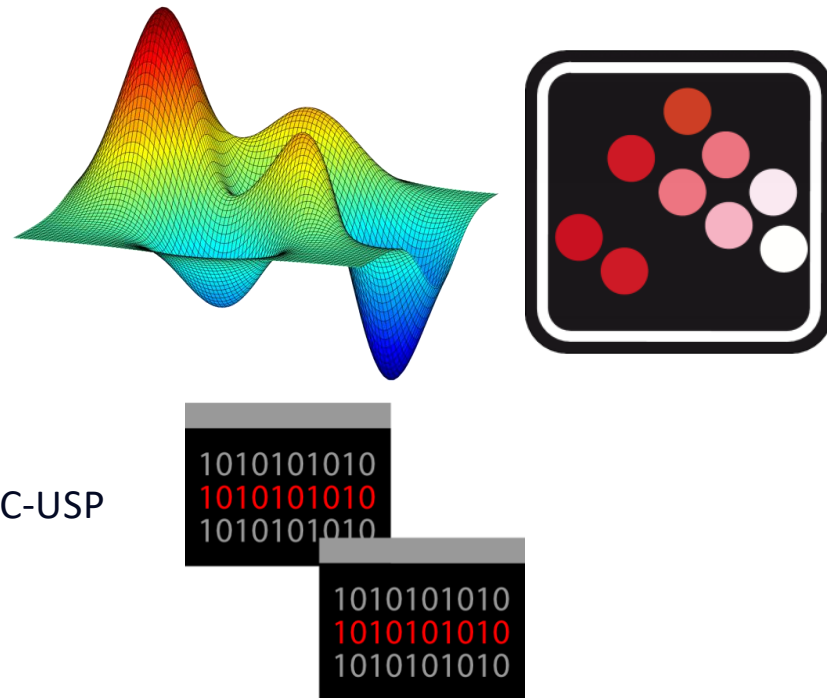


Scilab para *Machine Learning*

Prof. Dr. Diego Bruno

Education Tech Lead na DIO

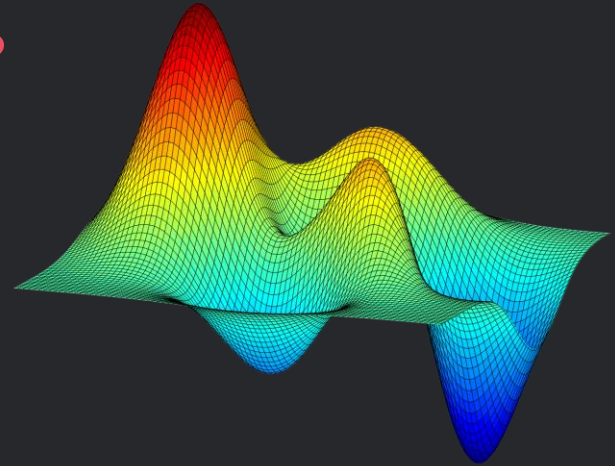
Doutor em Robótica e *Machine Learning* pelo ICMC-USP



Vamos começar a programar...

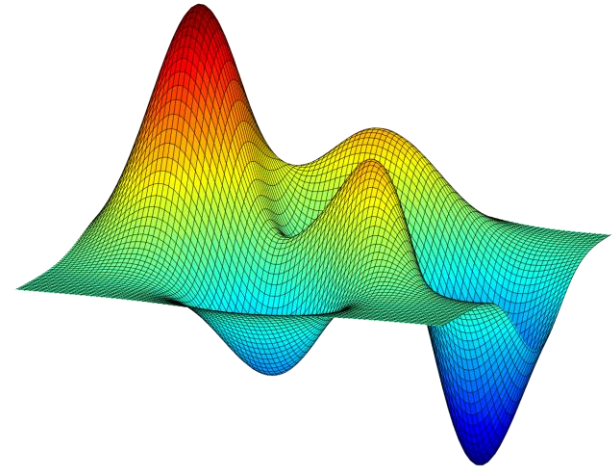
Prof. Dr. Diego Bruno

Machine Learning

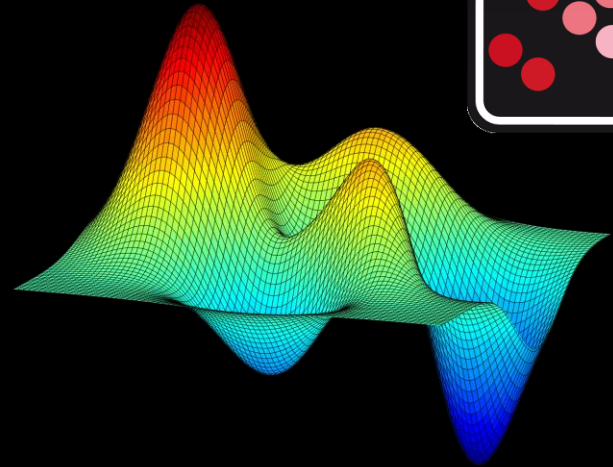


Aplicações do Scilab

O **Scilab** é um software científico para computação numérica semelhante ao MATLAB que fornece um poderoso ambiente computacional aberto para aplicações científicas e de engenharia. Disponível gratuitamente para várias plataformas: Windows, Linux e Mac OS X.

