Scientific Computing

Lecture 1

Delft University of Technology

Vandana Dwarka and Kees Vuik

September 4, 2024

Course Structure and Planning

▸ Course Website

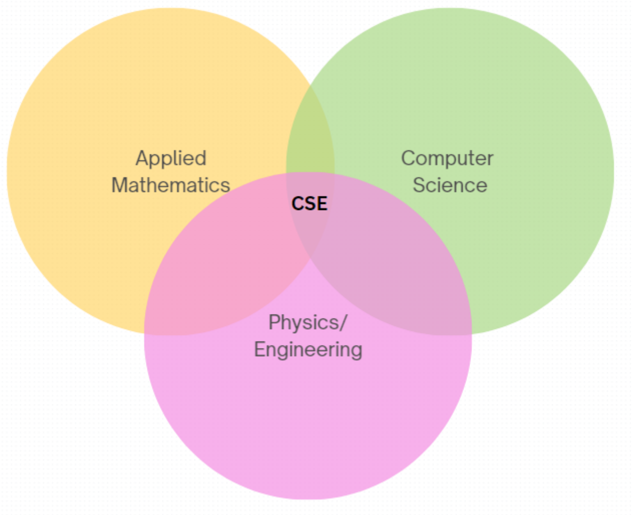
- Weekly 2-hour lecture (schedule on Brightspace)
- G_1: Take-home exam (theoretical and/or practical)
  - Deadline: 15-11-2024
- G_2: Take-home exam (mostly practical)
  - Deadline: 10-01-2025
  - Group of 2 students - PhD TA's
  - Check-in with supervisor in December
- G_3: Final written exam
  - Exam on 22-01-2025
- Final grade: $(G_1 + G_2 + 2G_3)/4$, with the condition that all grades must be ≥ 5

Prerequisite Knowledge

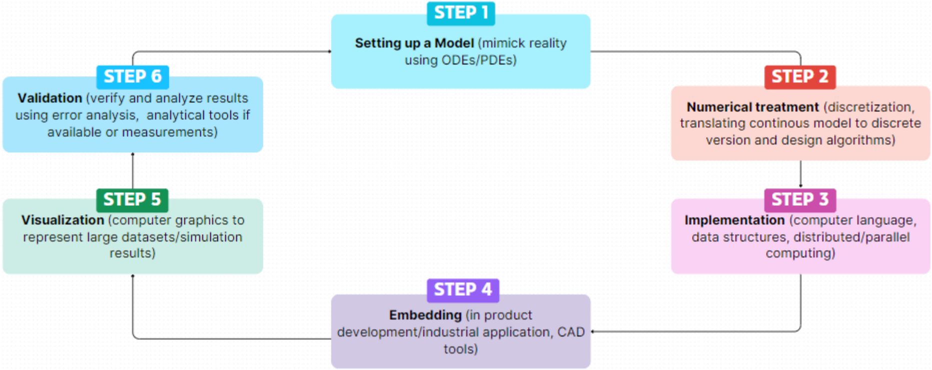
We assume you are familiar with material from the courses:

- Linear Algebra
- Calculus or Analysis
- Differential Equations
- Numerical Methods I (Book used)

Compututational Science and Engineering



Six Steps of a Simulation Process

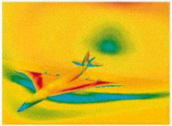


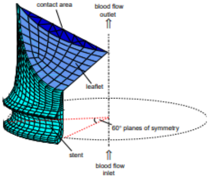
Example: Turbulence

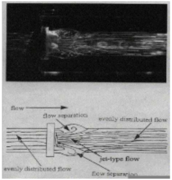
Real-world phenomena

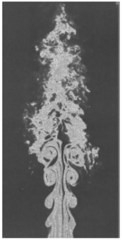
Numerical Model

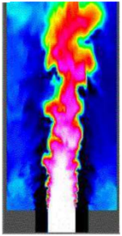
Numerical Solution











Algorithms, Methods, Code

Scientific Computing focuses on developing algorithms, defined as:

A set of instructions to carry out certain mathematical, arithmetical and logical operations (or already known algorithms) for solving a prescribed problem.

Algorithms, Methods, Code

Scientific Computing focuses on developing algorithms, defined as:

A set of instructions to carry out certain mathematical, arithmetical and logical operations (or already known algorithms) for solving a prescribed problem.

Finite vs.infinite algorithms: solution is obtained after a finite or infinite number of steps

Algorithms, Methods, Code

Scientific Computing focuses on developing algorithms, defined as:

A set of instructions to carry out certain mathematical, arithmetical and logical operations (or already known algorithms) for solving a prescribed problem.

