HW 6 - ASTR510

Daniel George

Q1

Taylor expansions

q(t+dt)

In[1]:=
```
TE1 = Series[q[t + dt], {dt, 0, 3}]
```

Out[1]= $q[t] + q'[t]\,dt + \frac{1}{2} q''[t]\,dt^2 + \frac{1}{6} q^{(3)}[t]\,dt^3 + O[dt]^4$

q(t+1/2dt)

In[2]:=
```
TE2 = Series[q[t + 1/2 dt], {dt, 0, 3}]
```

Out[2]= $q[t] + \frac{1}{2} q'[t]\,dt + \frac{1}{8} q''[t]\,dt^2 + \frac{1}{48} q^{(3)}[t]\,dt^3 + O[dt]^4$

Defining linear operators A and B

In[3]:=
```
A[x_ + y_] = A[x] + A[y]; A[c_ q_[t_]] = c A[q[t]]; A[0] = 0;
B[x_ + y_] = B[x] + B[y]; B[c_ q_[t_]] = c B[q[t]]; B[0] = 0;
```

Correct value of q after time dt

In[5]:=
```
eqC = q[t + dt] → TE1 //.
  Derivative[n_][q][t] → A[Derivative[n - 1][q][t]] + B[Derivative[n - 1][q][t]]
```

Out[5]= $q[dt + t] \to q[t] + (A[q[t]] + B[q[t]])\,dt + \frac{1}{2} (A[A[q[t]]] + A[B[q[t]]] + B[A[q[t]]] + B[B[q[t]]])\,dt^2 + \frac{1}{6} (A[A[A[q[t]]]] + A[A[B[q[t]]]] + A[B[A[q[t]]]] + A[B[B[q[t]]]] + B[A[A[q[t]]]] + B[A[B[q[t]]]] + B[B[A[q[t]]]] + B[B[B[q[t]]]])\,dt^3 + O[dt]^4$

a)

Computing value using Godunov method

Replacing derivatives in Taylor expansion using operator B

In[6]:=
```
step1G = TE1 //. Derivative[n_][q][t] → B[Derivative[n - 1][q][t]] // Normal
```

Out[6]= $\frac{1}{6}\,dt^3\,B[B[B[q[t]]]] + \frac{1}{2}\,dt^2\,B[B[q[t]]] + dt\,B[q[t]] + q[t]$

Substituting this expression for q(t) in the Taylor expansion having operator A

In[7]:=
```
step2G = (step1G /. {B → A}) /. q[t] -> step1G // Simplify // Normal[# + O[dt]^4] &
```

Out[7]= $\frac{1}{6}\,dt^3\,(A[A[A[q[t]]]] + 3\,A[A[B[q[t]]]] + 3\,A[B[B[q[t]]]] + B[B[B[q[t]]]]) + \frac{1}{2}\,dt^2\,(A[A[q[t]]] + 2\,A[B[q[t]]] + B[B[q[t]]]) + dt\,(A[q[t]] + B[q[t]]) + q[t]$

Final value of q after time dt using Godunov method

In[8]:=
```
eqG = q[t + dt] → step2G + O[dt]^4
```

Out[8]= $q[dt + t] \to q[t] + (A[q[t]] + B[q[t]])\,dt + \frac{1}{2}\,(A[A[q[t]]] + 2\,A[B[q[t]]] + B[B[q[t]]])\,dt^2 + \frac{1}{6}\,(A[A[A[q[t]]]] + 3\,A[A[B[q[t]]]] + 3\,A[B[B[q[t]]]] + B[B[B[q[t]]]])\,dt^3 + O[dt]^4$

Error when using Godunov method

Subtracting the value of q obtained using Godunov method from the correct value, after time dt

In[9]:=
```
eqC[[2]] - eqG[[2]] // Simplify
```

Out[9]= $\frac{1}{2}\,(-A[B[q[t]]] + B[A[q[t]]])\,dt^2 + \frac{1}{6}\,(-2\,A[A[B[q[t]]]] + A[B[A[q[t]]]] - 2\,A[B[B[q[t]]]] + B[A[A[q[t]]]] + B[A[B[q[t]]]] + B[B[A[q[t]]]])\,dt^3 + O[dt]^4$

Therefore we can see that the error after one time step is of order dt^2 if the operators A and B are not commutative.

b)

Computing value using Strang splitting

Replacing derivatives in Taylor expansion for $\frac{1}{2}$dt using operator A

```
In[10]:= step1S = TE2 //. Derivative[n_][q][t] → A[Derivative[n - 1][q][t]] // Normal
```

