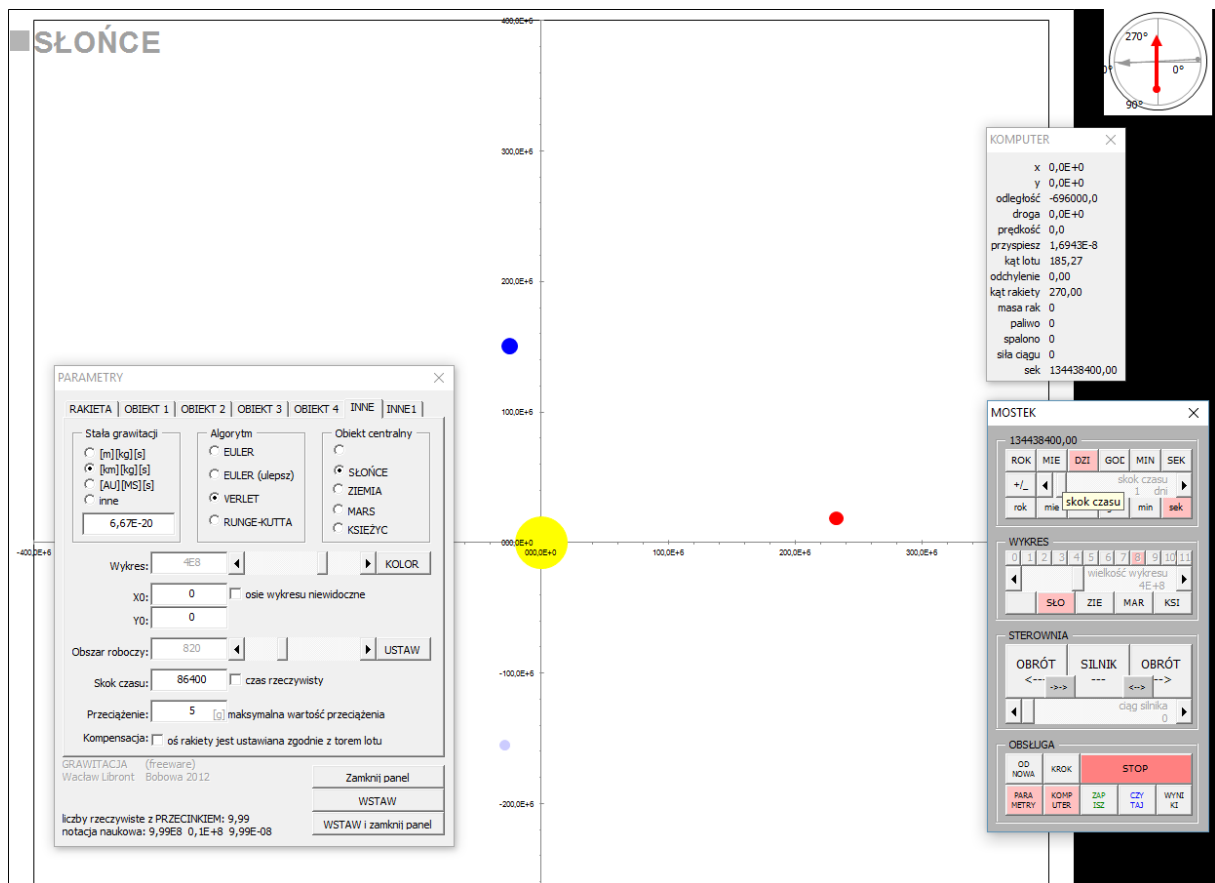


Cyfrowe projekty – GRAWITACJA



Projekt GRAWITACJA służy do przeprowadzania eksperymentów fizycznych na komputerze. Eksperymenty te służą do symulowania ruchu obiektów w polu grawitacyjnym i symulowania lotów kosmicznych. Projekt powstał jako praca dyplomowa z fizyki, którą studiowałem na Wydziale Fizyki i Informatyki Stosowanej mojej ulubionej uczelni, czyli AGH.

```
'WYKRES
Public SKALA_skala As Double           'wielkość wykresu
Public SKALA_wykład As Integer
'WYKRES_xs=SKALA_skala*10^SKALA_wykład
Public WYKRES_xs As Double             'wykres na obu osiach od -WYKRES_xs
do +WYKRES_xs
Public WYKRES_xś As Double             'środek wykresu
Public WYKRES_yś As Double             'nie w (0,0), gdy jako centralny
ustawiony inny obiekt
Public WYKRES_ŚRODEK_NR As Integer     'który obiekt (0-4) ustawiony jako
centralny
'RAKIETA
Public MASA_PALIWA_PO CZ As Double     'początkowa masa paliwa ustawiona na
suwaku [kg]
Public MASA_PALIWA_AKT As Double       'bieżąca masa paliwa [kg]
Public SZYBK OŚĆ_SPALANIA As Double    'szybkość spalania [kg/s]
Public SZYBK OŚĆ_SPALIN As Double      'szybkość wypływu spalin [km/s]
Public SIŁA_CIA GU As Double           'siła ciągu silnika rakiety
=SZYBK OŚĆ_SPALANIA*SZYBK OŚĆ_SPALIN [kN]
Public MASA_RAKIETY As Double          'aktualna masa rakiety z paliwem
```

```

Public SILNIK_PRACUJE As Boolean      'czy silnik pracuje - pole wyboru
Public PALIWO_POBIERAJ As Boolean     'czy silnik pobiera paliwo - pole
wyboru
Public RAKIETA_PRACUJE As Boolean     'czy uwzględniać rakieta w
obliczeniach
nie będzie ulegała grawitacji
'OBROTY-RADAR
Public KĄT_SKOKU_OBROTU As Double     'ile wynosi skok obrotu rakiety
podczas skoku czasu
Public KIERUNEK_OBROTU As Integer     '+1 lub -1 w prawo lub w lewo obrót
Public KĄT_LOTU As Double             'kąt wynikający z wyliczenia
przyspieszeń 0 stopni-kierunek osi 0X i zgodnie z zegarem
Public KĄT_RAKIETY As Double          'kąt pochylenia rakiety - rakieta
leci odchyłona od osi lotu
Public ODCHYLENIE As Double           'odchylenie rakiety od osi lotu
Public OBRACAĆ As Boolean              'przyciski obrotów działają na
zasadzie włącz-wyłącz
' CZAS
Public SKOK_CZASU_PRZ As Long          'ustawiony przyciskami: SEK, MIN,
GOD...
Public SKOK_CZASU_ZNAK As Integer     '+ lub -1
Public SKOK_CZASU As Double           'aktualny skok czasu =
SKOK_CZASU_PRZ*SKOK_CZASU_SUWAK
Public SKOK_CZASU_SUWAK As Double     'na suwaku można dodatkowo powiększyć
skok
'np. zamiast 1 sekundy przycisk SEK
można zrobić 12 sekund
Public SKOK_CZASU_WIDOK As Double     'dolnymi przyciskami na panelu CZAS
można zmienić wymiar czasu
'np. skok czasu w godzinach, a
wyświetlamy czas np. w dniach
Public CZAS_FORMAT As String          'jak sformatowane wyświetlanie czasu
Public CZAS As Double                 'główna zmienna czasu upływającego -
w sekundach
Public CZAS_ZEGARA As Double          'czas systemowego zegara z Timer
'gdy CZAS_REAL=true to puszczaamy
pętlę animacji gdy można
Public CZAS_REAL As Boolean            'jeśli prawda to czas 1 sekundy
rzeczywisty
'jeśli fałsz to ile procesor wytrzyma
bez ograniczeń
Public PUŚĆ_KROK As Integer           'gdy klikamy w przycisk KROK
'animacja - 0 krokowo - 1
Public Const ADRES = "C20"           'do tej komórki skok po operacjach na
elementach graficznych
'za każdym razem trzeba zaznaczyć
obiekt a nie można tego zrobić
'nie będąc na arkuszu - arkusz nie
jest wybrany
Public Const PI = 3.14159265358979
Public Const WYM = 4                  'ile obiektów obsługujemy, wymiar
tablicy
'można więcej, ale tabelki z danymi
i wynikami trzeba by powiększyć
Public GRAW As Double                 'stała grawitacji
Public KONIEC As Boolean               'koniec animacji pętli głównej
Public EcMIN, EcMAX, EcPRO As Double  'energia minimalna, maksymalna i
procent odchyłki
Public KOREK As Boolean                'NIE UŻYWANE

