10.34: Numerical Methods Applied to Chemical Engineering

Lecture 4:
Gaussian elimination
Sparse matrices

Recap

- Vector spaces
- Linear dependence
- Existence and uniqueness of solutions
- Four fundamental subspaces

Recap

- What is the column space of a matrix?
- What is the null space of a matrix?
- What are the conditions for existence and uniqueness of solutions to linear equations?

Easy to Solve Linear Equations

- Diagonal:
  - Go row by row

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

- Triangular:
  - Back substitution

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ 3 \end{pmatrix}$$

- Goal: transform complicated equations into easy ones

Gaussian Elimination

- Solving N equations with N unknowns:
  - Example:
$$\begin{array}{cccc} 2x_1 & -x_2 & 0 & = & 0 \\ -x_1 & +2x_2 & -x_3 & = & 1 \\ 0 & -x_2 & +2x_3 & = & 0 \end{array}$$

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\left[\begin{array}{ccc|c} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 1 \\ 0 & -1 & 2 & 0 \end{array}\right]$$

    - Convert to triangular form using elementary row operations
      - $(\text{row})_1 \rightarrow c(\text{row})_1$
      - $(\text{row})_1 \rightarrow a(\text{row})_1 + b(\text{row})_2$
      - $(\text{row})_1 \leftrightarrow (\text{row})_2$

Gaussian Elimination

- Solving N equations with N unknowns:
  - Example: $\left[\begin{array}{ccc|c} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 1 \\ 0 & -1 & 2 & 0 \end{array}\right]$
    - step 1: $(\text{row})_2 \rightarrow (\text{row})_2 + (1/2)(\text{row})_1$

$$\left[\begin{array}{ccc|c} 2 & -1 & 0 & 0 \\ 0 & 3/2 & -1 & 1 \\ 0 & -1 & 2 & 0 \end{array}\right]$$

    - step 2: $(\text{row})_3 \rightarrow (\text{row})_3 + (2/3)(\text{row})_2$

$$\left[\begin{array}{ccc|c} 2 & -1 & 0 & 0 \\ 0 & 3/2 & -1 & 1 \\ 0 & 0 & 4/3 & 2/3 \end{array}\right]$$

    - solve by back substitution.

Gaussian Elimination

- Solving N equations with N unknowns:
  - Example:
$$\left[\begin{array}{cccc|c} A_{11} & A_{12} & \dots & A_{1N} & b_1 \\ A_{21} & A_{22} & \dots & A_{2N} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \\ A_{N1} & A_{N2} & \dots & A_{NN} & b_N \end{array}\right]$$
    - step 1, select pivot: A_{11} $\quad \lambda_{k1} = A_{k1}/A_{11}$
    - step 2, do row operations: $(\text{row})_k \to (\text{row})_k - \lambda_{k1}(\text{row})_1 \quad k > 1$

$$\left[\begin{array}{cccc|c} A_{11} & A_{12} & \dots & A_{1N} & b_1 \\ 0 & A_{22} - \lambda_{21}A_{12} & \dots & A_{2N} - \lambda_{21}A_{1N} & b_2 - \lambda_{21}b_1 \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & A_{N2} - \lambda_{N1}A_{12} & \dots & A_{NN} - \lambda_{N1}A_{1N} & b_N - \lambda_{N1}b_1 \end{array}\right]$$

    - step 3, select pivot: $A_{22} - \lambda_{21}A_{12}$ $\quad \lambda_{k2} = A_{k2}/(A_{22} - \lambda_{21}A_{12})$
    - step 4, do row operations: $(\text{row})_k \to (\text{row})_k - \lambda_{k2}(\text{row})_2 \quad k > 2$
    - rinse and repeat until upper triangular
    - solve by back substitution

Gaussian Elimination

- Gaussian elimination requires how many operations?
- Is Gaussian elimination reliable (stable)?
  - Example:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

  - Partial pivoting:
    - If a selected pivot is zero, perform an additional row operation and reselect the pivot.
      - Swap the pivot row for a row with a non-zero pivot: $(\text{row})_k \leftrightarrow (\text{row})_l$
      - What if all potential pivots are zero?

Gaussian Elimination

- Gaussian elimination requires how many operations?
- Is Gaussian elimination reliable (stable)?
  - Example:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

  - Partial pivoting:
    - If a selected pivot is zero, perform an additional row operation and reselect the pivot.
      - Swap the pivot row for a row with a non-zero pivot: $(\text{row})_k \leftrightarrow (\text{row})_l$
      - What if all potential pivots are zero?

