MATLAB 工程应用

Partial Differential Equations

CAE(Computer Aided Engineering) for Engineers

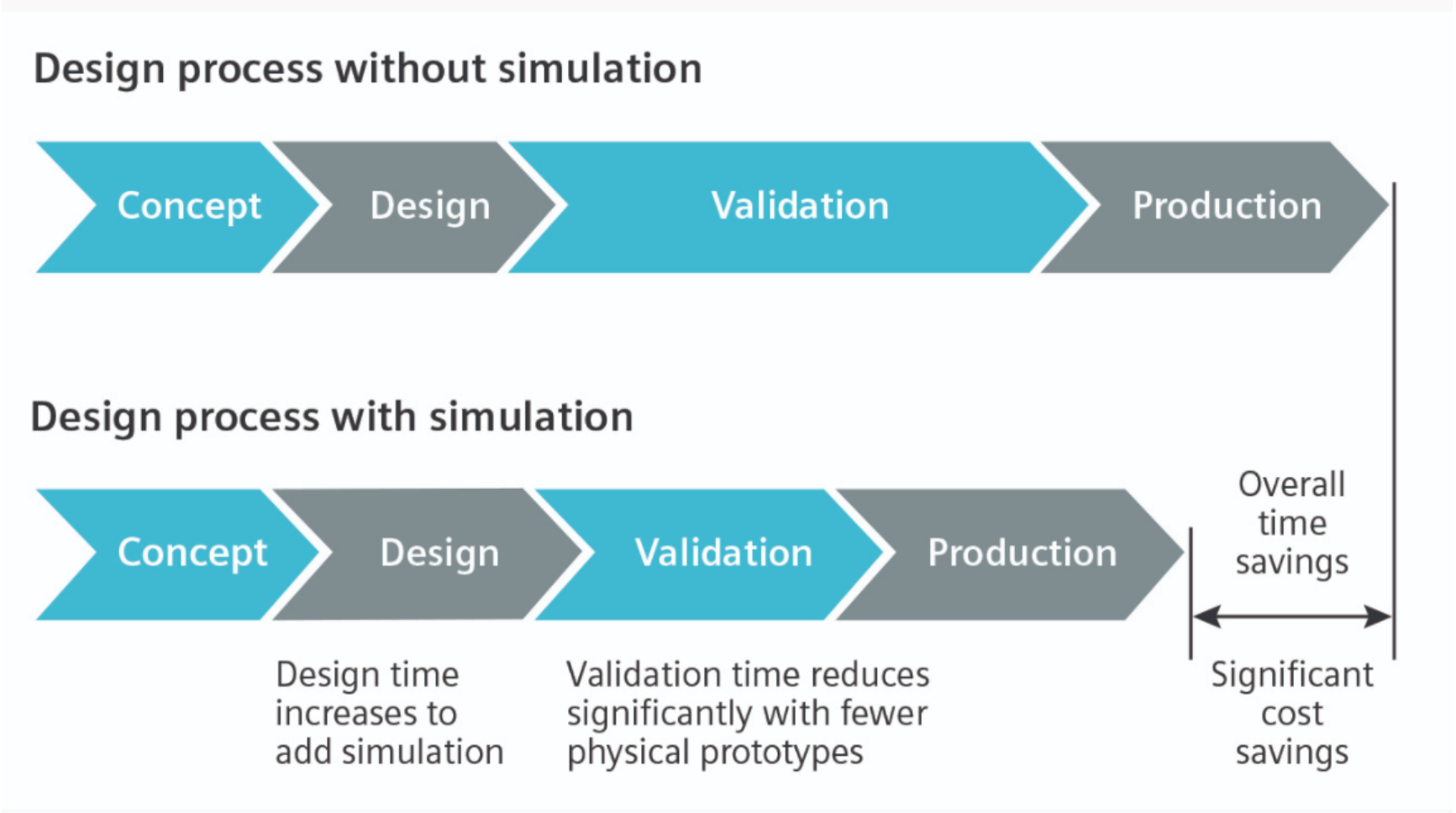
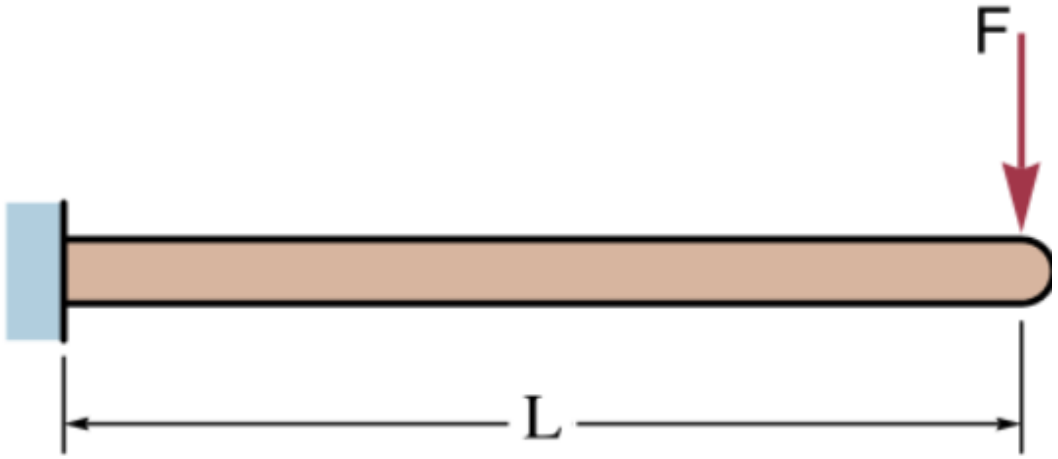
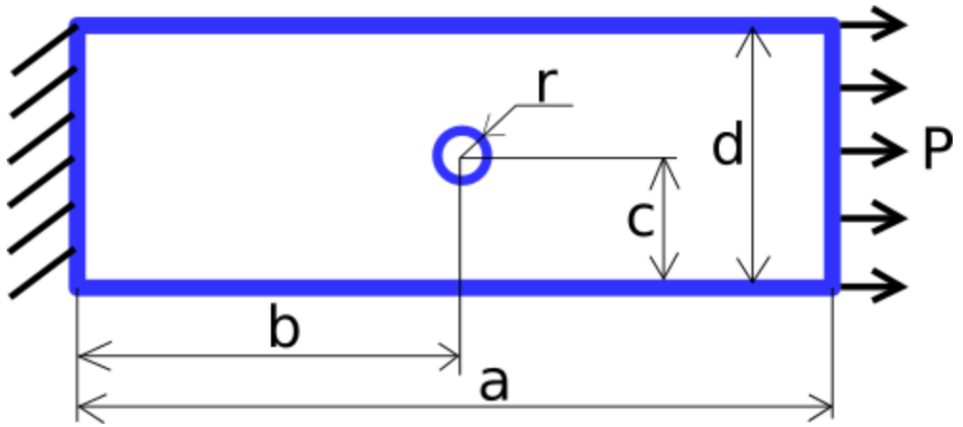
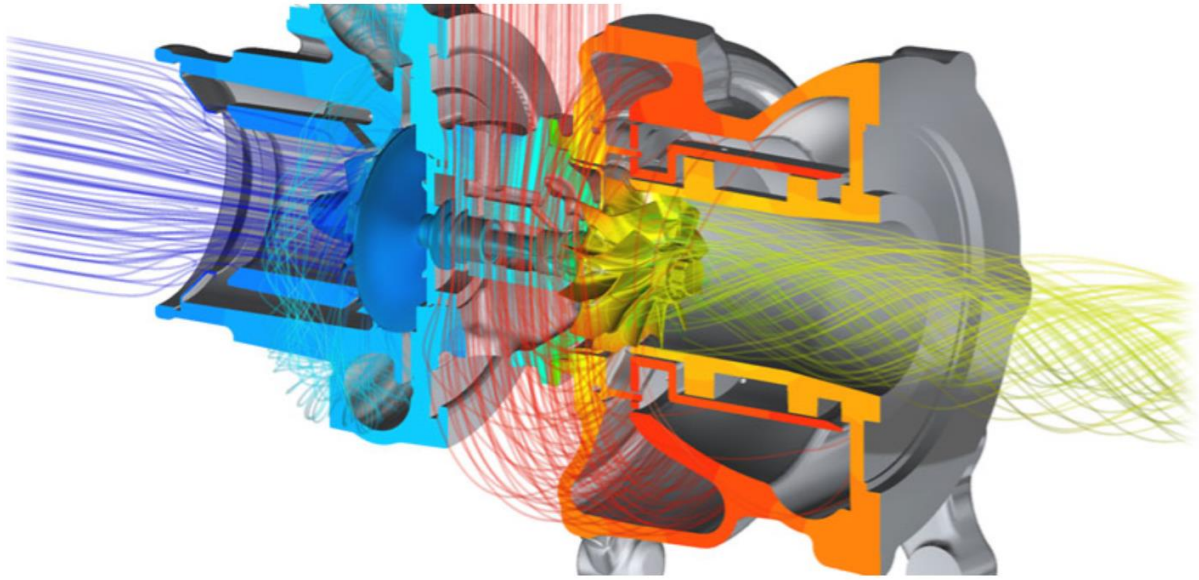


Fig. 1: Speeding up time-to-market through virtual testing instead of physical prototypes for design validation.

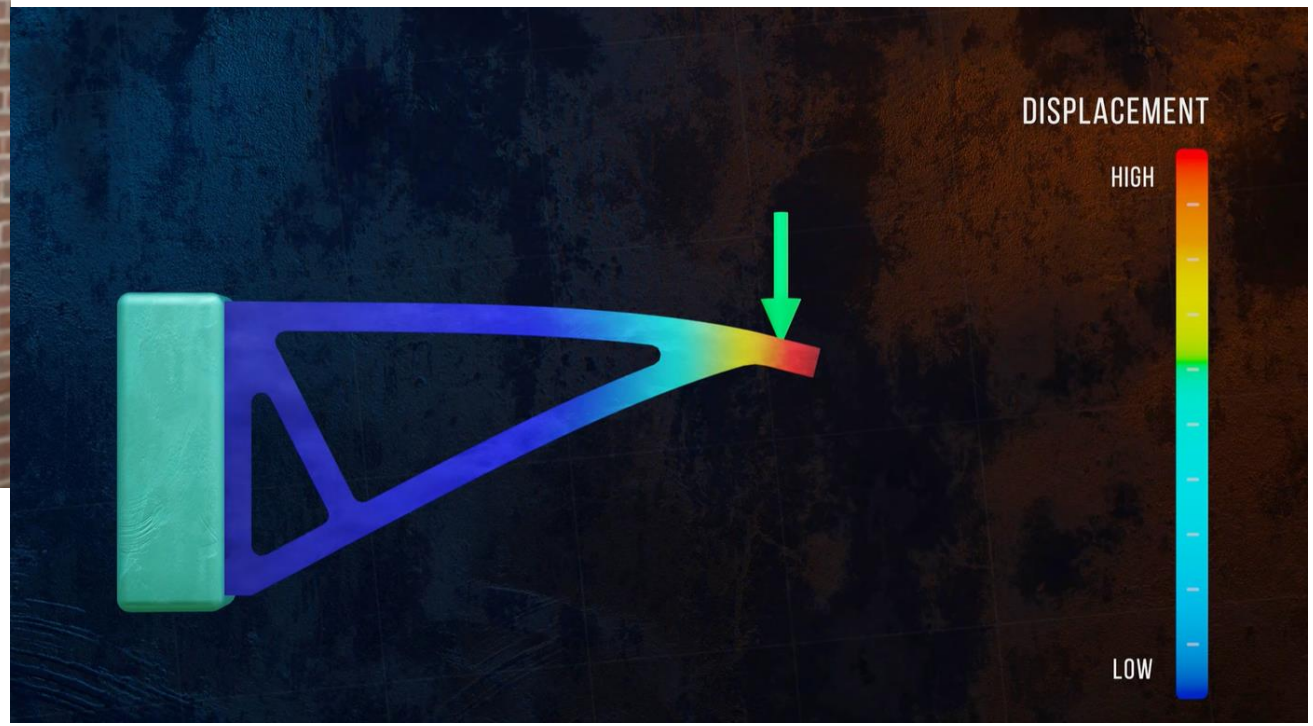
Background



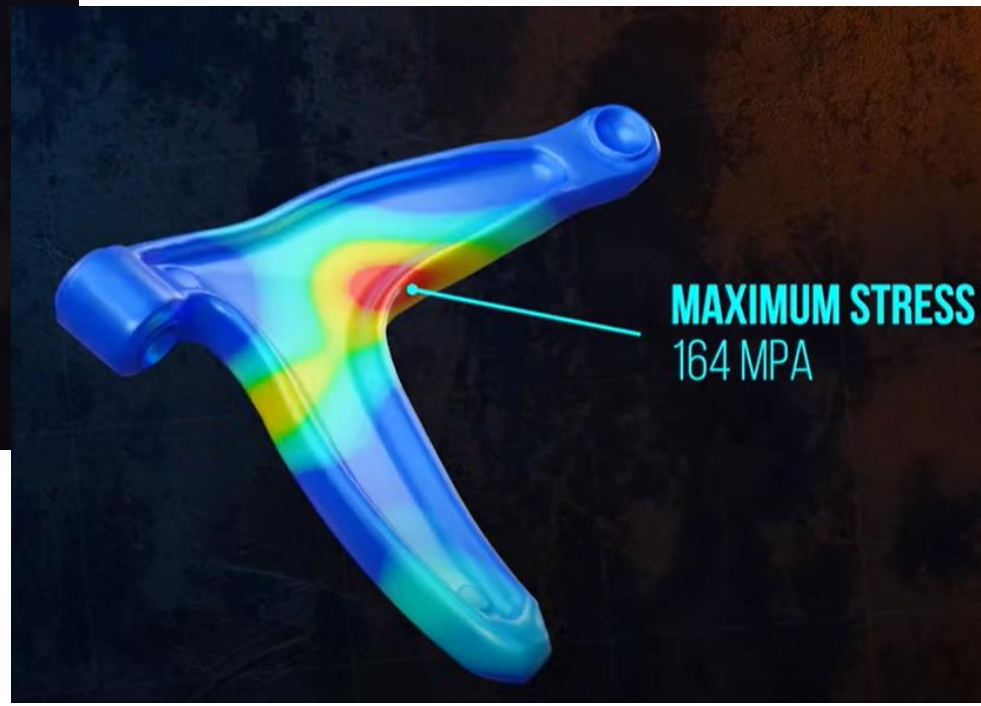




Background



Background



CAE analysis

- **Solid Mechanics**
  - Static Analysis
  - Dynamic Analysis
  - Buckling Analysis

