Phenomena

- Physics: heat, flow, space, time
- Mathematics: continuous functions, (partial) differential equations
- Computer science: Discrete simulation of physical phenomena through **Finite Difference Methods**
Differentials

- Physical phenomena like the flow of heat are modeled with differentials:

\[
\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}
\]

- A differential describes rate of change, e.g. velocity is the rate of change of position, \( v = \frac{df}{dx} \), and acceleration is the rate of change of velocity, \( a = \frac{dv}{dx} \), which is the second derivative (the derivative of the derivative of position)
Partial Differential Equations

Partial differential equations are differential equations in higher dimensions expressed in a coordinate system, e.g. in 2D:

\[
\frac{\partial u}{\partial x} \text{ and } \frac{\partial u}{\partial y}
\]

describe the change of \( u \) in the \( x \) and \( y \) direction.
Laplace

- Laplace described physical phenomena in 2 and 3D, e.g. heat in 2D

\[
\begin{align*}
\Delta V_x \Delta y &= \text{heat removed} \\
\Delta V_y \Delta x &= \text{heat removed}
\end{align*}
\]

- In X direction: cell receives heat \(V_x \Delta y\), loses heat \((V_x + \Delta V_x) \Delta y\), hence \(\Delta V_x \Delta y\) heat removed

- Similarly, in Y direction: \(\Delta V_y \Delta x\) heat removed
CS view

- Nearest neighbor computation, checkerboard or block row partitioning
- Some stencil computation represents the solution to the difference equation, which represents the differential equation, which represents the physical model of the natural phenomenon.
- At every level simplifying assumptions approximate the next level.
CS view: stencil computations

- Stencil computation approximates the solution to a differential equation
- Domain decomposition: Exchange of data along borders
- Trick: overlapping areas (see e.g. Quinn Ch. 13)
  - Re-computation to reduce communication frequency
  - Potentially more complicated communication pattern
    - Especially in higher dimensions
Numerical integration

- Finding the surface under $f(x) \cdot x$ in $[a,b]$
- Approximate $f(x)$ and derive simple formula for area
  - Linear: two points, quadratic: three, etc.
- Approximate in a number of intervals
  - Applying any form of above approximation methods
Integration, linear approximation

\[ f(x) \]

\[ x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9 \]
Trapezoidal rule

- Intervals: \( x_0, x_1 = x_0 + h, x_2 = x_0 + 2h, \ldots X_n, \)

  \[ h = \frac{b - a}{n} \]

  \[
  I = h \left( \frac{f(x_0) + f(x_1)}{2} + \ldots + \left( \frac{f(x_{n-1}) + f(x_n)}{2} \right) \right)
  \]

  \[
  f(x_0) + 2 \sum_{i=1}^{n-1} f(x_i) + f(x_n)
  \]

  \[
  = \frac{(b - a)}{2n}
  \]
Better approximations

- Either: more points (increase n)
- or higher order polynomials
  - E.g. Simpsons rule uses quadratic approximation over 3 points

\[ I = \frac{h}{3} (f(x_0) + 4f(x_1) + f(x_2)) \]

- Intervals:
  \[ I = \frac{b-a}{3n} (f(x_0) + 4 \sum_{i=1,3,5,..}^{n-1} f(x_i) + 2 \sum_{i=2,4,6,..}^{n-1} f(x_i) + f(x_n)) \]
Iterative / adaptive approach

- Iterate with smaller and smaller segments until $I_i \sim I_{i+1}$
  
  $h_1 = (b-a)/n, \quad h_{2\text{etc.}} = (h_1)/2, \text{ etc.}$

- Error: use relative error

\[ \varepsilon_r = \frac{\text{present approx} - \text{previous approx}}{\text{present approx}} \times 100\% \]

\[ \leq (0.5 \times 10^{2-n})\% \]

$n$: number of significant digits
Recursive approach: adaptive quadrature

\[
\text{trap}(left, right) = \{ \text{return } (right-left)*(f(left)+f(right))/2; \}
\]

\[
\text{tol} = (0.5\times\exp(10,2-n));
\]

\[
\text{area}(left, right, est) = \{
\text{mid}=(left+right)/2; \\
a1=\text{trap}(left,mid); \ a2=\text{trap}(mid,right); \\
\text{newest} = a1+a2; \\
\text{if}(\text{abs}((\text{newest-est})/\text{newest})<\text{tol}) \\
\quad \text{return newest;}
\text{else return } \text{area}(left,mid,a1) + \text{area}(mid,right,a2)
\}\]