Capítulo 12

Gráficos com 2007, Parte II

Representações 2D e curvas paramétricas

Curvas de Lissajous são curvas paramétricas definidas pelas equações

$$x = a\ sen(\omega_1 t)$$
$$y = b\ sen(\omega_2 t + \varphi)$$

que foram descobertas em 1857 por Jules Antoine Lissajous, físico francês.
Uma curva de Lissajous pode ser observada facilmente na tela de um osciloscópio, colocando a componente x no canal horizontal e a componente y no canal vertical.
Só podemos ver curvas fechadas quando a razão das freqüências é um número racional, ou seja, quando ω_1 e ω_2 não possuem divisor comum. Neste caso, temos $\omega_1:\omega_2 = n_1:n_2$ onde os números n_1 e n_2 são inteiros e não possuem divisor comum.
Se a razão das freqüências angulares é irracional, resultam oscilações não periódicas.
Se ω1:ω2 não for um número racional, então a curva será "aberta" e, após um longo tempo, o ponto que traça a curva terá passado por todos os pontos do retângulo limitado por x = ± a e y = ± b. Ele nunca passará duas vezes por um dado ponto com a mesma velocidade.

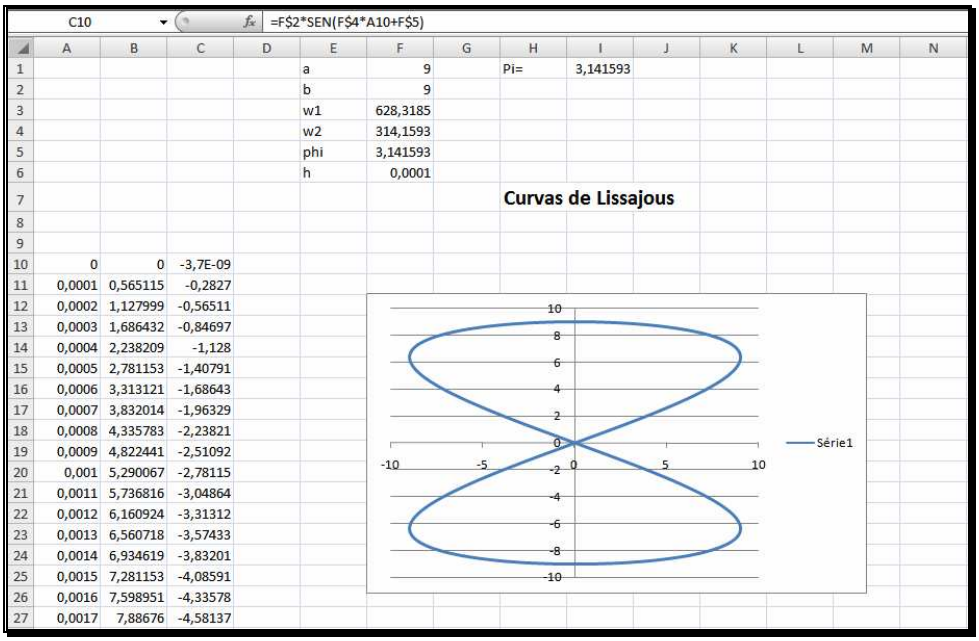
Para a *primeira* figura foram elegidos os seguintes parâmetros: a = b = 9; ω_1 = 200π; $\omega_2 = \omega_1/2$ e φ = π e h = 0,0001.

Entradas para a planilha:

1. As amplitudes estão em F1 e F2. ω_1 em F3 e ω_2 em F4. φ fica em F5.

2. O incremento de tempo h = 0,0001 está em F6. Visto que utilizaremos 300 pontos, t vai variar entre 0 e 0,03.
3. Na linha 10 começaremos com os cálculos:
   A10: 0
   B10: =F$1*SEN(F$3*A10) (=x)
   C10: =F$2*SEN(F$4*A10+F$5) (=y)
   A11: =A10+F$6 (t = t+h)
4. Copiar tudo até a linha 310

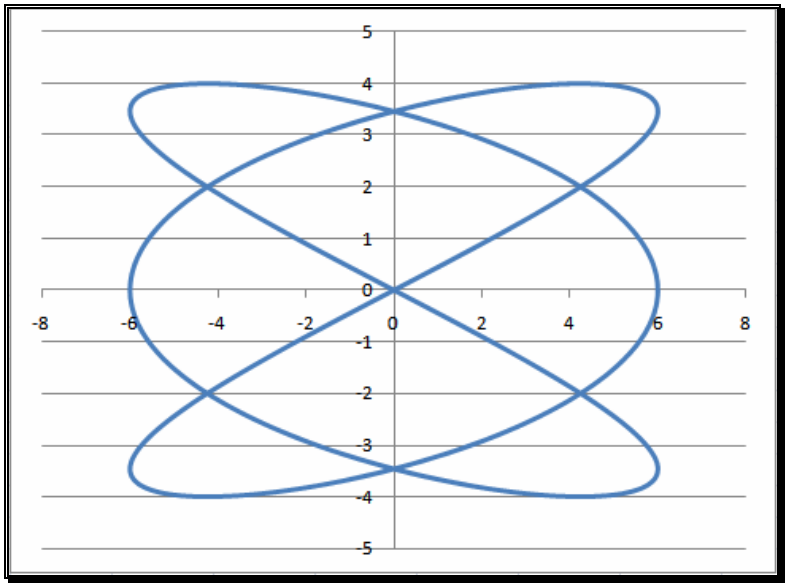
Para desenhar os dados, selecionamos o intervalo B10:C310.



C10 =F$2*SEN(F$4*A10+F$5)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1					a	9		Pi=	3,141593					
2					b	9								
3					w1	628,3185								
4					w2	314,1593								
5					phi	3,141593								
6					h	0,0001								
7								Curvas de Lissajous						
8														
9														
10	0	0	-3,7E-09											
11	0,0001	0,565115	-0,2827											
12	0,0002	1,127999	-0,56511											
13	0,0003	1,686432	-0,84697											
14	0,0004	2,238209	-1,128											
15	0,0005	2,781153	-1,40791											
16	0,0006	3,313121	-1,68643											
17	0,0007	3,832014	-1,96329											
18	0,0008	4,335783	-2,23821											
19	0,0009	4,822441	-2,51092											
20	0,001	5,290067	-2,78115											
21	0,0011	5,736816	-3,04864											
22	0,0012	6,160924	-3,31312											
23	0,0013	6,560718	-3,57433											
24	0,0014	6,934619	-3,83201											
25	0,0015	7,281153	-4,08591											
26	0,0016	7,598951	-4,33578											
27	0,0017	7,88676	-4,58137											

Agora podemos usar nossa planilha à vontade, é só necessário variar os parâmetros.
Por exemplo: a = 6; b = 4; ω_1 = 3; ω_2 = 2; φ = 0 e h = 0,05 produzem a seguinte figura



Interessantes curvas resultam com só mudar do valor de φ. Por exemplo π/8 e π/2.

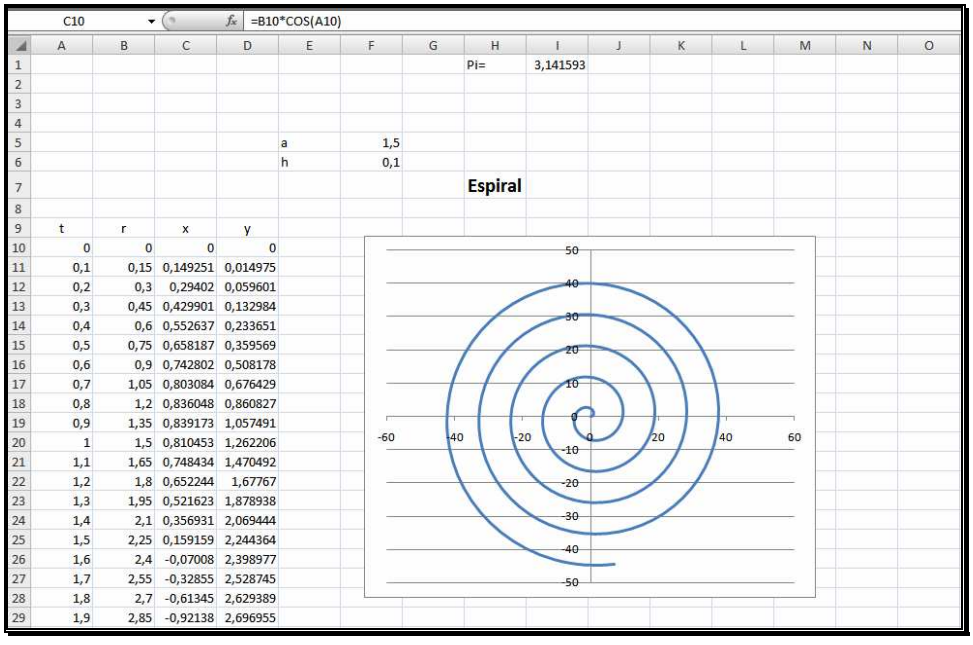
A espiral

As **espirais** continuam a ser um mistério para a ciência. Pense nas Galáxias e nas outras espirais mais pequenas da natureza. Bem conhecidas são as espirais dos caracoles e dos girassóis. Mas, para a vida a mais importante forma espiral é a DNA, duas faixas paralelas espiraladas, batizada por dupla hélice por James Watson e Francis Crick.