Out[10]= $\frac{1}{48}\,dt^3\,A[A[A[q[t]]]] + \frac{1}{8}\,dt^2\,A[A[q[t]]] + \frac{1}{2}\,dt\,A[q[t]] + q[t]$

Substituting this expression for q(t) in the Taylor expansion having operator B

```
In[11]:= step2S = (step1S /. {A → B}) /. q[t] → step1S // Normal[# + O[dt]^4] &
```

Out[11]= $\frac{1}{48}\,dt^3\,(A[A[A[q[t]]]] + 3\,B[A[A[q[t]]]] + 3\,B[B[A[q[t]]]] + B[B[B[q[t]]]]) + \frac{1}{8}\,dt^2\,(A[A[q[t]]] + 2\,B[A[q[t]]] + B[B[q[t]]]) + \frac{1}{2}\,dt\,(A[q[t]] + B[q[t]]) + q[t]$

Substituting this expression for q(t) in the Taylor expansion having operator B

```
In[12]:= step3S = (step1S /. {A → B}) /. q[t] → step2S // Simplify // Normal[# + O[dt]^4] &
```

Out[12]= $\frac{1}{48}\,dt^3\,(A[A[A[q[t]]]] + 6\,B[A[A[q[t]]]] + 12\,B[B[A[q[t]]]] + 8\,B[B[B[q[t]]]]) + \frac{1}{8}\,dt^2\,(A[A[q[t]]] + 4\,B[A[q[t]]] + 4\,B[B[q[t]]]) + dt\left(\frac{1}{2}\,A[q[t]] + B[q[t]]\right) + q[t]$

Substituting this expression for q(t) in the Taylor expansion having operator A

```
In[13]:= step4S = step1S /. q[t] → step3S // FullSimplify // Normal[# + O[dt]^4] &
```

Out[13]= $\frac{1}{24}\,dt^3\,(4\,A[A[A[q[t]]]] + 3\,A[A[B[q[t]]]] + 6\,A[B[A[q[t]]]] + 6\,A[B[B[q[t]]]] + 3\,B[A[A[q[t]]]] + 6\,B[B[A[q[t]]]] + 4\,B[B[B[q[t]]]]) + \frac{1}{2}\,dt^2\,(A[A[q[t]]] + A[B[q[t]]] + B[A[q[t]]] + B[B[q[t]]]) + dt\,(A[q[t]] + B[q[t]]) + q[t]$

Final value of q after time dt using Strang splitting

In[14]:=
```
eqS = q[t + dt] → step4S + O[dt] ^ 4 // Simplify
```

Out[14]=
$$q[dt+t] \to q[t] + (A[q[t]] + B[q[t]])\, dt + \frac{1}{2}(A[A[q[t]]] + A[B[q[t]]] + B[A[q[t]]] + B[B[q[t]]])\, dt^2 + \frac{1}{24}(4\,A[A[A[q[t]]]] + 3\,A[A[B[q[t]]]] + 6\,A[B[A[q[t]]]] + 6\,A[B[B[q[t]]]] + 3\,B[A[A[q[t]]]] + 6\,B[B[A[q[t]]]] + 4\,B[B[B[q[t]]]])\, dt^3 + O[dt]^4$$

Error when using Strang splitting

Subtracting the value of q obtained using Strang splitting from the correct value, after time dt

In[15]:=
```
eqC[[2]] - eqS[[2]] // Simplify
```

Out[15]=
$$\frac{1}{24}(A[A[B[q[t]]]] - 2\,A[B[A[q[t]]]] - 2\,A[B[B[q[t]]]] + B[A[A[q[t]]]] + 4\,B[A[B[q[t]]]] - 2\,B[B[A[q[t]]]])\, dt^3 + O[dt]^4$$

Therefore we can see that the error after one time step is of order dt^3 if the operators A and B are not commutative.

c)

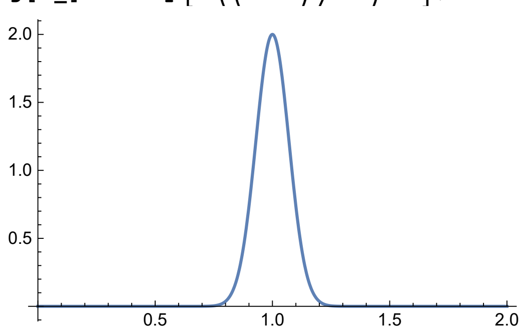
Burger's Equation

$$\frac{\partial u(x,t)}{\partial t} + u(x,t)\frac{\partial u(x,t)}{\partial x} = \mu\frac{\partial^2 u(x,t)}{\partial x^2}$$