Finite vs.infinite algorithms: solution is obtained after a finite or infinite number of steps

Example: Babylonian Root Extraction

$$\sqrt{a} \ (a > 0): \quad x_0 > 0 \quad \text{arbitrary}$$
$$n = 1, 2, 3, \ldots \quad x_n = \frac{1}{2}\left(x_{n-1} + \frac{a}{x_{n-1}}\right)$$

Stopping criterion: $|x_n - x_{n-1}| \leq \epsilon$

Babylonian Root - Derivation

Desired: zeros of $g(x) = x^2 - a = 0$

Derivation via ▸ Newton-Raphson algorithm:

$$x_n = x_{n-1} - \frac{g\left(x_{n-1}\right)}{g'\left(x_{n-1}\right)}$$

$$x_n = x_{n-1} - \frac{x_{n-1}^2 - a}{2x_{n-1}} = x_{n-1} - \frac{1}{2}x_{n-1} + \frac{1}{2}\frac{a}{x_{n-1}}$$

$$x_n = \frac{1}{2}\left(x_{n-1} + \frac{a}{x_{n-1}}\right)$$

Babylonian Root - Derivation

Desired: zeros of $g(x) = x^2 - a = 0$

Derivation via Newton-Raphson algorithm:

$$x_n = x_{n-1} - \frac{g(x_{n-1})}{g'(x_{n-1})}$$

$$x_n = x_{n-1} - \frac{x_{n-1}^2 - a}{2x_{n-1}} = x_{n-1} - \frac{1}{2}x_{n-1} + \frac{1}{2}\frac{a}{x_{n-1}}$$

$$x_n = \frac{1}{2}\left(x_{n-1} + \frac{a}{x_{n-1}}\right)$$

Infinite algorithm + stopping criterion = finite algorithm

Babylonian Root - Results

We take $a = 0.64$, $\sqrt{a} = 0.8$

n	x_n	$x_n - \sqrt{a}$	$(x_n - \sqrt{a}) / \sqrt{a}$	EB (3)
0	0.76	0.04	0.05	0.05
1	0.801052631	0.001052631	0.001316...	0.001316
2	0.800000691	0.000000691	0.000000864	0.000658
3	0.800000000...	$< 10^{-10}$	$< 1.2510^{-10}$	0.000329

Babylonian Root - Results

We take $a = 0.64$, $\sqrt{a} = 0.8$

n	x_n	$x_n - \sqrt{a}$	$(x_n - \sqrt{a})/\sqrt{a}$	EB (3)
0	0.76	0.04	0.05	0.05
1	0.801052631	0.001052631	0.001316...	0.001316
2	0.800000691	0.000000691	0.000000864	0.000658
3	0.800000000...	$< 10^{-10}$	$< 1.2510^{-10}$	0.000329

Improved Error Bound

$$\frac{x_n - \sqrt{a}}{\sqrt{a}} = \frac{1}{2}\left(\frac{x_{n-1} - \sqrt{a}}{\sqrt{a}}\right)^2 \frac{\sqrt{a}}{x_{n-1}}$$

$$\text{If } x_0 \geq \sqrt{a} : \frac{x_n - \sqrt{a}}{\sqrt{a}} \leq \frac{1}{2^{2^n - 1}}\left(\frac{x_0 - \sqrt{a}}{\sqrt{a}}\right)^{2^n}$$

Errors in Scientific Computing

'"If mathematical theories refer to reality, then they are not certain. If they are certain, then they do not refer to reality.'

Albert Einstein

Errors in Scientific Computing

> *'"If mathematical theories refer to reality, then they are not certain. If they are certain, then they do not refer to reality.'*
>
> Albert Einstein

An important aspect of numerical math is understanding the errors made in computing solutions.

Errors in Scientific Computing

> *'"If mathematical theories refer to reality, then they are not certain. If they are certain, then they do not refer to reality.'*
>
> Albert Einstein

An important aspect of numerical math is understanding the errors made in computing solutions.

Errors in Scientific Computing

> *'"If mathematical theories refer to reality, then they are not certain. If they are certain, then they do not refer to reality.'*
>
> Albert Einstein

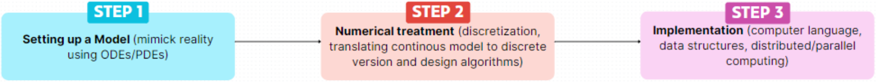
An important aspect of numerical math is understanding the errors made in computing solutions.