Gaussian Elimination

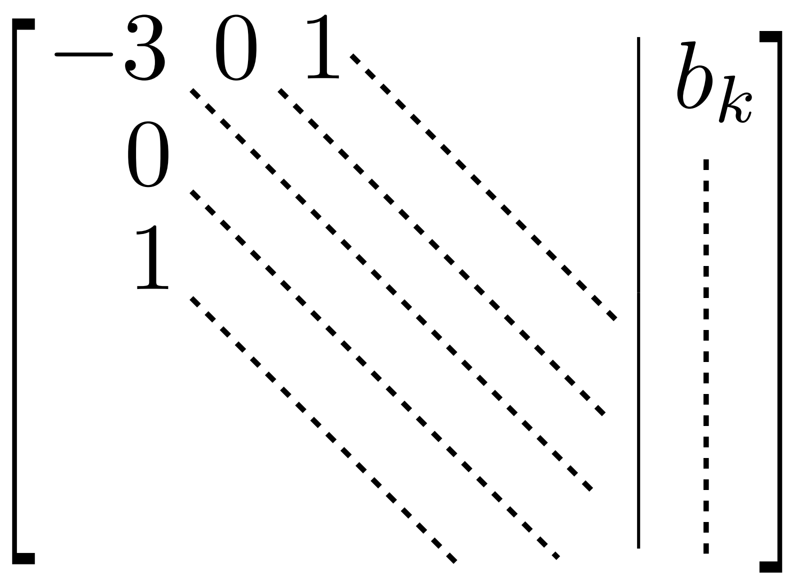
- Pivoting to improve accuracy:
  - Example with three digit accuracy: $\left[\begin{array}{cc|c} 10^{-4} & 1 & 1 \\ 1 & -1 & 0 \end{array}\right]$
    - eliminate first column:

$$\left[\begin{array}{cc|c} 10^{-4} & 1 & 1 \\ 0 & -1.00 \times 10^4 & -1.00 \times 10^4 \end{array}\right]$$

    - solve by back substitution: $x_1 = 0.00 \quad x_2 = 1.00$
    - exact solution is: $x_1 = 0.9999 \quad x_2 = 0.9999$
    - repeat after swapping rows 1 and 2…
  - Small pivots can lead to large errors.
    - Therefore, many algorithms implement a pivoting strategy that uses the largest available pivot to minimize numerical errors.

