Capítulo 7

Matemática I (MDC e MMC, núm. complexos, equações)

Divisibilidade (MDC e MMC)

Na última seção (Calendário), o problema da divisibilidade era de alta importância. Para discutir este problema mais a fundo, vamos desenvolver uma planilha de Excel que nos ajudará na busca dos divisores de um número.

1. **MDC** (máximo divisor comum)
   O máximo divisor comum dos inteiros *a* e *b*, não nulos, é o maior inteiro que divide simultaneamente *a* e *b*.

Existem vários métodos para encontrar o MCD, por exemplo o método de decomposição de um número num produto de fatores primos. Mas, na planilha, queremos usar o método das divisões sucessivas, também conhecido como o *algoritmo de Euclides*. Quando se trata de números muito grandes, pode não ser fácil encontrar a decomposição em fatores primos. O método de Euclides é muito prático, pois é baseado apenas em divisões sucessivas, como vemos no seguinte exemplo.

Exemplo: Determine o **MDC** dos inteiros 240 e 408 pelo método das divisões sucessivas (o algoritmo de Euclides)

408:240 = 1 com resto 168 (= Mod(408,240))
240:168 = 1 com resto 72
168:72 = 2 com resto 24
72: 24 = 3 com resto zero. Portanto o MDC procurado é igual a 24.

C7 =SE(E(B7>0;B7<>"");MOD(A7;B7);"")

	A	B	C	D	E	F	G	H	I
1				a=	240		Verificação:		
2	MDC e MMC			b=	408		a*b=	97920	
3							MDC*MMC	97920	
4									
5	MDC =	24			MMC =	4080			
6	Nk-2	Nk-1	Nk						
7	408	240	168		1	240	240		
8	240	168	72		2	480	72		
9	168	72	24		3	720	312		
10	72	24	0	24	4	960	144		
11	24				5	1200	384		
12					6	1440	216		
13					7	1680	48		
14					8	1920	288		
15					9	2160	120		
16					10	2400	360		
17					11	2640	192		
18					12	2880	24		
19					13	3120	264		
20					14	3360	96		
21					15	3600	336		
22					16	3840	168		
23					17	4080	0		
24					18				
25									

Podemos formular o algoritmo de Euclides como segue

$$n_0 = \max(|a|,|b|)$$
$$n_1 = \min(|a|,|b|)$$
$$n_k = n_{k-2} \bmod n_{k-1}$$
$$k = 2,3,...$$

A planilha foi construída da seguinte maneira:

A7: =MÁXIMO(ABS(E1);ABS(E2))
B7: =MÍNIMO(ABS(E1);ABS(E2))
C7: =SE(E(B7>0;B7<>"");MOD(A7;B7);"")
D7: =SE(C7=0;B7;"")
B5: =MÁXIMO(D7:D50)

A8: =SE(B7>0;B7;"")
B8: =SE(C7>0;C7;"")
C8: = SE(E(B8>0;B8<>"");MOD(A8;B8);"")

D8: =SE(C8=0;B8;"")

Copie os conteúdos das células A8:D8 até linha 50.

Utilizamos a coluna D, para encontrar um zero na coluna C. Quando o zero é encontrado, o número à esquerda, na coluna B, será o MDC desejado, ele é o máximo na coluna D.

2. **MMC** (mínimo múltiplo comum)
   O mínimo múltiplo comum dos números inteiros *a* e *b* não-nulos é o menor inteiro no conjunto dos seus múltiplos comuns.

Pode-se provar que $a \cdot b = \text{MDC}(a,b) \cdot \text{MMC}(a,b)$
Utilizamos este teorema na planilha para verificar os valores de MDC e MMC encontrados separadamente.

Multiplicamos *a* sucessivamente por $i = 1,2,...$, até o produto $a \cdot i \bmod b$ seja igual a zero. Neste momento, $i \cdot a$ é sem resto divisível por *b*, ou seja, $i \cdot a$ é o MMC(*a,b*).

E7: 1
E8: =SE(F7<>"";E7+1;"")
F7: =E1*E7
F8: =SE(E(G7<>0;G7<>"");E1*E8;"")
G7: =MOD(F7;E2)
G8: =SE(F8<>"";MOD(F8;E2);"")
Copiar E8:G8 até linha 200 (ou maior!)

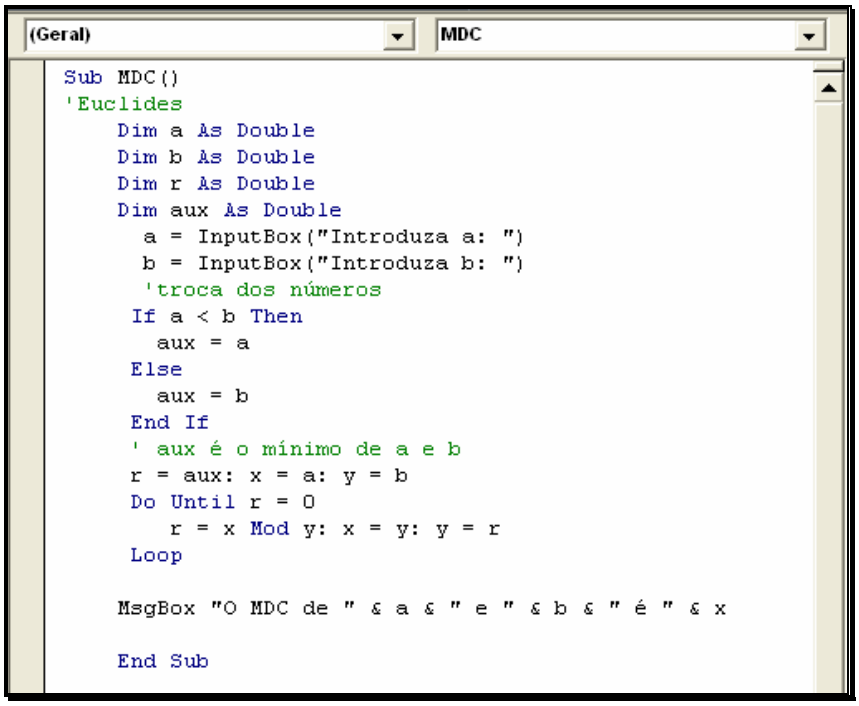
O **MMC** é =MÁXIMO(F7:F400) e fica em F5.

H2: =E1*E2; H3: =B5*F5 -os dois números devem ser idênticos.

(Pode ser preciso copiar E8:G8 até uma linha bem maior do que 140, para obter $a \cdot i \bmod b = 0$, por isso, usamos F400, para poder extender a buscada de zero até a linha 400. Tente os números a=339 e b=1128; apenas na linha 382 aparece zero!)

Para praticar os nossos conhecimentos na programação em VBA, escrevemos alguns pequenos programas sobre a determinação do MDC. (O Excel mesmo tem incorporado as funções MDC e MMC por meio do **Analysis ToolPak**.)

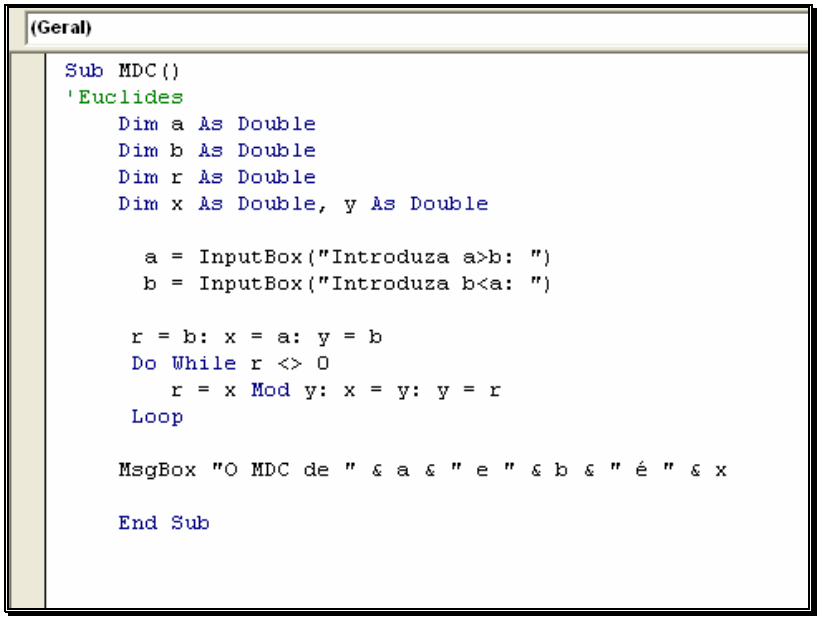
Programa 1:



No seguinte programa seja $a > b$. O programa calcula para cada número n < b se é um divisor comum de a e b, ou seja, determinamos, se valem simultâneamente as duas relações $a \bmod n = 0$ e $b \bmod n = 0$. Se for assim, sabemos que este número é o MDC(a,b).

