# AC274 - Homework #4

Matheus C. Fernandes

October 13, 2015

## Problem 1: Write a third order accurate finite-difference expression of $\frac{df}{dx}$

*From: Lecture 6*

To obtain a third order accurate finite-difference scheme (forward difference stencil), we know that we must cancel out the third term of the taylor series expansion and thus we can use the anzats for our scheme that:

$$f'(x) = af(x) + bf(x+h) + cf(x+2h) + df(x+3h) \tag{1}$$

We therefore taylor expand each term in our anzats:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + O(h^4)$$

$$f(x+2h) = f(x) + 2hf'(x) + \frac{4h^2}{2!}f''(x) + \frac{8h^3}{3!}f'''(x) + O(h^4)$$

$$f(x+3h) = f(x) + 3hf'(x) + \frac{9h^2}{2!}f''(x) + \frac{27h^3}{3!}f'''(x) + O(h^4)$$

Plugging these in to our anzats eq. (1) and solving for like terms we obtain:

$$f'(x) = (a+b+c+d)f(x) + (b+2c+3d)hf'(x) + (b+4c+9d)\frac{h^2}{2!}f''(x) + (b+8c+27d)\frac{h^3}{3!}f'''(x) + O(h^4).$$

Solving for the individual terms such that they are all set equal to zero except for the term we want to keep, namely $(b+2c+3d)h = 1$.

$$\left.\begin{array}{rcl} a+b+c+d &=& 0 \\ b+2c+3d &=& \frac{1}{h} \\ b+4c+9d &=& 0 \\ b+8c+27d &=& 0 \end{array}\right\} \begin{array}{rcl} a &=& -\frac{11}{6h} \\ b &=& \frac{3}{h} \\ c &=& -\frac{3}{2h} \\ d &=& \frac{1}{3h} \end{array}$$

Therefore the scheme is:

$$f'(x) = \frac{-\frac{11}{6}f(x) + 3f(x+h) - \frac{3}{2}f(x+2h) + \frac{1}{3}f(x+3h)}{h} + O(h^3)$$

The third order accuracy comes from plugging in the taylor series and seeing that we divide through by h and thus the $O(h^4)$ becomes $O(h^3)$ (this part omitted from write up as it takes a lot of typing, furthermore we based our derivation on solving the taylor series up to third order, so a proof of accuracy order is not needed).

## Problem 2: Write a fourth order accurate $\frac{d^2f}{dx^2}$

*From: Lecture 6*

Lets now use the center difference scheme to shake things up a bit. Also, we can save some work knowing that we get an extra order of accuracy from the center difference, therefore we can get rid of some algebra, thus let our anzats be:

$$f''(x) = af(x-2h) + bf(x-h) + cf(x) + df(x+h) + ef(x+2h) \tag{2}$$

Therefore using taylor expansion we get:

$$f(x-2h) = f(x) - 2hf'(x) + \frac{4h^2}{2!}f''(x) - \frac{8h^3}{3!}f'''(x) + \frac{16h^4}{4!}f^{(4)}(x) + O(h^5)$$

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(x) + O(h^5)$$

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(x) + O(h^5)$$

$$f(x+2h) = f(x) + 2hf'(x) + \frac{4h^2}{2!}f''(x) + \frac{8h^3}{3!}f'''(x) + \frac{16h^4}{4!}f^{(4)}(x) + O(h^5)$$

Plugging these in to our anzats eq. (2) and solving for like terms we obtain:

$$\begin{aligned} f''(x) = (a+b+c+d+e)f(x) + (-2a-b+d+2e)hf'(x) + (4a+b+d+4e)\frac{h^2}{2!}f''(x) + \\ (-8a-b+d+8e)\frac{h^3}{3!}f'''(x) + (16a+b+d+16e)\frac{h^4 f^{(4)}(x)}{4!} + O(h^5). \end{aligned}$$

Solving for the individual terms such that they are all set equal to zero except for the term we want to keep, namely $(4a+b+d+4e)\frac{h^2}{2!} = 1$.

$$\left.\begin{aligned} a+b+c+d+e &= 0 \\ -2a-b+d+2e &= 0 \\ 4a+b+d+4e &= \tfrac{2}{h^2} \\ -8a-b+d+8e &= 0 \\ 16a+b+d+16e &= 0 \end{aligned}\right\} \begin{aligned} a &= -\tfrac{1}{12h^2} \\ b &= \tfrac{4}{3h^2} \\ c &= -\tfrac{5}{2h^2} \\ d &= \tfrac{4}{3h^2} \\ e &= -\tfrac{1}{12h^2} \end{aligned}$$

Therefore the scheme is

$$\boxed{f''(x) = \frac{-\frac{1}{12}f(x-2h) + \frac{4}{3}f(x-h) - \frac{5}{2}f(x) + \frac{4}{3}f(x+h) - \frac{1}{12}f(x+2h)}{h^2} + O(h^4)}$$

