Introduction to Simulation - Lecture 6

Krylov-Subspace Matrix Solution Methods

Jacob White

Thanks to Deepak Ramaswamy, Michal Rewienski,
and Karen Veroy

Outline

- General Subspace Minimization Algorithm
  - Review orthogonalization and projection formulas
- Generalized Conjugate Residual Algorithm
  - Krylov-subspace
  - Simplification in the symmetric case.
  - Convergence properties
- Eigenvalue and Eigenvector Review
  - Norms and Spectral Radius
  - Spectral Mapping Theorem

Arbitrary Subspace methods

Approach to Approximately Solving Mx=b

Pick a k-dimensional Subspace ⇒ $\left\{ \begin{bmatrix} w_{0_1} \\ \vdots \\ w_{0_N} \end{bmatrix}, \ldots, \begin{bmatrix} w_{k-1_1} \\ \vdots \\ w_{k-1_N} \end{bmatrix} \right\} \equiv \left\{ \vec{w}_0, \ldots, \vec{w}_{k-1} \right\}$

Approximate x^k as a weighted sum of $\left\{ \vec{w}_0, \ldots, \vec{w}_{k-1} \right\}$

$$\Rightarrow \quad x^k = \sum_{i=0}^{k-1} \alpha_i \vec{w}_i$$

Arbitrary Subspace methods

Residual Minimization

The residual is defined as $r^k \equiv b - Mx^k$

If $x^k = \sum_{i=0}^{k-1} \alpha_i \vec{w}_i$

$$\Rightarrow r^k = b - Mx^k = b - \sum_{i=0}^{k-1} \alpha_i M\vec{w}_i$$

Residual Minimizing idea: pick α_i 's to minimize

$$\left\| r^k \right\|_2^2 \equiv \left(r^k\right)^T \left(r^k\right) = \left(b - \sum_{i=0}^{k-1} \alpha_i M\vec{w}_i \right)^T \left(b - \sum_{i=0}^{k-1} \alpha_i M\vec{w}_i \right)$$

Arbitrary Subspace methods

Residual Minimization

Computational Approach

Minimizing $\left\| r^k \right\|_2^2 = \left\| b - \sum_{i=0}^{k-1} \alpha_i M\vec{w}_i \right\|_2^2$ is easy if

$\left(M\vec{w}_i\right)^T \left(M\vec{w}_j\right) = 0,\ i \neq j$ or $\left(M\vec{w}_i\right)$ orthogonal to $\left(M\vec{w}_j\right)$

Create a set of vectors $\left\{ \vec{p}_0, \ldots, \vec{p}_{k-1} \right\}$ such that

$span\left\{ \vec{p}_0, \ldots, \vec{p}_{k-1} \right\} = span\left\{ \vec{w}_0, \ldots, \vec{w}_{k-1} \right\}$

and $\left(M\vec{p}_i\right)^T \left(M\vec{p}_j\right) = 0,\ i \neq j$

Arbitrary Subspace methods

Residual Minimization

Algorithm Steps

Given M, b and a set of search directions $\{\vec{w}_0, \ldots, \vec{w}_{k-1}\}$

1) Generate $\vec{p}_j$'s by orthogonalizing $\mathrm{M}w_j$'s

$$\text{For } j = 0 \text{ to } k-1 \quad p_j = w_j - \sum_{i=0}^{j-1} \frac{(Mw_j)^T (Mp_i)}{(Mp_i)^T (Mp_i)} p_i$$

2) compute the r minimizing solution x^k

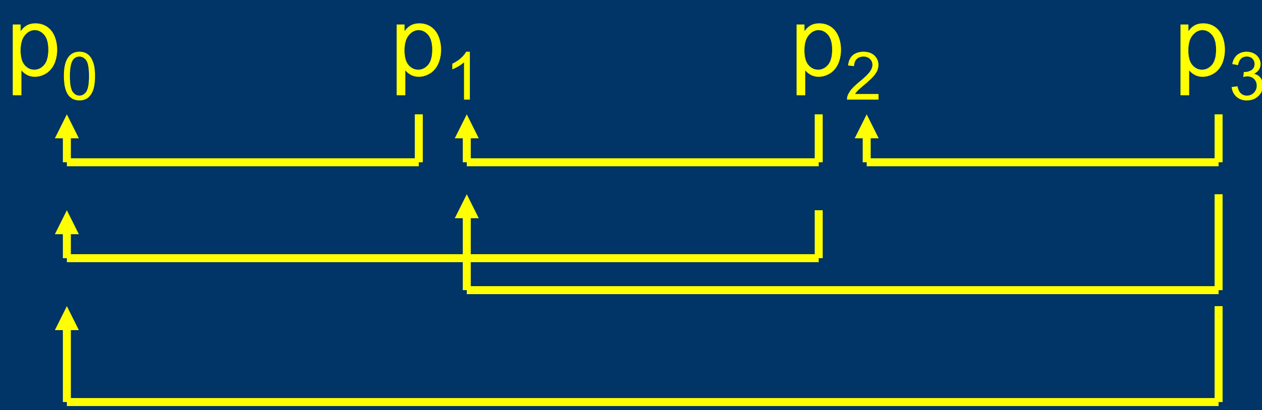
$$x^k = \sum_{i=0}^{k-1} \frac{(r^0)^T (Mp_i)}{(Mp_i)^T (Mp_i)} p_i = \sum_{i=0}^{k-1} \frac{(r^i)^T (Mp_i)}{(Mp_i)^T (Mp_i)} p_i$$

