Introduction to Simulation - Lecture 21

Boundary Value Problems - Solving 3-D Finite-Difference problems

Jacob White

Thanks to Deepak Ramaswamy, Michal Rewienski, and Karen Veroy

Outline

- Reminder about FEM and F-D
  - 1-D Example
- Finite Difference Matrices in 1, 2 and 3D
  - Gaussian Elimination Costs
- Krylov Method
  - Communication Lower bound
  - Preconditioners based on improving communication

Heat Flow

1-D

Normalized 1-D Equation

Normalized Poisson Equation

$$\frac{\partial}{\partial x}\kappa\frac{\partial T(x)}{\partial x} = -h_s \Rightarrow -\frac{\partial^2 u(x)}{\partial x^2} = f(x)$$

$$-u_{xx}(x) = f(x)$$

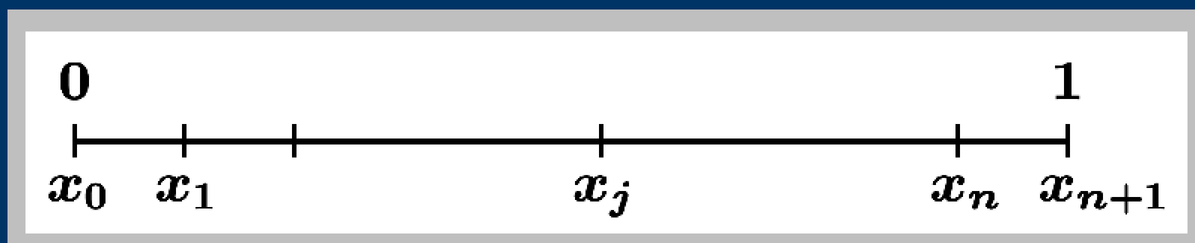
Numerical Solution

Finite Differences

Discretization

Subdivide interval $(0, 1)$ into $n + 1$ equal subintervals

$$\Delta x = \frac{1}{n+1}$$



$$x_j = j\Delta x, \qquad \hat{u}_j \approx u_j \equiv u(x_j)$$

for $\quad 0 \leq j \leq n+1$

Numerical Solution

Finite Differences

Approximation

For example . . .

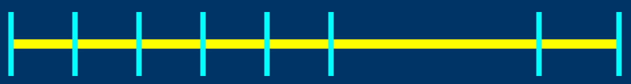
$$
\begin{aligned}
v''(x_j) &\approx \frac{1}{\Delta x}\left(v'(x_{j+1/2}) - v'(x_{j-1/2})\right) \\
&\approx \frac{1}{\Delta x}\left(\frac{v_{j+1} - v_j}{\Delta x} - \frac{v_j - v_{j-1}}{\Delta x}\right) \\
&= \frac{v_{j+1} - 2v_j + v_{j-1}}{\Delta x^2}
\end{aligned}
$$

for Δx small

FD Matrix properties

1-D Poisson $-u_{xx} = f$

Finite Differences



$$-\frac{\hat{u}_{j+1} - 2\hat{u}_j + \hat{u}_j}{\Delta x^2} = f(x_j)$$

$$\underbrace{\frac{1}{\Delta x^2}\begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix}}_{A} \begin{bmatrix} \hat{u}_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \hat{u}_n \end{bmatrix} = \begin{bmatrix} f(x_1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f(x_n) \end{bmatrix}$$

Using Basis Functions

Residual Equation

Partial Differential Equation form

$$-\frac{\partial^2 u}{\partial x^2} = f \qquad u(0) = 0 \quad u(1) = 0$$

Basis Function Representation

$$u(x) \approx u_h(x) = \sum_{i=1}^{n} \omega_i \underbrace{\varphi_i(x)}_{\text{Basis Functions}}$$

Plug Basis Function Representation into the Equation

$$R(x) = \sum_{i=1}^{n} \omega_i \frac{d^2 \varphi_i(x)}{dx^2} + f(x)$$

Using Basis functions

Basis Weights

Galerkin Scheme

Force the residual to be "orthogonal" to the basis functions

$$\int_0^1 \varphi_l(x) R(x)\, dt = 0$$

Generates n equations in n unknowns

$$\int_0^1 \varphi_l(x) \left[\sum_{i=1}^{n} \omega_i \frac{d^2 \varphi_i(x)}{dx^2} + f(x) \right] dx = 0 \quad l \in \{1, \ldots, n\}$$

Using Basis Functions

Basis Weights

Galerkin with integration by parts

Only first derivatives of basis functions

$$\int_0^1 \frac{d\varphi_l(x)}{dx} \frac{d\sum_{i=1}^{n} \omega_i \varphi_i(x)}{dx} dx - \int_0^1 \varphi_l(x) f(x) dx = 0$$

$$l \in \{1, \ldots, n\}$$

Structural Analysis of Automobiles

- Equations
  - Force-displacement relationships for mechanical elements (plates, beams, shells) and sum of forces = 0.
  - Partial Differential Equations of Continuum Mechanics

Drag Force Analysis of Aircraft

- Equations
  - Navier-Stokes Partial Differential Equations.

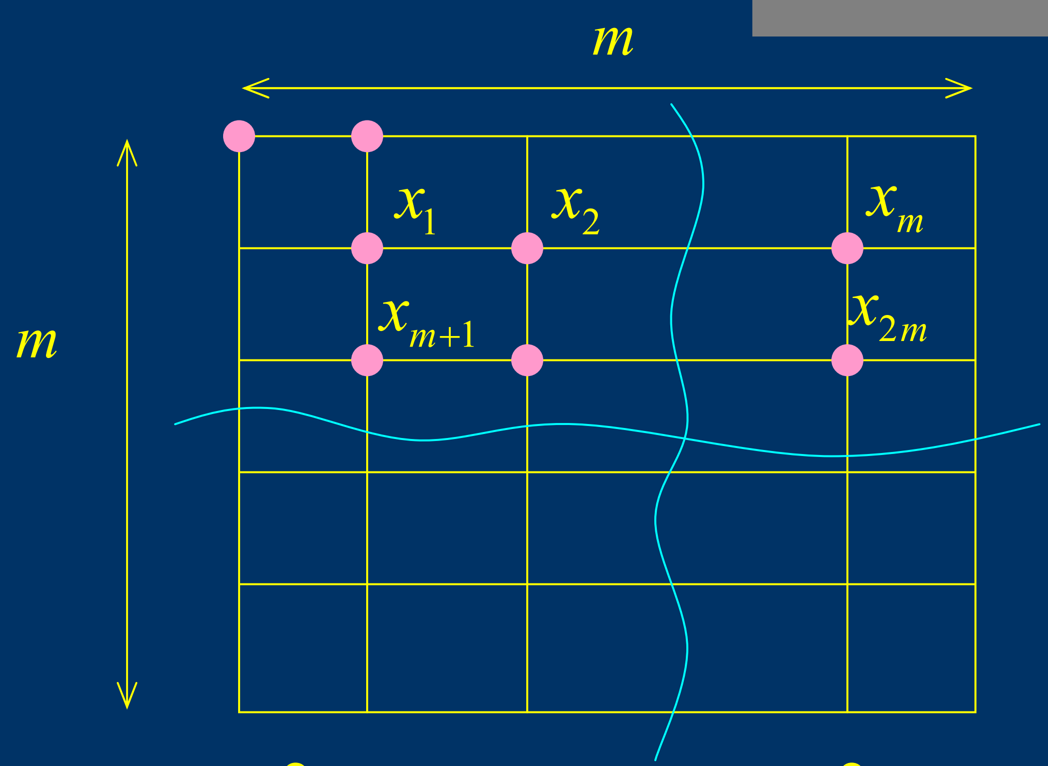
Engine Thermal Analysis

- Equations
  - The Poisson Partial Differential Equation.

FD Matrix properties

2-D Discretized Problem

Discretized Poisson

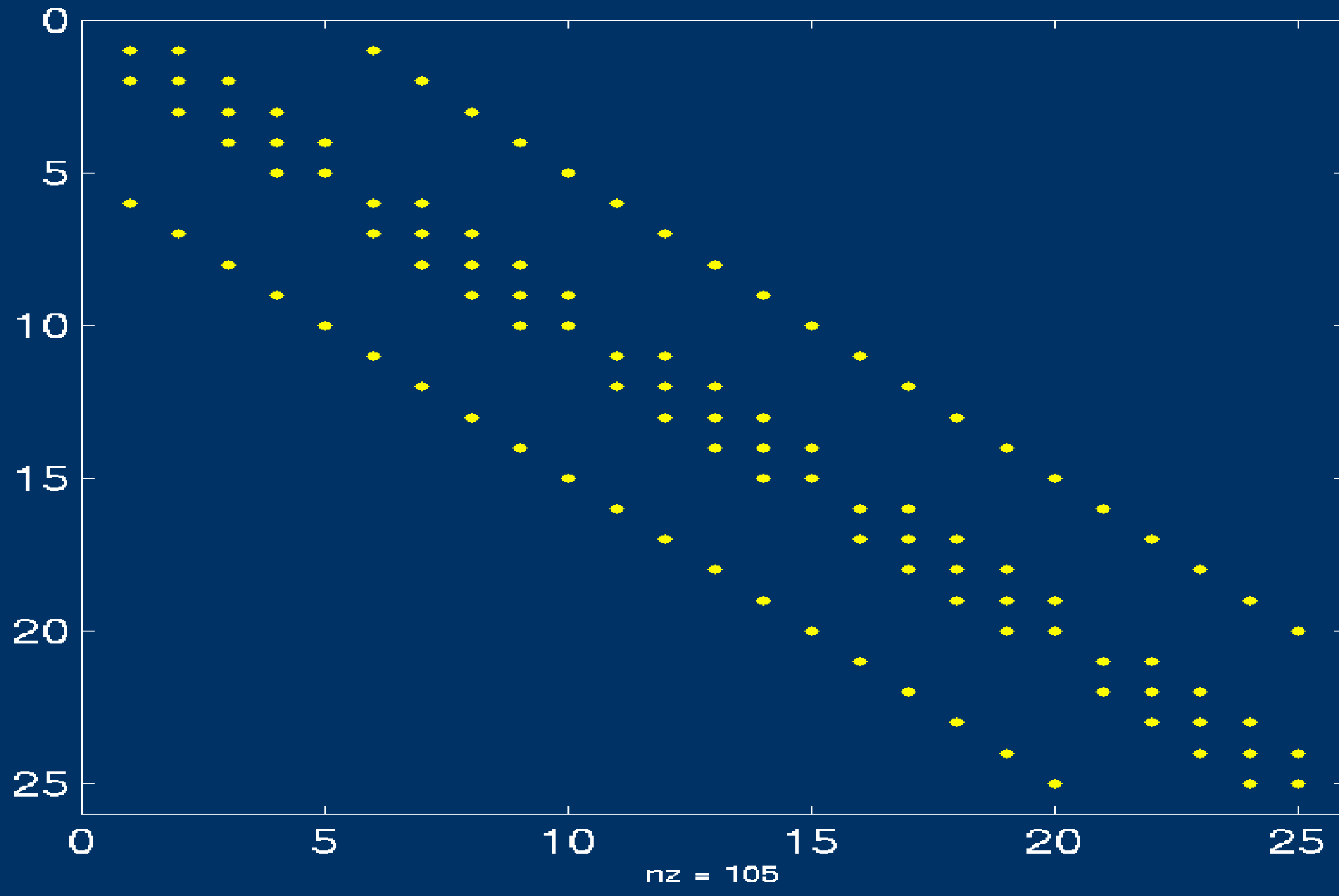


$$\underbrace{\frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2}}_{u_{xx}} + \underbrace{\frac{u_{j+m} - 2u_j + u_{j-m}}{\Delta y^2}}_{u_{yy}} = f(x_j)$$

