix} = \begin{bmatrix} \hat{w} \\ p \end{bmatrix}, \qquad (3.5.2.5)$$

3.5.3 Method-3.

In Method-3 we construct a preconditioner for the preconditioned ESP matrix (3.5.2.2) with the same definitions as in Section 3.5.2.

Let p(x) denote a Choleski JMP precondtion polynomial as described in Chapter 2 with respect to the SPD matrix $\hat{A}$. Then we approximately solve

the ESP matrix

$$\begin{bmatrix} p(\hat{A})^{-2} & \hat{B}^t \\ \hat{B} & 0 \end{bmatrix} \tag{3.5.3.1}$$

as a preconditioner for (3.5.2.2), using the block-LU factorization

$$\begin{bmatrix} p(\hat{A})^{-2} & \hat{B}^t \\ \hat{B} & 0 \end{bmatrix} = \begin{bmatrix} p(\hat{A})^{-2} & 0 \\ \hat{B} & -\hat{B}\, p(\hat{A})^2\, \hat{B}^t \end{bmatrix} \begin{bmatrix} I & p(\hat{A})\, \hat{B}^t \\ 0 & I \end{bmatrix} . \tag{3.5.3.2}$$

In Method-3 we apply the factorization (3.5.3.2) in the following algorithm.

Step-1: Approximately solve (3.5.3.3) as follows:

$$\begin{bmatrix} p(\hat{A})^{-2} & 0 \\ \hat{B} & -\hat{B}\, p(\hat{A})^2\, B^t \end{bmatrix} \begin{bmatrix} \hat{w} \\ p \end{bmatrix} = \begin{bmatrix} \hat{f} \\ g \end{bmatrix} \tag{3.5.3.3}$$

(Step-1a) $\hat{w} = p(\hat{A})^2\, \hat{f}$,

(Step-1b) approximately solve sandwich system $\hat{B}\, p(\hat{A})^2\, B^t\, \hat{p} = \hat{B}\, \hat{w} - \hat{g}$ by applying type1 (type-3) sandwich preconditioner to $\hat{B}\, \hat{w} - \hat{g}$.

Step-2: Solve (3.5.2.5) using $\hat{u} = \hat{w} - p(\hat{A})\, B^t\, p$.

Chapter 4.

ESP Matrices in Flow in Porous Media.

The purpose of this chapter is to formulate and solve 2-d and 3-d elliptic boundary value problems which arise in flow in porous media. The elliptic equations and the mixed finite element approximations are described in Section 4.1, and 4.2, respectively. The conductvitiy tensor is described in Section 4.3. Test problems and numerical results are discussed in Sections 4.4 and 4.5, respectively.

4.1 The Elliptic Flow Equation.

Let Ω denote be the unit-square in $\Re^d$, d = 2, 3, with $\Gamma = d\Omega$ the boundary of Ω. Let Γ_1 denote the set of edges (or faces) of Γ with Neumann boundary conditions, and let Γ_2 denote the set of edges (or faces) of Γ with Dirichlet boundary conditions. Assume that $\Gamma = \Gamma_1 \cup \Gamma_2$, and let $\underline{n}$ denote the unit outward normal vector along Γ.

Definition 4.1.1 Let $f(\underline{x}) \in L^2(\Omega)$, $g(\underline{x}) \in H^{1/2}(\Gamma)$,$h(\underline{x}) \in H^{3/2}(\Gamma)$, and assume that the symmetric tensor $k(\underline{x})$ is uniformly-bounded in the sense that

$$\langle k(\underline{x})\, v\,,\ v\rangle_2 \geq \alpha \mid v \mid_2^2 \qquad (4.1.1)$$

holds for $v \in \Re^2$ (or $v \in \Re^3$) and $\underline{x}$ a.e. in Ω.

Then the elliptic flow equation is given by

$$\nabla \bullet \{ -k(\underline{x}) \nabla p(\underline{x}) \} = f(\underline{x}) \quad \text{on} \quad \Omega \tag{4.1.2a}$$

$$\{-k(\underline{x}) \nabla p(\underline{x}) \} \bullet n = g(\underline{x}) \quad \text{on} \quad \Gamma_1 \tag{4.1.2b}$$

$$p(\underline{x}) = h(\underline{x}) \quad \text{on} \quad \Gamma_2 . \tag{4.1.2c}$$

For the pure Neumann boundary conditions ($\Gamma_2 = \varnothing$) , we require the compatibility condition (4.1.3) in order for (4.1.2) to have a unique solution modulo the subspace of constant functions.

$$\int_{\Omega} f(\underline{x}) \, d\underline{x} + \int_{\Gamma} g(\underline{x}) \, d\underline{x} = 0 \tag{4.1.3}$$

Following Raviart-Thomas [19], define the velocity space V and pressure space W as follows.

Definition 4.1.2 Let H(Ω ; div) denote the linear space of vector functions $v \in (L^2(\Omega))^d$ such that $\nabla \bullet v \in L^2(\Omega)$, and let V denote the linear subspace of vectors v such that $v \bullet n = 0$ on Γ_1 .

