Homework 7: Computational Physics

Pablo Lopez Duque

November 23^{rd}, 2020

1 Problem 1

The dimensionless Navier-Stokes (NS) equations are $\vec{\nabla} \cdot \vec{u} = 0$ and $\dot{\vec{u}} + (\vec{u} \cdot \vec{\nabla})\vec{u} = -\vec{\nabla} p + \frac{1}{Re} \nabla^2 \vec{u}$. This problem will consider this equation in two dimensions only, with $\vec{u} = (u_x; u_y; 0)$. The fluid is trapped between two walls (with the boundary conditions $u(y = 1) = u(y = -1) = 0$, and periodic boundary conditions on $\vec{u}$ in the x direction)

(a) **Show analytically that a solution to the NS equations is $p = p0 - \alpha x$, $u_x = \alpha Re(1 - y^2) = 2$, and $u_y = 0$, for any α. This parabolic flow profile is referred to as Poiseuille flow.**

In order to find to show that the given solution indeed satisfies the NS equations, we first find a general solution by considering the two dimensional problem and the boundary conditions imposed by the infinite planes.

The equation for u_y is trivially satisfied as all terms become zero.

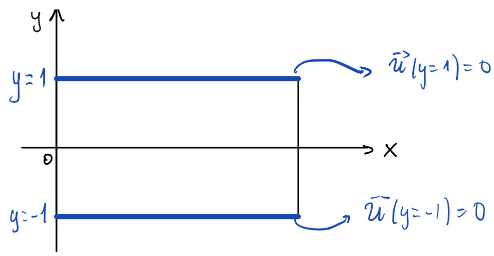
Then, we consider the equations for u_x, and apply the steady state conditions, i.e. $\partial_t = 0$ and $\partial_x = 0$

Finally, we integrate and find the integration constants by applying the symmetry and BC's.

1

$$\vec{\nabla}\cdot\vec{u} = 0$$

$$\dot{\vec{u}} + (\vec{u}\cdot\vec{\nabla})\vec{u} = -\vec{\nabla}p + \frac{1}{Re}\nabla^2\vec{u}$$



Consider $\vec{u} = (u_x, u_y)$

in x periodic B.C. $u(x=0) = u(x=L)$

a

(1) $\dot{u}_x + u_x\partial_x u_x + u_y\partial_y u_x = -\frac{\partial p}{\partial x} + \frac{1}{Re}\left[\partial_x^2 + \partial_y^2\right]u_x$

(2) $\dot{u}_y + u_x\partial_x u_y + u_y\partial_y u_y = -\frac{\partial p}{\partial y} + \frac{1}{Re}\left[\partial_x^2 + \partial_y^2\right]u_y$

we expect $u_y = 0$ (flow parallel to walls) $\quad \frac{dp}{dy} = 0 \quad u_x = u_x(y)$

$\Rightarrow$ (2) $0=0$ ✓

$$\dot{u}_x + u_x\partial_x u_x = -\frac{\partial p}{\partial x} + \frac{1}{Re}\partial_y^2 u_x$$

when we reach steady state $\frac{\partial}{\partial t} = 0 \quad \frac{\partial}{\partial x} = 0 \Rightarrow 0 = -\frac{\partial p}{\partial x} + \frac{1}{Re}\frac{\partial^2 u_x}{\partial y^2}$

$$\left.\begin{aligned} &\Rightarrow \int \frac{\partial p}{\partial x}\,dy = \frac{1}{Re}\int \frac{\partial^2 u_x}{\partial y^2}\,dy \\ &y\frac{\partial p}{\partial x} = \frac{1}{Re}\cdot\frac{\partial u_x}{\partial y} + C_1 \end{aligned}\right\} \Rightarrow \text{at } y=0,\ \frac{\partial u_x}{\partial y} = 0 \Rightarrow \boxed{C_1 = 0}$$

