UNIVERSITY AT ALBANY
STATE UNIVERSITY OF NEW YORK

CSI 401 (Fall 2025)

Numerical Methods

Lecture 18: Advanced Topic: Differential Equations

Chong Liu

Department of Computer Science

Nov 24, 2025

Announcements

- Today's two sessions are your last chances to earn your participation points.
  - Up to 3 points can be earned by joining in-class discussions
  - The remaining 2 points will be given to all students if at least 60% students submit their course evaluations
    - Why at least 60%? Statistically significant conclusions on my teaching!

Recap: numerical integration

- For f: $R \to R$, definite integral over interval $[a, b]$

$$I(f) = \int_a^b f(x)\,dx$$

  - is defined by limit of Riemann sums

$$R_n = \sum_{i=1}^{n} (x_{i+1} - x_i)\, f(\xi_i)$$

  - Riemann integral exists provided integrand f is bounded and continuous almost everywhere
- Key question today: How can we use computers to calculate the integration by querying f only?
- Discussion: What's your idea?

Recap: Quadrature Rules

- An n-point quadrature rule has form
  - Points x_i are called nodes
  - Multipliers w_i are called weights

$$Q_n(f) = \sum_{i=1}^{n} w_i \, f(x_i)$$

- Quadrature rules are based on polynomial interpolation
  - Integrand function f is sampled at finite set of points
  - Integral of interpolant is taken as estimate for integral of original function

- In practice, interpolating polynomial is not determined explicitly but used to determine weights corresponding to nodes

Recap: Newton-Cotes Quadrature

- Midpoint rule

$$M(f) = (b-a)\, f\left(\frac{a+b}{2}\right)$$

- Trapezoid rule

$$T(f) = \frac{b-a}{2}\,(f(a) + f(b))$$

- Simpson's rule

$$S(f) = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$$

Recap: Composite Quadrature

- Subdivide interval $[a, b]$ into k subintervals
  - Length $h = (b - a)/k$, for $x_j = a + jh, j = 0, \dots, k$
- Composite midpoint rule

$$M_k(f) = \sum_{j=1}^{k} (x_j - x_{j-1}) f\left(\frac{x_{j-1} + x_j}{2}\right) = h \sum_{j=1}^{k} f\left(\frac{x_{j-1} + x_j}{2}\right)$$

- Composite trapezoid rule

$$\begin{aligned} T_k(f) &= \sum_{j=1}^{k} \frac{(x_j - x_{j-1})}{2} \left(f(x_{j-1}) + f(x_j)\right) \\ &= h\left(\tfrac{1}{2} f(a) + f(x_1) + \cdots + f(x_{k-1}) + \tfrac{1}{2} f(b)\right) \end{aligned}$$

Recap: Summary of numerical integration

- Integral is approximated by weighted sum of sample values of integrand function

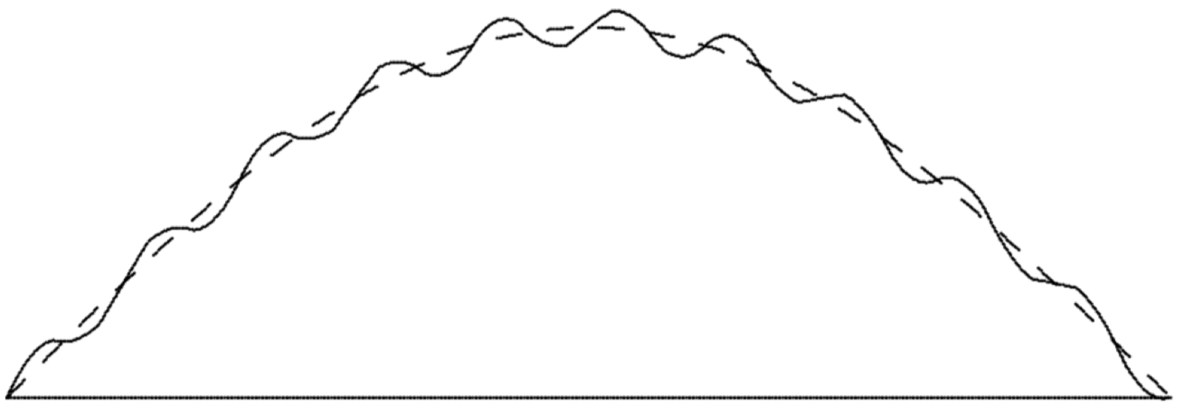
- Nodes and weights chosen to achieve required accuracy at least cost (fewest evaluations of integrand)