# Problem3V2

October 13, 2015

## 1 Problem 3 (from lecture 6)

Solve the damped-oscillator equation:

$$\frac{d^2x(t)}{dt^2} + \gamma * \frac{dx(t)}{dt} + \omega^2 * x(t) = 0$$

$$x(0) = 0$$

$$\frac{dx(0)}{dt} = 1.$$

Take $\gamma = 0.1$, $\omega = 1$ and explore the numerical stability by changed the time-step $dt$

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import cmath as math
        %matplotlib inline
        gamma=0.1
        omega=1
        dxdt=1.0
        totaltime=50.0
        dt=0.1
```

The scheme of choice is center difference of O($\mathrm{dt}^3$):

$$x'(t) = \frac{-x_{i-1} + x_{i+1}}{2dt} + O(dt^3)$$

$$x''(t) = \frac{x_{i-1} - 2x_i + x_{i+1}}{dt^2} + O(dt^3)$$

Therefore, we must solve the following equation:

$$\frac{x_{i-1} - 2x_i + x_{i+1}}{dt^2} + \gamma \frac{-x_{i-1} + x_{i+1}}{2dt} + \omega^2 * x_i = 0$$

After derivation of the analytical form we obtain that the analytical solution takes form

$$x(t) = \frac{1}{\lambda_+ - \lambda_-} \left(e^{\lambda_+ t} - e^{\lambda_- t}\right)$$

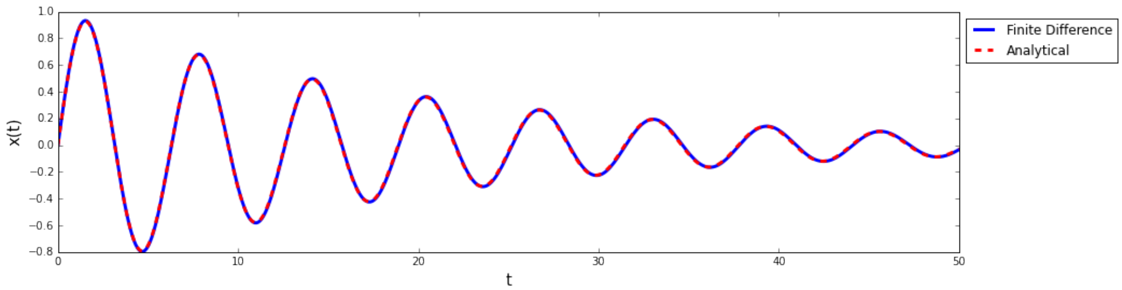
where

$$\lambda_+ = \frac{-\gamma + \sqrt{\gamma^2 - 4\omega^2}}{2}$$

$$\lambda_- = \frac{-\gamma - \sqrt{\gamma^2 - 4\omega^2}}{2}$$

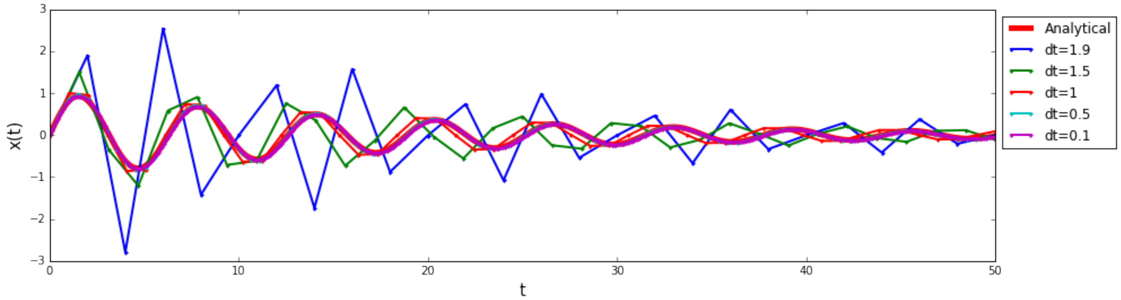
```
In [2]: def FD(totaltime,dt,omega,gamma):
            t=np.linspace(0,totaltime,totaltime/dt)
            x=np.zeros((totaltime/dt))
            x[0]=0
            x[1]=dt # here we impletmetn the boundary conditon of dxdt=1 at x=0
            for i in range(2,len(t)):
                x[i]=-(x[i-2]*(1/dt**2-gamma/(2*dt))+x[i-1]
                    *(-(2/dt**2)+omega**2))*(1/dt**2+gamma/(2*dt))**(-1)
            return t,x
        def analytical(totaltime,dt,omega,gamma):
            tall=np.linspace(0,totaltime,totaltime/dt)
            lambda_p=(-gamma+math.sqrt(gamma**2-4*omega**2))/2
            lambda_m=(-gamma-math.sqrt(gamma**2-4*omega**2))/2
            x=[(1/(lambda_p-lambda_m)*(math.exp(lambda_p*t)-math.exp(lambda_m*t))) for t in tall]
            return tall,x
```

