MOTIVAÇÃO PARA O ENSINO DE DISCIPLINAS BÁSICAS NOS CURSOS DE ENGENHARIA

Glaucia Maria Bressan – glauciabressan@uniararas.br
Délson Luiz Módolo – delsonmodolo@uniararas.br
Fundação Hermínio Ometto – UNIARARAS
Núcleo de Engenharia
Rua Maximiliano Baruto, 500
13607-339 – Araras - SP

Resumo: *O crescimento tecnológico do Brasil tem sido bastante divulgado e perceptível nos últimos anos. Neste cenário, é notável e também anunciada a carência de profissionais qualificados, especialmente engenheiros, para atuarem no desenvolvimento de novas tecnologias e nos avanços na produção do país. Uma das maiores dificuldades enfrentadas pelos cursos de Engenharia da atualidade é a grande evasão de alunos que ocorre logo nos anos iniciais do curso, com as disciplinas básicas. Com o objetivo de motivar o processo de ensino-aprendizagem destas disciplinas, o presente trabalho mostra de que forma algumas disciplinas, como Métodos Numéricos, podem ter um enfoque computacional e ser associadas com aplicações reais, baseadas em dados experimentais, para, por exemplo, ajustar curvas e interpolar funções, utilizando implementações no MATLAB. Além disso, este trabalho pretende servir de exemplo e de motivação para alunos e professores de graduação e pós-graduação.*

Palavras-chave: *Ensino de Engenharia, Métodos Numéricos, MATLAB*

1 INTRODUÇÃO

Nos últimos anos, o Brasil vem tendo um visível crescimento de seu parque industrial e isto pode ser facilmente observado pelo aumento da oferta de empregos. Segundo o IBGE (2010), somente no ano de 2010, o emprego industrial cresceu 3,4%, o maior desde 2002. Todos os 14 locais pesquisados contribuíram com resultados positivos para o crescimento do emprego industrial em comparação com dezembro de 2009. A principal contribuição ficou com São Paulo. Na indústria paulista, as atividades de meios de transporte (8,7%) e de máquinas e equipamentos (7,5%) exerceram as maiores contribuições. Observando a mesma fonte consultada do IBGE (2010) nas outras regiões do Brasil (Norte, Nordeste e Sul) o emprego industrial avançou também em grande parte nas áreas de máquinas, aparelhos eletrônicos e de comunicações, ou seja, todos no ramo tecnológico.

No Brasil, há grandes exemplos de desenvolvimentos tecnológicos que resultaram em relevantes contribuições para o país e também o projetaram no mercado internacional. Entre eles, destacam-se a exploração do petróleo em águas profundas, a produção de biocombustíveis (etanol e biodiesel) como fontes renováveis de energia em substituição aos combustíveis fósseis e tendo como conseqüência uma redução na emissão dos gases responsáveis pelo efeito estufa, a indústria aeronáutica com aviões de uso civil e militar que têm obtido grande penetração em países desenvolvidos entre outros.

Embora os exemplos acima citados possam parecer distantes entre si, há pontos em comum. Um deles, é a capacidade da engenharia de transformar o conhecimento das ciências básicas em produtos de alta tecnologia que melhoram a qualidade de vida das pessoas e alavancam o desenvolvimento econômico do país. O outro é a interdisciplinaridade dos projetos envolvendo várias áreas distintas do conhecimento, inclusive entre as engenharias. Esses avanços são frutos das interações entre pesquisadores das grandes universidades e departamentos de Pesquisa & Desenvolvimento das empresas.

Às universidades cabe o papel da formação de recursos humanos com a competência necessária para o desenvolvimento dos projetos tecnológicos demandados pelas empresas. Sendo assim, um dos maiores desafios de um país é a formação de uma grande quantidade de engenheiros munidos de uma sólida qualificação necessários para a ampliação da capacidade tecnológica de um país. Segundo dados da ABENGE (2001), no Brasil, a população de engenheiros é pequena quando comparada a de países do primeiro mundo: cinco engenheiros para cada mil trabalhadores da população economicamente ativa contra 15 a 25 nesses outros países.

