- 4^{th}-order PADE

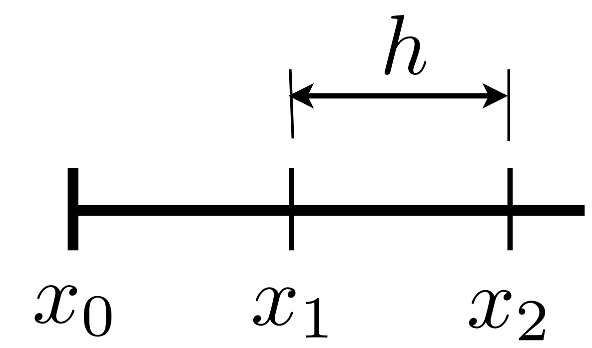
$$f'_{i-1} + 4f'_i + f'_{i+1} = \frac{3}{h}\left(f_{i+1} - f_{i-1}\right) + O(h^4)$$

  - only 2 slight problems:
    - to get f'_i, we need f'_{i-1} and f'_{i+1} ⇒ coupled system ⇒ implicit
    - to get f'_0, we need f'_{-1}, and to get f'_N we need f'_{N+1} ⇒ N-1 equations for N+1 unknowns!
  - Solution: apply difference formula of lower order at boundaries using only “inner” points
  - Why lower order?

- Example 6: left boundary

$$f_0' + a_0 f_0 + a_1 f_1 + a_2 f_2 + a_3 f_1' = O(?)$$

stencil:



- Following procedure results in

$$f_0' + 2f_1' = \frac{1}{h}\left(-\frac{5}{2}f_0 + 2f_1 + \frac{1}{2}f_2\right) + O(h^3)$$

- similar for right boundary

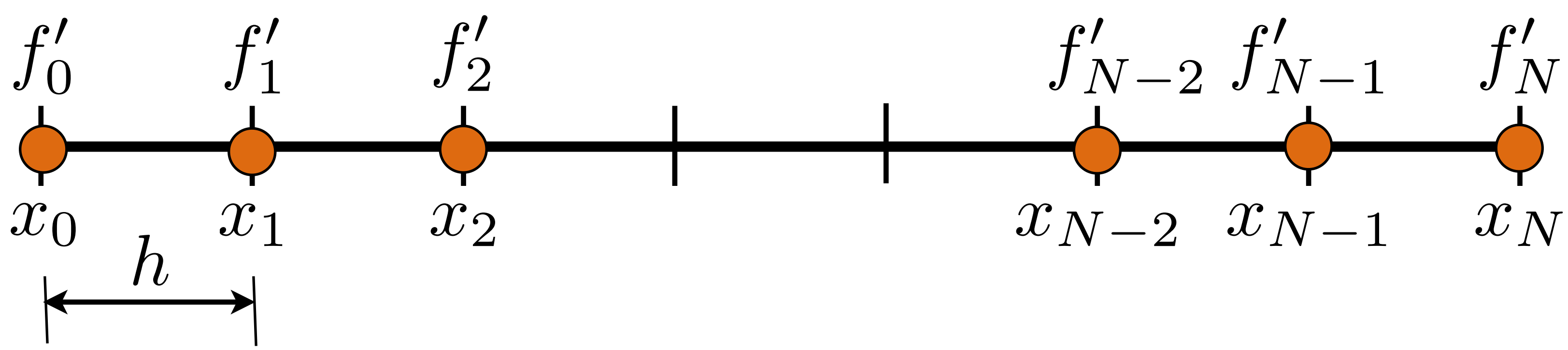
$$2f_{N-1}' + f_N' = \frac{1}{h}\left(\frac{5}{2}f_N - 2f_{N-1} - \frac{1}{2}f_{N-2}\right) + O(h^3)$$

- recap: in the interior we had

$$f_{i-1}' + 4f_i' + f_{i+1}' = \frac{3}{h}\left(f_{i+1} - f_{i-1}\right) + O(h^4)$$

- So what do we have now?

