

3.4 Movimento ao longo de uma curva no espaço (terça parte)

Antes de começar com a nova matéria, vamos considerar um problema sobre o material recentemente visto.

Problema: (Projeção de uma trajetória sobre os planos $-xy$, $-xz$, $-yz$)

Uma partícula se move ao longo de uma curva no 3-espaço, com posição no tempo dada pelo seguinte vetor (t é dado em segundos e x em metros):

$$\mathbf{r}(t) = 6t \cdot \mathbf{i} + (5t + 4) \cdot \mathbf{j} + 6t^2 \cdot \mathbf{k}$$

- Representar graficamente a trajetória e suas projeções sobre os planos de coordenadas.
- Determine os vetores $\mathbf{v}(t)$ e $\mathbf{a}(t)$.
- Quais são os valores de \mathbf{r} , \mathbf{v} , e \mathbf{a} no instante $t = 3s$?
- Qual a distância que a partícula percorre em $3s$? (Veja as fórmulas para determinar o comprimento de uma curva na seção 2.2.)

Matematicamente o comprimento s de uma curva parametrizada $\mathbf{r}(t)$, quando t varie de a a b , é dada por

$$s = \int_a^b |\dot{\mathbf{r}}(t)| \cdot dt$$

Solução:

```

• reset() :
a:=0:b:=1:
x:=t->6*t:y:=t->5*t+4:z:=t->6*t^2:
curve:=plot::Curve3d([x(t),y(t),z(t)],t=a..b):
projexz:=plot::Curve3d([x(t),0,z(t)],Color=RGB::Red,t=a..b)
:
projexy:=plot::Curve3d([x(t),y(t),0],Color=RGB::Green,t=a..b)
:
projeyz:=plot::Curve3d([0,y(t),z(t)],Color=RGB::Magenta,t=a..b)
:
plot(curve,projexz,projexy,projeyz, Scaling=Constrained)

```

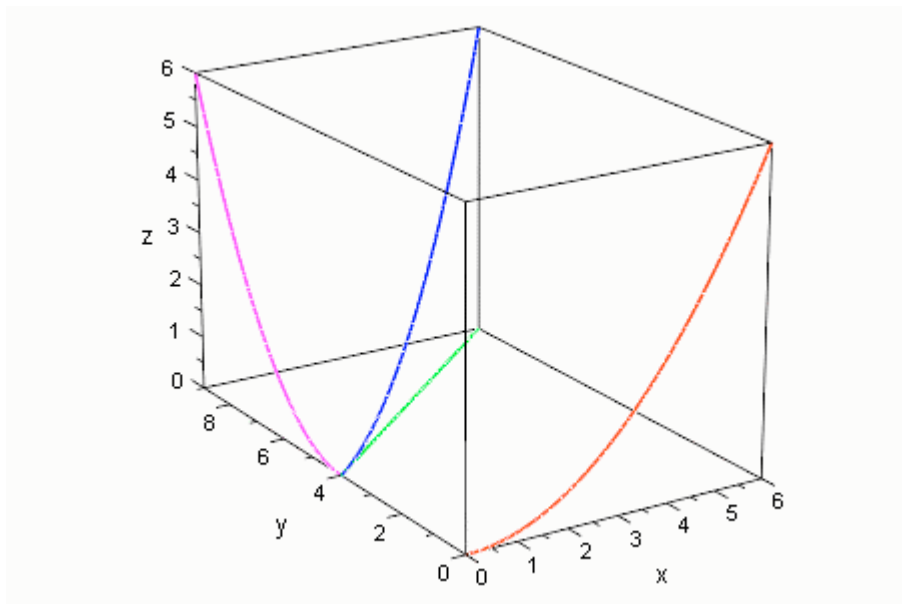


Fig. 3.4-23

```

• reset() :
x:=t->6*t:
y:=t->5*t+4:
z:=t->6*t^2:
t1:=3:
pos:=matrix([[x(t), y(t), z(t)]]);
v:=matrix([[x'(t), y'(t), z'(t)]]);
a:=matrix([[x''(t), y''(t), z''(t)]]);
ds:=sqrt(v[1,1]^2+v[1,2]^2+v[1,3]^2);
arco:=numeric::int(ds, t=0..t1)

