Atividades de apoio para implementações e/ou simulações de métodos numéricos com o auxílio do software GNU Octave

BOLSISTAS DO PROJETO: PAULO HENRIQUE CARDOSO DE NOVAIS e YURE MORAES PIRES
ORIENTADOR DO PROJETO: PROFESSOR GISLAN SILVEIRA SANTOS

Métodos Iterativos

1 Método de Jacobi-Richardson

O método de Jacobi-Richardson trata-se de uma forma de solucionar, aproximadamente, sistemas de equações. Então, dessa forma, considere o sistema $Ax = b$, de ordem n, dado abaixo:

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

Logo, para encontrar a solução aproximada dele, pelo método de Jacobi-Richardson, é necessário encontrar uma sequência de aproximantes da iteração k, ou seja,

$$x_1^{(k)}, x_2^{(k)}, \dots, k = 1, 2, 3, \dots,$$

A partir do vetor inicial dado por:

$$x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(}0)$$

Portanto, a sequência de aproximantes da iteração k é determinada através do processo iterativo abaixo.

$$\left\{ \begin{array}{ccccccccc} x_1^{(k+1)} & = & -a_{12}^* x_2^{(k)} & - & a_{13}^* x_3^{(k)} & - & \dots & - & a_{1n}^* x_n^{(k)} \\ + & b_1^* & & & & & & & \\ x_2^{(k+1)} & = & -a_{21}^* x_1^{(k)} & - & a_{23}^* x_3^{(k)} & - & \dots & - & a_{2n}^* x_n^{(k)} \\ + & b_2^* & & & & & & & \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ x_n^{(k+1)} & = & -a_{n1}^* x_1^{(k)} & - & a_{n2}^* x_2^{(k)} & - & \dots & - & a_{n,n-1}^* x_{n-1}^{(k)} \\ + & b_n^* & & & & & & & \end{array} \right. \tag{2}$$

Onde,

$$a_{ij}^* = \frac{a_{ij}}{a_{ii}} \text{ e } b_i^* = \frac{b_i}{a_{ii}}$$

Obs: O método de Jacobi-Richardson também é conhecido como Método dos Deslocamentos Simultâneos, devido a mudança simultânea das equações, usando o valor mais recente do vetor x.

Critérios de Convergência
O método de Jacobi-Richardson é convergente caso seja atendido um dos critérios abaixo:

1. O critério das linhas

$$\underset{1 \le i \le n}{max} \sum_{\substack{j=1, \\ i \ne j}}^{n} |a_{ij}^*| < 1$$

2. O critério das colunas

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

1.1 Implementação

Apos definir o método na sessão anterior, o algoritmo será representado da seguinte forma.

```
function x=jacobirichardson(A,b,x0,k)
  n=length(A);
  x=zeros(n,((k+1)+1));
  x=[x0 x];
  for s=1:n
    g(s,1:n)=A(s,1:n)/A(s,s); %equivalente ao a*
    p(s)=b(s)/A(s,s); %equivalente ao b*
  endfor
  q=[g p']; % matriz expandida composta por a* e b*
  %crit rios de converg ncia
  criteriolinhas=zeros(n,1);
  criteriocolunas=zeros(1,n);
  criteriolinhas(1,1)=sum(g(1,2:n));
  criteriocolunas(1,1)=sum(g(2:n,1));
  criteriolinhas(n,1)=sum(g(n,n-1:-1:1));
  criteriocolunas(1,n)=sum(g(n-1:-1:1,n));
  for s=2:n-1
    criteriolinhas(s,1)=sum(g(s,1:n))-1;
    criteriocolunas(1,s)=sum(g(1:n,s))-1;
  endfor
%testando a converg ncia do sistema
    if (max(criteriolinhas)<1) %crit rio das linhas
        for j=1:(k+1)
          x(1,j+1)=q(1,n+1)-q(1,2:n)*x(2:n,j);
          for t=2:n-1
            x(t,j+1)=q(t,n+1)-q(t,1:t-1)*x(1:t-1,j)-q(t,t+1:n)*x(t+1:n,j);
          endfor
            x(n,j+1)=q(n,n+1)-q(n,1:n-1)*x(1:n-1,j);
        endfor
        x=x(:,(k+1));
      elseif (max(criteriocolunas)<1) %crit rio das colunas
          for j=1:(k+1)
            x(1,j+1)=q(1,n+1)-q(1,2:n)*x(2:n,j);
            for t=2:n-1
              x(t,j+1)=q(t,n+1)-q(t,1:t-1)*x(1:t-1,j)-q(t,t+1:n)*x(t+1:n,j);
            endfor
            x(n,j+1)=q(n,n+1)-q(n,1:n-1)*x(1:n-1,j);
          endfor
          x=x(:,(k+1));
      else
          x='O␣m todo␣de␣Jacobi-Richardson␣n o␣ ␣convergente␣para␣esse␣sistema';
          criteriocolunas
          criteriolinhas
    endif
endfunction
```

2 Referências

1. CHAPRA, Steven C.; CANALE, Raymond P.. **Métodos Numéricos para Engenharia, 5ª Edição**. São Paulo, McGraw-Hill, 2011. 809p.

2. EATON, John W.; BATEMAN, David; HAUBERG, Soren; WEHBRING, Rik. **GNU Octave: A high-level interactive language for numerical computations, 5ª edição**. 2020. 1077p.

3. FRANCO, Neide Maria Bertoldi. **Cálculo Numérico**. São Paulo: Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, 2006. 489 p.

4. Todos os Colaboradores. **Cálculo Numérico Um Livro Colaborativo Versão Octave**. Porto Alegre: Projeto REAMAT da UFRGS, 2020. Disponível em:
https://www.ufrgs.br/reamat/CalculoNumerico/livro-oct/main.html