- Quadrature rules derived by integrating polynomial interpolant
  - Newton-Cotes rules use equally spaced nodes and choose weights to maximize polynomial degree

- Composite Quadrature divides original interval into subintervals
  - Works using piecewise interpolation

Recap: Numerical differentiation

- Differentiation is inherently sensitive, as small perturbations in data can cause large changes in result
- Integration is inherently stable because of its smoothing effect
  - For example, two functions shown below have very similar definite integrals but very different derivatives



Recap: numerical differentiation

- Given smooth function $f\colon R \to R$, we wish to approximate its first and second derivatives at point x
- Key question today: How can we use computers to calculate the differentiation by querying f only?
  - Discussion: what is your idea?
- Consider Taylor series expansions

$$
\begin{aligned}
f(x+h) &= f(x) + f'(x)h + \frac{f''(x)}{2}h^2 + \frac{f'''(x)}{6}h^3 + \cdots \\
f(x-h) &= f(x) - f'(x)h + \frac{f''(x)}{2}h^2 - \frac{f'''(x)}{6}h^3 + \cdots
\end{aligned}
$$

Recap: Finite Difference Approximations

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{f''(x)}{2}h + \cdots \approx \frac{f(x+h) - f(x)}{h}$$

$$\begin{aligned} f'(x) &= \frac{f(x) - f(x-h)}{h} + \frac{f''(x)}{2}h + \cdots \\ &\approx \frac{f(x) - f(x-h)}{h} \end{aligned}$$

$$\begin{aligned} f'(x) &= \frac{f(x+h) - f(x-h)}{2h} - \frac{f'''(x)}{6}h^2 + \cdots \\ &\approx \frac{f(x+h) - f(x-h)}{2h} \end{aligned}$$

$$\begin{aligned} f''(x) &= \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - \frac{f^{(4)}(x)}{12}h^2 + \cdots \\ &\approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \end{aligned}$$

Recap: Numerical Differentiation

- Differentiation is inherently sensitive to perturbations

- For continuously defined smooth function, finite difference approximations to derivatives can be derived by Taylor series or polynomial interpolation

- Another option is that computer program expressing given function is differentiated step by step to compute derivative

Recap: Example of Richardson Extrapolation

- Use Richardson extrapolation to improve accuracy of finite difference approximation to derivative of function sin(x) at x=1
  - Discussion: what's the result of forward difference approximation with step size 0.5?
- Using first-order accurate forward difference approximation, we have $F(h) = a_0 + a_1 h + \mathcal{O}(h^2)$
  - so $p = 1$ and $r = 2$ in this instance
- Using step sizes of $h = 0.5$ and $h/2 = 0.25$ i.e., $q = 2$, we obtain

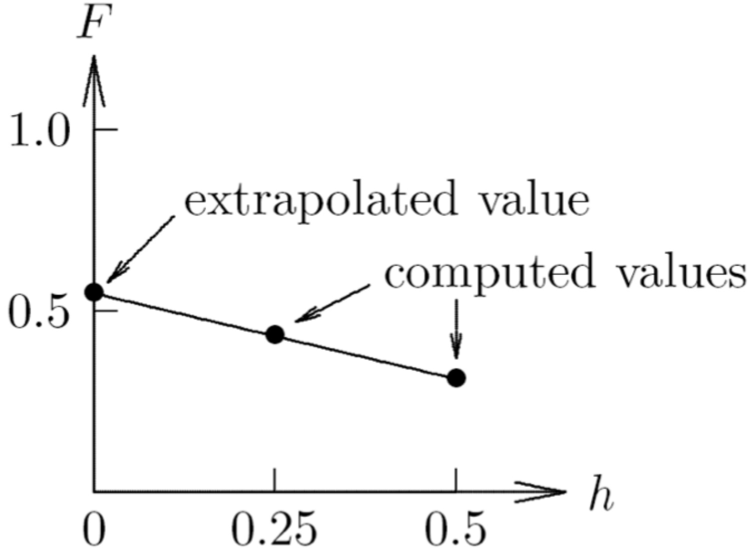
$$\begin{aligned} F(h) &= \frac{\sin(1.5) - \sin(1)}{0.5} = 0.312048 \\ F(h/2) &= \frac{\sin(1.25) - \sin(1)}{0.25} = 0.430055 \end{aligned}$$

