

Introdução ao *Software*



Licenciatura em Engenharia Informática

Licenciatura em Segurança Informática e Redes de Computadores



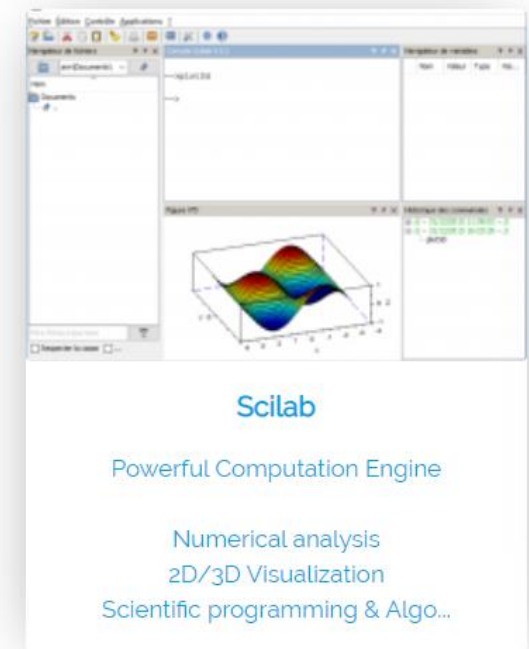
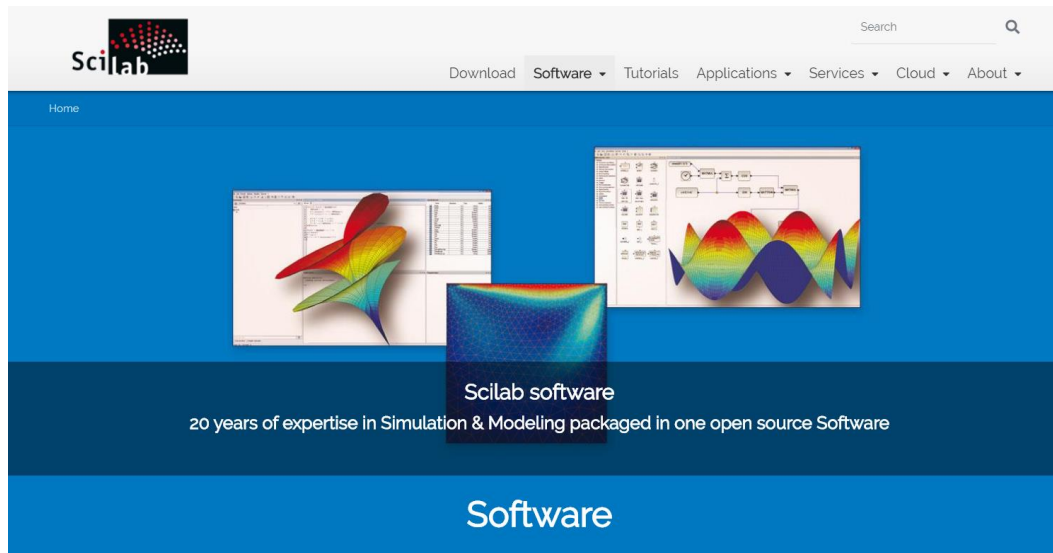
Tópicos

- O que é o *Scilab*?
- Um pouco de história ... e instalação do *Scilab*
- As principais janelas de interface com o utilizador
- Interação com a *Console*
- Na linha de Comandos
- Manipulação e informação sobre o espaço de trabalho
- Guardar e imprimir todo o trabalho
- Matrizes: criação, manipulação e operações



■ O que é o *Scilab*?

- É um *software* livre e de código aberto para computação numérica, utilizado hoje em dia, por exemplo, em muitas das aplicações de engenharia.
- É um *software* grátis e compatível com alguns dos sistemas operativos mais usuais: *MS Windows*, *Linux* e *Mac OS X*.





■ Um pouco de história...

- A história do *software Scilab* começa em França, em 1982, com o **Blaise** (um *software* criado no Instituto Francês de Pesquisa em Ciência da Computação e Controlo) desenvolvido, principalmente, por François Delebecque e Serge Steer, com o objetivo de fornecer uma ferramenta de controlo automático para investigadores;
- Em 1984, **Blaise** tornou-se **Basile** e foi distribuído durante alguns anos pela **Simulog**, o primeiro **INRIA** (Instituto Nacional Francês de Pesquisa em Ciências da Computação e Controlo);
- No início dos anos 90, **Simulog** parou de distribuir o **Basile**. O nome do *software* tornou-se **Scilab** e foi desenvolvido pelo **INRIA** e um grupo alargado de investigadores Franceses;
- A primeira versão, **Scilab 1.1**, foi lançada a 2 de janeiro de 1994 e disponibilizada *online*. O grupo *Scilab*, desenvolveu-o até ao final de 2002 (versão 2.7);



