

Introdução ao Cálculo Numérico

Solução Numérica de EDOs - Problemas de Valor Inicial

- Equações diferenciais aparecem com grande frequência em modelos que descrevem quantitativamente fenômenos em diversas áreas, como por exemplo:
 - Mecânica dos fluidos;
 - Termodinâmica;
 - Reações químicas e nucleares;
 - Economia;
 - Biologia e Genética.

- Historicamente, a motivação para a construção dos primeiros computadores foi ocasionada, em grande parte, pela necessidade de serem calculadas trajetórias balísticas de mísseis de uma forma precisa e rápida.
- Tais trajetórias são calculadas através da resolução de um sistema de equações diferenciais ordinárias, cuja solução deve ser obtida de forma numérica.

- Os problemas típicos a serem resolvidos são aqueles nos quais:
 - um *valor inicial* é conhecido para a variável de interesse e deseja-se conhecer o *valor final* dessa variável;
 - um ou mais *valores de fronteira* são especificados para a variável de interesse, e deseja-se determinar o seu valor tal que as condições de fronteira sejam satisfeitas;
- Esses dois tipos de problemas são denominados de *Problema de Valor Inicial* (PVI) e *Problema de Valor de Fronteira* (PVF).

- Problema de Valor Inicial (PVI)

- Seja o PVI

$$\begin{cases} x' = f(t, x) \\ x(t_0) = x_0 \end{cases}$$

onde x é uma função de t , com $x' = \frac{d}{dt}x(t)$, e deseja-se calcular $x(t_1)$. A função f é a “inclinação” de x no ponto t .

- Tipicamente, tais problemas não tem solução analítica, sendo necessário recorrer a métodos numéricos que permitam aproximar adequadamente $x(t_1)$.

- Observe, no entanto, que nem todo PVI apresenta solução; tentar resolver numericamente tal problema resultará em divergência do processo numérico!
- Em existindo a solução, em geral só se pode afirmar que ela existe na *vizinhança* de t_0 .
- Existem três teoremas que garantem a existência e unicidade da solução (vide D.R. Kincaid and W. Cheney. *Numerical Analysis*. Brooks/Cole, Pacific Grove, 1991).

• Erros na solução numérica

- Ao se aproximar numericamente a solução de uma equação diferencial – através de um processo de integração numérica – uma série de *erros* surgem, os quais podem ser classificados como:
 - **Erro de truncamento local (ETL)**: é o erro existente em uma iteração da integração numérica ao substituirmos um processo *infinito* por outro, *finito*;
 - **Erro de arredondamento local (EAL)**: é causado pela precisão finita do computador;
 - **Erro de truncamento global (ETG)**: é a acumulação dos ETLs ao longo do processo de integração; é *dependente do método* e *independente da precisão* utilizada;
 - **Erro de arredondamento global (EAG)**: é a acumulação de todos os EALs;
 - **Erro total (ETT)**: é a soma dos ETG e EAG.

- Métodos numéricos para resolução de PVIs
 - Existe uma variedade de métodos numéricos para a resolução de PVIs, os quais distinguem-se uns dos outros pelas diferentes aproximações utilizadas e exibindo, conseqüentemente, diferentes valores assintóticos para os erros envolvidos.
 - Os métodos podem ser classificados, de forma ampla, em duas classes distintas:
 - Métodos de passo simples : *Taylor, Euler, Heun, Runge-Kutta*;
 - Métodos de passo múltiplo: *Milne, Adams-Bashforth, Adams-Moulton*.

- Método da série de Taylor

- Recapitulando, veja que a derivada x' é uma função $f(t, x(t))$, e (t_0, x_0) é um ponto único através do qual passa a curva solução particular
- O *método da série de Taylor* consiste em obtermos uma expansão em x de $f(t, x)$, de acordo com a série de Taylor, até um determinado número de termos,

$$\begin{aligned} x(t+h) &= x(t) + hx'(t) + \frac{h^2}{2!}x''(t) + \frac{h^3}{3!}x'''(t) + \dots \\ &= x(t) + h \left(x'(t) + \frac{h}{2} \left(x''(t) + \frac{h}{3} \left(x'''(t) + \dots + \frac{h}{n} x^{(n)}(t) \right) \right) \right) \end{aligned}$$

$$\text{onde } x''(t) = \frac{d}{dt}f(t, x(t)), \dots$$

- Método da série de Taylor

- Tipicamente, o número de termos a serem utilizados na expansão de Taylor, n , é fixado de antemão.
- Uma vez obtida a expansão, integramo-la nos intervalos $t_0 \leq t \leq t_1$ e $x_0 \leq x \leq x_1$, usando um *passo* (ou *espaçamento*) h .
- É recomendável que h seja escolhido de forma a garantir que o último valor calculado para t seja igual a t_1 . Preferencialmente, h deve ser um número representável, para minimizar o acúmulo de erros de ponto-flutuante em t .

- Método da série de Taylor

- Inicializam-se então duas variáveis, t e x com os valores iniciais prescritos no PVI, i.e. t_0 e x_0 .
- Para cada iteração, calcula-se o valor da expressão da expansão de Taylor, armazenando-o na própria variável x , e adiciona-se h a t .
- As derivadas x' , x'' , x''' , ..., $x^{(n)}$ são armazenadas em variáveis d_1 , d_2 , d_3 , ..., d_n
 - A derivada de primeira ordem é definida no problema;
 - As derivadas de maior ordem devem ser calculadas aplicando-se a regra da cadeia, já que a f é, em geral, função de t e de x .

- Método da série de Taylor
 - O método pode ser descrito de forma algorítmica como segue:

```
proc taylor(entrada: t0,t1,x0,h; saida: x)
  t:= t0
  x:= x0
  enquanto t<t1
    d1:= f(t,x)
    // calcule as demais derivadas x'', x''', ...,
    // armazenando-as em d2, d3, ...
    x:= x+h*(d1+(h/2)*(d2+(h/3)*(d3+...+(h/n)dn))
    t:= t+h
  fim
fim proc
```

- Método da série de Taylor

- Como a série de Taylor é *truncada* em n termos, todos os termos de ordem superior a n são descartados, constituindo o ETL do método, calculado como

$$\frac{1}{(n+1)!} h^{n+1} x^{(n+1)}(t + \theta h), 0 < \theta < 1.$$

- Observe que, como são feitas $m = \frac{|t_1 - t_0|}{h} + 1$ iterações, então teremos um ETG de ordem $O(h^n)$, no mínimo.

• Método da série de Taylor

• Vantagens:

- O método é extremamente simples;
- Se for possível utilizar derivadas até ordem n grande, então a precisão do método será bastante alta.

• Desvantagens:

- O método exige que derivadas parciais de ordem superior a 1 sejam contínuas na região considerada no plano tx , mas o PVI pode apresentar solução mesmo que haja descontinuidades dessas derivadas;
- Tais derivadas exigem codificação separada – utilizar aproximações para as derivadas introduz mais erros de arredondamento locais.

- Método de Taylor

- Exemplo: para analisar graficamente os resultados obtidos com os métodos, consideraremos o PVI

$$\begin{cases} x' &= 2tx - \frac{x}{2} \\ x(0) &= 4 \end{cases}$$

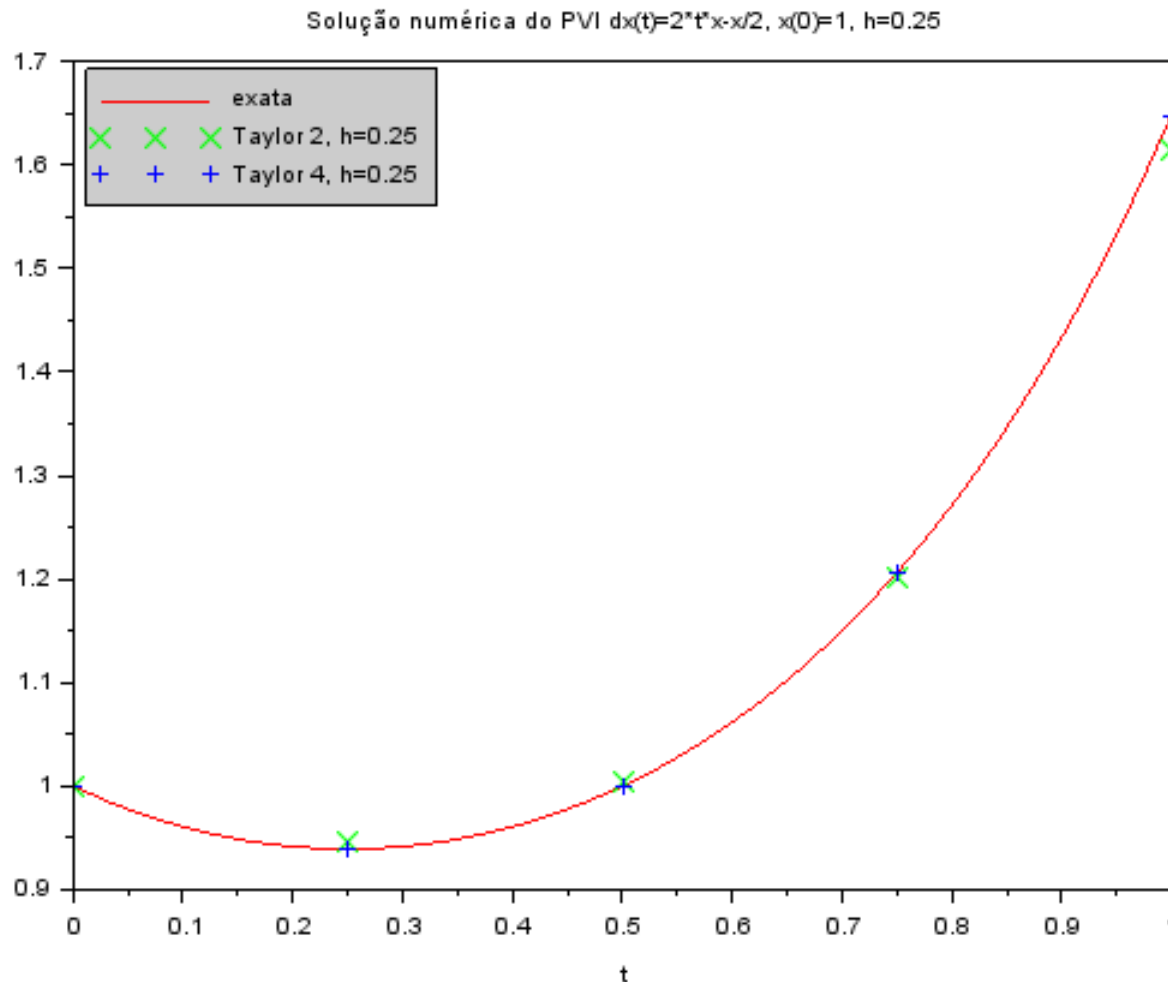
e deseja-se calcular $x(1)$.