Recap: Example of Richardson Extrapolation

- Extrapolated value is then given by

$$F(0) = a_0 = F(h) + \frac{F(h) - F(h/2)}{(1/2) - 1} = 2F(h/2) - F(h) = 0.548061$$

- For comparison, correctly rounded result is $\cos(1) = 0.540302$



Agenda

- Ordinary differential equations (ODE)
  - Applications
  - Initial value problems
    - Euler's method
  - Boundary value problems

- Partial differential equations (PDE)
  - Applications

What’s a differential equation?

- Differential equations involve derivatives of unknown solution function
- Ordinary differential equation (ODE): all derivatives are with respect to single independent variable, often representing time
- Solution of differential equation is continuous function in infinite-dimensional space of functions
- Numerical solution of differential equations is based on finite-dimensional approximation

Example: Newton's Second Law

- $F = ma$
  - second-order ODE, since acceleration a is second derivative of position coordinate, which we denote by y
- Thus, ODE has form

$$y'' = F/m$$

  - where F and m are force and mass, respectively
- Defining $u_1 = y$ and $u_1 = y'$ yields equivalent system of two first-order ODEs

$$\begin{bmatrix} u_1' \\ u_2' \end{bmatrix} = \begin{bmatrix} u_2 \\ F/m \end{bmatrix}$$

- We can now use methods for first-order equations to solve this system
  - First component of solution u_1 is solution y of original second-order equation
  - Second component of solution u_2 is velocity y'

Ordinary Differential Equations

- General first-order system of ODEs has form

$$\boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y})$$

where $\boldsymbol{y} : \mathbb{R} \to \mathbb{R}^n$, $\boldsymbol{f} : \mathbb{R}^{n+1} \to \mathbb{R}^n$, and $\boldsymbol{y}' = d\boldsymbol{y}/dt$ denotes derivative with respect to t,

$$\begin{bmatrix} y_1'(t) \\ y_2'(t) \\ \vdots \\ y_n'(t) \end{bmatrix} = \begin{bmatrix} dy_1(t)/dt \\ dy_2(t)/dt \\ \vdots \\ dy_n(t)/dt \end{bmatrix}$$

  - Function f is given and we wish to determine unknown function y satisfying ODE

A lot of applications of ODEs

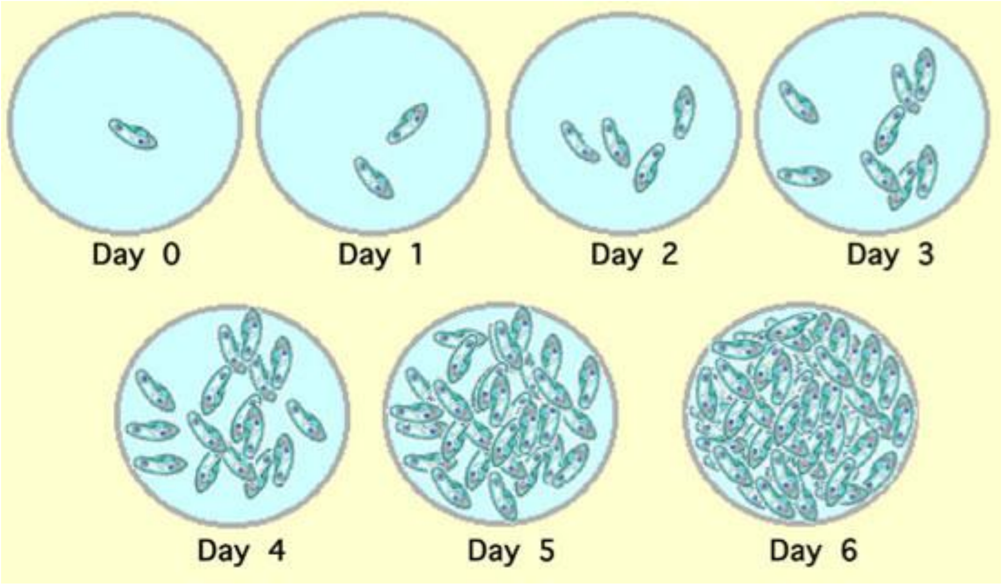
- Newton’s Law of Cooling
  - Cooling of engines, heat sinks, electronics temperature decay.

