Prisma infinito

João Medeiros (joaomedeiros@ect.ufrn.br)

Alocação de memória em C

- Alocação estática
  double V[N][N]
- Alocação dinâmica, ponteiros
  double **V;

Alocação dinâmica de vetor

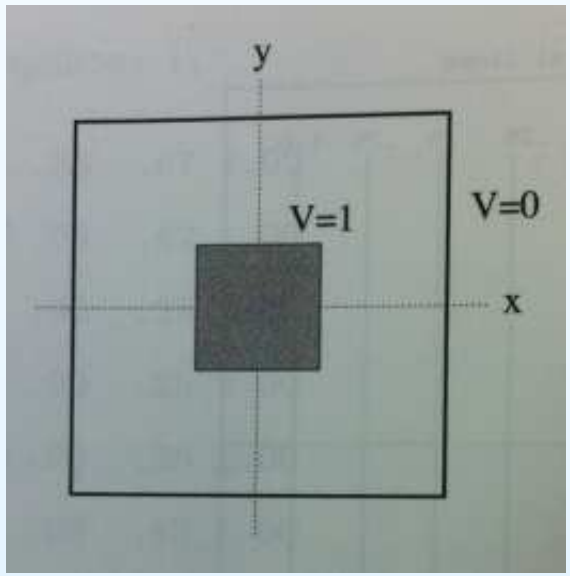
```
main() {
    double *Vetor;
    Vetor = alocaVetor(N);
}
double * alocaVetor(int tam) {
    double *vetor;
    // aloca as linhas
    vetor = (double *) malloc(sizeof(double)*tam);
    // sempre teste se a alocacao foi realizada com sucesso
    if( !vetor) {
        fprintf(stderr,"Erro ao alocar vetor\n");
        exit(1);
    }
 return vetor;
}
```

Alocação dinâmica de matriz

```
main() {
    double **V;
    V = alocaMatriz(N, N);
}
double ** alocaMatriz(int nlinhas, int ncolunas) {
    double **matriz;
    int i;
    // aloca as linhas
    matriz = (double **) malloc(sizeof(double)*nlinhas);
    // sempre teste se a alocacao foi realizada com sucesso
    if( !matriz) {
        fprintf(stderr,"Erro ao alocar linhas da matriz\n");
        exit(1);
    }
    for(i=0; i<nlinhas;i++) {
     matriz[i] = (double *) malloc(sizeof(double)*ncolunas);
    // sempre teste se a alocacao foi realizada com sucesso
     if( !matriz[i]) {
       fprintf(stderr,"Erro ao alocar colunas da matriz\n");
       exit(1);
     }
  }
```

Exercícios

1. Modifique o programa desenvolvido nessa aula para resolver a equação de laplace para a situação mostrada na figura abaixo. Que esquematiza um prisma infinito, na direção z, com uma parte central condutora. O potencial nas paredes externas do prisma é nulo e na parte interna é mantida e a parte interna dimensão de 0.6 unidades. Utilize $dx = 0.1$.



Prisma

- Iremos utilizar o $dx = 0.1$.
- Número de pontos na matriz

$$N = 1 + (Xmax - Xmin)/dx;$$

- Conversão entre as coordenadas cartesianas e os índices da matriz, lembre que estamos considerando $dx = dy$ e $Xmax = Ymax$.

$$i = (x + Xmax)/dx$$

$$j = (Xmax - y)/dx$$

Prisma

- iremos precisar determinar os limites dos laços para o quadrado interno.

```
// o termo +0.5 serve para evitar erros de arredondamentos
iinicial = (xinicial + Xmax)/dx +0.5;
jinicial = (Xmax - yinicial)/dx +0.5;
largura = larguraInterna / dx + 0.5;
```

i↓j→	0	1	2	3	4	5	6	7	8	9
0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0
1	V=0									V=0
2	V=0									V=0
3	V=0			V=1	V=1	V=1	V=1			V=0
4	V=0			V=1	V=1	V=1	V=1			V=0
5	V=0			V=1	V=1	V=1	V=1			V=0
6	V=0			V=1	V=1	V=1	V=1			V=0
7	V=0									V=0
8	V=0									V=0
0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0

Prisma

Caso particular

- $dx = 0.1, Xmax = 1, larguraInterna = 0.6, xinicial = -0.3, yinicial = 0.3$

```
iinicial = (xinicial + Xmax)/dx +0.5;
jinicial = (Xmax - yinicial)/dx +0.5;
largura = larguraInterna / dx + 0.5;
```

