ITERATIVE METHODS

Numerical methods in chemical engineering

Edwin Zondervan

OVERVIEW

- Iterative methods for large systems of equations
- We will solve Laplace's equation for steady state heat conduction

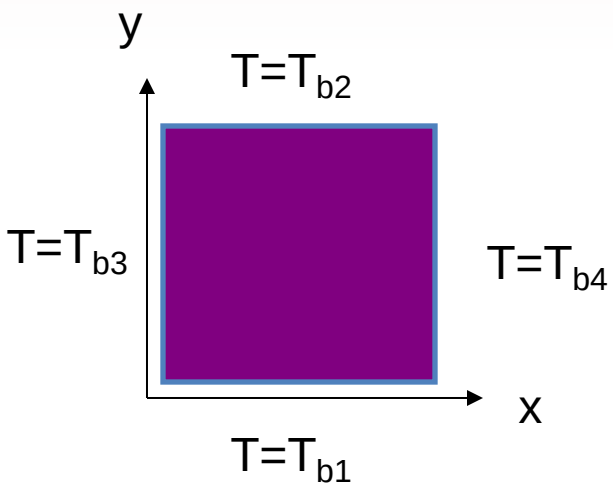
LAPLACE'S EQUATIONS

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T \qquad (4\text{-}1)$$

Thermal diffusivity

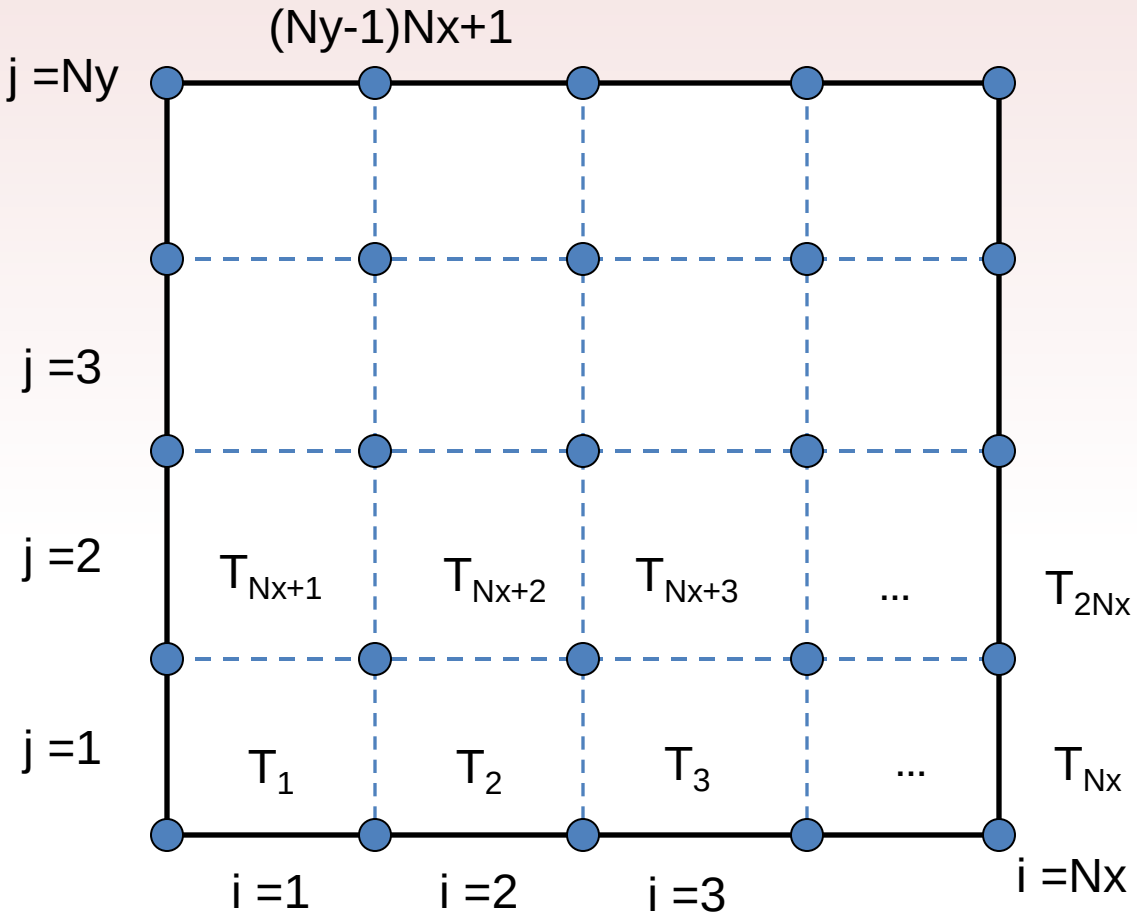
The steady state problem:

$$\nabla^2 T = 0 \qquad (4\text{-}2)$$



$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \qquad (4\text{-}3)$$

TRACK TEMPERATURE ON A GRID



Index of a node is given by:

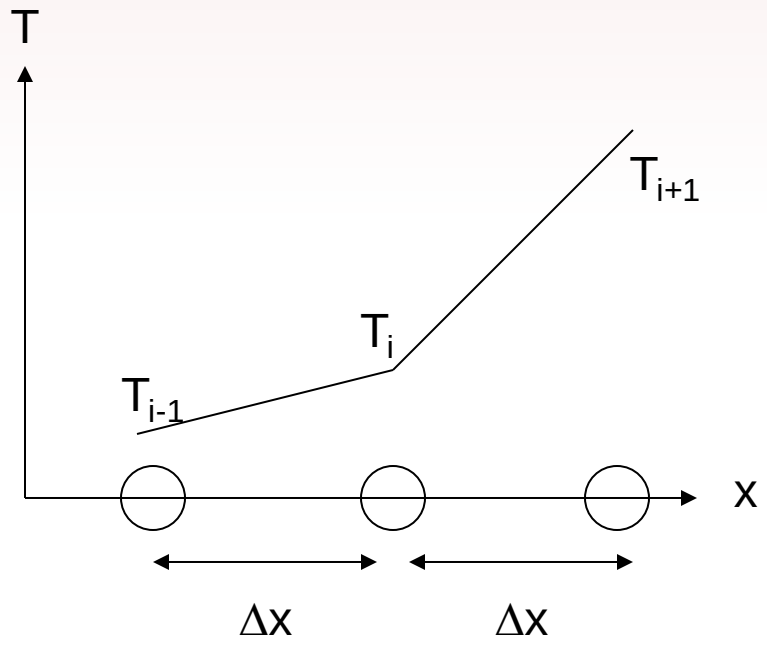
$K = i + Nx(j-1)$

So that:

$T_{i,j} = T_{k=i+Nx(j-1)}$

ESTIMATES OF THE SECOND DIFFERENTIALS FOR X

- Assume a piece-wise linear profile in the temperature, e.g.:



$$\frac{\partial^2 T}{\partial x^2} \approx \frac{\left.\frac{\partial T}{\partial x}\right|_{i+1/2} - \left.\frac{\partial T}{\partial x}\right|_{i-1/2}}{\Delta x} \qquad (4\text{-}4)$$

$$\approx \frac{\frac{\left(T_{i+1,j} - T_{i,j}\right)}{\Delta x} - \frac{\left(T_{i,j} - T_{i-1,j}\right)}{\Delta x}}{\Delta x}$$

$$= \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} \qquad (4\text{-}5)$$

INCLUDE THE ESTIMATES OF THE SECOND DIFFERENTIALS FOR Y

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} + \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta y^2} = 0 \qquad (4\text{-}6)$$

$$\frac{T_{k+1} - 2T_k - T_{k-1}}{\Delta x^2} + \frac{T_{k+Nx} - 2T_k - T_{k-Nx}}{\Delta y^2} = 0 \qquad (4\text{-}7)$$