Set the pressure space W be the space $L^2(\Omega)$. We introduce velocity as a new variable defined by (4.1.4), which is called Darcy's Law, and rewrite (4.1.2a) as (4.1.5).

$$u(\underline{x}) = -k(\underline{x}) \nabla p(\underline{x}) \tag{4.1.4}$$

$$\nabla \bullet u(\underline{x}) = f(\underline{x}) \tag{4.1.5}$$

If $a(\underline{x})$ denotes the tensor $k^{-1}(\underline{x})$, we obtain

$$\nabla p(\underline{x}) = -a(\underline{x}) u(\underline{x}) \tag{4.1.6}$$

Multiplying (4.1.6) by $v \in V$, and integrating, and integrating by parts, we get a weak form of Darcy's Law:

$$\langle a(\underline{x})\, u(\underline{x}) , v(\underline{x}) \rangle = \langle -\nabla p(\underline{x}) , v(\underline{x}) \rangle = \langle p(\underline{x}) , \nabla \bullet v(\underline{x}) \rangle = - \int_{\Gamma_2} g\, v \bullet n \, d\underline{x}$$

Multiplying (4.1.5) by $w \in W$ and integrating, we get a weak form of the conservation of mass equation:

$$\langle \nabla \bullet u(\underline{x}) , w(\underline{x}) \rangle = \langle f(\underline{x}) , w(\underline{x}) \rangle .$$

A weak formulation of the elliptic flow equation (4.1.2) is to solve the first order system

$$\langle a(\underline{x})\, u(\underline{x}) , v(\underline{x}) \rangle = \langle p(\underline{x}) , \nabla \bullet v(\underline{x}) \rangle = - \int_{\Gamma_2} g\, v \bullet n \, d\underline{x} \qquad (4.1.7a)$$

$$\langle \nabla \bullet u(\underline{x}) , w(\underline{x}) \rangle = \langle f(\underline{x}) , w(\underline{x}) \rangle . \qquad (4.1.7b)$$

We next describe the finite element spaces used by Raviart-Thomas [19] to discretize this mixed variational formulation.

4.2 The Mixed Finite Element Equations.

Definition 4..2.1 Let $\Delta_x = \{x_i\}_{i=0}^{nx}$, $\Delta_y = \{y_j\}_{j=0}^{ny}$, ($\Delta_z = \{z_k\}_{k=0}^{nz}$), denote partitions of the interval I_x, I_y (and I_z) for the domain Ω. Let $\{ dx_i \}_{i=1}^{nx}$, $\{ dy_j \}_{j=1}^{ny}$, and $\{ dz_k \}_{k=1}^{nz}$ denote the lengths of the subintervals of Δ_x, Δ_y, and Δ_z, respectively. Let $\{ xm_i \}_{i=1}^{nx}$, $\{ ym_j \}_{j=1}^{ny}$, and $\{ zm_k \}_{k=1}^{nz}$ denote the midpoints, and $\{ dmx_i \}_{i=1}^{nx}$, $\{ dmy_j \}_{j=1}^{ny}$, and $\{ dmz_k \}_{k=1}^{nz}$ the lengths of the intervals $[xm_i, xm_{i+1}]$, $[ym_j, ym_{j+1}]$, and $[zm_k, zm_{k+1}]$, respectively.

Definition 4.2.2 If $\Delta = \{x_i\}_{i=0}^{n}$ is a partition of the interval [0,1], define $M_q^r(\Delta)$ to be the linear space of piecewise polynomial functions $v(x) \in C^q([0,1])$ such that for $0 \le i \le nx$, $v(x)\big|_{[x_i,x_{i+1}]}$ is a polynomial of degree $\le r$ (if $q = -1$, the function v(x) may be discontinuous at the end-points of each interval).

Definition 4.2.3 For the partitions $\Delta_x, \Delta_y, (\Delta_z)$, define the discretization parameter h with the formula

$$h = \max_{\substack{1 \le i \le nx \\ 1 \le j \le ny \\ 1 \le k \le nz}} \{ dx_i, dy_j, dz_k \}. \qquad (4.2.1)$$

Definition 4.2.4 The Raviart-Thomas (RT) pressure space is the tensor-product space (4.2.5) for 2-d and (4.2.6) for 3-d.

$$W_q^r(h) = M_q^r(\Delta_x) \otimes M_q^r(\Delta_y) \qquad (4.2.5)$$

$$W_q^r(h) = M_q^r(\Delta_x) \otimes M_q^r(\Delta_y) \otimes M_q^r(\Delta_z). \qquad (4.2.6)$$

Definition 4.2.5 The Raviart-Thomas (RT) velocity space is (in 2-d)

$$V_q^r(h) = V_q^r(h)_x \times V_q^r(h)_y \qquad (4.2.7a)$$

$$V_q^r(h)_x = M_{q+1}^{r+1}(\Delta_x) \otimes M_q^r(\Delta_y) \qquad (4.2.7b)$$

$$V_q^r(h)_y = M_q^r(\Delta_x) \otimes M_{q+1}^{r+1}(\Delta_y) \qquad (4.2.7c)$$

and (in 3-d)

$$V_q^r(h) = V_q^r(h)_x \times V_q^r(h)_y \times V_q^r(h)_z \quad (4.2.8a)$$

$$V_q^r(h)_x = M_{q+1}^{r+1}(\Delta_x) \otimes M_q^r(\Delta_y) \otimes M_q^r(\Delta_z) \quad (4.2.8b)$$

$$V_q^r(h)_y = M_q^r(\Delta_x) \otimes M_{q+1}^{r+1}(\Delta_y) \otimes M_q^r(\Delta_z) \quad (4.2.8c)$$

$$V_q^r(h)_z = M_q^r(\Delta_x) \otimes M_q^r(\Delta_y) \otimes M_{q+1}^{r+1}(\Delta_z). \quad (4.28d)$$

We compute with the "lowest-order" Raviart-Thomas spaces, such that r = 0, and q = -1, and simplify the notation such that $W_h \equiv W_{-1}^0(h)$ is the pressure space, $V_h^x \equiv (V_{-1}^0(h))_x$ is the x-component velocity space, $V_h^y \equiv (V_{-1}^0(h))_y$ is the y-component velocity space, and $V_h^z \equiv (V_{-1}^0(h))_z$ is the z-component velocity space.