$$\begin{bmatrix} 1 & 2 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 1 & 4 & 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & 4 & 1 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 1 & 4 & 1 & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 & 4 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} f_0' \\ f_1' \\ f_2' \\ \vdots \\ f_{N-2}' \\ f_{N-1}' \\ f_N' \end{bmatrix} = \frac{1}{h} \begin{bmatrix} -\frac{5}{2} f_0 + 2 f_1 + \frac{1}{2} f_2 \\ 3\left(f_2 - f_0\right) \\ 3\left(f_3 - f_1\right) \\ \vdots \\ 3\left(f_{N-1} - f_{N-3}\right) \\ 3\left(f_N - f_{N-2}\right) \\ \frac{5}{2} f_N - 2 f_{N-1} - \frac{1}{2} f_{N-2} \end{bmatrix}$$



$$f_0' + 2 f_1' = \frac{1}{h}\left(-\frac{5}{2} f_0 + 2 f_1 + \frac{1}{2} f_2\right)$$

$$f_0' + 4 f_1' + f_2' = \frac{3}{h}\left(f_2 - f_0\right)$$

$$f_1' + 4 f_2' + f_3' = \frac{3}{h}\left(f_3 - f_1\right)$$

$$f_{N-3}' + 4 f_{N-2}' + f_{N-1}' = \frac{3}{h}\left(f_{N-1} - f_{N-3}\right)$$

$$f_{N-2}' + 4 f_{N-1}' + f_N' = \frac{3}{h}\left(f_N - f_{N-2}\right)$$

$$2 f_{N-1}' + f_N' = \frac{1}{h}\left(\frac{5}{2} f_N - 2 f_{N-1} - \frac{1}{2} f_{N-2}\right)$$

$$
\begin{bmatrix}
1 & 2 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
1 & 4 & 1 & 0 & \cdots & \cdots & \cdots & 0 \\
0 & 1 & 4 & 1 & \cdots & \cdots & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & \cdots & \cdots & 1 & 4 & 1 & 0 \\
0 & \cdots & \cdots & \cdots & 0 & 1 & 4 & 1 \\
0 & \cdots & \cdots & \cdots & 0 & 0 & 2 & 1
\end{bmatrix}
\begin{bmatrix}
f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N
\end{bmatrix}
= \frac{1}{h}
\begin{bmatrix}
-\frac{5}{2} f_0 + 2 f_1 + \frac{1}{2} f_2 \\
3\left(f_2 - f_0\right) \\
3\left(f_3 - f_1\right) \\
\vdots \\
3\left(f_{N-1} - f_{N-3}\right) \\
3\left(f_N - f_{N-2}\right) \\
\frac{5}{2} f_N - 2 f_{N-1} - \frac{1}{2} f_{N-2}
\end{bmatrix}
$$

- So what do we have now? A tri-diagonal system!
- Was this inevitable? No, it depends on the choice of boundary scheme!
- Recipe:
  - Choose a boundary scheme such that
    a) matrix form is preserved (here tri-diagonal)
    b) boundary stencil ≤ 1^{st} interior stencil (reasons later)
- Does the lower order @ boundary not pollute the higher order in the interior?
  - Depends on the case. In most cases, the additional error remains @ or near the boundary
- How do we solve the resulting tri-diagonal system?

How to solve a tri-diagonal system?

- Gaussian elimination (direct solve)

1) NEVER store the entire matrix !!!

- instead: store the 3 diagonals in separate vectors (1D arrays)

$$\begin{bmatrix} b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ a_1 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & a_2 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_{N-2} & b_{N-2} & c_{N-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & a_N & b_N \end{bmatrix} \begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix}$$

- store right hand side in vector d_i with $i = 0 \ldots N$

1) NEVER store the entire matrix !!!

- instead: store the 3 diagonals in separate vectors (1D arrays)

$$\begin{bmatrix} b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ a_1 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & a_2 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_{N-2} & b_{N-2} & c_{N-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & a_N & b_N \end{bmatrix} \begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix}$$

- store right hand side in vector d_i with $i = 0 \ldots N$
- store main diagonal in vector b_i with $i = 0 \ldots N$

1) NEVER store the entire matrix !!!