$$\left.\begin{aligned} &\frac{\partial p}{\partial x}\int y\,dy = \frac{1}{Re}\int \frac{\partial u_x}{\partial y}\,dy \\ &\frac{y^2}{2}\frac{\partial p}{\partial x} = \frac{1}{Re}u_x + C_2 \end{aligned}\right\} \Rightarrow \text{at } y=\pm 1\ \ u_x = 0 \Rightarrow \boxed{\frac{1}{2}\frac{dp}{dx} = C_2}$$

$$\Rightarrow u_x(y) = Re\left(\frac{y^2}{2}\frac{dp}{dx} - \frac{1}{2}\frac{dp}{dx}\right) = -\frac{Re}{2}\frac{dp}{dx}\left(1-y^2\right) //$$

let $p = \alpha x + p_0 \Rightarrow \frac{dp}{dx} = \alpha \qquad \therefore\ u_x = -\frac{\alpha Re}{2}\left(1-y^2\right) //$

$$\therefore \begin{cases} u_x = -\alpha\frac{Re}{2}\left(1-y^2\right) \\ u_y = 0 \end{cases} \quad \text{with } p = \alpha x + p_0 \text{ is a solution!}$$

(b) **The vorticity of a flow is defined as $\vec{\omega} = \vec{\nabla} \times \vec{u}$, with $\vec{\omega} = (0; 0; \omega)$ for a 2D system. For an arbitrary 2D flow (not the Poiseuille flow in (a)), show that $\dot{\omega} + (u \cdot \vec{\nabla})\omega = \frac{1}{Re}\nabla^2\omega$. What are the boundary conditions on ω?**

(b) $\vec{\omega} = \vec{\nabla} \times \vec{u} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \partial_x & \partial_y & \partial_z \\ u_x & u_y & 0 \end{vmatrix} = \hat{i}\left(-\partial_z u_y\right) - \hat{j}\left(-\partial_z u_x\right) + \hat{k}\left(\partial_x u_y - \partial_y u_x\right)$

$\Rightarrow \boxed{\partial_z u_y = 0} \quad \boxed{\partial_z u_x = 0} \quad \Rightarrow \quad u_x = u_x(x,y), \quad u_y = u_y(x,y)$

$\omega = \partial_x u_y - \partial_y u_x \quad \Rightarrow |\omega| = |\partial_x u_y - \partial_y u_x|$

$\dot{u}_x + u_x \partial_x u_x + u_y \partial_y u_x = -\frac{\partial p}{\partial x} + \frac{1}{Re}\left[\partial_x^2 + \partial_y^2\right] u_x$

$\dot{u}_y + u_x \partial_x u_y + u_y \partial_y u_y = -\frac{\partial p}{\partial y} + \frac{1}{Re}\left[\partial_x^2 + \partial_y^2\right] u_y$

$\dot{\omega} = \partial_t \partial_x u_y - \partial_t \partial_y u_x = \partial_x(\partial_t u_y) - \partial_y(\partial_t u_x)$

$\Rightarrow \dot{\omega} = \partial_x\left(-\partial_y p + \frac{1}{Re}(\partial_x^2 + \partial_y^2)u_y - u_x \partial_x u_y - u_y \partial_y u_y\right) - \partial_y\left(-\partial_x p + \frac{1}{Re}(\partial_x^2 + \partial_y^2)u_x - u_x \partial_x u_x - u_y \partial_y u_x\right)$

$= -\partial_x \cancel{\partial_y p} + \partial_y \cancel{\partial_x p} + \frac{1}{Re}\left[\partial_x\left((\partial_x^2 + \partial_y^2)u_y\right) - \left(\partial_y(\partial_x^2 + \partial_y^2)u_x\right)\right] - \partial_x\left(u_x \partial_x u_y + u_y \partial_y u_y\right) + \partial_y\left(u_x \partial_x u_x + u_y \partial_y u_x\right)$