FD Matrix properties

2-D Discretized Problem

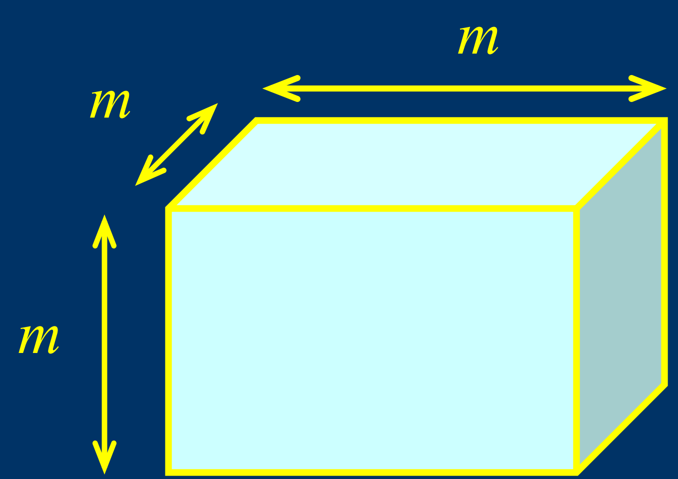
Matrix Nonzeros, 5x5 example

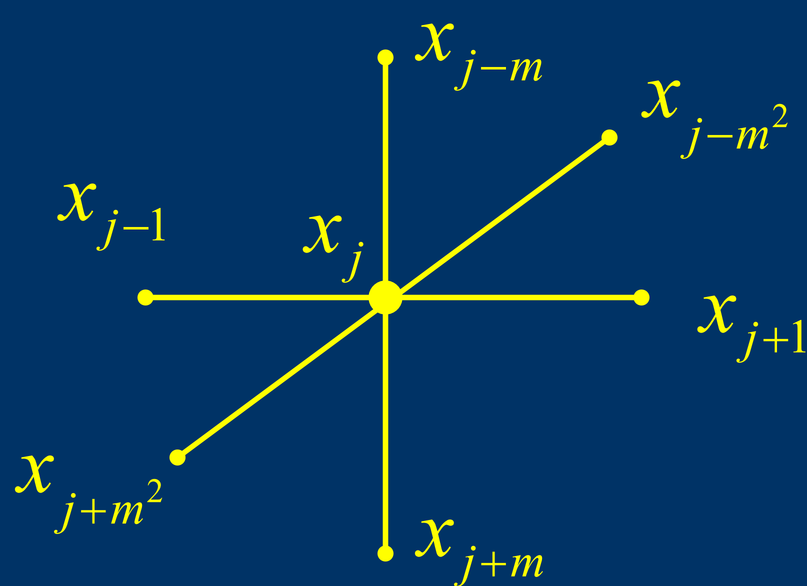


FD Matrix properties

3-D Discretization

Discretized Poisson



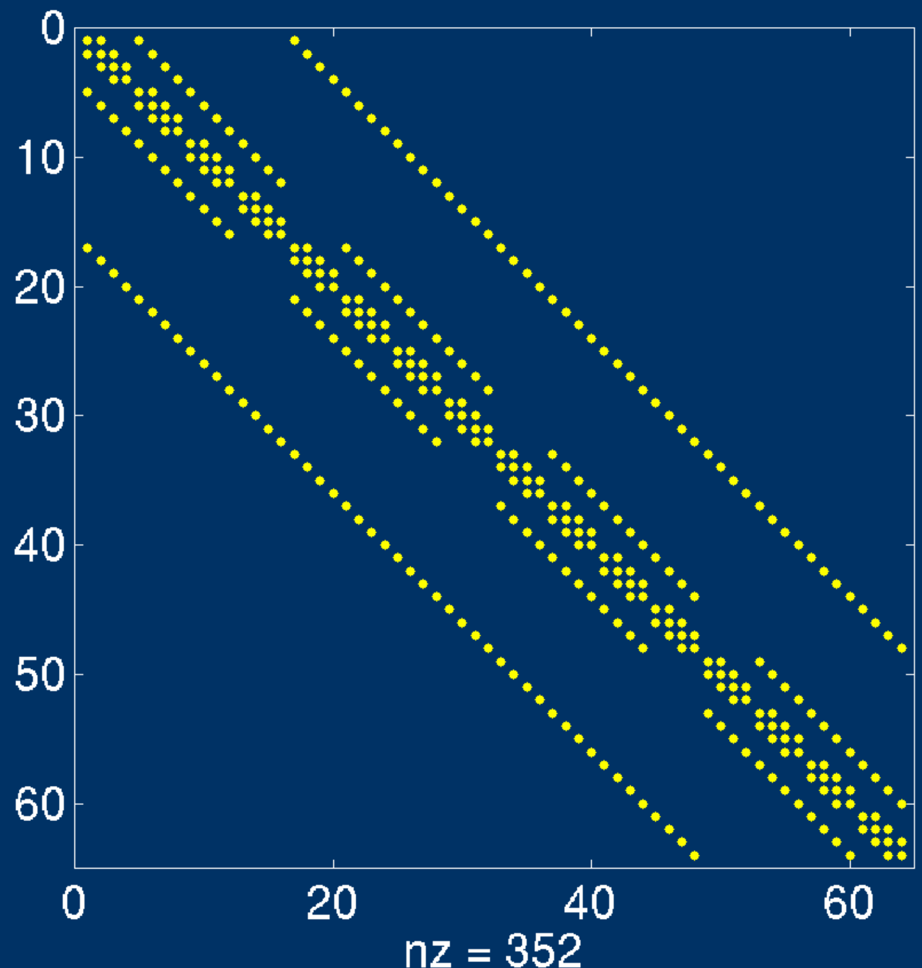


$$\underbrace{\frac{\hat{u}_{j+1} - 2\hat{u}_j + \hat{u}_{j-1}}{(\Delta x)^2}}_{u_{xx}} + \underbrace{\frac{\hat{u}_{j+m} - 2\hat{u}_j + \hat{u}_{j-m}}{(\Delta y)^2}}_{u_{yy}} + \underbrace{\frac{\hat{u}_{j+m^2} - 2\hat{u}_j + \hat{u}_{j-m^2}}{(\Delta z)^2}}_{u_{zz}} = f(x_j)$$

FD Matrix properties

3-D Discretization

Matrix nonzeros, m = 4 example



FD Matrix properties

Summary

Numerical Properties

Matrix is Irreducibly Diagonally Dominant

$$\left(|A_{ii}| \geq \sum_{j \neq i} |A_{ij}| \right)$$

Each row is strictly diagonally dominant, or path connected to a strictly diagonally dominant row

Matrix is symmetric positive definite

Assuming uniform discretization, diagonal is

$$1-D \rightarrow \frac{1}{\Delta^2}2, \quad 2-D \rightarrow \frac{1}{\Delta^2}4, \quad 3-D \rightarrow \frac{1}{\Delta^2}6,$$

FD Matrix properties

Summary

Structural Properties

Matrices in 3-D are LARGE

$$1-D \rightarrow m \times m, \quad 2-D \rightarrow m^2 \times m^2, \quad 3-D \rightarrow m^3 \times m^3$$

100x100x100 grid in 3-D = 1 million x 1 million matrix

Matrices are very sparse

Nonzeros per row $1-D \;\square\; 3, \quad 2-D \;\square\; 5, \quad 3-D \;\square\; 7$

Matrices are banded

$$1-D \qquad A_{ij} = 0 \qquad |i-j| > 1$$

$$2-D \qquad A_{ij} = 0 \qquad |i-j| > m$$

$$3-D \qquad A_{ij} = 0 \qquad |i-j| > m^2$$

Basics of GE

Triangularizing

Picture

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ 0 & A_{22} & A_{23} & A_{24} \\ 0 & 0 & A_{33} & A_{34} \\ 0 & 0 & 0 & A_{44} \end{bmatrix}$$

GE Basics

Triangularizing

Algorithm

For i = 1 to n-1 { "For each Row"

For j = i+1 to n { "For each Row below pivot"

For k = i+1 to n { "For each element beyond Pivot"

$$A_{jk} \Leftarrow A_{jk} - \frac{A_{ji}}{A_{ii}} A_{ik}$$

Multiplier

Pivot

}

}

}

Form n-1 reciprocals (pivots)

Form $\sum_{i=1}^{n-1}(n-i) = \frac{n^2}{2}$ multipliers

Perform $\sum_{i=1}^{n-1}(n-i)^2 \approx \frac{2}{3}n^3$ Multiply-adds

Complexity of GE

$\underline{1-}$D $\quad O(n^3) = O(m^3) \quad \leftarrow 100$ pt grid $O(10^6)$ ops

$\underline{2-}$D $\quad O(n^3) = O(m^6) \quad \leftarrow 100\times100$ grid $O(10^{12})$ ops

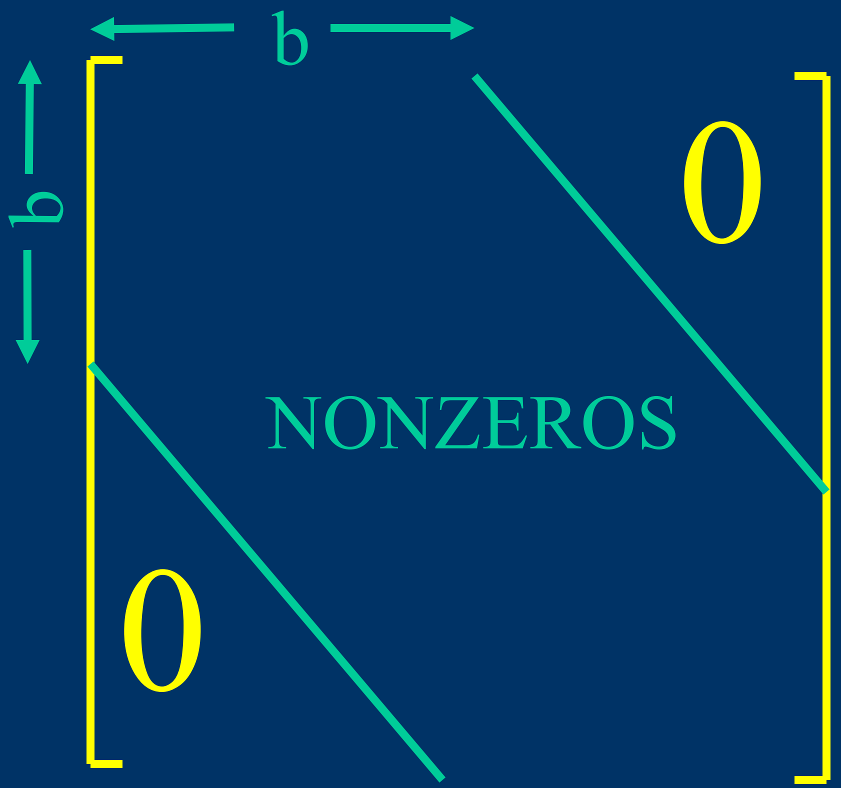
$\underline{3-}$D $\quad O(n^3) = O(m^9) \quad \leftarrow 100\times100\times100$ grid $O(10^{18})$ ops

<u>For 2- D and 3-D problems</u> Need a Faster Solver !

