2.2 Método da Bissecção

```
DADOS a0 , b0  : f(a0 ).f(b0 )<0
    PARA n=0,1,2,... ATÉ ONDE FOR CONVENIENTE
        SEJA m-(an +bn )/2
        SE f(an ).f(m )<0  SEJAM an+1 = an ; bn+1 = m
        CASO CONTRÁRIO SEJAM an+1 = m; bn+1 = bn
```

```
se   ( f(a) <= 0   E  f(m) >= 0 )  OU  ( f(a) > 0  E  f(m) <0)   então
raiz está no intervalo [ a, m ]
caso contrário
raiz está no intervalo ] m , b ]
```

2.3 Método da Corda

```
Algoritmo da corda
DADOS a0 , b0  : f(a0 ).f(b0 )<0
    PARA n=0,1,2,... ATÉ ONDE FOR CONVENIENTE
        SEJA w =(an*f(bn) - bn*f(an))/(f(bn)-f(an))
        SE f(an ).f(w )<0  SEJAM an+1 = an ; bn+1 = w
        CASO CONTRÁRIO SEJAM an+1 = w; bn+1 = bn
```

2.4 Método da tangente

método da tangente ou *método de Newton*

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

2.5 Método de iteração de Picard-Peano

$x = g(x)$ $\quad |g'(x)| < 1$ $\quad x_{k+1} = g(x_k)$

3.1 Eliminação Gaussiana

Condensar a Matriz e resolver sistema

```
    DADOS W(n,n+1)= A(n,n);b(n)
    INICIALIZE-SE p DE MODO QUE p(i) = i   (i = 1, 2, ..., n)
          PARA k =  1, 2, ..., n-1 {
                  ACHAR O MENOR i >=  k TAL QUE W(i,k) != 0
                  SE NÃO EXISTIR, A NÃO É INVERTÍVEL; PARAR
                  CASO CONTRÁRIO, TROCAR p(k)  COM p(i)  E AS LINHAS k E I DE W
                  PARA i = k+1, ..., n {
                          SEJAM m = W(k,k) , W(i,k) = W(i,k)/m
                                  PARA j = k+1, ..., n+1 {
                                          SEJA W(i,j) = W(i,j) - m. W(k,j)
                                  }
                  }
          }
  SE W(n,m)= 0 , A NÃO É INVERTÍVEL; PARAR
  SEJA x(n) = b(n)
  PARA k = n-1, ..., 1 {
      SEJA s= 0
      PARA j = k+1, ..., n {
          SEJA s = s + W(k,j).x(j)
      }
          SEJA x(k) = (b(k) - s)/W(k,k)
  }
```

Exemplo: 33.6 Estabilidade externa

Seja o sistema

$$\begin{cases} 7x+8y+9z=24 \\ 8x+9y+10z=27 \\ 9x+10y+8z=27 \end{cases}$$

e resolvamo-lo em **Maxima** pelo método de eliminação gaussiana, usando a colecção de funções para álgebra linear[a]:

```
(%i1) load(linearalgebra);
```

mediante os comandos `echelon(M)`[b] e `rowop(M,i,j,t)`[c]:

```
(%i2) A:matrix(
 [7,8,9],
 [8,9,10],
 [9,10,8]
);
```

(%o2) $$\begin{pmatrix} 7 & 8 & 9 \\ 8 & 9 & 10 \\ 9 & 10 & 8 \end{pmatrix}$$

```
(%i3) B:matrix(
 [24],
 [27],
 [27]
);
```

(%o3) $$\begin{pmatrix} 24 \\ 27 \\ 27 \end{pmatrix}$$

```
(%i4) AB:addcol(A,B);
```

(%o4) $$\begin{pmatrix} 7 & 8 & 9 & 24 \\ 8 & 9 & 10 & 27 \\ 9 & 10 & 8 & 27 \end{pmatrix}$$

```
(%i5) AB:echelon(AB);
```

(%o5) $$\begin{pmatrix} 1 & \frac{8}{7} & \frac{9}{7} & \frac{24}{7} \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

```
(%i6) AB:rowop(AB,2,3,2);
```

(%o6) $$\begin{pmatrix} 1 & \frac{8}{7} & \frac{9}{7} & \frac{24}{7} \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

```
(%i7) AB:rowop(AB,1,3,9/7);
```

