CI1164 – Introdução à Computação Científica

Prof. Guilherme Derenievicz
Prof. Armando Delgado

Exercícios de Revisão para Prova 01

GABARITO

Questão 1

Considere um equipamento cujo sistema de ponto flutuante **normalizado** é SPF(2,4,-5,5), ou seja, de **base 2**, possui **4 dígitos na mantissa**, **menor expoente -5** e **maior expoente 5**. Os números abaixo são fornecidos a este sistema:

(a) 0.1011×2^4 (b) 0.1101×2^{-1} (c) 0.1110×2^1

Qual é o resultado das seguintes operações, considerando que a máquina efetua o truncamento dos resultados. Calcule também para cada item o valor exato (sem considerar truncamento) .

1) $(a+b)+c$

$x = (1011 + 0{,}01101) + 1{,}110 = 1101{,}00101 = 13{,}15625_{10}$

$\bar{x} = (0{,}1011 * 2^4 + 0{,}000001101 * 2^4) + 0{,}1110 * 2^1 = 0{,}1011 * 2^4 + 0{,}0001110 * 2^4 = 0.1100 * 2^4 = 12_{10}$

2) $a+(b+c)$

$x = 1011 + (0{,}01101 + 1{,}110) = 1101{,}00101 = 13{,}15625_{10}$

$\bar{x} = 0{,}1011 * 2^4 + (0{,}001101 * 2^1 + 0{,}1110 * 2^1) =$
$0{,}1011 * 2^4 + 1{,}0001 * 2^1 = 0{,}1011 * 2^4 + 0{,}1000 * 2^2 = 0{,}1011 * 2^4 + 0{,}0010 * 2^4 = 0.1101 * 2^4 = 13_{10}$

3) Explique a diferença de resultados verificadas nos itens (1) e (2)

No item (1) a soma dos elementos **a** e **b** causa um cancelamento subtrativo pois a mantissa não é capaz de representar o valor de **b** quando seu expoente tem que ser igualado ao expoente de **a**. Este cancelamento tem efeito menor no item (2) porque ao menos parte do valor de **b** pode ser somado a **c** e não é perdido na subsequente operação com **a**.

Questão 2

Considere um equipamento cujo sistema de ponto flutuante **normalizado** é SPF(10,4,-5,5), ou seja, de **base 10**, possui **4 dígitos na mantissa**, **menor expoente -5** e **maior expoente 5**. Os números abaixo são fornecidos a este sistema:

(a) 0.4523×10^{4} (b) 0.2116×10^{-1} (c) 0.2583×10^{1}

Qual é o resultado das seguintes operações, considerando que a máquina efetua o truncamento dos resultados. Calcule os erros absolutos e relativos destas aproximações:

i. $(a+b)+c$

$$x = (4523{,}0 + 0.02116) + 2{,}583 = 4525{,}60416$$

$$\bar{x} = (0.4523 * 10^{4} + 0.000002116 * 10^{4}) + 0.2583 * 10^{1} = 0.4523 * 10^{4} + 0.0002583 * 10^{4} = 0.4525 * 10^{4}$$

$$EA = |x - \bar{x}| = |4525{,}60416 - 4525{,}0| = 0{,}60416$$

$$ER = \frac{EA}{|x|} \times 100 = \frac{0{,}60416}{4525{,}60416} \times 100 = 0{,}01334\,\%$$

ii. $a+(b+c)$

$$x = 4523{,}0 + (0.02116 + 2{,}583) = 4525{,}60416$$

$$\bar{x} = 0.4523 * 10^{4} + (0.002116 * 10^{1} + 0.2583 * 10^{1}) = 0.4523 * 10^{4} + 0.0002604 * 10^{4} = 0.4525 * 10^{4}$$

$$EA = |x - \bar{x}| = |4525{,}60416 - 4525{,}0| = 0{,}60416$$

$$ER = \frac{EA}{|x|} \times 100 = \frac{0{,}60416}{4525{,}60416} \times 100 = 0{,}01334\,\%$$