$= \frac{1}{Re}\left[\partial_x\left((\partial_x^2 + \partial_y^2)u_y\right) - \left(\partial_y(\partial_x^2 + \partial_y^2)u_x\right)\right] - (\partial_x u_x)(\partial_x u_y) - u_x \partial_x^2 u_y - (\partial_x u_y)(\partial_y u_y) - u_y \partial_x \overbrace{\partial_y u_y}^{-\partial_x u_x}$

$+ (\partial_y u_x)(\partial_x u_x) + u_x \partial_y \underbrace{\partial_x u_x}_{-\partial_y u_y} + (\partial_y u_y)(\partial_y u_x) + u_y \partial_y^2 u_x$

$= \frac{1}{Re}\left[\partial_x\left((\partial_x^2 + \partial_y^2)u_y\right) - \left(\partial_y(\partial_x^2 + \partial_y^2)u_x\right)\right] - \cancel{(\partial_x u_x)(\partial_x u_y)} - u_x \partial_x^2 u_y + \cancel{(\partial_x u_y)(\partial_x u_x)} + u_y \partial_x^2 u_x$

$+ \cancel{(\partial_y u_x)(\partial_x u_x)} - u_x \partial_y^2 u_y - \cancel{(\partial_x u_x)(\partial_y u_x)} + u_y \partial_y^2 u_x$

$= \frac{1}{Re}\left[\partial_x\left((\partial_x^2 + \partial_y^2)u_y\right) - \left(\partial_y(\partial_x^2 + \partial_y^2)u_x\right)\right] - u_x(\partial_x^2 + \partial_y^2)u_y + u_y(\partial_x^2 + \partial_y^2)u_x$

$(\vec{u} \cdot \vec{\nabla})\omega = (u_x \partial_x + u_y \partial_y)(\partial_x u_y - \partial_y u_x) = u_x \partial_x^2 u_y - u_x \partial_x \partial_y u_x + u_y \partial_y \partial_x u_y - u_y \partial_y^2 u_x$

$= u_x \partial_x^2 u_y - u_x \partial_y \partial_x u_x + u_y \partial_x \partial_y u_y - u_y \partial_y^2 u_x$

$\vec{\nabla} \cdot \vec{u} = 0 \quad \partial_x u_x + \partial_y u_y = 0$

$= u_x \partial_x^2 u_y - u_x \partial_y(-\partial_y u_y) + u_y \partial_x(-\partial_x u_x) - u_y \partial_y^2 u_x$

$= u_x \partial_x^2 u_y + u_x \partial_y^2 u_y - u_y \partial_x^2 u_x - u_y \partial_y^2 u_x$

$= u_x(\partial_x^2 + \partial_y^2)u_y - u_y(\partial_x^2 + \partial_y^2)u_x$

$$\frac{1}{Re}\nabla^2\omega = \frac{1}{Re}\left(\partial_x^2+\partial_y^2\right)\left(\partial_x u_y - \partial_y u_x\right) = \frac{1}{Re}\left(\partial_x^3 u_y - \partial_x^2\partial_y u_x + \partial_y^2\partial_x u_y - \partial_y^3 u_x\right)$$

$$= \frac{1}{Re}\left[\partial_x\left(\partial_x^2 u_y - \underbrace{\partial_x\partial_y u_x}_{\partial_y\partial_x u_x}\right) - \partial_y\left(\partial_y^2 u_x - \underbrace{\partial_y\partial_x u_y}_{\partial_x\partial_y u_y}\right)\right]$$

$$= \frac{1}{Re}\left[\partial_x\left(\partial_x^2 u_y + \partial_y^2 u_y\right) - \partial_y\left(\partial_y^2 u_x + \partial_x^2 u_y\right)\right]$$

$$= \frac{1}{Re}\left[\partial_x\left(\left(\partial_x^2+\partial_y^2\right)u_y\right) - \partial_y\left(\left(\partial_x^2+\partial_y^2\right)u_x\right)\right]$$

$$\therefore \quad \boxed{\dot{\omega} + (\vec{u}\cdot\vec{\nabla})\,\omega = \frac{1}{Re}\nabla^2\omega \quad //}$$

