Chapter 2

The Finite Difference Method

CONTENTS

• Basics, Taylor expansions, examples
• Explicit formulas on equidistant grids: Method of undetermined coefficients
• Explicit formulas on equidistant grids: Difference Operators
• Variable coefficient treatment
• Explicit formulas on variable spacing grids – The Fornberg method
• Implicit Formulas; Compact schemes

(I) Difference Operators
(II) Method of Undetermined Coefficients:

– tridiagonal and pentadiagonal systems
– boundary conditions
• Application: Temperature distribution on a rod
  – The concepts of equivalent differential and difference equations

• Tridiagonal systems: Thomas algorithm

• Multi-dimensional formulas
  – examples of different stencils
  – mixed derivatives
  – variable coefficients

• Mappings and non-uniform meshes
The Finite Difference Method

* Basics - Taylor’s expansion (Euler 1708)

![Diagram of Taylor’s expansion](image)

equidistant grid

**Expansions:** \( u_{i \pm 1} = u(x \pm \Delta x) = u(x) \pm \Delta x u_x(x) + \frac{\Delta x^2}{2} u_{xx}(x) \pm \ldots \)

- **forward:** \((u_x)_i = \frac{u_{i+1} - u_i}{\Delta x} + O(\Delta x)\)
- **backward:** \((u_x)_i = \frac{u_i - u_{i-1}}{\Delta x} + O(\Delta x)\) (upwind)
- **central:** \((u_x)_i = \frac{u_{i+1} - u_{i-1}}{2\Delta x} + O(\Delta x^2)\)

Notice that the forward/backward differences are \(O(\Delta x^2)\) approximation to derivatives at the half-point \((i \pm 1/2)\), respectively e.g

\[(u_x)_{i+1/2} = \frac{u_{i+1} - u_i}{\Delta x} + O(\Delta x^2)\]

Also, add expansions:

\[(u_{xx})_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} - \frac{1}{12} \frac{u_{xxxx}}{O(\Delta x^3)} \Delta x^2 + H.O.T.\]

These are **explicit formulas**, i.e., the derivative at a point is computed in terms of function values of neighboring points – no other derivatives involved unlike the implicit formulas.
Explicit Formulas - Equidistant grids

I. Method of Undetermined Coefficients

Example: One-sided, second-order differences for \( (u_x)_i \)

\[
(u_x)_i = \frac{au_i + bu_{i-1} + cu_{i-2}}{\Delta x} + \mathcal{O}(\Delta x^2)
\]

\begin{center}
\begin{tabular}{cccc}
\textbullet & \textbullet & \textbullet & \textbullet \\
\textbullet & \textbullet & \textbullet & \textbullet \\
\textbullet & \textbullet & \textbullet & \textbullet \\
\end{tabular}
\end{center}

(Phadom nodes)

Expand,

\[
c: u_{i-2} = u_i - 2\Delta x(u_x)_i + 2\Delta x^2 (u_{xx})_i - \frac{(2\Delta x)^3}{6} (u_{xxx})_i + \ldots
\]

\[
b: u_{i-1} = u_i - \Delta x(u_x)_i + \frac{\Delta x^2}{2} (u_{xx})_i - \frac{\Delta x^3}{6} (u_{xxx})_i + \ldots
\]

\[
a: u_i = u_i
\]

Then:

\[
(\Delta x)(u_x)_i = au_i + bu_{i-1} + cu_{i-2} + \mathcal{O}(\Delta x^3)
\]

\[
= (a + b + c)u_i - \Delta x(2c + b)(u_x)_i + \frac{\Delta x^2}{2}(4c + b)(u_{xx})_i + \mathcal{O}(\Delta x^3)
\]

Thus,

\[
\begin{cases}
a + b + c = 0 \\
2c + b = -1 \\
4c + b = 0
\end{cases}
\Rightarrow a = 3/2, b = -2, c = 1/2
\]

\[
(u_x)_i = \frac{3u_i - 4u_{i-1} + u_{i-2}}{2\Delta x} + \mathcal{O}(\Delta x^2)
\]

In general, a first-order derivative at mesh point \( i \) can be made of order of accuracy \( p \) by an explicit formula involving \( (p + 1) \) points.
II. Difference Operators - Equidistant Grid

Definitions:
- displacement: \(E^n u_i \equiv u_{i+n}\)
- forward: \(\delta^+ u_i \equiv u_{i+1} - u_i \Rightarrow \delta^+ = E - 1\)
- backward: \(\delta^- u_i \equiv u_i - u_{i-1} \Rightarrow \delta^- = 1 - E^{-1}\)
- half-central: \(\delta u_i \equiv (u_{i+1/2} - u_{i-1/2}) \Rightarrow \delta = \frac{1}{2}(E - E^{-1})\)
- central: \(\tilde{\delta} u_i \equiv \frac{1}{2}(u_{i+1} - u_{i-1}) \Rightarrow \tilde{\delta} = \frac{1}{2}(E - E^{-1})\)
- average: \(\mu u_i \equiv \frac{1}{2}(u_{i+1/2} + u_{i-1/2}) \Rightarrow \mu = \frac{1}{2}(E^{1/2} + E^{-1/2})\)
- derivative: \(D u \equiv \partial u / \partial x\)

Symbolic manipulations, e.g. \(\delta^+ \delta^+ = (E - 1)(E - 1) = E^2 - 2E + 1\) also, \(\mu \tilde{\delta} = \tilde{\delta}\), etc.

Taylor expansion: \(u(x + \Delta x) = u(x) + \Delta x u_x + \frac{\Delta x^2}{2!} u_{xx} + \frac{\Delta x^3}{3!} u_{xxx} + \ldots\)

In operator form: \(E u(x) = \left[1 + \Delta x D + \frac{(\Delta x D)^2}{2!} + \frac{(\Delta x D)^3}{3!} + \ldots\right] u(x)\) or \(E u(x) = e^{\Delta x D} u(x) \Rightarrow E = e^{\Delta x D} \Rightarrow \Delta x D = \ln E\)

- **Forward differences**:
  \(\Delta x D = \ln(E) = \ln(1 + \delta^+)\) because \((\delta^+ = E - 1)\)
  \(= \delta^+ - \frac{\delta^{+2}}{2} + \frac{\delta^{+3}}{3} - \frac{\delta^{+4}}{4} + \ldots\),

where the first neglected term gives the truncation error. For example, if we keep two terms:

\[
\Delta x (Du)_i = \delta^+ u_i - \frac{\delta^{+2}}{2} u_i
\]

\[
= \left( u_{i+1} - u_i \right) - \frac{1}{2} (E^2 - 2E + 1) u_i
\]

\[
= \left( u_{i+1} - u_i \right) - \frac{1}{2} (u_{i+2} - 2u_{i+1} + u_i)
\]

Thus, we obtain a one-sided, second-order formula:

\[
Du_i = (u_x)_i = -3u_i + 4u_{i+1} - u_{i+2} \frac{2\Delta x}{2\Delta x} + \frac{\Delta x^2}{3} (u_{xxx})_i
\]

\[\text{truncation error}\]

- **Backwards differences**:
  \(\Rightarrow \Delta x D = \ln E = - \ln(1 - \delta^-)\)
or

\[
\ln E = \delta^2 + \frac{\delta^3}{2} + \frac{\delta^4}{3} + \frac{\delta^5}{4} + \ldots
\]

- **Central differences**: \( \delta u_i = u_{i+1/2} - u_{i-1/2} = (E^{1/2} - E^{-1/2})u_i \Rightarrow \)

\[
\Delta x D = 2 \sinh^{-1} \frac{\delta}{2} = 2 \left[ \frac{\delta}{2} - \frac{1}{2 \cdot 3} \left( \frac{\delta^2}{2} \right)^3 + \frac{1 \cdot 3}{2 \cdot 4 \cdot 5} \left( \frac{\delta^4}{2} \right)^5 + \ldots \right]
\]

