

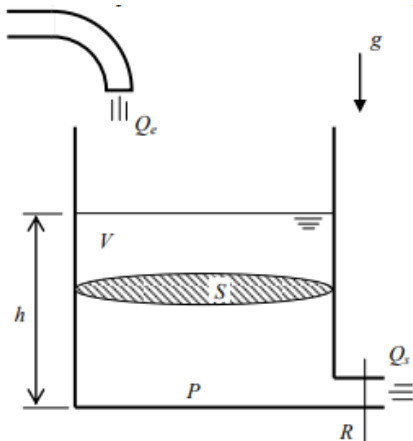
# PME3380 - Lista B

Aluno - Bruno Akira Oshiro

Nº USP - 10771667

## 1. Reservatório único

Para o primeiro exercício considerou-se o reservatório esquematizado a seguir:



Reservatório com água

Parâmetros:

$S = 10 \text{ m}^2$  - área da seção transversal (constante)

$R = 2 \times 10^8 \text{ Pa}/(\text{m}^3/\text{s})^2$  - parâmetro que relaciona vazão com queda de pressão (perda de carga)

$\rho = 1000 \text{ kg}/\text{m}^3$  - massa específica da água

$G = 10 \text{ m}/\text{s}^2$  - aceleração da gravidade na superfície da terra

Variáveis:

$Q_e = 0,010247 \text{ m}^3/\text{s}$  - vazão de entrada

$h$ : nível do reservatório [m]

$V$ : volume de água no reservatório [ $\text{m}^3$ ]

$P$ : pressão relativa à atmosférica, no fundo do reservatório [Pa]

$Q_s$ : vazão de saída [ $\text{m}^3/\text{s}$ ]

Admite-se que a água seja incompressível.

Além disso, assim como demonstrado no enunciado, considerou-se:

$$Q_s = \sqrt{\frac{P}{R}} \quad \text{Vazão de saída}$$

$$P = \rho gh \quad \text{Pressão no ponto mais baixo}$$

$$\dot{h} = \frac{\left(-\sqrt{\frac{\rho gh}{R}} + Q_e\right)}{S} \quad \text{derivada de } h$$

Assim, resolveu-se o problema através de métodos numéricos: Euler e Runge Kutta

### 1.1. Método de Euler

Conhecendo a derivada e o valor inicial de  $h$ , podemos calcular o método de Euler da seguinte maneira:

$$h(i+1) = h(i) + dt * \dot{h}(h(i))$$

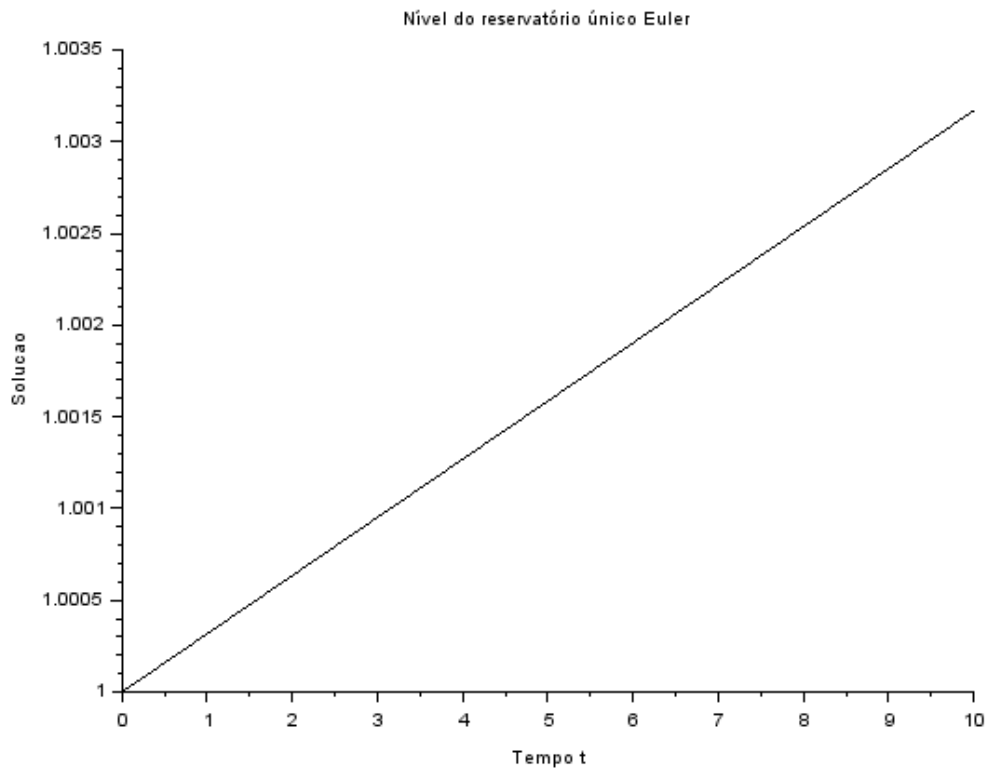
## PME3380 - Lista B

Aluno - Bruno Akira Oshiro

Nº USP - 10771667

---

O resultado obtido para  $h_0 = 1$  foi o seguinte:



### 1.2. Método de Runge-Kutta 4ª ordem

Assim como o método de Euler, por meio da derivada de  $h$ , pode-se implementar uma solução numérica pelo método de runge-kutta da seguinte maneira:

$$\begin{aligned}K_1(i) &= \dot{h}(h(i)) \\K_2(i) &= \dot{h}\left(h(i) + \frac{dt}{2} * K_1\right) \\K_3(i) &= \dot{h}\left(h(i) + \frac{dt}{2} * K_2\right) \\K_4(i) &= \dot{h}\left(h(i) + dt * K_3\right) \\h(i + 1) &= h(i) + \frac{dt}{6} (K_1 + 2K_2 + 2K_3 + K_4)\end{aligned}$$

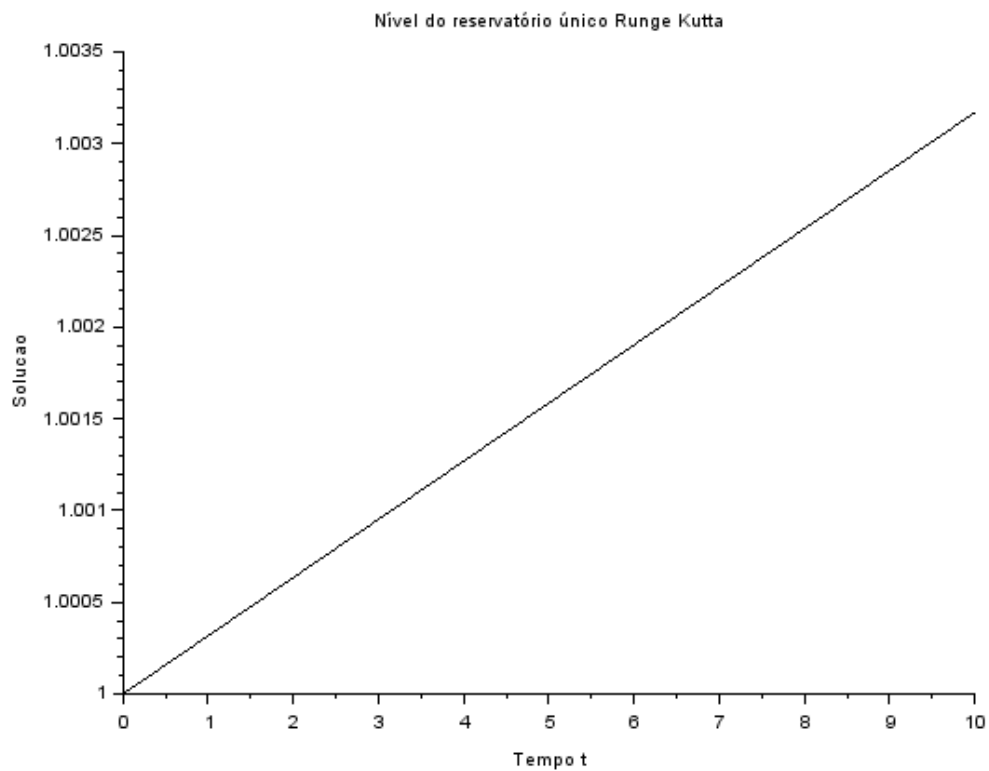
## PME3380 - Lista B

Aluno - Bruno Akira Oshiro

Nº USP - 10771667

---

O resultado, para  $h_0 = 1$ , encontrado foi o seguinte:



Percebe-se que o resultado é muito semelhante ao encontrado no método de Euler. Dessa forma, valida-se os programas utilizados e seus resultados.