```

Resultados:

$(3t, 5t + 4, 6t)$ posição
 $(6, 5, 12t)$ velocidade
 $(0, 0, 12)$ aceleração
 $(144t^2 + 61)^{1/2}$ caminho depois de t segundos
 60.93123879 (m) caminho depois de 3 s

Vamos, agora, falar sobre os três **sistemas de coordenadas** usadas comumente na física. Esta seção serve tanto de revisão de conceitos básicos da geometria analítica como de preparo para os capítulos posteriores.

3.4.9 Sistema de coordenadas polares e movimento circular

Se queremos estudar o movimento contínuo de uma partícula ao longo de uma curva plana, utilizamos freqüentemente coordenadas retangulares (x - y). Este sistema de coordenadas é particularmente útil para descrever movimentos onde as componentes x e y da aceleração são geradas ou determinadas independentemente.

Uma aplicação importante foi o problema do movimento de um projétil. Para as condições de um projétil simples, $a_x = 0$, $a_y = -g$, pode-se ver que os movimentos em x e y são independentes.

Em outra descrição do movimento curvilíneo fizemos uso de variáveis da trajetória, que eram medições feitas ao longo da tangente \mathbf{t} e da normal \mathbf{n} à trajetória da partícula. Considerava-se que os vetores (sistema de coordenadas) \mathbf{t} e \mathbf{n} movem-se ao longo da trajetória com a partícula. Aplicações práticas são, p. ex., projetos de construção de estradas e pistas para carros de corrida.

Consideremos agora uma terceira descrição do movimento curvilíneo plano, a saber, as **coordenadas polares** onde a partícula é localizada pela distância radial r medida a partir de um pólo fixo e por uma medida angular ϕ a partir do eixo x até a linha radial.

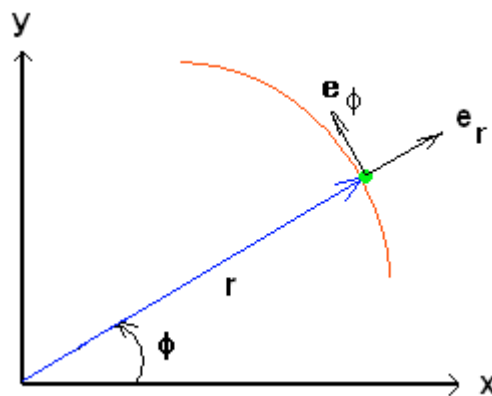


Fig. 3.4-24

Coordenadas polares são particularmente úteis quando o movimento possui restrição por meio do controle de uma distância radial e de uma posição angular ou também, quando um movimento sem restrições é observado pela medição de uma distância radial e de uma posição angular.

A partícula está acompanhada por dois vetores unitários \mathbf{e}_r e \mathbf{e}_ϕ , que estão estabelecidos nas direções positivas de r e ϕ , respectivamente.

O vetor posição \mathbf{r} até a partícula possui módulo igual à distância radial r e direção especificada pelo vetor unitário \mathbf{e}_r . Assim, expressamos a posição da partícula pelo vetor

$$\mathbf{r} = r \cdot \mathbf{e}_r \quad (1)$$

Sendo r , r e \mathbf{e}_r funções do tempo, temos para a velocidade

$$\mathbf{v} = dr/dt = dr/dt \cdot \mathbf{e}_r + d\mathbf{e}_r/dt \cdot r$$

Para derivar \mathbf{e}_r com respeito ao tempo, introduzimos as componentes retangulares dos dois vetores unitários (confira a descomposição de \mathbf{t} e \mathbf{n} em 3.4.3):

$$\begin{aligned} \mathbf{e}_r &= \cos\varphi \mathbf{i} + \sin\varphi \mathbf{j} \\ \mathbf{e}_\varphi &= \cos(\varphi + \pi/2) \cdot \mathbf{i} + \sin(\varphi + \pi/2) \cdot \mathbf{j} \\ &= -\sin\varphi \cdot \mathbf{i} + \cos\varphi \cdot \mathbf{j} \end{aligned} \quad (2)$$