Equal spaced grid $\Delta x = \Delta y = 1$

$$T_{k-Nx} + T_{k-1} - 4T_k + T_{k+1} + T_{k+Nx} = 0 \qquad (4\text{-}8)$$

BOUNDARY CONDITIONS

- For the nodes on the boundary, we have a simple equations:

$$T_{k,boundary} = \text{some fixed temperature} \quad (4\text{-}9)$$

- The equation on the previous slide and the boundaries can be written as a matrix equation:

$$AT = b \quad (4\text{-}10)$$

LET'S WRITE THIS STUFF AS MATRIX EQUATIONS

```
Nx=5;          %number of points along x direction
Ny=5;          %number of points in the y direction
d = 1/Nx;      %the grid spacing
Alpha = 1;     %thermal diffusivity

e = ones(Nx*Ny,1);
A = spdiags([e,e,-4*e,e,e],[-Nx,-1,0,1,Nx],Nx*Ny,Nx*Ny);
A = A*alpha/d^2;
```

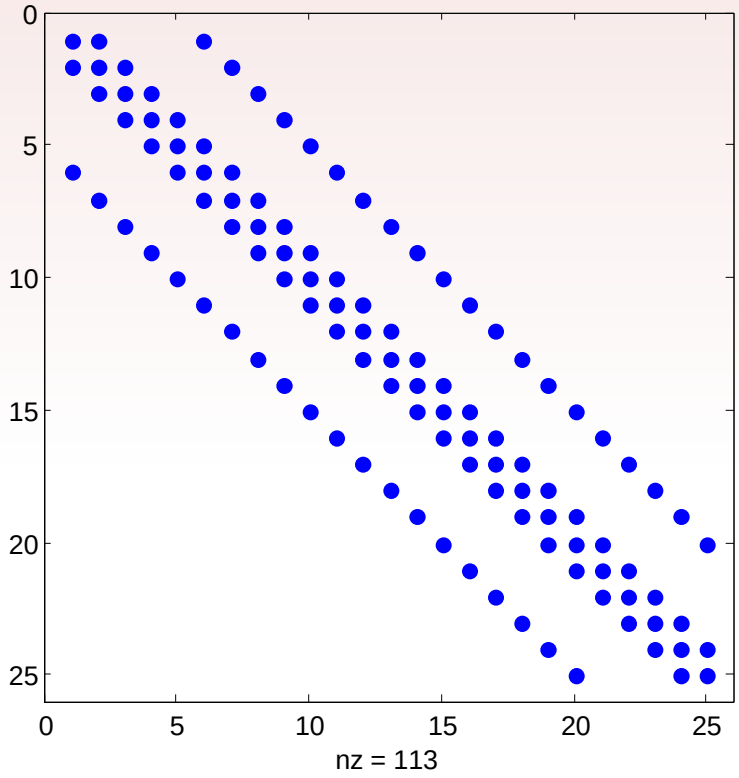
MATRIX SPARSITY

`Spy(A)` Nx=Ny = 5

A sparse matrix structure,

which is not tridiagonal: there are offset bands.

Offset bands can cause your trouble!!

