

# UTILIZAÇÃO DO SOFTWARE SCILAB COMO FERRAMENTA COMPUTACIONAL EM SUBSTITUIÇÃO AO MATLAB® – APLICAÇÃO EM SISTEMAS COM REPRESENTAÇÃO LINEAR

**Francisco Damasceno Freitas** – e-mail: [ffreitas@ene.unb.br](mailto:ffreitas@ene.unb.br)  
Universidade de Brasília, Faculdade de Tecnologia – Departamento de Engenharia Elétrica  
Departamento de Engenharia Elétrica, Universidade de Brasília  
CEP 70910-900 – Brasília - DF

***Resumo:** Este artigo mostra uma aplicação do software Scilab em um problema de computação numérica em engenharia. São apresentadas comparações entre o Scilab - que é um software livre e aberto - e o Matlab®. O problema simulado diz respeito à análise da estabilidade linear de um sistema de potência. Simultaneamente à formulação do problema e sua modelagem, são apresentados os códigos relativos ao Scilab bem como os respectivos resultados. Adicionalmente, são comentadas as comparações com códigos de funções similares do Matlab®. A evidência apresentada pelo amplo kernel de aplicações do Scilab e a disponibilidade de telas de ajuda, contendo exemplos ilustrativos, faz desse software uma excelente ferramenta para utilização em computação numérica aplicada aos mais diversos cursos de engenharia.*

***Palavras-chave:** Scilab, Software livre e aberto, Simulação computacional, Ensino.*

## 1. INTRODUÇÃO

Nos últimos anos, a discussão sobre a utilização de software livre tem sido motivada, em parte, pelo interesse em se dispor do código fonte dos aplicativos. Uma outra motivação é o fato desse tipo de software ficar disponível para acesso, sem custos comerciais. As duas motivações apontadas são igualmente relevantes, principalmente, em um contexto acadêmico. Exatamente, porque é interessante se trabalhar com o código fonte dos aplicativos, estando aí uma forma de se desenvolver novos algoritmos, novas técnicas numéricas etc. E também, porque os recursos que deveriam ser investidos em software podem ser destinados a outros investimentos, como aquisição de novos hardwares, atualização da infraestrutura dos laboratórios computacionais, obtenção de referências bibliográficas, entre outras alternativas.

O *Matlab*® (NAKAMURA (2001)) é um exemplo de software para computação numérica, sendo muito utilizado atualmente nas mais diversas áreas de estudos. Trata-se de software comercial cujo valor da licença e das suas *toolboxes* atinge custos relativamente elevados para a realidade que hoje atravessa as universidades públicas. Há softwares livres que se tornam atrativos para uso sem que se observe deficiência na qualidade das simulações. O aplicativo *Scilab* é uma dessas opções, conforme se observa em Scilab CONSORTIUM (2004), PIRES e ROGERS (2002) e em PIRES (2001).

Este artigo aborda uma aplicação com o *Scilab*, sendo que o principal objetivo é a divulgação do potencial do software para trabalhos que envolvem computação numérica. As aplicações abordadas aqui são limitadas a problemas com representação linear. Entretanto, tais aplicações constituem-se em apenas uma pequena percentagem do *kernel* do programa. Busca-se descrever de forma sucinta o potencial do software, bem como destacar as principais

características do aplicativo. Alguns exemplos envolvendo simulações numéricas são mostrados para fins de ilustrar a apresentação dos resultados, as manipulações dos comandos, e as analogias com o aplicativo *Matlab*®, como é feito em URROZ (2004).

## 2. O APLICATIVO *Scilab*

O *Scilab* foi concebido com a finalidade de ser um ambiente para resolução de problemas numéricos. Ele foi desenvolvido desde 1993 por um grupo de pesquisadores do *Institute National de Recherche en Informatique et en Automatique* (INRIA) e *École Nationale des ponts et Chaussées* (ENPC), sediados na França. Atualmente, é mantido e desenvolvido pelo consórcio *Scilab*, criado em maio de 2003. O software pode ser baixado livremente, via *internet*, do sítio [www.scilab.org](http://www.scilab.org). O aplicativo está na sua versão 2.7.2 e versão beta 3.0. O programa é distribuído livremente pela *internet* desde 1994, sendo utilizado hoje em ambientes acadêmicos e industriais, em todo o mundo. Na página da *internet* dedicada ao aplicativo podem ser encontrados livros e artigos relativos sobre o assunto, *links* para locais que utilizam o programa, grupos de discussão, contribuições de usuários (*toolboxes*), demonstrações, documentações, entre outras informações. A primeira conferência internacional sobre o *Scilab* está programada para dezembro de 2004, sendo organizada pelo consórcio que mantém e desenvolve o software. Os tópicos que serão abordados nessa conferência, preferencialmente, são:

- Simulação;
- Sistemas dinâmicos;
- Processamento de imagens e de sinais;
- Controle;
- Modelagem e identificação;
- Otimização;
- Métodos computacionais;
- Aplicações em engenharia;
- Ensino.

O *Scilab* é distribuído com o código fonte. Ele é disponibilizado, também, em versões pré-compiladas para plataformas como o Windows® e o Linux. Por ser um sistema aberto, o usuário pode definir novos tipos de dados e novas operações sobre esses tipos de dados.