Questão 3

Considere os métodos numéricos estudados nesta disciplina e o cálculo de erros apresentados abaixo.

i) Dada uma função $f(x)$ definida e contínua no intervalo I, chamamos de zero (ou raiz) da função a todo $\alpha \in I \mid f(\alpha)=0$. Considere $x_i \approx \alpha$ o resultado da i-ésima iteração de algum método numérico para o cálculo de α. Indique em que situações cada uma das formas de cálculo de erro abaixo é mais adequada:

a) $|f(x_i)|$ *: Quando a função próxima da raiz faz com que o método não produza valores que convergem rápido. Este erro não nos diz nada com relação à raiz propriamente dita e tem pouca utilidade para funções mal comportadas na proximidade da raiz.*

b) $\dfrac{|x_i - x_{i-1}|}{|x_i|}$: Quando a função produz valores que convergem rápido e que não está próxima da origem. Este erro nos dá informação mais adequada da variação da aproximação,

c) $|x_b - x_a|$, onde $x_a \le \alpha \le x_b$: Quando o método lida com intervalos contendo a raiz e este intervalo vai se estreitando sempre (bisseção e falsa posição)

ii) Seja um Sistema de Equações Lineares da forma $Ax=b, A \in \mathbb{R}^{n\times n}, \{x,b\} \in \mathbb{R}^n$ e $\overline{x}^{(k)} \approx x$ o resultado da k-ésima iteração de algum método para a solução de sistemas lineares. Lembrando que o resíduo é definido $r = b - A\overline{x}^{(k)}$, indique em que situações cada uma das formas de cálculo de erro abaixo é mais adequada:

d) $\|x\|_\infty = max\left(|\overline{x}_i^{(k)} - \overline{x}_i^{(k-1)}|\right), i=1,2,\ldots,n$: Quando se deseja ter indicação da contribuição de cada incógnita para o erro.

e) $\|r\|_2 = \sqrt{r_1^2 + r_2^2 + \ldots + r_n^2}$: Quando se deseja saber se o S.L. está satisfeito de forma geral. O que não sabemos é quanto cada variável contribui para cada erro.

iii) Sejam `xd` e `raiz` a representação em ponto flutuante IEEE754 (`double`) dos valores de x_i e α do item (i), respectivamente. Considerando ainda que o método numérico esteja convergindo, explique porque o laço a seguir (em linguagem C) não é uma opção para testar a convergência do método.

```
while( fabs(xd - raiz) > 0.0) )
{
    ...
}
```

Devido à representação IEEE754, a diferença **xd – raiz** pode nunca chegar a ser 0.0, mesmo havendo convergência do método. Portanto o valor de **fabs()** nunca será igual a zero e o laço nunca termina.

Questão 4

O algoritmo abaixo pode ser utilizado para calcular com quantos dígitos um computador trabalha.

```
ε ← 1.0
j = 1
Enquanto (1.0 + ε > 1.0) faça
   ε ←  ε / 2.0
   j ← j + 1

Escreva o valor de j
```

Explique por que o algoritmo não entra em laço infinito.

R: O valor de ε é dividido ao meio a cada iteração, o que equivale a reduzir o expoente em uma unidade a cada iteração (consequentemente um dígito da mantissa é retirado da precisão do número ε no momento da soma). O laço termina obrigatoriamente porque em algum momento todos os dígitos da mantissa de ε serão deslocados para igualar o expoente de 1.0, deixando ε =0.0 na operação de soma.

Questão 5

Considere as duas expressões equivalentes abaixo para calcular a abcissa da interseção da reta que passa pelos pontos $(x_{0,}y_0)$ e (x_1,y_1) com o eixo x :

(a) $x=\frac{x_0y_1-x_1y_0}{y_1-y_0}$ (b) $x=x_0-\frac{(x_1-x_0)y_0}{y_1-y_0}$

i. Usando os pontos $(0.131\times10^1,0.324\times10^1)$ e $(0.193\times10^1,0.476\times10^1)$ calcule o valor de x em um SPF(10,3,-5,5). Calcule os erros absolutos e relativos destas aproximações.

