EE324 Control Systems Lab

Problem Sheet 3

Param Rathour | 190070049

Autumn Semester 2021-22

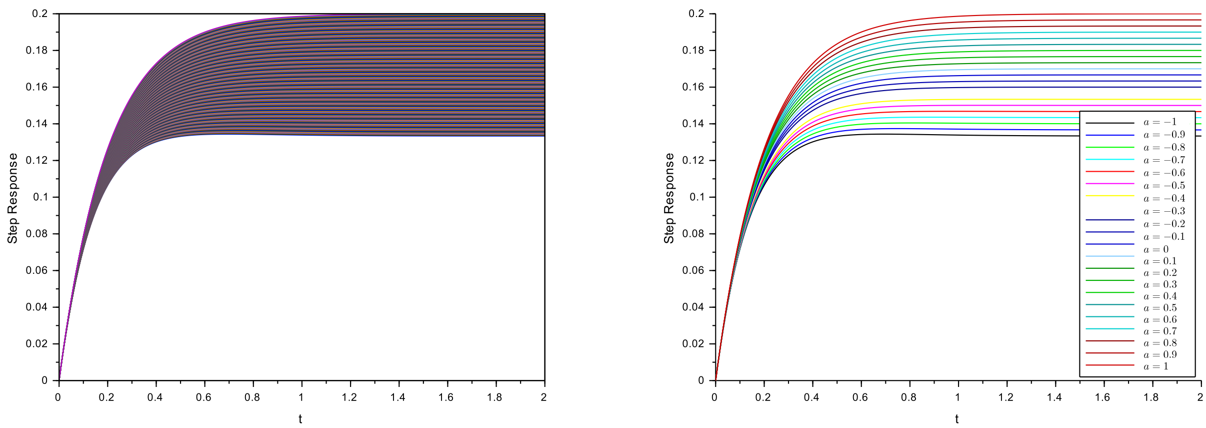
Contents

1 Q1

a)

$$G(s) = \frac{s+5+a}{s^2+11s+30} = \frac{s+5+a}{(s+5)(s+6)}$$

So, $-5, -6$ are poles of $G(s)$, for pole-zero cancellation $s+5+a =$ either $(s+5)$ or $(s+6) \rightarrow a = 0$ or $a = 1$



(a) $a \in$ `-1:0.01:1`

(b) $a \in$ `-1:0.1:1`

Figure 1.1: Variation in Step Response of $G(s)$

The Figure 1.1b has values of a in a lower number to increase visibilty.

The steady state value is increasing 'smoothly' with a and there is no discontinuity in graph for $a = 0$.

Hence, pole-zero cancellation works in this case.

Code to obtain Figure 1.1a

```
s = %s;
tMax = 2;
tStep = 0.01;
t = 0:tStep:tMax;
aValues = -1:0.01:1;
y = [];
dims = 1;
for x = 1:length(aValues)
    a = aValues(x);
    G = simp((s + 5 + a) / (s^2 + 11 * s + 30));
    sys = syslin('c',G);
    y = cat(dims, y, csim('step', t, sys));
end
plot(t',y'); xlabel(" t" ); ylabel(" Step Response" );
xs2pdf(0,'Q1a');
```

Code to obtain Figure 1.1b

```
s = %s;
tMax = 2;
tStep = 0.01;
t = 0:tStep:tMax;
aValues = -1:0.1:1;
legendValues = []
y = [];
dims = 1;
for x = 1:length(aValues)
    a = aValues(x);
    str = "$a = " + string(a) + "$";
    legendValues = cat(dims, legendValues, str);
    G = simp((s + 5 + a)/(s^2 + 11 * s + 30));
    sys = syslin('c',G);
    y = cat(dims, y, csim('step', t, sys));
end
plot2d(t',y'); xlabel(" t" ); ylabel(" Step Response" );
legend(legendValues,opt=4);
xs2pdf(0,'Q1aLR');
```

b)

$$G_1(s) = \frac{1}{s^2 - s - 6} = \frac{1}{(s+2)(s-3)}$$

The Step response of $G_1(s)$ (Figure 1.2a) doesn't have any steady state value (unbounded).

