

**SIMULAÇÃO NUMÉRICA EM DINÂMICA DOS FLUIDOS APLICADO A  
SOLUÇÃO DO PROBLEMA DO ESCOAMENTO FORÇADO SOBRE UMA  
CAVIDADE ABERTA**

Ramiro de Oliveira Castro Prado<sup>49</sup> (UBC, Bolsista PIBIC/CNPq)  
Dr. Maurício Ribeiro Baldan<sup>50</sup> (LAS/INPE)

**RESUMO**

Neste trabalho estamos interessados na fundamentação básica em técnicas computacionais aplicada a dinâmica dos fluidos. Com o objetivo de discretizar as equações diferenciais parciais (EDP) em uma região  $R$ , fizemos um estudo simples, mas necessário, ao entendimento da solução de uma EDP do ponto de vista computacional. No caso de técnicas numéricas não é possível tratar uma região  $R$  como contínua, devemos então escolher alguns pontos dentro de  $R$  e somente neles calcular a solução do problema. Para o cálculo de aproximações em diferenças finitas utilizamos a expansão em série de Taylor e interpolação polinomial. Fizemos um exemplo simples onde calculamos aproximações por diferenças finitas progressivas e atrasadas, este cálculo foi realizado tanto do ponto de vista analítico como numérico onde um programa em C foi elaborado para isto. Uma outra técnica de obtenção das aproximações de diferenças finitas é a interpolação polinomial. A função  $f$  é aproximada por um polinômio de grau arbitrário e os coeficientes desse polinômio são determinados em função dos valores de  $f$  sobre os quais o polinômio passa. Para uma maior clareza e entendimento de uma interpolação foram estudadas três técnicas diferentes de interpolação: interpolação polinomial, por Lagrange e Newton. Uma pequena rotina em MATLAB foi utilizada para visualizar as etapas de interpolação.

---

<sup>49</sup>Aluno do Curso de Bacharelado e Tecnologia em Ciências Aeronáuticas, UBC. E-mail: [ramiro@las.inpe.br](mailto:ramiro@las.inpe.br)

<sup>50</sup>Pesquisador do Laboratório Associado de Sensores e Materiais. E-mail: [baldan@las.inpe.br](mailto:baldan@las.inpe.br)

temperatura da superfície do mar (TSM) acima da média no Atlântico Tropical Sul e abaixo da média no Atlântico Norte.

Para este episódio, foi selecionado o período de 16 a 22 de janeiro de 1999. Conforme as classificações da Revista Climanálise (1999a), em janeiro, o fenômeno La Niña encontrava-se em sua fase madura. As TSMs no Pacífico Equatorial Central e Oriental registraram desvios negativos de temperatura entre  $-1.5$  e  $-3.0^{\circ}$  C. O campo de anomalias de pressão ao nível médio do mar (PNMM) apresentou anomalias positivas no setor tropical oriental e central do Pacífico, e anomalias negativas no setor ocidental, contribuindo para a manutenção do La Niña. A situação sinótica deste episódio é mostrada nas figuras 3.9 e 3.10, para os campos de vento em 925 e 200 hPa, respectivamente. No nível de 925 hPa, observa-se que os ventos alísios apresentam-se mais intensos na Região do Pacífico Equatorial Central do que no Episódio de El Niño; nota-se também, a presença da ZCIT em torno de  $5^{\circ}$  N. Em 200 hPa, verifica-se uma circulação anticiclônica próximo à costa do Peru e Chile e, uma região de ventos de oeste, relativamente mais fortes, próximo ao equador, entre  $100^{\circ}$  W e  $150^{\circ}$  W.

### Agradecimentos

Meus agradecimentos ao CNPq / PIBIC pela bolsa concedida e ao INPE, os quais deram condições a oportunidade de adquirir novos conhecimentos.



## **SUMÁRIO**

### **CAPÍTULO I**

- Introdução
- Objetivo
- Estruturação

### **CAPÍTULO II**

- Fundamentação Teórica
  - Equações Diferenciais Parciais
  - Equações de Diferenças Finitas
    - Erro de Arredondamento
- Interpolação Polinomial

### **CAPÍTULO III**

- Materiais e Métodos

### **CAPÍTULO IV**

- Resultados, Análises e Conclusão

## **REFERÊNCIAS BIBLIOGRÁFICAS**

## **ANEXOS**

# CAPÍTULO I

## 1. Introdução.

A simulação numérica é uma alternativa empregada a partir do advento do computador digital, como complemento as análises teóricas e as técnicas experimentais da mecânica dos fluidos (DFC - Dinâmica de Fluidos Computacional), basicamente, consiste na realização de simulações do comportamento de fenômenos físicos, utilizando-se de métodos computacionais, que permitem estudar o escoamento de fluidos com transmissão de calor e/ou massa, tensões, entre outros [1].

Devido a complexidade das equações abrangentes no movimento do fluido, as técnicas computacionais de solução exigem grande poder de processamento, que neste projeto de Investigação Científica, necessitam de interpretação, para um bom entendimento dos procedimentos numéricos envolvidos a solucionar o problema.

Através das técnicas experimentais, pode-se dividir o regime do escoamento em dois tipos: laminares, aqueles que em regime de escoamentos viscosos, se caracterizam pelo movimento suave em camadas, Istoé, as partículas se movem em lâminas planas bem definidas, sem que haja fuga de uma lâmina para outra, e, os

turbulentos, que se caracterizam por partículas em movimentos aleatórios, tridimensionais, com grandes flutuações de velocidade [3].

Tratado de se obter um modelo computacional, é necessário expressar de forma adequada as equações e a região (domínio) em que elas são válidas. Já que uma vez definido o problema, as equações diferenciais parciais não são tratadas de uma forma contínua, mas sim como um conjunto discreto e finito de pontos. A idéia geral do método de diferenças finitas é a discretização do domínio e a substituição das derivadas presentes na equação diferencial por aproximações envolvendo somente valores numéricos da função em pontos particulares [6].

Ao conjunto discreto de pontos distribuídos no domínio, que é a cavidade, dá-se o nome de malha, esta é a idéia fundamental das técnicas numéricas: a dimensão espacial é dividida em um conjunto discreto de pontos ou células e o tempo é discretizado em pequenos intervalos e neles pode-se obter um conjunto de equações em função dos valores das propriedades em pontos adjacentes da malha. Este procedimento resulta em um sistema de equações algébricas, geralmente lineares, que representam as equações diferenciais parciais no espaço computacional.

A descrição deste projeto, foi dada seqüencialmente por uma metodologia dialética de leitura sistemática do livro-texto de Armando de Oliveira Fortuna [1], e o entendimento e discretização das equações mais importantes em DFC.

## 1.1 Objetivo

Este trabalho de Iniciação Científica tem como objetivo a familiarização com as metodologias numéricas modernas, abordando principalmente os respectivos fundamentos teóricos, dando ênfase as técnicas computacionais (DFC). No uso desta ferramenta de análise numérica, relaciona-se a fundo a teoria e desenvolvimento apropriado aos métodos numéricos aplicados a solução de um fluxo de corrente incompressível, típico escoamento confinado. Escoamento este, considerado em coordenadas ortogonais bidimensionais: todas as linhas de corrente emanam de um ponto central (0), desprezando qualquer valor de alteração da densidade ao longo do escoamento, permitindo, a compreensão da dinâmica de escoamento forçado pelo movimento de uma tampa de uma caixa ou cavidade modelada pela Equação de Navier-Stokes (fig. 1a).