Arbitrary Subspace methods

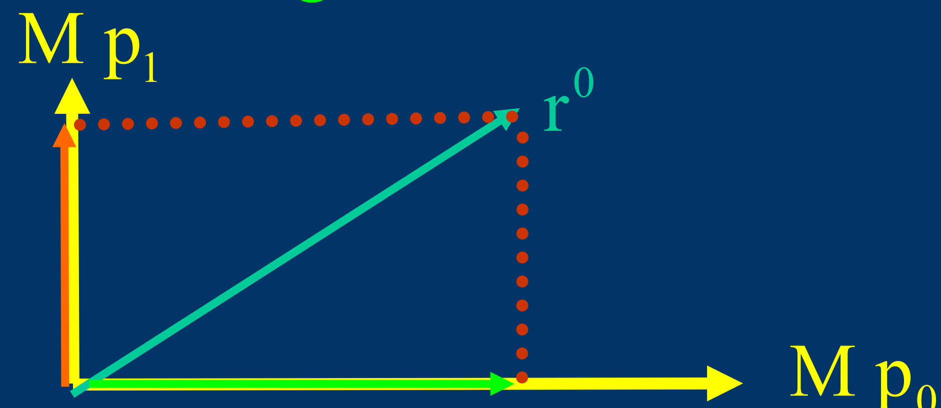
Residual Minimization

Algorithm Steps by Picture

1) orthogonalize the Mw_i's



2) compute the r minimizing solution x^k



Arbitrary Subspace Solution Algorithm

Minimization Algorithm

$r^0 = b - Ax^0$

For j = 0 to k-1

$p_j = w_j$

For i = 0 to j-1

$p_j \leftarrow p_j - \left(Mp_j\right)^T \left(Mp_i\right) p_i$ } Orthogonalize Search Direction

$p_j \leftarrow \dfrac{1}{\sqrt{\left(Mp_j\right)^T \left(Mp_j\right)}} p_j$ } Normalize

$x^{j+1} = x^j + \left(r^j\right)^T \left(Mp_j\right) p_j$ } Update Solution

$r^{j+1} = r^j - \left(r^j\right)^T \left(Mp_j\right) M\, p_j$ } Update Residual

Arbitrary Subspace methods

Subspace Selection

Criteria

Criteria for selecting $w_0, \ldots, w_{k-1}$

All that matters is the $span\{w_0, \ldots, w_{k-1}\}$

$\exists\ \alpha_i$'s such that $b - Mx^k = b - \sum_{i=0}^{k-1} \alpha_i M \vec{w}_i$ is small

$A^{-1}b \approx$ in the $span\{w_0, \ldots, w_{k-1}\}$ for $k \ll N$

One choice, unit vectors, $x^k \in span\{\vec{e}_1, \ldots, \vec{e}_k\}$

- Generates the QR algorithm if k=N
- Can be terrible if k < N

Arbitrary Subspace methods

Subspace Selection

Historical Development

Consider minimizing $f\left(x\right)=\dfrac{1}{2}x^{T}Mx-x^{T}b$

Assume $M=M^{T}$ (symmetric) and $x^{T}Mx>0$ (pos. def)

$\nabla_{x}f\left(x\right)=Mx-b \Rightarrow x=M^{-1}b$ minimizes f

Pick $span\left\{w_{0},\ldots,w_{k-1}\right\}=span\left\{\nabla_{x}f\left(x^{0}\right),\ldots,\nabla_{x}f\left(x^{k-1}\right)\right\}$

Steepest descent directions for f, but f is not residual

Does not extend to nonsymmetric, non pos def case

Arbitrary Subspace methods

Subspace Selection

Krylov Subspace

Note: $span\left\{\nabla_x f\left(x^0\right),\ldots,\nabla_x f\left(x^{k-1}\right)\right\} = span\left\{r^0,\ldots,r^{k-1}\right\}$

If: $span\left\{\vec{w}_0,\ldots,\vec{w}_{k-1}\right\} = span\left\{r^0,\ldots,r^{k-1}\right\}$

then $r^k = r^0 - \sum_{i=0}^{k-1} \alpha_i M r^i$

and $span\left\{r^0,\ldots,r^{k-1}\right\} = \underbrace{span\left\{r^0, Mr^0,\ldots,M^{k-1}r^0\right\}}_{Krylov\ Subspace}$

Krylov Methods

The Generalized Conjugate Residual Algorithm

The kth step of GCR

$$\alpha_k = \frac{\left(r^k\right)^T \left(Mp_k\right)}{\left(Mp_k\right)^T \left(Mp_k\right)}$$

Determine optimal stepsize in kth search direction

$$x^{k+1} = x^k + \alpha_k p_k$$
$$r^{k+1} = r^k - \alpha_k M p_k$$

Update the solution and the residual

$$p_{k+1} = r^{k+1} - \sum_{j=0}^{k} \frac{\left(Mr^{k+1}\right)^T \left(Mp_j\right)}{\left(Mp_j\right)^T \left(Mp_j\right)} p_j$$

Compute the new orthogonalized search direction

Krylov Methods

The Generalized Conjugate Residual Algorithm

Algorithm Cost for iter k

$$\alpha_k = \frac{\left(r^k\right)^T \left(Mp_k\right)}{\left(Mp_k\right)^T \left(Mp_k\right)}$$

Vector inner products, O(n)
Matrix-vector product, O(n) if sparse

$$x^{k+1} = x^k + \alpha_k p_k$$
$$r^{k+1} = r^k - \alpha_k M p_k$$

Vector Adds, O(n)

$$p_{k+1} = r^{k+1} - \sum_{j=0}^{k} \frac{\left(Mr^{k+1}\right)^T \left(Mp_j\right)}{\left(Mp_j\right)^T \left(Mp_j\right)} p_j$$

O(k) inner products,
total cost O(nk)

If M is sparse, as k (# of iters) approaches n,

$$\text{total cost } = O(n) + O(2n) + \ldots + O(kn) = O(n^3)$$

Better Converge Fast!

Krylov Methods

The Generalized Conjugate Residual Algorithm

Symmetric Case

<u>An Amazing fact that will not be derived</u>

$$\text{If } M = M^T \text{ then } r^{k+1} \perp Mp^j \quad j < k$$

$$p_{k+1} = r^{k+1} - \sum_{j=0}^{k} \frac{\left(Mr^{k+1}\right)^T \left(Mp_j\right)}{\left(Mp_j\right)^T \left(Mp_j\right)} p_j \quad \Rightarrow \quad p_{k+1} = r^{k+1} - \frac{\left(Mr^{k+1}\right)^T \left(Mp_k\right)}{\left(Mp_k\right)^T \left(Mp_k\right)} p_k$$