### 2.1 O Prompt no *Scilab*

A tela mostrada na Figura 2.1 ilustra como é a página de trabalho para comandos utilizando o *Scilab*. Os usuários que utilizam o ambiente computacional *Matlab*® podem observar semelhanças entre os dois ambientes. A sintaxe dos comandos também é muito semelhante. Mas há algumas diferenças.

#### ***Definição de funções e outros comandos***

As funções definidas no *Scilab* não são consideradas arquivos separados, tal como ocorre no *Matlab*®. Elas constituem variáveis no ambiente *Scilab*. Uma ou mais funções definidas pelo usuário podem ser definidas em um único arquivo. E o nome do arquivo não necessariamente precisa estar relacionado ao nome da função. Ainda, as funções não são automaticamente carregadas no *Scilab*, como são no *Matlab*® após serem comandadas através dos seus nomes. Normalmente, deve ser executado o comando *getf("function\_name")*, antes de usar a função.

As funções podem também ser definidas on-line (são as denominadas funções internas), através do comando *deff*.

Para executar um arquivo com código em *Scilab* é preciso usar o comando *exec("filename")*. Já no *Matlab*® basta digitar o nome do arquivo e executá-lo.

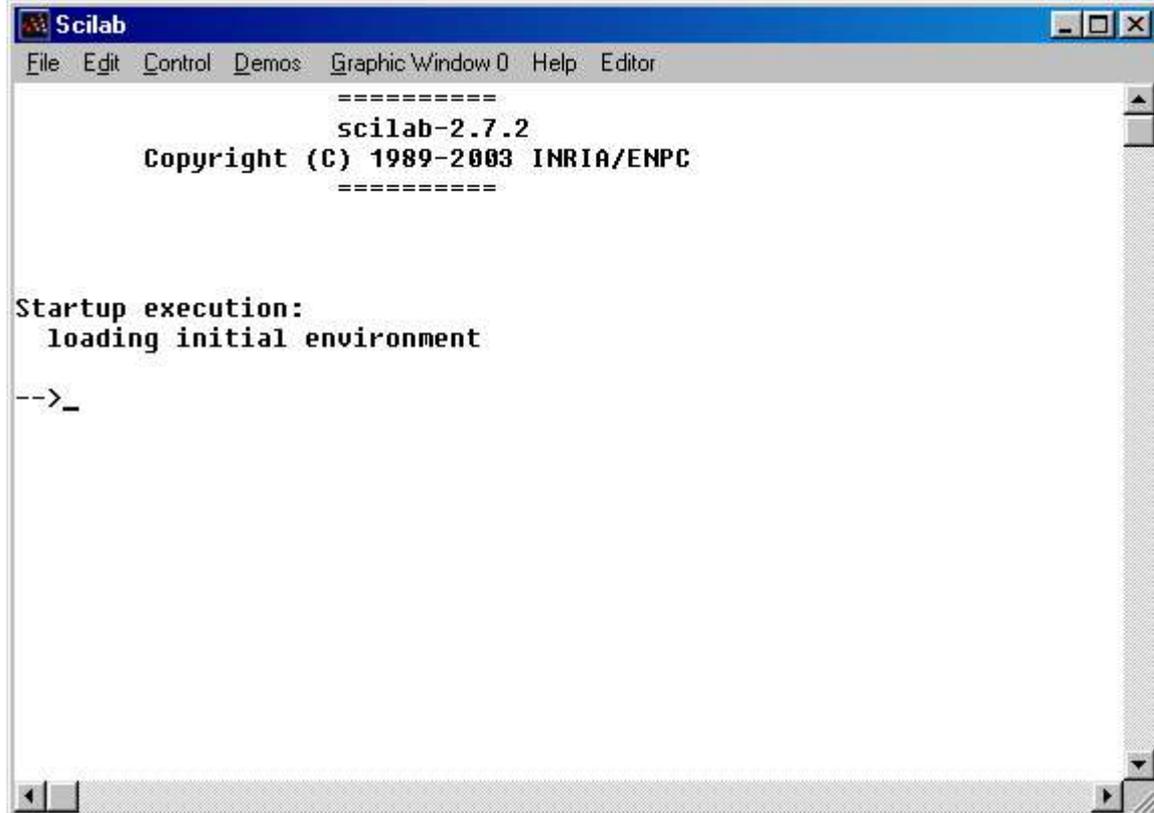


Figura 2.1 – Prompt do *Scilab*.

Tanto o *Scilab* quanto o *Matlab*® podem efetuar simulações através de um ambiente em que é possível se descrever um sistema físico através de diagramas em blocos. No *Scilab* esse ambiente é denominado *Scicos* e no *Matlab*® é chamado *Simulink*.

A maioria das funções pré-definidas nos aplicativos *Scilab* e *Matlab*® é idêntica. Algumas delas apresentam pequenas diferenças em relação à sintaxe.

Para os usuários acostumados ao ambiente *Matlab*®, no *Scilab* é possível emular alguns tipos de comandos com a finalidade de utilizar funções do *Matlab*®. Essas funções começam com *mtlb\_*. Os comandos listados a seguir ilustram algumas das funções.