- **Fluid Mechanics**
  - Heat Transfer
  - Flow
  - Electromagnetics

Car's door attached to an electromagnetic shaker

Engineering Problem Solution

- **Analytical Method**
  Classical method, 100% accurate results
  Closed form solution
  Applicable only for simple problems like cantilever and simply supported beams, etc

- **Numerical Method**
  Mathematical representation
  Approximate, assumptions made
  Real life complex problems
  Results cannot be believed blindly. **Certain results must be validated by experiments and /or analytical method**
  FEM, FVM, FDM

- **Experimental Method**
  Actual measurement
  Time consuming and expensive set up
  Applicable only if physical prototypes is available
  Results cannot be believed blindly and a minimum of 3 to 5 prototypes must be tested
  Strain gauges, sensors, accelerometers

Widely Used CAE SoftWares

CAD Software's		CAE Software's	
Commercials:	Freeware and open source:	Commercials	Freeware and open source:
• AutoCAD/Inventor (Autodesk)	• 123D	• ANSYS/Fluent/CFX	• CalculiX
• Creo/Pro-Engineer(PTC)	• LibreCAD	• Abaqus	• JCMsuite
• Unigraphics/Solid Edge (Siemens)	• FreeCAD	• Nastran/Patran	• ANSYS (student version, 32000 nodes)
• CATIA/Solidworks/Draftsight (Dassault systems)	• BRL-CAD	• Hypermesh	• FEBio
• Spaceclaim/Design Modeler (ANSYS)	• OpenSCAD	• Radioss	• FeatFlow
• MicroStation(Bentley)	• QCad	• LS-Dyna	• SimScale
• TurboCAD Pro	• SolveSpace	• Simufact Welding	• Z88/Z88Aurora
• Onshape		• Autodesk CFD/Moldflow	• Fidesys (Academic version)
		• Modex3D	• VisualFEA (Educational version)
		• ANSA	• CAEplex (Free Plan Available)
		• COMSOL Multiphysics	

Types of Analysis

Static Analysis	Dynamic Vibration Analysis	Explicit Dynamic Analysis	Fatigue / Durability	Thermal Analysis	Optimization	High End Analysis	Computational Fluid Dynamics
•Linear Analysis	•Modal Analysis	•Impact Analysis	•Stress-Life Approach	•Steady State Thermal Analysis	•Topology Optimization	•Mold Flow Analysis	•Pressure drop ration & Flow characteristic
•Non Linear Analysis	•Transient Dynamic Analysis	•Drop Test	•Strain-Life Approach	•Transient Thermal Analysis	•Size and Shape Optimization	•Multi-physics analysis	•Cavitation study
•Quasi-Static Analysis	•Frequency Response Analysis	•Can Crush	•ASME standard Fatigue Analysis	•Coupled / Thermal Stress Analysis	•Topography Optimization	•Manufacturing Simulations (Weld analysis)	•Thermal, transient flow analysis
•Contact Bolt-joint Simulations	•Spectrum Analysis	•Projectile Impact	•DNV standard Fatigue Analysis	•Transient Weld Analysis		•NVH	•Laminar turbulent flow
•Leakage, Interference Analysis	•Multi Body Dynamics	•Car Crash Analysis					•Flow coefficient & pressure recovery

Different Numerical Methods

- Finite Element Method (FEA)

  The Finite Element Method (FEM) is popular numerical technique used to determine the approximated solution for a partial differential equations(PDE)

  Applications-Linear, nonlinear, buckling, thermal, dynamic and fatigue analysis, etc

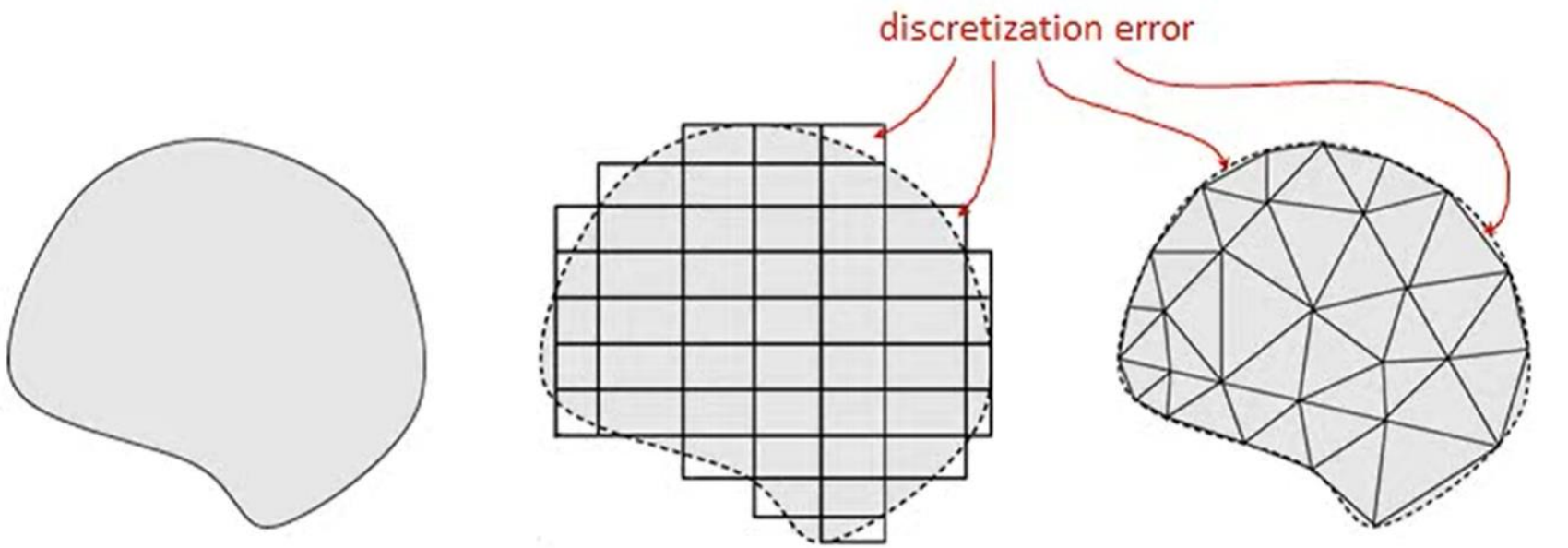
- Finite Difference Method (FDM)

  It uses Taylor's series to convert a differential equation to an algebraic equation. In the conversion process, higher order terms are neglected.

  It is used to solve thermal and CFD couple problems.

FEM Vs. Finite-Difference Grids

The finite element mesh offers minimal discretization error.



Arbitrary Object

Finite-Difference Discretization
- *Simpler to implement*
- *Larger error*

Triangular Discretization
- *More complex implementation*
- *Smaller error*

PDEs (Partial Differential Equations)

A differential equation involving partial derivatives of a dependent variable(one or more) with more than one independent variable is called a partial differential equation.

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0 \qquad \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \qquad ux\frac{\partial^2 u}{\partial x^2} + u^2 xy\frac{\partial^2 u}{\partial x \partial y} + uy\frac{\partial^2 u}{\partial y^2} + \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + u^3 = 0$$

$$\frac{\partial u}{\partial t} - T\frac{\partial^2 u}{\partial x^2} = 0 \qquad \left(x^2 + y^2\right)\frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial x \partial y} - 3u = 0 \qquad \frac{\partial^2 u}{\partial x^2} + \left(\frac{\partial^2 u}{\partial x \partial y}\right)^2 + \frac{\partial^2 u}{\partial y^2} = x^2 + y^2$$

Order of a PDE: The order of the highest derivative term in the equation is called the order of the PDE

Linear PDE: If the dependent variable and all its partial derivatives occur linearly in any PDE then such an equation is called linear PDE otherwise a non-linear PDE.

Governing Equations

Conservation Equations

Continuity

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0$$

Momentum

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial P}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j}$$

Energy

$$\frac{\partial}{\partial t}(\rho h_{tot}) + \frac{\partial}{\partial x_j}(\rho h_{tot} u_j) = \frac{\partial P}{\partial t} + \frac{\partial}{\partial x_j}(u_i \tau_{ij} + \lambda \frac{\partial T}{\partial x_j})$$

where

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{2}{3}\delta_{ij}\frac{\partial u_i}{\partial x_j}\right) \qquad h_{tot} = h + \frac{1}{2}u_i^2$$

General form of PDE

$$A\frac{\partial^2 u}{\partial x^2} + B\frac{\partial^2 u}{\partial x \partial y} + C\frac{\partial^2 u}{\partial y^2} + D = 0$$

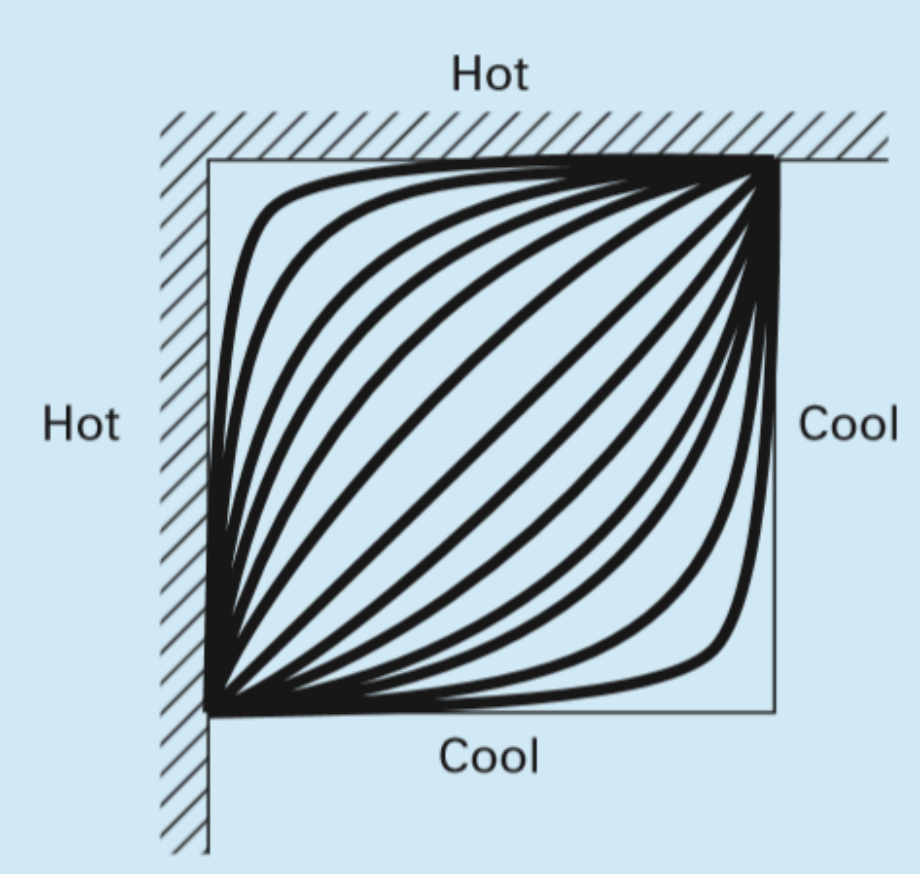
Categories into which linear, second-order partial differential equations in two variables can be classified.

