Math 151BH - HW4

Will Firmin

May 29, 2023

Problem 1:

(a)

$$\begin{aligned}
c_k &= \int_{-\pi}^{\pi} \cos(mx) \frac{1}{\sqrt{2\pi}} e^{-ikx} \\
&= \frac{1}{m} \sin(mx) \frac{1}{\sqrt{2\pi}} e^{-ikx}|_{-\pi}^{\pi} - \frac{ik}{m\sqrt{2\pi}} \int_{-\pi}^{\pi} \sin(mx) e^{-ikx} dx \\
&= \frac{-ik}{m\sqrt{2\pi}} \left[-\cos(mx) e^{ikx} \frac{1}{m}|_{-\pi}^{\pi} + \frac{ik}{m} \int_{-\pi}^{\pi} \cos(mx) e^{-ikx} dx \right] \\
&= \frac{-ik}{m^2\sqrt{2\pi}} [-\cos(m\pi)\cos(k\pi) + \cos(-m\pi)\cos(-k\pi)] + \left(\frac{k}{m}\right)^2 c_k \\
&= \left(\frac{k}{m}\right)^2 c_k
\end{aligned}$$

$$\Rightarrow c_k = 0 \text{ unless } \left(\frac{k}{m}\right)^2 = 1 \iff |k| = m$$

$$\begin{aligned}
\cos(mx) &= \frac{1}{2}(\cos(mx) + i\sin(mx)) + \frac{1}{2}(\cos(mx) - i\sin(mx)) \\
&= \frac{1}{2}(\cos(mx) + i\sin(mx)) + \frac{1}{2}(\cos(-mx) + i\sin(-mx)) \\
&= \frac{1}{2}e^{imx} + \frac{1}{2}e^{-imx}
\end{aligned}$$

$$\Rightarrow c_k = \begin{cases} \frac{\sqrt{\pi}}{\sqrt{2}} & : |k| = m \\ 0 & : \text{else} \end{cases}$$

(b)

$$\begin{aligned}
\sin(mx) &= \frac{-i}{2}(\cos(mx) + i\sin(mx)) - \frac{i}{2}(-\cos(mx) + i\sin(mx)) \\
&= \frac{-i}{2}e^{imx} - \frac{i}{2}(-\cos(-mx) - i\sin(-mx)) \\
&= \frac{-i}{2}e^{imx} + \frac{i}{2}(\cos(-mx) + i\sin(-mx)) \\
&= \frac{-i}{2}e^{imx} + \frac{i}{2}e^{-imx}
\end{aligned}$$

$$\Rightarrow c_k = \begin{cases} \frac{-i\sqrt{\pi}}{\sqrt{2}} & : k = m \\ \frac{i\sqrt{\pi}}{\sqrt{2}} & : k = -m \\ 0 & : \text{else} \end{cases}$$

Problem 2:

(a) Let $\hat{c}_k$ denote the k^{th} Fourier coefficient of $\mathcal{F}(\tau_\mu f)$.

$$\begin{aligned}
\hat{c}_k &= \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} f(x-\mu)e^{-ikx}dx \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\pi-\mu}^{\pi-\mu} f(x)e^{-ik(x+\mu)}dx \\
&= \frac{1}{\sqrt{2\pi}} e^{ik\mu} \int_{-\pi-\mu}^{\pi-\mu} f(x)e^{-ikx}dx \\
&= \frac{1}{\sqrt{2\pi}} e^{-ik\mu} \int_{-\pi}^{\pi} f(x)e^{-ikx}dx \\
&= e^{-ik\mu}c_k
\end{aligned}$$

(b)

$$\begin{aligned}
\overline{c_{-k}} &= \overline{\frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} f(x)e^{ikx}dx} \\
&= \frac{1}{\sqrt{2\pi}} \overline{\int_{-\pi}^{\pi} f(x)e^{ikx}dx} \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} \overline{f(x)e^{ikx}}dx \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} f(x)\overline{e^{ikx}}dx \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} f(x)e^{-ikx}dx \\
&= c_k
\end{aligned}$$