Um exemplo famoso na Física é o cíclotron. (Quando uma partícula carregada em movimento uniforme penetra um campo magnético, também uniforme, perpendicular à direção de seu movimento, as forças que atuam fazem com que a nova trajetória da partícula seja circular, -veja a seção anterior. Assim, se o campo elétrico presente nos "dês" estiver em ressonância com a revolução das partículas, essa é acelerada a cada travessia do intervalo entre os "dês".)
Para obter as equações que descrevem uma espiral, partimos das equações paramétricas de um círculo

$$x = r\cos(t)$$
$$y = rsen(t);\ \ -\infty < t < \infty$$

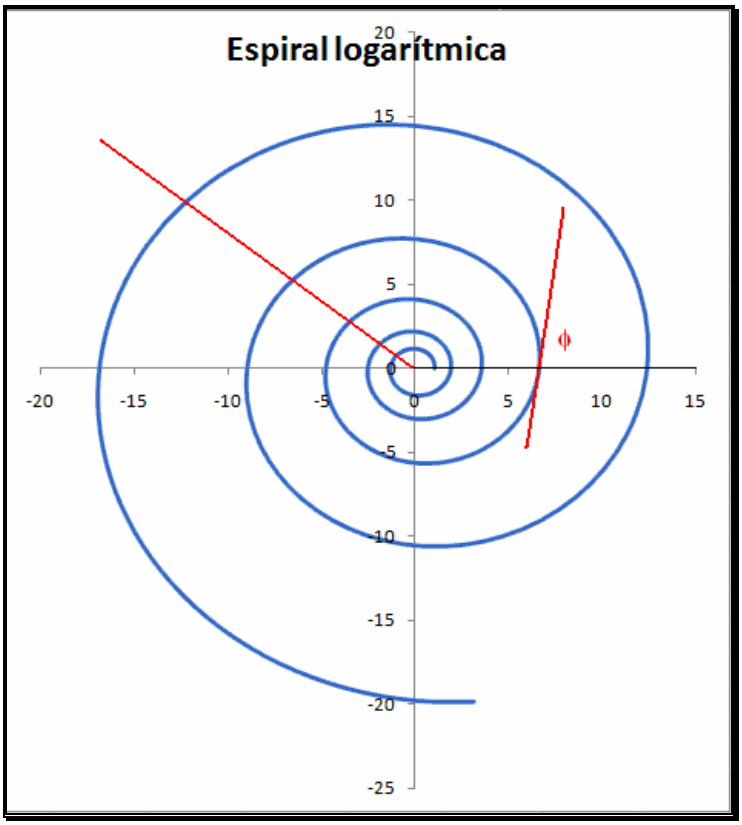
Vamos obter espirais fazendo depender o raio em forma apropriada do parâmetro t. Obtém-se as *espirais arquimédicas* por meio da tentativa $r^m = a^m t$. A mais conhecida resulta com m = 1, ou seja *r = at*. A figura a seguir foi desenhada com a = 1,5. A coluna B contém os valores de r, por exemplo: B10: =F$5*A10.



C10 =B10*COS(A10)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1								Pi=	3,141593						
2															
3															
4															
5					a	1,5									
6					h	0,1									
7								Espiral							
8															
9	t	r	x	y											
10	0	0	0	0											
11	0,1	0,15	0,149251	0,014975											
12	0,2	0,3	0,29402	0,059601											
13	0,3	0,45	0,429901	0,132984											
14	0,4	0,6	0,552637	0,233651											
15	0,5	0,75	0,658187	0,359569											
16	0,6	0,9	0,742802	0,508178											
17	0,7	1,05	0,803084	0,676429											
18	0,8	1,2	0,836048	0,860827											
19	0,9	1,35	0,839173	1,057491											
20	1	1,5	0,810453	1,262206											
21	1,1	1,65	0,748434	1,470492											
22	1,2	1,8	0,652244	1,67767											
23	1,3	1,95	0,521623	1,878938											
24	1,4	2,1	0,356931	2,069444											
25	1,5	2,25	0,159159	2,244364											
26	1,6	2,4	-0,07008	2,398977											
27	1,7	2,55	-0,32855	2,528745											
28	1,8	2,7	-0,61345	2,629389											
29	1,9	2,85	-0,92138	2,696955											

A *espiral logarítmica* com $r = e^{at}$ era a favorita de Jakob Bernoulli (1654-1705), quem fez o primeiro uso extenso das coordenadas polares. (Na época de Bernoulli a função exponencial ainda não era considerada como uma função independente, pois o número *e* nem tinha ainda um símbolo especial, e Jakob utilizou a equação $\ln r = a\theta$. Assim, explica-se o nome de espiral logarítmica.)
A nossa espiral logarítmica tem a = 0,1:



Duas propriedades da espiral logarítmica são especialmente interessantes:

1. Cada raio que passa pela origem atravessa a espiral com mesmo ângulo.
2. O comprimento do arco de qualquer ponto da espiral logarítmica até o centro é finito, embora sejam necessárias infinitas rotações para se chegar ao centro.

Para o ângulo vale a fórmula $a = \mathrm{ctg}(\varphi)$. Utilizando a relação $\mathrm{arc\,ctg}(x) = \mathrm{arc\,tg}(1/x)$, obteremos $\varphi = \mathrm{arc\,tg}(1/0{,}1) = 1{,}4711$ Rad ou $\varphi = 84{,}29^{o}$.

Isso significa que a espiral corta o eixo X sempre sob $84{,}29^{o}$, pois o eixo X é também um raio pela origem.

O ciclóide

Em 1696 Johann Bernoulli, o irmão mais velho de Jakob, propus um problema de mecânica: encontrar a curva ao longo da qual uma partícula deslizará sob a força da gravidade no tempo mais curto possível. (Este famoso problema é conhecido como o problema da braquistócrona, *tempo mais curto*.) Cinco soluções corretas foram apresentadas: por Newton, Leibniz, L'Hospital e pelos dois irmãos Bernoulli.

A curva pedida revelou-se um **ciclóide**, a curva traçada por um ponto P na borda de uma roda, à medida que ela rola, sem escorregamento, por uma superfície horizontal.

Sobre o movimento ao longo de uma ciclóide compare o site

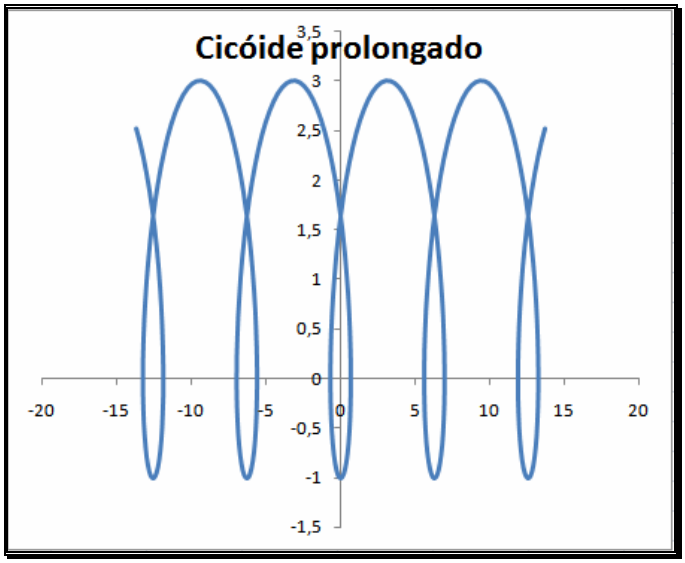
http://www.geocities.com/Athens/Agora/6594/Mechsub/mech3_3.pdf

(No caso geral, P pode também ficar no interior ou no exterior de um círculo. A distância do centro do círculo é chamado de b.)

As equações paramétricas do ciclóide são:

$$x = at - bsen(t)$$
$$y = a - b\cos(t); \quad -\infty < t < \infty$$

a é o raio do círculo, t = ângulo de rolagem.



Caso b > a

Devemos diferenciar o caso P sobre o circulo (a = b) do caso P no interior do circulo (b < a) e do caso de P no exterior do círculo (b > a). No gráfico utilizamos a = 1 (F4), b = 2 (F5), h = 0,1 (F6); o valor inicial para t é -15 (A10)

C10: =A10*F$4-F$5*SEN(A10); D10: =F$4-F$5*COS(A10), copiar até D310.

A ciclóide de Bernoulli tem a = b (=1); aqui pomos h=0,03.

Representações 3D e curvas paramétricas

Muitas vezes curvas em 2D são dadas por meio de equações paramétricas

$$(x(t),y(t)), \quad t \varepsilon [a,b]$$

Exemplos conhecidos são círculos, elipses e trajetórias de partículas carregadas (alfas, elétrons, prótons etc.) movendo-se em campos eletromagnéticos. Uma curva paramétrica em 3D é representada por (x(t),y(t),z(t)), t ε [a,b].