$B^2 - 4AC$	Category	Example
< 0	Elliptic	Laplace equation (steady state with two spatial dimensions) $\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$
$= 0$	Parabolic	Heat conduction equation (time variable with one spatial dimension) $\frac{\partial T}{\partial t} = k' \frac{\partial^2 T}{\partial x^2}$
> 0	Hyperbolic	Wave equation (time variable with one spatial dimension) $\frac{\partial^2 y}{\partial x^2} = \frac{1}{c^2} \frac{\partial^2 y}{\partial t^2}$

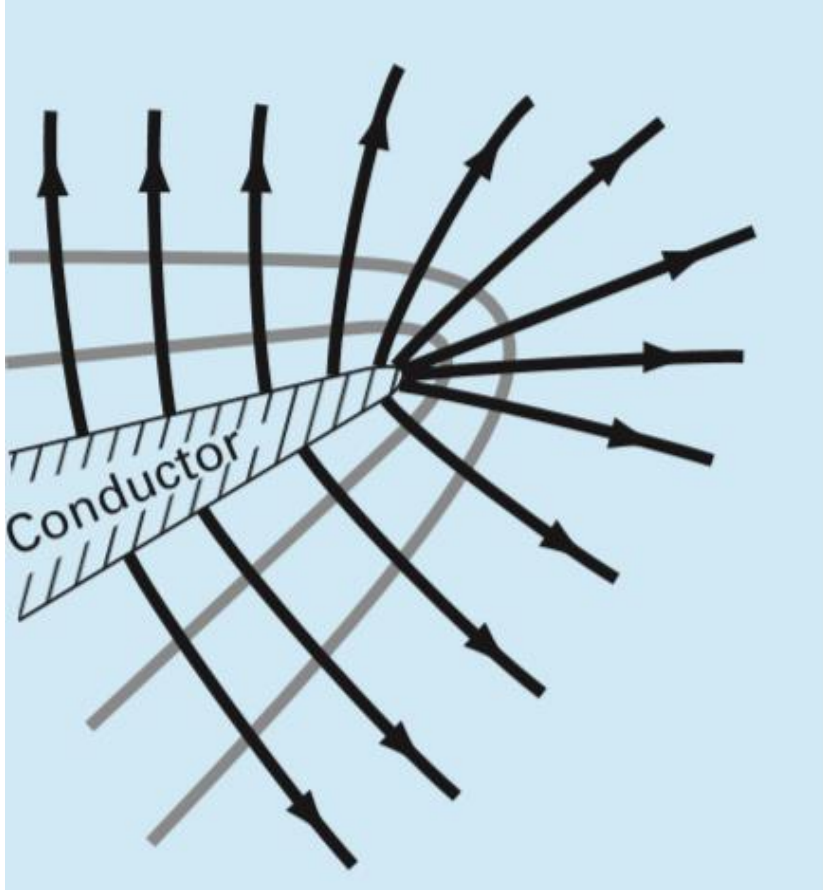
where A, B, and C are functions of x and y and D is a function of x, y, u, $\partial u/\partial x$, and $\partial u/\partial y$.

Elliptic equation

Elliptic equations are typically used to characterize steady-state systems.

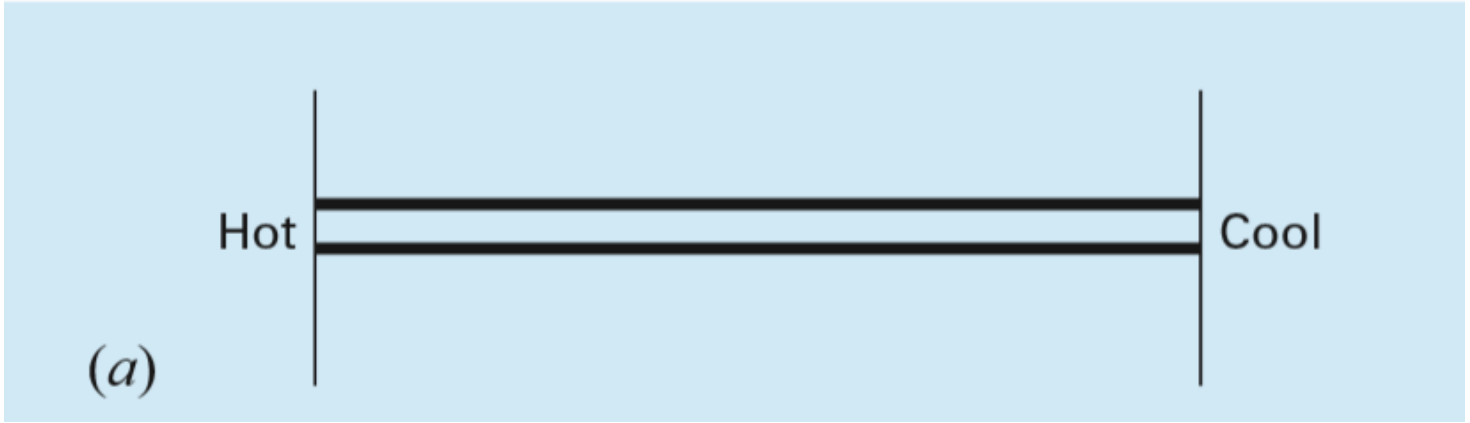


Temperature distribution on a heated plate

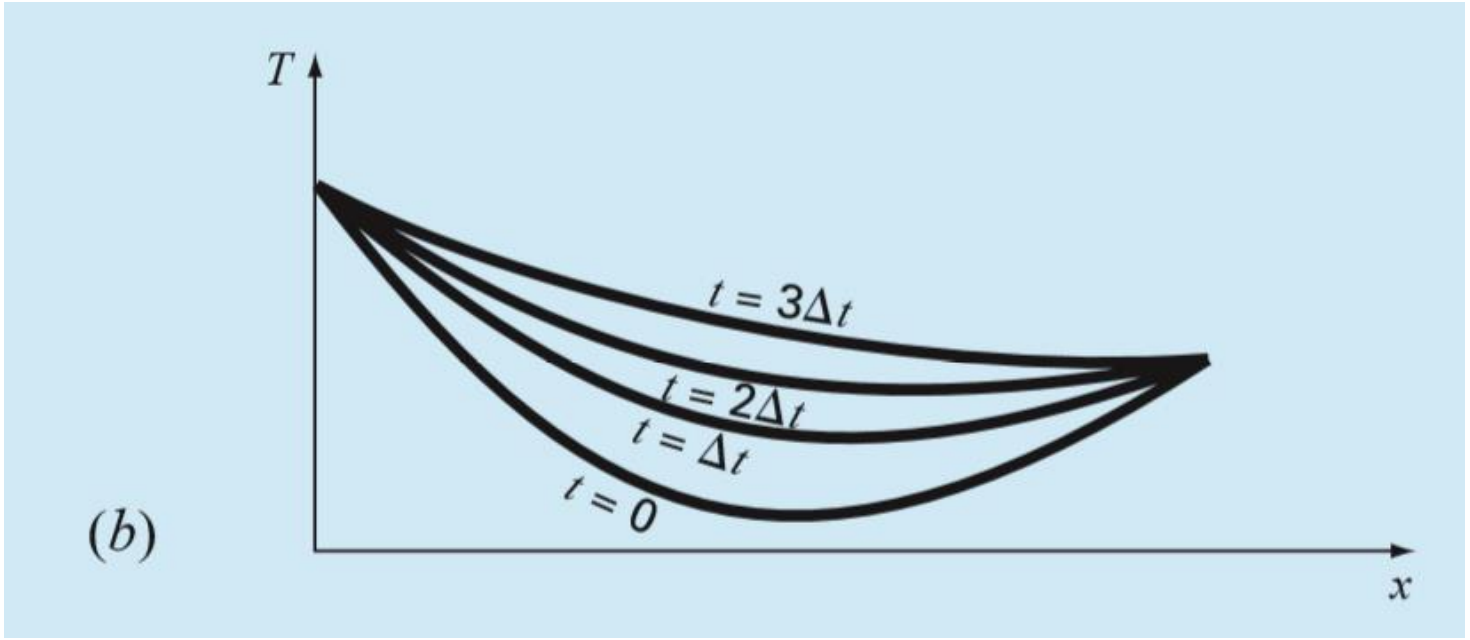


Electrical field near the point of a conductor

Parabolic equation

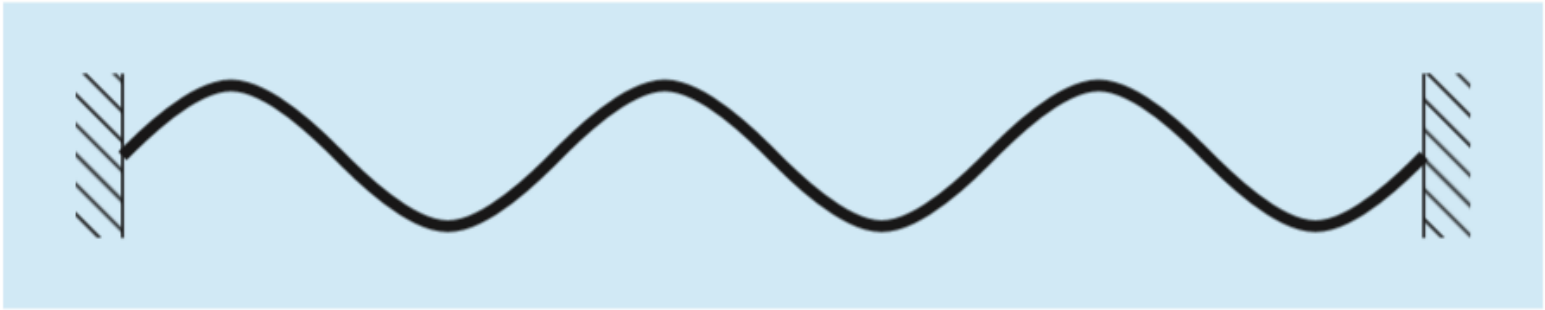


A long, thin rod that is insulated everywhere but at its end.



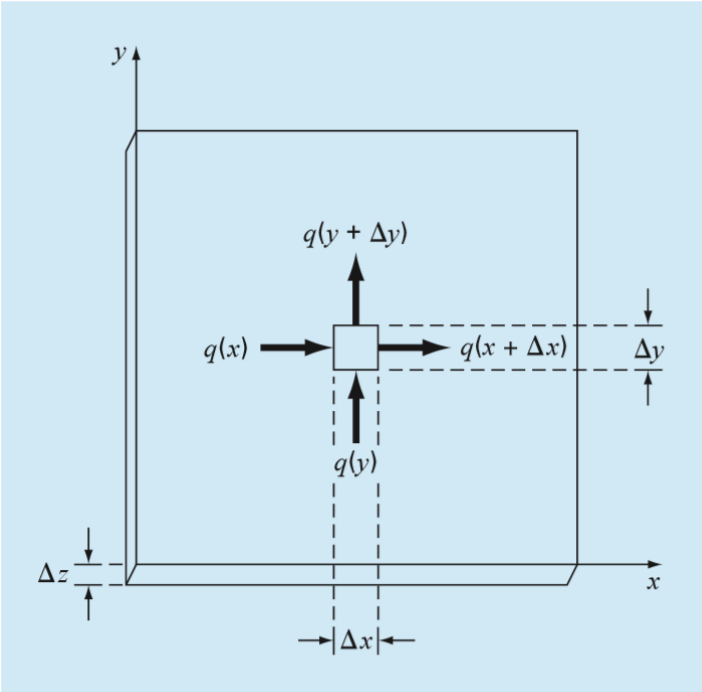
The solution, consisting of distributions corresponding to the state of the rod at various times.

