



COBENGE 2005

XXXIII - Congresso Brasileiro de Ensino de Engenharia

“Promovendo e valorizando a engenharia em um cenário de constantes mudanças”

12 a 15 de setembro - Campina Grande - Pb

Promoção/Organização: ABENGE/UFPE

UTILIZAÇÃO DO MAPLE, MATLAB E SCILAB NOS CURSOS DE ENGENHARIA

Viviana Cocco Mariani – viviana.mariani@pucpr.br
Programa de Pós-Graduação em Engenharia Mecânica – PPGEM
Pontifícia Universidade Católica do Paraná - PUCPR
Rua Imaculada Conceição, 1155, Prado Velho
81611-970 - Curitiba, PR, Brasil

Tânia Martins Preto - tania@inf.ufpr.br
André Luiz Pires Guedes - andre@inf.ufpr.br

***Resumo:** O presente trabalho tem como objetivo apresentar as principais características dos ambientes computacionais Maple, Matlab e Scilab aplicados no ensino de Cálculo Numérico, disciplina presente em todos os cursos de Engenharia. Estes ambientes computacionais são utilizados com o intuito de estimular e facilitar o aprendizado tornando mais atrativo o estudo e entendimento dos principais métodos de Cálculo Numérico. Em todos estes ambientes o aluno tem a opção de entrar com os dados e obter as respostas prontas, além de utilizar estes ambientes para criar os seus próprios códigos e analisar suas respostas. Assim, ocorre um melhor aproveitamento por parte dos alunos, capacitando-os a interpretar os fenômenos físicos envolvidos nos problemas propostos.*

***Palavras-chave:** Cálculo Numérico, Matlab, Scilab, Engenharia, Programação*

1. INTRODUÇÃO

Hoje vivemos o desenvolvimento tecnológico em todas as áreas o que causa uma revolução na criação e exploração de novas metodologias de ensino. O aluno que chega a Universidade é um aluno crítico e ativo, ele quer saber porque e onde irá utilizar os conteúdos que estão sendo ministrados, a pergunta mais ouvida em sala de aula é: “Para que serve isto?”. Algumas vezes esta questão não tem resposta, até porque nem todo conteúdo tem uma aplicação imediata, alguns conteúdos são básicos para a resolução de problemas mais avançados que serão apresentados no decorrer do curso.

O professor deve ver o aluno como uma pessoa engajada em uma sociedade que convive diariamente, mesmo que não diretamente, com a tecnologia. Esta se popularizou e é utilizada por muitos, contudo poucos conhecem os seus mecanismos de funcionamento. O papel do professor é estar em constante atualização para usar exemplos aplicados e instigantes para os alunos, uma vez que estão sendo formados novos Cientistas da Computação, Engenheiros.

Geralmente os métodos de ensino não têm se beneficiado de forma significativa de todas as opções de programas específicos e ambientes de programação. Sob este aspecto, programas como o Maple, Matlab e Scilab têm grande potencial para melhorar a relação ensino-aprendizado dos cursos de Engenharia e Computação. Estes programas oferecem diversas funcionalidades para a resolução de problemas matemáticos tais como cálculos matriciais, resolução de integrais, derivadas, manipulação de vetores em geometria, análise numérica, além de permitir a visualização de gráficos de funções. Os mesmos são utilizados para resolver diversos problemas de Engenharia, Matemática, possibilitando a solução numérica dos problemas como a maioria dos programas convencionais. Além disso, representam boas opções de ferramentas para o ensino de matemática, possuindo a capacidade de permitir a formulação dos mesmos a partir dos princípios fundamentais e a solução algébrica desses mesmos problemas em um ambiente computacional integrado e interativo.

Outra grande vantagem desses ambientes é que possuem uma linguagem de programação de alto nível, exigindo que o usuário tenha apenas conhecimentos básicos de programação, não sendo necessário usar técnicas avançadas de programação e estruturas de dados. Em especial o Scilab é um programa livre oferecendo vantagens ao usuário quanto a possibilidades de uso.

O uso dos citados programas no ensino, deve ser feito com cautela, pois o aluno até pode exercitar e expandir a concepção intelectual dos fundamentos embutidos nos modelos a serem estudados e estimular a sua criatividade no desenvolvimento de modelos mais apropriados para a situação física, porém deve preocupar-se também com o tratamento matemático, algébrico e as soluções das equações geradas. Os programas são manipulados pelos alunos e não estão isentos de erros, logo os alunos devem interpretar a solução fornecida pelos mesmos, tendo conhecimento dos métodos utilizados e da solução esperada.

