Determinação Automática de Escalas em Gráficos

ADELARDO ADELINO DANTAS DE MEDEIROS

LECA - Laboratório de Engenharia de Computação e Automação
Departamento de Engenharia Elétrica
UFRN - CT - DEE
Campus Universitário
59072-970 Natal RN Brasil
E-mail: leca@brufrn.bitnet(.br)

Abstract - This paper shows an algorithm for, if we know the maximum and minimum values the coordinates of a set of points that we want to present on a graphic can assume, determining the suitable numerical bounds to the axes where those points will be presented.

1. Introdução

O primeiro passo quando se pretende representar um conjunto de ordenadas e abscissas em um dispositivo de saída com capacidade gráfica (impressora, monitor de vídeo, etc.) é estabelecer uma correspondência entre as dimensões da área de trabalho do dispositivo que se pretende empregar e o intervalo do conjunto dos números reais que se quer representar ao longo de cada um dos eixos.

O estabelecimento dos limites da área de trabalho usualmente é feito com base no sistema de coordenadas e na resolução do dispositivo gráfico e no percentual da área disponível que se pretende utilizar. Por exemplo, caso se queira que a representação ocupe o quarto superior direito de um monitor VGA que tem resolução de 640x480 *pixels* e coordenadas do canto superior esquerdo (0,0), esses limites seriam $i_{sup}=0$, $i_{inf}=239$, $j_{esq}=320$ e $j_{dir}=639$, utilizando-se a letra "i" para denotar as linhas e "j" para as colunas.

Caso sejam conhecidos os limites (lx_{min}, lx_{max}, ly_{min} e ly_{max}) dos intervalos que se quer representar ao longo dos eixos (eixo x horizontal e eixo y vertical) é trivial estabelecer-se a correspondência entre os dois sistemas de coordenadas:

$$j = \frac{j_{esq}(lx_{max}-x)+j_{dir}(x-lx_{min})}{lx_{max}-lx_{min}}$$

$$i = \frac{i_{inf}(ly_{max}-y)+i_{sup}(y-ly_{min})}{ly_{max}-ly_{min}}$$

Os limites dos intervalos muitas vezes se confundem com os valores extremos das coordenadas a serem representadas (vx_{min}, vx_{max}, vy_{min} e vy_{max}). Porém, em algumas situações é desejável que essas grandezas assumam valores diferentes. Por exemplo, suponha que se deseja representar em um gráfico um conjunto de pontos (x,y) gerados a partir da função $y=2{,}05+2{,}87\text{sen}(x)$, com x variando de 0 a 3π.

No caso, $vx_{min}=0$, $vx_{max}=3\pi$, $vy_{min}=-0{,}82$ e $vy_{max}=4{,}92$. Pode-se fazer $lx_{min}=0$, $lx_{max}=9{,}425$, $ly_{min}=-0{,}82$ e $ly_{max}=4{,}92$; todavia, é mais apropriado utilizar-se $lx_{min}=0$, $lx_{max}=10$, $ly_{min}=-1$ e $ly_{max}=5$, pois números "quebrados" nos limites dos eixos tornam mais difícil a visualização, conforme pode ser visto no gráfico da figura 1.

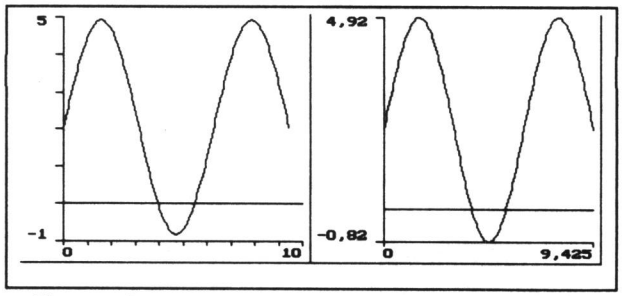

Figura 1: Comparação entre limites ajustados e não ajustados

A escolha dos limites não apresenta dificuldades para os habituadas a esse trabalho. Contudo, não é imediata a definição de um conjunto de regras (ou heurísticas) que permita sistematizar esse procedimento. Ou, em outras palavras, nosso problema consiste em expressar um conhecimento composto de fatos simples em regras lógicas [Rich (1988)].

Nesse artigo, a partir de observações sobre o comportamento humano usual na resolução desse problema, são apresentadas algumas regras que devem ser obedecidas na determinação dos limites e os algoritmos que as implementam, bem como uma codificação em C e exemplos de utilização.