Initial condition

In[16]:=
```
g[x_] = 2 Exp[-((x - 1) / .1) ^2]; Plot[g[x], {x, 0, 2}, PlotRange → All]
```

Out[16]=



Function to initialize array given g(x) and dx

```
In[17]:= init = Function[{g, dx}, g /@ Range[0, 2, dx]];
```

Lax Wendroff (Richtmyer) scheme for advection

Source: https://en.wikipedia.org/wiki/Lax–Wendroff_method

```
f[x_] = x^2 / 2;
In[19]:= up = 1 / 2 (u[i + 1] + u[i]) - dt / (2 dx) (f[u[i + 1]] - f[u[i]]);
In[20]:= um = 1 / 2 (u[i] + u[i - 1]) - dt / (2 dx) (f[u[i]] - f[u[i - 1]]);
```

Update rule for LW Richtmyer scheme

```
In[21]:= rLWR = u[i] -> u[i] - dt / dx (f[up] - f[um]); rLWR // TraditionalForm
```

Out[21]//TraditionalForm=

$$u(i) \to u(i) - \frac{1}{\mathrm{dx}}\mathrm{dt}\left(\frac{1}{2}\left(\frac{1}{2}(u(i)+u(i+1)) - \frac{\mathrm{dt}\left(\frac{1}{2}u(i+1)^2 - \frac{u(i)^2}{2}\right)}{2\,\mathrm{dx}}\right)^2 - \frac{1}{2}\left(\frac{1}{2}(u(i-1)+u(i)) - \frac{\mathrm{dt}\left(\frac{u(i)^2}{2} - \frac{1}{2}u(i-1)^2\right)}{2\,\mathrm{dx}}\right)^2\right)$$

Simplifying RHS

```
In[22]:= exp1 = rLWR[[2]] /. {u[i + 1] → #3, u[i] → #2, u[i - 1] → #1} // Simplify // InputForm
```

Out[22]//InputForm=

```
#2 - (dt*(-((2*dx + dt*(#1 - #2))^2*(#1 + #2)^2) + (2*dx + dt*(#2 - #3))^2*(#2 + #3)
```

Function to advance by one time-step using LW Richtmyer method

```
In[23]:= LWR = Function[{v, dx, dt}, ArrayPad[ArrayFilter[
        #2 - (dt * (- ((2 * dx + dt * (#1 - #2)) ^2 * (#1 + #2) ^2) + (2 * dx + dt * (#2 - #3)) ^2 *
              (#2 + #3) ^2)) / (32 * dx ^3) &@@# &, v, 1][[2 ;; -2]], 1]];
```

Crank-Nicholson scheme for diffusion part

Equation to solve to update by one time step (u → v)

```
In[31]:= (u[i][n + 1] - u[i][n]) / dt - μ / 2 / dx^2 (u[i + 1][n + 1] -
              2 u[i][n + 1] + u[i - 1][n + 1] + u[i + 1][n] - 2 u[i][n] + u[i - 1][n]) == 0 /.
       {u[x_][n + 1] → v[x], u[x_][n] → u[[x]]} // TraditionalForm // Quiet
```

Out[31]//TraditionalForm=

$$\frac{v(i) - u[\![i]\!]}{\mathrm{dt}} - \frac{\mu\,(u[\![i-1]\!] - 2\,u[\![i]\!] + u[\![i+1]\!] + v(i-1) - 2\,v(i) + v(i+1))}{2\,\mathrm{dx}^2} = 0$$

Function to update by one time step using CN with built-in LinearSolve function

```
In[32]:= CN = Function[{u, dx, dt, μ}, Module[{v, n}, n = Length@u;
      v[1] = 0;
      v[n] = 0;
      ArrayPad[
       LinearSolve[#[[2]], -#[[1]]] &@CoefficientArrays[Table[(-u[[i]] + v[i]) / dt -
            (μ * (u[[-1 + i]] - 2 * u[[i]] + u[[1 + i]] + v[-1 + i] - 2 * v[i] + v[1 + i])) /
             (2 * dx^2) == 0, {i, 2, n - 1}], Table[v[i], {i, 2, n - 1}]], 1]]];
```

Faster function specifically for dx=0.02 and μ=0.01

```
In[148]:= sol[dt_] :=
     sol[dt] = LinearSolve[CoefficientArrays[Table[-25 dt v[-1 + i] + 2 (1 + 25 dt) v[i] -
              25 dt v[1 + i] == 25 dt u[-1 + i] + 2 (1 - 25 dt) u[i] + 25 dt u[1 + i],
            {i, 2, 100}], Table[v[i], {i, 2, 100}]][[2]]];
```

```
In[149]:= fCN =
     Function[{u, dx, dt, μ}, ArrayPad[sol[dt]@
         (ArrayFilter[25 dt #3 + 2 (1 - 25 dt) #2 + 25 dt #1 &@@# &, u, 1][[2 ;; -2]]), 1]];
```