■ Um pouco de história e sua instalação

- No início de 2003, **INRIA** criou o **Scilab Consortium**, com o apoio de empresas e instituições acadêmicas, para garantir o futuro do software, seu desenvolvimento, manutenção, suporte e promoção;
- A **Scilab Enterprises** é fundada em 2010 e com o apoio do **INRIA** começa a desenvolver o software em 2012, oferecendo também serviços profissionais e de suporte;



■ Um pouco de história e sua instalação

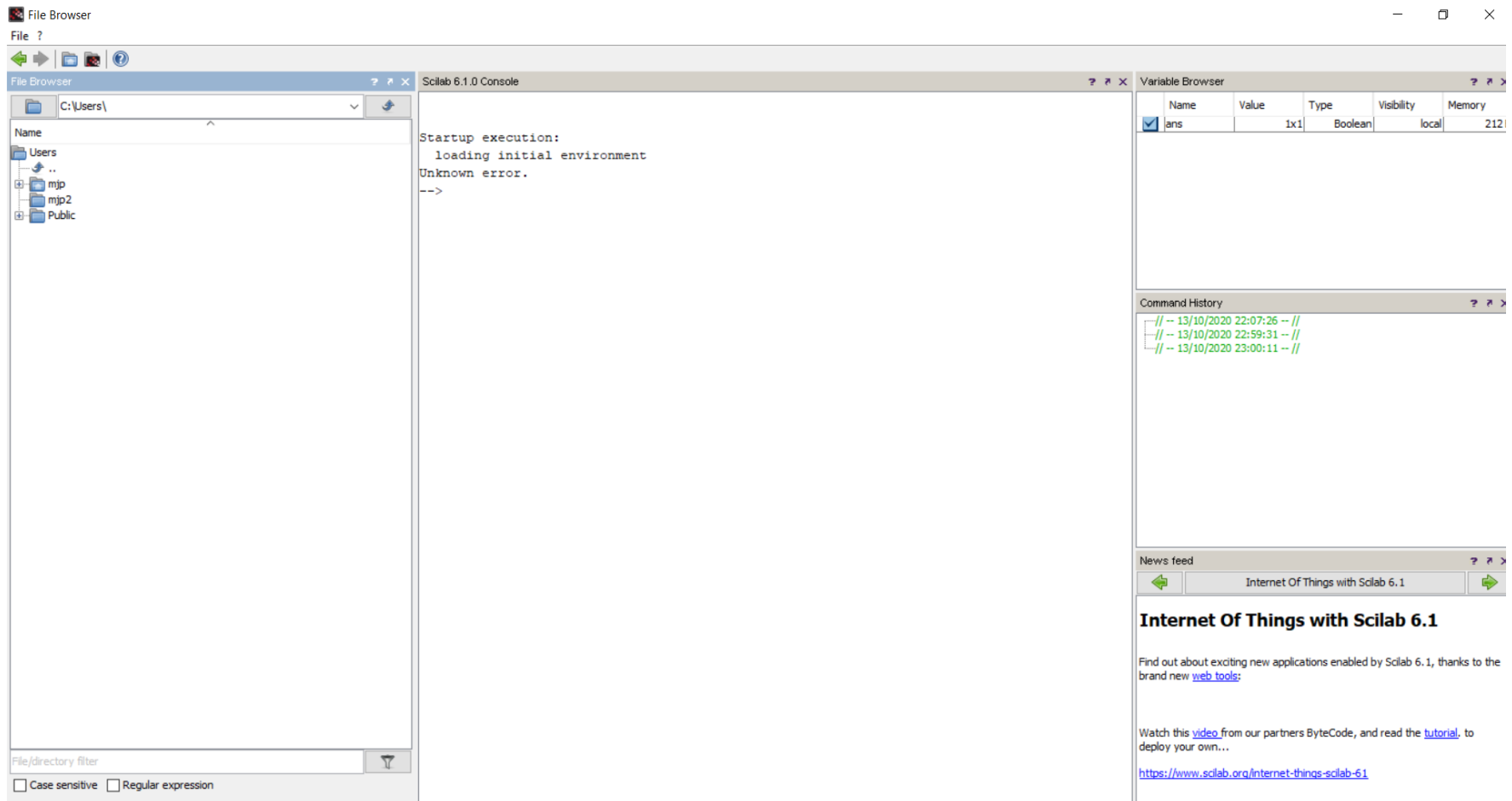
- Após 5 anos de versões, a **Scilab Enterprises** junta-se ao grupo **ESI** em 2017, mantendo o software livre e grátis (última versão 6.1.1): <https://www.scilab.org/>

A screenshot of the Scilab website homepage. The header features the Scilab logo on the left, a search bar with a magnifying glass icon on the right, and a navigation menu with links: Download, Software, Tutorials, Applications, Services, Cloud, and About. Below the header is a blue horizontal bar. The main content area is divided into two columns. The left column, titled "Software versions", lists links for Scilab 6.1.0, Scilab 6.0.2, Scilab 5.5.2, Scilab, Previous versions, Latest version, System Requirements, and To donate. The right column contains text about the release date (Tue, 25 Feb 2020), system requirements, and a change log link. It also lists download links for Windows Vista, 7, 8, 10 (64 bits and 32 bits), GNU/Linux (64 bits and 32 bits), and Mac OS X (64 bits). A note at the bottom states that the version is based on further developments of Scilab 6.1 built on 2020/04/20.