```

```

'pomocne przy wychodzeniu z
korkociągów
'gdz prędkość maleje do zera
'to samodzielnie zaczyna się obracać
Public ADANE As String 'na tym arkuszu ustawienia początkowe
'które obsługują formularze
Public ARKUSZ As String 'tutaj odbywa się animacja
Public ALGORYTM As Integer '1-euler 0-verlet 2-RK4
Public KOMPENSACJA As Boolean 'czy oś rakiety ma być ustawiana jak
oś lotu
Public WŁĄCZ_ZAPISZ As Boolean 'zapisywanie do arkusza WYNIKI
Public ZAPISZ_CYKLE As Long 'po zapisaniu tylu cykli w ZAPISZ
koniec
Public EKTRAN_WYS As Double 'ile zajmują dodatki ekranowe
Public RAKIETA_S As Double 'ile kilometrów przeleciała rakieta
Public RAKIETA_Xpop As Double 'poprzednie położenie rakiety
Public RAKIETA_Ypop As Double '

'0-rakieta
'1-WYM - obiekty
Public XY(0 To WYM, 1 To 2) As Double 'współrzędne 1-X, 2-Y
Public VV(0 To WYM, 1 To 2) As Double 'prędkości 1-Vx, 2-Vy
Public AA(0 To WYM, 1 To 2) As Double 'przyspieszenia 1-ax, 2ay
Public MA(0 To WYM) As Double 'masy
Public RR(0 To WYM) As Double 'promienie
'do RK4
Public VV1(0 To WYM, 1 To 2) As Double 'prędkości 1-Vx, 2-Vy
Public VV2(0 To WYM, 1 To 2) As Double 'prędkości 1-Vx, 2-Vy
Public VV3(0 To WYM, 1 To 2) As Double 'prędkości 1-Vx, 2-Vy
Public VV4(0 To WYM, 1 To 2) As Double 'prędkości 1-Vx, 2-Vy
Public VV0(0 To WYM, 1 To 2) As Double 'prędkości 1-Vx, 2-Vy
Public XY0(0 To WYM, 1 To 2) As Double 'prędkości 1-Vx, 2-Vy

'ta procedura uruchamia się automatycznie podczas startu arkusza
kalkulacyjnego
'rozdzielczość
Public Declare Function GetSystemMetrics _
    Lib "user32.dll" (ByVal nIndex As Long) As Long
Const SM_CXSCREEN = 0
Const SM_CYSCREEN = 1

Public Function TrybVideo_WY() As Integer
    Dim vidSzer As Integer
    Dim vidWys As Integer
    vidSzer = GetSystemMetrics(SM_CXSCREEN)
    vidWys = GetSystemMetrics(SM_CYSCREEN)
    TrybVideo_WY = vidWys
End Function
Public Function TrybVideo_SZ() As Integer
    Dim vidSzer As Integer
    Dim vidWys As Integer
    vidSzer = GetSystemMetrics(SM_CXSCREEN)
    vidWys = GetSystemMetrics(SM_CYSCREEN)
    TrybVideo_SZ = vidSzer
End Function

Sub auto_open()
On Error GoTo bład:
    Application.ScreenUpdating = False
    ARKUSZ = "GRAWITACJA"

```

```

ADANE = "DANE"
KONIEC = True
'ustawienie przycisku start
UF_MOSTEK.CommandButton5.Caption = "START"
UF_MOSTEK.CommandButton5.BackColor = &H80000002 ' &H8000000F
Sheets("GRAWITACJA").Select

'maksymalizacja, schować paski
'paski i menu pasek formuły i stanu
  With ActiveWindow
    .DisplayHorizontalScrollBar = False
    .DisplayVerticalScrollBar = False
    .DisplayWorkbookTabs = False
    .DisplayHeadings = False
  End With
  With Application
    .WindowState = xlMaximized
    .CommandBars.ActiveMenuBar.Enabled = False
    .CommandBars("Standard").Visible = False
    .CommandBars("Formatting").Visible = False
    .ReferenceStyle = xlA1
    .DisplayFormulaBar = False
    .DisplayStatusBar = False
    .DisplayFormulaBar = False
    .DisplayStatusBar = False
    .ShowWindowsInTaskbar = False
  End With
  EKRAN_WYS = 3.15
  w_m = 320
  w_k = 210
  wys = TrybVideo_WY
  sze = TrybVideo_SZ
  wys = wys - Int(wys / EKRAN_WYS)
  'panel komp
  If wys - (w_m + w_k) > 60 Then
    Sheets("DANE").Range("B36") = wys + 10
    Sheets("DANE").Range("B37") = wys - (w_m + w_k)
  Else
    If sze > 1024 Then
      Sheets("DANE").Range("B36") = wys + 194
    Else
      Sheets("DANE").Range("B36") = wys - 120
    End If
    Sheets("DANE").Range("B37") = wys - w_k
  End If
  'panel mostek
  Sheets("DANE").Range("B38") = wys + 10
  Sheets("DANE").Range("B39") = wys - w_m

'resetujemy wszystkie ustawienia
KLAWISZ_ustaw
UF_DANE.DOPASUJ_WYSOKOŚĆ

UF_MOSTEK.Show
UF_KOMPUTER.Show
UF_MOSTEK.CommandButton18.BackColor = &HC0C0FF
BŁOKADA_OFF
Application.ScreenUpdating = True
Exit Sub
błąd:
Resume Next

```

```

End Sub
'przywrócenie pasków podczas zamykania
Sub auto_close()
On Error GoTo błąd:
    With ActiveWindow
        .DisplayHorizontalScrollBar = True
        .DisplayVerticalScrollBar = True
        .DisplayWorkbookTabs = True
        .DisplayHeadings = True
    End With
    With Application
        .CommandBars.ActiveMenuBar.Enabled = True
        .CommandBars("Standard").Visible = True
        .CommandBars("Formatting").Visible = True
        .ReferenceStyle = xlA1
        .DisplayFormulaBar = True
        .DisplayStatusBar = True
        .DisplayFormulaBar = True
        .DisplayStatusBar = True
        .ShowWindowsInTaskbar = True
    End With
Exit Sub
błąd:
    Resume Next
End Sub
'odblokowanie arkusza i ustawienie komórki aktywnej
'żeby można było wybrać (selection) obiekt formularza do zmiany
'można zabezpieczać i odbezpieczać, ale zwalnia i mruga
Public Sub BLOKADA_OFF()
    'ActiveSheet.Unprotect
    Sheets("GRAWITACJA").Range(ADRES).Select
End Sub
'na wszelki wypadek znów komórkę wybieramy
'zablokowanie arkusza
Public Sub BLOKADA_ON()
    Sheets("GRAWITACJA").Range(ADRES).Select
    'ActiveSheet.Protect
End Sub

'uruchamianie rakiety - klawisz START-STOP
Sub KLAWISZ_start()
    If KONIEC = True Then
        KONIEC = False
        UF_MOSTEK.CommandButton5.Caption = "STOP"
        UF_MOSTEK.CommandButton5.BackColor = &H8080FF    '&HC0C0FF
        'główna pętla animacji
        ANIMACJA
    Else
        KONIEC = True
        UF_MOSTEK.CommandButton5.Caption = "START"
        UF_MOSTEK.CommandButton5.BackColor = &H80000002 '&H8000000F
    End If
End Sub

'pobieranie danych z komórek arkusza do tablic - tylko za pierwszym razem
Sub PobierzDane()
    For i = 0 To WYM
        w = i + 3
        MA(i) = Sheets("DANE").Cells(9, i + 2)
        RR(i) = Sheets("DANE").Cells(10, i + 2)
        'dane bierzące

```

```

w = i + 10
XY(i, 1) = Sheets("DANE").Cells(1, i + 2) 'Cells(w, 3)
XY(i, 2) = Sheets("DANE").Cells(2, i + 2) 'Cells(w, 4)
VV(i, 1) = Sheets("DANE").Cells(3, i + 2) 'Cells(w, 5)
VV(i, 2) = Sheets("DANE").Cells(4, i + 2) 'Cells(w, 6)
Next i
RAKIETA_Xpop = XY(0, 1)
RAKIETA_Ypop = XY(0, 2)
RAKIETA_S = 0 'Sqr((0 - RAKIETA_Xpop) ^ 2 + (0 - RAKIETA_Ypop) ^ 2)
End Sub

'reset ustawień - przywracamy wszystko do początkowych ustawień
Sub KLAWISZ_ustaw()
On Error GoTo błąd
Application.ScreenUpdating = False
'licznik
Sheets("GRAWITACJA").Range("B16") = 0
'ENERGIA - początkowe wartości minimum i maksimum
EcMIN = 1E+40
EcMAX = -1E+40
EcPRO = 0
RAKIETA_S = 0
Sheets("GRAWITACJA").Range("C15") = EcMIN
Sheets("GRAWITACJA").Range("D15") = EcMAX
Sheets("GRAWITACJA").Range("E15") = EcPRO
'ENERGIA_oblicz
'najpierw wyzerujemy tabelkę chwilowych wartości
'wyzerować też masy bo pamiętało i obliczało
BLOKADA_OFF
For w = 3 To 7
For k = 3 To 8
Sheets("GRAWITACJA").Cells(w + 7, k) = 0
Next k
MA(w - 3) = 0
Next w
'przepisujemy wartości początkowe do tabeli chwilowych
For w = 10 To 14
For k = 3 To 6
Sheets("GRAWITACJA").Cells(w, k) = Sheets("DANE").Cells(k - 2, w - 8)
'jeśli są zera to ustawia na wykresie obiekty
'gdz pusto, to obiektów początkowo nie widać
If Sheets("GRAWITACJA").Cells(w, k) = 0 Then Sheets("DANE").Cells(k -
2, w - 8) = 0
Next k
Next w
'czas w symulacji
CZAS = 0
SKOK_CZASU_ZNAK = 1
CZAS_ZEGARA = Timer
UF_MOSTEK.CZAS_USTAW_POCZ
UF_MOSTEK.CZAS_NA_PANEL
UF_MOSTEK.Frame1.Caption = "CZAS"
'jeszcze raz suwak bo poprzednie psują
SKOK_CZASU_PRZ = Sheets("DANE").Range("B42")
SKOK_CZASU_SUWAK = Sheets("DANE").Range("C35")
'zabezpieczenie starsze wersje
If SKOK_CZASU_SUWAK = 0 Then SKOK_CZASU_SUWAK = 100
UF_MOSTEK.ScrollBar5.Value = SKOK_CZASU_SUWAK
SKOK_CZASU_ZNAK = 1
'pobranie wartości stałej grawitacji
GRAW = Sheets("DANE").Range("B33")

