THE UNIVERSITY OF WARWICK

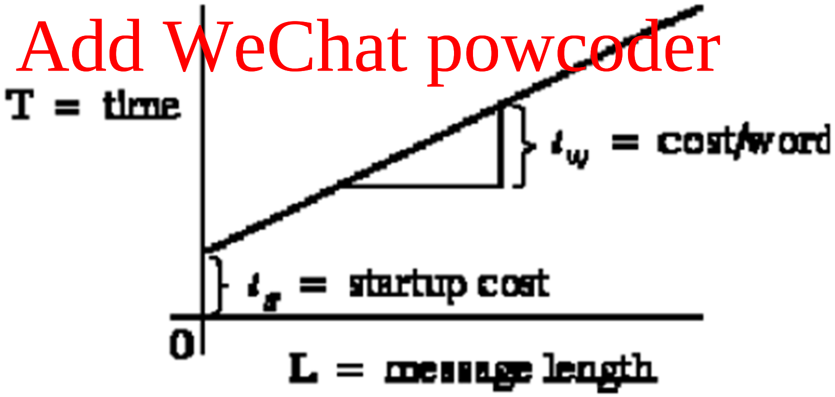
High Performance Computing
Course Notes
Performance III

Dr Ligang He

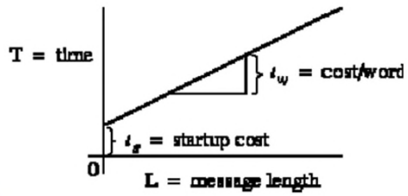
Time for sending a message

$T_{msg}=t_s+t_w * L$

Question: How to determine t_s and t_w



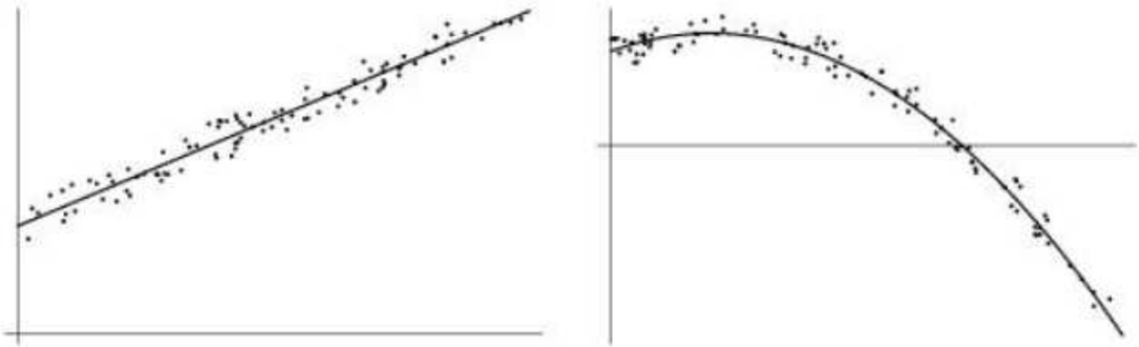
Curve fitting



- Suppose that we obtained a set of measurement values, $\{(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\}$, which is called the measurement sample
- The goal is to obtain a "fitting function", f(x), that is the best fit to the data

- The quality of the fitting lies in the residuals:

$$\{r_i=y_i-f(x_i), i=0, 1, \ldots, n\}$$



LEAST SQUARES FITTING

- Selecting the fitting function that minimizes the sum of the squares of residuals ($r_i = y_i - f(x_i)$).
- Suppose that the fitting function is written $y = f(x, a_1, \ldots, a_n)$ where $\{a_1, \ldots, a_n\}$ are the fitting parameters
- Define the sum of squares of the residuals, R^2, over a set of n data pairs $\{(x_1, y_1), \ldots (x_n, y_n)\}$,

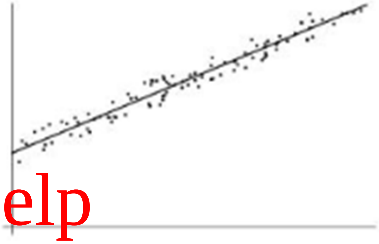
$$R^2 = \sum_{i=1}^{n} [y_i - f(x_i, a_1, a_2, \ldots a_n)]^2$$

- The condition for R^2 to be a minimum is that

$$\frac{\partial (R^2)}{\partial a_i} = 0 \quad (i=1, \ldots, n)$$

Linear function

Finding the straight line (slope and intercept) that best fits a set of data points



$$f(a, b) = a + b\, x,$$

$$R^2(a, b) \equiv \sum_{i=1}^{n} [y_i - (a + b\, x_i)]^2$$

$$\begin{cases} \dfrac{\partial(R^2)}{\partial a} = -2 \displaystyle\sum_{i=1}^{n} [y_i - (a + b\, x_i)] = 0 \\ \dfrac{\partial(R^2)}{\partial b} = -2 \displaystyle\sum_{i=1}^{n} [y_i - (a + b\, x_i)]\, x_i = 0. \end{cases} \Longrightarrow \begin{cases} n\, a + b \displaystyle\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \\ a \displaystyle\sum_{i=1}^{n} x_i + b \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i\, y_i. \end{cases}$$