$\rightarrow$ partial, non linear diff. eqn

$\omega = \partial_x u_y - \partial_y u_x \qquad \vec{\omega} = \vec{\nabla}\times\vec{u} \qquad \Rightarrow \quad \omega = 0$ at $y = \pm 1$

(c) **Can the PDE in (b) be solved using the methods for solving PDE's we described in class? If so, explain how. If not, describe a predictor-corrector method (like the SIMPLE algorithm described in class) that could be used?**

The issue to solve the original N-S equations are that we don't know the pressure p and that the equation is non linear, so we cannot use the linear algebra packages directly without some manipulation. By using the rotational of $\vec{u}$, we get rid of the problem of not knowing p, but we still have issues due to having ω and $\vec{u}$ unknowns. The term $(u \cdot \vec{\nabla})\omega$ is the source of the problem; without it, the equation reduces to a diffusion-like type. Still, once we find u, we can find ω easily. Now, it is conveninent to define a scalar function Ψ, such that:

$$\frac{\partial \Psi}{\partial x} = -u_y$$

$$\frac{\partial \Psi}{\partial y} = u_x$$

Now, recalling the equation for omega:

$$\partial_t \omega + (\vec{u} \cdot \nabla)\omega = \frac{1}{Re}\nabla^2 \omega$$

in 2d, it becomes:

$$\partial_t \omega + (u_x \partial_x + u_y \partial_y)\omega = \frac{1}{Re}(\partial_x^2 + \partial_y^2)\omega \tag{1}$$

And we can clearly see that:

$$\nabla^2 \Psi = (\partial_x^2 + \partial_y^2)\Psi = -\omega \tag{2}$$

Hence, we can implement a solution by first solving eqn (1) for ω and then use the updated values of $omega$ to solve eqn (2), which is essentially the Poisson equation. Notice that eqn(2) can be solved directly with any of the methods we used in class. However, eqn (1) cannot be solved using conventional methods such as overrelaxation. But we can still use a modified SIMPLE algorithm to address this problem (recall that the idea behind the SIMPLE algorithm was to initially solve the differential equation without pressure dependence and later correct it by adding an extra term to account for it). The algorithm is as follows:

- Guess an initial value for ω and Ψ.

- Solve eqn (1) to get updated values for ω.

- Update the values of the velocity by the discretized version of $\partial_x \Psi = -u_y$ and $\partial_y \Psi = u_x$ (correction step).

- Repeat until the algorithm converges to a predefined tolerance.

2 Problem 2

Firstly, let's solve the problem in general:

let $A[n(r)] = \int dr' \, a\left(n(r'), \frac{dn(r')}{dr'}\right)$

$$\Rightarrow \frac{\delta A[n(r)]}{\delta n(r)} = \int dr' \frac{\delta}{\delta n(r)} a\left(n(r'), \frac{dn(r')}{dr'}\right) = \int dr' \cdot \left(\frac{\partial a}{\partial n(r')} \cdot \overbrace{\frac{\delta n(r')}{\delta n(r)}}^{\delta(r-r')} + \frac{\partial a}{\partial\left(\frac{dn(r')}{dr'}\right)} \cdot \frac{\delta\left(\frac{dn(r')}{dr'}\right)}{\delta n(r)} \right)$$

$$= \int dr' \frac{\partial a}{\partial n(r')} \cdot \delta(r-r') + \int dr' \frac{\partial a}{\partial\left(\frac{dn(r')}{dr'}\right)} \cdot \frac{d}{dr'} \frac{\delta(n(r'))}{\delta n(r)}$$