## PME3380 - Lista B

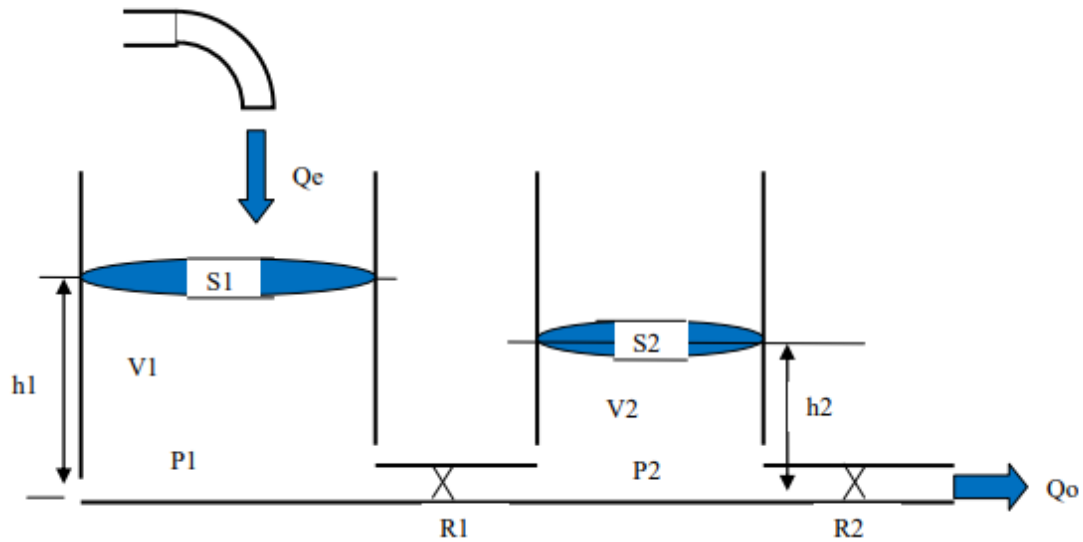
Aluno - Bruno Akira Oshiro

Nº USP - 10771667

---

### 2. Dois reservatórios unidos

No segundo exercício resolve-se o problema para o seguinte esquema:



Os parâmetros considerados forma replicados do primeiro exercício para os dois tanques:

$$\begin{aligned}Q_e &= 0.010247 \\ \rho &= 1000 \\ g &= 10 \\ R_a &= 2 * 10^8 \\ R_s &= 2 * 10^8 \\ S_1 &= 10 \\ S_2 &= 10\end{aligned}$$

Por meio do enunciado também sabemos que:

$$h_1 = \frac{Q_e - \sqrt{\rho \frac{g}{R_a}} (h_1 - h_2)}{S_1}$$
$$h_2 = \frac{\sqrt{\rho \frac{g}{R_a}} (h_1 - h_2) - \sqrt{\rho g \frac{h_2}{R_s}}}{S_2}$$