O ensino de Cálculo Numérico deve abranger cinco etapas fundamentais (Lima, 1995):

- 1) *Conceituação*: Ato de formular corretamente e objetivamente as definições, enunciados e proposições de Cálculo Numérico, mostrando as conexões entre os diversos conceitos e métodos e desenvolvendo o raciocínio dedutivo.
- 2) *Manipulação*: Prática principalmente de exercícios de caráter algébrico e numérico. Permitindo adquirir habilidade e destreza no manuseio de equações, fórmulas, construções geométricas, algoritmos, preparando o aluno para a aquisição de novos conceitos.
- 3) *Aplicações*: Utilização das noções e teorias aprendidas para obter resultados, conclusões e previsões em problemas que podem variar desde os mais simples até aplicações em outras áreas, sejam elas científicas, tecnológicas ou mesmo sociais. A conceituação e manipulação são indispensáveis para o entendimento das aplicações. As aplicações incluem a resolução de problemas que, pelo fato de representar desafios, auxiliam no desenvolvimento da criatividade, estimulam a imaginação e recompensam o esforço de aprender. As aplicações representam a parte mais interessante do processo, pois possibilita que o aluno entenda a origem de tantos conceitos e a sua importância na resolução de problemas matemáticos e de outras áreas.
- 4) *Programação*: Elaboração de programas computacionais envolvendo os métodos de Cálculo Numérico os quais devem ter como objetivo principal contemplar a aquisição do conhecimento tanto na matemática como na programação. A aprendizagem de conceitos de programação e algoritmos envolve a aquisição de alguns conhecimentos e habilidades específicas, que serão citadas no decorrer deste trabalho.
- 5) *Avaliação*: Ato ou efeito de apreciação e análise dos resultados obtidos pelo aluno no decorrer do programa de Cálculo Numérico, visando julgá-lo proferindo um conceito ou nota.

A tarefa de ensinar é uma tarefa profissional que, no entanto, exige amor, criatividade, competência científica. Programar e avaliar não são momentos separados, um a espera do outro, são momentos interligados. Avaliar implica, quase sempre, em re-programar, retificar. A avaliação, por isso mesmo, não se dá no momento que nos parece ser o final de certa prática (Freire, 2003). A avaliação é um processo contínuo que envolve muitas variantes, tanto relativas aos conteúdos avaliados quanto às concepções do professor sobre os fins da avaliação e também quanto às diversas habilidades que os alunos disponibilizam em uma resolução de questão.

O uso do computador como ferramenta de apoio ao ensino é tema de muitos trabalhos e tem contribuído para a melhoria na qualidade do ensino. Diversos ambientes computacionais foram desenvolvidos nos últimos anos, sendo que uma boa parte destes, concentra-se na tarefa de apoiar as etapas de conceituação e manipulação, porém também existem diversos ambientes computacionais que atendem a etapa de aplicação.

Em geral os ambientes onde as etapas de conceituação e manipulação são o maior foco não exigem que o aluno execute tarefas relacionadas com a elaboração de algoritmos, já em algumas situações na etapa de aplicações propõem-se ao aluno situações onde ele é responsável por elaborar uma solução através de alguma ferramenta computacional. Os programas computacionais abordados neste trabalho podem ser utilizados para auxiliar o ensino de matemática apoiando a formulação e execução de programas.

As aplicações envolvendo programação e matemática devem objetivar a aquisição do conhecimento tanto na matemática como na programação. A aprendizagem de conceitos de programação e algoritmos envolve a aquisição de alguns conhecimentos e habilidades específicas, dentre eles pode-se citar o domínio dos aspectos relacionados com sintaxe (regras de escrita), semântica (significado de comandos), estratégias para decomposição (dividir um problema grande em subproblemas menores), estratégias para composição (união e comunicação entre diferentes partes de um programa), conhecimento pragmático (usar recursos básicos do sistema), escolha da metodologia de desenvolvimento, capacidade de adaptar soluções conhecidas para a resolução de novos problemas e capacidade de entender o que a memória do computador faz durante a execução de um programa (Direne e Pimentel, 1998).

Em alguns cursos como Engenharia Mecânica, Engenharia da Computação e Ciência da Computação, é desejável que os alunos pratiquem as técnicas de programação, já em outros cursos como Engenharia Química, Engenharia de Alimentos etc, os alunos podem utilizar os ambientes computacionais Matlab, Maple ou Scilab sem a necessidade de programar em linguagens que exigem conhecimentos avançados de técnicas de programação, como C, Fortran, Pascal, dentre outras.