Combining both schemes using Godunov method

Function to iterate upto one second given initial f(x), dx, dt, and μ

```
In[177]:= iterG = Function[{f, dx, dt, μ}, NestWhileList[fCN[#, dx, dt, μ] &[LWR[#, dx, dt]] &,
       init[f, dx], Max@Abs@# < 3.5 &, 1, Floor[1/ dt]]];
```

Combining both schemes using Strang splitting

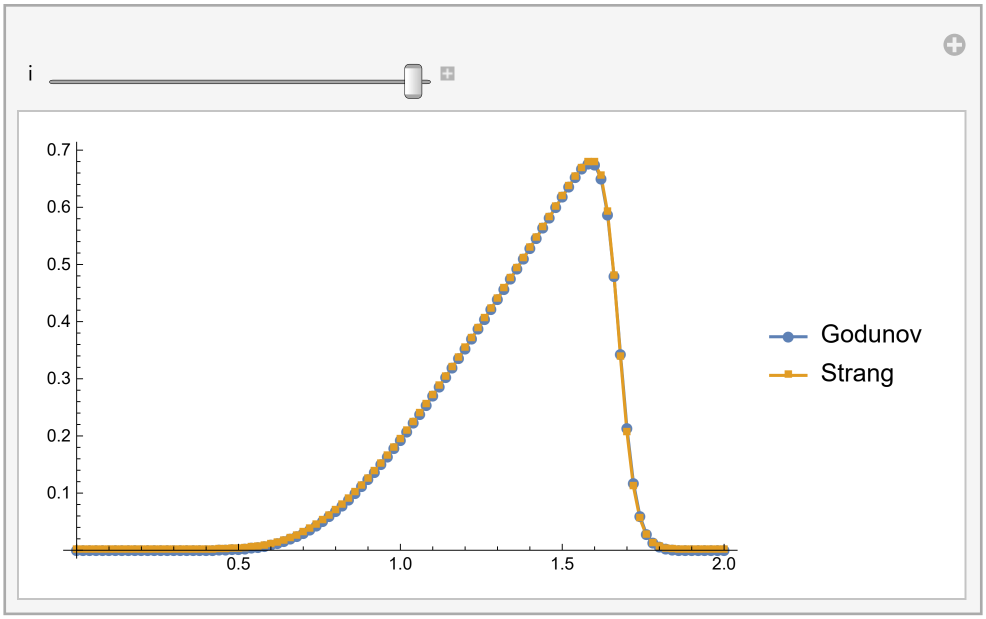
Function to iterate upto one second given initial f(x), dx, dt, and μ

```
In[179]:= iterS = Function[{f, dx, dt, μ}, NestWhileList[LWR[#, dx, 1/2 dt] &[
          fCN[#, dx, 1/2 dt, μ] &[fCN[#, dx, 1/2 dt, μ] &[LWR[#, dx, 1/2 dt]]]] &,
       init[f, dx], Max@Abs@# < 3.5 &, 1, Floor[1/ dt]]];
```

Plotting solutions with both methods at t=1 for dx=0.02, dt=0.01, and μ=0.01

In[185]:=
```
With[{S = iterS[g, .02, .01, .01], G = iterG[g, .02, .01, .01]},
 Manipulate[ListLinePlot[Transpose[{Range[0, 2, .02], #[[i]]}] & /@ {G, S},
   PlotRange → All, PlotMarkers → Automatic,
   PlotLegends → {"Godunov", "Strang"}], {i, 1, Length@S, 1}]]
```

Out[185]=



Time step stability criterion

The maximum allowed timestep is determined by the CFL criterion for the Lax-Wendroff method:

```
Max[u] dt / dx == 1
```

Here maximum value of u is 2 from the given initial function.