Banded GE

Triangularizing

Algorithm



For i = 1 to n-1 {

  For j = i+1 to n i+b-1 {

    For k = i+1 to n i+b-1 {

$$A_{jk} \Leftarrow A_{jk} - \frac{A_{ji}}{A_{ii}} A_{ik}$$

    }

  }

}

Perform

$$\sum_{i=1}^{n-1} (\min(b-1, n-i))^2 \approx O\left(b^2 n\right)$$

Multiply-adds

Complexity of Banded GE

$\underline{1-\text{D}}$ $\quad O(b^2 n) = O(m) \quad \leftarrow 100$ pt grid $O(100)$ ops

$\underline{2-\text{D}}$ $\quad O(b^2 n) = O(m^4) \quad \leftarrow 100 \times 100$ grid $O(10^8)$ ops

$\underline{3-\text{D}}$ $\quad O(b^2 n) = O(m^7) \quad \leftarrow 100 \times 100 \times 100$ grid $O(10^{14})$ ops

<u>For 3 - D problems</u> Need a Faster Solver !

The World According to Krylov

Preconditioning

Start with $Ax = b$, Form $PAx = Pb$

Determine the Krylov Subspace

$$r^0 = Pb - PAx$$

Krylov Subspace $\equiv span\left\{r^0, PAr^0, \ldots, (PA)^k r^0\right\}$

Select Solution from the Krylov Subspace

$$x^{k+1} = x^0 + y^k, \quad y^k \in span\left\{r^0, PAr^0, \ldots, (PA)^k r^0\right\}$$

GCR picks a residual minimizing y^k.

Krylov Methods

Preconditioning

Diagonal Preconditioners

$$\text{Let } \mathrm{A} = D + A_{nd}$$

Apply GCR to $\left(D^{-1}A\right)x = \left(I + D^{-1}A_{nd}\right)x = D^{-1}b$

- The Inverse of a diagonal is cheap to compute
- Usually improves convergence

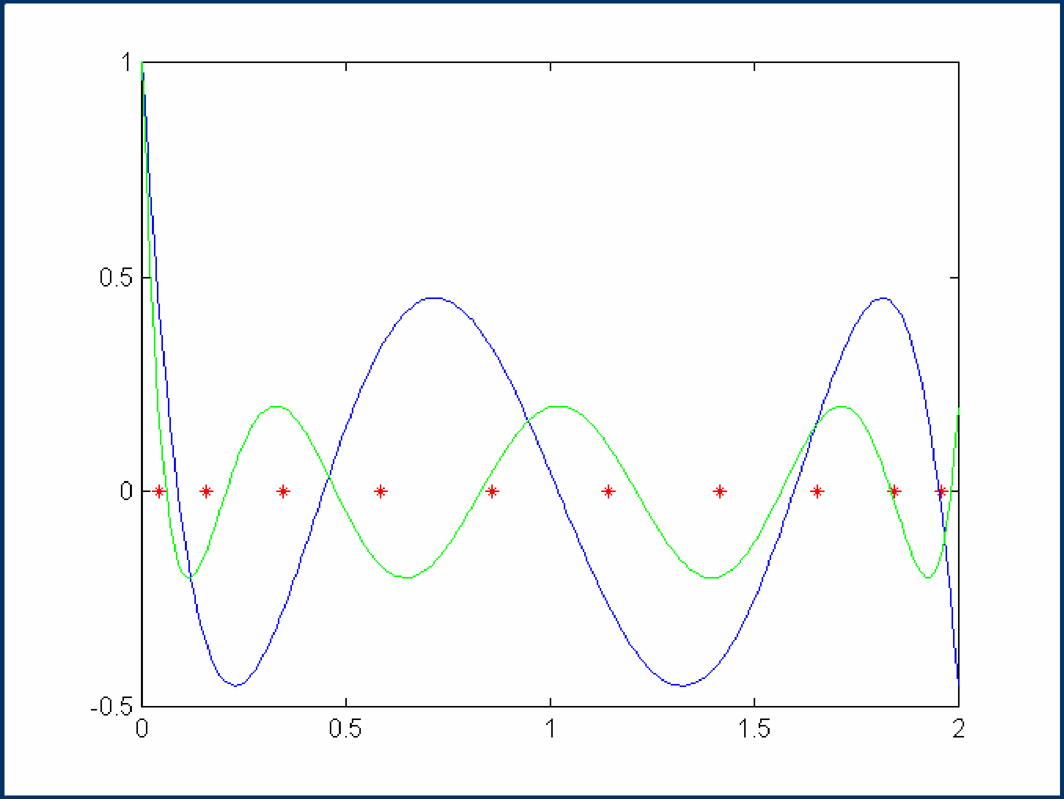
GCR Optimality Property

$\left\|r^{k+1}\right\| \leq \left\|\tilde{\wp}_{k+1}(PA)\right\| \left\|\mathrm{r}^{0}\right\|$ 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 constraints can be used to get an upper bound on $\Rightarrow \dfrac{\left\|r^{k+1}\right\|}{\left\|r^{0}\right\|}$

Residual Poly Picture for Heat Conducting Bar Matrix
No loss to air (n=10)



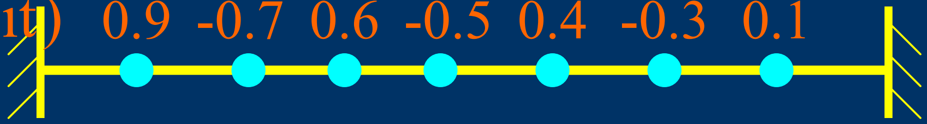
Keep $\left|\wp_k(\lambda_i)\right|$ as small as possible:

Easier if eigenvalues are clustered!

The World According to Krylov

Krylov Vectors for diagonal Preconditioned A

1-D Discretized PDE



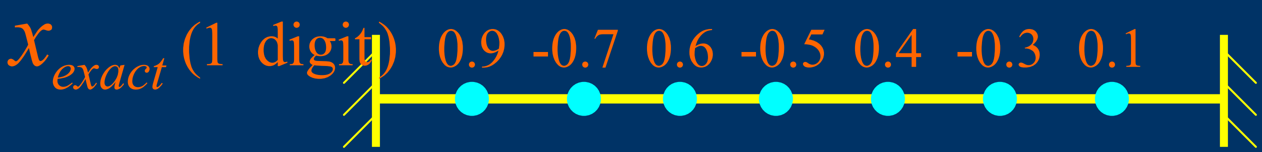
x_{exact} (1 digit)	0.9	-0.7	0.6	-0.5	0.4	-0.3	0.1
$b = r^0$	1	0	0	0	0	0	0
$D^{-1}A$	1	-.5	0	0	0	0	0
$D^{-1}A$	1.25	-1	-.5	0	0	0	0

$$\underbrace{\begin{bmatrix} 1 & -0.5 & 0 & 0 \\ -0.5 & 1 & \ddots & 0 \\ 0 & \ddots & \ddots & -0.5 \\ 0 & 0 & -0.5 & 1 \end{bmatrix}}_{D^{-1}A} \begin{bmatrix} 1 \\ -0.5 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 1.25 \\ -1 \\ -0.5 \\ 0 \end{bmatrix}$$

The World According to Krylov

Krylov Vectors for Diagonal Preconditioned A

Communication Lower Bound



$b = r^0$	1	0	0	0	0	0	0
$D^{-1}A$	1	-.5	0	0	0	0	0
$D^{-1}A$	1.25	-1	-.5	0	0	0	0

Communication Lower Bound for m gridpoints

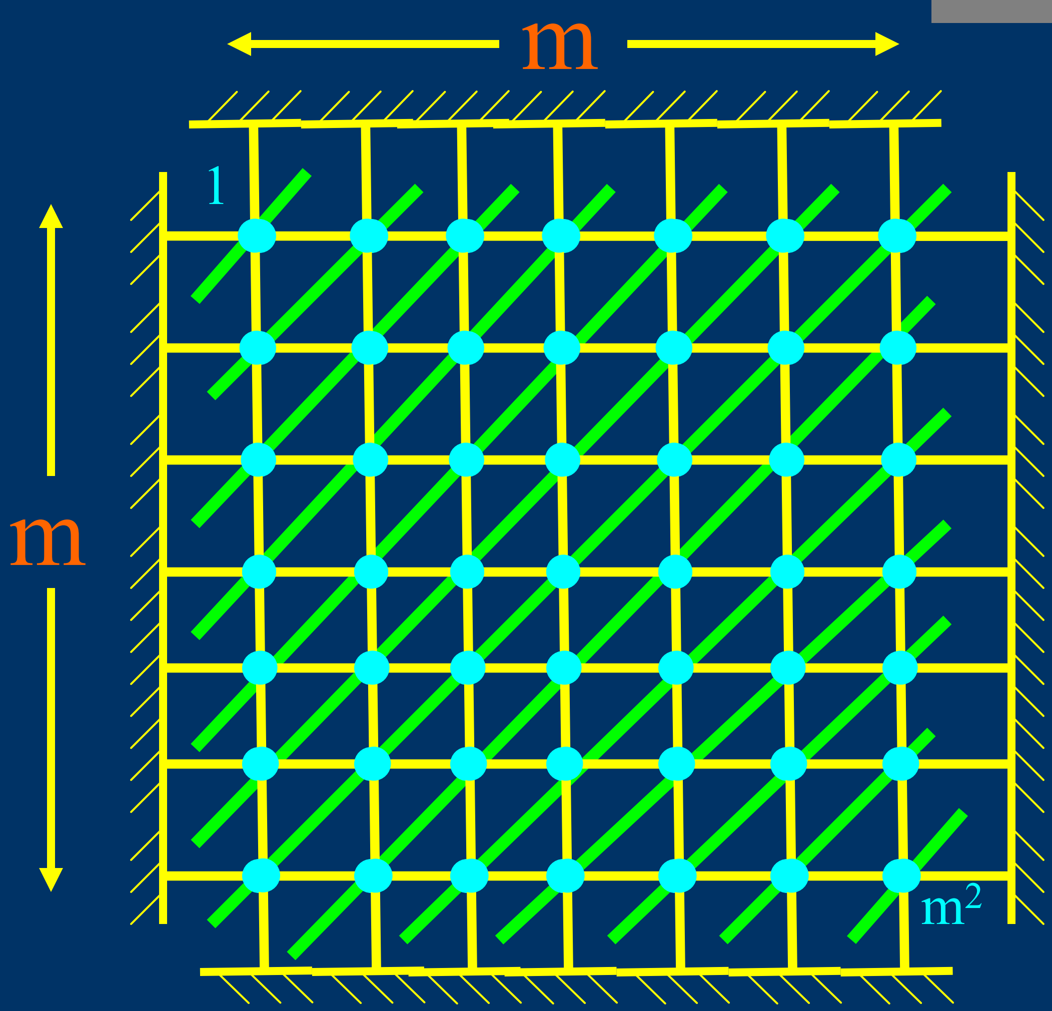
$\left(D^{-1}A\right)^k r^0$ is nonzero in m^{th} entry after $k = m$ iters

Need at least m iterations for $\left(x^{k+1}\right)_m = \left(x^0 + \sum_{j=1}^{k} \alpha_j r^j\right)_m \neq 0$

