MC102 - Algoritmos e Progração de Computador

Prof. Alexandre Xavier Falcão

11° Aula: Operações com Vetores

1 Operações com vetores

Uma das aplicações para vetores é a representação de sinais e funções discretas. Em telecomunicações, por exemplo, é muito comum obter amostras de um sinal elétrico, que representa um trecho de um sinal de voz, e armazená-las em um vetor. O processamento do sinal no computador é essencialmente uma seqüência de operações aplicadas ao vetor. O sinal processado pode então ser transformado de volta em sinal elétrico, e por sua vez transformado em som.

1.1 Reflexão

Seja $f_1(x)$ um sinal discreto definido no intervalo de inteiros $[0, n) = 0, 1, \ldots, n-1$. Assumimos que $f_1(x) = 0$ fora deste intervalo. A reflexão $f_2(x) = f_1(-x)$ do sinal em torno da origem é um sinal definido no intervalo $(-n, 0]$, com valores nulos fora deste intervalo (Figura 1). Muito embora as amostras desses sinais estejam definidas em intervalos diferentes do eixo x, ambos são armazenados em vetores com n posições, cujos índices variam de 0 a $n-1$. O problema consiste em encontrar primeiro a relação entre x e os índices i e j dos vetores de $f1[i]$ e $f2[j]$, que é diferente para cada caso, e depois encontrar a relação entre i e j. Para $f1[i]$, $i = x = 0, 1, \ldots, n-1$, mas para $f2[j]$, $j = -x + n - 1$ e $x = -n + 1, -n + 2, \ldots, 0$. Então, a relação entre i e j é $j = -i + n - 1$, $i = 0, 1, \ldots, n-1$, $j = n-1, n-2, \ldots, 0$; e a reflexão implica em fazer $f2[-i + n - 1] \leftarrow f1[i]$.

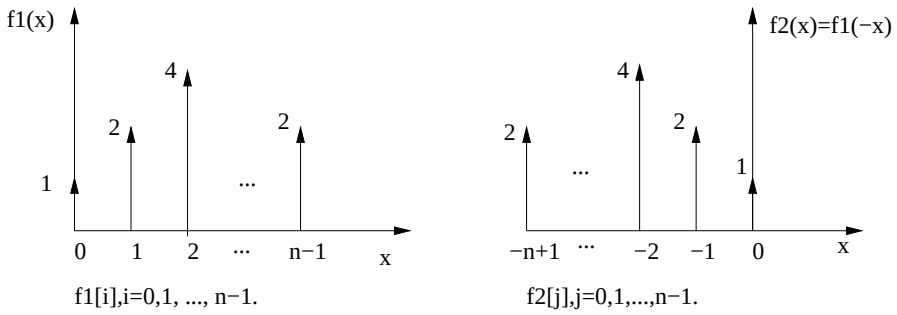

Figura 1: Reflexão de um sinal discreto.

```
#include <stdio.h>

#define N 100

int main()
{
  float f1[N],f2[N];
  int i,n;

  printf("Entre com o número de amostras\n");
  scanf("%d",&n);
  printf("Entre com os valores das amostras\n");
  for (i=0; i < n; i++)
    scanf("%f",&f1[i]);

  /* calcula a reflexão */

  for (i=0; i < n; i++)
    f2[-i+n-1] = f1[i];

  printf("Vetor resultante\n");
  for (i=0; i < n; i++)
    printf("%5.2f ",f2[i]);
  printf("\n");

  return(0);
}
```

1.2 Convolução

A **convolução** entre dois sinais discretos $f_1(x)$, $x \in [0, n_1)$, e $f_2(x)$, $x \in [0, n_2)$, é um terceiro sinal discreto $f_3(x)$, $x \in [0, n_1 + n_2 - 1)$, com $n_1 + n_2 - 1$ amostras (Figura 2).

$$f_3(x) \quad = \quad \sum_{x'=-\infty}^{x'=+\infty} f_1(x') f_2(x - x'). \tag{1}$$

O sinal $f_3(x)$ é calculado pela soma do produto entre $f_1(x')f_2(x - x')$ ao deslizarmos $f_2(x - x')$ sobre o eixo x' para cada deslocamento x. No entanto, $f_3(x) \neq 0$ apenas para $x \in [0, n_1 + n_2 - 1)$. Sendo i, j, k os índices dos vetores $f_1[i]$, $f_2[j]$ e $f_3[k]$, respectivamente, temos que $x' = i$, $x = k$, e $x - x' = j$ (i.e., $j = k - i$).

