Programming Assignment

Nathan Tipton, Aaron Yardley,Tarrin Rasmussen

12/5/2016

Contents

1 Problem 1

For the differential equation

$$(D + 3)y_0(t) = 0 \tag{1}$$

with initial condition

$$y_0(0) = 2$$

1.1 a) Problem Statement

Find and plot the analytical (exact) solution to the differential equation for $0 \le t \le 10$.

1.2 a) Solution

Solve $(D+3)\,y_0(t) = 0$ when $y_0(0) = 2$

$\lambda = -3$

$y(t) = Ae^{-3t}$

$y(0) = 2 = A(1)$

$y(t) = 2e^{-3t}$

Figure 1: Analytical solution for differential equation

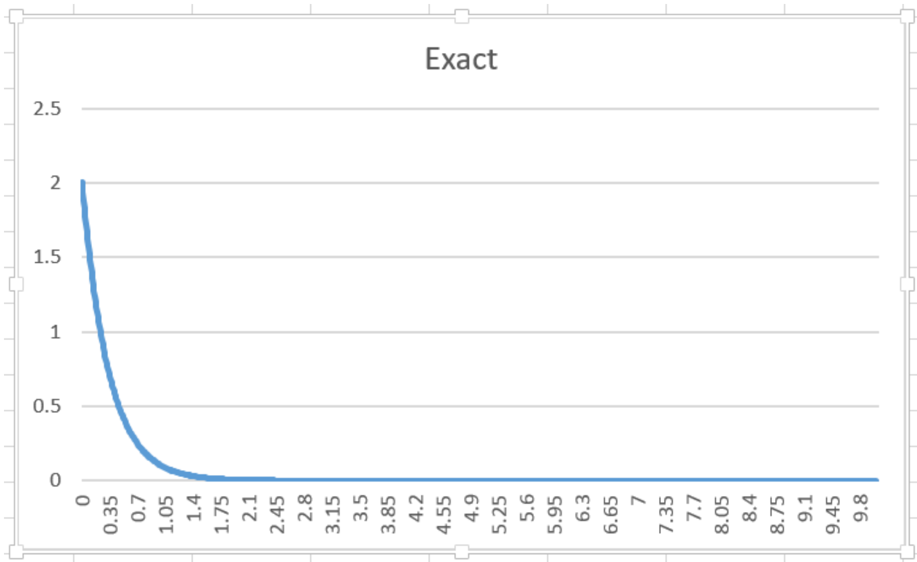


Figure 2: Exact Solution

1.3 b) Problem Statement

Write a program to plot a numerical solution using

$$y_0(t + \Delta t) = (1 + a\Delta t)y_0(t) \tag{2}$$

1.4 b) Solution

See Appendix for Code

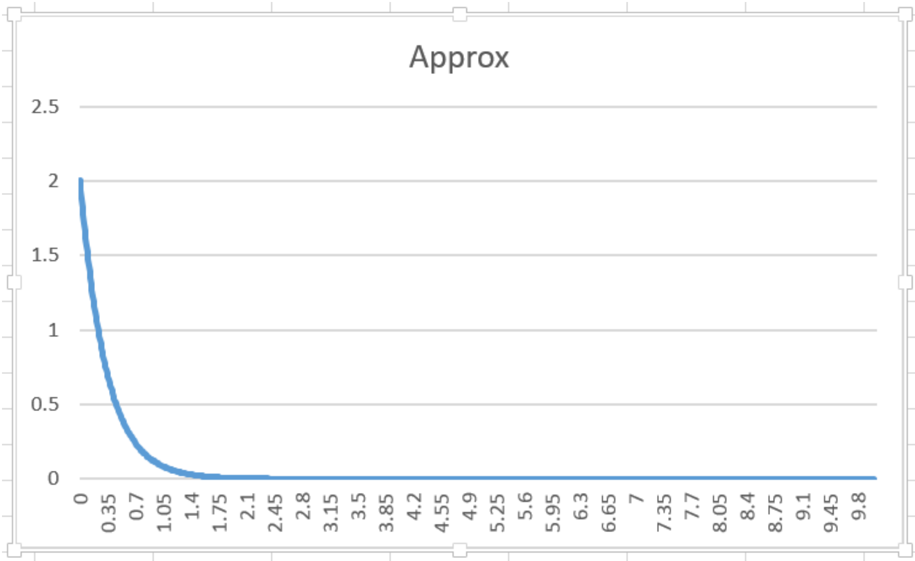


Figure 3: Approximate Solution

1.5 c) Problem Statement

Compare the exact solution with the approximate solution.

1.6 c) Solution

Figure 4 shows the error when comparing the exact and approximate solutions.

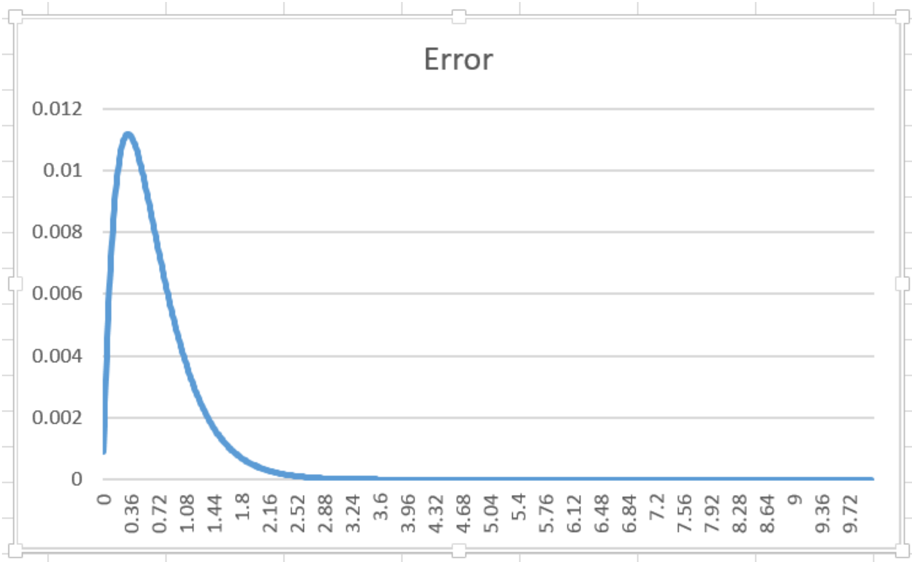


Figure 4: Error

2 Problem 2

For the third-order differential equation

$$(D^3 + 0.45D^2 + 36.06D + 1.802)y_0(t) = 0 \tag{3}$$

with initial conditions

$$y_0(0) = 2, y_0'(0) = 3, y_0''(0) = 1$$

2.1 a) Problem Statement

Find and plot the analytical solution to the differential equation for $0 \leq t \leq 10$. Identify the roots of the characteristic equation and plot them in the complex plane.

2.2 a) Solution

$$(D^3 + .45D^2 + 36.06D + 1.802)\, y_0(t)$$

$$\lambda_1 = -.05 \qquad \lambda_{2,3} = -.2 \pm j6 \qquad y_0(t) = C_1 e^{-.05t} + e^{-.2t}\big(B_1 \cos(6t) + \sin(6t\, B_2)\big)$$

$$2 = C_1 + B_1$$

$$3 = -.05C_1 - .2B_1 + 6B_2$$

$$1 = .0025C_1 + .04B_1 - 36B_1 - (.4 \cdot 6)B_2$$

$$C = \sqrt{B_1^2 + B_2^2}$$

$$\theta = \tan^{-1}\left(\frac{-B_2}{B_1}\right)$$

exact $\underline{y_0(t) = 2.062e^{-.05t} + .5788e^{-.2t}\cos(6t - 96.8634^\circ)}$

2.3 b) Problem Statement

Put the third-order differential equation into state-space form.