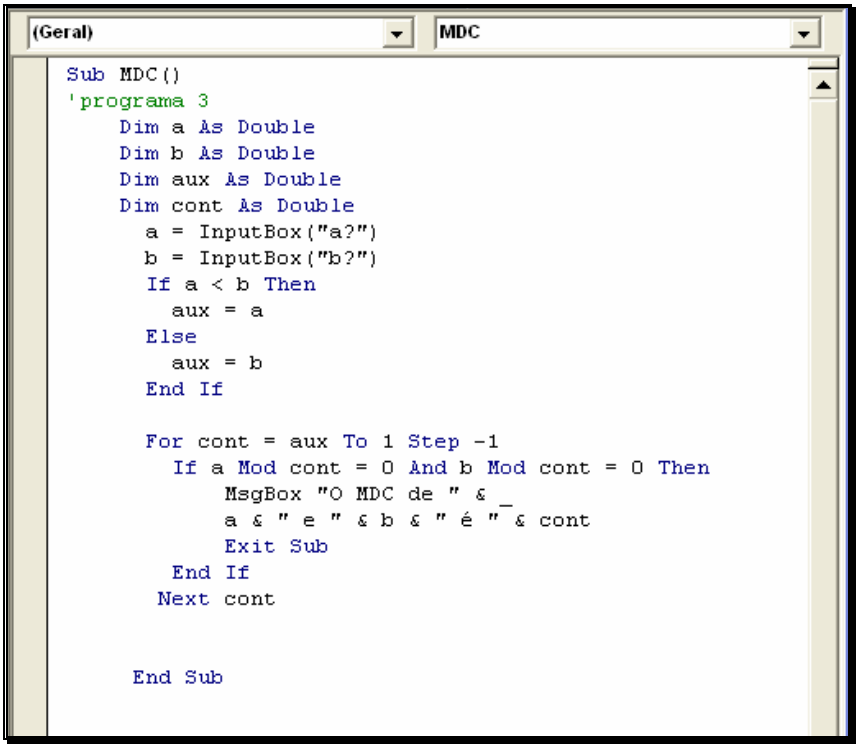
Este programa não troca os números e utiliza `Do While` , em vez de `Do Until`.

Programa 2:



```
(Geral)

Sub MDC()
'Euclides
    Dim a As Double
    Dim b As Double
    Dim r As Double
    Dim x As Double, y As Double

      a = InputBox("Introduza a>b: ")
      b = InputBox("Introduza b<a: ")

     r = b: x = a: y = b
     Do While r <> 0
        r = x Mod y: x = y: y = r
     Loop

    MsgBox "O MDC de " & a & " e " & b & " é " & x

    End Sub
```

Programa 3:

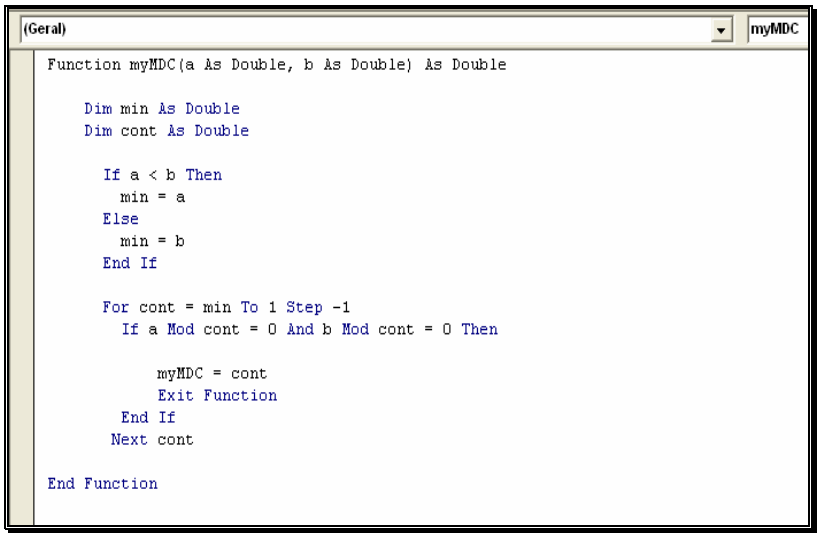


```
(Geral)                                    MDC

Sub MDC()
'programa 3
    Dim a As Double
    Dim b As Double
    Dim aux As Double
    Dim cont As Double
      a = InputBox("a?")
      b = InputBox("b?")
      If a < b Then
        aux = a
      Else
        aux = b
      End If

      For cont = aux To 1 Step -1
        If a Mod cont = 0 And b Mod cont = 0 Then
            MsgBox "O MDC de " & _
            a & " e " & b & " é " & cont
            Exit Sub
        End If
       Next cont


     End Sub
```

O último Programa 3 mostra um método que determina de cada um dos números menores do menor de *a* e *b*, se ele é um divisor comum de *a* e *b*. O primeiro número que cumpre com esta condição será o maior comum divisor de *a* e *b*

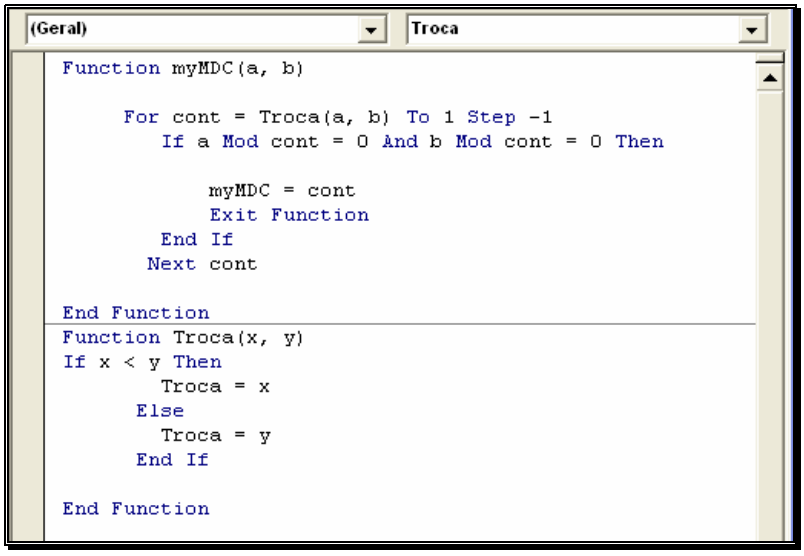
Sub-rotinas e Funções

Todos os SUB-Programas são executadas desde o editor do VBA com F5.
Seria muito mais prático tê-las como funções, pois dessa forma poderíamos usá-las como as outras funções embutidas no Excel. (Na realidade, o MDC é uma função, pois ele retorna um só valor.)