O presente artigo apresenta uma descrição breve dos ambientes de programação que são utilizados para o ensino de Cálculo Numérico em alguns cursos de Engenharia, dando ênfase ao Scilab por estar na categoria de software livre, fazendo um comparativo entre exemplos de rotinas feitas nos três programas, finalizando com as conclusões e as perspectivas de trabalhos futuros.

2. CARACTERÍSTICAS DOS AMBIENTES COMPUTACIONAIS

2.1 Matlab

A primeira versão do Matlab foi escrita na Universidade do Novo México e na Universidade de Stanford, no final da década de 70, e foi destinado a cursos de teoria matricial, álgebra linear e análise numérica. Hoje sua capacidade foi estendida, e o Matlab é muito utilizado no desenvolvimento de projetos de Engenharia, devido a sua poderosa capacidade de processamento matemático e visualização gráfica. Matlab apresenta diversos conjuntos de funções voltadas para aplicações específicas, os chamados *toolboxes*.

O Matlab pode funcionar tanto como uma simples calculadora como um ambiente para desenvolvimento de aplicativos através de uma linguagem de programação, proporcionando o desenvolvimento de projetos de diversos tipos de complexidade. Sua utilização faz com que o aluno se prenda mais ao entendimento físico dos problemas, pois não exige conhecimentos profundos de programação e estruturas de dados. O uso do Matlab nos Cursos de Engenharia é interessante porque coloca o aluno em contato com diversas funcionalidades, não envolvendo a utilização direta de técnicas avançadas de programação e estruturas de dados, assim a atenção do aluno fica concentrada na resolução do problema e na análise dos resultados obtidos (Martim e Mariani, 2004).

2.2 Maple

O Maple é um programa versátil onde, pode-se explorar computação algébrica e numérica, gráficos, animações, possuindo uma linguagem de programação. Devido a esta versatilidade, o Maple tem encontrado grande aplicação, tanto do ponto de vista técnico como do ponto de vista educacional. O aplicativo Maple foi desenvolvido pela Universidade de Waterloo, Canadá, e pelo Instituto ETH, de Zurich, Suíça.

A capacidade principal do Maple está nos algoritmos para resolução de problemas simbólicos. Ao contrário dos programas matemáticos convencionais, que podem trabalhar somente com números em ponto flutuante, compreendendo e operando, por exemplo, com frações, raízes quadradas de números não perfeitos, valores inexatos de senos, co-senos, logaritmos e outros.

2.3 Scilab

O Scilab é um programa livre disponível em www.scilab.org. O uso do programa livre na educação é uma alternativa imprescindível a qualquer projeto educacional, tanto no setor público como no privado. Fatores tais como liberdade, custo, flexibilidade são estratégicos para a condução bem sucedida de projetos educacionais mediados por computador. Para o setor educacional, muitas vezes carente de recursos, o programa livre é uma alternativa viável e que deve ser considerada seriamente (Almeida, 2002).

Desenvolvido no INRIA, na França, o Scilab tem sido utilizado para aplicações em sistemas de controle e processamento de sinais. É um código fonte livre feito de três partes distintas: um interpretador, livrarias de funções (procedimentos Scilab) e livrarias de rotinas em Fortran e C. Estas rotinas não pertencem ao Scilab mas são interativamente chamadas pelo interpretador, algumas foram modificadas para melhor compatibilidade com o interpretador do Scilab.

O Scilab é um ambiente de programação numérica bastante flexível, cujas principais características são (Pires, 2004): (1) é um programa de distribuição gratuita, com código fonte disponível, sua linguagem é simples e de fácil aprendizado; (2) possui um sistema de auxílio ao usuário (*help*); (3) apresenta recursos para geração de gráficos bidimensionais, tridimensionais e

animações; (4) implementa diversas funções para manipulação de matrizes, incluindo operações de união, acesso e extração de elementos, transposição, adição e multiplicação de matrizes; (5) permite trabalhar com polinômios, funções de transferência, sistemas lineares e grafos; (6) apresenta facilidades para a definição de funções; (7) permite o acesso a rotinas escritas nas linguagens FORTRAN ou C; (8) pode ser acessado por programas de computação simbólica como o Maple, que é um programa comercial, ou o MuPAD, que é um programa livre para uso em instituições de ensino/pesquisa; (9) suporta o desenvolvimento de conjuntos de funções voltadas para aplicações específicas, os chamados *toolboxes*.