Gaussian Elimination

- Example: Gaussian elimination of a sparse $N \times N$ system

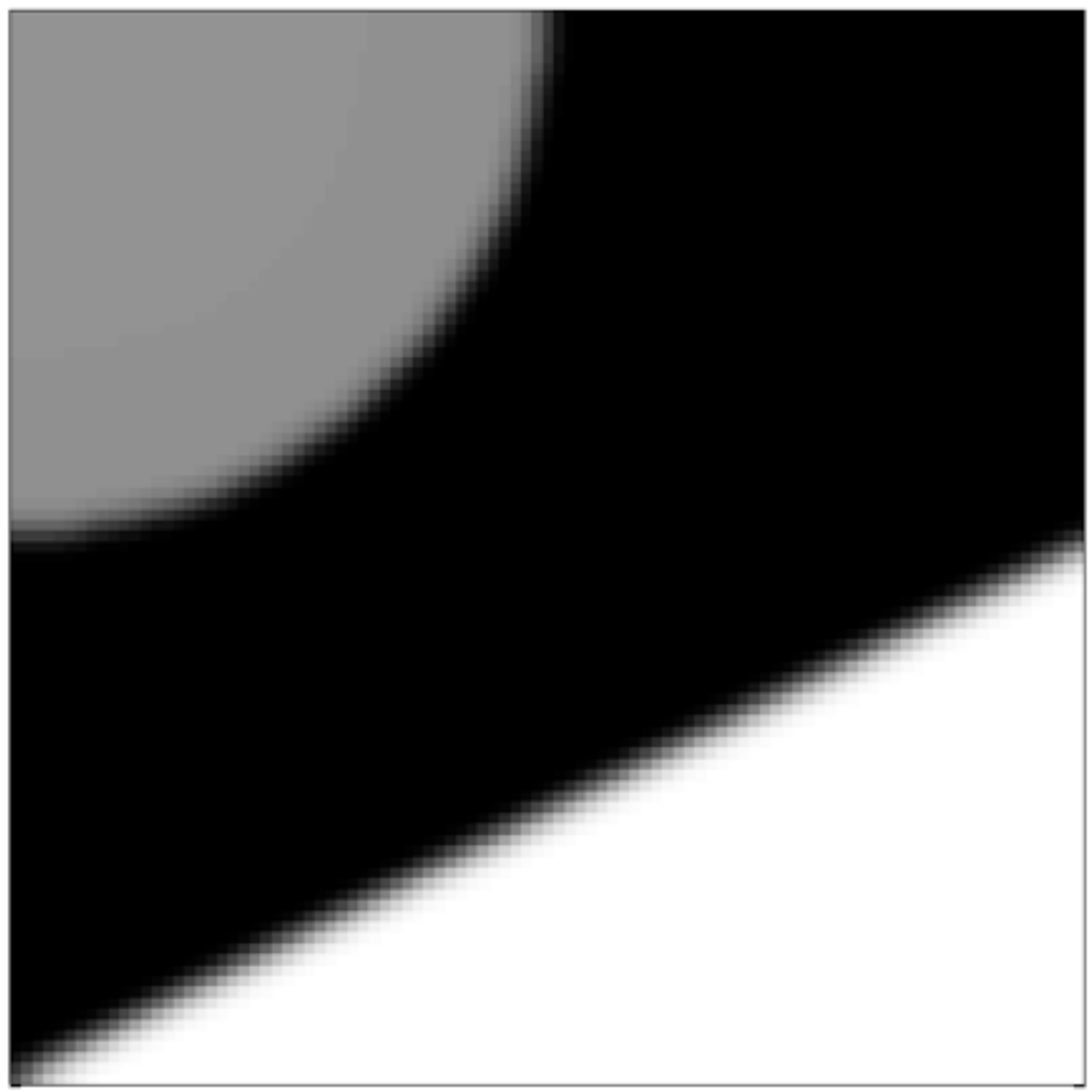


  - What is the most memory I would need to perform Gaussian elimination?
  - What is the least amount of memory I would need to perform Gaussian elimination?
  - How should I store the matrix?

Sparse Matrices

- Example: a finite volume model of diffusion

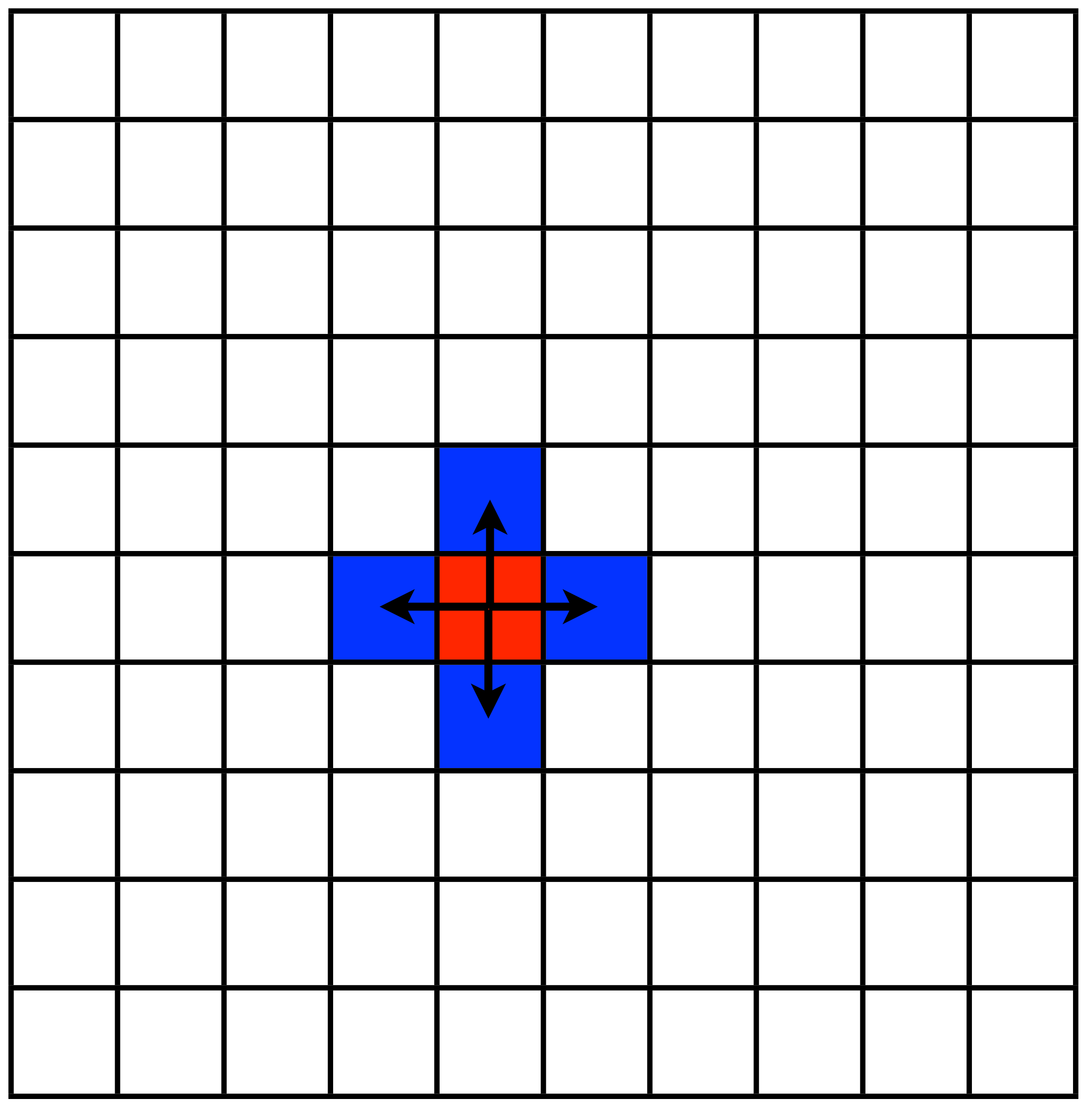
$$\frac{\partial c}{\partial t} = D\nabla^2 c$$



