CS475 Parallel Programming

Differentiation and Integration

Wim Bohm

Colorado State University

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:

$$\frac{df}{dx} = \lim_{\Delta x \to o} \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

- A differential describes **rate of change**, e.g. velocity is the rate of change of position, v = df/dx, and acceleration is the rate of change of velocity, a = dv/dx, which is the second derivative (the derivative of the derivative of position)

Partial Differential Equations

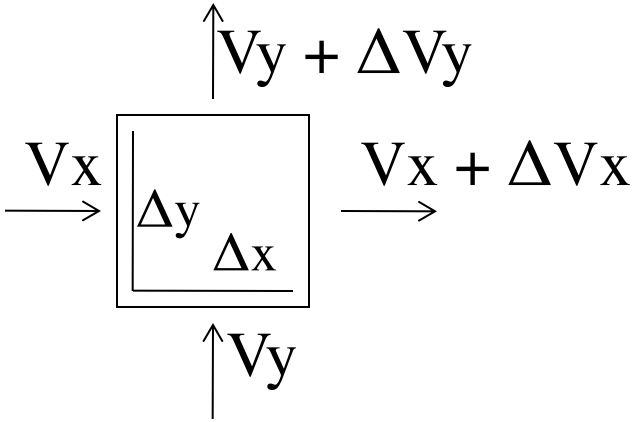
- Partial differential equations are differential equations in higher dimensions expressed in a coordinate system, e.g in 2D:

$$\frac{\partial u}{\partial x} and \frac{\partial u}{\partial y}$$

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



- 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 solu\tion 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.

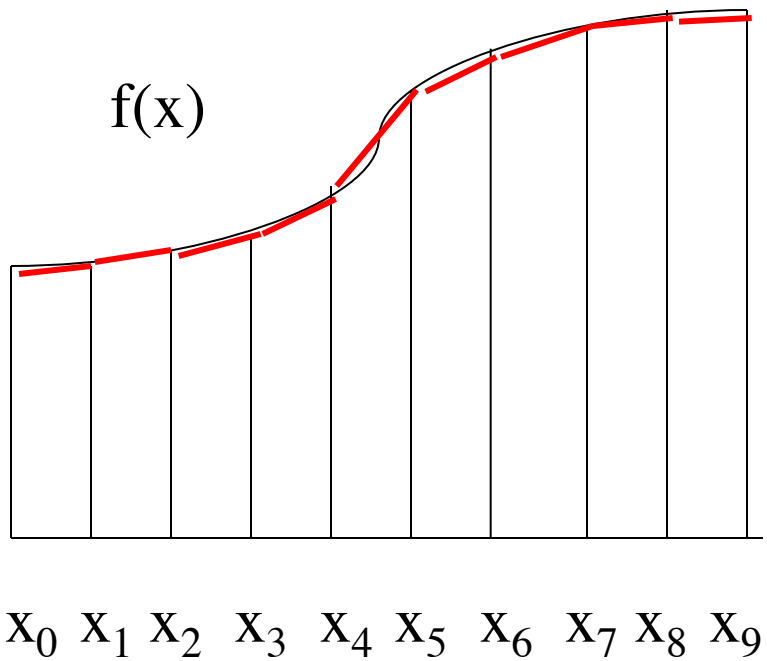
CS view: stencil computations

- 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) 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



Trapezoidal rule

- Intervals: x_0, $x_1 = x_0 + h$, $x_2 = x_0 + 2h$, …. X_n,

  $h = (b-a)/n$

$$I = h((f(x_0)+f(x_1))/2 \ + \ \ldots + ((f(x_{n-1})+f(x_n)/2))$$

$$= \quad (b-a)\frac{f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)}{2n}$$

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 $I_i \sim I_{i+1}$

  $h_1=(b-a)/n \qquad h_{2etc.} = (h_1)/2$ etc.

- Error: use relative error

$$\varepsilon_r = \frac{present \quad approx \quad - \quad previous \quad approx}{present \quad approx}.100\%$$

$$\leq (0.5*10^{2-n})\%$$

n: number of significant digits

Recursive approach: adaptive quadrature

```
trap(left,right) = { return (right-left)*(f(left)+f(right))/2;}
tol = (0.5*exp(10,2-n));
area(left,right,est) ={
   mid=(left+right)/2;
   a1=trap(left,mid); a2=trap(mid,right);
   newest = a1+a2;
   if(abs((newest-est)/newest)<tol)
      return newest;
   else return area(left,mid,a1) +area(mid,right,a2)
}
```