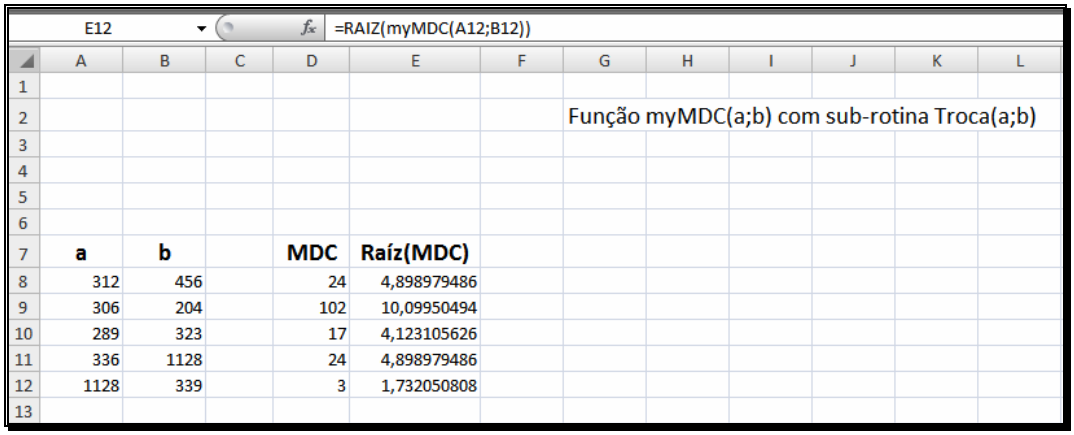

```
Function myMDC(a As Double, b As Double) As Double

    Dim min As Double
    Dim cont As Double

      If a < b Then
        min = a
      Else
        min = b
      End If

      For cont = min To 1 Step -1
        If a Mod cont = 0 And b Mod cont = 0 Then

            myMDC = cont
            Exit Function
        End If
       Next cont

End Function
```

No seguinte programa definimos a função-VBA "myMDC" junto com uma função "Troca" que se ocupa com determinar o menor de *a* e *b*.




```
Function myMDC(a, b)

     For cont = Troca(a, b) To 1 Step -1
        If a Mod cont = 0 And b Mod cont = 0 Then

            myMDC = cont
            Exit Function
        End If
       Next cont

End Function
Function Troca(x, y)
If x < y Then
        Troca = x
      Else
        Troca = y
      End If

End Function
```

Para simplificar, foram deixados afora do programa as instruções Dim.
Agora demonstramos o uso da função "myMDC". Ela não só determina o MDC mas também calcula a raiz quadrada dele. (Compare com a função "Montante" no terceiro capítulo.)

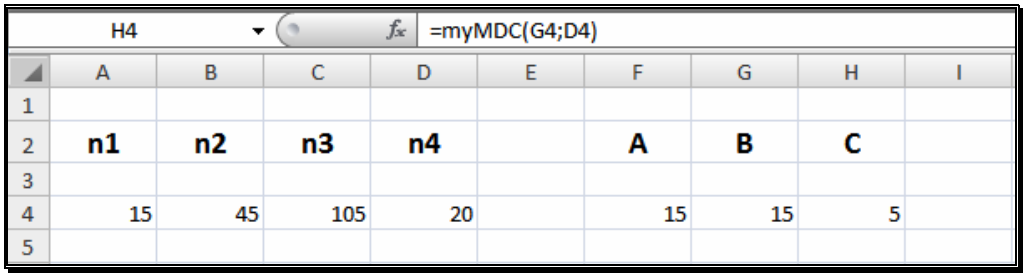


E12 =RAIZ(myMDC(A12;B12))

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2							Função myMDC(a;b) com sub-rotina Troca(a;b)					
3												
4												
5												
6												
7	**a**	**b**		**MDC**	**Raíz(MDC)**							
8	312	456		24	4,898979486							
9	306	204		102	10,09950494							
10	289	323		17	4,123105626							
11	336	1128		24	4,898979486							
12	1128	339		3	1,732050808							
13												

Com nossas funções podemos facilmente determinar o MDC de n números inteiros com n > 2. É só preciso calcular primeiro A = MDC(A1,B1), depois B = MDC(A,C10) etc.

Exemplo: =MDC(15;45;105;20) = 5 (com a função MDC do Excel)



H4 =myMDC(G4;D4)

	A	B	C	D	E	F	G	H	I
1									
2	**n1**	**n2**	**n3**	**n4**		**A**	**B**	**C**	
3									
4	15	45	105	20		15	15	5	
5									

Equações do segundo grau

Dada $ax^2 + bx + c = 0,\ com\ [(a,b,c) \in R; a \neq 0]$.
Essa é uma equação do segundo grau, cuja resolução calculamos com a chamada fórmula de *Bhashara* (um matemático hindu do século XII):

$$x_1, x_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

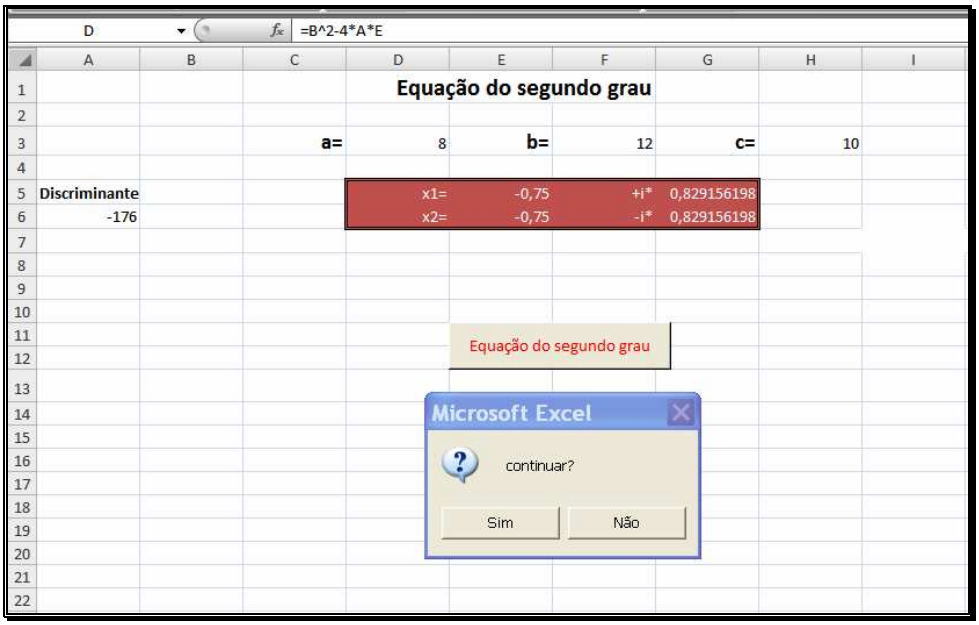
A expressão $D := b^2 - 4ac$ chama-se *discriminante.* Usando a discriminante, podemos escrever a resolução na forma

$$S = \left\{ \frac{-b+\sqrt{D}}{2a}, \frac{-b-\sqrt{D}}{2a} \right\}$$

A resolução de nossa equação dependerá do valor de *D*. Devemos considerar três casos:

D > 0, neste caso, a equação terá duas raízes distintas
D =0, neste caso, a equação terá duas raízes reais idênticas
D < 0, neste caso, a equação terá raízes imaginárias

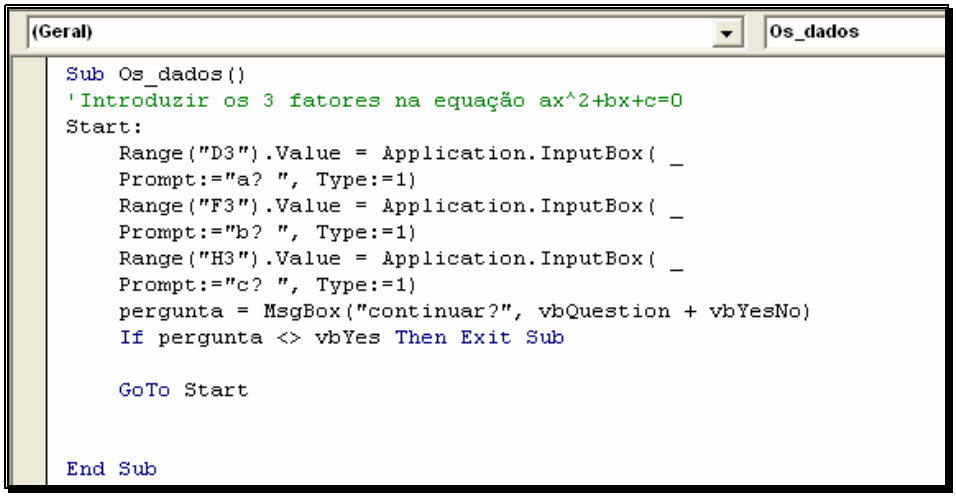
Na planilha vamos tomar em consideração os três casos.