Problem 3:

(a)

$$\begin{aligned}
< \varphi_1, L\varphi_2 > &= \int_{-\pi}^{\pi} \varphi_1 \overline{L\varphi_2} dx \\
&= \int_{-\pi}^{\pi} \varphi_1 \sum_k \frac{k^2 c_{2,k}}{\sqrt{2\pi}} e^{-ikx} dx \\
&= \sum_k \frac{k^2 c_{2,k}}{\sqrt{2\pi}} \int_{\pi}^{\pi} \varphi_1 e^{-ikx} dx \\
&= \sum_k k^2 c_{2,k} c_{1,k} \qquad (*) \\
&= \sum_k \frac{k^2 c_{1,k}}{\sqrt{2\pi}} \int_{-\pi}^{\pi} \varphi_2 e^{-ikx} dx \\
&= \int_{-\pi}^{\pi} \varphi_2 \sum_k \frac{k^2 c_{1,k}}{\sqrt{2\pi}} e^{-ikx} dx \\
&= \int_{-pi}^{\pi} \varphi_2 \overline{L\varphi_1} dx \\
&= < \varphi_1, L\varphi 1 >
\end{aligned}$$

The above shows symmetry, and for $\varphi = \varphi_1 = \varphi_2$, $(*)$ gives

$$< \varphi, L\varphi > = \sum_k k^2 c_k^2 > 0$$

For $\varphi \neq 0$ and $c_0 = 0$

(b) Clearly $e^{ikx} = \sqrt{2\pi} q_k(x) \Rightarrow c_k = \sqrt{2\pi}$ with all other coefficients being 0. Then

$$\begin{aligned}
L\varphi &= \sum_j \frac{j^2 c_j}{\sqrt{2\pi}} e^{ijx} \\
&= \frac{k^2 \sqrt{2\pi}}{\sqrt{2\pi}} e^{ikx} \\
&= k^2 e^{ikx} \\
&= k^2 \varphi \\
\Rightarrow \lambda &= k^2
\end{aligned}$$

Problem 4: Let $\hat{\varphi} = \varphi - \varphi_{avg}$. Then $\hat{\varphi}_{avg} = 0$ and $\hat{\varphi}' = \varphi'$.

$$\begin{aligned}
<\hat{\varphi}, \hat{\varphi}> &= \int_{-\pi}^{\pi} \hat{\varphi}\overline{\hat{\varphi}} dx \\
&= \int_{\pi}^{\pi} \hat{\varphi} \sum_k \frac{c_k}{\sqrt{2\pi}} e^{-ikx} dx \\
&= \sum_k \frac{c_k}{\sqrt{2\pi}} \int_{-\pi}^{\pi} \hat{\varphi} e^{-ikx} dx \\
&= \sum_k c_k^2 \\
<\varphi', \varphi'> &= <\hat{\varphi}', \hat{\varphi}'> \\
&= \int_{-\pi}^{\pi} \hat{\varphi}'\overline{\hat{\varphi}'} dx \\
&= \int_{-\pi}^{\pi} \hat{\varphi}' \sum_k \frac{-ikc_k}{\sqrt{2\pi}} e^{-ikx} dx \\
&= \sum_k \frac{-ikc_k}{\sqrt{2\pi}} \int_{-\pi}^{\pi} \hat{\varphi}' e^{-ikx} dx \\
&= \sum_k -ikc_k c_k' \\
&= \sum_k -ikc_k (ikc_k) \\
&= \sum_k k^2 c_k^2 \\
\Rightarrow \|\hat{\varphi}\| = \left(\sum_k c_k^2\right)^{\frac{1}{2}} &\leq \left(\sum_k k^2 c_k^2\right)^{\frac{1}{2}} = \|\varphi'\|
\end{aligned}$$

Since $k^2 \geq 1 \; \forall k \neq 0$ and $c_0 = 0$.