```
In [6]: t,x=FD(totaltime,dt,omega,gamma)
        ta,xa=analytical(totaltime,dt,omega,gamma)
        plt.plot(t,x,linewidth=3,label="Finite Difference")
        plt.plot(ta,xa,'r--',linewidth=3,label="Analytical")
        plt.xlabel('t',fontsize=15)
        plt.ylabel('x(t)',fontsize=15)
        plt.legend(bbox_to_anchor=(1, 1),loc=2)
        plt.gcf().set_size_inches(15, 4)
```



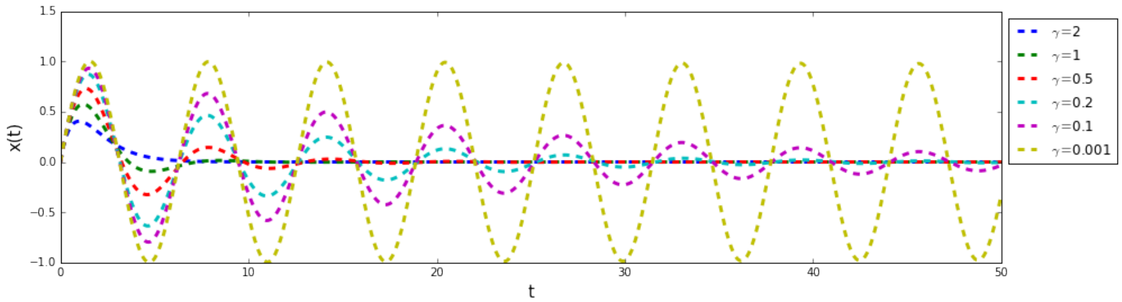

### 1.1 Playing with stability by changing the time-step $dt$

```
In [4]: plt.plot(ta,xa,'r-',linewidth=5,label="Analytical")
        for dtchange in [1.9,1.5,1,0.5,0.1]:
            t,x=FD(totaltime,dtchange,omega,gamma)
            plt.plot(t,x,'.-',linewidth=2,label="dt={}".format(dtchange))
        plt.xlabel('t',fontsize=15)
        plt.ylabel('x(t)',fontsize=15)
        plt.legend(bbox_to_anchor=(1, 1),loc=2)
        plt.gcf().set_size_inches(15, 4)
```



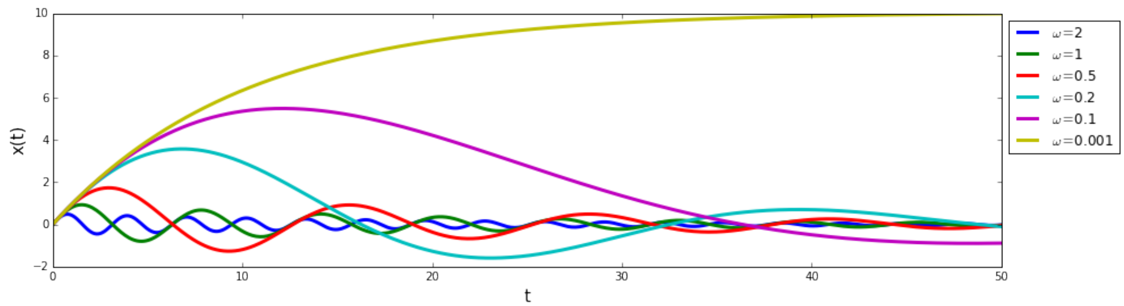

## 1.2 Playing with the $\gamma$ parameter and seeing the system's response to different inputs

```
In [5]: for gammachange in [2,1,0.5,0.2,0.1,0.001]:
            t,x=FD(totaltime,dt,omega,gammachange)
            plt.plot(t,x,'--',linewidth=3,label="$\gamma=${}".format(gammachange))
        plt.xlabel('t',fontsize=15)
        plt.ylabel('x(t)',fontsize=15)
        plt.legend(bbox_to_anchor=(1, 1),loc=2)
        plt.gcf().set_size_inches(15, 4)
```




## 1.3 Playing with the $\omega$ parameter and seeing the system's response to different inputs

```
In [7]: for omegachange in [2,1,0.5,0.2,0.1,0.001]:
            t,x=FD(totaltime,dt,omegachange,gamma)
            plt.plot(t,x,'-',linewidth=3,label="$\omega=${}".format(omegachange))
        plt.xlabel('t',fontsize=15)
        plt.ylabel('x(t)',fontsize=15)
        plt.legend(bbox_to_anchor=(1, 1),loc=2)
        plt.gcf().set_size_inches(15, 4)
```


x(t)
t
10
8
6
4
2
0
−2
0
10
20
30
40
50
ω=2
ω=1
ω=0.5
ω=0.2
ω=0.1
ω=0.001