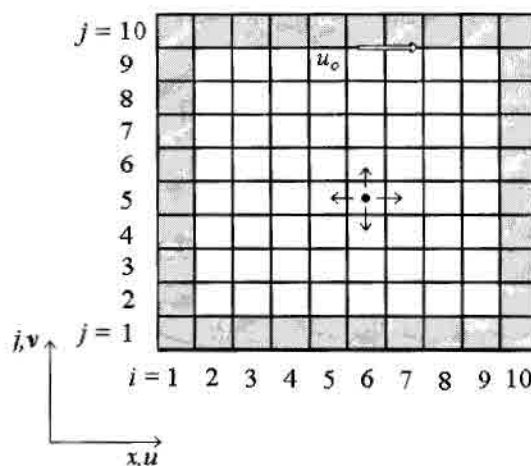


Fig. 1a – Cavidade discretizada por meio de uma malha estruturada



## 1.2 – Estruturação.

As atividades deste projeto foram divididos em etapas seguindo o livro-texto do Fortuna [1].

As linguagens C e MATLAB, foram utilizadas na elaboração dos exemplos propostos no livro texto. Um tempo considerável foi dedicado a esta parte do trabalho.

## CAPÍTULO II

### 2.1 Equações Diferenciais Parciais

Através do livro-texto [Fortuna] foi demonstrado os três tipos básicos de EDPs, e salientando a diferença entre eles, tanto do ponto de vista físico como matemático.

Tratados de obter de soluções numéricas, para aplicar-se técnicas computacionais. As EDPs descrevem os fenômenos de interesse da dinâmica de fluidos computacional, classificando em três categorias básicas: *Elíptica*, *Parabólica e Hiperbólica*.

*Elíptica*

$$\left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \right) \quad \text{Eq .04}$$

Quando se tem equilíbrio nas propriedades de interesse com o tempo.

*Parabólica*

$$\left( \frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \right) \quad \text{Eq .05}$$

Quando apresentam mecanismo de dissipação de energia em evolução temporal efetiva da grandeza física.

*Hiperbólica*

$$\left( = \frac{\partial \phi}{\partial t} = -v \frac{\partial \phi}{\partial x} \right) \quad \text{Eq .06}$$

Quando não apresentam fenômenos dissipativos de calor e no escoamento de fluidos viscosos mas, se encontram em evolução temporal efetiva da grandeza física.

Em problemas que envolvem EDPs, podem-se especificar três tipos básicos de condições iniciais ou de fronteira sobre a variável dependente (i) condições de *Dirichlet*, condições de *Neumann*, e condições *Periódicas*.

Obviamente, um problema pode apresentar uma mistura dos três tipos básicos de condições de fronteira. Nesse caso, dizemos que ele possui condições de fronteira *mistas*.

Para as condições iniciais ou de fronteira especificadas incorretamente, por “falta” ou “excesso”, freqüentemente fazem com que a solução não seja a única ou mesmo fisicamente plausível. Pequenas variações das condições auxiliares devem produzir mudanças compatíveis a solução numérica. O contrário sugere que alguma condição auxiliar está incorretamente especificada. Portanto, é necessário muito cuidado na especificação das condições auxiliares do problema estudado.

## 2.2 Equações de Diferenças Finitas

A idéia geral do método de diferenças finitas é escolher alguns pontos dentro do Domínio e somente neles calcular a solução do problema. Este processo denominado de discretização do domínio, substitui as derivadas [6] presentes nas equações diferenciais por aproximações destes pontos envolvendo somente valores numéricos da função  $f$ . Desta maneira, encontrando os termos que aparecem nas

formas numéricas das EDP's escritas em função dos pontos da malha. Que esta relacionada as expressões que identificam os valores das grandezas nos pontos discretos  $(i\Delta x, j\Delta y)$  da região de Domínio, como na fig. 2.

As expressões encontradas na discretização dos pontos são denominadas de aproximações por diferenças finitas. O resultado dá-se de uma equação algébrica, dos pontos discretizados em um sistema de coordenadas na intersecção das linhas horizontais com as verticais, separados entre si, por uma distância (fig.2)  $\Delta x$  e  $\Delta y$ . Por índices como  $i$  e  $j$  indentificam a  $i$ -ésima coluna e na  $j$ -ésima linha, respectivamente, denominando assim de equação de diferenças finitas (EDP's).

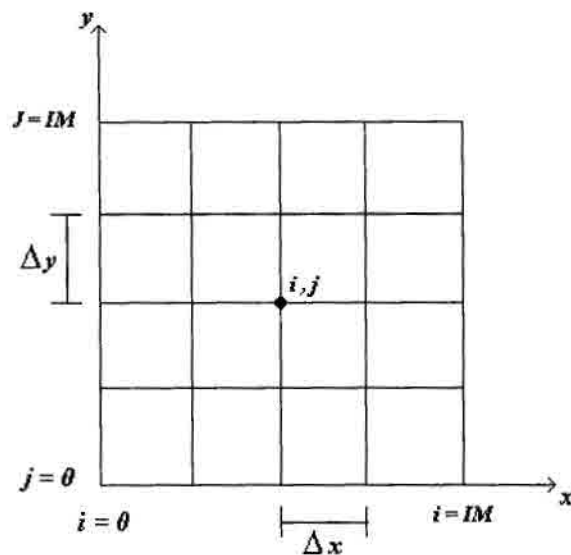


Fig 2 – Região Domínio discretizada.

Num exemplo, seja o  $X_0$  um número real do domínio em questão, e  $\Delta x$  um número positivo, definimos a malha de passo  $\Delta x$  associado a  $X_0$  como o conjunto de pontos.

$$x_i = x_0 \pm i\Delta x, \quad i = 1, 2, \dots, N.$$

Os pontos dessa malha serão calculadas as aproximações de uma função  $f$ . Os valores, serão utilizados na aproximação da derivada presente na equação diferencial, e, substituindo o operador diferencial contínuo (por exemplo  $\left(\frac{d}{dx}\right)$ ), por aproximação discreta em um número finito de pontos.

Ferramenta básica utilizada foi a expansão em série de Taylor :

$$f(x) = f(x_0) + \Delta x f'(x) + \frac{\Delta x^2}{2!} f''(x) + \dots + \frac{\Delta x^N}{N!} f^N(x) \quad \text{Eq. 07}$$

em que  $\Delta x = x - x_0$  e sendo  $\frac{\Delta x^N}{N!} f^N(x)$  chamado de  $RN$  que é o resto.

Considerando pontos uniformemente espaçados de  $x_i - x_{i-1} = \Delta x$ .



A expansão da série de Taylor serviu para escrever a aproximação no ponto  $x_i$  tendo  $x_i = i \Delta x$  (como única variável):

$$f(x + \Delta x) = f(x) + \Delta x f'(x) + \frac{\Delta x^2}{2!} f''(x) + \dots + \frac{\Delta x^N}{N!} f^N(x) \quad \text{Eq. 08}$$

Determinado a primeira derivada da função  $f$ , isolando a derivada de maior ordem:

$$\left. \frac{\partial f}{\partial x} \right|_i = \frac{f(x_i + \Delta x) - f(x_i)}{\Delta x} + \left[ -\frac{\Delta x^N}{N!} f^N(x) \right] \quad \text{Eq.09}$$

A expressão da Eq. 09, esta dividindo os termos da série pelo espaçamento  $\Delta x$ . Os termos isolados representam o erro local de truncamento, devido ao número finito de pontos, considerando a partir da segunda ordem, indicando o valor exato da derivada e sua aproximação numérica, e também, a variação da redução do espaçamento  $\Delta x$ , o refinamento da malha [1].