Sparse Matrices

- Example: a finite volume model of diffusion

$$\frac{\partial c}{\partial t} = D\nabla^2 c$$



conserve the flux from one cell to the next

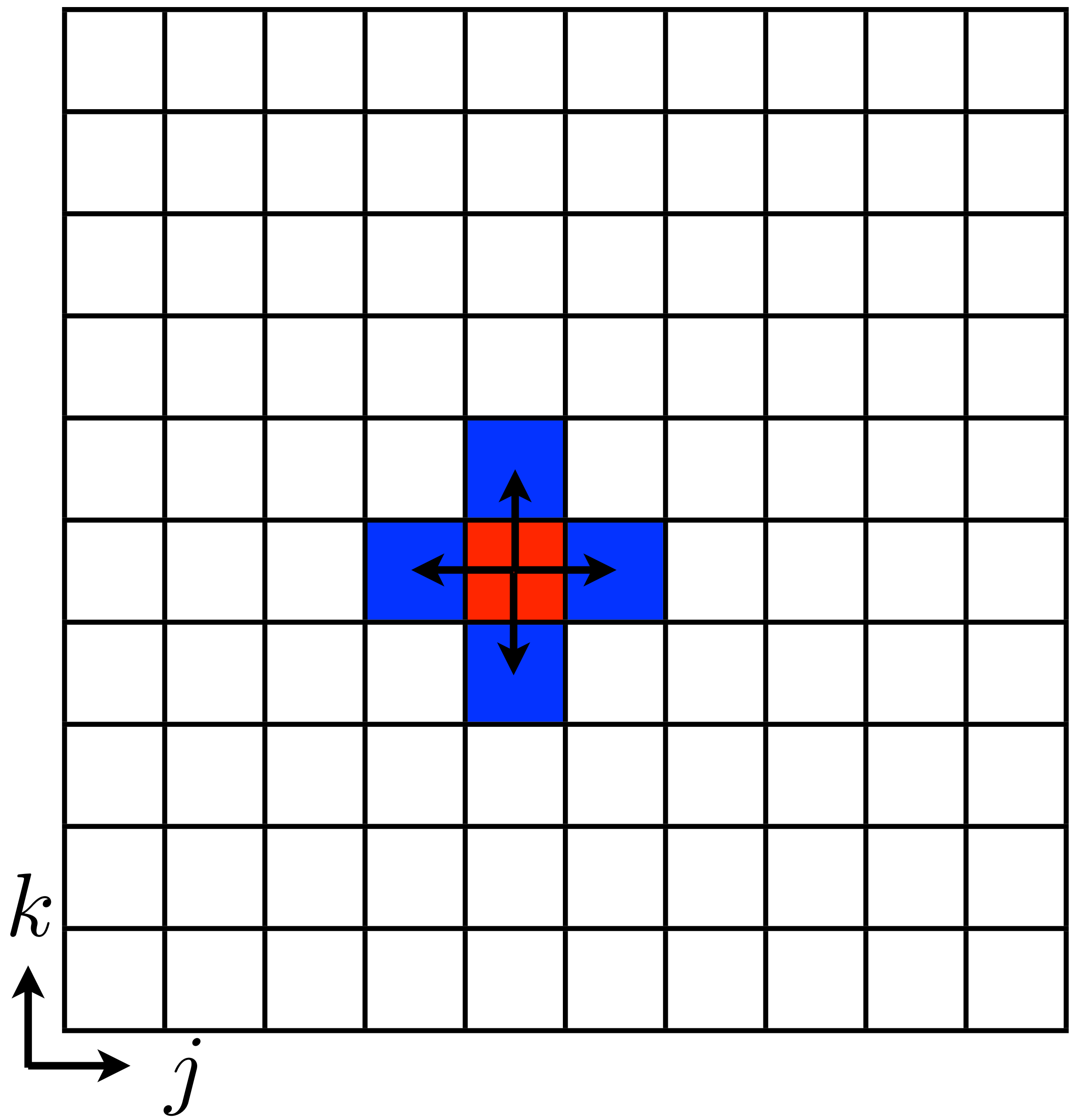
only neighboring cells interact

$$\mathbf{c}_{i+1} = \mathbf{c}_i + \frac{\Delta t D}{\Delta x^2}\mathbf{A}\mathbf{c}_i$$

