

Prílohy

Príloha 1 – zdrojový kód procedúry redgrad

```
Sub redgrad()  
    '----- deklaracia premennych -----      i,j,p,q su pomocne  
    premenne  
    Dim F As Range                ' ucelova funkcia  
    Dim DF As Range               ' gradient ucelovej funkcie  
    Dim A As Range                ' lave strany ohraniceni  
    Dim B As Range                ' prave strany ohraniceni  
    Dim X As Range                ' pole premennych  
    Dim vyp As Range              ' bunka od ktorej sa vypisuju vypocty  
    Dim az As Range                ' prepocty az  
    Dim Rr As Range  
    Dim RB As Range  
    Dim RN As Range  
    Dim RNT As Range  
    Dim RBDF As Range  
    Dim RNEG As Range  
    Dim RNEG2 As Range  
    Dim Rzj As Range  
    Dim RzN As Range  
    Dim RzB As Range  
    Dim Rzk As Range  
    Dim Ralfa As Range  
    Dim Ralfak As Range  
    Dim BI(100) As Double          ' indexy stlpcov pre B  
    Dim NI(100) As Double          ' indexy stlpcov pre N  
    Dim DFset(100) As Double        ' pomocna mnozina1 pre prepocet  
    indexov  
    Dim ite As Integer  
  
    '----- priradenie poli zadavanych uzivatelom -----  
    Set F = Application.InputBox("Oznac pole s ucelovou funkciou (F)",  
    Type:=8)  
    Set DF = Application.InputBox("Oznac pole gradientu UF (DF)",  
    Type:=8)  
    Set A = Application.InputBox("Oznac pole lavych stran ohraniceni  
    (A)", Type:=8)  
    Set B = Application.InputBox("Oznac pole pravych stran ohraniceni  
    (B)", Type:=8)  
    Set X = Application.InputBox("Oznac pole premennych (X)", Type:=8)
```

```

Set az = Application.InputBox("Oznac pole (az)", Type:=8)
Set vyp = Application.InputBox("Oznac pole pre pre vypis
vypoctov", Type:=8)
ite = InputBox("Zadaj pocet iteracii")

'----- definicia pomocnych premennych -----
l = az.Count           ' pocet premennych v UF
m = B.Count           ' pocet ohraniceni
n = X.Count           ' celkovy pocet premennych
Vx = vyp.Row          ' suradnice vystupu
Vy = vyp.Column
k = 0                 ' iteracia
inf = 10000000000     ' inf predstavuje najväcsie mozne
cislo

'----- vymedzenie oblasti vystupov -----
Cx = Cells(Vx + 2, Vy + 1).Row
cy = Cells(Vx + 2, Vy + 1).Column
BDFx = Cells(Vx + 4 + 2 * m, Vy + 1).Row
BDFy = Cells(Vx + 4 + 2 * m, Vy + 1).Column
Rx = Cells(Vx + 6 + 2 * m, Vy + 1).Row
Ry = Cells(Vx + 6 + 2 * m, Vy + 1).Column

Set Rr = Range(Cells(Rx, Ry), Cells(Rx, Ry + n - 1))
Set RB = Range(Cells(Cx, cy), Cells(Cx + m - 1, cy + m - 1))
Set RN = Range(Cells(Cx + m + 1, cy), Cells(Cx + n, cy + n - m - 1))
Set RNEG = Range(Cells(Cx, cy + m + n + 1), Cells(Cx + n - 1, cy + m + 2 * n))
Set RNT = Range(Cells(RNEG.Row + n + 1, RNEG.Column), Cells(RNEG.Row + 2 * n - m, RNEG.Column + m - 1))
Set RNEG2 = Range(Cells(Cx, cy + m + n + 1), Cells(Cx + m - 1, cy + 2 * m + n))
Set RBDF = Range(Cells(BDFx, BDFy), Cells(BDFx, BDFy + m - 1))
Set Rzj = Range(Cells(Rr.Row + 2, Rr.Column), Cells(Rr.Row + 2, Rr.Column + n - 1))
Set RzN = Range(Cells(Rzj.Row + 2, Rzj.Column), Cells(Rzj.Row + 2, Rzj.Column + n - m - 1))
Set RzB = Range(Cells(RzN.Row + 2, RzN.Column), Cells(RzN.Row + 2, RzN.Column + m - 1))
Set Rzk = Range(Cells(RzB.Row + 2, RzB.Column), Cells(RzB.Row + 2, RzB.Column + n - 1))