As células A6, D3, F3, H3 foram denominados com D, A, B, E na "Caixa de nome" (ao canto esquerdo da "Barra de Fórmulas"). O nome "C" não foi aceitado pelo Excel, por isso utilizei E. (Só aparece na célula A6.)

A6: =B^2-4*A*E (E em lugar de C; Discriminante)
E5: =SE(D>0;(-B+RAIZ(D))/(2*A);-B/(2*A))
F5: =SE(D<0;"+i*";""))
G5: =SE(D<0;RAIZ(-D)/(2*A);"")
E6: =SE(D>0;(-B-RAIZ(D))/(2*A);-B/(2*A))
F6: =SE(D<0;"-i*";""))
G6 =SE(D<0;RAIZ(-D)/(2*A);"")

Poderíamos inscrever os números a, b, c diretamente nas células D3, F3, H3, mas, é mais chique utilizar uma macro: (Type := 1 significa que se introduz um número; 2 = texto (uma *string*), 4 = valor lógico, 16 = valor de erro, tal como #N/A, 64 = array (matriz)).




```
Sub Os_dados()
'Introduzir os 3 fatores na equação ax^2+bx+c=0
Start:
    Range("D3").Value = Application.InputBox( _
    Prompt:="a? ", Type:=1)
    Range("F3").Value = Application.InputBox( _
    Prompt:="b? ", Type:=1)
    Range("H3").Value = Application.InputBox( _
    Prompt:="c? ", Type:=1)
    pergunta = MsgBox("continuar?", vbQuestion + vbYesNo)
    If pergunta <> vbYes Then Exit Sub

    GoTo Start


End Sub
```

A macro é executada por meio de um botão. (Em 2007 se faz assim: *Desenvolver>Controles>Inserir>Botão* + Atribuir o nome da macro "Os_dados" em nosso caso.)

Resolução de equações do segundo grau com VBA

No programa a seguir, utilizamos uma função matricial cujos argumentos são os três coeficientes a, b, c. A função determina os soluções da equação e as coloca num array 2X2 de células.

```
(Geral)                                          segundograu

Public Function segundograu(coef)

 Dim discr1 As Double, discr2 As Double
 Dim a As Double, b As Double, c As Double
 Dim result(1 To 2, 1 To 2) As Double

 a = coef(1)
 b = coef(2)
 c = coef(3)
 discr1 = b ^ 2 - 4 * a * c

 If discr1 >= 0 Then
    result(1, 1) = (-b + Sqr(discr1)) / (2 * a)
    result(1, 2) = 0
    result(2, 1) = (-b - Sqr(discr1)) / (2 * a)
    result(2, 2) = 0
   Else
    discr2 = Sqr(Abs(discr1))
    If b = 0 Then
        result(1, 1) = 0
        result(2, 1) = 0
    Else
        result(1, 1) = -b / (2 * a)
        result(2, 1) = -b / (2 * a)
    End If

        result(1, 2) = discr2 / (2 * a)
        result(2, 2) = -discr2 / (2 * a)

  End If

  segundograu = result


End Function
```

Introduza os valores de a, b, c nas células A1, A2, A3. Selecione A5:B6 e clique uma vez sobre o ícone f_x para *Inserir Função* "segundograu" que fica na categoria *Definida pelo usuário*. Na janela "Coef" escrevemos (A1:A3) y em seguida pressionamos Ctrl+Shift+Enter e vemos os valores originais e os resultados.

A5 | fx | {=segundograu(A1:A3)}

	A	B	C	D	E	F
1	2					
2	4					
3	8					
4						
5	-1	1,732051				
6	-1	-1,73205				
7						

Para a equação $2x^2 + 4x + 8 = 0$ teremos o resultado

x1 = -1 + 1,732... (*i) e x2 = - 1 – 1,732... (*i)

Se tiver selecionado um intervalo 2x2 (=array de 4 células) antes de ativar a função "segundograu", então ela escreverá os resultados também na pasta de trabalho, -depois de pressionar Ctrl+Shift+Enter (com OK não funciona!).

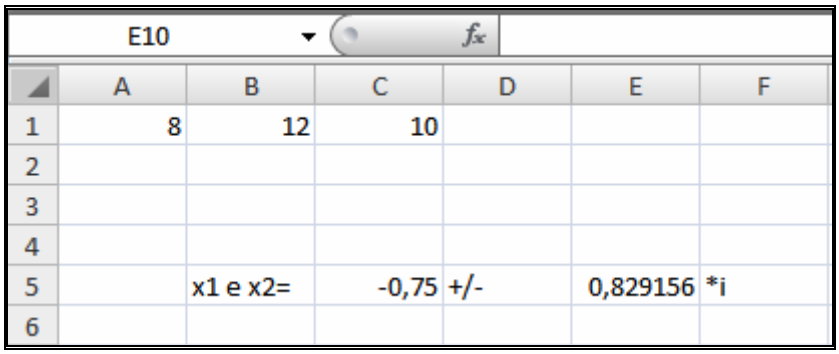
O seguinte programa "SegGrau", que também resolve uma equação de segundo grau, demonstra outra vez uma aplicação da propriedade Cells.

(Geral) SegGrau

```
Sub SegGrau()

    Dim a As Double, b As Double, c  As Double
    Dim discr As Double, parte_real As Double, parte_imag As Double
    Dim x1 As Double, x2 As Double


    a = Sheets(1).Cells(1, 1)
    b = Sheets(1).Cells(1, 2)
    c = Sheets(1).Cells(1, 3)

    Range("A5:F5").Clear

    discr = b ^ 2 - 4 * a * c

    If discr > 0 Then
        x1 = (-b + Sqr(discr)) / (2 * a)
        x2 = (-b - Sqr(discr)) / (2 * a)
        Sheets(1).Cells(5, 2) = "x1= "
        Sheets(1).Cells(5, 3) = x1
        Sheets(1).Cells(5, 4) = "x2= "
        Sheets(1).Cells(5, 5) = x2

       ElseIf discr = 0 Then
        x1 = -b / (2 * a)
        Sheets(1).Cells(5, 2) = "x1= "
        Sheets(1).Cells(5, 3) = x1
        Sheets(1).Cells(5, 4) = "x2= "
        Sheets(1).Cells(5, 5) = x1
       ElseIf discr < 0 Then
        parte_real = -b / (2 * a)
        parte_imag = Sqr(Abs(discr)) / (2 * a)
        Sheets(1).Cells(5, 2) = "x1 e x2= "
        Sheets(1).Cells(5, 3) = parte_real
        Sheets(1).Cells(5, 4) = "+/-"
        Sheets(1).Cells(5, 5) = parte_imag
        Sheets(1).Cells(5, 6) = "*i"

      End If

 End Sub
```

Esta sub-rotina espera os coeficientes a, b, c nas células A1, B1, C1, pois na propriedade Cells(i,j), i significa *linha* e j *coluna*. Cells(3,2) seria célula B3.

Sheets(1) refere-se a Plan1 (=planilha 1) da pasta de trabalho ativo. Com a = 8, b = 12, c = 10 temos o seguinte resultado.



E10

	A	B	C	D	E	F
1	8	12	10			
2						
3						
4						
5		x1 e x2=	-0,75	+/-	0,829156	*i
6						

Para não perder o contato com VBA, lhe ofereço uma última versão de uma sub-rotina para a fórmula de *Bhashara*. O programa trabalha, entre outras coisas, com a variable String e com InputBox e MsgBox .