For i=1, nx, the functions $w_i^x(x) = 1$ for $x \in [x_{i-1},x_i]$ form a basis for the spaces $M_{-1}^0(\Delta_x)$. For j=1, ny, the functions $w_j^y(x) = 1$, form a basis for the spaces $M_{-1}^0(\Delta_y)$. For k=1, nz, the functions $w_j^z(x) = 1$ for $z \in [z_{k-1},z_k]$, form a basis for the spaces $M_{-1}^0(\Delta_z)$. For i=1, nx, the functions $v_i^x(x)$ denote the standard basis for the space $M_0^1(\Delta_x)$. $v_j^y(y_m)$ and $v_k^z(z_m) = \delta_k^m$ are defined similarly.

Therefore, for i=1, nx, j=1, ny, (k=1,nz), the functions $\{ w_i^x w_j^y (w_k^z) \}$, form a basis of the pressure space W_h, $\{ v_i^x w_j^y (w_k^z) \}$ form a basis of the x-velocity space V_h^x, $\{ w_i^x v_j^y (w_k^z) \}$ form a basis of the y-velocity space V_h^y, (and $\{ w_i^x w_j^y v_k^z \}$ form a basis of the z-velocity space V_h^z).

With the Raviart-Thomas pressure and velocity finite element spaces defined, we seek the solution pair

$$(U_h , P_h) \in V_h \times W_h \tag{4.2.12}$$

for the system

$$\langle a\, U_h , v_h \rangle - \langle P_h , \nabla\bullet v_h \rangle = = - \int_{\Gamma_2} g\, v\bullet n\, d\underline{x} , \quad v_h \in V_h \tag{4.2.13a}$$

$$\langle -\nabla\bullet U_h , w_h \rangle = \langle -f , w_h \rangle , \qquad w_h \in W_h. \tag{4.2.13b}$$

We have negated equation (4.2.13b) so that the resulting stiffness matrix becomes symmetric. Assembling the blocks of the mixed finite element stiffness matrix S, we get an ESP matrix, since A is SPD.

If we illustrate the block structure of A, B, and B^t, we get the ESP linear system

$$\begin{bmatrix} A^{xx} & A^{xy} & B^{x\,t} \\ A^{yx} & A^{yy} & B^{y\,t} \\ B^{x} & B^{y} & 0 \end{bmatrix} \begin{bmatrix} U^x \\ U^y \\ P \end{bmatrix} = \begin{bmatrix} rhsx \\ rhsy \\ rr \end{bmatrix} \tag{4.2.14}$$

for 2-d, and

$$\begin{bmatrix} A^{xx} & A^{xy} & A^{xz} & B^{x\,t} \\ A^{yx} & A^{yy} & A^{yz} & B^{y\,t} \\ A^{zx} & A^{zy} & A^{zz} & B^{z\,t} \\ B^{x} & B^{y} & B^{z} & 0 \end{bmatrix} \begin{bmatrix} U^x \\ U^y \\ U^z \\ P \end{bmatrix} = \begin{bmatrix} rhsx \\ rhsy \\ rhsz \\ rr \end{bmatrix} \tag{4.2.15}$$

for 3-d.

4.3 The Conductivity Tensor.

Let k(x,y) (k(x,y,z)) be a scalar function, which measures an absolute permeability at the center of each cell of the domain Ω. We shall call such a function a pmap. We have used the program RFG [34,35] to generate random pmaps for both 2-d and 3-d rectangular domains.

A conductivity tensor for a 2-d domain Ω is constructed using the following 3-step algorithm.

(step 1) Construct a cell-centered pmap k(x,y) for Ω.

(step 2) Interpolate the pmap from the cell-centered grid to the cell corners to construct a diagonal (k_{11}, k_{22}), using harmonic averaging in the direction of flow and arithmetic averaging in the orthogonal direction to transfer the pmap to the component velocity grids.

Introduce the notation

$p1 = k(xm_i, ym_j)$, $p2 = k(xm_{i+1}, ym_j)$,

$p3 = k(xm_i, ym_{j+1})$, $p4 = k(xm_{i+1}, ym_{j+1})$,

$havg(p_1,p_2)$ = the harmonic average of p_1 and p_2.

Then $$k_{11}(x_i,y_j) = \frac{havg(p1,p2)+havg(p3,p4)}{2.},$$

and $$k_{22}(x_i,y_j) = \frac{havg(p1,p3)+havg(p2,p4)}{2.}.$$

(step3) Rotate each cell of the grid by θ radians in the xy-plane.

The diagonal tensor (k_{11},k_{22}) is rotated with respect to the Cartesian axes by an angle θ using the similarity transformation;

$$\begin{bmatrix} k_{11}(x,y) & k_{12}(x,y) \\ k_{21}(x,y) & k_{22}(x,y) \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} k_{11}(x,y) & 0 \\ 0 & k_{22}(x,y) \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{4.3.1}$$

$$= \begin{bmatrix} \cos^2\theta * k_{11}(x,y) + \sin^2\theta * k_{22}(x,y) & \sin\theta * \cos * (k_{11}(x,y) - k_{22}(x,y)) \\ \sin\theta * \cos * (k_{11}(x,y) - k_{22}(x,y)) & \sin^2\theta * k_{11}(x,y) + \cos^2\theta * k_{22}(x,y) \end{bmatrix}, \tag{4.3.2}$$

which results in a full tensor.

A conductivity tensor for a 3-d domain Ω is constructed using the following 3-step algorithm.

(step 1) Construct a cell-centered pmap k(x,y,z) for Ω.

(step 2) Interpolate the pmap from the cell-centered grid to the cell corners to construct a diagonal tensor $(k_{11},k_{22}k_{33})$, using harmonic averaging in the direction of flow and arithmetic averaging in the orthogonal directions to transfer the pmap to the component velocity grids.

Introduce the notation

$p1 = k(xm_i,ym_j,zm_k)$, $k2 = p(xm_{i+1},ym_j,zm_k)$,

$p3 = k(xm_i,ym_{j+1},zm_k)$, $k4 = p(xm_{i+1},ym_{j+1},zm_k)$,

$p5 = k(xm_i,ym_j,zm_{k+1})$, $k6 = p(xm_{i+1},ym_j,zm_{k+1})$,

$p7 = k(xm_i,ym_{j+1},zm_{k+1})$, $k8 = p(xm_{i+1},ym_{j+1},zm_{k+1})$.