- instead: store the 3 diagonals in separate vectors (1D arrays)

$$\begin{bmatrix} b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ a_1 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & a_2 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_{N-2} & b_{N-2} & c_{N-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & a_N & b_N \end{bmatrix} \begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix}$$

- store right hand side in vector d_i with $i = 0 \ldots N$
- store main diagonal in vector b_i with $i = 0 \ldots N$
- store lower diagonal in vector a_i with $i = 1 \ldots N$?

a bit impractical! extend vector to $i = 0 \ldots N$ and skip $i = 0$

1) NEVER store the entire matrix !!!

- instead: store the 3 diagonals in separate vectors (1D arrays)

$$
\begin{bmatrix}
b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
a_1 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & a_2 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & a_{N-2} & b_{N-2} & c_{N-2} & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & a_N & b_N
\end{bmatrix}
\begin{bmatrix}
f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N
\end{bmatrix}
=
\begin{bmatrix}
d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N
\end{bmatrix}
$$

- store right hand side in vector d_i with $i = 0 \dots N$
- store main diagonal in vector b_i with $i = 0 \dots N$
- store lower diagonal in vector a_i with $i = 0 \dots N$

1) NEVER store the entire matrix !!!

- instead: store the 3 diagonals in separate vectors (1D arrays)

$$
\begin{bmatrix}
b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
a_1 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & a_2 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & a_{N-2} & b_{N-2} & c_{N-2} & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & a_N & b_N
\end{bmatrix}
\begin{bmatrix}
f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N
\end{bmatrix}
=
\begin{bmatrix}
d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N
\end{bmatrix}
$$

- store right hand side in vector d_i with $i = 0 \ldots N$
- store main diagonal in vector b_i with $i = 0 \ldots N$
- store lower diagonal in vector a_i with $i = 0 \ldots N$
- store upper diagonal in vector c_i with $i = 0 \ldots N-1$?

again extend vector to $i = 0 \ldots N$ and skip $i = N$

1) NEVER store the entire matrix !!!

- instead: store the 3 diagonals in separate vectors (1D arrays)

$$\begin{bmatrix} b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ a_1 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & a_2 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_{N-2} & b_{N-2} & c_{N-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & a_N & b_N \end{bmatrix} \begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix}$$

- store right hand side in vector d_i with $i = 0 \ldots N$
- store main diagonal in vector b_i with $i = 0 \ldots N$
- store lower diagonal in vector a_i with $i = 0 \ldots N$
- store upper diagonal in vector c_i with $i = 0 \ldots N$

2) Elimination

$$
\begin{bmatrix}
b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
a_1 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & a_2 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & a_{N-2} & b_{N-2} & c_{N-2} & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & a_N & b_N
\end{bmatrix}
\begin{bmatrix}
f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N
\end{bmatrix}
=
\begin{bmatrix}
d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N
\end{bmatrix}
$$

- 1st pivot b_0: subtraction multiplier: a_1/b_0

$$
b_1 \to b_1 - c_0\frac{a_1}{b_0} \qquad d_1 \to d_1 - d_0\frac{a_1}{b_0} \qquad (c_1 \to c_1 \qquad a_1 \to 0)
$$

2) Elimination

$$\begin{bmatrix} b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & a_2 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_{N-2} & b_{N-2} & c_{N-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & a_N & b_N \end{bmatrix} \begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix}$$

- 1st pivot b_0: subtraction multiplier: a_1/b_0

$$b_1 \to b_1 - c_0 \frac{a_1}{b_0} \qquad d_1 \to d_1 - d_0 \frac{a_1}{b_0} \qquad (c_1 \to c_1 \qquad a_1 \to 0)$$

- 2nd pivot b_1: subtraction multiplier: a_2/b_1

$$b_2 \to b_2 - c_1 \frac{a_2}{b_1} \qquad d_2 \to d_2 - d_1 \frac{a_2}{b_1} \qquad (c_2 \to c_2 \qquad a_2 \to 0)$$

