

Dokumentace výsledku
**Software MN_SGM_KRY pro
tvorbu 3D modelu míry nejistoty
strukturně geologického modelu
(pro lokalitu Kaňk)**
Programátorská dokumentace

Autoři:

Staněk, F. – Jelínek, J. – Grygar, R. –
Kryl, J. – Skřivánková, J.

Označení výsledku ISTA:

SS02030023-V41



**T A
Č R**

Projekt č. „SS02030023 Horninové prostředí a suroviny“
je spolufinancován se státní podporou Technologické
agentury ČR v rámci Programu Prostředí pro život.

www.tacr.cz

Programátorská dokumentace software MN_SGM_KRY (software pro tvorbu Modelu míry Nejistoty Strukturně Geologického Modelu v revírech s hlubinnou těžbou v KRYstaliniku)

Níže jsou v přílohách uvedeny zdrojové kódy tří vybraných modulů software MN_SGM_KRY, jejichž podrobný popis je v uživatelské příručce v kapitolách 2.1.5 (*Dopln_Z_povrchu_a_Top_krystal* a *Typy_hornin*) a 2.7:

Příloha 1: Zdrojový kód makra <i>Dopln_Z_povrchu_a_Top_krystal</i>	3
Příloha 2: Zdrojový kód makra <i>Typy_hornin</i>	4
Příloha 3: Zdrojový kód makra <i>Nejistota_Kank</i>	8

Příloha 1: Zdrojový kód makra Dopln_Z_povrchu_a_Top_krystal.

```
Public Sub Dopln_Z_povrchu_a_Top_krystal()  
'  
' Makro  
' Makro vytvořené 2024, František Staněk  
'  
Dim XMin As Double, XMax As Double, YMin As Double, YMax As Double, Krok_gridu_hor As  
Double  
Dim SurferApp As Object, Grid_DMR As Object, Grid_Top_kryst As Object  
Dim i As Integer  
Dim i_rad As Integer, i_sl As Integer  
Dim Blank As Double  
Dim n_gridx_hor As Integer, n_gridy_hor As Integer  
Dim x As Double, y As Double, Z_DMR As Double, Z_Top_kryst As Double  
  
Set SurferApp = CreateObject("Surfer.Application")  
SurferApp.DefaultFilePath = ThisWorkbook.Path  
  
' Objekty gridů báze a stropu  
Set Grid_DMR = SurferApp.NewGrid  
Grid_DMR.LoadFile ThisWorkbook.Path & "\" & "DMR_n.grd", False  
Blank = Grid_DMR.BlankValue  
XMin = Grid_DMR.XMin  
XMax = Grid_DMR.XMax  
YMin = Grid_DMR.YMin  
YMax = Grid_DMR.YMax  
Krok_gridu_hor = Grid_DMR.xSize  
n_gridx_hor = Grid_DMR.NumCols  
n_gridy_hor = Grid_DMR.NumRows  
Set Grid_Top_kryst = SurferApp.NewGrid  
Grid_Top_kryst.LoadFile ThisWorkbook.Path & "\" & "Top krystalinika_upr_n.grd",  
False  
  
Sheets("AllVrtl0x10m").Select  
i = 2  
Do While Not IsEmpty(Cells(i, 1))  
    x = Cells(i, 2)  
    y = Cells(i, 3)  
    i_rad = Int(y - YMin) / Krok_gridu_hor  
    i_sl = Int(x - XMin) / Krok_gridu_hor  
    Z_DMR = Grid_DMR.GetNode(i_rad + 1, i_sl + 1)  
    Z_Top_kryst = Grid_Top_kryst.GetNode(i_rad + 1, i_sl + 1)  
    Cells(i, 6) = Z_DMR  
    Cells(i, 7) = Z_Top_kryst  
    i = i + 1  
Loop  
End Sub
```