Para reduzir o erro, apresentamos mais pontos na malha, de forma que o erro local de truncamento sendo representado por  $O(\Delta x)$  apareça elevado a primeira potência, indicando variação devido ao refinamento.

Como usado na derivação de  $x_i$  um ponto adiante de  $x_i$ , no caso  $x_{i+1}$ , a representação da aproximação de primeira ordem para a primeira derivada de  $f$ , é também conhecida por diferença finita progressiva, que é dada como:

$$\left. \frac{\partial f}{\partial x} \right|_i = \frac{f_{i+1} - f_i}{\Delta x} + O(\Delta x) \quad \text{Eq.10}$$

Uma segunda aproximação de diferenças finitas pode ser obtida a partir da expansão de  $f(x_i - \Delta x)$  em série de Taylor em torno do ponto  $x_i$ :

$$f(x_i - \Delta x) = f(x_i) - \Delta x f'(x) + \frac{\Delta x^2}{2!} f''(x) + \dots + \frac{\Delta x^N}{N!} f^N(x) \quad \text{Eq.11}$$

Isolando a primeira derivada, temos:

$$\left. \frac{\partial f}{\partial x} \right|_i = \frac{f_i - f_{i-1}}{\Delta x} + O(\Delta x) \quad \text{Eq.12}$$

Após, a primeira aproximação para eliminarmos a segunda derivada de  $f$  usando novamente as expansões derivamos entre  $x_{i-1}$  e  $x_{i+1}$  encontramos:

$$\left. \frac{\partial f}{\partial x} \right|_i = \frac{f_{i+1} - f_{i-1}}{2\Delta x} + O(\Delta x)^2 \quad \text{Centrais} \quad \text{Eq. 13}$$

Podendo também ser escrita relacionada com  $f(i, j)$ :

$$f(x_i + \Delta x, j) + f(x_i - \Delta x, j) = 2f(x_i, j) + (\Delta x)^2 \left. \frac{\partial^2 f}{\partial x^2} \right|_{i,j} + O(\Delta x)^4 \quad \text{Eq. 14}$$

Reescrevendo:

$$\left. \frac{\partial^2 f}{\partial x^2} \right|_{i,j} = \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{(\Delta x)^2} + O(\Delta x^2) \quad \text{Eq. 15}$$

Relacionando os termos para em torno do ponto  $(x_i, y_j)$ , resolvemos as derivadas mistas:

$$\begin{aligned} f(x_i + \Delta x, y_j + \Delta y) = & f(x_i, y_j) + (\Delta x) \left. \frac{\partial f}{\partial x} \right|_{i,j} + (\Delta y) \left. \frac{\partial f}{\partial y} \right|_{i,j} + \\ & + \frac{(\Delta x)^2}{2!} \left. \frac{\partial^2 f}{\partial x^2} \right|_{i,j} + 2 \frac{(\Delta x)(\Delta y)}{2!} \left. \frac{\partial^2 f}{\partial x \partial y} \right|_{i,j} + \frac{(\Delta y)^2}{2} \left. \frac{\partial^2 f}{\partial y^2} \right|_{i,j} + \text{Erro} \end{aligned} \quad \text{Eq. 16}$$



### 2.2.1 Erro de Arredondamento

Para ilustrar o efeito do erro de arredondamento e do erro de truncamento sobre a aproximação de uma derivada, considere-se o cálculo numérico da primeira derivada de  $f(x) = 10 \cos(x)$  em  $x = \frac{\pi}{6}$ , utilizando as três fórmulas anteriores e diferentes espaços  $\Delta x$ . O valor exato da derivada é  $f'\left(\frac{\pi}{6}\right) = -10 \operatorname{sen}\left(\frac{\pi}{6}\right) = -5$ .

O erro relativo percentual (ER%) será definido como:

$$ER\% = \left| \frac{\text{solução\_exata} - \text{solução\_numérica}}{\text{solução\_exata}} \right| \cdot 100\% \quad \text{Eq. 17}$$

Através de uma linguagem de programação em C (ANEXO A), foi realizado um programa para calcular as derivadas parciais de primeira ordem, e, as aproximações por diferenças atrasadas e adiantadas por diferenças finitas, realizando também os cálculos das derivadas parciais de aproximação de primeira e segunda ordem, as diferenças centrais (ANEXO B), que cooperou obter a plotagem gráfica (fig. 3 e 4) no uso do programa Oringin, dando a possibilidade da confirmação visual da variação da redução do espaçamento dos pontos, com relação a aproximação numérica, que por fim viabiliza o refinamento da malha, o

erro relativamente (ER%) pequeno em relação espaçamento, até certa aproximação dos pontos devido ao número finito de termos da Série de Taylor, pelo truncamento apresentado nesta Série a partir da derivada da segunda ordem [1].

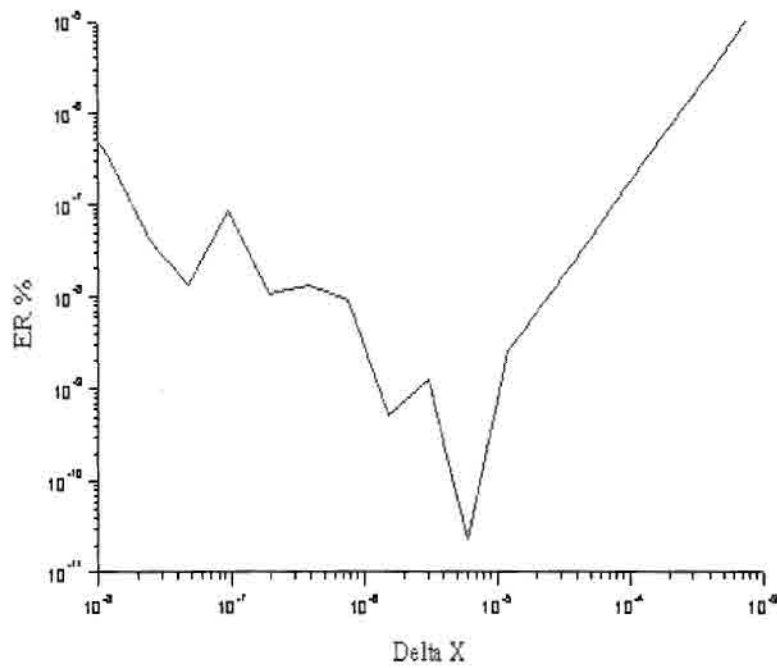


Fig. 3 e 4 – O valor de ER% em função do espaçamento  $\Delta x$ , no cálculo de  $f'\left(\frac{\pi}{6}\right)$  por diferenças centrais de segunda ordem.

### 2.3 Interpolação Polinomial

Esta é uma outra técnica de obtenção das aproximações de diferenças finitas. A função  $f$  é aproximada por um polinômio de grau arbitrário, e os coeficientes desse polinômio são determinados em função dos valores de  $f$  sobre os quais o polinômio passa [1].

Para simplificar esses cálculos e sem perda de generalidade, o eixo  $y$  foi colocado sobre o ponto  $x_i$  no qual queremos calcular a derivada. Assim, as coordenadas dos pontos são:  $x_i = 0$ ;  $x_{i-1} = -\Delta x$  e  $x_{i-2} = -2\Delta x$ .