Error Measures: for $x, \hat{x} \in \mathbb{R}$:

$$|x - \hat{x}| \leq \epsilon \qquad \text{Estimate of absolute error}$$

$$\frac{|x - \hat{x}|}{|x|} \leq \epsilon \qquad \text{Estimate of relative error}$$

Sources of Errors



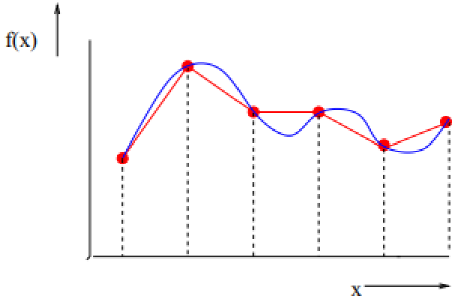
- Modelling error
- Error in data

- Discretization error
- Truncation error
- Round-off error

- Programming errors

Sources of Errors - Discretization

Discretization error



Only n values are obtained: gives an approximate solution curve.

Sources of Errors - Round-off

Numbers can only be stored with finite number of bits/digits

Generally round-off error small

Problematic when used with unstable algorithm

Sources of Errors - Round-off

Ill-conditioned linear system

$$\begin{pmatrix} 1 & 1 \\ 1 & 1-\varepsilon \end{pmatrix} x = \begin{pmatrix} 4 \\ 4-\varepsilon \end{pmatrix} \quad Ax = b$$

$$x^* = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 \\ 1 & 1-\varepsilon \end{pmatrix} \tilde{x} = \begin{pmatrix} 4+\varepsilon \\ 4-2\varepsilon \end{pmatrix} \quad A\tilde{x} = \tilde{b}$$

$$\tilde{x}^* = \begin{pmatrix} 1+\varepsilon \\ 3 \end{pmatrix}$$

Sources of Errors - Round-off

Ill-conditioned linear system

$$\begin{pmatrix} 1 & 1 \\ 1 & 1-\varepsilon \end{pmatrix} x = \begin{pmatrix} 4 \\ 4-\varepsilon \end{pmatrix} \quad Ax = b$$

$$x^* = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 \\ 1 & 1-\varepsilon \end{pmatrix} \tilde{x} = \begin{pmatrix} 4+\varepsilon \\ 4-2\varepsilon \end{pmatrix} \quad A\tilde{x} = \tilde{b}$$

$$\tilde{x}^* = \begin{pmatrix} 1+\varepsilon \\ 3 \end{pmatrix}$$

Also: $\|b-\tilde{b}\|_\infty = \varepsilon$, but $\|x-\tilde{x}\| = \max\{|2-\varepsilon|, 2\} = 2$

Sources of Errors - Round-off

Problem: Computation of

$$x_n := \int_0^1 \frac{t^n}{t+10} dt \text{ for larger } n$$
$$> 0 \text{ clearly}$$

$$x_0 = \int_0^1 \frac{1}{t+10} dt = \ln(10+t)|_0^1 = \ln\frac{11}{10} = \ln 1.1 \approx 0.0953$$

$$x_n := \int_0^1 \frac{t^n}{t+10} dt = \int_0^1 \frac{t^{n-1}(t+10) - 10t^{n-1}}{t+10} dt$$

$$= \int_0^1 t^{n-1} dt - 10x_{n-1} = \frac{1}{n} - 10x_{n-1}$$

Recursion: $x_n = \frac{1}{n} - 10x_{n-1}$

Sources of Errors - Round-off

Practically: computation with 3 decimals gives the approximations $\tilde{x}_n$

$$
\begin{aligned}
\tilde{x}_0 &= 0.0953 \\
\tilde{x}_1 &= 1 - 0.953 = 0.047 \quad (x_1 = 0.0469..) \\
\tilde{x}_2 &= 0.5 - 0.470 = 0.030 \quad (x_2 = 0.0310..) \\
\tilde{x}_3 &= 0.333 - 0.300 = 0.033 \quad (x_3 = 0.0232..) \\
\tilde{x}_4 &= 0.250 - 0.330 = -0.08 < 0! \quad (x_4 = 0.0185..)
\end{aligned}
$$

