The Numerical Solution of the Large Scale Eigenvalue Problem

R. B. Lehoucq

Mathematics and Computer Science Division,

Argonne National Laboratory

lehoucq@mcs.anl.gov

http://www.mcs.anl.gov/home/lehoucq/index.html

Problem

Let $\mathbf{A}$ be a real matrix of order n. Find $k \ll n$ solutions to the matrix eigenvalue problem

$$\mathbf{Ax} = \lambda \mathbf{x},$$

Interested in methods that only require matrix vector products with $\mathbf{A}$.

No **dense similarity transformations** as in EISPACK and LAPACK for they **destroy the sparsity** of $\mathbf{A}$ and **require $\mathcal{O}(n^2)$ storage.** These are implementations of the QR algorithm.

Definition: Large is something for which the full QR algorithm is prohibitive, both in storage and efficiency.

QR Algorithm

Let $\mathbf{A}^+ \leftarrow \mathbf{A}$. and $\mathbf{W} = \mathbf{I}$.

- Let $\mathbf{QR} \leftarrow \mathbf{A}^+ - \sigma_i \mathbf{I}$ be a QR factorization.
- $\mathbf{A}^+ \leftarrow \mathbf{RQ} - \sigma_i \mathbf{I}$ and $\mathbf{W} \leftarrow \mathbf{WQ}$.

Let $\mathcal{P}(\lambda) \equiv (\lambda - \mu_1) \cdots (\lambda - \mu_N)$. If we take the QR factorization $\mathcal{P}(\mathbf{A}) = \mathbf{QR}$, then

$$\mathbf{W} = \mathbf{Q} \qquad \text{and} \qquad \mathbf{Q}^T \mathbf{A} \mathbf{Q} = \mathbf{A}^+.$$

The QR algorithm is simultaneous iteration in disguise. The polynomial $\mathcal{P}(\mathbf{A})$ is **neither formed nor applied.** However, we do need to store an entire matrix.

Schur Decomposition

$$\mathbf{AZ} = \mathbf{Z} \begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1,\bar{n}} \\ 0 & T_{22} & \cdots & T_{2,\bar{n}} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_{\bar{n},\bar{n}} \end{bmatrix}$$

- $\mathbf{Z}^H\mathbf{Z} = \mathbf{I}$, The columns of $\mathbf{Z}$ are the Schur vectors.
- Each $T_{i,i}$ is an eigenvalue.
- Can re-order $\mathbf{T}$ so that the k eigenvalues of interest are located in $\mathbf{T}_k$ (the leading principle matrix of order k).

Since we're only interested in k eigenvalues, why not just only compute $\mathbf{T}_k$, the leading principle matrix of order k of $\mathbf{T}_n$? And thus remove the $\mathcal{O}(n^2)$ storage requirements.

Outline

- An example from computational chemistry. (Joint work with S. Gray, Theoretical Chemistry, Argonne).
- Truncated QR algorithms. (No need to store the entire matrix). An unification and extension of work of Sorensen.
- An example from the analysis of guided modes in optical fibers. (Joint work with L. Cowsar, Lucent Technologies).
- Inexact rational Krylov method. (Joint work with K. Meerbergen, Leuven).

Engineering/science examples drive the algorithmic/software design & development. Goal: a computational toolkit for large scale eigenvalue problems

Vibrational Energy Levels of HOCO

Interest in the four-atom system Hydrogen-Oxygen-Carbon-Oxygen. The energy levels are the eigenvalues $\mathcal{E}$ of the time independent six degree of freedom Schrödinger equation:

$$\mathcal{H}\boldsymbol{\Psi} = (\mathcal{T} + \mathcal{V})\boldsymbol{\Psi} = \mathcal{E}\boldsymbol{\Psi}$$

$\mathcal{T}$ is Kinetic energy, $\mathcal{V}$ is the Potential energy. (Think of $\mathcal{H} = \Delta + \mathcal{V}$.)

We use a diatom-diatom Jacobi coordinate system.

Three radial $\mathbf{R} = (\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3)$ and angular degrees of freedom **Theta** $= (\Theta_1, \Theta_2, \Theta_3)$ and standard boundary conditions.

We want at least 50 eigenpairs.

Motivation

- Understanding of a basic molecule and its reactions.
- Applications to combustion and atmospheric chemistry. For example, it is one of the most important chemical reactions in the process of burning hydrocarbon fuels. HOCO is an intermediate state of this process.
- Why so many vibrational levels ? Understand the more complex motions of HOCO and possibly refine the underlying potential.
- Mix state-art-techniques from computationally chemistry & numerical analysis along with massively parallel computing. **Is it worth the effort ?**
- Writing up our results. For more information, see `http://www.mcs.anl.gov/home/lehoucq/hoco.html`.

Discretization Scheme

The matrix eigenvalue problem has the form

$$\mathcal{H}_N \Psi_i = (\mathcal{T}_N + \mathcal{V}_N)\Psi_i = \mathcal{E}_i \Psi_i.$$

We use state-of-the-art methods from the chemical physics literature for compactly and efficiently representing the discrete Hamiltonian $\mathcal{H}_N$ and computing the associated matrix vector products.

We implement a discrete variable representation (DVR) (Light, Hill & Lill 1985) on a direct product of six one dimensional problems.

This is formally equivalent to a Pseudo–spectral method. We use a certain choice of basis functions. (Echave & Clary 1992).

Discretization Scheme, cont.

First difficulty is that Oxygen and Carbon are each 16 and 12 times as massive as Hydrogen, respectively.

We'll need alot of grid points to represent the angular coordinates.

The problem size is $N = N_R \cdot N_A$, where

- N_R denotes the product $N_{OH} \cdot N_{CO} \cdot N_{OH-CO}$. Each integer is the number of basis functions used for each radial degree of freedom.
- N_A is the number of angular basis functions.

N_R **is** $\mathcal{O}(100)$ **and** N_A **is** $\mathcal{O}(10,000)$ **so that** N **is** $\mathcal{O}(1,000,000)$.

Cost to apply the Discrete Hamiltonian

The idea behind DVR is to make a change of coordinates so that $\mathcal{V}_N$ is diagonalized.

Because we suppose that the six degree PDE is separable,

$$\mathbf{\Psi}(\mathbf{R}, \mathbf{\Theta}) \equiv \mathbf{\Psi}_1(\mathbf{R}_1)\mathbf{\Psi}_2(\mathbf{R}_2)\mathbf{\Psi}_3(\mathbf{R}_3)\mathbf{\Psi}_4(\mathbf{\Theta}_1, \mathbf{\Theta}_2, \mathbf{\Theta}_3)$$

the cost in applying $\mathcal{H}_N$ is not N^2.