Thus, \( \Delta x D = \delta - \frac{\delta^3}{24} + \frac{3\delta^5}{120} - \frac{5\delta^7}{720} + \ldots \). Keep first term only:

\[
Du_i = \frac{u_{i+1/2} - u_{i-1/2}}{\Delta x} = -\frac{\Delta x^2}{24} (u_{xxx})_i + \ldots , \text{ need values at } \Delta x \text{ half-points}
\]

For function values at the integer grid points we obtain:

\[
\bar{\delta} = \frac{1}{2} (E - E^{-1}) = \frac{1}{2} (e^{\Delta x D} - e^{-\Delta x D}) = \sinh(\Delta x D)
\]

\[
\Rightarrow \Delta x D = \sinh^{-1} \bar{\delta} = (\bar{\delta} - \frac{\bar{\delta}^3}{6} + \frac{3}{2 \cdot 4 \cdot 5} \bar{\delta}^5 + \ldots)
\]

**Second-order**: \( Du_i = \bar{\delta} / \Delta x = \frac{u_{i+1} - u_{i-1}}{2 \Delta x} - \frac{\Delta x^2}{6} (u_{xxx})_i + \ldots \)

**Fourth-order**: involves \( \bar{\delta}^3 \) and therefore a long stencil (7-point stencil) \((i \pm 3)\). We can do better:

\[
\mu^2 = 1 + \delta^2 / 4 \Rightarrow \mu(1 + \delta^2 / 4)^{-1/2} = 1
\]

\[
\Rightarrow \mu(1 - \frac{\delta^2}{8} + \frac{3\delta^4}{128} - \frac{5\delta^6}{1024} + \ldots) = 1
\]

But: \( \Delta x D = \delta - \frac{\delta^3}{24} + \frac{3\delta^5}{640} - \frac{5\delta^7}{7168} + \ldots \) from above. So multiply by

\[
1 \times [\Delta x D] = \left[ \delta - \delta^3 / 24 + \frac{3\delta^5}{640} - \frac{5\delta^7}{7168} \right] \times 1
\]

\[
= \mu(\delta - \frac{1}{3!} \delta^3 + \frac{2^2 2^2}{5!} \delta^5 - \ldots)
\]

\[
= \bar{\delta}(1 - \frac{\delta^2}{3!} + \frac{2^2}{5!} \delta^4 - \frac{2^2 3^2}{7!} \delta^6 + \ldots)
\]
Thus, 4\textsuperscript{th}-order \((u_x)_i = \frac{-u_{i+2} + 8u_{i+1} - 8u_{i-1} + u_{i-2}}{12\Delta x} + \frac{\Delta x^4}{30} \frac{\partial^5 u}{\partial x^5}\) on a 5-point stencil

\[
\begin{array}{cccccc}
\cdots & i-2 & i-1 & i & i+1 & i+2 & \cdots
\end{array}
\]

\textbf{Higher-Order Derivatives}

\textbf{Forward:}

\[
\left( \frac{\partial^n u}{\partial x^n} \right)_i = D^n u_i = \frac{1}{\Delta x^n} \ln(1 + \delta^+)^n u_i
\]

\[
= \frac{1}{\Delta x^n} \left[ \delta^{+n} - \frac{n}{2} \delta^{+(n+1)} + \frac{n(3n+5)}{24} \delta^{+(n+2)} - \frac{n(n+2)(n+3)}{48} \delta^{+(n+3)} + \ldots \right] u_i
\]

\textbf{Backward:}

\[
\left( \frac{\partial^n u}{\partial x^n} \right)_i = -\frac{1}{\Delta x^n} \ln(1 - \delta^-)^n u_i
\]

\[
= \frac{1}{\Delta x^n} (\delta^- + \frac{\delta^{-2}}{2} + \frac{\delta^{-3}}{3} + \ldots)^n u_i
\]

\[
= \frac{1}{\Delta x^n} \left[ \delta^{-n} + \frac{n}{2} \delta^{-(n+1)} + \frac{n(3n+5)}{24} \delta^{-(n+2)} + \frac{n(n+2)(n+3)}{48} \delta^{-(n+3)} + \ldots \right] u_i
\]

\textbf{Central:}

\[
D^n u_i = \left( \frac{2}{\Delta x} \sinh^{-1} \frac{\delta}{2} \right)^n u_i = \frac{1}{\Delta x^n} \left[ \delta - \frac{\delta^3}{24} + \frac{3\delta^5}{640} - \frac{5\delta^7}{7168} + \ldots \right]^n u_i
\]

\[
= \frac{1}{\Delta x^n} \left[ 1 - \frac{n}{24} \delta^2 + \frac{n}{64} \left( \frac{22 + 5n}{90} \right) \delta^4 - \frac{n}{45} \left( \frac{57 + 5n}{10} + \frac{(n-1)(n-2)}{35} \right) \delta^6 + \ldots \right] u_i
\]
\[ n = \text{even} \Rightarrow \text{function values at integer mesh points.} \]

**Example:** Second-order derivative

- **Forward:** \( (u_{xx})_i = \frac{1}{\Delta x^2} \left( \delta^{+2} - \delta^{+3} + \frac{11}{12} \delta^{+4} - \frac{5}{6} \delta^{+5} + \ldots \right) u_i \)

- **Backward:** \( (u_{xx})_i = \frac{1}{\Delta x^2} \left( \delta^{-2} - \delta^{-3} + \frac{11}{12} \delta^{-4} + \frac{5}{6} \delta^{-5} + \ldots \right) u_i \)

- **Central:** \( (u_{xx})_i = \frac{1}{\Delta x^2} \left( \delta^2 - \frac{\delta^4}{12} + \frac{\delta^6}{90} - \frac{\delta^8}{630} + \ldots \right) u_i \)

Note that central is of higher-order by maintaining first-term only.

**Example:** (keep two terms)

- **Forward:** \( (u_{xx})_i = \frac{1}{\Delta x^2} \left( 2u_i - 5u_{i+1} + 4u_{i+2} - u_{i+3} \right) + \frac{11}{12} \Delta x^2 \left( \frac{\partial^2 u}{\partial x^2} \right) \)

- **Backward:** \( (u_{xx})_i = \frac{1}{\Delta x^2} \left( 2u_i - 5u_{i-1} + 4u_{i-2} - u_{i-3} \right) - \frac{11}{12} \Delta x^2 \left( \frac{\partial^2 u}{\partial x^2} \right) \)

- **Central:** \( (u_{xx})_i = \frac{1}{12\Delta x^2} (-u_{i+2} + 16u_{i+1} - 30u_i + 16u_{i-1} - u_{i-2}) + \frac{\Delta x^4}{90} \left( \frac{\partial^4 u}{\partial x^4} \right) \)

- **Variable coefficient:**

\[
\frac{\partial}{\partial x} \left[ \nu(x) \frac{\partial}{\partial x} \right] u_i = \frac{1}{\Delta x^2} \delta^+(\nu_{i+1/2} \delta^+) u_i + O(\Delta x^2) = \frac{1}{\Delta x^2} \delta^+(\nu_{i-1/2} \delta^-) u_i + O(\Delta x^2)
\]

Thus,

\[
\frac{\partial}{\partial x} \left[ \nu(x) \frac{\partial}{\partial x} \right] u_i = \frac{\nu_{i+1/2}(u_{i+1} - u_i)}{\Delta x^2} - \frac{\nu_{i-1/2}(u_i - u_{i-1})}{\Delta x^2} + O(\Delta x^2)
\]
Explicit Formulas - Variable $\Delta x$

$$(N + 1) \text{ grid points}$$

\[ \alpha_0 \alpha_1 \alpha_2 \ldots x_0 \ldots \alpha_{N} \]

Consider the function $f(x)$ and look for approximation

\[
\begin{align*}
& n \rightarrow \text{stencil} & d^n f \bigg|_{x=x_0} &= \sum_{\nu=0}^{n} c_{m\nu}^{m} f(\alpha_{\nu}) \quad m = 0, 1, \ldots, M \\
& m \rightarrow \text{order} & n = m, m+1, \ldots, N
\end{align*}
\]

accuracy-order $(n-m+1)$

(1)