Utilizando essas componentes, obtemos

$$d\mathbf{e}_r/dt = -\sin\varphi \cdot d\varphi/dt \cdot \mathbf{i} + \cos\varphi \cdot d\varphi/dt \cdot \mathbf{j} = d\varphi/dt \cdot \mathbf{e}_\varphi$$

Compare com Eq. (10) de 3.4.3.

Para a derivada de \mathbf{e}_φ com relação ao tempo obteremos

$$d\mathbf{e}_\varphi/dt = -\cos\varphi \cdot d\varphi/dt \cdot \mathbf{i} - \sin\varphi \cdot d\varphi/dt \cdot \mathbf{j} = -d\varphi/dt \cdot \mathbf{e}_r \quad (3)$$

Logo, podemos escrever a velocidade da partícula como $\mathbf{v} = \mathbf{v}_r + \mathbf{v}_\varphi$, ou seja

$$\mathbf{v} = dr/dt \cdot \mathbf{e}_r + r \cdot d\varphi/dt \cdot \mathbf{e}_\varphi \quad (4)$$

A primeira parte dessa equação é denominada *velocidade radial*. (A componente- r de \mathbf{v} é simplesmente a taxa em que o vetor \mathbf{r} se alonga.)

A segunda parte é um vetor perpendicular a \mathbf{r} e é devida à variação da direção de \mathbf{r} , por exemplo, uma rotação da partícula em torno de O .

Essa parte é denominada *velocidade transversa*. $\omega := d\varphi/dt$ a velocidade angular.

No *movimento circular* não há velocidade radial porque o raio é constante, isto é, $dr/dt = 0$, sendo, então, a velocidade inteiramente transversa: $v = \omega \cdot r$.

Agora diferenciamos a expressão de \mathbf{v} para obtermos a aceleração \mathbf{a} .

Observe que a derivada de $r \cdot d\varphi/dt \cdot \mathbf{e}_\varphi$ produzirá três termos, já que todos os três fatores são funções do tempo.

Assim,

$$\mathbf{a} = d\mathbf{v}/dt = (d^2r/dt^2 \cdot \mathbf{e}_r + dr/dt \cdot d\mathbf{e}_r/dt) + (dr/dt \cdot d\varphi/dt \cdot \mathbf{e}_\varphi + r \cdot d^2\varphi/dt^2 \cdot \mathbf{e}_\varphi + r \cdot d\varphi/dt \cdot d\mathbf{e}_\varphi/dt)$$

Substituindo $d\mathbf{e}_r/dt$ e $d\mathbf{e}_\varphi/dt$ da Eq. (3) e reunindo termos, temos

$$\mathbf{a} = \mathbf{a}_r + \mathbf{a}_\varphi = a_r \cdot \mathbf{e}_r + a_\varphi \cdot \mathbf{e}_\varphi, \text{ ou seja}$$

$$\mathbf{a} = (d^2r/dt^2 - r \cdot (d\varphi/dt)^2) \cdot \mathbf{e}_r + (r \cdot d^2\varphi/dt^2 + 2 \cdot dr/dt \cdot d\varphi/dt) \cdot \mathbf{e}_\varphi \quad (5)$$

onde $a = (a_r^2 + a_\varphi^2)^{1/2}$.

No caso do *movimento circular* não há nem dr/dt nem d^2r/dt^2 porque o raio é constante, isto é

$$\mathbf{a} = \mathbf{a}_r + \mathbf{a}_\varphi = -R \cdot \omega^2 \cdot \mathbf{e}_r + R \cdot \alpha \cdot \mathbf{e}_\varphi \quad (6)$$

R = raio de círculo (é igual a $r \cdot \sin\beta$)

$\omega := d\varphi/dt$ = velocidade angular em rad/s ou simplesmente s^{-1}

$\alpha := d\omega/dt = d^2\varphi/dt^2$ = aceleração angular em rad/s^2 ou simplesmente s^{-2}

$-R \cdot \omega^2 \cdot \mathbf{e}_r$ = aceleração centrípeta

$R \cdot \alpha \cdot \mathbf{e}_\varphi$ = aceleração tangencial

Note que, no movimento circular uniforme (sem aceleração angular, $\alpha := 0$) não há aceleração tangencial, mas sim aceleração centrípeta devido à variação de direção da velocidade.