Sparse Matrices

- Example: a finite volume model of diffusion

$$\frac{\partial c}{\partial t} = D\nabla^2 c$$



conserve the flux from one cell to the next

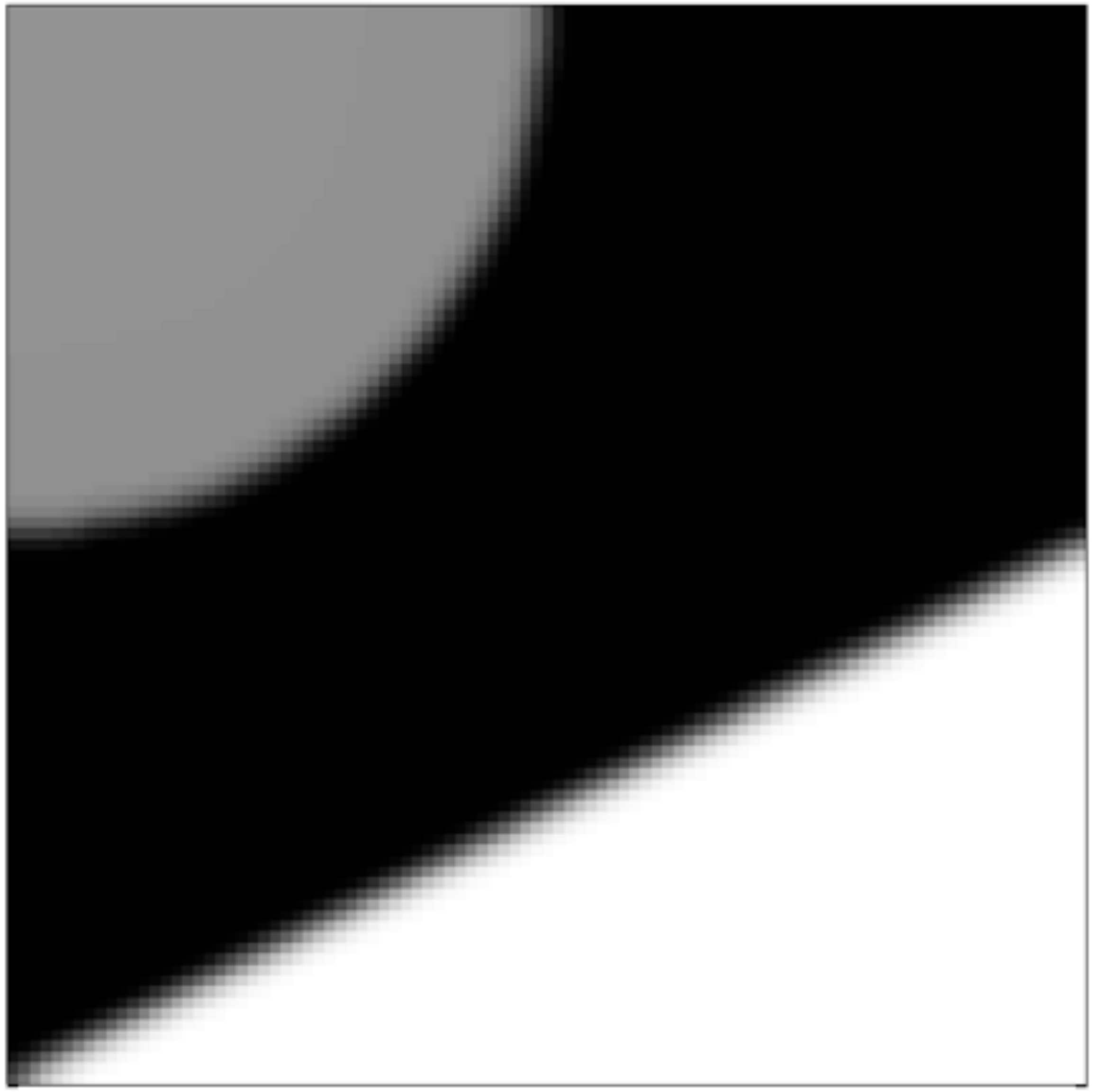
only neighboring cells interact

$$c_{i+1}^{j+Nk} = c_{i+1}^{j+Nk} + \frac{\Delta t D}{\Delta x^2}\left(c_i^{j+1+Nk} + c_i^{j-1+Nk} + c_i^{j+N(k+1)} + c_i^{j+N(k-1)} - 4c_i^{j+Nk}\right)$$

Sparse Matrices

- Example: a finite volume model of diffusion

$$\frac{\partial c}{\partial t} = D\nabla^2 c$$



$$c_{i+1}^{j+Nk} = c_{i+1}^{j+Nk} + \frac{\Delta t D}{\Delta x^2}\left(c_i^{j+1+Nk} + c_i^{j-1+Nk} + c_i^{j+N(k+1)} + c_i^{j+N(k-1)} - 4c_i^{j+Nk}\right)$$

