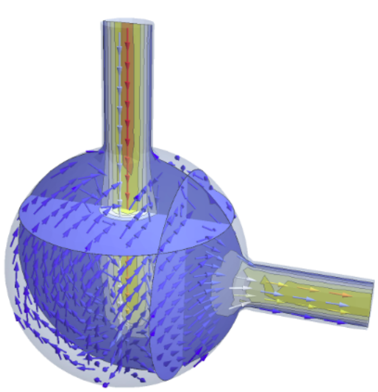
WOLFRAM U

Scientific Computing with Mathematica

PART (3): PARTIAL DIFFERENTIAL EQUATIONS



Outline

Symbolic Solution to PDEs

Numerical Solution to PDEs

PDE Domain Models

Symbolic Solution to PDEs

First-order PDEs

PDEs in physics and engineering

Differential eigensystems

First-Order PDEs

The transport equation

The transport equation is a linear first-order PDE:

In[ ]:=
```
TransportEquation = ∂tu[x,t]+ ∂xu[x,t] == 0;
```

Find the general solution for this equation:

In[ ]:=
```
DSolveValue[TransportEquation ,u[x,t],{x,t}];
```

Solve an initial value problem:

In[ ]:=
```
InitialValue = u[x,0] == E^(-x^2);
sol = DSolveValue[{TransportEquation ,InitialValue },u[x,t],{x,t}];
```

Plot the traveling wave:

In[ ]:=
```
Manipulate[Plot[sol /. t→tf,{x,-10,10},PlotRange→All ],{tf,0,10,.1}];
```

First-Order PDEs

The inhomogeneous transport equation

Use **DSolve** with an inhomogeneous PDE with a given initial condition:

In[ ]:=
```
sol1 = DSolveValue [{D[y[x, t], t] + 2 D[y[x, t], x] == Sin[x], y[0, t] == Cos[t]}, y[x, t], {x, t}];
```

Evaluate the solution for given values of the parameters:

In[ ]:=
```
sol1 /. {x→5,t→1};
```

Now use **Plot3D** to plot the solution:

In[ ]:=
```
Plot3D [sol1 , {x, -10 , 10}, {t, -5, 5}];
```

The coefficients of the PDE can also be symbolic:

```
sol1 = DSolveValue [{a D[y[x, t], t] + b D[y[x, t], x] == Sin[x], y[0, t] == Cos[t]}, y[x, t], {x, t}
```

Out[ ]=

$$\frac{1 - \mathrm{Cos}[x] + b\,\mathrm{Cos}\left[\frac{b\,t - a\,x}{b}\right]}{b}$$

PDEs in Physics and Engineering

Wave equation

Heat equation

Laplace’s equation

Wave equation: The one-dimensional string

The wave equation is a linear second-order hyperbolic PDE:

In[ ]:=
```
waveEq = D[u[x,t],{x,2}] == D[u[x,t],{t,2}];
```

Use it to model the vibrations of a plucked string as follows.

Specify that the ends of the string remain fixed during the vibrations:

In[ ]:=
```
bc = {u[0,t] == 0, u[1,t] == 0};
```

Give the initial position of the string:

In[ ]:=
```
ic = {u[x,0] == Piecewise[{{3x, 0 ≤ x ≤ 1/3},{3×(1-x)/2, 1/3 ≤ x ≤ 1}}], (D[u[x,t],{t,1}] /. t→0) == 0};
```

Solve the initial value problem:

In[ ]:=
```
solWaveEq = DSolveValue[{waveEq,ic,bc},u[x,t],{x,t}]
```

Out[ ]=
```
$Aborted
```

Extract the first 10 terms of the sum:

In[ ]:=
```
solseries = Activate[solWaveEq /. {K[1] →m,∞ -> 10}];
```

Animate the string vibrations:

In[ ]:=
```
Manipulate[Plot[solseries /. t→tf,{x,0,1},PlotRange →All],{tf,0,10,.01}];
```

Wave equation: The two-dimensional membrane

Model the oscillations of a circular membrane of radius 1 using the wave equation in 2D:

In[ ]:=
```
eqn=r D[u[r,t],{t,2}]==D[r D[u[r,t],r],r];
```

Specify that the boundary of the membrane remain fixed:

In[ ]:=
```
bc=u[1,t]==0;
```

Initial condition for the problem:

In[ ]:=
```
ic={u[r,0]==0,Derivative[0,1][u][r,0]==1};
```

Solve the initial value problem:

In[ ]:=
```
(dsol=DSolveValue[{eqn,bc,ic},u[r,t],{r,t}])
```

Out[ ]=

$$\sum_{K[1]=1}^{\infty} (2\,\text{BesselJ}[0,\ r\,\text{BesselJZero}[0,\ K[1]]]\ \text{BesselJ}[1,\ \text{BesselJZero}[0,\ K[1]]]\ \text{Sin}[t\,\text{BesselJZero}[0,\ K[1]]]) / ((\text{BesselJ}[0,\ \text{BesselJZero}[0,\ K[1]]]^2 + \text{BesselJ}[1,\ \text{BesselJZero}[0,\ K[1]]]^2)\ \text{BesselJZero}[0,\ K[1]]^2)$$

Extract the first 3 terms of the sum:

In[ ]:=
```
h[r_, t_]=dsol/.{Infinity→3}//Activate//N;
```

Animate the membrane vibrations:

In[ ]:=
```
Animate[Plot3D[h[r,t]/.{r→Sqrt[x^2+y^2]},{x,y}∈Disk[],PlotRange→{-1,1},Ticks→None,Mesh→True,Mesh
```

Heat equation: Heat flow in a bar

The heat equation is a linear second-order parabolic PDE:

In[ ]:=
```
heatEq = D[u[x,t],{x,2}] == D[u[x,t],{t,1}];
```

Use it to model the heat flow in a bar that is insulated at both ends.

Specify that there be no heat flux at either end:

In[ ]:=
```
bc = {(D[u[x,t],{x,1}] /. x→0) == 0, (D[u[x,t],{x,1}] /. x→1) == 0}
```

Out[ ]= $\{u^{(1,0)}[0, t] == 0, u^{(1,0)}[1, t] == 0\}$

Specify the initial temperature distribution:

In[ ]:=
```
ic = u[x,0] == 10x ;
```

Solve the PDE:

In[ ]:=
```
solHeatEq = DSolveValue [{heatEq ,ic ,bc},u[x,t],{x,t}]
```

Out[ ]= $$5 + 2 \sum_{K[1]=1}^{\infty} \frac{10 \times \left(-1 + (-1)^{K[1]}\right) e^{-\pi^2 t K[1]^2} \mathrm{Cos}[\pi x K[1]]}{\pi^2 K[1]^2}$$

Extract the first 20 terms of the sum:

In[ ]:=
```
solseries = Activate [solHeatEq /. {K[1] →m,∞ -> 20}];
```

Animate the heat diffusion:

In[ ]:=
```
Manipulate [Plot[solseries /. t→tf,{x,0,1},PlotRange →{0,10} ],{tf,0,.1,.001}];
```

Laplace's equation

The Laplace equation is a linear second-order elliptic PDE:

In[ ]:=
```
laplaceEq = D[u[x,y],{x,2}] + D[u[x,y],{y,2}] == 0
```

Out[ ]= $u^{(0,2)}[x, y] + u^{(2,0)}[x, y] == 0$

Solve the Dirichlet problem in a square:

In[ ]:=
```
bc = {u[x,0] == 1, u[x,1] == 1, u[0,y] ==0 , u[1,y] == 0};
```

Solve the PDE:

In[ ]:=
```
solHeatEq = DSolveValue [{laplaceEq ,bc},u[x,y],{x,y}]
```

Out[ ]=
$$\sum_{K[1]=1}^{\infty}\left(-\frac{2\times\left(-1+(-1)^{K[1]}\right)\text{Csch}[\pi\,K[1]]\,\text{Sin}[\pi\,x\,K[1]]\,\text{Sinh}[\pi\,(1-y)\,K[1]]}{\pi\,K[1]}-\frac{2\times\left(-1+(-1)^{K[1]}\right)\text{Csch}[\pi\,K[1]]\,\text{Sin}[\pi\,x\,K[1]]\,\text{Sinh}[\pi\,y\,K[1]]}{\pi\,K[1]}\right)$$