A velocidade angular pode ser expressa como uma grandeza vetorial de direção perpendicular ao plano do movimento e de sentido coincidente com o do avanço de um saca-rolhas, com rosca à direita, que gira no mesmo sentido do movimento da partícula. (Ver a seguinte figura feita para o caso geral de três dimensões.)

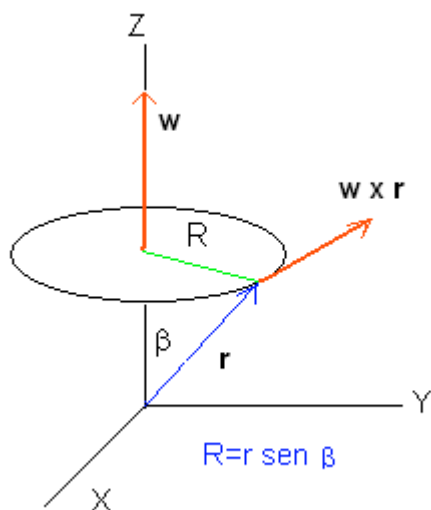


Fig. 3.4-25

Na figura (3.4-22) vemos que $R = r \sin\beta$ e que $\boldsymbol{\omega} = d\varphi/dt \cdot \mathbf{k}$.

Em lugar da equação $v = \omega R$, podemos escrever, $v = \omega r \sin\beta$, o que seria o módulo

da equação vetorial

$$\mathbf{v} = \boldsymbol{\omega} \times \mathbf{r} \quad (7).$$

Esta relação, que define o vetor $\boldsymbol{\omega}$, é válida somente para o movimento circular geral com r e β constantes. No caso especial de $\omega = \text{constante}$, temos o movimento circular uniforme.

3.4.10 Coordenadas cilíndricas e esféricas

É comum o uso de dois outros sistemas de coordenadas no espaço: as coordenadas cilíndricas e as coordenadas esféricas.

As coordenadas *cilíndricas* são obtidas trocando-se duas das três coordenadas retangulares pelas coordenadas polares correspondentes, por exemplo substituir (x,y) por (r,ϕ) e manter a coordenada z . As coordenadas cilíndricas da partícula são (r,ϕ,z) . As equações de transformação entre coordenadas retangulares e cilíndricas são

$$x = r \cos\phi, \quad y = r \sin\phi, \quad z = z \quad (8)$$

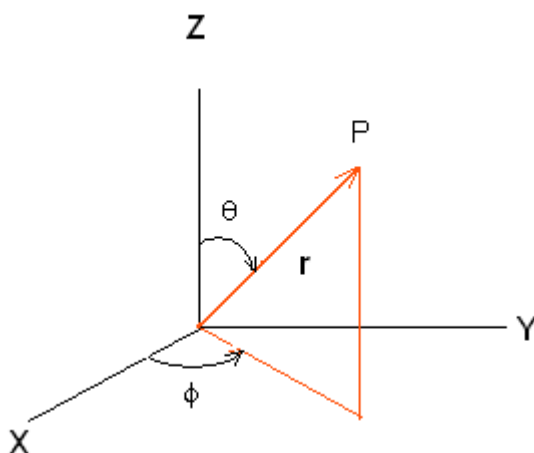


Fig. 3.4-26

Nas coordenadas *esféricas*, usa-se o ângulo φ como coordenada polar num plano coordenado, a distância da origem O ao ponto P, e o ângulo θ entre OP e o terceiro eixo. As relações entre as coordenadas retangulares e esféricas são

$$x = r \sin\theta \cos\varphi, \quad y = r \sin\theta \sin\varphi, \quad z = r \cos\theta \quad (9)$$

$$0 \leq \varphi < 2\pi, \quad 0 \leq \theta < \pi$$

Levando ao quadrado cada uma das relações em (9) e somando, obtemos

$x^2 + y^2 + z^2 = r^2$ o que é a equação *implícita* de uma esfera de raio r e centro na origem.