- A solução exata desse problema é

$$x(t) = 4e^{t^2 - \frac{t}{2}}$$

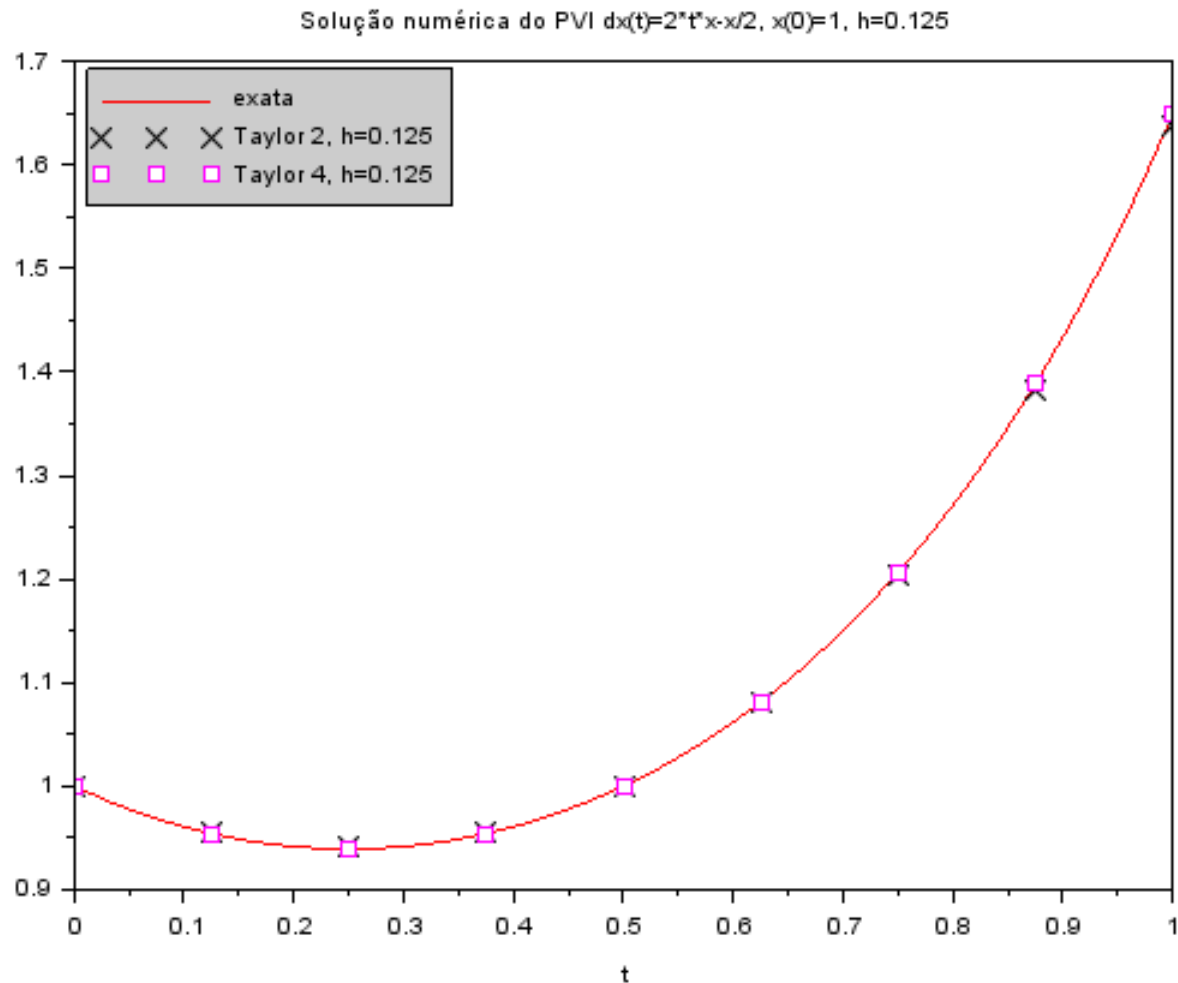
o que nos permite calcular $x(1) \cong 6,594885082$.

• Método da série de Taylor



Solução Numérica de EDOs – Problema de Valor Inicial

• Método da série de Taylor



Solução Numérica de EDOs – Problema de Valor Inicial

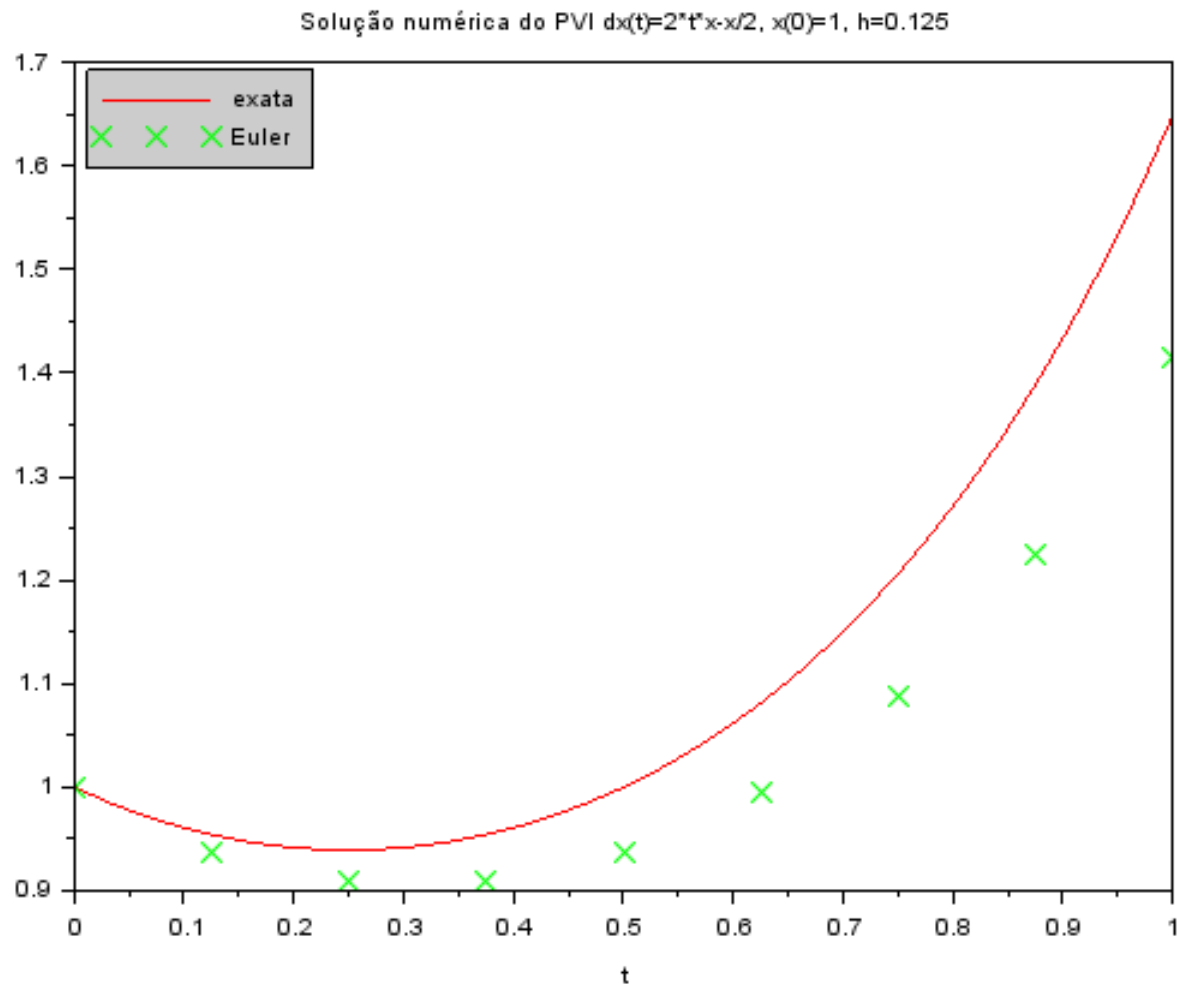
• Método de Euler

- Leonhard Euler (1707-1783) foi o pioneiro na tentativa de se resolver uma EDO de forma numérica, por volta de 1768.
- O método de Euler nada mais é do que um método da série de Taylor de primeira ordem, i.e.
$$x(t + h) = x(t) + hx'(t) = x(t) + hf(t, x).$$

• Método de Euler

- A simplicidade do método é seu forte, pois não é necessária nenhuma outra informação além daquela presente na especificação do PVI!
- Por outro lado, o seu ETL é de ordem $O(h)$, o que poderá exigir um passo de integração muito pequeno – e, conseqüentemente, serão necessárias muitas iterações para se obter uma precisão adequada (se comparado com métodos com ETL de ordem menor).

• Método de Euler



Solução Numérica de EDOs – Problema de Valor Inicial

- Método de Euler

- O método pode ser descrito de forma algorítmica como segue:

```
proc euler(entrada: t0,t1,x0,h; saida: x)
  t:= t0
  x:= x0
  enquanto t<t1
    d1:= f(t,x)
    x:= x+h*d1
    t:= t+h
  fim
fim proc
```

- Método de Heum

- O método de Heum é um método de passo simples que incorpora uma técnica de *correção* de um valor *previsto* como sendo o novo valor de x , sendo por isso um método do tipo *previsor-corretor*.