Essa pequena participação da engenharia na sociedade é insuficiente para sustentar o processo de desenvolvimento e tornar a economia brasileira mais competitiva. Menos de 10% dos alunos de graduação das universidades brasileiras estão matriculados em cursos de engenharia, contra mais de 25% nos Estados Unidos. Esse quadro mostra que esta profissão foi nitidamente desvalorizada com o tempo, sendo uma conseqüência de baixos investimentos em pesquisa e desenvolvimento e em infra-estrutura no país. Embora os dados possam parecer antigos, o quadro atual não mudou substancialmente. Este cenário resultou em uma falta de engenheiros com formação sólida no mercado, chegando a situações tão críticas que, em alguns casos, fez-se necessário a “importação” de profissionais devido à sua ausência no mercado de trabalho.

O atual cenário sócio-econômico brasileiro e a necessidade de se impulsionar o desenvolvimento científico e tecnológico da nação tornam imperativa a formação de uma grande quantidade de engenheiros capazes de se adaptar a novos ambientes onde o impacto social, econômico e ambiental de sua atuação é cada vez mais imprescindível. No entanto, apesar da necessidade de qualificação e competência dos recém formados, o Brasil convive com uma quantidade muito grande de profissionais que não preenchem as vagas do mercado devido à falta de qualificação para o mercado de trabalho.

Uma das causas é a falta de estrutura das escolas de primeiro e segundo grau e outra é a progressão continuada, que desestimula o estudo dos alunos, pois já sabem que não precisam se esforçar em disciplinas difíceis, uma vez que não há reprovação. Disciplinas das áreas das exatas, como matemática, física e química (fundamentais para os cursos de engenharia), acabam sendo os principais motivos da grande evasão destes cursos, devido à enorme dificuldade dos alunos no entendimento e aprendizado delas. Outro problema atual é que a proliferação de cursos noturnos de 5 anos (no caso das engenharias) demanda grade curricular mais compacta, exigindo um esforço maior do aluno. Muitas vezes, este aluno não tem o tempo necessário para dedicar-se às disciplinas fundamentais, implicando em um desempenho abaixo do esperado e contribuindo para o aumento da evasão.

Uma forma de contribuir para o entendimento e aprendizado dos alunos em disciplinas básicas é a utilização de recursos computacionais que permitam a visualização dos fenômenos. No caso específico das disciplinas da área da matemática, softwares como MATLAB, MATHEMATICA, MAPLE entre outros consistem em poderosas ferramentas auxiliares de ensino.

O presente trabalho tem como principal objetivo mostrar de que forma algumas disciplinas básicas, como Métodos Numéricos, podem ter um enfoque computacional e ser associadas com aplicações reais, baseadas em dados experimentais, para, por exemplo, ajustar curvas e interpolar funções, utilizando implementações no MATLAB. Também procura ser um exemplo e uma motivação para alunos e professores de graduação e pós-graduação.

2 MODELOS MATEMÁTICOS COM ENFOQUE COMPUTACIONAL

A capacidade de visualização dos dados é um fator importante na solução de problemas de engenharia. Às vezes, o dado é um número real; outras, um grupo de coordenadas x-y-z que representam os quatro vértices de uma pirâmide com uma base triangular no espaço. Estes exemplos podem ser representados usando um tipo especial de estrutura de dados: matriz. Desta forma, um número pode ser considerado uma matriz com uma linha e uma coluna, uma coordenada x-y pode ser considerada uma matriz com uma linha e duas colunas, e um grupo de quatro coordenadas x-y-z, como uma matriz com quatro linhas e três colunas e assim por diante.