Observe que a coordenada- z depende somente do ângulo θ . Geometricamente isto significa que todos os pontos com a mesma *latitude* θ têm a mesma coordenada $-z$

3.4.11 Aplicações

A biblioteca `linalg` de MuPAD tem a função `linalg::ogCoordTab` com a qual podem-se efetuar os cálculos para transformar entre diferentes coordenadas.

(O nome significa: *table of orthogonal coordinate transformation*.)

Com a seguinte instrução MuPAD obtém as equações de transformação *esféricas* -> *cartesianas*:

```
linalg::ogCoordTab[Spherical, Transformation](r, phi, theta)
```

```
[r cos(phi) sin(theta), r sin(phi) sin(theta), r cos(theta)]
```

Para a transformação inversa, utiliza MuPAD

```
linalg::ogCoordTab[Spherical, InverseTransformation](x,y,z)
```

No caso de coordenadas cilíndricas, temos:

- `linalg::ogCoordTab[Cylindrical, Transformation](r, phi, z)`

```
[r cos(phi), r sin(phi), z]
```

- `linalg::ogCoordTab[Cylindrical, InverseTransformation](x,y,z)`

```
[sqrt(x^2+y^2), arccos(x/sqrt(x^2+y^2)) - sign(y) * (arccos(x/sqrt(x^2+y^2)) - pi) * (sign(y) - 1), z]
```


Aqui temos um exemplo numérico para transformar as coordenadas cilíndricas $r = 8$, $\varphi = 2\pi/3$, $z = -3$ em coordenadas cartesianas:

- `linalg::ogCoordTab[Cylindrical,Transformation](8,2*PI/3,-3)`

$$[-4, 4 \cdot \sqrt{3}, -3]$$

Para a transformação inversa, ou seja $x = -4$, $y = 4 \cdot 3^{1/2}$, $z = -3$ em (r, φ, z) , temos

- `linalg::ogCoordTab[Cylindrical,InverseTransformation](-4,4*sqrt(3),-3)`

$$\left[8, \frac{2 \cdot \pi}{3}, -3\right]$$

- Quais são as coordenadas cartesianas pertencentes às coordenadas esféricas $(1, \pi/4, -\pi/2)$?

`linalg::ogCoordTab[Spherical, Transformation](1,-PI/2,PI/4)`

$$\left[0, -\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right]$$

Então, $x = 0$, $y = -1/2^{1/2}$, $z = 1/2^{1/2}$

Agora temos em coordenadas cartesianas a área $x \cdot z = 1$. Qual será sua forma em coordenadas esféricas?

Primeiramente pedimos as equações de transformação:

- `reset() :`

`ar:=x*z=1 :`

`linalg::ogCoordTab[Spherical, Transformation](r, phi, theta) :`

`[r cos(phi) sin(theta), r sin(phi) sin(theta), r cos(theta)]`

Logo, trocamos as coordenadas:

`area:=subs(ar,x=%[1],z=%[3]);`

Resultado:

$$r^2 \cos(\phi) \cos(\theta) \sin(\theta) = 1$$

Agora, aplicamos **combine**, para simplificar o termo $\cos(\theta) * \sin(\theta)$

- **combine(cos(theta)*sin(theta), sincos)**

$$\frac{\sin(2 \cdot \theta)}{2}$$

Temos, então, $r^2 \cos(\phi) \sin(2 \cdot \theta)/2 = 1$. Nossa área $x \cdot z = 1$ tem em coordenadas esféricas a forma: $r^2 \cos\phi \cdot \sin(2\theta) = 2$.