- Recapitulando, um PVI é especificado na forma

$$\begin{cases} x' = f(t, x) \\ x(t_0) = x_0 \end{cases}$$

e deseja-se determinar $x(t_1)$.

- Método de Heum

- Usando o teorema fundamental do cálculo, e integrando $x'(t)$ no intervalo $[t_0, t_1]$, vem

$$\int_{t_0}^{t_1} x'(t) dt = x(t_1) - x(t_0)$$

onde a antiderivada de $x'(t)$ é a função desejada, $x(t)$.

- Isolando $x(t_1)$ na equação acima, escrevemos

$$\begin{aligned} x(t_1) &= x(t_0) + \int_{t_0}^{t_1} x'(t) dt \\ &= x(t_0) + \int_{t_0}^{t_1} f(t, x(t)) dt \end{aligned}$$

- Método de Heun

- A integral no lado direito da equação pode ser calculada utilizando um processo de integração numérica; utilizando-se a regra dos trapézios, com passo $h = t_1 - t_0$, podemos escrever

$$x(t_1) \approx x(t_0) + \frac{h}{2} \left(f(t_0, x(t_0)) + f(t_1, x(t_1)) \right).$$

- No entanto, essa fórmula ainda contém $x(t_1)$ nos dois lados da aproximação!

• Método de Heum

- Para eliminar tal termo no lado direito, substituímo-lo por uma aproximação de Euler, i.e.

$$x(t_1) \approx x(t_0) + \frac{h}{2} \left(f(t_0, x(t_0)) + f(t_1, x(t_0) + hf(t_0, x(t_0))) \right).$$

- Tomando h suficientemente pequeno, e introduzindo uma variável adicional p , i.e. a aproximação por Euler do valor *previsto* para $x(t + h)$, obtemos o método de Heum:

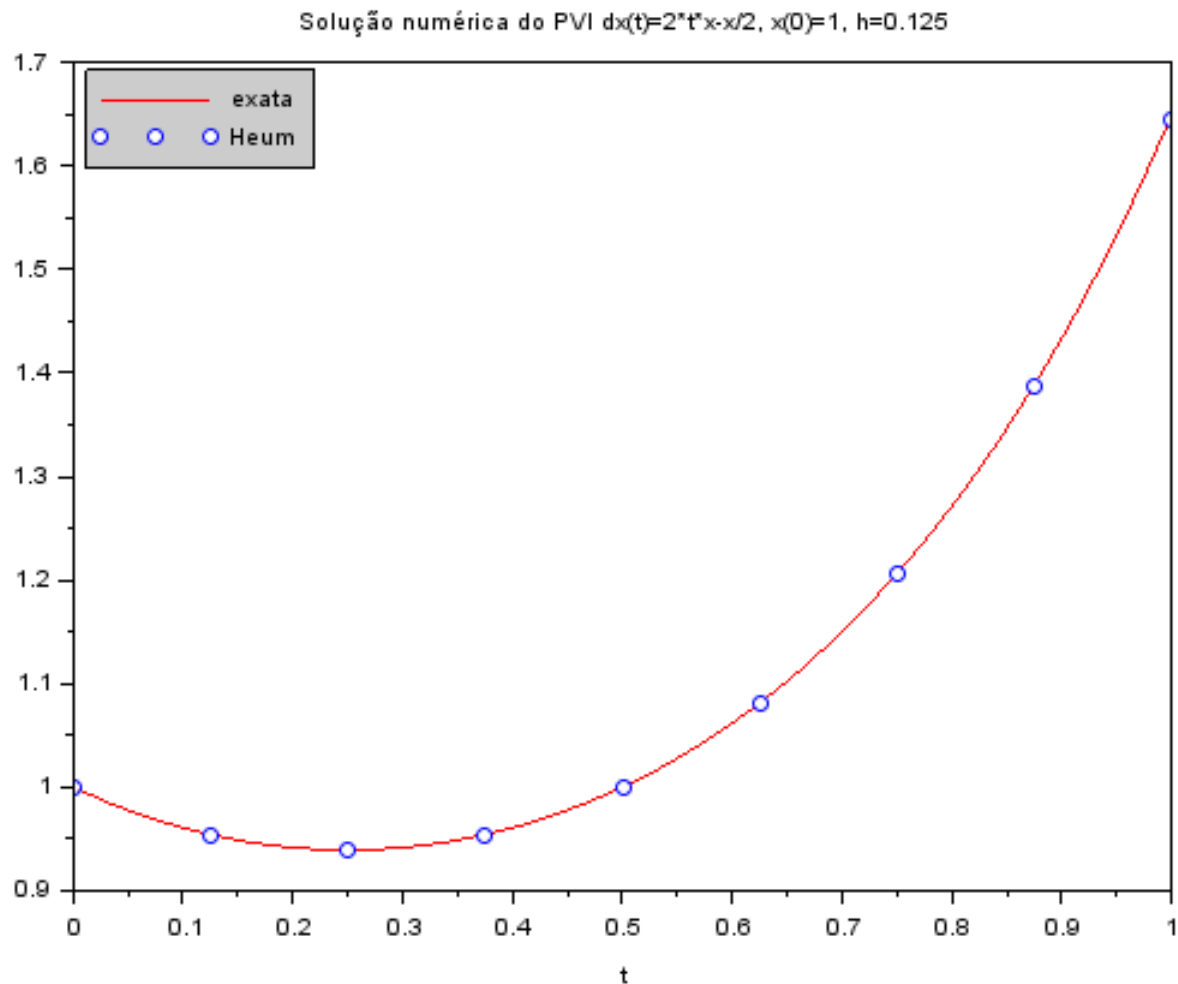
$$p(t + h) = x(t) + hf(t, x(t))$$

$$x(t + h) = x(t) + \frac{h}{2} \left(f(t, x(t)) + f(t + h, p(t + h)) \right)$$

- Método de Heun

- O método de Heun apresenta um ETL de ordem $O(h^3)$ e um ETG, após m iterações, da ordem $O(h^2)$.
- Essa maior exatidão do método é conseguida à custa de uma avaliação adicional por iteração da função f .

• Método de Heum



Solução Numérica de EDOs – Problema de Valor Inicial

- Método de Heum

- O método pode ser descrito de forma algorítmica como segue:

```
proc heum(entrada: t0,t1,x0,h; saida: x)
  t:= t0
  x:= x0
  enquanto t<t1
    d1:= f(t,x)
    p:= x+h*d1
    x:= x+(h/2)*(d1+f(t+h,p))
    t:= t+h
  fim
fim proc
```

- Métodos de Runge-Kutta

- Os métodos de Runge-Kutta são similares ao método da série de Taylor, mas as equações são escritas de tal forma que as derivadas de maior ordem são desnecessárias.
- Vejamos o método de Runge-Kutta de 2ª ordem: a série de Taylor até o termo de 3ª ordem é

$$x(t + h) = x(t) + hx'(t) + \frac{h^2}{2!}x''(t) + \frac{h^3}{3!}x'''(t) + \dots$$

e, como $x'(t) = f$ (pelo PVI), as derivadas acima podem ser escritas, usando a regra da cadeia, como:

• Métodos de Runge-Kutta

(cont.:)

$$x'(t) = f$$

$$x''(t) = f_t + f_x x' = f_t + f_x f$$

$$x'''(t) = f_{tt} + f_{tx}f + (f_t + f_x f)f_x + f(f_{xt} + f_{xx}f)$$

onde o subscrito indica derivação parcial em relação àquela variável.

• Assim, podemos escrever:

$$\begin{aligned} x(t+h) &= x + hf + \frac{1}{2}h^2(f_t + f_x f) + O(h^3) \\ &= x + \frac{1}{2}hf + \frac{1}{2}h(f + hf_t + hf f_x) + O(h^3) \end{aligned}$$

onde $x \equiv x(t)$ e $f \equiv f(t, x)$.

- Métodos de Runge-Kutta

- Usando a expressão para a série de Taylor em duas variáveis,

$$f(t + h, x + hf) = f + hf_t + hf f_x + O(h^2)$$

podemos eliminar os termos envolvendo f_t e f_x na expressão anterior e, descartando o termo $O(h^3)$, escrevemos

$$x(t + h) = x(t) + \frac{1}{2} hf(t + h, x + hf)$$

ou,

$$x(t + h) = x(t) + \frac{1}{2} (F_1 + F_2)$$

$$F_1 = hf(t, x)$$

$$F_2 = hf(t + h, x + F_1)$$

- Métodos de Runge-Kutta

- As últimas três equações definem o método de Runge-Kutta de 2ª ordem (o qual é equivalente ao método de Heum, visto anteriormente).
- Outros métodos dessa família são:
 - O *método modificado de Euler*:

$$\begin{aligned}x(t + h) &= x(t) + F_2 \\F_1 &= hf(t, x) \\F_2 &= hf\left(t + \frac{h}{2}, x + \frac{F_1}{2}\right)\end{aligned}$$

- Métodos de Runge-Kutta

- (cont.):