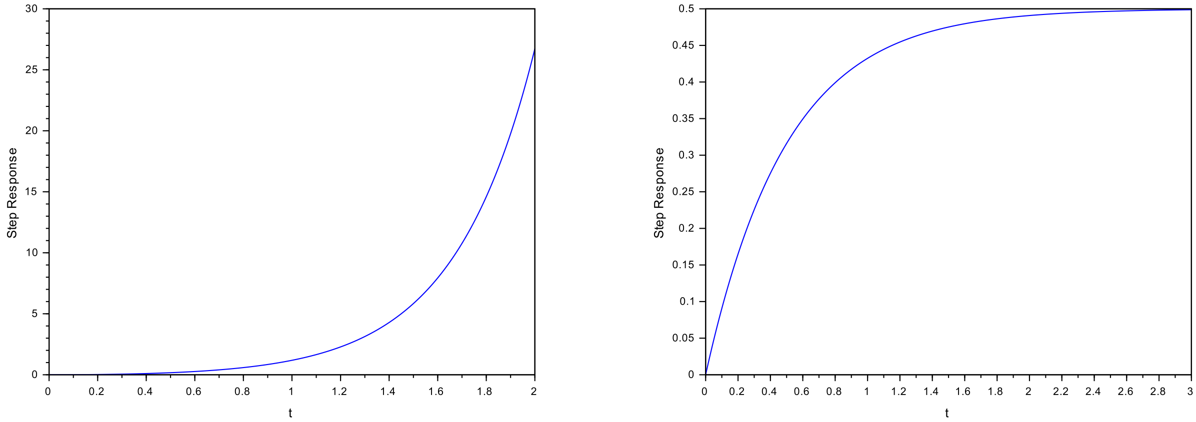
So, $G_1(s)$ is unstable.

This unstability is caused by the pole 3. To cancel it we add a zero at 3. Now,

$$G_2(s) = \frac{(s-3)}{(s+2)(s-3)} = \frac{1}{(s+2)}$$

The Step response of $G_2(s)$ (Figure 1.2b) has a steady state value at 0.5 (bounded).

So, $G_2(s)$ is stable.



(a) Step Respose of G_1 (b) Step Respose of G_2

Code to obtain Figure 1.2a

```
s = %s;
G = simp(1 / (s^2 - s - 6));
sys = syslin('c',G);
tStep = 0.001;
t = 0:tStep:2;
gp = csim('step', t, sys);
plot(t,gp); xlabel("t"); ylabel("Step Response");
xs2pdf(0,'Q1b1');
```

Code to obtain Figure 1.2b

```
s = %s;
G = simp((s - 3) / (s^2 - s - 6));
sys = syslin('c',G);
tStep = 0.001;
t = 0:tStep:3;
gp = csim('step', t, sys);
plot(t,gp); xlabel("t"); ylabel("Step Response");
xs2pdf(0,'Q1b2');
```

Now we shift the zero slightly by adding a small variable parameter, ‘a’ , and plot the response for different values of this parameter.

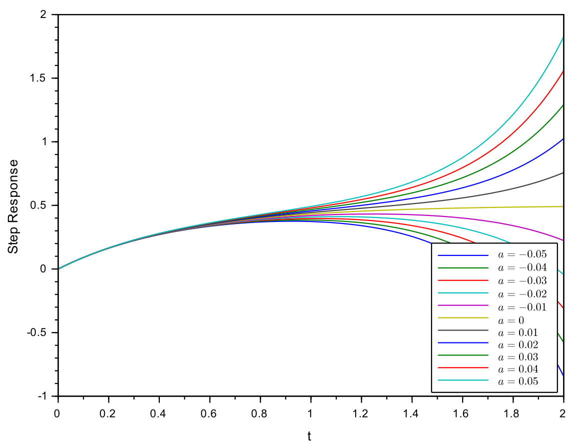


Figure 1.3: Variation in Step Response of $G_2(s)$ in a from `-1:0.01:1`

Code to obtain Figure 1.3

```
s = %s;
tStep = 0.001;
t = 0:tStep:2;
aValues = -0.05:0.01:0.05;
legendValues = []
y = [];
dims = 1;
for x = 1:length(aValues)
    a = aValues(x);
    str = "$a = " + string(a) + "$";
    legendValues = cat(dims, legendValues, str);
    G = simp((s - 3 + a) / (s^2 - s - 6));
    sys = syslin('c',G);
    y = cat(dims, y, csim('step', t, sys));
end
plot(t',y'); xlabel(" t" ); ylabel(" Step Response" );
legend(legendValues,opt=4);
xs2pdf(0,'Q1b3');
```

The system (Figure 1.3) is stable only for exact value of zero required. Even for slightest variations of zero the step response is unbounded, so the system becomes unstable. At a physical location, it is extremely difficult to get **exact** value of pole in RHP and implement that value exactly as a zero (to implement this gain we will need op-amps or other physical devices which have non-idealities).