So,

$$\begin{cases} a = \dfrac{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i\, y_i}{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2} \\ b = \dfrac{n \sum_{i=1}^{n} x_i\, y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2} \end{cases}$$

Polynomial function

Finding the polynomial function that best fits a set of data points. In this case, the fitting function has the form

$$f(x)=a_0+a_1x+a_2x^2+\ldots+a_nx^n$$

Solving partial differential equations

When we introduce the background of the Deqn code, we studied the finite difference method for solving partial differential equations

$$\frac{\partial f(x)}{\partial x} = \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

Now, we introduce the finite volume method for solving partial differential equations

Partial differential equations

- **Fluid flow can be modelled as partial differential equations**

$$\nabla^2\phi = 0 \qquad (\text{i.e. } \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0)$$

- where the velocity potential function $\phi(x, y)$ is related to the flow velocity $\vec{v}$ by

$$\vec{v} = \nabla\phi \qquad (\text{i.e. } v_x = \frac{\partial\phi}{\partial x},\ v_y = \frac{\partial\phi}{\partial y})$$

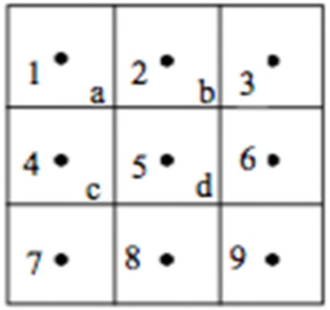
- **The aim is to find the numerical solution for ϕ**

Numerical solution for partial differential equations

- First, approximating continuous space with a set of discrete points
- Then, finding the value of the function of interest at each discrete point
  - The finer points the continuous space is partitioned into, the more accurate the solution will be
  - In some situations, it is adequate to partition the space into a regular grid where the distance between points is uniform
  - When we need to get more accuracy in certain areas (e.g. the function changes rapidly), we need to place more points in those areas

Finite volume method for solving differential equations

- A continuous space is broken down into a set of volumes (cells)



- A cell surrounds one of the discrete points

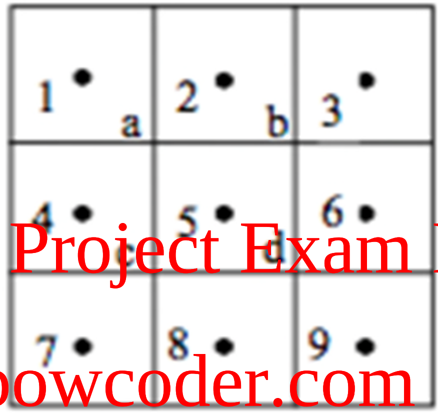
- Using these cells to solve φ in the fluid flow problem, expressed as differential equations $\nabla^2\phi = 0$ (i.e. $\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0$)

  - The net flow into a cell has to be zero

  - We can set up a linear equation for each cell to express the above relationship
  - The unknown variables in a linear equation are the values of the function φ at the points
  - For n cells, there are n unknown variables and n equations

How to set up the linear equations

Consider this example:



The linear equation for cell 5 can be setup as

$$\frac{(\phi_2-\phi_5)l_{a-b}}{l_{2-5}}+\frac{(\phi_6-\phi_5)l_{b-d}}{l_{6-5}}+\frac{(\phi_8-\phi_5)l_{d-c}}{l_{8-5}}+\frac{(\phi_4-\phi_5)l_{c-a}}{l_{4-5}}=0$$

We can write similar equations for each of the nine cells, then we get a set of equations of the form AΦ=b, A is the matrix of the coefficients in the equations, Φ is the vector with the value of φ to be calculated at each point

Using the iterative method to solve the linear equations

- **Aim: solve** AΦ=b

- **Method: repeating iterative steps and each step generates a better approximation of the solution**

- Step 1: Guess a initial solution Φ^0

- Step 2: Check if convergence is reached by checking the residual $b-A\Phi^i$<tolerance

- Step 3: For Φ^i (i>=0), $\Phi^{(i+1)} = \Phi^{(i)} + (b - A\Phi^{(i)})$, go to Step 2

Successive Over-Relaxation(SOR)

- The SOR method can speed up convergence
- For a set of linear equations

- let A=D+U+L, where D, L and U denote the diagonal, strictly lower triangular, and strictly upper triangular parts of A, respectively

- The successive over-relaxation (SOR) iteration is defined by the recurrence relation

$$(D + \omega L)\phi^{(k+1)} = (-\omega U + (1 - \omega)D)\phi^{(k)} + \omega b. \qquad (*)$$

- Where values of w > 1 are used to speedup convergence of a slow-converging process, while values of w < 1 are often help to establish convergence of diverging iterative process