Příloha 2: Zdrojový kód makra Typy_hornin.

```
Public Sub Typy_hornin()  
'  
' Makro - vytvoření 3D gridu geomodelu hodnot typů hornin a tektoniky  
' Makro vytvořené 2024, František Staněk  
'  
Dim Pocet_typu_hornin As Integer, krok_Z As Single, Baze_modelu As Single  
Dim i As Integer, j_vrty As Long, j_lit As Long, k As Integer, j As Integer  
Dim Nazev_vrtu As String, x As Double, y As Double, z As Double, Z_zac As Single,  
Z_Top_krystal As Single  
Dim Z_poloha As Single, pocet_typu_3az5 As Integer  
Dim j_Typy_hornin As Long, pocy_typu_hornin(1 To 10) As Long  
Dim Pocet_bodu_vrt As Integer, Max_pocet_radku As Long, Cislo_sloupce As Integer  
Dim Typ_hor_vrtu(1 To 1000) As Byte, Z_vrtu_Typ_hor(1 To 1000) As Single  
Dim Z_vrt_lit(1 To 20) As Single, Z_vrt_Name(1 To 20) As String, Zrud_a As Byte, Zrud_b  
As Byte, Lampro As Byte, Tekt As Byte  
Dim Z_od As Single, Z_do As Single, i_poc As Integer, i_od As Integer, i_do As Integer  
  
Sheets("Legenda").Select  
Pocet_typu_hornin = Cells(10, 4)  
krok_Z = Cells(11, 4)  
Baze_modelu = Cells(12, 4)  
Max_pocet_radku = Cells(9, 4)  
For i = 1 To Pocet_typu_hornin  
    pocy_typu_hornin(i) = 0  
Next i  
  
j_vrty = 2  
j_lit = 2  
j_Typy_hornin = 2  
Cislo_sloupce = 1  
Sheets("AllVrt10x10m").Select  
Do While Not IsEmpty(Cells(j_vrty, 1))  
    Nazev_vrtu = Cells(j_vrty, 1)  
    x = Cells(j_vrty, 2)  
    y = Cells(j_vrty, 3)  
    z = Cells(j_vrty, 4)  
    Z_Top_krystal = Cells(j_vrty, 7)  
    If krok_Z >= 1 Then  
        Z_zac = Int(z) 'krok po metru nebo víc - začátek je celý  
    Else  
        Z_zac = Int(z * 10) / 10 'krok po méně než metru - začátek je v dese-  
tinách metru  
    End If  
    ' Naplnění litologie vrtu typy 1 a 2  
    i = 0  
    For Z_poloha = Z_zac To Baze_modelu Step -krok_Z  
        i = i + 1  
        Z_vrtu_Typ_hor(i) = Z_poloha  
        If Z_poloha > Z_Top_krystal Then  
            Typ_hor_vrtu(i) = 1 'kvartér  
        Else  
            Typ_hor_vrtu(i) = 2 'krystalinikum  
        End If  
    Next Z_poloha  
    Pocet_bodu_vrt = i  
    ' Naplnění litologie vrtu z tabulky virtualni-vrty typy 3 až 5  
    ' Typ 3 - zrudnění  
    Sheets("virtualni-vrty").Select
```

```

i = 0
Zrud_a = 0
Zrud_b = 0
Lampro = 0
Tekt = 0
Do While Cells(j_lit, 8) = Nazev_vrtu
    i = i + 1
    Z_vrt_lit(i) = Cells(j_lit, 3)
    Z_vrt_Name(i) = Cells(j_lit, 6)
    If InStr(1, Cells(j_lit, 6), "_a") > 0 Then          'začátek zrudnění
        Zrud_a = Zrud_a + 1
    Else
        If InStr(1, Cells(j_lit, 6), "_b") > 0 Then      'konec zrudnění
            Zrud_b = Zrud_b + 1
        Else
            If InStr(1, Cells(j_lit, 6), "lamprofyr") > 0 Then 'konec zrudnění
                Lampro = Lampro + 1
            Else
                Tekt = Tekt + 1
            End If
        End If
    End If
    j_lit = j_lit + 1
Loop
pocet_typu_3az5 = i
i_poc = 1
Dal_a:
Do While Zrud_a > 0
    For i = i_poc To pocet_typu_3az5
        If InStr(1, Z_vrt_Name(i), "_a") > 0 Then
            Z_od = Z_vrt_lit(i)
            Zrud_a = Zrud_a - 1
            If Z_od < Baze_modelu Then Exit For
            For j = i + 1 To pocet_typu_3az5
                If InStr(1, Z_vrt_Name(j), "_b") > 0 Then
                    Z_do = Z_vrt_lit(j)
                    Zrud_b = Zrud_b - 1
                    i_poc = j + 1
                    '
                    Zápis do Typ_hor_vrtu(i) - zrudnělá poloha
                    If krok_Z >= 1 Then
                        Z_od = Round(Z_od)                'krok po metru nebo
víc - začátek polohy je celý metr
                        Z_do = Round(Z_do)                'krok po metru nebo
víc - konec polohy je celý metr
                    Else
                        Z_od = Round(Z_od * 10) / 10      'krok po méně než
metru - začátek polohy je v desetínách metru
                        Z_do = Round(Z_do * 10) / 10      'krok po méně než
metru - konec polohy je v desetínách metru
                    End If
                    i_od = (Z_zac - Z_od) / krok_Z + 1    'index Z_od v
Typ_hor_vrtu(i)
                    i_do = (Z_zac - Z_do) / krok_Z + 1    'index Z_do v
Typ_hor_vrtu(i)
                    For k = i_od To i_do
                        Typ_hor_vrtu(k) = 3                'Zrudnění
                    Next k
                    GoTo Dal_a
                End If
            Next j
            '
            K_a neexistuje _b, tzn. že zrudnění je až po bázi modelu
        End If
    Next i

```

```

'                Zápis do Typ_hor_vrtu(i) - zrudnělá poloha až po bázi modelu
                If krok_Z >= 1 Then
                    Z_od = Round(Z_od)                                'krok po metru nebo
víc - začátek polohy je celý metr
                    Z_do = Round(Baze_modelu)                        'krok po metru nebo
víc - konec polohy je celý metr
                Else
                    Z_od = Round(Z_od * 10) / 10                    'krok po méně než
metru - začátek polohy je v desetinách metru
                    Z_do = Round(Baze_modelu * 10) / 10            'krok po méně než
metru - konec polohy je v desetinách metru
                End If
                i_od = (Z_zac - Z_od) / krok_Z + 1                  'index Z_od v
Typ_hor_vrtu(i)
                i_do = (Z_zac - Z_do) / krok_Z + 1                  'index Z_do v
Typ_hor_vrtu(i)
                For k = i_od To i_do
                    Typ_hor_vrtu(k) = 3                              'Zrudnění
                Next k
            End If
        Next i
    Loop
'    samostatné _b - neexistuje _a (zrudnění od povrchu po Z_do) to může nastat
pouze jednou po první _b
    If Zrud_b > 0 Then
        For j = 1 To pocet_typu_3az5
            If InStr(1, Z_vrt_Name(j), "_b") > 0 Then
                Z_do = Z_vrt_lit(j)
                Zrud_b = Zrud_b - 1
                If krok_Z >= 1 Then
                    Z_do = Round(Z_do)                                'krok po metru nebo
víc - konec polohy je celý metr
                Else
                    Z_do = Round(Z_do * 10) / 10                    'krok po méně než
metru - konec polohy je v desetinách metru
                End If
                i_od = 1                                              'index Z_od v
Typ_hor_vrtu(i) - od povrchu
                i_do = (Z_zac - Z_do) / krok_Z + 1                    'index Z_do v
Typ_hor_vrtu(i)
                For k = i_od To i_do
                    Typ_hor_vrtu(k) = 3                              'Zrudnění
                Next k
            End If
        Next j
    End If
'    Tektonika
    If Tekt > 0 Then
        For j = 1 To pocet_typu_3az5
            If ((InStr(1, Z_vrt_Name(j), "zlom") > 0) And (InStr(1, Z_vrt_Name(j),
"_" ) = 0)) Or ((InStr(1, Z_vrt_Name(j), "žíla") And (InStr(1, Z_vrt_Name(j), "_" ) = 0))
-
            Or ((InStr(1, Z_vrt_Name(j), "zlom") > 0) And (InStr(1, Z_vrt_Name(j),
"_" ) = 0)) Or ((InStr(1, Z_vrt_Name(j), "žíla") And (InStr(1, Z_vrt_Name(j), "_" ) = 0))
-
            Then
                Z_do = Z_vrt_lit(j)
                ' Zápis do Typ_hor_vrtu(i) - tektonické porušení
                If krok_Z >= 1 Then
                    Z_do = Round(Z_do)                                'krok po metru nebo
víc - konec polohy je celý metr
                Else