```

```

'wykres na mostku
WYKRES_ŚRODEK_NR = Sheets("DANE").Range("B34")
SKALA_skala = Sheets("DANE").Range("B19")
SKALA_wykład = Sheets("DANE").Range("B20")
UF_MOSTEK.Label3.Caption = CStr(SKALA_skala)
UF_MOSTEK.Label6.Caption = "E+" + CStr(SKALA_wykład)
UF_MOSTEK.ScrollBar3.Value = Int(SKALA_skala * 100)
UF_MOSTEK.ScrollBar1.Value = SKALA_wykład
UF_MOSTEK.SKALA_WYKŁAD_USTAW
'nazwy przycisków
UF_MOSTEK.CommandButton20.Caption = Left(Sheets("DANE").Range("C23"), 3)
UF_MOSTEK.CommandButton21.Caption = Left(Sheets("DANE").Range("C24"), 3)
UF_MOSTEK.CommandButton22.Caption = Left(Sheets("DANE").Range("C25"), 3)
UF_MOSTEK.CommandButton23.Caption = Left(Sheets("DANE").Range("C26"), 3)
UF_MOSTEK.CommandButton24.Caption = Left(Sheets("DANE").Range("C27"), 3)
'gdę badamy samą raketę, to się przydało żeby nie była w środku,
'bo inaczej nie liczyło
'na panelu zznaczyć przycisk
UF_MOSTEK.OBIEKTY_kolorowy WYKRES_ŚRODEK_NR
ALGORYT_M = Sheets("DANE").Range("C22")
'dane z tabeli wstawiamy do tabel
PobierzDane
'masa paliwa początkowa i bieżąca taka sama
MASA_RAKIETY = Sheets("DANE").Range("B9")
MASA_PALIWA_PO CZ = Sheets("DANE").Range("B11")
MASA_PALIWA_AKT = MASA_PALIWA_PO CZ
SZYBK OŚĆ_SPALANIA = Sheets("DANE").Range("B12")
SZYBK OŚĆ_SPALIN = Sheets("DANE").Range("B13")
SIŁA_CIAGU = SZYBK OŚĆ_SPALANIA * SZYBK OŚĆ_SPALIN * 9.81 'km*kg/s^2
UF_MOSTEK.SIŁA_CIAGU_ustaw (CLng(SIŁA_CIAGU))
KĄT_SKOKU_OBROTU = Sheets("DANE").Range("B8")
'wyliczenie odchylenia osi lotu od osi rakiety
KĄT_LOTU = Sheets("DANE").Range("B6")
KĄT_RAKIETY = Sheets("DANE").Range("B7")
ODCHYLENIE = KĄT_GRANICA(KĄT_RAKIETY - KĄT_LOTU)
RYSUJ_STRZAŁKI
OBRACAĆ = False
KOREK = False
KIERUNEK_OBROTU = 0
If Sheets("DANE").Range("C34") = "" Then
    KOMPENSACJA = False
Else
    KOMPENSACJA = Sheets("DANE").Range("C34")
End If
UF_MOSTEK.CommandButton1.BackColor = &H8000000F
UF_MOSTEK.CommandButton2.BackColor = &H8000000F
'czy uwzględniać raketę w animacji, bo można obserwować same planety,
'żeby od razu nie zderzyła się, jeśli ustawiona na powierzchni
'jeśli włączone to zmieniamy, a potem procedurą przywracamy
RAKIETA_PRACUJE = Not (Sheets("DANE").Range("B16"))
RAKIETA_onoff
'silnik może być włączony od startu - start z ziemi
'ustawiamy przeciwnie, a potem od razu zmieniamy
SILNIK_PRACUJE = Not (Sheets("DANE").Range("B15"))
SILNIK_onoff
'nie pobiera paliwa, gdy włączony silnik - żeby nie denerwować
niedoświadczonego użytkownika
PALIWO_POBIERAJ = False
PALIWO_ustaw
RAKIETA_oblicz
PALIWO_pobierajONOFF

```

```

WYKRES_osie
WYKRES_środek
'środek wykresu wyliczamy żeby rysować kółka dobrze
WYKRES_oblicz
'kółko radaru
ActiveSheet.Shapes.Range(Array("TARCZA", "R_KIERUNEK",
"R_SILNIK")).Select
Selection.Left = Sheets("DANE").Range("B40") + 45
Selection.Top = 25
BLOKADA_OFF

'nazwa pliku na wykres
Sheets("GRAWITACJA").Label1.Caption = CStr(Sheets("DANE").Range("C21"))
'żeby wystartować od zera
KONIEC = False
KLAWISZ_start
'automatyczny zas do arkusza ZAPISZ
WŁĄCZ_ZAPISZ = False
UF_MOSTEK.CommandButton44.BackColor = &H8000000F
Application.ScreenUpdating = True
Exit Sub
błąd:
Resume Next
End Sub

'WYKRES*****
*
'wyliczanie wielkości wykresu, maksymalnych i minimalnych wartości na
osiach
Sub WYKRES_osie()
    Sheets("GRAWITACJA").Select
    ActiveSheet.ChartObjects("W_wykres").Activate
    ActiveChart.PlotArea.Select
    ActiveChart.Axes(xlCategory).Select
    osie = Sheets("DANE").Range("B22")
    If osie = True Then
        'jak = Selection.Border.LineStyle
        'If jak = xlAutomatic Then
            With Selection.Border
                .Weight = xlHairline
                .LineStyle = xlNone
            End With
            With Selection
                .MajorTickMark = xlNone
                .MinorTickMark = xlNone
                .TickLabelPosition = xlNone
            End With
            ActiveChart.Axes(xlValue).Select
            With Selection.Border
                .Weight = xlHairline
                .LineStyle = xlNone
            End With
            With Selection
                .MajorTickMark = xlNone
                .MinorTickMark = xlNone
                .TickLabelPosition = xlNone
            End With
        Else
            With Selection.Border
                .Weight = xlHairline
            End With
        End If
    End If
End Sub

```



```

        .LineStyle = xlAutomatic
    End With
    With Selection
        .MajorTickMark = xlInside
        .MinorTickMark = xlOutside
        .TickLabelPosition = xlNextToAxis
    End With
    ActiveChart.Axes(xlValue).Select
    With Selection.Border
        .Weight = xlHairline
        .LineStyle = xlAutomatic
    End With
    With Selection
        .MajorTickMark = xlInside
        .MinorTickMark = xlOutside
        .TickLabelPosition = xlNextToAxis
    End With
End If
End Sub

'pobierz z pokrętła, która planeta na środku i narysuj wykres
'położenie planet z tabeli chwilowe na arkuszu, a nie z tablicy
Sub WYKRES_środek()
On Error GoTo błąd
    Application.ScreenUpdating = False
    xś = Sheets("GRAWITACJA").Cells(10 + WYKRES_ŚRODEK_NR, 3)
    yś = Sheets("GRAWITACJA").Cells(10 + WYKRES_ŚRODEK_NR, 4)

    'przesunięcie środka
    'ale bardzo wygodne - KINEMATYKA
    xp = Sheets("DANE").Range("C19")
    yp = Sheets("DANE").Range("C20")

    xs = SKALA_skala * 10 ^ SKALA_wykład
    ActiveSheet.ChartObjects("W_wykres").Activate
    If SKALA_wykład > 4 Then
        f = "##0.0E+0"
    Else
        f = "General"
    End If
    With ActiveChart.Axes(xlCategory)
        .MinimumScale = -xs + xś - xp
        .MaximumScale = xs + xś - xp
        .CrossesAt = xś
        .TickLabels.NumberFormat = f
    End With
    With ActiveChart.Axes(xlValue)
        .MinimumScale = -xs + yś - yp
        .MaximumScale = xs + yś - yp
        .CrossesAt = yś
        .TickLabels.NumberFormat = f
    End With
    KÓŁKO_obsługa
    Application.ScreenUpdating = True
błąd:
    Application.ScreenUpdating = True
' BLOKADA_ON
End Sub

'położenie wykresu sprawdzamy po przycisku OD NOWA
'bo użytkownik mógł zmienić wymiar i położenie