The computational concerns are that

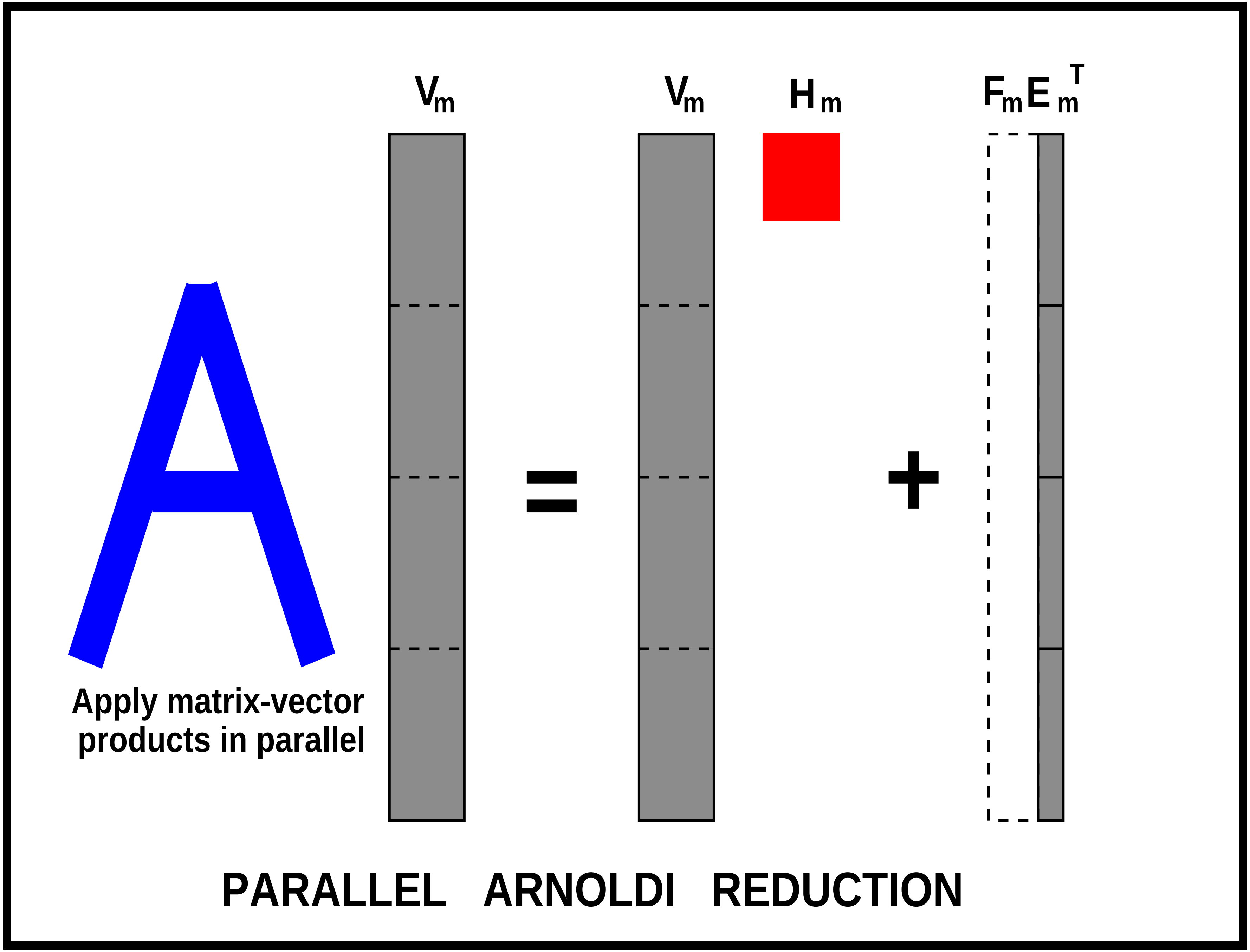
- **Cost of a matrix vector product with $\mathcal{H}_N$ is nearly $N_R^2 \cdot N_A$ or $\mathcal{O}(10^7)$–$\mathcal{O}(10^8)$ FLOPS.**
- **We need storage for a matrix with 50 columns and N rows.**

Eigensolver

Distributed memory version of ARPACK (Lehoucq,Maschhoff,Sorensen,Vu & Yang). **http://www.caam.rice.edu/software/ARPACK/** User's guide available.

ARPACK implements an implicitly restarted Arnoldi Method. (Sorensen 1992). (When the matrix is Hermitian, Arnoldi's is equivalent to Lanczos' method)

Distributed memory version uses a data parallel model of computation. Each processor holds a small multiple of N_A rows of the Lanczos vectors.


Vm
Vm
Hm
FmEm T
Apply matrix-vector
products in parallel
PARALLEL ARNOLDI REDUCTION

Results on 54 nodes of the IBM SP

	Experiment 1	Experiment 2	Experiment 3
N	1,092,420	1,638,630	2,287,044
Time(Hours)	5.5	7.9	11
$\mathcal{E}_i$	41	52	52
$\mathcal{H}_N\mathbf{x}$	733	903	894
Time per (seconds)	26.3	30.4	43.3
% Time for all $\mathcal{H}_N\mathbf{x}$	98 %	97 %	97 %

The first 50 eigenvalues appear to be accurate to 3–5 digits.

Review of Methods

Power Method with a vector $\mathbf{u}$

$$\mathbf{u}, \mathbf{Au}, \cdots, \mathbf{A}^{m-1}\mathbf{u}$$

Sequence converges to $\mathbf{A}$'s eigenvector associated with eigenvalue of largest magnitude.

Simultaneous iteration with a martrix $\mathbf{U}$ of b vectors

$$\mathbf{U}, \mathbf{AU}, \cdots, \mathbf{A}^{m-1}\mathbf{U}$$

Sequence converges to $\mathbf{A}$'s eigenspace associated with the b eigenvalues of largest magnitude.

A spectral transformation $\mathcal{P}(\cdot)$ can be used so that $\mathcal{P}(\mathbf{A})$ has eigenvalues $\mathcal{P}(\lambda)$. For example, $\mathcal{P}(\lambda) = (\lambda - \sigma)^{-1}$.

Brief History

- 1940's Wielandt introduces the power method.
- 1950,1951 Lanczos & Arnoldi methods. Round-off error causes loss orthogonality which renders methods unstable.
- 1957 Bauer & (1969) Rutihauser consider subspace iteration for Hermitian matrices. 1976 Stewart considers subspace iteration for general matrices.
- 1971 Paige's analysis leads to the understanding of the numerical problems with Lanczos's method.
- 1970's Cullum, Golub, Lewis, Parlett, Scott, Underwood exploit Paige's analysis and the three term recurrence.
- 1980 Saad **restarts** Arnoldi's method based on earlier ideas.
- 1992 Sorensen implicitly restarts Lanczos & Arnoldi methods.

Block Arnoldi Reduction of Length m

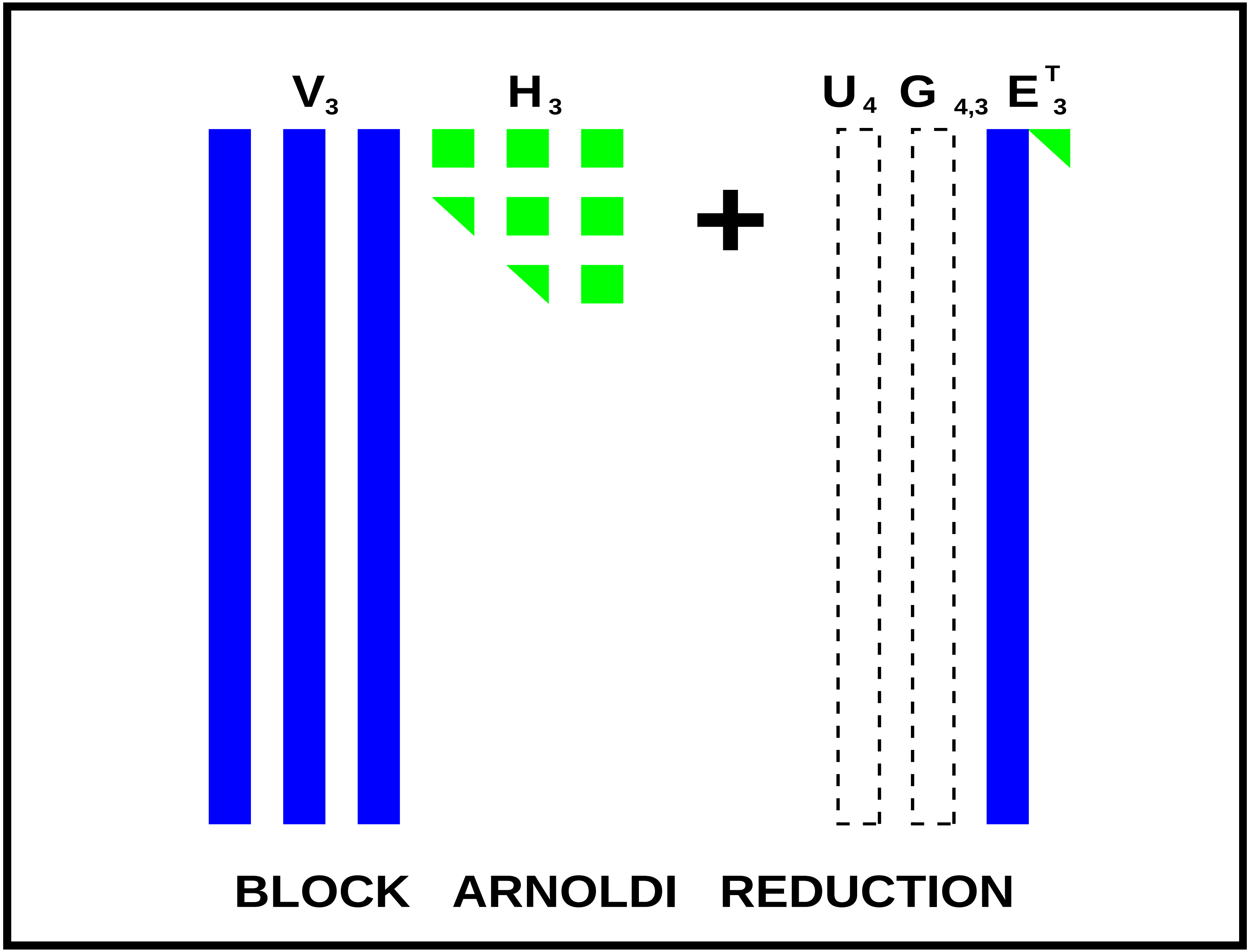
$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_m\mathbf{H}_m + \mathbf{F}_m\mathbf{E}_m^T$$