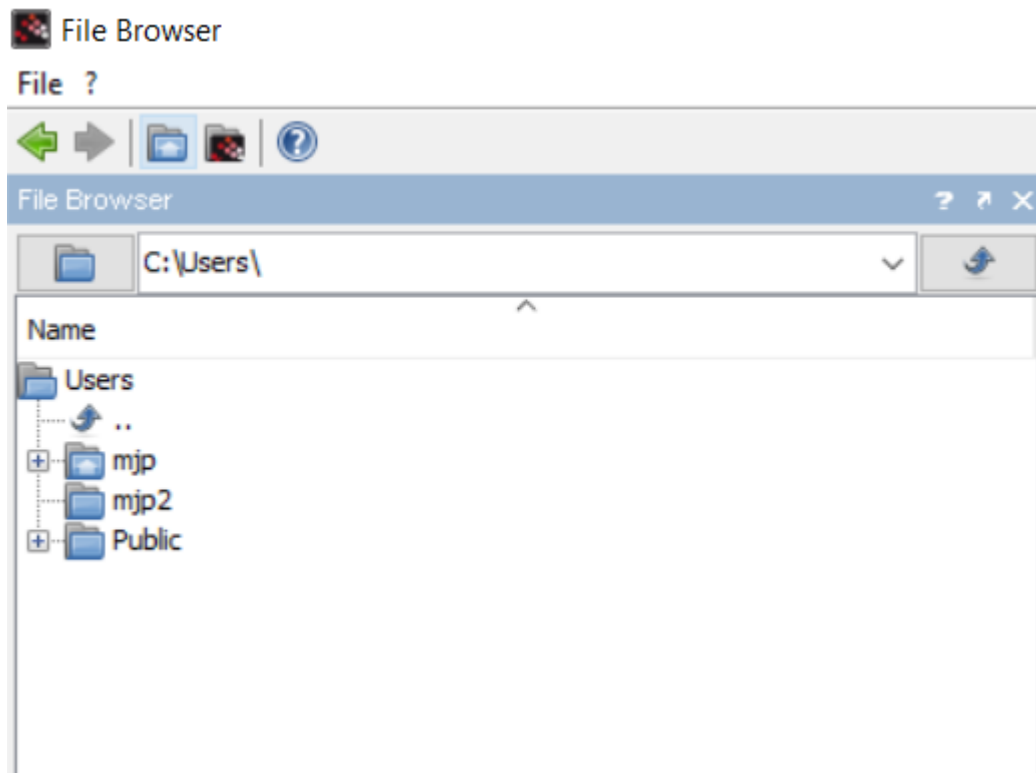
Ver: <https://www.scilab.org/tutorials>



- **As principais janelas de interface com o utilizador** (Estes slides versão Scilab 6.1.0)



- As principais janelas de interface com o utilizador (à esquerda):



■ As principais janelas de interface com o utilizador (à direita):

Variable Browser

	Name	Value	Type	Visibility	Memory
<input checked="" type="checkbox"/>	ans		1x1 Boolean	local	212 B

Command History

```
// -- 13/10/2020 22:07:26 -- //
// -- 13/10/2020 22:59:31 -- //
// -- 13/10/2020 23:00:11 -- //
```

News feed

Web tools in Scilab 6.1

Web tools in Scilab 6.1

Leverage the new functions for server communication, based on the well known web protocol HTTPS:
[GET](#) to simply get data from remote location [POST](#) to exchange data with the server through the request

It enables seamless exchanges with remote servers, by connecting to their API (i.e....

<https://www.scilab.org/web-tools-scilab-61>



■ As principais janelas de interface com o utilizador (no centro)

➤ **Console**

- Ao iniciar uma sessão no Scilab trabalhamos na janela **Console**. Esta janela exibe um cursor, designado de **prompt** (-->), que assinala onde são escritas as instruções (comandos), dados, etc., pelo utilizador. Os comandos são executados após pressionar a tecla **Enter** (↵) .