$$= \left.\frac{\partial a}{\partial n(r')}\right|_{r'=r} + \int dr' \frac{\partial a}{\partial\left(\frac{dn(r')}{dr'}\right)} \cdot \frac{d}{dr'} \delta(r-r') = \left.\frac{\partial a}{\partial n(r')}\right|_{r'=r} + \int d(\delta(r-r')) \cdot \frac{\partial a}{\partial\left(\frac{dn(r')}{dr'}\right)}$$

$$= \left.\frac{\partial a}{\partial n(r')}\right|_{r'=r} + \underbrace{\cancel{\int_{\mathbb{R}^3} dr' \cdot \frac{d}{dr'}\left[\frac{\partial a}{\partial\left(\frac{dn(r')}{dr'}\right)} \cdot \delta(r-r')\right]}}_{0 \text{ in a compact domain}} - \int dr' \, \delta(r-r') \cdot \frac{d}{dr'} \cdot \left(\frac{\partial a}{\partial\left(\frac{dn(r')}{dr'}\right)}\right)$$

$$\therefore \frac{\delta A[n(r)]}{\delta n(r)} = \left.\frac{\partial a}{\partial n(r')}\right|_{r'=r} - \left.\frac{d}{dr'}\left(\frac{\partial a}{\partial\left(\frac{\partial n(r')}{dr'}\right)}\right)\right|_{r'=r} //.$$

Now, let's find $\delta A[n(r)]/\delta n(r)$ for:

(a) $A[n(r)] = \int dr' dr'' \frac{n(r')n(r'')}{|r'-r''|}$

(a) $A[n(r)] = \int dr' \int dr'' \frac{n(r')n(r'')}{|r'-r''|} \Rightarrow a = \int dr'' \frac{n(r')n(r'')}{|r'-r''|} \Rightarrow a = a(n(r'), r')$

$$\frac{\delta A[n(r)]}{\delta n(r)} = \left.\frac{\partial a}{\partial n(r')}\right|_{r'=r} - \left.\frac{d}{dr'}\left(\frac{\partial a}{\partial\left(\frac{dn(r')}{dr'}\right)}\right)\right|_{r'=r}$$

$$= \left.\frac{\partial}{\partial n(r')} \int dr'' \frac{n(r')n(r'')}{|r'-r''|}\right|_{r'=r} = \left.\int dr'' \frac{\partial}{\partial n(r')}\left(\frac{n(r')n(r'')}{|r'-r''|}\right)\right|_{r'=r} = \left.\int dr'' \cdot \frac{n(r'')}{|r'-r''|}\right|_{r'=r} = \int dr'' \frac{n(r'')}{|r-r''|} //$$

$$\therefore \boxed{\frac{\delta A[n(r)]}{\delta n(r)} = \int dr'' \frac{n(r'')}{|r-r''|} //}$$

(b) $A[n(r)] = \int dr' |\vec{\nabla} n(r')|^2$

(b) $$A[n(r)] = \int dr' \overbrace{|\vec{\nabla} n(r')|^2}^{a = a\left(\frac{dn(r')}{dr'}\right)} = \int dr' \left| \frac{\partial n(r')}{\partial x'}\hat{i} + \frac{\partial n(r')}{\partial y'}\hat{j} + \frac{\partial n(r')}{\partial z'}\hat{k} \right| = \int dr' \underbrace{\left(\left(\frac{\partial n(r')}{\partial x'}\right)^2 + \left(\frac{\partial n(r')}{\partial y'}\right)^2 + \left(\frac{\partial n(r')}{\partial z'}\right)^2 \right)}_{a = a\left(\frac{dn(r')}{dr'}\right)}$$

$$= \frac{\partial a}{\partial n}\bigg|_{r'=r} - \frac{d}{dr'}\left(\frac{\partial a}{\partial\left(\frac{dn(r')}{dr'}\right)}\right)\bigg|_{r'=r}$$

in 3D $$= \frac{\partial a}{\partial n}\bigg|_{r'=r} - \vec{\nabla}' \cdot \frac{\partial a}{\partial(\vec{\nabla}' n(r'))}\bigg|_{r'=r}$$