```

```
'a my chcemy rosować dokładnie w środku o odpowiedniej wielkości planety
Sub WYKRES_oblicz()
  With ActiveSheet.Shapes("w_wykres")
    s = .Width
    w = .Height
    g = .Top
    l = .Left
  End With
  'środek wykresu na ekranie
  WYKRES_xs = l + s / 2
  WYKRES_ys = g + w / 2
  'średnia z wysokości i szerokości na ekranie - połowa nas interesuje
  WYKRES_xs = (s + w) / 4
End Sub
```

'rysowanie półprzezroczystego koła-planety o odpowiedniej wielkości w środku wykresu

```
Sub KÓŁKO_rysuj(promień As Variant)
On Error GoTo błąd
  'maksymalna wartość odciętych i rzędnych na wykresie
  xs = SKALA_skala * 10 ^ SKALA_wykład
  'wyliczamy średnicę rzeczywistą planety
  'powinno być *2 ale tak ładniej wygląda
  śre = promień * 1.95
  'średnica koła na wykresie
  wykres_śre = śre * WYKRES_xs / xs
  'lewy górny róg kółka - bo tak rysuje excel
  x = WYKRES_xs - wykres_śre / 2
  Y = WYKRES_ys - wykres_śre / 2
  ko = Sheets("DANE").Cells(23 + WYKRES_ŚRODEK_NR, 2).Interior.ColorIndex

  'sprawdzamy czy rysowane koło nie za małe lub za duże
  If wykres_śre > 20 And wykres_śre < WYKRES_xs * 1.5 Then
    'dodawanie na arkusz nowego obiektu o nazwie w_kółko
    With ActiveSheet.Shapes.AddShape(msoShapeOval, x, Y, wykres_śre,
wykres_śre)
      .Name = "w_kółko"
      .Fill.Visible = msoFalse
      .Line.Weight = 1
      .Line.DashStyle = msoLineSquareDot
      '.Line.ForeColor.SchemeColor=ko
      .Line.ForeColor.RGB = RGB(192, 192, 192)
      .Fill.Visible = msoTrue
      '.Fill.ForeColor.SchemeColor = ko
      .Fill.ForeColor.RGB = RGB(192, 192, 192)
      .Fill.Transparency = 0.8
    End With
  End If
błąd:
BLOKADA_ON
End Sub
```

'wymazujemy narysowane kółko z akranu
 'w danej chwili tylko jedno i to jedno właśnie usuwamy
 'gdyby coś nie tak, to nie wymaże wszystkich!

```
Sub KÓŁKO_wymaż()
On Error GoTo błąd
  BLOKADA_OFF
  ActiveSheet.Shapes("w_kółko").Select
  Selection.Delete
błąd:
```

```
BLOKADA_ON
End Sub
```

```
'podczas animacji cały czas kółko jest najpierw wymazywane
'i zaraz potem rysowane od nowa - i tak w koło
'za pierwszym razem nie ma czaego wymazać, dlatego obsługa błędu
Sub KÓŁKO_obsługa()
'czy przesuwamy punkt przecięcia osi
xp = Sheets("DANE").Range("C19")
yp = Sheets("DANE").Range("C20")
KÓŁKO_wymaż
promień = Sheets("DANE").Cells(10, WYKRES_ŚRODEK_NR + 2)
'Cells(WYKRES_ŚRODEK_NR + 3, 8)
'co prawda nie używane ale na wszelki wypadek
If xp = 0 And yp = 0 Then KÓŁKO_rysuj promień
End Sub
```

```
'RAKIETA
*****
```

```
'wyliczanie chwilowego przyspieszenia pochodzącego od siły ciągu rakiety
'siła ciągu zawsze działa zgodnie z kierunkiem pochylenia rakiety
'który może być inny niż kierunek lotu wyliczony z grawitacji
Sub AA_rakieta()
Dim ax, ay, vx, vy, a, v, vx1, vy1, ac, acx, acy As Double
On Error GoTo błąd
```

```
'wartości chwilowe przyspieszeń i prędkości rakiety
ax = AA(0, 1)
ay = AA(0, 2)
vx = VV(0, 1)
vy = VV(0, 2)
'wypadkowe
a = (ax ^ 2 + ay ^ 2) ^ (1 / 2)
v = (vx ^ 2 + vy ^ 2) ^ (1 / 2)
```

```
'przyspieszenia po osiach rakiety wyliczamy z Talesa
'proporcjonalne do wektorów prędkości rakiety
'przyspieszenie zgodne ze zwrotem przepustnicy rakiety
'i uwzględniamy obrót automatyczny
kał = KAŁ_GRANICA(KAŁ_RAKIETY + KAŁ_SKOKU_OBROTU)
kał = KAŁ_GRANICA(KAŁ_RAKIETY + KIERUNEK_OBROTU)
kał = KAŁ_RAKIETY
'wyliczone kierunki prędkości
vx1 = WERSOR_X(kał)
vy1 = WERSOR_Y(kał)
'nowe prędkości składowe po zmianie kąta
'po co to było liczone?
'vx = v * Abs(Cos(kał * PI / 180)) * vx1
'vy = v * Abs(Sin(kał * PI / 180)) * vy1
```

```
'przyspieszenie uwzględnia masę chwilową rakiety - Ciołkowski
'nie przeliczamy na inne jednostki, bo za szybko
'jest w N a my chcemy na kilogramy?
ac = SIŁA_CIAГУ / MASA_RAKIETY
'przyspieszenie ciągu działa w tą stronę, w którą jest zwrócona rakieta
acx = ac * Abs(Cos(kał * PI / 180)) * vx1
acy = ac * Abs(Sin(kał * PI / 180)) * vy1
'nowe składowe przyspieszeń rakiety
'po uwzględnieniu siły ciągu i kąta odchylenia
```

```

AA(0, 1) = AA(0, 1) + acx
AA(0, 2) = AA(0, 2) + acy
błąd:
End Sub

'jeśli nie chcemy uwzględniać rakiety w animacji
'bo np. ustawiona na powierzchni ziemi do startu
'to zerujemy jej masę w tablicach
'pole wyboru na panelu RAKIETA
Sub RAKIETA_onoff()
    RAKIETA_PRACUJE = Sheets("DANE").Range("B16")
    If RAKIETA_PRACUJE Then
        MA(0) = Sheets("DANE").Range("B9")
    Else
        MA(0) = 0
    End If
End Sub

'przełącznik na panelu RAKIETA
'jeśli wyłączymy to podczas pracy silnik a rakiety
'nie spalamy paliwa i nie zmniejsza się masa rakiety
'silnik może działać w nieskończoność
Sub PALIWO_pobierajONOFF()
    PALIWO_POBIERAJ = Sheets("DANE").Range("B14")
End Sub

'masa paliwa i rakiety oraz siła ciągu na panel KOMPUTER POKŁADOWY
Sub PALIWO_ustaw()
    UF_KOMPUTER.Label21.Caption = CStr(MASA_RAKIETY) & " Format(MASA_RAKIETY, " & "0.0")
    UF_KOMPUTER.Label22.Caption = CStr(MASA_PALIWA_AKT) & " Format(MASA_PALIWA_AKT, " & "0.0")
    UF_KOMPUTER.Label23.Caption = CStr(SIŁA_CIAGU)
    UF_MOSTEK.ScrollBar4.Value = SIŁA_CIAGU
    UF_MOSTEK.Label7.Caption = CStr(SIŁA_CIAGU)
End Sub

'obliczenie ilości paliwa i masy rakiety, gdy pracuje silnik
'i zaznaczono, że ma pobierać paliwo
Sub PALIWO_oblicz()
    If PALIWO_POBIERAJ And SILNIK_PRACUJE Then
        If MASA_PALIWA_AKT <= 0 Then
            PALIWO_POBIERAJ = False
            SILNIK_onoff 'wyłączenie silnika
        Else
            'powinniśmy uwzględniać skok czasu, ale bardzo szybko by spalało
            'nie uwzględniamy czasu spalania dlatego bez * CZAS_SKOKU
            MASA_PALIWA_AKT = MASA_PALIWA_AKT - SZYBKOŚĆ_SPALANIA * SKOK_CZASU
            MASA_RAKIETY = MASA_RAKIETY - SZYBKOŚĆ_SPALANIA * SKOK_CZASU
            MA(0) = MASA_RAKIETY 'zmniejszyć masę rakiety do obliczeń
        End If
        PALIWO_ustaw
    End If
End Sub

'przycisk SILNIK na panelu MOSTEK KAPITAŃSKI - przełącznik
Sub SILNIK_onoff()
    If SILNIK_PRACUJE = False Then
        SILNIK_PRACUJE = True
        UF_MOSTEK.CommandButton3.BackColor = &HC0C0FF
    Else