```

**SOFTWARE PRO TVORBU 3D MODELU MÍRY NEJISTOTY STRUKTURNĚ
GEOLOGICKÉHO MODELU (PRO LOKALITU KAŇK)**

```

        Z_do = Round(Z_do * 10) / 10                                'krok po méně než
metru - konec polohy je v desetínách metru
        End If
        i_do = (Z_zac - Z_do) / krok_Z + 1                        'index Z_do v
Typ_hor_vrtu(i)
        Typ_hor_vrtu(i_do) = 4                                    'tektonické porušení
        End If
    Next j
End If
'
    Lamprofyr
    If Lampro > 0 Then
        For j = 1 To pocet_typ_u_3az5
            If InStr(1, Z_vrt_Name(j), "lamprofyr") > 0 Then
                Z_do = Z_vrt_lit(j)
                ' Zápis do Typ_hor_vrtu(i) - lamprofyr
                If krok_Z >= 1 Then
                    Z_do = Round(Z_do)                                'krok po metru nebo
víc - konec polohy je celý metr
                Else
                    Z_do = Round(Z_do * 10) / 10                        'krok po méně než
metru - konec polohy je v desetínách metru
                End If
                i_do = (Z_zac - Z_do) / krok_Z + 1 'index Z_do v Typ_hor_vrtu(i)
                Typ_hor_vrtu(i_do) = 5                                'lamprofyr
            End If
        Next j
    End If

'
    Zapsání typů hornin virtuálního vrtu do tabulky Typy_hornin
    Sheets("Typy_hornin").Select
    For j = 1 To Pocet_bodu_vrt
        Cells(j_Typy_hornin, Cislo_sloupce) = x
        Cells(j_Typy_hornin, Cislo_sloupce + 1) = y
        Cells(j_Typy_hornin, Cislo_sloupce + 2) = Z_vrtu_Typ_hor(j)
        Cells(j_Typy_hornin, Cislo_sloupce + 3) = Typ_hor_vrtu(j)
        pocety_typ_u_hornin(Typ_hor_vrtu(j)) = pocety_typ_u_hornin(Typ_hor_vrtu(j)) + 1
        Cells(j_Typy_hornin, Cislo_sloupce + 4) = Nazev_vrtu
        j_Typy_hornin = j_Typy_hornin + 1
        If j_Typy_hornin = Max_pocet_radku + 1 Then
            Cislo_sloupce = Cislo_sloupce + 6
            j_Typy_hornin = 2
        End If
    Next j
    Sheets("AllVrt10x10m").Select
    j_vrty = j_vrty + 1
Loop
'
    Tabulka s počty typů hornin do tabulky Legenda
    Sheets("Legenda").Select
    Cells(39, 6) = "Typ horniny"
    Cells(39, 7) = "Počet"
    For i = 1 To Pocet_typ_u_hornin
        Cells(39 + i, 6) = i
        Cells(39 + i, 7) = pocety_typ_u_hornin(i)
    Next i
End Sub