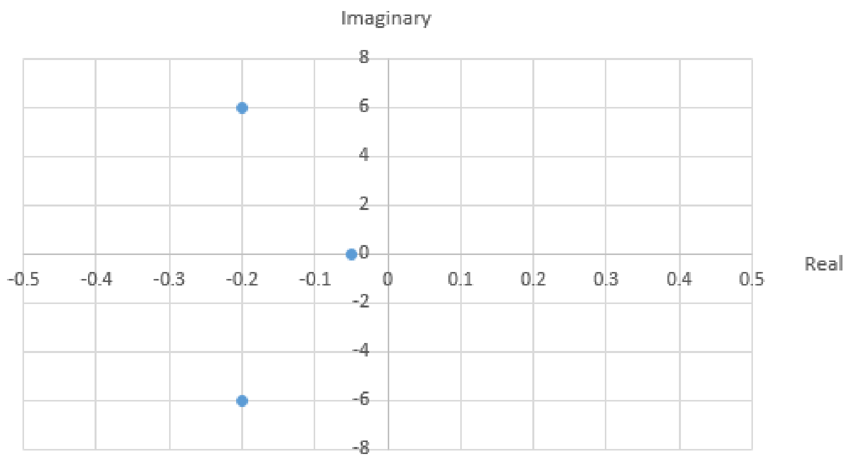


Figure 5: Roots on complex plane: -.05, -.2+6j, -.2-6j

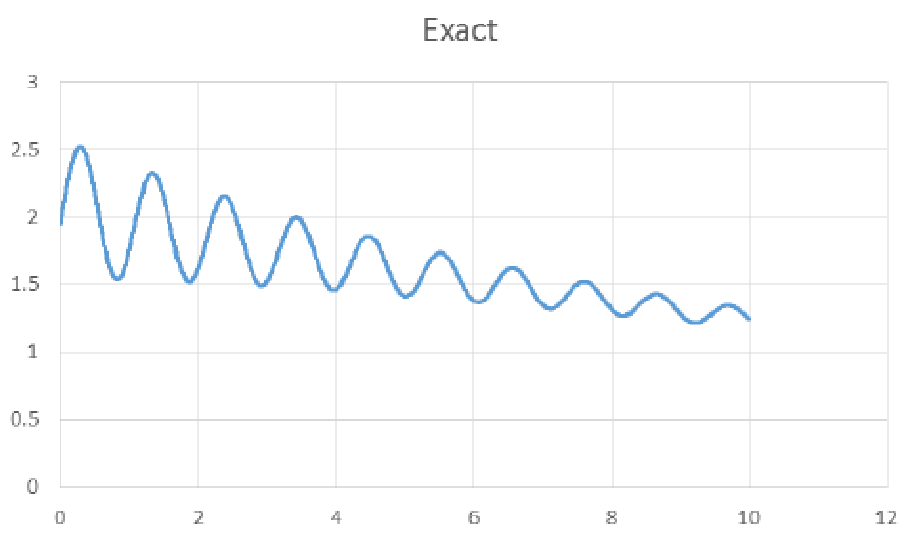


Figure 6: Exact Solution

2.4 b) Solution

$$\dot{x}_3 + .45x_3(t) + 36.06x_2(t) + 1802x_1(t) = 0$$

$$\dot{x}_3 = -.45x_3(t) - 36.06x_2(t) - 1.802x_1(t)$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} f(t)$$

Figure 7: State Space form

2.5 c) Problem Statement

Write a program to plot an approximate solution using

$$x(t + \Delta t) = (I + A\Delta t)x(t) \tag{4}$$

2.6 c) Solution

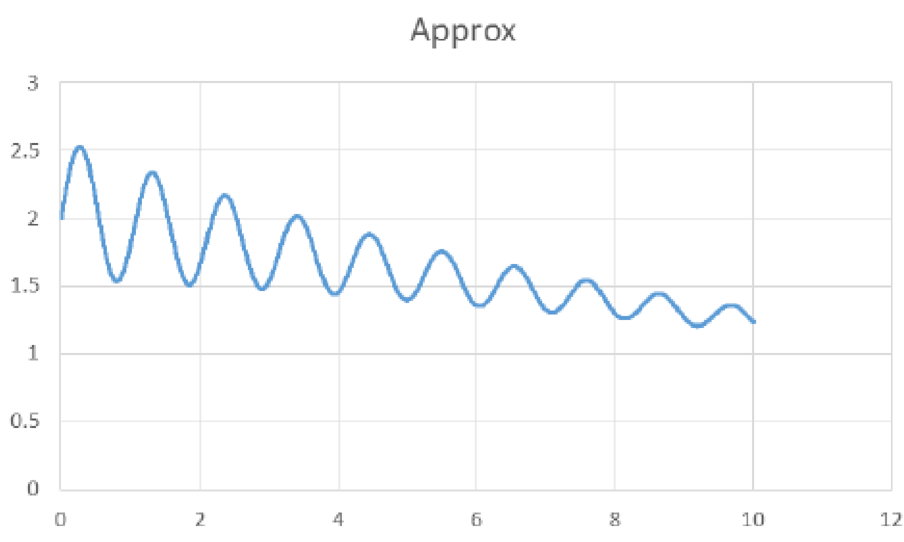


Figure 8: Approximate Solution

See Appendix for Code

2.7 d) Problem Statement

Compare the exact solution with the approximate solution.

2.8 d) Solution

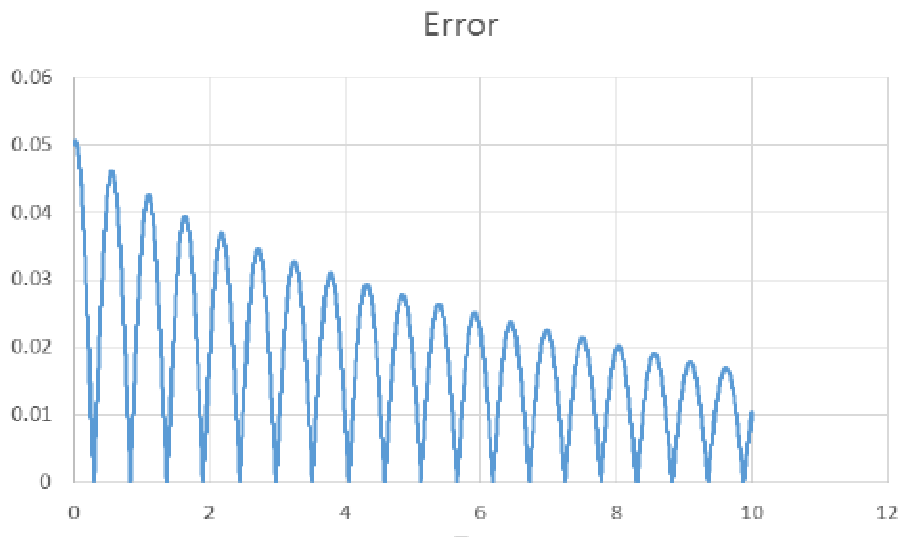


Figure 9: Error

3 Problem 3

For the circuit shown: where $R_1 = 1k\Omega, R_2 = 47k\Omega, C = 100\mu F$ and $L = 5H$

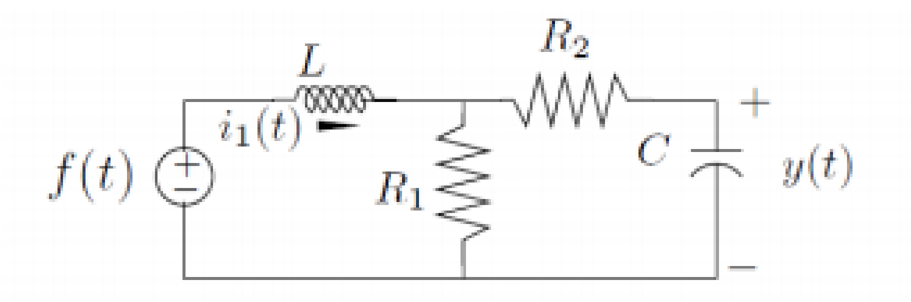


Figure 10: Problem 3 circuit diagram

3.1 a-d) Problem Statements

a)Detemine a differential equation relating the input f(t) to the output y(t).

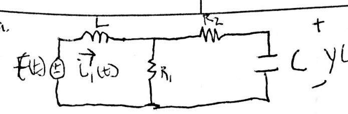
b)Determine the inital conditions on $y(t)$ if $i_1(0) = 0.2A$ and $y(0) = 5V$.

c)Determine the analytical solution for the zero-input response of the system with these initial conditions.

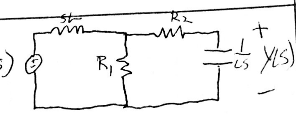
d)Represent the differential equation for the circuit in state variable form.

3.2 a-d) Solutions

3. a.

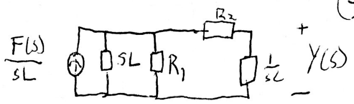


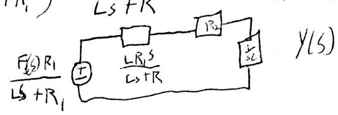
$\Leftrightarrow$



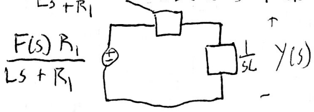
$$\left(\frac{1}{sL}+\frac{1}{R_1}\right)^{-1}=\frac{LR_1 s}{Ls+R}$$

$$\frac{F(s)}{\cancel{sL}}\cdot\frac{\cancel{Ls}R_1}{Ls+R_1}$$





$$\frac{Ls(R_1+R_2)+R_1R_2}{Ls+R_1} \quad \cancel{Ls(R_1+R_2)+R_1}$$



$$Y(s)=\frac{F(s)R_1}{Ls+R_1}\cdot\frac{\frac{1}{sC}}{\frac{1}{sC}+\frac{Ls(R_1+R_2)+R_1R_2}{Ls+R_1}}$$

$$H(s)=\frac{R_1}{sC(Ls+R_1)\left(\frac{1}{sC}+\frac{Ls(R_1+R_2)+R_1R_2}{Ls+R_1}\right)}$$