O MATLAB é uma linguagem de computação de nível alto usada para cálculos científicos e visualização de dados, construída em um ambiente de programação interativo. A grande vantagem de um sistema interativo é que os programas podem ser testados e executados rapidamente, podendo ser desenvolvidos em um tempo muito menor do que linguagens como Fortran ou C. O ponto negativo é que os programas só podem ser executados em computadores que tenham o MATLAB instalado. Os arquivos de comando e programas têm extensão *.m* (M-files) e os arquivos de dados binários *default* tem extensão *.mat* (MAT-files). Trata-se de um programa-base sucessivamente aumentado pela incorporação de toolboxes específicos (Otimização, Processamento de Sinais, Estatística, entre outros). Versões mais recentes dispõem de interface bastante amigável. O MATLAB pode ser muito bem aplicado na sala de aula para as disciplinas básicas dos cursos de Engenharia, mostrando enfoque computacional e aplicações reais que motivam o aprendizado dos alunos. As implementações deste trabalho são baseadas em KIUSALAAS, 2005.

2.1 Funções Matemáticas Implementadas em MATLAB

Um dos problemas que ocorrem mais frequentemente em trabalhos científicos é calcular as raízes de equações da forma: $f(x) = 0$, onde f(x) pode ser um polinômio em x ou uma função transcendente. Em raros casos é possível obter as raízes exatas de $f(x) = 0$, como ocorre por exemplo, supondo-se $f(x)$ um polinômio fatorável. Através de técnicas numéricas, é possível obter uma solução aproximada, em alguns casos, tão próxima da solução exata, quanto se deseje. A maioria dos procedimentos numéricos fornece uma sequência de aproximações, cada uma das quais mais precisa que a anterior, de tal modo que a repetição do procedimento fornece uma aproximação a qual difere do valor verdadeiro por alguma tolerância pré-fixada. Um dos métodos mais conhecidos para a resolução numérica de equações é o método de *Newton-Raphson* (ARENALES & DAREZZO, 2008; CHAPRA & CANALE, 2008). O processo iterativo é definido por:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

que converge sempre que $|x_0 - \bar{x}|$ for suficientemente pequeno.

Uma interpretação geométrica do método de Newton é dada na Figura 1. Dado x_k , o valor x_{k+1} pode ser obtido graficamente traçando-se pelo ponto (x_k , $f(x_k)$) a tangente à curva $y = f(x)$. O ponto de interseção da tangente com o eixo x determina x_{k+1}.

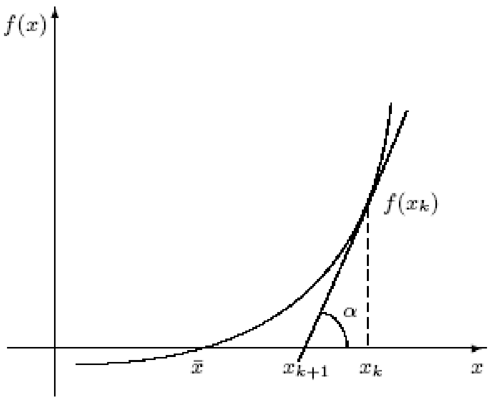

Figura 1 – Interpretação geométrica do método de Newton Raphson

No MATLAB, pode-se implementar uma função que execute o procedimento do método de Newton – Raphson. Os parâmetros são os seguintes:

tol*x*, tol*y* - tolerâncias para o critério de parada: ($|x(k)-x(k-1)| <$ tol*x*) ou ($|f(x(k))| <$ tol*y*)
maxit - número máximo de iterações

Parâmetros de saída:
x - vetor das aproximações da raiz da função
y - vetor dos valores de *f (x)*

