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 Gauss-Seidel

O método de Gauss-Seidel trata-se de, assim como o método de Jacobi-Richardson, uma forma de solucionar, aproximadamente, sistemas de equações. A sua principal diferença é a velocidade de convergência. Então, dessa forma, considere o sistema $Ax = b$, de ordem n, dado abaixo:

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

Logo, para encontrar a solução aproximada dele, pelo método de Gauss-Seidel, 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.

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

Onde,

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

Obs: O método de Gauss-Seidel difere do método de Jacobi-Richardson por usar uma componente $x^{(k+1)}$ na próxima iteração. Por conta disso, o método também é conhecido como Método dos Deslocamentos Sucessivos.

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

1. O critério das linhas
$$\underset{1\leq i\leq n}{max} \sum_{\substack{j=1,\\ i\neq j}}^{n} |a_{ij}^*| < 1$$
2. O critério de Sassenfeld
$$\underset{1\leq j\leq n}{max} \sum_{\substack{i=1,\\ i\neq j}}^{n} \beta_i < 1$$
   onde, os β_i são calculados,
$$\beta_i = \sum_{j=1}^{i-1} |a_{ij}^*\beta_j + \sum_{j=i+1}^{n} |a_{ij}^*$$

1.1 Implementação

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

```
function x=gaussseidel(A,b,x0,iteracoes)
  n=length(A);
  k=iteracoes+1;
  x=zeros(n,k);
  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);
  criteriolinhas(1,1)=sum(g(1,2:n));
  criteriolinhas(n,1)=sum(g(n,n-1:-1:1));
  beta=zeros(n,1);
  beta(1,1)=sum(g(1,2:n));
  if n>2
    for s=2:n-1
      criteriolinhas(s,1)=sum(g(s,1:n))-1;
      for p=1:s-1
      beta(s,1)=sum(g(s,p)*beta(p,1)+g(s,s+1:n));
      endfor
    endfor
  endif
  beta(n,1)=sum(g(n,n-1:-1:1)*beta(n-1:-1:1));
    if (max(criteriolinhas)<1) %crit rio das linhas
      for j=1:k %in cio das itera  es
         x(1,j+1)=q(1,n+1)-q(1,2:n)*x(2:n,j);
           for i=2:n-1
              x(i,j+1)=q(i,n+1)-q(i,1:i-1)*x(1:i-1,j+1)-q(i,i+1:n)*x(i+1:n,j);
           endfor
         x(n,j+1)=q(n,n+1)-q(n,1:n-1)*x(1:n-1,j+1);
      endfor
      x=x(:,k);
    elseif (max(beta)<1) %crit rio de Sassenfeld
      for j=1:k %in cio das itera  es
         x(1,j+1)=q(1,n+1)-q(1,2:n)*x(2:n,j);
           for i=2:n-1
              x(i,j+1)=q(i,n+1)-q(i,1:i-1)*x(1:i-1,j+1)-q(i,i+1:n)*x(i+1:n,j);
           endfor
         x(n,j+1)=q(n,n+1)-q(n,1:n-1)*x(1:n-1,j+1);
      endfor
      x=x(:,k);
    else
        x='O␣m todo␣de␣Gauss-Seidel␣n o␣ ␣convergente␣para␣esse␣sistema';
        criteriolinhas
        beta
    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