<code>mtlb</code>	<code>mtlb_all</code>	<code>mtlb_any</code>	<code>mtlb_axes</code>
<code>mtlb_cell</code>	<code>mtlb_choices</code>	<code>mtlb_clf</code>	<code>mtlb_cumsum</code>
<code>mtlb_diff</code>	<code>mtlb_e</code>	<code>mtlb_eval</code>	<code>mtlb_exist</code>
<code>mtlb_eye</code>	<code>mtlb_fft</code>	<code>mtlb_filter</code>	<code>mtlb_find</code>
<code>mtlb_findstr</code>	<code>mtlbfliplr</code>	<code>mtlb_flipud</code>	<code>mtlb_fprintf</code>
<code>mtlb_fread</code>	<code>mtlb_fscanf</code>	<code>mtlb_fwrite</code>	<code>mtlb_get</code>
<code>mtlb_hold</code>	<code>mtlb_i</code>	<code>mtlb_ifft</code>	<code>mtlb_is</code>
<code>mtlb_ishold</code>	<code>mtlb_isreal</code>	<code>mtlb_length</code>	<code>mtlb_load</code>
<code>mtlb_loglog</code>	<code>mtlb_max</code>	<code>mtlb_mean</code>	<code>mtlb_median</code>
<code>mtlb_mesh</code>	<code>mtlb_meshdom</code>	<code>mtlb_min</code>	<code>mtlb_ones</code>
<code>mtlb_plot</code>	<code>mtlb_prod</code>	<code>mtlb_qz</code>	<code>mtlb_rand</code>
<code>mtlb_save</code>	<code>mtlb_semilogx</code>	<code>mtlb_semilogy</code>	<code>mtlb_sprintf</code>
<code>mtlb_sscanf</code>	<code>mtlb_subplot</code>	<code>mtlb_sum</code>	<code>mtlb_zeros</code>

No *Scilab*, as variáveis pré-definidas devem ser precedidas do símbolo “%”, como na variável `%i`, que é interpretada como  $\sqrt{-1}$ . Elas não podem ser redefinidas. No *Matlab*®, uma vez predefinida a variável `i` ou `j` para essa equivalência, ela não mais poderá ser usada como uma outra variável nas simulações. Isso não acontece no *Scilab*.

Enquanto no *Scilab* a indicação de comentário deve ser precedida por “//”, no *Matlab*® o indicador é “%”.

### ***Tela com comandos de ajuda***

O *Scilab* tem um menu no qual é possível acessar uma tela de ajuda. A Figura 2.2 mostra as opções disponíveis na versão 2.7.2 do aplicativo. A Figura 2.3 ilustra algumas informações referentes a funções de álgebra linear.

Como exemplo, é digitado o comando “*help spec*” no prompt do *Scilab*, para consulta sobre a ajuda em relação ao disponibilizado no aplicativo com relação a autovalores. Os quadros a seguir ilustram a seqüência de resultados encontrados.

## Scilab Function

spec - eigenvalues of matrices and pencils

### Calling Sequence

```
evals=spec(A)
[X,diagevals]=spec(A)
evals=spec(A,E)
[al,be]=spec(A,E)
[al,be,Z]=spec(A,E)
[al,be]=spec(A,E)
[al,be,Q,Z]=spec(A,E)
```

### Parameters

- \* A : real or complex square matrix
- \* E : real or complex square matrix with same dimensions as A
- \* evals : real or complex vector, the eigenvalues
- \* diagevals : real or complex diagonal matrix (eigenvalues along the diagonal)
- \* al : real or complex vector, al./be gives the eigenvalues
- \* be : real vector, al./be gives the eigenvalues
- \* X : real or complex invertible square matrix, matrix eigenvectors.
- \* Q : real or complex invertible square matrix, pencil left eigenvectors.
- \* Z : real or complex invertible square matrix, pencil right eigenvectors.

### Description

- spec(A) : evals=spec(A) returns in vector evals the eigenvalues of A.

[evals,X]=spec(A) returns in addition the eigenvectors A (if they exist). See also bdiag

- spec(A,B) : evals=spec(A,E) returns the spectrum of the matrix pencil  $sE - A$ , i.e. the roots of the polynomial matrix  $sE - A$ .

[al,be] = spec(A,E) returns the spectrum of the matrix pencil  $sE - A$ , i.e. the roots of the polynomial matrix  $sE - A$ . The eigenvalues are given by al./be and if be(i) = 0 the ith eigenvalue is at infinity. (For E = eye(A), al./be is spec(A)).

[al,be,Z] = spec(A,E) returns in addition the matrix Z of generalized right eigenvectors of the pencil.

[al,be,Q,Z] = spec(A,E) returns in addition the matrix Q and Z of generalized left and right eigenvectors of the pencil.

### REFERENCES

Matrix eigenvalues computations are based on the Lapack routines DGEEV and ZGEEV. Pencil eigenvalues computations are based on the Lapack routines DGGEV and ZGGEV.