**Question:** How can we compute $c_{n\nu}^{m}$ efficiently?

Let us consider the point $x_0 = 0$, for simplicity. Define $F_{n}(x) \equiv \Pi_{k=0}^{n}(x-\alpha_{k})$ and thus the $n^{th}$-degree polynomial $h_{\nu}^{n}(x) \equiv \frac{F_{n}(x)}{F_{n}(\alpha_{\nu})(x-\alpha_{\nu})}$; $h_{\nu}^{n}(\alpha_{k}) = \delta_{\nu k}$. Then Lagrangian interpolation for $f(x)$ is

\[
f(x) \approx \sum_{\nu=0}^{n} h_{\nu}^{n}(x) f(\alpha_{\nu})
\]

(2)

Thus, compare (1) and $m^{th}$-derivative of (2) we obtain

\[
c_{n\nu}^{m} = \frac{d^{m}}{dx^{m}} h_{\nu}^{n}(x) \bigg|_{x=0}
\]

Thus, inversely the $n^{th}$-degree polynomial $h_{\nu}^{n}(x)$ can be expressed \textit{a la} Taylor:

\[
h_{\nu}^{n}(x) = \sum_{m=0}^{n} \frac{c_{n\nu}^{m}}{m!} x^{m}
\]

The next step is to obtain recurrence formulas for $h_{\nu}^{n}(x)$ so that they can be computed easily, and subsequently compute $c_{n\nu}^{m}$ recursively. To this end,

\[
F_{n}(x) = (x-\alpha_{n})F_{n-1}(x)
\]

\[
F_{n}'(x) = (x-\alpha_{n})F_{n-1}'(x) + F_{n-1}(x)
\]
Thus:

\[ h^\nu_n(x) = \frac{F_n(x)}{F_n(\alpha_\nu)(x-\alpha_\nu)} = \frac{(x-\alpha_n)F_{n-1}(x)}{(\alpha_\nu - \alpha_n)F'_{n-1}(\alpha_\nu)(x-\alpha_\nu)} \]

\[ = \frac{x-\alpha_n}{\alpha_\nu - \alpha_n} \frac{F_{n-1}(x)}{F'_{n-1}(\alpha_\nu)(x-\alpha_\nu)} \]

\[ = \frac{x-\alpha_n}{\alpha_\nu - \alpha_n} h_{\nu,n-1}(x), \]

\[ \nu \neq n: \quad h^n_n(x) = \frac{F_{n-1}(x)}{F_{n-1}(\alpha_n)} = \frac{F_{n-2}(\alpha_{n-1})}{F_{n-1}(\alpha_n)} (x-\alpha_{n-1}) h_{n,n-1}(x) \]

By equating coefficients in the expansion for \( h^\nu_n(x) \) we obtain:

\[ x_0 = 0 \]

\[ \begin{cases} 
  n \neq \nu: & c_{n,\nu} = \frac{1}{\alpha_n - \alpha_\nu} (\alpha_n c_{n-1,\nu} - mc_{n-1,\nu-1}) \\
  n = \nu: & c_{n,n} = \frac{F_{n-2}(\alpha_{n-1})}{F_{n-1}(\alpha_n)} (mc_{n-1,n-1} - \alpha_{n-1}c_{n-1,n-1}) 
\end{cases} \]

We can also use \( \sum_{\nu=0}^{n-1} c_{n,\nu} = \begin{cases} 
  1, m = 0 \\
  0, m \neq 0 
\end{cases} \) but it may induce round-off error.

Note that for \( x_0 \neq 0 \) we replace \( \begin{cases} 
  \alpha_n \rightarrow (\alpha_n - x_0) \\
  \alpha_{n-1} \rightarrow (\alpha_{n-1} - x_0) 
\end{cases} \)

There is no restriction on \( x_0 \) coinciding with any \( \alpha_\nu \).
II. Implicit Formula - Equidistant Grids

Locality/high-order accuracy

Use the same difference operators \( \Delta x D = \mu (\delta - \frac{\delta^3}{6x} + \frac{12\delta^5}{120x^5} - \ldots) \)

4\textsuperscript{th}-order accuracy:

\[
\Delta x D = \mu \delta (1 - \frac{\delta^2}{6}) + O(\Delta x^5)
\]

or \( \Delta x D \approx \frac{\mu \delta}{1 + \frac{\delta^2}{6}} + O(\Delta x^5) \), PADE approximation

Thus,

\[
(1 + \frac{\delta^2}{6})D = \frac{\mu \delta}{\Delta x} + O(\Delta x^4)
\]

But

\[
\mu \delta u_i = \mu [u_{i+1/2} - u_{i-1/2}] = \frac{1}{2} [u_{i+1} + u_i - u_i - u_{i-1}]
\]

so,

\[
\text{RHS} = \frac{1}{2} \frac{u_{i+1} - u_{i-1}}{\Delta x} + O(\Delta x^4)
\]

LHS:

\[
\left(1 + \frac{\delta^2}{6}\right)Du_i = \left( (u_x)_i + \frac{1}{6} (E + E^{-1} - 2) (u_x)_i \right)
\]

\[
= (u_x)_i + \frac{1}{6} [(u_x)_{i+1} + (u_x)_{i-1} - 2(u_x)_i]
\]

\[
= \frac{1}{6} [(u_x)_{i+1} + 4(u_x)_i + (u_x)_{i-1}]
\]

Thus, \( \frac{1}{6} [(u_x)_{i+1} + 4(u_x)_i + (u_x)_{i-1}] = \frac{u_{i+1} - u_{i-1}}{2\Delta x} + O(\Delta x^4) \)

So we have 3-pt stencil, 4\textsuperscript{th}-order:

\[ i - 1 \quad i \quad i + 1 \]

- General scheme:

\[
\beta(u_x)_{i-1} + \alpha(u_x)_i + \beta(u_x)_{i+1} = \frac{u_{i+1} - u_{i-1}}{2\Delta x} + O(\Delta x^p)
\]
or more extended stencil to match the order.

Note, contrast with the 4th-order explicit formula. (Locality)

\[
(u_x)_i = \frac{-u_{i+2} + 8u_{i+1} - 8u_{i-1} + u_{i-2}}{12\Delta x} + \frac{\Delta x^4 \partial^5 u}{30 \partial x^5}
\]

5-pt stencil

\[
\cdots i-3 \ i-2 \ i-1 \ i \ i+1 \ i+2 \ i+3 \ \cdots
\]

* 2-\text{pt implicit formula} \ [\cdots i \ i+1 \ \cdots]

2nd-order for \((u_x)_i\). Consider: \(\Delta x D = \ln E = \ln(1 + \delta^+ \cdot \text{or})\)

\[
\Delta x D = \delta^+ - \frac{\delta^{+2}}{2} + \frac{\delta^{+3}}{3} - \frac{\delta^{+4}}{4} + \cdots
\]

\[
= \delta^+ \left( 1 - \frac{\delta^+}{2} \right) + \mathcal{O}(\Delta x^3)
\]

\[
= \delta^+ \frac{1}{1 + \frac{\delta^+}{2}} + \mathcal{O}(\Delta x^3), \text{ geometric series}
\]

\[
(1 + \frac{\delta^+}{2}) D = \frac{\delta^+}{\Delta x} + \mathcal{O}(\Delta x^2)
\]

or

\[
\{ (u_x)_i + \frac{1}{2}[ (u_x)_{i+1} - (u_x)_i ] \} = \frac{u_{i+1} - u_i}{\Delta x} + \mathcal{O}(\Delta x^2)
\]

or

\[
\frac{1}{2}[(u_x)_i + (u_x)_i] = \frac{u_{i+1} - u_i}{\Delta x} + \mathcal{O}(\Delta x^2)
\]