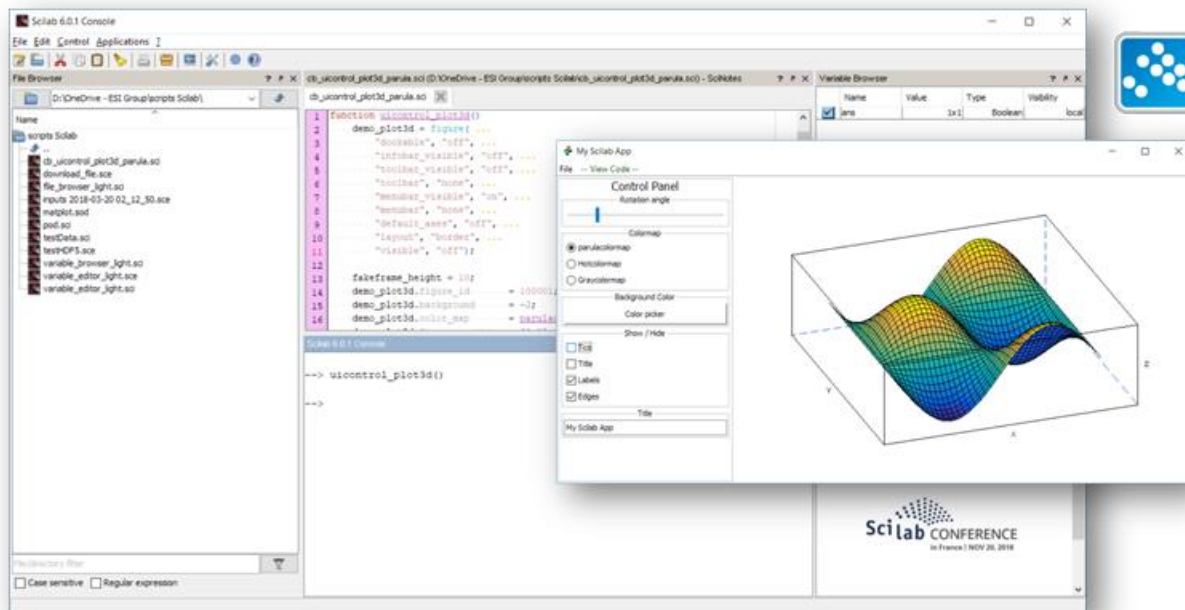


Matemática Básica para Scilab



Programação Orientada a Objetos

Scilab (laboratório de matriz) é um ambiente de computação numérica multi-paradigma.



Multi-paradigma

Funções básicas no Scilab

O sinal de prontidão “-->” indica que o Scilab aguarda a digitação de um comando ou expressão, que deve ser finalizado pela tecla ENTER. Exemplos:

```
-->5+10  
ans =  
    15.  
-->10/4  
ans =  
    2.5
```

Operações Matemáticas

As operações básicas da matemática podem ser realizadas no Scilab por meio dos seguintes operadores:

+	Soma
-	Subtração
*	Multiplicação
/	Divisão
^	Potenciação (x^y)

Exemplo:

```
-->2*6  
ans  =  
    12.
```

Precedência de Operadores

Quando uma expressão envolve diversos operadores, o Scilab considera a ordem de precedência dos mesmos para avaliar a expressão:

Prioridade	Operação
1ª	Potenciação: ^
2ª	Multiplicação e divisão: *, /
3ª	Soma e subtração: +, -

Exemplos:

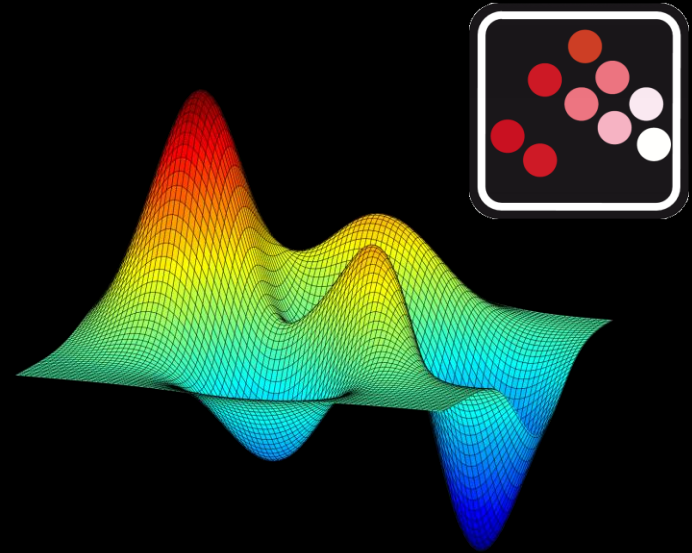
<pre>-->4*3^2 ans = 36.</pre>	Como o operador de potenciação tem maior prioridade em relação ao de multiplicação, a operação 3^2 é avaliada primeiro e o seu resultado (9) é multiplicado por 4.
<pre>-->10+4/2 ans = 12.</pre>	$4/2$ é avaliado primeiro, pois o operador de divisão tem prioridade sobre o de adição. O resultado é então somado com 10.