No princípio deste capítulo, vamos estudar a representação 3D de objetos geométricos, dados por pontos ou por equações paramétricas. Depois, consideraremos alguns exemplos tomados da Física.

Preparação

Se queremos desenhar um objeto geométrico perspectivamente, temos que fazer os seguintes três passos:

1. O corpo deve ser descrito num sistema tridimensional de coordenadas retangulares (X, Y, Z). Cada um dos seus pontos tem três coordenadas (x,y,z).
2. O corpo será girado sobre três eixos. As equações que descrevem as rotações dependem da ordem e do sentido em que as rotações são executadas. (Rotações não são comutativas! Ou seja, a ordem da aplicação dos fatores de rotação influencia no resultado final.) As coordenadas x,y,z se transformam em x',y',z'.
3. O corpo deve ser projetado sobre um plano de projeção (por exemplo a tela do computador) onde corresponde a cada ponto com as coordenadas "giradas" x',y',z' um ponto com as coordenadas x_s, y_s .

O eixo X aponta para a direita, o Y para cima, e o Z aponta para o observador. Primeiro giramos o corpo um ângulo beta (b) ao redor do eixo Y, depois um ângulo alfa (a) em torno do eixo X e, finalmente, um ângulo gama (c) à volta de Z.

As equações da transformação são

$$
\begin{aligned}
x' &= [\cos(c)\cos(b)-\text{sen}(b)\text{sen}(a)\text{sen}(c)]\cdot x \\
&\quad - [\cos(c)\text{sen}(b)+\text{sen}(a)\cos(b)\text{sen}(c)]\cdot z \\
&\quad + [\cos(a)\text{sen}(c)]\cdot y \\
y' &= [-\cos(b)\text{sen}(c)-\text{sen}(b)\text{sen}(a)\cos(c)]\cdot x \\
&\quad + [\text{sen}(b)\text{sen}(c)-\text{sen}(a)\cos(b)\cos(c)]\cdot z \\
&\quad + [\cos(a)\cos(c)]\cdot y \\
z' &= [\text{sen}(b)\cos(a)]\cdot x + [\cos(a)\cos(b)]\cdot z + [\text{sen}(a)]\cdot y
\end{aligned}
\qquad (1)
$$

A projeção em perspectiva é feita através de cálculos simples de "semelhança de triângulos". As equações para o cálculo da projeção são:

$$
\begin{aligned}
x_s &= xa + (xa - x')za/(z' - za) \\
y_s &= ya + (ya - y')za/(z' - za)
\end{aligned}
\qquad (2)
$$

As coordenadas (xa,ya,za) são as coordenadas no espaço (space coordenates) do centro de projeção ("ponto do olho").

Nós escolhemos como centro de projeção um ponto situado no eixo Z com za := D = distância do observador ao plano de projeção. Com esta simplificação, obtemos as seguintes formulas de projeção:

$$
\begin{aligned}
xs &= D\cdot x'/(D-z') \\
ys &= D\cdot y'/(D-z')
\end{aligned}
\qquad (3)
$$

A prática demonstra que ângulos de visada entre 40 e 60 graus proporcionam experiências visuais bastante próximas da realidade. Ao contrário, quando a distância do observador à tela aumenta, o seu campo de visão se estreita e a projeção em perspectiva não é muito pronunciada. Por meio da planilha que vamos desenvolver agora podemos confirmar estas observações.

Para não repetir o cálculo dos fatores de sen e cos permanentemente, os colocamos como valores fixos nas células J1 até J6. As três fórmulas em H22, F22, G22 são simples, mas extensos. Visto que este programa será de grande utilidade para todo este capítulo, devemos sentir-nos cheios de anseio para preencher as células com tudo o que segue. Não é assim?

1. J1: =SEN(B8*I1); J2: =COS(B8*I1); J3: =SEN(B9*I1);
   J4: =COS(B9*I1); J5: =SEN(B10*I1); J6: =SEN(B10*I1)
2. Em H22 inserimos a primeira das fórmulas (3)
   =J$3*J$2*B22+J$2*J$4*D22+J$1*C22

F22: =((J$6*J$4-J$3*J$1*J$5)*B22-(J$6*J$3+J$1*J$4*J$5)* D22 +J$2*J$5*C22)/(B$11-H22)*B$11+H$8+H$10 (= x)

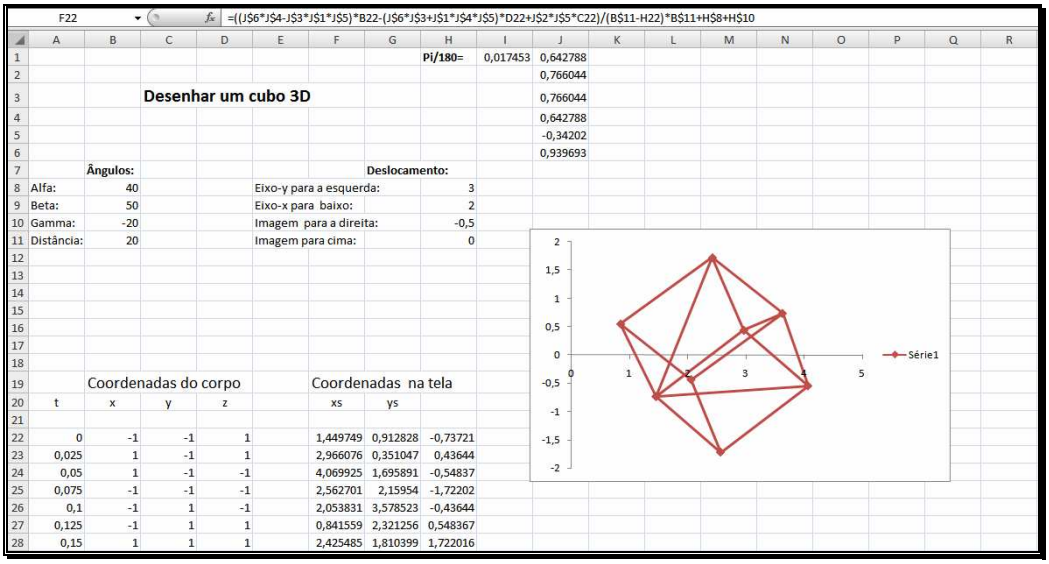
G22: =((-J$4*J$5-J$3*J$1*J$6)*B22+(J$3*J$5-J$1*J$4*J$6)*D22 +J$2*J$6*C22)/(B$11-H22)*B$11+H$9+H$11 (=y)

Copiamos todas as fórmulas até linha 38 (use F8, F5).

4. Em seguida devemos inserir as coordenadas de espaço das esquinas do cubo:

B22 até D22 : -1; -1; 1
B23 " D23 : 1; -1; 1
B24 " D24 : 1; -1; -1
B25 " D25 : -1; -1; -1
B26 " D26 : -1; 1; -1
B27 " D27 : -1; 1; 1
B28 " D28 : 1; 1; 1
B29 " D29 : 1; 1; -1
B30 " D30 : -1; 1; -1
B31 " D31 : -1; 1; 1
B32 " D32 : -1; -1; 1
B33 " D33 : 1; 1; 1
B34 " D34 : 1; -1; 1
B35 " D35 : 1; 1; -1
B36 " D36 : 1; -1; -1
B37 " D37 : -1; -1; 1
B38 " D38 : -1; -1; -1

Agora, é só selecionar as colunas F e G (F22:G38) e depois buscar *Inserir* > XY*Dispersão com Linhas Retas com Marcadores.*



Para poder obter na tela uma posição perfeita da figura, foram adicionadas nas fórmulas para x_s e y_s (F22, G22) dois constantes: F22: =H$8+H$10; G22: =H$9+H$11. É muito recomendável guardar a planilha, pois vamos utilizá-la, com poucas mudanças, nos próximos exemplos.