bi-diagonal system

* Second-order derivatives, similarly using the formal expansion, e.g.,

\[
(u_{xx})_i = \frac{1}{\Delta x^2} \left[ \delta^2 - \frac{\delta^4}{12} + \frac{\delta^6}{90} - \frac{\delta^8}{560} + \cdots \right] u_i, \text{ (derived from } D^2 u_i = \cdots)
\]

\[
= \frac{\delta^2}{\Delta x^2} \left[ 1 - \frac{\delta^2}{12} \right] u_i + \mathcal{O}(\Delta x^4)
\]

\[
= \frac{1}{\Delta x^2} \frac{\delta^2 u_i}{1 + \delta^2/12} + \mathcal{O}(\Delta x^4)
\]
\[
\Rightarrow \left( 1 + \frac{\delta^2}{12} \right) (u_{xx})_i = \frac{1}{\Delta x^2} \delta^2 u_i + O(\Delta x^4)
\]
or
\[
\frac{1}{12} \left[ (u_{xx})_{i+1} + 10(u_{xx})_i + (u_{xx})_{i-1} \right] = \frac{1}{\Delta x^2} \left[ u_{i+1} - 2u_i + u_{i-1} \right] + O(\Delta x^4)
\]
Solve the tri-diagonal system \[
\frac{1}{12} \begin{bmatrix}
\ddots & \ddots & 0 \\
1 & 10 & 1 \\
0 & \ddots & \ddots 
\end{bmatrix} [(u_{xx})_i] = \text{known}
\]
General Derivation of Compact/Implicit Schemes: (Method of Undetermined Coefficients)

First-order derivative \((u_x)_i\):

\[
\begin{array}{cccccc}
\cdots & i-2 & i-1 & i & i+1 & i+2 & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
5\text{-pt stencil/LHS}
\end{array}
\]

\[
\begin{array}{cccccc}
c & b & a & \cdots & \cdots & \cdots \\
i-3 & i-2 & i-1 & i & i+1 & i+2 \cdots & i+3
\end{array}
\]

7\text{-pt stencil/RHS}

\[
\beta (u_x)_{i-2} + \alpha (u_x)_{i-1} + (u_x)_i + \alpha (u_x)_{i+1} + \beta (u_x)_{i+2} =
\]

\[
c \frac{u_{i+3} - u_{i-3}}{6\Delta x} + b \frac{u_{i+2} - u_{i-2}}{4\Delta x} + a \frac{u_{i+1} - u_{i-1}}{2\Delta x}
\]

We have, then, 5 unknowns: \(\{\alpha, \beta, c, b, a\}\)

Note: If \(\beta = 0\) \(\Rightarrow\) tri-diagonal sytem; \(\beta \neq 0\) \(\Rightarrow\) penta-diagonal.

To find unknowns, match Taylor expansion on \((u_x)_{i\pm 1}, u_{i\pm 1}\), etc; the first unmatched coefficient will give the truncation error.

Constraints:

\[
a + b + c = 1 + 2\alpha + 2\beta, \quad O(\Delta x^2)
\]

\[
a + 2^2b + 3^2c = 2 \frac{3!}{2!} (\alpha + 2^2 \beta), \quad O(\Delta x^4)
\]

\[
a + 2^4b + 3^4c = 2 \frac{5!}{4!} (\alpha + 2^4 \beta), \quad O(\Delta x^6)
\]

\[
a + 2^6b + 3^6c = 2 \frac{7!}{6!} (\alpha + 2^6 \beta), \quad O(\Delta x^8)
\]

\[
\vdots
\]
Tri-diagonal schemes ($\beta = 0$) ($5N$ operations)

\(\alpha\)-family

\[
\begin{array}{c}
\text{LHS} \\
\bullet \bullet \bullet \bullet \bullet \bullet \bullet \\
\end{array}
\]

\[
\begin{array}{c}
\text{RHS} \\
\bullet \bullet \bullet \bullet \bullet \bullet \\
\end{array}
\]

3/5 stencil: use 2 equations \(\mathcal{O}(\Delta x^4)\) \(\Rightarrow c = 0 \Rightarrow a = \frac{2}{3}(\alpha + 2), b = \frac{1}{3}(4\alpha - 1)\). Truncation error: \(T^t_{3/5} = \frac{4}{3\alpha}(3\alpha - 1)\Delta x^4 \left(\frac{\partial^3 u}{\partial x^3}\right)_i\)

Typical members of the \(\alpha\)-family:

- \(\alpha \to 0\), recover central difference (explicit)
- \(\alpha = 1/4 \Rightarrow b = 0\), 3/3 stencil, classical Padé scheme, \(\mathcal{O}(\Delta x^4)\)
- \(\alpha = 1/3\), recover an \(\mathcal{O}(\Delta x^6)\) scheme

3/7 stencil: Use 3 equations, 6th-order:

\[
\begin{array}{c}
\bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \\
\bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \\
\end{array}
\]

\[
\begin{array}{c}
i - 3 & i - 1 & i & i + 1 & i + 3 \\
\end{array}
\]

\[
a = \frac{1}{6}(\alpha + 9)
\]

\[
b = \frac{1}{15}(32\alpha - 9)
\]

\[
c = \frac{-3\alpha + 10}{10}
\]

We can obtain 8th-order \(\mathcal{O}(\Delta x^8)\) accuracy by setting \(\alpha = 3/8\) which zeroes out the truncation error. This scheme provides the highest order of the family.
**Penta-diagonal schemes** ($\beta \neq 0$) (11 $N$ operations) (LU + backsolve, multiplies only)

\[
c = 0, a = \frac{2}{9}(8 - 3\alpha), \beta = \frac{-1 + 3\alpha}{12}, b = \frac{-17 + 57\alpha}{18},
\]

\[
T'_{5/5} = \frac{4}{7!} (9\alpha - 4) \Delta x^6 \left( \frac{\partial^7 u}{\partial x^7} \right)
\]

**5/5 stencil:**

\[
\begin{array}{c}
\text{i} \\
\text{i + 2} \\
\end{array}
\]

**5/7 stencil:**

\[
\begin{array}{c}
\text{i} \\
\text{i + 2} \\
\text{i + 3} \\
\end{array}
\]

\[
a = \frac{12 - 7\alpha}{6}, b = \frac{568\alpha - 183}{150}, \beta = \frac{-3 + 8\alpha}{20}, c = \frac{9\alpha - 4}{50}
\]

$O(\Delta x^8)$

To obtain 10th-order: $T'_{5/7} = \frac{144}{9!} (2\alpha - 1) \Delta x^8 \frac{\partial^9 u}{\partial x^9}$ set $\alpha = 1/2$, highest-order in this family.

**Note:** the corresponding explicit formula would require an 11-point stencil!

**Second-Derivative** $(u_{xx})_i$

\[
\begin{array}{c}
\text{i - 2} \\
\text{i - 3} \\
\text{i} \\
\text{i + 3} \\
\end{array}
\]

\[
\text{LHS} \\
\text{RHS}
\]

\[
\beta (u_{xx})_{i-2} + \alpha (u_{xx})_{i-1} + (u_{xx})_i + \alpha (u_{xx})_{i+1} + \beta (u_{xx})_{i+2}
\]
\[
\begin{align*}
&= a \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} + b \frac{u_{i+2} - 2u_i + u_{i-2}}{4\Delta x^2} + c \frac{u_{i+3} - 2u_i + u_{i-3}}{9\Delta x^2} \\
\text{Constraints:} \\
&\quad a + b + c = 1 + 2\alpha + 2\beta, \quad \mathcal{O}(\Delta x^2) \\
&\quad a + 2^2b + 3^2c = \frac{4!}{2!}(\alpha + 2^2\beta), \quad \mathcal{O}(\Delta x^4) \\
&\quad a + 2^4b + 3^4c = \frac{6!}{4!}(\alpha + 2^4\beta), \quad \mathcal{O}(\Delta x^6) \\
&\quad a + 2^6b + 3^6c = \frac{8!}{6!}(\alpha + 2^6\beta), \quad \mathcal{O}(\Delta x^8) \\
&\quad a + 2^8b + 3^8c = \frac{10!}{8!}(\alpha + 2^8\beta), \quad \mathcal{O}(\Delta x^{10}) \\
&\quad \vdots \\
\text{General: } a + 2^{p-2}b + 3^{p-2}c = \frac{p!}{(p-2)!}(\alpha + 2^{p-2}\beta), \quad \mathcal{O}(\Delta x^p)
\end{align*}
\]