2. Determinação pelo número mínimo de algarismos significativos

Os limites dos eixos dependem dos valores extremos das coordenadas. Para que todos os pontos tenham representação no espaço de trabalho, deve-se ter $lx_{min} \leq vx_{min}$, $lx_{max} \geq vx_{max}$, $ly_{min} \leq vy_{min}$ e $ly_{max} \geq vy_{max}$[1].

O primeiro procedimento de determinação que se sugere é adotar-se como limite máximo (mínimo) o menor (maior) número com um algarismo significativo que seja maior (menor) ou igual que o valor máximo (mínimo) a ser representado. Por exemplo,

[1] Como o problema é idêntico para os dois eixos, em alguns casos nos referiremos apenas aos limites do eixo (l_{min} e l_{max}) e aos valores extremos da coordenada (v_{min} e v_{max}), sem especificar de qual dimensão se trata.

se $v_{min}=1,03$ e $v_{max}=2,97$, faz-se $l_{min}=1$ e $l_{max}=3$; se $v_{min}=10,3$ e $v_{max}=29,7$, $l_{min}=10$ e $l_{max}=30$.

Este algoritmo, bastante simples, deve sofrer alguns aprimoramentos. O principal deles é que os dois limites, além de terem um único algarismo significativo, devem ser da mesma ordem de grandeza. Exemplificando, para $v_{min}=-0,88$ e $v_{max}=6,72$, deve-se fazer $l_{min}=-1$ e $l_{max}=7$, e não $l_{min}=-0,9$ e $l_{max}=7$.

Com esta modificação chega-se ao procedimento da figura 2, que determina os limites e o número de marcas igualmente espaçadas que devem ser feitas ao longo do eixo para indicação dos valores notáveis. Na figura 3 tem-se um exemplo de emprego do algoritmo em um caso onde $v_{ymax}=6,72$ e $v_{ymin}=-0,88$, sendo os pontos gerados a partir da função $y=2,92-3,8\sin(x)$.

```
Parâmetros de Entrada
|  Reais: vmax,vmin
Fim Parâmetros de Entrada

Parâmetros de Saída
|  Reais: lmax,lmin
|  Inteiros: nmarc
Fim Parâmetros de Saída

Variáveis
|  Reais: expo,y
|  Inteiras: imax,imin,expoente
Fim Variáveis

Início Programa
|  Se (vmax+vmin>=0)
|  |  y:=vmax
|  Senão
|  |  y:=-vmin
|  Fim Se
|  expoente:=RoundDown(Log10(y)) (*)
|  expo:=10.0^expoente
|  imax:=0
|  Se (vmax>=0.0)
|  |  Enquanto (imax*expo<vmax)
|  |  |  imax:=imax+1
|  |  Fim Enquanto
|  Senão
|  |  Enquanto ((imax-1)*expo>=vmax)
|  |  |  imax:=imax-1
|  |  Fim Enquanto
|  Fim Se
|  imin:=0
|  Se (vmin>0.0)
|  |  Enquanto ((imin+1)*expo<=vmin)
|  |  |  imin:=imin+1
|  |  Fim Enquanto
|  Senão
|  |  Enquanto (imin*expo>vmin)
|  |  |  imin:=imin-1
|  |  Fim Enquanto
|  Fim Se
|  ..................................  <- Ponto A
|  nmarc:=imax-imin+1
|  ..................................  <- Ponto B
|  lmax:=imax*expo
|  lmin:=imin*expo
|  Retorne(lmax,lmin,nmarc)
Fim Programa

(*) RoundDown(x)=maior inteiro menor ou igual que x
```

Figura 2: Procedimento para determinação dos limites com um único algarismo significativo

3. Determinação pela taxa mínima de ocupação

Empregando-se o procedimento da figura 2 a um conjunto de pontos gerados a partir da função $y=250+2,8\sin(x)$, obtém-se o resultado da figura 4, que claramente não é satisfatório. O problema acontece porque, no caso, $v_{ymax}=252,8$ e $v_{ymin}=247,2$. O menor número com um algarismo significativo maior que 252,8 é 300, enquanto que o maior número com um algarismo significativo menor que 247,2 é 200. Desta forma, o intervalo numérico associado ao eixo é grande em comparação com a magnitude de variação dos valores da coordenada.