Desenhando uma flor

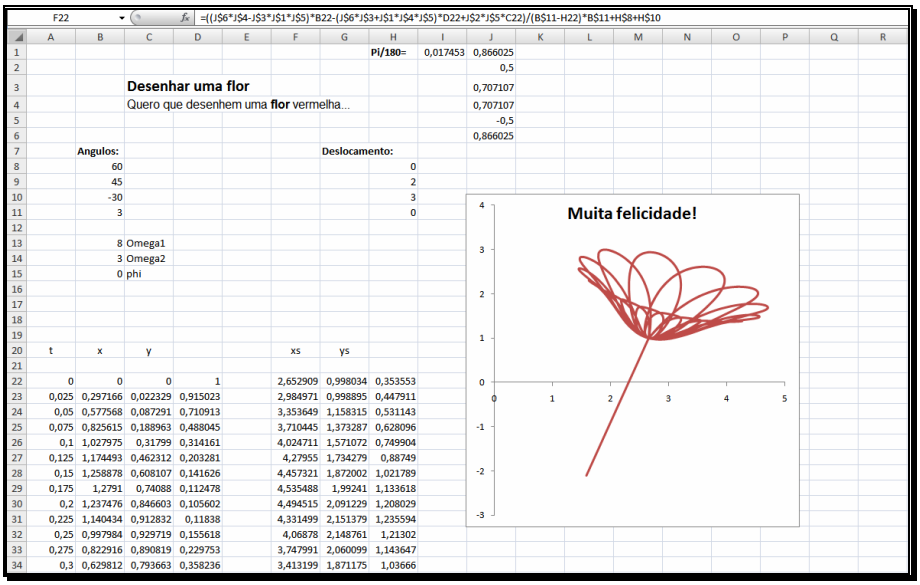
Na planilha anterior, foi preciso introduzir as coordenadas dos pontos "manualmente", o que foi bem cansativo.
Agora, tudo vai ser diferente, pois calcularemos as coordenadas apoiando-nos nas seguintes fórmulas:

$$x = asen(\omega_1 t)\cos(\omega_2 t)$$
$$y = bsen(\omega_1 t)sen(\omega_2 t + \varphi) \qquad (4)$$
$$z = ce^{-(x^2+y^2)}$$

Trata-se de **figuras de Lissajous** no espaço 3D.
Usamos a planilha do exemplo anterior com as mudanças necessárias:

1. B8: 60; B9: 45; B10: -30; B11: 3
   H8: 0; H9: 2; H10: 3; H11: 0
2. B13: 8 (= ω_1); B14: 3 (= ω_2); B15: 0 (= φ)
3. B22: =1,5*SEN(B$13*A22)*COS(B$14*A22) (= x)
   C22: =1,5*SEN(B$13*A22)*SEN(B$14*A22+B$15) (= y)
   D22: =EXP(-(B22^2+C22^2))
   A22: 0
4. Copiar as fórmulas em B22,C22,D22 (F5>D422>Shift>OK; Ctrl+d) até a linha 422. Em A23 temos =A22+0,025; copiar até A422.
5. Para desenhar o talo da flor, deixamos a linha 423 vazia e adicionamos B424: 0; C424: 0; D424: 1; B425: 0; C425: 0 e D425: 3
6. As equações para x_s e y_s , compare a planilha anterior, devem ser copiadas até a linha 425 (depois limpar F423:H423)
7. Selecionar F22:G425 e *Inserir >XY Dispersão Linhas Suaves ...*

Os valores de ω_1 e ω_2 determinam o número das pétalas e a forma da flor.
O ângulo de fase, φ, destrói geralmente a harmonia da forma.



Mostramos também as últimas linhas da planilha:

421	9,975	-0,11395	1,423667	0,130053		2,147699	2,888495	1,238625
422	10	-0,22996	1,47299	0,108329		1,960056	2,989147	1,232643
423								
424		0	0	1		2,652909	0,998034	0,353553
425		0	0	3		1,579065	-2,10189	1,06066

Partículas carregadas num campo eletromagnético

Um elétron num tubo de raios catódicos (= elétrons) sofre uma deflexão, se for aplicado um campo magnético B. Suponhamos que o elétron entre num campo magnético homogêneo de um ângulo α. A trajetória do elétron será uma hélice cilíndrica com distância característica, s, constante. O valor desta constante (= passo ou "pitch" da hélice) vem dado pala seguinte equação:

$$s = \frac{2\pi v sen(\alpha)}{\frac{e}{m}B} = Tvsen(\alpha)$$

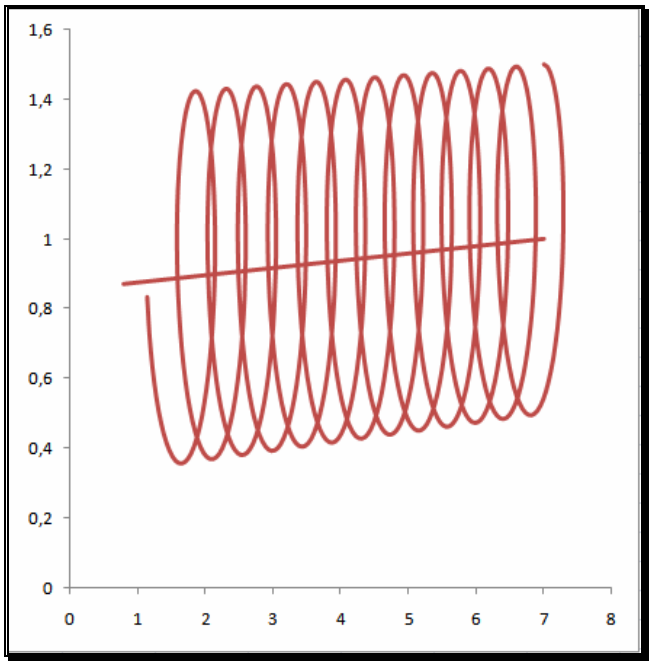
O fator sen(α) é uma constante e o tempo T para um círculo completo não depende do ângulo α. Para α = 0°, o elétron percorre um círculo em torno do campo magnético. (A trajetória completa de um elétron consiste da super-posição do movimento retilíneo uniforme paralelo ao eixo, e da revolução num plano perpendicular ao eixo.) v é a velocidade do elétron.

Em 1922, Busch desenvolveu um método para a determinação da carga específica de elétrons conhecido como "Método da Hélice". A carga específica do elétron é e/m = 1,7589· 10^{11} C/kg.

Por meio das seguintes equações paramétricas podemos representar este "movimento completo", fazendo uso do Excel:

$$x = rsen(\omega t)$$
$$y = r\cos(\omega t)$$
$$z = vsen(\alpha)t$$

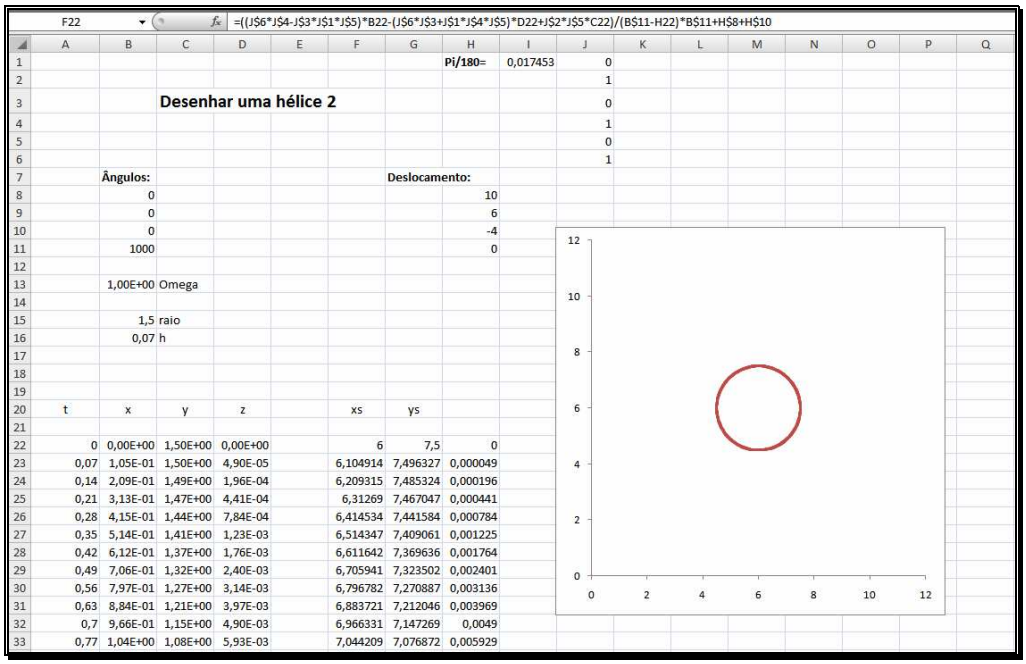
Trate de inserir os dados em nossa planilha que produzem a seguinte figura:



(Dados: ω = 1; v = 1; r = 0,5; sen(α) = 0,1; h = 0,2; ângulos: 1; 40; 0 e D = 100; deslocamentos 7, 1, 0, 0. v = 0 dá um círculo. O segmento do eixo Z foi desenhado a partir das seguintes entradas nas linhas 424 e 425: todos zero, salvo D425 = 9. Consulte também as explicações sobre o próximo caso com E e B paralelos.)

Quando um elétron se move na presença de campos **E** e **B** *paralelos*, sua trajetória consiste da superposição de um movimento circular uniforme num plano perpendicular aos campos e do movimento *acelerado* na direção dos campos.
Outra vez podemos fazer uso da planilha anterior com as devidas mudanças. A equação para z é, agora, $z = kt^2$, onde, nas figuras a seguir, a constante k foi tomada igual a 0,01. Tomamos r = 1,5 e ω = 1. O incremento do tempo é h = 0,07.