(a)

$$x_a = \frac{1{,}31*4{,}76 - 1{,}93*3{,}24}{4{,}76 - 3{,}24} = \frac{6{,}2356 - 6{,}2532}{1{,}52} = \frac{-0{,}0176}{1{,}52} = -0{,}011578947$$

$$\bar{x_a} = \frac{0.131\times10^1*0.476\times10^1-0{,}193\times10^1*0{,}324\times10^1}{0{,}476\times10^1-0{,}324\times10^1} = \frac{0{,}0623\times10^1-0{,}0625\times10^1}{0{,}152\times10^1}$$

$$= \frac{0{,}623\times10^0 - 0{,}625\times10^0}{0{,}152\times10^1} = \frac{-0{,}002\times10^0}{0{,}152\times10^1}$$

$$= -0{,}00131$$

$$EA = |x - \bar{x}| = |-0{,}011578947 - (-0{,}00131)| = 0{,}010268947$$

$$ER = \frac{EA}{|x|}\times100 = \frac{0{,}010268947}{0{,}011578947}\times100 = 88{,}6864\%$$

(b)

$$x_b = 1{,}31 - \frac{(1{,}93 - 1{,}31) * 3{,}24}{4{,}76 - 3{,}24} = 1{,}31 - \frac{0{,}62 * 3{,}24}{1{,}52} = 1{,}31 - \frac{2{,}0088}{1{,}52} = 1{,}31 - 1{,}321578947$$
$$= -0{,}011578947$$

$$\bar{x_b} = 0.131 \times 10^1 - \frac{(0.193 \times 10^1 - 0.131 \times 10^1) * 0.324 \times 10^1}{0.476 \times 10^1 - 0.324 \times 10^1} = 0.131 \times 10^1 - \frac{0{,}062 \times 10^1 * 0.324 \times 10^1}{0.152 \times 10^1}$$
$$= 0.131 \times 10^1 - \frac{0{,}020 \times 10^1}{0.152 \times 10^1} = 0.131 \times 10^1 - 0.131 \times 10^1$$
$$= 0.000$$

$$EA = |x - \bar{x}| = |-0{,}011578947 - 0{,}00| = 0{,}011578947$$

$$ER = \frac{EA}{|x|} \times 100 = \frac{0{,}011578947}{0{,}011578947} \times 100 = 100\,\%$$

ii. Qual dos dois métodos é melhor? Justifique.

O primeiro método é o melhor, pois o efeito do cancelamento subtrativo e menor do que no segundo método.

Questão 6

Considere um equipamento cujo sistema de ponto flutuante **normalizado** é SPF(2,3,-4,4). Responda:

1) Qual o menor número positivo exatamente representável, em base 2?

0.100×2^{-4}

2) Qual o próximo positivo, depois do menor positivo representável, em base 2?

0.101×2^{-4}

3) Verifique se existem números reais entre o menor e o próximo positivo. Comente as implicações de sua verificação.

R: Existem infinitos números reais no intervalo entre o menor e o próximo positivo, entretanto eles não podem ser representados no SPF em questão devido ao número limitado de dígitos na mantissa. Desta forma, qualquer número neste intervalo será truncado/arredondado para o menor/próximo.

Questão 7

Um paralelepípedo retangular **tem** dimensões $x = 3\,\text{cm},\ y = 4\,\text{cm}$ e $z = 5\,\text{cm}$. Ele foi medido com um paquímetro com precisão de $\pm 0{,}1\,\text{cm}$.