A screenshot of the Scilab 6.1.0 Console window. The window has a title bar that says "Scilab 6.1.0 Console" and standard window control buttons (minimize, maximize, close) on the right. The main area of the window is white and contains the following text: "Startup execution:", "loading initial environment", "Unknown error.", and "-->". The "-->" is the prompt character, indicating where the user can enter commands.

```
Scilab 6.1.0 Console

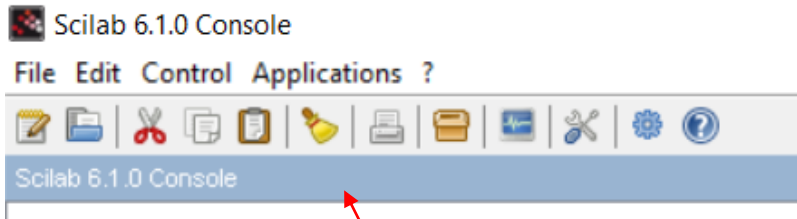
Startup execution:
  loading initial environment
Unknown error.
-->
```



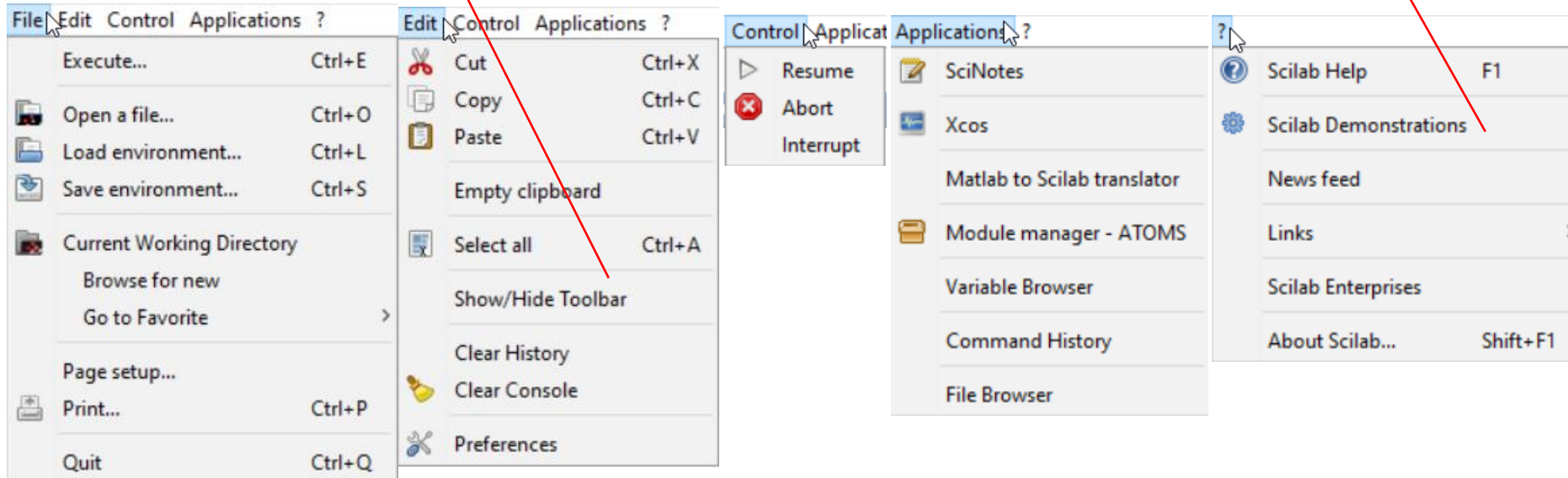
➤ Console

- Detalhes da janela **Console**

Menus



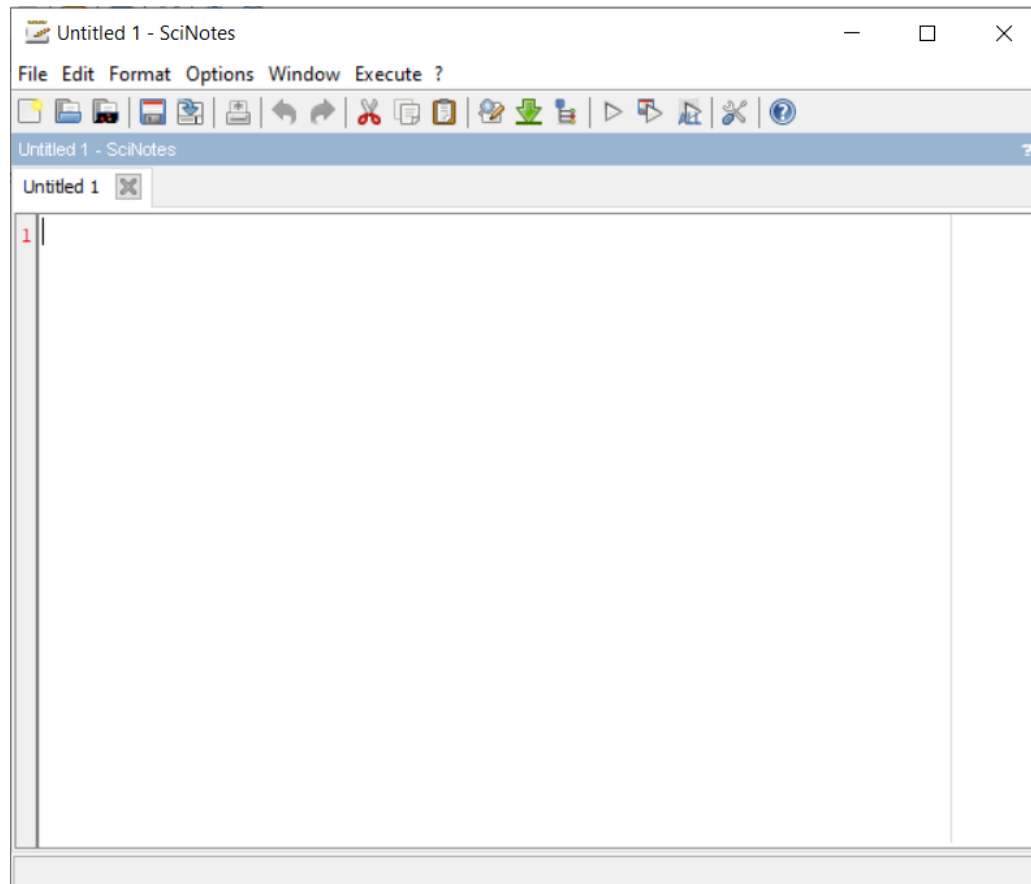
Scilab 6.1.0 Console





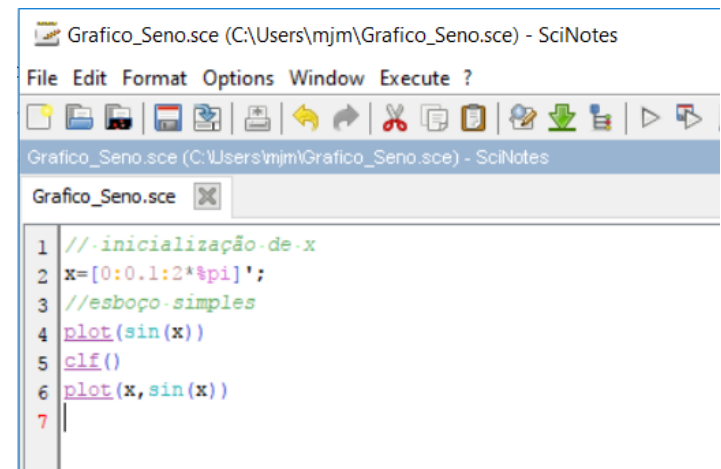
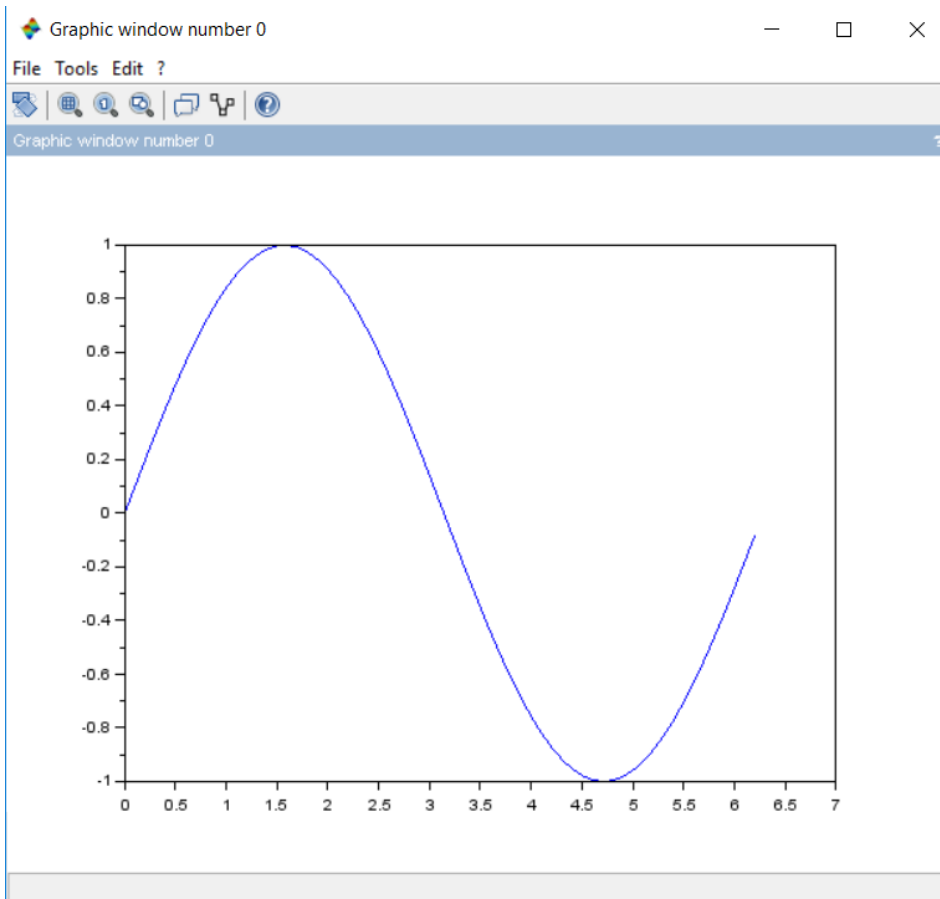
➤ SciNotes

- A janela **SciNotes** é um processador de texto que facilita a programação.



