

Trabalho 1

Métodos Iterativos para Sistemas Lineares

Eduardo Brunaldi dos Santos
8642515

Jorge Ashkar Ferreira Simondi
8517081

Victor Luiz da Silva Mariano Pereira
8602444

2018

Introdução

1	Método Iterativo de Gauss-Seidel
---	----------------------------------

--

2 Códigos Fonte

2.1 Função principal (main.c)

```
1  /**
2   *   Trabalho 1 - Métodos Iterativos para Sitemas Lineares
3   *
4   *   Cálculo Numérico   SME-0104
5   *   Prof.: Murilo Francisco Tomé
6   *
7   *   Eduardo Brunaldi dos Santos           8642515
8   *   Jorge Ashkar Ferreira Simondi         8517081
9   *   Victor Luiz da Silva Mariano Pereira  8602444
10  */
11
12  #include <stdio.h>
13  #include <stdlib.h>
14  #include <gauss_seidel.h>
15
16  int main (int argc, char *argv[]){
17      // Variáveis de entrada
18      int n;           // Quantidade de equações do sistema linear
19      int itmax;       // Número máximo de iterações
20      long double **A; // Matriz de sistemas lineares
21      long double *b;  // Vetor resultado das equações
22      long double *x;  // Vetor inicial
23      long double e;   // Erro permitido, precisão
24
25      // Iteradores
26      int i;
27      int j;
28
29      // Dimensão da matriz (número de equações do sistema linear)
30      scanf("%d", &n);
31
32      // Alocação da matriz
33      A = malloc(sizeof(long double *) * n);
34      for (i = 0; i < n; i++)
35          A[i] = malloc(sizeof(long double) * n);
36
37      // Pegando valores de A
38      for (i = 0; i < n; i++)
39          for (j = 0; j < n; j++)
40              scanf("%Lf", &(A[i][j]));
41
42      // Alocação do vetor resultado
43      b = malloc(sizeof(long double) * n);
44
45      // Pegando os valores de b
46      for(i = 0; i < n; i++)
47          scanf("%Lf", &b[i]);
48
49      // Alocação do vetor chute
50      x = malloc(sizeof(long double) * n);
51
52      // Pegando os valores do x(0), o vetor inicial
53      for(i = 0; i < n; i++)
54          scanf("%Lf", &x[i]);
55
56      // Pegando o valor da precisão (erro permitido)
```

```

57     scanf("%Lf", &e);
58
59     // Pegando a quantidade máxima de iterações
60     scanf("%d", &itmax);
61
62     // Calcula um valor aproximado para a resposta
63     // usando o método de Gauss-Seidel
64     x = gauss_seidel(A, b, x, n, e, itmax);
65
66     // Imprime a solução na tela
67     imprime_vetor(x, n);
68
69     // Liberando a memória
70     free(x);
71     free(b);
72     for (i = 0; i < n; i++)
73         free(A[i]);
74     free(A);
75
76     return 0;
77 }

```

2.2 Biblioteca auxiliar

2.2.1 Header (gauss_seidel.h)

```

1  /**
2   *   Trabalho 1 - Métodos Iterativos para Sitemas Lineares
3   *
4   *   Cálculo Numérico   SME-0104
5   *   Prof.: Murilo Francisco Tomé
6   *
7   *   Eduardo Brunaldi dos Santos           8642515
8   *   Jorge Ashkar Ferreira Simondi         8517081
9   *   Victor Luiz da Silva Mariano Pereira  8602444
10  */
11
12  #ifndef GAUSS_SEIDEL_H
13  #define GAUSS_SEIDEL_H
14
15  /**
16   * Função para imprimir de forma mais legível uma matriz quadrada
17   * @param A Matriz a ser impressa
18   * @param n dimensão da matriz
19   */
20  void imprime_matriz(long double **A, int n);
21
22  /**
23   * Função para imprimir um vetor de forma mais legível
24   * @param v vetor a ser impresso
25   * @param n tamanho do vetor
26   */
27  void imprime_vetor(long double *v, int n);
28
29  /**
30   * Função para retornar a norma infinita de um vetor obtido pela subtração
31   * de dois vetores
32   * @param xk Vetor  $x(k+1)$ 