(a) Calcule o erro absoluto máximo e o erro relativo máximo no volume do paralelepípedo.

$$EA = |3 \times 4 \times 5 - 3{,}1 \times 4{,}1 \times 5{,}1| = |60 - 64{,}821| = 4{,}821$$

$$ER = \frac{|4{,}821|}{|60|} = 0{,}080 = 8\,\%$$

(b) Este erro é Real ou Aproximado? Justifique.

O erro é real porque conhecemos as medidas reais do paralelepípedo

Questão 8

Observe o trecho de código a seguir, considere as variáveis **soma1** e **soma2** e responda:

```
float soma1=0.0f, soma2=0.0f;

for (int i=1; i<=200; ++i)
   soma1 += 1.0f / (i*i);

for (int i=200; i>=1; --i)
   soma2 += 1.0f / (i*i);
```

a) Qual variável terá o valor mais exato? Por que isso ocorre?

soma2 porque adicionar pequenos valores a uma soma com acumulador implica em erros de arredondamento devido ao cancelamento subtrativo. Iniciando pelos menores valores diminui a ocorrência de erros porque o somador não é muito maior do que os valores sendo somados a ele.

b) A precisão das variáveis é a mesma? Justifique sua resposta.

Sim, pois ambas utilizam a mesma representação em ponto flutuante, ou seja, possuem a mesma quantidade de dígitos na mantissa. Entretanto, **soma1** terá menos dígitos significativos do que **soma2**.

Questão 9

Considere um equipamento cujo sistema de ponto flutuante (SPF) **normalizado** de **base 2** possui **4 dígitos na mantissa**, **menor expoente -5** e **maior expoente 5** (SPF(2,4,-5,5)). Para este sistema:

a) Qual a diferença entre o menor número positivo representável e o próximo número, imediatamente maior (em base 2)?

min = ,1000x2^{-5}, prox = ,1001x2^{-5}, diff = ,0001x2^{-5} = 1 x 2^{-9}

b) Qual a diferença entre o maior número positivo representável e o número anterior, imediatamente menor (em base 2)?

max = ,1111x2^{5}, antes = ,1110x2^{5}, diff = ,0001x2^{5} = 1 x 2^{1}

c) Qual é o maior número inteiro ímpar que este sistema pode representar (em base 2 ou base 10)?

maxImpar = 0,1111x2^{4}

d) Explique uma implicação de suas respostas anteriores.

(a) Existem infinitos números reais no intervalo entre o menor e o próximo positivo, entretanto eles não podem ser representados no SPF em questão devido ao número limitado de dígitos na mantissa. Desta forma, qualquer número neste intervalo será truncado/arredondado para o menor ou para o próximo valor.

(b) O mesmo acontece para valores entre o maior valor e o anterior.

(c) Em ponto flutuante, apenas números inteiros pares podem ser representados a partir do maior inteiro ímpar representável em PF.

Observa-se que sempre há 5 números em ponto flutuante entre duas potências de dois:

- Para cada aumento na potência de 2, a quantidade de inteiros representados duplica, mas a quantidade de números em ponto flutuante é constante.
- A precisão do número em ponto flutuante é proporcional à sua magnitude. Quanto maior o número, menor a precisão.
- À medida em que o valor aumenta, diminui a precisão do número em ponto flutuante.

Questão 10

Escreva uma função em linguagem C que receba como parâmetros de entrada os limites de um intervalo (a,b) e um valor de tolerância, e calcule uma raiz da equação $sen(x)-x^3+x+1=0$ utilizando duas iterações do método da Bisseção e em seguida o método da Secante até que o erro aproximado absoluto em x seja menor do que a tolerância estipulada.