```

```
SILNIK_PRACUJE = False
KIERUNEK_OBROTU = 0
UF_MOSTEK.CommandButton3.BackColor = &H8000000F
End If
End Sub
```

```
'przycisk OBRÓT na panelu MOSTEK KAPITAŃSKI
'przycisk OBRÓT<<<< - obrót w lewo - przeciwnie do ruchu wskazówek zegara
'gdy wciśniemy, to rakieta obraca się sama o wybrany kąt w stosunku do osi
lotu
```

```
'prawdziwa zmiana kierunku lotu nastąpi, gdy włączymy silniki
```

```
Sub OBRÓT_lewy()
  If OBRACAĆ = False Then
    OBRACAĆ = True
    KOREK = False
    KIERUNEK_OBROTU = -1
    UF_MOSTEK.CommandButton1.BackColor = &HC0C0FF
    UF_MOSTEK.CommandButton2.BackColor = &H8000000F
  Else
    OBRACAĆ = False
    KIERUNEK_OBROTU = 0
    UF_MOSTEK.CommandButton1.BackColor = &H8000000F
  End If
End Sub
```

```
'przycisk WYRÓWNAJ na panelu MOSTEK KAPITAŃSKI
'wyrównujemy kąt rakiety do kąta osi lotu
```

```
Sub OBRÓT_doosi()
  OBRACAĆ = False
  KOREK = False
  ODCHYLENIE = 0
  KĄT_RAKIETY = KĄT_LOTU
  KĄTY_oblicz
  RYSUJ_STRZAŁKI
  UF_MOSTEK.CommandButton1.BackColor = &H8000000F
  UF_MOSTEK.CommandButton2.BackColor = &H8000000F
End Sub
```

```
Sub OBRÓT_odosi()
  OBRACAĆ = False
  KOREK = False
  ODCHYLENIE = 180
  KĄT_RAKIETY = KĄT_GRANICA(KĄT_LOTU + ODCHYLENIE)
  KĄTY_oblicz
  RYSUJ_STRZAŁKI
  UF_MOSTEK.CommandButton1.BackColor = &H8000000F
  UF_MOSTEK.CommandButton2.BackColor = &H8000000F
End Sub
```

```
'przycisk OBRÓT na panelu MOSTEK KAPITAŃSKI
'działanie identyczne jak obrót w lewo
```

```
Sub OBRÓT_prawy()
  If OBRACAĆ = False Then
    OBRACAĆ = True
    KOREK = False
    KIERUNEK_OBROTU = 1
    UF_MOSTEK.CommandButton1.BackColor = &H8000000F
    UF_MOSTEK.CommandButton2.BackColor = &HC0C0FF
  Else
    OBRACAĆ = False
    UF_MOSTEK.CommandButton2.BackColor = &H8000000F
  End If
End Sub
```

End Sub

```
'przeliczanie kąta na wersory zgodne z układem współrzędnych
'pewnie można by to jakąś mądrzejszą funkcją zrobić
Function WERSOR_X(kąt As Variant) As Variant
    If (kąt >= 0) And (kąt <= 90) Then v = 1
    If (kąt > 90) And (kąt <= 180) Then v = -1
    If (kąt > 180) And (kąt <= 270) Then v = -1
    If (kąt > 270) And (kąt <= 360) Then v = 1
    WERSOR_X = v
End Function
```

```
'przeliczanie kąta na wersory zgodne z układem współrzędnych
Function WERSOR_Y(kąt As Variant) As Variant
    If (kąt >= 0) And (kąt <= 90) Then v = -1
    If (kąt > 90) And (kąt <= 180) Then v = -1
    If (kąt > 180) And (kąt <= 270) Then v = 1
    If (kąt > 270) And (kąt <= 360) Then v = 1
    WERSOR_Y = v
End Function
```

```
'podajemy składowe prędkości i wyliczamy kąt
'kąt zerowy - poziomo w kierunku osi oX
'kąt na + - zgodnie z ruchem wskazówki zegara
'bo tak obracają się obiekty na arkuszu
'niezgodne to co prawda z matematyczny opisem w trygonometrii, ale...
Function KĄT_oblicz(vx, vy As Variant) As Variant
    If vx = 0 Then
        If vy < 0 Then k = 90 Else k = 270
    Else
        k = Atn(vy / vx) * 180 / PI
        If vx >= 0 And vy >= 0 Then k = -k + 360
        If vx < 0 And vy >= 0 Then k = -k + 180
        If vx < 0 And vy < 0 Then k = -k + 180
        If vx >= 0 And vy < 0 Then k = -k
    End If
    KĄT_oblicz = k
End Function
```

```
'przeskok przez granicę kątową - żeby kąt nie był ujemny ani większy od 360
Function KĄT_GRANICA(kąt As Variant) As Variant
    If kąt >= 360 Then kąt = kąt - 360
    If kąt < 0 Then kąt = 360 + kąt
    KĄT_GRANICA = kąt
End Function
```

```
'obliczanie kątów, położenia i kierunku rakiety
'gdy pracuje silnik
'i próby automatycznego wychodzenia z korkociągu,
'gdy prędkość spada do zera a przeciwny zwrot
Sub KĄTY_oblicz()
    Dim vx, vy, vx1, vy1 As Variant

    'położenie rakiety
    vx = Sheets("GRAWITACJA").Range("E10")
    vy = Sheets("GRAWITACJA").Range("F10")
    'położenie planety środkowej
    vx1 = Sheets("GRAWITACJA").Cells(10 + WYKRES_ŚRODEK_NR, 5)
    vy1 = Sheets("GRAWITACJA").Cells(10 + WYKRES_ŚRODEK_NR, 6)
    'nowe położenie rakiety
    vx = vx - vx1
```

```

vy = vy - vy1
'v = (vx ^ 2 + vy ^ 2) ^ (1 / 2)
'wylczenie kąta lotu
KĄT_LOTU = KĄT_oblicz(vx, vy)
'jeśli niema kompensacji to nie ma problemów z korkociągami
'bo sobie sam oblicza i reguluje
'tu wstawić czy kompensacja obrotów rakiety
If KOMPENSACJA Then
  'prawidłowa reakcja na korkociąg gdy silnik pracuje
  If Not (SILNIK_PRACUJE) Then
    KĄT_RAKIETY = KĄT_GRANICA(KĄT_LOTU + ODCHYLENIE)
  Else
    ODCHYLENIE = KĄT_GRANICA(KĄT_RAKIETY - KĄT_LOTU)
  End If
End If
'obraca się zgodnie z szybkością zmian kąta lotu
If OBRACAĆ = True Then
  KĄT_RAKIETY = KĄT_GRANICA(KĄT_RAKIETY + KĄT_SKOKU_OBROTU *
KIERUNEK_OBROTU)
  ODCHYLENIE = KĄT_GRANICA(KĄT_RAKIETY - KĄT_LOTU)
Else
  End If
End Sub

'ALGORYTMY
*****

'obliczanie odległości pomiędzy obiektami
Function R_oblicz(i, j As Variant) As Double
  r2 = (XY(i, 1) - XY(j, 1)) ^ 2 + (XY(i, 2) - XY(j, 2)) ^ 2
  R_oblicz = r2 ^ (1 / 2)
End Function
'obliczanie przyspieszeń z wzoru na grawitację
'a=-G*M/R^2
Sub A_oblicz()
On Error GoTo błąd
Dim R, R3 As Double

For i = 0 To WYM
  AA(i, 1) = 0
  AA(i, 2) = 0
  For j = 0 To WYM
    'sam ze sobą nie oddziałuje i sprawdzamy tylko obiekty z masą >0
    If i <> j And MA(i) > 0 And MA(j) > 0 Then
      'odległości pomiędzy planetami
      R = R_oblicz(i, j)
      R3 = R ^ 3
      'suma wszystkich składowych przyspieszeń daje przyspieszenie
wypadkowe
      For k = 1 To 2
        'sumujemy przyspieszenie chwilowe
        AA(i, k) = AA(i, k) + -GRAW * MA(j) * (XY(i, k) - XY(j, k)) / R3
      Next k
    End If
  Next j
Next i
Exit Sub
błąd:
'gdy błąd, bo np promień równy 0 to omijamy i realizujemy dalej
Resume Next

```

```

End Sub

'RUNGE-KUTTY 4
'zapamiętaj początkowe
Sub XV0_oblicz()
  For i = 0 To WYM
    For k = 1 To 2
      XY0(i, k) = XY(i, k)
      VV0(i, k) = VV(i, k)
    Next k
  Next i
End Sub

'nowe prędkości z początkowym V i obliczonym K
Sub KXV1_oblicz()
  For i = 0 To WYM
    For k = 1 To 2
      VV1(i, k) = AA(i, k) * SKOK_CZASU
      XY(i, k) = XY0(i, k) + VV1(i, k) * SKOK_CZASU / 2
    Next k
  Next i
End Sub

Sub KXV2_oblicz()
  For i = 0 To WYM
    For k = 1 To 2
      VV2(i, k) = AA(i, k) * SKOK_CZASU
      XY(i, k) = XY0(i, k) + VV2(i, k) * SKOK_CZASU / 2
    Next k
  Next i
End Sub

Sub KXV3_oblicz()
  For i = 0 To WYM
    For k = 1 To 2
      VV3(i, k) = AA(i, k) * SKOK_CZASU
      XY(i, k) = XY0(i, k) + VV3(i, k) * SKOK_CZASU
    Next k
  Next i
End Sub

Sub K4_oblicz()
  For i = 0 To WYM
    For k = 1 To 2
      VV4(i, k) = AA(i, k) * SKOK_CZASU
    Next k
  Next i
End Sub

'prędkość w metodzie RK
Sub VXRK_oblicz()
  For i = 0 To WYM
    For k = 1 To 2
      VV(i, k) = VV0(i, k) +
        (VV1(i, k) + 2 * VV2(i, k) + 2 * VV3(i, k) + VV4(i, k)) / 6
      XY(i, k) = XY0(i, k) + VV(i, k) * SKOK_CZASU
    Next k
  Next i
End Sub

'METODA Runge-Kutty4
Sub RK4()
  'zapamiętaj poprzednie V i X
  XV0_oblicz
  '1 krok RK
  A_oblicz
  If (SILNIK_PRACUJE) And (SKOK_CZASU > 0) Then AA_rakieta