(Geral) — Equação_segundo_grau

```
Sub Equação_segundo_grau()
Dim equação As String
Dim texto As String
Dim a As Double, b As Double, c As Double
Dim D As Double, x1 As Double, x2 As Double
Dim x As Double, y As Double

equação = "ax^2+bx+c=0"
texto = "Equação de segundo grau: "
MsgBox texto & equação
a = InputBox("Introduza o valor de a: ")
b = InputBox("Introduza o valor de b: ")
c = InputBox("Introduza o valor de c: ")
D = b ^ 2 - 4 * a * c 'discriminante
 If D < 0 Then
     x = -b / (2 * a)
     y = Sqr(Abs(D)) / (2 * a)
     MsgBox " a equação 0= " & a & " x^2+ " & b & "x + " & c & _
     " tem as soluções: "
     MsgBox " x1/x2= " & x & " +/- " & y & "*i"
    ElseIf D = 0 Then
     MsgBox "a equação tem a solução dupla: " & -b / (2 * a) & ""
    ElseIf D > 0 Then
     x1 = (-b + D ^ 0.5) / (2 * a)
     x2 = (-b - D ^ 0.5) / (2 * a)
     MsgBox "A equação tem as soluções: x1= " & x1 & " e x2= " & x2

  End If

End Sub
```

Números Complexos

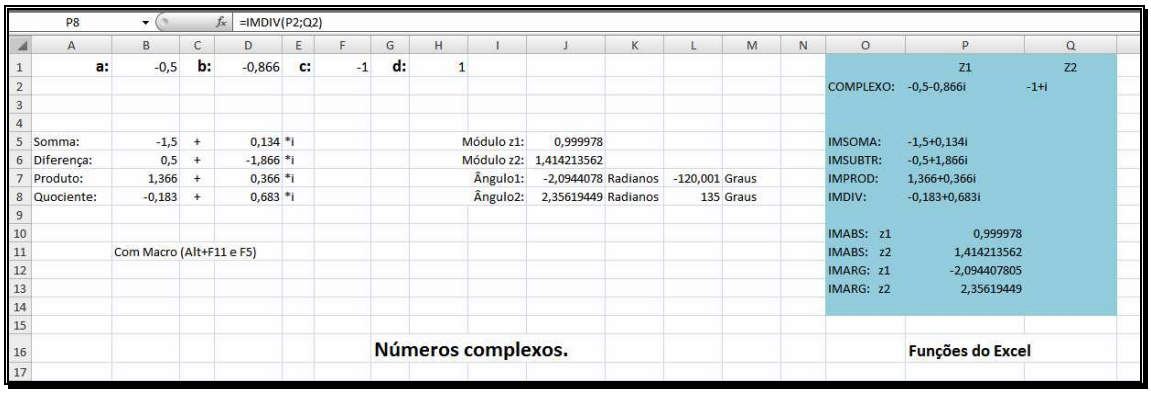
O cálculo básico com números complexos nos proporciona uma oportunidade para aprender alguns coisas novas em questões de programação em VBA.

Aprendemos

escrever um programa VBA com varias sub-rotinas de tipo função
para lidar com a função *InputBox*
escrever os dados e os resultados diretamente sobre uma planilha
usar a função *Val*

Já usamos varias vezes a função *InputBox* para a entrada de dados numéricos. Até agora, nunca tivemos problemas com esta função, mas esta vez podemos ficar surpreendidos.
Sem usar a função *Val*, obtemos uma *mensagem de erro* 13, pois a InputBox considera o número entrado pelo usuário como sendo uma *String* (= corrente de símbolos). Estas *Strings* são passadas para um procedimento, por exemplo "rsom", que está esperando como argumentos números simples.

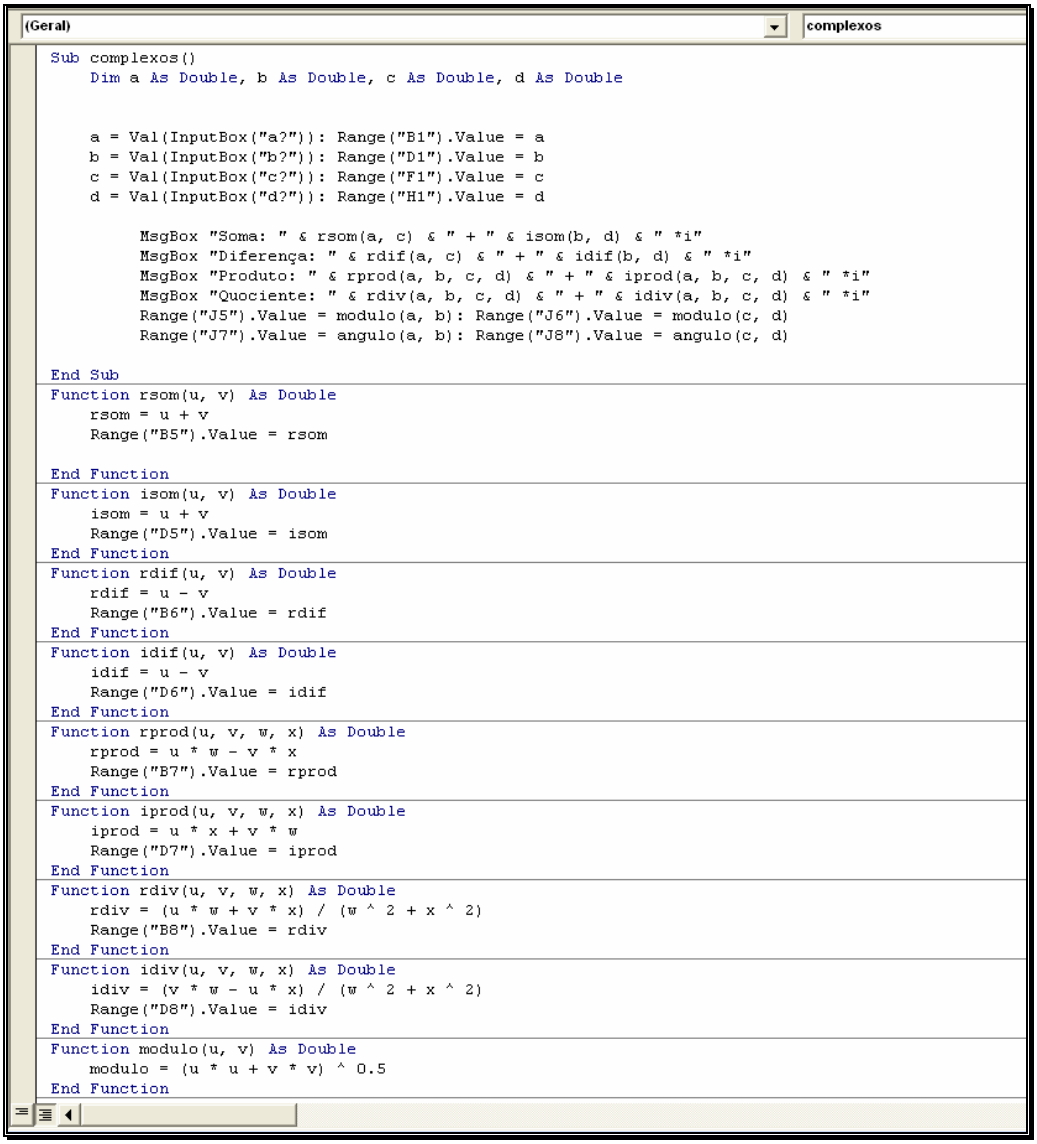
A função *Val* converte uma *String* para um número. (A função *Str* faz o contrario, ela converte um número, p. ex. 346, para a String "346".) Observe que *Val* não reconhece sinais de cifrão ou vírgulas. Assim, devemos escrever os números decimais *com ponto decimal* e não com vírgula. Experimente!
Na seguinte planilha, vemos os resultados que o próximo **programa** vai produzir com os números complexos z1 = -0.5 - 0.866i e z2 = -1 + 1i, (na região azul encontram-se os cálculos feitos com as funções complexas do Excel).