(%o7) $$\begin{pmatrix} 1 & \frac{8}{7} & 0 & \frac{15}{7} \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

```
(%i8) AB:rowop(AB,1,2,8/7);
```

(%o8) $$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

A solução é, portanto,

$$\begin{cases} x = 1 \\ y = 1 \\ z = 1 \end{cases}$$

e a estabilidade externa pode ser estimada mediante:

$$A.\delta x = \delta b - \delta A.x$$

em que δa é o erro absoluto dos coeficientes da equação e δb o dos segundos membros. Calculando mais uma vez em **Maxima**

```
(%i9) DA:zeromatrix(3,3)+da;
```

(%o9) $$\begin{pmatrix} da & da & da \\ da & da & da \\ da & da & da \end{pmatrix}$$

```
(%i10) DB:zeromatrix(3,1)+db;
```

(%o10) $$\begin{pmatrix} db \\ db \\ db \end{pmatrix}$$

```
(%i11) X:col(AB,4);
```

(%o11) $$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

```
(%i12) BP:DB-DA.X;
```

(%o12) $$\begin{pmatrix} db-3da \\ db-3da \\ db-3da \end{pmatrix}$$

```
(%i13) AP:addcol(A,BP);
```

Exemplo: 33.6 Estabilidade externa (cont.)

(%o13)
$$\begin{pmatrix} 7 & 8 & 9 & db-3\,da \\ 8 & 9 & 10 & db-3\,da \\ 9 & 10 & 8 & db-3\,da \end{pmatrix}$$

```
(%i14) AP:echelon(AP);
```

(%o14)
$$\begin{pmatrix} 1 & \frac{8}{7} & \frac{9}{7} & \frac{db-3da}{7} \\ 0 & 1 & 2 & db-3\,da \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

```
(%i15) AP:rowop(AP,2,3,2);
```

(%o15)
$$\begin{pmatrix} 1 & \frac{8}{7} & \frac{9}{7} & \frac{db-3da}{7} \\ 0 & 1 & 0 & db-3\,da \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

```
(%i16) AP:rowop(AP,1,3,9/7);
```

(%o16)
$$\begin{pmatrix} 1 & \frac{8}{7} & 0 & \frac{db-3da}{7} \\ 0 & 1 & 0 & db-3\,da \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

```
(%i17) AP:rowop(AP,1,2,8/7);
```

(%o17)
$$\begin{pmatrix} 1 & 0 & 0 & 3\,da-db \\ 0 & 1 & 0 & db-3\,da \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

teremos, portanto,

$$\begin{cases} \delta x = 3\delta A - \delta b \\ \delta y = -3\delta A + \delta b \\ \delta z = 0 \end{cases}$$

teremos, portanto,

$$\begin{cases} \delta x = 3\delta A - \delta b \\ \delta y = -3\delta A + \delta b \\ \delta z = 0 \end{cases}$$

o que mostra o facto surpreendente de (para este sistema particular!) a incógnita z ser insensível a (pequenos!) erros dos dados. Se supusermos que $\delta A = \delta b = 0.5$ teremos

$$\begin{cases} \delta x = 1 \\ \delta y = -1 \\ \delta z = 0 \end{cases}$$

[a] ver ficheiro SISTEMAS-estabilidade

[b] `echelon(M)` devolve uma matriz triangular superior de diagonal unitária obtida por eliminação gaussiana a partir da matriz `M`

[c] `rowop(M,i,j,t)` opera sobre a linha `i` da matriz `M` por $R_i \leftarrow R_i - t * R_j$

3.6.1 Método de Gauss-Jacobi

$$x_i^{(k)} = \frac{1}{a_{ii}} \left[-\sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{(k-1)} + b_i \right] \quad , \quad (1 \leq i \leq n)$$

X1(n+1) = (D1-B1*X2n – C1*X3n)/A1

…

3.6.2 Método de Gauss-Seidel

$$x_i^{(k)} = \frac{1}{a_{ii}} \left[-\sum_{\substack{j=1\\ j<i}}^{n} a_{ij}x_j^{(k)} - \sum_{\substack{j=1\\ j>i}}^{n} a_{ij}x_j^{(k-1)} + b_i \right] \quad , \quad (1 \le i \le n)$$

Método idêntico ao Gauss-Jacobi porém já se utiliza os valores já calculado atrás.