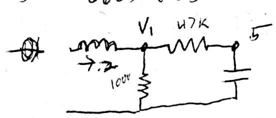
$$(24s^2+4705s+1000)Y(s)=\frac{1000}{24}(F(s))$$

$$(s^2+196.042s+41.6667)Y(s)=41.6667F(s)$$

$$(D^2+196.042D+41.6667)Y(t)=41.6667f(t)$$

b. $y(0)=5V$

$$\dot{y}(0)=C\frac{dV}{dt}=i_c(0)$$



$$.2=\frac{V_1-5}{47k}+\frac{V_1}{1000}$$

$$V_1=195.937$$

$$\dot{y}(0)=\frac{i_c(0)}{C}=\frac{4.06249\,mA}{100\,\mu F}=40.625V \qquad \frac{V_1-5}{47k}=i_2(0)=4mA$$

c.

$$\lambda^2+196.042\lambda+41.6667$$

$$(\lambda+.2128)(\lambda+195.83)=0 \qquad \lambda=-.2128;\ \lambda=-195.83$$

$$y_0(t)=c_1e^{-.2128t}+c_2e^{-195.83t} \qquad 5=c_1+c_2$$

$$y_0'(t)=-.2128c_1e^{-.2128t}-c_2 195.83e^{-195.83t} \qquad 40.625=-.2128c_1-195.83c_2$$

$$c_1=5.213 \qquad c_2=-.2131$$

$$\cancel{c_2=-.2131}$$

$$y_0(t)=5.213e^{-.2128t}-.2131e^{-195.83t}$$

d. $x_1 = y$

$x_2 = \dot{y}$

$\dot{x}_1 = x_2$

$$\dot{x}_2(t) = -196.042\, x_2(t) - 41.6667\, x_1(t)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -196.04 & -41.6667 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} E(t)$$

3.3 e) Problem Statement

Using your program, determine a numerical solution to the differential equation for the zero-input response.

3.4 e) Solution

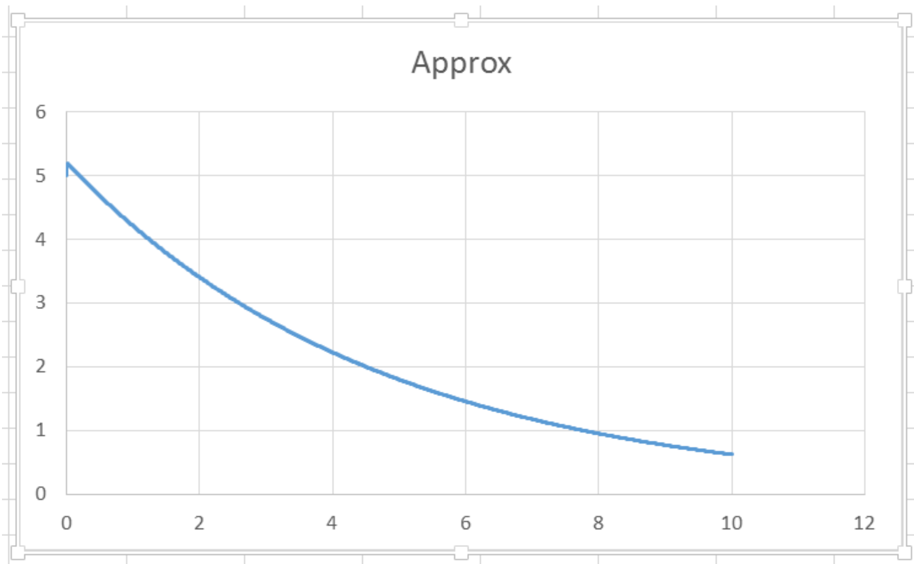


Figure 11: Numerical Solution

3.5 f) Problem Statement

Plot and compare the analytical and the numerical solution. Comment on your results.

3.6 f) Solution

Lower order differential equations resulted in smaller error when comparing the numerical and analytical solutions. The circuit is overdamped.

3.7 g) Problem Statement

Suppose that the circuit had nonlinear elements in it, such as dependent sources. Describe how the analytical solution and numerical solution would change.

3.8 g) Solution

A nonlinear element would raise the order of the differential equation.

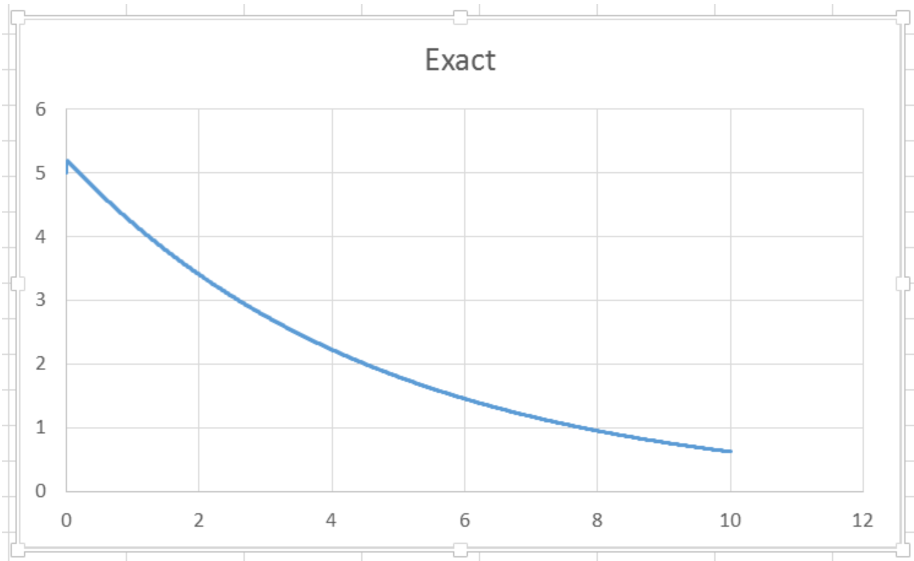


Figure 12: Exact Solution

Figure 13: Error

4 Problem 4

4.1 a) Problem Statement

Write a subroutine that will convolve two sequences using

$$y[n] = \sum_{m=0}^{n} f[m]h[n-m] \equiv f[n] * h[n] \tag{5}$$

. Test your program by convolving the following functions:

1. $f_1 * f_1$
2. $f_1 * f_2$
3. $f_1 * f_3$
4. $f_1 * f_4$
5. $f_2 * f_3$

where the sequences are described by the C/C++ arrays below

$$f_1[] = \{0, 1, 2, 3, 2, 1\};$$

$$f_2[] = \{-2, -2, -2, -2, -2, -2, -2\};$$

$$f_3[] = \{1, -1, 1, -1\};$$

$$f_4[] = \{0, 0, 0, -3, -3\};$$

4.2 Function Plots

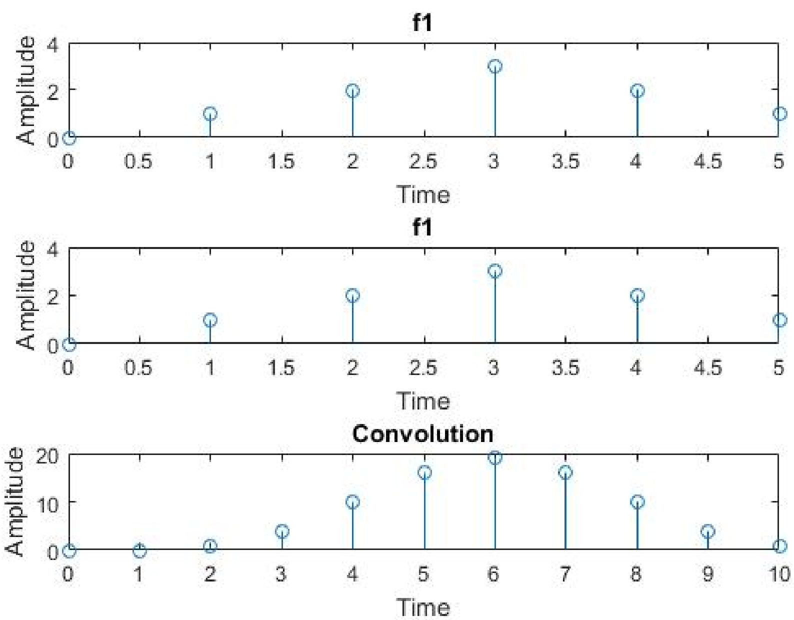


Figure 14: f1*f1

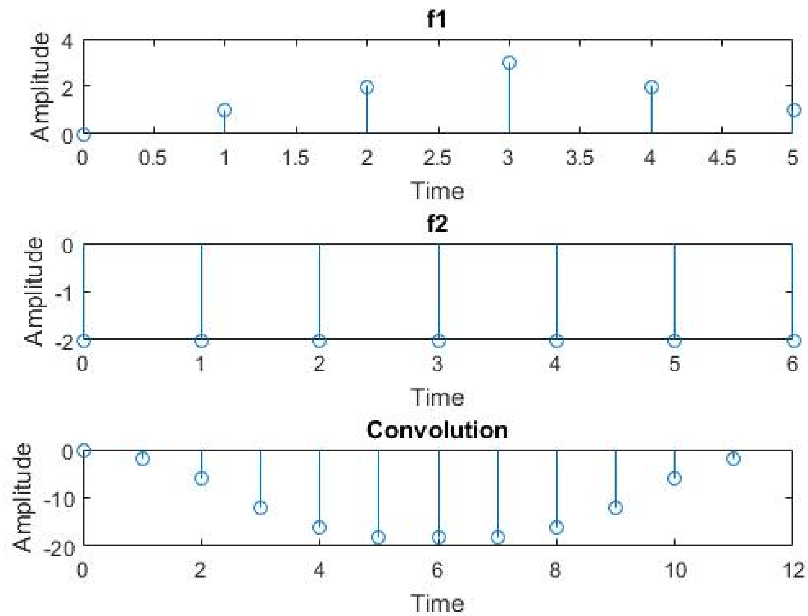


Figure 15: f1*f2

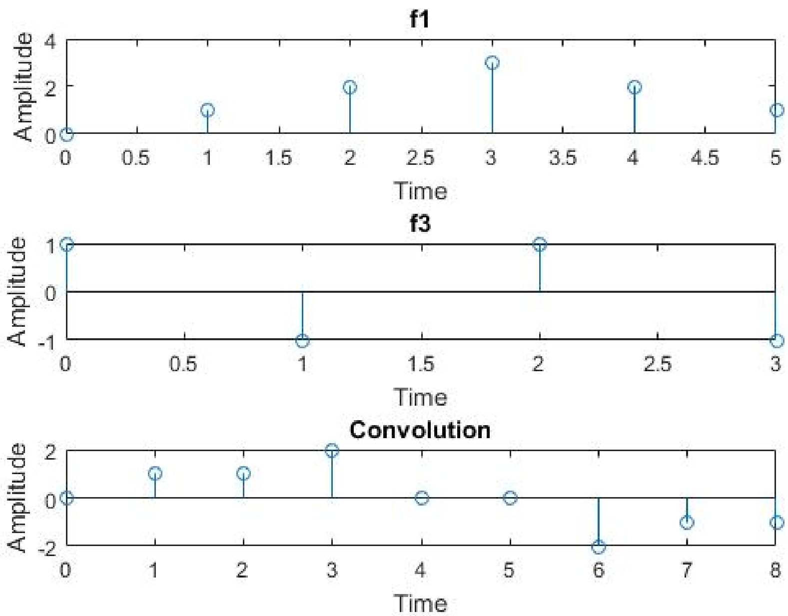


Figure 16: f1*f3

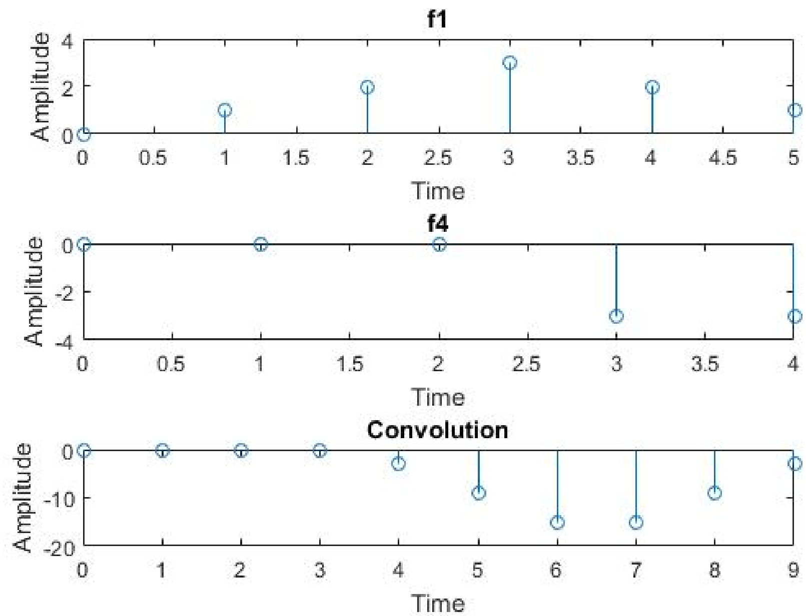


Figure 17: f1*f4

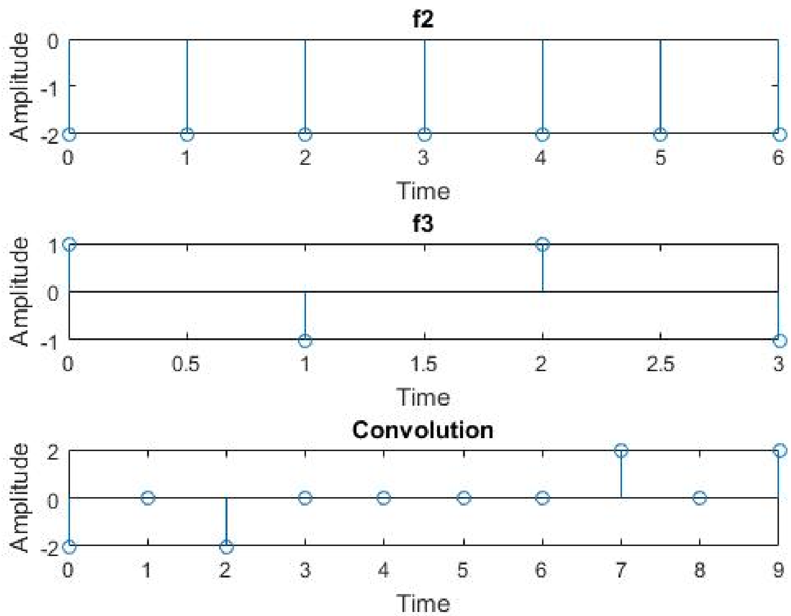


Figure 18: f2*f3

4.3 Convolution by hand and MATLAB

$f_1 = \{0, 1, 2, 3, 2, 1\}$

$f_2 = \{-2, -2, -2, -2, -2, -2, -2\}$

$f_3 = \{1, -1, 1, -1\}$

$f_4 = \{0, 0, 0, -3, -3\}$

$f_1 * f_1 = Y$

```
        1 2 3 2 1 0
   0, 1 2, 3, 2 1
0  0
1  1 0
2  2 1 0
3  3 2 1 0
4  2 3 2 1 0
5  1 2 3 2 1 0
6    1 2 3 2 1 0
7      1 2 3 2
8        1 2 3
9          1 2
10           1
```

$Y(0) = 0(0) = 0$

$Y(1) = 0(1) + 0(1) = 0$

$Y(2) = 0(2) + 1(1) + 2(0) = 1$

$Y(3) = 0(3) + 2(1) + 1(2) + 0(3) = 4$

$Y(4) = 0(2) + 1(3) + 2(2) + 3(1) + 2(0) = 10$

$Y(5) = 0(1) + 2(1) + 2(3) + 3(2) + 2(1) + 1(0) = 16$

$Y(6) = 1(1) + 2(2) + 3(3) + 2(2) + 1(1) = 19$

$Y(7) = 2(1) + 3(2) + 2(3) + 1(2) = 16$

$Y(8) = 3(1) + 2(2) + 1(3) = 10$

$Y(9) = 2(1) + 1(2) = 4$

$Y(10) = 1(1) = 1$

$Y = \{0, 0, 1, 4, 10, 16, 19, 16, 10, 4, 1\}$

$f_1 * f_2$

```
   0 1 2 3 2 1
0  -2
1  -2 -2
2  -2 -2 -2
3  -2 -2 -2 -2
4  -2 -2 -2 -2 -2
5  -2 -2 -2 -2 -2 -2
6  -2 -2 -2 -2 -2 -2 -2
7     -2 -2 -2 -2 -2 -2
8        -2 -2 -2 -2 -2
9           -2 -2 -2 -2
10             -2 -2 -2
11                -2 -2
                     -2
```

$Y(0) = 0(-2) = 0$

$Y(1) = 0(2) + 1(-2) = -2$

$Y(2) = 0(-2) + 1(-2) + 2(-2) = -6$

$Y(3) = 0(-2) + 1(-2) + 2(-2) + 3(-2) = -12$

$Y(4) = 0(-2) + 1(-2) + 2(-2) + 3(-2) + 2(-2) = -16$

$Y(5) = 0(-2) + 1(-2) + 2(-2) + 3(-2) + 2(-2) + 1(-2) = -18$

$Y(6) = 0(-2) + 1(-2) + 2(-2) + 3(-2) + 2(-2) + 1(-2) = -18$

$Y(7) = 1(-2) + 2(-2) + 3(-2) + 2(-2) + 1(-2) = -18$

$Y(8) = 2(-2) + 3(-2) + 2(-2) + 1(-2) = -16$

$Y(9) = 3(-2) + 2(-2) + 1(-2) = -12$

$Y(10) = 2(-2) + 1(-2) = -6$

$Y(11) = 1(-2) = -2$

$Y = \{0, -2, -6, -12, -16, -18, -18, -18, -16, -12, -6, -2\}$

$f_1 * f_3$

	0	1	2	3	2	1
0	1					
1	-1	1				
2	1	-1	1			
3	-1	1	-1	1		
4		-1	[illegible]	-1	1	
5			-1	[illegible]	-1	1
6				-1	[illegible]	-1
7					-1	1
8						-1