A função, chamada 'newtonraph', pode ser implementada no MATLAB da seguinte forma.

```
function [x,y]=newtonraph(fun,x0,tolx,toly,maxit)
x(1)=x0;
[y(1),dy(1)]=feval(fun,x(1));
iter=maxit+1;
 for k=2:iter
    x(k)=x(k-1)-y(k-1)/dy(k-1);
   [y(k),dy(k)]=feval(fun,x(k));
   difx(k)=abs(x(k)-x(k-1));
      if ((difx(k)<tolx) | (abs(y(k))<toly))
      disp('O método de Newton-Raphson converge.')
      break;
   end
 end
 if ((difx(k)<tolx) & (abs(y(k))<toly))
   disp('A raiz foi encontrada com as duas tolerâncias desejadas.')
elseif (difx(k)<tolx)
   disp('A raiz foi encontrada com |x(k)-x(k-1)| < tolerância.')
elseif (abs(y(k))<toly)
   disp('A raiz foi encontrada com |f(xk)| < tolerância.')
else
   disp('*******************************************************')
   disp('A raiz não foi encontrada com as tolerâncias desejadas.')
   disp('*******************************************************')
end
ym=abs(y); n=length(x);
k=0:n-1;
out=[k' x' difx' y' ym'];
disp('----------------------------------------------------')
disp('  iteração     xk    |x(k)-x(k-1)|     yk     |f(xk)|')
disp('----------------------------------------------------')
disp(out)
disp('----------------------------------------------------')
```

Exemplo de execução

Na tela de execução do MATLAB pode-se compilar a função com os seguintes parâmetros

```
>> r=newtonraph('f2',0.8,0,0.0005,10)
```

O parâmetro f2 é o nome da função do MATLAB como string. Deve conter o valor da função, *y=f(x)* e da sua primeira derivada *dy=f'(x)*. O arquivo f2.m deve ser, por exemplo, da seguinte forma, utilizando a função $\cos(x^3) - \log(x)$.

```
function [y, y1] = f2(x0)
  syms x;
  f = cos(x^3)-log(x);
  f1 = diff(f);
  y = subs(f,x0);
  y1 = subs(f1, x0);
```

A resposta fornecida é

```
O método de Newton-Raphson converge.
A raiz foi encontrada com |f(xk)| < tolerância.
----------------------------------------------------
  iteração     xk    |x(k)-x(k-1)|    yk      |f(xk)|
----------------------------------------------------
         0    0.8000         0     1.0949    1.0949
    1.0000    1.2998    0.4998    -0.8475    0.8475
    2.0000    1.1261    0.1737     0.0235    0.0235
    3.0000    1.1312    0.0051    -0.0001    0.0001
----------------------------------------------------

r =     0.8000    1.2998    1.1261    1.1312
```

Portanto, a aproximação para a raiz é 1,1312.

2.2 Método dos Mínimos Quadrados Utilizando MATLAB

Dado um conjunto de pontos X e Y, obtidos experimentalmente, o método dos mínimos quadrados pode ser aplicado para ajustar uma função (curva) a estes pontos, aumentando o grau da função polinomial ajustada. A princípio, uma reta pode ser ajustada. Os comandos a seguir são executados na própria tela de execução do MATLAB ou também podem ser implementados em forma de função programa com extensão .m.

Para tanto, é considerado o conjunto de pontos X e Y coletados experimentalmente no trabalho de SOUZA (2010). Os pontos X se referem às rotações por minuto (RPM) de um motor de combustão interna e os pontos Y representam a pressão média efetiva indicada líquida (bar), de um coletor de comprimento primário de 300mm.

X=[1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000, 5500, 6000]
Y=[10,7, 11,5, 12,1, 13,3, 14,0, 13,2, 12,2, 11,0, 10,1, 8,5, 7,4]

São executados os comando polyfit(X,Y, grau do polinômio) para o ajuste polinomial e o comando polyval() para o cálculo do erro.