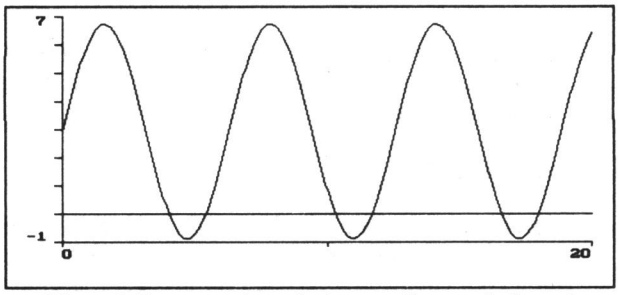

Figura 3: Emprego adequado do procedimento da figura 2

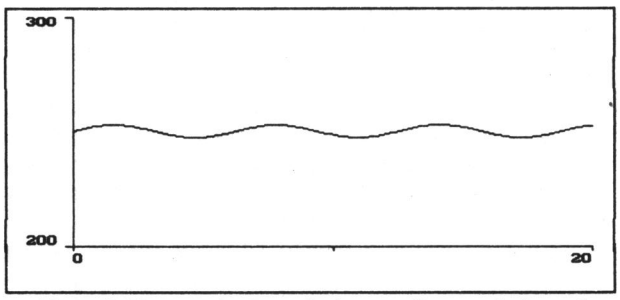

Figura 4: Emprego inadequado do procedimento da figura 2

Assim, surge a necessidade de incorpor-se algum refinamento adicional ao procedimento básico. A opção escolhida foi prever-se uma taxa mínima de ocupação do eixo. Caso o percentual de ocupação efetiva esteja abaixo desse limite pré-estabelecido, procede-se, tantas vezes quantas necessárias ou até que seja atingido o número máximo de algarismos significativos, aos seguintes passos:

a) aumenta-se de um a quantidade de algarismos significativos dos limites;
b) diminui-se o máximo possível o limite máximo; e
c) aumenta-se o limite mínimo o máximo possível.

Para tanto, o procedimento básico deve sofrer os acréscimos apresentados na figura 5. Na figura 6 mostra-se o resultado do emprego do procedimento atualizado (com os acréscimos da figura 5) no caso do início dessa seção, onde $v_{ymax}=252,8$ e $v_{ymin}=247,2$, fixando-se uma taxa mínima de ocupação dos eixos de 80% (TAXA_MIN_OCUP = 0,8).

4. Determinação pelo número de marcas

Na forma como se encontra no momento, o algoritmo ainda gera resultados indesejáveis em alguns casos no que diz respeito ao número de marcas. Se esta preocupação não for relevante, pode-se adotar o procedimento visto até aqui.

Para exemplificar o problema basta que se tente empregar o procedimento da figura 5 em um conjunto de pontos gerados a partir da função $y=100+12{,}4\sin(x)$. O resultado está na figura 7, onde se vê que o número de marcas no eixo é excessivo.

```
Definições
    TAXA_MIN_OCUP = ##
    NUM_MAX_DIGIT = ##
Fim Definições

Variáveis
    Inteiras: i
Fim Variáveis

Início Bloco ..... Ponto A ->
    i:=1
    Enquanto (((vmax-vmin) < TAXA_MIN_OCUP*(imax-
                imin)*expo) E (i < NUM_MAX_DIGIT))
        i:=i+1
        expoente:=expoente-1
        expo:=10^expoente
        imax:=10*imax
        imin:=10*imin
        Enquanto ((imax-1)*expo>=vmax)
            imax:=imax-1
        Fim Enquanto
        Enquanto ((imin+1)*expo<=vmin)
            imin:=imin+1
        Fim Enquanto
    Fim Enquanto
Fim Bloco
```

Figura 5: Procedimento adicional para determinação dos limites com taxa mínima de ocupação