Sparse Matrices

- Example: a finite volume model of diffusion

$$\frac{\partial c}{\partial t} = D\nabla^2 c$$

$$\mathbf{c}_{i+1} = \mathbf{c}_i + \frac{\Delta t D}{\Delta x^2}\mathbf{A}\mathbf{c}_i$$

$$c_{i+1}^{j+Nk} = c_{i+1}^{j+Nk} + \frac{\Delta t D}{\Delta x^2}\left(c_i^{j+1+Nk} + c_i^{j-1+Nk} + c_i^{j+N(k+1)} + c_i^{j+N(k-1)} - 4c_i^{j+Nk}\right)$$

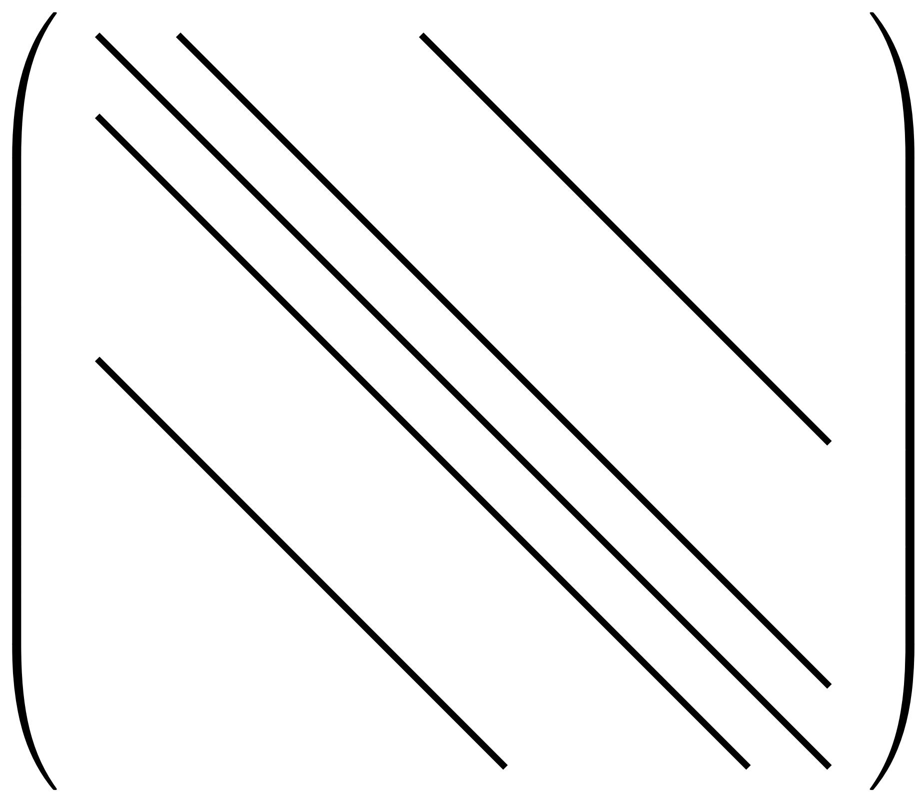
$$\boldsymbol{A} = \begin{pmatrix} \\ \\ \\ \end{pmatrix}$$

Sparse Matrices

- Example: a finite volume model of diffusion

$$\mathbf{c}_{i+1} = \mathbf{c}_i + \frac{\Delta t D}{\Delta x^2}\mathbf{A}\mathbf{c}_i$$

$\boldsymbol{A} =$



- How many operations to compute $\mathbf{A}\mathbf{c}_i$ when $\mathbf{A}$ is $N^2 \times N^2$
- How much memory to store $\mathbf{A}$ as a full matrix?
- How much memory to store $\mathbf{A}$ as a sparse matrix?

Sparse Matrices

- Example: Gaussian elimination of a structured matrix
  - Before elimination: $$\left[\begin{array}{cccccc|c} \times & \times & \times & \times & \times & \times & \times \\ \times & \times & & & & & \times \\ \times & & \times & & & & \times \\ \times & & & \times & & & \times \\ \times & & & & \times & & \times \\ \times & & & & & \times & \times \end{array}\right]$$
  - First column eliminated: $$\left[\begin{array}{cccccc|c} \times & \times & \times & \times & \times & \times & \times \\ & \times & \times & \times & \times & \times & \times \\ & \times & \times & \times & \times & \times & \times \\ & \times & \times & \times & \times & \times & \times \\ & \times & \times & \times & \times & \times & \times \\ & \times & \times & \times & \times & \times & \times \end{array}\right]$$
  - After elimination: $$\left[\begin{array}{cccccc|c} \times & \times & \times & \times & \times & \times & \times \\ & \times & \times & \times & \times & \times & \times \\ & & \times & \times & \times & \times & \times \\ & & & \times & \times & \times & \times \\ & & & & \times & \times & \times \\ & & & & & \times & \times \end{array}\right]$$