$$\frac{dT}{dt} = -k(T - T_\infty)$$

  - $T(t)$:object temperature
  - T_∞ :ambient temperature
  - k: heat-transfer coefficient

A lot of applications of ODEs

- Population Growth (Chemical/Biological Engineering)
  - Exponential model $\frac{dN}{dt} = rN$
  - Logistic model $\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right)$



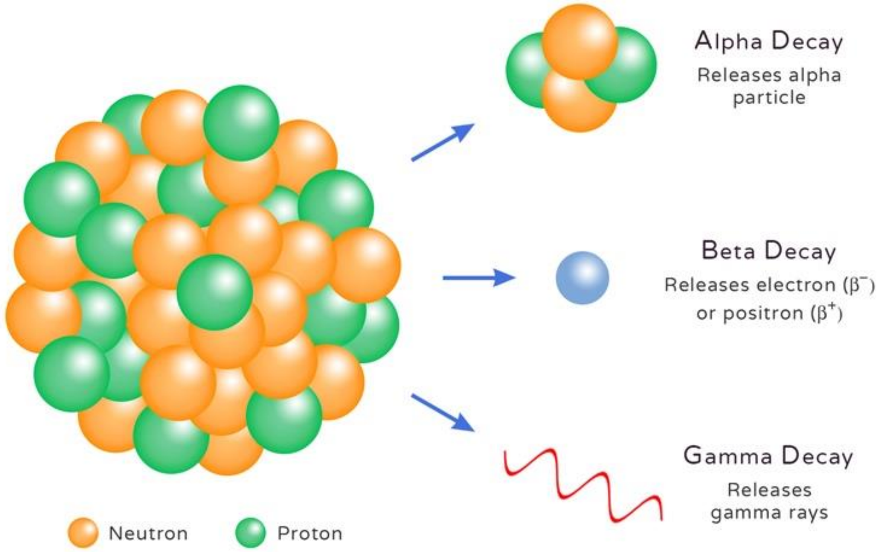
- $N(t)$:population (cells, bacteria, chemical species)
- r: growth rate
- K: carrying capacity

A lot of applications of ODEs

- Radioactive Decay (Nuclear & Medical Engineering)
  - Nuclear reactor design, PET imaging tracers, radiation shielding.

$$\frac{dN}{dt} = -\lambda N$$

  - $N(t)$: amount of radioactive substance
  - λ: decay constant



Initial Value Problems

- By itself, ODE $y' = f(t, y)$ does not determine unique solution function
- This is because ODE merely specifies slope $y'(t)$ of solution function at each point, but not actual value $y(t)$ at any point
- If $y(t)$ is solution and c is any constant, then $y(t) + c$ is also a solution because $d(y(t) + c)/dt = y'(t) + 0 = y'(t)$
- Infinite family of functions satisfies ODE, in general
  - To single out particular solution, value y_0 of solution function must be specified at some point t_0

Initial Value Problems

- Thus, part of given problem data is requirement that $y(t_0) = y_0$, which determines unique solution to ODE

- Because of interpretation of independent variable t as time, we think of t_0 as initial time and y_0 as initial value

- Hence, this is termed initial value problem, or IVP

- ODE governs evolution of system in time from its initial state y_0 at time t_0 onward, and we seek function $y(t)$ that describes state of system as function of time