X1(n+1) = (D1-B1*X2n-C1*X3n)/A1

X2(n+1) = (D2-A2*X1(n+1) – C2*X3n)/B2

X3(n+1) = (D3-A3*X1(n+1) – B3*X2(n+1))/C3

4.2 Regra dos Trapézios

$$\int_{x_0}^{x_n} y.dx = \frac{h}{2} \times \left[y_0 + y_n + 2 \times \sum_{\substack{i=1\\ i=i+2}}^{n-1} y_i \right]$$

Sendo y a função. yi = F(a + i*h)

Ter em atenção que é duas vezes yn

Coeficiente de convergência:

$$\frac{S' - S}{S'' - S'}$$

Tem de ser aproximadamente igual a 2^ordem do método

Erro cometido:

$$\varepsilon'' \approx \frac{S'' - S'}{}$$

k-> 2^(ordem do método) - 1

4.3 Regra de Simpson

$$\int_{x_0}^{x_{2n}} y.dx = \frac{h}{3}.\left[y_0 + y_{2n} + 4\times\sum_{\substack{i=1\\ i=i+2}}^{2n-1} y_i + 2\times\sum_{\substack{i=2\\ i=i+2}}^{2n-2} y_i\right]$$

O número de intervalos tem de ser sempre par (N = (b – a) / h)

Y2n = 2*y2n

Termos pares (i par) : 2*f(a + i*h)
Termos ímpares (i ímpar): 4*f(a + i*h)

4.6 Cubatura

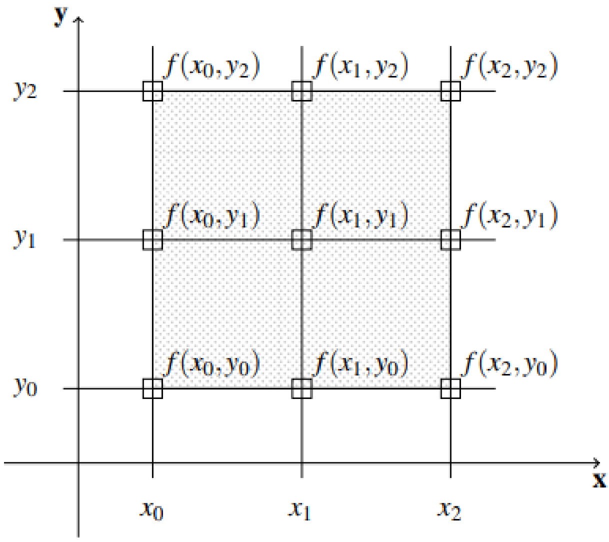

Figura 4.1: Pontos usados na Regra de Simpson

com

$$h_x = \frac{A-a}{2} \qquad h_y = \frac{B-b}{2}$$

A fórmula de Simpson da cubatura é, portanto, da forma:

$$\iint f(x,y)\,.dx.dy = \frac{h_x.h_y}{9}.\left[\Sigma_0 + 4\Sigma_1 + 16\Sigma_2\right]$$

em que:

$\Sigma_0 =$ soma dos valores de f nos vértices da malha
$\Sigma_1 =$ soma dos valores de f nos pontos médios dos lados da malha
$\Sigma_2 =$ valor de f no centro da malha

Cubatura pela forma tos trapézios:

com

$$h_x = \frac{A-a}{2} \qquad h_y = \frac{B-b}{2}$$

A fórmula de [redacted] da cubatura é, portanto, da forma:

$$\iint f(x,y)\,.dx.dy = \frac{h_x.h_y}{4}.\,[\Sigma_0 + 2\Sigma_1 + 4\Sigma_2]$$

em que:

Σ_0 = soma dos valores de f nos vértices da malha
Σ_1 = soma dos valores de f nos pontos médios dos lados da malha
Σ_2 = valor de f no centro da malha

5.2 Método de Euler

1. Calcular o incremento de y usando y'_n que é a inclinação da curva no ponto (x_n, y_n), e Δx_n, o incremento de x no intervalo n:

$$\Delta y_n = f(x_n, y_n)\,\Delta x_n$$

2. Calcular o ponto seguinte:

$$\begin{cases} y_{n+1} & \leftarrow & y_n + \Delta y_n \\ x_{n+1} & \leftarrow & x_n + \Delta x_n \end{cases}$$

e repetir esses passos até cobrir por completo o intervalo que nos interessa.

