2.2.4 Análise do Erro

Vejamos o que foi estudado até aqui. No capítulo 1 apresentamos um contato direto com o erro de origem externa à máquina, e o erro da máquina, no momento em fizemos a mudança de base, e não encontramos os mesmos valores no tratamento de casas decimais. Ao estudarmos os erros de arredondamento e truncamento comparado com dois softwares do mercado (Pascal e MatLab) . No capítulo 2 estudamos a solução de sistemas de n equações e n variáveis muito utilizadas no tratamento de imagem, ou cálculo de eletricidade. Desenvolvemos sistemas automatizados que transformam matrizes quadradas homogêneas em diagonais superiores (método de Gauss) e resolução de sistemas de diagonais superiores através do método da substituição.

Agora faremos a união dos dois sistemas, entretanto ao invés de utilizarmos a entrada de dados via teclado, faremos a leitura de um arquivo onde todos os valores são colocados em linhas, ou seja, cada valor estará em uma linha, considerando a entrada da esquerda para direita, de cima para baixo. Em seguida faremos o retorno do cálculo, para definição do erro.

Erros

Nenhum resultado obtido através de cálculos eletrônicos ou métodos numéricos tem valor se não tivermos conhecimento e controle sobre os possíveis erros envolvidos no processo.

A análise dos resultados obtidos através de um método numérico representa uma etapa fundamental no processo das soluções numéricas.

Número Aproximado

Um número $\tilde{x}$ é dito uma aproximação para o número exato x se existe uma *pequena* diferença entre eles. Geralmente, nos cálculos os números exatos não são conhecidos e deste modo são substituidos por suas aproximações. Dizemos que $\tilde{x}$ é um número aproximado por falta do valor exato x se $\tilde{x} < x$. Se $\tilde{x} > x$ temos uma aproximação por excesso.

Exemplo

Como $1.41 < \sqrt{2} < 1.42$ temos que 1.41 uma aproximação de $\sqrt{2}$ por falta e 1.42 uma aproximação de $\sqrt{2}$ por excesso.

Erros Absolutos e Relativos

Erro Absoluto

A diferença entre um valor exato x e sua aproximação $\tilde{x}$ é dito **erro absoluto** o qual denotamos por e_x.

$$e_x := x - \tilde{x}$$

Cota para o Erro

Na prática, o valor exato é quase sempre não conhecido. Como o erro é definido por $e_x := x - \tilde{x}$ conseqüentemente também será não conhecido. Uma solução para este problema é ao invés de determinar o erro determinar uma **cota** para o erro. Isso permitirá que, mesmo não conhecendo o erro, saber que ele está entre dois valores conhecidos.
Dizemos que um número $\epsilon > 0$ é uma cota para o erro e_x se $|e_x| < \epsilon$

$$\therefore \quad |e_x| < \epsilon \iff |x - \tilde{x}| < \epsilon \iff \tilde{x} - \epsilon < x < \tilde{x} + \epsilon$$

Assim, mesmo não conhecendo o valor exato, podemos afirmar que ele esta entre $\tilde{x} - \epsilon$ e $\tilde{x} + \epsilon$ que são valores conhecidos.

É evidente que uma cota ϵ só tem algum valor prático se $\epsilon \approx 0$

Erro Relativo

Considere : $x = 100$; $\tilde{x} = 100.1$ e $y = 0.0006$; $\tilde{y} = 0.0004$.

Assim $e_x = 0.1$ e $e_y = 0.0002$.

Como $|e_y|$ é muito menor que $|e_x|$ poderiamos "imaginar" que a aproximação $\tilde{y}$ de y é *melhor* que a $\tilde{x}$ de x. Numa análise mais cuidadosa percebemos que as grandezas dos números envolvidos são muito diferentes.

Inspirados nessa observação definimos:

$$E_x := \frac{e_x}{\tilde{x}}$$

que é denominado **erro relativo**. Temos então para os dados acima:

$E_x = e_x/\tilde{x} = 0.1/100.1 = 0.000999$

$E_y = e_y/\tilde{y} = 0.0002/0.0006 = 0.333333$

Agora podemos concluir que a aproximação $\tilde{x}$ de x é melhor que a $\tilde{y}$ de y pois $|E_x| < |E_x|$.