The World According to Krylov

Krylov Vectors for Diagonal Preconditioned A

Two Dimensional Case



For an mxm Grid

$$\text{If } r^0 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Takes $\approx \sqrt{2}m = O(m)$ iters

for $\left(x^{k+1}\right)_{m^2} \neq 0$

Convergence for GCR

Eigenanalysis

$$\underbrace{\begin{bmatrix} 1 & -0.5 & 0 & 0 \\ -0.5 & 1 & \ddots & 0 \\ 0 & \ddots & \ddots & -0.5 \\ 0 & 0 & -0.5 & 1 \end{bmatrix}}_{D^{-1}A} \begin{bmatrix} 1 \\ -0.5 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 1.25 \\ -1 \\ -0.5 \\ 0 \end{bmatrix}$$

Recall Eigenvalues of $D^{-1}A \; = 1 - \cos\left(\frac{k\pi}{m+1}\right)$

The World According to Krylov

Convergence for GCR

Eigenanalysis

$$\text{For } D^{-1}A,\ \kappa \equiv \frac{\lambda_{\max}}{\lambda_{\min}} = \left(1-\cos\left(\frac{m\pi}{m+1}\right)\right)\left(1-\cos\left(\frac{\pi}{m+1}\right)\right)^{-1}$$

Iters for GCR to achieve convergence

$$\frac{\left\|r^k\right\|}{\left\|r^0\right\|} \le 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \le \gamma\begin{pmatrix} convergence \\ tolerance \end{pmatrix}$$

$$k = \frac{\log\frac{\gamma}{2}}{\log\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)} \underset{m\to\infty}{\cong} O(m)$$

GCR achieves Communication lower bound O(m)!

The World According to Krylov

Work for Banded Gaussian Elimination, Relaxation and GCR

Dimension	Banded GE	Sparse GE	GCR
1	$O(m)$	$O(m)$	$O(m^2)$
2	$O(m^4)$	$O(m^3)$	$O(m^3)$
3	$O(m^7)$	$O(m^6)$	$O(m^4)$

GCR faster than banded GE in 2 and 3 dimensions

Could be faster, 3-D matrix only m^3 nonzeros.

Must defeat the communication lower bound!

Preconditioning is the Only Hope

GCR already achieves Communication Lower bound for a diagonally preconditioned A

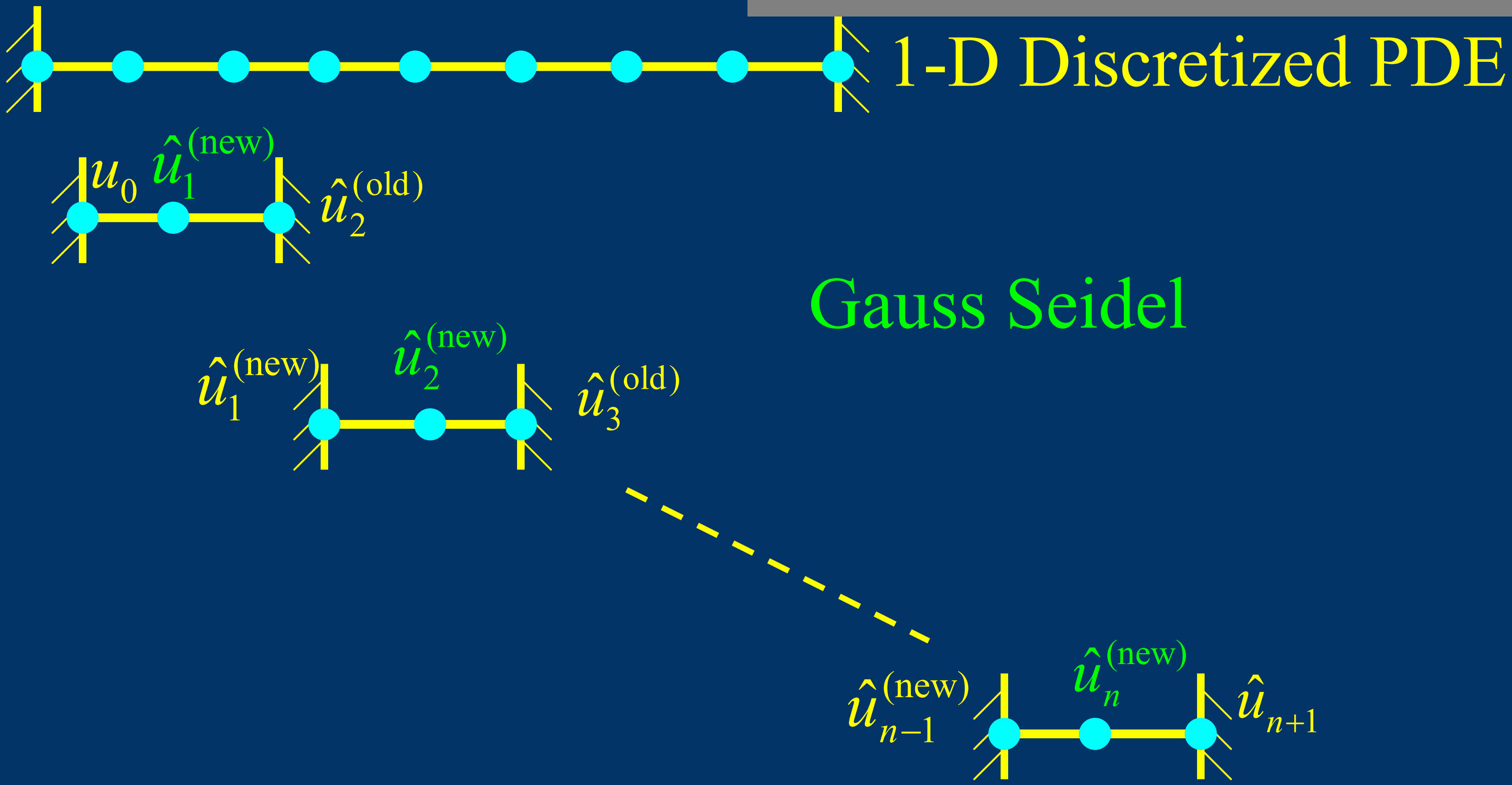
Preconditioner must accelerate communication

Multiplying by PA must move values by more than one grid point.

Preconditioning Approaches

Gauss-Seidel Preconditioning

Physical View



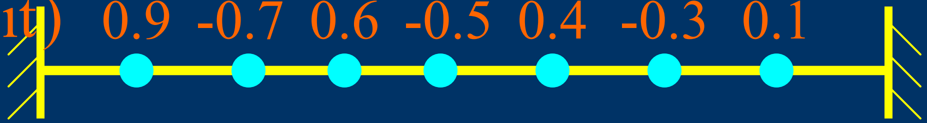
Each Iteration of Gauss-Seidel Relaxation moves data across the grid

Preconditioning Approaches

Gauss-Seidel Preconditioning

Krylov Vectors

x_{exact} (1 digit) 0.9 -0.7 0.6 -0.5 0.4 -0.3 0.1



1-D Discretized PDE

$b = r^0$	1	0	0	0	0	0	0
$(D+L)^{-1}A$	x	x	x	x	x	x	x
$(D+L)^{-1}A$	x	x	x	x	x	x	x

X=nonzero

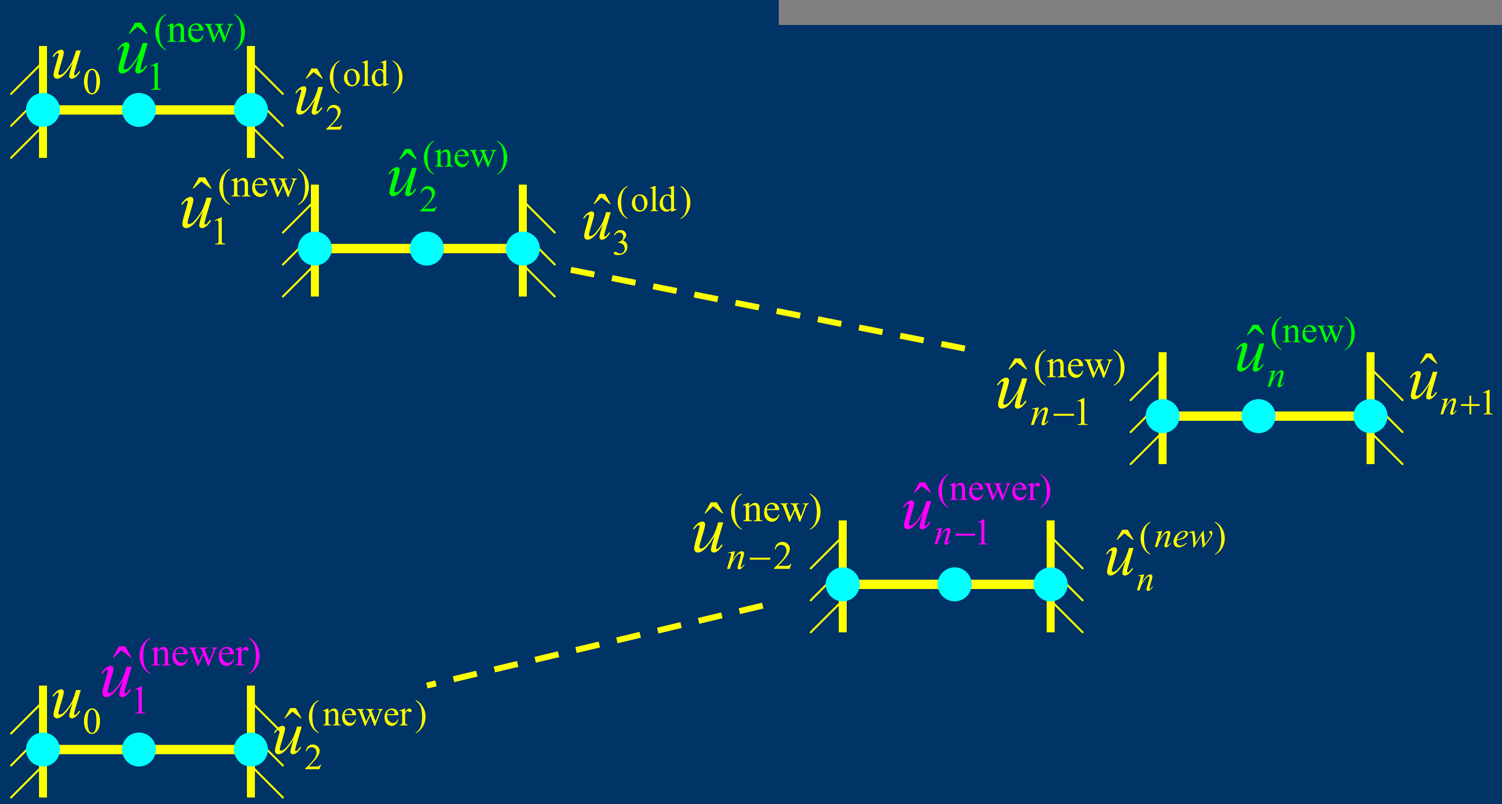
$b = r^0$	0	0	0	0	0	0	1
$(D+L)^{-1}A$	0	0	0	0	0	x	x
$(D+L)^{-1}A$	0	0	0	0	x	x	x