Orthogonalization in one step

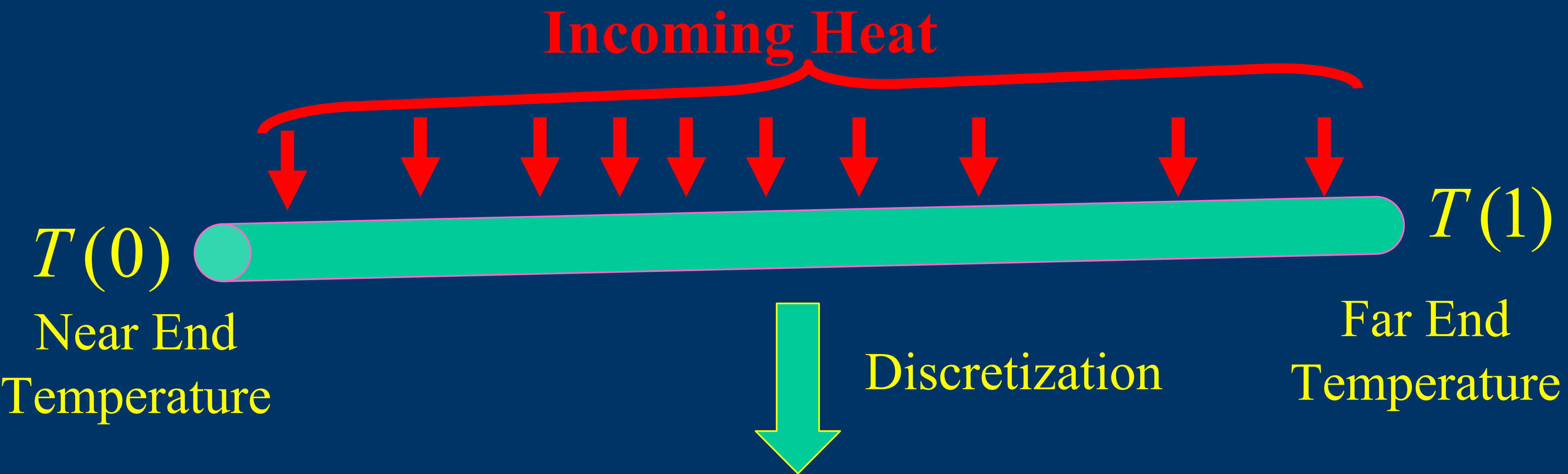
If k (# of iters) → n, then symmetric, sparse, GCR is $O(n^2)$

Better Converge Fast!

Krylov Methods

"No-leak Example"

Insulated bar and Matrix



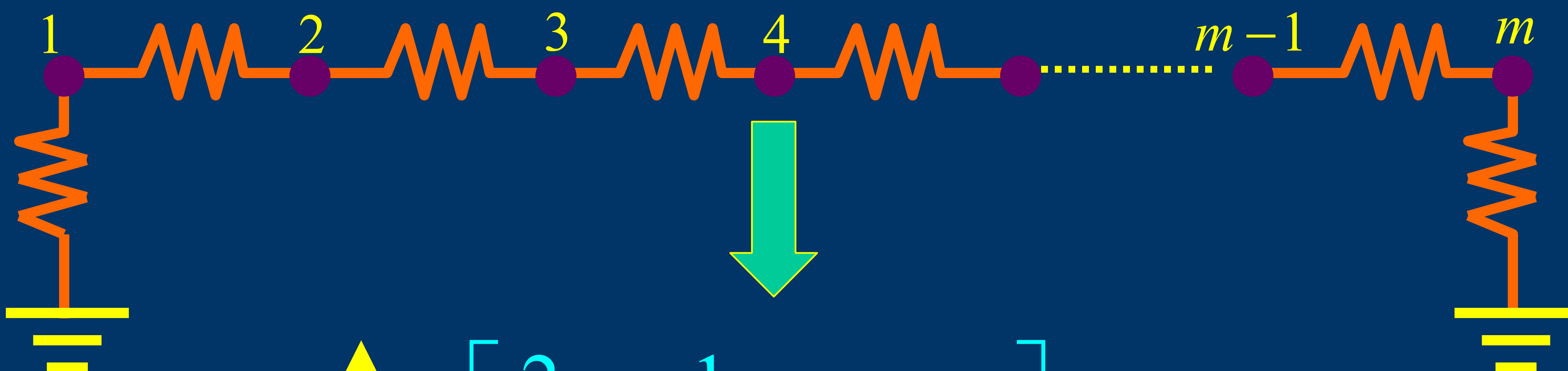
$$m \updownarrow \underbrace{\begin{bmatrix} 2 & -1 & & \\ -1 & 2 & & \\ & & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}}_{M}$$

Nodal Equation Form

Krylov Methods

"No-leak Example"

Circuit and Matrix



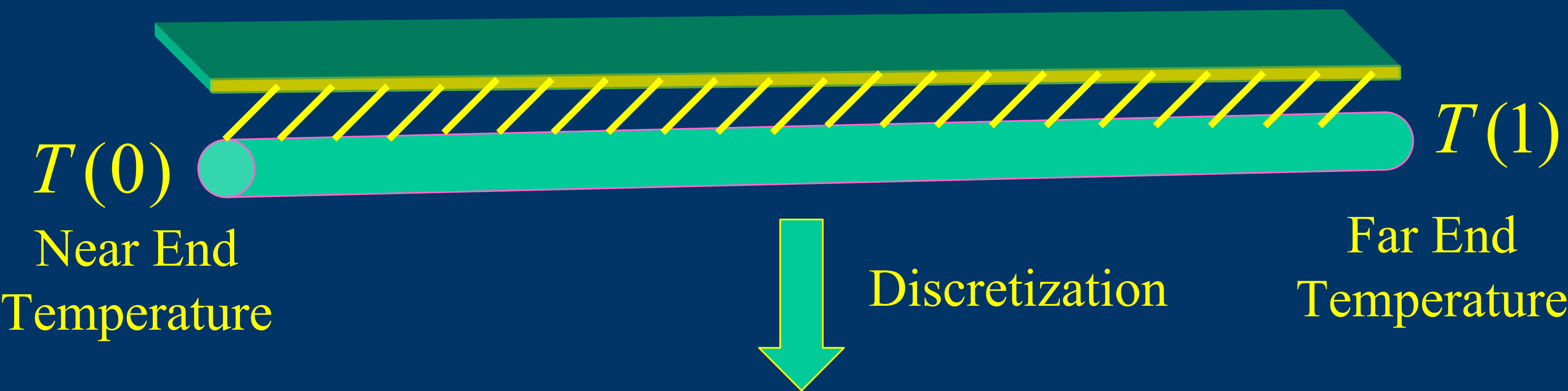
$$
\underbrace{\begin{bmatrix} 2 & -1 & & \\ -1 & 2 & & \\ & & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}}_{M} \quad \updownarrow m
$$