O programa a seguir ler os dados do arquivo em disco ex12.txt.

Borland Pascal - [c:\pendrive\iccii\solusis.pas]

File Edit Search Run Compile Tools Options Window Help

```pascal
program solusis;
uses wincrt;
var
   a,m,av : array[1..50,1..50] of real;
   n : integer; { Número lido no teclado para o cálculo}
   x,b,bv,h    : array[1..50] of real; { Variáveis procuradas }
   arq: Text;
   L,C,k,i,j,linha2: integer; { Contadores do Loop }
   s,linha : real;
   nome_arq : string; opcao :char;
begin
  repeat
     clrscr;
     write('Digite o nome do arquivo com a extensao ');readln(nome_arq);

     Assign ( arq, nome_arq );
     Reset ( arq );
     ReadLn ( arq, linha2 );
     n := linha2;
     for L := 1 to n do begin
         for C := 1 to n do begin
                ReadLn ( arq, linha );
                a[L,C] := linha;
                av[L,C]:= linha;
         end;
         ReadLn ( arq, linha );
         b[L] := linha;
         bv[L]:= linha;
     end;
     close( arq );
     writeln('Solução de sistemas ');
     writeln;
     writeln('Valores de Entrada');
     writeln;
```

25:40 Ins

Borland Pascal - [c:\pendrive\iccii\solusis.pas]

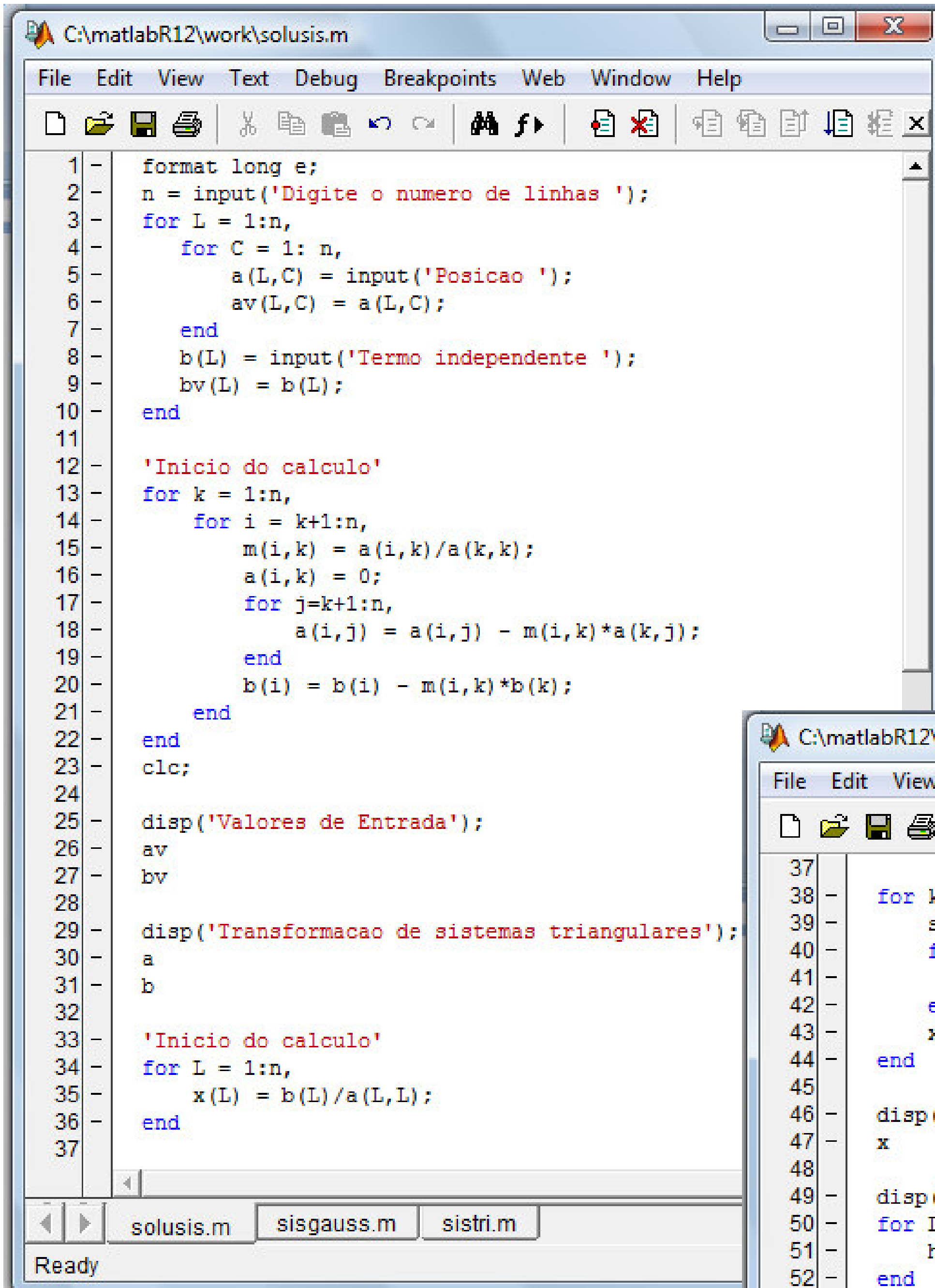
File Edit Search Run Compile Tools Options Window Help

```pascal
     for l := 1 to n do begin
         for c := 1 to n do begin
             write(a[l,c]:10:5,'      ');
         end;
         writeln(' = ',b[l]:10:5);
     end;
     { Início da transformação }

     for k := 1 to n do begin
         for i := k + 1 to n do begin
           m[i,k] := a[i,k]/a[k,k];
           a[i,k] := 0;
           for j := k + 1 to n do begin
               a[i,j] := a[i,j] - m[i,k]*a[k,j];
           end;
           b[i] := b[i] - m[i,k]*b[k];
         end;
     end;

     writeln('Transformação em sistemas triangulares');
     writeln;
     for l := 1 to n do begin
         for c := 1 to n do begin
             write(a[l,c]:10:5,'      ');
         end;
         writeln(' = ',b[l]:10:5);
     end;
```

