(Custom CSS files are not reliable for controlling Jupyter font style. To establish the same appearance as the original notebook, depend on the browser to control the font, by setting the desired font faces in the browser settings. For example, Chrome 135 or Firefox 134 can do this. In this notebook series, Bookerly font is for markdown and Monaco is for code.)

Chapter 31-8 Method of Characteristics.

In mathematics, the method of characteristics is a technique for solving partial differential equations. Typically, it applies to first-order equations, although more generally the method of characteristics is valid for any hyperbolic partial differential equation. The method is to reduce a partial differential equation to a family of ordinary differential equations along which the solution can be integrated from some initial data given on a suitable hypersurface.

Characteristics of first-order partial differential equation

For a first-order PDE (partial differential equation), the method of characteristics discovers curves (called characteristic curves or just characteristics) along which the PDE becomes an ordinary differential equation (ODE). Once the ODE is found, it can be solved along the characteristic curves and transformed into a solution for the original PDE.

For the sake of simplicity, attention is restricted to the case of a function of two independent variables x and y for the moment. Consider a quasilinear PDE of the form

$$a(x, y, z)\frac{\partial z}{\partial x} \ + \ b(x, y, z)\frac{\partial z}{\partial y} \ = \ c(x, y, z) \tag{1}$$

Suppose that a solution z is known, and consider the surface graph $z = z(x, y)$ in $\mathbb{R3}$. A normal vector to this surface is given by

$$\left(\frac{\partial z}{\partial x}(x, y), \ \frac{\partial z}{\partial y}(x, y), \ -1\right)$$

As a result, equation (1) is equivalent to the geometrical statement that the vector field

$$(a(x, y, z), b(x, y, z), c(x, y, z))$$

is tangent to the surface $z = z(x, y)$ at every point, for the dot product of this vector field with the above normal vector is zero. In other words, the graph of the solution must be a union of integral curves of this vector field. These integral curves are called the characteristic curves of the original partial differential equation and are given by the Lagrange–Charpit equations

$$\begin{aligned} \frac{dx}{dt} &= a(x, y, z) \\ \frac{dy}{dt} &= b(x, y, z) \\ \frac{dz}{dt} &= c(x, y, z) \end{aligned}$$

These equations constitute a template, and by this means a solution can be discovered.

Following is an example in Octave, based on the material at https://terpconnect.umd.edu/~petersd/462/charact.pdf (https://terpconnect.umd.edu/~petersd/462/charact.pdf). The first two scripts must be available in the same directory as the third at the time the third script is executed.

In [ ]:

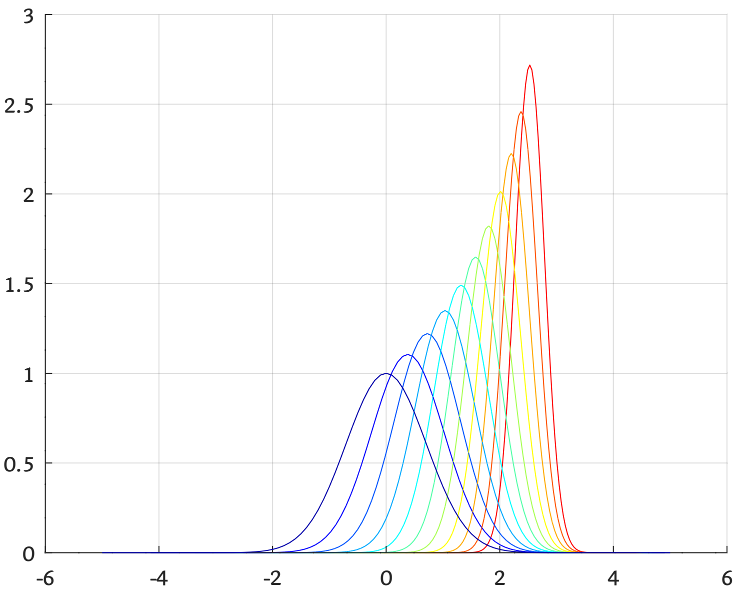
```
function colorcurves(x,y,z)
%   colorcurves(x,y,z)
%   colorcurves(z)     % use 1,2,3,... for x,y
% plot columns of array z as curves vs x
% Note that you can rotate the graph for a 3 dimensional view
%   z: array of size m by n
%   x: vector of length m, or array of size m by n
%   y: vector of length n, or array of size m by n
%
% Example: show graphs of x^4+a*x^2-a/2 for x=-2:.1:2, a=-4:.4:4 together
%   [X,A] = ndgrid(-2:.1:2,-4:.4:4);
%   colorcurves(X,A,X.^4+A.*X.^2-A/2)  % rotate graph with mouse!

if nargin==1
  z = x;
  [m,n] = size(z);
  x = 1:n;
  y = 1:m;
end
[m,n] = size(z);
if min(size(x))==1
  x = repmat(x(:),1,n);
end
if min(size(y))==1
  y = repmat(y(:)',m,1);
end
newplot
set(gca,'colororder',jet(n))
line(x,y,z)
grid on;
view(0,0)
cameratoolbar('SetMode','orbit');
```