$y[0] = 0(1) = 0$
$y[1] = 0(-1) + 1(1) = 1$
$y[2] = 0(1) + 1(-1) + 2(1) = 1$
$y[3] = 0(-1) + 1(1) + 2(-1) + 3(1) = 2$
$y[4] = 1(-1) + 2(1) + 3(-1) + 2(1) = 0$
$y[5] = 2(-1) + 3(1) + 2(-1) + 1(1) = 0$
$y[6] = 3(-1) + 2(1) + 1(-1) = -2$
$y[7] = 2(-1) + 1(1) = -1$
$y[8] = 1(-1) = -1$

$Y = \{0, 1, 1, 2, 0, 0, -2, -1, -1\}$

$f_1 * f_4$

0 0 0 -3 -3

	0	1	2	3	2	1
0	0					
1	0	0				
2	0	0	0			
3	-3	0	0	0		
4	-3	-3	0	0	0	
5		-3	-3	0	0	0
6			-3	-3	0	0
7				-3	-3	0
8					-3	-3
9						-3

$y(0) = 0$
$y(1) = 0$
$y(2) = 0$
$y(3) = 0$
$y(4) = -3$
$y(5) = -9$
$y(6) = -15$
$y(7) = -15$
$y(8) = -9$
$y(9) = -3$

$Y = \{0, 0, 0, 0, -3, -9, -15, -15, -9, -3\}$

$f_2 * f_3$

	-2	-2	-2	-2	-2	-2	-2
0	1						
1	-1	1					
2	1	-1	1				
3	-1	1	-1	1			
4		-1	1	-1	1		
5			-1	1	-1	1	
6				-1	1	-1	1
7					-1	1	-1
8						-1	1
9							-1

$Y = \{-2, 0, -2, 0, 0, 0, 0, 2, 0, 2\}$

$f_1 * f_3$

	0	1	2	3	2	1
0	1					
1	-1	1				
2	1	-1	1			
3	-1	1	-1	1		
4		-1	1	-1	1	
5			-1	1	-1	1
6				-1	1	-1
7					-1	1
8						-1

f_3: 1 -1 1 -1

$Y[0] = 0(1) = 0$

$Y[1] = 0(-1) + 1(1) = 1$

$Y[2] = 0(1) + 1(-1) + 2(1) = 1$

$Y[3] = 0(-1) + 1(1) + 2(-1) + 3(1) = 2$

$Y[4] = 1(-1) + 2(1) + 3(-1) + 2(1) = 0$

$Y[5] = 2(-1) + 3(1) + 2(-1) + 1(1) = 0$

$Y[6] = 3(-1) + 2(1) + 1(-1) = -2$

$Y[7] = 2(-1) + 1(1) = -1$

$Y[8] = 1(-1) = -1$

$Y = \{0, 1, 1, 2, 0, 0, -2, -1, -1\}$

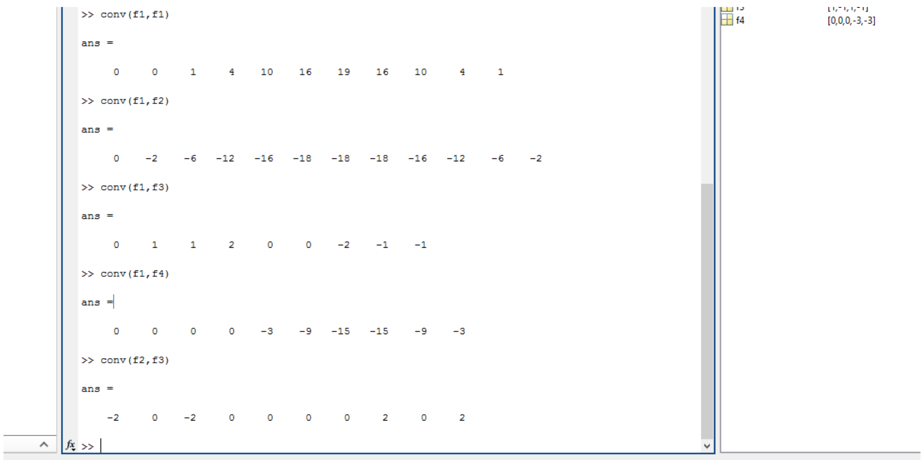


Figure 19: Convolution using MATLAB conv function

5 Problem 5

Using the results from PART I (which computed the zero-input response, find the total solution to the differential equation

$$(D^3 + 3D^2 + 38.25D + 72.5)y(t) = (D + 1.5)f(t) \tag{6}$$

with initial conditions $y(0) = -2, y'(0) = 3, y''(0) = -1.7$ and $f(t) = sin(4\pi t)u(t)$. Use $T = 0.001$, and let $0 \leq t \leq 10$ You will need to incorporate part of PART I into your new program to complete this part.

5.1 a) Problem Statement

Using paper-and- pencil analysis, find the impulse response of the system h(t). Then compute and plot its sampled values $h[k] = h(kT)$. You may use Laplace transform methods if you want.

5.2 a) Solution

$$(D^3 + 3D^2 + 38.25D + 72.5)\,y(t) = (D + 1.5)\,f(t)$$

$$(\lambda + 2)(\lambda^2 + \lambda + 36.25) = 0$$

$$\lambda_1 = -2 \qquad \lambda_{2,3} = -\tfrac{1}{2} \pm j6$$

$$y_0(t) = c_1 e^{-2t} + e^{-\frac{1}{2}t}\left(B_1 \cos(6t) + B_2 \sin(6t)\right)$$

$y_0(0) = -2$

$\dot{y}_0(0) = 3$

$y''_0(0) = -17$

$$\text{ZI } y_0(t) = 1.86144e^{-2t} + .19139\cos(6t + 136.384)$$

$\cancel{y_n(0) = 0}$ $\qquad \cancel{y_n(t) = .0261438e^{-2t} + .026948e^{-\frac{1}{2}t}\cos(6t - 165.96}$

$\cancel{\dot{y}_n(0) = 0}$ $\qquad \cancel{y_n(t) = -.05\ldots e^{-2t} + .161688e^{-\frac{1}{2}t}\sin(165.964}$

$\cancel{y''_n(0) = 1}$ $\qquad \cancel{-.013474e^{-.5t}\cos(165.964 - 6t)}$

computer

$$H(s) = \frac{s + 1.5}{(s+2)(s^2 + s + 36.25)} = -\frac{.0130719}{s+2} + \frac{\overbrace{.0130719}^{A}s + \overbrace{.986928}^{B}}{s^2 + s + 36.25}$$

$\cancel{\text{Alt}}$ $a = \frac{1}{2}$ $\quad b = 6$

$c = 36.25$

$r = .163867$

$\theta = -85.4261$

$$) = \left(-.0130719e^{-2t} + .163867e^{-\frac{1}{2}t}\cos(.986928t - 85.4261)\right)u(t)$$

Figure 20: Exact Solution

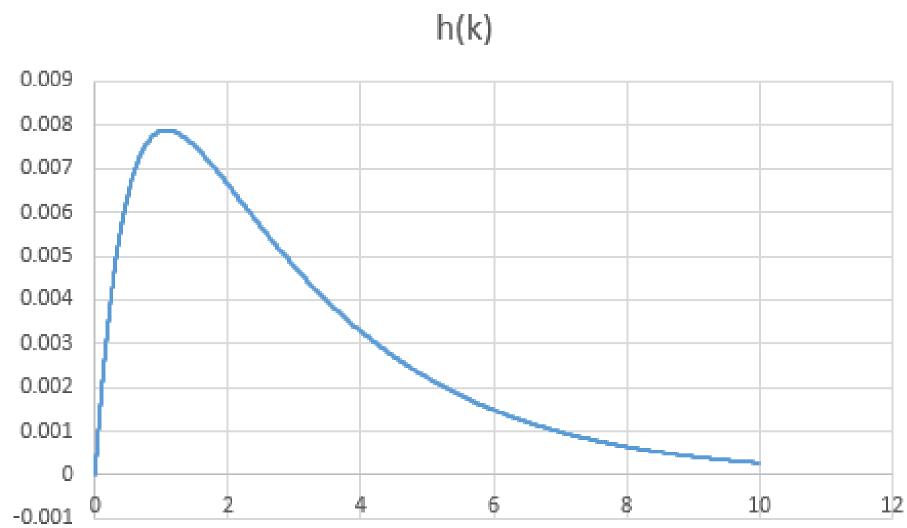


Figure 21: Impulse Response

5.3 b) Problem Statement

Find the sampled values of the input function $f[k] = f(kT)$. Plot these values.