- 3/5 stencil \(\Rightarrow c = \beta = 0\): \(a = \frac{4}{3}(1-\alpha), \quad b = \frac{1}{3}(-1+10\alpha), \quad \mathcal{O}(\Delta x^4), \quad T_{3/5}'' = -\frac{4}{6}(11\alpha - 2)\Delta x^4 \frac{\partial^2 u}{\partial x^2}\)

[\(\alpha \to 0, \text{ central; } \alpha = 1/10 \to 3/3 \text{ classical Padé; } a = \frac{2}{11} \Rightarrow \mathcal{O}(\Delta x^6)\)]

- 5/7 stencil: A 2-parameter family: \(\mathcal{O}(\Delta x^6)\)

\[
T_{5/7}'' = -\frac{8}{8!}(9 - 38\alpha + 214\beta)\Delta x^6 \frac{\partial^3 u}{\partial x^3} \begin{cases}
\begin{aligned}
a &= \frac{6 - 9\alpha - 12\beta}{4} \\
b &= \frac{-3 + 24\alpha - 6\beta}{5} \\
c &= \frac{2 - 11\alpha + 124\beta}{20}
\end{aligned}
\end{cases}
\]
Require $\mathcal{O}(\Delta x^8)$ (4 constraints; 5 unknowns 1-parameter ($\alpha$):

\[
\beta = \frac{38\alpha - 9}{214}; \quad \alpha = \frac{696 - 1191\alpha}{428}; \quad b = \frac{2454\alpha - 294}{530}
\]

\[
T''_{5/7} = \frac{899\alpha - 334}{2696400} \Delta x^8 \frac{\partial^{10} u}{\partial x^{10}}
\]

We obtain $\mathcal{O}(\Delta x^{10})$ for $\alpha = 334/899$.

**Third-Dervative**

\[
\begin{array}{c c c}
\bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet
\end{array}
\]

\[
\alpha(u_{xxx})_{i-3} + (u_{xxx})_i + \alpha(u_{xxx})_{i+1} = b^* \frac{u_{i+3} - 3u_{i+1} + 3u_{i-1} - u_{i-3}}{8\Delta x^3}
\]

\[
+ a^* \frac{u_{i+2} - 2u_{i+1} + 2u_{i-1} - u_{i-2}}{2\Delta x^3}
\]

Constraints: $a = 2, b = 2\alpha - 1, T''_{3/7} = \frac{42}{11}(16\alpha - 7)\Delta x^4 \frac{\partial^7 u}{\partial x^7}$.

- For $\alpha = 1/2 \implies b = 0 \implies 3/5$ most compact $\mathcal{O}(\Delta x^4)$ scheme
- For $\alpha = 7/16 \implies \mathcal{O}(\Delta x^6)$, highest possible accuracy.

* Note: • 2\textsuperscript{nd}-order accuracy/integer pts: $(u_{xxx})_i = \frac{1}{2\Delta x^3}(u_{i+2} - 2u_{i+1} + 2u_{i-1} - u_{i-2}) - \frac{1}{4}\Delta x^2(u^{(iv)})$  
  
  * 2\textsuperscript{nd}-order accuracy/(1/2) pts: $(u_{xxx})_i = \frac{1}{\Delta x^3}(u_{i+3/2} - 3u_{i+1/2} + 3u_{i-1/2} - u_{i-3/2}) - \frac{\Delta x^2}{8}(u^{(v)})$
Fourth-order Derivative

- 3/7 stencil:

\[
\begin{align*}
\alpha u_{i-1}^{iv} + u_i^{iv} + \alpha u_{i+1}^{iv} &= b \frac{u_{i+3} - 9u_{i+1} + 16u_i - 9u_{i-1} + u_{i-3}}{6 \Delta x^4} \\
&+ a \frac{u_{i+2} - 4u_{i+1} + 6u_i - 4u_{i-1} + u_{i-2}}{\Delta x^4}
\end{align*}
\]

- For \( a = 2(1 - \alpha) \) and \( b = 4\alpha - 1 \Rightarrow \mathcal{O}(\Delta x^4) \)
- For \( \alpha = 1/4 \Rightarrow b = 0 \Rightarrow \) compact
- For \( \alpha = 7/26 \Rightarrow \mathcal{O}(\Delta x^6) \)

Boundary Conditions:

\[
\begin{array}{c}
\text{left boundary} \\
\hline
1 & 2 & 3 & 4 & 5
\end{array}
\]

First-Derivative: \( (u_x)_1 + \alpha (u_x)_2 = \frac{1}{\Delta x} (au_1 + bu_2 + cu_3 + du_4) \)

- \( \mathcal{O}(\Delta x^2) \): \( a = -\frac{3+\alpha+2d}{2}, b = 2 + 3d, c = \frac{1-\alpha+6d}{2} \)
  truncation: \( \sim \frac{2-\alpha-6d}{3} \Delta x^2 \left( \frac{\partial^2 u}{\partial x^2} \right)_1 \)

- \( \mathcal{O}(\Delta x^3) \): \( a = -\frac{11+2\alpha}{6}, b = \frac{6-\alpha}{2}, c = \frac{2\alpha-3}{2}, d = \frac{2-\alpha}{6} \)

- \( \mathcal{O}(\Delta x^4) \): \( \alpha = 3, a = -17/6, b = 3/2, d = -1/6 \)
Note: $T_1 \sim \Delta x^p u_1^{(p+1)}$. If $p$ = even, dispersive errors; $p$ = odd, dissipative errors.

Second-Derivative: $(u_{xx})_1 + 11(u_{xx})_2 = \frac{1}{\Delta x^2} (13u_1 - 27u_2 + 15u_3 - u_4) + \frac{\Delta x^3}{12} \left( \frac{\partial^3 u}{\partial x^3} \right)_1$

The corresponding explicit one-sided formula: $(u_{xx})_1 = \frac{1}{\Delta x^2} \left( \frac{35}{12} u_1 - \frac{26}{3} u_2 + \frac{10}{7} u_3 - \frac{14}{3} u_4 + \frac{11}{12} u_5 \right) + \frac{5}{3} \Delta x^3 \left( \frac{\partial^3 u}{\partial x^3} \right)_1$. Thus, the corresponding truncation error is 10 times larger than that of the 3rd-order compact scheme!

Application: Temperature Distribution in a Rod:

\[
\begin{array}{cccccc}
0 & 1 & 2 & 3 & \cdots & N \\
\bullet & \bullet & \bullet & \bullet & \bullet \\
\theta_0 & \theta_1 & \theta_2 & \theta_3 & \cdots & \theta_N
\end{array}
\]

\[
\begin{cases}
\theta'' = q(x) & \text{steady state} \\
\theta_0 = \theta_N = 0
\end{cases}
\]

Second-order, $O(\Delta x^2)$ differencing: \(\frac{1}{\Delta x^2} (\theta_{i+1} - 2\theta_i + \theta_{i-1}) = q_i\), where \(i = 1, \ldots, N-1\). In matrix form:

\[
\begin{pmatrix}
1 & 1/\Delta x^2 & & & \\
1/\Delta x^2 & -2/\Delta x^2 & 1/\Delta x^2 & & \\
& \ddots & \ddots & \ddots & \\
0 & 1/\Delta x^2 & -2/\Delta x^2 & & 1
\end{pmatrix}
\begin{bmatrix}
\theta_0 \\
\theta_1 \\
\vdots \\
\theta_{N-1} \\
\theta_N
\end{bmatrix}
= 
\begin{bmatrix}
q_0 \\
q_1 \\
\vdots \\
q_{N-1} \\
q_N
\end{bmatrix}
\]

\(\begin{pmatrix}
A \\
\theta
\end{pmatrix} = \begin{pmatrix}
q
\end{pmatrix}\); \(A\) sparse

Question: Is the inverse of \(A\) a sparse matrix? Explain

Multiply by the columns \([a_i]\) of \(A\), i.e. \(\sum_{i=0}^{N-1} \theta_i [a_i] = [q]\). The solution is \([\theta] = [A]^{-1} [q]\) and therefore \([\theta] = \sum_{i=0}^{N} q_i \left[a_i^{-1}\right]\), where \([a_i^{-1}]\) are the
columns of $[A]^{-1}$. Now consider that we apply a discrete unit heat source at node $(i)$ but nowhere else, then: $[\theta] = [a_i^{-1}]$. In other words the elements of the column-vector $[a^{-1}]$ are the temperature at the nodes due to a unit heat source at node $i$, thus they should be $\neq 0$ (ellipticity, zero at end points).