In [ ]:

```
function [X,v] = quasilin(c,f,u0,xval,tval)
% [X,v] = quasilin(c,f,u0,xval,tval)
% solve quasilinear 1st order PDE
%   u_t + c(x,t,u)*u_x = f(x,t,u),    u(0)=u0(x)
% using method of characteristics (solving ODEs with ode45)
%
% inputs:
%   c:    coefficient fct c, define by c = @(x,t,u) ...
%   f:    right hand side fct f, define by f = @(x,t,u) ...
%   u0:   initial condition fct u0, define by u0 = @(x) ...
%   xval: values for x0, define by xval = xmin:step:xmax
%   tval: values for t,  define by tval = tmin:step:tmax
% outputs:
%   X: x-values of characteristics
%   v: u-values of solution
% X,v are arrays where row index corresponds to x0-values xval
%                   column index corresponds to t-values tval
% at time tval(j) the x-values are X(:,j), the u-values are v(:,j)
% to plot the solution u vs. x at time tval(3) use  plot(X(:,3),v(:,3))
%
% Example: Burgers equation u_t + u*u_x = 0, u(x,0)=atan(x)
%   c = @(x,t,u) u;       % c(x,t,u)=u
%   f = @(x,t,u) 0;       % f(x,t,u)=0
%   u0 = @(x) atan(x);    % initial condition u0(x)=atan(x)
%  [X,v] = quasilin(c,f,u0,-5:.1:5,0:.1:3);

% This uses colorcurves.m
% Download colorcurves.m from the course web page!

Nx = length(xval);
Nt = length(tval);
X = zeros(Nx,Nt);
v = zeros(Nx,Nt);
%  need to solve ODE system  X'=c(X,t,v), v'=F(X,t,v)
%  let w = (X,v), i.e., w(1)=X, w(2)=v
%  then we can write ODE system as w'=G(t,w)
G = @(t,w) [ c(w(1),t,w(2)) ; f(w(1),t,w(2)) ]; % define fct G
for i=1:Nx
  x0 = xval(i);
  [ts,ws] = ode45(G,tval,[x0;u0(x0)]); % solve system of ODEs w'=G(t,w)
  X(i,:) = ws(:,1);   % 1st column of ws are values of X, store as row
  v(i,:) = ws(:,2);   % 2nd column of ws are values of v, store as row
end
% each row of array X contains solution for one value of x0
% each row of array v contains solution for one value of x0

% plot characteristics in (x,t) plane
figure(1)
plot(X',tval,'k')     % plot rows of X vs tval
xlabel('x'); ylabel('t')
title('characteristics')

% plot solutions u(x,t_j) vs x, for all times
figure(2)
colorcurves(X,tval,v)  % plot columns of v vs. columns of X
xlabel('x'); ylabel('t'); zlabel('u')
title('Rotate this graph with the mouse')
```

In [ ]:

```
c = @(x,t,u) 4-x;
f = @(x,t,u) u;
u0 = @(x) exp(-x^2);
[X,v] = quasilin(c,f,u0,-5:.1:5,0:.1:1);
```


3
2.5
2
1.5
1
0.5
0
-6
-4
-2
0
2
4
6