From this, we can infer "An unstable plant cannot be rendered stable by canceling unstable poles by adding zeros attempting to cancel the unstable pole".[1]

2 Q2

a)

$$G_1(s) = \frac{85}{(s^3 + 7s^2 + 27s + 85)}$$

Using **tf2zp**, the poles of $G_1(s)$ are at $-5, -1 \pm 4\iota$. The 3$^{\text{rd}}$ pole is atleast 5 times away from orgin as other two.

Hence, Dominant Pole Approximation neglecting the pole 5 will give us 2$^{\text{nd}}$ order approximation of the system

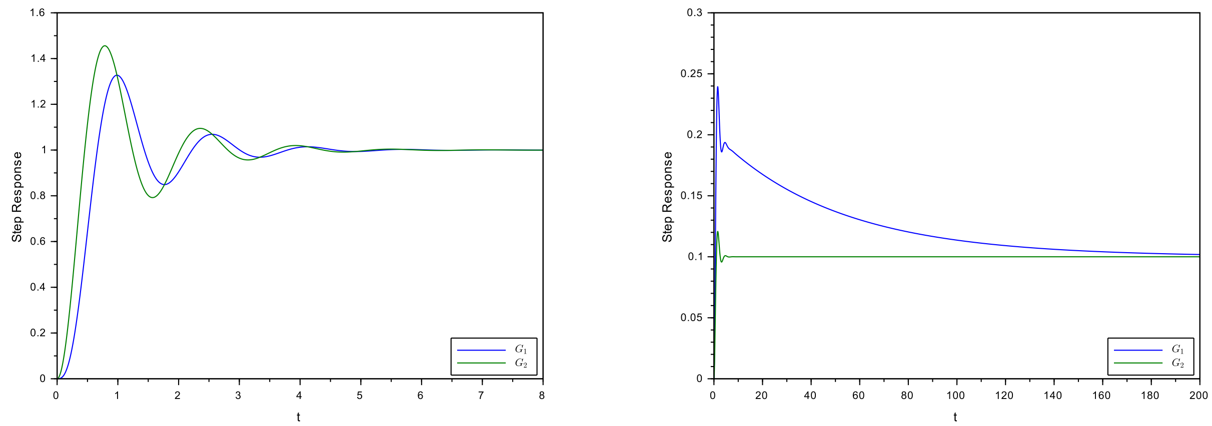
$$G_2(s) = G_1 \cdot \frac{(s+5)}{5} = \frac{17}{(s - (-1 + 4\iota))(s - (-1 - 4\iota))} = \frac{17}{s^2 + 2s + 17} \qquad \text{(Division by 5 to maintain steady state value)}$$

Code to obtain Figure 2.1a

```
s = %s;
G_1 = 85 / (s^3 + 7 * s^2 + 27 * s + 85);
sys = syslin('c',G_1);
tStep = 0.001;
t = 0:tStep:8;
[z, p, k] = tf2zp(G_1)
y = [];
dims = 1;
y = cat(dims, y, csim('step', t, sys));

G_2 = G_1 * ((s + 5) / 5);
sys = syslin('c',G_2);
y = cat(dims, y, csim('step', t, sys));
plot(t',y'); xlabel(" t" ); ylabel(" Step Response" );
legend(["$G_1$";"$G_2$"],opt=4);
xs2pdf(0,'Q2a');
```

[1] Non-zero initial conditions also play a significant role. One can't cancel poles with zeros directly if non-zero initial conditions exists



(a) Dominant Pole Approximation

(b) Pole Zero Cancellation

Figure 2.1: Step Response of $G_1(s)$ (3^{rd} order system) & approximated by a (2^{nd} order system) a

b)

$$G_1 = \frac{(s+0.01)}{\left(s^3 + \frac{101}{50}s^2 + \frac{126}{25}s + 0.1\right)}$$

