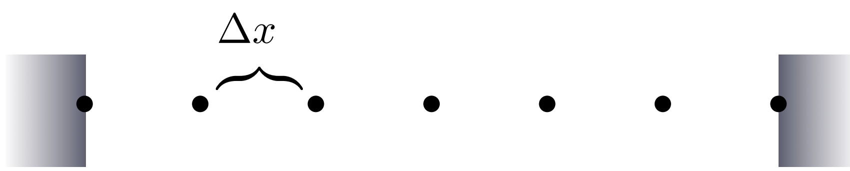
2. Matrix approach to B.V.P. : Implicit methods

How does this become a linear algebra (matrix · vector = vector) problem?
Consider $d^2_x T - T = 0$, $T(0) = T_0$, $T(1) = T_1$.



a. Set up grid -- here, constant Δx.
b. Store solution in array [T(1), T(2), T(3)...] = T(I).
c. Linear algebra (**A · T**=**B**).

Setting up the **A** *matrix:*

Say $d^2_x T - T = 0$ with $T(0)=T_0$; $T(1)=T_1$. Recall second order central

$$\frac{\delta^2 T_i}{\delta x^2} = \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2}$$

(T(i),i=1...I) i.e. T(i) = **T**

As a vector product: column solution vector

i th row of matrix

$$\frac{\delta^2 T_i}{\delta x^2} = \left(\underbrace{0, 0, \ldots\ldots, 0}_{i-2}, \frac{1}{\Delta x^2}, \frac{-2}{\Delta x^2}, \frac{1}{\Delta x^2}, \underbrace{0, \ldots\ldots, 0}_{I-i-1} \right) \cdot \boldsymbol{T}$$

Stack this for all i=2,I-1. B.c.s on first and last row (add -1 to diag for diff eq.)

d. Solve simultaneous equations for T(i) including boundary conditions

Store only the 3 diagonal elements, not the whole matrix

$$
\begin{pmatrix}
\bullet \quad 0 \quad\quad 0 \quad 0,0,\ldots\ldots\ldots\ldots,0,\ 0 \\
\frac{1}{\Delta x^2},\frac{-2}{\Delta x^2}-1,\frac{1}{\Delta x^2},0,0,\ldots\ldots\ldots\ldots,0,\ 0 \\
0,0\ldots..0,\frac{1}{\Delta x^2},\frac{-2}{\Delta x^2}-1,\frac{1}{\Delta x^2},0,\ldots\ldots\ldots,0 \\
0,0\ldots\ldots..0,\frac{1}{\Delta x^2},\frac{-2}{\Delta x^2}-1,\frac{1}{\Delta x^2},0\ldots\ldots0 \\
0,0\ldots\ldots\ldots..0,\frac{1}{\Delta x^2},\frac{-2}{\Delta x^2}-1,\frac{1}{\Delta x^2},0..0 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
0,0,0,\ldots\ldots\ldots\ldots\ldots\ldots,0,0,\bullet
\end{pmatrix}
\qquad
\begin{pmatrix}
T_0 \\ 0 \\ 0 \\ . \\ . \\ . \\ . \\ T_1
\end{pmatrix}
$$

Boundary conditions first and last rows

$A_{i,i-1}=1/\Delta x^2$, $A_{i,i}=-2/\Delta x^2-1$, $A_{i,i+1}=1/\Delta x^2$; all other $A_{i,j}$'s are 0: matrix of size `A(N,3)`

Note first and last row have only two elements inside tridiag-structure. (Another element could be handled by modifying Gauss elimination.)

Matrix for $d^2_x T - T = 0$

B(2:I-1) = 0
A(2:I-1,1) = $1/\Delta x^2$
A(2:I-1,2) = $-2/\Delta x^2 - 1$
A(2:I-1,3) = $1/\Delta x$

b.c.'s: A(1,2) = ?
A(1,3) = ?
A(N,1) = ?
A(N,2) = ?
B(1) = ?
B(N) = ?

What goes into the B.C. rows?