- O método de Runge-Kutta de 4ª ordem (RK4):

$$x(t + h) = x(t) + \frac{1}{6} (F_1 + 2F_2 + 2F_3 + F_4)$$

$$F_1 = hf(t, x)$$

$$F_2 = hf\left(t + \frac{h}{2}, x + \frac{F_1}{2}\right)$$

$$F_3 = hf\left(t + \frac{h}{2}, x + \frac{F_2}{2}\right)$$

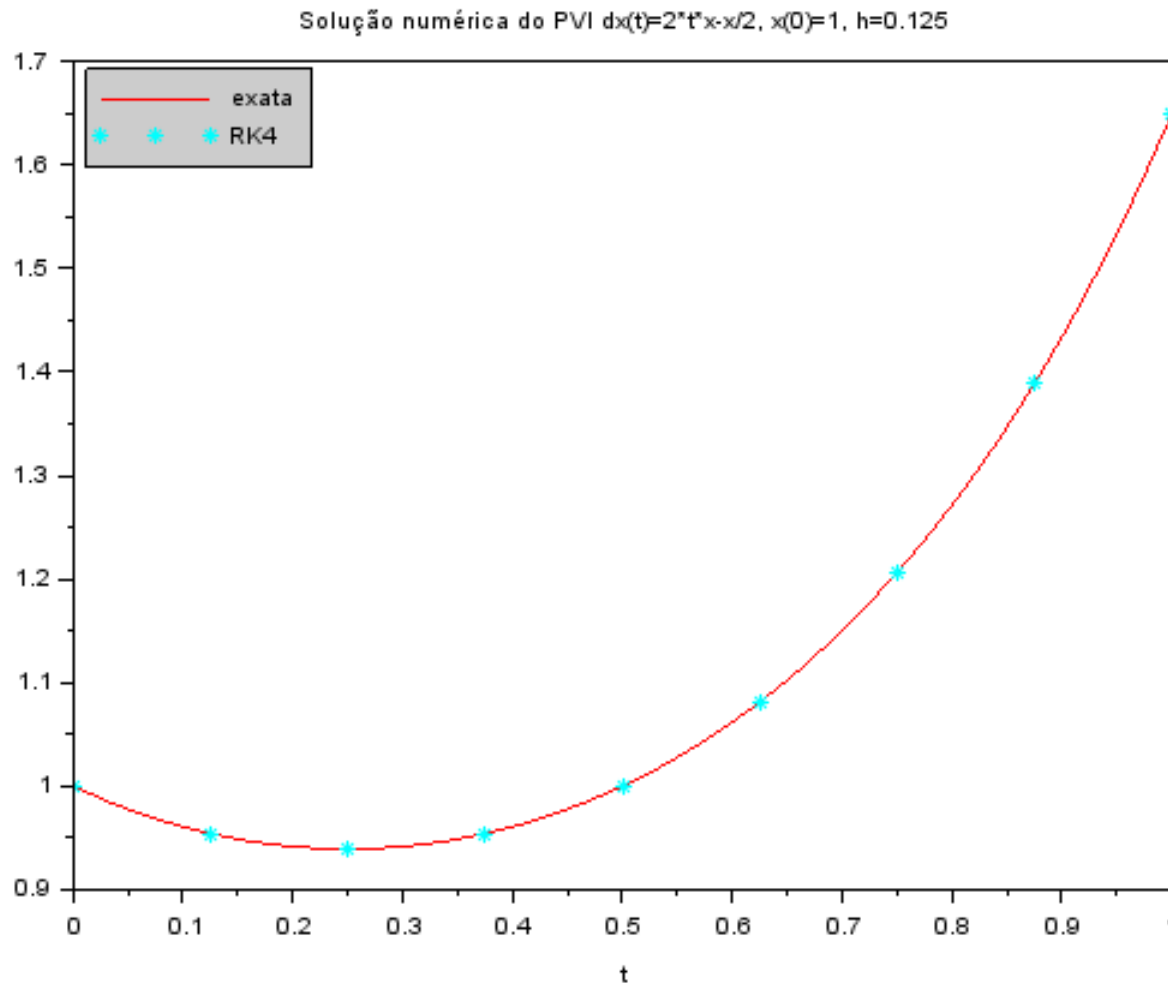
$$F_4 = hf(t + h, x + F_3)$$

- Métodos de Runge-Kutta

- O método RK4 pode ser descrito de forma algorítmica como segue:

```
proc rk4(entrada: t0,t1,x0,h; saida: x)
  t:= t0
  x:= x0
  enquanto t<t1
    F1:= f(t,x)
    F2:= f(t+h/2,x+F1/2)
    F3:= f(t+h/2,x+F2/2)
    F4:= f(t+h,x+F3)
    x:= x+(h/6)*(F1+2*F2+2*F3+F4)
    t:= t+h
  fim
fim proc
```

• Métodos de Runge-Kutta



Solução Numérica de EDOs – Problema de Valor Inicial

- Métodos de Runge-Kutta

- O ETL do método de Runge-Kutta de 4ª ordem é de ordem $O(h^5)$ e pode-se obter uma estimativa para o erro comparando os ETL de duas soluções, uma delas com passo h , e outra com $h/2$:

$$\begin{aligned}x\left(t + \frac{h}{2} + \frac{h}{2}\right) &= u + 2C\left(\frac{h}{2}\right)^5 \\x(t + h) &= v + Ch^5\end{aligned}$$

de onde

$$Ch^5 = \frac{u - v}{1 - 2^{-4}}$$

- Métodos de Runge-Kutta

- A expressão anterior permite estabelecer um procedimento adaptativo para o controle do passo h :

- Calculam-se as duas aproximações, u e v , para a solução numa iteração;
- Se $|u - v|$ for muito menor do que uma tolerância pré-especificada, isso significa que a convergência está sendo muito boa e, portanto, o passo pode ser maior (normalmente, duplica-se o passo);
- Caso contrário, deve-se diminuir o passo (usualmente, dividindo-o pela metade).

- Método adaptativo de Runge-Kutta-Fehlberg (RKF):
 - É um método desenvolvido em 1969 que combina os métodos de Runge-Kutta de 4ª e 5ª ordens (ambos com cinco avaliações da função f), de tal forma que as expressões contenham avaliações de f nos *mesmos* pontos.

- Método adaptativo de Runge-Kutta-Fehlberg (RKF):

- O método RKF é de 5ª ordem e é expresso por:

$$x(t+h) = x(t) + \sum_{i=1}^6 a_i F_i$$

$$\bar{x}(t+h) = x(t) + \sum_{i=1}^6 b_i F_i$$

$$F_i = hf \left(t + c_i h, x + \sum_{j=1}^{i-1} d_{ij} F_j \right), i = 1, 2, \dots, 6$$

- Método adaptativo de Runge-Kutta-Fehlberg (RKF):

- De posse das expressões $x(t + h)$ e $\bar{x}(t + h)$, pode-se definir o *erro* como

$$e = x(t + h) - \bar{x}(t + h) = \sum_{i=1}^6 (a_i - b_i)F_i$$

cujo valor, em módulo, é uma estimativa para o ETL Ch^5 , e pode ser utilizado para controle do passo h .

• Método adaptativo de Runge-Kutta-Fehlberg (RKF):

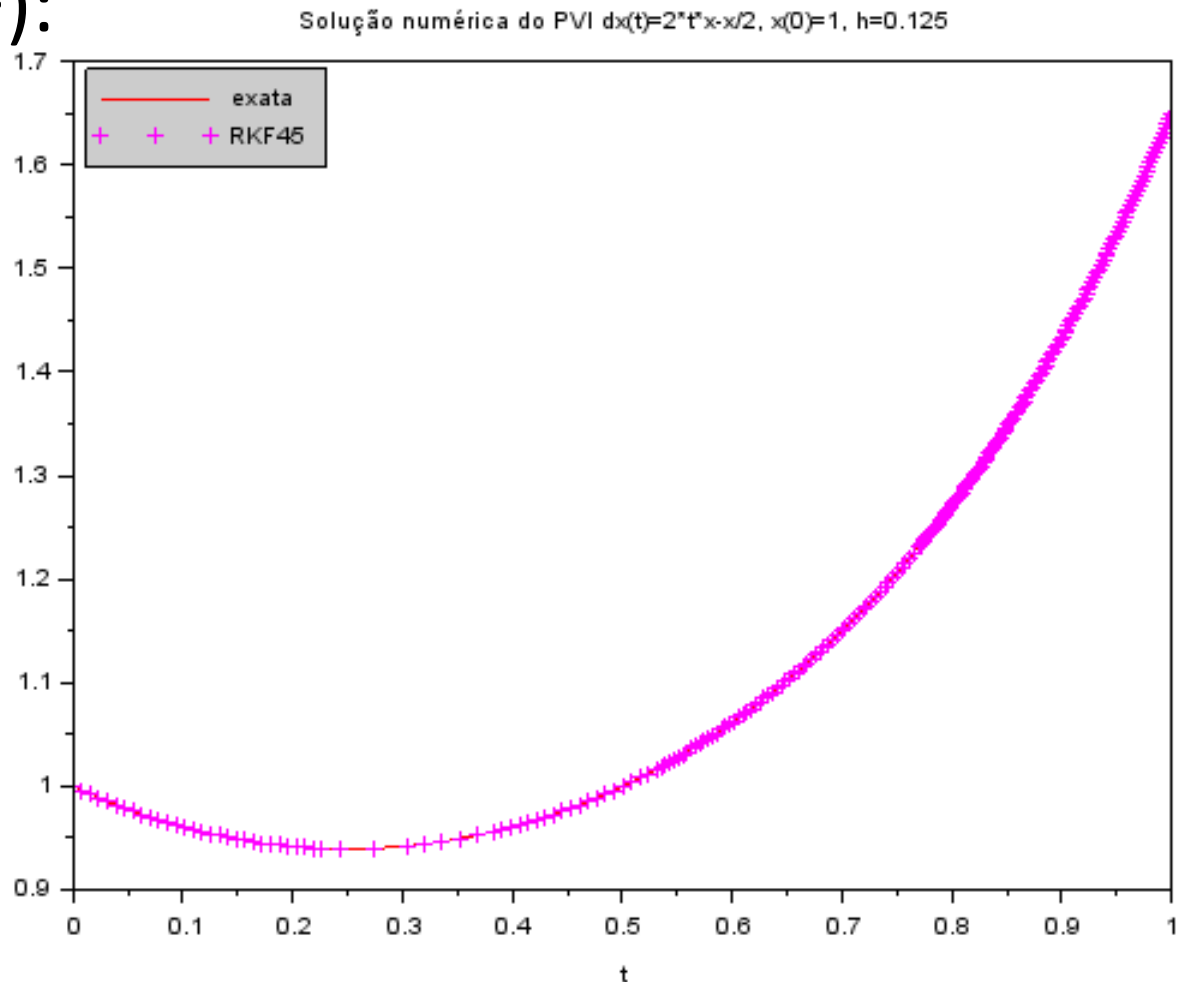
• As constantes a_i , $a_i - b_i$, c_i e d_{ij} são:

i	a_i	$a_i - b_i$	c_i	d_{i1}	d_{i2}	d_{i3}	d_{i4}	d_{i5}
1	$\frac{16}{135}$	$\frac{1}{360}$	0	0	0	0	0	0
2	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0
3	$\frac{6656}{12825}$	$-\frac{128}{4275}$	$\frac{3}{8}$	$\frac{3}{32}$	$\frac{9}{32}$	0	0	0
4	$\frac{28561}{56430}$	$-\frac{2197}{75240}$	$\frac{12}{13}$	$\frac{1932}{2197}$	$-\frac{7200}{2197}$	$\frac{7296}{2197}$	0	0
5	$-\frac{9}{50}$	$\frac{1}{50}$	1	$\frac{439}{216}$	-8	$\frac{3680}{513}$	$-\frac{845}{4104}$	0
6	$\frac{2}{55}$	$\frac{2}{55}$	$\frac{1}{2}$	$-\frac{8}{27}$	2	$-\frac{3644}{2565}$	$\frac{1859}{4104}$	$-\frac{11}{40}$