Hyperbolic equation



A taut string vibrating at a low amplitude

Finite Difference: Elliptic Equations



A thin plate of thickness Δz. An element is shown about which a heat balance is taken

Heat balance

$$q(x)\,\Delta y\,\Delta z\,\Delta t + q(y)\,\Delta x\,\Delta z\,\Delta t = q(x+\Delta x)\,\Delta y\,\Delta z\,\Delta t + q(y+\Delta y)\Delta x\,\Delta z\,\Delta t$$

$$[q(x) - q(x+\Delta x)]\Delta y + [q(y) - q(y+\Delta y)]\Delta x = 0$$

$$\frac{q(x) - q(x+\Delta x)}{\Delta x}\,\Delta x\,\Delta y + \frac{q(y) - q(y+\Delta y)}{\Delta y}\,\Delta y\,\Delta x = 0$$

$$-\frac{\partial q}{\partial x} - \frac{\partial q}{\partial y} = 0$$

Fourier's law of heat conduction

$$q_i = -k\rho C\frac{\partial T}{\partial i}$$

where q_i = heat flux in the direction of the i dimension [cal/(cm^2 · s)], k = coefficient of *thermal diffusivity* (cm^2/s), ρ = density of the material (g/cm^3), C = heat capacity of the material [cal/(g · °C)], and T = temperature (°C), which is defined as

Finite Difference-Elliptic Equations

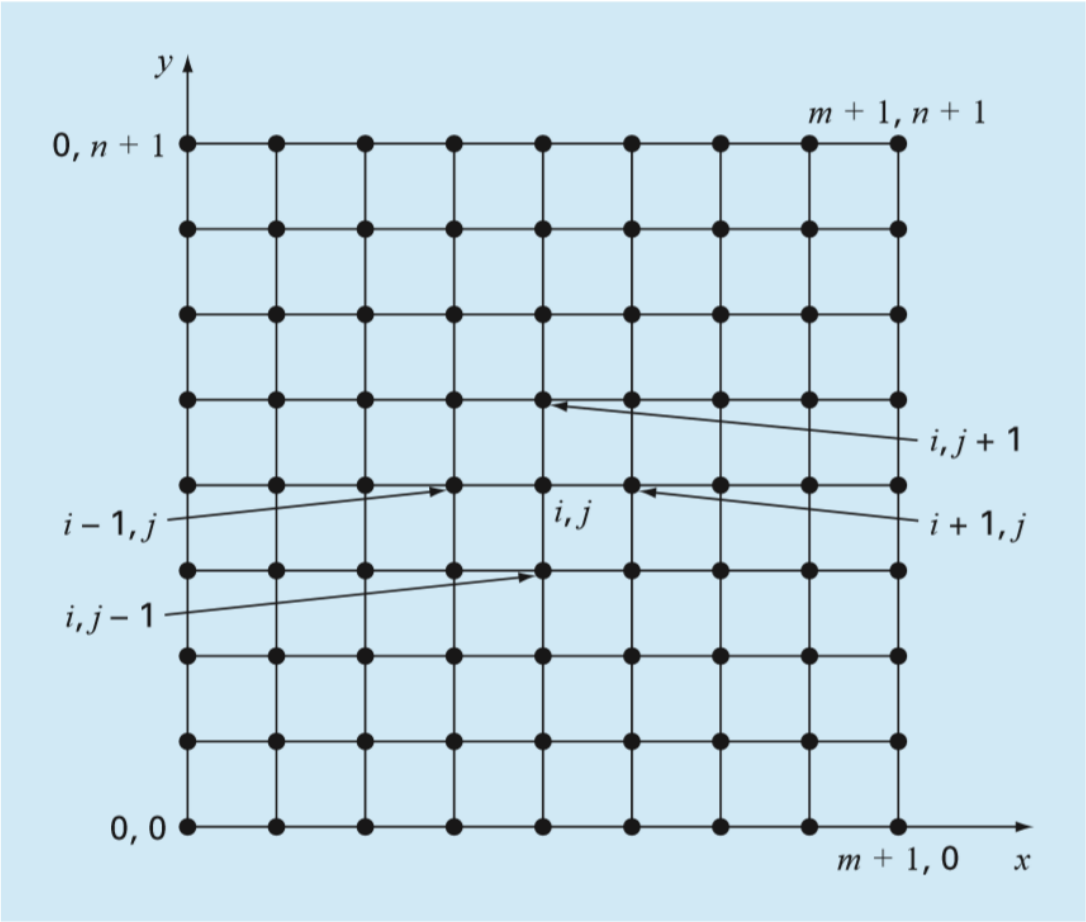
Laplace equation

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

Poisson equation

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = f(x, y)$$

Solution Technique



The Laplacian Differential Equation

Central Differentiation

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2}$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{\Delta y^2}$$

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{\Delta y^2} = 0$$

$$T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4T_{i,j} = 0$$

The Laplacian Differential Equation

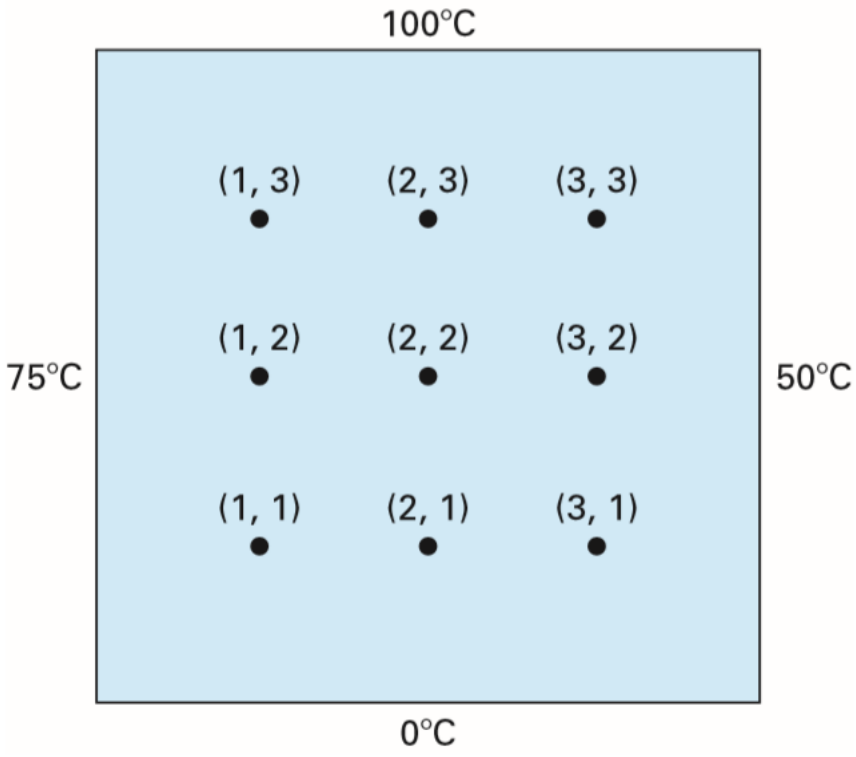
$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

$$T_{21} + T_{01} + T_{12} + T_{10} - 4T_{11} = 0$$

$T_{01} = 75$ and $T_{10} = 0$,

$$-4T_{11} + T_{12} + T_{21} = -75$$

$$\begin{array}{rrrrrrrrr}
4T_{11} & -T_{21} & & -T_{12} & & & & & = 75 \\
-T_{11} & +4T_{21} & -T_{31} & & -T_{22} & & & & = 0 \\
 & -T_{21} & +4T_{31} & & & -T_{32} & & & = 50 \\
-T_{11} & & & +4T_{12} & -T_{22} & & -T_{13} & & = 75 \\
 & -T_{21} & & -T_{12} & +4T_{22} & -T_{32} & & -T_{23} & = 0 \\
 & & -T_{31} & & -T_{22} & +4T_{32} & & & -T_{33} = 50 \\
 & & & -T_{12} & & & +4T_{13} & -T_{23} & = 175 \\
 & & & & -T_{22} & & -T_{13} & +4T_{23} & -T_{33} = 100 \\
 & & & & & -T_{32} & & -T_{23} & +4T_{33} = 150
\end{array}$$



Gauss-Seidel: The Liebmann Method

$$T_{i,j} = \frac{T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1}}{4}$$

Overrelaxation is sometimes employed to accelerate the rate of convergence by applying the following formula after iteration

$$T_{i,j}^{\text{new}} = \lambda T_{i,j}^{\text{new}} + (1 - \lambda) T_{i,j}^{\text{old}}$$

where $T_{i,j}^{\text{new}}$ and $T_{i,j}^{\text{old}}$ are the values of $T_{i,j}$ from the present and the previous iteration, respectively, and λ is a weighting factor that is set between 1 and 2.

As with the conventional Gauss-Seidel method, the iterations are repeated until the absolute values of all the percent relative errors $(\varepsilon_a)_{i,j}$ fall below a prespecified stopping criterion ε_s.These percent relative errors are estimated by

$$|(\varepsilon_a)_{i,j}| = \left| \frac{T_{i,j}^{\text{new}} - T_{i,j}^{\text{old}}}{T_{i,j}^{\text{new}}} \right| 100\%$$

Temperature of a Heated Plate with Fixed Boundary Conditions with Liebmann's method (Gauss-Seidel)

$$T_{i,j} = \frac{T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1}}{4}$$

$$T_{11} = \frac{0 + 75 + 0 + 0}{4} = 18.75$$

and applying overrelaxation yields

$$T_{11} = 1.5(18.75) + (1 - 1.5)0 = 28.125$$

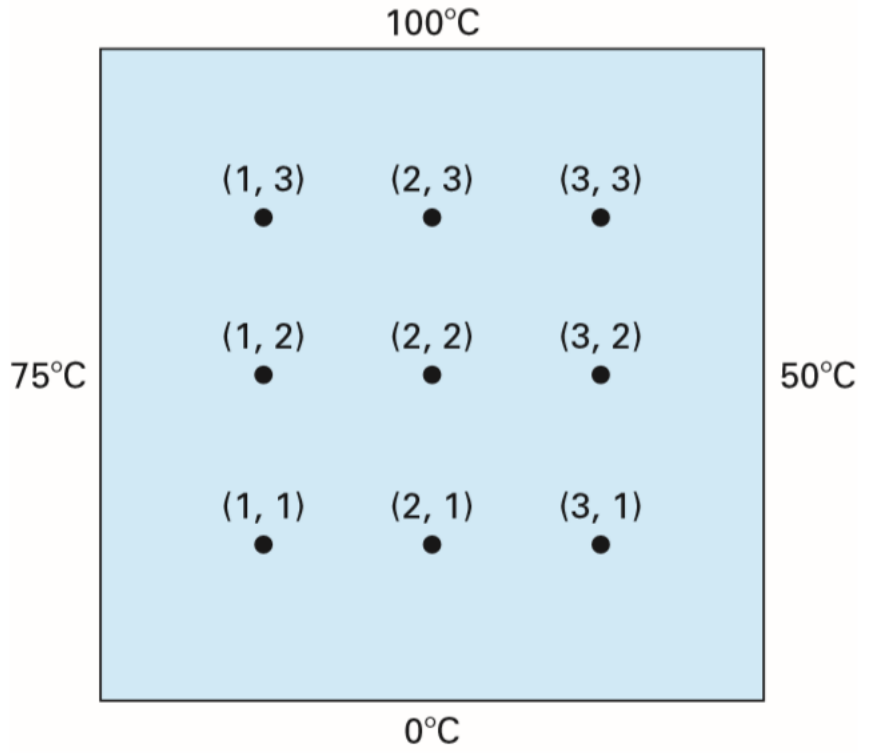
For $i = 2, j = 1$,

$$T_{21} = \frac{0 + 28.125 + 0 + 0}{4} = 7.03125$$

$$T_{21} = 1.5(7.03125) + (1 - 1.5)0 = 10.54688$$

For $i = 3, j = 1$,

$$T_{31} = \frac{50 + 10.54688 + 0 + 0}{4} = 15.13672$$



$$T_{31} = 1.5(15.13672) + (1 - 1.5)0 = 22.70508$$

The computation is repeated for the other rows to give

$$T_{12} = 38.67188 \qquad T_{22} = 18.45703 \qquad T_{32} = 34.18579$$
$$T_{13} = 80.12696 \qquad T_{23} = 74.46900 \qquad T_{33} = 96.99554$$

Because all the $T_{i,j}$'s are initially zero, all ε_a's for the first iteration will be 100%.

For the second iteration the results are

$$T_{11} = 32.51953 \qquad T_{21} = 22.35718 \qquad T_{31} = 28.60108$$
$$T_{12} = 57.95288 \qquad T_{22} = 61.63333 \qquad T_{32} = 71.86833$$
$$T_{13} = 75.21973 \qquad T_{23} = 87.95872 \qquad T_{33} = 67.68736$$

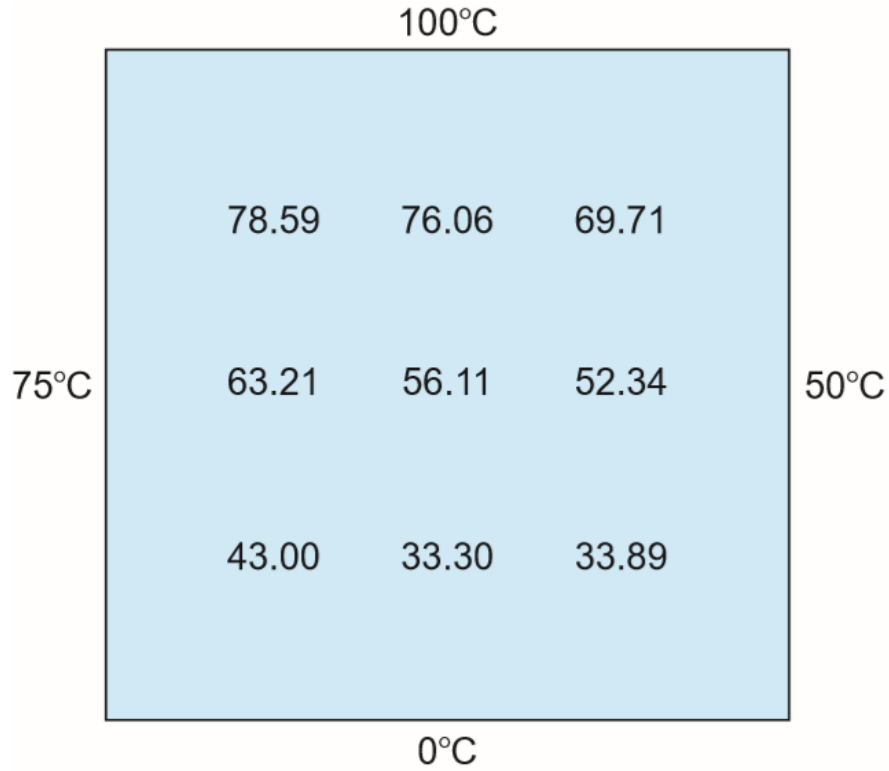
The error for $T_{1,1}$ can be estimated as [Eq. (29.13)]

$$|(\varepsilon_a)_{1,1}| = \left|\frac{32.51953 - 28.12500}{32.51953}\right| 100\% = 13.5\%$$

Because this value is above the stopping criterion of 1%, the computation is continued. The ninth iteration gives the result

$$T_{11} = 43.00061 \qquad T_{21} = 33.29755 \qquad T_{31} = 33.88506$$
$$T_{12} = 63.21152 \qquad T_{22} = 56.11238 \qquad T_{32} = 52.33999$$
$$T_{13} = 78.58718 \qquad T_{23} = 76.06402 \qquad T_{33} = 69.71050$$

where the maximum error is 0.71%.