Gauss-Seidel communicates quickly in <u>only</u> one direction

Preconditioning Approaches

Gauss-Seidel Preconditioning

Symmetric Gauss-Seidel



This symmetric Gauss-Seidel Preconditioner Communicates both directions

Derivation of the SGS Iteration Equations

Forward Sweep (half step): $(D+L)x^{k+1/2} + Ux^k = b$

Backward Sweep (half step): $(D+U)x^k + Lx^{k+1/2} = b$

$$\Rightarrow \quad x^{k+1} = (D+U)^{-1} L (D+L)^{-1} Ux^k + (D+U)^{-1} b - (D+U)^{-1} L (D+L)^{-1} b$$

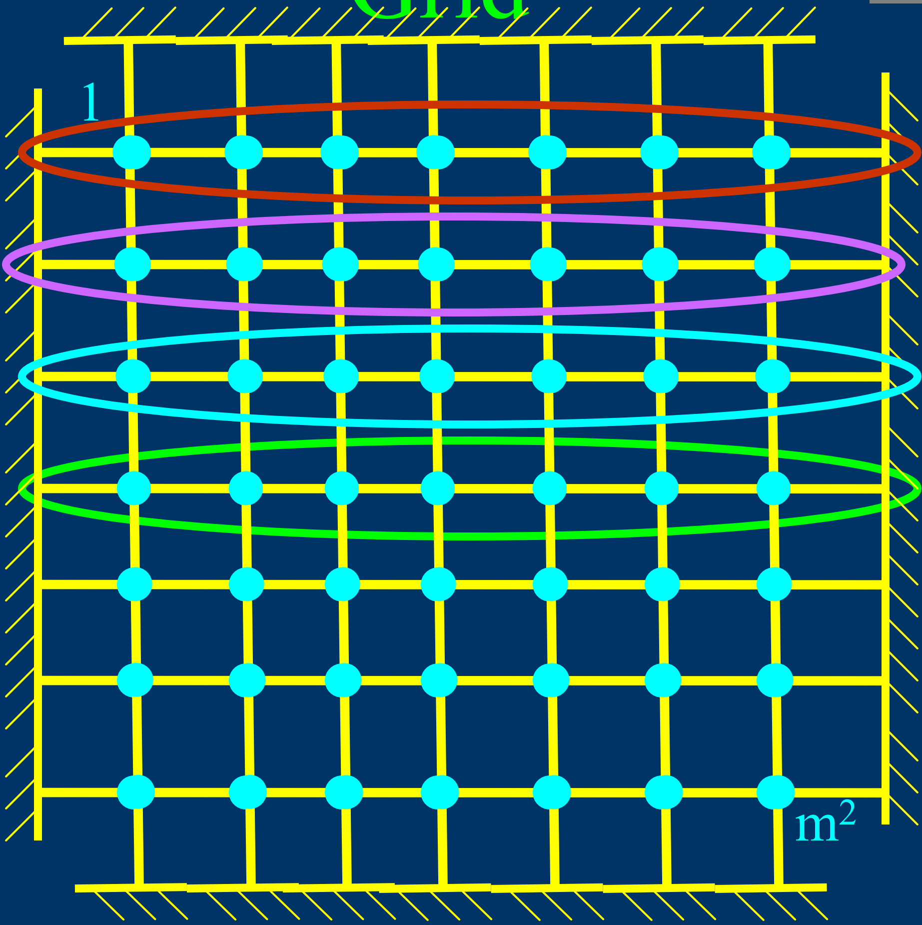
$$\Rightarrow \quad x^{k+1} = x^k - (D+U)^{-1} D (D+L)^{-1} Ax^k + (D+U)^{-1} D (D+L)^{-1} b$$

Preconditioning Approaches

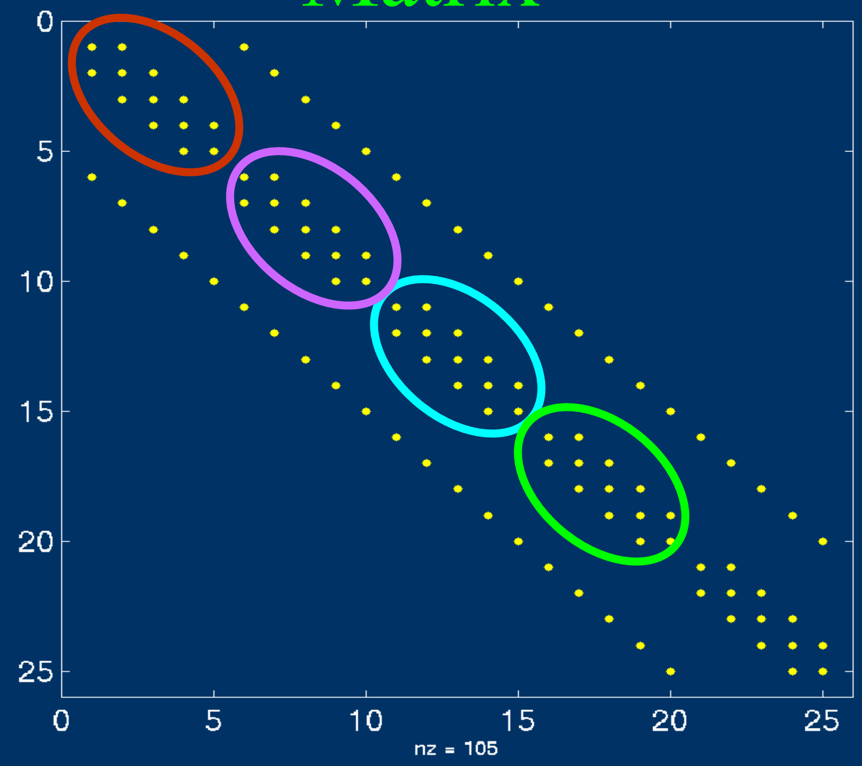
Block Diagonal Preconditioners

Line Schemes

Grid



Matrix

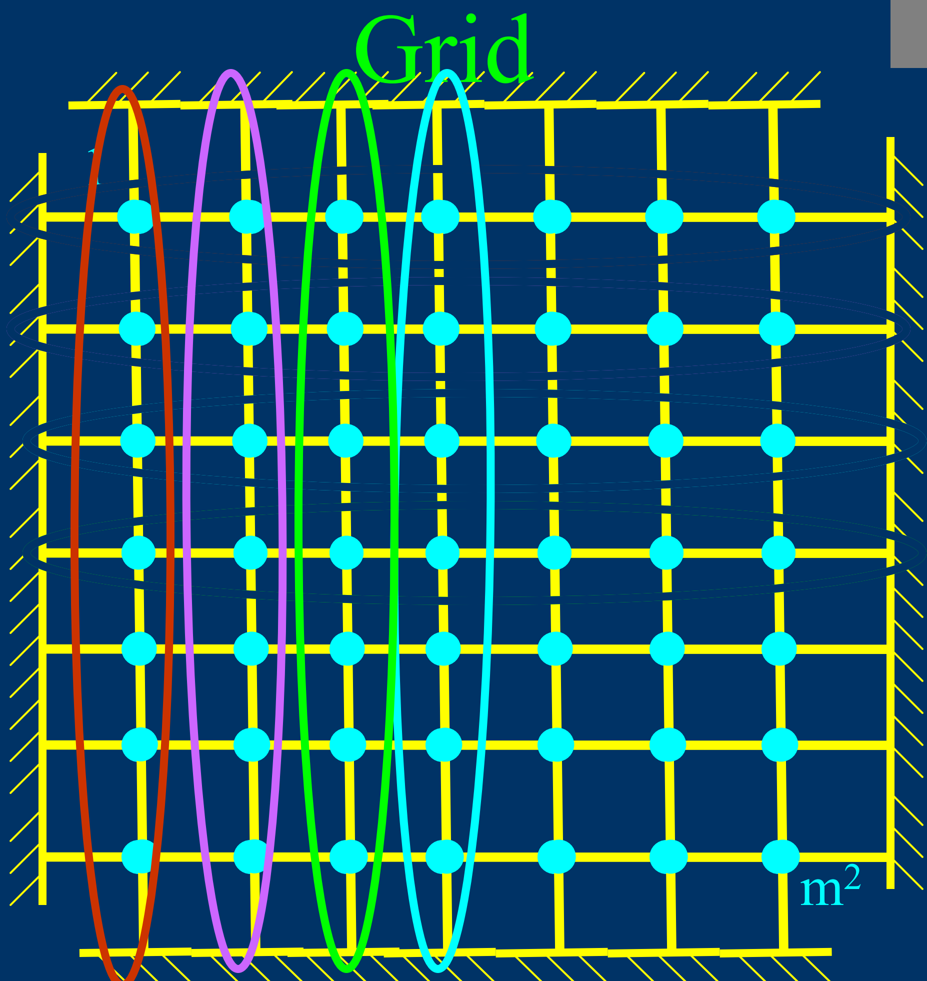


Tridiagonal Matrices factor quickly

Preconditioning Approaches

Block Diagonal Preconditioners

Line Schemes



Problem

Lines preconditioners communicate rapidly in only one direction

Solution

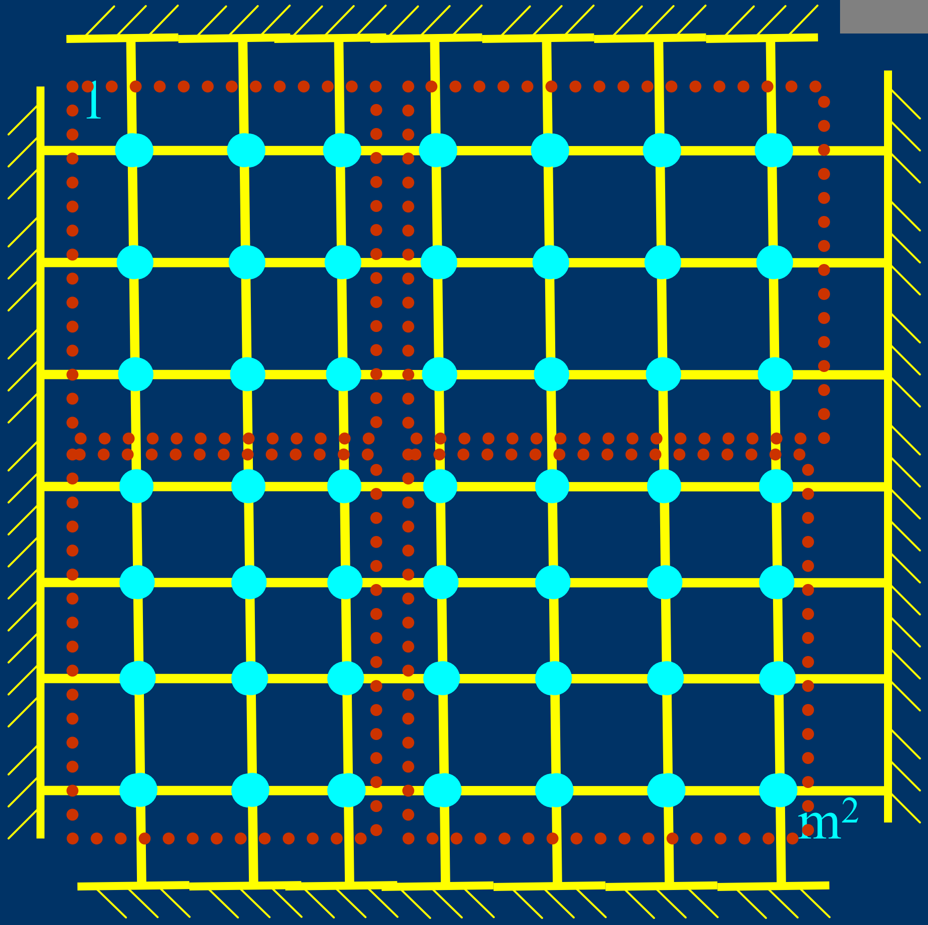
Do lines first in x, then in y.