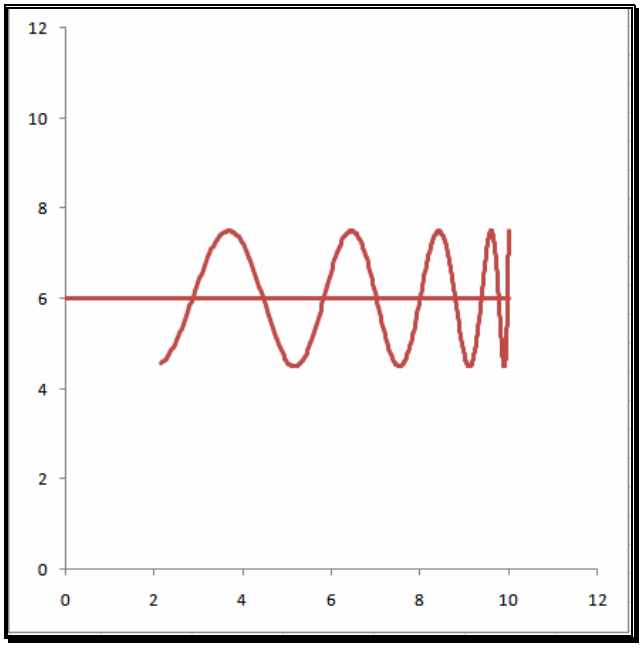
A *primeira* figura foi calculada com os ângulos 0, 0 (ou 180) e 0. A distância foi grande: D = 1000. Para os deslocamentos foram tomadas os valores 10, 6, -4, 0. A figura mostra a projeção da trajetória sobre o plano XY (um círculo).



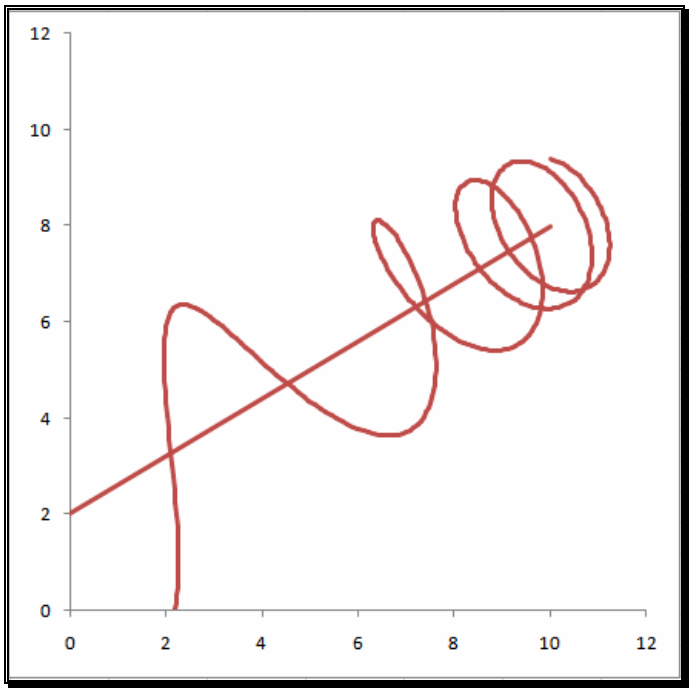
F22 =((J$6*J$4-J$3*J$1*J$5)*B22-(J$6*J$3+J$1*J$4*J$5)*D22+J$2*J$5*C22)/(B$11-H22)*B$11+H$8+H$10

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1								Pi/180=	0,017453	0							
2										1							
3			Desenhar uma hélice 2							0							
4										1							
5										0							
6										1							
7		Ângulos:					Deslocamento:										
8		0						10									
9		0						6									
10		0						-4									
11		1000						0									
12																	
13		1,00E+00	Omega														
14																	
15		1,5	raio														
16		0,07	h														
17																	
18																	
19																	
20	t	x	y	z		xs	ys										
21																	
22	0	0,00E+00	1,50E+00	0,00E+00		6	7,5	0									
23	0,07	1,05E-01	1,50E+00	4,90E-05		6,104914	7,496327	0,000049									
24	0,14	2,09E-01	1,49E+00	1,96E-04		6,209315	7,485324	0,000196									
25	0,21	3,13E-01	1,47E+00	4,41E-04		6,31269	7,467047	0,000441									
26	0,28	4,15E-01	1,44E+00	7,84E-04		6,414534	7,441584	0,000784									
27	0,35	5,14E-01	1,41E+00	1,23E-03		6,514347	7,409061	0,001225									
28	0,42	6,12E-01	1,37E+00	1,76E-03		6,611642	7,369636	0,001764									
29	0,49	7,06E-01	1,32E+00	2,40E-03		6,705941	7,323502	0,002401									
30	0,56	7,97E-01	1,27E+00	3,14E-03		6,796782	7,270887	0,003136									
31	0,63	8,84E-01	1,21E+00	3,97E-03		6,883721	7,212046	0,003969									
32	0,7	9,66E-01	1,15E+00	4,90E-03		6,966331	7,147269	0,0049									
33	0,77	1,04E+00	1,08E+00	5,93E-03		7,044209	7,076872	0,005929									

A *segunda* figura mostra uma vista lateral com 0, 90, 0 e D = 1000. Os deslocamentos são 10, 6, 0,0.



Finalmente, vemos, na *terceira* figura, uma representação perspectiva com 30, 40, 0 e D = 10. Os deslocamentos são: 10, 8, 0, 0:



O segmento do eixo Z foi desenhado a partir das seguintes entradas nas linhas 424 e 425.

420	27,86	6,04E-01	-1,37E+00	7,76E+00		1,296848	-0,37542	4,798902
421	27,93	5,06E-01	-1,41E+00	7,80E+00		1,185983	-0,33206	4,751089
422	28	4,06E-01	-1,44E+00	7,84E+00		1,069802	-0,28008	4,705419
423								
424		0	0	0		10	8	0
425		0	0	10		-9,09727	-3,37962	6,634139

Trabalhando com o clsMathParser

Dentre os **programas auxiliares** dedicados a VBA destaca-se o programa clsMathParser que foi desenvolvido por Leonardo Volpi, Foxes Team. O programa pode ser baixado gratuitamente do site
http://digilander.libero.it/foxes/mathparser/MathExpressionsParser.htm

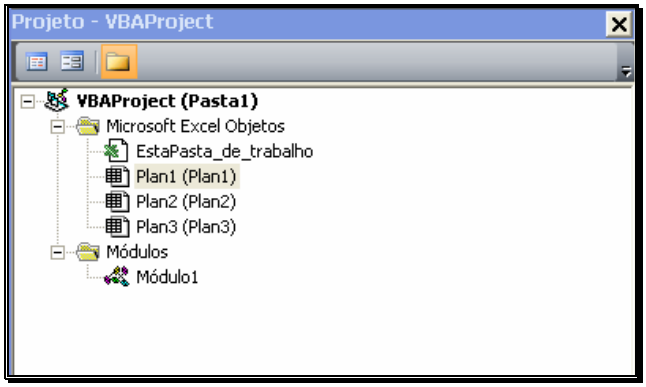
Este site oferece também um manual completo com exemplos e aplicações. O Parser permite que o usuário escreva as fórmulas na forma usual e é de grande ajuda na criação de gráficos usando Excel e VBA. Já estão outros programas no mercado que utilizam o clsMathParser como programa de fundo como, por exemplo, **tc**^{2}, http://www.tcquadrat.de/downloads/tc2-Handbuch.pdf que permite fazer cálculos escrevendo em Word.

Se você quer trabalhar com o parser, é preciso baixar o programa e descompactá-lo numa pasta, por exemplo em "clsMathParser". Nesta pasta encontram-se, além da documentação em pdf, dois arquivos: clsMathParser.cls e o arquivo mMath-SpecFun.bas.
O primeiro é o parser e o segundo é uma biblioteca com funções especiais já implementadas. (Se pode descompactar o arquivo com ExtractNow.)

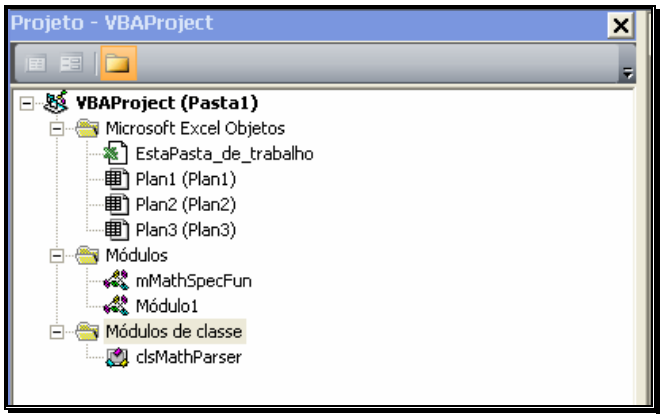
Para ver um **exemplo** da aplicação do parser, implementamos uma planilha do Excel:

Os passos a seguir são:

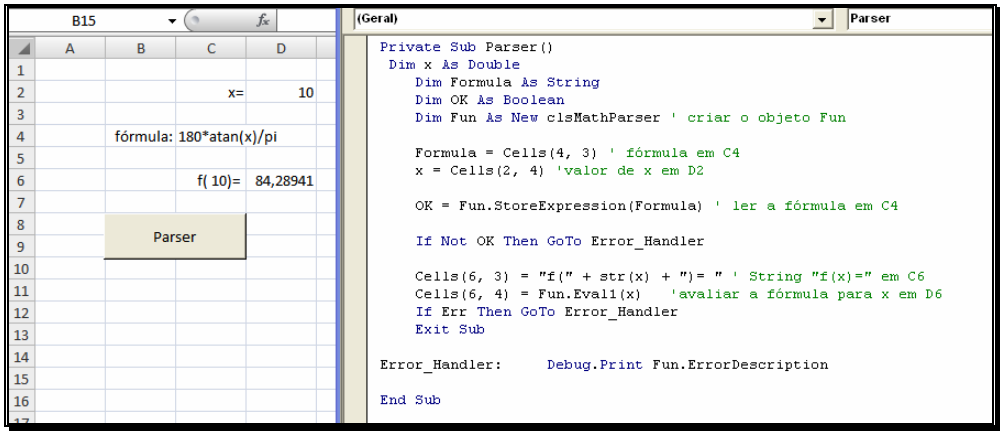
1. No editor de VBA selecionamos *Inserir>Módulos>Project Explorer.*



Com o botão direito da mouse fazemos click no *Plan1* e elegemos *Importar arquivo*.