Ajuste de uma reta

Primeiramente, os dados devem ser inseridos no MATLAB como vetores X e Y. Em seguida, são aplicadas as funções polyfit() e polyval() e, finalmente, são executados os comandos necessários para a exibição gráfica do ajuste.

```
>> X=[1000 1500 2000 2500 3000 3500 4000 4500 5000 5500 6000]';
>> Y=[10.7 11.5 12.1 13.3 14.0 13.2 12.2 11.0 10.1 8.5 7.4]';
>> Coef=polyfit(X,Y,1)        % constante + Coef1 xj
Coef =    -0.0007   13.8755   %são os coeficientes da reta
>> y1=polyval(Coef,X);        % cálculo do erro (resíduo do ajuste)
>> res=Y-y1;
>> residuoMMQ=res'*res
residuoMMQ =  26.1145   % resíduo do método pelo ajuste da reta
% para plotar o gráfico do ajuste, executamos:
>> [sx k]= sort(X);      % ordenado em sx, índices originais em k
>> sy1=y1(k);            % reordena y para ser coerente com x original
>> sy=Y(k);
>> plot(sx,sy,'b*',sx,y1(k),'r')
```

A saída é o ajuste da reta aos pontos, conforme Figura 2a.

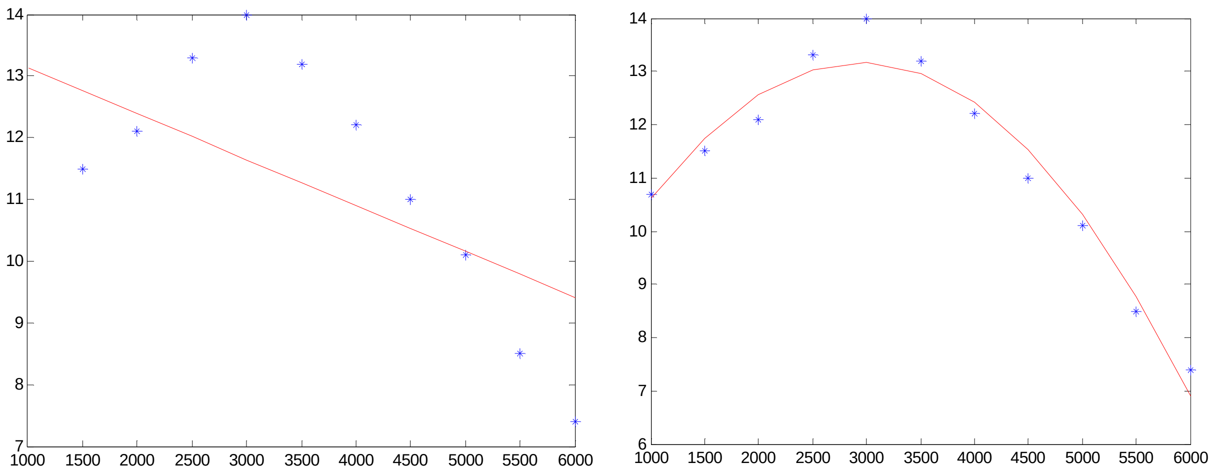

Figura 2. a) Ajuste da reta aos pontos; b) Ajuste de um polinômio de grau 2 aos pontos.

Portanto, a reta que melhor se ajusta aos dados, segundo o método dos mínimos quadrados tem equação y = -0,0007 + 13,8755x e o resíduo do ajuste é 26,1145.

Ajuste de um Polinômio de grau 2

Utilizando os comandos polyfit() e polyval() após a inserção dos dados X e Y, o mesmo algoritmo pode ser empregado para o ajuste de um polinômio de grau 2 aos pontos coletados.

```
>> Coef=polyfit(X,Y,2)              % constante + Coef1 xj + Coef2 xj^2
Coef = -0.0000  0.0040  7.3164      % coeficientes do polinômio de grau 2
>> y1=polyval(Coef,X);              % cálculo do resíduo do ajuste
>> res=Y-y1;
>> residuoMMQ=res'*res
residuoMMQ =  1.8458                % valor do resíduo
>> [sx k]= sort(X);
>> sy1=y1(k); sy=Y(k);
>> plot(sx,sy,'b*',sx,y1(k),'r')
```

A saída é o ajuste da reta aos pontos, conforme Figura 2b.