The Preconditioner is now two Tridiagonal solves, with variable reordering in between.

Preconditioning Approaches

Block Diagonal Preconditioners

Domain Decomposition



Approach

Break the domain into small blocks each with the same # of grid points

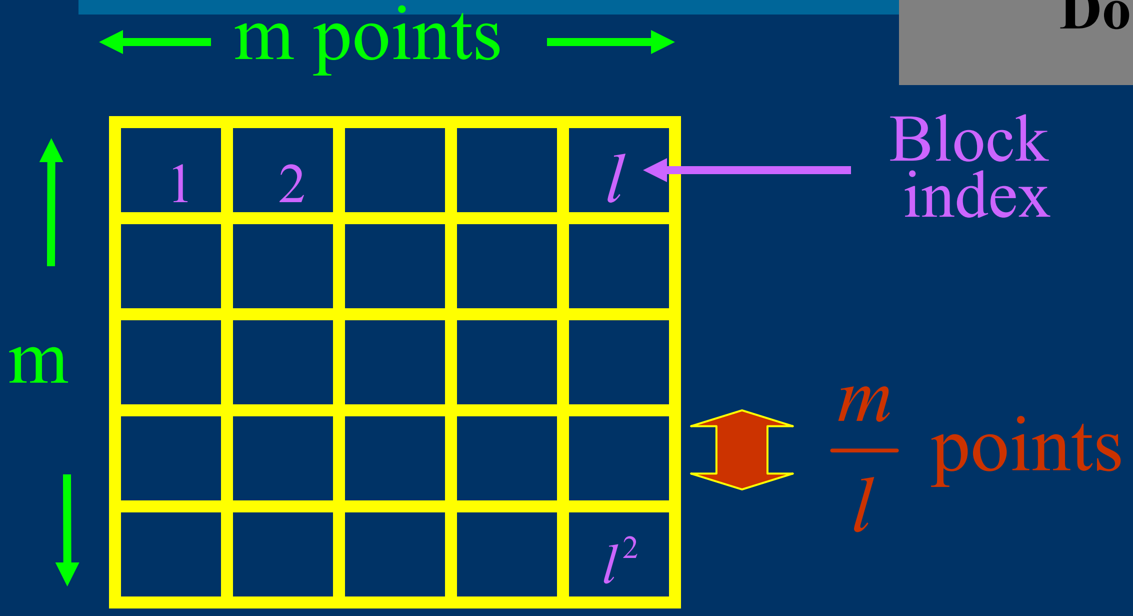
The trade-off

Fewer blocks means faster convergence, but more costly iterates

Preconditioning Approaches

Block Diagonal Preconditioners

Domain Decomposition



Block cost: factoring $\left(\frac{m}{l}\right)^2$ $l \times l$ grids, $O\left(m^2 l\right)$, sparse GE

GCR iters: Communication bound gives $O\left(\frac{m}{l}\right)$ iterations.

Suggests insensitivity to l: Algorithm is $O\left(m^3\right)$.

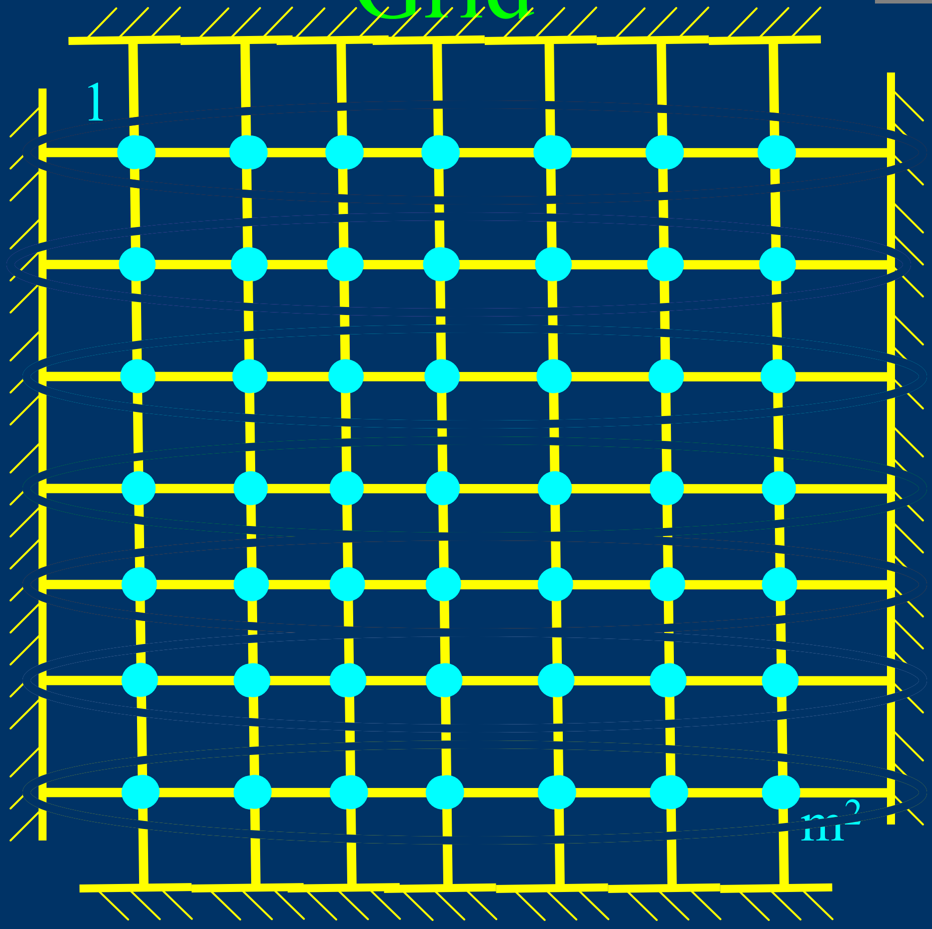
Do you have to refactor every GCR iteration?

Preconditioning Approaches

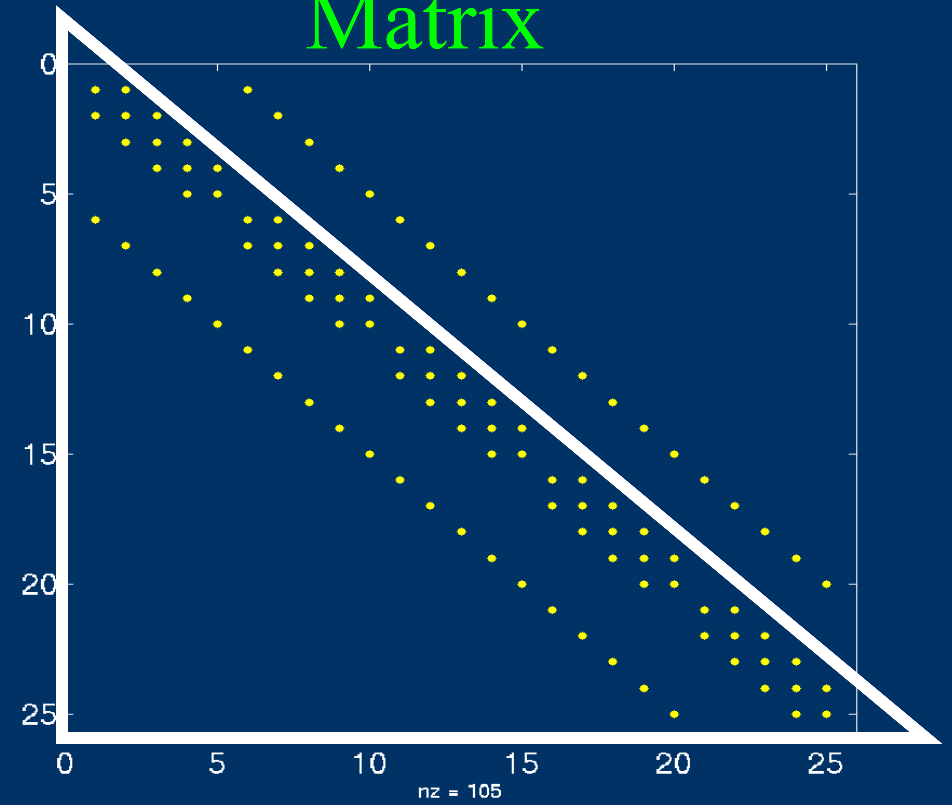
Siedelerized Block Diagonal Preconditioners

