

# 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  #include <stdio.h>
2  #include <stdlib.h>
3  #include <gauss_seidel.h>
4
5  int main (int argc, char *argv[]){
6      // Variáveis de entrada
7      int n;                // Quantidade de equações do sistema linear
8      int itmax;            // Número máximo de iterações
9      long double **A;      // Matriz de sistemas lineares
10     long double *b;       // Vetor resultado das equações
11     long double *x;       // Vetor inicial
12     long double e;        // Erro permitido, precisão
13
14     // Iteradores
15     int i;
16     int j;
17
18     // Dimensão da matriz (número de equações do sistema linear)
19     scanf("%d", &n);
20
21     // Alocação da matriz
22     A = malloc(sizeof(long double *) * n);
23     for (i = 0; i < n; i++)
24         A[i] = malloc(sizeof(long double) * n);
25
26     // Pegando valores de A
27     for (i = 0; i < n; i++)
28         for (j = 0; j < n; j++)
29             scanf("%Lf", &(A[i][j]));
30
31     // Alocação do vetor resultado
32     b = malloc(sizeof(long double) * n);
33
34     // Pegando os valores de b
35     for(i = 0; i < n; i++)
36         scanf("%Lf", &b[i]);
37
38     // Alocação do vetor chute
39     x = malloc(sizeof(long double) * n);
40
41     // Pegando os valores do x(0), o vetor inicial
42     for(i = 0; i < n; i++)
43         scanf("%Lf", &x[i]);
44
45     // Pegando o valor da precisão (erro permitido)
46     scanf("%Lf", &e);
47
48     // Pegando a quantidade máxima de iterações
49     scanf("%d", &itmax);
50
51     x = gauss_seidel(A, b, x, n, e, itmax);
52
53     imprime_vetor(x, n);
54
55     // Liberando a memória
56     free(x);
```

```

57     free(b);
58     for (i = 0; i < n; i++)
59         free(A[i]);
60     free(A);
61
62     return 0;
63 }

```

## 2.2 Biblioteca auxiliar

### 2.2.1 Header (gauss\_seidel.h)

```

1  #ifndef GAUSS_SEIDEL_H
2  #define GAUSS_SEIDEL_H
3
4  void imprime_matriz(long double **A, int n);
5  void imprime_vetor(long double *v, int n);
6  long double norma_infinita(long double *xk, long double *x, int n);
7  long double *gauss_seidel(long double **A, long double *b, long double *x, int n, long
↪   double e, int itmax);
8
9  #endif

```

### 2.2.2 Implementação da biblioteca (gauss\_seidel.c)

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  #include <gauss_seidel.h>
5
6  /**
7   * Função para imprimir de forma mais legível uma matriz quadrada
8   * @param A Matriz a ser impressa
9   * @param n dimensão da matriz
10  */
11 void imprime_matriz(long double **A, int n){
12     int i;
13     int j;
14
15     for (i = 0; i < n; i++){
16         for (j = 0; j < n; j++){
17             printf("%Lf\t", A[i][j]);
18             printf("\n");
19         }
20     }
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     int i;
29
30     for (i = 0; i < n; i++)

```

```

31     printf("%.16Lf\n", v[i]);
32 }
33
34 /**
35  * Função para retornar a norma infinita de um vetor obtido pela subtração
36  * de dois vetores
37  * @param xk Vetor x(k+1)
38  * @param x Vetor x(k)
39  * @param n Dimensão dos vetores
40  * @return norma do vetor obtido pela subtração
41  */
42 long double norma_infinita(long double *xk, long double *x, int n){
43     int i;
44     long double maximo = 0;
45
46     for(i = 0; i < n; i++)
47         if(fabs(xk[i] - x[i]) > maximo)
48             maximo = fabs(xk[i] - x[i]);
49
50     return maximo;
51 }
52
53 /**
54  * Função para resolver o sistema linear usando o método de gauss-seidel
55  * @param A Matriz de funções do sistema linear
56  * @param b Resultados das equações do sistema linear
57  * @param x Vetor contendo os resultados iniciais
58  * @param n Dimensão do sistema linear
59  * @param e Precisão
60  * @param itmax Número máximo de iterações
61  */
62 long double *gauss_seidel(long double **A, long double *b, long double *x, int n, long
↵ double e, int itmax){
63     // Variáveis auxiliares
64     long double somaL;
65     long double somaU;
66     long double *x_ant;
67
68     // Iteradores
69     int i;
70     int j;
71     int it = 0;
72
73     // Alocando espaço da memória para o vetor auxiliar
74     x_ant = malloc(sizeof(long double) * n);
75
76     do{
77         for(i = 0; i < n; i++)
78             x_ant[i] = x[i];
79
80         for(i = 0; i < n; i++){
81             somaL = 0;
82             somaU = 0;
83             for(j = 0; j < i; j++)
84                 somaL += A[i][j] * x[j];
85             for(j = i + 1; j < n; j++)
86                 somaU += A[i][j] * x[j];
87             x[i] = (b[i] - somaL - somaU) / A[i][i];
88         }
89

```

```
90     if(norma_infinita(x, x_ant, n) <= e){
91         free(x_ant);
92         return x;
93     }
94
95     it++;
96 }while(it < itmax);
97
98 free(x_ant);
99 return x;
100 }
```

Considerações finais