5.4 b) Solution

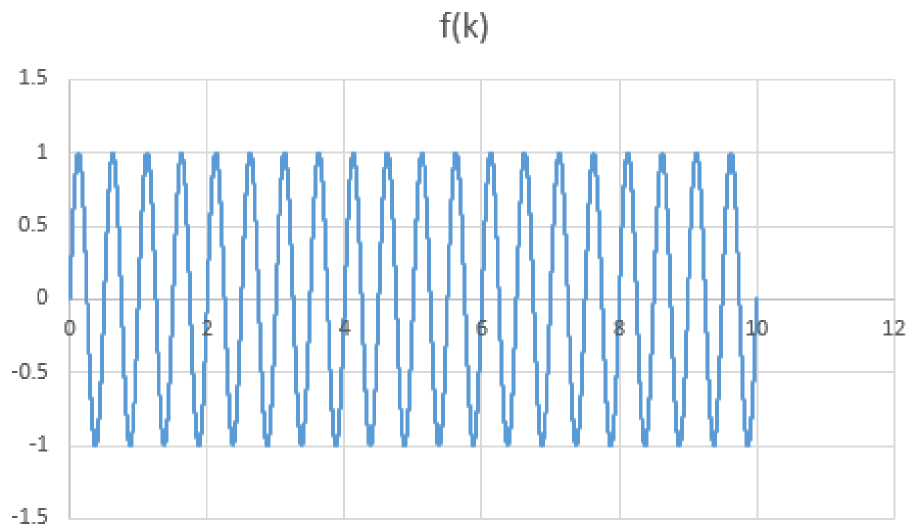


Figure 22: Input Function

5.5 c) Problem Statement

The zero-state solution is the scaled convolution $T(f[k] * h[k])$.

5.6 c) Solution

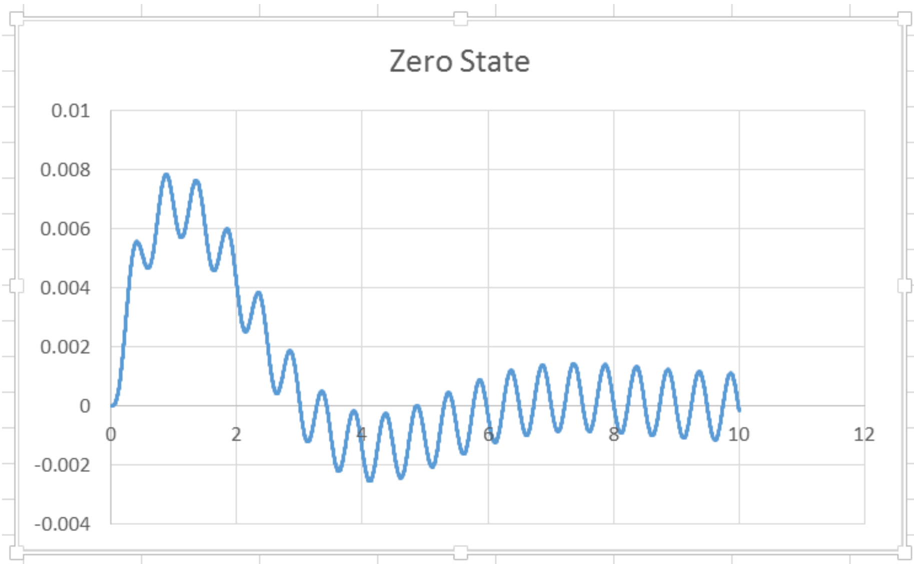


Figure 23: Zero State

5.7 d) Problem Statement

The zero-input solution found using program from PART I.

5.8 d) Solution

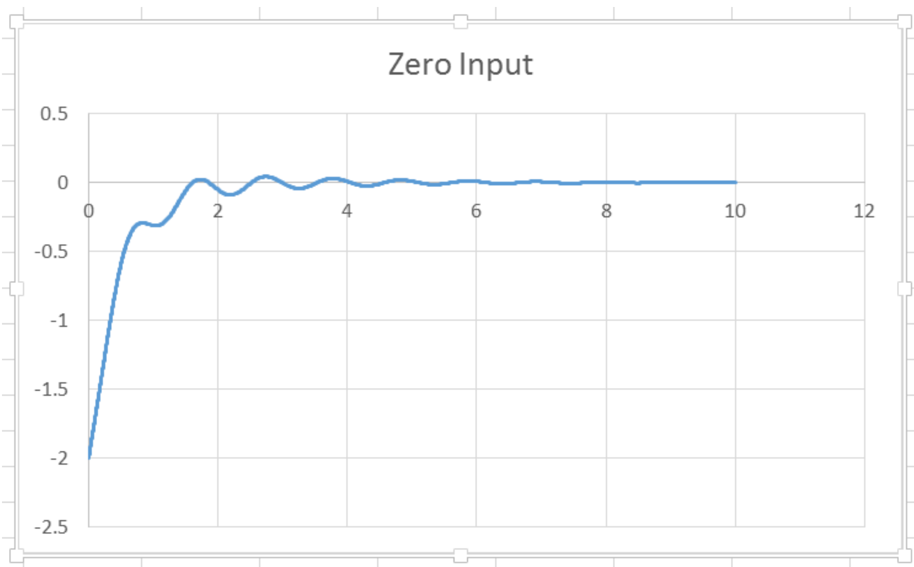


Figure 24: Zero Input

5.9 e) Problem Statement

The total solution is the sum of the zero-state solution and the zero-input solution.

5.10 f) Problem Statement

Compute numerically and plot the total solution.

5.11 f) Solution

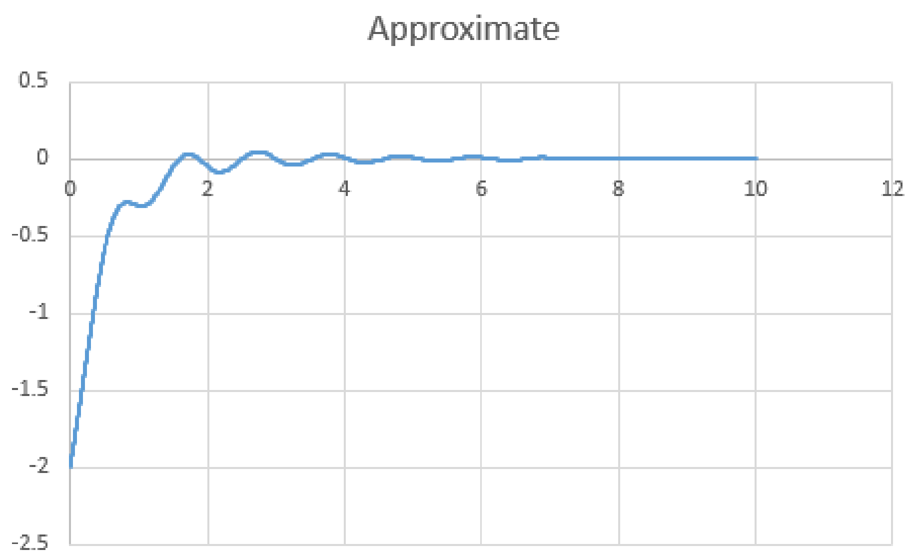


Figure 25: Numerical Solution

5.12 g) Problem Statement

Find an analytical solution to the DE and plot it.

5.13 g) Solution

$$s^3 Y(s) + 2s^2 - 3s + 1.7 + 3[s^2 Y(s) + 2s - 3] + 38.25[sY(s) + 2]$$
$$+ 72.5 Y(s) = sF(s) + 1.5F(s) \qquad F(s) = \frac{4\pi}{s^2 + (4\pi)^2}$$

$$Y(s)[s^3 + 3s^2 + 38.25s + 72.5] = \frac{s4\pi}{s^2+16\pi^2} + \frac{6\pi}{s^2+16\pi^2} - 2s^2 - 3s - 69.2$$

$$Y(s) = \frac{-.0011492s - .101172}{s^2 + 16\pi^2} - \frac{1.86245}{s+2} - \left[\frac{0.1364s + 0760128}{s^2 + s + 36.25}\right]$$

$$y(t) = -.0011492 \cos(4\pi t) - .008051 \sin(4\pi t) - 1.8624 e^{-2t}$$
$$- .178617 e^{-\frac{1}{2}t} \cos(6t - 40.2133^\circ)$$

5.14 h) Problem Statement

Compare the analytical and the numerical solution.