Sabendo as derivadas, pode-se calcular as soluções numéricas

## PME3380 - Lista B

Aluno - Bruno Akira Oshiro

Nº USP - 10771667

---

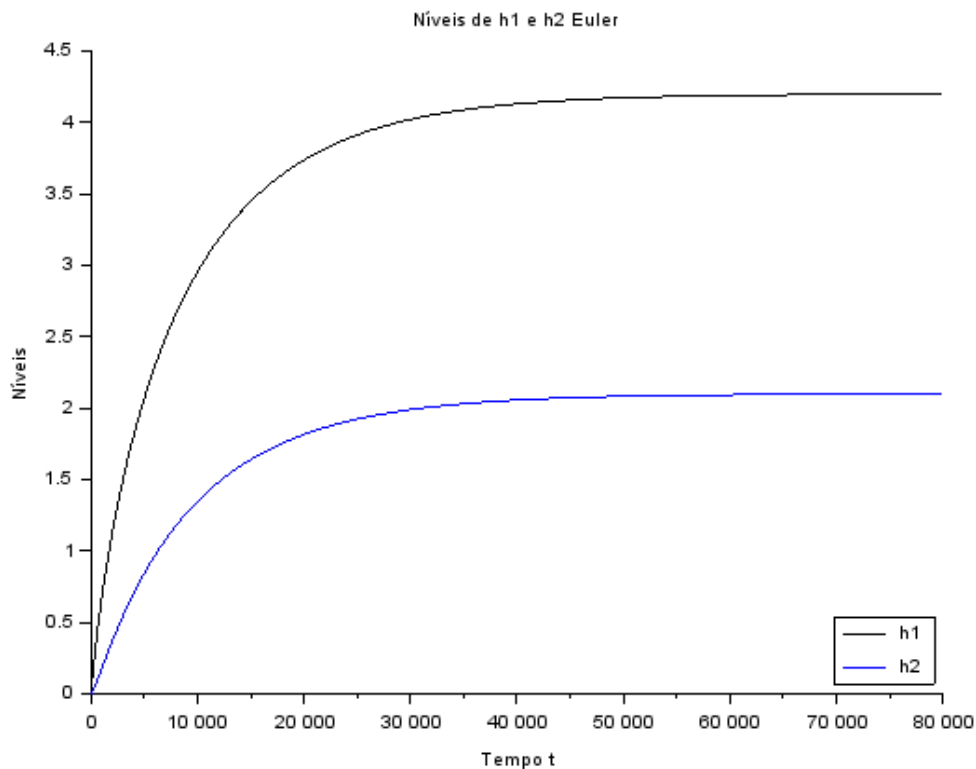
### 2.1. Método de Euler

O método de Euler não possui grandes modificações em relação ao primeiro caso. Basicamente, é realizado o mesmo processo nas duas equações.

$$h_1(i+1) = h_1(i) + \frac{Q_e - \sqrt{\rho \frac{g}{R_a} (h_1(i) - h_2(i))}}{S1} * dt$$

$$h_2(i+1) = h_2(i) + \frac{\sqrt{\rho \frac{g}{R_a} (h_1(i) - h_2(i))} - \sqrt{\rho g \frac{h_2(i)}{R_s}}}{S2} * dt$$

Os resultados encontrados para  $h_{1_0} = 0$  e  $h_{2_0} = 0$  são mostrados a seguir:



Observa-se que a Vazão de entrada é maior que a de saída do primeiro tanque. E também vale notar que após algum tempo as alturas se estabilizam. Ou seja a partir de, aproximadamente, 30.000 segundos as vazões de entrada e saída se igualam.

## PME3380 - Lista B

Aluno - Bruno Akira Oshiro

Nº USP - 10771667

---

### 2.2. Método de Runge-Kutta 4ª ordem

O método de runge-kutta tem algumas peculiaridades nesse exercício. Todos os coeficientes do método devem ser calculados para as duas variáveis simultaneamente, pois elas são dependentes entre si. Assim, calcula-se a solução numérica da seguinte maneira:

$$K_{1h_1}(i) = \dot{h}_1(h_1(i), h_2(i))$$

$$K_{1h_2}(i) = \dot{h}_2(h_1(i), h_2(i))$$

$$K_{2h_1}(i) = \dot{h}_{h_1}(h_{h_1}(i) + \frac{dt}{2} * K_{1h_1}, h_{h_2}(i) + \frac{dt}{2} * K_{1h_2})$$

$$K_{2h_2}(i) = \dot{h}_{h_2}(h_{h_1}(i) + \frac{dt}{2} * K_{1h_1}, h_{h_2}(i) + \frac{dt}{2} * K_{1h_2})$$

$$K_{3h_1}(i) = \dot{h}_{h_1}(h_{h_1}(i) + \frac{dt}{2} * K_{2h_1}, h_{h_2}(i) + \frac{dt}{2} * K_{2h_2})$$