Impondo que os pontos  $x_i$ ;  $x_{i-1}$  e  $x_{i-2}$  o polinômio  $P_2$  assuma os valores de  $f_i$ ,  $f_{i-1}$  e  $f_{i-2}$ , respectivamente, escrevendo aos pontos discretos de  $x_0$ ,  $x_1$  e  $x_2$ , podemos escrever um sistema em um polinômio de segundo grau

$$P_2(x) = ax^2 + bx + c:$$

$$x_2 = P_2(-2\Delta x) = f_{i-2} = 4a(\Delta x)^2 - 2b(\Delta x) + c \quad \text{Eq.18}$$

$$x_1 = P_2(-\Delta x) = f_{i-1} = a(\Delta x)^2 - b(\Delta x) + c \quad \text{Eq.19}$$

$$x_0 = P_2(0) = f_i = c \quad \text{Eq.20}$$

Por técnicas de substituição ou eliminação encontramos os valores para os coeficientes de  $P_2$ , em função de  $f_i$ ,  $f_{i-1}$  e  $f_{i-2}$ :

$$x_2 = a = \frac{f_{i-2} - 2f_{i-1} + f_i}{2(\Delta x)^2} \quad \text{Eq.21}$$

$$x_1 = b = \frac{f_{i-2} - 4f_{i-1} + 3f_i}{2(\Delta x)} \quad \text{Eq.22}$$

$$x_0 = c = f_i \quad \text{Eq.23}$$

Devido ao cancelamento de termos na série de Taylor, obtemos uma aproximação de  $O(\Delta x)^2$ .

Existem várias técnicas de se obter o polinômio de interpolação nas quais, verificamos algumas que implementamos. Destas, realizado programa em MATLAB destacamos: forma de Solução de Sistemas, forma de Lagrange e forma de Newton [7], conforme Anexos C , D e E respectivos. Seguindo a cada programa realizado os gráficos plotados em cada forma: Solução de Sistema conforme anexo C-1, Lagrange conforme anexo D-1,D-2, e de Newton no anexo E-1 e E-2.

## CAPÍTULO III

### 3.1 Materiais e Métodos

A abordagem deste projeto consistiu na leitura continuada e sistemática do livro-texto com algumas implementações numéricas de exemplos dos mais interessantes.

Para o desenvolvimento utilizou-se das ferramentas de programação de linguagem C, que é adotada pelo grupo, e, a implementação numérica com a ferramenta MATLAB para geração de programas de cálculos e gráficos para o melhor entendimento das formas de solução numérica.

## **CAPÍTULO IV**

### **4 Resultados, Análises e Conclusão**

O projeto dado início a leitura do livro-texto, convergiu com uma necessidade de aprendizagem e uso de linguagem de programação em ambiente LINUX, não somente pelo grupo, mas sim para o entendimento de cada um dos passos apresentados pelo autor.

Decisões sugeridas pelo orientador, como a implementação gráfica nas interpolações numéricas, serviram a enriquecer ao meu conhecimento, tendo uma grande importância para a minha formação profissional.

## ANEXO A

Este anexo mostra o programa em linguagem C, do calculo das diferenças de primeira e segunda ordem.

```
/* Aproximacao por diferencas finitas de primeira ordem & adiantadas, atrasadas e
centrais*/

#include <math.h>

#include <stdlib.h>

#include <string.h>

#include <stdio.h>

#define delta 0.1

#define var 0.5

#define iter 50

int main()
{
int i,j;

double fexata;

double delta1,delta2,delta3,pi,f;
```

```
double erad,erat,eradtr,fad[iter],fat[iter],fadtr[iter];
```

```
FILE *in, *out,*out1,*out2,*out3;
```

```
pi=4*atan(1.0);
```

```
printf("pi=%f\n",pi);
```

```
f=10*cos(pi/6);
```

```
fexata=-5;
```

```
printf("f=%f\n",f);
```

```
if((out1=fopen("aproxadicorrida.dat","w")) !=NULL)
```

```
    printf("abriu arquivo de saida .\n");
```

```
else
```

```
{
```

```
    printf("impossivel abrir arquivo de saida.\n");
```

```
}
```

```
if((out2=fopen("aproxatrascorrigida.dat","w")) !=NULL)
```

```
    printf("abriu arquivo de saida .\n");
```

```
else
```

```
{  
    printf("impossivel abrir arquivo de saida.\n");  
}
```

```
if((out3=fopen("aproxadiatrascorrigida.dat","w")) !=NULL)
```

```
    printf("abriu arquivo de saida .\n");
```

```
else
```

```
{  
    printf("impossivel abrir arquivo de saida.\n");  
}
```

```
/* Calculando aproximacao adiantada por diferencas finitas progressivas de  
primeira ordem para
```

```
f(x)=10cos(x) para x=pi/6 */
```

```
delta1=delta;
```

```
for(j=1;j<= iter; j++)
```

```
{
```

```
fad[j]=(10*cos((pi/6)+delta1)-f)/delta1;
```



```
erat=((fexata-fad[j])/fexata)*100;
```

```
fprintf(out1,"%d %16.10f %16.10f %16.10f\n",j,delta1,fad[j],erat);
```

```
delta1=delta1*var;
```

```
}
```

```
/* Calculando aproximacao atrasada por diferencas finitas atrasadas de primeira  
ordem para
```

```
f(x)=10cos(x) para x=pi/6 */
```

```
delta2=delta;
```

```
for(j=1;j<= iter; j++)
```

```
{
```

```
fat[j]=(f-10*cos((pi/6)-delta2))/delta2;
```

```
erat=((fexata-fat[j])/fexata)*100;
```

```
fprintf(out2,"%d %16.10f %16.10f %16.10f\n",j,delta2,fat[j],erat);
```

```
delta2=delta2*var;
```

```
}
```

```
/* Calculando aproximacao adiantadas e atrasadas por diferencas finitas centrais de
segunda ordem para
```

```
f(x)=10cos(x) para x=pi/6 */
```

```
delta3=delta;
```

```
for(j=1;j<= iter; j++)
```

```
{
```

```
fadtr[j]=((10*cos((pi/6)+delta3))-(10*cos((pi/6)-delta3)))/(2*delta3);
```

```
eradtr=((fexata-fadtr[j])/fexata)*100;
```

```
fprintf(out3,"%d %16.10f %16.10f %16.10f\n",j,delta3,fadtr[j],eradtr);
```

```
delta3=delta3*var;
```

```
}
```

```
/*for(i=1; i<=iter; i++)
```

```
    fprintf(out1,"%d %f\n",i,fad[i]);*/
```

```
fclose(out1);
```

```
/*for(i=1; i<=iter; i++)
```

```
fprintf(out2,"%d %f\n",i,fat[i]);*/
```

```
fclose(out2);
```

```
/*for(i=1; i<=iter; i++)
```

```
fprintf(out3,"%d %f\n",i,fadtr[i]);*/
```

```
fclose(out3);
```

```
}
```

## ANEXO B

Este anexo mostra o cálculo de aproximação numérica por diferenças centrais de

segunda ordem com espaçamento  $\Delta x$ , para  $f'\left(\frac{\pi}{6}\right) = -10\text{sen}\left(\frac{\pi}{6}\right) = -5$ .