```



```

KXV1_oblicz
'2 krok RK
A_oblicz
If (SILNIK_PRACUJE) And (SKOK_CZASU > 0) Then AA_rakieta
KXV2_oblicz
'3 krok RK
A_oblicz
If (SILNIK_PRACUJE) And (SKOK_CZASU > 0) Then AA_rakieta
KXV3_oblicz
'4 krok RK
A_oblicz
'uważamy silnik
If (SILNIK_PRACUJE) And (SKOK_CZASU > 0) Then AA_rakieta
K4_oblicz 'K4
'a teraz obliczenie V i X jako 1/6
VXRK_oblicz 'do tablic VV ze startowego V i 1/6(K1+2K2+2K3+k4)
'do tablic XY z przyrostem czasu dt
End Sub

'METODA VERLET'a
'obliczanie położenia metodą Verleta
'X=X+V*dt+a*dt^2/2
'V=V+a*dt/2
Sub XV_oblicz()
For i = 0 To WYM
For k = 1 To 2
XY(i, k) = XY(i, k) + VV(i, k) * SKOK_CZASU +
AA(i, k) * SKOK_CZASU * SKOK_CZASU / 2
VV(i, k) = VV(i, k) + AA(i, k) * SKOK_CZASU / 2
Next k
Next i
End Sub
'obliczanie prędkości metodą Verleta
'V=V+a*dt/2
Sub V_oblicz()
For i = 0 To WYM
For k = 1 To 2
VV(i, k) = VV(i, k) + AA(i, k) * SKOK_CZASU / 2
Next k
Next i
End Sub
Sub Verlet()
'obliczanie przyspieszeń
A_oblicz
'nowe położenia i prędkości
If (SILNIK_PRACUJE) And (SKOK_CZASU > 0) Then AA_rakieta
XV_oblicz
'obliczanie przyspieszeń
A_oblicz
'wyliczone nowe przyspieszenia związane z ciągiem rakiety
If (SILNIK_PRACUJE) And (SKOK_CZASU > 0) Then AA_rakieta
'nowe prędkości - drugi raz, bo tak działa Verlet
'ale już po obliczeniu przesunięć związanych z rakieta
V_oblicz
End Sub

'METODA EULERA
'obliczenia dla Eulera - V i X bo przyspieszenie tak samo
'obliczanie prędkości
'V=V+a*dt

```

```

'X=X+V*dt
Sub VXE_oblicz()
  For i = 0 To WYM
    For k = 1 To 2
      VV(i, k) = VV(i, k) + AA(i, k) * SKOK_CZASU
      XY(i, k) = XY(i, k) + VV(i, k) * SKOK_CZASU
    Next k
  Next i
End Sub
Sub Euler()
  'obliczanie przyspieszeń
  A_oblicz
  'wyliczone nowe przyspieszenia związane z ciągiem rakiety
  If (SILNIK_PRACUJE) And (SKOK_CZASU > 0) Then AA_rakieta
  'prędkości i położenia
  VXE_oblicz
End Sub

'EULER ULEPSZONY
'nowe prędkości z początkowym V i obliczonym K
Sub K1_oblicz()
  For i = 0 To WYM
    For k = 1 To 2
      VV1(i, k) = AA(i, k) * SKOK_CZASU
      XY(i, k) = XY0(i, k) + VV1(i, k) * SKOK_CZASU
    Next k
  Next i
End Sub
Sub K2_oblicz()
  For i = 0 To WYM
    For k = 1 To 2
      VV2(i, k) = AA(i, k) * SKOK_CZASU
      VV(i, k) = VV0(i, k) + (VV1(i, k) + VV2(i, k)) / 2
      XY(i, k) = XY0(i, k) + VV(i, k) * SKOK_CZASU
    Next k
  Next i
End Sub
Sub Euler2()
  XV0_oblicz
  A_oblicz
  If (SILNIK_PRACUJE) And (SKOK_CZASU > 0) Then AA_rakieta
  K1_oblicz
  A_oblicz
  If (SILNIK_PRACUJE) And (SKOK_CZASU > 0) Then AA_rakieta
  K2_oblicz
End Sub

'daje TRUE jeśli minął skok czasu
'możemy w czasie rzeczywistym - sekundy lub coraz szybciej
'aż do wyczerpania możliwości kompa
Function PUŚĆ_CZAS(skok As Double) As Boolean
On Error GoTo błąd
  'czas w pełnych sekundach ale tylko gdy chcemy
If CZAS_REAL Then
  cz = Timer
  PUŚĆ_CZAS = False
  If cz >= CZAS_ZEGARA + skok Then
    PUŚĆ_CZAS = True
    CZAS_ZEGARA = cz
  End If

```

```

Else
  PUŚĆ_CZAS = True
End If
błąd:
End Function

' GŁÓWNA PĘTLA SYMULACJI
Sub ANIMACJA()
  Do
    'gdy nie krokowo ale automatycznie
    'gdy muszczamy kolejną pętlę bo czas upłynął - gdy badamy czas
    If PUŚĆ_KROK >= 0 And PUŚĆ_CZAS(SKOK_CZASU) Then
      'upływ czasu rzeczywisty (zwłaszcza 1s) ale animacja skokowa
      'albo upływa czasu zależny od procesora, tak lepiej bo płynna animacja
      'żeby można było zrobić następny krok
      If PUŚĆ_KROK > 0 Then PUŚĆ_KROK = 1
      CZAS = CZAS + SKOK_CZASU
      If SKOK_CZASU <> 0 Then
        Select Case ALGORYTM
          Case 1
            Euler
          Case 0
            Verlet
          Case 2
            RK4
          Case 3
            Euler2
        End Select
      End If
      'wyniki na ekran
      WYNIKI
    End If
    'skierowanie obsługi na zewnątrz, aby użytkownik mógł dokonywać zmian
    gdy działa pętla
    DoEvents
  Loop Until KONIEC
End Sub

```

```
'WYNIKI *****
```

```
'wszystkie obliczenia, które wykonywane są w trakcie symulacji
'pokazywane są na ekranie, głównie w panelu KOMPUTER POKŁADOWY
'oraz w tabeli CHWILOWE - a stąd rysowane są obiekty na wykresie
Sub WYNIKI()

```

```

On Error GoTo błąd
  Dim xś, yś, xs As Double
  'można by wyłączyć odświeżanie ekranów przed wszystkimi wyliczeniami
  'a na końcu włączyć - może by wzrosła szybkość
  'ale szybkość nie wzrasta a wszystko mruga
  Application.ScreenUpdating = False
  'BLOKADA_OFF
  'uwzględnić środek wykresu - wybrany obiekt ustawiamy na środek
  'a pozostałe przeliczamy względem tego środka
  xś = XY(WYKRES_ŚRODEK_NR, 1)
  yś = XY(WYKRES_ŚRODEK_NR, 2)
  xs = SKALA_skala * 10 ^ SKALA_wykład

```

```

For i = 0 To WYM
  If MA(i) > 0 Then 'tylko dla obiektów z masą
    w = 10