$$\Rightarrow \frac{\delta A[n(r)]}{\delta n(r)} = -\vec{\nabla}' \cdot \frac{\partial}{\partial(\vec{\nabla}'(n(r')))}\left[\vec{\nabla}' n(r') \cdot \vec{\nabla}' n(r')\right]$$

$$\boxed{\frac{\partial f}{\partial \vec{\nabla} g} = \frac{\partial f}{\partial g_x}\hat{i} + \frac{\partial f}{\partial g_y}\hat{j} + \frac{\partial f}{\partial g_z}\hat{k}}$$

$$= -\vec{\nabla}' \cdot \left[\frac{\partial}{\partial\left(\frac{\partial n}{\partial x'}\right)}\left(\frac{\partial n}{\partial x'}\right)^2 \hat{i} + \frac{\partial}{\partial\left(\frac{\partial n}{\partial y'}\right)}\left(\frac{\partial n}{\partial y'}\right)^2 + \frac{\partial}{\partial\left(\frac{\partial n}{\partial z'}\right)}\left(\frac{\partial n}{\partial z'}\right)^2\right]$$

$$g_x = \frac{\partial g}{\partial x}$$

$$= -\vec{\nabla}' \cdot \left[2\frac{\partial n}{\partial x'}\hat{i} + 2\frac{\partial n}{\partial y'}\hat{j} + 2\frac{\partial n}{\partial z'} \right.$$

$$= -\vec{\nabla}' \cdot \left[2\vec{\nabla} n(r')\right]\bigg|_{r'=r} = -2\,\vec{\nabla}' \cdot \left(\vec{\nabla}' n(r')\right)\bigg|_{r'=r} = -2\nabla^2 n(r)$$

$$\boxed{\therefore \frac{\delta A[n(r)]}{\delta n(r)} = -2\nabla^2 n(r)}$$

3 Problem 3

In class we discussed an algorithm for solving eigenvalue problems. In this question, you'll implement that algorithm. You may use any language you choose, you do not need to use all three languages. Your code should solve the problem $x''(t) = Ef(x)$ with the boundary conditions $x(0) = x(\pi) = 0$ by:

(a) **Starting with with $E = 0.01$.**

(b) **Use rk4 to solve the initial value problem $x''(t) = Ef(x)$ with $x(0) = 0$ and $x'(0) = 1$. Set $b(E) = x(\pi)$ from this numerical calculation.**

(c) **Use rk4 to solve the initial value problem $x''(t) = (E + dE)f(x)$ with $x(0) = 0$ and $x0(0) = 1$. Set $b(E + dE) = x(\pi)$ from this numerical calculation, and define $b'(E) = (b(E + dE) - b(E))/dE$.**

(d) **Use Newton's method to update the eigenvalue, $E = E - b(E)/b'(E)$.**

(e) **Return to step (b), unless b(E) is less than some tolerance. You are free to choose the tolerance, dE, and dt (the latter for the timestep in rk4). Hint:** Make sure you integrate all the way to $t = \pi$ in your implementation of rk4.

The code to implement this algorithm is shown below and analyzed below. Firstly, the parameters for the system were initialized:

```
clear;
E=0.01;
dE=1.0e-2;
x0=0;
v0=1;
Y0=[x0;v0];
dt=0.000001;
tol=1.0e-8;
options = odeset('RelTol', 1.0e-8, 'AbsTol', 1.0E-8);
iter=1;
```