Secondary Variables

For the heated plate, a secondary variable is the rate of heat flux across the plate's surface. This quantity can be computed from Fourier's law.

$$q_x = -k' \frac{T_{i+1,j} - T_{i-1,j}}{2\ \Delta x}$$

$$q_x = -k' \frac{T_{i,j+1} - T_{i,j-1}}{2\ \Delta y}$$

The resultant heat flux can be computed from these two quantities by

$$q_n = \sqrt{q_x^2 + q_y^2}$$

where the direction of q_n is given by

$$\theta = \tan^{-1}\left(\frac{q_y}{q_x}\right)$$

Flux Distribution for a Heated Plate

Assume that the plate is 40 × 40 cm and made out of aluminum [$k' = 0.49$ cal/(s · cm · °C)]

For $i = j = 1$

$$q_x = -0.49 \frac{\text{cal}}{\text{s} \cdot \text{cm} \cdot {}^\circ\text{C}} \frac{(33.29755 - 75)^\circ\text{C}}{2(10\ \text{cm})} = 1.022\ \text{cal/(cm}^2 \cdot \text{s)}$$

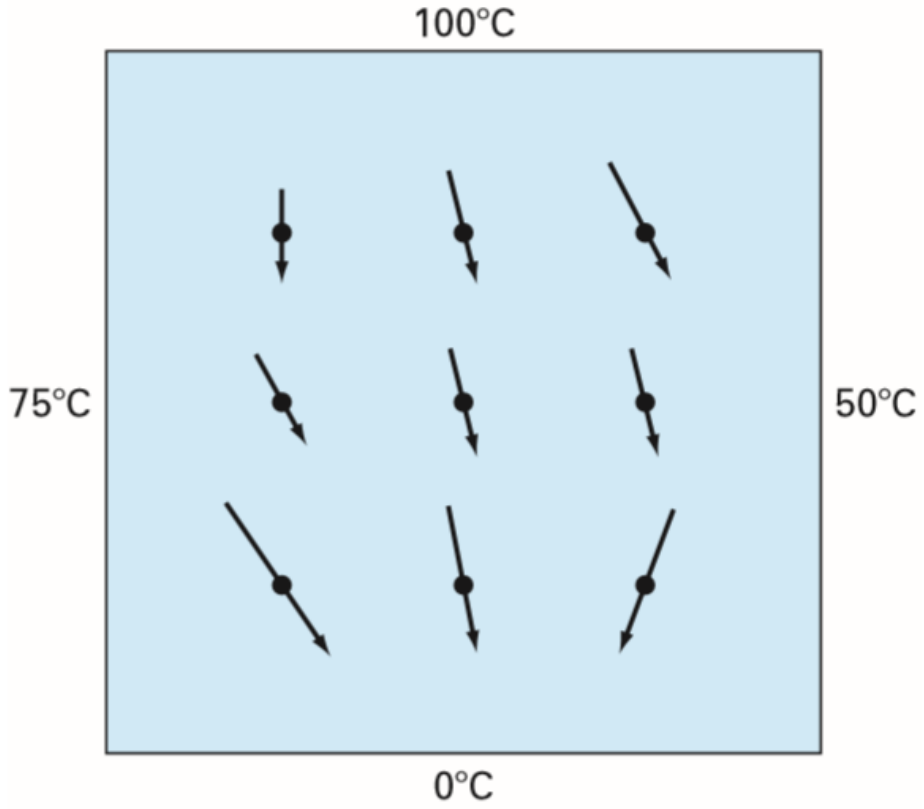
$$q_y = -0.49 \frac{\text{cal}}{\text{s} \cdot \text{cm} \cdot {}^\circ\text{C}} \frac{(63.21152 - 0)^\circ\text{C}}{2(10\ \text{cm})} = -1.549\ \text{cal/(cm}^2 \cdot \text{s)}$$

The resultant flux can be computed

$$q_n = \sqrt{(1.022)^2 + (-1.549)^2} = 1.856\ \text{cal/(cm}^2 \cdot \text{s)}$$

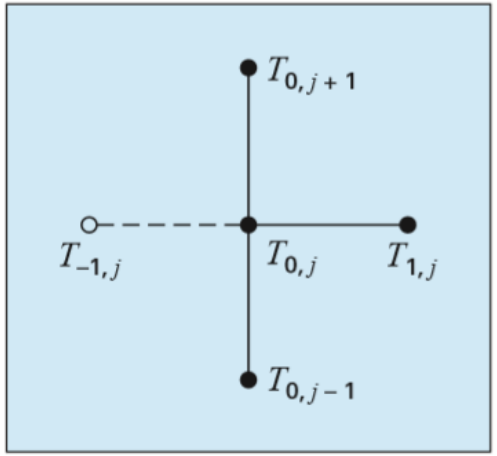
and the angle of its trajectory by Eq. (29.17)

$$\theta = \tan^{-1}\left(\frac{-1.549}{1.022}\right) = -0.98758 \times \frac{180^\circ}{\pi} = -56.584^\circ$$



Derivative Boundary Conditions

The fixed or Dirichlet boundary condition discussed to this point is but one of several types that are used with partial differential equations. A common alternative is the case where the derivative is given. This is commonly referred to as a *Neumann boundary condition.* For the heated-plate problem, this amounts to specifying the heat flux rather than the temperature at the boundary. One example is the situation where the edge is insulated. In this case, the derivative is zero. Another example would be where heat is lost across the edge by predictable mechanisms such as radiation or convection.



A boundary node (0,j) on the left edge of a heated plate. To approximate the derivative normal to the edge, an imaginary point (-1,j) is located a distance Δx beyond the edge.

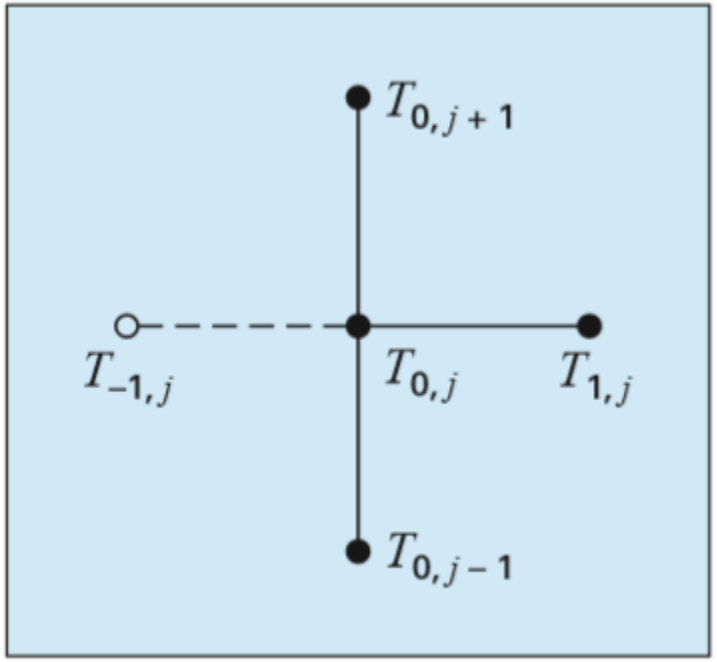
Derivative Boundary Conditions

$$T_{1,j} + T_{-1,j} + T_{0,j+1} + T_{0,j-1} - 4T_{0,j} = 0$$

$$\frac{\partial T}{\partial x} \cong \frac{T_{1,j} - T_{-1,j}}{2\,\Delta x}$$

$$T_{-1,j} = T_{1,j} - 2\,\Delta x \frac{\partial T}{\partial x}$$

$$2T_{1,j} - 2\,\Delta x \frac{\partial T}{\partial x} + T_{0,j+1} + T_{0,j-1} - 4T_{0,j} = 0$$



Heated plate with an insulated edge

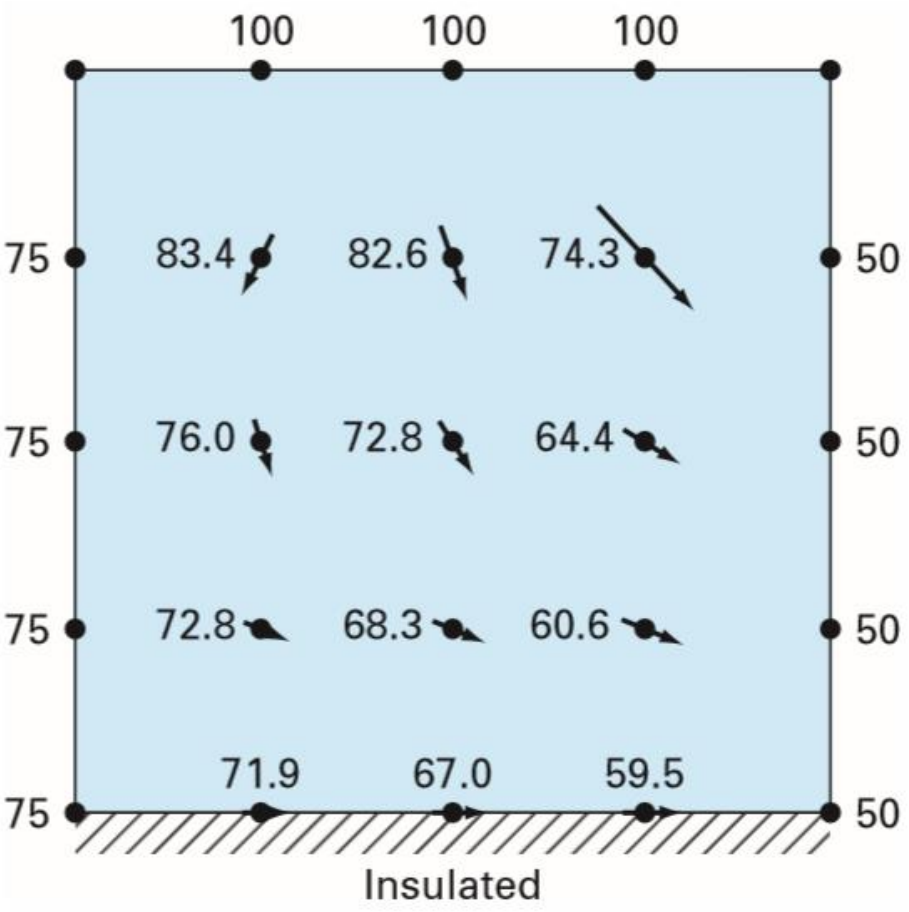
The general equation to characterize a derivative at the lower edge (that is, at $j = 0$) of a heated plate is

$$T_{i+1,0} + T_{i-1,0} + 2T_{i,1} - 2\,\Delta y \frac{\partial T}{\partial y} - 4T_{i,0} = 0$$

For an insulated edge, the derivative is zero and the equation becomes

$$T_{i+1,0} + T_{i-1,0} + 2T_{i,1} - 4T_{i,0} = 0$$

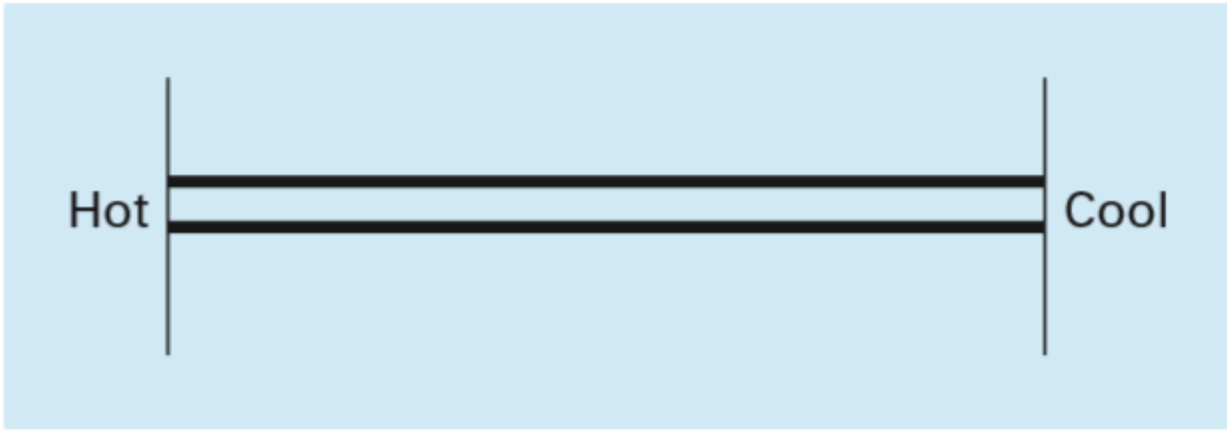
$$\begin{bmatrix}
4 & -1 & & -2 & & & & & & & & \\
-1 & 4 & -1 & & -2 & & & & & & & \\
 & -1 & 4 & & & -2 & & & & & & \\
-1 & & & 4 & -1 & & -1 & & & & & \\
 & -1 & & -1 & 4 & -1 & & -1 & & & & \\
 & & -1 & & -1 & 4 & & & -1 & & & \\
 & & & -1 & & & 4 & -1 & & -1 & & \\
 & & & & -1 & & -1 & 4 & -1 & & -1 & \\
 & & & & & -1 & & -1 & 4 & & & -1 \\
 & & & & & & -1 & & & 4 & -1 & \\
 & & & & & & & -1 & & -1 & 4 & -1 \\
 & & & & & & & & -1 & & -1 & 4
\end{bmatrix}
\begin{Bmatrix}
T_{10} \\ T_{20} \\ T_{30} \\ T_{11} \\ T_{21} \\ T_{31} \\ T_{12} \\ T_{22} \\ T_{32} \\ T_{13} \\ T_{23} \\ T_{33}
\end{Bmatrix}
=
\begin{Bmatrix}
75 \\ 0 \\ 50 \\ 75 \\ 0 \\ 50 \\ 75 \\ 0 \\ 50 \\ 175 \\ 100 \\ 150
\end{Bmatrix}$$