$$K_{3h_2}(i) = \dot{h}_{h_2}(h_{h_1}(i) + \frac{dt}{2} * K_{2h_1}, h_{h_2}(i) + \frac{dt}{2} * K_{2h_2})$$

$$K_{4h_1}(i) = \dot{h}_{h_1}(h_{h_1}(i) + dt * K_{3h_1}, h_{h_2}(i) + dt * K_{3h_2})$$

$$K_{4h_2}(i) = \dot{h}_{h_2}(h_{h_1}(i) + dt * K_{3h_1}, h_{h_2}(i) + dt * K_{3h_2})$$

$$h_1(i + 1) = h_1(i) + \frac{dt}{6} (K_{1h_1} + 2K_{2h_1} + 2K_{3h_1} + K_{4h_1})$$

$$h_2(i + 1) = h_2(i) + \frac{dt}{6} (K_{1h_2} + 2K_{2h_2} + 2K_{3h_2} + K_{4h_2})$$

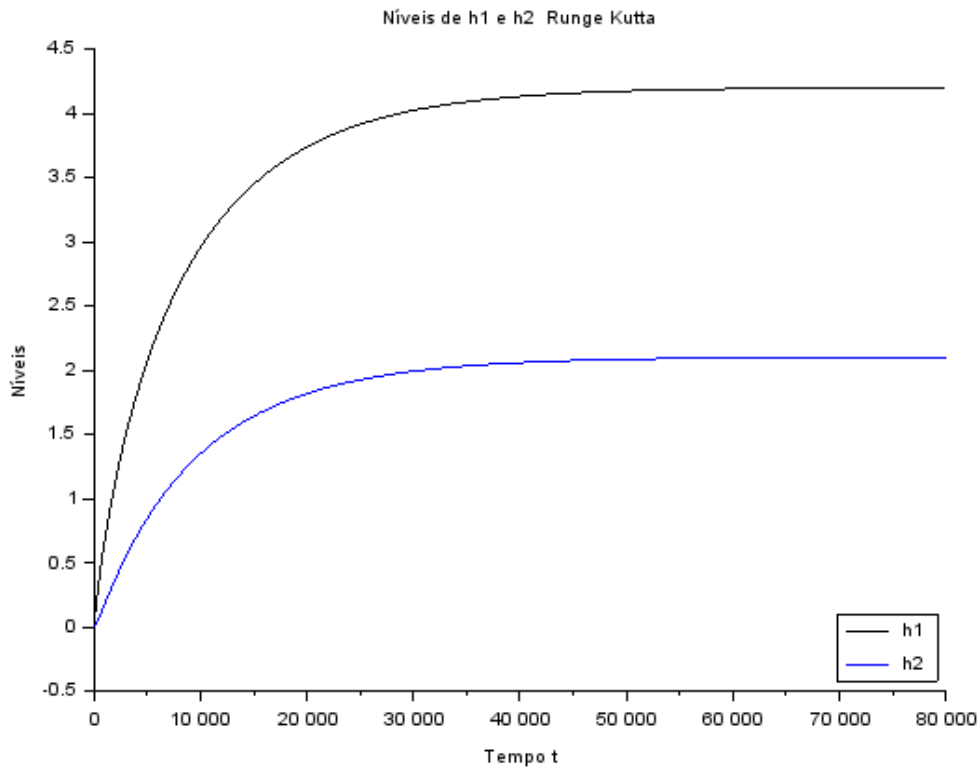
# PME3380 - Lista B

Aluno - Bruno Akira Oshiro

Nº USP - 10771667

---

Os resultados para  $h_{1_0} = 0$  e  $h_{2_0} = 0$  são mostrados a seguir:



Observa-se que o resultado é muito semelhante ao encontrado pelo método de Euler. Dessa forma, validam-se os programas utilizados.