Expressões com frações

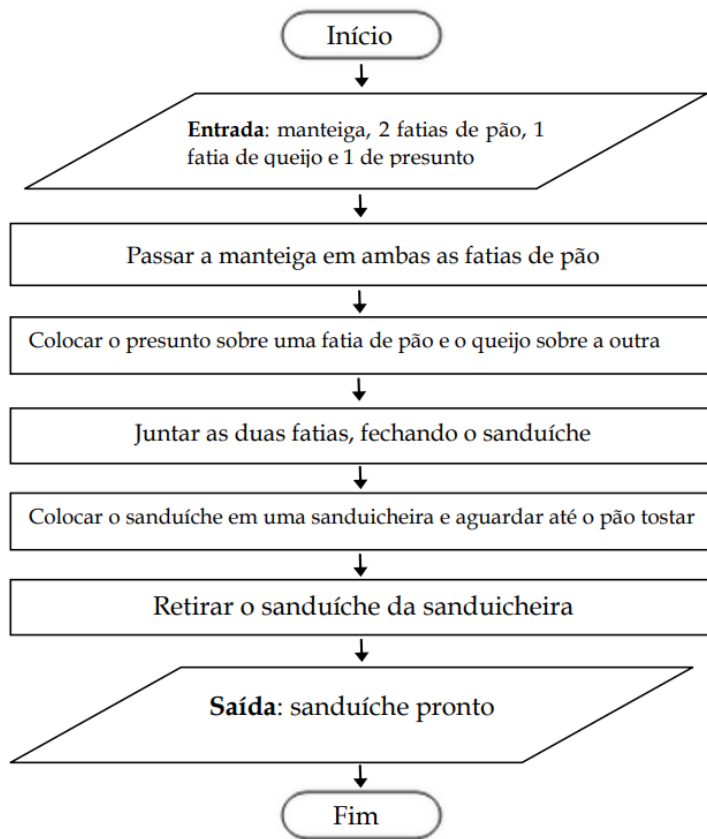
Repare que, se os parênteses não fossem utilizados, a expressão $10+4/2$ no Scilab seria equivalente à expressão matemática, $10 + \frac{4}{2}$,

Expressão Matemática	Expressão correspondente no Scilab
$\frac{1}{2} + \frac{3}{5} - \frac{5}{8}$	-->1/2 + 3/5 - 5/8
$\frac{3 + 2^5}{5}$	--> (3+2^5) / 5
$\frac{5(4 + 2) - 1}{10 + 3^2}$	--> (5* (4+2) -1) / (10+3^2)
$\frac{2^4 + 2^6}{2^5 - 1} + 20$	--> (2^4+2^6) / (2^5-1) +20