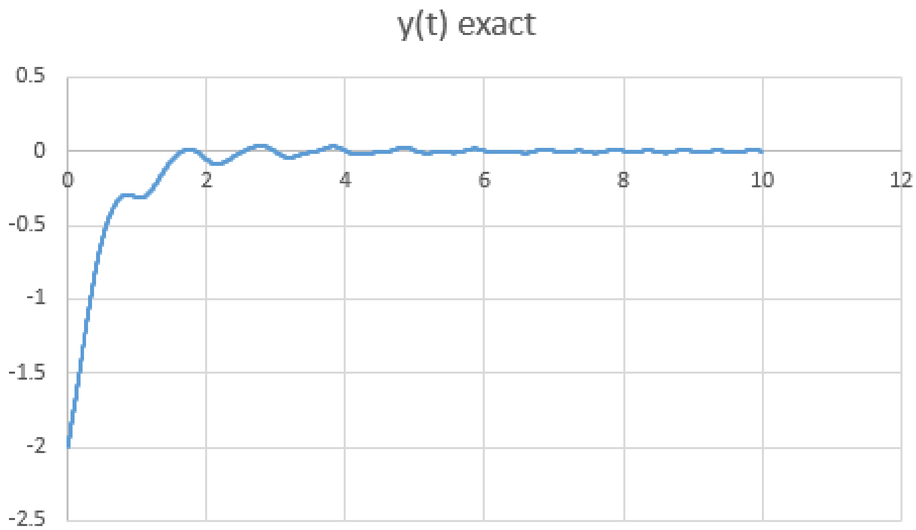


Figure 26: Exact Solution

5.15 h) Solution

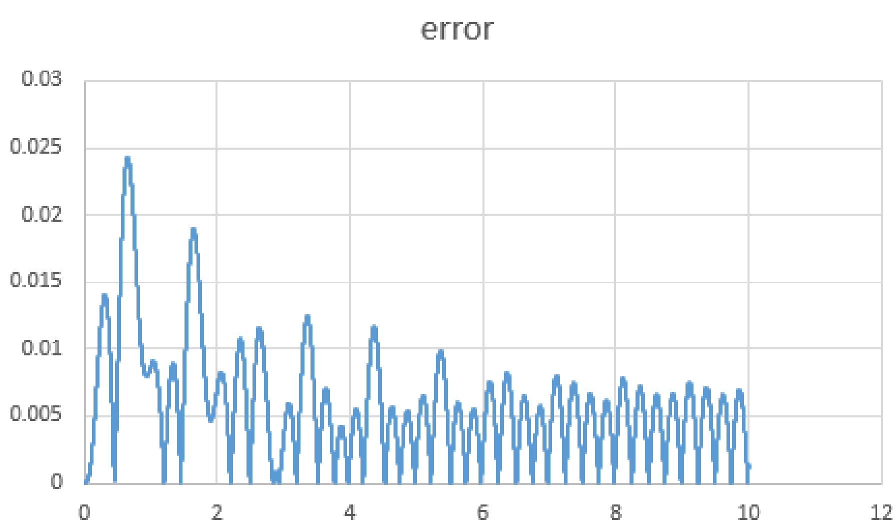


Figure 27: Error when comparing analytical and numerical solutions

As we decreased the stepping size the error is reduced. The greatest amount of error corresponds with the highest values of the derivatives of y(t).

6 Problem 6

Note that the solution to the system output in 5 above settles to a ”steady-state” solution after a few seconds, where the signal form continues unchanged. Your plots should indicate this. What is the steady state output amplitude? Why does this happen?

6.1 Solution

The steady state output amplitude is .00813268. This happens because of the input function f(k).

7 Appendix

7.1 Code for Part I Problem 1

7.1.1 Twin.h

```
#ifndef __T_WIN__
#define __T_WIN__

#include <windows.h>
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <vector>

class Twin {
public:
        Twin(const std::string &);
        template<typename T>
        void println(const T &);
        void println();
        template<typename T>
        void print(const T &);
        template<typename T>
        void printmulti(const T &);
        template<typename T>
        void getInput(T &);
        void getInput();
        void exit();
        void toFile();
private:
        void size();
        void line();
        int rows();
        int columns();
        void display();
        void clear();
        void wincheck();
        int x, y;
        std::vector<std::string> buffer;
        std::string pname;
```

```
};

void Twin::size() {

        CONSOLE_SCREEN_BUFFER_INFO csbi;

        GetConsoleScreenBufferInfo(GetStdHandle(STD_OUTPUT_HANDLE), &csbi);
        x = csbi.srWindow.Right - csbi.srWindow.Left + 1;
        y = csbi.srWindow.Bottom - csbi.srWindow.Top + 1;

}

int Twin::rows() {
        size();
        return y;
}

int Twin::columns() {
        size();
        return x;
}

template<typename T>
void Twin::print(const T &in) {

        if (buffer.size() == 0) buffer.push_back("");

        std::ostringstream oss;
        oss << in;

        std::string temp = buffer.back() + oss.str();

        buffer.pop_back();
        buffer.push_back(temp);

}

template<typename T>
void Twin::println(const T &in) {

        std::ostringstream oss;
        oss << in;

        buffer.push_back(oss.str());

}

template<typename T>
void Twin::printmulti(const T &in) {

        for (auto &i : in)
                println(i);

}

void Twin::println() {
        buffer.push_back("");
}

void Twin::display() {

        wincheck();

        line();
        std::cout << " " << pname << std::endl;
        line();
        std::cout << std::endl;
```

```
	for (auto&& i : buffer) {
		if (i != "") std::cout << " > " << i << std::endl;
		else std::cout << std::endl;
	}

	for (int i = 0; i < y - 7 - int(buffer.size()); i++) std::cout << std::endl;
	line();
	std::cout << " " << "Input:" << std::endl << " > ";
	buffer.clear();

}

void Twin::wincheck() {

	size();

	int b = 0;

	for (auto &i : buffer) {

		if (int(i.length()) > b) b = int(i.length());

	}

	while (y - 8 - int(buffer.size()) < 1 || int(pname.length()) > x - 4 || int(b) > x - 8) {

		std::cout << std::endl << "Window too small.";
		std::cout << std::endl << "Enter 'x' to output to file, or resize window and enter 'r': ";

		char temp;
		std::cin >> temp;
		std::cin.ignore(1000, '\n');
		std::cin.clear();

		if (temp == 'x' || temp == 'X') {
			toFile();
			buffer.push_back("[See " + pname + ".txt for output.]");
		}

		size();

	}

}

Twin::Twin(const std::string &name) {

	pname = name;

}

void Twin::line() {

	size();

	std::cout << " ";
	for (int i = 0; i < x - 2; i++) std::cout << "=";
	std::cout << " ";

}

void Twin::clear() {
	buffer.clear();
}

template<typename T>
void Twin::getInput(T &var) {
```

```
    std::vector<std::string> backup = buffer;

    while (true) {
        display();

        if (std::cin >> var) {
            std::cin.ignore(1000, '\n');
            std::cin.clear();

            buffer.push_back("Input:");
            buffer.push_back("");
            println(var);
            buffer.push_back("");
            buffer.push_back("Enter 'x' to save and continue.");
            display();

            char temp;
            std::cin >> temp;
            std::cin.ignore(1000, '\n');
            std::cin.clear();

            if (temp == 'x' || temp == 'X') break;
            else buffer = backup;
        }
        else {

            std::cin.clear();
            std::cin.ignore(1000, '\n');

            buffer.push_back("Invalid entry.");
            buffer.push_back("");
            buffer.push_back("Press enter to continue.");
            display();
            buffer = backup;

            std::cin.ignore(1000, '\n');
            std::cin.clear();
        }
    }

}

void Twin::getInput() {

    buffer.push_back("Press enter to continue.");
    display();

    std::cin.ignore(1000, '\n');
    std::cin.clear();

}


void Twin::exit() {

    buffer.push_back("Press enter to exit. . .");
    display();

    std::cin.ignore(1000, '\n');
    std::cin.clear();

}

void Twin::toFile() {

    std::ofstream fout(pname + ".txt");
    if (fout.is_open()) {
        for (auto &i : buffer)
```

```
                    fout << i << std::endl;

                fout.close();
                buffer.clear();
                buffer.push_back("Saved output to " + pname + ".txt.");
                buffer.push_back("");
                getInput();
        }
        else std::cout << std::endl << "Unable to open file.";
}
#endif
```

7.1.2 Main.c

```
#include "Twin.h"

int main() {

        //
        // Get input from user
        //

        double a, dt, lt, ut, iy;
        std::vector<double> result;

        Twin t("Euler's Method");

        t.println("For the equation (D + a) * y(t) = 0, enter a.");
        t.println("Example: 1.2");
        t.getInput(a);

        t.println("Enter lower bound of t.");
        t.println("Example: 2.3");
        t.getInput(lt);

        t.println("Enter upper bound of t.");
        t.println("Example: 3.4");
        t.getInput(ut);

        t.println("Enter step size of t.");
        t.println("Example: .25");
        t.getInput(dt);

        t.println("Enter initial condition, y(");
        t.print(lt);
        t.print(").");
        t.println("Example: 6.5");
        t.getInput(iy);

        t.println("Enter 'd' to show results, or 'f' to save results to file.");
        char d;
        t.getInput(d);

        //
        // Calculate
        //

        result.push_back(iy);
        for (double i = lt + dt; i <= ut; i += dt)
                result.push_back((1 - dt * a) * result.back());

        //
        // Display results
        //

        t.println("Values of y for (D + ");
        t.print(a * -1);
        t.print(") * y(t) = 0 when t is between ");
```

```
        t.print(lt);
        t.print(" and ");
        t.print(ut);
        t.print(" with a step size of ");
        t.print(dt);
        t.print(", and ");
        t.print("y(");
        t.print(lt);
        t.print(") = ");
        t.print(iy);
        t.print(".");

        t.println();
        t.printmulti(result);
        t.println();

        if (d != 'd' && d != 'D') t.toFile();

        t.exit();
        return EXIT_SUCCESS;
}
```