Problem 5:

$$
\begin{aligned}
\vec{c} &= F\vec{x} \\
\Rightarrow c_k &= \sum_{j=1}^{N} F_{k,j} x_j \\
&= \sum_{j=1}^{N} (e^{-2\pi i/N})^{kj} x_j \\
\Rightarrow \overline{c_k} &= \sum_{j=1}^{N} (e^{2\pi i/N})^{kj} x_j \\
&= \sum_{j=1}^{N} e^{2\pi i} (e^{2\pi i/N})^{kj} \\
&= \sum_{j=1}^{N} e^{-2\pi ij} (e^{2\pi i/N})^{kj} \\
&= \sum_{j=1}^{N} (e^{-2\pi i/N})^{Nj} (e^{-2\pi i/N})^{-kj} \\
&= \sum_{j=1}^{N} (e^{-2\pi i/N})^{(N-k)j} \\
&= c_{N-k}
\end{aligned}
$$

Problem 6: Let $v = y'$, then $v' = y''$.

$$
v'(t) + a(t)v(t) + b(t)y(t) = f(t) \Rightarrow v'(t) = f(t) - a(t)v(t) - b(t)y(t)
$$

Then the system is

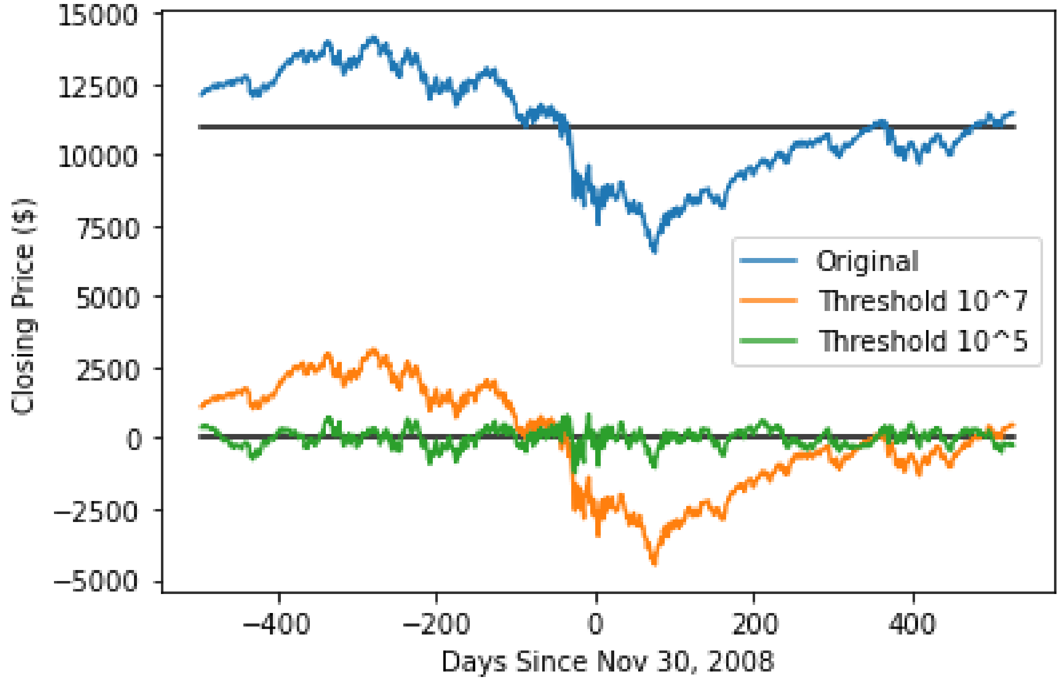
$$
\begin{aligned}
v'(t) &= f(t) - a(t)v(t) - b(t)y(t) \\
y'(t) &= v(t) \\
y(t_0) &= c_1 \\
v(t_0) &= c_2
\end{aligned}
$$

