

Scilab

Curso de Ciência da Computação
Campus Kobrasol



Prof. Denise Prado Kronbauer

denise.kronbauer@univali.br

denipk@gmail.com

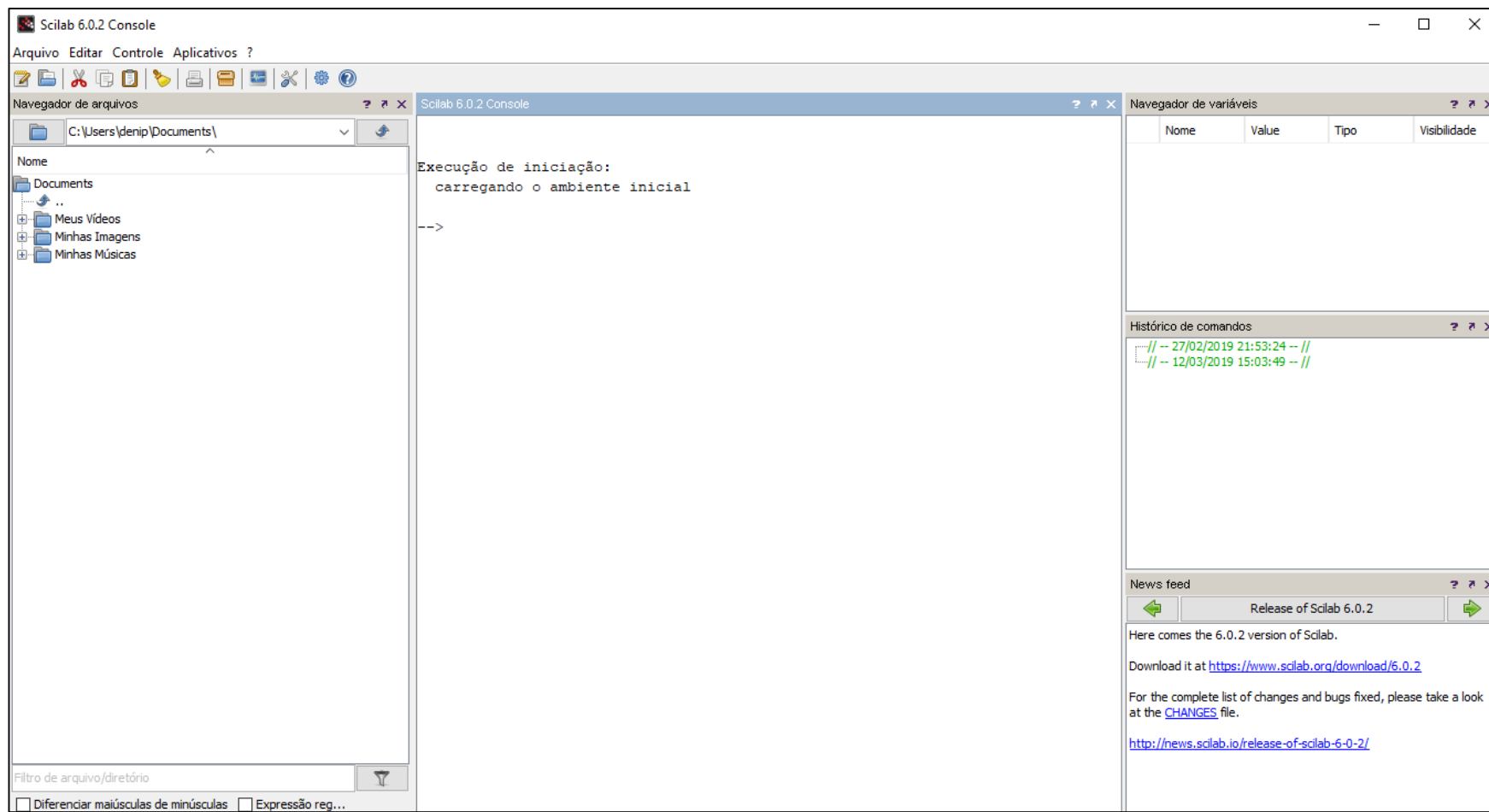


O software Scilab, que é um ambiente numérico, gráfico e programável, desenvolvido pelo Institut Nationale de Recherche en Informatique et en Automatique (INRIA), da França; seu uso é livre e está disponível para plataformas Linux, Windows e Mac.