Um dos objetivos secundários deste trabalho é mostrar que a utilização de programas livres e de código aberto traz grandes vantagens do ponto de vista do usuário. Algumas destas vantagens são apresentadas em Palacios (2001) e destacadas a seguir: (1) a última versão do programa está sempre disponível, geralmente através da Internet; (2) o programa pode ser legalmente utilizado, copiado, distribuído, modificado; (3) os resultados obtidos podem ser divulgados sem nenhuma restrição; (4) os programas desenvolvidos podem ser transferidos para outras pessoas sem imposições ou constrangimentos de qualquer natureza; (5) o acesso ao código fonte, evitando surpresas desagradáveis; (6) participação em uma comunidade cujo principal valor é a irrestrita difusão do conhecimento.

O presente trabalho tem como objetivo principal propor sugestões de como utilizar os programas Matlab, Maple e Scilab no ensino de Cálculo Numérico e relatar a experiência dos autores no uso destes programas em aula. Além disso, pretende-se mostrar um comparativo entre a implementação através dos três programas, através dos exemplos apresentados. Nota-se que a aprendizagem independe do ambiente computacional adotado, o aluno deve desenvolver a lógica ao elaborar as etapas para a solução de um problema, para depois programá-lo e analisar os resultados obtidos. A “chave” da disciplina Cálculo Numérico é o uso da lógica matemática na solução dos problemas.

3. APLICAÇÕES EM CÁLCULO NUMÉRICO

Com o objetivo de apresentar a versatilidade e robustez destes programas, na solução numérica de problemas, são apresentados alguns métodos da disciplina Cálculo Numérico e a forma como os mesmos podem ser resolvidos através dos três programas abordados neste trabalho.

A disciplina de Cálculo Numérico consiste na resolução de vários problemas que podem ser resolvidos manualmente ou por programas. A solução fornecida pelos programas geralmente são aproximações da solução real, isto ocorre devido ao sistema de representação da máquina, ou a base numérica empregada, ou o número total de dígitos na mantissa, ou da forma como são realizadas as operações ou do compilador utilizado (Mariani, 2002). Estes aspectos não serão abordados neste trabalho, no entanto vale ressaltar que todos os métodos são interrompidos quando um determinado valor aproximado satisfaz o problema.

A seguir serão descritas três aplicações: desenho de gráficos, zero de funções e resolução de sistemas lineares.

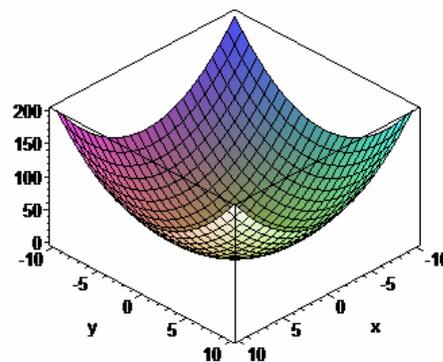
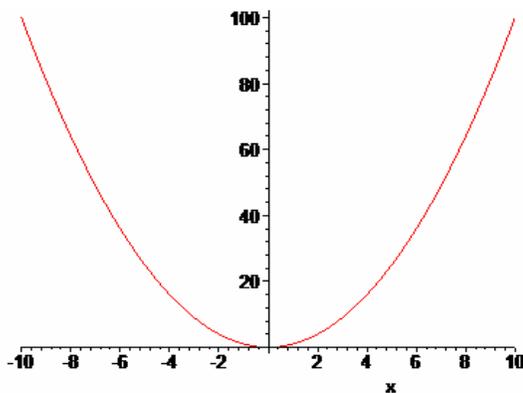
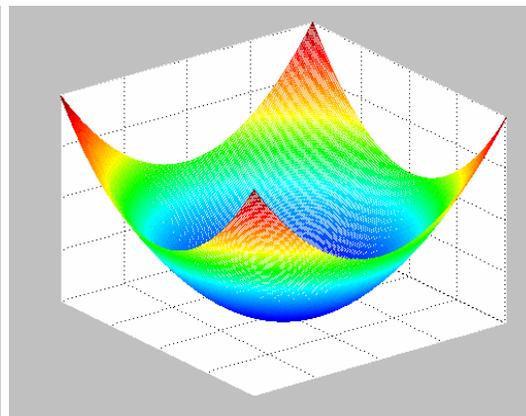
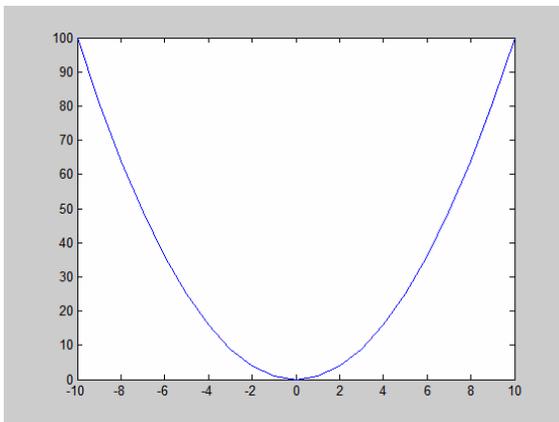
3.1 Geração de Gráficos