Nodal Equation Form

Krylov Methods

"leaky" Example

Conducting bar and Matrix



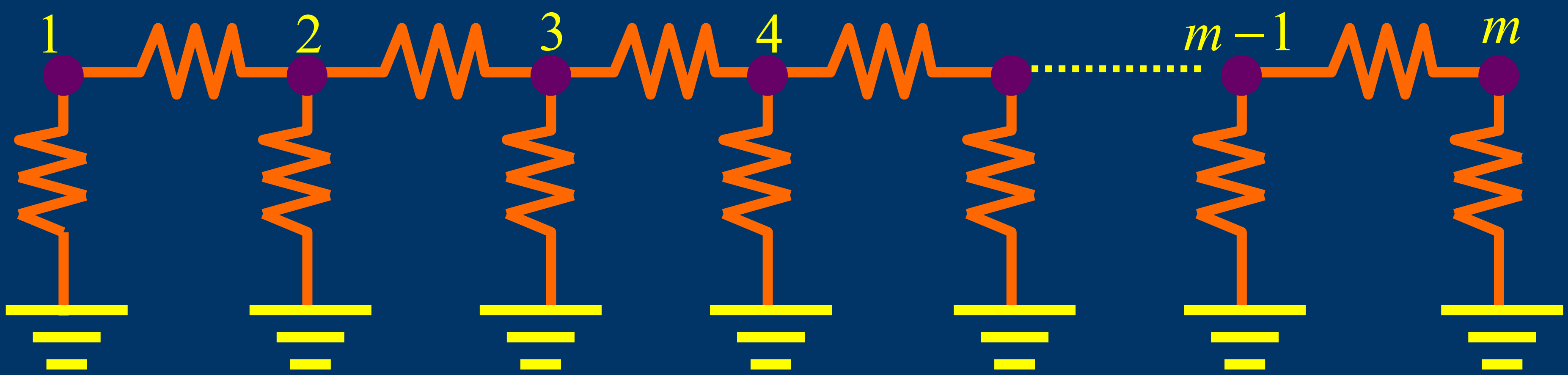
$$
m \left\{ \underbrace{\begin{bmatrix} 2.01 & -1 & & \\ -1 & 2.01 & & \\ & & \ddots & -1 \\ & & -1 & 2.01 \end{bmatrix}}_{M} \right.
$$

Nodal Equation Form

Krylov Methods

“leaky” Example

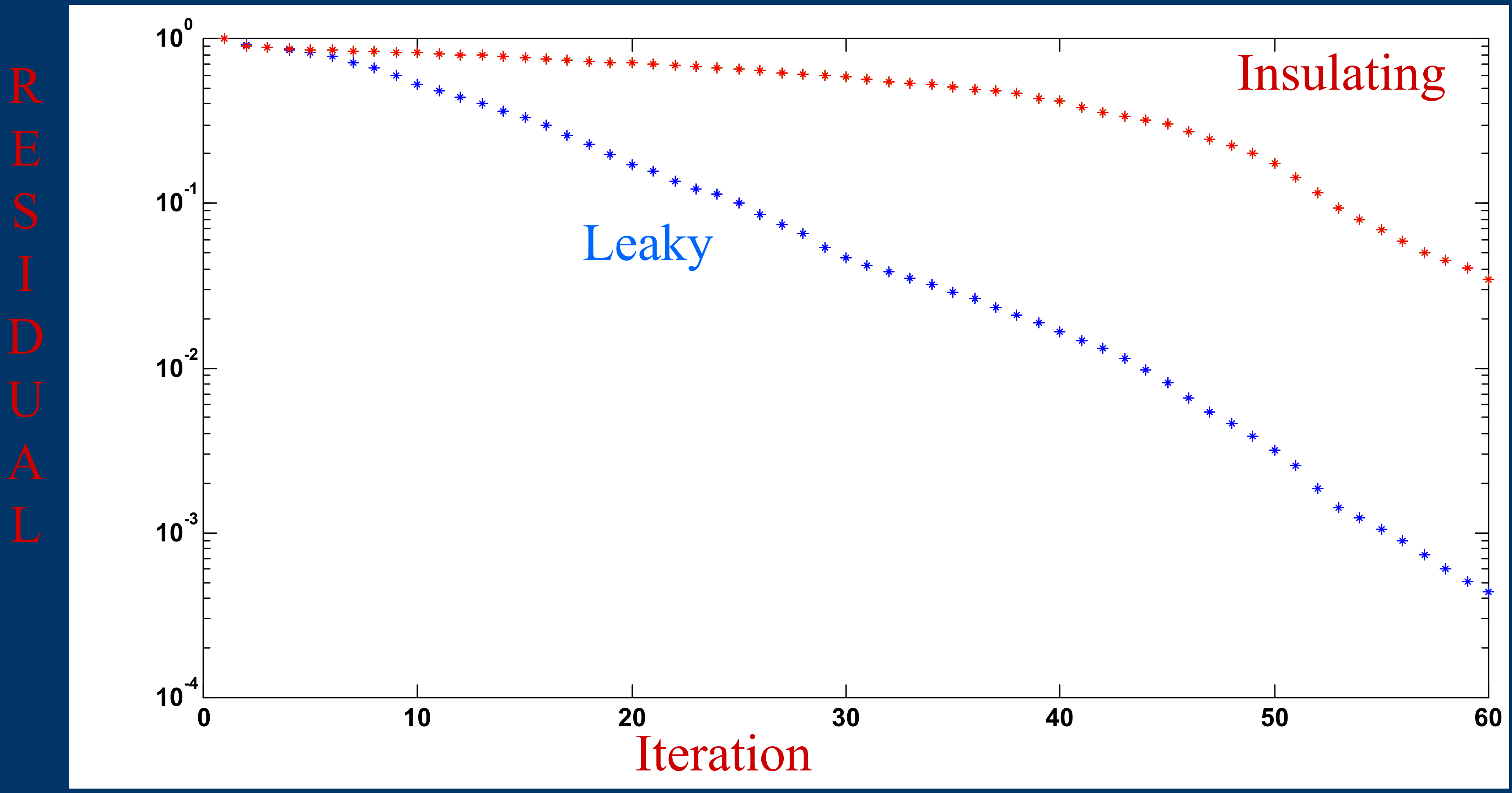
Circuit and Matrix



$$\underbrace{\begin{bmatrix} 2.01 & -1 & & \\ -1 & 2.01 & & \\ & & \ddots & -1 \\ & & -1 & 2.01 \end{bmatrix}}_{M} \quad \updownarrow m$$