A convolução pode ser usada para filtrar o sinal, suavizando transições abruptas (e.g. $f_2(x) = \{1, 2, 1\}$), detectando transições abruptas (e.g. $f_2(x) = \{-1, 2, -1\}$), ou realçando essas transições (e.g. $f_2(x) = \{1, -1\}$). Na maioria dos casos, no entanto, usa-se $f_2(x + n_2/2)$ deslocada para que a origem $x = 0$ fique na amostra do meio. Isto apenas desloca o sinal $f_3(x)$ para $f_3(x + n_2/2)$, não afetando o algoritmo nem o conteúdo do vetor.

```
#include <stdio.h>
#define N1 100
#define N2 9
#define N3 110

int main()
{
  float f1[N1],f2[N2],f3[N3];
  int i,j,k,n1,n2,n3;

  printf("Entre com o número de amostras\n");
  scanf("%d",&n1);
  printf("Entre com os valores das amostras\n");
  for (i=0; i < n1; i++)
    scanf("%f",&f1[i]);

  printf("Entre com o número de coeficientes do filtro\n");
  scanf("%d",&n2);
  printf("Entre com os valores dos coeficientes\n");
  for (i=0; i < n2; i++) /* ler f2 sem reflexão */
    scanf("%f",&f2[i]);

  n3 = n1+n2-1;

  /* calcula a convolução */

  for (k=0; k < n3; k++) {
    f3[k]=0.0;
    for (i=0; i < n1; i++){
      j = k-i; // reflexao
      if ((j >= 0)&&(j < n2))
        f3[k] += f1[i]*f2[j];
    }
  }
  printf("Vetor resultante\n");
  for (k=0; k < n3; k++)
    printf("%5.2f ",f3[k]);
  printf("\n");
  return 0;
}
```