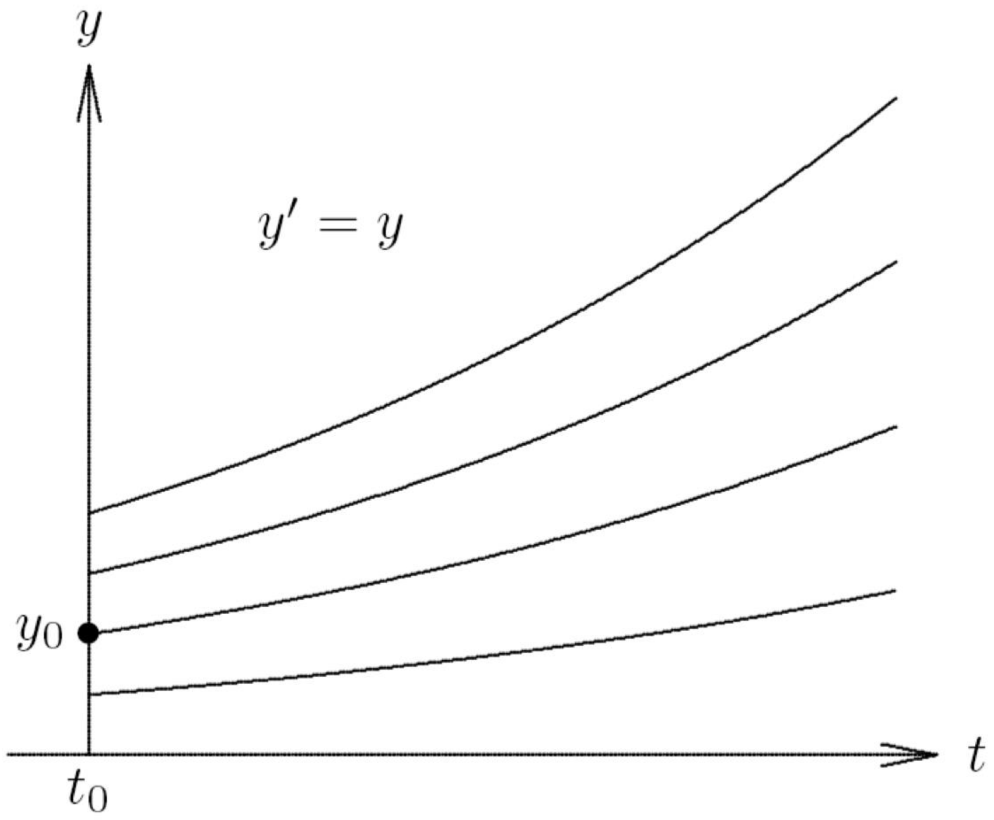
Example of Initial Value Problem

- Consider scalar ODE $y' = y$
- Discussion: Could you try to guess the solution?

- Family of solutions is given by $y(t) = ce^t$, where c is any real constant
- Imposing initial condition $y(t_0) = y_0$ singles out unique particular solution
  - For this example, if $t_0 = 0$, then $\mathrm{c} = y_0$, which means that solution is $y(t) = y_0 e^t$

Example of Initial Value Problem

Family of solutions for ODE $y' = y$



How can we solve an IVP problem?

- Euler's method
  - For general system of ODEs $y' = f(t, y)$, consider Taylor series

$$
\begin{aligned}
\boldsymbol{y}(t+h) &= \boldsymbol{y}(t) + h\boldsymbol{y}'(t) + \frac{h^2}{2}\boldsymbol{y}''(t) + \cdots \\
&= \boldsymbol{y}(t) + h\boldsymbol{f}(t, \boldsymbol{y}(t)) + \frac{h^2}{2}\boldsymbol{y}''(t) + \cdots
\end{aligned}
$$

  - Euler's method results from dropping terms of second and higher order to obtain approximate solution value

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + h_k \boldsymbol{f}(t_k, \boldsymbol{y}_k)$$