Solução Numérica de EDOs – Problema de Valor Inicial

- Método adaptativo de Runge-Kutta-Fehlberg (RKF):
 - O controle do passo h é feito comparando-se o valor de $|e|$ com uma tolerância δ :
 - Se $|e| > \delta$, então h é dividido por dois e a correção calculada para $x(t + h)$ é descartada;
 - Se $|e| < \delta/128$, então h é multiplicado por dois.
 - Os gráficos a seguir mostram o comportamento do método para o problema-exemplo:

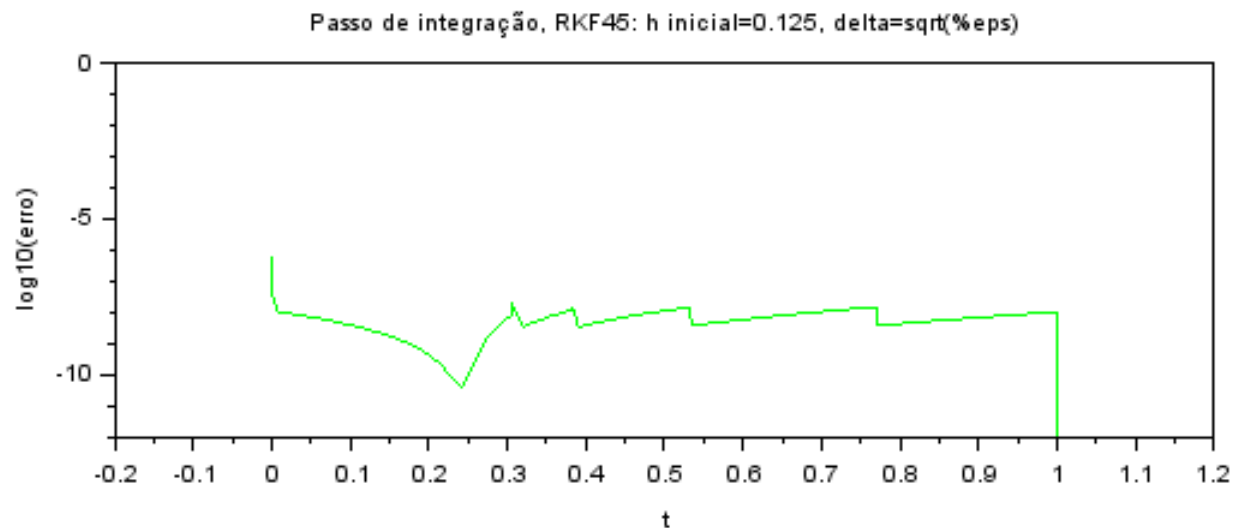
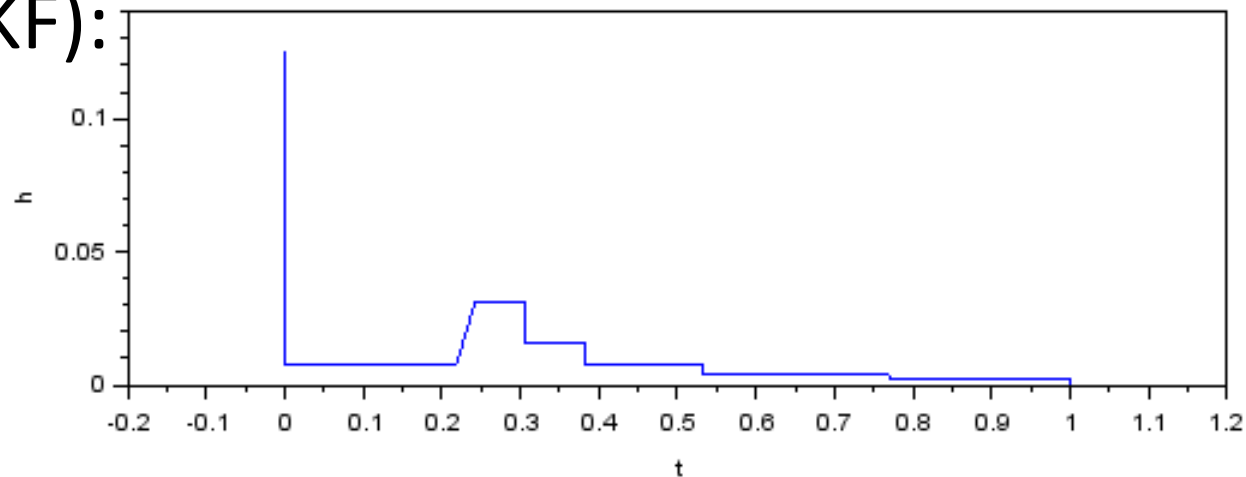
- Método adaptativo de Runge-Kutta-Fehlberg (RKF):



Nesse exemplo, com $\delta = \sqrt{\varepsilon_M}$, foram necessárias 245 iterações até obter o valor de $x(1)$.

Solução Numérica de EDOs – Problema de Valor Inicial

- Método adaptativo de Runge-Kutta-Fehlberg (RKF):



Solução Numérica de EDOs – Problema de Valor Inicial

- Método adaptativo de Runge-Kutta-Fehlberg (RKF):
 - O método RKF pode ser descrito de forma algorítmica como segue:

```
proc rkf(entrada: t0,t1,x0,h,delta; saida: x)
  t:= t0
  x:= x0
  continua:= VERDADEIRO
  k:= 0
  enquanto continua
    dt:= t1-t
    se abs(dt)<=h então
      h:= dt
      continua:= FALSO
    fim
```

• Método adaptativo de Runge-Kutta-Fehlberg (RKF):

```
para i:=1 até 6 faça
    s:= x
    para j:=1 até i-1 faça
        s:= s+d(i,j)*F(j)
    fim
    F(i):= h*f(t+c(i)*h,s)
fim
x_ant:= x
s:= 0
e:= 0
para i:=1 até 6 faça
    s:= s+a(i)*F(i)
    e:= e+a_b(i)*F(i)
fim
x:= x+s
```

- Método adaptativo de Runge-Kutta-Fehlberg (RKF):

```
t_ant:= t
t:= t+h
se continua então
  se abs(e)>=delta então
    h:= h/2
    x:= x_ant
    t:= t_ant
  senão se abs(e)<delta/128 então
    h:= h*2
  fim
  k:= k+1
fim
fim
fim proc
```

- Métodos de passo múltiplo

- Os métodos de passo múltiplo diferenciam-se dos anteriores por utilizarem na fórmula de correção para $x(t + h)$, um conjunto de pontos t_i nos quais a função f é avaliada.
- Esses pontos, ou *nós*, não necessariamente precisam estar igualmente espaçados.
- Novamente retomando o PVI, observe que se $x(t)$ é a solução exata do problema, então

$$\int_{t_n}^{t_{n+1}} x'(t) dt = x(t_{n+1}) - x(t_n).$$

- Métodos de passo múltiplo

- Podemos então escrever

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} f(t, x(t)) dt.$$

- A integral pode ser aproximada por

$$\int_{t_n}^{t_{n+1}} f(t, x(t)) dt \approx h(Af_n + Bf_{n-1} + Cf_{n-2} + Df_{n-3})$$

onde $f_i \equiv f(t_i, x(t_i))$, e A, B, C e D são coeficientes obtidos exigindo-se que essa aproximação seja exata sempre que o integrando se um polinômio de grau menor ou igual a 3

- Métodos de passo múltiplo

- Usando a base de Newton para o espaço polinomial Π^3 ,

$$\begin{cases} p_0(t) = 1 \\ p_1(t) = t \\ p_2(t) = t(t+1) \\ p_3(t) = t(t+1)(t+2) \end{cases}$$

e usando $t_0 = 0, h = 1$, escrevemos

$$\int_0^1 p_i(t) dt = Ap_i(0) + Bp_i(-1) + Cp_i(-2) + Dp_i(-3),$$

para $0 \leq i \leq 3$, o que nos leva ao sistema de equações lineares

- Métodos de passo múltiplo

$$\begin{cases} Ap_0(0) + Bp_0(-1) + Cp_0(-2) + Dp_0(-3) &= \int_0^1 p_0(t) dt \\ Ap_1(0) + Bp_1(-1) + Cp_1(-2) + Dp_1(-3) &= \int_0^1 p_1(t) dt \\ Ap_2(0) + Bp_2(-1) + Cp_2(-2) + Dp_2(-3) &= \int_0^1 p_2(t) dt \\ Ap_3(0) + Bp_3(-1) + Cp_3(-2) + Dp_3(-3) &= \int_0^1 p_3(t) dt \end{cases}$$

e, como o termo independente é perfeitamente definido, obtemos o sistema de equações lineares expresso a seguir:

- Métodos de passo múltiplo

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ & -1 & -2 & -3 \\ & & 2 & 6 \\ & & & -6 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} 1 \\ 1/2 \\ 5/6 \\ 9/4 \end{bmatrix}$$

cuja solução é:

$$A = 55/24, B = -59/24, C = 37/24, D = -9/24,$$

de onde

$$\int_{t_n}^{t_{n+1}} f(t, x(t)) dt \approx \frac{h}{24} (55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3})$$

- Métodos de passo múltiplo

- Com isso, podemos escrever

$$x(t_{n+1}) = x(t_n) + \frac{h}{24} (55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3})$$

a qual é conhecida como a fórmula de Adams-Bashforth de 4ª ordem (AB4).