- temos

$$iinicial = (-0.3 + 1.0)/0.1 + 0.5 = 7 \quad (1)$$

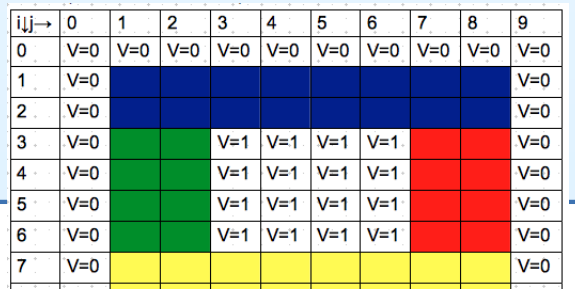
$$jinicial = (1.0 - 0.3)/0.1 + 0.5 = 7 \quad (2)$$

$$largura = 0.6/0.1 + 0.5 = 6 \quad (3)$$

Prisma

Inicia

```
void inicia(double **V, double **Vn1) {
    int i,j;
    // Vamos inicilizar todos os elementos de matriz com zero
    for(i=0; i<N; i++) {
        for(j=0; j<N; j++) {
            V[i][j]=Vn1[i][j]=0.0;
        }
    }
    for(i=iinicial; i<= iinicial + largura; i++) {
        for(j=jinicial; j<= jinicial + largura; j++) {
            V[i][j]=Vn1[i][j]=1.0;
        }
    }
}
```



i↓j→	0	1	2	3	4	5	6	7	8	9
0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0
1	V=0									V=0
2	V=0									V=0
3	V=0			V=1	V=1	V=1	V=1			V=0
4	V=0			V=1	V=1	V=1	V=1			V=0
5	V=0			V=1	V=1	V=1	V=1			V=0
6	V=0			V=1	V=1	V=1	V=1			V=0
7	V=0									V=0

Prisma

- Podemos usar 4 laços para atualizar a matriz
- Um laço para cada região: azul, verde, vermelha e amarela.

i↓j→	0	1	2	3	4	5	6	7	8	9
0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0
1	V=0									V=0
2	V=0									V=0
3	V=0			V=1	V=1	V=1	V=1			V=0
4	V=0			V=1	V=1	V=1	V=1			V=0
5	V=0			V=1	V=1	V=1	V=1			V=0
6	V=0			V=1	V=1	V=1	V=1			V=0
7	V=0									V=0
8	V=0									V=0
0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0

Prisma

Região azul

```
for(i=1; i<iinicial; i++) {
    for(j=1; j<N-1; j++) {
        Vn1[i][j] = 0.25*( V[i+1][j] + V[i-1][j] +
                V[i][j+1] + V[i][j-1] );
        deltaV += fabs(V[i][j] - Vn1[i][j]);
    }
}
```

i↓j→	0	1	2	3	4	5	6	7	8	9
0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0
1	V=0									V=0
2	V=0									V=0
3	V=0			V=1	V=1	V=1	V=1			V=0
4	V=0			V=1	V=1	V=1	V=1			V=0
5	V=0			V=1	V=1	V=1	V=1			V=0
6	V=0			V=1	V=1	V=1	V=1			V=0
7	V=0									V=0
8	V=0									V=0
0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0

Prisma

Região verde

```
for(i=iinicial; i<=iinicial+largura; i++) {
    for(j=1; j<jinicial; j++) {
        Vn1[i][j] = 0.25*( V[i+1][j] + V[i-1][j] +
                V[i][j+1] + V[i][j-1] );
        deltaV += fabs(V[i][j] - Vn1[i][j]);
    }
}
```

i↓j→	0	1	2	3	4	5	6	7	8	9
0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0
1	V=0									V=0
2	V=0									V=0
3	V=0			V=1	V=1	V=1	V=1			V=0
4	V=0			V=1	V=1	V=1	V=1			V=0
5	V=0			V=1	V=1	V=1	V=1			V=0
6	V=0			V=1	V=1	V=1	V=1			V=0
7	V=0									V=0
8	V=0									V=0
0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0

Prisma

Região vermelha

```
for(i=iinicial; i<=iinicial+largura; i++) {
    for(j=jinicial+largura; j<N-1; j++) {
        Vn1[i][j] = 0.25*( V[i+1][j] + V[i-1][j] +
                V[i][j+1] + V[i][j-1] );
        deltaV += fabs(V[i][j] - Vn1[i][j]);
    }
}
```

i↓j→	0	1	2	3	4	5	6	7	8	9
0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0
1	V=0									V=0
2	V=0									V=0
3	V=0			V=1	V=1	V=1	V=1			V=0
4	V=0			V=1	V=1	V=1	V=1			V=0
5	V=0			V=1	V=1	V=1	V=1			V=0
6	V=0			V=1	V=1	V=1	V=1			V=0
7	V=0									V=0
8	V=0									V=0
0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0