	$\Delta x$	calculado	ER%
1	0.100000000000	-4.991670832341	0.163913046539
2	0.050000000000	-4.997916927068	0.038987810647
3	0.025000000000	-4.999479182942	0.007741857422
4	0.012500000000	-4.999869792684	0.007054636513
5	0.006250000000	-4.999967447980	0.002023704534
6	0.003125000000	-4.999991861983	0.002511997648
7	0.001562500000	-4.999997965495	0.002634071153
8	0.000781250000	-4.999999491373	0.002664589538
9	0.000390625000	-4.999999872843	0.002672219129
10	0.000195312500	-4.999999968213	0.002674126577
11	0.000097656250	-4.999999992052	0.002674603383
12	0.000048828125	-4.999999998015	0.002674722633
13	0.000024414063	-4.999999999501	0.002674752358
14	0.000012207031	-4.999999999874	0.002674759826

15	0.000006103516	-4.999999999999	0.002674762319
16	0.000003051758	-5.000000000064	0.002674763621
17	0.000001525879	-5.000000000027	0.002674762876
18	0.000000762939	-4.999999999546	0.002674753269
19	0.000000381470	-4.999999999347	0.002674749290
20	0.000000190735	-4.999999999479	0.002674751928
21	0.000000095367	-5.000000004297	0.002674848291
22	0.000000047684	-4.999999999327	0.002674748881
23	0.000000023842	-5.000000002183	0.002674805999
24	0.000000011921	-5.000000016589	0.002675094134
25	0.000000005960	-5.000000081200	0.002676386379
26	0.000000002980	-4.999999846477	0.002671691805
27	0.000000001490	-4.999999665306	0.002668068282
28	0.000000000745	-4.999999484862	0.002664459310
29	0.000000000373	-5.000001505250	0.002704868149
30	0.000000000186	-4.999997646082	0.002627682726
31	0.000000000093	-4.999994696118	0.002568681863
32	0.000000000047	-4.999991776422	0.002510286376
33	0.000000000023	-5.000024084002	0.003156455261

34	0.000000000012	-4.999962337315	0.001921488496
35	0.000000000006	-4.999915137887	0.000977474684
36	0.000000000003	-4.999868422747	0.000043146887
37	0.000000000001	-5.000385344028	0.010381849052
38	0.000000000001	-4.999397397041	0.009377619196
39	0.000000000000	-4.998642206192	0.024481840173
40	0.000000000000	-4.997894763947	0.039431084929
41	0.000000000000	-5.006165504456	0.125988149705
42	0.000000000000	-4.990358352661	0.190163342259
43	0.000000000000	-4.978275299072	0.431830877895
44	0.000000000000	-4.966316223145	0.671018793987
45	0.000000000000	-5.098648071289	1.975688960152
46	0.000000000000	-4.845733642578	3.082734911265
47	0.000000000000	-4.652404785156	6.949415481452
48	0.000000000000	-4.461059570313	10.776422138922
49	0.000000000000	-6.578369140625	31.570901927309

## ANEXO C

Programa de Interpolação por Forma de Solução de Sistema em MATLAB.

```
function interp_sistema
```

```
x=input('Entre os pontos de interpolação na forma [x0 x1 ... xn]');
```

```
y=input('Entre os valores da função na forma [y0 y1 ... yn]');
```

```
res=input('Voce deseja ver o grafico dos pontos da funcao dada? S/N [S] ','s');
```

```
if isempty(res)
```

```
res='S';
```

```
end
```

```
n = length(x);
```

```
hold;
```

```
xmin = min(x) - 1;
```

```
xmax = max(x) + 1;
```

```
ymin = min(y) - 1;
```

```
ymin = max(y) + 1;
```

```
axis([xmin xmax ymin ymax]);  
if((res=='S') | (res=='s'))  
for i=1:n  
plot(x(i),y(i), 'o');  
  
end  
  
disp('Pressione qualquer tecla para continuar')  
pause;  
  
end  
  
for i=1:n  
for j=1:n  
a(i,j) = x(i)^(j-1);  
  
end  
  
end
```



```

coef = a \ y';

res=input('Voce deseja ver o grafico do polinômio interpolador? S/N [S] ','s');

if isempty(res)

res='S';

end

if((res=='S') | (res=='s'))

h = (xmax - xmin)/100;

xx = xmin : h : xmax;

yy = polyval(fliplr(coef),xx);

plot(xx,yy);

end

ss=sprintf('Os coeficientes do polinomio interpolador na ordem a0 a1 a2 ... an
são:');

disp(' ');

disp(ss);

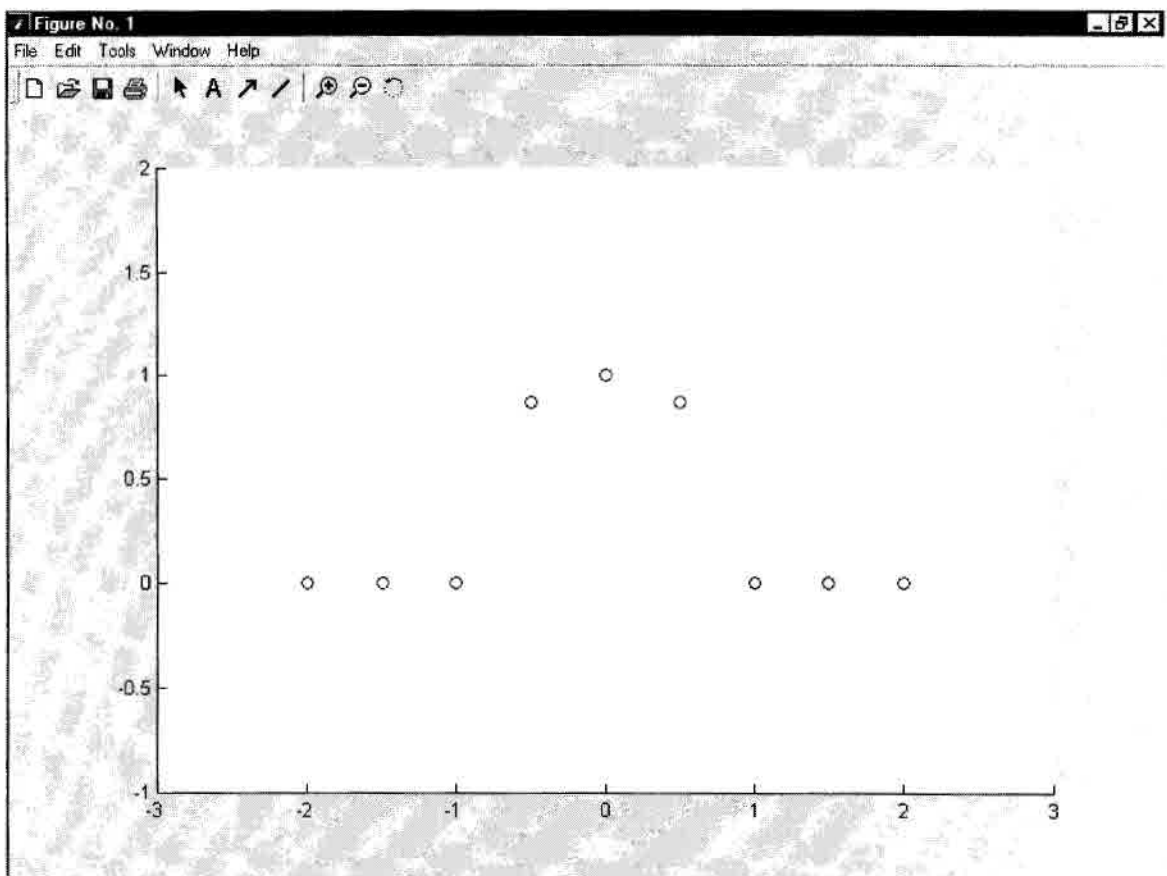
coef