### Examples

```
// MATRIX EIGENVALUES
A=diag([1,2,3]);X=rand(3,3);A=inv(X)*A*X;
spec(A)
//
x=poly(0,'x');
pol=det(x*eye()-A)
roots(pol)
//
[S,X]=bdiag(A);
clean(inv(X)*A*X)

// PENCIL EIGENVALUES
A=rand(3,3);
```

O comando equivalente no *Matlab*® é “*help eig*”. É interessante observar que nas consultas à ajuda do *Scilab*, na maioria das vezes, encontra-se um exemplo ilustrativo sobre a função investigada.

Informações semelhantes podem ser obtidas para as demais funções discriminadas no menu “ajuda” do aplicativo.

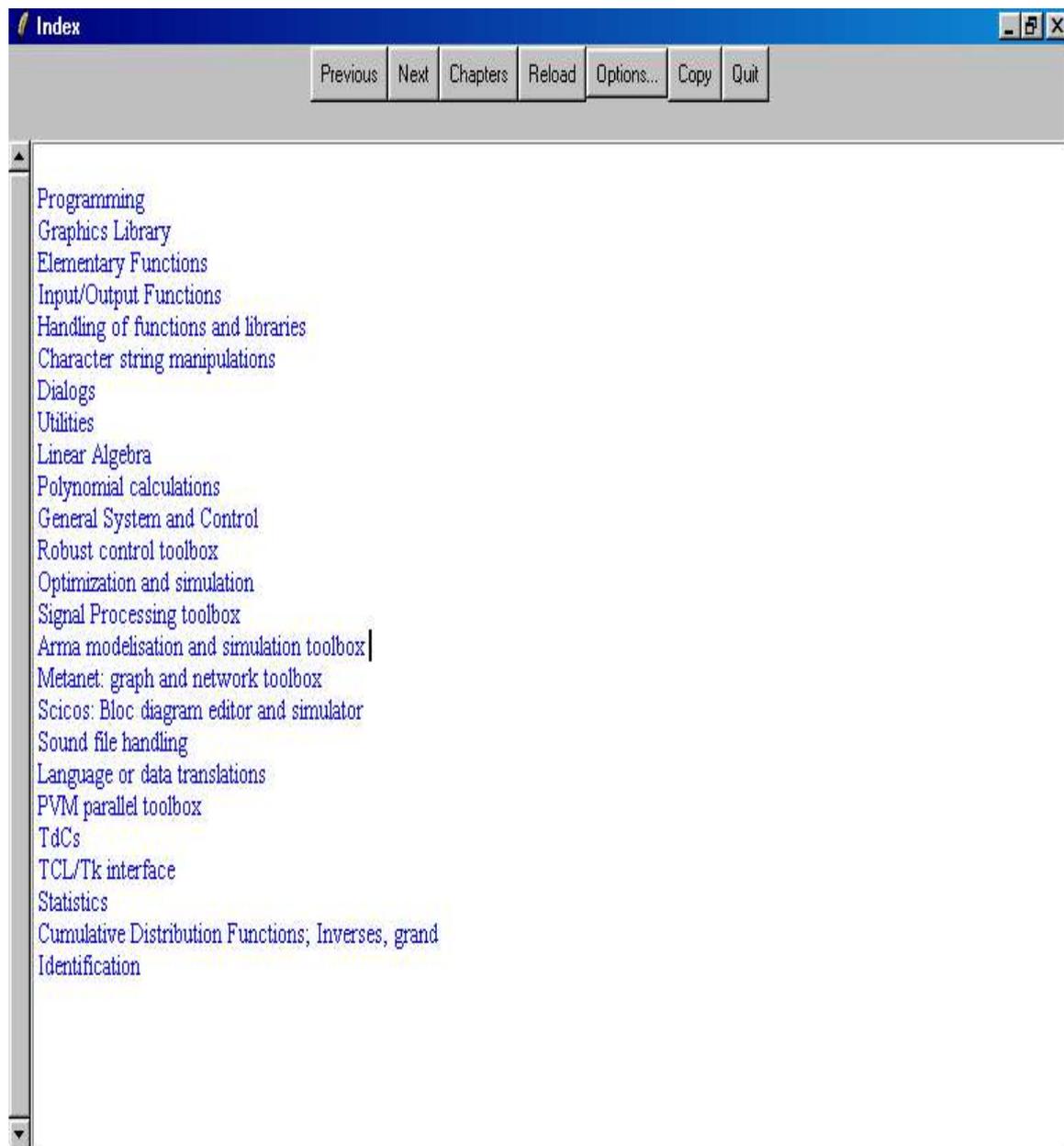


Figura 2.2 – Tela indicativa de opções no *help* do aplicativo.