Recall that, using this algorithm, not all solutions satisfy the BC's. Hence, we must loop around solutions of the system until we reach one that satisfies the BC's within a certain tolerance. This corresponds to selecting different values for the energy E, and checking if the BC's is satisfied for that E.

```
while(1)
    tspan = 0:dt:pi;
    iter=iter+1
    [t,y] = ode45(@(t,y) odehw7(t,y,E), tspan, Y0, options);

    bE=y(length(t),1)

    [tdE,ydE] = ode45(@(t,y) odehw7(t,y,E+dE), tspan, Y0, options);

    bEdE=ydE(length(tdE),1);
    bpE=(bEdE-bE)/dE
    E=E-bE/bpE
    if (abs(bE)<tol)
        break;
    end
end
deriv=diff(y(:,2))./diff(t);
deriv(length(diff(y(:,2)))+1)=diff(y(1:2,2))./diff(t(1:2));
f1=figure;
hold on
index=1:10000:length(t);
```

```
tplot=t(index);
xplot=y(index,1);
vplot=y(index,2);
derivplot=deriv(index);
plot ( tplot, xplot, 'k.');
plot ( tplot, vplot, 'b.');
plot ( tplot, derivplot, 'g.');
plot ( tplot, 1, 'r.');
plot ( tplot, -1, 'r.');
legend('pos','vel','acc')
title ( 'f1 Problem 3 HW7' )
saveas(gcf,'HW7','epsc');
hold off
```

We can then use the following known problem in order to test the precision of our implementation. The expected allowed energies for the system: $x''(t) = -E * x(t)$ are proportional to n^2. Since we non-dimensionalized the system, the allowed energies must satisfy $E_n = n^2$.

- $x''(t) = -E * x(t)$, with $x(0) = x(\pi) = 0$. You should find an eigenvalue satisfying $E \approx n^2$, with n an integer,such that $n > 0$.
  After running the program $HW7\_3.m$, the output energy was $E \approx 1$ with a precision of 10^{-7}. This precision can be directly controlled by varying the timestep dt. The smaller dt, the more precise our BC for $x(t)$ at $t = \pi$ will become. Figure 1 shows the result for the position, velocity and acceleration, confirming the expected result (see Appendix). When the program converges, the solution for the energy is close to $E = 1$, so the amplitude that we see, is very close to $\frac{1}{\sqrt{E}} \approx 1$.

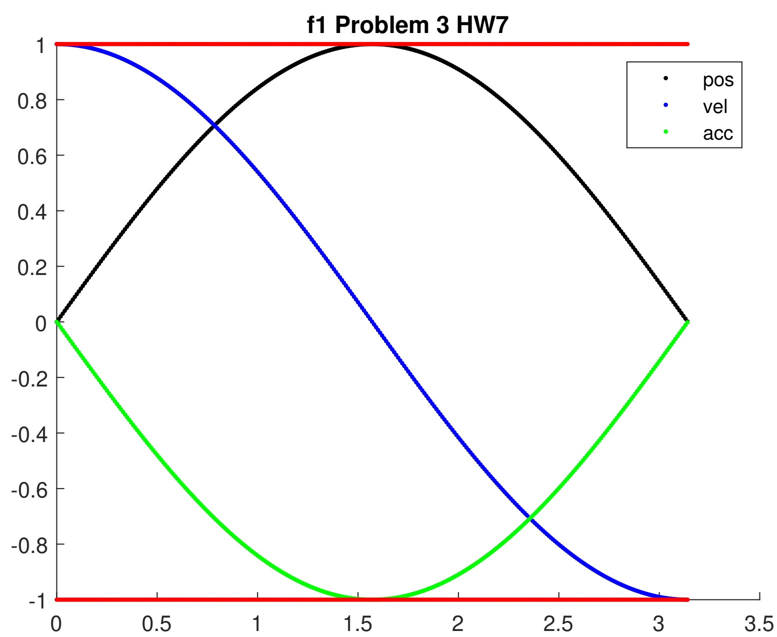


Figure 1: Solutions for $x''(t) = -E * x(t)$ with the specified BCs.

- $x''(t) = -Ex^{1/3}(t)$, with $x(0) = x(\pi) = 0$. Note: $E = 1$ is not an eigenvalue for this boundary value problem.

  After solving the previous problem and confirming the precision of our implementation, we can start to analyze related problems with a bit more confidence. The results for this system are shown in figure 2.