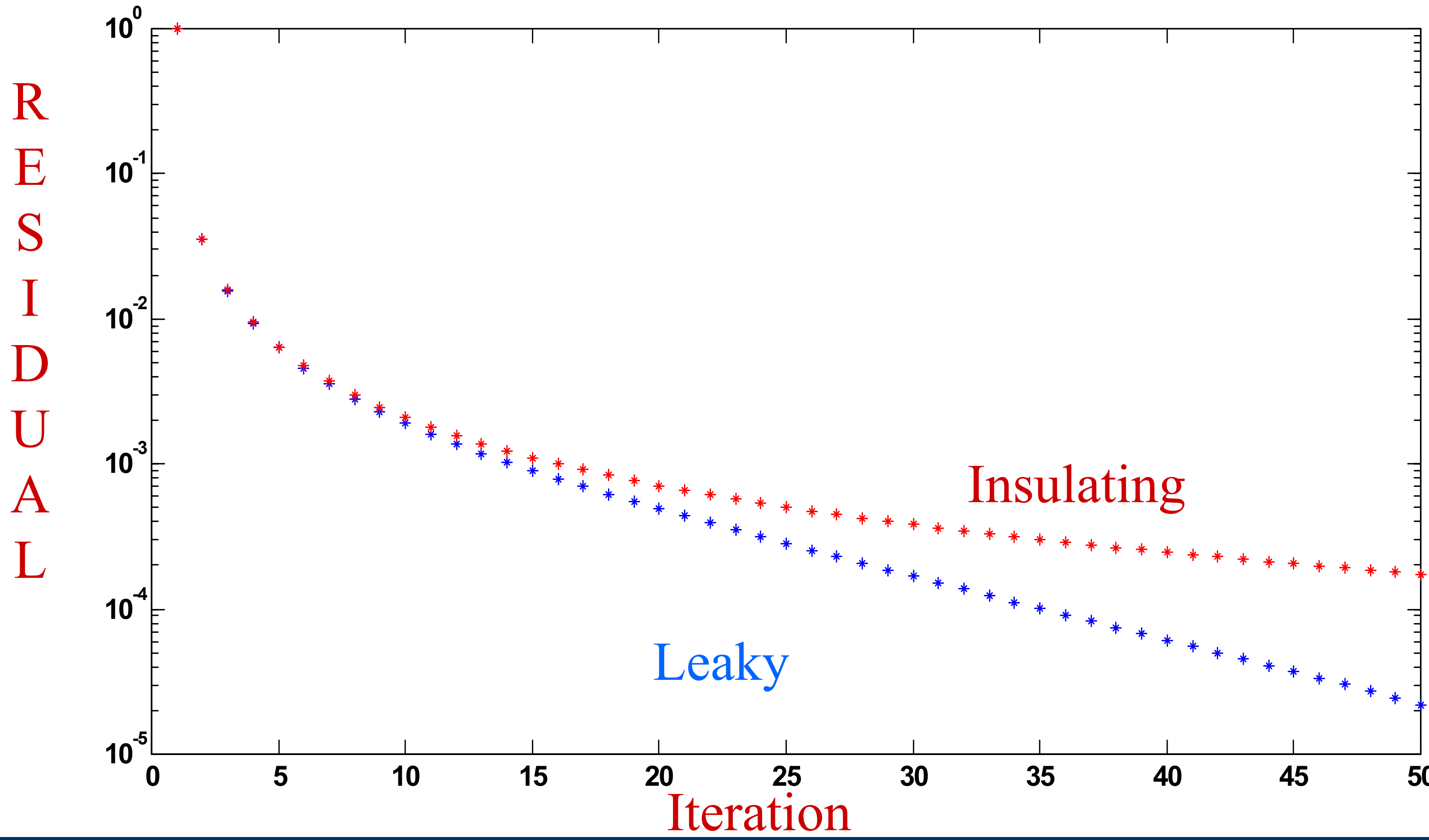
Nodal Equation Form

GCR Performance(Random Rhs)



Plot of *log*(residual) versus Iteration

GCR Performance(Rhs = -1,+1,-1,+1....)



Plot of *log*(residual) versus Iteration

Krylov Subspace Methods

Convergence Analysis

Polynomial Approach

If $\text{span}\left\{w_0,\ldots,w_k\right\} = \text{span}\left\{r^0, Mr^0,\ldots,M^k r^0\right\}$

$$x^{k+1} = \sum_{i=0}^{k} \alpha_i M^i r^0 = \underbrace{\xi_k\left(M\right)}_{\text{kth order polynomial}} r^0$$

$$r^{k+1} = r^0 - \sum_{i=0}^{k} \alpha_i M^{i+1} r^0 = \left(I - M\xi_k\left(M\right)\right) r^0$$

Note: for any $\alpha_0 \neq 0$

$$\text{span}\left\{r^0, r^1 = r^0 - \alpha_0 M r^0\right\} = \text{span}\left\{r^0, Mr^0\right\}$$

Krylov Methods

Convergence Analysis

Basic Properties

If $\alpha_j \neq 0$ for all $j \leq k$ in GCR, then

1) $\text{span}\left\{p_0, p_1, \ldots, p_k\right\} = \text{span}\left\{r^0, Mr^0, \ldots, M^k r^0\right\}$

2) $x^{k+1} = \xi_k(M)r^0$, ξ_k is the kth order polynomial which minimizes $\left\|r^{k+1}\right\|_2^2$

3) $r^{k+1} = b - Mx^{k+1} = r^0 - M\xi_k(M)r^0$

$$= \left(I - M\xi_k(M)\right)r^0 \equiv \wp_{k+1}\left(M\right)r^0$$

where $\wp_{k+1}\left(M\right)r^0$ is the $\left(k+1\right)^{th}$ order poly minimizing $\left\|r^{k+1}\right\|_2^2$ subject to $\wp_{k+1}\left(0\right)=1$