- $\mathbf{V}_m^H\mathbf{A}\mathbf{V}_m = \mathbf{H}_m$ is a band Hessenberg matrix of order $b \cdot m$.
- $\mathbf{V}_m^H\mathbf{V}_m = \mathbf{I}_{bm}$, and $\mathbf{V}_m^H\mathbf{F}_m = \mathbf{0}$.
- $\mathcal{K}_m(\mathbf{A}, \mathbf{U}_1) \equiv \{\mathbf{U}_1, \mathbf{A}\mathbf{U}_1, \cdots, \mathbf{A}^{m-1}\mathbf{U}_1\} = \mathcal{R}(\mathbf{V}_m)$.

Block Arnoldi Reduction of Length m, cont.

$$\mathbf{A}\,\mathbf{V}_m = \begin{bmatrix} \mathbf{U}_1 & \cdots & \mathbf{U}_m \end{bmatrix} \begin{bmatrix} \mathbf{G}_{1,1} & \cdots & \cdots & \mathbf{G}_{1,m} \\ \mathbf{G}_{2,1} & \ddots & \vdots & \vdots \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{G}_{m,m-1} & \mathbf{G}_{m,m} \end{bmatrix} + \mathbf{U}_{m+1}\mathbf{G}_{m+1,m}\mathbf{E}_m^T$$

- $\mathbf{U}_{m+1}\mathbf{G}_{m+1,m} = \mathbf{A}\mathbf{U}_m - \sum_{i=1}^{m} \mathbf{U}_i\mathbf{G}_{i,m}$ where $\mathbf{G}_{i,m} = \mathbf{U}_i^H\mathbf{A}\mathbf{U}_m$.
- The columns of $\mathbf{U}_{m+1}$ are an unitary basis for the orthonormal projection of $\mathbf{A}^m\mathbf{U}_1$ onto $\mathbf{V}_m$.



V3
H3
+
U4 G4,3 E3T
BLOCK ARNOLDI REDUCTION

Computing Eigenvalues

Let $1 \le i \le bm$ & $\mathbf{H}_m \mathbf{s}_i = \theta_i \mathbf{s}_i$ so that

$$\|\mathbf{A}\mathbf{V}_m \mathbf{s}_i - \mathbf{V}_m \mathbf{s}_i \theta_i\| = \|\mathbf{F}_m \mathbf{E}_m^T \mathbf{s}_i\|$$

should be small for the k desired eigenvalues.

The extremal, well–separated eigenvalues of $\mathbf{A}$ are well approximated by the θ_i for m large. (Saad).

A truncated QR algorithm (restarting) pushes the interesting directions to the leading portion of the reduction.

Increasing b (blocksize) does a better job of getting clusters and/or multiplicities. **Increase the convergence tolerance** Can also be more efficient since $\mathbf{A}$ is applied to a group of vectors. **Amortize the cost of applying A**

Truncated QR Algorithm

Let $\psi_p(\lambda) = \prod_{i=1}^{p}(\lambda - \tau_i)$ where $1 \le p < m$ and let $r = m - p$.

Let $\psi_p(\mathbf{H}) \begin{bmatrix} \mathbf{E}_1 & \cdots & \mathbf{E}_r \end{bmatrix} = \mathbf{QR}$.

Suppose we run p steps of the implicitly shifted QR algorithm on $\mathbf{H}$ with $\tau_1, \ldots, \tau_p$ resulting in $\mathbf{H}_m\mathbf{W} = \mathbf{W}\mathbf{H}_m^+$. A classical result gives that $\mathbf{W} \begin{bmatrix} \mathbf{E}_1 & \cdots & \mathbf{E}_r \end{bmatrix} = \mathbf{Q}$.

Theorem Let $\mathbf{V}_r^+ = \mathbf{V}_m\mathbf{Q}$. Given a length m block Arnoldi reduction

$$\psi_p(\mathbf{A})\mathbf{V}_r = \mathbf{V}_r^+\mathbf{R}$$

and

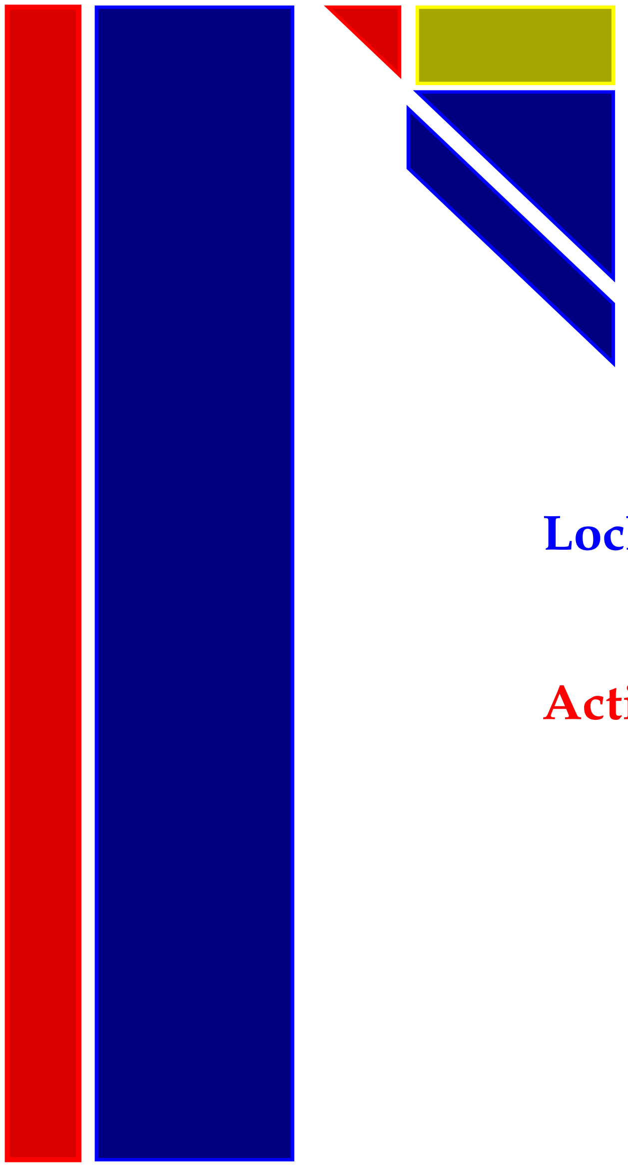
$$\mathbf{A}\mathbf{V}_r^+ = \mathbf{V}_r^+\mathbf{H}_r^+ + \mathbf{F}_r^+\mathbf{E}_r^+.$$