➤ Gráfica

- A janela **Gráfica** abre quando é executada a instrução de um gráfico.



```
1 //inicialização-de-x
2 x=[0:0.1:2*pi]';
3 //esboço-simples
4 plot(sin(x))
5 clf()
6 plot(x, sin(x))
7
```



■ Interação com a *Console*

➤ Em geral, a interação com a *Console*, isto é, a forma geral do(s) comando(s) é:

--> nome=instrução (função do *Scilab*)

neste caso, a(s) operação(ões) contida(s) no termo instrução são realizadas e o objeto resultante é atribuído à variável nome.

-->instrução

neste caso, a(s) operação(ões) contida(s) no termo instrução são realizadas e o resultado é atribuído à variável *ans*.



■ Interação com a *Console*

➤ Observações

- Uma linha de comando que termina com **ponto e vírgula**, “;”, significa que o *Scilab* não mostra o resultado na *Console*;
- **Dois pontos seguidos**, “..”, permitem a continuação do comando na linha seguinte (após pressionar a tecla “*Enter*”);
- **Duas barras**, “//”, iniciam um comentário ou indicam que um comando não será executado.

■ Na linha de comandos

➤ Alguns Comandos Básicos

- Para se **mover o cursor** na linha de comandos usar as **teclas com setas**, como num processador de texto;



- As teclas **Home** e **End** levam ao princípio e fim da linha, respetivamente;
- Se colocarmos o cursor no **prompt** e pressionarmos sucessivamente a tecla ↑ vamos chamando sucessivas linhas de comando, a começar na última introduzida;
- Uma linha ou qualquer região da janela de comandos pode ser seleccionada usando o rato, copiada e colada no **prompt** ativo ou noutro documento.



■ Na linha de comandos

➤ Variáveis especiais

Existem algumas variáveis pré-definidas no *Scilab*. Essas variáveis iniciam-se com o carater “%”. Alguns exemplos:

- %i – o número complexo (ou imaginário) $i = \sqrt{-1}$ (ou $i^2 = -1$);
- %e – o número de Neper, $e=2.7182818$;
- %pi – o número pi, $\pi=3.1415927$;
- %inf – infinito, ∞ .

Por defeito, os números são apresentados com, no máximo, 9 posições, sendo a parte inteira separada da parte decimal por um **ponto final**;

Para aumentar o número posições, no máximo até 25, usa-se a função *format(n);número*.



■ Na linha de comandos

➤ Regras para os nomes das variáveis

O nome das variáveis, ou objetos, devem seguir as seguintes regras:

- Começar com um letra seguida de letras, algarismos ou o símbolo *underscore*;
- Carateres especiais não são permitidos, tais como: #, &, %, ?, !, @, :, ... etc;
- Carateres acentuados não são permitidos;
- Os nomes das variáveis são sensíveis a letras maiúsculas e minúsculas.



■ Na linha de comandos

➤ Operadores aritméticos e relacionais

- + Soma
- - Subtração
- * Multiplicação
- / Divisão
- ^ Potenciação
- < menor que
- <= menor ou igual que
- > maior que
- >= maior ou igual que
- <> ou ~= diferente de

■ Na linha de comandos

➤ Funções Matemáticas, função de x

- $\log(x)$ logaritmo natural ou de base e de x
- $\log_{10}(x)$ logaritmo de base 10 de x
- $\log_2(x)$ logaritmo de base 2 de x
- $\sin(x)$ seno de x
- $\cos(x)$ cosseno de x
- $\tan(x)$ tangente de x
- $\text{asin}(x)$ arcosseno de x
- $\text{acos}(x)$ arcocosseno de x
- $\exp(x)$ exponencial de x
- $\text{sqrt}(x)$ raiz quadrada de x
- rand número aleatório entre zero e um
- $\text{abs}(x)$ valor absoluto ou módulo de x
- $\text{round}(x)$ arredondamento para o inteiro mais próximo
- $\text{floor}(x)$ arredondamento para o inteiro anterior
- $\text{ceil}(x)$ arredondamento para o inteiro seguinte
- $\text{int}(x)$ extrai a parte inteira de um número x
- $\text{conj}(a+b*\%i)$ conjugado de um número complexo $a+bi$



■ Manipulação e informação sobre o espaço de trabalho

➤ Quando trabalhamos no *Scilab*, o software guarda numa área de memória chamada “*workspace*” (espaço de trabalho) informação sobre as variáveis e funções existentes. Para eliminar ou apagar informação que existe no espaço de trabalho, o *Scilab* disponibiliza os seguintes comandos:

- `clc()` limpa a janela de comandos;
- `clc(n)` limpa n linhas acima da atual linha de comandos;
- `clear()` remove todas as variáveis da memória;
- `clear x` remove apenas a variável x da memória.