Krylov Methods

Convergence Analysis

Optimality of GCR poly

GCR Optimality Property

$$\left\|r^{k+1}\right\|_2^2 \leq \left\|\tilde{\wp}_{k+1}(M) r^0\right\|_2^2$$ where $\tilde{\wp}_{k+1}$ is any k^{th} order polynomial such that $\tilde{\wp}_{k+1}(0)=1$

Therefore

Any polynomial which satisfies the zero constraint can be used to get an upper bound on $\Rightarrow \left\|r^{k+1}\right\|_2^2$

Eigenvalues and Vectors Review

Basic Definitions

Eigenvalues and eigenvectors of a matrix M satisfy

eigenvalue

$$M\vec{u}_i = \lambda_i \vec{u}_i$$

eigenvector

Or, λ_i is an eigenvalue of M if

$$M - \lambda_i I \text{ is singular}$$

$\vec{u}_i$ is an eigenvector of M if

$$\left(M - \lambda_i I\right)\vec{u}_i = 0$$

Eigenvalues and Vectors Review

Basic Definitions

Examples

$$\begin{bmatrix} 1.1 & -1 \\ -1 & 1.1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

Eigenvalues?
Eigenvectors?

$$\begin{bmatrix} M_{11} & 0 & \cdots & 0 \\ M_{21} & M_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ M_{N1} & \cdots & M_{NN-1} & M_{NN} \end{bmatrix}$$

What about a lower triangular matrix

Eigenvalues and Vectors Review

A Simplifying Assumption

Almost all NxN matrices have N linearly independent Eigenvectors

$$M \begin{bmatrix} \uparrow & \uparrow & \uparrow & \cdots & \uparrow \\ \vec{u}_1 & \vec{u}_2 & \vec{u}_3 & \cdots & \vec{u}_N \\ \downarrow & \downarrow & \downarrow & \cdots & \downarrow \end{bmatrix} = \begin{bmatrix} \uparrow & \uparrow & \uparrow & \cdots & \uparrow \\ \lambda_1 \vec{u}_1 & \lambda_2 \vec{u}_2 & \lambda_3 \vec{u}_3 & \cdots & \lambda_N \vec{u}_N \\ \downarrow & \downarrow & \downarrow & \cdots & \downarrow \end{bmatrix}$$

The set of all eigenvalues of M is known as the Spectrum of M

Eigenvalues and Vectors Review

A Simplifying Assumption Continued

Almost all NxN matrices have N linearly independent Eigenvectors

$$MU = U \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_N \end{bmatrix}$$

⇓

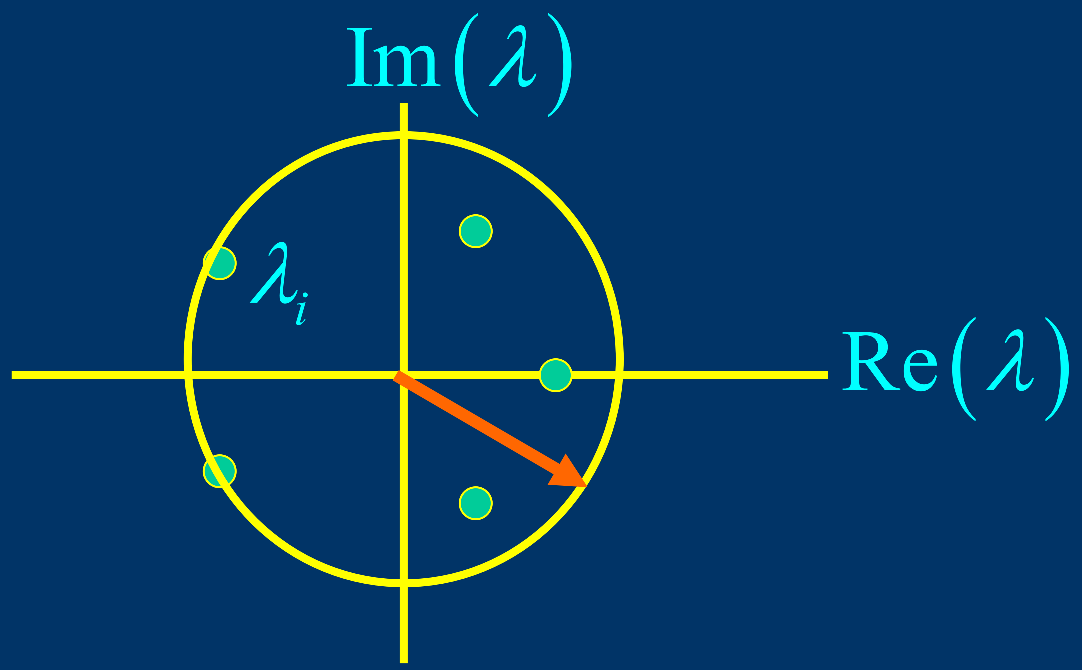
$$U^{-1} M U = \lambda \quad \text{or} \quad M = U \lambda U^{-1}$$