Line Schemes

Grid



Matrix

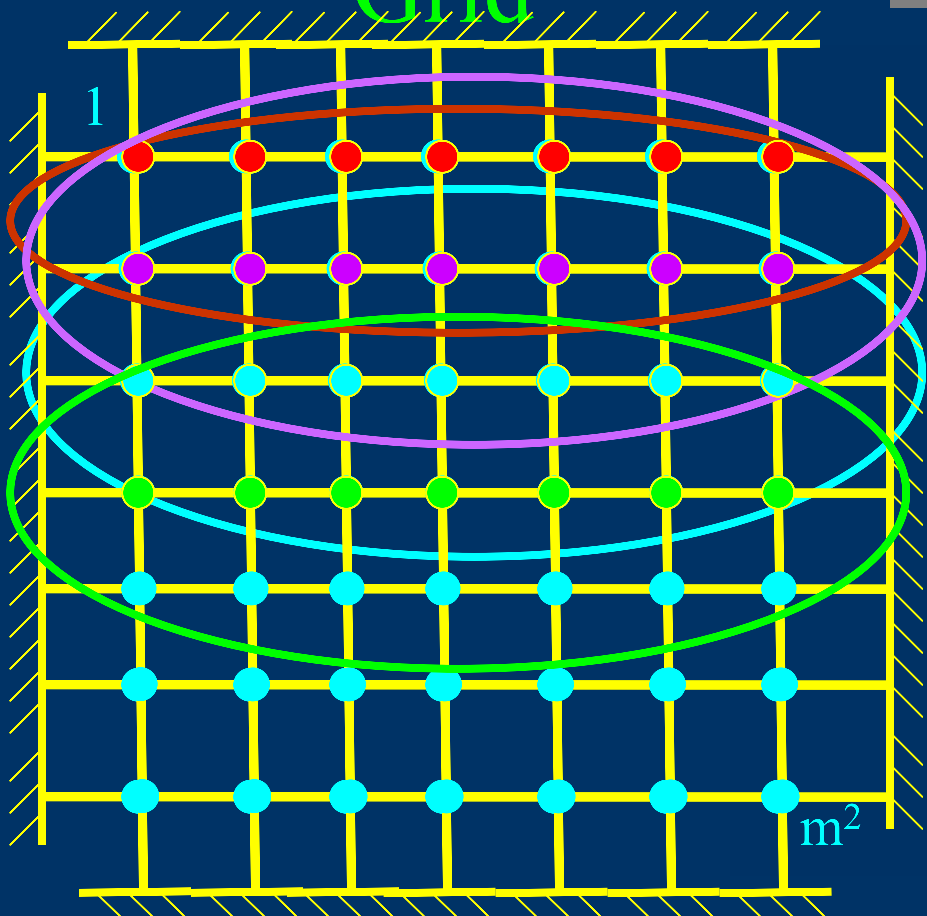


Preconditioning Approaches

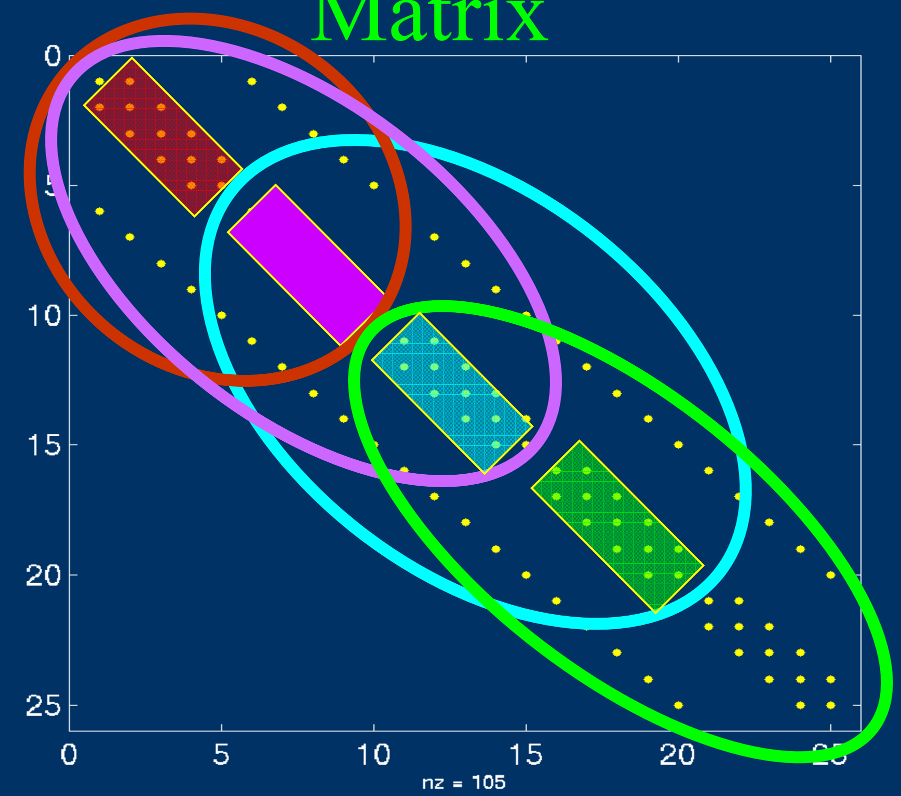
Overlapping Domain Preconditioners

Line based Schemes

Grid



Matrix



Bigger systems to solve, but can have faster convergence on tougher problems (not just Poisson).

Preconditioning Approaches

Incomplete Factorization Schemes

Outline

Reminder about Gaussian Elimination

- Computational Steps

Fill-in for Sparse Matrices

- Greatly increases factorization cost

Fill-in in a 2-D grid

Incomplete Factorization Idea

Sparse Matrices

Fill-In

Example

Matrix Non zero structure

$$\begin{bmatrix} X & X & X \\ X & X & 0 \\ X & 0 & X \end{bmatrix}$$

X= Non zero

Matrix after one GE step

$$\begin{bmatrix} X & X & X \\ X & X & X \\ X & X & X \end{bmatrix}$$

Fill-ins

Sparse Matrices

Fill-In

Second Example

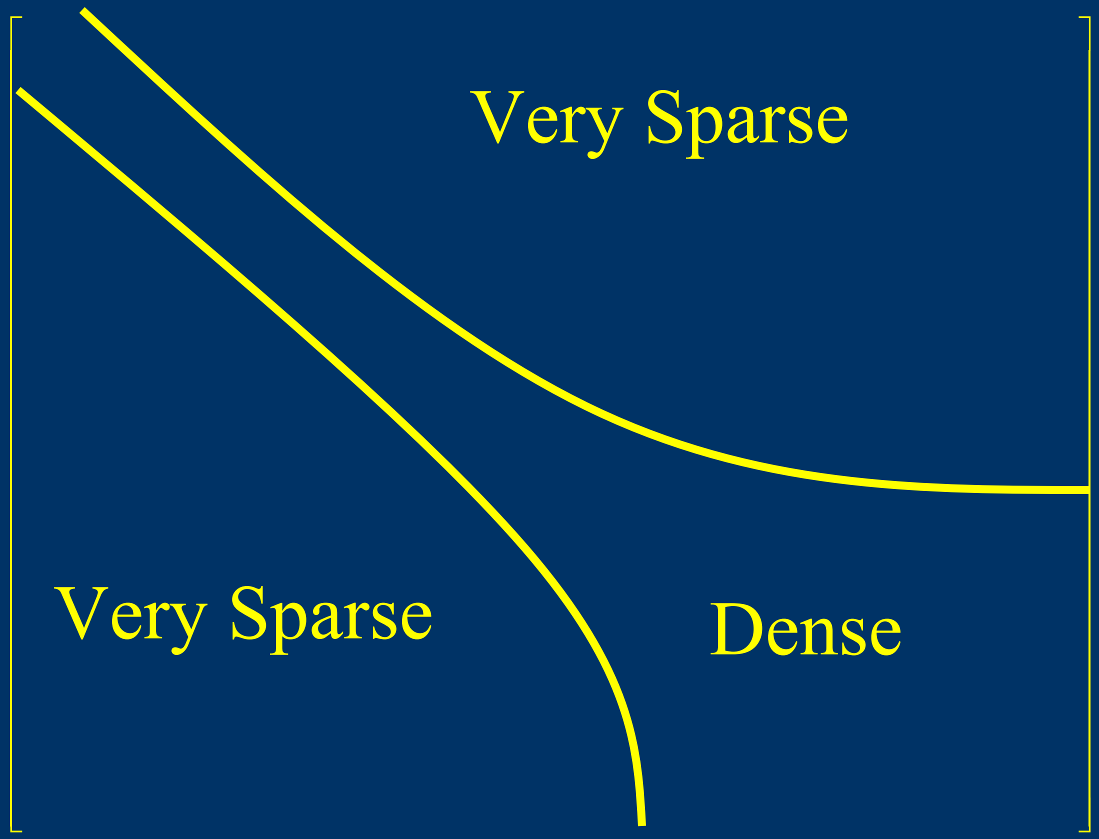
Fill-ins Propagate

$$\begin{bmatrix} X & X & X & X \\ X & X & X & X \\ 0 & X & X & X \\ 0 & X & X & X \end{bmatrix}$$