Um tópico usado em várias disciplinas, não só em Cálculo Numérico, é a representação gráfica de funções no espaço bidimensional ou tridimensional. Assim, na tabela 2 são

apresentados os principais comandos utilizados para ilustrar as funções $f(x) = x^2$ e $g(x, y) = z = x^2 + y^2$, em cada um dos programas. Na figura 1 são ilustrados os gráficos obtidos primeiro para o Matlab, em seguida o Maple e após o Scilab. A tabela 1 ilustra os comandos usados para a geração dos gráficos nos respectivos programas. Percebe-se que todos os programas ilustram adequadamente as funções solicitadas. Nota-se na figura 1 que a representação gráfica do programa Scilab ainda está deficiente, principalmente para o gráfico tridimensional. Para criar este gráfico foi necessário criar uma matriz $g(i,j)$, conforme apresentado na tabela 2, e o Scilab só permite utilizar no laço do for valores positivos o que prejudicou a representação gráfica.

Tabela 1. Comandos para construir gráficos 2D e 3D.

Matlab	Maple	Scilab
<pre>X=-10:10; y=x^2; y=x.^2; plot(x,y)</pre>	<pre>plot(x^2,x);</pre>	<pre>x=[-10:0.1:10]'; y=x^2; plot2d(x,y,rect=[-10,0,10,100])</pre>
<pre>[x,y]=meshgrid(-10:0.1:10); z=x.^2+y.^2; mesh(x,y,z)</pre>	<pre>plot3d(x^2+y^2,x=-10..10,y=-10..10);</pre>	<pre>x=1:10; y=1:10 for i=1:10 for j=1:10 g(i,j)=i^2+j^2;end;end; plot3d(x,y,u)</pre>



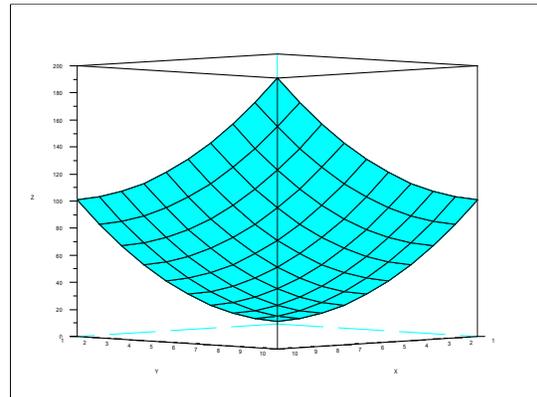
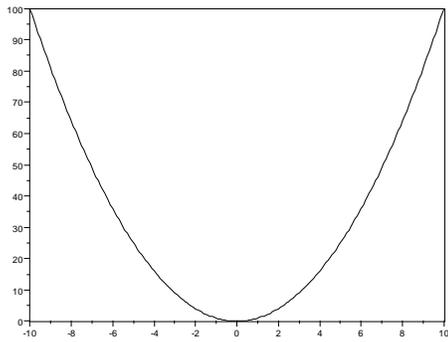


Figura 1. Funções $f(x)$ (esquerda) e $g(x,y)$ (direita) no Matlab, Maple e Scilab.

3.2 Zero de Funções

Um assunto importante na disciplina Cálculo Numérico é a obtenção numérica de zeros de funções ou raízes de equações. O zero de uma classe restrita de funções algébricas pode ser obtido de forma rápida e fácil, contudo na maior parte das funções algébricas e transcendentais esta não é uma tarefa simples e os métodos numéricos foram criados para auxiliar na obtenção de tais zeros (Penny e Lindfield, 2000). Um dos métodos vastamente usados é o método de Newton-Raphson. O algoritmo deste método (Burden e Faires, 2000) é apresentado na tabela 2.

Tabela 2. Algoritmo do método de Newton-Raphson.

Algoritmo do método de Newton-Raphson

Seja $f(x)$ contínua em $[a, b]$ e tal que $f(a) \cdot f(b) < 0$. Supor que as condições $f(x)$ e $f'(x)$ sejam não nulas e preservem o sinal em $[a, b]$ e tal que $f(x_0) \cdot f''(x_0) > 0$, $f'(x) \neq 0$ estejam satisfeitas (Burden e Faires, 2000).

Dados iniciais

precisão ε

$k = 0$ (contador de iterações)

x_0 (aproximação inicial da raiz)

Calcular $f = f(x_0)$ e $df = f'(x_0)$.