Then

$$k_{11}(x_i,y_j,z_k) = \frac{havg(p1,p2)+havg(p3,p4)+havg(p5,p6)+havg(p7,p8)}{4.}$$

$$k_{22}(x_i,y_j,z_k) = \frac{havg(p1,p4)+havg(p2,p3)+havg(p5,p8)+havg(p6,p7)}{4.}$$

$$k_{33}(x_i,y_j,z_k) = \frac{havg(p1,p5)+havg(p2,p6)+havg(p3,p4)+havg(p4,p8)}{4.}$$

(step3) Rotate each cell of the grid by θ radians in the xz-plane.

The diagonal tensor $(k_{11},k_{22}k_{33})$ is rotated with respect to the Cartesian axes by an angle θ using the similarity transformation;

$$\begin{bmatrix} k_{11}(x,y,z) & k_{12}(x,y,z) & k_{13}(x,y,z) \\ k_{21}(x,y,z) & k_{22}(x,y,z) & k_{23}(x,y,z) \\ k_{31}(x,y,z) & k_{32}(x,y,z) & k_{33}(x,y,z) \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & 0 & \cos\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & -\sin\theta \end{bmatrix} \begin{bmatrix} k_{11}(x,y,z) & 0 & 0 \\ 0 & k_{22}(x,y,z) & 0 \\ 0 & 0 & k_{33}(x,y,z) \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ \cos\theta & 0 & -\sin\theta \end{bmatrix} \qquad (4.3.3)$$

$$= \begin{bmatrix} \cos^2\theta * k_{11}(x,y,z)+\sin^2\theta * k_{33}(x,y,z) & 0 & \sin\theta*\cos*(k_{33}(x,y,z)-k_{11}(x,y,z)) \\ 0 & k_{22}(x,y,z) & 0 \\ \sin\theta*\cos\theta*(k_{33}(x,y,z)-k_{11}(x,y,z)) & 0 & \sin^2\theta * k_{11}(x,y,z)+\cos^2\theta * k_{33}(x,y,z) \end{bmatrix},$$

which is also a tensor.

4.4 The Flow Problems Solved.

In this Section we describe the porous media problems, which are used for the numerical work in Section 4.5. For 2-d we solve two series of models. Series-1 tests the robustness of the ESP solvers, such that the model is kept the same, but the grid is refined by a factor of 2 in x and y . Pure Neumann

boundary conditions are used. The conductivity tensor is constructed as described in Section 4.3 from the following pmap:

perm(x,y)	=	160	0. < y ≤ .125
perm(x,y)	=	100	.125 < y ≤ .250
perm(x,y)	=	550	.250 < y ≤ .375
perm(x,y)	=	160	.375 < y ≤ .500
perm(x,y)	=	5	.500 < y ≤ .625
perm(x,y)	=	5	.625 < y ≤ .750
perm(x,y)	=	15	.750 < y ≤ .975
perm(x,y)	=	60	.975 < y ≤ 1.000

Here is a summary of the 2-d models of Series-1 including the condition numbers of the ESP stiffness matrices.

Model	Pmap	Grid Size	Matrix Size	Condition Num.
s2d-20pn	s2d-20	20 x 20	1,160	31,111.
s2d-40pn	s2d-40	40 x 40	4,720	62,230.
s2d-80pn	s2d-80	80 x 80	19,040	126,300.
s2d-160pn	s2d-160	160 x 160	76,480	248,980.

The Series-2 2-d models use a 295 x 295 grid. The program RFG [34,35] is used to generate random 2-d pmaps, and the algorithm described in Section 4.3 is used to construct a 2-d conductivity tensor.

Two models are constructed. The first 2-d pmap (s2d_01) has a coefficient of variation $\sigma = 1.0$, and the second pmap (s2d_02) has $\sigma = 2.3$.

Each model assumes no-flow boundary conditions with an injection well with unit flow rate at (0,0) and a production well with unit flow rate at

(1,1). Since pure Neumann boundary conditions are assumed, each ESP stiffness matrix is singular.

Here is a summary of the 2-d models of Series-2 including condition numbers for the ESP stiffness matrices.

Model	Pmap	Grid Size	Matrix Size	Condition Num.
s2d-01pn	s2d-01	295 x 295	260,485	11,894.
s2d-02pn	s2d-02	295 x 295	260,485	1,369,210.

The Series-3 consists of four 3-d models, which use a 80 x 40 x 20 grid, chosen to produce an ESP matrix with approximately the same size as the Series-2 matrices for comparison purposes. The program RFG [34,35] is used to generate two random 3-d pmaps. The first 3-d pmap (s3d_01) has a coefficient of variation $\sigma = 2.3$, and the second (s3d_02) has $\sigma = 1.0$. The third 3-d pmap (s3d_03), which has moderate (constant 1. milli-darcy) permeability throughout the model, except for a narrow high (100. milli-darcy) permeability strip, which is defined as

$$\{ (x,y,z) \mid 0. < x < 1., \ .5\text{-}4\,\Delta y \ \leq \ y \ \leq .5\text{+}4\,\Delta y, \quad z = .5 \ \}.$$

The algorithm described in Section 4.3 is used to construct a 3-d conductivity tensor using the pmaps. Each model has an injection well with unit flow rate at (0,0,1), and a production well with unit flow rate at (1,1,1).

The first two models use pmaps s3d_01 and s3d_02, respectively, and pure Neumann (no-flow) boundary conditions are assumed. The third model uses the striped permeability map s3d_03, and no-flow boundary conditions are assumed. The fourth model uses pmap s3d_01, and assumes no-flow boundary conditions of the front, back, left, and right, but assumes a constant pressure of 1. on the bottom and a constant pressure of .01 on the top of the model.