DICA: a função de iteração do método da Secante é dada por

$$x_{k+1}=\frac{x_{k-1}f(x_k) - x_k f(x_{k-1})}{f(x_k) - f(x_{k-1})}$$

```
double raizEquacao( double a, double b, double tol )
{
   double fx( double x) {
      return sin(x) + x*x*x + 1;
   }

   // Bissection
   double fa = fx(a);
   double fb = fx(b);
   double xAnt, fxAnt, fxM=fb, xM=b;
   for (int i=0; i<2; ++i) {
      xAnt = xM;
      fxAnt = fxM;
      double xM = a+b/2.0;
      double fxM = fx(xM);
      if( fa*fxM < 0.0 ) {
         b = xM; fb = fxM;
      } else {
         a = xM; fa = fxM;
      }
   } // end bissection

   // Secante
   double xAntAnt, fxAntAnt;
   do {
      xAntAnt = xAnt; fxAntAnt = fxAnt;
      xAnt = xM; fxAnt = fxM;
      xM = (xAntAnt * fxAnt - xAnt * fxAntAnt) / (fxAnt - fxAntAnt);
      fxM = fx(xM);
   } while ( fabs(xAnt - xM) < tol );

   return xM;
}
```

Questão 11

Escreva uma função em linguagem C que receba como parâmetros de entrada as diagonais de uma matriz tri-diagonal, o vetor de termos independentes $\vec{b}$, o vetor com os valores iniciais para $\vec{x}$, a ordem **n** da matriz, e um valor de tolerância tol, e devolva sua solução $\vec{x}$ usando o método de Gauss-Seidel até 5 (cinco) iterações. Se existir alguma condição ou propriedade para os vetores das diagonais e termos independentes, isto deve ser indicado na resposta.

```
/* double ds[N], ds[n-1] = 0
   double di[N], di[0] = 0
   double Xi[N+1],  Xi[0] = 0, X = Xi+1
*/
void gaussSeidel( double *ds, double *dp, double *di, double *b,
                  int n, double *X)
{
  int i, j;
  double s;

  for (j=0; j < 5; ++j) {
    for (i=0; i < n; ++i) {
       s = di[i]*X[i-1] + ds[i]*X[i+1];
       X[i] = (b[i] - s)/dp[i];
    }
  }
}
```

Questão 12

Seja um Sistema de Equações Lineares da forma $Ax=b, A\in\mathbb{R}^{n\times n}, \{x,b\}\in\mathbb{R}^n$. Para a solução deste sistema, quantas operações aritméticas são realizadas EM CADA ITERAÇÃO dos Métodos de Jacobi e de Gauss-Seidel quando:

O algoritmos Gauss-Seidel e Jacobi executam o mesmo número de operações aritméticas, o que muda é a origem dos valores das variáveis do SL (iteração atual x iteração anterior).

$$x_i^{(k)}=\frac{1}{a_{ii}}[b_i - \sum_{j=i,j\neq i}^{n} a_{ij}.x_j^{(k-1)}],\ i=1,2,\ldots,n \quad \text{(Jacobi)}$$

$$x_i^{(k)}=\frac{1}{a_{ii}}[b_i - \sum_{j=1,j\neq i}^{i-1} a_{ij}.x_j^{(k)} - \sum_{j=i+1,j\neq i}^{n} a_{ij}.x_j^{(k-1)}],\ i=1,2,\ldots,n \quad \text{(Gauss-Seidel)}$$

a) **A** é uma matriz $n\times n$ *?*

Para cada equação do SL são efetuados **(n-1)** produtos, **(n-1)** somas e **1** (uma) divisão, isto é, **2n-1** operações aritméticas (em ponto flutuante). Cada ITERAÇÃO DO MÉTODO (cálculo de todos os valores $\mathbf{x}_i$), portanto, realiza **n(2n – 1) = 2n² - n** operações aritméticas. Para **n** muito grandes, isto é da ordem de **2n²** operações.

b) **A** é uma matriz de banda *k*-diagonal?