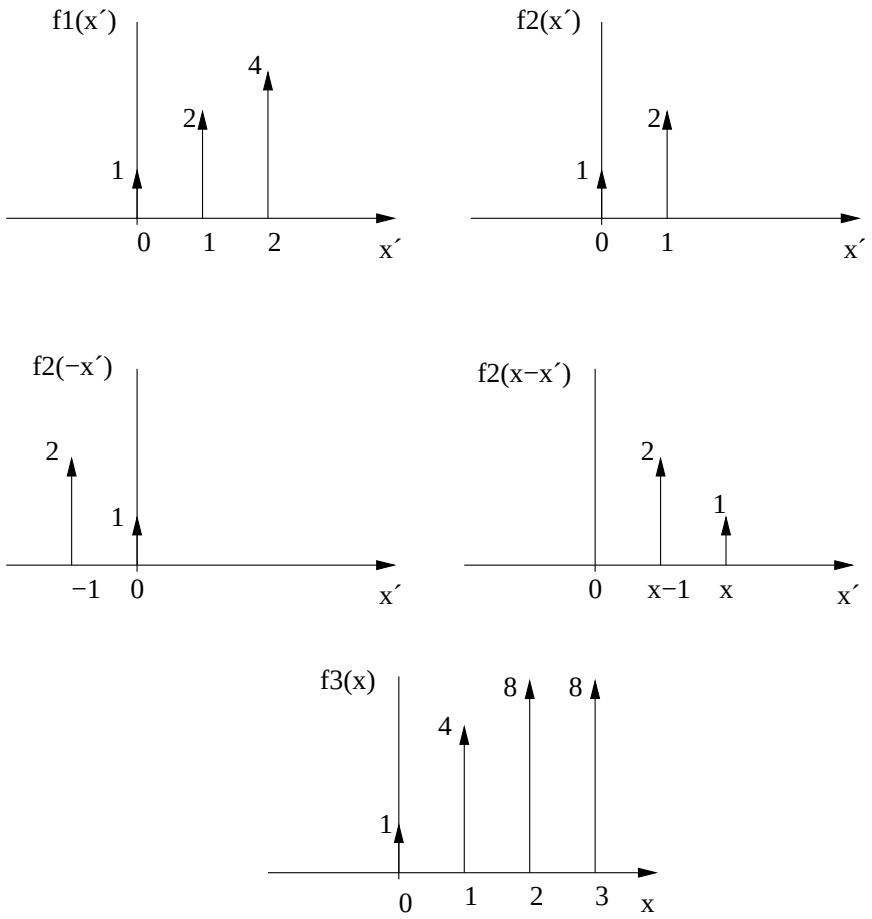

Figura 2: Convolução entre sinais discretos.

1.3 Correlação e correlação circular

A **correlação** entre dois sinais discretos é definida como

$$f_3(x) = \sum_{x'=-\infty}^{x'=+\infty} f_1(x') f_2(x + x'). \tag{2}$$

Sua implementação é muito parecida com a da convolução, exceto que $j = i + k$. A correlação é normalmente usada para alinhar (registrar) dois sinais, minimizando a distância entre eles. Neste caso, porém, ela deve ser calculada de forma circular. Supondo que $f_1(x)$ e $f_2(x)$ possuem o mesmo número n de amostras (caso contrário, podemos completar com zeros o sinal com menos amostras), a correlação circular é definida por:

$$f_3(x) = \sum_{x'=0}^{x'=n-1} f_1(x') f_2((x + x')\%n). \tag{3}$$

Note que, $f_3(x) \neq 0$ para $x \in [0, n)$, e os índices i, j, k de $f_1[i]$, $f_2[j]$, e $f_3[k]$ ficam $x = k$, $x' = i$, e $j = (i + k)\%n$. O valor máximo de $f_3(x)$ representa o deslocamento circular x necessário para o alinhamento de $f_2(x)$ com $f_1(x)$.

```
#include <stdio.h>
#define N 100

int main()
{
  float f1[N],f2[N],f3[N];
  int i,j,k,n,imax;

  printf("Entre com o número de amostras dos sinais\n");
  scanf("%d",&n);
  printf("Entre com os valores das amostras do 1o. sinal\n");
  for (i=0; i < n; i++)
    scanf("%f",&f1[i]);
  printf("Entre com os valores das amostras do 2o. sinal\n");
  for (i=0; i < n; i++)
    scanf("%f",&f2[i]);

  /* calcula a correlação circular */

  for (k=0; k < n; k++) {
    f3[k]=0.0;
    for (i=0; i < n; i++){
      f3[k] += f1[i]*f2[(i+k)%n];
    }
  }

  printf("Vetor resultante\n");
  for (i=0; i < n; i++)
    printf("%5.2f ",f3[i]);
  printf("\n");

  /* alinha o vetor f2 com f1 */
  imax = 0;
  for (i=1; i < n; i++) /* encontra o máximo */
    if (f3[i] > f3[imax])
      imax = i;

  for (i=0; i < n; i++) /* alinha f2 copiando o resultado para f3 */
    f3[i] = f2[(i+imax)%n];
  printf("Vetor alinhado\n");
  for (i=0; i < n; i++)
    printf("%5.2f ",f3[i]);
  printf("\n");
  return 0;
}
```

2 Exercícios

1. O histograma de um sinal discreto $f(x)$ com n amostras e L valores inteiros no intervalo $[0, L-1]$ é uma função discreta $h(l)$, $l = 0, 1, \ldots, L-1$, onde $h(l)$ é o número de ocorrências do valor $f(x) = l$, para $x = 0, 1, \ldots, n-1$. Escreva um programa para calcular o histograma de um sinal discreto.

2. Seja $p[i]$, $i = 0, 1, \ldots, n-1$, um vetor que armazena em cada posição i o coeficiente a_i de um polinômio de grau $n-1$: $a_0 + a_1 x^1 + \ldots + a_{n-1} x^{n-1}$. Faça um programa para ler um dado polinômio em p e avaliar seus valores para diferentes valores de x lidos da entrada padrão.