Implications

- Connects subspace iteration with the a truncated QR algorithm. Generalization of the result on the full QR algorithm.
- Define an iterative procedure by building out the compressed block reduction to a length m one. Restarting retains the good information. **Fixed Storage Requirements**
- No need for an explicit restart from scratch. Nor apply some polynomial filter in $\mathbf{A}$. **Unnecessary matrix vector products are avoided.**
- **Sorensen (1992) introduced implicit restarting for $b = 1$ (blocksize).**
- **A truncated QR algorithm is simultaneous iteration with an implicit Rayleigh–Ritz step to accelerate convergence.** See Preprint MCS-P648-0297, *Truncated QR Algorithms and the Solution of Large-Scale Eigenvalue Problems* for more details.

Practical Implementation

- An analysis of an implicitly restarting Arnoldi iteration (L. Thesis 1995). **How to go (is it possible ?) to production quality software**.
- Deflation strategies. (L. & Sorensen SIMAX 1996)
- Block Formulation (L. & Maschhoff 1997)
- Software comparison subspace iteration and Arnoldi methods for non-symmetric eigenvalue problems (L. & Scott 1996). **ARPACK** best over all method.



Locked Vectors

Active Reduction

Stability of a Shifting Strategy

Partition the eigenvalues of $\mathbf{H}_m$ ($b = 1$) so that

$$\underbrace{\{\theta_1, \ldots, \theta_k\}}_{\text{wanted}} \cup \underbrace{\{\theta_{k+1}, \ldots, \theta_m\}}_{\text{unwanted}}$$

If we use the p unwanted eigenvalues of $\mathbf{H}_m$ as shifts, then the new starting vector $\mathbf{V}_m\mathbf{Q}_m\mathbf{e}_1$ is a linear combination of approximate *Schur* vectors and

$$\mathbf{H}_m^+ = \begin{bmatrix} \mathbf{H}_k & \mathbf{M} \\ \mathbf{0} & \mathbf{G} \end{bmatrix}$$

The unwanted and wanted ones are decoupled from each other. The wanted ones are pushed to the front.

This is the situation in exact arithmetic. In practical computation, we are using only approximations to the θ_i's.

More Challenging Eigenvalue Problems

Previous methods only require the **application** of a matrix vector product. Sufficient for the well separated extremal eigenvalues.

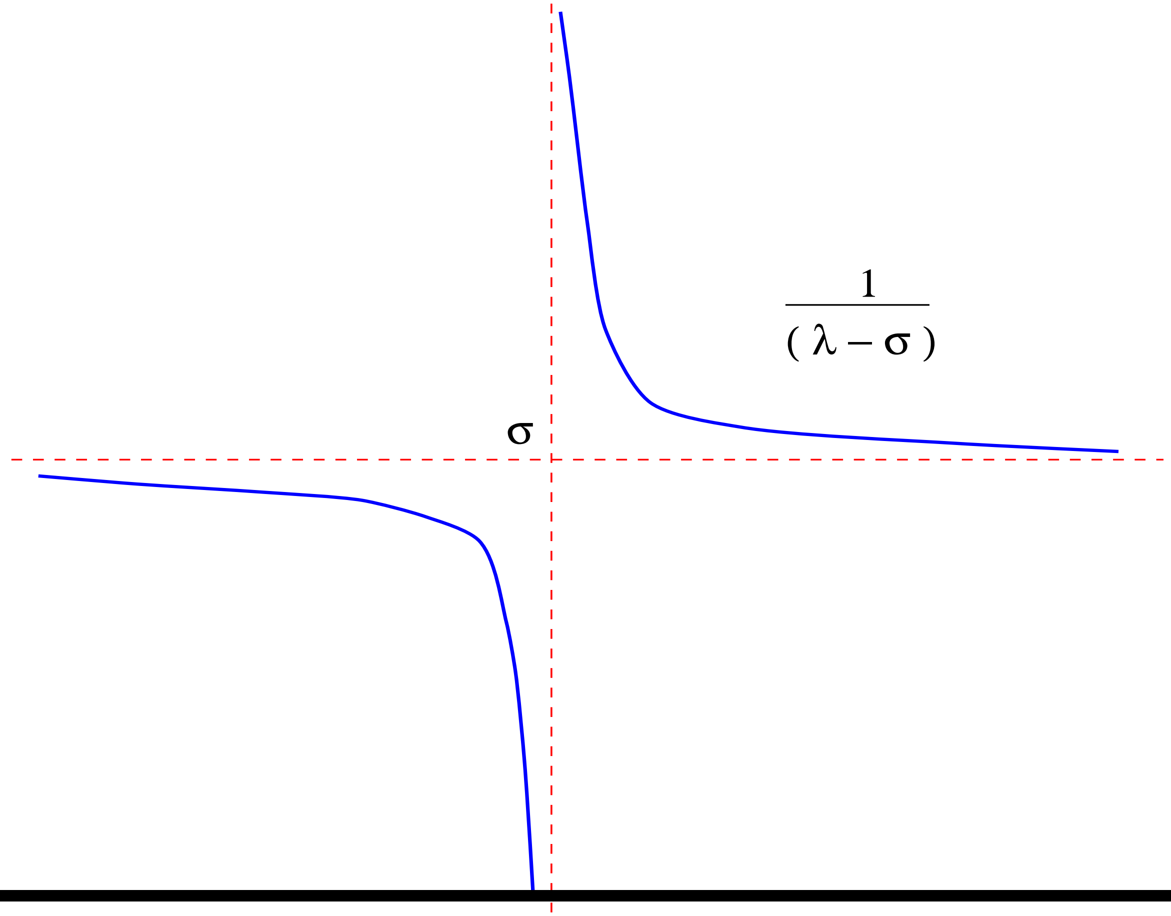
- What if we're interested in the interior eigenvalues ?
- What about a generalized eigenvalue problem $\mathbf{Ax} = \mathbf{Bx}\lambda$?

We assume that the generalized problem cannot be put into standard form. ($\mathbf{LU} = \mathbf{B}$ so that $\mathbf{L}^{-1}\mathbf{AU}^{-1}\mathbf{y} = \mathbf{y}\lambda$ where $\mathbf{y} = \mathbf{Ux}$.)

Generalized eigenvalue problems typically arise in structural analysis problems and linear stability analysis for fluid flow when using the FEM. (I am also involved in project with co–workers in England at Harwell Lab. for the latter application)

Shift and Invert Spectral Transformation

- $(\mathbf{A} - \sigma\mathbf{B})^{-1}\mathbf{B}\mathbf{x} = \mathbf{x}\nu$ where $\nu = 1/(\lambda - \sigma)$.
- ν is large for eigenvalues near σ and small otherwise. The transformation also improves the relative separation.



Computing Guided Waves in Optical Fibers

Consider a 3D closed dielectric wave guide. Assume that the dielectric material is linear, isotropic and translation invariant in the x_3 direction.