Here is a summary of the 3-d models of Series-3 including condition numbers for the ESP stiffness matrices.

Model	Pmap	Grid Size	Matrix Size	Condition Num.
s3d-01pn	s3d-01	80 x 40 x 20	250,400	1,209,443.
s3d-02pn	s3d-02	80 x 40 x 20	250,400	3,937.
s3d-03pn	s3d-03	80 x 40 x 20	250,400	9,285.
s3d-04mix	s3d-01	80 x 40 x 20	256,800	1,242,709.

4.5 Summary of Computer Runs.

Each problem is solved such that the final residual is less than 10^{-6}. To estimate the computational complexity of each iterative ESP solver, we have recorded the number of matrix-vector multiplies which occur during the solution process. This is recorded as three sums:

(1) matvecs = the number of matrix-vector multiplies of S,

(2) matveca = the number of matrix-vector multiplies of A,

(3) matvecb = the number of matrix-vector multiplies of B B^t.

Each ESP matrix is scaled symmetrically as in (4.5.1) such that the diagonal matrix $D_{ii} = 1./\mathrm{sqrt}(A_{ii})$. In each problem the scaled matrix DAD has condition number less than 10.

$$\begin{bmatrix} D & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A & B^t \\ B & 0 \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & I \end{bmatrix} \tag{4.5.1}$$

This symmetric scaling has the consequence of shifting the spectrum of BB^t from a range typically $[10^{-7}, 10^{-3}]$ such that the matrix BD^2B^t has a spectrum typically with range $[10^{-7}, 10^{-3}]$, which using the formulas in Corollary 3.1 for the spectrum of the scaled ESP matrix, is better balanced for solution.

Single cpu times on a Cray C-90 are provided as a check that things are in order, but the principal criterion of comparision of the methods used is the computational complexity measured by total number of matvecs required for solution.

Parallelization of the solvers was not a design goal, but rather to establish the effectiveness of the preconditioners for ESP matrices described in Chapter 3.

Arnoldi iteration applied to a JMP polynomial in the matrices A and BD^2B^t, as described in Section 3.2, are used to estimate spectral covers for the matrices A and BD^2B^t, respectively. However, all of these estimates were quite similar, and the JMP preconditioners were not noticeably sensitive to the estimate for the inner-most eigenvalue, but the computation diverged if the upper eigenvalue estimate was not large enough.

For all the JMP preconditioners for SPD matrices, it was found experimentally that using $\alpha = -.95$, $\beta = -.5$ provided effective preconditioners.

For all the JMP Choleski preconditioners, it was found experimentally that using $\alpha = -.99$, $\beta = -.5$ provided effective preconditioners.

For the 2-band JMP preconditioners used by Method-I , it was found experimentally that using $\alpha = -.95$, $\beta = -.5$ on both eigenbands, provided effective preconditioners. Since the matrices BD^2B^t had condition numbers in the range 10^4-10^5, it was found experimentally that the degree of the JMP preconditioners for the matrix BD^2B^t could be increased to about 20, resulting in reducing the computational complexity of the solution of the ESP system as well as reducing the runtime.

Here are the runtime characteristics on a Cray C-90 using the preconditioners of Chapter 3 for the **Series-1 model problems**.

Method-1

model	gmres its	matveca	matvecb	matvecs	cpu (secs)
s2d_20pn	6	36	756	36	.1627
s2d_40pn	11	66	726	66	.4542
s2d_80pn	9	54	1134	54	3.062
s2d_160pn	8	48	1008	48	17.420

Method-2

model	gmres its	matveca	matvecb	matvecs	cpu (secs)
s2d_20pn	3	226	223	3	.0448
s2d_40pn	3	274	303	3	.1389
s2d_80pn	2	291	362	2	.5601
s2d_160pn	1	92	101	1	.9178

Method-3

model	gmres its	matveca	matvecb	matvecs	cpu (secs)
s2d_20pn	2	123	182	2	.0294
s2d_40pn	3	214	333	3	.1324
s2d_80pn	1	72	221	1	.2659
s2d_160pn	1	72	111	1	.9327

Here are the runtime characteristics on a Cray C-90 using the preconditioners of Chapter 3 for the **Series-2 model problems**.

Method-1

model	gmres its	matveca	matvecb	matvecs	cpu (secs)
s2d_01pn	11	66	726	66	16.0998
s2d_02pn	7	42	252	42	5.6085

Method-2

model	gmres its	matveca	matvecb	matvecs	cpu (secs)
s2d_01pn	1	56	61	1	1.0981
s2d_02pn	2	249	242	2	4.3480

Method-3

model	gmres its	matveca	matvecb	matvecs	cpu (secs)
s2d_01pn	1	72	111	1	1.0981
s2d_02pn	1	42	51	1	.8866

Here are the runtime characteristics on a Cray C-90 using the preconditioners of Chapter 3 for the **Series-3 model problems**.

Method-1

model	gmres its	matveca	matvecb	matvecs	cpu (secs)
s3d_01pn	14	168	924	84	21.132
s3d_02pn	8	96	528	48	12.042
s3d_03pn	10	120	660	60	15.012
s3d_04mix	20	121	4880	121	54.676

Method-2

model	gmres its	matveca	matvecb	matvecs	cpu (secs)
s3d_01pn	1	128	111	1	3.277
s3d_02pn	1	128	111	1	3.289
s3d_03pn	1	110	101	1	2.895
s3d_04mix	2	345	642	2	12.570

Method-3

model	gmres its	matveca	matvecb	matvecs	cpu (secs)
s3d_01pn	1	72	111	1	2.445
s3d_02pn	1	72	111	1	2.455
s3d_03pn	1	52	71	1	1.685
s3d_04mix	3	214	663	3	10.706

Chapter 5. Concluding Remarks.