Portanto, o polinômio de grau 2 que melhor se ajusta aos dados, segundo o método dos mínimos quadrados, tem equação $y = 0{,}0040x + 7{,}3164x^2$ e o resíduo do ajuste é 1,8458.

Executando esta sequência de comandos no Matlab, é possível aumentar o grau do polinômio a ser ajustado e verificar o resíduo do ajuste da curva, até obter um resultado satisfatório.

Ajuste de um Polinômio de grau 5

O mesmo procedimento é aplicado para aumentar o grau do polinômio para a melhora do ajuste da curva e a conseqüente diminuição do resíduo.

```
>>Coef=polyfit(X,Y,5) %const+Coef1xj+Coef2xj^2+Coef3xj^3+Coef4xj^4+Coef5 xj^5
Coef =  -0.0000    0.0000   -0.0000    0.0000   -0.0182   18.4136
>> y1=polyval(Coef,X);
>> res=Y-y1;
>> residuoMMQ=res'*res
residuoMMQ =   0.4004
>> [sx k]= sort(X);
>> sy1=y1(k); sy=Y(k);
>> plot(sx,sy,'b*',sx,y1(k),'r')
```

A saída é o ajuste da reta aos pontos, conforme Figura 3a.

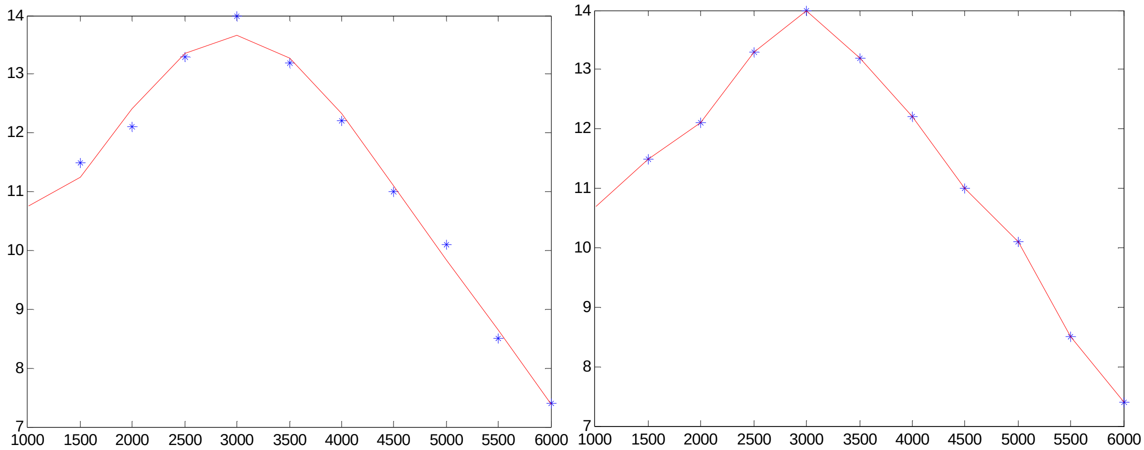

Figura 3. a) Ajuste de um polinômio de grau 5 aos pontos;
b) Ajuste de um polinômio de grau 10 aos pontos.

Portanto, o polinômio de grau 5 que melhor se ajusta aos dados, segundo o método dos mínimos quadrados tem equação $y = -0{,}0182\, x^4 + 18{,}4136\, x^5$ e o resíduo do ajuste é 0,4004.