```

```

'od razu uwzględniamy przesunięcie środka wykresu dla wybranego
obiektu
    Sheets("GRAWITACJA").Cells(w + i, 3) = XY(i, 1) - xś
    Sheets("GRAWITACJA").Cells(w + i, 4) = XY(i, 2) - yś
    Sheets("GRAWITACJA").Cells(w + i, 5) = VV(i, 1)
    Sheets("GRAWITACJA").Cells(w + i, 6) = VV(i, 2)
    Sheets("GRAWITACJA").Cells(w + i, 7) = AA(i, 1)
    Sheets("GRAWITACJA").Cells(w + i, 8) = AA(i, 2)
End If
Next i

'przy pierwszym obrocie pętli głównej symulacji
'musi narysować wykres bo głupiało
If CZAS = SKOK_CZASU Then WYKRES_środek
UF_MOSTEK.CZAS_NA_PANEL
KĄTY_oblicz
RYSUJ_STRZAŁKI
PALIWO_oblicz
RAKIETA_oblicz
ENERGIA_oblicz
ZDERZENIE_oblicz
Sheets("GRAWITACJA").Range("B16") = Sheets("GRAWITACJA").Range("B16") + 1
If WŁĄCZ_ZAPISZ Then ZAPISZ_DoWyniki
'Application.ScreenUpdating = True
Exit Sub
błąd:
Resume Next
End Sub

'początkowo wszystko kręciło się "wokół słońca" - planety ustawionej w
punkcie (0,0)
'za pomocą parametru WYKRES_ŚRODEK_NR uwzględniamy położenie planety
'która jest aktualnie na środku wykresu
'wyboru aktualnej planety dokonujemy przyciskami na panelu OBIEKTY

'wyliczenia dotyczące parametrów rakiety na panelu KOMPUTER POKŁADOWY
Sub RAKIETA_oblicz()
On Error GoTo błąd
'położenie rakiety
x = Sheets("GRAWITACJA").Range("C10")
Y = Sheets("GRAWITACJA").Range("D10")
s = Sqr((x - RAKIETA_Xpop) ^ 2 + (Y - RAKIETA_Ypop) ^ 2)
RAKIETA_S = RAKIETA_S + s
RAKIETA_Xpop = x
RAKIETA_Ypop = Y
'położenie planety i promień
x1 = Sheets("GRAWITACJA").Cells(10 + WYKRES_ŚRODEK_NR, 3)
y1 = Sheets("GRAWITACJA").Cells(10 + WYKRES_ŚRODEK_NR, 4)
r1 = Sheets("DANE").Cells(10, 2 + WYKRES_ŚRODEK_NR)
'nowe położenie rakiety względem nie słońca, ale wybranej planety
x = x - x1
Y = Y - y1
'odległość rakieta-planeta
R = (x ^ 2 + Y ^ 2) ^ (1 / 2)
R = R - r1
'w zależności od wielkości liczb ustalamy odpowiedni format
'wartości na panel KOMPUTER POKŁADOWY
If Abs(x) > 1000000 Or Abs(x) < 0.000001 Then f = "0.0###E+0" Else f =
"0.0###"
UF_KOMPUTER.Label14.Caption = Format(x, f)

```

```

If Abs(Y) > 1000000 Or Abs(Y) < 0.000001 Then f = "0.0####E+0" Else f =
"0.0###"
UF_KOMPUTER.Label15.Caption = Format(Y, f)
If Abs(R) > 1000000 Or Abs(R) < 0.000001 Then f = "0.0####E+0" Else f =
"0.0###"
UF_KOMPUTER.Label16.Caption = Format(R, f)
If Abs(RAKIETA_S) > 1000000 Or Abs(RAKIETA_S) < 0.000001 Then f =
"0.0####E+0" Else f = "0.0###"
UF_KOMPUTER.Label30.Caption = Format(RAKIETA_S, f)

'prędkość też przeliczamy względem obiektu środkowego
vx = Sheets("GRAWITACJA").Range("E10")
vy = Sheets("GRAWITACJA").Range("F10")
vx1 = Sheets("GRAWITACJA").Cells(10 + WYKRES_ŚRODEK_NR, 5)
vy1 = Sheets("GRAWITACJA").Cells(10 + WYKRES_ŚRODEK_NR, 6)
vx = vx - vx1
vy = vy - vy1
v = (vx ^ 2 + vy ^ 2) ^ (1 / 2)
If Abs(v) > 1000000 Or Abs(v) < 0.000001 Then f = "0.0####E+0" Else f =
"0.0###"
UF_KOMPUTER.Label17.Caption = Format(v, f)
'przyspieszenie przeliczamy względem obiektu środkowego
ax = Sheets("GRAWITACJA").Range("G10")
ay = Sheets("GRAWITACJA").Range("H10")
ax1 = Sheets("GRAWITACJA").Cells(10 + WYKRES_ŚRODEK_NR, 7)
ay1 = Sheets("GRAWITACJA").Cells(10 + WYKRES_ŚRODEK_NR, 8)
ax = ax - ax1
ay = ay - ay1
a = (ax ^ 2 + ay ^ 2) ^ (1 / 2)

'zawsze w metrach na sekundę
'a może jeszcze w [g]
gr = Sheets("DANE").Range("C33") 'jaka stała grawitacji
Select Case gr
Case 0
a = a * 1000 'km/s2
Case 3
a = a * 149597870.691 * 1000
Case 4
'a tu nie wiadomo
End Select
If Abs(a) > 1000000 Or Abs(a) < 0.000001 Then f = "0.0####E+0" Else f =
"0.0###"
UF_KOMPUTER.Label18.Caption = Format(a, f)
UF_KOMPUTER.Label25.Caption = Format(ODCHYLENIE, "0.00")
UF_KOMPUTER.Label27.Caption = CStr(Sheets("DANE").Range("B9") -
MASA_RAKIETY)

'reakcja na przeciążenie
kr = Sheets("DANE").Range("B41") 'przeciążenie
'jak przeciążenie to zmniejszaj ciąg
If SILNIK_PRACUJE And SIŁA_CIAGU > 0 And a >= kr * 9.81 Then
'wylicz teoretycznie a potem stopniowo zmniejszaj
Fx = MA(0) * kr * 9.81
SIŁA_CIAGU = SIŁA_CIAGU - (SIŁA_CIAGU - Fx / 1000) / 2
'UF_MOSTEK.ScrollBar4.Value = SIŁA_CIAGU
UF_MOSTEK.SIŁA_CIAGU_ustaw (SIŁA_CIAGU)
UF_MOSTEK.Label7.Caption = CStr(SIŁA_CIAGU)
Else
'UF_MOSTEK.SIŁA_CIAGU_ustaw (CLng(SIŁA_CIAGU))
'UF_MOSTEK.Label7.Caption = CStr(SIŁA_CIAGU)

```

```

End If

Exit Sub
błąd:
Resume Next
End Sub

'strzałki rakiety i lotu rysujemu na radarze - KOMPUTER POKŁADOWY
Sub RYSUJ_STRZAŁKI()
    BLOKADA_OFF
    ActiveSheet.Shapes("R_KIERUNEK").Rotation = KĄT_LOTU
    ActiveSheet.Shapes("R_SILNIK").Rotation = KĄT_RAKIETY
    UF_KOMPUTER.Label19.Caption = Format(KĄT_LOTU, "0.00")
    UF_KOMPUTER.Label20.Caption = Format(KĄT_RAKIETY, "0.00")
End Sub

'wyliczamy energię całkowitą układu
Function ENERGIA_wylicz() As Double
    Dim R, EcSUMA, EpSUMA, EkSUMA As Double
    Dim ep, ek, v, vx, vy As Double

    EcSUMA = 0
    EpSUMA = 0
    EkSUMA = 0
    'Ep=-G*m1*m2/R
    For i = 0 To WYM - 1
        For j = i + 1 To WYM
            ep = 0
            If i <> j And MA(i) > 0 And MA(j) > 0 Then
                R = R_oblicz(i, j)
                ep = -(MA(i) * MA(j) * GRAW / R)
                EpSUMA = EpSUMA + ep
            End If
            'Cells(18 + i, 3 + j) = ep
        Next j
    Next i
    Sheets("GRAWITACJA").Range("F15") = EpSUMA
    'Ek=m*v^2/2
    For i = 0 To WYM
        ek = 0
        If MA(i) > 0 Then
            vx = VV(i, 1)
            vy = VV(i, 2)
            v = (vx ^ 2 + vy ^ 2) ^ (1 / 2)
            ek = (MA(i) * v ^ 2) / 2
            EkSUMA = EkSUMA + ek
        End If
        'Cells(18 + i, 2) = ek
    Next i
    Sheets("GRAWITACJA").Range("G15") = EkSUMA
    'suma całkowita
    EcSUMA = EpSUMA + EkSUMA
    ENERGIA_wylicz = EcSUMA
End Function

'obliczono energię całkowitą w jednym przebiegu
'trzeba sprawdzić i rozdzielić na max i min
'algotym jednoczesnego wyszukiwania minimum i maksimum
Sub ENERGIA_oblicz()
    Dim ene As Double
    On Error GoTo błąd

```

```

BLOKADA_ON
ene = ENERGIA_wylicz
If ene < EcMIN Then EcMIN = ene
If ene > EcMAX Then EcMAX = ene
If EcMIN + EcMAX <> 0 Then EcPRO = (EcMAX - EcMIN) / ((EcMIN + EcMAX) /
2) * 100
Sheets("GRAWITACJA").Range("C15") = EcMIN
Sheets("GRAWITACJA").Range("D15") = EcMAX
Sheets("GRAWITACJA").Range("E15") = EcPRO
błąd:
BLOKADA_OFF
End Sub

```

```

'obliczanie zderzeń obiektów
Sub ZDERZENIE_oblicz()
'sprawdzamy wszystkie obiekty
For i = 0 To WYM
For j = 0 To WYM
If i <> j And MA(i) > 0 And MA(j) > 0 Then
'wyliczamy odległość środków obiektów
R = R_oblicz(i, j)
'promienie planet
r1 = Sheets("DANE").Cells(10, i + 2)
r2 = Sheets("DANE").Cells(10, j + 2)
r0 = Sheets("DANE").Range("B10") 'promień rakiety

'sprawdzanie zderzeń - odległość mniejsza niż suma promieni
'ale można by uwzględnić start i lądowanie rakiety
If R <= r1 + r2 - r0 Then
'gdy zderzenie to koniec pętli symulacji
'ale gdy prędkość mała to może lądowanie albo start
'i wtedy powrót do poprzedniej pozycji
'ale na początku jak zwykle źle bo 0 w arkuszu
MsgBox "zderzenie"
SKOK_CZASU = 0
KONIEC = True
Exit Sub
End If 'promień
End If 'różne i masy
Next j
Next i
End Sub