■ Guardar e imprimir todo o trabalho na janela de comandos

- O ambiente de trabalho pode ser guardado numa *pen*, num ficheiro com o comando **save("nomeficheiro.sav")**, ou clicar no menu **File→Save environment**;
- Para carregar o ficheiro, clicar no menu **File→Load environment**;
- Para imprimir toda a sessão de trabalho presente na janela de comandos do *Scilab*, clicar no ícone que representa uma impressora ou utilizando no menu **File→Print**.

Nota: O *Scilab* trabalha, por defeito, num determinado diretório atual do computador, para visualizar qual é o diretório clicar no menu **File→Current Working Directory**. Se desejar alterar o diretório clicar no menu **File→Browse for new**.



■ Matrices

➤ O *Scilab* é baseado fundamentalmente na manipulação de vetores e matrizes.

Vamos, por isso, abordar os seguintes pontos:

- Criar matrizes;
- Conhecer as matrizes especiais que o *Scilab* cria;
- Identificar elementos de um vetor ou de uma matriz;
- Realizar operações com vetores e matrizes;
- Manipular matrizes, como acrescentar e eliminar elementos/linhas/colunas.



■ Matrizes

- Para definir uma matriz é necessário identificar o seu número de linhas (m) e o seu número de colunas (n) assim como o tipo de dados que a compõe;
- No *Scilab* podemos definir matrizes cujos dados são números complexos, reais, inteiros, booleanos, *strings* ou polinómios;
- Um vetor é um caso particular de uma matriz, onde o número de linhas (colunas) é igual a 1;
- Um escalar também é interpretado como uma matriz com apenas 1 linha e 1 coluna.

Nota (vetores particulares):

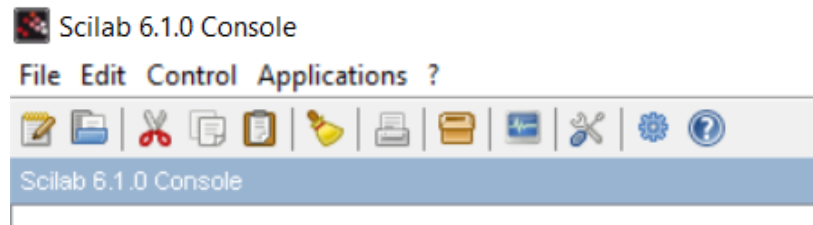
- $a:b$ sequência de números entre a e b ($b > a$), espaçados de uma unidade;
- $a:c:b$ sequência de números entre a e b , espaçados de c unidades;



■ Matrices

➤ Definição

- Os parêntesis retos indicam que se está a definir uma matriz;
- A vírgula ou o espaço é utilizado para separar os elementos nas diferentes colunas;
- O ponto e vírgula ou quebra de linha é utilizado para separar elementos nas diferentes linhas.
- Ou, quando possível, utilizando a função **matrix(elementos,nºlin,nºcol)'**



Startup execution:

loading initial environment

```
--> A=[1 2 3;4 5 6;7 8 9]
```

A =

```
1.  2.  3.
4.  5.  6.
7.  8.  9.
```

```
--> A=[1,2,3;4,5,6;7,8,9]
```

A =

```
1.  2.  3.
4.  5.  6.
7.  8.  9.
```

```
--> A=[1 2 3;
```

```
> 4 5 6;
```

```
> 7 8 9]
```

A =

```
1.  2.  3.
4.  5.  6.
7.  8.  9.
```

```
--> A=matrix(1:9,3,3)'
```

A =

```
1.  2.  3.
4.  5.  6.
7.  8.  9.
```

```
--> A=matrix(1:9,3,3)
```

A =

```
1.  4.  7.
2.  5.  8.
3.  6.  9.
```



■ Matrizes

- Definição de matrizes especiais do *Scilab*
Sintaxe geral: `nome(nºlin,nºcol)`

Estas funções criam as seguintes matrizes especiais:

- `eye(n,n)` matriz Identidade de ordem n
- `k*eye(n,n)` matriz Escalar k
- `eye(m,n)` matriz com uma Submatriz Identidade $n \times n$ ($n < m$) ou $m \times m$ ($n > m$)
- `zeros(m,n)` matriz Nula $m \times n$
- `rand(m,n)` matriz Números Aleatórios (entre 0 e 1) $m \times n$
- `diag(a)` matriz Diagonal com os elementos do vetor a na sua diagonal principal, (por exemplo, $a=1:4$)