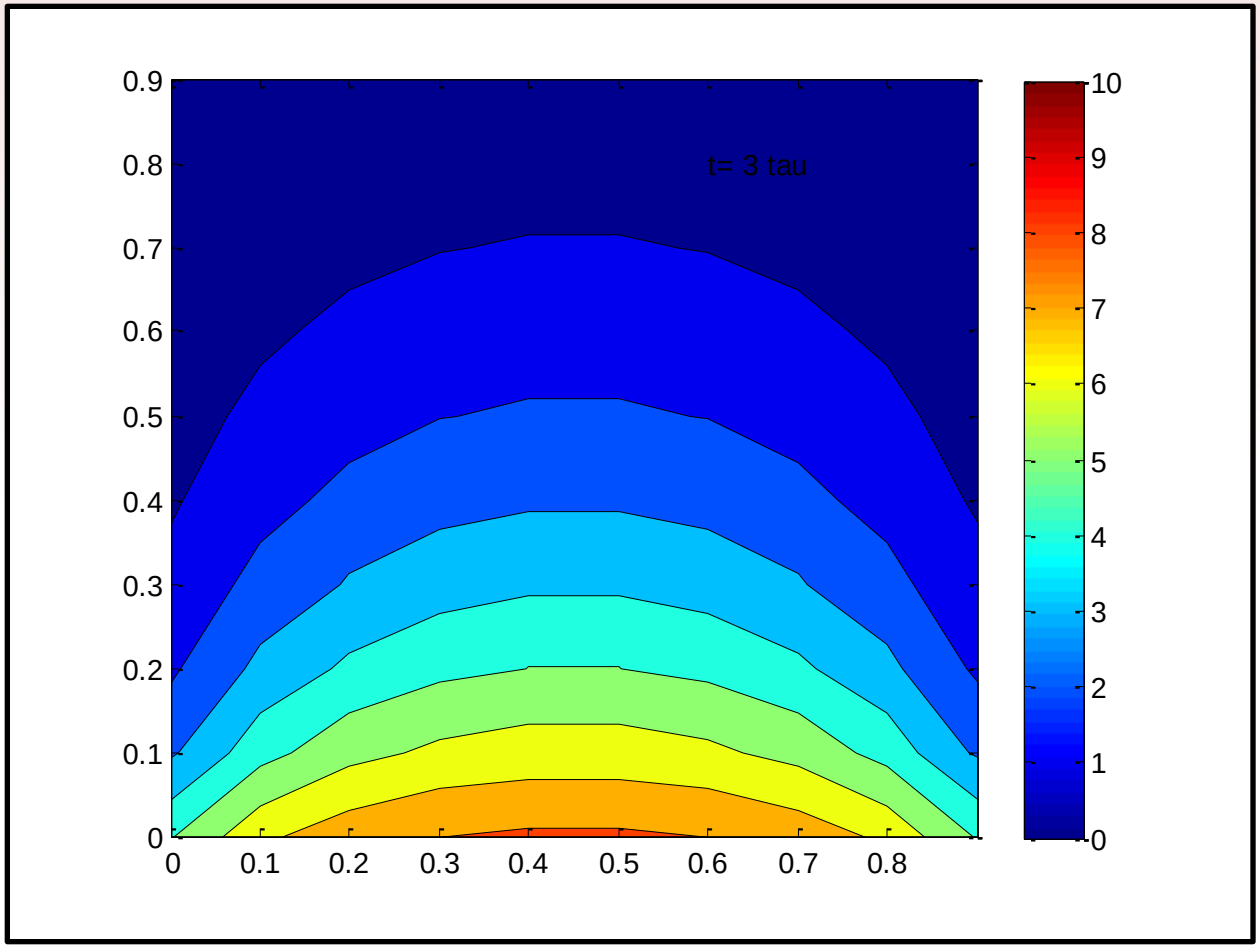


SOLUTION TO THE LAPLACE EQUATION

You can solve the system with \, for Nx=Ny=5 and

Tb1 = 10

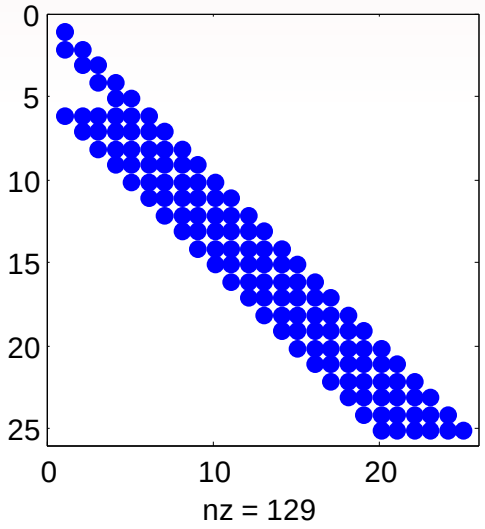
Tb2 = Tb3 = Tb4 = 0

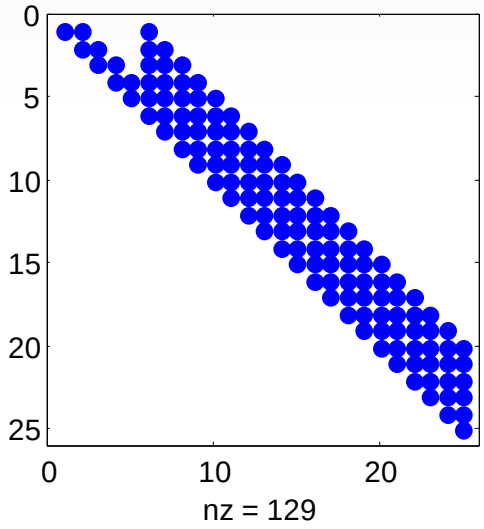


LU DECOMPOSITION

```
[L,U,P] = lu(A)
subplot(1,2,1)
spy(L)
subplot(1,2,2)
spy(U)
```

With LU decomposition we produce matrices that are less sparse than the original matrix.





LU DECOMPOSITION

- Gaussian elimination on a matrix like A requires more memory (with 3D problems, the offset in the diagonal would even be bigger!)
- In general extra memory allocation will not be a problem for MATLAB
- MATLAB is clever, in that sense that it attempts to reorder equations, to move elements closer to the diagonal)

ITERATIVE METHODS

- Alternatives for Gaussian elimination
- Use iterative methods when systems are large and sparse.
- Often such systems are encountered when we want to solve PDE's of higher dimensions (>1D)

EXAMPLES

- **Jacobi method**
- Gauss-Seidel method
- Succesive over relaxation

- `bicg` – Bi-conjugate gradient method
- `gmres` – generalized minimum residuals method
- `bicgstab` – Bi-conjugate gradient method

THE JACOBI METHOD

- In our previous example we derived the following equation:

$$T_{k-Nx} + T_{k-1} - 4T_k + T_{k+1} + T_{k+Nx} = 0 \qquad (4\text{-}11)$$

- Rearranging:

$$T_k = \frac{T_{k-Nx} + T_{k-1} + T_{k+1} + T_{k+Nx}}{4} \qquad (4\text{-}12)$$

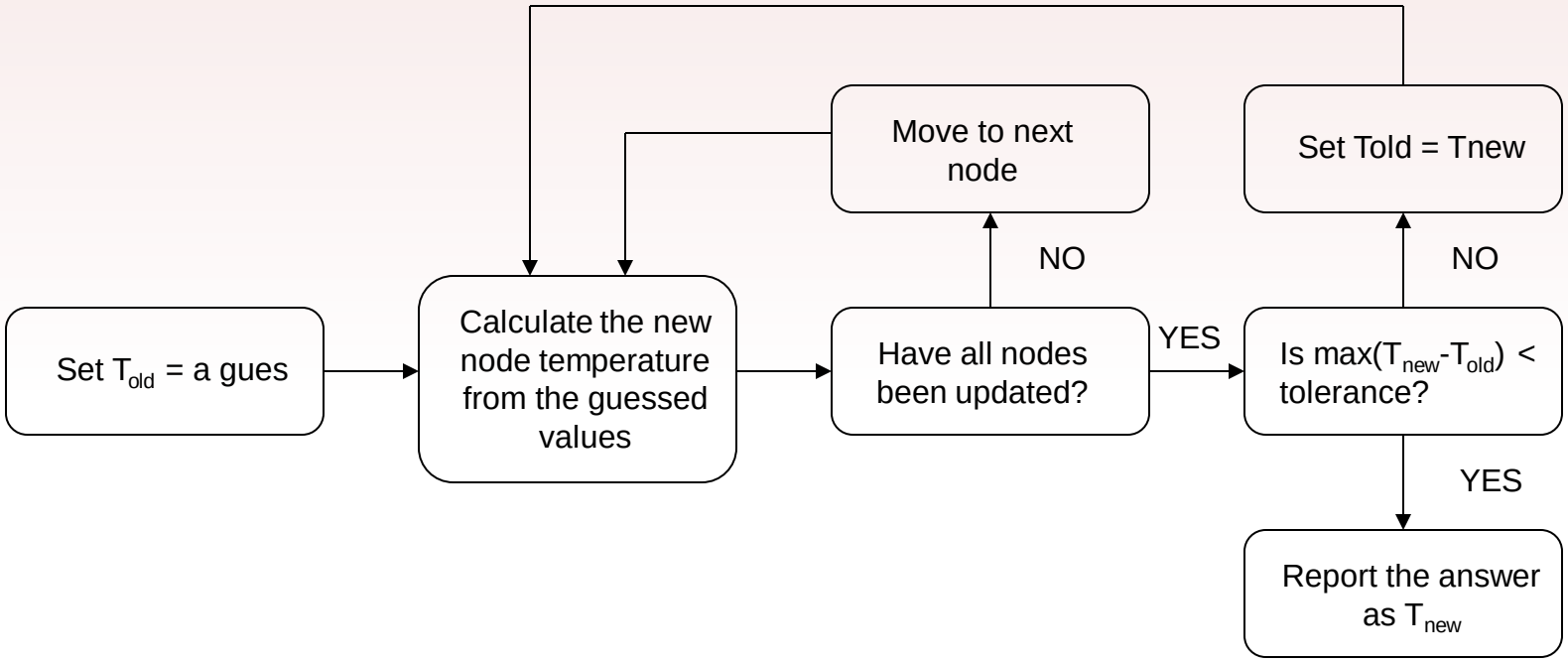
THE JACOBI METHOD

- In the Jacobi scheme the iteration proceeds as follows:
  - Start with an initial guess for the values of T at each node, we calculate an new, updated values using the equation of the previous slide and store a new vector:

$$T_{k,new} = \frac{T_{k-Nx,old} + T_{k-1,old} + T_{k+1,old} + T_{k+Nx,old}}{4} \qquad (4\text{-}13)$$

  - Do this for all other nodes, and use new values as guess, Repeat!

DIAGRAM OF JACOBI METHOD



Only two vectors need to be stored: the old Temp. And the new Temp.

NOW WITH MATRICES

Consider a matrix A:

$$A = \begin{pmatrix} * & * & & & * & \\ * & * & * & & & * \\ & * & * & * & & \\ & & * & * & * & \\ * & & & * & * & * \\ & * & & & * & * \end{pmatrix}$$

Split it into a diagonal matrix D and another matrix S:

$$A = \begin{pmatrix} * & & & & & \\ & * & & & & \\ & & * & & & \\ & & & * & & \\ & & & & * & \\ & & & & & * \end{pmatrix} + \begin{pmatrix} & * & & & * & \\ * & & * & & & * \\ & * & & * & & \\ & & * & & * & \\ * & & & * & & * \\ & * & & & * & \end{pmatrix}$$

SOLVE AT-B

- Now we can solve AT=b by:
  - $(D+S).T = b$
  - $D.T = b - S.T$
  - $D.T^{new} = b - S.T^{old}$
  - $T^{new} = D^{-1}(b-S.T^{old})$

CONVERGENCE OF THE METHOD

- Now we define an error at the k-th iteration:
  - $error^k = T^k - T$
- The error at the next iteration is:
  - $T^{k+1} = D^{-1}(b - S.T^k)$
  - $Error^{k+1} + T = D^{-1}(b - S.error^k - S.T)$
  - $D.error^{k+1} = -S.error^k$
  - $Error^{k+1} = -D^{-1}S.error^k$

Eigenvalues of $D^{-1}S$ must always have modulus < 1, to ensure that $|error^{k+1}| < |error^k|$