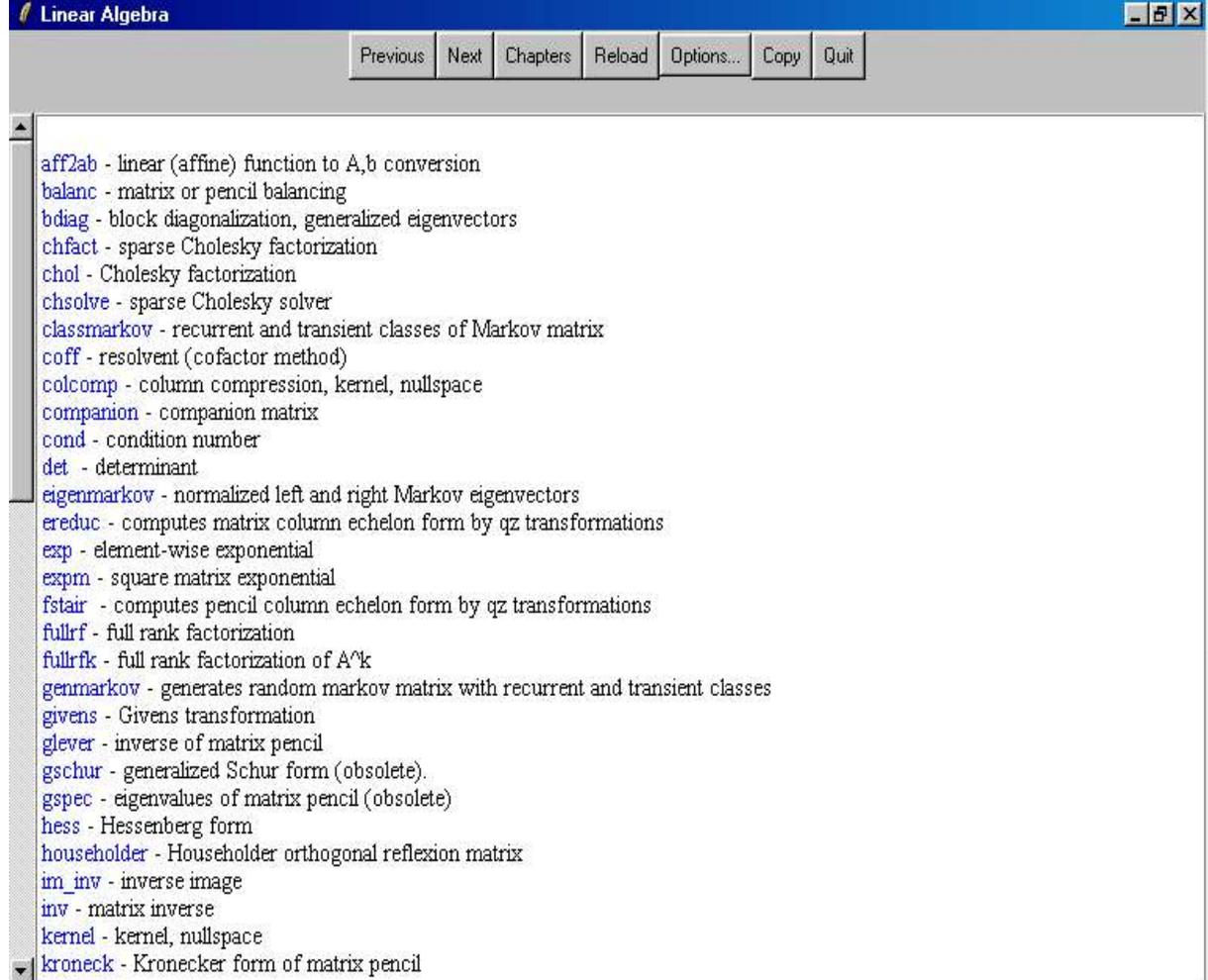


Figura 2.3 - Algumas opções de funções disponíveis sobre álgebra linear

## 2.2 Simulações através do Scilab

Nesta subseção são apresentados resultados referentes à aplicação do *Scilab* para resolução de problemas envolvendo a manipulação de sistemas lineares, tanto no domínio do tempo (através de variáveis de estados), como no domínio da frequência. Evidentemente, o problema físico tratado aqui poderia ser um outro, de natureza qualquer. Por limitação de espaço, o autor considera suficiente os detalhes apresentados para estender os resultados para aplicações em outras engenharias, em química, biologia, economia etc.

Suponha que o sistema físico que se queira representar seja apresentado na seguinte forma de espaço de estado:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (1)$$

onde  $A$  é denominada matriz de estado;  $B$  é a matriz de controle;  $C$  é a matriz de saída;  $D$  é a matriz de transmissão;  $x$  é um vetor contendo os estados e de dimensão  $n \times 1$ ; e  $y$  é um vetor contendo as variáveis de saída e de dimensão  $m \times 1$ .

O sistema descrito por (1) pode ser colocado no domínio da frequência ou na forma de uma função de transferência no domínio  $s$ . A transformação abaixo permite obter esse resultado.

$$H(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1}B + D \quad (2)$$

onde  $I$  é a matriz identidade e  $s$  é o operador da transformada de Laplace.