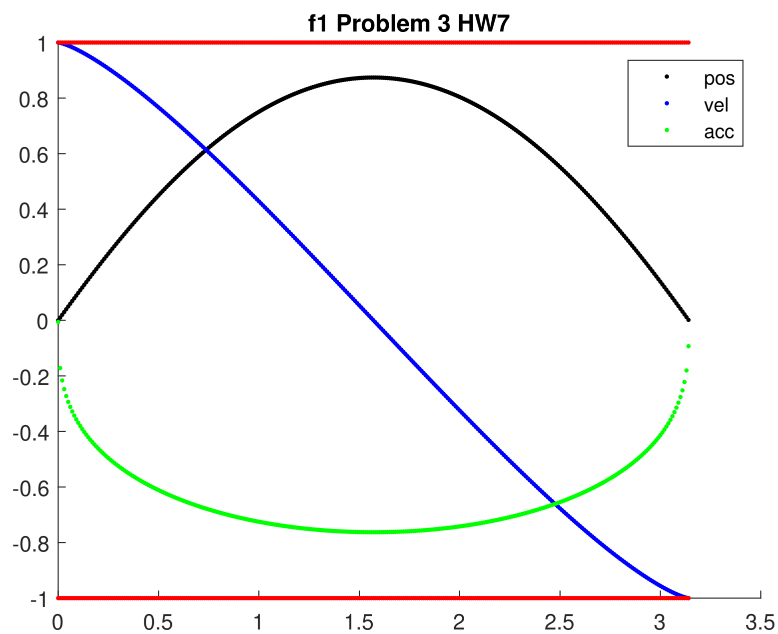


Figure 2: Solutions for $x''(t) = -E * x^{1/3}(t)$ with the specified BCs.

4 Appendix

$$x''(t) = E \cdot f(x)$$

$$\dot{x} = v \qquad x(0) = 0$$
$$\dot{v} = \ddot{x} = E f(x) \qquad \dot{x}(0) = 1$$
$$\Rightarrow \begin{pmatrix} v \\ \dot{v} \end{pmatrix} = \begin{pmatrix} \dot{x} \\ E f(x) \end{pmatrix} \quad \text{with} \quad b(E) = x(\pi)$$

$$\left.\begin{aligned} x'' &= -E x \\ \frac{d v(t)}{dt} &= -E\, x(t) \end{aligned}\right\} \Rightarrow \frac{dv}{dx} \cdot \frac{dx}{dt} = -E\, x(t)$$

$$\int v\, dv = -E x\, dx$$

$$\left.\frac{v^2}{\cancel{2}}\right|_{v_0}^{v} = -E \left.\frac{x^2}{\cancel{2}}\right|_{x_0}^{x}$$

$$v^2 - v_0^2 = -E(x^2 - x_0^2)$$

$$v^2 = v_0^2 - E(x^2 - x_0^2)$$

$$v^2 = 1 - E(x^2)$$

$$\Rightarrow v = \sqrt{1 - E x^2}$$

$$\frac{dx}{dt} = \sqrt{1 - E x^2}$$

$$\int \frac{dx}{\sqrt{1 - E x^2}} = \int dt$$

$$\frac{\arcsin(\sqrt{E}\, x)}{\sqrt{E}} = (t)$$

$$\arcsin(\sqrt{E}\, x) = \sqrt{E}\, t$$

$$\sqrt{E}\, x = \sin(\sqrt{E}\, t)$$

$$x = \frac{1}{\sqrt{E}} \sin(\sqrt{E}\, t)$$

$$\Rightarrow \text{amplitude } \frac{1}{\sqrt{E}}$$

Check:

$$x' = \frac{\sqrt{E}}{\sqrt{E}} \cos(\sqrt{E}\, t) = \cos(\sqrt{E}\, t) \;/\!/ \;\checkmark$$

$$x'' = -\sqrt{E} \sin(\sqrt{E}\, t)$$

$$x'' = -E x \quad \checkmark$$