CONVERGENGE OF THE METHOD

- We can express the vector of error in terms of eigenvectors of $D^{-1}S$:
  - $\text{error} = a_1u_1 + a_2u_2 + a_3u_3 + a_4u_4 + \ldots$ (4-14)
- So:
  - $\text{error}^{k+1} = -D^{-1}S(a_1u_1 + a_2u_2 + a_3u_3 + a_4u_4 + \ldots)$

$$= -a_1D^{-1}Su_1 - a_2D^{-1}Su_2 - a_3D^{-1}Su_3 - a_4D^{-1}Su_4 + \ldots$$

$$= -a_1\lambda_1u_1 - a_2\lambda_2u_2 - a_3\lambda_3u_3 - a_4\lambda_4u_4 + \ldots \quad (4\text{-}15)$$

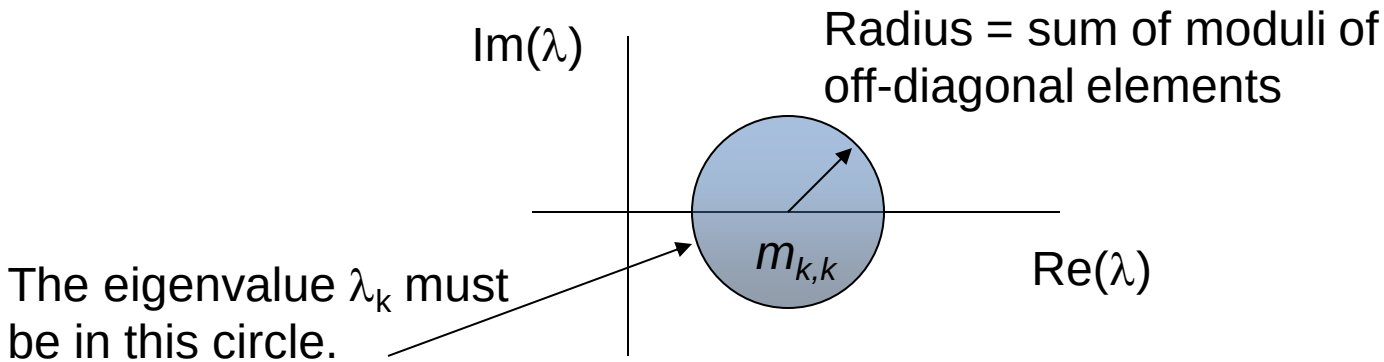
The magnitude of the error will grow each iteration, if λ's have a complex module >1, So: Find the largest magnitude of eigenvalues for the matrix $D^{-1}S$

GERSHGORIN'S THEOREM

- For a square matrix and row *k*, an eigenvalue is located on the complex plane within a radius equal or less to the sum of the moduli of the off-diagonal elements of that row.

$$|\lambda - m_{k,k}| \leq |m_{k,1}| + |m_{k,2}| + \ldots + |m_{k,N-2}| + |m_{k,N-1}| + |m_{k,N}|$$

(4-16)



APPLICATION OF GERSHGORIN'S THEOREM

- Appying Gershgorin's theorem to our Jacobi iteration, the off-diagonal elements of row *k* of $D^{-1}S$ are $1/a_{k.k}$ times the off-diagonal elements of our original matrix A, while the diagonal element is zero:
- For $|\lambda_k|<1$:

$$\sum_{j \neq k} | a_{k,j} | < | a_{k,k} | \qquad (4\text{-}17)$$

Stability: size of diagonal element must be larger that the sum of moduli of the other elements, such matrix is called *diagonally dominant*.

SUMMARY

- Partial differential equations can be written as sparse systems of linear equations
- Sparse systems can be handled with a direct method like Gaussian elimination
- If you have systems of more than 1 dimension, a direct method still can be used, if there are no memory issues, otherwise an iterative method may be attractive.
- The Jacobi method was introduced. Many other methods are based on the Jacobi method (SOR method, for example)