```

Příloha 3: Zdrojový kód makra Nejistota_Kank.

```
Public Sub Nejistota_Kank()
'
' Makro - vytvoření 3D gridu geomodelu hodnot typů hornin a tektoniky
' Makro vytvořené 2025, František Staněk
'
Dim krok_Z As Single, Baze_modelu As Single, R_koule As Single
Dim i As Integer, j As Integer, j_vrty As Integer, k As Integer, Pocet_typu_lit As Integer
Dim Nazev_vrtu As String, x As Double, y As Double, z As Double, Z_cele As Double
Dim j_Typy_hornin As Long, pocty_typu_lit() As Long
Dim X_bod As Double, Y_bod As Double, Z_bod As Double, Typ_horniny As Byte, Celkem_v_R As Integer, Typ_lit As Byte
Dim Entropie As Double, Nejistota As Double, Pravdep As Double, Nejistota_lit As Double
Dim Nejistota_obec_dole As Double, hloubka_pro_nejistotu_obec_dole As Double, Nejistota_obec_povrch As Double
Dim Exist_jina_obecna As Boolean, pom As String, Nej_obec_vrt As Double, Pocet_polyg As Integer
Dim Pocet_bodu_polyg() As Integer, Nejistota_polyg_povrch() As Double, Nejistota_polyg_dole() As Double, Max_bodu_polyg As Integer
Dim Polyg_X() As Double, Polyg_Y() As Double
Dim LOC As Integer, I1 As Integer, I2 As Integer, Uvnitr As Boolean, Nejistota_polyg_povrch_XY As Double, Nejistota_polyg_dole_XY As Double
Dim X_Rez As Double, Y_Rez As Double, V As Double, Pocet_useku_Rezu As Integer
Dim X1_Rezy() As Double
Dim Y1_Rezy() As Double
Dim X2_Rezy() As Double
Dim Y2_Rezy() As Double
Dim Hloubka_Rezy() As Double, Vzdal_rez() As Double, Nejistota_Rez_minVzdal() As Double
Dim smernice As Double, Sniz_rez_proc As Double, exist As Boolean
Dim Nazev_souboru As String, Pocet_vrtu As Integer
Dim Vrty_X() As Double
Dim Vrty_Y() As Double
Dim Vrty_Z() As Double
Dim Vrty_Hloubka() As Double, Vrty_Verohodnost() As Double, Vrty_R() As Double
Dim Sniz_vrt_proc As Double, Min_celk_nejis As Single
Dim X_vrt As Double, Y_vrt As Double, Z_vrt As Double, Hloubka_vrt As Double, Baze_vrt As Double
Dim Sniz_dul_dila_proc As Double
Dim Upr_Nej_max As Double, Nej_max As Double, Max_rezy As Double, Pocet_useku As Integer
Dim Krok_gridu_XY As Double, Posun As Integer, Cislo_sloupce As Integer, Cisla_radku_sl_X() As Long, X_puv As Double
Dim Nej_max_lit_obec As Double, Upr_Nej_max_lit_obec As Double
Dim Nej_obec_zakl As Double, Pocet_bodu_1 As Long
Dim Max_pocet_radku As Long, Cisla_Sloupce_sl_X() As Long, Sloupec_Typy_hornin As Long
Dim Pocet_useku_DD As Integer
Dim X1_DD() As Double
Dim Y1_DD() As Double
Dim Z1_DD() As Double
Dim X2_DD() As Double
Dim Y2_DD() As Double
Dim Z2_DD() As Double
Dim Nejistota_DD_minVzdal() As Double
Dim Vzdal_DD() As Double
Dim typ_DD() As String
Dim MensiZ As Double, VetsiZ As Double, Vzdal As Double, Max_DD As Double, j_vrty_Od As Integer, j_vrty_Do As Integer

Sheets("Legenda").Select
Cells(28, 1) = "Spuštěno: " & Date & ", " & Time
Cells(29, 1) = ""
Pocet_typu_lit = Cells(10, 4)
ReDim pocty_typu_lit(1 To Pocet_typu_lit)
krok_Z = Cells(11, 4)
Baze_modelu = Cells(12, 4)
R_koule = Cells(14, 4)

Nejistota_obec_povrch = Cells(15, 4)
Nejistota_obec_dole = Cells(16, 4)
If Cells(17, 4) = "Ano" Then Exist_jina_obecna = True Else Exist_jina_obecna = False
hloubka_pro_nejistotu_obec_dole = Cells(18, 4)
Krok_gridu_XY = Cells(19, 4)
Posun = R_koule \ Krok_gridu_XY + 1
```



```

Max_rezy = Cells(20, 4)
Min_celk_nejis = Cells(21, 4)
Max_pocet_radku = Cells(9, 4)
Max_DD = Cells(22, 4)
j_vrty_Od = Cells(23, 4)
j_vrty_Do = Cells(24, 4)

'   Načtení pole Císla_radku_sl_X
i = 0
Do While Not IsEmpty(Cells(i + 80, 6))
    i = i + 1
Loop
ReDim Císla_radku_sl_X(1 To i)
ReDim Císla_Sloupce_sl_X(1 To i)
For j = 1 To i
    Císla_radku_sl_X(j) = Cells(79 + j, 7)
    Císla_Sloupce_sl_X(j) = Cells(79 + j, 9)
Next j

'   Existuje jiná obecná nejistota?
If Exist_jina_obecna Then
    Open ThisWorkbook.Path & "\Polyg_obec.dat" For Input As #1
    Line Input #1, pom
    Pocet_polyg = Left(pom, InStr(pom, " ") - 1)
    ReDim Pocet_bodu_polyg(1 To Pocet_polyg)
    ReDim Nejistota_polyg_povrch(1 To Pocet_polyg)
    ReDim Nejistota_polyg_dole(1 To Pocet_polyg)
    Max_bodu_polyg = 0
    For i = 1 To Pocet_polyg
        Input #1, Pocet_bodu_polyg(i), Nejistota_polyg_povrch(i), Nejistota_po-
lyg_dole(i)
        'Zjištění max. počtu bodů polygonu
        If Pocet_bodu_polyg(i) > Max_bodu_polyg Then Max_bodu_polyg = Pocet_bodu_po-
lyg(i)
        For j = 1 To Pocet_bodu_polyg(i)
            Line Input #1, pom
        Next j
    Next i
    Close #1
    ReDim Polyg_X(1 To Pocet_polyg, 1 To Max_bodu_polyg)
    ReDim Polyg_Y(1 To Pocet_polyg, 1 To Max_bodu_polyg)
    '   Načtení souřadnic polygonů
    Open ThisWorkbook.Path & "\Polyg_obec.dat" For Input As #1
    Line Input #1, pom
    For i = 1 To Pocet_polyg
        Line Input #1, pom
        For j = 1 To Pocet_bodu_polyg(i)
            Input #1, Polyg_X(i, j), Polyg_Y(i, j)
        Next j
    Next i
    Close #1
End If

'   Vrty
Nazev_souboru = ""
Application.DisplayAlerts = False
With Application.FileDialog(msoFileDialogFilePicker)
    .InitialFileName = ThisWorkbook.Path
    .Title = "Vyber soubor s údaji o vrtech"
    'zrušení minulého filtru
    .Filters.Clear
    .Filters.Add "Soubory Excelu (xls/xlsx)", "*.xl*", 1
    .Show
    If .SelectedItems.Count = 0 Then
        MsgBox "Nebyly načteny žádné soubory": Exit Sub
    ElseIf .SelectedItems.Count > 1 Then
        MsgBox "Vyberte pouze jeden soubor!": Exit Sub
    Else
        Nazev_souboru = .SelectedItems(1)
    End If
End With
Workbooks.Open (Nazev_souboru)