Does NOT imply distinct eigenvalues, λ_i can equal λ_j

Does NOT imply M is nonsingular

Eigenvalues and Vectors Review

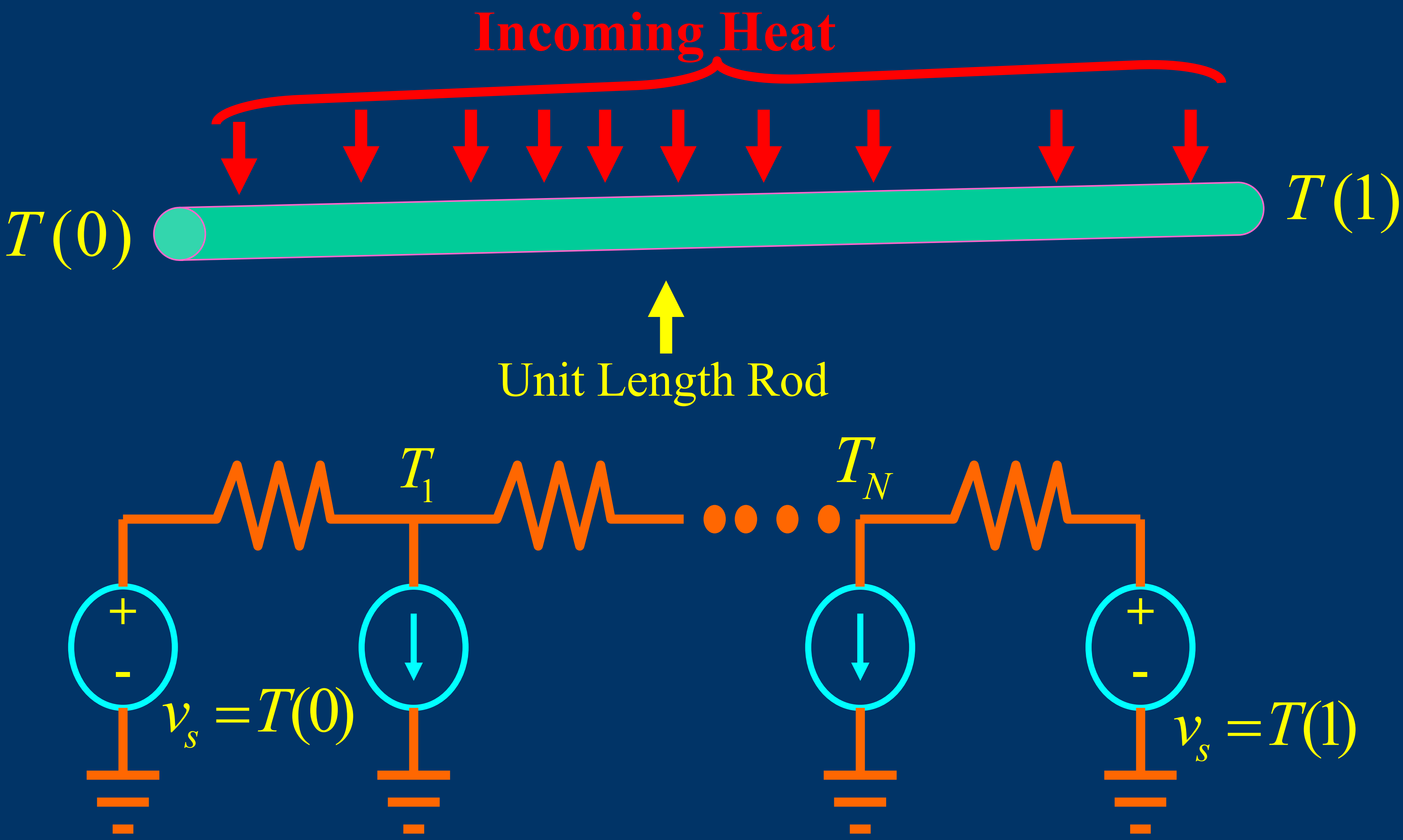
Spectral Radius



The spectral Radius of M is the radius of the smallest circle, centered at the origin, which encloses all of M's eigenvalues