Onde Δx_n = h

$$\frac{S'-S}{S''-S'} \approx 2 \qquad\qquad S''-S' \approx \varepsilon''$$

5.3 Um melhoramento do Método de Euler

Calcular declive

$$y'_n = f(x_n, y_n)$$

Calculando em seguida um *ponto previsto*:

$$p_{n+1} = y_{n-1} + 2y'_n \times \Delta x_n$$

Calcular a derivada nesse ponto previsto

$$p'_{n+1} = f(x_{n+1}\ ,\ p_{n+1})$$

calculando em seguida o *incremento corrigido* de y em x_{n+1}

$$\Delta y_n = \frac{p'_{n+1} + y'_n}{2} \times \Delta x_n$$

aplicando no fim as expressões recorrentes para obter o novo ponto:

$$\begin{cases} y_{n+1} & \leftarrow & y_n + \Delta y_n \\ x_{n+1} & \leftarrow & x_n + \Delta x_n \end{cases}$$

5.4.1 Método de Runge-Kuta de Segunda Ordem

1. começa-se por calcular y' no início do intervalo: $y'_n = f(x_n, y_n)$
2. a partir daí constrói-se uma estima de y' no meio do intervalo
$$y'_a = f\left(x_n + \frac{\Delta x_n}{2}\ ,\ y_n + \frac{h}{2}.y'_n\right)$$
3. Calcular o incremento de y usando y'_a , e Δx_n, o incremento de x no intervalo n:
$$\Delta y_{n+1} = y'_a\,\Delta x_n$$
4. Calcular o ponto seguinte:
$$\begin{cases} y_{n+1} & \leftarrow & y_n + \Delta y_n \\ x_{n+1} & \leftarrow & x_n + \Delta x_n \end{cases}$$
e repetir esses passos até cobrir por completo o intervalo que nos interessa.

$$\frac{S' - S}{S'' - S'} \approx 4 \qquad S'' - S' \approx 3.\varepsilon''$$

5.4.2 Método de Runge-Kuta de Quarta Ordem

1. começa-se por, usando y'_n no início do passo de integração, calcular uma primeira estima δ_1 para o incremento de Δy_n ;
$$\delta_1 = \Delta x_n.f\,(x_n\ ,\ y_n)$$
2. a partir desta estima, calcula-se uma primeira estima dey' no meio do passo e, a partir deste valor, uma segunda estima δ_2 para o incremento de Δy_n :
$$\delta_2 = \Delta x_n.f\left(x_n + \frac{\Delta x_n}{2}\ ,\ y_n\ + \frac{\delta_1}{2}\right)$$
3. calcula-se uma segunda estima de y'_n no meio do passo, usando a anterior, e a partir deste valor, uma terceira estima δ_3 para o incremento de Δy_n :
$$\delta_3 = \Delta x_n.f\left(x_n + \frac{\Delta x_n h}{2}\ ,\ y_n + \frac{\delta_2}{2}\right)$$
4. finalmente, obtém-se uma última estima de y'_n no fim do passo, usando a anterior, e a partir deste valor, uma quarta estima δ_4 para o incremento de Δy_n :
$$\delta_4 = \Delta x_n.f\,(x_n + \Delta x_n\ ,\ y_n + \delta_3)$$
5. calcula-se a média ponderada das diversas estimas δ_i estabelecendo os pesos de modo a garantir um erro proporcional a $(\Delta x_n)^4$
$$\Delta y_n = \frac{\delta_1}{6} + \frac{\delta_2}{3} + \frac{\delta_3}{3} + \frac{\delta_4}{6}$$
6. e usa-se essa média para calcular o valor y_{n+1} em x_{n+1}
7. Calcular o ponto seguinte:
$$\begin{cases} y_{n+1} & \leftarrow & y_n + \Delta y_n \\ x_{n+1} & \leftarrow & x_n + \Delta x_n \end{cases}$$

e repetir esses passos até cobrir por completo o intervalo que nos interessa.