```

```
' Naplnění polí s potřebnými údaji o vrtech
Pocet_vrtu = Cells(Rows.Count, "B").End(xlUp).Row - 1
ReDim Vrtty_X(1 To Pocet_vrtu)
ReDim Vrtty_Y(1 To Pocet_vrtu)
ReDim Vrtty_Z(1 To Pocet_vrtu)
ReDim Vrtty_Hloubka(1 To Pocet_vrtu)
ReDim Vrtty_Verohodnost(1 To Pocet_vrtu)
ReDim Vrtty_R(1 To Pocet_vrtu)
i = 2
Do While Not IsEmpty(Cells(i, 2))
    Vrtty_X(i - 1) = Cells(i, 2)
    Vrtty_Y(i - 1) = Cells(i, 3)
    Vrtty_Z(i - 1) = Cells(i, 4)
    Vrtty_Hloubka(i - 1) = Cells(i, 6)
    Vrtty_Verohodnost(i - 1) = Cells(i, 7)
    Vrtty_R(i - 1) = Cells(i, 8)
    i = i + 1
Loop
ActiveWorkbook.Close False
Application.DisplayAlerts = True 'opětovné povolení excelovských hlášené

' Řezy
Open ThisWorkbook.Path & "\Rezy.dat" For Input As #1
i = 0
Do While Not EOF(1)
    i = i + 1
    Line Input #1, pom
Loop
Pocet_useku_Rezu = i
Close #1
ReDim X1_Rezy(1 To Pocet_useku_Rezu)
ReDim Y1_Rezy(1 To Pocet_useku_Rezu)
ReDim X2_Rezy(1 To Pocet_useku_Rezu)
ReDim Y2_Rezy(1 To Pocet_useku_Rezu)
ReDim Hloubka_Rezy(1 To Pocet_useku_Rezu) 'Dosah po Z m n.m.
ReDim Vzdal_rez(1 To Pocet_useku_Rezu)
ReDim Nejistota_Rez_minVzdal(1 To Pocet_useku_Rezu)
Open ThisWorkbook.Path & "\Rezy.dat" For Input As #1
i = 0
Do While Not EOF(1)
    i = i + 1
    Input #1, X1_Rezy(i), Y1_Rezy(i), X2_Rezy(i), Y2_Rezy(i), Hloubka_Rezy(i),
    Vzdal_rez(i), Nejistota_Rez_minVzdal(i)
Loop
Close #1

' Důlní díla
Open ThisWorkbook.Path & "\Dulni_dila.dat" For Input As #1
i = 0
Do While Not EOF(1)
    i = i + 1
    Line Input #1, pom
Loop
Pocet_useku_DD = i
Close #1
ReDim X1_DD(1 To Pocet_useku_DD)
ReDim Y1_DD(1 To Pocet_useku_DD)
ReDim Z1_DD(1 To Pocet_useku_DD)
ReDim X2_DD(1 To Pocet_useku_DD)
ReDim Y2_DD(1 To Pocet_useku_DD)
ReDim Z2_DD(1 To Pocet_useku_DD)
ReDim Nejistota_DD_minVzdal(1 To Pocet_useku_DD)
ReDim Vzdal_DD(1 To Pocet_useku_DD)
ReDim typ_DD(1 To Pocet_useku_DD)
Open ThisWorkbook.Path & "\Dulni_dila.dat" For Input As #1
i = 0
Do While Not EOF(1)
    i = i + 1
    Input #1, X1_DD(i), Y1_DD(i), Z1_DD(i), X2_DD(i), Y2_DD(i), Z2_DD(i), Nejis-
    tota_DD_minVzdal(i), Vzdal_DD(i), typ_DD(i), pom
Loop
Close #1
```

**SOFTWARE PRO TVORBU 3D MODELU MÍRY NEJISTOTY STRUKTURNĚ
GEOLOGICKÉHO MODELU (PRO LOKALITU KAŇK)**

```

'      Zapis geometrie 3D gridu
      Open ThisWorkbook.Path & "\Nejistota_3D.dat " For Output As 1      '3D model nejistoty
      Print #1, "Nazev X Y Z Nej_lit Nej_obec Nej_lit_obec_max Sniz_vrt_proc Sniz_rez_proc
      Sniz_dul_dila_proc Upr_Nej_lit_obec_max"
      Pocet_bodu_1 = 0
      Sheets("AllVrt10x10m").Select
      j_vrty = 2
      Cislo_sloupce = 1
      X_puv = Cells(2, 2)
      k = 0
      Do While Not IsEmpty(Cells(j_vrty, 1))
        If j_vrty < j_vrty_Od Then GoTo Dalsi_vrt      'od zadaného čísla řádku vrtu
        Nazev_vrtu = Cells(j_vrty, 1)
        x = Cells(j_vrty, 2)
        If x <> X_puv Then
          Cislo_sloupce = Cislo_sloupce + 1
          X_puv = x
        End If
        y = Cells(j_vrty, 3)
        z = Cells(j_vrty, 4)

        If krok_Z >= 1 Then
          Z_cele = Int(z)      'krok po metru nebo víc - začátek je celý
          metr
        Else
          Z_cele = Int(z * 10) / 10      'krok po méně než metru - začátek je v
          desetínách metru
        End If

'      Zjištění, zda je X,Y v jiném polygonu obecné nejistoty
      Uvnitr = False
      If Exist_jina_obecna Then
        For i = 1 To Pocet_polyg
          '      zjištění, zda je bod gridu v polygonu
          PTLOC_2_rozm_pole LOC, I1, I2, Pocet_bodu_polyg(i), i, Polyg_X, Polyg_Y,
          x, y
          If LOC <> -1 Then      'Uvnitr
            Nejistota_polyg_povrch_XY = Nejistota_polyg_povrch(i)
            Nejistota_polyg_dole_XY = Nejistota_polyg_dole(i)
            Uvnitr = True
            GoTo dal_hor      'polygony jsou disjunktní
          End If
        Next i
      End If

      dal_hor:
      Do While Z_cele >= Baze_modelu
        Celkem_v_R = 0
        For i = 1 To Pocet_typy_lit
          pocy_typy_lit(i) = 0
        Next i

        If Cislo_sloupce > Posun Then      'pro urychlení se začne prohledávat až od
        sloupce, který teoreticky spadá do koule výběru
          j_Typy_hornin = Ciska_radku_sl_X(Cislo_sloupce - Posun)
          Sloupec_Typy_hornin = Ciska_Sloupce_sl_X(Cislo_sloupce - Posun)
        Else
          j_Typy_hornin = 2
          Sloupec_Typy_hornin = 1
        End If
        Sheets("Typy_hornin").Select

        Do While Not IsEmpty(Cells(j_Typy_hornin, Sloupec_Typy_hornin))
          X_bod = Cells(j_Typy_hornin, Sloupec_Typy_hornin)
          If X_bod > x + R_koule Then GoTo dalsi      'vzhledem k utřídění tabulek
          dle názvu ( a tedy dle X sestupně a Y sestupně) je možno na začátku testovat X_bod >
          X+R_koule - v tomto případě jít na další bod bez výpočtu vzdálenosti k X_bod, Y_bod, Z_bod
          If X_bod < x - R_koule Then GoTo ven      'vzhledem k utřídění tabulek
          dle názvu ( a tedy dle X sestupně a Y sestupně) je možno ukončit čtení z tabulky Typy_hornin
          jakmile X_bod < X-R_koule
          Y_bod = Cells(j_Typy_hornin, Sloupec_Typy_hornin + 1)

