

Métodos Numéricos

Trabalho 2 - Algoritmos

Carlos Eduardo Cassimiro da Silva

a = matriz com os coeficientes das equações
 b = matriz com as soluções da equações de a

```
Eliminacao_Gauss(a, b):  
{  
  n <- b.tamanho  
  
  Para k=0 passo 1 até n-1 faça:  
    Para i=k+1 passo 1 até n-1 faça:  
      m <- a[i][k]/a[k][k]  
      a[i][k] <- 0  
      Para j=k+1 passo 1 até n-1 faça:  
        a[i][j] <- a[i][j] - m*a[k][j]  
        b[i] <- b[i] - m*b[k]  
  
  x[n]  
  x[n-1] <- b[n-1]/a[n-1][n-1]  
  Para i=n-1 passo -1 até 0 faça:  
    soma <- 0  
    Para j=i+1 passo 1 até n-1 faça:  
      soma <- soma + a[i][j]*x[j]  
    x[i] <- (b[i]-soma)/a[i][i]  
  Retorne x  
}
```

```

Jacobi(a, b, itr):
{
    n <- b.tamanho

    vmax[n]

    Para i = 0 passo 1 até n-1 faça:
        vmax[i] <- a[i][i]
        Para k=0 passo 1 até n-1 faça:
            a[i] <- -1*k/vmax[i]
            a[i][i] <- b[i]/vmax[i]

    Para i = 0 passo 1 até n-1 faça:
        b[i] <- a[i][i]

    Para m = 0 passo 1 até itr faça:
        soma[n]
        Para j = 0 passo 1 até n-1 faça:
            soma[j] <- 0
            Para i = 0 passo 1 até n-1 faça:
                Se i!=j faça:
                    soma[j] <- soma + b[i] * a[j][i]
                Senão:
                    soma[j] <- soma + a[j][i]
            b <- soma
    Retorne b
}

```

```

Seidel(a, b, itr):
{
    n <- b.tamanho

    vmax[n]

    Para i = 0 passo 1 até n-1 faça:
        vmax[i] <- a[i][i]
        Para k=0 passo 1 até n-1 faça:
            a[i] <- -1*k/vmax[i]
            a[i][i] <- b[i]/vmax[i]

    b[0] <- 0
    Para m = 0 passo 1 até itr faça:
        soma[n]
        Para j = 0 passo 1 até n-1 faça:
            Para i = 0 passo 1 até n-1 faça:
                soma[j] <- 0
                Se i!=j faça:
                    soma[j] <- soma + b[i] * a[j][i]
                Senão:
                    soma[j] <- soma + a[j][i]
            b[j] <- soma[j]
    Retorne b
}

```