34:14 Insert

```pascal
    { Início do cálculo  da soluçao }
    for L := 1 to n do x[L] := b[L]/a[L,L];

    for k := n-1 downto 1 do begin
        s := 0;
        for j := k + 1 to n do begin
          s := s + a[k,j]*x[j];
        end;
        x[k] := (b[k] - s)/a[k,k];
    end;

    writeln('Solução de sistemas triangulares');
    writeln;
    for l := 1 to n do writeln(x[l]);

    writeln('Cálculo do Erro Absoluto e Relativo');
    for L := 1 to n do begin
        for c:= 1 to n do begin
            h[L] := h[L] + av[l,c]*x[c];
        end;
    end;

    for l := 1 to n do begin
        h[l] := bv[l]-h[l];
    end;

    for l := 1 to n do writeln(h[l],'   ',h[l]*100/(bv[l]-h[l]),' %');
    write('Deseja continuar ?');readln(opcao);
  until (opcao = 'N') or (opcao = 'n');
end.
```

Implementação no MatLab.

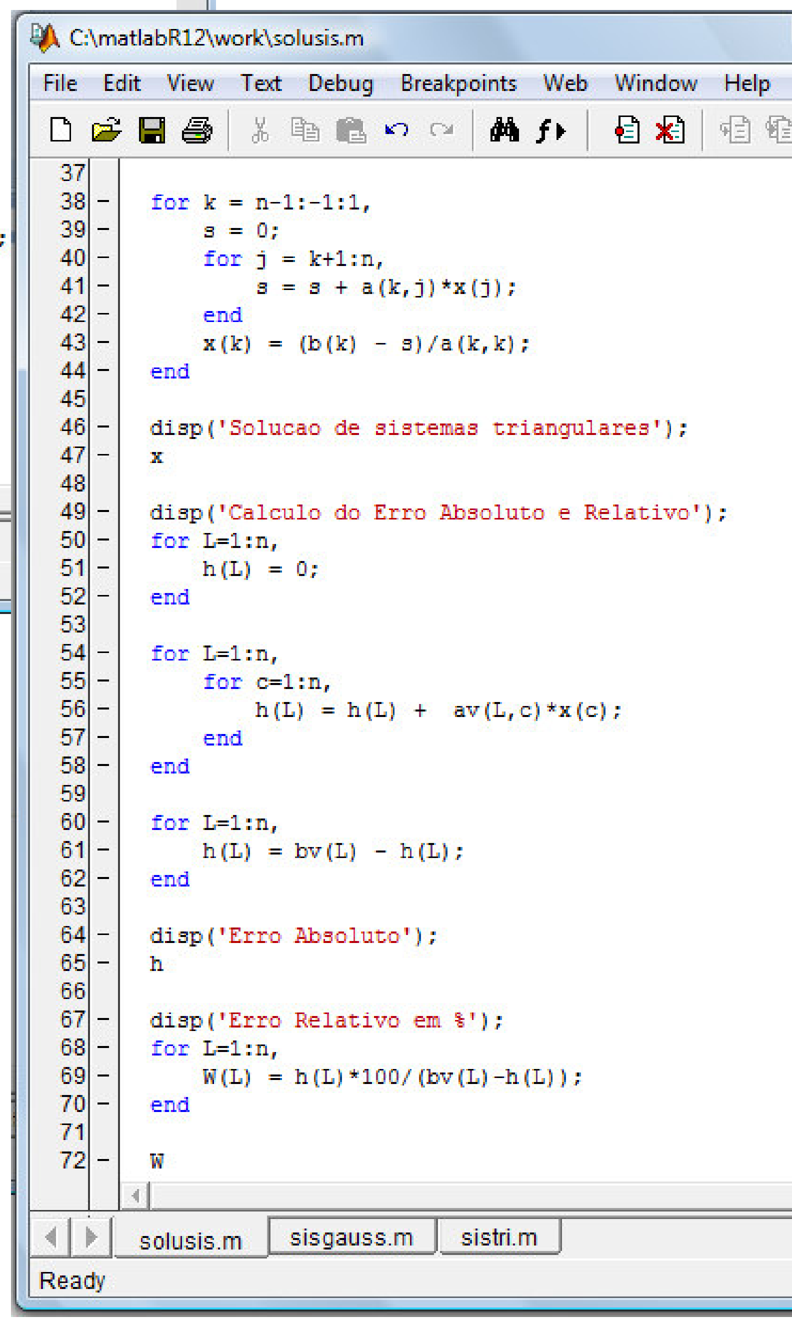


C:\matlabR12\work\solusis.m

```
format long e;
n = input('Digite o numero de linhas ');
for L = 1:n,
   for C = 1: n,
       a(L,C) = input('Posicao ');
       av(L,C) = a(L,C);