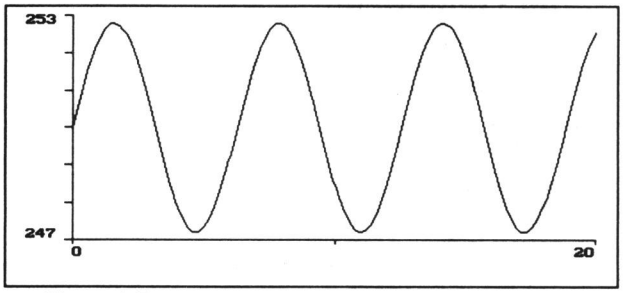

Figura 6: Emprego adequado do procedimento da figura 5

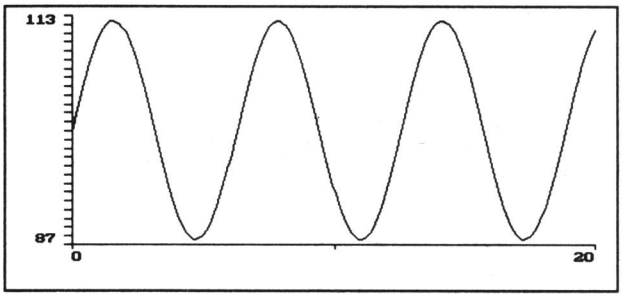

Figura 7: Emprego inadequado do procedimento da figura 5

Deve-se ter, então, uma definição sobre a faixa admissível para a quantidade de marcas no eixo. Caso o número de marcas sugerido pelo algoritmo esteja fora dessa faixa, deve-se aumentá-lo ou diminuí-lo (talvez até alterando os limites) conforme a situação.

Caso o número de marcas esteja abaixo do mínimo, a solução é simples: basta inserir uma nova marca entre cada duas existentes e proceder desta forma até que o número desejado seja atingido.

Sendo o número de marcas excessivo, a solução possível é procurar-se, no universo admissível, aquele número de marcas que melhor se ajusta aos limites do eixo. A forma de escolha que nos parece mais racional é feita com base nos seguintes critérios:

a) caso o conjunto de pontos inclua um valor nulo ou tenha tanto elementos positivos quanto negativos, uma das marcas deve coincidir com o número 0 (zero);
b) satisfeita a condição anterior, escolhe-se o número de marcas que implicar em menor aumento no comprimento do eixo; e
c) atendidos os requisitos precedentes, a opção recairá sobre o menor (ou o maior, a depender do gosto pessoal) número de marcas.

A incorporação desse refinamento ao algoritmo pode ser feita através dos acréscimos da figura 8.

```
Definições
    NUM_MIN_MARC = ##
    NUM_MAX_MARC = ##
Fim Definições

Variáveis
    Inteiras: resid,rmin,zero,zmin,exces,emin,passo
Fim Variáveis

Início Bloco ..... Ponto B ->
    Enquanto (nmarc<NUM_MIN_MARC)
        nmarc:=2*nmarc-1
    Fim Enquanto
    Se (nmarc>NUM_MAX_MARC)
        zmin:=Numero Muito Grande
        rmin:=Numero Muito Grande
        emin:=Numero Muito Grande
        Para i de NUM_MIN_MARC a NUM_MAX_MARC (*)
            passo:=RoundUp((imax-imin)/(i-1))
            Se (imax>=0 E imin<=0)
                zero:=imin+passo*RoundUp(-imin/passo)
            Senão
                zero:=0
            Fim Se
            resid:=(i-1)*passo-(imax-imin)
            Se (resid>=0)
                exces:=0
            Senão
                exces:=zero-resid
            Fim Se
            Se((emin!=0 E exces=0)OU(resid<=rmin E
                    ((emin!=0)OU(emin=0 E exces=0))))
                nmarc:=1
                emin:=exces
                rmin:=resid
                zmin:=zero
            Fim Se
        Fim Para
        Se (emin=0)
            imin:=imin-zmin
            imax:=imax+rmin-zmin
        Senão
            imin:=imin-rmin
        Fim Se
    Fim Se
Fim Bloco

(*) RoundUp(x)=menor inteiro maior ou igual que x
```

Figura 8: Procedimento adicional para determinação dos limites com número de marcas apropriado

Na figura 9 mostra-se o resultado do emprego do algoritmo final (com os acréscimos da figura 8) no caso apresentado no início dessa seção, onde

$v_{y_{max}} = 112{,}4$ e $v_{y_{min}} = 87{,}6$. Foi adotada uma taxa mínima de ocupação de 80% e a faixa admissível para o número de marcas é de 4 a 12.

5. Conclusões

A necessidade de geração desse algoritmo surgiu quando se passou a desenvolver uma biblioteca para interfaceamento homem-máquina que englobasse diversas formas de comunicação entre o *software* e o usuário. No presente momento, as funções de determinação de escala estão sendo usadas por mais de 20 usuários no desenvolvimento de seus próprios programas, não tendo sido registrado até o momento qualquer anomalia na lógica do algoritmo. Em anexo se encontra a listagem de uma implementação do procedimento em Turbo C 2.0 [BORLAND (1988)].

A aplicação mais natural e imediata desse procedimento é na visualização científica de funções no domínio do tempo ou de sinais temporais. Contudo pode-se empregá-lo igualmente no desenvolvimento de *software* para CAD em várias áreas, como Arquitetura, Engenharia Civil e Projetos Mecânicos, onde a visualição gráfica consiste em um dos aspectos mais importantes do problema [Besant (1985)].