P8 =IMDIV(P2;Q2)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	**a:**	-0,5	**b:**	-0,866	**c:**	-1	**d:**	1								Z1	Z2
2															COMPLEXO:	-0,5-0,866i	-1+i
3																	
4																	
5	Somma:	-1,5	+	0,134	*i				Módulo z1:	0,999978					IMSOMA:	-1,5+0,134i	
6	Diferença:	0,5	+	-1,866	*i				Módulo z2:	1,414213562					IMSUBTR:	-0,5+1,866i	
7	Produto:	1,366	+	0,366	*i				Ângulo1:	-2,0944078	Radianos	-120,001	Graus		IMPROD:	1,366+0,366i	
8	Quociente:	-0,183	+	0,683	*i				Ângulo2:	2,35619449	Radianos	135	Graus		IMDIV:	-0,183+0,683i	
9																	
10															IMABS: z1	0,999978	
11		Com Macro (Alt+F11 e F5)													IMABS: z2	1,414213562	
12															IMARG: z1	-2,094407805	
13															IMARG: z2	2,35619449	
14																	
15																	
16							**Números complexos.**									**Funções do Excel**	
17																	

Programa "Números_complexos":



```
(Geral)                                                    complexos

Sub complexos()
    Dim a As Double, b As Double, c As Double, d As Double


    a = Val(InputBox("a?")): Range("B1").Value = a
    b = Val(InputBox("b?")): Range("D1").Value = b
    c = Val(InputBox("c?")): Range("F1").Value = c
    d = Val(InputBox("d?")): Range("H1").Value = d

         MsgBox "Soma: " & rsom(a, c) & " + " & isom(b, d) & " *i"
         MsgBox "Diferença: " & rdif(a, c) & " + " & idif(b, d) & " *i"
         MsgBox "Produto: " & rprod(a, b, c, d) & " + " & iprod(a, b, c, d) & " *i"
         MsgBox "Quociente: " & rdiv(a, b, c, d) & " + " & idiv(a, b, c, d) & " *i"
         Range("J5").Value = modulo(a, b): Range("J6").Value = modulo(c, d)
         Range("J7").Value = angulo(a, b): Range("J8").Value = angulo(c, d)

End Sub
Function rsom(u, v) As Double
    rsom = u + v
    Range("B5").Value = rsom

End Function
Function isom(u, v) As Double
    isom = u + v
    Range("D5").Value = isom
End Function
Function rdif(u, v) As Double
    rdif = u - v
    Range("B6").Value = rdif
End Function
Function idif(u, v) As Double
    idif = u - v
    Range("D6").Value = idif
End Function
Function rprod(u, v, w, x) As Double
    rprod = u * w - v * x
    Range("B7").Value = rprod
End Function
Function iprod(u, v, w, x) As Double
    iprod = u * x + v * w
    Range("D7").Value = iprod
End Function
Function rdiv(u, v, w, x) As Double
    rdiv = (u * w + v * x) / (w ^ 2 + x ^ 2)
    Range("B8").Value = rdiv
End Function
Function idiv(u, v, w, x) As Double
    idiv = (v * w - u * x) / (w ^ 2 + x ^ 2)
    Range("D8").Value = idiv
End Function
Function modulo(u, v) As Double
    modulo = (u * u + v * v) ^ 0.5
End Function
```

Com a instrução "Range("B6").Value = rdif" o Excel escreve o valor redif na célula B6 da nossa planilha, etc.
Para sua conveniência, serão dadas aqui as regras usadas nas sub-rotinas:

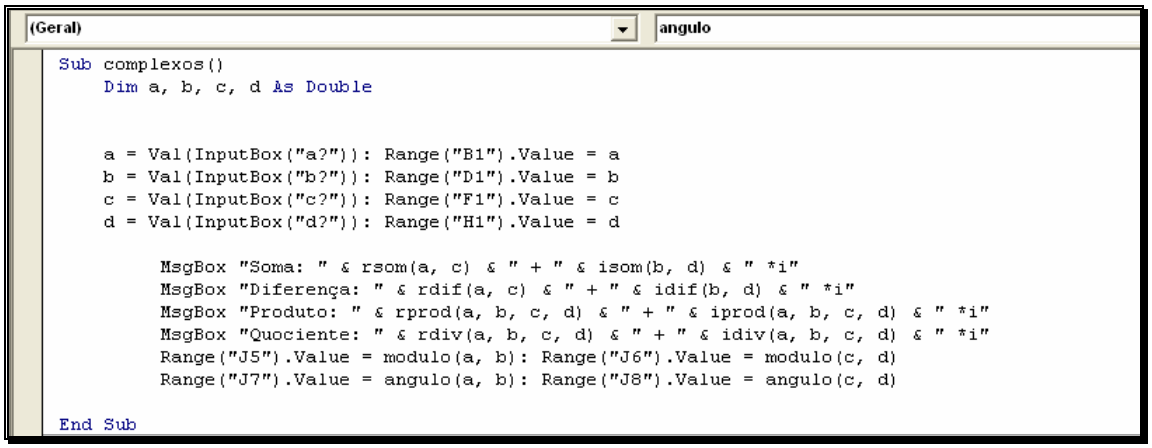
Soma: z1+z2 = (a+c)+(b+d)i
Diferença: z1-z2 = (a-c)+(b-d)i
Produto: z1*z2 = (ac-bd)+(ad+bc)i
Quociente: z1/z2 = x + yi, onde

$$x = \frac{ac + bd}{c^2 + d^2} \quad \text{e} \quad y = \frac{bc - ad}{c^2 + d^2}$$

Para determinar a forma polar, $z = r(\cos\varphi + i \cdot sen\varphi)$, do número complexo z, é preciso calcular o módulo $r = \sqrt{a^2 + b^2}$ e o ângulo $\varphi = \tan^{-1}(\frac{b}{a})$.

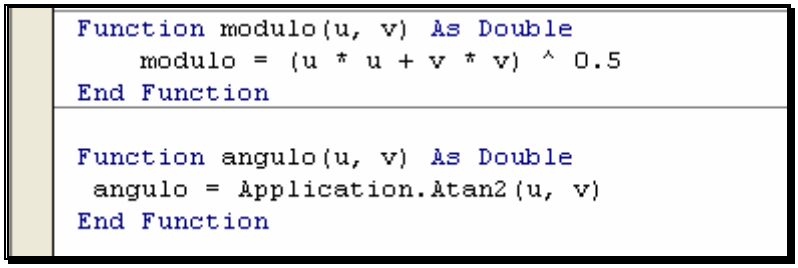
Para o arco tangente, o Excel tem embutido as funções =ATAN2(x;y) e =ATAN(x). A segunda retorna só ângulos em [o;π], mas nós precisamos ângulos em [-π;π], que são retornados pela primeira função =ATAN2(x;y). A função *Atn* do VBA retorna o resultado em radianos apenas entre -π/2 e π/2. Por isso obtemos para z=-1+1i um ângulo de -0.78539816 radianos = -45° em vez dos 135° esperados. (É altamente aconselhável fazer se um esboço do vetor correspondente ao número complexo.)

Modificação do program principal para a avaliação dos módulos e dos ângulos:



Mas, afortunadamente, existe uma saída deste problema com *Atn*. Pois no VBA é permitido usar também as funções próprias do Excel (Worksheet Functions, funções de planilha de trabalho).
É só necessário substituir a linha `angulo = Atn(v/u)` por a linha `angulo = Application.Atan2(u,v)`, isso é tudo. (Veja o comentário no final desta seção.)
Application tem o significado de "worksheet". Com este pequeno truco obtemos ângulos aceitáveis, por exemplo: z = -4 + 6i tem o ângulo φ = 123,69° (= 2,1588 Radianos).

Mas, não devemos esquecer adicionar as duas funções seguintes:



Agora vou revelar um segredo, pois podemos também encontrar os Números Complexos no Excel na categoria *Engenharia*. Para efetuar os cálculos elementares, soma, diferença, produto, ... precisamos da função COMPLEXO para converter os coeficientes reais e imaginários em números complexos no formato x + yi ou x + yj. (Se não usamos o "Sufixo" i ou j, o Excel põe automaticamente i .)

Por exemplo: se B1: -0,5 e D1: -0,866, então =COMPLEXO(B1;D1) retorna o número complexo -0,5-0,866i, compare a primeira planilha acima. Determinamos a soma de z1 e z2 com a função IMSOMA.

Resultado: =IMSOMA(P2;Q2)= -1,5+0,134i.

Veja também a calculadora que vamos criar no capítulo 18.