Fazemos o mesmo para a área $x^2 + y^2 - z^2 = 1$:

reset() :

ar:=x^2+y^2-z^2=1 :

linalg::ogCoordTab[Spherical, Transformation](r, phi, theta) :

area:=subs(ar, x=%[1], y=%[2], z=%[3]) ;

- **combine(%, sincos)**

$$-r^2 \cdot \cos(2 \cdot \theta) = 1$$

Resultado: A área $x^2 + y^2 - z^2 = 1$ tem em coordenadas esféricas a forma

$$-r^2 \cdot \cos(2\theta) = 1.$$

Calculemos, finalmente, a seguinte **integral tripla**

$$I_{zz} = \int_V (x^2 + y^2) \rho dV$$

Trata-se do *momento de inércia* em torno do eixo-z. A densidade ρ tomamos igual a um e a integração se faz sobre a esfera com raio R .

(O elemento de volume é $dV = dx dy dz$. Em coordenadas esféricas é $dV = r^2 \sin\theta \cdot dr \, d\phi \, d\theta$. O fator de transformação do elemento de volume é $r^2 \sin\theta$. Em coordenadas cilíndricas temos $dV = r \cdot dr \, d\phi \, dz$.)

```
fz:=x^2 + y^2://para fzz:=1 obtemos o volume
linalg::ogCoordTab[Spherical, Transformation](r, phi,
theta):
fzz:=subs(fz,x=%[1],y=%[2],z=%[3]):
factor(%);
subs(%,cos(phi)^2+sin(phi)^2=1)
```

$$\sin(\theta)^2 \cdot r^2 \cdot (\cos(\phi)^2 + \sin(\phi)^2)$$

$$1 \cdot \sin(\theta)^2 \cdot r^2$$

Para calcular a *integral tripla*, precisamos do valor absoluto do *Jacobiano* da transformação. Para coordenadas esféricas, o Jacobiano é $-r^2 \cdot \sin(\theta)$, para cilíndricas simplesmente r . Ver "Com lápis e papel".

- `jacob:=r^2*sin(theta)://valor absoluto do Jacobiano`
- ```
int(int(int(r^2*sin(theta)^2*jacob, r = 0..R),
theta=0..PI),
phi= 0..2*PI)
```

$$\frac{8 \cdot \pi \cdot R^5}{15}$$

### 3.4.12 Com lápis e papel

#### Integrais duplas e triplas

Acabamos de ver um exemplo de uma integral tripla  $I_{zz} = \int_V (x^2 + y^2) \rho dV$

Normalmente calculamos integrais duplas e triplas usando integrais ordinárias em uma variável. O cálculo das integrais se faz por meio da integração repetida. Se  $f(x,y)$  é uma função contínua de duas variáveis e  $R$  um retângulo  $a \leq x \leq b$ ,  $c \leq y \leq d$ , então

$$\int_R f dA = \int_c^d \left( \int_a^b f(x,y) dx \right) dy := \int_c^d \int_a^b f(x,y) dx dy \quad (10)$$

é chamada uma integral *iterada*.

A integral de dentro é feita em relação a  $x$ , mantendo  $y$  constante, e depois o resultado é integrado em relação a  $y$ .

Para qualquer função que tenhamos alguma probabilidade de encontrar, não importa em qual ordem integramos sobre uma região retangular  $R$ ; de qualquer modo encontramos o mesmo valor para a integral dupla. Usando a notação intuitiva de MuPAD calculamos o volume da cabana da Figura 3.4-22 da seção 3.4.7:

- `int(int(12-x/4-y/8, x = 0..8), y=0..16)`

1280

- `int(int(12-x/4-y/8, y = 0..16), x=0..8)`

1280

Mudando a ordem da integração não teve nenhum efeito sobre o resultado de  $1280 \text{ m}^3$ .

Alias, o volume da cabana podemos calcular, também, usando uma *integral tripla*:

- `int(int(int(1, z = 0..12-x/4-y/8), y=0..16), x=0..8)`

1280

**Observa:**

1. Os limites para a integral externa são constantes.
2. Os limites para a integral do meio só podem envolver uma variável (a da integral externa).
3. Os limites para a integral interior podem envolver duas variáveis (as das duas integrais externas).