In-class exercise of Euler's method

- Consider ODE: $\frac{dy}{dt} = y - t^2 + 1$ with initial condition $y(0) = 0.5$
- Use Euler's Method with step size $h = 0.2$ to approximate $y(1.0)$.
- Solution:

$$y_{n+1} = y_n + hf(t_n, y_n)$$

$$t_{n+1} = t_n + h.$$

$$t = 0,\ 0.2,\ 0.4,\ 0.6,\ 0.8,\ 1.0$$

Iteration 0 → 1

$$f(t_0, y_0) = 0.5 - 0^2 + 1 = 1.5$$

$$y_1 = y_0 + hf(t_0, y_0) = 0.5 + 0.2(1.5) = 0.8$$

Iteration 1 → 2

$$f(t_1, y_1) = 0.8 - 0.2^2 + 1 = 0.8 - 0.04 + 1 = 1.76$$

$$y_2 = y_1 + 0.2(1.76) = 0.8 + 0.352 = 1.152$$

Iteration 2 → 3

$$f(t_2, y_2) = 1.152 - 0.4^2 + 1 = 1.152 - 0.16 + 1 = 1.992$$

$$y_3 = 1.152 + 0.2(1.992) = 1.152 + 0.3984 = 1.5504$$

Iteration 3 → 4

$$f(t_3, y_3) = 1.5504 - 0.6^2 + 1 = 1.5504 - 0.36 + 1 = 2.1904$$

$$y_4 = 1.5504 + 0.2(2.1904) = 1.5504 + 0.43808 = 1.98848$$

Iteration 4 → 5

$$f(t_4, y_4) = 1.98848 - 0.8^2 + 1 = 1.98848 - 0.64 + 1 = 2.34848$$

$$y_5 = 1.98848 + 0.2(2.34848) = 1.98848 + 0.469696 = 2.458176$$

Boundary Value Problems

- Side conditions prescribing solution or derivative values at specified points are required to make solution of ODE unique
- For initial value problem, all side conditions are specified at single point, say t_0
- For boundary value problem (BVP), side conditions are specified at more than one point
- For ODEs, side conditions are typically specified at end points of interval [a,b], so we have two-point boundary value problem with boundary conditions (BC) at a and b.

Example of boundary value problems

- Two-point BVP for second-order scalar ODE
  - $u'' = f(t, u, u'), \quad a < t < b$
- with boundary conditions

Partial differential equations (PDEs)

- Partial differential equations (PDEs) involve partial derivatives with respect to more than one independent variable

- Independent variables typically include one or more space dimensions possibly time dimension as well

- More dimensions complicate problem formulation: we can have
  - pure initial value problem
  - pure boundary value problem
  - or mixture of both

Partial Differential Equations

- For simplicity , we will deal only with PDEs with only two independent variables, either
  - two space variables, denoted by x and y
  - or one space variable denoted by x and one time variable denoted by t

- Partial derivatives with respect to independent variables are denoted by subscripts, for example

$$u_t = \partial u / \partial t$$

$$u_{xy} = \partial^2 u / \partial x \partial y$$

Example: Advection Equation

- $u_t = -cu_x$
  - where c is nonzero constant, with spatial domain R and $t \geq 0$

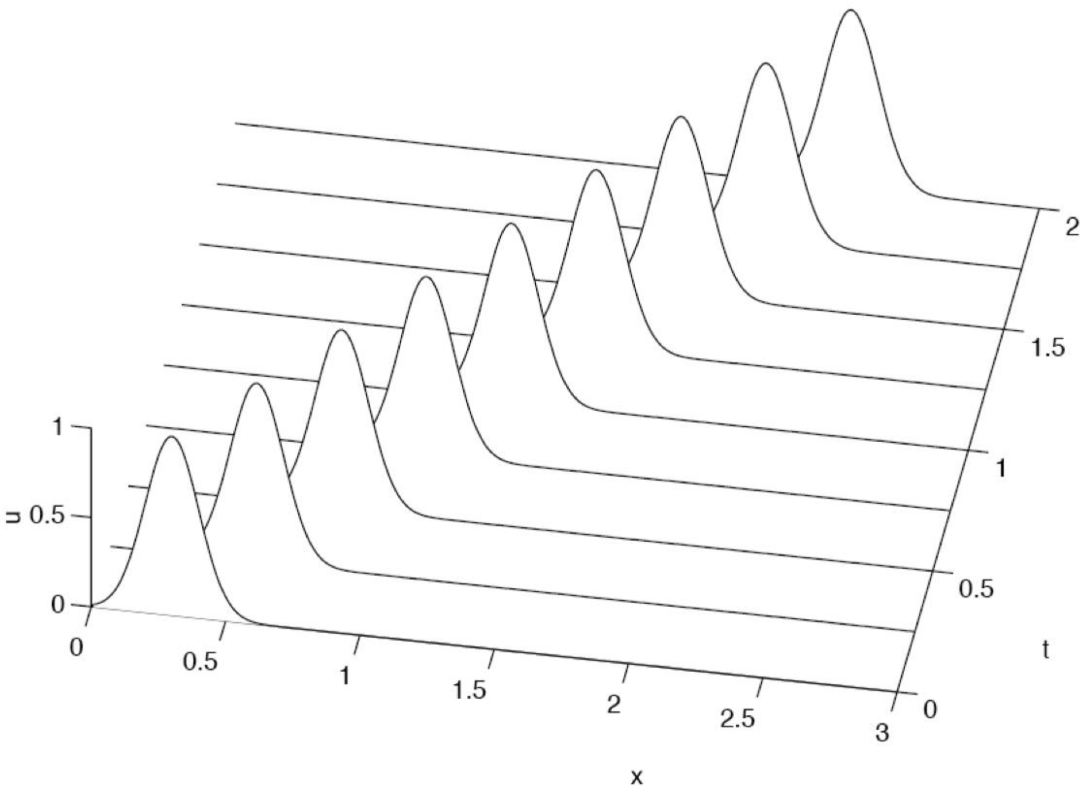
- Unique solution is determined by initial condition