Se você deseja instalar o Scilab em seu computador pessoal, basta acessar a página <http://www.scilab.org/index.php/download/> e selecionar o programa de instalação adequado a seu sistema operacional.



Janela aberta ao inicializar o Scilab:





Na janela central, chamada Console, ou workspace, é que todos os comandos são executados e os resultados são visualizados.

De tempos em tempos, é recomendável limpar o conteúdo da Console, através do comando *clc*.

Quando pedimos gráficos, eles serão exibidos em janelas separadas. Quando quisermos fechar as janelas, podemos digitar o comando *close*.



Nome das variáveis

Na composição dos nomes das variáveis são permitidos os caracteres alfanuméricos (os caracteres %,_,#, !,\$ e ? Também são permitidos, mas o caractere % não deve iniciar o nome). O Scilab distingue caracteres maiúsculos e minúsculos, de modo que var1, Var1, VAR1, por exemplo, são variáveis distintas.



Variáveis pré-definidas

No Scilab, diversas variáveis são pré-definidas e seus nomes começam com o caractere % e, por esse motivo, os nomes de variáveis do usuário não podem começar com %. Alguns exemplos são:

<code>%pi</code>	π
<code>%i</code>	Número imaginário, i
<code>%e</code>	Constante de Euler



Atribuição de valores

O Scilab é uma linguagem interpretada e, por isso, não é necessário declarar a variável antes de usá-la (como é o caso de linguagens compiladas como o C).

A variável é criada no momento em que lhe atribuímos um valor. Assim, no exemplo abaixo, criamos a variável real x e lhe atribuímos o valor 1.

Execute:

-->x=1 (ele retornará um 'espelho')

-->x=1; (o ponto e vírgula faz omitir a repetição)



Comentários e continuação de linhas

Qualquer linha iniciada com duas barras “//” é considerada pelo Scilab como um comentário e não como um comando a ser executado e, portanto, o conteúdo dessa linha é ignorado. No Scilab, entretanto, não é possível comentar um bloco inteiro de linhas, isto é, não existe, por exemplo, o recurso da linguagem C de colocar um bloco de comentários entre “/* ... */”.

Execute:

→//Resolução de sistemas pelo método de Gauss-Seidel

→a=[2 4 6 8];



Comentários e continuação de linhas

Quando um comando é longo demais para caber em uma única linha ele pode ser subdividido em duas ou mais linhas. Sempre que uma linha terminar com dois pontos “..”, o Scilab entende que a próxima linha é uma continuação da anterior e não um novo comando.

Execute:

```
-->x=1 ..  
  > +2 ..  
  > +3 ..  
  > +4
```



Operações aritméticas

O Scilab oferece um elenco de operações aritméticas elementares como qualquer calculadora. São elas:

+	Adição
—	Subtração
*	Multiplicação
/	Divisão pela direita (x/y ou xy^{-1})
\	Divisão pela esquerda ($x\backslash y$ ou $x^{-1}y$)
^	x^y Potenciação (x^y)
**	Potenciação (o mesmo que ^)



Operações aritméticas

Execute:

```
—→((4+16)/4)^2
```

```
—→a=[2 4 6 2];
```

```
—→s=sum(a);
```

```
—→size(a)
```



Funções matemáticas elementares

O Scilab oferece um grande número de funções matemáticas básicas como funções trigonométricas, exponenciais, logarítmicas, etc, como qualquer calculadora científica. Nos quadros abaixo, é apresentada uma lista (incompleta) das funções matemáticas disponíveis:

sin cos tan exp log max min sqrt mean

Execute:

```
--> sin(%pi/2)
--> sqrt(81)
--> mean([2 5 7 12])
```



Cadeia de caracteres (“strings”)

Strings podem ser armazenadas em variáveis, desde que Delimitadas por aspas duplas (“). Duas strings podem ser concatenadas usando o operador +.

Execute:

```
--> ab=“cálculo”;  
--> cd=“ numérico”;  
--> ad=ab+cd
```



Matrizes e vetores

No Scilab, o objeto básico é a Matriz, um objeto matemático definido por seu número de linhas, número de colunas e tipo do dado armazenado em cada elemento da matriz, que pode ser um número inteiro ou real, uma variável lógica, uma cadeia de caracteres, etc.

Para o Scilab, todos os objetos são matrizes. Assim, vetores são matrizes de dimensão $(n \times 1)$ (vetor coluna) ou $(1 \times n)$ (vetor linha) e números usuais são matrizes de dimensão (1×1) .