Problem 7:

$$
\begin{aligned}
\frac{dw}{dx} &= \frac{w'(t)}{x'(t)} \\
&= \frac{-x(t)}{x'(t)} \\
&= \frac{-x(t)}{w(t)} \\
\Rightarrow w(t)dw &= -x(t)dx \\
\Rightarrow \int w(t)dw &= -\int x(t)dx \\
\Rightarrow \frac{1}{2}w(t)^2 &= -\frac{1}{2}x(t)^2 + C_0 \\
\Rightarrow w(t)^2 + x(t)^2 &= 2C_0
\end{aligned}
$$

Problem 8:

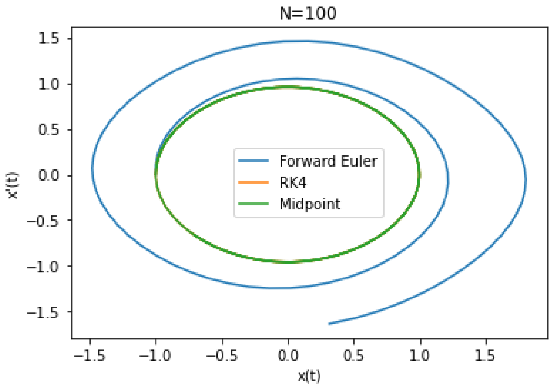
```
import numpy as np
from scipy.fft import rfft, irfft
import matplotlib.pyplot as plt
import pandas


file = open("dow.txt", "r")
dow = file.read()
dow = dow.replace('\n', ' ').split(" ")[:-1]
dow = np.array(list(map(float,dow)))
file.close()

# rfft is designed for real input and only gives first half of coefficients
y = rfft(dow)
ynorm = np.abs(y)

y1 = y
y1[ynorm > 10000000] = 0
x1 = irfft(y1)

y2 = y
y2[ynorm > 100000] = 0
x2 = irfft(y2)

days = np.arange(1024)-496

plt.plot(days,np.zeros(1024),color="k")
plt.plot(days,np.ones(1024)*np.mean(dow),color="k")
plt.plot(days,dow, label="Original")
plt.plot(days,x1, label="Threshold 10^7")
plt.plot(days,x2, label="Threshold 10^5")
plt.xlabel("Days Since Nov 30, 2008")
plt.ylabel("Closing Price ($)")
plt.legend()
plt.show()
```