Note: matrix condensation also to impose Dirichlet b.c. i.e., $\sum_{i=1}^{N-1} \theta_i [a_i] = [q] - \theta_0[a_0] - \theta_N[a_N]$

**Compact scheme:** 3/3 stencil

$$
\begin{align*}
\bullet & \theta \\
\bullet & \quad i - 1 \quad i \quad i + 1 \quad \bullet \\
\end{align*}$$

$q$, equidistant

$$\theta_{i \pm 1} = \theta_i \pm \Delta x^2 \theta_i' + \frac{\Delta x^2}{2} \theta_i'' + \frac{\Delta x^4}{24} \theta_i^{iv} + \ldots$$

$$\alpha \theta_{i-1} + \beta \theta_i + \alpha \theta_{i+1} = a q_{i-1} + b q_i + a q_{i+2} + T_i$$

But $\theta'' = q \Rightarrow q_{i \pm 1} = q_i \pm \Delta x q_i' + \frac{\Delta x^2}{2} q_i'' + \mathcal{O}(\Delta x^4)$, etc. Thus,

$$\alpha = \frac{1}{\Delta x^2}, \beta = -2/\Delta x^2 \quad \alpha = \frac{1}{12}, b = \frac{5}{6}$$

Thus, $\frac{1}{\Delta x^2}(\theta_{i-1} - 2\theta_i + \theta_{i+1}) = \frac{1}{12}(q_{i-1} + 10 q_i + q_{i+1}) + \mathcal{O}(\Delta x^4)$. Modify RHS through a mass matrix $\rightarrow$ distribute the forcing similarly as in FEM:

$$
\text{RHS} : \frac{1}{12} \begin{bmatrix}
\vdots & \vdots \\
1 & 10 & 1 \\
\vdots & \vdots \\
\hline \\
\text{Mass Matrix} \\
\end{bmatrix} \begin{bmatrix}
\vdots \\
q_{i-1} \\
q_i \\
q_{i+2} \\
\vdots \\
\end{bmatrix}
$$
This is the same as the most compact scheme of $\mathcal{O}(\Delta x^4)$ for $\alpha = 1/10$ i.e.,

$$\frac{1}{10} (\theta_{xx})_{i-1} + (\theta_{xx})_i + \frac{1}{10} (\theta_{xx})_{i+1} = \frac{12 \theta_{i+1} - 2 \theta_i + \theta_{i+1}}{10 \Delta x^2} + \mathcal{O}(\Delta x^4)$$

- Application: 1D heat conduction

$$\begin{cases} 
\theta'' = -\sin 2\pi x \\
\theta_0 = \theta_N = 0 & \text{Differential Equation}
\end{cases}$$

$$\Theta(x) = \frac{1}{4\pi^2} \sin 2\pi x, \text{ exact solution}$$

- Difference equation: $\frac{1}{\Delta x^2} (\theta_{i-1} - 2\theta_i + \theta_{i+1}) = -\sin 2\pi x_i$

$$\theta_0 = \theta_N = 0$$

- Equivalent differential equation: $\hat{\theta}'' = -\sin 2\pi x - \frac{\Delta x^2}{12} \hat{\theta}^{(iv)} + \ldots$

truncation error

Note, $T_i \to 0$ as $\Delta x \to 0 \Rightarrow EqDE \to DE$ (consistency $\to$ convergence, Lax)

To solve the difference equation, we assume the solution is of the form: $\theta_i = A \sin 2\pi i \Delta x \ (i = 0, N \Rightarrow \theta_i = 0)$ Thus

$$\frac{A}{\Delta x^2} [\sin 2\pi (i - 1) \Delta x - 2 \sin 2\pi i \Delta x + \sin 2\pi (i + 1) \Delta x] = -\sin 2\pi i \Delta x$$

or

$$\frac{A}{\Delta x^2} \sin 2\pi i \Delta x (\cos 2\pi \Delta x - 2 + \cos 2\pi \Delta x) = -\sin 2\pi i \Delta x$$

$$\Rightarrow A = \frac{\Delta x^2}{2(1 - \cos 2\pi \Delta x)} = \frac{\Delta x^2}{2[1 - (1 - \frac{(2\pi)^2}{2} \Delta x^2 + \frac{(2\pi)^4}{24} \Delta x^4 + \ldots)]}$$

$$= \frac{\Delta x^2}{(2\pi)^2 \Delta x^2 - \frac{(2\pi)^4}{12} \Delta x^4 + \ldots}$$
\( = \frac{1}{4\pi^2} \left[ 1 - \frac{(2\pi)^2 \Delta x^2}{12} + \ldots \right] \)
\( = \frac{1}{4\pi^2} \left[ 1 + \frac{(2\pi)^2 \Delta x^2}{12} + \ldots \right] \)

thus, the solution of the difference equation is:

\[ \theta_i = \frac{1}{4\pi^2} \sin 2\pi x_i + \frac{\Delta x^2}{12} \sin 2\pi x_i + \ldots \]

so we have shown that the approximation of the second derivative with second-order accuracy leads to a solution which is of second-order accurate.

To solve the equivalent D.E. we use perturbation expansions, \((\Delta x^2\) small parameter): \( \hat{\theta} = \theta_0 + \Delta x^2 \hat{\theta}_2 + \Delta x^4 \hat{\theta}_4 + \ldots \), so:

\[
\begin{align*}
\Delta x^0: & \quad \hat{\theta}_0'' = -\sin 2\pi x \\
& \quad \hat{\theta}_0(0) = \hat{\theta}_0(1) = 0 \\
\rightarrow & \quad \hat{\theta}_0 = \Theta = \frac{1}{4\pi^2} \sin 2\pi x
\end{align*}
\]

\[
\begin{align*}
\Delta x^2: & \quad \hat{\theta}_2'' = -\frac{1}{12}(2\pi)^4 \sin 2\pi x \\
& \quad \hat{\theta}_2(0) = \hat{\theta}_2(1) = 0 \\
\rightarrow & \quad \hat{\theta}_2 = \frac{1}{12} \sin 2\pi x
\end{align*}
\]

so, \( \hat{\theta}(x) = \frac{1}{4\pi^2} \sin 2\pi x + \frac{\Delta x^2}{12} \sin 2\pi x + \ldots \). Notice that \( \hat{\theta}(x_i) = \theta_i \), from above, in other words \( \hat{\theta}(x) \), solution of the equivalent D.E., collocates the solution at the nodes. Thus, the numerical solution is an “exact” solution at the nodes of the corresponding equivalent D.E. but not of the differential equation.