5.1 Which Methods Performed Best?

The preconditioners Method-2 and Method-3, based on block-LU factorization of an approximation of the ESP matrix S, proved to be more efficient than Method-1, because Method-1 requires evaluating the composition of the polynomial which preconditions the whole matrix S with the polynomial which preconditions the BB^t.

Symmetric scaling the matrix A of the ESP stiffness matrix by its diagonal (as described in Remark 3.5.1) has the effect that the resulting sandwich matrices, which arise in the three ESP preconditioners are well conditioned by a split preconditioner.

In the numerical work in Chapter 4 we applied each of the three sandwich preconditioners to porous media problems. The type-1 sandwich preconditioner was found to be the most efficient in all cases. The reason for this is that the type-2 requires approximate solution of two linear systems, and type-3 requires approximate solution of three linear systems. The type-1 method requires only the approximate solution of a single linear system.

5.2 Choosing Effective Parameters.

Experimentation with the degree of the JMP preconditioners for the SPD matrix A showed that since A was so well preconditioned by symmetric scaling with the diagonal matrix diag(A), that each of the Darcy flow problems was efficiently solved with a JMP polynomial with degree 3 to 5.

The JMP Choleski polynomials used to precondition the blocks $B\,B^t$ required larger degree since the condition number of the matrix BB^t (which is a discretization of the Laplacian) is known to be proprotional to h^{-2}. Experimentation indicated that using a JMP polynomial of degree of about 10 minimized the total work of the solvers.

The 2-band preconditioner used in Method-1 was most efficient with degree in the range 5-10 in all the Darcy flow problems.

Experimentation showed that effective parameters to use for defining the Jacobi polynomials were $\alpha = -.95$ and $\beta = -.5$ for the single-function JMP preconditioners of definite matrices.

The effective parameters used for the JMP Choleski preconditioners were found experimentally to be $\alpha = -.995$ and $\beta = -.5$ (this value for α produced numerical instabilty in construction of the single-function JMP preconditioner).

The effective parameters used for the 2-band preconditioners for the ESP stiffness matrices from the Darcy flow problems were found to be $\alpha = -.5$ and $\beta = -.95$ on the negative eigenband and $\alpha = -.95$ and $\beta = -.5$ on the positive eigenband.

5.3 Summary.

In this Thesis we have developed a general theory of polynomial matrix preconditioners. Of particular interest is the use of functions other than the function f(x) = 1 as ideal functions to be approximated. For example, we have demonstated that approximations to the square-root function $f(x) = \sqrt{x}$ can lead to effective split preconditioners of positive definite linear systems. Moreover, we have also demonstrated that polynomial matrices using approximations to the piecewise constant function, which is -1 on the negative part of the spectrum and +1 on the positive part of the spectrum of an symmetric indefinite matrix can also provide effective preconditioners.

We have described efficient methods based upon Arnoldi's iteration for estimating spectral covers for saddle-point matrices, which are necessary for defining the domain of definition of the polynomials.

We have demonstrated computationally that these polynomial matrices can be used quite effectively in the solution of saddle-point problems which arise in mixed finite element solution of flow in porous media problems.

In order to solve ESP linear systems we have developed several effective methods for solving Schur complements of ESP matrices (sandwich matrices), which arise in block factorization of ESP matrices.

Finally, through experimentation we have found parameters, which produce effective preconditioners, but we were unable to provide a theorem, which predicts optimality.

Appendix A: Proof of Spectral Cover Theorem.

To prove the Spectral Cover Theorem for a general saddle-point matrix

$$\begin{bmatrix} A & B^t \\ B & -C \end{bmatrix}$$

we shall use the following well known formulas

$$\Re^n = \mathrm{Im}(B^t) \oplus \mathrm{null}(B) \tag{A1}$$

$$\lambda_n \| x \|^2 \le \langle A x, x \rangle \le \lambda_1 \| x \|^2 \quad \text{for } x \in \Re^n \tag{A2}$$

$$\gamma_m \| y \|^2 \le \langle C y, y \rangle \le \gamma_1 \| y \|^2 \quad \text{for } y \in \Re^m \tag{A3}$$

$$\sigma_m \| y \| \le \| B^t y \| \le \sigma_1 \| y \| \quad \text{for } y \in \Re^m \tag{A4}$$

$$\sigma_m \| x \| \le \| B x \| \le \sigma_1 \| x \| \quad \text{for } x \in \mathrm{null}(B)^{\perp} . \tag{A5}$$

Theorem 3.1 (Spectral Cover for saddle-point matrices) If S is a non-singular saddle-point matrix, then { [a,b], [c,d] } is a spectral cover for S provided the formulas

$$a = 1/2 \left[(\lambda_n - \gamma_1) - \sqrt[2]{(\lambda_n + \gamma_1)^2 + 4 \sigma_1^2} \right] \tag{A6}$$

$$b = 1/2 \left[(\lambda_1 - \gamma_m) - \sqrt[2]{(\lambda_1 + \gamma_m)^2 + 4 \sigma_m^2} \right] \tag{A7}$$

$$c = \lambda_n \tag{A8}$$

$$d = 1/2 \left[(\lambda_1 - \gamma_m) + \sqrt[2]{(\lambda_1 + \gamma_m)^2 + 4 \sigma_1^2} \right] \tag{A9}$$

hold.

Proof: Let $\lambda \in \Lambda(S)$ and (x,y) be a corresponding eigenvector. Then

$$A x + B^t y = \lambda x \qquad \text{(A10)}$$

$$B x - C y = \lambda y. \qquad \text{(A11)}$$

Assume $x = 0$. Since B^t is is injective, (A10) implies that $y = 0$, which is impossible; therefore, $x \neq 0$. To establish the lower bound c for the positive eigenband of S, assume $\lambda > 0$ and take the inner-product of (A10) with x and (A11) with y. Then we get formulas (A12) and (A13).

$$\langle A x, x \rangle + \langle B^t y, x \rangle = \lambda \| x \|^2 \qquad \text{(A12)}$$

$$\langle B x, y \rangle = \langle B^t y, x \rangle = \langle (\lambda+C)\ y, y \rangle \qquad \text{(A13)}$$

Therefore, $\langle A x, x \rangle + \langle (\lambda+C)\ y, y \rangle = \lambda \| x \|^2$, which implies using (A1) and (A2) that $(\lambda_n - \lambda) \| x \|^2 \leq -(\lambda+\gamma_m) \| y \|^2 \leq 0$. Hence, $\lambda \geq \alpha n$.