Ajuste de um Polinômio de grau 10

Finalmente, o ajuste é feito até o décimo grau do polinômio, já que são 11 pontos coletados.

```
>> Coef=polyfit(x,y, 10)
Coef =
  1.0e+003 *
0.0000  -0.0000   0.0000   -0.0000   0.0000   -0.0000   0.0000  -0.0000
0.0000  -0.0055   1.4077
>> y1=polyval(Coef,X);
>> res=Y-y1;
>> residuoMMQ=res'*res
```

```
residuoMMQ =  3.2864e-017
>> [sx k]= sort(X);
>> sy1=y1(k);
>> sy=Y(k);
>> plot(sx,sy,'b*',sx,y1(k),'r')
```

A saída é o ajuste da reta aos pontos, conforme Figura 3b.

Desta forma, pode-se aumentar o grau do polinômio até o grau n, conhecidos n+1 pontos. Quanto maior o grau do polinômio, menor será o erro residual. Portanto, o polinômio de grau 10 corresponde à curva que melhor se ajusta aos 11 dados experimentais, e tem equação y=-5,50x^9 + 1407,7x^{10}. O resíduo do ajuste, menor de todos os anteriores, é $3{,}2864\times10^{-17}$.

2.3 Interpolação Polinomial – Forma de Lagrange no MATLAB

O problema geral da interpolação por meio de polinômios consiste em, dados n+1 pontos distintos $x_0, x_1, ..., x_n$ e n+1 valores de uma função $y=f(x)$, dados por $y_0, y_1, ..., y_n$, determina-se um polinômio $P_n(x)$ de grau máximo n tal que

$$P_n(x_0)=y_0\,;\ P_n(x_1)=y_1\,;\ \ldots;\ P_n(x_n)=y_n.$$

Sejam $x_0, x_1, ..., x_n$, n+1 pontos distintos. Considere para k = 0, 1,..., n, os seguintes polinômios de grau n:

$$\ell_k(x) = \frac{(x-x_0)\ldots(x-x_{k-1})(x-x_{k+1})\ldots(x-x_n)}{(x_k-x_0)\ldots(x_k-x_{k-1})(x_k-x_{k+1})\ldots(x_k-x_n)}.$$

Para valores dados: $f_0=f(x_0)$; $f_1=f(x_1)$; …; $f_n=f(x_n)$ de uma função $y=f(x)$, o polinômio:

$$P_n(x) = \sum_{k=0}^{n} f_k\ \ell_k(x)\ ,$$

é o polinômio de interpolação de f(x) nos pontos $x_0, x_1, ..., x_n$ (ARENALES & DAREZZO, 2008; CHAPRA & CANALE, 2008).

No MATLAB, duas funções podem ser implementadas separadamente. A primeira calcula os coeficientes do polinômio de Lagrange e a segunda calcula os valores do polinômio interpolador de Lagrange para pontos desejados.

Função `“lagrangecoef”` – Calcula os coeficientes dos polinômios de Lagrange. Os vetores x e y correspondem aos dados de entrada e o vetor c, que contém os coeficientes dos polinômios de Lagrange.

```
function c = lagrangecoef(x,y)
n=length(x);
for k = 1 : n
   d(k) = 1;
   for i = 1 : n
      if i ~= k
         d(k)=d(k)*(x(k) - x(i));
      end
      c(k) = y(k)/d(k);
   end
end
```

Função "lagrange" – Calcula os valores do polinômio interpolador de Lagrange para pontos desejados, cujos valores pertencem ao intervalo de valores experimentados. O vetor *t* corresponde à entrada da função e o vetor *p* contém os valores do polinômio Pn(x) calculados nas abcissas de *t*.

```
function p = lagrange(t,x,y)
c = lagrangecoef(x,y);
m = length(x);
for i = 1 : length(t)
   p(i) = 0;
   for j = 1 : m
      N(j) = 1;
      for k = 1 : m
         if j ~= k
            N(j) = N(j) * (t(i) - x(k));
         end
      end
      p(i) = p(i) + N(j) * c(j);
   end
end
```

Exemplo de execução

O conjunto de pontos X e Y coletados experimentalmente no trabalho de SOUZA (2010) são considerados. Os pontos X se referem às rotações por minuto (RPM) de um motor de combustão interna e os pontos Y representam a eficiência volumétrica, expressa em porcentagem, de um coletor de comprimento primário de 300mm.