Eigenvalues and Vectors Review

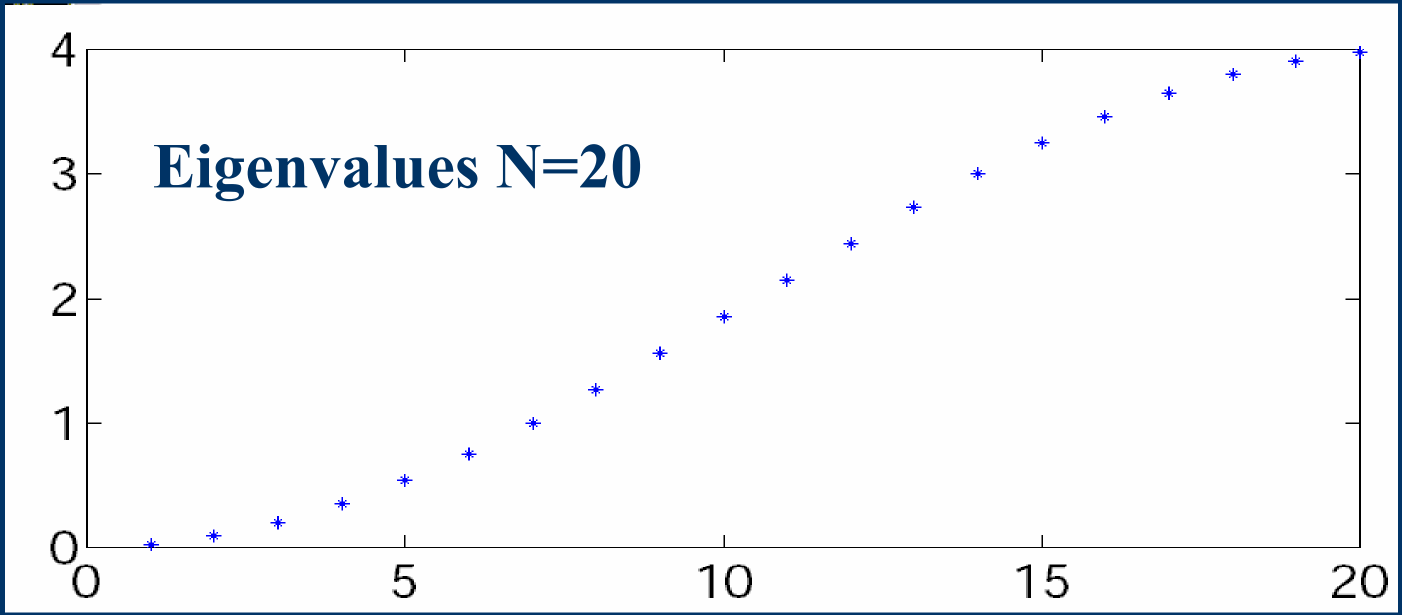
Heat Flow Example



Eigenvalues and Vectors Review

Heat Flow Example Continued

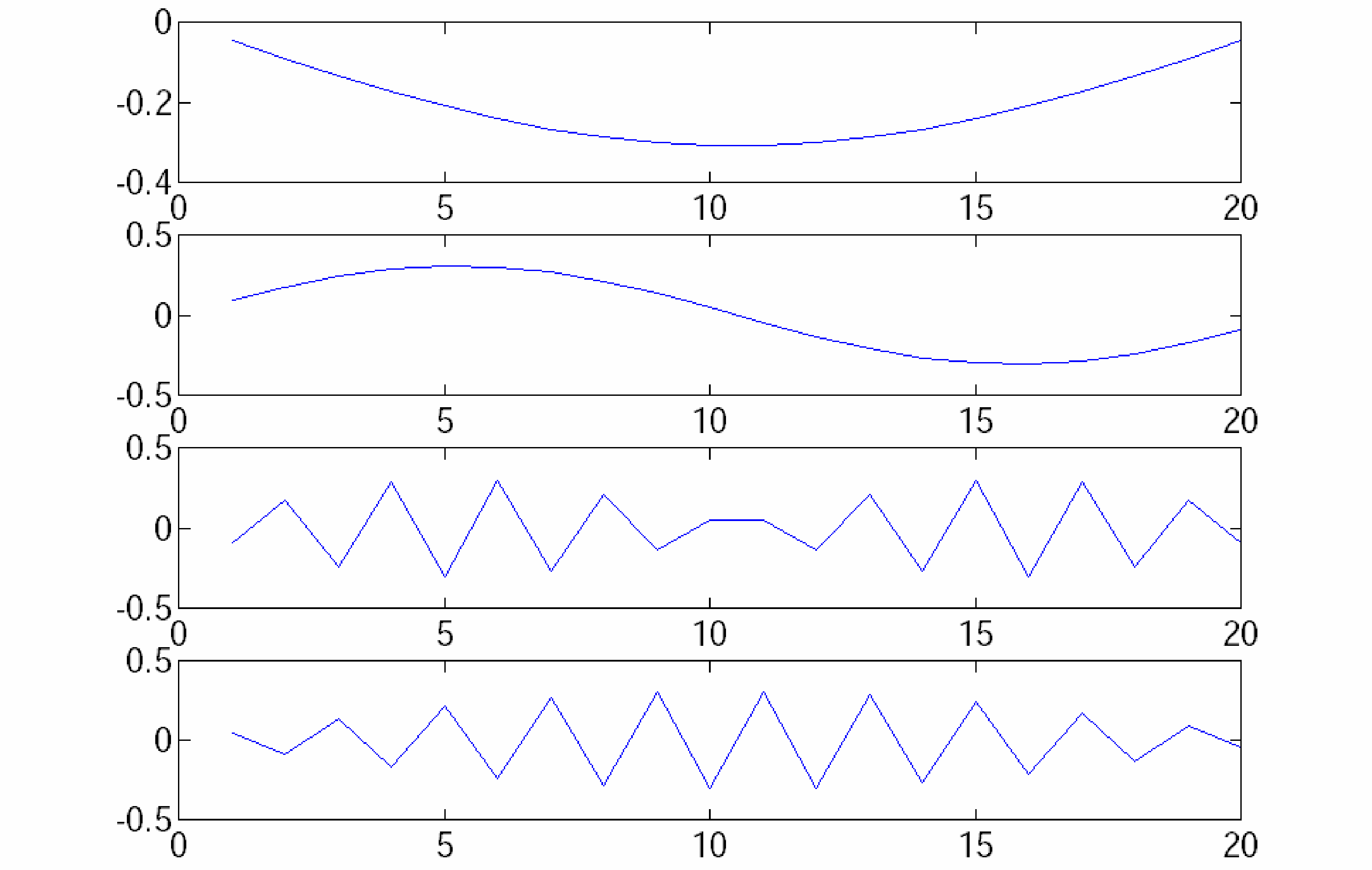
$$\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & \ddots & 0 \\ 0 & \ddots & \ddots & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$



Eigenvalues and Vectors Review

Heat Flow Example Continued

Four Eigenvectors – Which ones?



Useful Eigenproperties

Spectral Mapping Theorem

Given a polynomial

$$f(x) = a_0 + a_1 x + \ldots + a_p x^p$$

Apply the polynomial to a matrix

$$f(M) = a_0 + a_1 M + \ldots + a_p M^p$$

Then

$$spectrum(f(M)) = f(spectrum(M))$$

Useful Eigenproperties

Spectral Mapping Theorem Proof

Note a property of matrix powers

$$MM = U\lambda U^{-1}U\lambda U^{-1} = U\lambda^2 U^{-1}$$

$$\Rightarrow M^p = U\lambda^p U^{-1}$$

Apply to the polynomial of the matrix

$$f(M) = a_0 UU^{-1} + a_1 U\lambda U^{-1} + \ldots + a_p U\lambda^p U^{-1}$$

Factoring $$f(M) = U\underbrace{\left(a_0 I + a_1\lambda + \ldots + a_p\lambda^p\right)}_{Diagonal}U^{-1}$$

$$f(M)U = U\left(a_0 I + a_1\lambda + \ldots + a_p\lambda^p\right)$$

Useful Eigenproperties

Spectral Decomposition

Decompose arbitrary x in eigencomponents

$$x = \alpha_1 \vec{u}_1 + \alpha_2 \vec{u}_2 + \ldots + \alpha_N \vec{u}_N$$

Compute by solving $U \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix} = x \Rightarrow \vec{\alpha} = U^{-1} x$

Applying M to x yeilds

$$Mx = M\left(\alpha_1 \vec{u}_1 + \alpha_2 \vec{u}_2 + \ldots + \alpha_N \vec{u}_N\right)$$

$$= \alpha_1 \lambda_1 \vec{u}_1 + \alpha_2 \lambda_2 \vec{u}_2 + \ldots + \alpha_N \lambda_N \vec{u}_N$$

Krylov Methods

Convergence Analysis

Important Observations

1) The GCR Algorithm converges to the exact solution in at most n steps

$$\text{Proof: Let } \tilde{\wp}_n(x) = (x-\lambda_1)(x-\lambda_2)\ldots(x-\lambda_n)$$
$$\text{where } \lambda_i \in \lambda(M).$$

$$\Rightarrow \left\| \tilde{\wp}_n(M) r^0 \right\| = 0 \quad \text{and therefore } \left\| r^n \right\| = 0$$

2) If M has only q distinct eigenvalues, the GCR Algorithm converges in at most q steps

$$\text{Proof: Let } \tilde{\wp}_q(x) = (x-\lambda_1)(x-\lambda_2)\ldots(x-\lambda_q)$$

Summary

- Arbitrary Subspace Algorithm
  - Orthogonalization of Search Directions
- Generalized Conjugate Residual Algorithm
  - Krylov-subspace
  - Simplification in the symmetric case.
  - Leaky and insulating examples
- Eigenvalue and Eigenvector Review
  - Spectral Mapping Theorem
- GCR limiting Cases
  - Q-step guaranteed convergence