Na pasta "clsMathParser" selecionamos clsMathParser. O segundo arquivo, o mMathSpecFun.bas, que se encontra na pasta "clsMathParser" instalamos da mesma maneira no Plan2. A última vista sobre VBAProject vai ter o aspecto mostrado. O parser aparece como Módulo de classe.

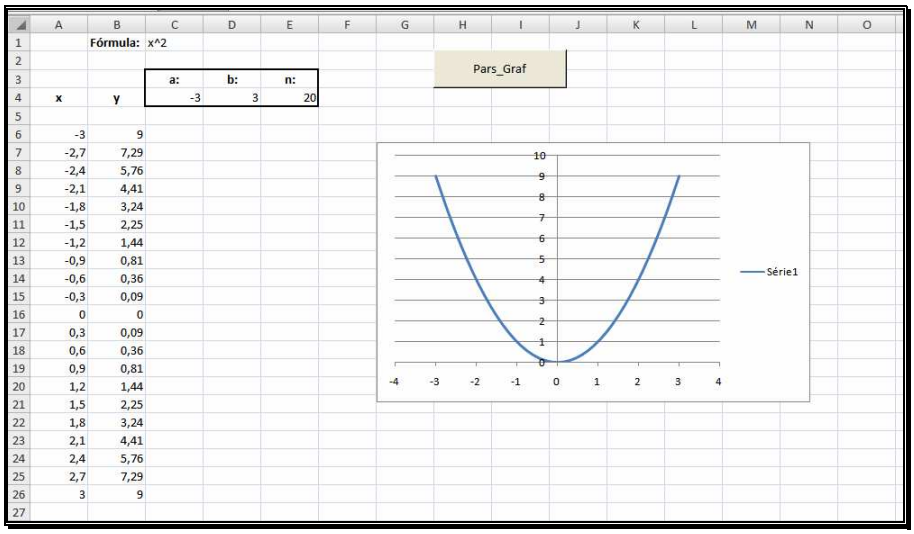


O exemplo mostra que não escrevemos =180*ATAN(D2)/PI(), mas sim 180*atan(x)/pi. Temos que levar em conta que as variáveis empeçam com uma letra. Para x, y, z permita-se multiplicação implícita, ou seja 3x é interpretada como sendo 3*x. Em nosso caso, devemos escrever 180*atan(x)/pi .
Podemos escrever, utilizando expressões lógicas, uma função *definida por troços*. Como exemplo encontramos na documentação do clsMathParser a seguinte função:

$$f(x) = \begin{cases} x^2 & ; \quad x \le 0 \\ \ln(x+1) & ; \quad 0 < x \le 1 \\ \sqrt{x - \ln(2)} & ; \quad x > 1 \end{cases}$$

Escreveremos esta função como uma string da seguinte forma:

f(x):="(x<=0)*x^2 + (0<x<=1)*Ln(x+1) + (x>1)*Sqr(x-Ln(2))"



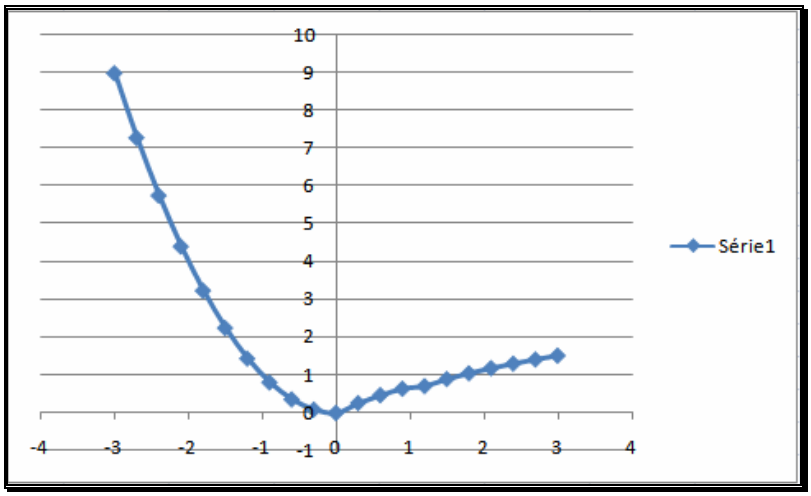
A figura mostra a imagem de uma parábola $f(x) = x^2$ que fica na célula C1. O seguinte programa criou a tabela e o gráfico.

```
Private Sub Pars_Graf()
'com clsMathParser e mMathSpecFun.bas de clsMathParser.zip
'veja http://www.cidse.itcr.ac.cr/cursos-linea/NUMERICO/excel/node28.html
    Dim n As Integer
    Dim h As Double
    Dim Formula As String
    Dim graf   As Chart
    Dim chartsTemp As ChartObjects 'contador de charts (gráficos) para eliminar o anterior
    Dim OK As Boolean
    Dim Fun As New clsMathParser

    n = Cells(4, 5)
    a = Cells(4, 3)
    b = Cells(4, 4)
    h = (b - a) / n
    Formula = Cells(1, 3)

    OK = Fun.StoreExpression(Formula)     'leitura da fórmula
    If Not OK Then GoTo Error_Handler
    For i = 0 To n
    Cells(6 + i, 1) = a + i * h
    Cells(6 + i, 2) = Fun.Eval1(a + i * h)
    Next i
'----------------------- eliminar gráficos anteriores-------------
    Set chartsTemp = ActiveSheet.ChartObjects
    If chartsTemp.Count > 0 Then
         chartsTemp(chartsTemp.Count).Delete
    End If
'-----------------------------------------------------------------
    dados = Range(Cells(6, 1), Cells(6 + n, 2)).Address 'intervalo a desenhar
    Set graf = Charts.Add    'gráfico y suas caracteristicas
    With graf
        .Name = "Gráfico"
        .ChartType = xlXYScatterSmoothNoMarkers
        .SetSourceData Source:=Sheets("Plan1").Range(dados), PlotBy:=xlColumns
        .Location Where:=xlLocationAsObject, Name:="Plan1"
      End With
'-----------------------------------------------------------------
    If Err Then GoTo Error_Handler
Error_Handler:     Cells(2, 1) = Fun.ErrorDescription 'imprimir mensagem de erro
'-----------------------------------------------------------------
End Sub
```

Se colocamos a expressão (x<=0)*x^2 + (0<x<=1)*Ln(x+1) + (x>1)*Sqr(x-Ln(2)) na célula C1, deixando o resto como no caso da parábola, obtemos a figura em baixo.



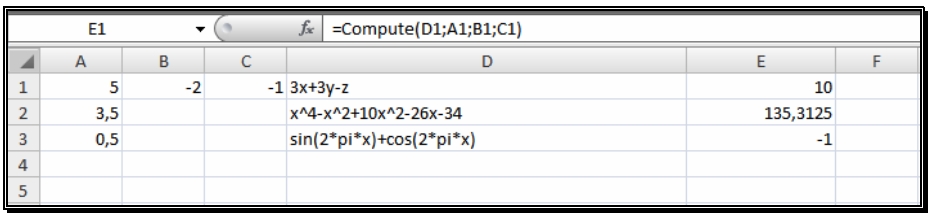
Veja também o programa "Eval_Pieces_Function" na pagina 50 da documentação do clsMathParser.

O loop
```
For i = 0 To n
 Cells(6 + i, 1) = a + i * h
 Cells(6 + i, 2) = Fun.Eval1(a + i * h)
Next i
```

determina os dados da tabela. Fun.Eval1(x) avalia o valor da função para x = a+i*h.
Tecnicamente falado, `Eval1(x)` é um *método* assim como também `Eval(x)`e `StoreExpression` que armazena a expressão e que revisa a sua sintaxe. `Eval(x)` avalia uma expressão de vários parâmetros e/ou variáveis. `Eval1(x)` aplicamos quando queremos avaliar uma expressão de uma variável só.
O curto programa a seguir utiliza `Eval` para poder, também, avaliar funções de varias variáveis como f(x,y,z) = 3x+3y-z nos pontos x = 5, y = -2, z = -1. Resultado: 10.