```

```

        If Y_bod < y - R_koule Then GoTo dalsi      'vzhledem k utřídění tabulek
dle názvu ( a tedy dle X sestupně a Y sestupně) je možno na začátku testovat Y_bod < Y-
R_koule - v tomto případě jít na další bod bez výpočtu vzdálenosti k X_bod, Y_bod, Z_bod
        If Y_bod > y + R_koule Then GoTo dalsi      'vzhledem k utřídění tabulek
dle názvu ( a tedy dle X sestupně a Y sestupně) je možno na začátku testovat Y_bod >
Y+R_koule - v tomto případě jít na další bod bez výpočtu vzdálenosti k X_bod, Y_bod, Z_bod
        Z_bod = Cells(j_Typý_hornin, Sloupec_Typý_hornin + 2)
        If Z_bod < Baze_modelu - R_koule Then GoTo dalsi      'max do dosahu
vyhledávací koule z báze modelu
    '
        Je bod v kouli o poloměru R_koule?
        If Vzdal_3D(x, y, Z_cele, X_bod, Y_bod, Z_bod) <= R_koule Then
            Typ_lit = Cells(j_Typý_hornin, Sloupec_Typý_hornin + 3)
            Celkem_v_R = Celkem_v_R + 1
            pocy_typý_lit(Typ_lit) = pocy_typý_lit(Typ_lit) + 1
        End If
dalsi:
        If j_Typý_hornin = Max_pocet_radku Then
            Sloupec_Typý_hornin = Sloupec_Typý_hornin + 7
            j_Typý_hornin = 1
        End If
        j_Typý_hornin = j_Typý_hornin + 1
    Loop

ven:
    '
        Výpočet entropie a nejistoty
        Entropie = 0
    '
        Nejistota litologie
        Entropie = 0
        If Celkem_v_R > 0 Then
            For i = 1 To Pocet_typý_lit
                Pravdep = pocy_typý_lit(i) / Celkem_v_R
                If Pravdep > 0 Then Entropie = Entropie - Pravdep * Log(Pravdep)
            'pravdepodobnost
            Next i
        End If
        Nejistota_lit = Entropie / Log(Pocet_typý_lit)

    '
        Zpracování obecné nejistoty
    '
        Nejprve základní
        Nej_obec_zakl = Nejistota_obec_povrch + (Nejistota_obec_dole - Nejis-
tota_obec_povrch) * (z - Z_cele) / hloubka_pro_nejistotu_obec_dole
        If Exist_jina_obecna Then
            If Uvnitr Then
                Nej_obec_vrt = Nejistota_polyg_povrch_XY + (Nejistota_polyg_dole_XY -
Nejistota_polyg_povrch_XY) * (z - Z_cele) / hloubka_pro_nejistotu_obec_dole
            Else
                Nej_obec_vrt = Nej_obec_zakl
            End If
        Else
            Nej_obec_vrt = Nej_obec_zakl
        End If

    '
        Nej_max_lit_obec
        If Nejistota_lit > Nej_obec_vrt Then
            Nej_max_lit_obec = Nejistota_lit
        Else
            Nej_max_lit_obec = Nej_obec_vrt
        End If

    '
        Zpracování vrtů
        Sniz_vrt_proc = 0
        For j = 1 To Pocet_vrtu
            Z_vrt = Vrtý_Z(j)
            Hloubka_vrt = Vrtý_Hloubka(j)
            Baze_vrt = Z_vrt - Hloubka_vrt
            If (Z_cele <= Z_vrt) And (Z_cele >= Baze_vrt) Then      'Bod výškově sedí
                X_vrt = Vrtý_X(j)
                Y_vrt = Vrtý_Y(j)
                V = Vzdal_2D(x, y, X_vrt, Y_vrt)
                If V <= Vrtý_R(j) Then      'Zvýšit věrohodnost poměrně podle vzdálenosti
od vrtu od Vrtý_Verohodnost(j) přímo ve vrtu) po 0 (ve vzdálenosti R_vrt)
                    Sniz_vrt_proc = Sniz_vrt_proc + Vrtý_Verohodnost(j) * (1 - V /
Vrtý_R(j))