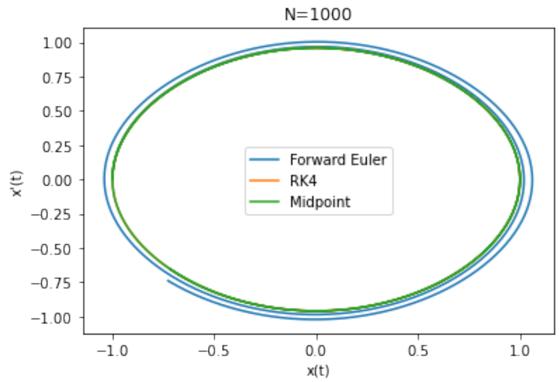


Problem 9:

```
import math

x0 = -1
v0 = 0

def ForwardEuler(N):
    w = [x0]
    v = [v0]
    h = 4*np.pi/N
    for i in range(N):
        w.append(w[-1]+h*v[-1])
        v.append(v[-1]+h*(-math.sin(w[-2])))
    return (w,v)

def RK4(N):
    w = [x0]
    v = [v0]
    h = 4*np.pi/N
    for i in range(N):
        j1 = h*(-math.sin(w[-1]))
        k1 = h*v[-1]
        j2 = h*(-math.sin(w[-1]+k1/2))
        k2 = h*(v[-1]+j1/2)
        j3 = h*(-math.sin(w[-1]+k2/2))
        k3 = h*(v[-1]+j2/2)
        j4 = h*(-math.sin(w[-1]+k3))
        k4 = h*(v[-1]+j3)
        v.append(v[-1]+(j1+2*j2+2*j3+j4)/6)
        w.append(w[-1]+(k1+2*k2+2*k3+k4)/6)
    return (w,v)
```

```
iterF = lambda x, xnew, h: x + h*np.array([-math.sin()])
def Midpoint(N):
    w = [x0]
    v = [v0]
    h = 4*np.pi/N
    for i in range(N):
        wold, vold = 0, 0
        wnew, vnew = w[-1], v[-1]
        start = False
        while np.abs([vold-vnew,wold-wnew]).max() >= 10**-10 or start==False:
            wold, vold = wnew, vnew
            vnew = v[-1] + h*(-math.sin((w[-1]+wold)/2))
            wnew = w[-1] + h*(v[-1]+vold)/2
            start = True
        w.append(wnew)
        v.append(vnew)
    return (w,v)


Euler1 = ForwardEuler(100)
RK1 = RK4(100)
Midpoint1 = Midpoint(100)
plt.plot(Euler1[0], Euler1[1], label="Forward Euler")
plt.plot(RK1[0], RK1[1], label="RK4")
plt.plot(Midpoint1[0], Midpoint1[1], label="Midpoint")
plt.xlabel("x(t)")
plt.ylabel("x'(t)")
plt.title("N=100")
plt.legend()
plt.show()

Euler2 = ForwardEuler(1000)
RK2 = RK4(1000)
Midpoint2 = Midpoint(1000)
plt.plot(Euler2[0], Euler2[1], label="Forward Euler")
plt.plot(RK2[0], RK2[1], label="RK4")
plt.plot(Midpoint2[0], Midpoint2[1], label="Midpoint")
plt.xlabel("x(t)")
plt.ylabel("x'(t)")
plt.title("N=1000")
plt.legend()
plt.show()
```





The results confirm what was found in question 7, in that the phase plot should be a circle. As N increases, the plots become more circular. The Forward Euler method is clearly the least accurate of the three, as it appears to spiral while the other two are indistinguishable at this point from a circle. The RK4 and Midpoint methods are difficult to compare since they have a high enough accuracy to be plotted over each other.

Problem 10:

```
d,a,b,W=0.3,-1,1,1.2
hmin,hmax=10**-3,0.5
qmin,qmax=0.1,4
tol=10**-5
x0,v0=1,0.2

def ATS(gamma, tend):
    f = lambda x,v,t: gamma*math.cos(W*t)-d*v-a*x-b*x**3
    h = 0.1
    H = [h] #need to remove at end
    Ts = [0]
    w = [x0]
    v = [v0]
    while Ts[-1] < tend:
        h = min(h, tend-Ts[-1])
        while True:
            y1 = w[-1]+h/2*v[-1]
            j1 = v[-1]+h/2*f(w[-1],v[-1],Ts[-1])
            y2 = w[-1]+h*j1
            j2 = v[-1]+h*f(y1,j1,Ts[-1]+h/2)
            eps = np.abs([y2-y1,j2-j1]).max()/h
            if eps < tol or h == hmin:
                w.append(y2)
                v.append(j2)
                H.append(h)
                Ts.append(Ts[-1]+h)
                h = min(h*2,hmax)
                break
            else:
                q=tol/(2*eps)
                q = min(max(q,qmin),qmax)
                h = q*h
                h = min(max(h,hmin),hmax)
    return np.array(w),np.array(v),np.array(Ts),np.array(H)
```

```
w,v,Ts,H = ATS(0.28,200)
plt.plot(w[Ts>=100], v[Ts>=100])
plt.title("Gamma=0.28, T_end=200")
plt.xlabel("x(t)")
plt.ylabel("x'(t)")
plt.show()

w,v,Ts,H = ATS(0.37,200)
plt.plot(w[Ts>=100], v[Ts>=100])
plt.title("Gamma=0.37, T_end=200")
plt.xlabel("x(t)")
plt.ylabel("x'(t)")
plt.show()

w,v,Ts,H = ATS(0.5,500)
plt.plot(w, v)
plt.title("Gamma=0.5, T_end=500")
plt.xlabel("x(t)")
plt.ylabel("x'(t)")
plt.show()

plt.plot(Ts,np.log(H))
plt.title("Step vs Time")
plt.xlabel("t")
plt.ylabel("h")
plt.show()
```