Nota:

- `size(A)` dá a dimensão da matriz A (número de linhas e número de colunas)
- `diag(A)` extrai os elementos da diagonal principal da matriz A
- `conj(A)` calcula a matriz conjugada da matriz de números complexos A
- `conj(A')` calcula matriz transconjugada da matriz de números complexos A
- `A'` calcula a transposta da matriz A

■ Matrices

➤ Operações com os vetores a e b da mesma dimensão

- $a+b$ Adição
- $a-b$ Subtração
- $a.*b$ ou $a*b$. Multiplicação elemento a elemento
- $a./b$ ou a/b . Divisão elemento a elemento
- $a.^b$ ou $b.^a$ Potência elemento a elemento
- $a*b'=b*a'=k$ Produto escalar de dois vetores
- $k*a$ Produto de um escalar por um vetor

■ Matrices

➤ Operações com duas matrizes A e B da mesma dimensão

- $A+B$ Adição
- $A-B$ Subtração
- $A*B$ Multiplicação (se e só se, A e B forem matrizes quadradas)

➤ Se a Matriz A tiver um número de colunas igual ao número de linhas da matriz B

- $A*B$ Multiplicação ($A_{mn} * B_{np} = C_{mp}$)

➤ Muitos dos comandos que se aplicam a escalares podem ser aplicados a matrizes:

- $\log(A)$
- \sqrt{A}
- $\exp(A)$
- etc.

■ Matrizes

➤ Analisar matrizes

- `det(A)` calcula o determinante da matriz A
- `inv(A)` calcula a matriz Inversa da matriz A: A^{-1}
- `trace(A)` calcula o traço da matriz A
- `rank(A)` calcula a característica da matriz A
- `triu(A)` extrai a matriz triangular superior da matriz A
- `tril(A)` extrai a matriz triangular inferior da matriz A

As instruções para obter os vetores e valores próprios da matriz A são

```
-->A=.... // definir a matriz A
-->[Ab,X,bs]=bdiag(A,1/%eps) // função bdiag()
-->X // dá os vetores próprios da matriz A
-->Ab // dá os valores próprios da matriz A
-->poly(A,"x") // dá o polinómio característico da matriz A
```

Nota: As funções do *Scilab* para o cálculo, de outros capítulo da UC de ALGA, vão ser apresentados nos slides das respetivas aulas teórico-práticas.

■ **Matrizes** (outras situações particulares)

- Identificar elementos de um vetor ou de uma matriz

Sintaxe geral: nome do vetor ou matriz(nº da lin,nº da col)

- $a(j)$ ou $A(i,j)$ extração do elemento da linha i e coluna j

- Extrair os todos/alguns elementos de uma linha(s)/coluna(s) de uma matriz

Sintaxe geral: nome da matriz(nº da lin,primeira coluna:última coluna)

nome da matriz(nº da col,primeira linha:última linha)

- $A(i,:)$ extração de toda a linha i da matriz A
- $A(:,j)$ extração de toda a coluna j da matriz A
- $A(i,j*:j**)$ extração da linha i da matriz A , da coluna j^* à coluna j^{**}
- $A(i*:i**,j)$ extração da coluna j da matriz A , da linha i^* à linha i^{**}
- $A(i*:i**,j*:j**)$ extração das linhas i e colunas j da matriz A

■ **Matrizes** (outras situações particulares)

➤ Manipulação de vetores e matrizes

- Acrescentar um elemento k a um vetor linha existente (concatenação)

a - vetor linha a
 $[a \ k]$ ou $[k \ a]$ - novo vetor

- Eliminar o elemento j a um vetor linha existente

$a(j)=[]$

- Acrescentar um elemento k a um vetor coluna existente (concatenação)

$b=a'$ - vetor coluna b
 $[b;k]$ ou $[k;b]$ - novo vetor

- Eliminar o elemento j a um vetor coluna existente

$b(i)=[]$

■ **Matrizes** (outras situações particulares)

➤ Manipulação de vetores e matrizes

- Acrescentar uma linha a uma matriz existente

a - vetor linha a
 $[A;a]$ ou $[a;A]$ - nova matriz

- Acrescentar uma coluna a uma matriz existente

$b=a'$ - vetor coluna b
 $[A,b]$ ou $[b,A]$ - novo vetor

- Eliminar a linha i ou as linhas i^* a i^{**} , da matriz A existente

$A(i,:)=[]$ ou $A([i^* i^{**}],:)=[]$

- Eliminar a coluna j ou as colunas j^* a j^{**} , da matriz A existente

$A(:,j)=[]$ ou $A(:,j^*:j^{**})=[]$