```

```

        End If
    End If
Next j

'
Zpracování řezů
Sniz_rez_proc = 0
For i = 1 To Pocet_useku_Rezu
    If Z_cele >= Hloubka_Rezy(i) Then 'jen do Z m n.m. řezu
        If Abs(X2_Rezy(i) - X1_Rezy(i)) < 0.001 Then
            smernice = 1000 'kolmice
        Else
            smernice = (Y2_Rezy(i) - Y1_Rezy(i)) / (X2_Rezy(i) - X1_Rezy(i))
        End If
        Prus_kruz_prim x, y, Vzdal_rez(i), smernice, Y1_Rezy(i) - smernice *
X1_Rezy(i), exist
        If exist Then
            Prus_primek smernice, Y1_Rezy(i) - smernice * X1_Rezy(i), -1 /
smernice, y + x / smernice, X_Rez, Y_Rez
            If (X_Rez >= X1_Rezy(i) - Vzdal_rez(i)) And (X_Rez <= X2_Rezy(i)
+ Vzdal_rez(i)) Or (X_Rez >= X2_Rezy(i) - Vzdal_rez(i)) And (X_Rez <= X1_Rezy(i) +
Vzdal_rez(i)) Then
                If (Y_Rez >= Y1_Rezy(i) - Vzdal_rez(i)) And (Y_Rez <=
Y2_Rezy(i) + Vzdal_rez(i)) Or (Y_Rez >= Y2_Rezy(i) - Vzdal_rez(i)) And (Y_Rez <= Y1_Rezy(i)
+ Vzdal_rez(i)) Then
                    V = Vzdal_2D(x, y, X_Rez, Y_Rez)
                    Sniz_rez_proc = Sniz_rez_proc + Nejistota_Rez_minVzdal(i)
                * (1 - V / Vzdal_rez(i))
            End If
        End If
    End If
Next i
If Sniz_rez_proc > Max_rezy Then
    Sniz_rez_proc = Max_rezy
End If

'
Zpracování důlních děl
Sniz_dul_dila_proc = 0
For i = 1 To Pocet_useku_DD
    If typ_DD(i) = "H" Then 'horizontální dílo
        If (Z_cele > (Z1_DD(i) + Vzdal_DD(i))) Or (Z_cele < (Z1_DD(i) -
Vzdal_DD(i))) Then GoTo Dalsi_usek 'jsme mimo vliv důlního díla
    Else 'vertikální dílo
        If Z1_DD(i) < Z2_DD(i) Then
            MensiZ = Z1_DD(i)
            VetsiZ = Z2_DD(i)
        Else
            MensiZ = Z2_DD(i)
            VetsiZ = Z1_DD(i)
        End If
        If (Z_cele > (VetsiZ + Vzdal_DD(i))) Or (Z_cele < (MensiZ -
Vzdal_DD(i))) Then GoTo Dalsi_usek 'jsme mimo vliv důlního díla
    End If
    Vzdal_bod_prim3D x, y, Z_cele, X1_DD(i), Y1_DD(i), Z1_DD(i), X2_DD(i),
Y2_DD(i), Z2_DD(i), Vzdal, Uvnitr
    If Uvnitr Then 'Bod [x,y,Z_cele] leží v oblasti vlivu úseku
        důlního díla (kolmo na úsek)
        If Vzdal < Vzdal_DD(i) Then
            Sniz_dul_dila_proc = Sniz_dul_dila_proc + Nejistota_DD_min-
Vzdal(i) * (1 - Vzdal / Vzdal_DD(i))
        End If
    End If
Dalsi_usek:
Next i
If Sniz_dul_dila_proc > Max_DD Then
    Sniz_dul_dila_proc = Max_DD
End If

'
Upr_Nej_max_lit_obec
Upr_Nej_max_lit_obec = Nej_max_lit_obec - Nej_max_lit_obec * (Sniz_vrt_proc +
Sniz_rez_proc + Sniz_dul_dila_proc) * 0.01
If Upr_Nej_max_lit_obec < Min_celk_nejis Then Upr_Nej_max_lit_obec =
Min_celk_nejis

```

**SOFTWARE PRO TVORBU 3D MODELU MÍRY NEJISTOTY STRUKTURNĚ
GEOLOGICKÉHO MODELU (PRO LOKALITU KAŇK)**

```

        Print #1, Nazev_vrtu & " " & x & " " & y & " " & Z_cele & " " & Round(Nejis-
tota_lit, 2) & " " & Round(Nej_obec_vrt, 2) & " " & Round(Nej_max_lit_obec, 2) & " " & Round(Sniz_vrt_proc, 2) & " " &
        Round(Sniz_rez_proc, 2) & " " & Round(Sniz_dul_dila_proc, 2) & " " &
Round(Upr_Nej_max_lit_obec, 2)
        Pocet_bodu_1 = Pocet_bodu_1 + 1
        Z_cele = Z_cele - krok_Z
    Loop

Dalsi_vrt:
    j_vrty = j_vrty + 1
    If j_vrty Mod 100 = 1 Then
        Sheets("Legenda").Select
        k = k + 1
        Application.ScreenUpdating = True
        Cells(29 + k, 1) = "zpracováno: " & j_vrty & ", " & Date & ", " & Time
        Application.Wait Now + TimeValue("00:00:01") 'zpoždění 1 sec
        Application.ScreenUpdating = False
    End If
    If j_vrty = j_vrty_Do + 1 Then GoTo ladeni 'Do zadaného řádku vrtu
    Sheets("AllVrt10x10m").Select
Loop
ladeni:
    Close #1
    Sheets("Legenda").Select
    Cells(29, 1) = "Ukončeno: " & Date & ", " & Time & ", Počet bodů 3D modelu nejistoty:
" & Pocet_bodu_1
    MsgBox ("Konec výpočtu, Nejistota_3D.dat vytvořen!"), , "Oznámení"
End Sub

Public Function Vzdal_2D(ByVal x1 As Double, ByVal y1 As Double, ByVal x2 As Double, ByVal
y2 As Double) As Double
    Vzdal_2D = Sqr((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1))
End Function

Public Sub Vyres_kv_rov(ByVal a As Double, ByVal b As Double, ByVal c As Double, exist As
Boolean)
'Public Sub Vyres_kv_rov(ByVal a As Double, ByVal b As Double, ByVal c As Double, x1 As
Double, x2 As Double, exist As Boolean)
Dim d As Double
    d = b * b - 4 * a * c
    If d <= 0 Then 'Průsečíky neexistují nebo je to tečna (d=0)
        exist = False
    Else
        exist = True
        x1 = (-b + Sqr(d)) / (2 * a)
        x2 = (-b - Sqr(d)) / (2 * a)
    End If
End Sub

Public Sub Prus_kruz_prim(ByVal m As Double, ByVal n As Double, ByVal r As Double, ByVal
k As Double, _
ByVal q As Double, exist As Boolean)
'ByVal q As Double, x1 As Double, y1 As Double, x2 As Double, y2 As Double, exist As
Boolean) nám stačí, zda existují
'    Pruseciiky [x1,y1] a [x2,y2] kruznice se stredem [m,n] a polomeru r s primkou y=kx+q
'    Vyres_kv_rov 1 + k * k, -2 * m + 2 * k * q - 2 * n * k, m * m + q * q - 2 * n * q +
n * n - r * r, x1, x2, exist
'    Vyres_kv_rov 1 + k * k, -2 * m + 2 * k * q - 2 * n * k, m * m + q * q - 2 * n * q + n
* n - r * r, exist
End Sub

Public Sub Prus_primek(ByVal k1 As Double, ByVal q1 As Double, ByVal k2 As Double, ByVal
q2 As Double, _
xp As Double, yp As Double)
'prusecik přímek y=k1*x+q1 a y=k2*x+q2,
    xp = (q1 - q2) / (k2 - k1)
    yp = (q1 * k2 - q2 * k1) / (k2 - k1)
End Sub

Public Function Vzdal_3D(ByVal x1 As Double, ByVal y1 As Double, ByVal z1 As Double, ByVal
x2 As Double, ByVal y2 As Double, ByVal z2 As Double) As Double
    Vzdal_3D = Sqr((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1) + (z2 - z1) * (z2 - z1))