- Observe que são necessários quatro valores de f para cada correção de x , já na primeira iteração ($n = 0$):
 - Para tal, utiliza-se uma aproximação de Euler para esses quatro valores;
 - Após, basta ser feita uma reutilização de valores previamente calculados, juntamente com valores novos.

- Métodos de passo múltiplo
 - O método AB4 pode ser descrito de forma algorítmica como segue:

```
proc ab4(entrada: t0,t1,x0,h; saida: x)
  t:= t0
  x:= x0
  F_0:= h*f(t,x) ; x:= x+F_0 ; t:= t+h
  F_1:= h*f(t,x) ; x:= x+F_1 ; t:= t+h
  F_2:= h*f(t,x) ; x:= x+F_2 ; t:= t+h
  F_3:= h*f(t,x) ; x:= x+F_3 ; t:= t+h
  enquanto t<t1
    x:= x+(1/24)*(55*F_3-59*F_2+37*F_1-9*F_0)
    F_0:= F_1 ; F_1:= F_2 ; F_2:= F_3
    t:= t+h
    F_3:= h*f(t,x)
  fim
fim proc
```

- Métodos de passo múltiplo

- Um método de passo múltiplo relacionado a esse é o de *Adams-Moulton de 4ª ordem* (AM4), obtido incluindo-se a função f_{n+1} e descartando-se a função f_{n-3} na aproximação para a integral:

$$x(t_{n+1}) = x(t_n) + \frac{h}{24} (9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2})$$

- Note que essa fórmula não pode ser utilizada diretamente para se obter $x(t_{n+1})$, já que

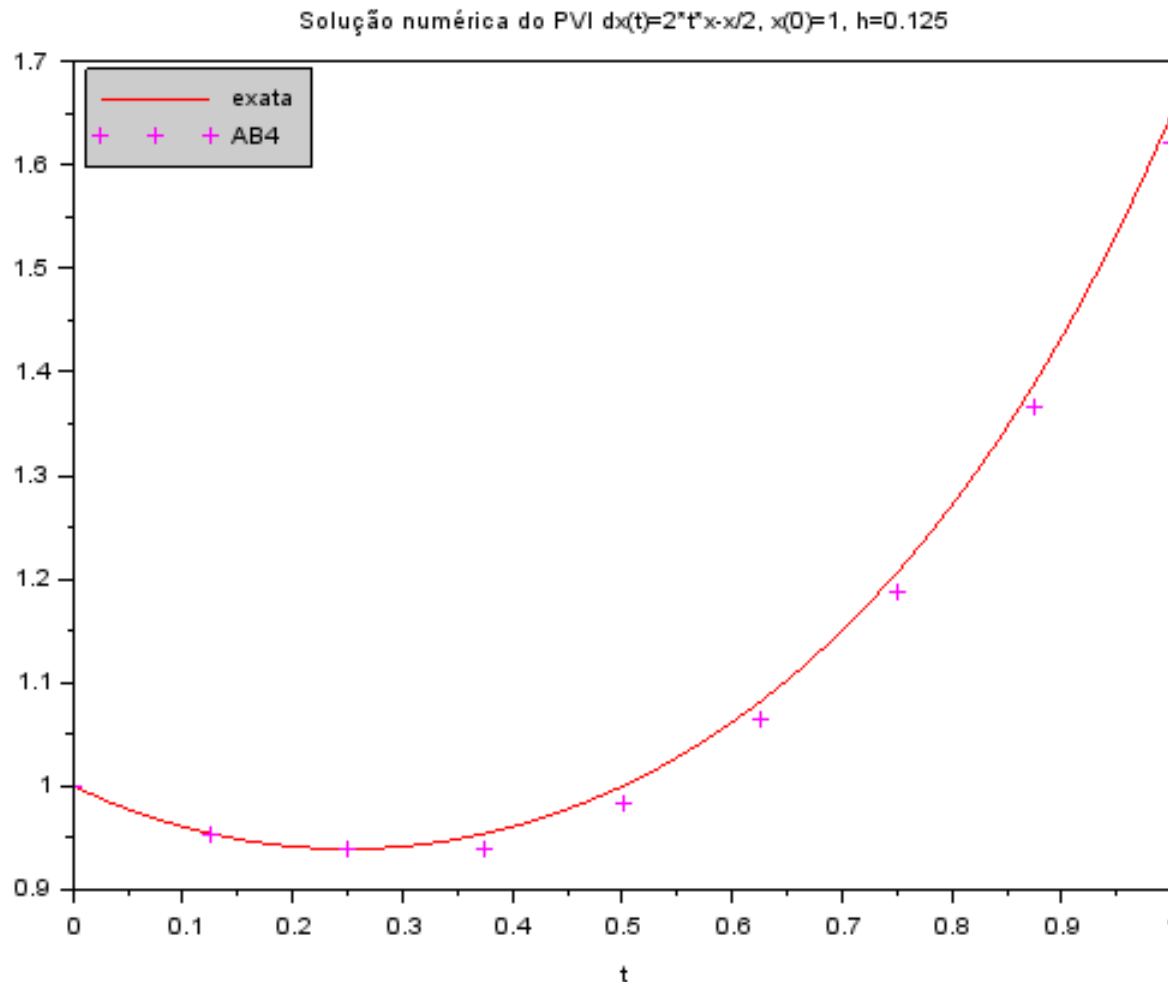
$$f_{n+1} \equiv f(t_{n+1}, x(t_{n+1})) !$$

- Métodos de passo múltiplo
 - No entanto, podemos combinar esse dois métodos de passo múltiplo, obtendo um método do tipo previsor-corretor:
 - Utiliza-se a fórmula de Adams-Bashforth para calcular uma estimativa $\tilde{x}(t_{n+1})$ para $x(t_{n+1})$;
 - Essa estimativa $\tilde{x}(t_{n+1})$ é usada para se calcular f_{n+1} e utilizá-la na fórmula de Adams-Moulton, com a qual calculamos, efetivamente, $x(t_{n+1})$.

- Métodos de passo múltiplo
 - O método previsor-corretor AB4_AM4 pode ser descrito de forma algorítmica como segue:

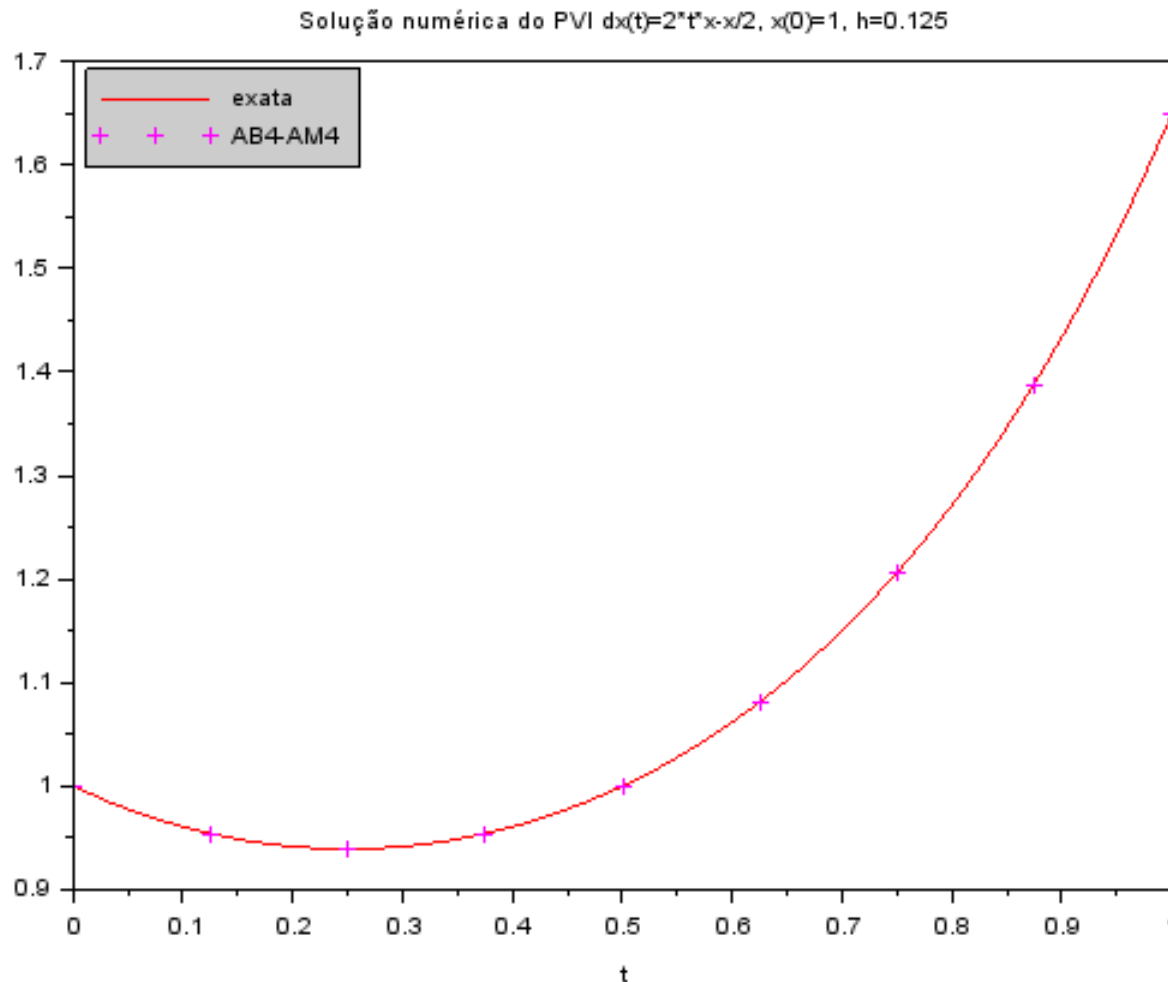
```
proc ab4_am4(entrada: t0,t1,x0,h; saida: x)
  t:= t0
  x:= x0
  F_0:= h*f(t,x) ; x:= x+F_0 ; t:= t+h
  F_1:= h*f(t,x) ; x:= x+F_1 ; t:= t+h
  F_2:= h*f(t,x) ; x:= x+F_2 ; t:= t+h
  F_3:= h*f(t,x) ; x:= x+F_3 ; t:= t+h
  enquanto t<t1
    x_til:= x+(1/24)*(55*F_3-59*F_2+37*F_1-9*F_0)
    F_0:= F_1 ; F_1:= F_2 ; F_2:= F_3
    t:= t+h
    F_3:= h*f(t,x_til)
    x:= x+(1/24)*(9*F_3+19*F_2-5*F_1+F_0)
  fim
fim proc
```