Prisma

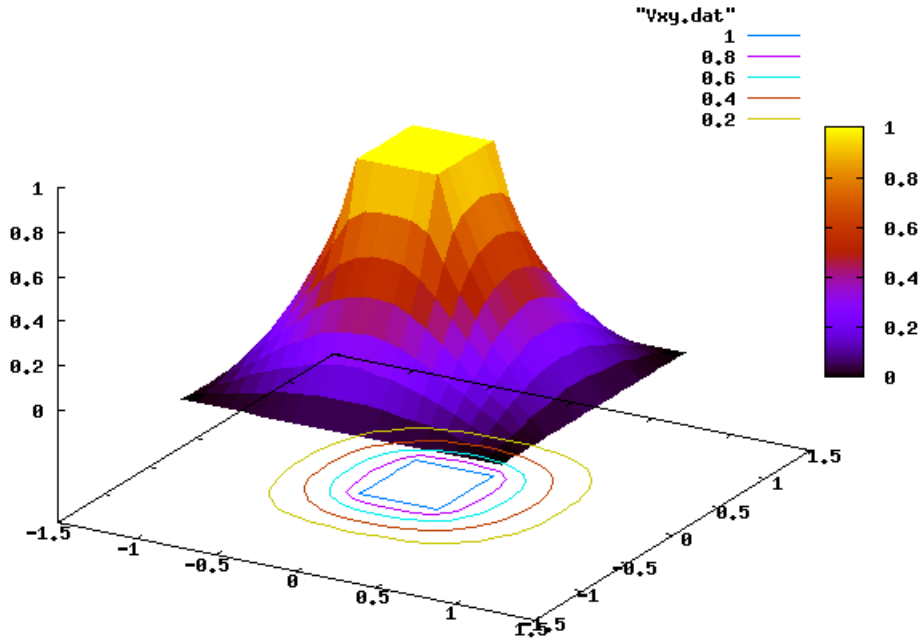
Região amarela

```
for(i=iinicial+largura+1; i<N-1; i++) {
    for(j=1; j<N-1; j++) {
        Vn1[i][j] = 0.25*( V[i+1][j] + V[i-1][j] +
                V[i][j+1] + V[i][j-1] );
        deltaV += fabs(V[i][j] - Vn1[i][j]);
    }
}
```

i↓j→	0	1	2	3	4	5	6	7	8	9
0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0
1	V=0									V=0
2	V=0									V=0
3	V=0			V=1	V=1	V=1	V=1			V=0
4	V=0			V=1	V=1	V=1	V=1			V=0
5	V=0			V=1	V=1	V=1	V=1			V=0
6	V=0			V=1	V=1	V=1	V=1			V=0
7	V=0									V=0
8	V=0									V=0
0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0	V=0

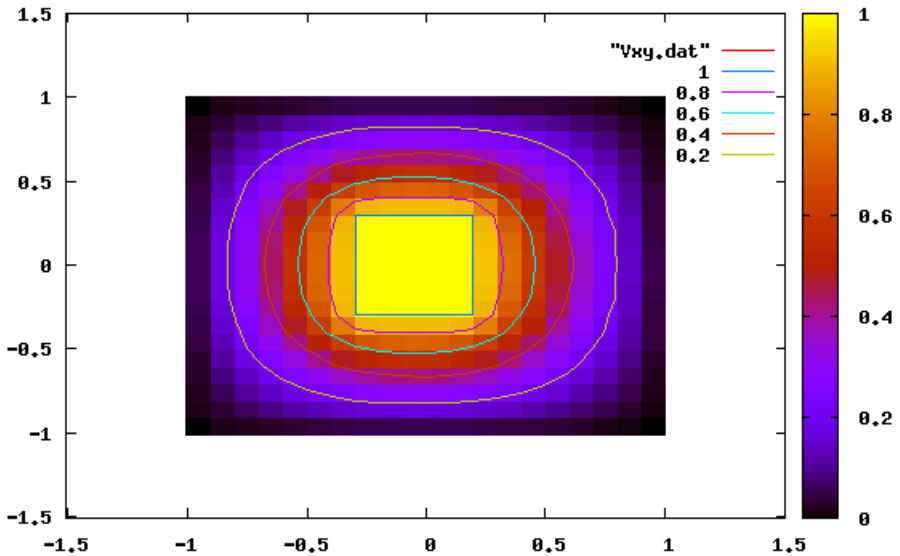
Visualizando os resultados - Potencial

```
set xrange [-1.5:1.5]
set yrange [-1.5:1.5]
set pm3d
unset surface
set hidden3d
set contour base
set term png
set output ”PrismaVxy.png”
splot ”Vxy.dat”with lines
```



Visualizando os resultados - Equipotenciais

```
set xrange [-1.5:1.5]
set yrange [-1.5:1.5]
set view map
set pm3d
set surface
set hidden3d
set contour base
set term png
set output ”PrismaVxy2.png”
splot ”Vxy.dat”with lines
```



Visualizando os resultados - Campo elétrico

```
set xrange [-1.5:1.5]
set yrange [-1.5:1.5]
set view map
unset pm3d
set surface
set hidden3d
set contour base
set term png
set output ”PrismaExy.png”
plot ”Exy.dat”using
($1):($2):($3/8):($4/8) with vec
```