Matrizes e vetores

O Scilab possui uma sintaxe simples para a criação de vetores e matrizes.

Os seguintes símbolos são empregados na definição de uma matriz:

- Colchetes `[]` delimitam o início e o fim da matriz;
- Vírgulas `,` separam os valores em diferentes colunas;
- Pontos e vírgulas `;` separam os valores de diferentes linhas

Execute:

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

--> `B=[1, 2; 3, 4; 5, 6]`

--> `V1=[1:0.2:2]'`

--> `V2=[1; 2; 3; 4]`



Matrizes especiais

Para alguns tipos de matrizes usadas frequentemente o Scilab oferece comandos específicos para sua criação. Nestes casos, é preciso fornecer apenas as dimensões da matriz

Comando	Resultado
<code>eye(n,m)</code>	Gera a matriz identidade
<code>zeros(n,m)</code>	Inicializa todos os elementos com 0
<code>ones(n,m)</code>	Inicializa todos os elementos com 1

Execute:

```
--> A=zeros(2,5)  
--> B=ones(3,1)
```




Operações com vetores e matrizes: adição e subtração

A adição (ou subtração) de duas matrizes é, inerentemente, uma operação elemento por elemento (*se* $C = A + B$, $C_{ij} = A_{ij} + B_{ij}$);
por isso, o operador “.+” sequer é definido.

Execute:

```
--> A=[2 4 -1 5]
```

```
--> B=[2 -1 3 0]
```

```
--> C=A+B
```



Operações com vetores e matrizes: multiplicação

A multiplicação usual entre duas matrizes, representada aqui pelo operador “*” é aquela em que os elementos da linha i são multiplicados pelos elementos da coluna j e somados, por sua vez, o operador “.*” é executado elemento por elemento de forma que se $C = A.*B$. Execute os comandos abaixo e compare os resultados.

Execute:

```
--> A=[1 2;3 4]
```

```
--> B=[1 2;3 4]
```

```
--> C=A*B
```

```
--> C=A.*B
```



Operações com vetores e matrizes: divisão

A divisão será abordada no contexto da resolução de sistema de equações lineares. Vejamos como proceder com a resolução do sistema de equações:

$$\begin{cases} 2x + y = 3 \\ -x - 2y = 0 \end{cases}$$

Execute:

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

```
--> b=[3;0]
```

```
--> X=A\b
```



Gráficos

O Scilab permite a criação de vários tipos de gráficos em 2D e 3D.

A única forma de conhecer todos os recursos gráficos do Scilab é explorá-los, com o auxílio do Help. Para construir o gráfico da função $y = x^2$,

Execute:

```
--> x=[-10:10];
```

```
--> y=x^2;
```

```
--> plot(x,y)
```

Execute:

```
--> x=linspace(-8,8,50);
```

```
--> y=sin(x);
```

```
--> plot(x,y,'*')
```



Gráficos

Ao utilizar o comando `plot(x,y,'tipo_de_linha')`, podemos escolher entre diferentes estilos de linha, conforme a tabela abaixo:

Símbolo	Cor	Símbolo	Estilo de linha
y	amarela	.	ponto
m	lilás	o	círculo
c	turquesa	x	marca x
r	vermelho	+	mais
g	verde	*	asterisco
b	azul	-	linha sólida
w	branco	:	linha pontilhada
k	preto	-.	linha de traço e ponto
		--	linha tracejada



Resolução de sistemas lineares

$$a) \begin{cases} 2x + 3y + z = 2 \\ x - y + z = 3 \\ x - y + 2z = 0 \end{cases}$$

$$c) \begin{cases} 2x - y + z = 2 \\ y + 2z = 3 \\ z = 1 \end{cases}$$

$$b) \begin{cases} x + y + z + t = 1 \\ -x + 2y + z = 2 \\ 2x - y - z - t = -1 \\ x - 3y + z + 2t = 0 \end{cases}$$



Comandos Save e Load

As variáveis são apagadas quando o usuário termina a execução do Scilab.

