

Parte 3

(Programação no Mathcad)

Comandos básicos

While e For - Loop
IF - Condicional

Somatório

$$f1(x, n) := \sum_{i=1}^n (x + i)$$

$$f1(5, 3) = 21$$

Somatorio(x, n) :=	sum ← 0	Adicionar 3 linhas
	for i ∈ 1 .. n	na linha 1 iniciar a variável
	sum ← (x + i) + sum	Na linha 2 adicionar o for
	sum	Na linha 3 inserir a operação do somatório
		Na linha 4 valor de saída

$$\text{Somatorio}(5, 3) = 21$$

Produtório

$$f2(x, n) := \prod_{i=1}^n (x + i)$$

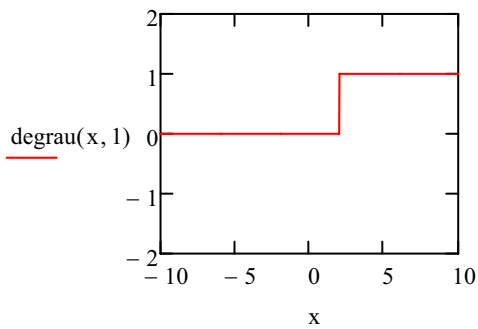
$$f2(5, 3) = 336$$

Produtorio(x, n) :=	prod ← 1	Adicionar 3 linhas
	for i ∈ 1 .. n	na linha 1 iniciar a variável
	prod ← (x + i) · prod	Na linha 2 adicionar o for
	prod	Na linha 3 inserir a operação do produtório
		Na linha 4 valor de saída

$$\text{Produtorio}(5, 3) = 336$$

Função degrau

degrau(x, pd) :=	var ← 0
	var ← 1 if x > 2
	var



Algumas Operações Com Matrizes

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

Multiplicação por escalar

$\text{Multi}(A, li, lj, nun) := \left \begin{array}{l} \text{for } i \in 0..li-1 \\ \quad \text{for } j \in 0..lj-1 \\ \quad \quad C_{i,j} \leftarrow nun \cdot A_{i,j} \end{array} \right $	C	<p>Inserir 2 linhas linha 1 for linha 1 for inserir outro for na linha 1 for inserir a operação linha 2 saída</p>
--	-----	---

$$\text{Multi}(A, 3, 3, 2) = \begin{pmatrix} 8 & 10 & 4 \\ 2 & 14 & 16 \\ 4 & 10 & 16 \end{pmatrix} \quad 2 \cdot A = \begin{pmatrix} 8 & 10 & 4 \\ 2 & 14 & 16 \\ 4 & 10 & 16 \end{pmatrix}$$

Multiplicação de matrizes

$\text{Prod_Mat}(A, B, dim) := \left \begin{array}{l} \text{for } i \in 0..dim-1 \\ \quad \text{for } j \in 0..dim-1 \\ \quad \quad C_{i,j} \leftarrow 0 \\ \quad \quad \text{for } k \in 0..dim-1 \\ \quad \quad \quad C_{i,j} \leftarrow A_{i,k} \cdot B_{k,j} + C_{i,j} \end{array} \right $	C	<p>Inserir 2 linhas linha 1 for linha 1 for inserir outro for inserir 2 linhas na linha 1 for inicializar variável na linha 2 for outro for na linha 1 do for operação linha 2 saída</p>
---	-----	---

$$\text{Prod_Mat}(A, B, 3) = \begin{pmatrix} 33 & 76 & 43 \\ 78 & 91 & 43 \\ 73 & 88 & 43 \end{pmatrix} \quad A \cdot B = \begin{pmatrix} 33 & 76 & 43 \\ 78 & 91 & 43 \\ 73 & 88 & 43 \end{pmatrix}$$

Método de Newton-Raphson

$$f(x) \approx f(x_k) + f'(x_k)(x - x_k) + \frac{f''(x_k)}{2}(x - x_k)^2$$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Função Derivada da função

$$g(x) := x^2 - 4$$

$$dg(x) := \frac{d}{dx}g(x)$$

```
N_Raph(x0, max, TOL) :=
  X0,0 ← 0
  X0,1 ← x0
  X0,2 ← g(X0,1)
  for i ∈ 0..max - 1
    Xi+1,0 ← i + 1
    Xi+1,1 ← Xi,1 -  $\frac{g(X_{i,1})}{dg(X_{i,1})}$ 
    Xi+1,2 ← g(Xi+1,1)
    break if Xi+1,2 < TOL
  X
```

Adicionar 5 linhas
 linha 1 contador de iterações
 linha 2 valor de x
 linha 3 valor da função
 linha 4 for
 no for adicionar 4 linhas
 linha 1 iteração
 linha 2 calculo de x(i+1)
 linha 3 avaliação da função
 linha 4 critério de parada
 linha 5 vetor que contem
 dados de saída

$$\text{Sol} := \text{N_Raph}(5, 100, 10^{-10})$$

$$\text{Sol} = \begin{pmatrix} 0 & 5 & 21 \\ 1 & 2.9 & 4.41 \\ 2 & 2.14 & 0.578 \\ 3 & 2.005 & 0.018 \\ 4 & 2 & 2.072 \times 10^{-5} \\ 5 & 2 & 2.684 \times 10^{-11} \end{pmatrix}$$

$$g(x) \text{ solve, } x \rightarrow \begin{pmatrix} -2 \\ 2 \end{pmatrix}$$

Método de Euler

$$y'(t) = f(t, y) \quad \text{with } y(t_0) = y_0$$

$$y_{k+1} = y_k + hf(t_k, y_k) \quad \text{with } y(t_0) = y_0$$

EDO

Solução Analítica

$$dy(t) := t^2 + t$$

$$y(t, y_0) := y_0 + \frac{t^3}{3} + \frac{t^2}{2}$$

```
Euler(y0, t0, tf, N) :=
  dt ← (tf - t0) / N
  V0,0 ← 0
  V0,1 ← t0
  V0,2 ← y0
  for i ∈ 0 .. N - 1
    Vi+1,0 ← i + 1
    Vi+1,2 ← Vi,2 + dy(Vi,1) · dt
    Vi+1,1 ← Vi,1 + dt
  V
```

Adicionar 6 linhas

linha 1 incremento

linha 2 contador de iterações

linha 2 valor de t0

linha 3 valor da função em t0

linha 4 for

no for adicionar 3 linhas

linha 1 iteração

linha 2 calculo de f(t)

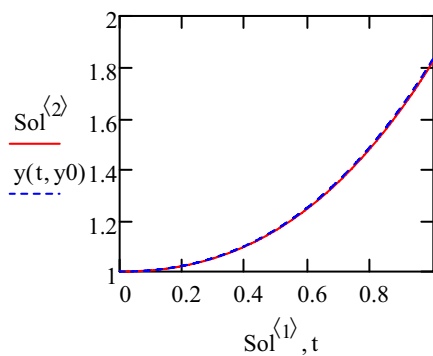
linha 3 incremento de t+dt

linha 5 vetor que contem os

dados de saída

y0 := 1 t0 := 0 tf := 1 N := 100

Sol := Euler(y0, t0, tf, N)



$Sol^{(2)} - y(Sol^{(1)}, y_0)$