**Thomas algorithm for tridiagonal system** \([A][x] = [q]\)

\[
\begin{bmatrix}
  a_1 & c_1 & 0 & \cdots & 0 \\
a_2 & c_2 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \cdots & \cdots & c_n & b_n \\
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
x_2 \\
\vdots \\
x_n \\
\end{bmatrix}
= 
\begin{bmatrix}
  1 & \ell_2 & 1 & \cdots & 0 \\
\ell_2 & 1 & \cdots & \cdots & 0 \\
\ell_3 & \cdots & \cdots & \cdots & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & \cdots & \cdots & \cdots \\
\ell_N & \cdots & \cdots & \cdots & 1 \\
\end{bmatrix}
\begin{bmatrix}
  d_1 \\
d_2 \\
\vdots \\
d_n \\
\end{bmatrix}
\]
Step 1: (LU decomposition)

1st \( d_1 = a_1, u_1 = c_1 \)

\[ \ell_i d_{i-1} = b_i \Rightarrow \ell_i, \quad N \text{ mults} \]

\[ \ell_i u_{i-1} + d_i = a_i \Rightarrow d_i \quad N \text{ mults, } N \text{ adds} \]

\[ u_i = c_i \]

\[ \ell_N d_{N-1} = b_N \Rightarrow \ell_N \]

\[ \ell_N u_{N-1} + d_N = a_N \Rightarrow d_N \]

Total: \(2N\) mults, \(N\) adds

Step 2: (Forward Substitution) \([L][y] = [q]\)

\[
\begin{bmatrix}
1 \\
\ell_2 & 1 \\
& 1 \\
& & \ddots \\
& & & \ell_N & 1
\end{bmatrix}
\begin{bmatrix}
y_1 \\
y_2 \\
& & \ddots \\
& & & y_{N-1} \\
& & & & y_N
\end{bmatrix}
= 
\begin{bmatrix}
q_1 \\
q_2 \\
& & \ddots \\
& & & q_{N-1} \\
& & & & q_N
\end{bmatrix}
\Rightarrow
\begin{cases}
y_1 = q_1 \\
\ell_i y_{i-1} + y_i = q_i \Rightarrow y_i
\end{cases}
\]

Operation count \((mN)\) mults; \((mN)\) adds (where \(m = \text{bandwidth}\))

Step 3: (Back substitution) \([U][x] = [y]\)

\[
\begin{bmatrix}
d_1 & u_1 \\
d_2 & u_2 \\
d_3 & u_3 \\
& \ddots \\
0 & \cdots & u_{N-1} \\
& & d_N
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
& & \ddots \\
x_{N-1} \\
x_N
\end{bmatrix}
= 
\begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
& & \ddots \\
y_{N-1} \\
y_N
\end{bmatrix}
\Rightarrow
\begin{cases}
x_N = y_N / d_N \\
d_i x_i + u_i x_{i+1} = y_i \Rightarrow x_i \quad i = N - 1, \ldots, 1
\end{cases}
\]

Operation count: \((m + 1)N\) mults ; \(mN\) adds (where \(m = \text{bandwidth}\))

Total operation count:
1) LU:  
2N mults,  N adds

2) Forwd:  
N mults,  N adds

3) Back:  
2N mults,  N adds

Total:  5N mults,  3N adds

Note: It can be shown that the above algorithm will always converge if the tridiagonal system is diagonal dominant, i.e.

\[
|a_k| \geq |b_k| + |c_k|, \quad k = 2, \ldots, N - 1 \\
|a_1| > |c_1| \text{ and } |a_N| > |b_N|
\]

If \(a, b, c\) are matrices we have a block-tridiagonal system and the same algorithm can be applied.

**Exercise:** Count flops in triangular matrix multiplication. For square matrices is \(2N^3\), where we account for both mults and adds. Note that if \([A]\) and \([B]\) are upper triangular \(\Rightarrow [C]\) is upper triangular since 
\[
d_{ij} = \sum_{k=i}^j a_{ik} b_{kj} \text{ which corresponds to } 2(j - i + 1)ops \text{ in the innermost loop.} 
\]
Thomas algorithm for periodic tridiagonal systems:

\[
\begin{bmatrix}
  a_1 & c_1 & & b_1 \\
  b_2 & a_2 & c_2 & 0 \\
  & \ddots & \ddots & \ddots \\
  c_{n+1} & 0 & b_{N+1} & a_{N+1}
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_N \\
  x_{N+1}
\end{bmatrix}
= \begin{bmatrix}
  q
\end{bmatrix}
\]

where: \( x_1 = x_{N+2} \), periodicity

\( b_1 \) and \( c_{N+1} \) coefficients due to periodic b.c. = 1. By "condensing" the matrix (eliminate last row/column):

\[
\begin{bmatrix}
  a_1 & c_1 & & b_1 \\
  b_2 & a_2 & c_2 & 0 \\
  & \ddots & \ddots & \ddots \\
  0 & b_N & c_{N-1} & a_N
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_N
\end{bmatrix}
= \begin{bmatrix}
  q
\end{bmatrix} - \begin{bmatrix}
  b_1 \\
  0 \\
  \vdots \\
  0 \\
  c_N
\end{bmatrix}
\]

Thus, solution is of the form: \( x = x^{(1)} + x^{(2)} \cdot x_{N+1} \), where \( x^{(1)} \) and \( x^{(2)} \) solutions of the tridiagonal "condensed" system with \( N \) unknowns.

i.e.,

\[
[A^c] \begin{bmatrix} x^{(1)} \end{bmatrix} = [q]
\]

\[
[A^c] \begin{bmatrix} x^{(2)} \end{bmatrix} = \begin{bmatrix}
  -b_1 \\
  0 \\
  \vdots \\
  0 \\
  -c_N
\end{bmatrix}
\]
We finally compute \( x_{N+1} \) from the last equation in the original system by back substitution, i.e.
\[
c_{N+1}(x_1^{(1)} + x_{N+1}x_1^{(2)}) + b_{N+1}(x_N^{(1)} + x_{N+1}x_N^{(2)}) + a_{N+1}x_{N+1} = q_{N+1}
\]
\[
\Rightarrow x_{N+1} = \frac{q_{N+1} - c_{N+1}x_1^{(1)} - b_{N+1}x_N^{(1)}}{a_{N+1} + c_{N+1}x_1^{(2)} + b_{N+1}x_N^{(2)}}
\]

**Multi-dimensional formulas**

![Diagram](image)

\[
x \ (u_x)_{ij} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \mathcal{O}(\Delta x^2)
\]

\[
0 \ (u_{yy})_{ij} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} - \frac{\Delta y^2}{12} \left( \frac{\partial^4 u}{\partial y^4} \right)
\]

Laplacian \( \nabla^2 u_{ij} = (u_{xx} + u_{yy})_{ij} \)
\[
= \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\Delta x^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\Delta y^2} - \mathcal{O}(\Delta x^2, \Delta y^2)
\]

Thus, in symbolic form: \( \nabla^2 u_{ij} = \left( \frac{\delta^2}{\Delta x^2} + \frac{\delta^2}{\Delta y^2} \right) u_{ij} \)
For $\Delta x = \Delta y$

$$
\begin{array}{|c|c|c|}
\hline
1 & 1 & 1 \\
1 & -4 & 1 \\
1 & 1 & 1 \\
\hline
\end{array}
$$

- variable coefficient:

$$
\nabla \cdot (\nu \nabla u)_{ij} = \frac{1}{2\Delta x^2} \delta_x^-(\nu_{i+1/2,j} \delta_x^+)u_{ij}
+ \frac{1}{2\Delta y^2} \delta_y^-(\nu_{i,j+1/2} \delta_y^+)u_{ij} + \mathcal{O}(\Delta x^2, \Delta y^2)
$$

- Another operator is

$$
\nabla_x^2 u_{ij} = \left[ \frac{(\mu_y \delta_x)^2}{\Delta x^2} + \frac{(\mu_x \delta_y)^2}{\Delta y^2} \right] u_{ij}
= \frac{1}{4\Delta x^2} [(E_y + E_y^{-1} + 2)(E_x + E_x^{-1} - 2)]u_{ij}
+ \frac{1}{4\Delta y^2} [(E_x + E_x^{-1} + 2)(E_y + E_y^{-1} - 2)] u_{ij}
$$