Using `tf2zp`, the poles of $G_1(s)$ are at $-0.02, -1 \pm 2\iota$ and the zero is at -0.01 The zero -0.01 is very close to the pole -0.02. Hence, Pole Zero Cancellation will give us 2^{nd} order approximation of the system

$G_2(s) = G_1(s) \cdot \frac{\frac{(s+0.02)}{0.02}}{\frac{(s+0.01)}{0.01}} = \frac{1}{2(s^2+2s+5)}$ (Multiplication & Division by constants is done to maintain steady state value)

Code to obtain Figure 2.1b

```
s = %s;
G_1 = (s + 0.01) / (s^3 + (101 / 50) * s^2 + (126 / 25) * s + 0.1);
sys = syslin('c',G_1);
tStep = 0.01;
t = 0:tStep:200;
y = [];
dims = 1;
y = cat(dims, y, csim('step', t, sys));

[z, p, k] = tf2zp(G_1)

G_2 = G_1 / ((s + 0.01) / 0.01) * ((s + 0.02) / 0.02);
sys = syslin('c',G_2);
y = cat(dims, y, csim('step', t, sys));

plot(t',y'); xlabel(" t" ); ylabel(" Step Response" );
legend(["$G_1$";"$G_2$"],opt=4);
ax = gca();
ax.data_bounds=[0 0;200 0.3];
xs2pdf(0,'Q2b');
```

Functions to determine Time Domain Parameters

I have used above functions extensively to calculate Time Domain Parameters of a response using it's output.

To use these functions you must pass the output of `csim` command as a input to the function.

Percent Overshoot

```
function pOS = percentOvershoot(gp)
    finalValue = gp($);
    maxValue = 0;
    for x = 1:length(gp)
        if gp(x) > maxValue
            maxValue = gp(x)
        end
    end
    pOS = (maxValue - finalValue) * 100 / finalValue
endfunction
```

2% Settling Time

```
function Ts = settlingTime(gp, tStep)
    finalValue = gp($);
    Ts = 0;
    for x = length(gp):-1:1
        if abs(gp(x) - finalValue) > (0.02 * finalValue)
            Ts = x;
        break
        end
    end
    Ts = Ts * tStep;
endfunction
```

Delay Time

```
function Td = delayTime(gp, tStep)
    finalValue = gp($);
    Td = 0
    for x = 1:length(gp)
        if (gp(x) > finalValue / 2)
            Td = x;
            break
        end
    end
    Td = Td * tStep;
endfunction
```

Peak Time

```
function Tp = peakTime(gp, tStep)
    Tp = -1;
    for x = 2:length(gp)-1
        if (gp(x) > gp(x-1)) & (gp(x) > gp(x+1))
            Tp = x;
            break
        end
    end
    Tp = Tp * tStep;
endfunction
```

Rise Time

```
function Tr = riseTime(gp, tStep)
    finalValue = gp($);
    Tr = 0;
    for x = 1:length(gp)
        if gp(x) > 0.9*finalValue
            Tr = Tr + x;
            break
        end
        if gp(x) < 0.1*finalValue
            Tr = -x;
        end
    end
    Tr = Tr * tStep;
endfunction
```

Note. *These functions give correct values when those values are defined for the system, otherwise it is unpredictable.*

3 Q3

$$G_1(s) = \frac{9}{(s^2 + 2s + 9)}$$

Using `tf2zp`, the poles of $G_1(s)$ are at $-1 \pm \sqrt{2}\iota$.

a)

Let's add -5 as a zero to $G_1(s)$

$$G_2(s) = \frac{9(s+5)}{(s^2 + 2s + 9)}$$

System	Rise Time (in s)	% Overshoot
G_1	0.456	32.93
G_2	0.356	40.11

b)

Let's add -1 (as the nearer pole) and -20 (as a farther pole) to $G_1(s)$

$$G_3(s) = \frac{9}{(s^2 + 2s + 9)(s+1)}$$

$$G_4(s) = \frac{9}{(s^2 + 2s + 9)(s+20)}$$

System	Rise Time (in s)	% Overshoot
G_3	2.178	0
G_4	0.465	32.54

c) Observations

Effect of Additional Zeroes

- Adding zeroes changes the peak of the response, but the overall shape is maintained. Rise Time decreases and % Overshoot increases as additional zero values gets farther from origin than original poles.