2) Elimination

$$
\begin{bmatrix}
b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & a_{N-2} & b_{N-2} & c_{N-2} & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & a_N & b_N
\end{bmatrix}
\begin{bmatrix}
f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N
\end{bmatrix}
=
\begin{bmatrix}
d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N
\end{bmatrix}
$$

- 1st pivot b_0: subtraction multiplier: a_1/b_0

$$b_1 \to b_1 - c_0 \frac{a_1}{b_0} \qquad d_1 \to d_1 - d_0 \frac{a_1}{b_0} \qquad (c_1 \to c_1 \qquad a_1 \to 0)$$

- 2nd pivot b_1: subtraction multiplier: a_2/b_1

$$b_2 \to b_2 - c_1 \frac{a_2}{b_1} \qquad d_2 \to d_2 - d_1 \frac{a_2}{b_1} \qquad (c_2 \to c_2 \qquad a_2 \to 0)$$

- ith pivot b_{i-1}: subtraction multiplier: a_i/b_{i-1}

$$b_i \to b_i - c_{i-1} \frac{a_i}{b_{i-1}} \qquad d_i \to d_i - d_{i-1} \frac{a_i}{b_{i-1}} \qquad (c_i \to c_i \qquad a_i \to 0)$$

2) Elimination

$$\begin{bmatrix} b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & b_{N-2} & c_{N-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & a_N & b_N \end{bmatrix} \begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix}$$

- 1^{st} pivot b_0: subtraction multiplier: a_1/b_0

$$b_1 \to b_1 - c_0 \frac{a_1}{b_0} \qquad d_1 \to d_1 - d_0 \frac{a_1}{b_0} \qquad (c_1 \to c_1 \qquad a_1 \to 0)$$

- 2^{nd} pivot b_1: subtraction multiplier: a_2/b_1

$$b_2 \to b_2 - c_1 \frac{a_2}{b_1} \qquad d_2 \to d_2 - d_1 \frac{a_2}{b_1} \qquad (c_2 \to c_2 \qquad a_2 \to 0)$$

- i^{th} pivot b_{i-1}: subtraction multiplier: a_i/b_{i-1}

$$b_i \to b_i - c_{i-1} \frac{a_i}{b_{i-1}} \qquad d_i \to d_i - d_{i-1} \frac{a_i}{b_{i-1}} \qquad (c_i \to c_i \qquad a_i \to 0)$$

- N^{th} pivot b_{N-1}: subtraction multiplier: a_N/b_{N-1}

$$b_N \to b_N - c_{N-1} \frac{a_N}{b_{N-1}} \qquad d_N \to d_N - d_{N-1} \frac{a_N}{b_{N-1}} \qquad (c_N \to c_N \qquad a_N \to 0)$$

2) Elimination

$$\begin{bmatrix} b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & b_{N-2} & c_{N-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & b_N \end{bmatrix} \begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix}$$

- i^{th} pivot b_{i-1}: subtraction multiplier: a_i/b_{i-1}

$$b_i \rightarrow b_i - c_{i-1}\frac{a_i}{b_{i-1}} \qquad d_i \rightarrow d_i - d_{i-1}\frac{a_i}{b_{i-1}} \qquad (c_i \rightarrow c_i \qquad a_i \rightarrow 0)$$

```
loop from i = 1 to N   # loop over rows to eliminate
   b(i) = b(i) - c(i-1)*a(i)/b(i-1)
   d(i) = d(i) - d(i-1)*a(i)/b(i-1)
end loop i
```

3) Back-Substitution

$$\begin{bmatrix} b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & b_{N-2} & c_{N-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & b_N \end{bmatrix} \begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix}$$

- last row:

$$d_N \to \frac{d_N}{b_N} \qquad (b_N \to 1)$$

3) Back-Substitution

$$
\begin{bmatrix}
b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 0 & b_{N-2} & c_{N-2} & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & b_{N-1} & c_{N-1} \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N
\end{bmatrix}
=
\begin{bmatrix}
d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N
\end{bmatrix}
$$