Para usá-las da próxima vez que executar o Scilab, você deve salvá-las com o comando: *save*("arquivo");

Execute:

```
-->a=2;
```

```
-->b=3;
```

```
-->c=4;
```

```
-->save("dados.dat");
```



As variáveis foram salvas no arquivo dados.dat. O comando *load(arquivo)*; é usado para recuperar variáveis que foram salvas no arquivo. Por exemplo,

Execute:

—→clear

—→load(“arquivo.dat”);

—→a+b



Formato de visualização dos números

O comando *format()* modifica a quantidade de dígitos com que os números são mostrados no Scilab. Por exemplo, o comando

```
→format(5)
```

fará com que todas os números sejam visualizados em 5 posições (incluindo o ponto decimal e um espaço para o sinal). Por exemplo:

```
→sqrt(3)
```

Para aumentar o números de posições para 16, usa-se:

```
→format(16)
```

```
→sqrt(3)
```



O comando *format('e')* mostra os números em notação científica. Por exemplo:

```
-->format('e')  
-->2*%pi/10
```

Para retornar ao formato inicial usa-se,

```
-->format('v') // que é chamado de “formato de variável”.
```

Vejamos outras formas de usar o comando *format*:

```
-->format('v',10) //mostra os números em formato de variável com 10 posições.  
-->format('e',8) //mostra os números em notação científica com 8 posições.
```



Exercícios

Considere as variáveis $A = 11$, $B = 5$, $C = -4$ e $D = 2$.

Calcule as expressões abaixo:

a) $3 * \text{modulo}(A, 3) - C$

b) $2^{(2 * \text{abs}(C))} / 8$

c) $(A/B - \text{fix}(A/B) + 2.8)^{(15/B)}$

d) $\text{sqrt}(\cos(A)^2 + \sin(A)^2) + \sin(D * \%pi/4)$

Pesquisar:

--> help modulo

--> help abs

--> help fix



Input, disp, printf

Dois tipos de função são imprescindíveis para a programação: os comandos de entrada e saída de dados. No Scilab, quando o programador quer que o usuário atribua um valor para uma variável, aquele faz o uso da função *input()*. A variável é atribuída da seguinte forma:

→ `x=input("Expressão");`



Input, disp, printf

O comando de saída tradicional é *disp()*, que retorna o conteúdo entre parênteses. Uma forma mais elegante é feita com o uso *printf*, que combina texto de exibição e variáveis. A seguir, exemplos de aplicação das funções:

- `printf("Alô mundo\n")`
- `printf('O número é: %f',435.7666)`
- `disp('Olá turma')`



Input, disp, printf

O caracter `\n` (chamado de *new line*) avisa ao comando `printf` para gerar uma nova linha. Mais precisamente, `\n` move o cursor para o começo da linha seguinte. Por exemplo, colocando `\n` após o string `Alô` faz com que `printf` gere uma nova linha após `Alô`:

→ `printf("Alô\nmundo")`



Input, disp, printf

A forma geral do comando printf é: `printf(<formato>,<lista de dados>);`

Exemplo:

→ `A=2;`

→ `printf("A variável A contém o valor %g\n",A);`

A símbolo `%g` (chamado de caractere de formatação) indica como cada variável da lista de dados será exibido dentro da string de formatação . Neste último exemplo, `%g` é substituído pelo valor da variável `A` no momento da impressão.



Input, disp, printf

No seguinte exemplo, as variáveis A e B substituirão os caracteres de formatação %g nas posições correspondentes:

```
--> A=8/4;  
--> B=A+3;  
--> printf("Os valores calculados foram %g e %g\n",A,B);
```

Se a variável for do tipo string, usa-se o caractere de formatação

%s em vez de %g. Por exemplo:

```
--> nome="Fabrício";  
--> altura=1.68;  
--> printf("A altura de %s é %g",nome,altura);
```




Comandos de fluxo

Os laços são, de longe, a parte mais importante para se programar com o Scilab. Os laços reconhecidos pelo ambiente são: *for*, *if*, e *while*.

O laço for segue a seguinte lógica: para um determinado índice variando de um valor inicial até um valor final com um determinado incremento, execute os comando sob o laço. Como exemplo, considere o seguinte exemplo, que carrega para uma matriz M os números pares de 1 a 20:



Comandos de fluxo

→ *clear*

→ *close*

→ *clc*

→ *N=20;*

→ *M=[];* // Matriz *M* vai armazenar os dados a seguir

→ *for i=1:2:(N-1)*

M=[M (i+1)];

end

→ *disp(M)*



Comandos de fluxo

O laço if funciona com a seguinte lógica: caso a condição seja verdadeira, execute os comandos. Considere o exemplo, para separar os números pares dos ímpares de 1 a 20:

→ *clear*
→ *close*
→ *clc*

→ *N=20;*
→ *P=[];*
→ *I=[];*

→ *for i=1:N*
 if modulo(i,2)==0
 P=[P i];
 end
 if modulo(i,2)==1
 I=[I i];
 end
end

→ *disp(P)*
→ *disp(I)*



Comandos de fluxo

A esse laço, podem ser agregados dois outros comandos: `elseif` e `if`. Estes dois tornam a programação mais eficiente, elegante e “imune” a erros. O programa anterior, se utilizada a função `else`, ficaria:

```
--> clear
--> close
--> clc

--> N=20;
--> P=[ ];
--> I=[ ];

--> for i=1:N
    if modulo(i,2)==0
        P=[P i];
    else
        I=[I i];
    end
end

--> disp(P)
--> disp(I)
```



Exercícios

- 1) Elabore um programa que calcule e imprima o valor da soma de números inteiros, por exemplo, de 1 a 20.
- 2) Elabore agora um programa que calcule e imprima o valor de S:

$$s = \frac{1}{1} + \frac{3}{2} + \frac{5}{3} + \frac{7}{4} + \dots + \frac{99}{50}$$

(Dica: encontre um padrão entre o numerador e o denominador)



Exercícios

3) Vamos mudar o problema anterior para:

$$s = \frac{1}{1} + \frac{5}{3} + \dots + \frac{97}{49}$$

(Dica: O padrão entre o numerador e o denominador é o mesmo, mas agora o denominador varia de forma diferente).



Exercícios

4) Considere o problema do cálculo do fatorial. Sabemos que o fatorial de um número n ($n!$) é dado por:

$$n! = 1 * 2 * 3 * \dots * (n - 1) * n$$

Embora exista uma função no Scilab que retorna o fatorial de um número (***factorial**(n)*), tente montar um programa que resolve o problema.



Resolução – Exercício 1

→ *clear*

→ *close*

→ *clc*

→ *n=20;*

→ *s=0;*

→ *for i=1:n*

s=s+i;

end

→ *disp(s) //*



Resolução – Exercício 2

*Podemos perceber que o numerador = $2 * denominador - 1$*

→ clear

→ close

→ clc

→ s=0;

→ for d=1:50

*s=s+(2*d-1)/d;*

end

→ disp(s) //



Resolução – Exercício 3

*Podemos perceber que o numerador = $2 * denominador - 1$
e os valores somados são tomados a cada dois*

→ clear

→ close

→ clc

→ s=0;

→ for d=1:2:50

*s=s+(2*d-1)/d;*

end

→ disp(s) //



Resolução – Exercício 4

```
--> clear  
--> close  
--> clc  
  
--> n=input("Entre com um número: ");  
--> fat=1;  
--> for cont=2:n  
    fat=fat*cont;  
end  
  
--> disp(fat)
```

Scilab

Curso de Ciência da Computação
Campus Kobrasol



Prof. Denise Prado Kronbauer

denise.kronbauer@univali.br

denipk@gmail.com



Exercícios

1) Elabore um programa para:

- Ler as nossas 6 notas;
- Calcular as médias M1, M2 e M3;
- Calcular a média final; e
- Informar se o aluno foi aprovado ou reprovado.

Sabendo que:

- As avaliações 1-M1, 1-M2 e 1-M3 têm peso 4,0 e as avaliações 2-M1, 2-M2 e 2-M3 têm peso 6,0.
- A média para aprovação é 6,0.



2) O índice de massa corporal (IMC) é uma medida internacional usada para calcular se a pessoa está no peso ideal. O IMC é determinado pela divisão da massa do indivíduo pelo quadrado de sua altura, onde a massa está em quilogramas e a altura está em metros.

$$IMC = \frac{Massa}{Altura^2}$$

IMC	Classificação
< 18,5	Abaixo do Peso
[18,5; 24,9]	Saudável
[25,0; 29,9]	Sobrepeso
[30,0; 34,9]	Obesidade Grau I
[35,0; 39,9]	Obesidade Grau II (severa)
≥ 40	Obesidade Grau III (mórbida)

O resultado do cálculo do IMC é utilizado para classificar o grau de obesidade de um indivíduo, de acordo com a tabela:

Codifique um programa que leia a massa e a altura de uma pessoa, calcule o IMC dessa pessoa e exiba a classificação conforme a tabela acima.