Therefore for dx=0.02 the maximum allowed value of dt is 0.01.

d)

Range of time steps used

In[134]:=
```
trange = .01 × 10^Range[-4, 0]
```

Out[134]= $\{1. \times 10^{-6}, 0.00001, 0.0001, 0.001, 0.01\}$

Computing solutions using Godunov method for different time steps

In[138]:=
```
dataG = ParallelMap[iterG[g, .02, #, .01][[-1]] &, trange]; // AbsoluteTiming
```

Out[138]=
```
{3928.74212217, Null}
```

Computing solutions using Strang splitting for different time steps

In[139]:=
```
dataS = ParallelMap[iterS[g, .02, #, .01][[-1]] &, trange]; // AbsoluteTiming
```

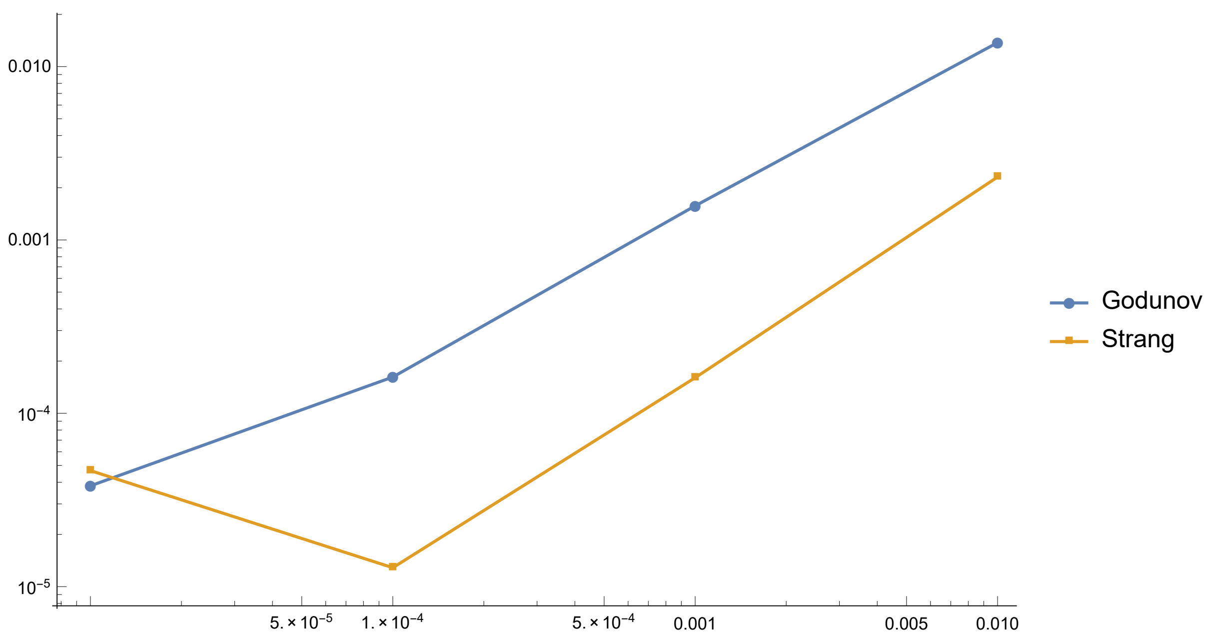
Out[139]=
```
{7324.2500477, Null}
```

Computing error

Plotting error (L2 norm) for both methods vs step size

In[194]:=
```
ListLogLogPlot[
 Function[d, Transpose[{trange[[2 ;;]], Norm[d[[1]] - d[[#]]] & /@ Range[2, 5]}]] /@
  {dataG, dataS}, Joined → True, PlotMarkers → Automatic,
 PlotLegends → {"Godunov", "Strang"}, ImageSize → Large]
```

Out[194]=



So in most cases, Strang Splitting seems to have steeper convergence as we have shown earlier (dt^3) compared to the Godunov method (which locally converges as dt^2).

Q2

a)

Finite volume methods can be set up so that the fluxes leaving and entering each zone are exactly conserved. Therefore conserved quantities remain conserved when using finite volume methods whereas this is not always the case for finite difference methods.

b)

The information takes O(n) steps to propagate from one end of the array to the opposite end for each wavenumber. Therefore it takes $O(n)^2$ steps to converge.

c)

No idea. Wthout internet, I guess I'm not really much better than a caveman.

d)

When particles get too close to each other, the forces and potentials can become inifinitely large causing overflows. Also this can lead to all the particles converging to a single point. Therefore force softening is used to prevent this from happening.

e)

Riemann solvers are used to calculate the exact value of the fluxes at the cell boundaries.

f)

Verification means ensuring that the solutions to a mathematical model are correct. Validation means that the solution works to model the real world in some meaningful manner.

g)

Magnetohydrodynamics allows for more types of characterstics waves (around 7) compared to hydrodynamics (around 3).

h)

Latency and throughput are important for a parallel computer for achieving good parallel efficiency.

i)

No clue. Was in India during classes on radiation.

j)

Stiff equations require timesteps to be extremely small compared to the total time.