Para uma matriz k-diagonal, para cada equação do SL são efetuados **(k-1)** produtos, **(k-1)** somas e **1** (uma) divisão, isto é, **2k-1** operações aritméticas (em ponto flutuante). Cada ITERAÇÃO DO MÉTODO (cálculo de todos os valores $\mathbf{x_i}$), portanto, realiza **n(2k – 1) = 2kn - n** operações aritméticas. Para **n** muito grandes, isto é da ordem de **2kn** operações.

c) Em quais condições estes métodos iterativos serão competitivos com o Método da Eliminação Gaussiana (sem considerar pivotamento)?

Por competitivo entenda-se equivalentes, ou a partir de que condições um executa mais operações que outro.

Na eliminação de Gauss, a quantidade de operações é **$(2/3)n^3 + O(n^2)$** para matrizes **n x n**, ou seja, da ordem de **$O(n^3)$** operações. Para sistemas k-diagonais o número de operações é da ordem de **$O(kn^2)$**.

Os métodos iterativos são competitivos (ou calculam menos operações aritméticas) sempre que o número de iterações for menor do que **n**. A partir do momento em que são necessárias aproximadamente **n** iterações para convergência, eles deixam de ser competitivos.

Questão 13

1) Explique como a inversa de uma matriz $A, n \times n$, pode ser obtida através da resolução de n sistemas lineares $n \times n$.

R: Seja M a inversa da matriz A tal que $A.M = I$. Podemos calcular cada coluna k da matriz M resolvendo um sistema linear $A.m_k = I_k$ para $k = 1,2,\ldots,n$

2) Entre o método da Eliminação de Gauss e a decomposição LU, qual o mais indicado para este caso? Justifique.

R: A decomposição LU é mais indicada pois todos n sistemas lineares a serem resolvidos compartilham a mesma matriz de coeficientes A . Desta forma, ela precisa ser fatorada apenas uma vez.

Questão 14

Matrizes tridiagonais são aquelas em que apenas os elementos da diagonal principal, e os elementos das diagonais imediatamente acima e abaixo são não nulos. Sistemas lineares com matrizes de coeficientes tridiagonais são bastante comuns na solução de problemas de computação científica.

OPÇÃO A

a) Elabore uma estrutura de dados em linguagem C para armazenar um sistema linear com matriz de coeficientes tridiagonal, que seja eficiente para resolução pelo método de Gauss-Seidel;

```
struct t_SistLinear3Diag {
  double *dp,*ds,*di; /* Diagonal principal, superior */
  double *b; /* Termos independentes */
  unsigned int n; /* numero de equações do SL */
};
```

b) Implemente uma função em C que resolva pelo método de Gauss-Seidel um sistema linear tridiagonal;

```
#define MAXIT 100

void gaussSeidel (struct t_SistLinear3Diag *SL, double *x, double
erro)
{
  double norma, diff, xk;
  int k = 1, i;
  do {
    // primeira equação fora do laço
    i = 0;
    xk = (SL->b[i] - SL->ds[i]*x[i+1]) / SL->dp[i];
    norma = fabs(xk - x[0]);
    x[i] = xk;

    // equações centrais
    for (i=1; i<SL->n-1; ++i) {
      xk = (SL->b[i] - SL->ds[i]*x[i+1]- SL->di[i]*x[i-1]) / SL-
>dp[i];
      // Calcula norma ||  x^(k) - x^(k - 1) ||
      diff = fabs(xk - x[i]);
      norma = (diff > norma) ? (diff) : (norma);
      x[i] = xk;
    }

    // ultima equação fora do laço
    xk = (SL->b[i] - SL->di[i]*x[i-1]) / SL->dp[i];
    diff = fabs(xk - x[i]);
    norma = (diff > norma) ? (diff) : (norma);
    x[i] = xk;
```

```
    ++k;
  } while (norma > erro && k < MAXIT) ;
}
```