Extract the first 50 terms of the sum:

In[ ]:=
```
solseries = Activate [solHeatEq /. {K[1] →m,∞ -> 50}];
```

Visualize the solution on the square:

In[ ]:=
```
ContourPlot [solseries ,{x,0,1},{y,0,1} ];
```

Differential Eigensystems

The vibrating circular membrane

Compute the first 6 eigenfunctions for a circular membrane with the edges clamped:

```
In[ ]:= {vals,funs}=DEigensystem[{-Laplacian[u[x,y],{x,y}],DirichletCondition[u[x,y]==0,True]},u[x,y],{x,y}
vals //N
```

```
Out[ ]= {5.78319, 14.682, 14.682, 26.3746, 26.3746, 30.4713}
```

Visualize the eigenfunctions:

```
In[ ]:= Table[Plot3D[funs[[i]]//N//Evaluate,{x,y}∈Disk[],PlotRange→All,PlotTheme→"Minimal"],{i,Length[vals
```

Differential Eigensystems

Compute eigenfunctions in an L-shaped region

Specify an L-shaped region:

In[ ]:=
```
L = Polygon[{{1, 0}, {2, 0}, {2, 2}, {0, 2}, {0, 1}, {1, 1}}];
RegionPlot[L];
```

Specify a Laplacian operator:

In[ ]:=
```
ℒ = Laplacian[u[x, y], {x, y}];
```

Specify a Dirichlet boundary condition:

In[ ]:=
```
ℬ = DirichletCondition[u[x, y] == 0., True];
```

Compute the eigenfunctions in the L-shaped region:

In[ ]:=
```
{vals, funs} = NDEigensystem[{ℒ, ℬ}, u[x, y], {x, y} ∈ L, 2];
```

Visualize the eigenfunctions:

In[ ]:=
```
Table[Plot3D[funs[[i]]//N//Evaluate,{x,y}∈L,PlotRange→All,PlotTheme→"Scientific"],{i,Length[vals]}]
```

Numerical Solution to PDEs

Numerical Solution to PDEs

Here are the main steps to solve a PDE problem:

1. Specify the domain
2. Define the PDE
3. Define boundary/initial conditions
4. Solve
5. Visualize

Solving PDEs Numerically

NDSolve

Specify domain:

In[ ]:=
```
Ω = Disk[];
RegionPlot[Ω];
```

Define a PDE:

In[ ]:=
```
op = -Laplacian[u[x,y],{x,y}]-1;
```

Define boundary/initial conditions:

In[ ]:=
```
BC = DirichletCondition[u[x,y]==1,x≥ 0];
```

Solve:

In[ ]:=
```
usol = NDSolveValue[{op==0, BC},u[x,y],{x,y}∈ Ω];
```

Visualize:

In[ ]:=
```
Plot3D[usol,{x,y}∈ Ω,ColorFunction→"Rainbow",PlotLegends→Automatic];
```

PDEs That Can Be Solved Numerically in the Wolfram Language™

What types of PDEs can be solved with the FEM as implemented in NDSolve?

Consider a single partial differential equation in u:

$$m \frac{\partial^2}{\partial t^2} u + \frac{\partial}{\partial t} u + \nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + a u - f = 0$$

The Laplace equation simply contains a diffusive term:

$$\nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + a u - f = 0.$$

To model Poisson's equation, add a load term :

$$\nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + a u - f = 0.$$

Helmholtz's equation adds a reaction term $a u$:

$$\nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + a u - f = 0.$$

The heat equation adds time dependence to the Poisson equation:

$$m \frac{\partial^2}{\partial t^2} u + \frac{\partial}{\partial t} u + \nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + a u - f = 0.$$

Similarly, the wave equation is given as:

$$m \frac{\partial^2}{\partial t^2} u + \frac{\partial}{\partial t} u + \nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + a u - f = 0.$$

Boundary Conditions

DirichletCondition—specify Dirichlet conditions for partial differential equations

NeumannValue—specify Neumann and Robin conditions

PeriodicBoundaryCondition—specify periodic boundary conditions

Poisson Equation

Neumann condition

Specify domain:

In[ ]:=
```
Ω = ImplicitRegion [0<x<5 && 0<y<10&& !((x-5)^2 + (y-5)^2 ≤ 3^2),{x,y}];
RegionPlot [Ω];
```

Define a PDE:

In[ ]:=
```
op = -Laplacian [u[x,y],{x,y}]-1;
```

Define boundary/initial conditions:

In[ ]:=
```
neumann = NeumannValue [3u[x,y]+1,(x-5)^2 + (y-5)^2 == 3^2];
dBC = {DirichletCondition [u[x,y] ==0, x == 0 && 8 ≤ y ≤ 10]};
```

Solve:

In[ ]:=
```
usol = NDSolveValue [{op ==neumann ,dBC},u[x,y],{x,y}∈ Ω];
```

Visualize:

In[ ]:=
```
Plot3D [usol ,{x,y}∈ Ω,ColorFunction →"Rainbow ",PlotLegends →Automatic ];
```

PDE Domain Models

Partial Differential Equation Terms (Mathematica® 12.2)

For specific physics fields, relevant PDE terms have been packaged as components:

Use partial differential equations of the form $\nabla\cdot(-c\,\nabla u-\alpha\,u+\gamma)+\beta\cdot\nabla u+a\,u=f$

DiffusionPDETerm—model diffusion with $\nabla\cdot(-c\,\nabla u)$

ConvectionPDETerm—model convection with $\beta\cdot\nabla u$

ReactionPDETerm—model reaction with $a\,u$

SourcePDETerm—model a source with f

ConservativeConvectionPDETerm—model conservative convection with $\nabla\cdot(-\alpha\,u)$

DerivativePDETerm—model a derivative of a term with $\nabla\cdot(\gamma)$

Animate convection and diffusion:

In[ ]:=
```
Manipulate[(ufun=NDSolveValue[{D[u[t,x],t]+
ConvectionPDETerm[{u[t,x],{x}},{β}]+
DiffusionPDETerm[{u[t,x],{x}},c]==0,u[0,x]==Exp[-x^2]},u,{t,0,1},{x,-2π,2π}];
Plot[ufun[tt,x],{x,-2π,2π},PlotRange→{0,1}]),{tt,0,1,.01},{{β,0},-5,5},{{c,1},1,5}];
```

Named Partial Differential Equation Terms (Mathematica 12.2)

Named partial differential equation terms:

LaplacianPDETerm—model with $\nabla^2 u$

PoissonPDEComponent—model with $\nabla^2 u - f$

HelmholtzPDEComponent—model with $\nabla^2 u + k^2 u$

WavePDEComponent—model with $\partial u^2 / \partial t^2 - c^2 \nabla^2 u$

DirichletCondition ▪ NeumannValue ▪ PeriodicBoundaryCondition

Find eigenvalues of a Laplacian term:

```
NDEigenvalues [LaplacianPDETerm [{u[x],{x}}],u,{x}∈Line[{{0},{1}}],3]
```

Solve a Poisson equation:

```
NDSolveValue [{PoissonPDEComponent  [{u[x,y],{x,y}},<|"PoissonSourceTerm "→1|>]==0,DirichletConditior
```

Domain Models

Acoustics

AcousticPDEComponent—model acoustics in the time or frequency domain

AcousticAbsorbingValue—truncate an infinite region to a finite one

AcousticImpedanceValue—model partially sound-transparent boundary

AcousticNormalVelocityValue—model sound sources on the boundary

AcousticPressureCondition—set a pressure source at the boundary

AcousticRadiationValue—model sound sources and sinks on the boundary

AcousticSoundHardValue—model an acoustic wall on the boundary

AcousticSoundSoftCondition—set pressure equal to the ambient reference pressure

Domain Models

Heat transfer

HeatTransferPDEComponent—model conservative and non-conservative heat transfer

HeatFluxValue—model heat flow through a boundary

HeatInsulationValue—model thermal insulation

HeatOutflowValue—model a heat outlet

HeatRadiationValue—model heating or cooling through radiation

HeatSymmetryValue—model a boundary with mirror symmetry

HeatTemperatureCondition—set a specific temperature on the boundary

HeatTransferValue—model a cooling or heating flow

PDEModels Overview

Structural mechanics

Electromagnetics

Mass transport