This is because adding zeros takes Linear Combination of the step response of $G_1(s)$ and it's derivative. Closer the zero to 0 larger the effect it has on system.

Effect of Additional Poles

- Adding poles nearer to origin than original poles has a significant effect on the step responses of the system $(G_1(s), G_3(s))$. Rise Time increases and % Overshoot becomes 0 with additional pole values getting closer to origin than original poles. So, the system behaves like first order approximately.
- Adding poles farther to origin than original poles has a negligible effect on the step responses of the system $(G_1(s), G_4(s))$. Rise Time and % Overshoot becomes closer to orignial values with additional pole values getting farther from origin than original poles.

This is because additional poles add exponential terms in the step response. Larger the magnitude of power of exponential decay smaller the effect it has on system.

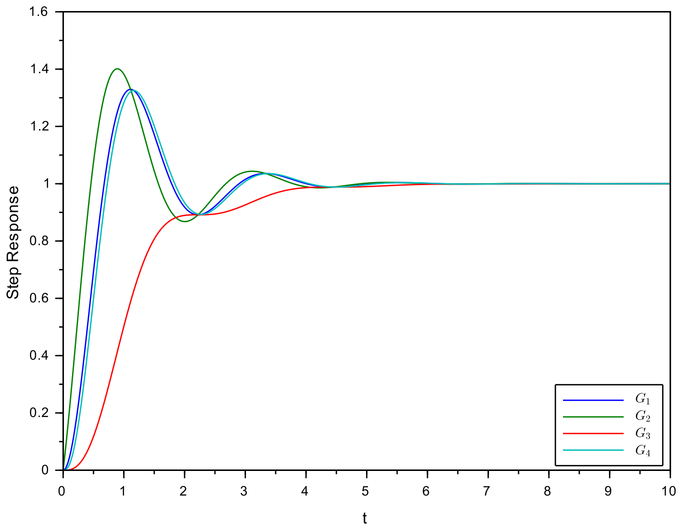


Figure 3.1: Variation in Step Response of $G_1(s)$, $G_2(s)$, $G_3(s)$ & $G_4(s)$

Code to obtain Figure 3.1

```
s = %s;
tStep = 0.001;
tMax = 20;
t = 0:tStep:tMax;
TF = 9 / (s^2 + 2*s + 9);
G = [TF ; TF * (s + 5) / 5; TF / (s + 1); TF * 20 / (s + 20)];
[z,p,k]=tf2zp(G(2)(1)/G(3)(1))

y = [];
legendValues = [];
dims = 1;
for x = 1:length(G)
    str = "$G_{" + string(x) + "}$";
    legendValues = cat(dims, legendValues, str);
    sys = syslin('c',G(2)(x),G(3)(x));
    gp = csim('step', t, sys);
    y = cat(dims, y, csim('step', t, sys));
    disp(riseTime(gp,tStep), percentOvershoot(gp));
end
plot(t',y'); xlabel(" t" ); ylabel(" Step Response" );
legend(legendValues,opt=4);
ax = gca();
ax.data_bounds=[0 0;10 1.5];
xs2pdf(0,'Q3');
```

4 Q4

General second order continuous time system with no zeros has the transfer function as

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{1}{s^2 + 2\zeta s + 1} \quad \text{(take } \omega_n = 1\text{)}$$

Now,

$$G_1(s) = \frac{1}{s^2 + 1} \qquad \text{(take } \zeta = 0\text{), Undamped} \tag{4.1}$$

$$G_2(s) = \frac{1}{s^2 + s + 1} \qquad \text{(take } \zeta = 1/2\text{), Underdamped} \tag{4.2}$$

$$G_3(s) = \frac{1}{s^2 + 4s + 1} \qquad \text{(take } \zeta = 2\text{), Overdamped} \tag{4.3}$$