$$u(0, x) = u_0(x), \qquad -\infty < x < \infty$$

  - where u_0 is given function defined on R

- We seek solution $u(t, x)$ for $t \geq 0$ and all $x \in R$
- From chain rule, solution is given by $u(t, x) = u_0(x - ct)$
  - Solution is initial function u_0 shifted by ct to right if $c > 0$, or to left if $c < 0$

Example: Advection Equation



Typical solution of advection equation, with initial function "advected" (shifted) over time

Classification of PDEs

- Second-order linear PDEs of general form

$$au_{xx} + bu_{xy} + cu_{yy} + du_x + eu_y + fu + g = 0$$

- are classified by value of discriminant $b^2 - 4ac$

$$b^2 - 4ac > 0: \ \textit{hyperbolic} \ \text{(e.g., wave equation)}$$

$$b^2 - 4ac = 0: \ \textit{parabolic} \ \text{(e.g., heat equation)}$$

$$b^2 - 4ac < 0: \ \textit{elliptic} \ \text{(e.g., Laplace equation)}$$

Classification of PDEs

- Classification of more general PDEs is not so clean and simple, but roughly speaking

  - Hyperbolic PDEs describe time-dependent, conservative physical processes, such as convection, that are not evolving toward steady state

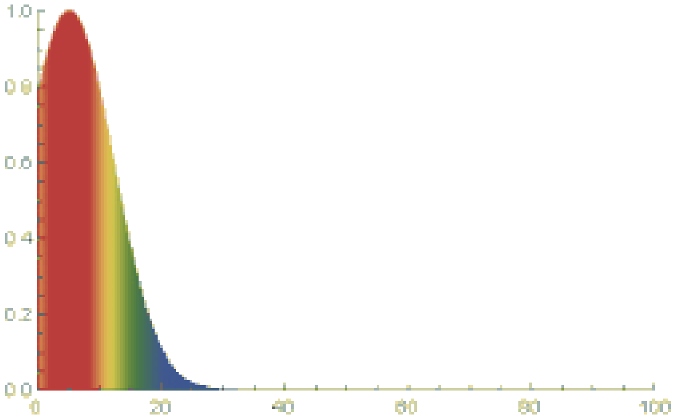
  - Parabolic PDEs describe time-dependent, dissipative physical processes, such as diffusion, that are evolving toward steady state

  - Elliptic PDEs describe processes that have already reached steady state, and hence are time-independent

A lot of applications of PDEs

- Heat Equation (Diffusion Equation)
  - **Heat conduction in a metal rod or engine block.**
    Predicts how heat diffuses through the material after heating one end.