```

```

33  * @param x Vetor x(k)
34  * @param n Dimensão dos vetores
35  * @return norma do vetor obtido pela subtração
36  */
37  long double norma_infinita(long double *xk, long double *x, int n);
38
39  /**
40  * Função para resolver o sistema linear usando o método de gauss-seidel
41  * @param A Matriz de funções do sistema linear
42  * @param b Resultados das equações do sistema linear
43  * @param x Vetor contendo os resultados iniciais
44  * @param n Dimensão do sistema linear
45  * @param e Precisão
46  * @param itmax Número máximo de iterações
47  */
48  long double *gauss_seidel(long double **A, long double *b, long double *x, int n, long
↳ double e, int itmax);
49
50  #endif

```

2.2.2 Implementação da biblioteca (gauss_seidel.c)

```

1  /**
2  * Trabalho 1 - Métodos Iterativos para Sistemas Lineares
3  *
4  * Cálculo Numérico SME-0104
5  * Prof.: Murilo Francisco Tomé
6  *
7  * Eduardo Brunaldi dos Santos 8642515
8  * Jorge Ashkar Ferreira Simondi 8517081
9  * Victor Luiz da Silva Mariano Pereira 8602444
10  */
11
12  #include <stdio.h>
13  #include <stdlib.h>
14  #include <math.h>
15  #include <gauss_seidel.h>
16
17  /**
18  * Função para imprimir de forma mais legível uma matriz quadrada
19  * @param A Matriz a ser impressa
20  * @param n dimensão da matriz
21  */
22  void imprime_matriz(long double **A, int n){
23      int i;
24      int j;
25
26      for (i = 0; i < n; i++){
27          for (j = 0; j < n; j++){
28              printf("%Lf\t", A[i][j]);
29          }
30      }
31  }
32
33  /**
34  * Função para imprimir um vetor de forma mais legível
35  * @param v vetor a ser impresso
36  * @param n tamanho do vetor

```

```

37  */
38  void imprime_vetor(long double *v, int n){
39      int i;
40
41      for (i = 0; i < n; i++)
42          printf("%.16Lf\n", v[i]);
43  }
44
45  /**
46   * Função para retornar a norma infinita de um vetor obtido pela subtração
47   * de dois vetores
48   * @param xk Vetor x(k+1)
49   * @param x Vetor x(k)
50   * @param n Dimensão dos vetores
51   * @return norma do vetor obtido pela subtração
52   */
53  long double norma_infinita(long double *xk, long double *x, int n){
54      int i;
55      long double maximo = 0;
56
57      for(i = 0; i < n; i++)
58          if(fabs(xk[i] - x[i]) > maximo)
59              maximo = fabs(xk[i] - x[i]);
60
61      return maximo;
62  }
63
64  /**
65   * Função para resolver o sistema linear usando o método de gauss-seidel
66   * @param A Matriz de funções do sistema linear
67   * @param b Resultados das equações do sistema linear
68   * @param x Vetor contendo os resultados iniciais
69   * @param n Dimensão do sistema linear
70   * @param e Precisão
71   * @param itmax Número máximo de iterações
72   */
73  long double *gauss_seidel(long double **A, long double *b, long double *x, int n, long
↵ double e, int itmax){
74      // Variáveis auxiliares
75      long double somaL;
76      long double somaU;
77      long double *x_ant;
78
79      // Iteradores
80      int i;
81      int j;
82      int it = 0;
83
84      // Alocando espaço da memória para o vetor auxiliar
85      x_ant = malloc(sizeof(long double) * n);
86
87      do{
88          // Para toda iteração, atualiza o vetor x anterior
89          for(i = 0; i < n; i++)
90              x_ant[i] = x[i];
91
92          for(i = 0; i < n; i++){
93              somaL = 0;
94              somaU = 0;
95              // Somatório da parte de baixo da matriz

```

```

96     for(j = 0; j < i; j++)
97         somaL += A[i][j] * x[j];
98     // Somatório da parte de cima da matriz
99     for(j = i + 1; j < n; j++)
100         somaU += A[i][j] * x[j];
101     // Aproximação do x
102     x[i] = (b[i] - somaL - somaU)/A[i][i];
103 }
104
105 // Calcula a norma infinita e compara com a tolerância
106 if(norma_infinita(x, x_ant, n) <= e){
107     free(x_ant);
108     return x;
109 }
110
111     it++;
112 }while(it < itmax);
113
114 // Libera memória
115 free(x_ant);
116 return x;
117 }

```


Considerações finais