If T(1) = T_0 then first row is

(1,0,0...0) or A(1,2)=1, A(1,3)=0

because (1,0,0...0)$\cdot$(T(1),T (2),...)=T (1) so

B(1)=T_0

Or, if heat flux is given, using one-sided, first order difference

If $\delta T/\delta x(1) = Q$ then first row is

($-1/\Delta x$,$1/\Delta x$,0...0) or A(1,2)=$-1/\Delta x$, A(1,3)=$1/\Delta x$ and

B(1)=Q

because ($-1/\Delta x$,$1/\Delta x$,0...0)$\cdot$(T (1),T (2),...)= (T (2) -T (1))/Δx

Similarly for last row A(I,2)=1, A(I,3)=0, B(I)=T_1

Formally $\mathbf{A} \cdot \mathbf{T} = \mathbf{B}$ is solved: $\mathbf{T} = \mathbf{A}^{-1} \cdot \mathbf{B}$.

However, a*lmost never invert a matrix explicitly. Will review Gauss elimination*

3. Gauss elimination; Thomas algorithm for tri-diagonal matrices

a. Reminder on linear algebra. System of equations **A · x = b** <- includes b.c.
E.g.

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad \boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} a_{11}x_1 + a_{12}x_2 \\ a_{21}x_1 + a_{22}x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

Formal solution **x** = $\mathbf{A}^{-1}$ · **b**. In CFD **A** might be 10^6x10^6 = # grid points. $\mathbf{A}^{-1}$ very expensive to compute; rarely done. But, don't have to find $\mathbf{A}^{-1}$, just solve equations.

Will see: for large systems even this is done approximately. Gauss elimination is an exact method.

b. Reminder on Gauss elimination

$$\text{e.g.} \qquad \begin{aligned} x_1 + x_2 + x_3 &= 0 \\ 2x_1 + 4x_2 + 3x_3 &= 1 \\ 2x_1 + 2x_2 + x_3 &= 2 \end{aligned}$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 4 & 3 \\ 2 & 2 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

Convert to lower triangular form **A** **b**

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 4 & 3 \\ 2 & 2 & 1 \end{pmatrix} \qquad \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 0 \\ 4 & 2 & 0 \\ 2 & 2 & 1 \end{pmatrix} \qquad \begin{pmatrix} 2 \\ 5 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 0 & 0 \\ 4 & 2 & 0 \\ 2 & 2 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 \\ 5 \\ 2 \end{pmatrix}$$

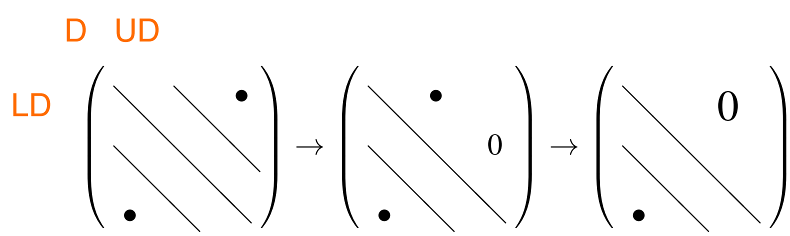
Back substitute

$$
\begin{aligned}
2x_1 &= 1 & &\rightarrow x_1 = 1/2 \\
4x_1 + 2x_2 &= 5 & &\rightarrow x_2 = 3/2 \\
2x_1 + 2x_2 + x_3 &= 2 & &\rightarrow x_3 = -2
\end{aligned}
$$

Schematic of elimination step

$$
\begin{pmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{pmatrix} \rightarrow \begin{pmatrix} \bullet & \bullet & 0 \\ \bullet & \bullet & 0 \\ \bullet & \bullet & \bullet \end{pmatrix} \rightarrow \begin{pmatrix} \bullet & 0 & 0 \\ \bullet & \bullet & 0 \\ \bullet & \bullet & \bullet \end{pmatrix}
$$