```

```

' CZYTAJ ZAPISZ *****
' zapis w arkuszu WYNIKI do kolejnego wiersza wszystko z tablic

```

```

Sub ZAPISZ_DoWyniki()
On Error GoTo błąd
w = CInt(Sheets("WYNIKI").Range("A1"))
'zawsze od wiersza 11
If w = 0 Then w = 10
'gdy koniec arkusza
WMAX = 65000
If w = WMAX Then
WŁĄCZ_ZAPISZ = False
UF_MOSTEK.CommandButton44.BackColor = &H8000000F
MsgBox "Zapisano 65000 wierszy - wyczerpały się możliwości arkusza
kalkulacyjnego"
Exit Sub

```

```

End If
'właściwe zapisywanie
CYKLE = Sheets("DANE").Range("C43")
OKRES = Sheets("DANE").Range("B43")
If (CYKLE > 0 And w - 9 > CYKLE) Or _
    (OKRES > 0 And CZAS > OKRES) Then _
    WŁĄCZ_ZAPISZ = False
    UF_MOSTEK.CommandButton44.BackColor = &H8000000F
    'zatrzymać wszystko
    KLAWISZ_start
    If UF_DANE.CheckBox5.Value = True Then
        ZAPISZ_WYNIKI
    End If
    Exit Sub
End If
w = w + 1
Sheets("WYNIKI").Range("A1") = w
With Sheets("WYNIKI")
    .Cells(w, 1) = w - 10
    'czas
    .Cells(w, 2) = CZAS
    'XY
    For i = 0 To 4
        For k = 1 To 2
            .Cells(w, 2 + i * 2 + k) = XY(i, k)
        Next k
    Next i
    'VV
    For i = 0 To 4
        For k = 1 To 2
            .Cells(w, 12 + i * 2 + k) = VV(i, k)
        Next k
    Next i
    'AA
    For i = 0 To 4
        For k = 1 To 2
            .Cells(w, 22 + i * 2 + k) = AA(i, k)
        Next k
    Next i
    'Ep i Ek
    .Cells(w, 33) = Sheets("GRAWITACJA").Range("F15")
    .Cells(w, 34) = Sheets("GRAWITACJA").Range("G15")
End With
Exit Sub
błąd:
    w = 0
    Resume Next
End Sub

Sub ZAPISZ_wróć()
    Sheets("GRAWITACJA").Select
End Sub

Public Sub ZAPISZ_czyść()
    napis = "Czy na pewno wyczyścić wszystkie dane w arkuszu WYNIKI?"
    wynik = MsgBox(napis, vbYesNo + vbCritical + vbDefaultButton2, "CZYŚĆ")
    If wynik = vbNo Then
        Sheets("GRAWITACJA").Select
        Exit Sub
    End If

```



```

Application.ScreenUpdating = False
Sheets("WYNIKI").Select
Rows("11:11").Select
Range(Selection, Selection.End(xlDown)).Select
Selection.ClearContents
Range("A1").Select
ActiveCell.FormulaR1C1 = "0"
Sheets("GRAWITACJA").Select
Application.ScreenUpdating = True
End Sub

Sub ZAPISZ_STARTOWE()
'z DANE do WYNIKI
'X i V
For k = 2 To 6
For w = 1 To 4
    Sheets("WYNIKI").Cells(w + 2, k) = Sheets("DANE").Cells(w, k)
Next w
Next k
'M i R
For k = 2 To 6
For w = 9 To 10
    Sheets("WYNIKI").Cells(w - 2, k) = Sheets("DANE").Cells(w, k)
Next w
Next k
'nazwy
For w = 23 To 27
    Sheets("WYNIKI").Cells(2, w - 21) = Sheets("DANE").Cells(w, 3)
Next w
End Sub
'zapisujemy wyniki do arkusza WYNIKI
'a potem tworzymy jako osobny arkusz na dysku
Public Sub ZAPISZ_WYNIKI()
On Error GoTo bład
    'ale wcześniej parametry startowe
    ZAPISZ_STARTOWE
    wynik = Application.GetSaveAsFilename("WYNIKI", fileFilter:="(*.xls),
*.xls")
    If wynik <> False Then
        Application.ScreenUpdating = False
        'Sheets.Add
        'n = ActiveSheet.Name
        'np = NAZWA_PLIKU(wynik)
        'Sheets(n).Name = np
        Sheets("WYNIKI").Select
        Sheets("WYNIKI").Copy
        nazwa = wynik
        ActiveWorkbook.SaveAs Filename:=nazwa
        ActiveWindow.Close
        Sheets("GRAWITACJA").Select
        Application.ScreenUpdating = True
    End If
    'od razu czyścimy
    ZAPISZ_czyść
    Exit Sub
bład:
    ActiveWindow.Close
    Sheets("GRAWITACJA").Select
    Application.ScreenUpdating = True
End Sub

```

```

'wczytaj z pliku GRA do arkusza DANE
Public Sub ZAPISZ_DANE()
On Error GoTo bład
nazwa = Sheets("DANE").Range("C21")
wynik = Application.GetSaveAsFilename(nazwa, fileFilter:="GRAWITACJA
(*.gra), *.gra")
If wynik <> False Then
Application.ScreenUpdating = False
'a teraz zapisać aktualne do ADANE z mostka
Sheets("DANE").Range("B34") = WYKRES_ŚRODEK_NR
Sheets("DANE").Range("B35") = SKOK_CZASU_SUWAK + 60
Sheets("DANE").Range("B42") = SKOK_CZASU_PRZ
Sheets("DANE").Range("B19") = SKALA_skala
Sheets("DANE").Range("B20") = SKALA_wykład
'ale z ciągiem nie wiadomo jak rozdzielić
Sheets("DANE").Select
'On Error GoTo bład
Sheets("DANE").Copy
nazwa = wynik
ActiveWorkbook.SaveAs Filename:=nazwa
ActiveWindow.Close
Sheets("GRAWITACJA").Select
pnazwa = NAZWA_PLIKU(nazwa)
Sheets("DANE").Range("C21") = pnazwa
Sheets("GRAWITACJA").Label1.Caption = CStr(pnazwa)
Application.ScreenUpdating = True
End If
Exit Sub
bład:
ActiveWindow.Close savechanges:=False
Sheets("GRAWITACJA").Select
Application.ScreenUpdating = True
End Sub

'zapisz arkusz DANE do pliku GRA
Public Sub WCZYTAJ_DANE()
Application.ScreenUpdating = False
Sheets("DANE").Select
aktywny = ActiveWindow.Caption
Set fdialog = Application.FileDialog(msoFileDialogFilePicker)
With fdialog
.AllowMultiSelect = False
.Title = "Otwórz"
.Filters.Clear
.Filters.Add "GRAWITACJA", "*.gra"
If .Show = True Then
For Each x In .SelectedItems
nazwa = x
'ale trzeba samą nazwę oddzielić
z = 1
Do
plik = Right(nazwa, z)
z = z + 1
Loop Until Left(Right(nazwa, z), 1) = "\"
Next
Else
Sheets("GRAWITACJA").Select
Exit Sub
End If
End With
Workbooks.Open Filename:=nazwa

```

```

Cells.Select
Range("A1").Activate
Selection.Copy
Windows(aktywny).Activate
Range("A1").Select
ActiveSheet.Paste
Range("A1").Select
Windows(plik).Activate
Application.CutCopyMode = False
ActiveWindow.Close
Range("A1").Select
Sheets("GRAWITACJA").Select
'samą nazwę wyciągnąć
pnazwa = NAZWA_PLIKU(nazwa)
Sheets("DANE").Range("C21") = pnazwa
Sheets("GRAWITACJA").Label1.Caption = CStr(pnazwa)
'panel mostek zapamiętać położenie
x = Sheets("DANE").Range("B38")
Y = Sheets("DANE").Range("B39")
UF_DANE.USTAW_WYKRES
KLAWISZ_ustaw
'żeby dopasowało się jednak do ekranu
UF_DANE.DOPASUJ_WYSOKOŚĆ
KLAWISZ_ustaw
'odtwórz położenie mostka
Sheets("DANE").Range("B38") = x
Sheets("DANE").Range("B39") = Y
Application.ScreenUpdating = True
End Sub

```

```

Function NAZWA_PLIKU(n As Variant) As Variant
'obetnij cztery ostatnie - rozszerzenie
'poszykaj pierwszego z prawej \ i obetnij
On Error GoTo błąd
'n = "d:\PULPIT\GRAWITACJA\SŁOŃCE.gra"
k = InStrRev(n, ".")
s = InStrRev(n, "\")
p = Mid(n, s + 1, k - s - 1)
NAZWA_PLIKU = p
Exit Function
błąd:
NAZWA_PLIKU = ""
End Function

```