Since there is no electric charge and current densities within the dielectric, Maxwell's equations are

$$\epsilon\frac{\partial \mathcal{E}}{\partial t} = \nabla \times \mathcal{H} \quad \text{and} \quad \mu\frac{\partial \mathcal{H}}{\partial t} = -\nabla \times \mathcal{E}$$

where

$$\mathcal{E} \equiv \begin{bmatrix} E_1(x_1, x_2) & E_2(x_1, x_2) & -iE_3(x_1, x_2) \end{bmatrix}^T e^{i(\omega t - \beta x_3)}$$

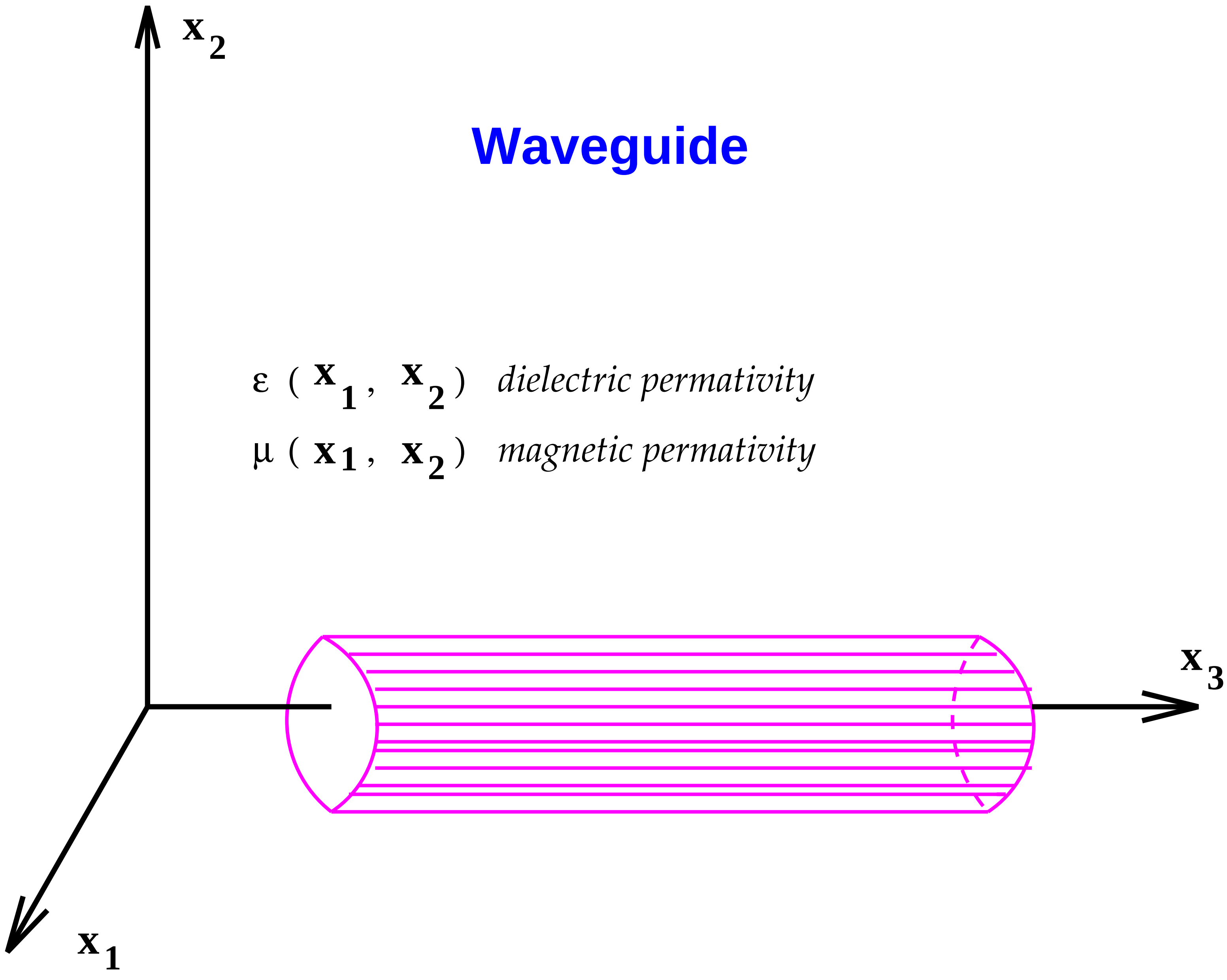
$$\mathcal{H} \equiv \begin{bmatrix} H_1(x_1, x_2) & H_2(x_1, x_2) & iH_3(x_1, x_2) \end{bmatrix}^T e^{i(\omega t - \beta x_3)}$$

are the electric and magnetic fields, respectively, that satisfy Maxwell's equations. **Interest in the fundamental wave numbers β.**

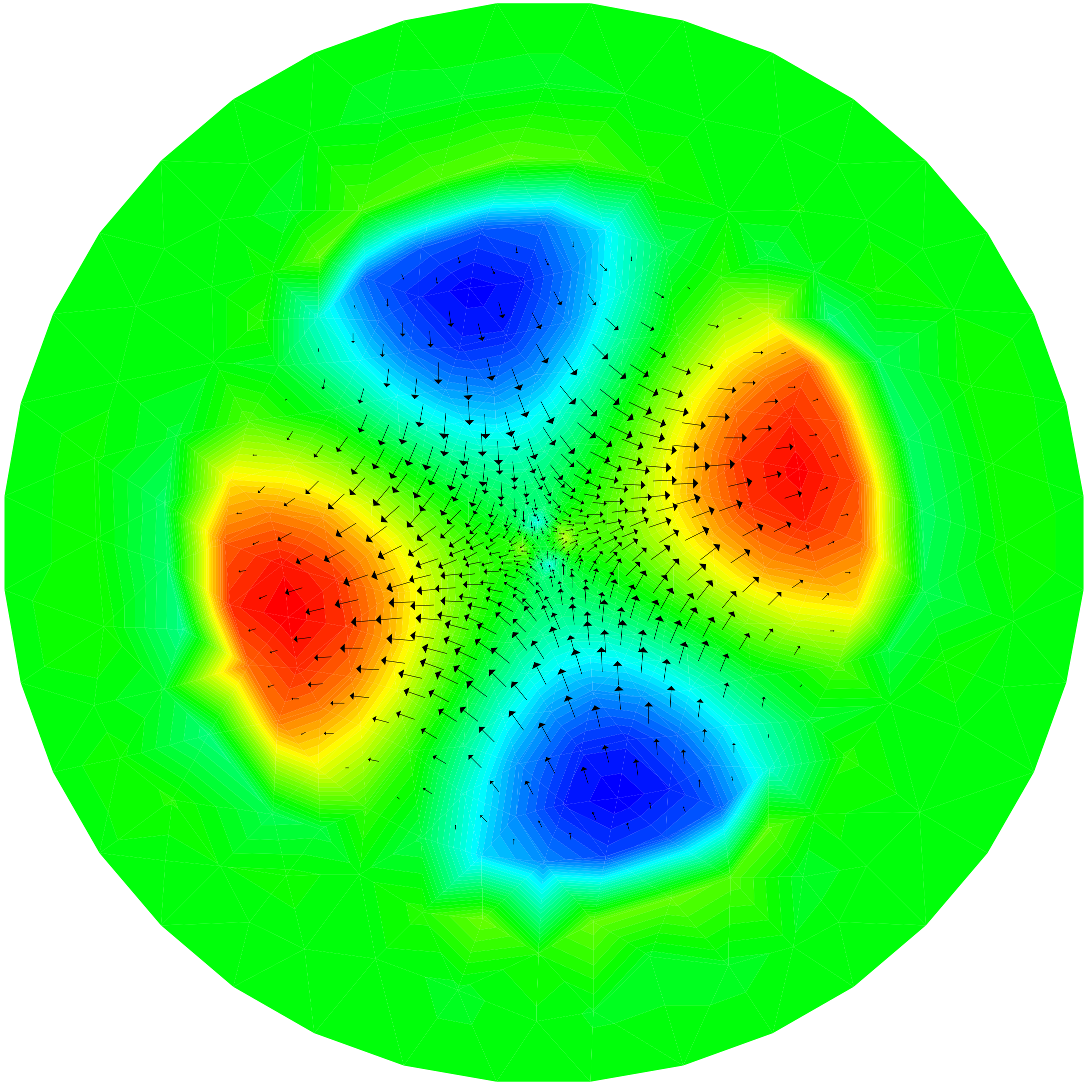
Waveguide

$\varepsilon\,(\mathbf{x}_1,\ \mathbf{x}_2)$ *dielectric permativity*

$\mu\,(\mathbf{x}_1,\ \mathbf{x}_2)$ *magnetic permativity*



Single Core Optic Fiber



Eigenvalue Problem

The solutions give rise to the symmetric eigenvalue problem

$$\nabla \times (\epsilon^{-1} \nabla \times H) = \omega^2 \mu H.$$

Unfortunately, the interest is in the wave number β not the (known) frequency ω. The design problem is to vary ϵ, μ, ω. Since the dielectric permativity varies in the transverse direction, there appear to be two choices. (We use the FEM for the discretization.)