Finite Difference: Parabolic Equations

The heat-conduction equation

In a fashion similar to the derivation of the Laplace equation, conservation of heat can be used to develop a heat balance for the differential element in the long, thin insulated rod. However, rather than examine the steady-state case, the present balance also considers the amount of heat stored in the element over a unit time period Δt. Thus the balance is in the form, inputs-outputs=storage



$$q(x)\ \Delta y\ \Delta z\ \Delta t - q(x+\Delta x)\ \Delta y\ \Delta z\ \Delta t = \Delta x\ \Delta y\ \Delta z \rho C\ \Delta T$$

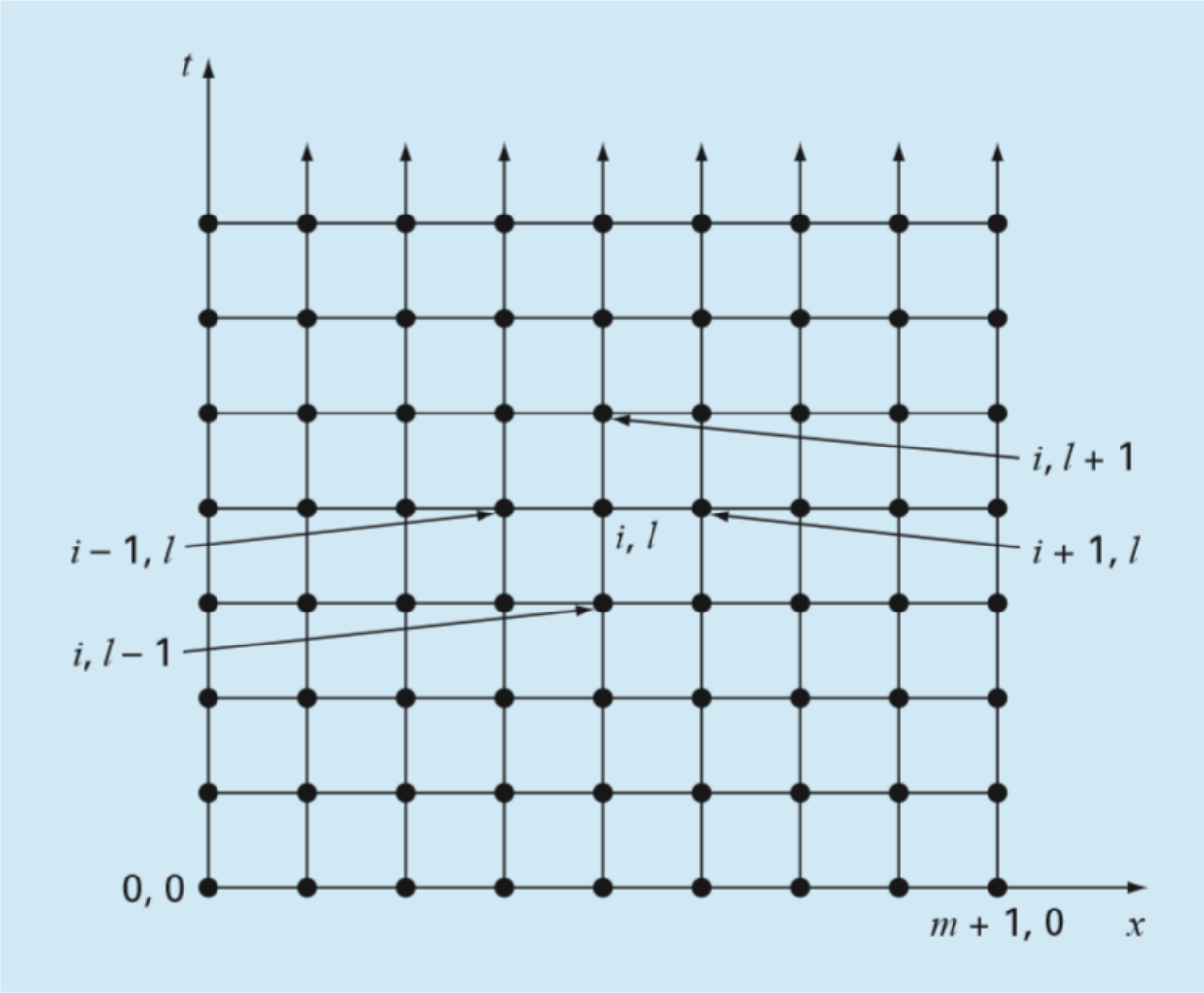
Dividing by the volume of the element ($= \Delta x\ \Delta y\ \Delta z$) and Δt gives

$$\frac{q(x) - q(x+\Delta x)}{\Delta x} = \rho C \frac{\Delta T}{\Delta t}$$

Taking the limit yields

$$-\frac{\partial q}{\partial x} = \rho C \frac{\partial T}{\partial t}$$

Grid system



Explicit methods

The heat-conduction equation requires approximations for the second derivative in space and the first derivative in time. The former is represented in the same fashion as for the Laplace equation by a centered finite-divided difference:

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{i+1}^l - 2T_i^l + T_{i-1}^l}{\Delta x^2}$$

A forward finite-divided difference is used to approximate the time derivative

$$\frac{\partial T}{\partial t} = \frac{T_i^{l+1} - T_i^l}{\Delta t}$$

$$k\frac{\partial^2 T}{\partial x^2} = \frac{\partial T}{\partial t}$$

$$k\frac{T_{i+1}^l - 2T_i^l + T_{i-1}^l}{(\Delta x)^2} = \frac{T_i^{l+1} - T_i^l}{\Delta t}$$

$$T_i^{l+1} = T_i^l + \lambda\left(T_{i+1}^l - 2T_i^l + T_{i-1}^l\right)$$

where $\lambda = k\,\Delta t/(\Delta x)^2$.

Explicit solution of the one-dimensional heat-conduction equation

Problem Statement. Use the explicit method to solve for the temperature distribution of a long, thin rod with a length of 10 cm and the following values: $k' = 0.49$ cal/(s · cm · °C), $\Delta x = 2$ cm, and $\Delta t = 0.1$ s. At $t = 0$, the temperature of the rod is zero and the boundary conditions are fixed for all times at $T(0) = 100°C$ and $T(10) = 50°C$. Note that the rod is aluminum with $C = 0.2174$ cal/(g·°C) and $\rho = 2.7$ g/cm^3. Therefore, $k = 0.49/(2.7 \cdot 0.2174) = 0.835$ cm^2/s and $\lambda = 0.835(0.1)/(2)^2 = 0.020875$.

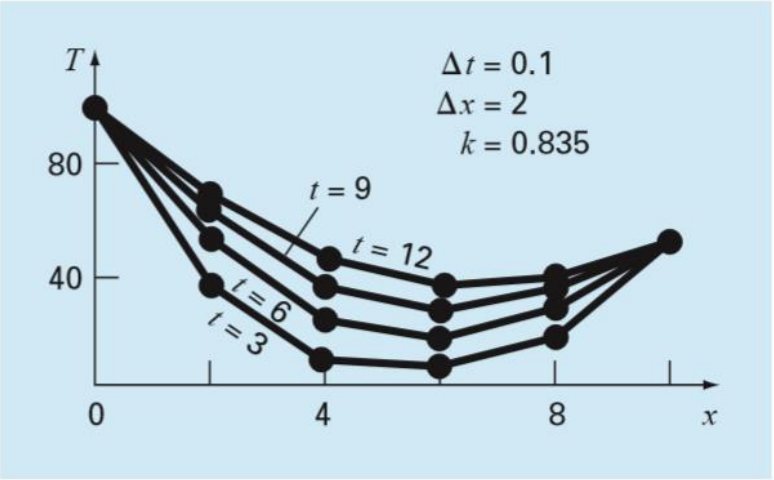
$$T_1^1 = 0 + 0.020875[0 - 2(0) + 100] = 2.0875$$

At the other interior points, $x = 4$, 6, and 8 cm, the results are

$$T_2^1 = 0 + 0.020875[0 - 2(0) + 0] = 0$$
$$T_3^1 = 0 + 0.020875[0 - 2(0) + 0] = 0$$
$$T_4^1 = 0 + 0.020875[50 - 2(0) + 0] = 1.0438$$

At $t = 0.2$ s, the values at the four interior nodes are computed as

$$T_1^2 = 2.0875 + 0.020875[0 - 2(2.0875) + 100] = 4.0878$$
$$T_2^2 = 0 + 0.020875[0 - 2(0) + 2.0875] = 0.043577$$
$$T_3^2 = 0 + 0.020875[1.0438 - 2(0) + 0] = 0.021788$$
$$T_4^2 = 1.0438 + 0.020875[50 - 2(1.0438) + 0] = 2.0439$$



The temperature distribution in a long, thin rod as computed with the explicit method. The general rise in temperature with time indicates that the computation captures the diffusion of heat from the boundaries into the bar.

Convergence and Stability

Convergence means that as Δx and Δt approach zero, the results of the finite-difference technique approach the true solution. *Stability* means that errors at any stage of the computation are not amplified but are attenuated as the computation progresses. It can be shown (Carnahan et al., 1969) that the explicit method is both convergent and stable if $\lambda \leq 1/2$, or

$$\Delta t \leq \frac{1}{2}\frac{\Delta x^2}{k}$$

In addition, it should be noted that setting $\lambda \leq 1/2$ could result in a solution in which errors do not grow, but oscillate. Setting $\lambda \leq 1/4$ ensures that the solution will not oscillate. It is also known that setting $\lambda = 1/6$ tends to minimize truncation error (Carnahan et al., 1969).

The crank-Nicolson method

The *Crank-Nicolson method* provides an alternative implicit scheme that is second-order accurate in both space and time. To provide this accuracy, difference approximations are developed at the midpoint of the time increment To do this, the temporal first derivative can be approximated at $t^{l+1/2}$ by

$$\frac{\partial T}{\partial t} \cong \frac{T_i^{l+1} - T_i^l}{\Delta t}$$

The second derivative in space can be determined at the midpoint by averaging the difference approximations at the beginning (t^l) and at the end (t^{l+1}) of the time increment

$$\frac{\partial^2 T}{\partial x^2} \cong \frac{1}{2}\left[\frac{T_{i+1}^l - 2T_i^l + T_{i-1}^l}{(\Delta x)^2} + \frac{T_{i+1}^{l+1} - 2T_i^{l+1} + T_{i-1}^{l+1}}{(\Delta x)^2}\right]$$

The crank-Nicolson method

$$k\frac{\partial^2 T}{\partial x^2} = \frac{\partial T}{\partial t}$$

$$-\lambda T_{i-1}^{l+1} + 2(1+\lambda)T_i^{l+1} - \lambda T_{i+1}^{l+1} = \lambda T_{i-1}^{l} + 2(1-\lambda)T_i^{l} + \lambda T_{i+1}^{l}$$

where $\lambda = k\,\Delta t/(\Delta x)^2$. As was the case with the simple implicit approach, boundary conditions of $T_0^{l+1} = f_0(t^{l+1})$ and $T_{m+1}^{l+1} = f_{m+1}(t^{l+1})$ can be prescribed to derive versions of

The above equation for the first and the last interior nodes. For the first interior node

$$2(1+\lambda)T_1^{l+1} - \lambda T_2^{l+1} = \lambda f_0(t^l) + 2(1-\lambda)T_1^l + \lambda T_2^l + \lambda f_0(t^{l+1})$$

For the last interior node

$$-\lambda T_{m-1}^{l+1} + 2(1+\lambda)T_m^{l+1} = \lambda f_{m+1}(t^l) + 2(1-\lambda)T_m^l + \lambda T_{m-1}^l + \lambda f_{m+1}(t^{l+1})$$