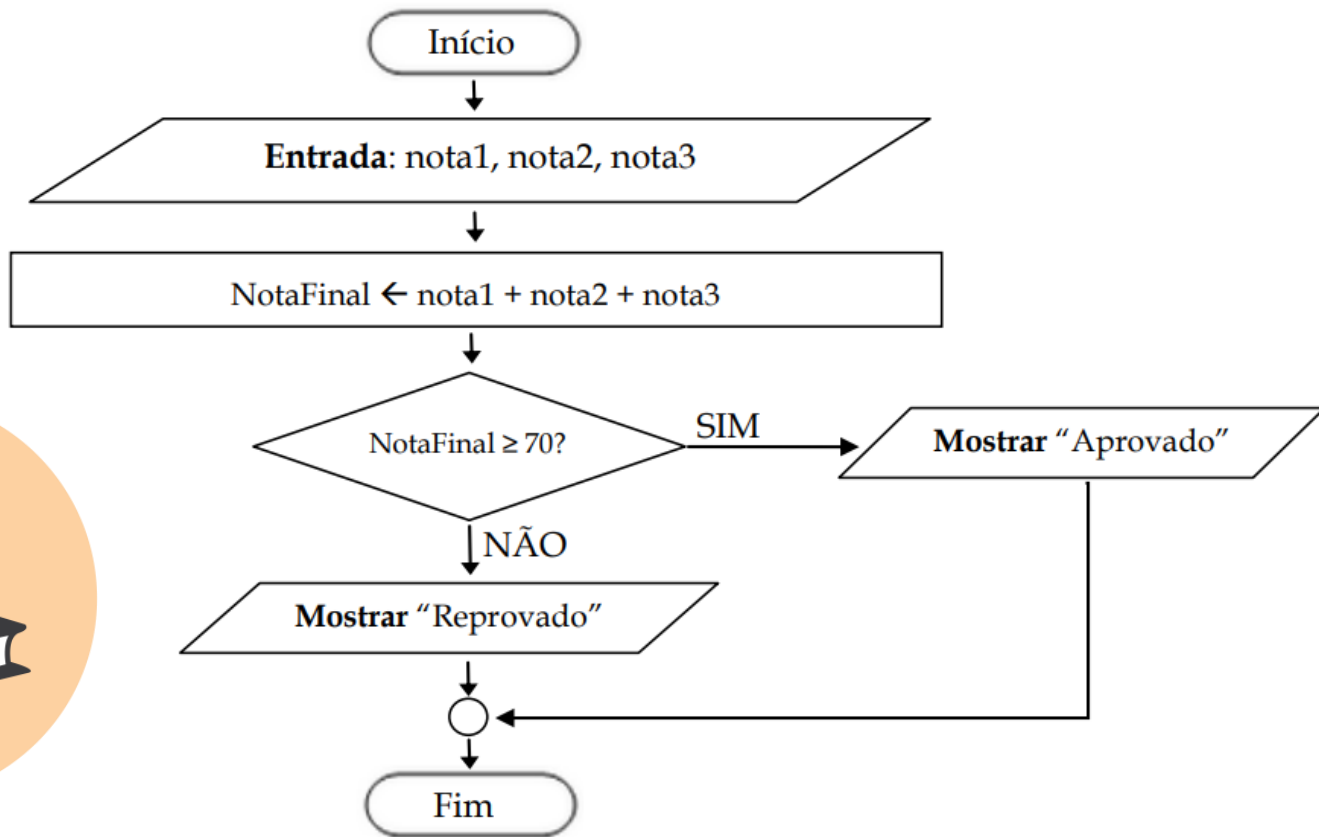
Algoritmos



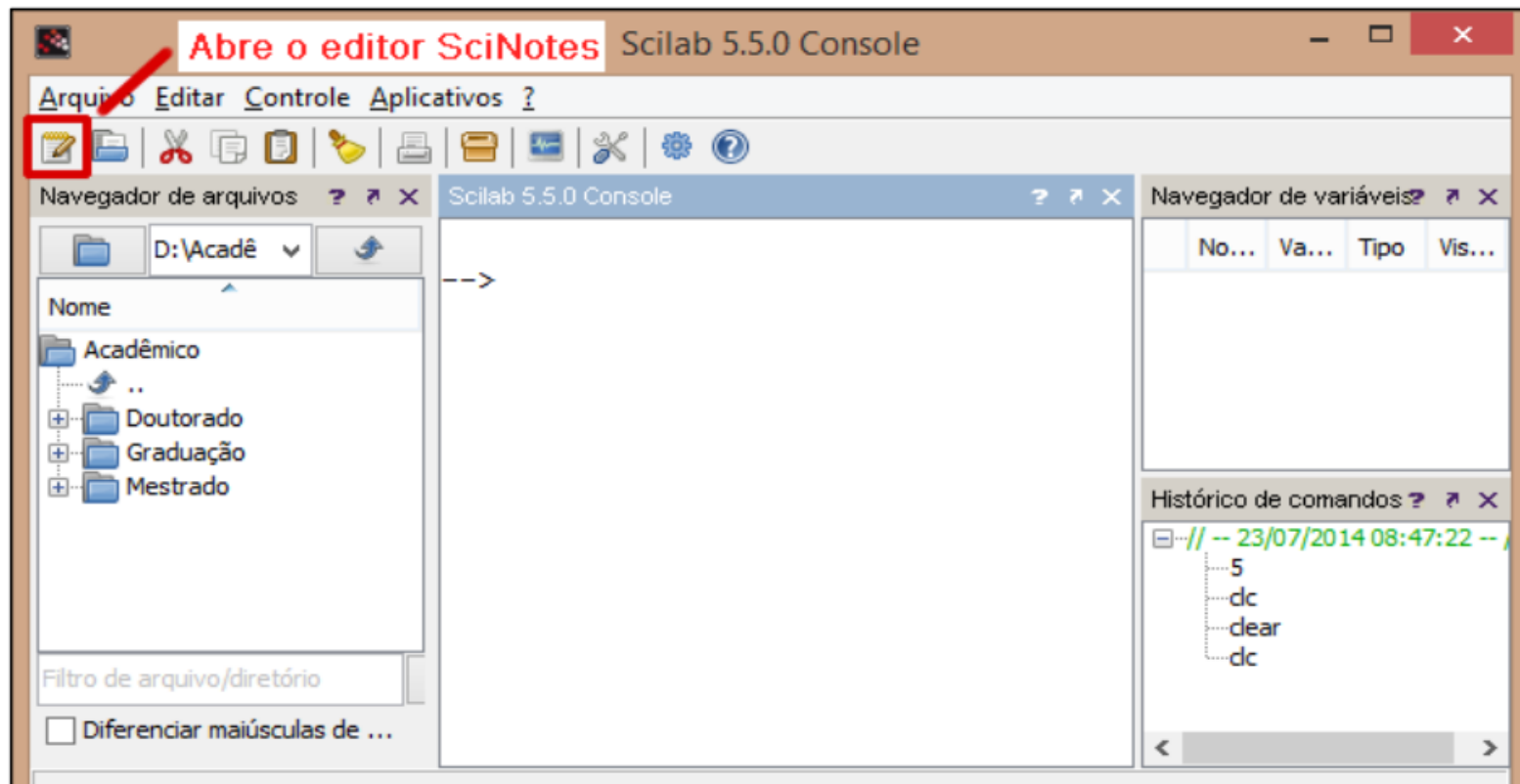
Introdução para Algoritmos



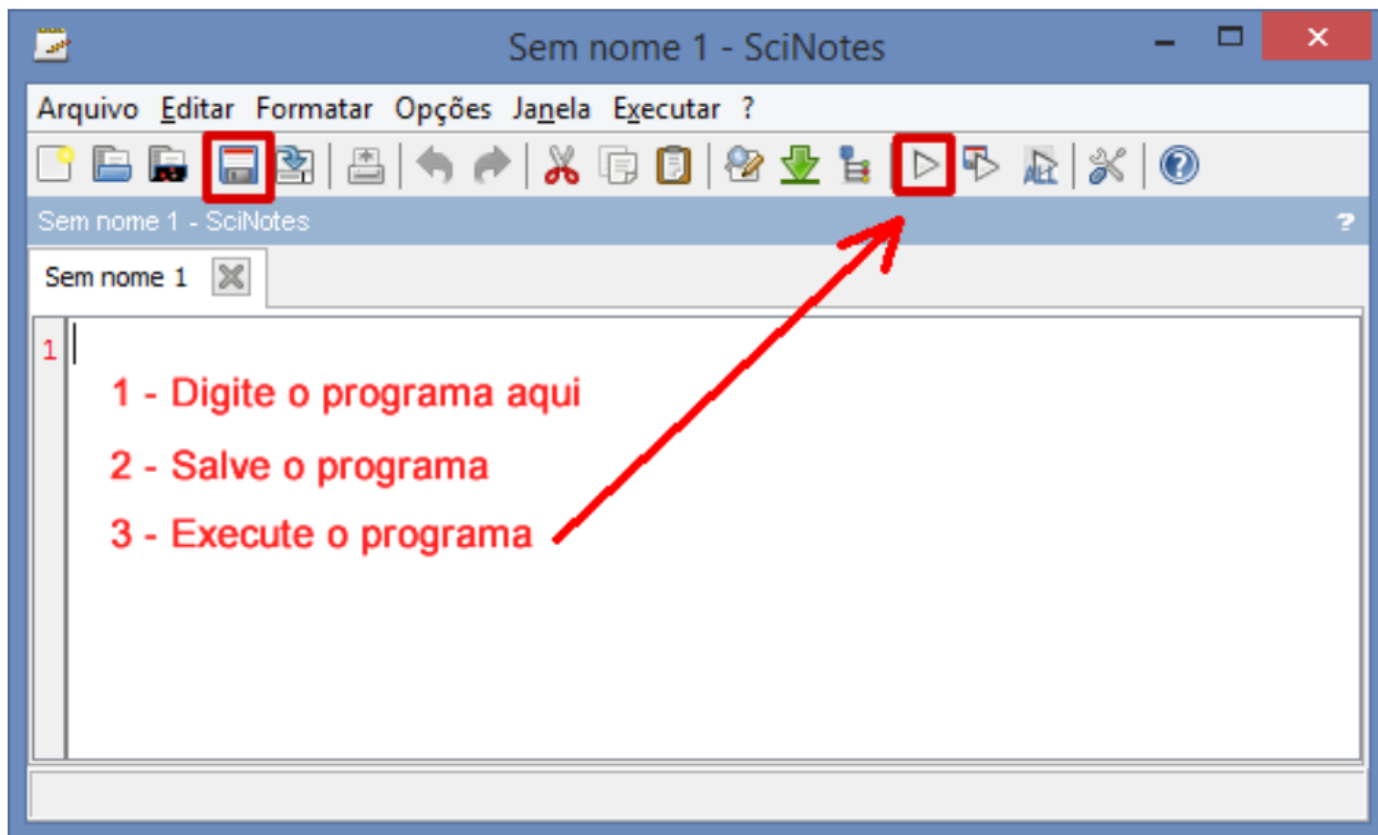
Introdução para Algoritmos



Scinotes



Scinotes



Funções *input* e *disp*

Entrada e saída de dados:

Exemplo para calcular a área de um quadrado.

```
lado = input("Informe a medida do lado de um quadrado: ");  
area = lado * lado;  
disp("A área do quadrado é: ");  
disp(area);
```

Função Printf

A diferença é que, com a função ***printf***, podemos apresentar a mensagem de texto juntamente com o valor da variável.

```
lado = input("Informe a medida do lado de um quadrado: ");  
area = lado * lado;  
printf("A área do quadrado é: %f", area);
```


Estrutura Condicional if-then-else

A estrutura condicional **if-then-else** (**if = se; then = então; else = senão**) é utilizada quando se deseja executar um bloco de operações ou outro, dependendo de uma condição ser verdadeira ou falsa, respectivamente.