Sources of Errors - Round-off

Practically: computation with 3 decimals gives the approximations $\tilde{x}_n$

$$
\begin{aligned}
\tilde{x}_0 &= 0.0953 \\
\tilde{x}_1 &= 1 - 0.953 = 0.047 \quad (x_1 = 0.0469..) \\
\tilde{x}_2 &= 0.5 - 0.470 = 0.030 \quad (x_2 = 0.0310..) \\
\tilde{x}_3 &= 0.333 - 0.300 = 0.033 \quad (x_3 = 0.0232..) \\
\tilde{x}_4 &= 0.250 - 0.330 = -0.08 < 0! \quad (x_4 = 0.0185..)
\end{aligned}
$$

Reason: $x_n = \frac{1}{n} - 10x_{n-1}$, x_{n-1} contains error ε

$\Rightarrow$ x_n contains error approx. 10ε

$\Rightarrow$ x_{n+1} contains error approx. 100 . "Unstable recursion"

Sources of Errors - Round-off

Fix it: reverse procedure, start with x_n

$$x_{n-1} = \frac{1}{10n} - \frac{1}{10}x_n = \frac{1}{10}\left(\frac{1}{n} - x_n\right)$$

Sources of Errors - Round-off

Fix it: reverse procedure, start with x_n

$$x_{n-1} = \frac{1}{10n} - \frac{1}{10}x_n = \frac{1}{10}\left(\frac{1}{n} - x_n\right)$$

How to pick x_n?

$$x_n = \int_0^1 \frac{t^n}{t+10}dt \leq \int_0^1 \frac{1}{10}t^n dt = \frac{1}{10(n+1)} \to 0 \quad (n \to \infty)$$

Employ recursion with $y_n = 0, y_{n-1} = \frac{1}{10}\left(\frac{1}{n} - y_n\right)$

Sources of Errors - Round-off

Fix it: reverse procedure, start with x_n

$x_{n-1} = \frac{1}{10n} - \frac{1}{10}x_n = \frac{1}{10}\left(\frac{1}{n} - x_n\right)$

How to pick x_n?

$x_n = \int_0^1 \frac{t^n}{t+10}dt \leq \int_0^1 \frac{1}{10}t^n dt = \frac{1}{10(n+1)} \to 0 \quad (n \to \infty)$

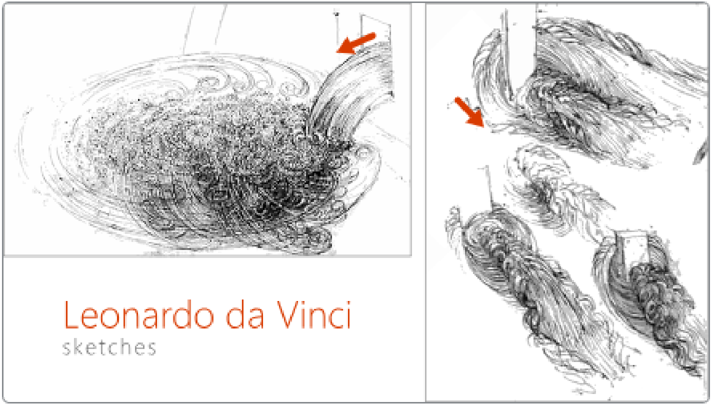
Employ recursion with $y_n = 0, y_{n-1} = \frac{1}{10}\left(\frac{1}{n} - y_n\right)$

$$
\begin{aligned}
y_8 &= 0 \\
\tilde{y}_7 &= 0.0125 \\
\tilde{y}_6 &= 0.0143 - 0.00125 = 0.0131 \\
\tilde{y}_5 &= 0.0167 - 0.00131 = 0.0154 \\
\tilde{y}_4 &= 0.0200 - 0.00154 = 0.0185
\end{aligned}
$$

$\tilde{y}_4$ is 3 digits accurate

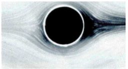
Example: Turbulence

Fluid mechanics: laminar (Euler equations) - turbulent flow (Navier-Stokes equation)

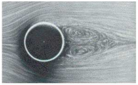


Example: Turbulence

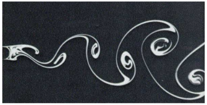
Reynolds Number represents balance between friction and non-friction



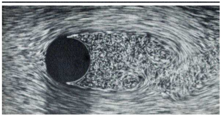
$Re = 0.16$,
Laminar, stationary, symmetric



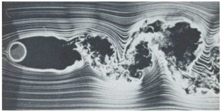
$Re = 26$,
Laminar, stationary, recirculation



$Re = 140$,
Transition, unsteady, von Kármán vortex street



$Re = 2000$,
Turbulent, unsteady



$Re = 10000$,
Turbulent, unsteady

Example: Turbulence - Mathematical Model

Flow mathematically modelled by the Navier-Stokes equation:

$$\underbrace{\frac{\partial u}{\partial t}}_{acceleration} + \underbrace{u\frac{\partial u}{\partial x}}_{convection} + \underbrace{v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}} + \underbrace{\frac{\partial p}{\partial x}}_{pressuregradient} =$$

$$\underbrace{\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right)}_{friction}$$