2) Enquanto $|f/df| > \varepsilon$ então

$$x_{k+1} = x_k - \frac{f}{df}$$

$k = k+1$

$x_k = x_{k+1}$

$f = f(x_k)$

$df = f'(x_k)$

senão escolha $\bar{\lambda} = x_{k+1}$ fim.

Nas tabelas 3, 4 e 5 são ilustradas as formas como o método de Newton-Raphson pode ser programado nos três ambientes computacionais.

Tabela 3. Método de Newton-Raphson no programa Matlab.

function [x, iter] = fnewton(f,df,x,tol)	iter = 0;x0 = x;
%	d = eval(f,x0)/eval(df,x0);
% Encontra a raiz de f(x) usando Newton	while abs(d)>tol
% Chame no prompt do Matlab	x1 = x0-d;
% [x, iter] = fnewton(f,df,x,tol)	iter = iter+1;
% onde f e a funcao f(x), df e a derivada	x0 = x1;
% x e o valor inicial e tol e a tolerancia	x=x0;
% f e df devem ser fornecidas	d = eval(f,x0)/eval(df,x0);
% Ex. fnewton('x*log(x)-3.2','1+log(x)',3,0.0001)	end
	res = x0;

Tabela 4. Método de Newton-Raphson no programa Maple.

```
newtonraphson := proc(x0,f,erro,itmax)
print("i", "xk=", "f(xk)=");
df:= unapply(diff(f(x),x),x);
x1:=evalf(x0-f(x0)/df(x0));
for i from 0 by 1 while (abs(f(x1)) > erro or abs(x1-x0) > erro) and (i < itmax) do
x0 := x1;
x1 := evalf(x0-f(x0)/df(x0));
end do;
print(i, x1, evalf(f(x1)));
end proc;
```

Tabela 5. Método de Newton-Raphson no programa Scilab.

<pre>function fnewton(x0,tol) //arquivo fnewton.sci // Encontra a raiz de f(x) usando Newton // f é a função f(x), df é a derivada da função f(x) // x0 é o valor inicial e tol é a tolerância requerida // iter = 0; d = f(x0)/df(x0); while (abs(d)> tol) x1 = x0-d iter = iter+1; x0 = x1;</pre>	<pre> d = f(x0)/df(x0); end; printf('iter=%d\n x0=%f\n',iter,x0) endfunction // arquivo func1.sci function[a] = f(x) a = x.*log(x)-3.2 // arquivo func2.sci function[b] = df(x) b = log(x) +1</pre>
--	--

Na tabela 5 nota-se a especificação de três funções diferentes, uma declarando a própria função $f(x)$ (arquivo *func1.sci*) outra declarando a derivada da função $f(x)$ (arquivo *func2.sci*) e a última função apresenta a resolução do próprio método de Newton-Raphson (arquivo *fnewton.sci*), implementado no programa Scilab. Para executar o programa no Scilab deve-se seguir as etapas:

```
-->getf('func1.sci')
-->getf('func2.sci')
-->getf('fnewton.sci')
-->fnewton(1,0.00001)
```

Neste caso, os valores 1 e 10^{-5} foram utilizados como dados de entrada para obter o zero da função especificada.

No Maple para obter a raiz de uma função também pode-se usar o comando *solve* com suas variantes. Os arquivos salvos no Matlab possuem extensão *.m*, no Maple a extensão é *.mws* e no Scilab a extensão é *.sci*. Para avaliar uma função ou expressão no Matlab usa-se o comando *feval* (quando o nome da função é chamado) ou *eval* (quando a própria função é chamada), no Maple o comando *evalf* e no Scilab basta digitar $f(x_0)$, onde x_0 é o valor onde a função será avaliada. O Matlab não calcula algebricamente a derivada de uma função, enquanto o Maple o faz com o comando *diff*. O Scilab calcula numericamente a derivada através do comando *diff*, sendo necessário especificar o intervalo de derivação. No exemplo dado a função derivada foi fornecida.

3.3 Resolução de Sistemas Lineares

Outro tópico apresentado na disciplina Cálculo Numérico é a solução de sistemas de equações lineares, assunto este que é de grande importância nos Cursos de Computação e de Engenharia. No Matlab pode-se resolver um sistema de equações lineares através de métodos diretos. Métodos diretos são aqueles que a menos de erros de arredondamento, fornecem a solução exata do sistema linear, caso ela exista, após um número finito de operações. A inversa da matriz pode ser usada através do comando *inv* ou através do comando *linsolve*; a decomposição de Cholesky usando o comando *chol* ou a decomposição LU usando o comando *lu*. Porém os demais métodos devem ser programados utilizando as ferramentas disponíveis no

Matlab. Até mesmo os métodos anteriormente citados podem ser programados e os alunos são incentivados a fazê-lo.