OPÇÃO B

a) Elabore uma estrutura de dados em linguagem C para armazenar um sistema linear com matriz de coeficientes tridiagonal, que seja eficiente para resolução pelo método de Gauss-Seidel;

```
typedef struct {
  double *A[3];
  double *x;
  double *b;
  int N;
} SLTridiag_t;
```

b) Implemente uma função em C que resolva pelo método de Gauss-Seidel um sistema linear tridiagonal;

```
void GaussSeidelTridiag ( SLTridiag sl, int maxIter ) {
   int i, iter, INF=0, PRI=1; SUP=2;
   double mult;

   for (iter=0; iter < maxIter; ++iter)
   {
      sl.x[0] = (sl.b[0] – sl[SUP][0]*sl.x[1]) / sl.A[PRI][0];
      for (i=1; i < sl.N-1; ++i)
      {
         sl.x[i] = (sl.b[i] – sl[INF][i]*sl.x[i-1] – sl[SUP][i]*sl.x[i+1])
                   / sl.A[PRI][i];
      }
      sl.x[0] = (sl.b[sl.N-1] – sl[INF][sl.N-1]*sl.x[sl.N-2])
                / sl.A[PRI][sl.N-1];
   }
}
```

Questão 15

Dado um sistema linear com N equações, da forma $Ax=b$, a decomposição de Cholesky fatoriza a matriz de coeficientes A em uma matriz triangular superior e sua conjugada transposta $A=RR^T$. Os elementos de $R=r_{i,j}$ $i \le j \le N$ são calculados da seguinte maneira:

Elementos da diagonal principal:

$$r_{ii}=\sqrt{a_{ii}-\sum_{k=1}^{i-1}(r_{ki})^2}$$

Elementos acima da diagonal principal $(i<j)$:

$$r_{ij}=\sqrt{a_{ij}-\sum_{k=1}^{i-1}(r_{ki})(r_{kj})}$$

Implemente uma função em linguagem C, utilizando o cabeçalho definido abaixo, para calcular a matriz R da decomposição de Cholesky.

```
/* A: matriz de coeficientes  de um S.L. de ordem 'n'
   R: decomposição de Cholesky
   n: ordem das matrizes A e R */
void cholesky( double A[][], double R[][], uint n )
{   }
```

Dica: A decomposição é calculada uma coluna de cada vez, iniciando pelo elemento

$r_{11}=\sqrt{a_{11}}$

```
/* A: matriz de coeficientes  de um S.L. de ordem 'n'
   R: decomposição de Cholesky
   n: ordem das matrizes A e R */
void cholesky( double A[][], double R[][], uint n )
{
   int i, j, k;
   for(j=0; j<n; ++j) {
      for (i=0; i<j; ++i) {
         R[i][j] = 0.0f;
         for (k=0; k<i; ++k)
            R[i][j] += R[k][i]*R[k][j];
         R[i][j] = sqrt(A[i][j] - R[i][j]);
      }
      /* agora i == j */
      R[i][i] = 0.0f;
      for (k=0; k<i; ++k)
         R[i][i] += R[k][i]*R[k][i];
      R[i][i] = sqrt(A[i][i] - R[i][i]);
   }
}
```

OU

```
void cholesky( double A[][], double R[][], uint n )
{
   int i, j, k;
   for(j=0; j<n; ++j) {
      for (i=0; i<=j; ++i) {
         R[i][j] = 0.0f;
         for (k=0; k<i; ++k)
            R[i][j] += R[k][i]*R[k][j];
         R[i][j] = sqrt(A[i][j] - R[i][j]);
      }
   }
}
```

Questão 16

Defina o que é um sistema linear bem condicionado (estável) e o que é um sistema linear mal condicionado.

R: Num sistema bem condicionado pequenas alterações na matriz de coeficientes não alteram a convergência ou o resultado do método. Num sistema mal condicionado ocorre o contrário.

Questão 17

Quando a decomposição LU é vantajosa computacionalmente se comparada ao Método da Eliminação de Gauss?

R: A decomposição LU é vantajosa sempre que for necessário resolver mais de um sistema linear com a mesma matriz de coeficientes, i.e., apenas alteram-se os valores dos termos independentes das equações.