• Métodos de passo múltiplo



Solução Numérica de EDOs – Problema de Valor Inicial

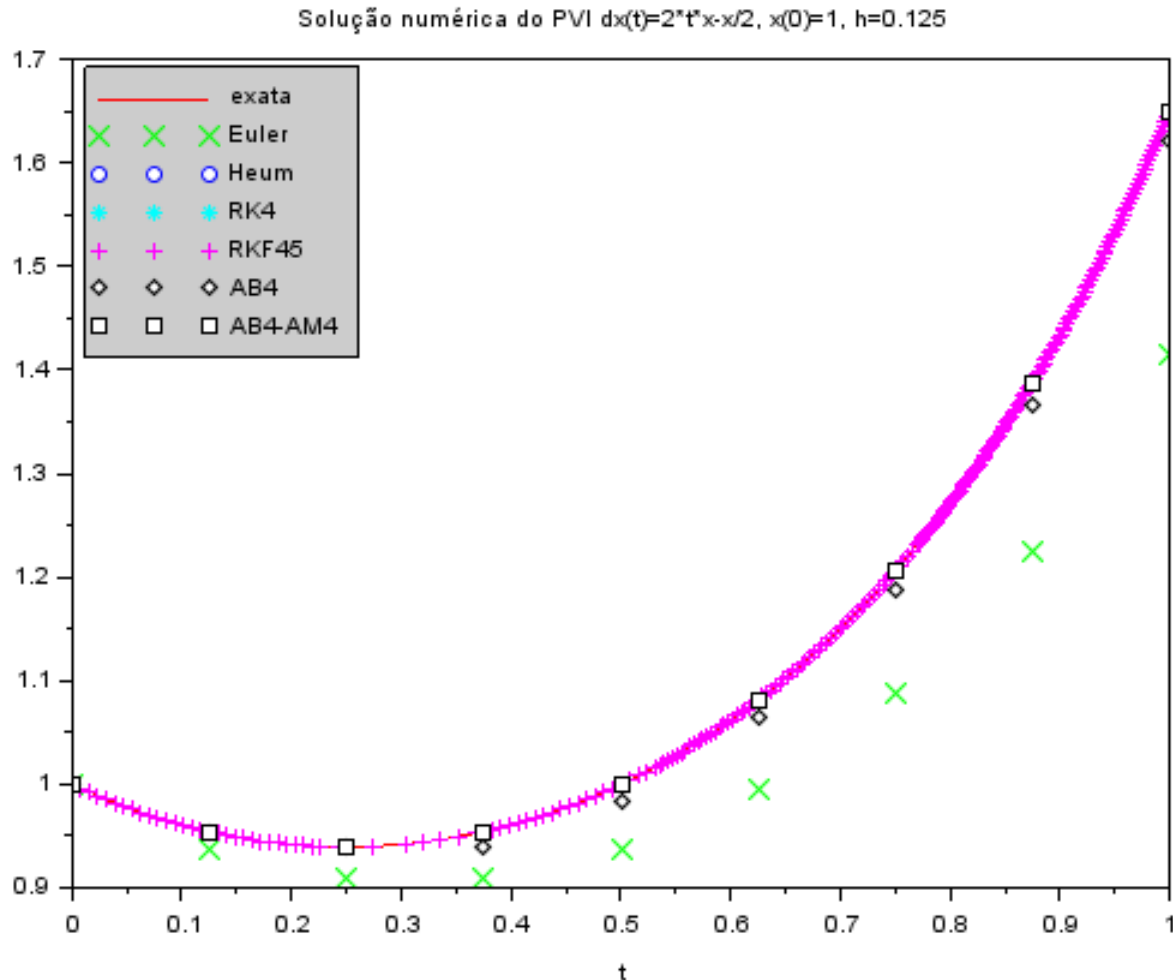
• Métodos de passo múltiplo



Solução Numérica de EDOs – Problema de Valor Inicial

- Comparação entre os métodos
 - Como vimos, os métodos apresentados têm comportamentos diferentes, em função das aproximações utilizadas, uso da técnica previsor-corretor, tamanho do passo de integração e controle adaptativo desse último.
 - O gráfico a seguir mostra as aproximações para a solução exata utilizando esses métodos:

• Comparação entre os métodos



Solução	$x(1)$
exata	1,648721271
Euler	1,415787998
Heum	1,644840524
RK4	1,648717517
RKF45	1,648703850
AB4	1,620943700
AB4-AM4	1,647923591

Solução Numérica de EDOs – Problema de Valor Inicial

• Sistema de EDOs

- Sistemas de EDOs surgem, por exemplo, ao se buscar resolver uma EDO não-linear, através da introdução de variáveis e equações auxiliares.
- Um PVI também pode ser expresso envolvendo mais de uma EDO, tipicamente envolvendo o acoplamento de uma ou mais equações através de uma ou mais variáveis.
- Os métodos vistos anteriormente são facilmente estendidos para se resolver sistemas de EDOs, bastando considerar o uso de vetores em lugar das variáveis escalares.

- Sistema de EDOs

- Formalmente, um PVI com mais do que uma equação pode ser escrito, em forma matricial, como

$$\begin{cases} \mathbf{X}' = \mathbf{F}(t, \mathbf{X}) \\ \mathbf{X}(t_0) = \mathbf{X}_0 \end{cases}$$

onde

$$\mathbf{X}' = \begin{cases} f_1(t, x_1, \dots, x_n) \\ \vdots \\ f_n(t, x_1, \dots, x_n) \end{cases}, \quad \mathbf{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

- Sistema de EDOs

- Porém, é mais adequado, inclusive por razões de implementação dos algoritmos, considerarmos t como se fosse uma variável adicional do problema, i.e. $t \equiv x_0$.
- Observe que, dessa forma, devemos introduzir também uma nova equação diferencial, em x_0 apenas, cuja integração resulta em t , i.e.

$$x_0' = f_0 = 1.$$

- Com isso, podemos escrever o sistema na *forma autônoma*.

- Sistema de EDOs
 - PVI: forma autônoma

$$\begin{cases} \mathbf{X}' = \mathbf{F}(\mathbf{X}) \\ \mathbf{X}(x_0) = \mathbf{X}_0 \end{cases}$$

Onde

$$\mathbf{X}' = \begin{cases} f_0 = 1 \\ f_1(x_0, x_1, \dots, x_n) \\ \vdots \\ f_n(x_0, x_1, \dots, x_n) \end{cases}, \quad \mathbf{X} = \begin{bmatrix} x_0 \equiv t \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

- Sistema de EDOs

- Por exemplo, considere o PVI

$$\begin{cases} x' &= 2x - 0,02xy \\ y' &= 0,0005xy - 0,8y \\ x(0) &= 3000 \\ y(0) &= 120 \end{cases}$$

- Para converter para a forma autônoma, escrevemos:

$$t = x_0$$

$$x = x_1$$

$$y = x_2$$

- Sistema de EDOs
 - Substituindo no PVI, vem

$$\left\{ \begin{array}{lcl} x_0' & = & 1 \\ x_1' & = & 2x_1 - 0,02x_1x_2 \\ x_2' & = & 0,0005x_1x_2 - 0,8x_2 \\ x_0 & = & 0 \\ x_1(x_0) & = & 3000 \\ x_2(x_0) & = & 120 \end{array} \right.$$

- Sistema de EDOs

- Ou, em forma matricial:

$$\begin{cases} \mathbf{X}' = \mathbf{F}(\mathbf{X}) \\ \mathbf{X}(x_0) = \mathbf{X}_0 \end{cases}$$

onde

$$\mathbf{F}(\mathbf{X}) = \begin{cases} 1 \\ 2x_1 - 0,02x_1x_2 \\ 0,0005x_1x_2 - 0,8x_2 \end{cases},$$

$$\mathbf{X} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} t \\ x \\ y \end{bmatrix}, \quad \mathbf{X}_0 = \begin{bmatrix} 0 \\ 3000 \\ 120 \end{bmatrix}$$

- Sistema de EDOs

- Os métodos de passo simples, passo simples adaptativo e passo múltiplo vistos anteriormente são facilmente implementados de forma a tratar qualquer PVI na forma autônoma, seja ele com apenas uma EDO, ou com um sistema de EDOs.