Tanto técnicas no domínio do tempo, quanto no domínio da frequência são utilizadas para análise e síntese de sistemas lineares. Uma das aplicações interessantes diz respeito ao projeto de controladores de processos ou plantas industriais. As matrizes definidas a seguir, retiradas de KUNDUR (1994), constituem exemplos de um sistema por representação de estado, utilizado na análise da estabilidade linear em sistemas de potência. Evidentemente, a dimensão abordada aqui é bastante modesta, pois o objetivo é apresentar uma aplicação do software *Scilab*, como ferramenta para apoio à resolução do problema. Problemas com dimensões maiores podem ser tratados em situações específicas. As matrizes de interesse são:

$$A = \begin{bmatrix} 0 & -0.1092 & -0.1236 & 0 & 0 & 0 \\ 376.99 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.1938 & -0.4229 & -27.3172 & 0 & 27.3172 \\ 0 & -7.3125 & 20.8391 & -50 & 0 & 0 \\ 0 & -1.0372 & -1.1738 & 0 & -0.7143 & 0 \\ 0 & -4.8404 & -5.4777 & 0 & 26.9697 & -30.3030 \end{bmatrix}$$

$$B = [0 \ 1 \ 0 \ 0 \ 0 \ 0]^T \quad C = [1 \ 0 \ 0 \ 0 \ 0 \ 0] \quad D = 0$$

Para análise da estabilidade linear desse sistema, são calculados os autovalores da matriz de estado. Foi criado um arquivo com terminação *.sci*, no qual foram inseridos os dados anteriores e em seguida realizada a execução do mesmo através do comando *exec*, encontrado no *menu file* da tela principal do aplicativo. A tela a seguir ilustra como foi feita a programação cujos comandos foram inseridos em arquivo.

```
// Definição da matriz de estado A
A=[0 -0.1092 -0.1236 0 0 0
 376.99 0 0 0 0 0
 0 -0.1938 -0.4229 -27.3172 0 27.3172
 0 -7.3125 20.8391 -50 0 0
 0 -1.0372 -1.1738 0 -0.7143 0
 0 -4.8404 -5.4777 0 26.9697 -30.3030];
// controle da segunda variável de estado
B=[0 1 0 0 0 0]';
// saída referente à primeira variável de estado
C=[1 0 0 0 0 0]; D=0;
```

Através do comando *spec(A)*, obtém-se os autovalores do sistema, cujo resultado, copiado a partir da tela de *prompt* do *Scilab* é o seguinte:

```
-->spec(A)
ans =

! - 1.0055015 + 6.6072843i !
! - 1.0055015 - 6.6072843i !
! - 39.096742          !
! - .7385135          !
! - 19.796971 + 12.822377i !
! - 19.796971 - 12.822377i !
```

O resultado encontrado é o mesmo apresentado na referência KUNDUR (1994). Através dos autovalores calculados, observa-se que o sistema é estável sob o ponto de vista de análise linear.

Caso seja necessário analisar o problema no domínio da frequência, através dos vários recursos disponíveis, como diagrama de Bode, de Nyquist, lugar das raízes, de pólos e zeros etc., é necessário efetuar conversão de espaço de estados para a forma de função de

transferência. A conversão pode ser feita a partir da função *syslin*. Esse procedimento também pode ser aplicado caso se tratasse da conversão inversa.

A função de transferência fica definida a partir das matrizes *A*, *B* e *C*. Caso *D* não seja definida nos argumentos da função, assume-se que o seu valor é zero e a função de transferência resultante será formada pela razão entre dois polinômios, onde a ordem do denominador é superior a do numerador.

O seguinte comando define então o sistema linear, cuja representação original ocorre a partir de variáveis de estado: `SI=syslin('c',A,B,C)`. A letra *c* no argumento da função indica que se trata de sistema contínuo. As linhas de comando abaixo mostram a continuação da programação para conversão para função de transferência. Esta etapa final é feita através da função *ss2tf*.

```
// conversão de representação de estado para função de transferência
s=poly(0,'s');
H=ss2tf(sistema)
```

A função de transferência resultante  $H(s)$ , tal como é gerada no *prompt* do *Scilab* é mostrada a seguir.

```
H =
      2      3      4
- 1903.2394 - 2846.9422s - 260.36974s - 8.8693162s - .1092s
      5
+ 8.527E-14s
-----
      2      3      4
717502.21 + 1073268.7s + 159519.79s + 31201.457s + 2366.8412s
      5 6
+ 81.4402s + s
```

Para exibir o diagrama de Bode relativo a essa função de transferência, define-se o seguinte código no *Scilab*.

```
// diagrama de bode
title='funcao de transferencia';
bode(sistema,0.01,100,'title')
```

Na função *bode*, a frequência mínima definida é igual a 0.01 Hz e a máxima igual a 100 Hz. A Figura 2.4 mostra os gráficos resultantes (módulo e fase) relativos ao diagrama. Outras informações poderiam ser obtidas, tais como: margem de fase, margem de ganho, diagrama de Nyquist, entre outras.