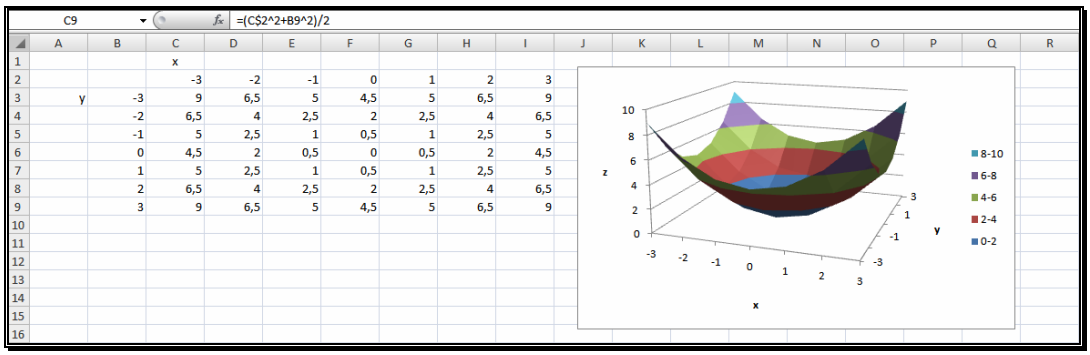
```
Function Compute(Formula As String, ParamArray Var())
Dim retval As Double, ParMax As Long
Dim Fun As New clsMathParser, OK As Boolean
On Error GoTo Error_Handler
'-----------------------------------
OK = Fun.StoreExpression(Formula)
'-----------------------------------
If Not OK Then Err.Raise 1001, , Fun.ErrorDescription
ParMax = UBound(Var) + 1 'número dos parâmetros
If ParMax < Fun.VarTop Then Err.Raise 1002, , "missing parameter"
ReDim Values(1 To ParMax)
'lê os valores dos parâmetros
For i = 1 To ParMax
Fun.Variable(i) = Var(i - 1)
Next
Compute = Fun.Eval() ' aqui aparece Eval
If Err <> 0 Then GoTo Error_Handler
Exit Function
Error_Handler:
Compute = Err.Description
End Function
```

Chamamos, usando o símbolo f_x , a função "Compute" e preenchemos os colchetes dela assim como mostra a figura. O cursor ficava na coluna E onde foram escritos os resultados.

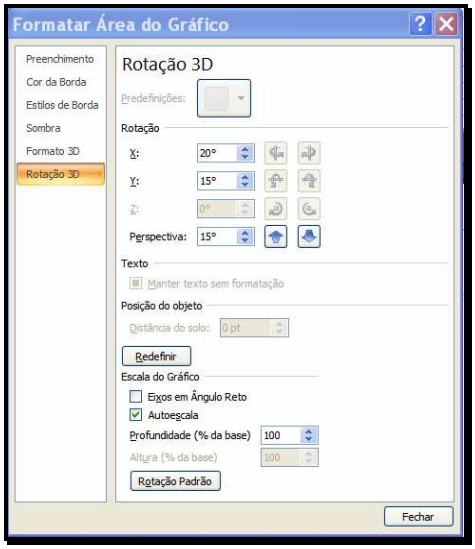


Superfícies 3D em Excel

O Excel suporta 73 tipos de gráficos, entre eles o tipo Superfície 3D. Encontramos este tipo via *Inserir>Gráficos>Outros Gráficos.*
Um gráfico de superfície é útil quando se pretende encontrar as melhores combinações entre dois conjuntos de dados. Nós pretendemos usar como conjuntos de dados as coordenadas x e y de um função z = f(x,y). A região dos pares (x,y) é definida por [xmin,xmax]x[ymin,ymax]. Veja o seguinte exemplo onde traçamos a função z = f(x,y) = $(x^2+y^2)/2$ sobre a região [-3,3] X [-3,3].



Os dados para o gráfico estão no intervalo C3:I9. A cada ponto neste intervalo pertencem duas coordenadas. As coordenadas x para o eixo na frente ficam em C2:I2, as coordenadas y do eixo lateral, que apresenta a profundidade do gráfico, estão em B3:B9. Os valores da função são apresentados vertical-mente, da base do gráfico até a superfície colorada.
Para criar este gráfico, selecionamos as células B2:I9. O gráfico pode ser editado da maneira usual. Mas, para o tipo Superfície 3D existem métodos específicos, como a Rotação 3D, a formatação dos paredes, da base etc.



À semelhança do que acontece num mapa topográfico, as cores indicam as áreas que pertencem ao mesmo intervalo de valores da função, veja a distribuição das cores junto com os intervalos de valores específicos.
(Um gráfico de superfície 3D apresentado sem cor é denominado gráfico de superfície vectorial 3D. Um Gráfico de Níveis é um gráfico de superfície visto de cima, em que as cores representam intervalos de valores específicos. Este tipo de gráfico apresentado sem cor é denominado Gráfico de Níveis Vectorial.)

O preenchimento da tabela fazemos em VBA com o seguinte código

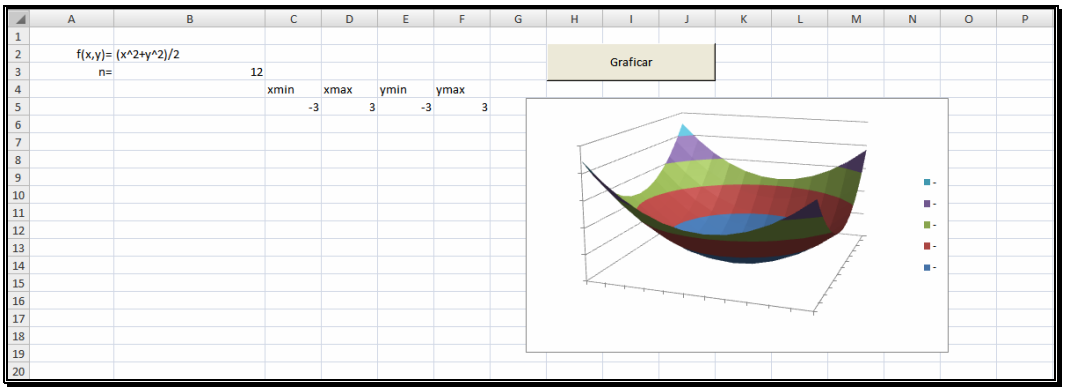
```
Sub preencher()

n = 6
xmin = -3
xmax = 3
ymin = -3
ymax = 3
hx = (xmax - xmin) / n
hy = (ymax - ymin) / n

  For i = 0 To n
    x = xmin + i * hx
      Cells(2, 2 + i) = x
      For j = 0 To n
        y = ymin + j * hy
        Cells(3 + j, 1) = y
            Cells(3 + j, 2 + i) = (x ^ 2 + y ^ 2) / 2
      Next j
   Next i

End Sub
```

A superfície será bastante mais liso e uniforme, se aumentarmos o número das subdivisões de n = 6 para n = 12, ou mais ainda.
Agora, não é difícil incorporar este código no programa Pars_Graf que vimos acima. Na planilha inserimos primeiro os dados mostrados na seguinte figura.
Observe que não aparecem na planilha os valores calculados pelo programa. Este se deve à linha `Selection.NumberFormat = ";;;"` que oculta um número com o formato ";;;", veja *Formatar Células >Personalizado*.



```
Private Sub graf3D()
    Dim xmin As Double, xmax As Double, ymin As Double, ymax As Double
    Dim hx As Double, hy As Double, x As Double, y As Double
    Dim n As Integer
    Dim fxy As String 'função f(x,y)
    Dim graf   As Chart
    Dim OK As Boolean
    Dim Fun As New clsMathParser

    fxy = Cells(2, 2)
    xmin = Cells(5, 3)
    xmax = Cells(5, 4)
    ymin = Cells(5, 5)
    ymax = Cells(5, 6)
    n = Cells(3, 2) ' número de subdivisões
    hx = (xmax - xmin) / n
    hy = (ymax - ymin) / n

    If hx > 0 And hy > 0 And n > 0 Then
        For i = 0 To n
            x = xmin + i * hx
            Cells(7, 2 + i) = x
            For j = 0 To n
                y = ymin + j * hy
                Cells(8 + j, 1) = y

                OK = Fun.StoreExpression(fxy)
                If Not OK Then GoTo Error_Handler
                Fun.Variable("x") = x
                Fun.Variable("y") = y
                Cells(8 + j, 2 + i) = Fun.Eval() 'retorna f(x,y)
            Next j
        Next i

    End If
    '----------------------- eliminar gráficos anteriores-------------
    Set chartsTemp = ActiveSheet.ChartObjects
    If chartsTemp.Count > 0 Then
        chartsTemp(chartsTemp.Count).Delete
    End If
```

Continuação:

```
    '-----------------------------------------------------------------
    dados = Range(Cells(7, 1), Cells(7 + n, n + 2)).Address 'intervalo a graficar
    Range(dados).Select
    Selection.NumberFormat = ";;;"  ' assim ocultam-se os dados
    Charts.Add
    ActiveChart.ChartType = xlSurface
    ActiveChart.SetSourceData Source:=Sheets("Plan1").Range(dados), PlotBy:=xlColumns
    ActiveChart.Location Where:=xlLocationAsObject, Name:="Plan1"
    '-----------------------------------------------------------------
    If Err Then GoTo Error_Handler
Error_Handler:     Cells(1, 1) = Fun.ErrorDescription  'mensagem de erro
    '-----------------------------------------------------------------

End Sub
```

Para terminar este capítulo, utilizamos outro programa da acima mencionada documentação do "clsMathParser".
Trata-se da avaliação da famosa fórmula de **Planck** para a intensidade da radiação de um *corpo negro*.