Two standard approaches.

Solve a generalized symmetric indefinite eigenvalue problem

$$\mathbf{Ax} = \mathbf{Wx}\lambda$$

or some non-linear iteration scheme.

A Non-symmetric Eigenvalue Problem

Solve the non-symmetric generalized eigenvalue problem

$$\mathbf{Cx} = \mathbf{Mx}\lambda$$

where $\mathbf{M}$ is symmetric positive definite. This is a **new approach.**(Special choice of FE's & a change of variables)

Use $(\mathbf{A} - \sigma\mathbf{M})^{-1}\mathbf{M}$ and the inner product induced by $\mathbf{M}$. Here σ is an estimate to the smallest wave number.

We want the three smallest wave numbers each of which is a double root.

Current Formulation

Use ARPACK on $(\mathbf{A} - \sigma\mathbf{M})^{-1}\mathbf{M}$. Although this computes the correct wave numbers and associated modes, the solution is **expensive**.

For example, let $n = 13,000$.

- Using Super LU (direct sparse method) uses 32 megabytes of storage for the factorization.
- ARPACK requires 70-80 linear solves and over 200 multiplications with $\mathbf{M}$.
- ARPACK solution takes 17 minutes to get the wave numbers and modes.

A preconditioned iterative method for the linear solves was slower since accurate linear solves are required.

Current Work

Target platform is a 200 MHZ Pentium PC and want to be able to go $n = 50,000$.

- Using an implicitly restarted block Arnoldi method. Amortize costs of the linear solves. Can also increase the convergence tolerance for the eigensolver since the block approach is more reliable.
- **Can we just use a few steps of preconditioned iterative method for the linear solves along with a block approach ?**

Pre-conditioning the Eigenvalue Problem

If interior eigenvalues are wanted, or the extremal portion of the spectrum is poorly separated, then the use of spectral transformations are required.

$$(\mathbf{A} - \sigma \mathbf{M})^{-1} \mathbf{M} \mathbf{x} = \mathbf{x} \nu$$

where $\nu = 1/(\lambda - \sigma)$.

One option is to run Arnoldi with $(\mathbf{A} - \sigma \mathbf{M})^{-1} \mathbf{M}$ and use a direct method. Large storage requirements. Or use an iterative method with a small convergence tolerance.

Is there a middle ground ?

Generalized Block Davidson Method

Given $\mathbf{U}_1$, $\mathbf{U}_1^H\mathbf{U}_1 = \mathbf{I}_b$ & $\mathbf{V}_1 = [\mathbf{U}_1]$

For $j = 1$ to m do

- Compute $\mathbf{H}_j = \mathbf{V}_j^H\mathbf{A}\mathbf{V}_j$ & $\mathbf{K}_j = \mathbf{V}_j^H\mathbf{M}\mathbf{V}_j$.
- Compute $\mathbf{H}_j\mathbf{S} = \mathbf{K}_j\mathbf{S}\boldsymbol{\Theta}$.
- Apply Cayley transformation

$$\mathbf{W} = (\mathbf{A} - \sigma\mathbf{M})^{-1}(\mathbf{A} - \theta\mathbf{M})\mathbf{V}_j\mathbf{S}$$

  approximately.

- Orthonormalise $\mathbf{W}$ against $\mathbf{V}_j$ into $\mathbf{U}_{j+1}$

Problem with Block Davidson Method

Unfortunately, in Generalized Davidson, if $\sigma = \theta_i$, then

$$\begin{aligned}(\mathbf{A} - \theta_i \mathbf{M})\mathbf{w} &= (\mathbf{A} - \theta_i \mathbf{M})\mathbf{V}_j \mathbf{s}_i \\ \mathbf{w} &= \mathbf{V}_j \mathbf{s}_i\end{aligned}$$

Stagnation. Of course, for a Cayley transformation, can't choose the pole and zero equal.

Jacobi–Davidson

Sleijpen & van der Vorst 1996.

Let $\mathbf{y} = \mathbf{V}_j \mathbf{s}_i$.

Jacobi–Davidson attempts to compute $\mathbf{w} \perp \mathbf{y}$ by solving the *correction* equation

$$(\mathbf{I} - \frac{\mathbf{M}\mathbf{y}\mathbf{y}^H}{\mathbf{y}^H\mathbf{M}\mathbf{y}})(\mathbf{A} - \theta_i\mathbf{M})\mathbf{w} = (\mathbf{A} - \theta_i\mathbf{M})\mathbf{y}$$

Note that

$$(\mathbf{I} - \frac{\mathbf{M}\mathbf{y}\mathbf{y}^H}{\mathbf{y}^H\mathbf{M}\mathbf{y}})(\mathbf{A} - \theta_i\mathbf{M})\mathbf{y} = (\mathbf{A} - \theta_i\mathbf{M})\mathbf{y}$$

and that we have a consistent linear system. Set the null space component to zero to get uniqueness.

Jacobi–Davidson as an Inexact Newton Method

$$\mathbf{F}(\theta, \mathbf{y}) = \mathbf{A}\mathbf{y} - \mathbf{M}\mathbf{y}\theta \qquad \mathbf{y}^H\mathbf{y} = 1.$$

Compute corrections $\mathbf{y} + \mathbf{w}$ and $\theta + \omega$ with $\mathbf{y}^H\mathbf{w} = 0$.

A step of Newton's method gives

$$\frac{\partial \mathbf{F}(\theta, \mathbf{y})}{\partial \mathbf{y}}\mathbf{w} + \frac{\partial \mathbf{F}(\theta, \mathbf{y})}{\partial \theta}\omega = -\mathbf{F}(\theta, \mathbf{y}) \equiv \mathbf{r}$$

Postmultiply by $\mathbf{y}^H$ and solving for ω reduces the above equation to the Jacobi Davidson correction equation

$$(\mathbf{I} - \frac{\mathbf{M}\mathbf{y}\mathbf{y}^H}{\mathbf{y}^H\mathbf{M}\mathbf{y}})(\mathbf{A} - \theta\mathbf{M})\mathbf{y} = (\mathbf{A} - \theta\mathbf{M})\mathbf{y}.$$

Using a subspace to compute $\mathbf{y}$ accelerates the process.

A Connection with a Cayley Transformation

Correction equation can be manipulated to get

$$\mathbf{w} = (\mathbf{A} - \theta_i \mathbf{M})^{-1}(\mathbf{A} - (\theta_i + \varepsilon)\mathbf{M})\mathbf{y}.$$

Jacobi-Davidson may be thought of as a Davidson method with zero $\theta_i + \varepsilon$.

Note that

$$(\mathbf{A} - \theta_i \mathbf{M})^{-1}(\mathbf{A} - (\theta_i + \varepsilon_i)\mathbf{M}) = \mathbf{I} - \varepsilon(\mathbf{A} - \theta \mathbf{M})^{-1}$$

but inexactly solving a Cayley transformation leads to a more robust process than solving a shift-and-invert transformation.