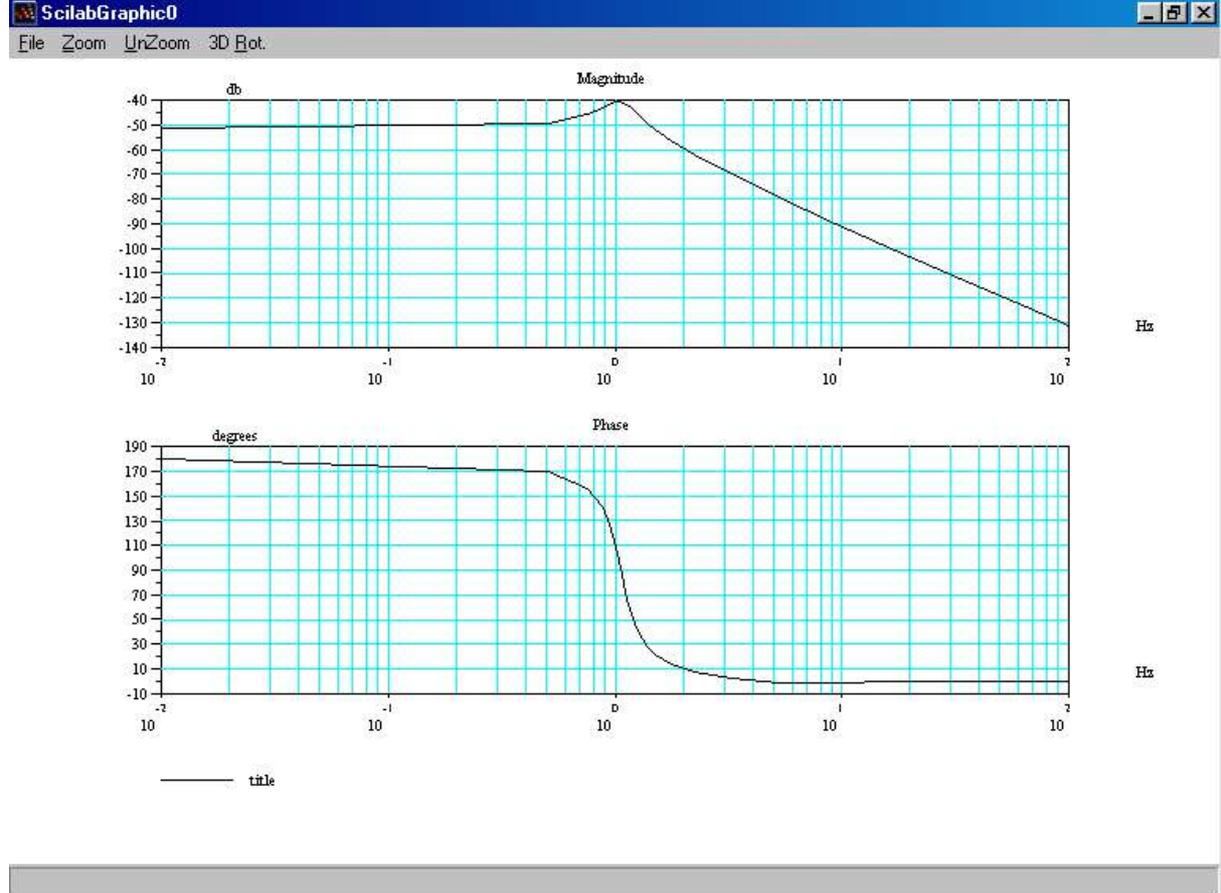


Figura 2.4 – Diagrama de Bode obtido a partir do *Scilab*.

Considere agora que o objetivo seja efetuar uma simulação no domínio do tempo. Suponha que um degrau unitário seja aplicado na entrada  $u(t)$  do sistema. O sistema nesse caso terá uma resposta  $y(t)$  que precisa ser calculada a partir dos dados do sistema. A programação apresentada a seguir realiza os cálculos e permite a exibição do gráfico.

```
// simulação no tempo do sistema definido por A,B e C.
// resposta a um degrau unitário
t=0:0.05:5;
y=csim('step',t,sistema);
//simple plot
//
xbasc()
plot2d(t,y)
```

No código apresentado, *csim* é a função que realiza a simulação ao degrau (*step*), considerando o sistema definido anteriormente. A simulação no tempo inicia-se no instante zero e se estende até 5 s. O passo utilizado é de 0,05 s. O comando *plot2d* (*t,y*) exibe a curva  $y(t)$ . A Figura 2.5 mostra a resposta  $y(t)$  ao degrau unitário. Observe-se que em função da definição da matriz *C*,  $y(t)$  corresponde à variável de estado  $x_1(t)$ .