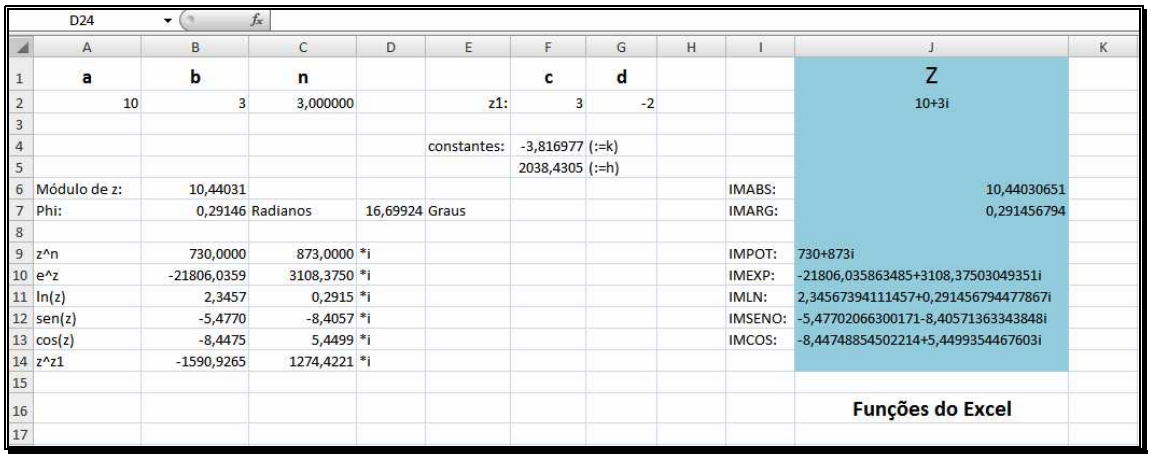
Funções de números complexos.

A seguinte lista contem algumas das importantes funções de números complexos, nas quais usamos as funções RAIZ, EXP, LN, ATAN2, SEN, COS :

1.	$z^n = r^n(\cos n\varphi + i \operatorname{sen} n\varphi)$ (*De Moivre*,1667-1754)
2.	$e^z = e^a(\cos b + i \operatorname{sen} b)$
3.	$\ln z = \ln r + i\varphi$; $(-\pi < \varphi <= \pi)$
4.	$\operatorname{sen} z = 0.5(e^b + e^{-b}) \operatorname{sen} a + 0.5(e^b - e^{-b}) \cos a \cdot i$
5.	$\cos z = 0.5(e^b + e^{-b}) \cos a - 0.5(e^b - e^{-b}) \operatorname{sen} a \cdot i$
6.	$z^{z1} = h\cos k + h\operatorname{sen} k \cdot i$; onde z1 = c+id e $h = r^c e^{-d\varphi}$; $k = d\ln r + c\varphi$

Veja a planilha a seguir que foi criada com as seguintes expressões:

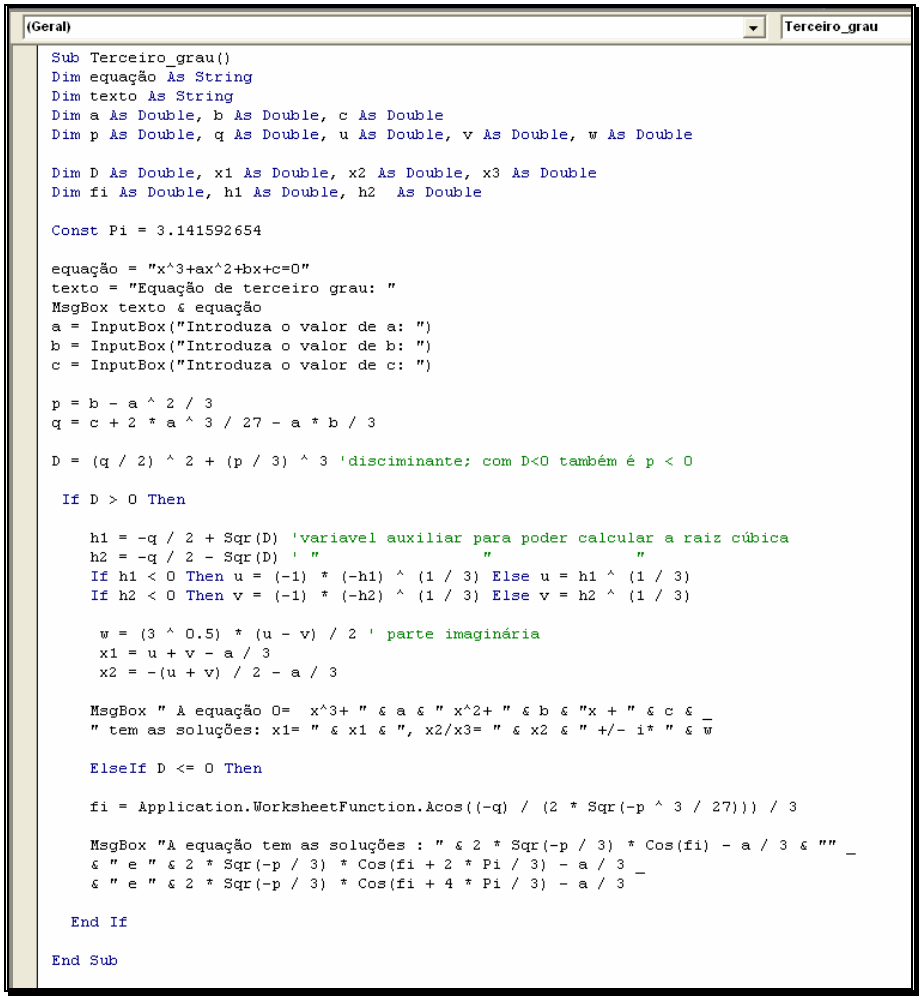
1. B6: =RAIZ(A2^2+B2^2); B7: =ATAN2(A2;B2)
   B9: =B6^C2*COS(C2*B7); C9: =B6^C2*SEN(C2*B7)
2. B10: =EXP(A2)*COS(B2); C10: =EXP(A2)*SEN(B2)
3. B11: =LN(B6) C11: =B7
4. B12: =0,5*(EXP(B2)+EXP(-B2))*SEN(A2)
   C12: =0,5*(EXP(B2)-EXP(-B2))*COS(A2)
5. B13: =0,5*(EXP(B2)+EXP(-B2))*COS(A2)
   C13: =-0,5*(EXP(B2)-EXP(-B2))*SEN(A2)
6. B14: =F5*COS(F4); C14: =F5*SEN(F4)
   com: F4: =G2*LN(B6)+F2*B7
   F5: =B6^F2*EXP(-G2*B7)



	A	B	C	D	E	F	G	H	I	J	K
1	**a**	**b**	**n**			**c**	**d**			**Z**	
2	10	3	3,000000		z1:	3	-2			10+3i	
3											
4					constantes:	-3,816977 (:=k)					
5						2038,4305 (:=h)					
6	Módulo de z:	10,44031							IMABS:	10,44030651	
7	Phi:	0,29146	Radianos	16,69924	Graus				IMARG:	0,291456794	
8											
9	z^n	730,0000	873,0000	*i					IMPOT:	730+873i	
10	e^z	-21806,0359	3108,3750	*i					IMEXP:	-21806,035863485+3108,37503049351i	
11	ln(z)	2,3457	0,2915	*i					IMLN:	2,345673941111457+0,291456794477867i	
12	sen(z)	-5,4770	-8,4057	*i					IMSENO:	-5,47702066300171-8,40571363343848i	
13	cos(z)	-8,4475	5,4499	*i					IMCOS:	-8,44748854502214+5,4499354467603i	
14	z^z1	-1590,9265	1274,4221	*i							
15											
16										**Funções do Excel**	
17											

O Excel permite introduzir, em estes casos, um número complexo como *String* e sem o uso da função COMPLEXO, veja 10+3i na célula J2.