Sparse Matrices

- Example: Gaussian elimination of a structured matrix
  - Before elimination:

$$\left[\begin{array}{cccccc|c} \times & \times & \times & \times & \times & \times & \times \\ \times & \times & & & & & \times \\ \times & & \times & & & & \times \\ \times & & & \times & & & \times \\ \times & & & & \times & & \times \\ \times & & & & & \times & \times \end{array}\right]$$

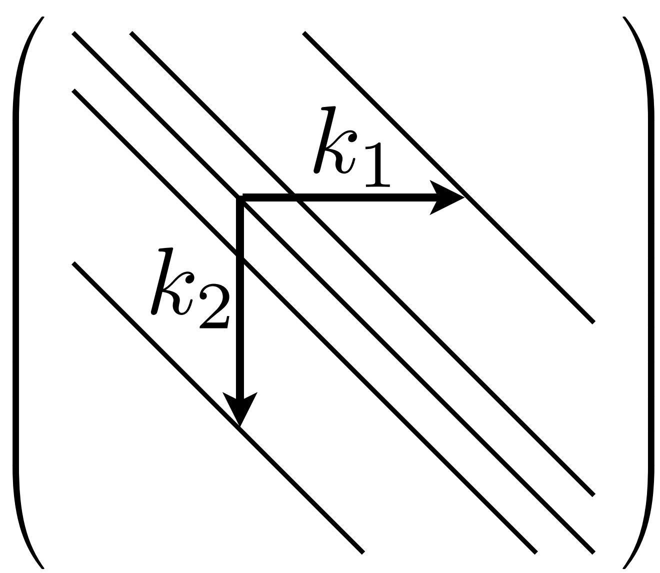
  - Swap last and first rows and columns:

$$\left[\begin{array}{cccccc|c} \times & & & & & \times & \times \\ & \times & & & & \times & \times \\ & & \times & & & \times & \times \\ & & & \times & & \times & \times \\ & & & & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \end{array}\right]$$

  - After elimination:

Fill-in

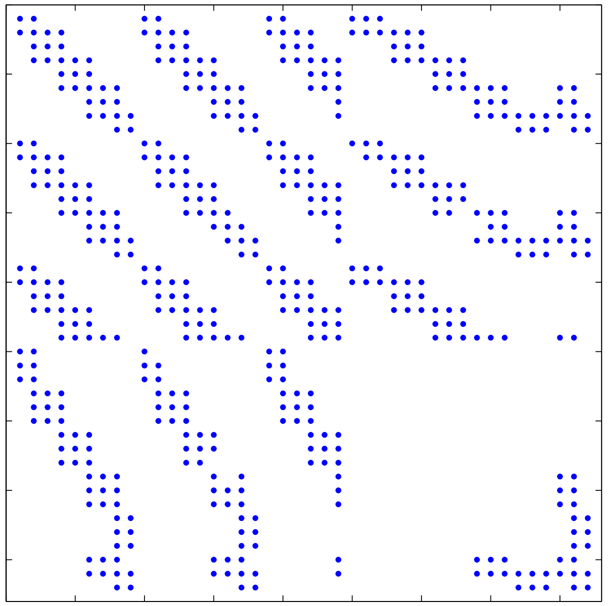
- Gaussian elimination fills in sparse matrices
  - The amount of fill-in depends on the sparse structure.
  - In general, lower bandwidth sparsity patterns, have smaller amounts of fill-in.
    - Bandwidth:



    - In the worst case, GE doubles the bandwidth
  - There are algorithms that reorder matrices with the goal of minimizing the amount of fill-in.

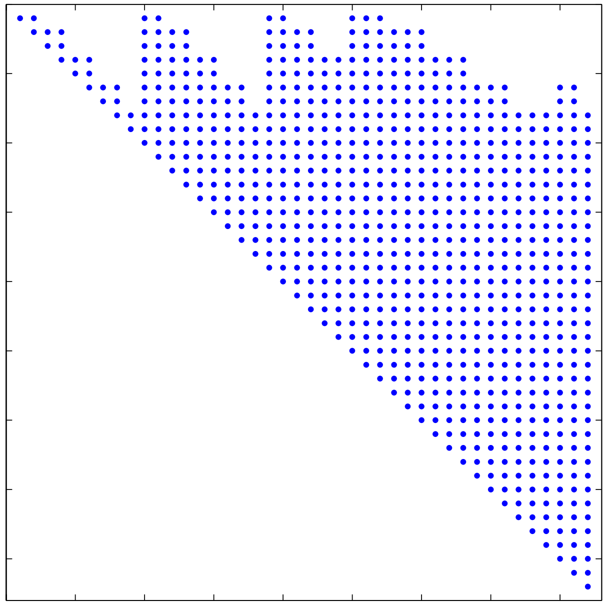
Fill-in

- Fill-in is reduced by reordering:



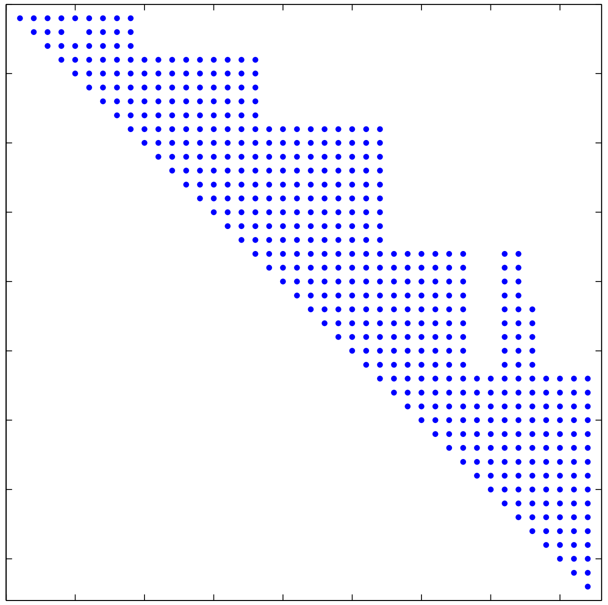
505 non-zero

original



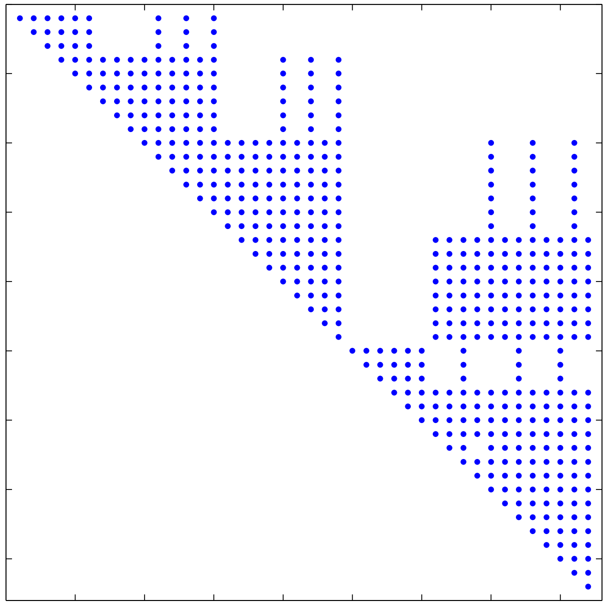
780 non-zero

symrcm



490 non-zero

symamd



479 non-zero

Permutation

- Reordering through use of permutation matrices:
  - Consider the operation of swapping two rows. This can be done through matrix multiplication.

$$
\boldsymbol{P} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}
$$

swap row 1 and 2

identity

  - For example:

$$
\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_2 \\ x_1 \\ x_3 \end{pmatrix}
$$

Permutation

- Reordering through use of permutation matrices:
  - Consider the operation of swapping two rows. This can be done through matrix multiplication.

$$\boldsymbol{P} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}$$

swap row 1 and 2

identity

$$\boldsymbol{PA} = \begin{pmatrix} \boldsymbol{PA}_1^C & \boldsymbol{PA}_2^C & \dots & \boldsymbol{PA}_N^C \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}_2^R \\ \boldsymbol{A}_1^R \\ \boldsymbol{A}_3^R \\ \vdots \\ \boldsymbol{A}_N^R \end{pmatrix}$$

Permutation

- Reordering through use of permutation matrices:
  - How do I swap columns?
- Permutation matrices are unitary:

$$\mathbf{P}\mathbf{P}^T = \mathbf{I}$$
$$\mathbf{P}^T = \mathbf{P}^{-1}$$

- Reordering a system of equations:

$$(\boldsymbol{P}_1 \boldsymbol{A} \boldsymbol{P}_2^T)(\boldsymbol{P}_2 \boldsymbol{x}) = \boldsymbol{P}_1 \boldsymbol{b}$$

  - Reordering is a form of preconditioning!
  - Reordering can be used for pivoting!

Permutation

- Reordering through use of permutation matrices:
  - Permutation matrices are sparse too. How are they stored?
    - Example reversing the order of 10 rows:

old position	1	2	3	4	5	6	7	8	9	10
new position	10	9	8	7	6	5	4	3	2	1

  - Permutation matrices are sparse too. How are they used?
    - `P = [ 10 9 8 7 6 5 4 3 2 1 ]`
    - `A = A( P, : )`

Permutation

- Reordering through use of permutation matrices:
  - Example:
    - `P = symrcm(A);`
    - `figure; spy(A);`
    - `figure; spy( A( P, P ) )`