7.2 Code for Part I Problem 2

```
#include "Twin.h"

int main() {

        Twin t("Euler's Method");
        mat A(3), I(3), c(3);

        A[0] = { 0, 1, 0 };
        A[1] = { 0, 0, 1 };
        A[2] = { 0, 0, 0 };

        I[0] = { 1, 0, 0 };
        I[1] = { 0, 1, 0 };
        I[2] = { 0, 0, 1 };

        c[0] = { 0, 0, 0 };
        c[1] = { 0, 0, 0 };
        c[2] = { 0, 0, 0 };

        vec a;

        std::vector<std::vector<double>> results;

        double lt, ut, dt;

        while (1) {
                t.println("For (D^3 + a2*D^2 + a1*D + a0) * y(t) = 0, enter a2, a1, and a0.");
                a = t.gvec();
                if (a.size() == 3) break;
                else t.println("Invalid vector.");
        }

        t.println("Enter lower bound of t.");
        t.getInput(lt);

        t.println("Enter upper bound of t.");
        t.getInput(ut);

        while (1) {
                t.println("Enter the initial condition, vector x(");
                t.print(lt);
                t.print(").");
                results.push_back(t.gvec());
```

```
            if (a.size() == 3) break;
            else t.println("Invalid vector.");
      }

      t.println("Enter step size of t.");
      t.getInput(dt);

      A[2][0] = a[2] * -1;
      A[2][1] = a[1] * -1;
      A[2][2] = a[0] * -1;

      //
      // Get constant matrix
      //

      for(int i = 0; i < 3; i++)
            for (int j = 0; j < 3; j++)
                  c[i][j] = A[i][j] * dt + I[i][j];

      //
      // Compute result vectors
      //

      for (double r = lt + dt; r <= ut; r += dt) {

            std::vector<double> vtemp;

            for (int i = 0; i < 3; i++) {
                  double temp = 0;
                  for (int j = 0; j < 3; j++) {
                        temp += c[i][j] * results.back()[j];
                  }
                  vtemp.push_back(temp);
            }

            results.push_back(vtemp);
      }

      //
      // Plot y
      //

      std::vector<double> plot;

      for (auto &i : results)
            plot.push_back(i[0]);

      t.printmulti(plot);
      t.toFile();

      t.exit();
      return EXIT_SUCCESS;
}
```

7.3 Code for Part II Problem 4

```
#include <iostream>
#include <algorithm>

using namespace std;

int conv(double*f1, int len1, double*f2, int len2, double *y) {
      int sum = 0;
      int k = 0;

      int j;
      int leny = (len1 + len2) - 1;
```

```
    int i=0;
    cout << "f1: ";
    for (int i = 0; i < len1; i++) { cout << f1[i] << " "; }
    cout << endl;
    cout << "f2: ";
    for (int i = 0; i < len2; i++) { cout << f2[i] << " "; }
    cout << endl;

    //For each element of convolution array
    for (int index = 0; i < leny; index++) {

        i = index;
            sum = 0;
            //iterate through f2
            for (int j = 0; j < len2; j++) {
                //Check convolution bounds
                if (i >= 0 && i < len1) {

                    sum += (f1[i] * f2[j]);
                }
                //iterate through f1
                    i = i - 1;
                    //store sum into convolution array
                    y[index] = sum;
                }
            }



        cout << " y= [";
        for (int i = 0; i < leny; i++) {
            cout << " " << y[i];




        }
        cout << "]";
        cout << endl;

    return leny;
}

int main() {
    int leny;
    double f1[] = { 0,1,2,3,2,1 };
    int len1 = 6;
    double f2[] = { -2,-2,-2,-2,-2,-2,-2 };
    int len2 = 7;
    double f3[] = { 1,-1,1,-1 };
    int len3 = 4;
    double f4[] = { 0,0,0,-3,-3 };
    int len4 = 5;
    double y[20] = { 0 };


    cout << "f1*f1 = [ 0 0 1 4 10 16 19 16 10 4 1]"<<endl;
    leny = conv(f1, len1, f1, len1, y);
    cout << endl;
    cout << "f1*f2 = [ 0 -2 -6 -12 -16 -18 -18 -18 -16 -12 -6 -2]" << endl;
    leny = conv(f1, len1, f2, len2, y);
    cout << endl;
    cout << "f1*f3 = [ 0 1 1 2 0 0 -2 -1 -1]" << endl;
    leny = conv(f1, len1, f3, len3, y);
    cout << endl;
    cout << "f1*f4 = [ 0 0 0 0 -3 -9 -15 -15 -9 -3]" << endl;
    leny = conv(f1, len1, f4, len4, y);
    cout << endl;
```

```
	cout << "f2*f3 = [-2 0 -2 0 0 0 0 2 0 2]" << endl;
	leny = conv(f2, len2, f3, len3, y);
	cout << endl;

	system("pause");
	return 0;
}
```

7.4 Code for Part II Problem 5

```
#include "Twin.h"

std::vector<double> fromFile(const std::string &, Twin &);
std::vector<double> conv2(const std::vector<double> &, const std::vector<double> &);
std::vector<double> zeroInput(Twin &t);

int main() {

	Twin t("Total Response");

	std::vector<double> h = fromFile("Impulse.txt", t);
	std::vector<double> f = fromFile("Input.txt", t);
	std::vector<double> zs = conv2(f, h);
	std::vector<double> zi = zeroInput(t);

	std::vector<double> total;

	for (int i = 0; i < zi.size(); i++)
		total.push_back(zi[i] + zs[i]);
	t.printmulti(total);
	t.toFile();

	t.exit();
	return EXIT_SUCCESS;
}

std::vector<double> fromFile(const std::string &name, Twin &t) {
	double temp;
	std::vector<double> h;

	std::ifstream fin(name);
	if (fin)
		while (fin >> temp)
			h.push_back(temp);
	else {
		t.println("Unable to open file " + name + ".");
		t.getInput();
	}
	fin.close();
```

```
        return h;
}

std::vector<double> conv2(const std::vector<double> &f, const std::vector<double> &h) {
        std::vector<double> y(f.size() + h.size() - 1);
        double sc = .001;

        int i, j, k;

        for (i = 0; i < h.size() - 1; ++i) {
                y[i] = 0;
                for (j = i, k = 0; j >= 0; --j, ++k)
                        y[i] += f[j] * h[k];
                y[i] *= sc;
        }

        for (i = h.size() - 1; i < f.size(); ++i) {
                y[i] = 0;
                for (j = i, k = 0; k < h.size(); --j, ++k)
                        y[i] += f[j] * h[k];
                y[i] *= sc;
        }

        return y;
}

std::vector<double> zeroInput(Twin &t) {

        mat A(3), I(3), c(3);

        A[0] = { 0, 1, 0 };
        A[1] = { 0, 0, 1 };
        A[2] = { 0, 0, 0 };

        I[0] = { 1, 0, 0 };
        I[1] = { 0, 1, 0 };
        I[2] = { 0, 0, 1 };

        c[0] = { 0, 0, 0 };
        c[1] = { 0, 0, 0 };
        c[2] = { 0, 0, 0 };

        vec a;

        std::vector<std::vector<double>> results;

        double lt, ut, dt;

        while (1) {
                t.println("For (D^3 + a2*D^2 + a1*D + a0) * y(t) = 0, enter a2, a1, and a0.");
                a = t.gvec();
                if (a.size() == 3) break;
                else t.println("Invalid vector.");
        }

        t.println("Enter lower bound of t.");
        t.getInput(lt);

        t.println("Enter upper bound of t.");
        t.getInput(ut);

        while (1) {
                t.println("Enter the initial condition, vector x(");
                t.print(lt);
                t.print(").");
                results.push_back(t.gvec());
                if (a.size() == 3) break;
                else t.println("Invalid vector.");
```

```
	}

	t.println("Enter step size of t.");
	t.getInput(dt);

	A[2][0] = a[2] * -1;
	A[2][1] = a[1] * -1;
	A[2][2] = a[0] * -1;

	//
	// Get constant matrix
	//

	for (int i = 0; i < 3; i++)
		for (int j = 0; j < 3; j++)
			c[i][j] = A[i][j] * dt + I[i][j];

	//
	// Compute result vectors
	//

	for (double r = lt + dt; r <= ut; r += dt) {

		std::vector<double> vtemp;

		for (int i = 0; i < 3; i++) {
			double temp = 0;
			for (int j = 0; j < 3; j++) {
				temp += c[i][j] * results.back()[j];
			}
			vtemp.push_back(temp);
		}

		results.push_back(vtemp);
	}

	//
	// Plot y
	//

	std::vector<double> plot;

	for (auto &i : results)
		plot.push_back(i[0]);

	return plot;

	t.exit();
}
```