```

## ANEXO C-1

Gráfico da função dada por um conjunto de pares de pontos de  $x_i$  e  $f(x_i)$  a que foram interpolados por forma de Solução de Sistema:

$$x_i = [-2 \ -1.5 \ -1 \ -0.5 \ 0 \ 0.5 \ 1 \ 1.5 \ 2] \text{ e } f(x_i) = [0 \ 0 \ 0 \ 0.87 \ 1 \ 0.87 \ 0 \ 0 \ 0].$$



## ANEXO D

Programa de Interpolação por forma de Lagrange em MATLAB.

```
function lagrange

x=input('Entre os pontos de interpolação na forma [x0 x1 ... xn]: ');
y=input('Entre os valores da função na forma [y0 y1 ... yn]: ');
xx=input('Entre o valor do ponto de avaliação do polinômio interpolador: ');
res=input('Voce deseja ver o grafico dos pontos da funcao dada? S/N [S] ','s');
if isempty(res)
res='S';

end

n = length(x);
xmin = min(x) - 1;
xmax = max(x) + 1;
ymin = min(y) - 1;
ymax = max(y) + 1;
```

```
if((res=='S') | (res=='s'))
hold;

axis([xmin xmax ymin ymax]);

for i=1:n
plot(x(i),y(i), 'o');

end

disp('Pressione qualquer tecla para continuar')

pause;

end

sol = pol(n, x, y, xx);

ss=sprintf('O valor do polinômio interpolador no ponto %f é: %f', xx,sol);

disp(' ');

disp(ss);

disp(' ');

res=input('Voce deseja ver o grafico do polinômio interpolador? S/N [S] ','s');
```

```
if isempty(res)
```

```
res='S';
```

```
end
```

```
if((res=='S') | (res=='s'))
```

```
h = (xmax - xmin)/100;
```

```
xxx = xmin : h : xmax;
```

```
m = length(xxx);
```

```
for i=1:m
```

```
yy(i) = pol(n, x, y, xxx(i));
```

```
end
```

```
plot(xxx,yy);
```

```
end
```

```
function solucao = pol(n, x, y, xx)
```

```
p = 1;
```

```
soma = 0;
```

```
for i=1:n
```

```
dd = xx-x(i);
```

```
if (dd == 0)
```

```
solucao = y(i);
```

```
return;
```

```
end
```

```
p = p*dd;
```

```
for j=1:i-1
```

```
dd = dd*(x(i)-x(j));
```

```
end
```

```
for j=i+1:n
```

```
dd = dd*(x(i)-x(j));
```

```
end
```

```
soma = soma + y(i)/dd;
```

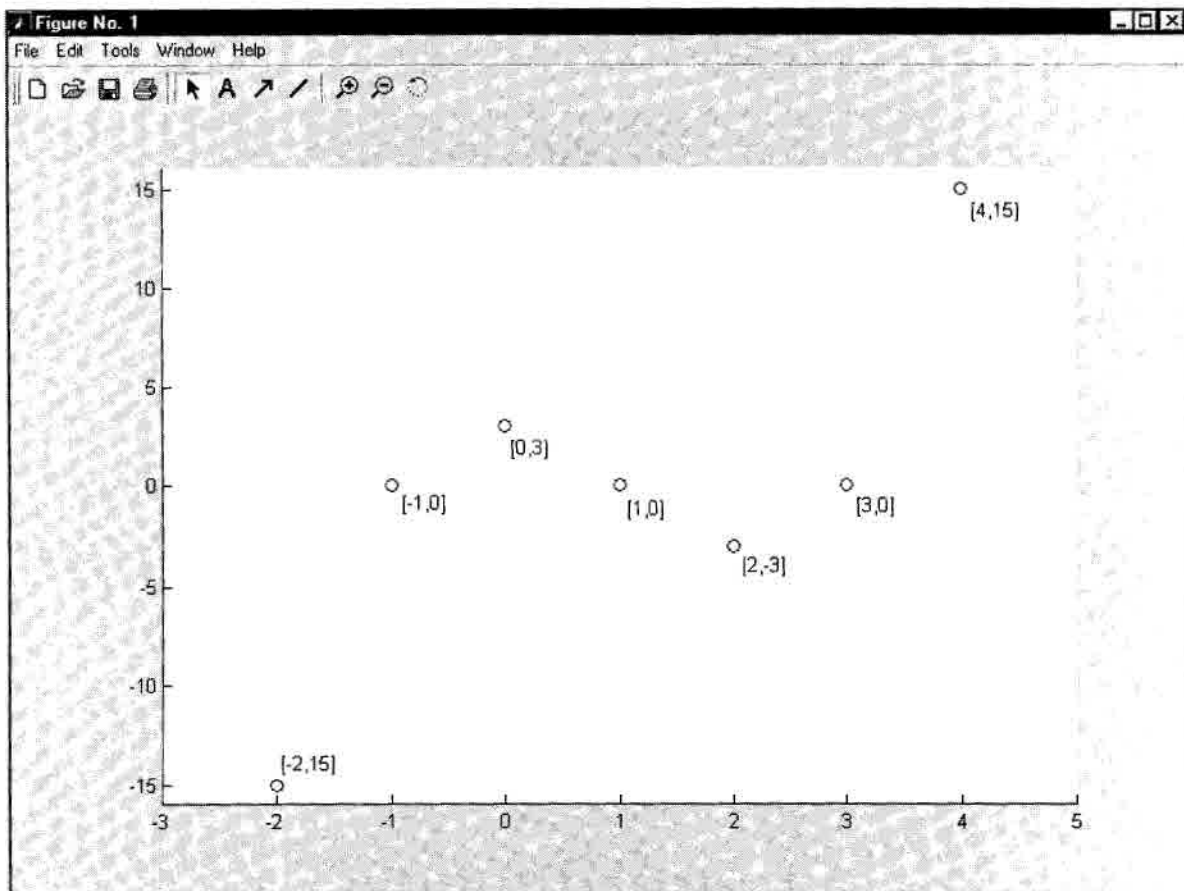
```
end
```

```
solucao = p*soma;
```

## ANEXO D-1

Gráfico da função dada por um conjunto de pares de pontos de  $x_i$  e  $f(x_i)$  a que foram interpolados por forma de Lagrange:

$$x_i = [-2 \ -1 \ 0 \ 1 \ 2 \ 3 \ 4] \text{ e } f(x_i) = [-15 \ 0 \ 3 \ 0 \ -3 \ 0 \ 15].$$