```
if (cond) then
    Comando C1;
    Comando C2;
    ⋮
    Comando Cn;
else
    Comando D1;
    Comando D2;
    ⋮
    Comando Dm;
end
```

Estrutura Condicional if-then-else

Calcular a área de um quadrado. Entretanto, o programa deve calcular a área da sala apenas quando o valor informado pelo usuário for **maior do que zero**, pois não há sala com lado negativo ou nulo.

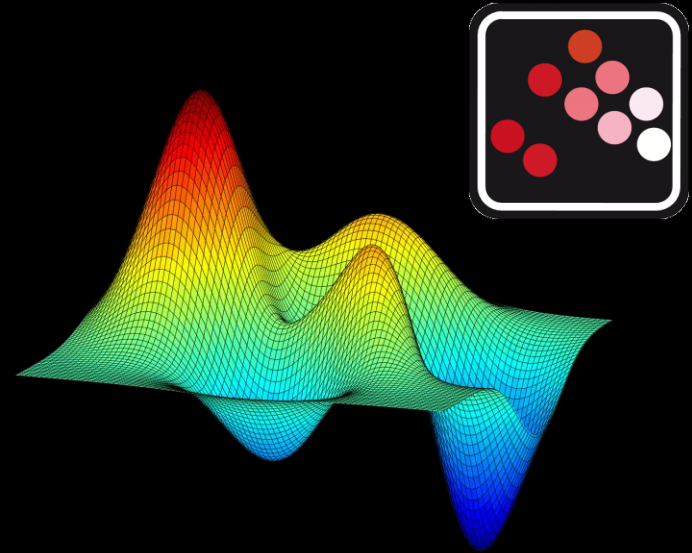
```
lado = input("Informe a medida do lado da sala: ");  
if (lado > 0) then  
    area = lado * lado;  
    printf("A area da sala é %f", area);  
else  
    printf("O valor informado é inválido! ");  
end
```

Estrutura Condicional if-then-else

Verificando se dois números são iguais ou não.

```
n1 = input("Informe o primeiro número: ");
n2 = input("Informe o segundo número: ");
if (n1 == n2) then
    printf("Os números fornecidos são iguais");
else
    printf("Os números fornecidos são diferentes");
end
```