Crank-Nicolson solution to heat-conduction equation

The tridiagonal set of equations are generated

$$\begin{bmatrix} 2.04175 & -0.020875 & & \\ -0.020875 & 2.04175 & -0.020875 & \\ & -0.020875 & 2.04175 & -0.020875 \\ & & -0.020875 & 2.04175 \end{bmatrix} \begin{Bmatrix} T_1^1 \\ T_2^1 \\ T_3^1 \\ T_4^1 \end{Bmatrix} = \begin{Bmatrix} 4.175 \\ 0 \\ 0 \\ 2.0875 \end{Bmatrix}$$

which can be solved for the temperatures at $t = 0.1$ s:

$$T_1^1 = 2.0450$$
$$T_2^1 = 0.0210$$
$$T_3^1 = 0.0107$$
$$T_4^1 = 1.0225$$

To solve for the temperatures at $t = 0.2$ s, the right-hand-side vector must be changed to

$$\begin{Bmatrix} 8.1801 \\ 0.0841 \\ 0.0427 \\ 4.0901 \end{Bmatrix}$$

The simultaneous equations can then be solved for

$$T_1^2 = 4.0073$$
$$T_2^2 = 0.0826$$
$$T_3^2 = 0.0422$$
$$T_4^2 = 2.0036$$

Parabolic equations in two spatial dimensions

$$\frac{\partial T}{\partial t} = k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right)$$

Standard explicit and implicit schemes

Criteria for stability

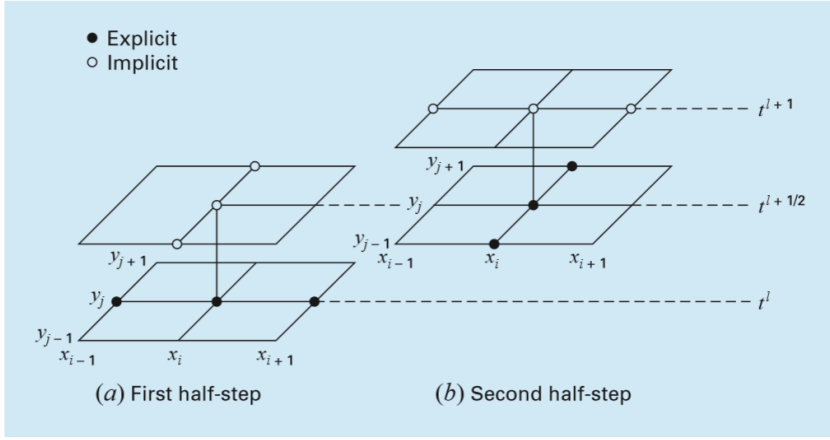
$$\Delta t \leq \frac{1}{8}\frac{(\Delta x)^2 + (\Delta y)^2}{k}$$

The ADI Scheme

The alternating-direction implicit, or ADI, scheme provides a means for solving parabolic equations in two spatial dimensions using tridiagonal matrices. To do this, each time

Increment is executed in two steps. For the first step, the governing equation is approximated by

$$\frac{T_{i,j}^{l+1/2} - T_{i,j}^{l}}{\Delta t/2} = k\left[\frac{T_{i+1,j}^{l} - 2T_{i,j}^{l} + T_{i-1,j}^{l}}{(\Delta x)^2} + \frac{T_{i,j+1}^{l+1/2} - 2T_{i,j}^{l+1/2} + T_{i,j-1}^{l+1/2}}{(\Delta y)^2}\right]$$



Thus, the approximation of $\partial^2 T/\partial x^2$ is written explicitly—that is, at the base point t^l where values of temperature are known. Consequently, only the three temperature terms in the approximation of $\partial^2 T/\partial y^2$ are unknown. For the case of a square grid ($\Delta y = \Delta x$), this equation can be expressed as

$$-\lambda T_{i,j-1}^{l+1/2} + 2(1+\lambda)T_{i,j}^{l+1/2} - \lambda T_{i,j+1}^{l+1/2} = \lambda T_{i-1,j}^{l} + 2(1-\lambda)T_{i,j}^{l} + \lambda T_{i+1,j}^{l}$$

which, when written for the system, results in a tridiagonal set of simultaneous equations.

For the second step from $t^{l+1/2}$ to t^{l+1}, Eq. (30.18) is approximated by

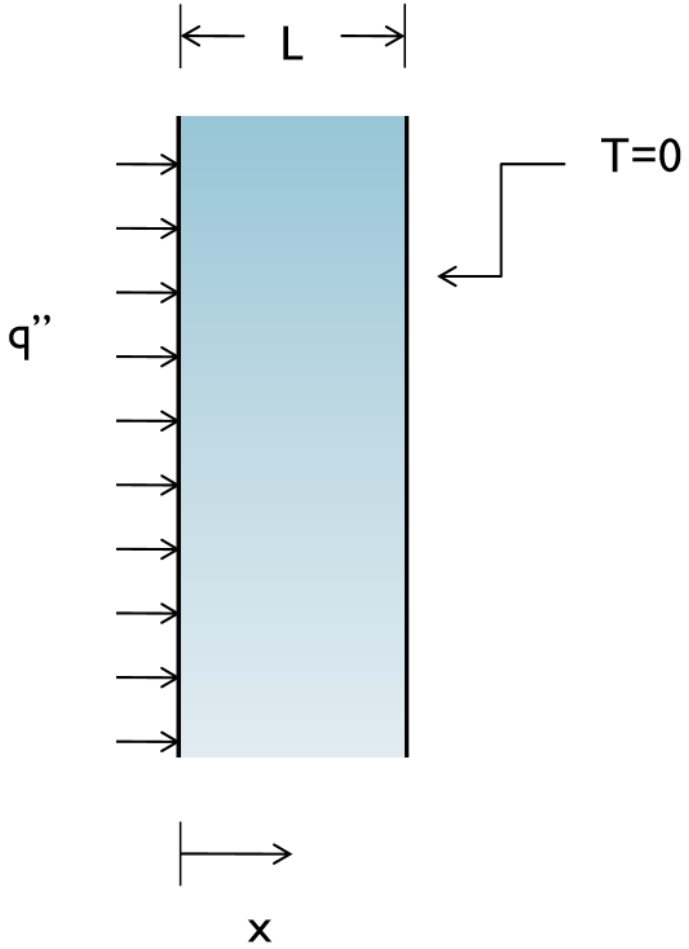
$$\frac{T_{i,j}^{l+1} - T_{i,j}^{l+1/2}}{\Delta t/2} = k\left[\frac{T_{i+1,j}^{l+1} - 2T_{i,j}^{l+1} + T_{i-1,j}^{l+1}}{(\Delta x)^2} + \frac{T_{i,j+1}^{l+1/2} - 2T_{i,j}^{l+1/2} + T_{i,j-1}^{l+1/2}}{(\Delta y)^2}\right]$$

$$-\lambda T_{i-1,j}^{l+1} + 2(1+\lambda)T_{i,j}^{l+1} - \lambda T_{i+1,j}^{l+1} = \lambda T_{i,j-1}^{l+1/2} + 2(1-\lambda)T_{i,j}^{l+1/2} + \lambda T_{i,j+1}^{l+1/2}$$

PDE solver in MATLAB

- Parabolic partial differential equations are encountered in many scientific applications
- Think of these as a time-dependent problem in one spatial dimension
- Matlab's **pdepe** command can solve these

Model Problem

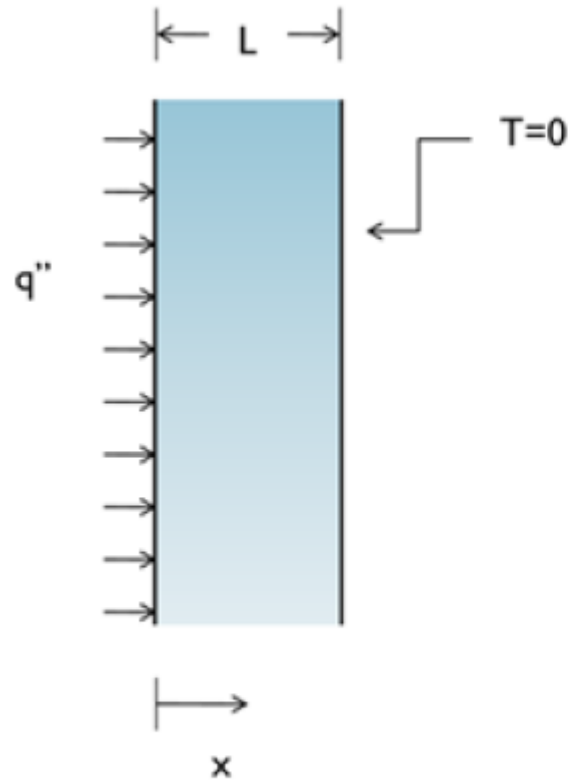


$$\rho c_p \frac{\partial T}{\partial x} = k \frac{\partial^2 T}{\partial x^2}$$

$$T(x,0) = 0$$

$$-k \left. \frac{\partial T}{\partial x} \right|_{x=0} = q''$$

$$T(L,t) = 0$$



$$\rho c_p \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2}$$

$$T(x,0) = 0$$

$$-k \left. \frac{\partial T}{\partial x} \right|_{x=0} = q''$$

$$-k \frac{dT}{dx}_{x=L} = h\left(T - T_{bulk}\right)$$

Pdepe solves the following

$$c\left(x,t,u,\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial t}=x^{-m}\frac{\partial}{\partial x}\left(x^{m}f\left(x,t,u,\frac{\partial u}{\partial x}\right)\right)+s\left(x,t,u,\frac{\partial u}{\partial x}\right)$$

m=0 for Cartesian, 1 for cylindrical, 2 for spherical

$$u(x,t_0)=u_0(x)$$

Initial Conditions

$$p(x,t,u)+q(x,t)f\left(x,t,u,\frac{\partial u}{\partial x}\right)=0$$

Boundary Conditions – one at each boundary

Pdepe solves the following

$$c\left(x,t,u,\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial t}=x^{-m}\frac{\partial}{\partial x}\left(x^{m}f\left(x,t,u,\frac{\partial u}{\partial x}\right)\right)+s\left(x,t,u,\frac{\partial u}{\partial x}\right)$$

$$\rho c_p\frac{\partial T}{\partial t}=k\frac{\partial^2 T}{\partial x^2}$$

$$c=\rho c_p$$

$$f=k\frac{\partial T}{\partial x}$$

$$s=0$$

Differential Equations

```
function [c,f,s] = pdex1pde(x,t,u,DuDx)
global rho cp k
c = rho*cp;
f = k*DuDx;
s = 0;
```

Initial Conditions

```
function u0 = pdex1ic(x)
u0 = 0;
```

Pdepe solves the following

$$-k\left.\frac{\partial T}{\partial x}\right|_{x=0} = q''$$

$$p(x,t,u)+q(x,t)f\left(x,t,u,\frac{\partial u}{\partial x}\right)=0$$

or

$$q''+k\left.\frac{\partial T}{\partial x}\right|_{x=0} = 0$$

$$x = 0$$
$$p = q''$$
$$q = 1$$

remember

$$f = k\frac{\partial T}{\partial x}$$

$$T(L,t) = 0$$

$$x = L$$
$$p = T = ur$$
$$q = 0$$

Boundary conditions

```
function [pl,ql,pr,qr] = pdex1bc(xl,ul,xr,ur,t)
global q
pl = q;
ql = 1;
pr = ur;
qr = 0;
```

At left edge, q+k*dT/dx=0

At right edge, ur=0

Calling the solver

```
tend=10
m = 0;
x = linspace(0,L,200);
t = linspace(0,tend,50);
sol =
  pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x,t);
```

200 spatial mesh points

50 time steps from t=0 to tend

Postprocessing

```
Temperature = sol(:,:, 1);
figure, plot(x, Temperature(end,:))

figure, plot(t, Temperature(:, 1))
```

Full Code

```
function parabolic
global rho cp k
global q
L=0.1      %m
k=200      %W/m-K
rho=10000 %kg/m^3
cp=500     %J/kg-K
q=1e6    %W/m^2
tend=10 %seconds

m = 0;
x = linspace(0,L,200);
t = linspace(0,tend,50);

sol = pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x,t);
Temperature = sol(:,:,1);
 figure, plot(x,Temperature(end,:))
```

```
function [c,f,s] = pdex1pde(x,t,u,DuDx)
global rho cp k
c = rho*cp;
f = k*DuDx;
s = 0;

function u0 = pdex1ic(x)
u0 = 0;

function [pl,ql,pr,qr] =
    pdex1bc(xl,ul,xr,ur,t)
global q
pl = q;
ql = 1;
pr = ur;
qr = 0;
```

Solve System of PDEs

$$\frac{\partial u_1}{\partial t} = 0.024 \frac{\partial^2 u_1}{\partial x^2} - F(u_1 - u_2),$$

$$\frac{\partial u_2}{\partial t} = 0.170 \frac{\partial^2 u_2}{\partial x^2} + F(u_1 - u_2).$$