   end
   b(L) = input('Termo independente ');
   bv(L) = b(L);
end

'Inicio do calculo'
for k = 1:n,
    for i = k+1:n,
        m(i,k) = a(i,k)/a(k,k);
        a(i,k) = 0;
        for j=k+1:n,
            a(i,j) = a(i,j) - m(i,k)*a(k,j);
        end
        b(i) = b(i) - m(i,k)*b(k);
    end
end
clc;

disp('Valores de Entrada');
av
bv

disp('Transformacao de sistemas triangulares');
a
b

'Inicio do calculo'
for L = 1:n,
    x(L) = b(L)/a(L,L);
end

```

solusis.m | sisgauss.m | sistri.m

Ready



C:\matlabR12\work\solusis.m

```

for k = n-1:-1:1,
    s = 0;
    for j = k+1:n,
        s = s + a(k,j)*x(j);
    end
    x(k) = (b(k) - s)/a(k,k);
end

disp('Solucao de sistemas triangulares');
x

disp('Calculo do Erro Absoluto e Relativo');
for L=1:n,
    h(L) = 0;
end

for L=1:n,
    for c=1:n,
        h(L) = h(L) +  av(L,c)*x(c);
    end
end

for L=1:n,
    h(L) = bv(L) - h(L);
end

disp('Erro Absoluto');
h

disp('Erro Relativo em %');
for L=1:n,
    W(L) = h(L)*100/(bv(L)-h(L));
end

W
```

solusis.m | sisgauss.m | sistri.m

Ready