Rational Krylov Method

Can we build some sort of block Krylov space and avoid involving A & M when computing the projection matrices ? And, while we're at it, vary the spectral transformation at every step ?

In other words, compute an orthonormal basis for

$$\mathrm{Span}\{\mathbf{U}_1,\ \mathcal{T}_1(\mathbf{A},\mathbf{M})\mathbf{U}_1,\ \cdots\ ,\ (\mathcal{T}_{m-1}(\mathbf{A},\mathbf{M})\cdots\mathcal{T}_1(\mathbf{A},\mathbf{M}))\mathbf{U}_1\}$$

where $\mathcal{T}_j$ is rational function in $\mathbf{A}$ & $\mathbf{M}$? Yes, extend Rational Krylov algorithm (Ruhe 1984) to a block formulation.

But we still need accurate solves.

Inexact Rational Krylov Method

L. & Meerbergen make a special choice for the rational functions, namely Cayley transformations and use inexact linear solves.

Avoid explicit computation of the projection matrices $\mathbf{V}_j^H \mathbf{A} \mathbf{V}_j$ and $\mathbf{V}_j^H \mathbf{M} \mathbf{V}_j$.

See *The Inexact Rational Krylov Sequence Method*, (with Karl Meerbergen). Preprint MCS-P612-1096. Accepted SIMAX pending revisions.

Cayley vs. Shift-and-invert

Approximately solve

$$\begin{array}{ll} \text{(Shift-and-invert:)} & (\mathbf{A}-\sigma\mathbf{B})\mathbf{w} = \mathbf{B}\mathbf{V}_j\mathbf{s} \\ \text{(Cayley:)} & (\mathbf{A}-\sigma\mathbf{B})\mathbf{w} = (\mathbf{A}-\theta\mathbf{B})\mathbf{V}_j\mathbf{s} \end{array}$$

for the portion of $\mathbf{w}$ orthonormal to $\mathbf{V}_j$. It follows that

$$(\mathbf{A}-\theta\mathbf{B})\mathbf{V}_j\mathbf{s} \equiv \mathbf{r}_j = \mathbf{f}_j + \mathbf{E}_j\mathbf{V}_j\mathbf{s}$$

where $\mathbf{E}_j$ is the backward error arising from solving each linear system approximately.

Theorem. Suppose that $\|\mathbf{f}_j\| \leq \rho^j\|\mathbf{f}_0\|$ and that the two linear systems are solved to a relative residual tolerance τ. Then

Cayley transformation	Shift-and-Invert transformation
$\|\mathbf{r}_j\| \leq (\rho+\tau)^j\|\mathbf{f}_0\| + \rho^j\|\mathbf{r}_0\|$	$\|\mathbf{r}_j\| \leq \rho^j\|\mathbf{f}_0\| + \tau\|\mathbf{B}\mathbf{V}_j\mathbf{s}\|$

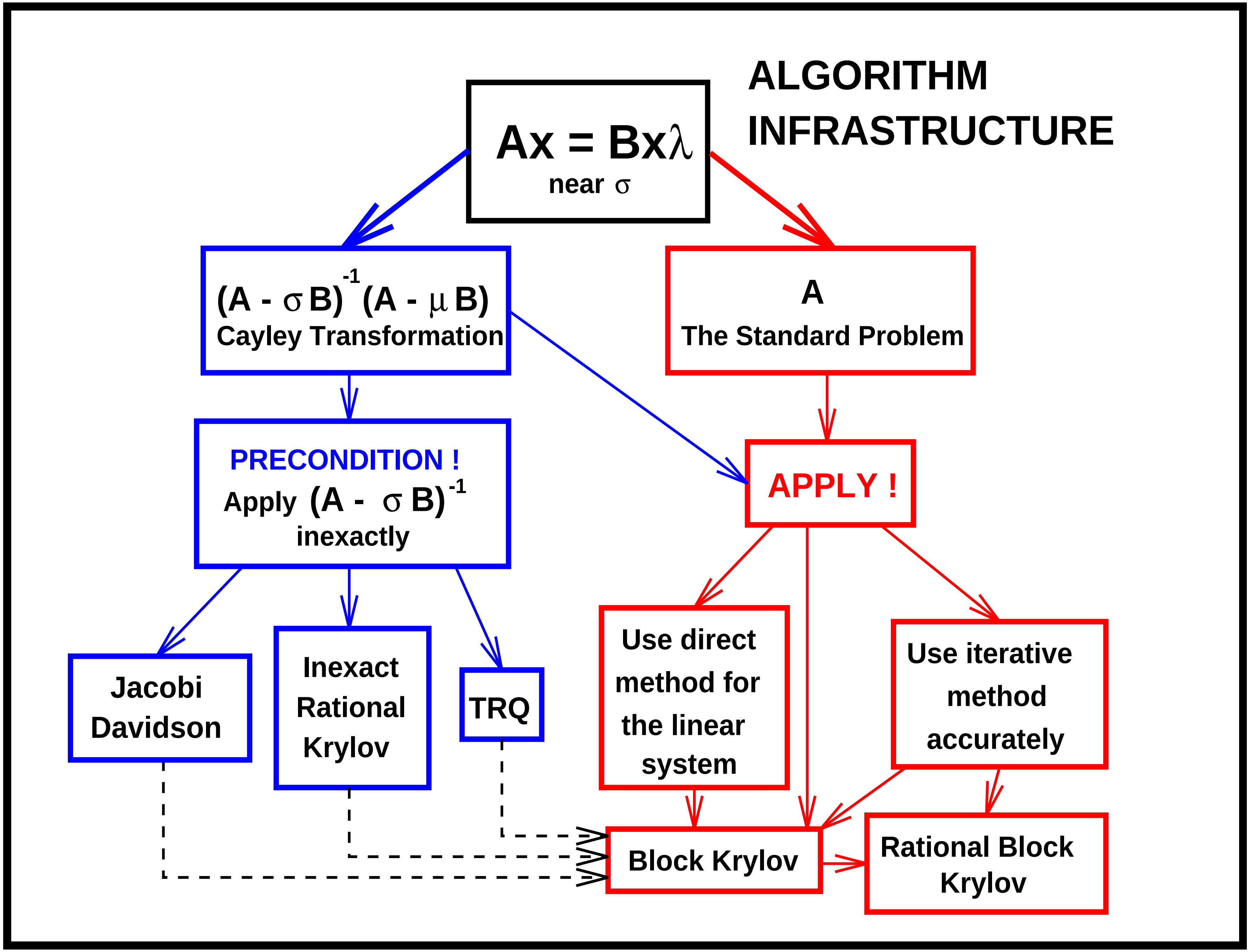
Pre-conditioning the Eigenvalue Problem

Much research remains.

- How to precondition the often indefinite linear systems ?
- How accurate should the linear systems be solved ?

Other approaches. Jacobi–Davidson, truncated RQ iteration by Sorensen & Yang 1996. Long history of special purpose approaches.

May 14–16,1997 Theory Institute on Pre-conditioning the Eigenvalue Problem at Argonne. (L. & Plassmann) See `http://www.mcs.anl.gov/home/lehoucq/workshop.html`


ALGORITHM INFRASTRUCTURE
$Ax = Bx\lambda$
near σ
$(A - \sigma B)^{-1}(A - \mu B)$
Cayley Transformation
A
The Standard Problem
PRECONDITION !
Apply $(A - \sigma B)^{-1}$ inexactly
APPLY !
Jacobi Davidson
Inexact Rational Krylov
TRQ
Use direct method for the linear system
Use iterative method accurately
Block Krylov
Rational Block Krylov

Computing in Finite Precision Arithmetic

The subdiagonal element in the (2,1) block will not be zero.