- For $\Delta x = \Delta y$

$$
\Rightarrow \nabla_x^2 = \frac{1}{2\Delta x^2} [E_x E_y + E_x^{-1} E_y^{-1} + E_x^{-1} E_y + E_x E_y^{-1} - 4]
$$

$$
\frac{1}{2\Delta x^2} [u_{i+1,j+1} + \ldots - 4u_{ij}]
$$
- odd-eval oscillations

Two independent stencils ⇒ odd-even oscillation \{ → odd-numbered grid and even-numbered grid\}

- Define a new operator: \( \nabla_x^2 u_{ij} = (a \nabla_x^2 + b \nabla_x^2) u_{ij} \) with \( a + b = 1 \) substitute for \( \nabla_x^2 \) and \( \nabla_x^2 \) and \( a = 1 - b \): \( \Delta x = \Delta y \)

\[
\frac{1}{\Delta x^2} (\delta_x^2 + \delta_y^2) + \frac{b}{\Delta x^2} \left[ (\mu_y \delta_x)^2 + (\mu_x \delta_y)^2 - (\delta_x^2 + \delta_y^2) \right] \\
= \frac{1}{\Delta x^2} \left\{ (\delta_x^2 + \delta_y^2) + \frac{b}{2} \left[ 2\delta_x^2 (\mu_y^2 - 1) + 2\delta_y^2 (\mu_x^2 - 1) \right] \right\}
\]

but

\[
2\delta_x^2 (\mu_y^2 - 1) = 2(E_x + E_x^{-1} - 2) \left[ \frac{1}{4} (E_y + E_y^{-1} + 2) - 1 \right] \\
= \frac{1}{2} (E_x + E_x^{-1} - 2)(E_y + E_y^{-1} - 2) \\
= \frac{1}{2} \delta_x^2 \delta_y^2
\]

by symmetry second term is the same, thus:

\[
\nabla_x^2 = \frac{1}{\Delta x^2} \left[ (\delta_x^2 + \delta_y^2) + \frac{b}{2} \delta_x^2 \delta_y^2 \right] = \nabla^2 + \frac{b}{2} \delta_x^2 \delta_y^2 \cdot \frac{1}{\Delta x^2}
\]
For $b = 2/3$:

The Dahlquist-Bjorck stencil:

For $b = 1/3$, we get: \[ \nabla_x^2 u_{ij} = \nabla^2 u_{ij} + \frac{\Delta x^2}{12} \left[ \frac{\partial^4 u}{\partial x^4} + \frac{\partial^4 u}{\partial y^2} + 6 \cdot \frac{\partial^4 u}{3 \partial x \partial y} \right] \]

Thus, \( \nabla_x^2 u_{ij} = \nabla^2 u_{ij} + \frac{\Delta x^2}{12} \nabla^4 u \). Thus, in solving the eigenvalue problem: \( \nabla^2 u = \lambda u \) the truncation error is \(-\frac{\Delta x^2}{12}(\lambda^2 u)\) and thus by ejecting its opposite on the RHS (booster method):

\[ \nabla_x^2 u_{ij} = (\lambda + \frac{\lambda^2 \Delta x^2}{12})u \]

has a 4th-order truncation error. (method of corrected differences).
Mixed Derivatives: \( (\frac{\partial^2 u}{\partial x \partial y}, D_x \equiv \frac{\partial}{\partial x}, D_y \equiv \frac{\partial}{\partial y}) \)

Use operators, then:

\[
    u_{xy} = \frac{1}{\Delta x \Delta y}\mu_x \delta_x \left(1 - \frac{\delta^2_x}{6} + \mathcal{O}(\Delta x^4)\right) \mu_y \delta_y \left(1 - \frac{\delta^2_y}{6}\right) + \mathcal{O}(\Delta y^4) u_{ij}
\]

The second-order accurate stencil: \( u_{xy} = \frac{1}{\Delta x \Delta y} \mu_x \delta_x \mu_y \delta_y u_{ij} + \mathcal{O}(\Delta x^2, \Delta y^2) \)

\[
    (u_{xy})_{ij} = \frac{1}{4\Delta x \Delta y} \left[u_{i+1,j+1} - u_{i+1,j-1} - u_{i-1,j+1} + u_{i-1,j-1}\right] + \mathcal{O}(\Delta x^2)
\]

(Note: no element on the diagonal \( u_{ij} \))

Notice that in the above if we omit the average operator \( \mu \) we loose an order of accuracy, i.e.,

\[
    (u_{xy})_{ij} = \frac{1}{4\Delta x \Delta y} (\mu_x \delta_x \delta_y^+) u_{ij} + \mathcal{O}(\Delta x^2, \Delta y)
\]
Also, to obtain a first-order in $\Delta x, \Delta y$: 

$$(u_{xy})_{ij} = \frac{1}{\Delta x \Delta y} \delta_x^+ \delta_y^+ u_{ij} + \mathcal{O}(\Delta x, \Delta y)$$

(Forward)

Similarly for backward. Thus the sum: 

$$(u_{xy})_{ij} = \frac{1}{2 \Delta x \Delta y} \left[ \delta_x^+ \delta_y^+ + \delta_x^- \delta_y^- \right]$$

is of 2nd-order (symmetric)

Here, non-zero coefficient for $u_{ij}$.

- **Mappings and Non-Uniform Meshes**

  We often need to compute derivatives on a non-uniform grid which is necessary because of non-uniform gradients in the solution (e.g., boundary layer behavior) or because of complex geometry (requiring mappings) or because of semi-infinite or infinite intervals.

  It is common practice to transform non-uniform grids to uniform (if possible), and on the uniform grid take derivatives. This is due to the fact that the formal accuracy (e.g., second) of the scheme is maintained on the uniform grid. The actual accuracy depends strongly on the
transformation between the non-uniform grid \([x]_i\) and the uniform grid \([\xi]_i\).

\[ x = \phi(\xi) \]

If the transformation is a polynomial of second-degree (or less) then the accuracy is identical by either discretizing on \([x]_i\) on \([\xi]_i\) (see J.D. Hoffman, JCP, vol. 46, p. 469–474, (1982)). In general however

\[ \tilde{f}_x = \tilde{f}_x + \mathcal{O}(\delta\xi^2) \]

where \( \tilde{f}_x \) is the derivative on the non-uniform grid and \( \tilde{f}_{xx} \) on the uniform grid.

Note: \( \tilde{f}_{xx} = \tilde{f}_{xx} \) for \( x = \phi(\xi) \) which are first-degree polynomials.

So

\[ \tilde{f}_x - \frac{f_{i+1} - f_{i-1}}{x_{i+1} - x_{i-1}} + \frac{1}{2} (f_{xx})_i (\Delta x_+ - \Delta x_-) + \mathcal{O}(\Delta x^2) \]

\[ \tilde{f}_{xx} = \frac{d}{dx}(\phi^{-1}(x)) \bigg| \frac{f_{i+1} - f_{i-1}}{\xi_{i+1} - \xi_{i-1}} + \mathcal{O}(\Delta x^2) \]

Some commonly used mappings are:

- **Exponential Stretching**

  \[ x = L(e^{\alpha \xi} - 1) \]

  with the purpose of increasing resolution in a certain area

- **Semi-infinite integral**

  - algebraic: \( x = L \frac{1 + \xi}{1 - \xi} \), \( x \in [0, \infty] \rightarrow \xi \in [-1, 1] \)
  
  - logarithmic \( x = -L \ln \left( \frac{1 - \xi}{2} \right) \), \( [0, \infty] \rightarrow [-1, 1] \)
  
  - exponential \( x = x_{max} \frac{e^{\alpha(1 - \xi)} - e^{2\alpha}}{1 - e^{2\alpha}} \)
• Infinite-Interval

- exponential \( x = L \tanh^{-1} \xi \ \epsilon \epsilon \epsilon \rightarrow \epsilon \epsilon \epsilon \epsilon \epsilon \]
- algebraic \( x = L \frac{\xi}{\sqrt{1-\xi^2}} \)
References