Example: Turbulence - Mathematical Model

Flow mathematically modelled by the Navier-Stokes equation:

$$\underbrace{\frac{\partial u}{\partial t}}_{acceleration} + \underbrace{u\frac{\partial u}{\partial x}}_{convection} + \underbrace{v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}} + \underbrace{\frac{\partial p}{\partial x}}_{pressuregradient} =$$

$$\underbrace{\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right)}_{friction}$$

Incompressible flow, also continuity equation:

$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$

Example: Turbulence - Mathematical Model

Flow mathematically modelled by the Navier-Stokes equation:

$$\underbrace{\frac{\partial u}{\partial t}}_{acceleration} + \underbrace{u\frac{\partial u}{\partial x}}_{convection} + \underbrace{v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}} + \underbrace{\frac{\partial p}{\partial x}}_{pressuregradient} =$$

$$\underbrace{\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right)}_{friction}$$

Incompressible flow, also continuity equation:

$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$

- $Re = \bar{U}L/\nu$ is the Reynolds number
- System of non-linear PDEs. **No** analytical solution exists (only for simple geometries/BCs).
- Solve using DNS (Direct Numerical Simulation)

Example: Turbulence - Reynolds Number

$Re = \bar{U}L/\nu$ is the Reynolds number

- Spoon stirring in a cup coffee ($Re \sim 10^4$)
- Airflow around a car ($Re \sim 3 \times 10^6$)
- Water flow in a river ($Re \sim 10^7$)
- Air flow around an aircraft ($Re \sim 3 \times 10^8$)

Example: Turbulence - DNS

How many computing operations to simulate a channel flow with $Re = 10^5$?

Example: Turbulence - DNS

How many computing operations to simulate a channel flow with $Re = 10^5$?

- Have number of grid points in all 'Eddies'
- Smallest length-scale: $\eta \approx L/1000$, i.e. 1000 gridpoints in every coordinate direction.
- In 3D this gives 10^9 number of unknowns (nou).

Example: Turbulence - DNS

How many computing operations to simulate a channel flow with $Re = 10^5$?

Have number of grid points in all 'Eddies'

Smallest length-scale: $\eta \approx L/1000$, i.e. 1000 gridpoints in every coordinate direction.

In 3D this gives 10^9 number of unknowns (nou).

Computer storage $> 10\times$ nou $= 10^{10}$

Example: Turbulence - DNS

How many computing operations to simulate a channel flow with $Re = 10^5$?

Have number of grid points in all 'Eddies'

Smallest length-scale: $\eta \approx L/1000$, i.e. 1000 gridpoints in every coordinate direction.

In 3D this gives 10^9 number of unknowns (nou).

Computer storage $> 10\times$ nou $= 10^{10}$

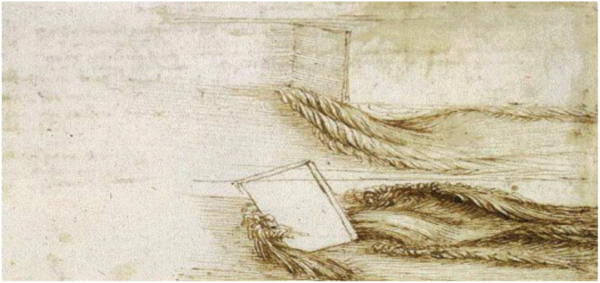
Computing operations $\sim 500\times$ nou per timestep $= 5 \times 10^{11}$

Number of timesteps $\sim 10^4$

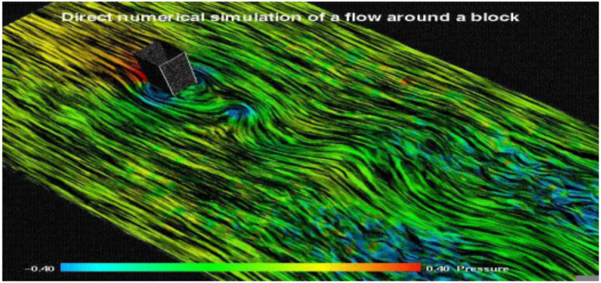
Total number of operations: 5×10^{15}

Example: Turbulence

Leonardo vs. DNS ($Re = 22000$)



Leonardo



DNS