X = [1000, 1500, 2000, 2500]
Y=[86,68, 88,92, 90,41, 94,74]

As funções são executadas no MATLAB da seguinte forma. Inserimos os dados X e Y em forma de vetores:

```
>> X = [1000, 1500, 2000, 2500]';
>> Y=[86.68, 88.92, 90.41, 94.74]';
```

Primeiramente, função "lagrangecoef" é executada para que os coeficientes sejam obtidos:

```
>> lagrangecoef(X,Y)
```

Os coeficientes do polinômio de Lagrange são $10^{-6}(-0,1156+0,3557x–0,3616x^2+0,1263x^3)$
Em seguida, a função "lagrange" é executada para obter o valor da função no ponto X=1600

```
>> lagrange(1600,X,Y)
```

A resposta é 89.1631, ou seja, 89,1631%. O valor da função em vários pontos do intervalo de interpolação pode ser obtido, por exemplo, nos pontos 1400, 1600, 2100. Para isso, a função 'lagrange' é executada da seguinte forma

```
>> lagrange([1400,1600,2100],X,Y)
```

E obtém-se, respectivamente, os valores 88,6469%, 89,1631% e 90,9339%.

3 CONSIDERAÇÕES FINAIS

Este trabalho mostrou que ferramentas computacionais transformaram-se, nos dias de hoje, em ótimos aliados dos docentes para auxiliar aplicações da matemática de forma rápida, compacta e adequada à carga horária dos cursos atuais. Além de todas estas vantagens, ainda facilita o trabalho de entendimento por parte dos alunos, geralmente carentes de uma formação básica mais apurada, servindo também de estímulo ao aprendizado e contribuindo de forma efetiva no combate à evasão.

REFERÊNCIAS BIBLIOGRÁFICAS

ARENALES, Selma e DAREZZO, Artur. **Cálculo Numérico – aprendizagem com apoio de software.** 1ª ed, São Paulo: Thomson, 2008. 364p.

ASSOCIAÇÃO BRASILEIRA DE ENSINO DE ENGENHARIA (ABENGE). **Programa de apoio ao ensino e a pesquisa da engenharia –** PAEPE. Brasília/DF, abril de 2001.

CHAPRA, Steven C. e CANALE, Raymond P. **Métodos Numéricos para Engenharia.** 5ª ed. McGraw Hill, 2008. 809p.

KIUSALAAS, Jaan. **Numerical Methods in Engineering with MATLAB.** Cambridge University Press, 2005. 435 p.

Pesquisa Industrial Mensal de Emprego e Salário - Fonte IBGE Base: Ano de 2010

SOUZA, Gustavo Rodrigues de. UNIVERSIDADE DE SÃO PAULO, Escola de Engenharia de São Carlos. **Estudo experimental e numérico do sistema de admissão de um motor de combustão interna,** 2010. 141p, il. Tese (Doutorado).

MOTIVATION FOR THE TEACHING OF BASIC DISCIPLINES IN ENGINEERING COURSES

Abstract: *Brazil's technological growth has been shown and noticed in the last years. In this context, what is also noticeable and which has been advertised is the deficit of qualified professionals, specially engineers, to work on the development of new technologies and to make the country's production increase. One of the most serious difficulties faced by engineering courses nowadays is the high student dropout rates that occur in the first years of the course, in basic disciplines. In order to motivate the teaching-learning process of these disciplines, this work shows how some disciplines, such as Numerical Methods, can present a computational focus and be associated to real applications based on empirical data in order to make curves fitting and to interpolate functions, for example, using implementations in MATLAB. Moreover, this work intends to be an example and a source of motivation for professors and undergraduate and postgraduate students.*

Key-words: *Engineering Teaching, Numerical Methods, MATLAB*