The equation holds on the interval $0 \leq x \leq 1$ for times $t \geq 0$. The initial conditions are

$$u_1(x, 0) = 1,$$

$$u_2(x, 0) = 0.$$

The boundary conditions are

$$\frac{\partial}{\partial x} u_1(0, t) = 0,$$

$$u_2(0, t) = 0,$$

$$\frac{\partial}{\partial x} u_2(1, t) = 0,$$

$$u_1(1, t) = 1.$$

Solve System of PDEs

Code Equation

Before you can code the equation, you need to make sure that it is in the form that the pdepe solver expects:

$$c\left(x,t,u,\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial t} = x^{-m}\frac{\partial}{\partial x}\left(x^m f\left(x,t,u,\frac{\partial u}{\partial x}\right)\right) + s\left(x,t,u,\frac{\partial u}{\partial x}\right).$$

In this form, the PDE coefficients are matrix-valued and the equation becomes

$$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\frac{\partial}{\partial t}\begin{bmatrix}u_1\\ u_2\end{bmatrix} = \frac{\partial}{\partial x}\begin{bmatrix}0.024\dfrac{\partial u_1}{\partial x}\\ 0.170\dfrac{\partial u_2}{\partial x}\end{bmatrix} + \begin{bmatrix}-F(u_1-u_2)\\ F(u_1-u_2)\end{bmatrix}.$$

So the values of the coefficients in the equation are

$$m = 0$$

$$c\left(x,t,u,\frac{\partial u}{\partial x}\right) = \begin{bmatrix}1\\ 1\end{bmatrix} \text{ (diagonal values only)}$$

$$f\left(x,t,u,\frac{\partial u}{\partial x}\right) = \begin{bmatrix}0.024\dfrac{\partial u_1}{\partial x}\\ 0.170\dfrac{\partial u_2}{\partial x}\end{bmatrix}$$

$$s\left(x,t,u,\frac{\partial u}{\partial x}\right) = \begin{bmatrix}-F(u_1-u_2)\\ F(u_1-u_2)\end{bmatrix}$$

Code Boundary Conditions

For $x = 0$, the equation is

$$\begin{bmatrix}0\\ u_2\end{bmatrix} + \begin{bmatrix}1\\ 0\end{bmatrix}\cdot\begin{bmatrix}0.024\dfrac{\partial u_1}{\partial x}\\ 0.170\dfrac{\partial u_2}{\partial x}\end{bmatrix} = 0.$$

for $x = 1$ the equation is

$$\begin{bmatrix}u_1-1\\ 0\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}\cdot\begin{bmatrix}0.024\dfrac{\partial u_1}{\partial x}\\ 0.170\dfrac{\partial u_2}{\partial x}\end{bmatrix} = 0.$$

```
function [c,f,s] = pdefun(x,t,u,dudx) % Equation to solve
c = [1; 1];
f = [0.024; 0.17] .* dudx;
y = u(1) - u(2);
F = exp(5.73*y)-exp(-11.47*y);
s = [-F; F];
end
% --------------------------------------------
function u0 = pdeic(x) % Initial Conditions
u0 = [1; 0];
end
% --------------------------------------------
function [pl,ql,pr,qr] = pdebc(xl,ul,xr,ur,t) % Boundary Conditions
pl = [0; ul(2)];
ql = [1; 0];
pr = [ur(1)-1; 0];
qr = [0; 1];
end
% --------------------------------------------
```

Select Solution Mesh

The solution to this problem changes rapidly when t is small. Although `pdepe` selects a time step that is appropriate to resolve the sharp changes, to see the behavior in the output plots you need to select appropriate output times. For the spatial mesh, there are boundary layers in the solution at both ends of $0 \le x \le 1$, so you need to specify mesh points there to resolve the sharp changes.

```
x = [0 0.005 0.01 0.05 0.1 0.2 0.5 0.7 0.9 0.95 0.99 0.995 1];
t = [0 0.005 0.01 0.05 0.1 0.5 1 1.5 2];
```

Solve Equation

Finally, solve the equation using the symmetry m, the PDE equation, the initial conditions, the boundary conditions, and the meshes for x and t.

```
m = 0;
sol = pdepe(m,@pdefun,@pdeic,@pdebc,x,t);
```

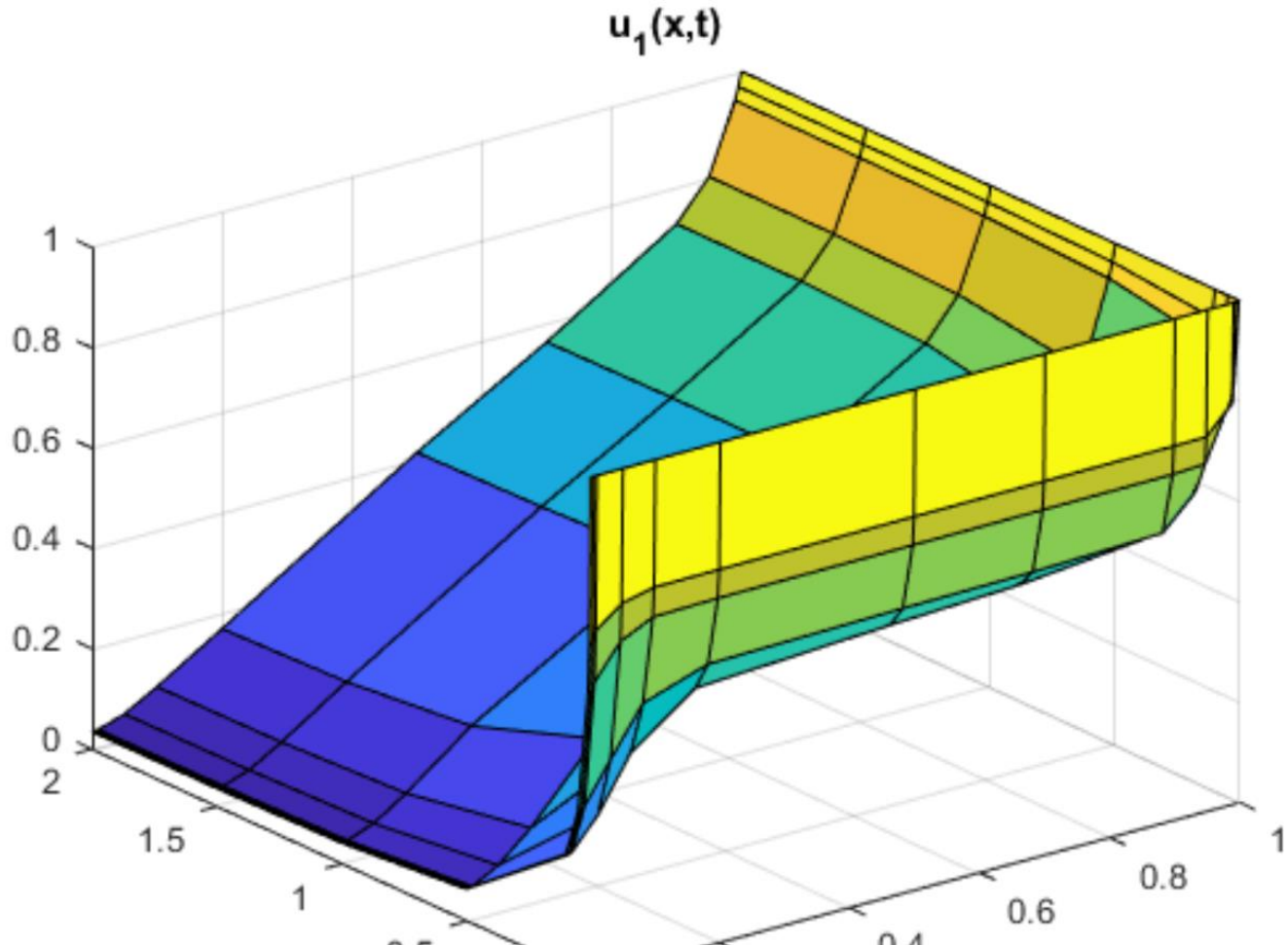
`pdepe` returns the solution in a 3-D array `sol`, where `sol(i,j,k)` approximates the kth component of the solution u_k evaluated at `t(i)` and `x(j)`. Extract each solution component into a separate variable.

```
u1 = sol(:,:,1);
u2 = sol(:,:,2);
```

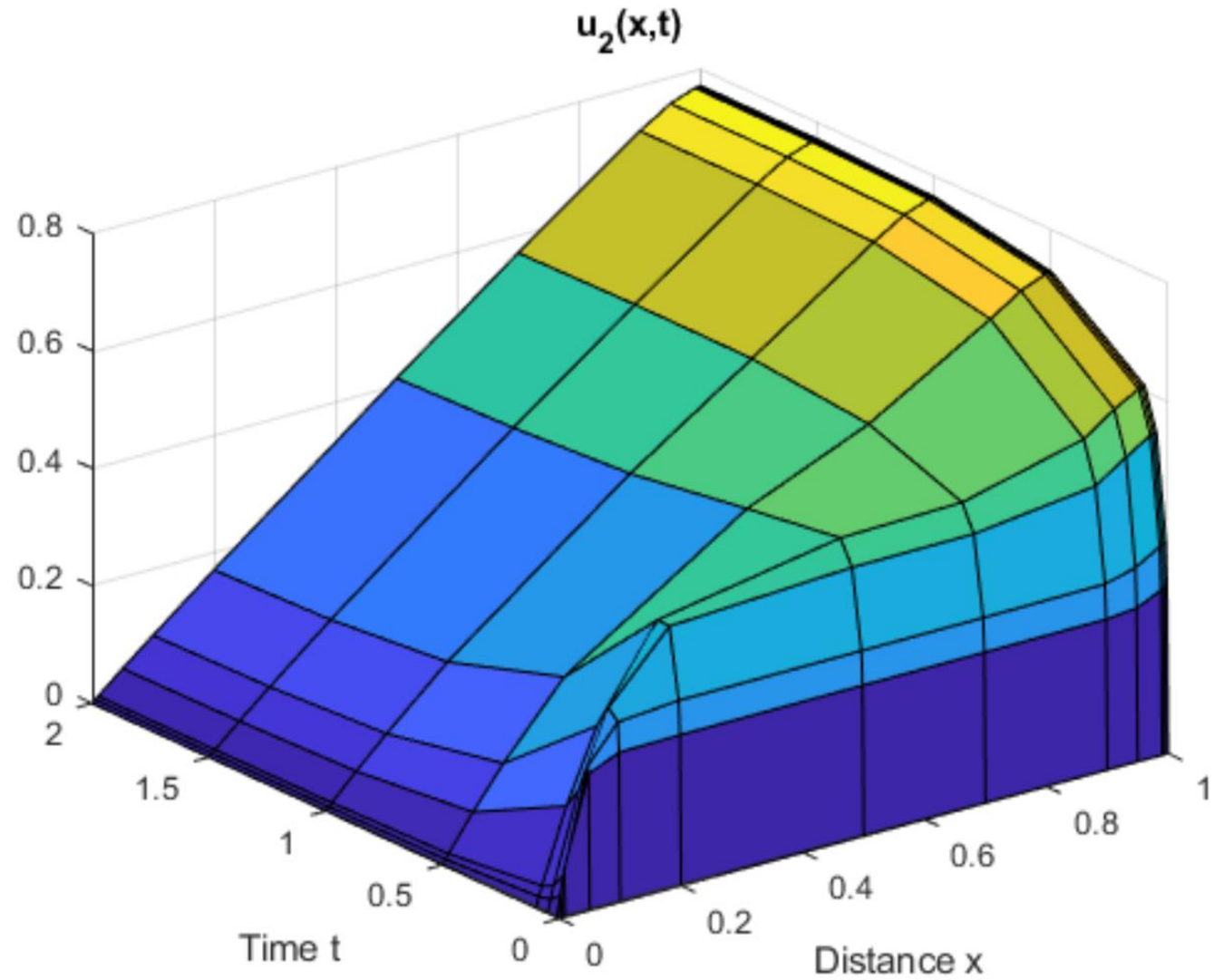
Plot Solution

Create surface plots of the solutions for u_1 and u_2 plotted at the selected mesh points for x and t.

```
surf(x,t,u1)
title('u_1(x,t)')
xlabel('Distance x')
ylabel('Time t')
```



```
surf(x,t,u2)
title('u_2(x,t)')
xlabel('Distance x')
ylabel('Time t')
```



30.7 The advection-diffusion equation is used to compute the distribution of concentration along the length of a rectangular chemical reactor (see Sec. 32.1),

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2} - U\frac{\partial c}{\partial x} - kc$$

where c = concentration (mg/m^3), t = time (min), D = a diffusion coefficient (m^2/min), x = distance along the tank's longitudinal axis (m) where $x = 0$ at the tank's inlet, U = velocity in the x direction (m/min), and k = a reaction rate (min^{-1}) whereby the chemical decays to another form. Develop an explicit scheme to solve this equation numerically. Test it for $k = 0.15$, $D = 100$, and $U = 1$ for a tank of length 10 m. Use a $\Delta x = 1$ m, and a step size $\Delta t = 0.005$. Assume that the inflow concentration is 100 and that the initial concentration in the tank is zero. Perform the simulation from $t = 0$ to 100 and plot the final resulting concentrations versus x.

Simulation Process