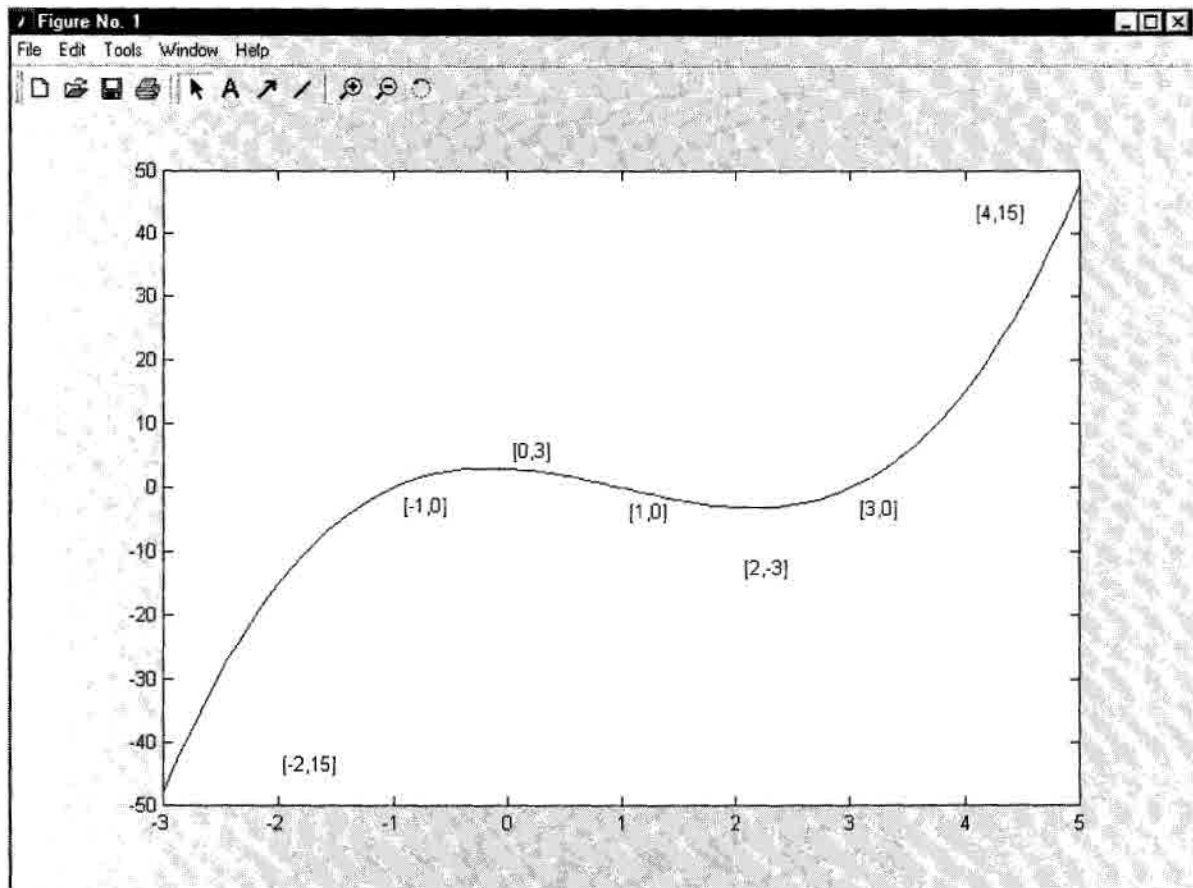




## ANEXO D-2

Gráfico do Polinômio Interpolador do conjunto de pares de pontos:

$x_i = [-2 \ -1 \ 0 \ 1 \ 2 \ 3 \ 4]$  e  $f(x_i) = [-15 \ 0 \ 3 \ 0 \ -3 \ 0 \ 15]$ , pela forma de Lagrange.



## ANEXO E

Programa de Interpolação por forma de Newton em MATLAB.

```
function interp_Newton

x=input('Entre os pontos de interpolação na forma [x0 x1 ... xn]: ');
y=input('Entre os valores da função na forma [y0 y1 ... yn]: ');
xx=input('Entre o valor do ponto de avaliação do polinômio interpolador: ');
res=input('Voce deseja ver o grafico dos pontos da funcao dada? S/N [S] ','s');
if isempty(res)
res='S';

end

n = length(x);

xmin = min(x) - 1;

xmax = max(x) + 1;

ymin = min(y) - 1;

ymax = max(y) + 1;
```

```
if((res=='S') | (res=='s'))  
  
hold;  
  
axis([xmin xmax ymin ymax]);  
  
for i=1:n  
  
plot(x(i),y(i), 'o');  
  
end  
  
disp('Pressione qualquer tecla para continuar')  
  
pause;  
  
end  
  
sol = dif_divididas(x, y);  
  
soma = sol(1, n);  
  
for i=n-1:-1:1  
  
aux = xx - x(i);  
  
soma = (soma * aux) + sol(1,i);
```

```
end

ss=sprintf('O valor do polinômio interpolador no ponto %f é: %f', xx,soma);

disp(' ');

disp(ss);

disp(' ');

res=input('Voce deseja ver o grafico do polinômio interpolador? S/N [S] ','s');

if isempty(res)

res='S';

end

if((res=='S') | (res=='s'))

h = (xmax - xmin)/100;

xxx = xmin : h : xmax;

m = length(xxx);

for i=1:m

yy(i) = pol(n, x, y, xxx(i));

end

plot(xxx,yy);
```

end

```
function solucao = pol(n, x, y, xx)
```

```
p = 1;
```

```
soma = 0;
```

```
for i=1:n
```

```
dd = xx-x(i);
```

```
if (dd == 0)
```

```
solucao = y(i);
```

```
return;
```

end

```
p = p*dd;
```

```
for j=1:i-1
```

```
dd = dd*(x(i)-x(j));
```

end

```
for j=i+1:n
```

```
dd = dd*(x(i)-x(j));
```

end

soma = soma + y(i)/dd;

end

solucao = p\*soma;

function dif = dif\_divididas(x, y)

n = length(x);

dif(:,1) = y';

for j=2:n

for i=1:n-j+1

dif(i,j) = (dif(i+1,j-1)-dif(i,j-1))/(x(i+j-1)-x(i));