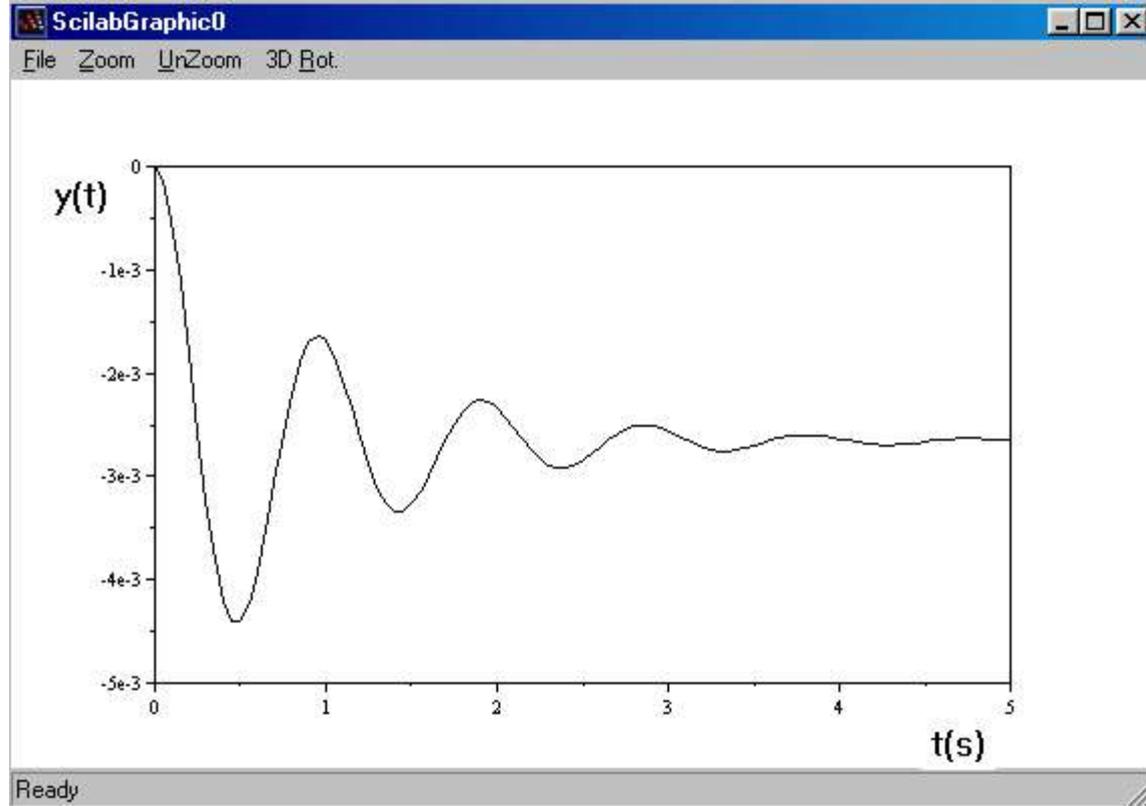


Figura 2.5 – Resposta  $y(t)$  ao degrau unitário.

Outros tipos de simulação poderiam ser incluídos, incluindo resolução de equações diferenciais ordinárias e parciais, identificação de sistemas, otimização, análise e síntese de sistemas discretos, ferramentas para análise estatística, para processamento de sinais e de imagens etc.

### 3. CONCLUSÕES

O artigo apresentou algumas aplicações do *Scilab*, que é uma ferramenta para simulação computacional disponibilizada livremente aos usuários. O software tem características bastante semelhantes ao *Matlab*® e muita das suas funções são idênticas ao aplicativo tradicional utilizado, principalmente em engenharia.

No artigo foi abordada uma aplicação voltada à simulação de um sistema com representação linear. Entretanto, outros tipos de problemas poderiam ter sido propostos e solucionados, utilizando a ferramenta computacional avaliada.

O software com as suas diversas funções é adequado para simulações computacionais e para utilização em cursos de graduação, pós-graduação e na indústria, pois possui, essencialmente, todos os recursos necessários para resolução de problemas numéricos. Ele pode substituir o software *Matlab*®, sem restrições em atividades de ensino. Essa substituição é fortemente recomendada, tendo em vista que o *Scilab* é um software livre e aberto, já sendo hoje utilizado em todo mundo.

### REFERÊNCIAS BIBLIOGRÁFICAS

KUNDUR, P. **Power System Stability and Control**. New York: McGraw-Hill, Inc., 1994.

NAKAMURA, S. **Numerical Analysis and Graphic Visualization with Matlab**® – **Second Edition**. New Jersey – USA: Prentice Hall PTR, 2002.

PIRES, P. S. M. Introdução ao Scilab – Versão 1.0. 2001. Nota Técnica, Departamento de Engenharia de Computação e Automação, Universidade Federal do Rio Grande do Norte, Natal-RN.

PIRES, P. S. M.; ROGERS, D. A. Free/Open Source Software: An Alternative for Engineering Students. In 32<sup>nd</sup> ASEE/IEEE Frontiers in Education Conference, Session T3G, 2002, Boston, MA-USA. **Meio Digital**, USA, 1994. p. 7-11.

Scilab CONSORTIUM. 2004. <http://scilabsoft.inria.fr/consortium/consortium.html>.

URROZ, G. E. Comparing Scilab and Matlab®. 2004. Class Notes, obtida em <http://www.engineering.usu.edu/cee/faculty/gurro/Scilab.html>.

## **USING SCILAB SOFTWARE AS A COMPUTATIONAL TOOL FOR REPLACING MATLAB® – A LINEAR SYSTEM REPRESENTATION APPLICATION**

***Abstract:** This paper shows an application based on Scilab software, considering a numerical computation problem in engineering study. Comparisons between Scilab – a free and open software - and Matlab® are carried out. The simulation is concerning to linear stability analysis applied to power systems. Simultaneously to the problem formulation and modelling, it has been presented the Scilab codes and their numerical results. Additionally, comments and comparisons about similar functions of Scilab and Matlab® are made. The large application of the Scilab kernel and facilities such as menu with illustrative examples make Scilab an excellent tool for using in engineering computational calculus field.*

***Key-words:** Scilab, Free and open software, Computational Simulation, Teaching.*