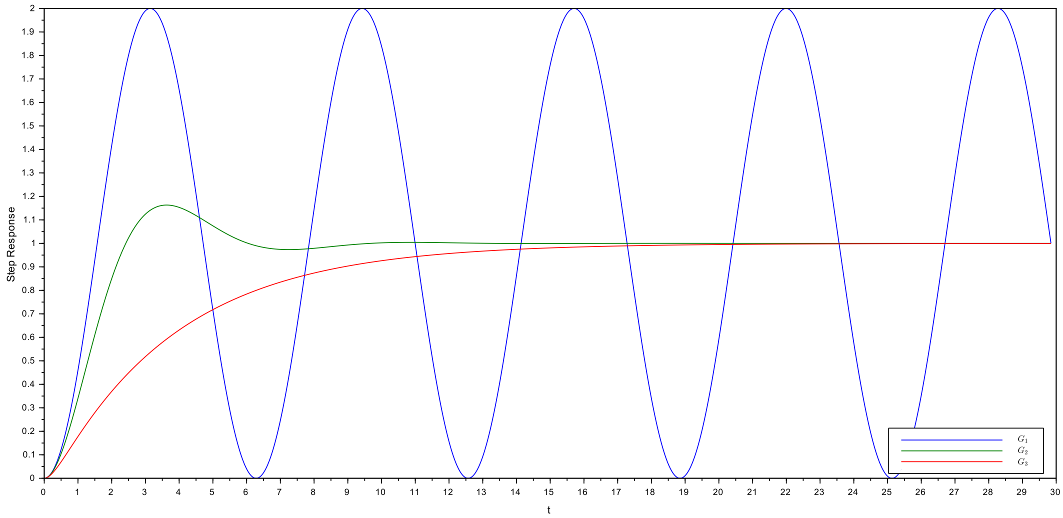


Figure 4.1: Variation in Step Response of $G_1(s)$, $G_2(s)$ & $G_3(s)$

System	% Overshoot	Peak Time (in s)	Delay Time (in s)	Rise Time (in s)	2% Settling Time (in s)
G_1	100%	3.142	1.047	1.020	Undefined
G_2	16.30%	3.628	1.294	1.638	8.076
G_3	0%	Undefined	2.864	8.217	14.812

Observations

With increase in Damping Ratio ζ (assuming same ω_n)

% Overshoot: value decreases from 100% at $\zeta = 0$ to 0 at $\zeta = 1$ and there is no % Overshoot for $\zeta > 1$

Peak Time: increases as ζ increases and it is only defined $\zeta < 1$ (for undamped & underdamped systems)

Delay Time: increases as ζ increases

Rise Time: increases as ζ increases

2% Settling Time: decreases as $\zeta = 0$ to $\zeta = 1$ then increases at higher ζ (undefined for $\zeta = 0$)

To find the last time the underdamped system was 1

```
s = %s;
tStep = 0.0001;
tMax = 30;
t = 0:tStep:tMax;

G_1 = 1 / (s^2 + 1);
sys1 = syslin('c',G_1);
gp1 = csim('step', t, sys1);
max(find(abs(gp1-1) <= 1e-4))*tStep
```

We get 29.8453, we will use this in below code. Now, the last ‘vector’ value is approximately steady state value for all 2^{nd} order step responses (including undamped)

Code to obtain Figure 4.1

```
--> s = %s;
--> tStep = 0.0001;
--> tMax = 29.8452;
--> t = 0:tStep:tMax;
--> G = [1 / (s^2 + 1); 1 / (s^2 + s + 1); 1 / (s^2 + 4*s + 1)];
--> y = [];
--> legendValues = [];
--> dims = 1;
--> for x = 1:length(G)
-->     str = "$G_{" + string(x) + "}$";
-->     legendValues = cat(dims, legendValues, str);
-->     sys = syslin('c',G(2)(x),G(3)(x));
-->     gp = csim('step', t, sys);
-->     y = cat(dims, y, csim('step', t, sys));
-->     disp(str+" =", percentOvershoot(gp));
-->     disp(peakTime(gp,tStep), delayTime(gp,tStep), riseTime(gp,tStep), settlingTime(gp,tStep));
--> end
--> plot(t',y'); xlabel(" t" ); ylabel(" Step Response" );
--> legend(legendValues,opt=4);
--> xs2pdf(0,'Q4');
  "$G_{1}$ ="
  100.0139591548
  3.1417
  1.047300000000
  1.0196
  29.82530000000 (this is undefined behaviour)

  "$G_{2}$ ="
  16.30339706800
  3.627700000000
  1.2942
  1.637700000000
  8.0764

  "$G_{3}$ ="
  0.
  -0.0001 (this is undefined behaviour)
  2.8637
  8.2173
  14.8123
```

5 References

2^{nd} order approximation