A intensidade da radiação térmica emitida na semi-esfera é dada pela seguinte equação:

$$I(\lambda,T)=\frac{2\pi c_1\lambda^{-5}}{e^{\frac{c_2}{\lambda T}}-1}$$

$c_1 = 5{,}955\cdot10^{-13}$ W·cm^2 ; $c_2 = 1{,}439$ cm·K; λ = comprimento de onda (cm);
T = temperatura da superfície em K.

O programa tem uma estrutura muito simples:

```
Sub multiparam()
Dim ok As Boolean, F As Double, Formula As String
Dim Funct As New clsMathParser

C1 = 5.955E-13 'W*cm^2
C2 = 1.439 ' cm*K

Formula = "(2*pi*C1/(L^5*(exp(C2/(L*T))-1)))" ' fórmula de Planck

ok = Funct.StoreExpression(Formula)
If Not ok Then GoTo Error_Handler
On Error GoTo Error_Handler

    Funct.Variable("C1") = C1
    Funct.Variable("C2") = C2
    Funct.Variable("L") = 0.0002
    Funct.Variable("T") = 1200

    F = Funct.Eval 'avaliando uma função de varios parâmetros
Debug.Print "evaluation = "; F ' aparece na janela
'da Verificação imediata: Resultado 29176,7339716322 ' W*cm^-3
Exit Sub
Error_Handler:
Debug.Print Funct.ErrorDescription
End Sub
```

Observe que o programa manda o resultado com `Debug.Print` à janela *Verificação imediata* que pode-se eleger com *Exibir* no editor do VBA (ou simplesmente escrevendo Ctrl+g). `Debug.Print` é muito útil para verificar rapidamente um resultado, se bem que é pensado para o processo de depuração do código.

Elementos para criar um gráfico

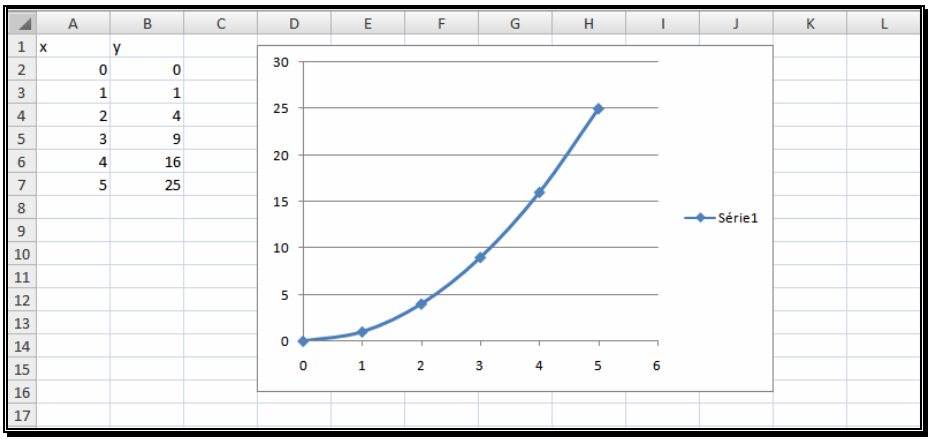
Neste lugar vou unir num simples programa alguns elementos para criar um gráfico "xlXYScatterSmooth".
Para desenhar uma função com valores em A2:B7, basta o seguinte código

```
Sub Charts()
Dim ch As ChartObject
Dim cw As Long, rh As Long
cw = Columns(1).Width
rh = Rows(1).Height
Set ch = ActiveSheet.ChartObjects.Add(cw * 3, rh * 0.5, cw * 7, rh * 15)
ch.Chart.ChartType = xlXYScatterSmooth

ch.Chart.SeriesCollection.Add _
    Source:=ActiveSheet.Range("A2:B7"), _
Rowcol:=xlColumns, SeriesLabels:=False, CategoryLabels:=True
End Sub
```

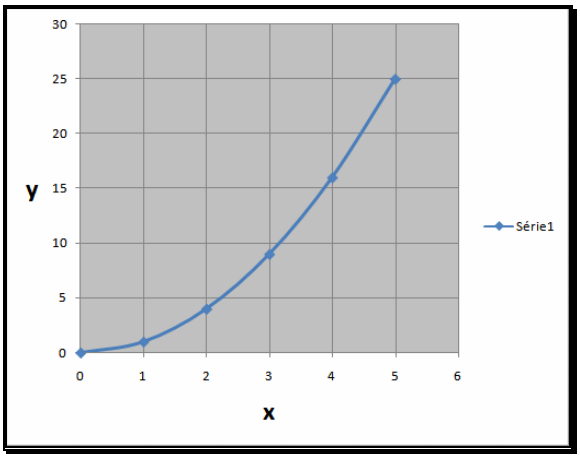
A linha `Set ch= ...` é um pouco assustador, mas, por meio dela podemos

variar o tamanho e a posição do gráfico, utilizando como unidades a largura e a altura de colunas e linhas. O objeto Worksheet tem uma propriedade ChartObjects que retorna uma coleção ChartObjects, que é a coleção de todos os ChartObjects na planilha. Os parâmetros do método Add são *Left, Top, Width, Height*. Left e Top dão as coordenadas do canto superior esquerdo do gráfico relativas ao canto superior esquerdo da célula A1 na planilha. Width e Height especificam o tamanho do gráfico em "pontos".



Os parâmetros de Add posicionam o gráfico inicialmente de modo que seu canto superior esquerdo tenha três colunas a partir da margem esquerda e meia linha para baixo da parte superior da planilha. O tamanho inicial do gráfico é sete colunas de largura por quinze linhas de altura. O gráfico corresponde com grande exatidão a estes valores. (A unidade básica de medição são os "pontos", definidos como 72 pontos = 1 polegada.)

O gráfico não é muito bonito, mas, neste exemplo, vemos como com poucas linhas de código é possível transformar rapidamente dados num simples gráfico.
Para satisfazer exigências mais sofisticadas, VBA oferece os médios para colocar rótulos nos eixos, para colorar o fundo, inserir linhas de grade verticais e outros.



Na figura acima foram adicionadas rótulos, linhas de grade verticais e um fundo gris.

Para lograr estas melhoras, foram adicionadas mais algumas linhas de código:

```
Sub CriarGrafico()
Dim ch As ChartObject
Dim cw As Long, rh As Long
cw = Columns(1).Width
rh = Rows(1).Height
'tamanho do gráfico
Set ch = ActiveSheet.ChartObjects.Add(cw * 3, rh * 0.5, cw * 7, rh * 15)
ch.Chart.ChartType = xlXYScatterSmooth

ch.Chart.SeriesCollection.Add _
    Source:=ActiveSheet.Range("A2:B7"), _
Rowcol:=xlColumns, SeriesLabels:=False, CategoryLabels:=True

'Rótulos para eixos
    With ch.Chart.Axes(xlCategory)
        .HasTitle = True
        .AxisTitle.Caption = "x"
        .AxisTitle.Font.Size = 18
    End With
    With ch.Chart.Axes(xlValue)
        .HasTitle = True
        .AxisTitle.Caption = "x"
        With .AxisTitle
         .Caption = "y"
         .Orientation = xlHorizontal
         .Font.Size = 18
        End With
 End With
 'color do interior gris
 ch.Chart.PlotArea.Interior.ColorIndex = 15
 'linhas de grade verticais
 ch.Chart.Axes(xlCategory).HasMajorGridlines = True

End Sub
```

Para informarmos sobre as infinitas outras possibilidades, devemos consultar a documentação do VBA. A maioria dos livros especializados em VBA ou em gráficos contêm listas com os 73 tipos de gráficos, chart elements etc.
Dois exemplos:

- Steven Roman, *Desenvolvendo macros no Excel*, Editora Ciência Moderna Ltda. 2000
- Bill Jelen, *Charts and Graphs for Excel 2007*, QUE Publishing 2007

Para aprender as técnicas da criação de gráficos, pode-se consultar efetivamente os exemplos que os especialistas no assunto oferecem gratuitamente no internet, veja, por exemplo, o programa "funcplot2point1.xls" de Milner e Watson no site http://www.waltermilner.com/vbmaths/