Operadores Relacionais

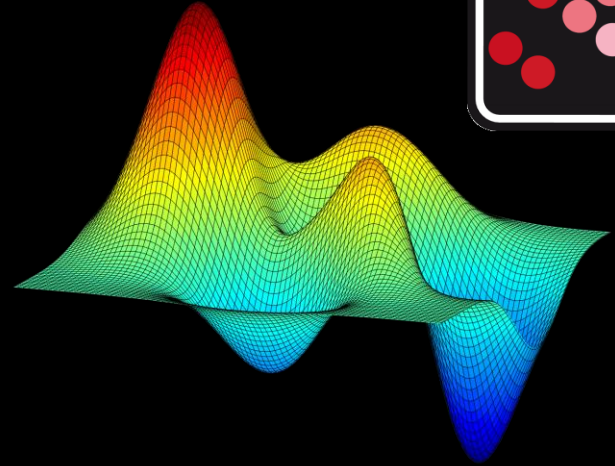


Operadores relacionais

O Scilab disponibiliza ao usuário um conjunto de operadores relacionais, que inclui os já utilizados `==` e `>`

Operador	Significado
<code>==</code>	Igual a
<code>~=</code>	Diferente de
<code>></code>	Maior que
<code>>=</code>	Maior ou igual a
<code><</code>	Menor que
<code><=</code>	Menor ou igual a

Operadores Lógicos



Operadores lógicos

Os operadores lógicos do Scilab podem ser utilizados para a formação de expressões lógicas mais complexas (que envolvem, por exemplo, **duas ou mais condições**).

Operador Lógico	Significado	Exemplo	Significado
&	“e” lógico	if (a > 0 & a < 10)...	Verifica se o valor da variável a é maior que 0 e menor que 10.
 	“ou” lógico	if (a == 0 b == 0) ...	Verifica se alguma das variáveis (a ou b) tem valor igual a zero.
~	Negação	if ~(a < 0) ...	Verifica se o valor da variável a NÃO é menor que zero. Isto é equivalente à condição a >= 0

Operadores lógicos

Os operadores lógicos do Scilab podem ser utilizados para a formação de expressões lógicas mais complexas (que envolvem, por exemplo, **duas ou mais condições**).

A	B	A & B	A B	$\sim A$
V	V	V	V	F
V	F	F	V	F
F	V	F	V	V
F	F	F	F	V

Operadores lógicos

Exemplos:

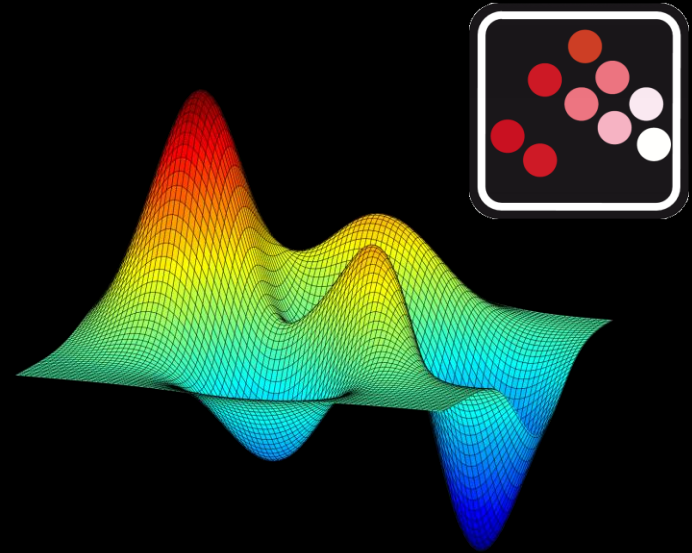
a)

```
a = 2; b = 3; c = 1;  
if (a > b) then  
    printf("%d",a);  
else  
    if (b > c) then  
        printf("%d",b);  
    else  
        printf("%d",c);  
    end  
end
```

b)

```
a = 10; b = 20; c = 10;  
if (a == b | a == c) then  
    printf("%d",a);  
else  
    printf("%d",b);  
end
```

Construção de Gráficos



Construção de gráficos

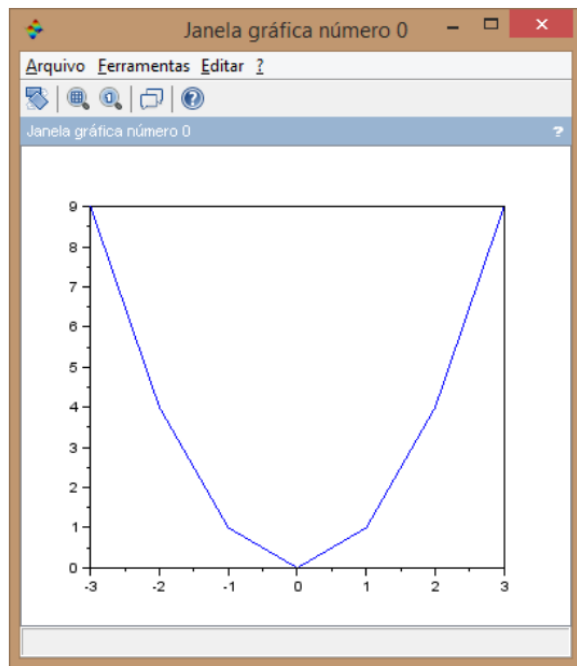
O Scilab disponibiliza uma série de recursos que possibilitam a construção de diversos tipos de gráficos.