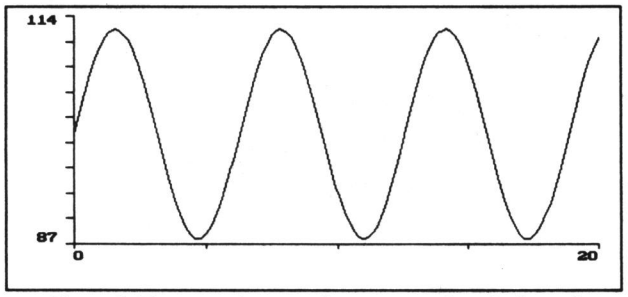

Figura 9: Emprego do procedimento completo da figura 8

Quanto às constantes a serem definidas, os melhores resultados têm sido obtidos com TAXA_MIN_OCUP entre 0,8 e 0,9 e a faixa de número de marcas admissíveis de 4 (no mínimo 3) a 12 (pode ser mais ou menos de acordo com a dimensão do eixo). O número máximo de algarismos significativos que os limites podem ter fica condicionado à quantidade de dígitos que um inteiro pode conter. Para inteiros de 2 bytes (-32768 a +32767), por exemplo, os limites só poderão ter até 4 algarismos significativos (+9999 a -9999).

6. Bibliografia

RICH, E. *Inteligência Artificial*. McGraw-Hill, 1988.

BORLAND INTERNATIONAL. *Turbo C 2.0 User's Guide* e *Turbo C 2.0 Reference's Guide*. Borland International Inc., 1988.

BESANT, C.B. *CAD/CAM - Projeto e Fabricação com Auxílio de Computador*. Ed. Campus, 1985.

Função de determinação de limites de escala em C

```
void escala(float vmax, float vmin, float *lmax,
                                    float *lmin, int *NMarcas)
{
   #define OCUP_MIN 0.8
   #define MAX_DIGIT 4
   #define MIN_MARCAS 4
   #define MAX_MARCAS 12

   float y,expo;
   int expoente,passo,i;
   int imax,imin,resid,rmin,zero,zmin,exces,emin;

   if (vmax==vmin) if (vmax==0)
   {
      vmax=1.0;
      vmin=-1.0;
   }
   else
   {
      vmax += 0.1*fabs(vmax);
      vmin -= 0.1*fabs(vmin);
   }
   if (vmax+vmin>=0) y=vmax; else y=-vmin;
   expoente = (int)floor(log10(y));
   expo = pow10(expoente);
   imax = 0;
   if (vmax>=0) while (imax*expo<vmax) imax++;
   else while ((imax-1)*expo>=vmax) imax--;
   imin = 0;
   if (vmin>0) while ((imin+1)*expo<=vmin) imin++;
   else while (imin*expo>vmin) imin--;
   i=1;
   while(((vmax-vmin)<OCUP_MIN*(imax-imin)*expo)
                                          &&(i<MAX_DIGIT))
   {
      i++;
      expoente--;
      expo = pow10(expoente);
      imax = 10*imax;
      imin = 10*imin;
      while ((imax-1)*expo>=vmax) imax--;
      while ((imin+1)*expo<=vmin) imin++;
   }
   *NMarcas = imax-imin+1;
   if (*NMarcas>MAX_MARCAS)
   {
      *NMarcas = 0;
      zmin = 30000;
      rmin = 30000;
      emin = 30000;
      for (i=MIN_MARCAS; i<=MAX_MARCAS; i++)
      {
         passo = (int)ceil((float)(imax-imin)/(i-1));
         if (imax>=0 && imin<=0)
         {
            zero = imin + passo*ceil(-(float)imin/passo);
         }
         else zero = 0;
         resid = (i-1)*passo-(imax-imin);
         if (resid>=zero) exces = 0;
         else exces = zero-resid;
         if ((emin!=0 && exces==0)||(resid<=rmin &&
                     ((emin!=0)||(emin==0 && exces==0))))
         {
            *NMarcas = i;
            emin = exces;
            rmin = resid;
            zmin = zero;
         }
      }
      if (emin==0)
      {
         imin -= zmin;
         imax += rmin-zmin;
      }
      else imin -= rmin;
   }
   else while(*NMarcas<MIN_MARCAS)
   {
      *NMarcas += *NMarcas-1;
   }
   *lmax = imax*expo;
   *lmin = imin*expo;
}
```