To establish the positive upper bound d in (A9), we use (A11) and (A12) to get

$$\langle A x, x \rangle + \langle (\lambda+C)^{-1} B x, B x \rangle = \lambda \| x \|^2 \qquad \text{(A14)}$$

and using (A2) and (A5) we get

$$\lambda \| x \|^2 \leq \left(\lambda_1 + \frac{\sigma_1^2}{\lambda+\gamma_m} \right) \| x \|^2$$

which implies that $\lambda (\lambda + \gamma_m) \leq \lambda_1 (\lambda+\gamma_m) + \sigma_1^2$. By the quadratic formula $\lambda \leq 1/2 \left[(\lambda_1 - \gamma_m) + \sqrt[2]{(\lambda_1 - \gamma_m)^2 + 4 (\sigma_1^2 + \lambda_1 \gamma_m)} \right]$, and after simplification $\lambda \leq 1/2 \left[(\lambda_1 - \gamma_m) + \sqrt[2]{(\lambda_1 + \gamma_m)^2 + 4 \sigma_1^2} \right]$, which establishes the bound d.

To establish the negative bounds, first observe that if $\lambda < 0$, then $\lambda \| x \|^2 < 0$, and $\langle A x, x \rangle > 0$, so (A14) implies (A15).

$$\langle (\lambda+C)^{-1} B x, B x \rangle < 0. \tag{A15}$$

To establish the bound a in (A6). Since (A15) holds, formulas (A2), (A4), and (A14) imply that

$$\lambda_n \| x \|^2 \leq \langle A x, x \rangle = \lambda \| x \|^2 - \langle (\lambda+C)^{-1} B x, B x \rangle$$

$$\leq (\lambda - \frac{\sigma_1^2}{\lambda+\gamma_1}) \| x \|^2,$$

and by the quadratic formula and simplification

$$\lambda \leq \quad 1/2 \, [\, (\lambda_n - \gamma_1) - \sqrt{(\lambda_n + \gamma_1)^2 + 4 \, \sigma_1^2} \,],$$

which establishes the bound a.

To establish the negative upper bound b in (A7), assume that $\lambda < 0$, that $x = v + w$, where $w \in$ null(B) and $v \in$ null(B)$^{\perp}$, and take the inner-product of (A10) with v and use (A11) to get

$$\langle A(v+w), v \rangle + \langle (\lambda+C)^{-1} B v, B v \rangle = \lambda \| v \|^2,$$

and thus

$$\langle A w, v \rangle = \lambda \| v \|^2 - \langle A v, v \rangle - \langle (\lambda+C)^{-1} B v, B v \rangle,$$

and using (A2) and (A4) we get (A16).

$$\langle A w, v \rangle \geq [\lambda - \lambda 1 - \frac{\sigma_m^2}{\lambda+\gamma_m}] \| v \|^2. \tag{A16}$$

We find an upper bound for $\langle A w, v \rangle$ by taking the inner-product of (A10) with w and use (A2). Since $w \in$ null(B) and $\lambda - \lambda_n < 0$, we get

$$\langle A w, v \rangle \leq \quad (\lambda - \lambda_n) \| w \|^2 \leq 0. \tag{A17}$$

Then (A16) and (A17) and the symmetry of A imply (A18).

$$0 \leq [\, \lambda^2 + (\gamma_m - \lambda_1)\lambda - (\gamma_m \lambda_1 + s_m^2) \,] \;\| v \|^2 \qquad \text{(A18)}$$

To show that $v \neq 0$, suppose, for the sake of contradiction, that $v = 0$. Then since $B x = B v = 0 = (I\text{-}C)\, y$, we deduce from (A15) that $y = 0$, and so (A10) implies that $A w = \lambda w$, which is a contradiction, because A is positive-definite and λ is negative. By the quadratic formula, and simplification, we get $\lambda \leq 1/2 \, [\, (\lambda_1 - \gamma_m) - \sqrt[2]{(\lambda_1 + \gamma_m)^2 + 4\, \sigma_m^2} \,]$. □

REFERENCES.

[1] W. E. Arnoldi, The Principle of Minimized Iterations in the Solution of the Matrix Eigenvalue Problem, **Quart. Appl. Math.**, Vol 9, pp. 17-29 (1951).

[2] S. F. Ashby, Polynomial Preconditioning for Conjugate Gradient Methods, PhD. Thesis, Dept. of Computer Science, University of Illinois at Urbana-Champaign (1988).

[3] S. F. Ashby, T. A. Manteuffel, and J. S. Otto, A Comparison of Adaptive Chebyshev and Least Squares Polynomial Preconditioning for Hermetian Positive Definite Linear Systems, Numerical Mathematics Group, LLNL, UCRL--JC-106726 (1991).

[4] K. Aziz, and A. Settari, **Petroleum Reservoir Simulation**, Applied Science Publishers, London, 1979.

[5] R. E. Bank, B. D. Welfert, and H. Yserentant, A Class of Iterative methods for Solving Saddle Point Problems, **Numerische Mathematik**, Math, 56, 645-666 (1990).

[6] J. Bramble, and J. E. Pasciak, A preconditioning Technique for Indefinite Systems Resulting from Mixed Approximations of Elliptic Problems., **Mathematics of Computation**, Vol. 50, No. 181, pp. 1-17 (1988).