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T$$



- $T = T(x, y, z, t)$: **temperature field** (°C or K)
- $\frac{\partial T}{\partial t}$: **rate of temperature change over time**
- α: **thermal diffusivity** $(\alpha = k/(\rho c))$
  - k: thermal conductivity
  - ρ: density
  - c: specific heat
- ∇^2: **Laplacian operator**

$$\nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}$$

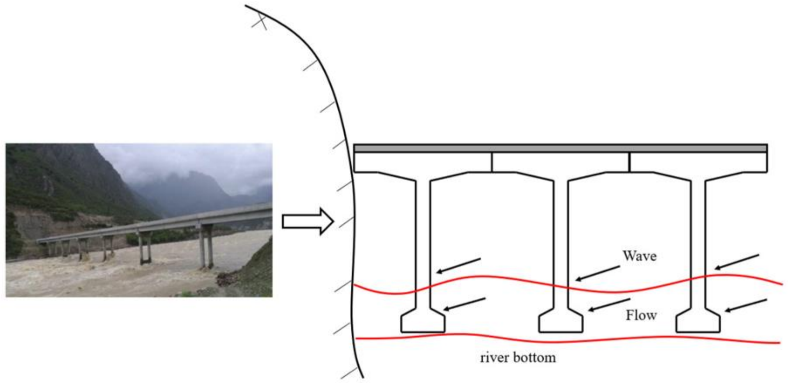
- Units: α [m²/s], $\nabla^2 T$ [K/m²]

A lot of applications of PDEs

- Wave Equation (Vibration / Acoustics)
  - **Vibration of a bridge, airplane wing, or building.**
    Predicts natural frequencies and vibration modes.

$$\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u$$

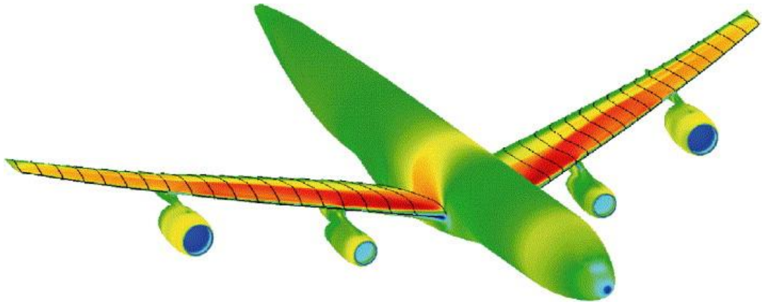
- $u(x, y, z, t)$: **displacement** (m) of a point on a structure
- $\dfrac{\partial^2 u}{\partial t^2}$: **acceleration**
- c: **wave propagation speed** (depends on tension, density, or material stiffness)
- $\nabla^2 u$: spatial curvature; where displacement is "bending" or "curving"



A lot of applications of PDEs

- Navier–Stokes Equations (Fluid Dynamics)
  - **Aerospace aerodynamics:** simulation of airflow over an aircraft wing.

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = -\nabla p + \mu \nabla^2 \mathbf{v}$$



- $\mathbf{v} = (v_x, v_y, v_z)$: **velocity field**
- ρ: **density**
- p: **pressure field**
- μ: **dynamic viscosity**
- $\mathbf{v} \cdot \nabla \mathbf{v}$: **convection of momentum**
- $-\nabla p$: force due to pressure
- $\mu \nabla^2 \mathbf{v}$: **viscous diffusion** of momentum
- LHS: "material acceleration"

A lot of applications of PDEs

- Advection–Diffusion Equation (Transport of Pollutants)
  - Modeling pollutant concentration in rivers, smoke in air, or heat in fluids.

$$\frac{\partial C}{\partial t} + \mathbf{v} \cdot \nabla C = D \nabla^2 C$$

- $C(x, y, z, t)$: **concentration** of pollutant or chemical species
- $\frac{\partial C}{\partial t}$: local accumulation
- $\mathbf{v} = (v_x, v_y, v_z)$: **velocity field** (flow speed of water/air)
- $\mathbf{v} \cdot \nabla C$: **advection term** (transport due to flow)

$$\mathbf{v} \cdot \nabla C = v_x \frac{\partial C}{\partial x} + v_y \frac{\partial C}{\partial y} + v_z \frac{\partial C}{\partial z}$$

- D: **diffusion coefficient**
- $D \nabla^2 C$: spreading due to diffusion/dispersion