No caso da integral dupla, devem ser constantes os limites na integral exterior. Se a integral interior é com relação a  $x$ , seus limites devem ser constantes ou expressões em termos de  $y$ , e vice-versa.

A região  $R$  não tem de ser, em todos os casos, retangular, como mostra o seguinte

**Exemplo:**

Ache a massa  $M$  de uma placa metálica  $R$  limitada pela reta  $y = 2x$  e pela parábola  $y = x^2$ , com densidade dada por  $\rho(x,y) = x \cdot y^2 \text{ kg/metro}^2$ . A massa é dada por

$$M = \int_R \rho(x,y) \cdot dA \quad (11)$$

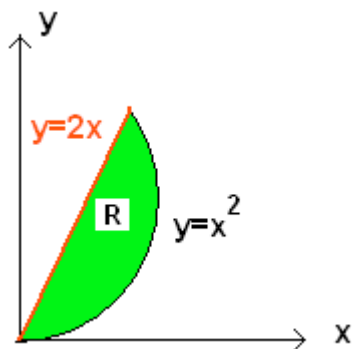


Fig.3.4-27

Primeiro fazemos a integral em  $y$ , que vai da fronteira inferior  $y = x^2$  até a fronteira superior  $y = 2x$ . A borda esquerda da região está em  $x = 0$  e a direita na interseção de  $y = 2x$  e  $y = x^2$ , que é  $(2,4)$ . Assim, a coordenada- $x$  das tiras verticais pode variar de  $x = 0$  a  $x = 2$ , de modo que a massa é dada por

$$M = \int_0^2 \int_{x^2}^{2x} xy^2 dy dx$$

Primeiro calculamos a integral interior considerando  $x$  constante:

$$\begin{aligned} \int_{x^2}^{2x} xy^2 dy &= x \frac{y^3}{3} \Big|_{y=x^2}^{y=2x} \\ &= \frac{1}{3} \cdot (8x^4 - x^7) \end{aligned}$$

Obtemos finalmente:

$$M = \frac{1}{3} \int_0^2 (8x^4 - x^7) dx = \frac{32}{5} = 6 \frac{2}{5} \cdot \text{kg}$$

O cálculo de MuPAD dá o mesmo resultado, a saber:

- `int(int(x*y^2, y = x^2..2*x), x=0..2)`

$$\frac{32}{5}$$

Em coordenadas cartesianas, o elemento de área foi  $dA = dx \cdot dy$ . Para integrar em coordenadas polares, temos que exprimir  $dA$  em termos das coordenadas polares  $(r, \varphi)$ . Podemos escrever  $dA = ds \cdot dr = r \cdot d\varphi \cdot dr$ .

Para calcular uma integral tripla em coordenadas cilíndricas, temos que exprimir o elemento de volume em coordenadas cilíndricas:  $dV = dA \cdot dz = ds \cdot dr \cdot dz = r \cdot d\varphi \cdot dr \cdot dz$ , ou seja  $dV = r \cdot dr \cdot d\varphi \cdot dz$ . Outras ordens de integração também são possíveis.

Para usar coordenadas esféricas em integrais triplas, precisamos expressar o elemento de volume  $dV$  em coordenadas esféricas. É fácil demonstrar que  $dV = r^2 \sin\theta \cdot dr \cdot d\varphi \cdot d\theta$ .

As fórmulas obtidas são casos especiais da teoria das transformações de variáveis em integrais múltiplas. A transformação dos elementos  $dA$  e  $dV$  se faz por meio de um determinante especial, chamado determinante de Jacobi ou simplesmente o Jacobiano. Para detalhes ver: Louis Leithold, *O Cálculo com Geometria Analítica*, vol.II (Carl Gustav Jakob Jacobi, matemático alemão, 1804-1851.)

## Algumas novas funções do MuPAD

Para conhecer mais algumas instruções do MuPAD, vamos demonstrar alguns teoremas sobre vetores.

