Cálculo Numérico - Jacobi

Este método é iterativo, ou seja, não exato, já que parte-se de um valor atribuído inicial e mediante buscas locais sucessivas, o mesmo converge para a resposta correta. A diferença entre o resultado sugerido e aquele real, é denominada erro, e pode ser tornada tão pequena quanto se queira. O método entretanto tem um senão, que é um critério de convergência bastante rígido. Se não obedecido o algoritmo não converge para a resposta correta, e portanto o método é inútil.

Considere $A.x = b$ um sistema de quações lineares de ordem n e cujo $det(A) \neq 0$. A maneira extensa de escrever tal sistema é

$$\begin{array}{ccccccc} a_{11}x_1 & + & a_{12}x_2 & +...+ & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & +...+ & a_{2n}x_n & = & b_2 \\ ... \\ a_{n1}x_1 & + & a_{n2}x_2 & +...+ & a_{nn}x_n & = & b_n \end{array}$$

A matriz A dos coeficientes, pode ser decomposta em $A = L + D + R$, onde

L é a matriz inferior de A (o resto é zero)

D é a matriz contendo apenas a diagonal principal de A, o resto é zero.

R é a matriz superior de A (o resto é zero)

Veja por exemplo, se

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \therefore L = \begin{pmatrix} 0 & 0 & 0 \\ 4 & 0 & 0 \\ 7 & 8 & 0 \end{pmatrix},$$

e

$$D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 9 \end{pmatrix}, R = \begin{pmatrix} 0 & 2 & 3 \\ 0 & 0 & 6 \\ 0 & 0 & 0 \end{pmatrix}$$

Supondo $det(D) \neq 0$, pode-se escrever: $(L + D + R)x = b$ e daqui $Dx = -(L + R)x + b$. e $x = -D^{-1}(L + R)x + D^{-1}b$. Chamando $B = -D^{-1}(L + R)$ e $g = D^{-1}.b$ obtem-se a equação geral deste método que é

$$x = Bx + g$$

. Note que é uma equação recursiva (ou iterativa) já que o x aparece nos 2 lados da mesma. A idéia aqui é atribuir um valor a x e com ele calcular um novo x, que é jogado no lugar do x velho e gera um novo x e assim sucessivamente até haver a convergência esperada (de antemão). Como supusemos $det(D) \neq 0$ pode-se dividir cada equação do sistema pelo coeficiente da diagonal principal, resultando

$$A^* = L^* + I + R^*$$

Agora o processo iterativo pode ser definido como

$$x^{k+1} = -(L^* + R^*)x^k + b^*$$

onde os elementos de L^*, R^* e b^* são:

$l_ij^* = \frac{a_{ij}}{a_{ii}}$ se $i > j$ e zero senão

$d_{ij} = a_{ij}$ se $i = j$ e zero senão

$r_ij^* = \frac{a_{ij}}{a_{ii}}$ se $i < j$ e zero senão

$b_ij^* = \frac{b_i}{a_{ii}}$

Critérios de convergência

Depois de dividida a matriz pelo elemento da diagonal principal, deve-se ter o **critério das linhas** que diz

$$max_{1 \leq i \leq n} \sum_{j=1, j \neq i}^{n} |a^*_{ij}| < 1$$

OU então o **critério das colunas** que diz

$$max_{1 \leq j \leq n} \sum_{i=1, i \neq j}^{n} |a^*_{ij}| < 1$$

Note que qualquer um dos dois critérios sendo satisfeitos, isso garante que o sistema tem convergência.

Definindo uma matriz como estritamente diagonalmente dominante se

$$\sum_{j=1, j \neq i}^{n} |a_{ij}| < |a_{ii}|$$

e se A é estritamente diagonalmente dominante, então A satisfaz o critério das linhas, e o sistema converge. Então, um bom critério de convergência pode ser

$$max_{1 \leq i \leq n} \sum_{j=1, j \neq i}^{n} |a_{ij}|^* < 1$$

Exemplo

Seja resolver o sistema abaixo, usando o método de Jacobi

$$\begin{array}{ccccccc} 10x & + & 2y & + & z & = & 7 \\ x & + & 5y & + & z & = & -8 \\ 2x & + & 3y & + & 10z & = & 6 \end{array}$$

com $x^0 = (0.7, -1.6 \ e \ 0.6)^t$ e $\epsilon < 0.11$.

Convergência ?