No Maple os métodos diretos estão todos disponíveis em forma de funções no pacote *linalg* (de Álgebra Linear), a função *inverse* pode ser usada para obter a matriz inversa, a função *gaussjord* para utilizar o método de Gauss-Jordan, a função *LUdecomp* para utilizar o método de Decomposição LU e a função *cholesky* para o método de Cholesky. Da mesma forma no Scilab muitos comandos já estão disponíveis para resolver sistemas, alguns semelhantes ao Matlab.

Nas tabelas 6 a 8 será apresentado o método de Eliminação de Gauss para os programas Matlab, Maple e Scilab, respectivamente. O método de eliminação de Gauss consiste em transformar o sistema linear original num sistema linear equivalente cuja matriz dos coeficientes fique na forma triangular superior, onde a solução é feita de forma retroativa. Dois sistemas são equivalentes quando possuem a mesma solução (Burden e Faires, 2000).

Tabela 6. Método de Eliminação de Gauss no programa Matlab.

<pre>function [x,iter] = EliGauss (Ab) n_l = size(Ab,1); % no. linhas da matriz estendida Ab n_c = size(Ab,2); % no. colunas da matriz estendida Ab for k=1:(n_l-1) for i=(k+1):n_l m = -Ab(i,k)/Ab(k,k) Ab(i,k) = 0 for j=(k+1):n_c Ab(i,j) = Ab(i,j) + m*Ab(k,j) end end</pre>	<pre>end end x(n_l) = Ab(n_l,n_c)/Ab(n_l,n_l); for i=(n_l-1):-1:1 s = 0 for j=i+1:n_l s = s + Ab(i,j)*x(j) end x(i) = (Ab(i,n_c)-s)/Ab(i,i) end</pre>
---	---

Tabela 7. Método de Eliminação de Gauss no programa Maple.

<pre>restart:with(linalg): gauss:=proc(C,B) local A,j,i,k,n,m,x,s; A:=concat(C,B); n:=rowdim(A); for k to n-1 do for i from k+1 to n do m:=evalf(-A[i,k]/A[k,k]); for j from k+1 to n+1 do A[i,j]:=evalf(A[i,j]+m*A[k,j]); od; A[i,k]:=0; od; od; # Retrosubstituição x:=vector(n);</pre>	<pre>x[n]:=A[n,n+1]/A[n,n]; for i from n-1 to 1 by -1 do s:=A[i,n+1]; for j from i+1 to n do s:=s-A[i,j]*x[j]; od; x[i]:=evalf(s/A[i,i]); od; x:=convert(x,matrix); #resultado um vetor coluna RETURN(op(x)); end: # Chamando Gauss para um sistema Ax=b. A1:=linalg[matrix](3,3,[1,2,-3,2,-4,1,3,2,1]); B1:=linalg[matrix](3,1,[5,0,5]); x:=gauss(A1,B1);</pre>
---	--

No Matlab para compilar o programa basta digitar `EliGauss(Ab)` na linha de comando, o programa se chama `EliGauss.m`, onde `Ab` é a matriz estendida, isto é, contendo os elementos da matriz `A` e do termo independente `b`, do sistema de equações lineares. No Maple a execução do programa é apresentada no final da tabela 7. Para executar o programa apresentado na tabela 8 no Scilab os seguintes passos devem ser seguidos:

```
-->getf('x.sci')
-->Eg(Ab)
```

Tabela 8. Método de Eliminação de Gauss no programa Scilab.

<pre>function x = Eg(Ab) // Programa que resolve o sistema Ax=b através do método de Eliminação de Gauss // nl é o número de linhas e nc é o número de colunas da matriz Ab [nl, nc]=size(Ab); for k =1:nl-1, for i=(k+1):nl, m=-Ab(i,k)/Ab(k,k); Ab(i,k)=0; for j=k+1:nc, Ab(i,j)=Ab(i,j)+ m*Ab(k,j); end, end, end,</pre>	<pre>n=nc-1; A = Ab(:,1:n); // matriz A b = Ab(:,nc) // vetor b x = zeros(n,1); x(n) = b(n)/A(n,n); for i = n-1:-1:1, soma = 0, for j = i+1:n, soma = soma + A(i,j)*x(j); end, x(i)=(b(i)-soma)/A(i,i); end Endfunction</pre>
---	---

4. CONCLUSÕES

Após a apresentação breve dos três ambientes computacionais e a aplicação dos mesmos para ilustrar gráficos de funções, obter o zero de uma função via o método de Newton-Raphson e resolver o sistema $Ax = b$ pelo método de Eliminação de Gauss percebe-se que todos os programas analisados têm potencial. Contudo, deve-se dar destaque ao programa Scilab, um programa gratuito e distribuído com o código fonte. Assim, todos os alunos podem ter acesso ao programa em suas casas sem restrições, bastando para isto ter um computador pessoal, enquanto os demais programas aqui apresentados são comerciais, o que dificulta o acesso de grande parte dos alunos.