Construção de um gráfico pela especificação de uma série de pares ordenados.

```
x = [-3, -2, -1, 0, 1, 2, 3];  
y = [9, 4, 1, 0, 1, 4, 9];  
plot(x, y);
```

Construção de gráficos

Construção de um gráfico pela especificação de uma série de pares ordenados.



	p_1	p_2	p_3	p_4	p_5	p_6	p_7
x	-3	-2	-1	0	1	2	3
$f(x)$	9	4	1	0	1	4	9

Construção de gráficos

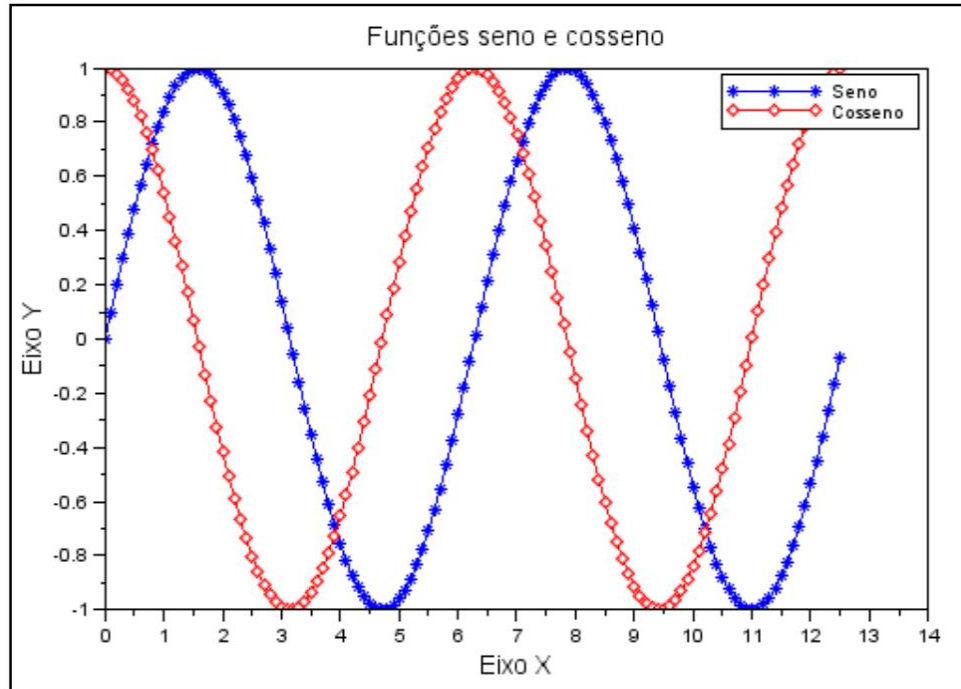
Representação de senos e cossenos:

```
clf
x = [0:0.1:4*pi];
y1 = sin(x);
y2 = cos(x);
plot(x,y1,'-b');
plot(x,y2,'-dr');
xtitle('Funções seno e cosseno');    // insere o título
xlabel('Eixo X');                    // insere o rótulo do eixo horizontal x
ylabel('Eixo Y');                    // insere o rótulo do eixo vertical y
legend('Seno', 'Cosseno');           // insere uma legenda para identificar,
                                     // na respectiva ordem, as duas curvas
```

Construção de gráficos

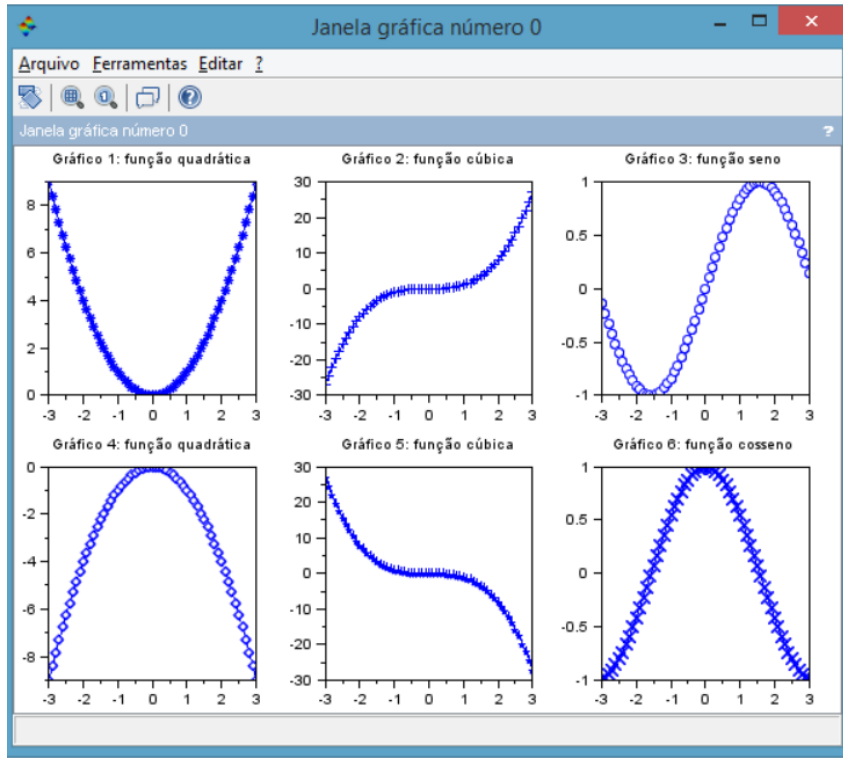
Representação de senos e cossenos:

Resultado:

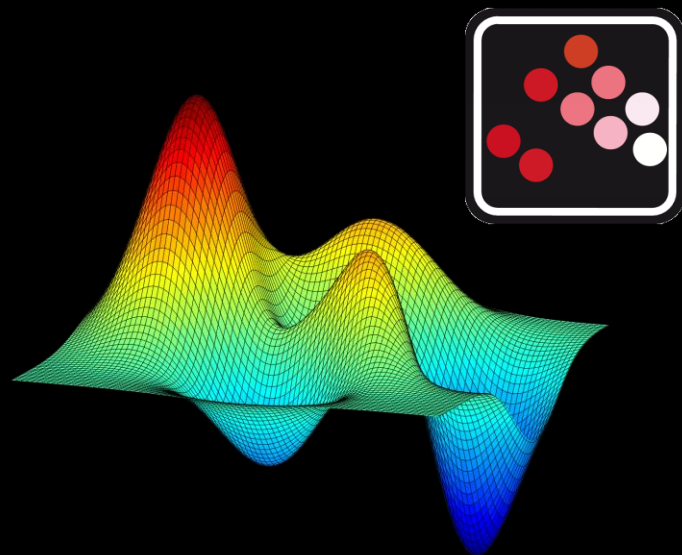


Inserindo diversos gráficos

Inserindo diversos gráficos na mesma janela com a função *subplot*:



Operações com Matrizes



Operações com Matrizes

Matrizes podem ser definidas no Scilab de maneira semelhante aos vetores. Os elementos devem ser especificados entre colchetes

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 1 \\ 7 & 4 & 9 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 1 \\ 3 & 2 \\ 4 & 3 \end{bmatrix} \quad C = \begin{bmatrix} 9 & 8 & 7 & 6 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

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

```
A =
```

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

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

```
B =
```

```
2.    1.
3.    2.
4.    3.
```

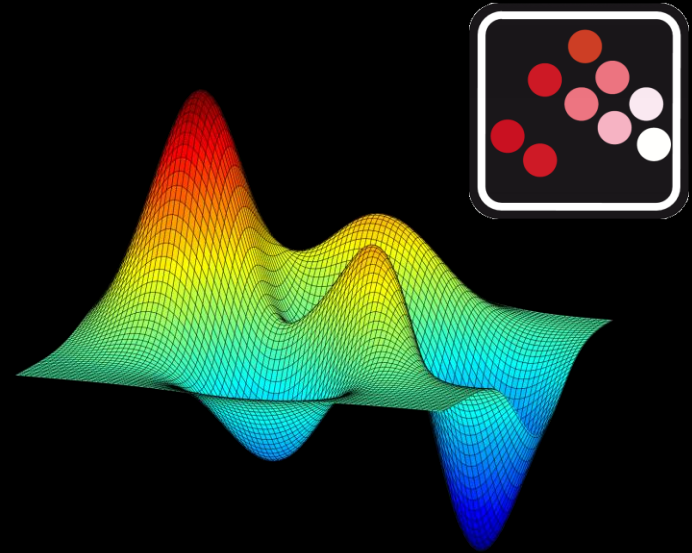
```
-->C = [9 8 7 6
```

```
-->1 2 3 4]
```

```
C =
```

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

Estruturas de Repetição



Estruturas de repetição

A estrutura de **repetição *while*** é utilizada quando se deseja repetir a execução de um bloco de comandos enquanto uma determinada condição for verdadeira.

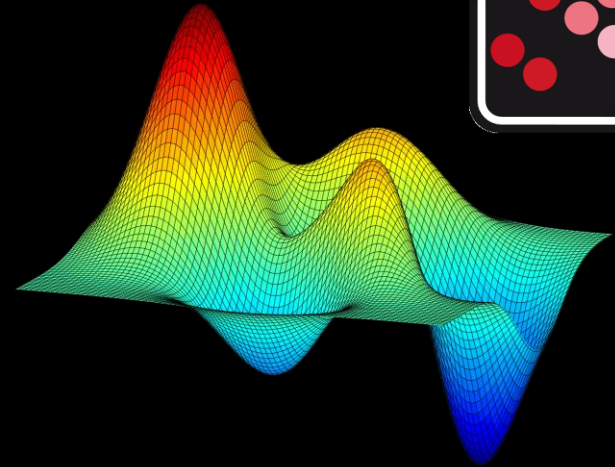
```
while (condição)
    comando 1
    comando 2
    ...
    comando n
end
```

Estruturas de repetição

Utilizando a estrutura **while**, faça um programa para ler números inteiros do teclado até que o número 0 seja lido. Ao término da leitura, o programa deverá apresentar a soma de todos os números informados.

```
total = 0;
x = input('Digite o primeiro número: ');
while (x ~= 0)
    total = total + x;
    x = input('Digite o próximo número (ou 0 para encerrar): ');
end
printf('A soma dos números informados é: %d', total);
```

Processamento de Imagens



Processamento de imagens

```
RGB = imread('teaset.png');  
imshow(RGB)f=gcf();  
f.name='Color Image';  
Image = rgb2gray(RGB);  
//figure('name','Gray Level Image'); imshow(Image); f=gcf();f.name='Gray  
Level Image'; imshow(Image,jetcolormap(256))f=gcf();  
f.name='Pseudo Color Image';
```

Obrigado!

Machine Learning

Prof. Dr. Diego Bruno