$|a_{12}| + |a_{13}| < |a_{11}|$? ou $|2| + |1| < |10|$? R: sim
$|a_{21}| + |a_{23}| < |a_{22}|$? ou $|1| + |1| < |5|$? R: sim
$|a_{31}| + |a_{32}| < |a_{33}|$? ou $|2| + |3| < |10|$? R: sim
logo, o sistema convergirá!

Solução

Dividindo cada equação pelo elemento da diagonal principal, fica

$$\begin{array}{ccccccc} x & + & 0.2y & + & 0.1z & = & 0.7 \\ 0.2x & + & y & + & 0.2z & = & -1.6 \\ 0.2x & + & 0.3y & + & z & = & 0.6 \end{array}$$

Apenas para ilustrar o critério das linhas para testar convergência (desnecessário no caso, já que já vimos que o sistema converge, mas em tese...)
$|a^*_{12}| + |a^*_{13}| = |0.2| + |0.1| = 0.3$ e
$|a^*_{21}| + |a^*_{23}| = |0.2| + |0.2| = 0.4$ e
$|a^*_{31}| + |a^*_{32}| = |0.2| + |0.3| = 0.5$ e
$\Rightarrow \ max_{1 \leq i \leq n}(0.3, 0.4 \ 0.5) = 0.5 < 1$

Iterações

$x^{k+1} = -0.2y^k - 0.1z^k + 0.7$
$y^{k+1} = -0.2x^k - 0.2z^k - 1.6$
$z^{k+1} = -0.2x^k - 0.3y^k + 0.6$
Começando com $(0.7, \ -1.6, \ 0.6)$ obtemos uma primeira aproximação que é $(0.96, \ -1.86, \ 0.94)$ e jogando estes novos valores na fórmula iterativa, obtem-se:

$x \rightarrow 0.7, \ 0.96, \ 0.978, \ 0.9994, \ 0.9979$
$y \rightarrow -1.6, \ -1.86, \ -1.98, \ -1.9888, \ -1.9996$
$z \rightarrow 0.6, \ 0.94, \ 0.966, \ 0.9984, \ 0.9968$

O critério de parada pode ser
$max(x^{k+1} - x^k) < \epsilon$. No exemplo acima $0.0108 < 0.011$. Ou seja, neste caso, a solução procurada é $x = 0.9979$, $y = -1.9996$ e $z = 0.9978$. Salta aos olhos neste caso que se a tolerância for um pouco menor, a resposta exata será encontrada que é $(1, -2, 1)$. Basta lançar este resultado no sistema original que a exatidão será confirmada.

Solução Freemat

```
function jacobi(a,b)
 n=size(a,1);
 xvelho=zeros(n);
 tol = 0.0001;
 disp('Jacobi');
 disp('Coeficientes');
 disp(a);
 disp('termo independente');
 disp(b);
 disp('solucao inicial');
 disp(xvelho);
 for i=1:n
  for j=1:n
   if i==j
    f(i,j)=0;
   else
    f(i,j)=-a(i,j)/a(i,i);
   end
  end
 end
 for i=1:n
  d(i)=b(i)/a(i,i);
 end
 erromax = tol+1;
 while (erromax > tol)
  for i=1:n
   xnovo(i)=0;
   for k=1:n
    xnovo(i)=xnovo(i)+f(i,k)*xvelho(k);
   end
  end
  for i=1:n
   xnovo(i)=xnovo(i)+d(i);
  end
  for i=1:n
   erro(i)=abs(xnovo(i)-xvelho(i));
  end
  erromax = erro(1);
  for i=2:n
   if erro(i)>erromax
    erromax=erro(i);
   end
  end
  disp('---parcial---');
  disp('vetor x');
  disp(xnovo);
  disp('erro');
  disp(erro);
  disp(['Er max=',num2str(erromax)]);
  for i=1:n
   xvelho(i)=xnovo(i);
  end
 end
 disp('---final----');
 disp(xnovo);
% para executar: a=[v1, v2,... ; w1,w2,...
%                     ; ...]
% b=[t1, t2, ...]  e depois jacobi(a,b)
```

☞ Para você fazer

Resolva usando este método e anexe a listagem do programa (ou copie no verso desta folha):

$$A = \begin{pmatrix} 91 & 28 & 20 & 7 & 35 \\ 29 & 87 & 6 & 19 & 32 \\ 10 & 24 & 57 & 11 & 8 \\ 18 & 9 & 2 & 39 & 3 \\ 30 & 26 & 4 & 21 & 82 \end{pmatrix}$$

e

$$B = (1747, 1178, 1375, 901, 1304)$$

As respostas encontradas, com $\epsilon < 0.0001$ e sugestão inicial igual à coluna B é

x_1	x_2	x_3	x_4	x_5