A análise dos programas mostrou que todos são adequados para o ensino de Cálculo Numérico. O Matlab e o Scilab são excelentes ferramentas de suporte para as linhas de pesquisa onde o uso de computadores na resolução numérica de problemas é intensivo, enquanto o Maple tem maior aplicação na solução simbólica e algébrica dos problemas.

Os conceitos de programação de computadores necessários para a utilização dos três ambientes computacionais estão relacionados a manipulação de estruturas de controle de repetição e decisão, além da elaboração de funções. Portanto pode-se afirmar que os alunos com conhecimentos básicos de programação têm condições de utilizar os três ambientes computacionais analisados.

Observou-se também que todos os ambientes computacionais são de fácil assimilação para os alunos, não havendo diferenças significativas a respeito das facilidades de aprendizagem e utilização. Como proposta de trabalhos futuros, pretende-se explorar os três programas em aplicações mais específicas dentro dos cursos de computação e engenharia, como, por exemplo, explorar determinados *toolbox* comparando suas funcionalidades e o uso dos programas para a realização de visualização científica de dados

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, R. Q. Software Livre na Educação. In: Boletim EAD - Unicamp/CentrodeComputação/EquipeEAD.http://www.ead.unicamp.br/ead/index_html?foco=Publicacoes, 2002. Acesso em 20/04/2005.

BURDEN, R. L. e FAIRES J. D., **Numerical Analysis**, 7ª Edição, PWS Publishing Co, 2000.

DIRENE, A. I. e PIMENTEL, A. R., Medidas Cognitivas no Ensino de Programação de Computadores com Sistemas Tutores Inteligentes. In: Simpósio Brasileiro de Informática na Educação, **Anais**: Fortaleza, 1998.

FREIRE, P., **Professora sim Tia não – Cartas a quem Ousa Ensinar**, São Paulo, Olho D'água, 14ª Edição, 2003.

LIMA, E. L., Conceituação, Manipulação e Aplicações. In: Revista do Professor de Matemática, 41. SBM. <http://cenp.edunet.sp.gov.br/RPM>, 1995. Acesso em 10/03/2005.

MARIANI, V. C., Laboratório Computacional na Disciplina de Cálculo Numérico – Um Relato, In: Congresso Brasileiro de Ensino de Engenharia, SP. **Anais**: Piracicaba, 2002.

MARTIM, E. e MARIANI, V. C., Modelagem Matemática de um Tanque de Armazenamento, In: Congresso Brasileiro de Ensino de Engenharia, **Anais**: Brasília, 2004.

PALACIOS, J. O., An Introduction to the Treatment of Neurophysiological Signals using Scilab-Version 0.02. <http://www.neurotraces.com/scilab/scilab2/node2.html>, 2001.

PENNY, J. e LINDFIELD, G., **Numerical Methods using Matlab**, Prentice Hall, 2000.

PIRES, P. S. M., Introdução ao Scilab Versão 3.0. – Apostila - UFRN <http://www.leca.ufrn.br/~pmotta/sciport.pdf>, 2004. Acesso em 01/03/2005.

ROCHA, A. A., SILVA, B. F., PRUCOLE, E. e BRASIL, O. I., O Ensino da Transferência de Calor em um Ambiente Computacional Interativo Integrado utilizando o Maple, In: Congresso Brasileiro de Ensino de Engenharia, **Anais**: Brasília, 2004.

USE OF THE MAPLE, MATLAB AND SCILAB INT THE ENGINEERING COURSES

Abstract. *The present work has as objective to present the main characteristics of the computational programs Maple, Matlab and Scilab in the education of Numerical Calculus present in all Engineering courses. These computational programs are used with intention to stimulate and to facilitate to the learning becoming more attractive the study and agreement of the main methods of Numerical Calculus. In all these programs the students has the option of only entering with the data and getting the answer and to use these programs to create its codes and to get its answers. Thus, the programs provide a better exploitation of the students in discipline, enabling them to interpret the problems and the phenomenon that are present in these problems.*

Key- Words: *Numerical Calculus, Maple, Matlab, Scilab, Engineering, Programation*