1. Mostre que  $(\mathbf{r} \times d\mathbf{r}/dt) \times \mathbf{r} = r^3 d\mathbf{e}_r/dt$  com  $\mathbf{r} := (x, y, z)$  e  $r := |\mathbf{r}|$

Para simplificar a notação, definimos com `mat:=Dom::Matrix()` a variável `mat` como sendo uma matriz. A instrução `export(linalg)` faz que em `linalg::crossProduct` podemos prescindir da introdução `linalg::`. Com `bool(esq=dir)` determinamos, se o lado esquerdo é igual ao lado direito. A variável `bool` analisa uma expressão lógica e determina se é "true" ou "false". Com `expand` expandimos (multiplicamos) os termos do resultado anterior, sem isto, a função `bool` não pode, em geral, fazer corretamente o análise.

- `reset()` :

```
mat:=Dom::Matrix(): export(linalg):
```

```
vr:=mat([[x(t), y(t), z(t)]])://vetor r
```

```
r:=sqrt(vr[1]^2+vr[2]^2+vr[3]^2)://módulo de vr
```

```

v:=diff(vr,t):
er:=vr/r:
direito:=r^3*diff(er,t):
dir:=expand(%):
esquerda:=crossProduct(crossProduct(vr,v),vr):
esq:=expand(%):
bool(esq=dir)
TRUE

```

Dados três vetores  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{c}$  do  $\mathbb{R}^3$ . Mostre as seguintes relações:

- $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \mathbf{b} \cdot (\mathbf{c} \times \mathbf{a}) = \mathbf{c} \cdot (\mathbf{a} \times \mathbf{b})$
- $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \mathbf{b}(\mathbf{a} \cdot \mathbf{c}) - \mathbf{c}(\mathbf{a} \cdot \mathbf{b})$  (12)

a.

```

• reset():
assume(Type::Real):
mat:=Dom::Matrix(): export(linalg):
a:=mat([[a1,a2,a3]])://vetor a
b:=mat([[b1,b2,b3]])://vetor b
c:=mat([[c1,c2,c3]])://vetor c
esquerda:=scalarProduct(a,crossProduct(b,c));
direito:=scalarProduct(b,crossProduct(c,a));
bool(esquerda=direito)
a1 b2 c3 - a1 b3 c2 - a2 b1 c3 + a2 b3 c1 + a3 b1 c2 - a3 b2 c1
a1 b2 c3 - a1 b3 c2 - a2 b1 c3 + a2 b3 c1 + a3 b1 c2 - a3 b2 c1
TRUE

```



b.

- `reset()` :

```
assume (Type::Real) :
```

```
mat:=Dom::Matrix() : export(linalg) :
```

```
a:=mat([[a1,a2,a3]])://vetor a
```

```
b:=mat([[b1,b2,b3]])://vetor b
```

```
c:=mat([[c1,c2,c3]])://vetor c
```

```
esquerda:=crossProduct(a,crossProduct(b,c)) :
```

```
esq:=expand(%) ;
```

```
direito:=b*scalarProduct(a,c)-c*scalarProduct(a,b) :
```

```
dir:=expand(%) ;
```

```
bool(esq=dir)
```

```
array(1..1, 1..3,
 (1, 1) = a2 b1 c2 - a2 b2 c1 + a3 b1 c3 - a3 b3 c1,
 (1, 2) = - a1 b1 c2 + a1 b2 c1 + a3 b2 c3 - a3 b3 c2,
 (1, 3) = - a1 b1 c3 + a1 b3 c1 - a2 b2 c3 + a2 b3 c2
)
array(1..1, 1..3,
 (1, 1) = a2 b1 c2 - a2 b2 c1 + a3 b1 c3 - a3 b3 c1,
 (1, 2) = - a1 b1 c2 + a1 b2 c1 + a3 b2 c3 - a3 b3 c2,
 (1, 3) = - a1 b1 c3 + a1 b3 c1 - a2 b2 c3 + a2 b3 c2
)
```

TRUE