Fill-ins from Step 1 result in Fill-ins in step 2

Fill-In

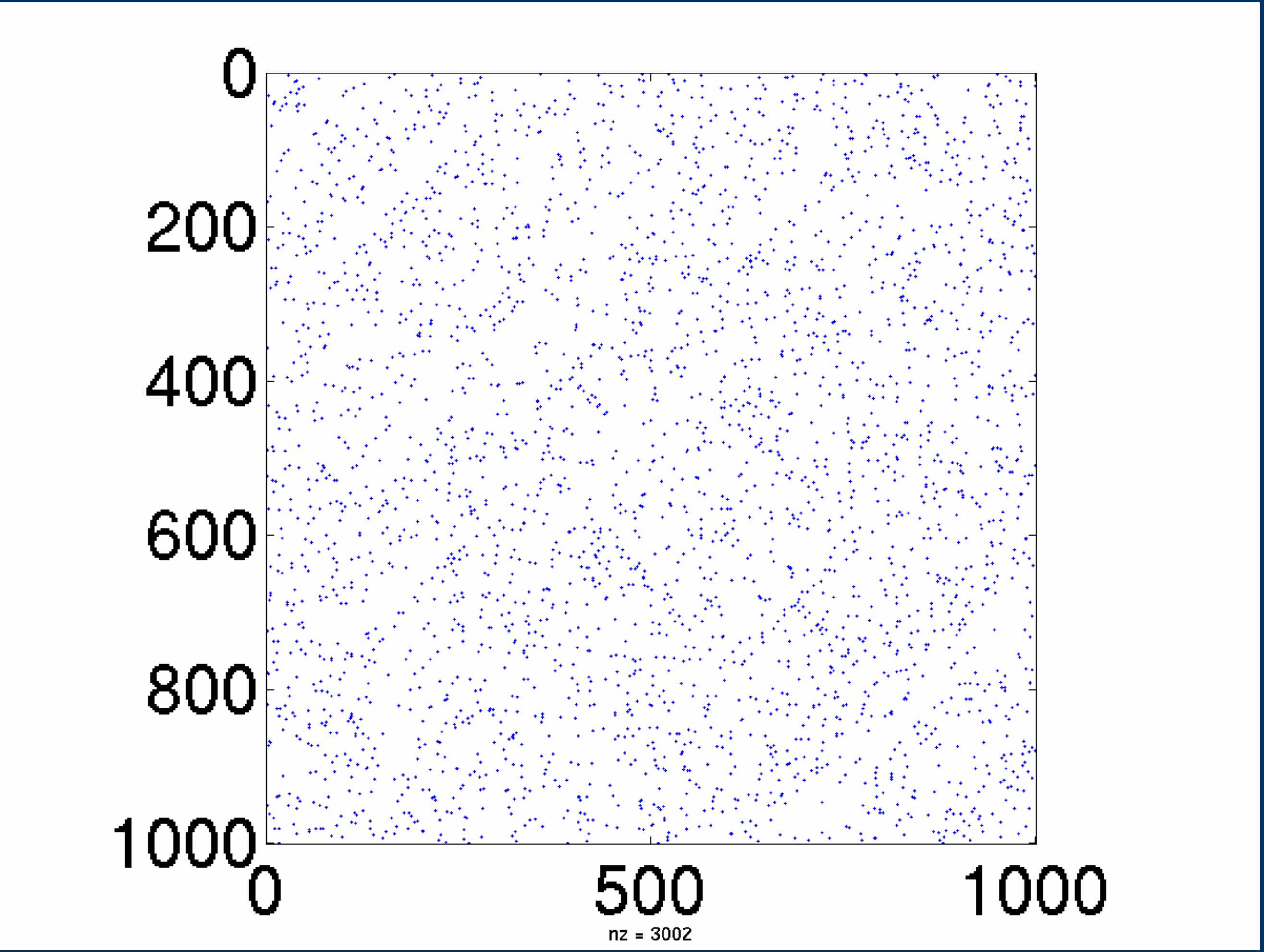
Pattern of a Filled-in Matrix



Sparse Matrices

Fill-In

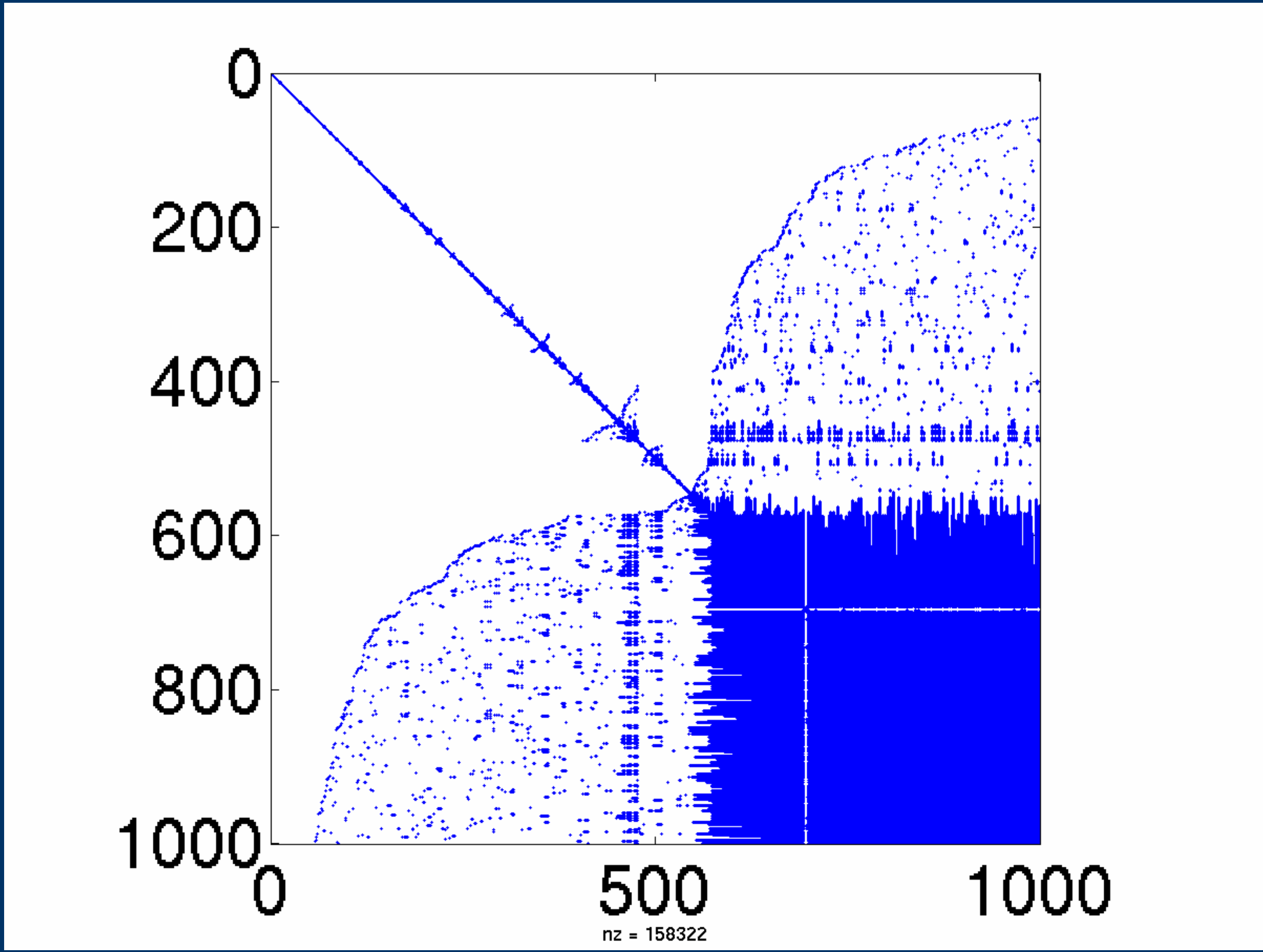
Unfactored Random Matrix



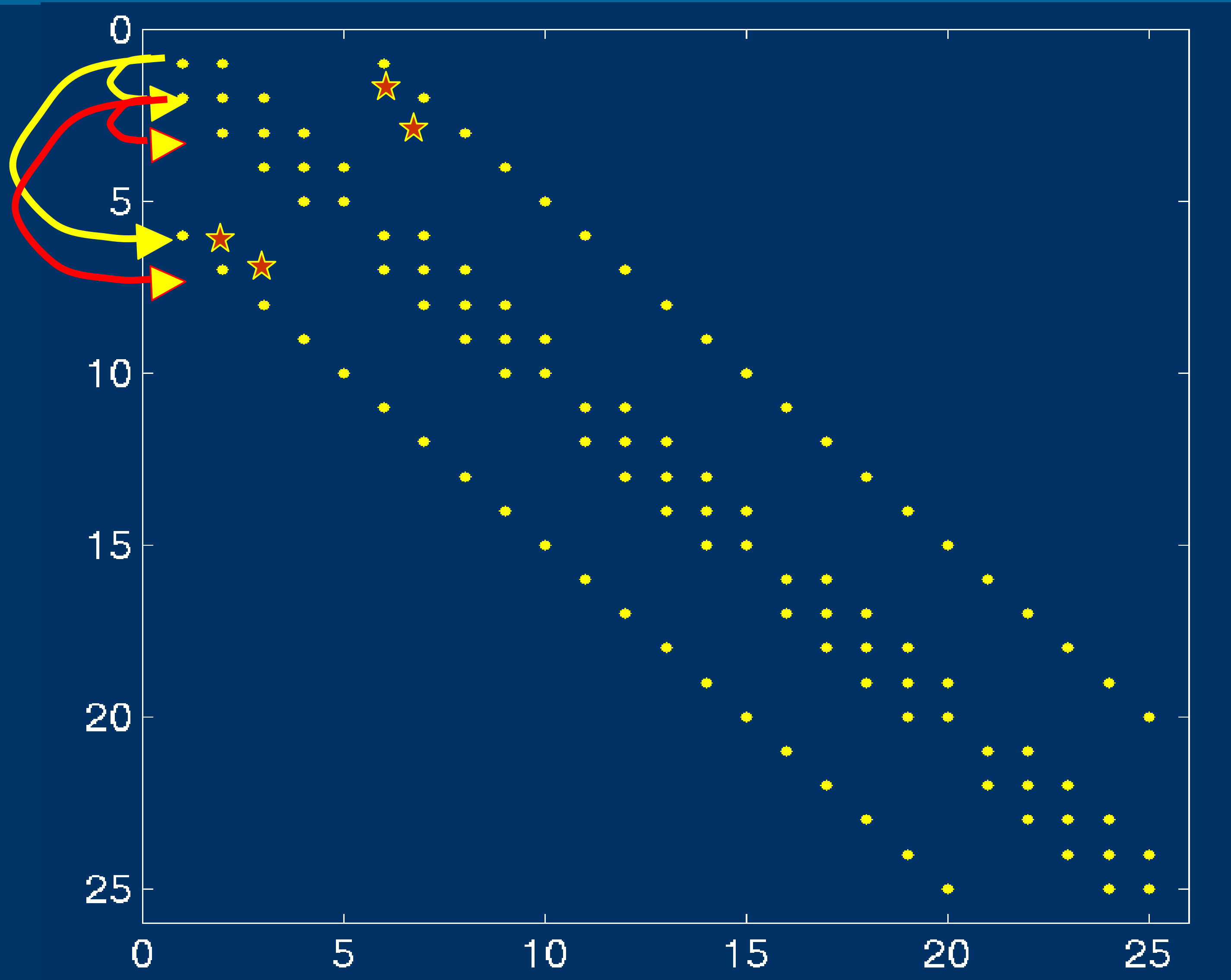
Sparse Matrices

Fill-In

Factored Random Matrix



Factoring 2-D Finite-Difference matrices

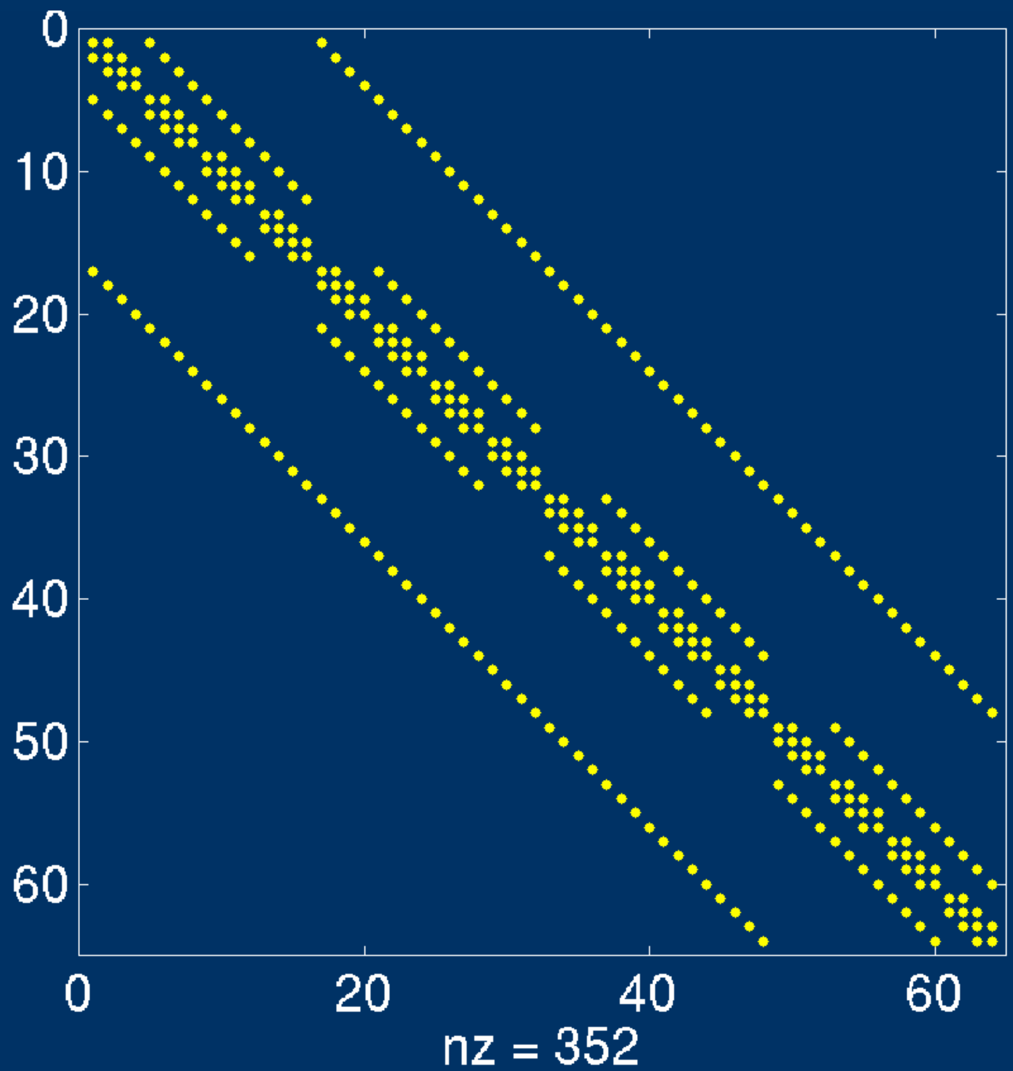


Generated Fill-in Makes Factorization Expensive

FD Matrix properties

3-D Discretization

Matrix nonzeros, m = 4 example



Preconditioning Approaches

Incomplete Factorization Schemes

Key idea

THROW AWAY FILL-INS!

Throw away all fill-ins

Throw away only fill-ins with small values

Throw away fill-ins produced by other fill-ins

Throw away fill-ins produced by fill-ins of other fill-ins, etc.

Summary

- 3-D BVP Examples
  - Aerodynamics, Continuum Mechanics, Heat-Flow
- Finite Difference Matrices in 1, 2 and 3D
  - Gaussian Elimination Costs
- Krylov Method
  - Communication Lower bound
  - Preconditioners based on improving communication