## 3. Códigos

### 3.1. Euler para o exercício 1

```
4. // Conjunto de comandos para solucao numerica de equacao diferencial dada pela funcao funcao.sci
5. // Apagando dados anteriores:
6. clear
7. // Carregando a equacao diferencial:
8. function [ydot]=funcao(y)
9. ydot=(-sqrt(rho*g*y/R)+Qe)*1/S;
10. endfunction
11.
12. //parâmetros do exercício
13. Qe=0.010247
14. rho=1000
15. g=10
16. R=2*10^8
17. S=10
18.
```

## PME3380 - Lista B

Aluno - Bruno Akira Oshiro

Nº USP - 10771667

---

```
19. // Instante inicial:
20. t(1)=0;
21. // Instante final:
22. tf=10;
23. // Condicao inicial:
24. y(1)=1;
25. // Valor inicial da solucao exata:
26. h=0.5;
27. // Calculo de numero de passos):
28. n=round(tf/h);
29. // Integracao numerica usando o metodo de Euler:
30. // Comando for:
31.
32. for i=1:n
33. // Vetor de tempo:
34. t(i+1)=t(i)+h;
35. //pressão
36. P=rho*g*y(i);
37. //Vazao de saída
38. Qs=sqrt(P/R);
39. // Solucao numerica:
40. y(i+1)=y(i)+h* funcao(y(i));
41.
42. // Termina do comando for:
43. end
44. // Plotando solucao numerica y versus vetor de tempo t e solucao exata ye versus vetor de tempo t:
45. plot2d(t,y);
46. // Colocando um titulo na figura e nomeando os eixos:
47. xtitle("Nível do reservatório único Euler","Tempo t","Solucao")
```

### 3.2. Runge-Kutta para o exercício 1

```
4. // Conjunto de comandos para solucao numerica de equacao diferencial [1-y(i)]/2
5. // Apagando dados anteriores:
6. clear
7. function [ydot]=funcao(y)
8. ydot=(-sqrt(rho*g*y/R)+Qe)*1/S;
9. endfunction
10.
11. //parâmetros do exercício
12. Qe=0.010247
13. rho=1000
14. g=10
15. R=2*10^8
16. S=10
17.
18. // Instante inicial:
19. t(1)=0;
20. // Instante final:
21. tf=10;
22. // Condicao inicial:
23. y(1)=1;
24. // Passo de integracao (experimente alterar o passo):
25. h=0.5;
26. // Calculo de numero de passos):
27. n=round((tf-t(1))/h);
28. // Integracao numerica usando o metodo de Runge Kutta:
29. // Comando for:
```



## PME3380 - Lista B

Aluno - Bruno Akira Oshiro

Nº USP - 10771667

---

```
30. for i=1:n
31. // Vetor de tempo:
32. t(i+1)=t(i)+h;
33. // Solucao numerica:
34. k1=funcao(y(i));
35. k2=funcao(y(i)+k1*h/2);
36. k3=funcao(y(i)+k2*h/2);
37. k4=funcao(y(i)+k3*h);
38.
39. y(i+1)=y(i)+(h*(k1+2*k2+2*k3+k4)/6);
40. // Termina do comando for:
41. end
42. // Plotando solucao numerica y versus vetor de tempo t e solucao exata ye versus vetor de tempo t:
43. plot2d(t,y);
44. // Colocando um titulo na figura e nomeando os eixos:
45. xtitle("Nível do reservatório único Runge Kutta","Tempo t","Solucao")
```

### 3.3. Euler para o exercício 2

```
4. // Conjunto de comandos para solucao numerica de equacao diferencial dada pela funcao funcao.sci
5. // Apagando dados anteriores:
6. clear
7. function [h1_dot,h2_dot]=funcao(h1,h2)
8. h1_dot=(Qe-sqrt(rho*g/Ra*(h1-h2)))/S1
9. h2_dot=(sqrt(rho*g/Ra*(h1-h2))-sqrt(rho*g*h2/Rs))/S2
10. endfunction
11.
12. // Instante final:
13. tf=80000;
14. // Condicao inicial:
15. h1(1)=0;
16. h2(1)=0;
17.
18. //parâmetros do exercício
19. Qe=0.010247
20. rho=1000
21. g=10
22. Ra=2*10^8
23. Rs=2*10^8
24. S1=10
25. S2=10
26.
27. // Instante inicial:
28. t(1)=0;
29. // Passo de integracao (experimente alterar o passo):
30. h=0.25;
31. // Calculo de numero de passos:
32. n=round(tf/h);
33.
34. // Integracao numerica usando o metodo de Euler:
35. // Comando for:
36. for i=1:n
37. // Vetor de tempo:
38. t(i+1)=t(i)+h;
39. //calculando as derivada
40. [h1_dot,h2_dot]=funcao(h1(i),h2(i))
41. // Solucao numerica:
42. h1(i+1)=h1(i)+h* h1_dot;
```