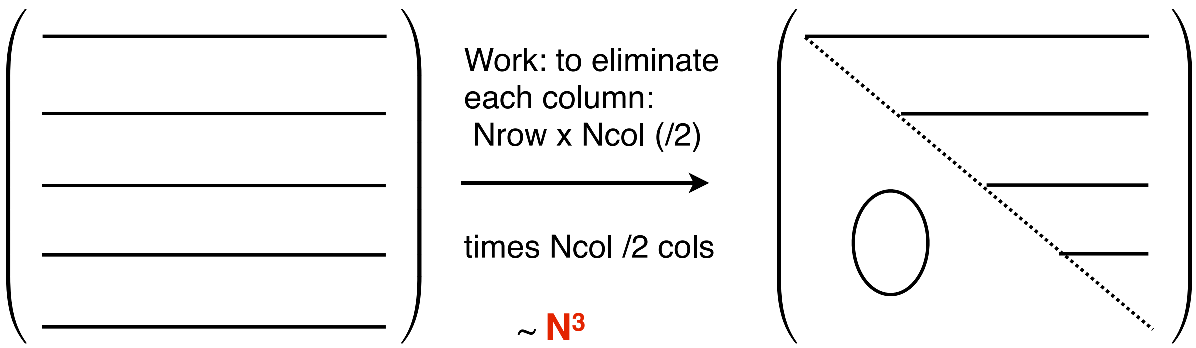
Or, sometimes think of it like this:



Main, upper and lower diagonals

Lower triangular (L)
(Upper triangular, U; c.f. LU)

c. Schematic of complexity (explain representation)



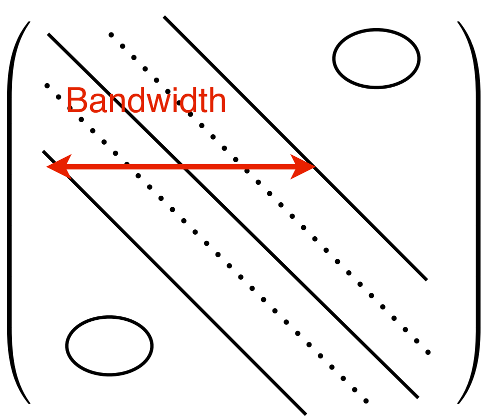
N^3 operations means 3 nested loops from 1 to N; i.e.,

```
DO nrow=1,N              ! Use the elements of current row
    DO i=nrow,N  rows    !  To eliminate whole column
    DO j=nrow,N   cols
        operate on A(i,j)
```

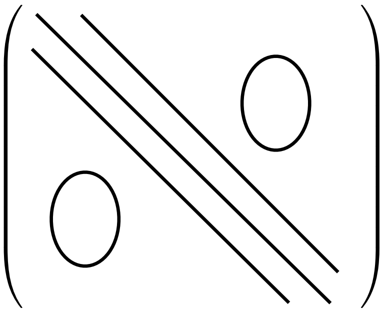
N x N/2 x N/2 ~ N^3 Hence work increases as size of matrix cubed: *computational complexity* in CFD, N may be *~10^6 So work ~10^{18} ops*

d. Simplifications: special form of matrix

Sparse - mostly 0's	< N^3
Banded - 0's outside diagonal band	N x BW^2
Tridiagonal	N (BW=3)



a. Thomas Algorithm for tridiagonal (handout posted on BlackBoard)



A_{ij}=0 for j≠ i±1

A(i,3) stores matrix

Solve

$$\begin{pmatrix} A_{12} & A_{13} & 0 & 0 & 0 & \dots \\ A_{21} & A_{22} & A_{23} & 0 & 0 & \dots \\ 0 & A_{31} & A_{32} & A_{33} & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & A_{N-1,1} & A_{N-1,2} & A_{N-1,3} \\ 0 & 0 & 0 & 0 & A_{N,1} & A_{N,2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_N \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_N \end{pmatrix}$$

by eliminating upper diagonal and back substituting.

```
!****************************************
 SUBROUTINE TDAG(a,b,N)
!****************************************
   REAL :: A(N,3),b(N),x(n)
!----
! eliminate A(*,3)
!----
 DO i=N-1,1,-1
    fac    = A(i,3)/A(i+1,2)
    A(i,2) = A(i,2)-fac*A(i+1,1)
    b(i)   = b(i)-fac*b(i+1)
 ENDDO
!----
! Now A is lower triangular.  Back substitution
!----
  x(1)     = b(1)/A(1,2)
 DO j=2,N
    x(j) = (b(j)-A(j,1)*x(j-1))/A(j,2)
 ENDDO

 RETURN
 END
```