Equações do terceiro grau



(Geral) — Terceiro_grau

```
Sub Terceiro_grau()
Dim equação As String
Dim texto As String
Dim a As Double, b As Double, c As Double
Dim p As Double, q As Double, u As Double, v As Double, w As Double

Dim D As Double, x1 As Double, x2 As Double, x3 As Double
Dim fi As Double, h1 As Double, h2  As Double

Const Pi = 3.141592654

equação = "x^3+ax^2+bx+c=0"
texto = "Equação de terceiro grau: "
MsgBox texto & equação
a = InputBox("Introduza o valor de a: ")
b = InputBox("Introduza o valor de b: ")
c = InputBox("Introduza o valor de c: ")

p = b - a ^ 2 / 3
q = c + 2 * a ^ 3 / 27 - a * b / 3

D = (q / 2) ^ 2 + (p / 3) ^ 3 'disciminante; com D<0 também é p < 0

 If D > 0 Then

    h1 = -q / 2 + Sqr(D) 'variavel auxiliar para poder calcular a raiz cúbica
    h2 = -q / 2 - Sqr(D) ' "                "                "
    If h1 < 0 Then u = (-1) * (-h1) ^ (1 / 3) Else u = h1 ^ (1 / 3)
    If h2 < 0 Then v = (-1) * (-h2) ^ (1 / 3) Else v = h2 ^ (1 / 3)

     w = (3 ^ 0.5) * (u - v) / 2 ' parte imaginária
     x1 = u + v - a / 3
     x2 = -(u + v) / 2 - a / 3

    MsgBox " A equação 0=  x^3+ " & a & " x^2+ " & b & "x + " & c & _
    " tem as soluções: x1= " & x1 & ", x2/x3= " & x2 & " +/- i* " & w

    ElseIf D <= 0 Then

    fi = Application.WorksheetFunction.Acos((-q) / (2 * Sqr(-p ^ 3 / 27))) / 3

    MsgBox "A equação tem as soluções : " & 2 * Sqr(-p / 3) * Cos(fi) - a / 3 & "" _
    & " e " & 2 * Sqr(-p / 3) * Cos(fi + 2 * Pi / 3) - a / 3 _
    & " e " & 2 * Sqr(-p / 3) * Cos(fi + 4 * Pi / 3) - a / 3

  End If

End Sub
```

Em 1539, um médico e cientista, rico e influente na época, Girolamo Cardano (1501-1576), obteve de Tartaglia a regra para se resolver a equação do terceiro grau, sob a forma de versos enigmáticos, sem demonstração. Mas Cardano jurou a Tartaglia que não divulgaria a regra. Tartaglia (1499-1557) era eminente professor em Veneza.

Hoje em dia, utiliza-se, geralmente, a seguinte regra *sem* versos –mas um pouco enigmático, sim (*fórmula de Cardano*):

1. Para resolver a equação $x^3 + ax^2 + bx + c = 0$ precisamos da discriminante

$$D = \left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3 \text{ com } p = b - \frac{a^2}{3} \text{ e } q = c + \frac{2}{27}a^3 - ab/3$$

2. Se D for positiva, resultarão uma solução real e duas soluções complexas conjugadas

$$x_1 = u + v - a/3$$
$$x_2 = -(u+v)/2 - a/3 \pm i\sqrt{3}(u-v)/2$$

3. As constantes *u* e *v* são dadas por

$$u = (-\frac{q}{2} + \sqrt{D})^{\frac{1}{3}} \; ; \; v = (-\frac{q}{2} - \sqrt{D})^{\frac{1}{3}}$$

4. Se D <= 0, teremos as seguintes soluções reais

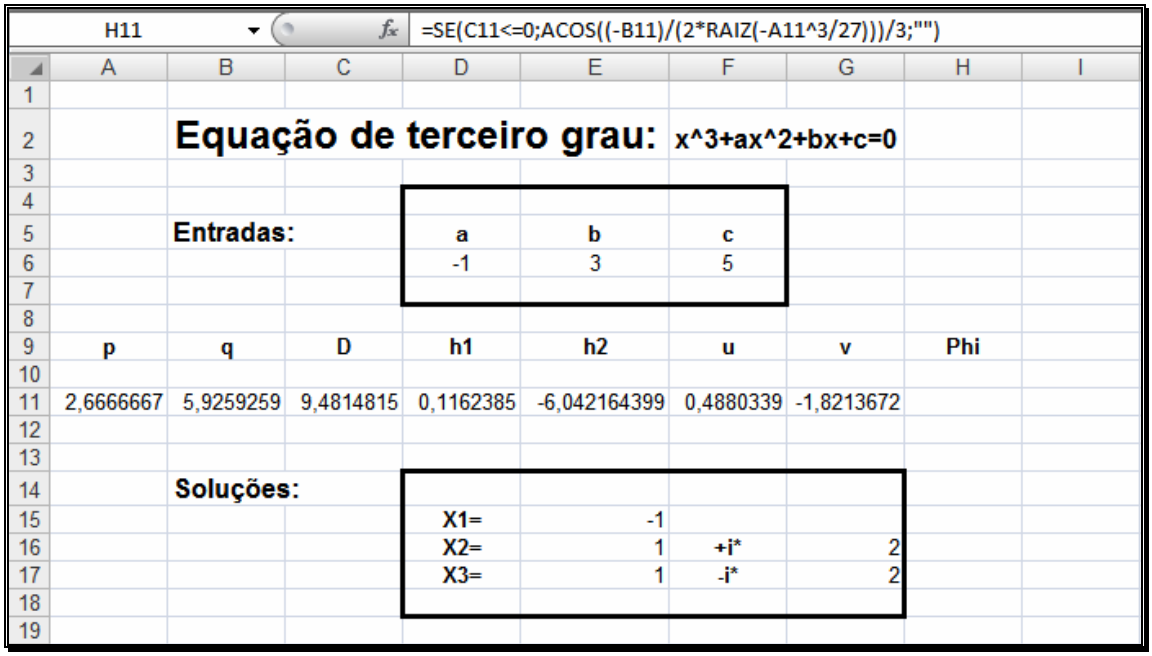
$$x_1 = 2\sqrt{-\frac{p}{3}}\cos(\varphi) - \frac{a}{3}$$
$$x_2 = 2\sqrt{-\frac{p}{3}}\cos(\varphi + 120^o) - \frac{a}{3}$$
$$x_3 = 2\sqrt{-\frac{p}{3}}\cos(\varphi + 240^o) - \frac{a}{3}$$

O ângulo φ está dado por $\varphi = \frac{1}{3}\arccos\frac{-q}{2\sqrt{-(\frac{p}{3})^3}}$

No código do program VBA "Sub Terceiro_grau()" foi preciso utilizar uma "Worksheet Function" para determinar o ângulo φ. (Veja acima debaixo da caixa "Complexos" na página 93.)

(Você pode usar a maioria das funções de planilha do Microsoft Excel em suas instruções de Visual Basic. Para obter uma lista das funções de planilha que você pode usar, consulte Lista de funções de planilha disponíveis para o Visual Basic.)

A **versão para Excel** foi feita com as entradas dadas abaixo da seguinte planilha:



A11: =E6-D6^2/3; B11: =F6+2*D6^3/27-D6*E6/3
C11: =(B11/2)^2; h1(=D11) e h2 (=E11) foram introduzidas para resolver o problema com raízes de números negativos:
D11: =SE(C11>0;-B11/2+RAIZ(C11);"")
E11: =SE(C11>0;-B11/2-RAIZ(C11);"")
F11: =SE(D11<0;(-1)*(-D11)^(1/3);SE(D11<>"";D11^(1/3);""))
G11: =SE(E11<0;(-1)*(-E11)^(1/3);SE(E11<>"";E11^(1/3);""))
H11: =SE(C11<=0;ACOS((-B11)/(2*RAIZ(-A11^3/27)))/3;"")
E15: =SE(C11>0;+F11+G11-D6/3;2*RAIZ(-A11/3)*COS(H11)-D6/3)
F15 e G15 ficam vazias
E16: =SE(C11>0;-(F11+G11)/2-D6/3;2*RAIZ(-A11/3)*COS(H11+2*PI()/3)-D6/3)
E17: =SE(C11>0;-(F11+G11)/2-D6/3;2*RAIZ(-A11/3)*COS(H11+4*PI()/3)-D6/3)
F17: =F16; G17: =G16