## PME3380 - Lista B

Aluno - Bruno Akira Oshiro

Nº USP - 10771667

---

```
43. h2(i+1)=h2(i)+h* h2_dot;
44. // Termino do comando for:
45. end
46. plot2d(t,h1)
47. // Plotando solucao numerica y versus vetor de tempo t e solucao exata ye versus vetor de tempo t:
48. plot2d(t,[h1,h2]);
49. // Colocando uma legenda na parte inferior direito da figura (parametro 4):
50. legends(["h1","h2"],[1 2],4)
51. // Colocando um titulo na figura e nomeando os eixos:
52. xtitle("Níveis de h1 e h2 Euler","Tempo t","Níveis")
```

### 3.4. Runge-Kutta para o exercício 2

```
4. // Conjunto de comandos para solucao numerica de equacao diferencial [1-y1(i)]/2
5. // Apagando dados anteriores:
6. clear
7.
8. //funções que calculam as derivadas
9. function [h1_dot]=Dh1(h1, h2)
10. h1_dot=(Qe-sqrt(rho*g/Ra*(h1-h2)))/S1
11. endfunction
12.
13. function [h2_dot]=Dh2(h1, h2)
14. h2_dot=(sqrt(rho*g/Ra*(h1-h2))-sqrt(rho*g*h2/Rs))/S2
15. endfunction
16.
17. // Instante final:
18. tf=80000;
19. // Condicao inicial:
20. h1(1)=0;
21. h2(1)=0;
22.
23. //parâmetros do exercício
24. Qe=0.010247
25. rho=1000
26. g=10
27. Ra=2*10^8
28. Rs=2*10^8
29. S1=10
30. S2=10
31.
32. // Instante inicial:
33. t(1)=0;
34. // Passo de integracao (experimente alterar o passo):
35. h=0.5;
36. // Calculo de numero de passos:
37. n=round((tf-t(1))/h);
38. // Integracao numerica usando o metodo de Runge Kutta:
39. // Comando for:
40. for i=1:n
41. // Vetor de tempo:
42. t(i+1)=t(i)+h;
43. // Solucao numerica:
44. k1_h1=Dh1(h1(i),h2(i));
45. k1_h2=Dh2(h1(i),h2(i));
46.
47. k2_h1=Dh1(h1(i)+k1_h1*h/2,h2(i)+k1_h2*h/2);
48. k2_h2=Dh2(h1(i)+k1_h1*h/2,h2(i)+k1_h2*h/2);
```

## PME3380 - Lista B

Aluno - Bruno Akira Oshiro

Nº USP - 10771667

---

```
49.
50. k3_h1=Dh1(h1(i)+k2_h1*h/2,h2(i)+k2_h1*h/2);
51. k3_h2=Dh2(h1(i)+k2_h1*h/2,h2(i)+k2_h1*h/2);
52.
53. k4_h1=Dh1(h1(i)+k3_h1*h,h2(i)+k3_h1*h);
54. k4_h2=Dh2(h1(i)+k3_h1*h,h2(i)+k3_h1*h);
55.
56. h1(i+1)=h1(i)+(h*(k1_h1+2*k2_h1+2*k3_h1+k4_h1)/6);
57. h2(i+1)=h2(i)+(h*(k1_h2+2*k2_h2+2*k3_h2+k4_h2)/6);
58.
59. // Termino do comando for:
60. end
61.
62. // Plotando solucao numerica y versus vetor de tempo t e solucao exata ye versus vetor de tempo t:
63. plot2d(t,[h1,h2]);
64. // Colocando uma legenda na parte inferior direito da figura (parametro 4):
65. legends(["h1","h2"],[1 2],4)
66. // Colocando um titulo na figura e nomeando os eixos:
67. xtitle("Níveis de h1 e h2 Runge Kutta","Tempo t","Níveis")
```