```

End Function

```

Sub PTLOC_2_rozm_pole(LOC As Integer, I1 As Integer, I2 As Integer, ByVal NP As Integer,
ByVal Rozmer_1 As Integer, x() As Double, y() As Double, _
xp As Double, yp As Double)
'Procedure PTLOC(var LOC,I1,I2:integer; NP:integer;var X,Y:polygon_delp; XP,YP:real);
' {Urceni polohy bodu [xp,yp] vuci polygonu [x1,y1]..[xnp,ynp],
' vystupni parametry LOC,I1,I2
' bod uvnitr 1, 0, 0
' bod vne - 1, 0, 0
' bod na hranici 0,I1,I2 (poradi vrcholu strany)
' bod ve vrcholu 2,I1,I1 }
Dim C180 As Double
Dim TSC As Double
Dim TCC As Double
Dim CC As Double
Dim x1 As Double
Dim y1 As Double
Dim x2 As Double
Dim y2 As Double
Dim R1 As Double
Dim R2 As Double
Dim VPL As Double
Dim ST As Double
Dim CT As Double
Dim SC As Double
Dim i As Integer
Dim IEC As Integer
Dim IP As Integer

C180 = 0.9999999998
I1 = 0
I2 = 0
IEC = 0
TSC = 0
TCC = 1
CC = 1
SC = 0
IP = NP
x1 = x(Rozmer_1, NP) - xp
y1 = y(Rozmer_1, NP) - yp ' (0,0) do testovaneho bodu
R1 = Sqr(x1 * x1 + y1 * y1)
For i = 1 To NP
    x2 = x(Rozmer_1, i) - xp
    y2 = y(Rozmer_1, i) - yp
    R2 = Sqr(x2 * x2 + y2 * y2)
    VPL = R1 * R2
    If R1 = 0 Or R2 = 0 Then 've vrcholu }
        LOC = 2
        I1 = i
        I2 = i
        If R1 = 0 Then
            I1 = IP
            I2 = IP
        End If
        Exit Sub
    End If
    ST = (x1 * y2 - x2 * y1) / VPL
    CT = (x1 * x2 + y1 * y2) / VPL
    If CT + C180 <= 0 Then 'na hranici
        LOC = 0
        I1 = IP
        I2 = i
        Exit Sub
    End If
    If i - NP >= 0 Then
        If (IEC \ 2) * 2 - IEC < 0 Then
            LOC = 1 'uvnitr
        Else
            LOC = -1 'vne
        End If
        Exit Sub
    End If
End For
End Sub

```

```

TSC = SC * CT + CC * ST
TCC = CC * CT - SC * ST
If Not ((TSC * SC > 0) Or (TSC * ST >= 0)) Then IEC = IEC + 1
SC = TSC
CC = TCC
x1 = x2
y1 = y2
R1 = R2
IP = i
Next i
LOC = -1
End Sub

Public Sub Vzdal_bod_prim3D(ByVal x As Double, ByVal y As Double, ByVal z As Double, ByVal
x1 As Double, _
ByVal y1 As Double, ByVal z1 As Double, ByVal x2 As Double, ByVal y2 As Double, ByVal z2
As Double, Vzdal As Double, Uvnitr As Boolean)
' Určení vzdálenosti bodu [x,y,z] s úsečkou zadanou body [x1,y1,z1] a [x2,y2,z2]. Pokud
není průsečík přímky a roviny kolmé
' na tuto přímku uvnitř úsečky [x1,y1,z1], [x2,y2,z2] nastaví se parametr Uvnitr na
false.
Dim u1 As Double, u2 As Double, u3 As Double, d As Double, t As Double
Dim xp As Double, yp As Double, zp As Double
' Směrnice přímky (u1,u2,u3)
u1 = x2 - x1
u2 = y2 - y1
u3 = z2 - z1
' Parametr d roviny ax+by+cz+d=0 kolmé k přímce
d = -u1 * x - u2 * y - u3 * z
' Výpočet parametru t pro určení průsečíku
t = (-u1 * x1 - u2 * y1 - u3 * z1 - d) / (u1 * u1 + u2 * u2 + u3 * u3)
' Průsečík
xp = x1 + u1 * t
yp = y1 + u2 * t
zp = z1 + u3 * t
' Leží uvnitř úsečky [x1,y1,z1], [x2,y2,z2]?
Uvnitr = False
Vzdal = 999999
If (xp >= x1) And (xp <= x2) Or (xp >= x2) And (xp <= x1) Then
If (yp >= y1) And (yp <= y2) Or (yp >= y2) And (yp <= y1) Then
If (zp >= z1) And (zp <= z2) Or (zp >= z2) And (zp <= z1) Then
Uvnitr = True
Vzdal = Vzdal_3D(x, y, z, xp, yp, zp)
End If
End If
End If
End Sub

```