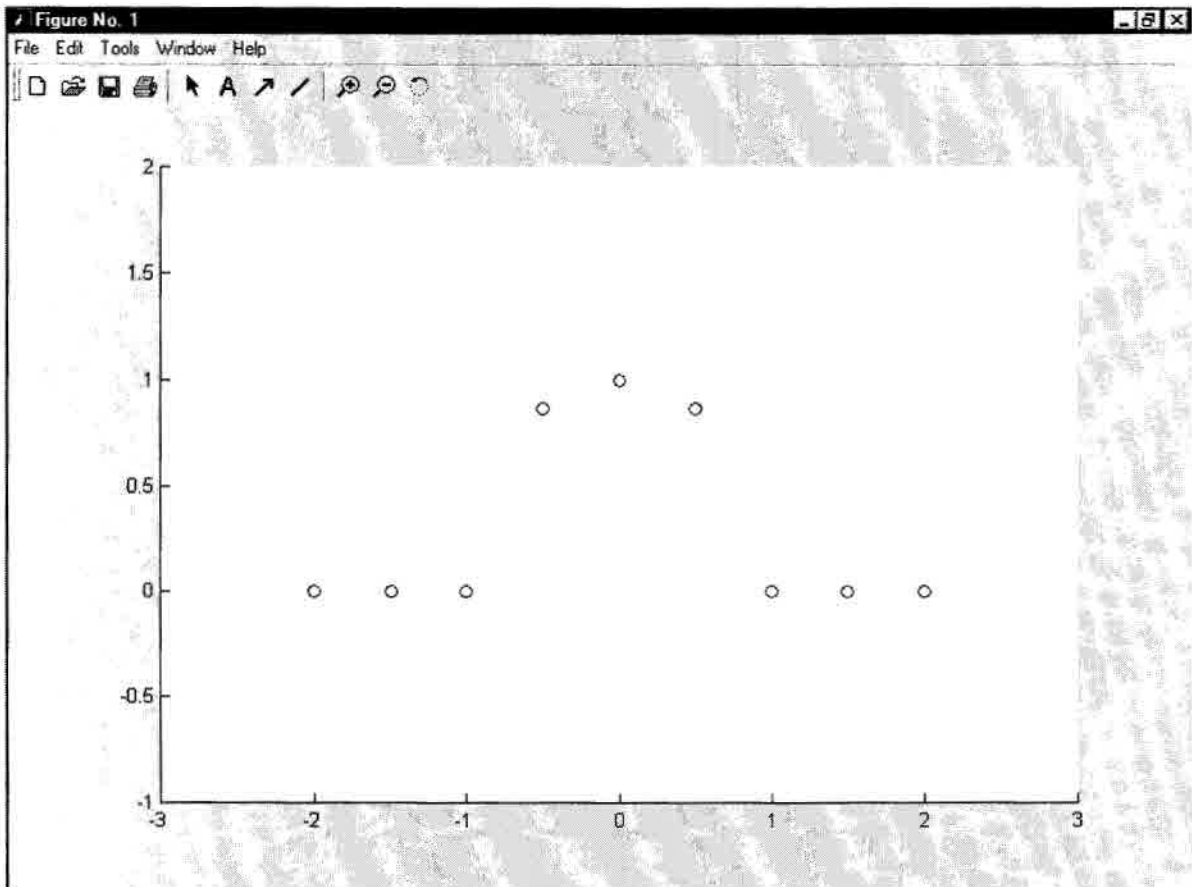
end

end

## ANEXO E-1

Gráfico da função dada por um conjunto de pares de pontos de  $x_i$  e  $f(x_i)$  a que foram interpolados por forma de Newton:

$$x_i = [-2 \ -1.5 \ -1 \ -0.5 \ 0 \ 0.5 \ 1 \ 1.5 \ 2] \text{ e } f(x_i) = [0 \ 0 \ 0 \ 0.87 \ 1 \ 0.87 \ 0 \ 0 \ 0].$$

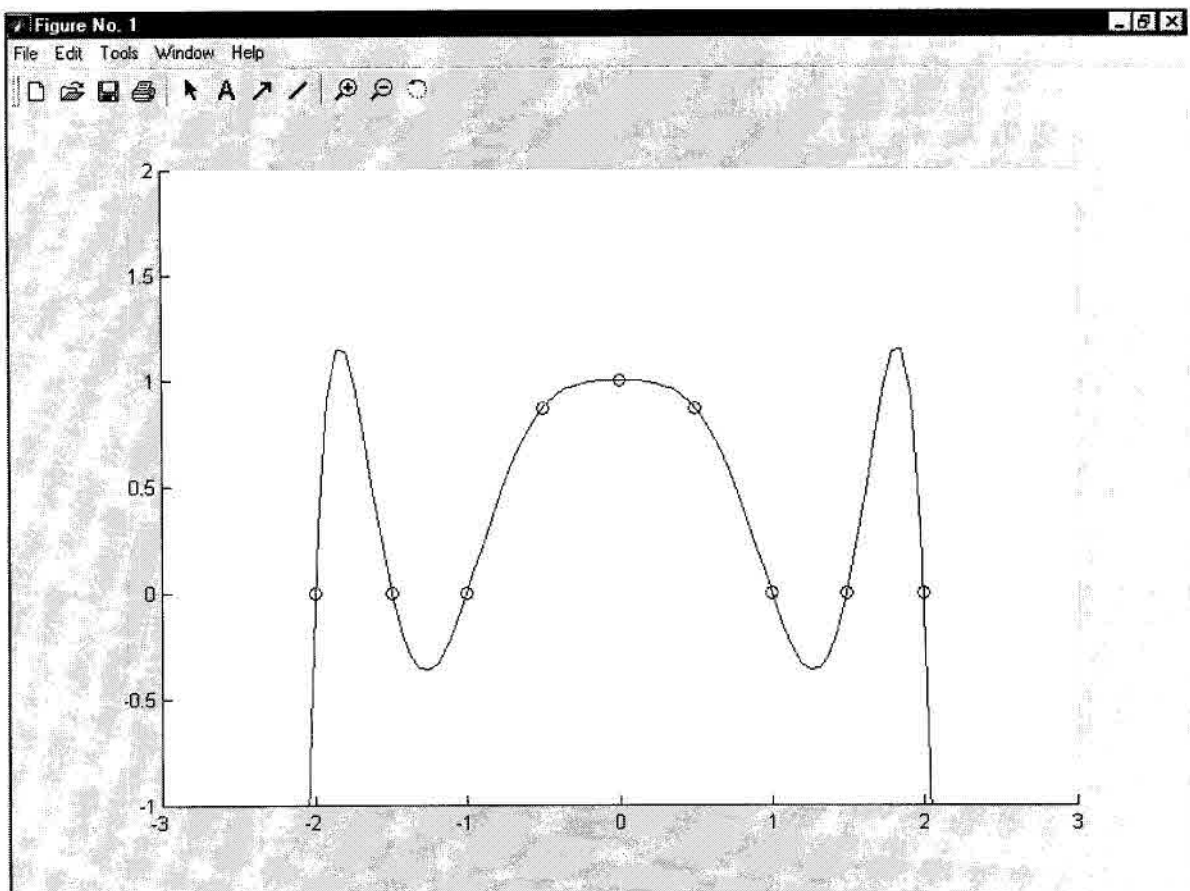


## ANEXO E-2

Gráfico do Polinômio Interpolador do conjunto de pares de pontos:

$$x_i = [-2 \ -1.5 \ -1 \ -0.5 \ 0 \ 0.5 \ 1 \ 1.5 \ 2] \text{ e } f(x_i) = [0 \ 0 \ 0 \ 0.87 \ 1 \ 0.87 \ 0 \ 0 \ 0],$$

pela forma de Newton.





## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] FORTUNA, A. de O. – “Técnicas Computacionais para Dinâmica dos Fluidos– Conceitos Básicos e Aplicações”, 2000;
- [2] Azevedo, N. J., - “Aerodinâmica Básica e Aplicada”, curso ministrado na Universidade Braz Cubas em 2001 e 2002;
- [3] Gaspar, P.D., Barroca, R.F. e Pitarma, R.A. – “Comparação das Simulações Numéricas Obtidas por Códigos Comerciais de Dinâmica de Fluidos Computacional”, Novembro de 2001;
- [4] Ozisik, M. N., - “Transferência de Calor” um texto Básico, 1985;
- [5] PUC-Rio – “Métodos Numéricos para Engenharia” – Cap. IV, 2002;  
Disponível: [http://www.venus.rdc.puc-rio.br/werneckr/mn\\_cap0400.htm](http://www.venus.rdc.puc-rio.br/werneckr/mn_cap0400.htm) (acessado em 17 de setembro de 2002).
- [6] Shimidith, M. – “Mecânica dos Fluidos”, 1999;
- [7] USP – Projeto SEA – “Solução Numérica de Equações Diferenciais Parciais”, 1998;  
Disponível: <http://www.lcad.icmc.sc.usp.br/~siae98/cap11/node3.html> (acessado em 07 de maio de 2003 – 08:50).