5.6 Sistemas de Equações e Equações de Ordem Superior

Exemplo: 55.8 Sistema de equações diferenciais

Vamos integrar o sistema de equações diferenciais

$$\begin{cases} z' = & x+y+z \\ y' = & 2.x-y-z \end{cases}$$

usando o *método de Euler*, com as expressões recorrentes

$$\begin{cases} z_{n+1} \leftarrow & z_n + (x_n + y_n + z_n).\Delta x_n \\ y_{n+1} \leftarrow & y_n + (2.x_n - y_n - z_n).\Delta x_n \\ x_{n+1} \leftarrow & x_n + \Delta x_n \end{cases}$$

começando em $(x_0, y_0, z_0) = (0, 1, 1)$, com passo constante $\Delta x = h = 1$
As três primeiras iterações são:

Exemplo: 55.9 Sistema de duas equações de segunda ordem

Reduzamos o sistema de duas equações de segunda ordem

$$\begin{cases} y'' = f(x, y, y', z, z') \\ z'' = g(x, y, y', z, z') \end{cases}$$

a um sistema de quatro equações de primeira ordem:

$$\begin{cases} u' = f(x, y, u, z, v) \\ y' = u \\ v' = g(x, y, u, z, v) \\ z' = v \end{cases}$$

Exemplo: 55.10 Equação diferencial de segunda ordem

Vamos integrar a equação diferencial de segunda ordem

$$y'' = 2.x - y - 0.5.y'$$

começando por a transformar num sistema de equações diferenciais de primeira ordem, fazendo $y' = z$:

$$\begin{cases} z' = & 2.x-y-0.5.z \\ y' = & z \end{cases}$$

Regra Áurea

O método da *secção áurea* utiliza esta condição. Com efeito, começando com o intervalo $[x_1, x_2]$ em que se sabe estar o mínimo, escolhemos

x_3	de modo a ser	$x_3 - x_1 = A.(x_2 - x_1)$
x_4	de modo que	$x_4 - x_1 = B.(x_2 - x_1)$

Assim:

- se for $f(x_3) < f(x_4)$ o mínimo está em $[x_1, x_4]$ e o próximo passo implica $x_2 \leftarrow x_4$ e $x_4 \leftarrow x_3$; será, portanto $x_3 - x_1 = B.(x_4 - x_1)$

- se for $f(x_3) > f(x_4)$ o mínimo está em x_3, x_2 e o próximo passo implica $x_1 \leftarrow x_3$ e $x_3 \leftarrow x_4$; será, portanto $x_4 - x_3 = B.(x_2 - x_3)$

Estas quatro equações exigem, como condição de compatibilidade, que seja $A - A.B^2 = B - A$, isto é, $A = B^2$, de modo que resulta $B^2 - B^4 = B - B^2$, ou seja,

$$B = \frac{\sqrt{5}-1}{2} \qquad = 0,6180339887498948482045868343656\!4$$

$$A = B^2 \qquad = 0,38196601125010515179541316563436$$

.

6.3.3 Pesquisa multidimensional

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - t.\frac{df^{(0)}}{d\mathbf{x}}$$

E pode acontecer o mesmo para y

Onde t = h

6.3.4 Método da quádrica

X(n+1) = Xn - h

Onde $$\mathbf{h} = \mathbf{x} - \mathbf{x}_n = -\mathbf{H}^{-1}(\mathbf{x}_n).\nabla f(\mathbf{x}_n)$$

Se o ponto origina valores crescentes da função (ou seja, afastamos do mínimo) devemos mudar o ponto de partida (decrementamos x e y)

6.3.5 Método de Levenberg-Marquardt

X(n+1) = Xn – h Onde $\mathbf{h} = \mathbf{h}_{quad} + \lambda.\mathbf{h}_{grad}$

- começa-se com um valor elevado de λ (em relação à norma de $\mathbf{h}_{quad}$) , isto é, começa-se virtualmente segundo o gradiente e continua-se decrementando o valor de λ enquanto os novos pontos calculados conduzirem a valores decrescentes da função objectivo; assim, o método aproxima-se progressivamente do da quádrica quando é bem sucedido;

- porém, cada vez que o novo ponto seja mal sucedido (isto é, corresponda a um incremento da função objectivo), o ponto é ignorado, o valor de λ é incrementado e faz-se nova tentativa.