If $\mathbf{s}_i$ is the eigenvector associated with θ_i, then the subdiagonal element could be as large as

$$\mathcal{O}(\epsilon_M / (\mathbf{e}_m^T \mathbf{s}_i \|\mathbf{A}\|)).$$

(Recall that $\|\mathbf{A}\mathbf{V}_m \mathbf{s}_i - \mathbf{V}_m \mathbf{s}_i \theta_i\| = \|\mathbf{f}_m\| \; |\mathbf{e}_m^T \mathbf{s}_i|$)

The problem is that the QR algorithm is NOT forward stable.

Thus, the wanted Ritz values occasionally cannot be pushed to the front with the shifting mechanism.

The QR algorithm

Backward stability. Let $\hat{\mathbf{Z}}^T\hat{\mathbf{Z}} = \mathbf{I}$,

$$(\mathbf{A}+\mathbf{E})\hat{\mathbf{Z}} \quad = \quad \hat{\mathbf{Z}}\hat{\mathbf{H}}^+,$$

where $\|\mathbf{E}\| \approx \epsilon_M\|\mathbf{A}\|$. The machine precision is denoted by ϵ_M.

Forward stability. Suppose that $\mathbf{AZ} = \mathbf{ZH}^+$ is the exact real Schur form. Is

$$\|\mathbf{H}^+ - \hat{\mathbf{H}}^+\| \quad \leq \quad \epsilon_M\|\mathbf{A}\|?$$

QR algorithm may be very sensitive to a shift. Equivalently, sensitive to the starting vector.

Related work

Wilkinson noticed that the subdiagonal element may not be small even when a computed eigenvalue is used as a shift.

Recently, Parlett & Le (1994) derived necessary and sufficient conditions for the forward instability of the symmetric QR algorithm.

Watkins (1995) examined the nonsymmetric QR algorithm.

New work (L. Thesis 1995) connected the above behavior to Arnoldi/Lanczos methods. Also, a connection with reordering Schur forms.

Significance

Implicit restarting attempts to restart with only wanted Schur vectors. But...

- Cannot use unwanted Ritz values with small Ritz estimates. Could incur violent forward instability of the QR algorithm. So how we purge these unwanted values ?
- Cannot guarantee that the converged Ritz values are located in the leading portion of the reduction. How can we lock these converged directions and thus reduce the size of the active reduction ?

Bounding the size of $\|\hat{\mathbf{f}}_k\| = \hat{\beta}_{k+1}$

$\mathbf{A}\mathbf{V}_k = \mathbf{V}_k\mathbf{H}_k$ ($\mathbf{f}_k = 0$) is a length m truncated Arnoldi reduction where $\mathbf{H}_k$ is unreduced.

$\mathbf{v}_1 = \mathbf{V}_k\mathbf{e}_1 = \mathbf{Z}_k\mathbf{y}$ where $\mathbf{A}\mathbf{Z}_k = \mathbf{Z}_k\mathbf{T}_k$ is a partial Schur form.

If $\tau\hat{\mathbf{v}}_1 = \mathbf{v}_1 + \mathbf{w}$ is an unit vector with $\mathbf{Z}_k^T\mathbf{w} = \mathbf{0}$
with$\mathbf{A}\hat{\mathbf{V}}_k = \hat{\mathbf{V}}_k\hat{\mathbf{H}}_k + \hat{\mathbf{f}}_k\mathbf{e}_k^T$ and $\hat{\mathbf{V}}_j\mathbf{e}_1 = \tau\hat{\mathbf{v}}_1$.

$$\|\hat{\mathbf{f}}_k\| \quad \leq \quad \frac{\|\mathbf{A}^k\mathbf{v}_1\|}{\hat{\beta}_2\cdots\hat{\beta}_k} \;\; \max\left\{\frac{\|\mathbf{K}_m(\mathbf{A},\mathbf{w})\|}{\|\mathbf{K}_m(\mathbf{A},\mathbf{v}_1)\|}, \frac{\|\mathbf{A}^m\mathbf{w}\|}{\|\mathbf{A}^m\mathbf{v}_1\|}\right\}.$$

Proof: Use Ruhe's (LAA 1984) least squares characterization of the Arnoldi reduction and use least squares perturbation theory.

Deflation (Locking) and Purging

Let $\mathbf{H}_m\mathbf{s} = \theta\mathbf{s}$ where θ is a real number and $\|\mathbf{s}\| = 1$. Let $\mathbf{P}$ be a Householder matrix such that $\mathbf{P}\mathbf{s} = \pm\mathbf{e}_1$. Then

$$\begin{aligned} \mathbf{e}_m^T\mathbf{P} &= \mathbf{e}_m^T + \mathbf{r}^T, \quad \|\mathbf{r}\| \leq \sqrt{2}|\mathbf{e}_m^T\mathbf{s}|, \\ \mathbf{P}^T\mathbf{H}_m\mathbf{P}\mathbf{e}_1 &= \mathbf{e}_1\theta. \end{aligned}$$

Thus,

$$\mathbf{A}\mathbf{V}_m\mathbf{P} = \mathbf{V}_m\mathbf{P}\begin{bmatrix} \theta & \mathbf{g}^T \\ 0 & \mathbf{G}_{m-1} \end{bmatrix} + \mathbf{f}_m\mathbf{e}_m^T + \mathbf{f}_m\mathbf{r}^T$$

$\mathbf{V}_m\mathbf{P}\mathbf{e}_1 = \mathbf{V}_m\mathbf{s}$ is a Ritz vector.

Purging uses a Sylvester equation approach to remove $\mathbf{g}^T$ and decouple $\mathbf{V}_m\mathbf{s}$ from the remainder of the reduction. (L. & Sorensen 1996, L. Thesis 1995)

ARPACK: FORTRAN 77 software

R.B. Lehoucq, D.C. Sorensen, P. Vu, C. Yang

Computes solutions to

$$\mathbf{Ax} \quad = \quad \mathbf{Mx}\lambda$$

M is symmetric positive semi-definite.

- Extensive use of LAPACK and Level I, II, III BLAS
- Real and complex arithmetic nonsymmetric, and symmetric (Lanczos) versions.
- Reverse communication interface.
- Fixed storage requirements.
- Distributed memory ARPACK—PARPACK. (MPI and BLACS implementations) K. J. Maschhoff

Available Software

- ARNCHEB - Braconnier, Bennai (1993). Saad & Chatelin ideas, ERA method with Chebyshev acceleration.
- EB13 - Scott 1993, HSL, ACM TOMS (1995). A deflated ERA method with Chebyshev acceleration.
- ARPACK - Lehoucq,Sorensen,Vu,Yang (1996). IRA method.
- SRRIT - Stewart and Bai (1996) ACM TOMS. Subspace iteration.
- EB12 - Duff and Scott (1994) ACM TOMS. Subspace iteration.

Model Problem

Consider the eigenvalue problem for the 2-D convection–diffusion operator,

$$-\Delta u + \rho(u_x + u_y) \quad = \quad \lambda u.$$

Discretize using centered finite differences on the unit square $[0,1] \times [0,1]$ with zero boundary data.

- Eigenvalues and eigenvectors of the resulting matrix are known explicitly
- Many eigenvalues have multiplicity two.
- As the mesh size decreases, relative separation of all the eigenvalues decreases. All are contained within the interval (0, 8).
- As ρ increases so does the non-normality of the matrix.

Timings for 2-D convection-diffusion ($\rho = 30$) matrix of order $10,000$.

Computed residuals are of $\mathcal{O}(10^{-12})$. All computed eigenvalues are seven digits accurate.

Table entries give time (seconds) and number of matrix vector products. Subspace dimension is denoted by m.

Code	$m = 18$	$m = 36$
SRRIT	*	*
EB12	284/41857	292/34268
ARNCHEB	*	*
ARPACK	61/991	80/1095
EB13	727/20004	436/7263

A $*$ denotes that code did not converge within $4000m$ matrix vector products. Timings on an IBM RS-6000 BST.