• Sistema de EDOs

- Observe, ainda, que em alguns ambientes de programação (Matlab, Scilab) não é possível fazer referência a elementos de vetores com índice “0”.
- Nesse caso, basta renumerar os índices das variáveis x_0, x_1, \dots, x_n , para x_1, x_2, \dots, x_{n+1}
- No nosso exemplo, teríamos, então

$$F(X) = \begin{cases} 1 \\ 2x_2 - 0,02x_2x_3 \\ 0,0005x_2x_3 - 0,8x_3 \end{cases},$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} t \\ x \\ y \end{bmatrix}, \quad X_0 = \begin{bmatrix} 0 \\ 3000 \\ 120 \end{bmatrix}$$

Solução Numérica de EDOs – Problema de Valor Inicial

- Sistema de EDOs

- Quando uma EDO é não-linear,

$$y^{(n)} = f(t, y, y', \dots, y^{(n-1)}), y^{(i)} = \frac{d^i}{dt^i} y$$

podemos introduzir variáveis auxiliares,

$$x_0 = t; x_1 = y; x_2 = y'; x_3 = y''; \dots x_n = y^{(n-1)}$$

de onde obtemos o sistema (na forma autônoma)

$$\begin{cases} x_0' &= & 1 \\ x_1' &= & x_2 \\ x_2' &= & x_3 \\ x_3' &= & x_4 \\ \vdots & \vdots & \vdots \\ x_n' &= & f(x_0, x_1, \dots, x_n) \end{cases}$$

• Sistema de EDOs

- Por exemplo, seja a EDO

$$\sin(t) y''' + \cos(ty) + \sin(t^2 + y'') + (y')^3 = \log t.$$

- Escrevendo

$$x_0 = t; x_1 = y; x_2 = y'; x_3 = y''$$

e isolando y''' na equação acima, vem

$$y''' = \frac{1}{\sin t} (\log t - \cos(ty) - \sin(t^2 + y'') - (y')^3)$$

$$(y'')' = \frac{1}{\sin x_0} (\log x_0 - \cos(x_0 x_1) - \sin(x_0^2 + x_3) - (x_2)^3)$$

$$x_3' = \frac{1}{\sin x_0} (\log x_0 - \cos(x_0 x_1) - \sin(x_0^2 + x_3) - (x_2)^3)$$

- Sistema de EDOs

- De onde obtemos o sistema

$$\begin{cases} x'_0 &= 1 \\ x'_1 &= x_2 \\ x'_2 &= x_3 \\ x'_3 &= \frac{1}{\sin x_0} (\log x_0 - \cos(x_0 x_1) - \sin(x_0^2 + x_3) - (x_2)^3) \end{cases}$$

o qual pode, agora, ser resolvido através dos métodos numéricos anteriormente vistos.

Métodos de passo simples para Problemas de Valor Inicial expressos na forma autônoma

- O uso da forma autônoma facilita a solução do PVI, por separar a especificação do PVI e o método usado para resolvê-lo.
- Além disso, por expressar de forma vetorial as variáveis envolvidas no método, permite que ele possa ser aplicado a qualquer PVI na forma autônoma, independente da quantidade de variáveis especificadas no problema.
- Considere, por exemplo, o método de Euler para resolver um PVI na forma

$$\begin{cases} x' &= f_1(t, x, y) \\ y' &= f_2(t, x, y) \\ x(t_0) &= x_0 \\ y(t_0) &= y_0 \end{cases}$$

tal que se deseja calcular $x(t_1)$ e $y(t_1)$.

- Um algoritmo que implementa o método de Euler para esse PVI pode ser expresso como:

```
proc Euler(entrada:  $t_0, x_0, y_0, h$ ; saída:  $t, x, y$ )  
1.   $t := t_0$   
2.   $x := x_0$   
3.   $y := y_0$   
4.  enquanto  $t < t_1$ ,  
5.     $\hat{x} := x + h \cdot f_1(t, x, y)$   
6.     $\hat{y} := y + h \cdot f_2(t, x, y)$   
7.     $t := t + h$   
8.     $x := \hat{x}$   
9.     $y := \hat{y}$   
10. fim  
fimproc
```

- Note que, para um PVI com mais equações, esse algoritmo tem de ser estendido, adicionando-se linhas semelhantes às linhas 5 e 6.

- Obviamente, as funções f_1 e f_2 também devem ser codificadas:

```
proc  $f_1$ (entrada:  $t, x, y$ ; saída:  $z$ )  
1.  $z := f_1(t, x, y)$   
fimproc
```

```
proc  $f_2$ (entrada:  $t, x, y$ ; saída:  $z$ )  
1.  $z := f_2(t, x, y)$   
fimproc
```

- Agora, na forma autônoma, escrevemos o PVI como

$$\mathbf{F}(\mathbf{X}) = \begin{pmatrix} 1 \\ f_1(t, x, y) \\ f_2(t, x, y) \end{pmatrix},$$

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} t \\ x \\ y \end{bmatrix}, \mathbf{X}_0 = \begin{bmatrix} t_0 \\ x_0 \\ y_0 \end{bmatrix}.$$

- Observe que a função $\mathbf{F}(\mathbf{X})$ retorna um **vetor** de valores, com a mesma quantidade de elementos que a dos vetores \mathbf{X} e \mathbf{X}_0 ; usaremos a notação $\mathbf{X}[i]$ para indicar o elemento i do vetor \mathbf{X} .

- Na forma autônoma, o algoritmo que implementa o método de Euler para esse PVI pode ser expresso como:

```
proc Euler(entrada:  $X_0, h$ ; saída:  $X$ )  
1.  $X := X_0$   
2. enquanto  $X[1] < t_1$ ,  
3.    $Y := F(X)$   
4.    $X := X + h \cdot Y$   
5. fim  
fim proc
```

- A função $F(X)$ implementa as funções especificadas no PVI:

```
proc  $F$ (entrada:  $X$ ; saída:  $Y$ )  
1.  $t := X[1], x := X[2], y := X[3]$   
2.  $Y[1] := 1$  // equação diferencial para  $t'$   
3.  $Y[2] := f_1(t, x, y)$   
4.  $Y[3] := f_2(t, x, y)$   
fim proc
```

- Vamos mostrar a equivalência entre ambas as formas do método de Euler:

- Para a forma clássica, temos:

$$x(t+h) = x(t) + hf_1(t, x(t), y(t))$$

$$y(t+h) = y(t) + hf_2(t, x(t), y(t))$$

- Já na forma autônoma:

$$\mathbf{X}(t+h) = \mathbf{X}(t) + h\mathbf{F}(\mathbf{X}(t))$$

$$\text{com } \mathbf{X}(t) = \begin{bmatrix} t \\ x(t) \\ y(t) \end{bmatrix}, \mathbf{F}(\mathbf{X}) = \begin{bmatrix} 1 \\ f_1(t, x(t), y(t)) \\ f_2(t, x(t), y(t)) \end{bmatrix}, \text{ de onde}$$

$$\begin{bmatrix} t+h \\ x(t+h) \\ y(t+h) \end{bmatrix} = \begin{bmatrix} t \\ x(t) \\ y(t) \end{bmatrix} + h \begin{bmatrix} 1 \\ f_1(t, x(t), y(t)) \\ f_2(t, x(t), y(t)) \end{bmatrix}. \blacksquare$$

- Observe que, dessa forma, desacoplamos o método, da especificação do PVI.
- Qualquer PVI pode ser resolvido usando a implementação proposta para o método de Euler, bastando prover a implementação adequada da função F .
- De forma semelhante, o método de Heun pode ser escrito na forma autônoma como

```
proc Heun(entrada:  $X_0, h$ ; saída:  $X$ )  
1.   $X := X_0$   
2.  enquanto  $X[1] < t_1$ ,  
3.     $Y := F(X)$   
4.     $P := P + h \cdot Y$  // valor previsto  
5.     $Y_1 := F(P)$   
6.     $X := X + h/2 \cdot (Y + Y_1)$  // valor corrigido  
7.  fim  
fim proc
```

- Mostraremos agora como essa implementação é equivalente a usarmos o método de Heun sobre cada variável, separadamente.
- Escrevendo o método de Heun sobre cada variável, temos:

$$p(t+h) = x(t) + hf_1(t, x(t), y(t))$$

$$q(t+h) = y(t) + hf_2(t, x(t), y(t))$$

$$x(t+h) = x(t) + \frac{h}{2} \left(f_1(t, x(t), y(t)) + f_1(t, p(t+h), q(t+h)) \right)$$

$$y(t+h) = y(t) + \frac{h}{2} \left(f_2(t, x(t), y(t)) + f_2(t, p(t+h), q(t+h)) \right)$$

- Na forma autônoma, temos:

$$\mathbf{P}(t + h) = \mathbf{X}(t) + h\mathbf{F}(\mathbf{X}(t))$$

$$\begin{bmatrix} t + h \\ p(t + h) \\ q(t + h) \end{bmatrix} = \begin{bmatrix} t \\ x(t) \\ y(t) \end{bmatrix} + h \begin{bmatrix} 1 \\ f_1(t, x(t), y(t)) \\ f_2(t, x(t), y(t)) \end{bmatrix}$$

e

$$\mathbf{X}(t + h) = \mathbf{X}(t) + \frac{h}{2} \left(\mathbf{F}(\mathbf{X}(t)) + \mathbf{F}(\mathbf{P}(t + h)) \right)$$

$$\begin{bmatrix} t + h \\ x(t + h) \\ y(t + h) \end{bmatrix} = \begin{bmatrix} t \\ x(t) \\ y(t) \end{bmatrix} + \frac{h}{2} \begin{bmatrix} 1 + 1 \\ f_1(t, x(t), y(t)) + f_1(t + h, p(t + h), q(t + h)) \\ f_2(t, x(t), y(t)) + f_2(t + h, p(t + h), q(t + h)) \end{bmatrix}. \blacksquare$$

- Observa-se a vantagem de se usar a forma autônoma, por reduzir o tamanho do código usado para implementar um método.
- Usando uma notação similar, o método RK4 pode ser expresso como

```
proc RK4(entrada:  $X_0, h$ ; saída:  $X$ )  
1.  $X := X_0$   
2. enquanto  $X[1] < t_1$ ,  
3.    $F1 := h \cdot F(X)$   
4.    $F2 := h \cdot F(X + F1/2)$   
5.    $F3 := h \cdot F(X + F2/2)$   
6.    $F4 := h \cdot F(X + F3)$   
7.    $X := X + 1/6 \cdot (F1 + 2 \cdot F2 + 2 \cdot F3 + F4)$   
8. fim  
fim proc
```