[7] C. C. Cheney, **Introduction to Approximation Theory**, McGraw-Hill, New York (1966).

[8] C. V. Deutsch and A. G. Journel, **GSLIB: Goestatistical Software Library and User's Guide**, Oxford University Press (1992).

[9] L. J. Durlofsky and M. C. H. Chien, Development of a mixed finite-element-based compositional reservoir simulator, Proc. 12th SPE Symposium on Reservoir Simulation, New Orleans, 1993.

[10] I. S. Duff, and J. A. Scott, Computing Selected Eigenvalues of Sparse Unsymmetric Matrices Using Subspace Iteration, **ACM Trans. Math. Software**, Vol. 19, No, 2, , pp. 137-159, (1993).

[11] N. Dyn and W. E. Ferguson Jr., The Numerical Solution of Equality-Constrained Quadratic Programming Problems, **Mathematics of Computation**, Vol. 41, No. 163, pp. 165-170 (1983).

[12] H. C. Elman, and G. H. Golub, Inexact and Preconditioned Uzawa Algorithms for Saddle Point Problems, CS-TR-3075, Computer Science Dept., University of Maryland (1993).

[13] M. Fortin, and R. Glowinski, **Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems,** North-Holland, New York, Chapter 1 (1983).

[14] W. Gautschi, On Generating Orthogonal Polynomials, **SIAM J. Sci. Statist. Comp.,** Vol. 3, pp. 289-317 (1982).

[15] G. H. Golub and C. F. Van Loan, **Matrix Computations,** John Hopkins University Press, Baltimore, MD, (1983).

[16] G. H. Golub and J. H. Welsch, Calculation of Gauss Quadrature Rules, **Math. Comp.,** 23, pp. 221-230 (1969).

[17] O. G. Johnson, C. A. Micchelli and G. Paul, Polynomial Preconditioning for Conjugate Gradient Calculations, **SIAM J. Numer. Anal.,** Vol 20, No. 2, pp. 362-376 (1983).

[18] D. W. Peaceman, **Fundamentals of Numerical Reservoir Simulation,** Elsevier, Amsterdam, 1977.

[19] P. A. Raviart, and J. M. Thomas, A Mixed Finite Element Method for 2nd Order Elliptic Problems, in **Mathematical Aspects of the Finite Element Method,** Rome, Lecture Notes in Mathematics, Springer-verlag, Berlin (1977).

[20] T. F. Russell, and M. F. Wheeler, Finite Element and Finite Difference Methods for Continuous Flows in Porous Media, in **The Mathematics of Reservoir Simulation,** R. E. Ewing, Editor, SIAM (1983).

[21] T. Rusten, Iterative Methods for Mixed Finite Element Systems, Thesis, Dept. of Informatics, University of Oslo (1991).

[22] T. Rusten and R. Winther, A Preconditioned Iterative Methods for Saddle Point Problems, Iterative Methods for Mixed Finite Element Systems, Institutt for Informatikk, University of Oslo (1991).

[23] Y. Saad and M. H. Schultz, **GMRES**: A Generalized Minimum Residual Algorithm for Solving Nonsymmetric Linear Systems, **SIAM J. Sci. Stat. Comput.,** 7 (1986), pp. 856-869.

[24] Y. Saad, Iterative Solution of Indefinite Symmetric Linear Systems by Methods Using Orthogonal Polynomials Over Two Disjoint Intervals, **SIAM J. Numer. Anal.,** Vol. 20, No. 4, (1983), pp. 784-810.

[25] Y. Saad, Practical Use of Polynomial Preconditionings for the Conjugate Gradient Method, **SIAM J. Sci. Stat. Comput.,** Vol. 6, No. 4 (1985).

[26] Y. Saad, Chebyshev Acceleration Techniques for Solving Nonsymmetric Eigenvalue Problems, **Math. of Comput.**, Vol. 42, 166 (1984), pp. 567-588.

[27] P. E. Saylor, Preconditioning Symmetric Indefinite Matrices. In: **Preconditioning Methods: Analysis and Applications**, David J. Evans, ed. Gordon and Breach Science Publishers, New York, , pp. 295-319 (1983).

[28] D. C. Sorensen, Implicit Application of Polynomial Filters in a k-step Arnoldi Method, TR90-27, Computational and Applied Mathematics Dept., Rice University (1990).

[29] D. C. Sorensen, Implitictly Restarted Arnoldi Methods, student notes, Rice University (1993).

[30] E. L. Stiefel, Relaxationsmethoden bester Strategie zur losung linearer Gleichungssystems, **Comment. Math. Helv.** , pp. 157-179 (1955).

[31] E. L. Stiefel, Kernel Polynomials in Linear Algebra and their Applications, **U.S. NBS Applied Math. Series** 49 (1958), pp. 1-24.

[32] T. J. Stieltjes, Quelques Recherches sur la Théorie des Quadratures Dites Méchaniques, **Ann. Sci. Ecole Norm. Paris Sér.** 3, 1 (1884), pp. 409-426. [Oeuvres I, pp. 377-396.]

[33] G. Szegö, **Orthogonal Polynomials**, Colloquium Publications 23, Third Edition, second printing, American Mathematical Society, Providence, R.I. (1974).

[34] A. F. B. Tompson, R. Ababou, L. W. Gelhar, Implementation of the Three-Dimensional Turning Bands Random Field Generator, **Water Resources Research**, 1989, Vol. 25, No. 10, pp. 2227-2243.

[35] A. F. B. Tompson, User's Guide for Turning Bands Codes, Earth Sciences Department, LLNL, 1993.

[36] H. F. Walker, Implementation of the GMRES method using Householder Transformations, SIAM J. Sci. Stat. Comp., 9, 152-163.

[37] A. Weiser, and M. F. Wheeler, On convergence of block-centered finite-differences for elliptic problems, **SIAM J. Numerical Analysis** 25, pp. 351-375, 1988.