- last row:

$$d_N \to \frac{d_N}{b_N} \quad (b_N \to 1)$$

3) Back-Substitution

$$\begin{bmatrix} b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & b_{N-2} & c_{N-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix}$$

- last row:

$$d_N \to \frac{d_N}{b_N} \quad (b_N \to 1)$$

- row N-1:

3) Back-Substitution

$$\begin{bmatrix} b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & b_{N-2} & c_{N-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & b_{N-1} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix}$$

- last row:

$$d_N \to \frac{d_N}{b_N} \quad (b_N \to 1)$$

- row N-1:

$$d_{N-1} \to (d_{N-1} - c_{N-1} d_N) \qquad (c_{N-1} \to 0 \qquad)$$

3) Back-Substitution

$$\begin{bmatrix} b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & b_{N-2} & c_{N-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix}$$

- last row:

$$d_N \rightarrow \frac{d_N}{b_N} \quad (b_N \rightarrow 1)$$

- row N-1:

$$d_{N-1} \rightarrow (d_{N-1} - c_{N-1} d_N) \frac{1}{b_{N-1}} \qquad (c_{N-1} \rightarrow 0 \qquad b_{N-1} \rightarrow 1)$$

3) Back-Substitution

$$
\begin{bmatrix}
b_0 & c_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & b_2 & c_2 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 0 & b_{N-2} & c_{N-2} & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N
\end{bmatrix}
=
\begin{bmatrix}
d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N
\end{bmatrix}
$$

- last row:

$$d_N \to \frac{d_N}{b_N} \quad (b_N \to 1)$$

- row N-1:

$$d_{N-1} \to (d_{N-1} - c_{N-1} d_N) \frac{1}{b_{N-1}} \qquad (c_{N-1} \to 0 \qquad b_{N-1} \to 1)$$

- ith row:

$$d_i \to (d_i - c_i d_{i+1}) \frac{1}{b_i} \qquad (c_i \to 0 \qquad b_i \to 1)$$

3) Back-Substitution

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix}$$

- last row:

$$d_N \to \frac{d_N}{b_N} \qquad (b_N \to 1)$$

- row N-1:

$$d_{N-1} \to (d_{N-1} - c_{N-1} d_N) \frac{1}{b_{N-1}} \qquad (c_{N-1} \to 0 \qquad b_{N-1} \to 1)$$

- i^{th} row:

$$d_i \to (d_i - c_i d_{i+1}) \frac{1}{b_i} \qquad (c_i \to 0 \qquad b_i \to 1)$$

- 1^{st} row:

$$d_0 \to (d_0 - c_0 d_1) \frac{1}{b_0} \qquad (c_0 \to 0 \qquad b_0 \to 1)$$

3) Back-Substitution

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f'_0 \\ f'_1 \\ f'_2 \\ \vdots \\ \vdots \\ f'_{N-2} \\ f'_{N-1} \\ f'_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix}$$

- last row:

$$d_N \to \frac{d_N}{b_N} \quad (b_N \to 1)$$

- ith row:

$$d_i \to (d_i - c_i d_{i+1}) \frac{1}{b_i} \qquad (c_i \to 0 \qquad b_i \to 1)$$

```
d(N) = d(N)/b(N)
loop from i = N-1 to 0 backwards    # loop over rows
   d(i) = (d(i) - c(i)*d(i+1))/b(i)
end loop i
```

How to solve a tri-diagonal system?

- Gaussian elimination (direct solve)

1) Store diagonals in vectors

2) Elimination

```
loop from i = 1 to N   # loop over rows to eliminate
      b(i) = b(i) - c(i-1)*a(i)/b(i-1)
      d(i) = d(i) - d(i-1)*a(i)/b(i-1)
end loop i
```

3) Back-Substitution

```
d(N) = d(N)/b(N)
loop from i = N-1 to 0 backwards   # loop over rows
      d(i) = (d(i) - c(i)*d(i+1))/b(i)
end loop i
```

Some notes for this algorithm:

- destroys (overwrites) a, b, and d
- solution is in vector d
- implement as function/subroutine
- alternatives can be found in ‘Numerical Recipes’ online.