Spectral Method for Wave Equation using FFT

In[1]:=
```
Clear["Global`*"]
```

Variable coefficient wave equation (Trefethen’s example p6.m)

Solve numerically the differential equation

$$\frac{\partial u(x,t)}{\partial t} = c(x)\frac{\partial u(x,t)}{\partial x}, \qquad c(x) = \frac{1}{5} + \sin^2(x-1) \tag{1}$$

for $x \in [0, 2\pi]$, $t > 0$ with the following initial condition

$$u(x,0) = e^{-100(x-1)^2} \tag{2}$$

and periodic boundary conditions

$$u(0,t) = u(2\pi, t) \tag{3}$$

Note that the initial function is not periodic but it is so close to zero at the ends of the interval that it can be regarded as periodic in practice.

In[2]:=
```
(* Grid, variable coefficient, and initial data *)
n = 128;
h = 2.0 * Pi / n;
x = Table[h * i, {i, 1, n}];
t = 0;
dt = h / 40
c = 0.2 + Sin[x - 1] ^2;
v = Exp[-100 * (x - 1) ^2];
(* For the leap-frog method below, we need another initial function for time -dt *)
vold = Exp[-100 * (x - 0.2 * dt - 1) ^2]; (* c(x) at x = 1 is close to -1/5 *)
ListPlot[{Transpose[{x, c}], Transpose[{x, v}], Transpose[{x, vold}]},
 PlotStyle → {PointSize[0.015]}, Joined → True, Mesh → All,
 PlotLegends → {"c(x)", "u(x,0)", "u(x,-dt)"}]
```

Out[6]= 0.00122718463031

General: Exp[−719.073126994] is too small to represent as a normalized machine number; precision may be lost.

General: Exp[−745.640177895] is too small to represent as a normalized machine number; precision may be lost.

General: Exp[−772.689143073] is too small to represent as a normalized machine number; precision may be lost.

General: Further output of General::munfl will be suppressed during this calculation.

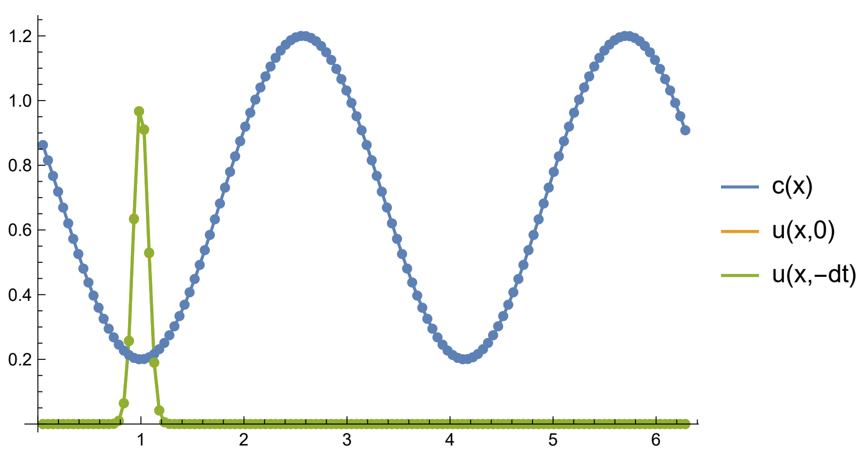
General: Exp[−718.941502549] is too small to represent as a normalized machine number; precision may be lost.

General: Exp[−745.506143879] is too small to represent as a normalized machine number; precision may be lost.

General: Exp[−772.552699486] is too small to represent as a normalized machine number; precision may be lost.

General: Further output of General::munfl will be suppressed during this calculation.

Out[10]=



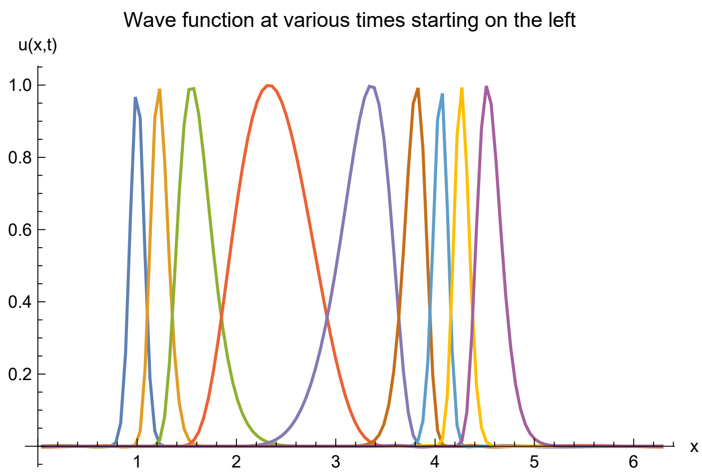
In[11]:=
```
(* Setting times for plotting and time step accordingly *)
tmax = 8;
tplot = 0.1;
plotgap = Round[tplot / dt];
dt = tplot / plotgap;
nplots = Round[tmax / tplot];
data = Join[{v}, Table[ConstantArray[0, n], {nplots}]];
tdata = {t};
```

In[18]:=
```
(* Time evolution of the wave *)

(* Create k vector for spectral differentiation to multiply the Fourier image by i*k *)
(* Notice 0 instead of n/2 ! *)
k = Join[Range[0, n / 2 - 1], {0}, Range[-n / 2 + 1, -1]];

(* Main time-stepping loop *)
Timing[For[i = 1, i ≤ nplots, i++,
   For[j = 1, j ≤ plotgap, j++, t = t + dt;
     (* Using Fourier transform to evaluate derivatives *)
     vhat = Fourier[v, FourierParameters → {1, -1}];
     what = I * k * vhat;
     w = Re[InverseFourier[what, FourierParameters → {1, -1}]];
     (* Time-stepping by the leap-frog formula *)
     vnew = vold - 2.0 * dt * c * w;
     vold = v;
     v = vnew;
   ];
   data[[i + 1]] = v;
   AppendTo[tdata, t];
  ];
]
```

Out[19]=
```
{0.140625, Null}
```

In[25]:=
```
ListPlot[Table[Transpose[{x, data〚it, All〛}], {it, 1, 81, 10}],
 PlotRange → All, Joined → True, AxesLabel → {"x", "u(x,t)"},
 PlotLabel → "Wave function at various times starting on the left"]
```

Out[25]=



In[26]:=
```
ListPlot3D[
 Flatten[Table[{x⟦j⟧, tdata⟦i⟧, data⟦i, j⟧}, {i, 1, Length[tdata]}, {j, 1, Length[x]}], 1],
 PlotRange → {{0, 2 * Pi}, {0, tmax}, {-1, 5}}, AxesLabel → {"x", "t", "u"}, Mesh → {nplots}]
```

Out[26]=