```

```

Set Ralfa = Range(Cells(Rzk.Row + 2, Rzk.Column), Cells(Rzk.Row +
2, Rzk.Column))
Set Ralfak = Range(Cells(Rzk.Row + 4, Rzk.Column), Cells(Rzk.Row +
4, Rzk.Column))
Cells(Vx, Vy) = "k"
Cells(Vx + 2, Vy) = "B"
Cells(Vx + 3 + m, Vy) = "N"
Cells(Vx + 4 + 2 * m, Vy) = "BDF"
Cells(Vx + 6 + 2 * m, Vy) = "r"
Cells(Vx + 8 + 2 * m, Vy) = "zj"
Cells(Vx + 10 + 2 * m, Vy) = "zN"
Cells(Vx + 12 + 2 * m, Vy) = "zB"
Cells(Vx + 14 + 2 * m, Vy) = "zk"
Cells(Vx + 16 + 2 * m, Vy) = "alfa"
Cells(Vx + 18 + 2 * m, Vy) = "alfak"

```

```

'----- zaciatok iteracie -----

```

```

For t = 1 To ite
k = k + 1
Cells(Vx, Vy + 1) = k

```

```

'----- identifikacia a zoradenie indexov NI a BI -----

```

```

For i = 0 To 100
    DFset(i) = -inf          ' vyprazdnenie pomocnej premennej
Next i

```

```

For j = 1 To n
    DFset(j) = X(j)          ' ulozenie hodnot x do pomocnej premennej
Next j

```

```

For i = 1 To m              ' vyber indexov do mnoziny B
    For j = 1 To n
        If (DFset(j) = Application.Max(DFset)) Then
            BI(i) = j
            DFset(j) = -inf
            Exit For
        End If
    Next j
Next i

```

```

For i = 1 To m          ' zoradenie prvkov v mnozine B
    For j = 1 To m
        If (BI(i) < (BI(j))) Then
            p = BI(i)
            q = BI(j)
            BI(i) = q
            BI(j) = p
            Exit For
        End If
    Next j
Next i

For i = 0 To 100
    DFset(i) = inf      ' vyprazdnenie pomocnej premennej
Next i

For j = 1 To n          ' ulozenie hodnot x do pomocnej premennej
    DFset(j) = X(j)
Next j

For i = 1 To n - m      ' vyber indexov do mnoziny N
    For j = 1 To n
        If (DFset(j) = Application.Min(DFset)) Then
            NI(i) = j
            DFset(j) = inf
            Exit For
        End If
    Next j
Next i

For i = 1 To n - m      ' zoradenie prvkov v mnozine N
    For j = 1 To n - m
        If (NI(i) < (NI(j))) Then
            p = NI(i)
            q = NI(j)
            NI(i) = q
            NI(j) = p
            Exit For
        End If
    Next j
Next i

```

```

'----- rozdelenie matice A na C a D, priradenie BDF-----
For i = 1 To m
    RBDF(i) = DF(BI(i))
    For j = 1 To m
        RB(i, j) = A(i, BI(j))
    Next j

    For j = 1 To n - m
        RN(i, j) = A(i, NI(j))
    Next j
Next i

'----- vypocet r -----
For i = 1 To n          ' tvorba -I matice
    For j = 1 To n
        If (i = j) Then
            RNEG(i, j) = -1
        Else
            RNEG(i, j) = 0
        End If
    Next j
Next i

Rr = Application.MMult(RBDF, Application.MInverse(RB), A)
Rr = Application.MMult(Rr, RNEG)
For i = 1 To n
    Rr(i) = Rr(i) + DF(i)
Next i

'----- urcenie zj zA a zB -----
For j = 1 To n
    If Rr(j) > 0 Then
        Rzj(j) = (-X(j)) * Rr(j)
    Else
        Rzj(j) = -Rr(j)
    End If
Next j

```

```

If Application.Sum(Rzj) < 1 / inf Then
    If Application.Sum(Rzj) > -1 / inf Then
        RZN = "N/A"
        RZB = "N/A"
        RZK = "N/A"
        Ralfa = "N/A"
        Ralfak = "N/A"
        az = "N/A"
        Exit For          ' ukoncenie algoritmu
    End If
End If

For j = 1 To n - m
    RZN(j) = Rzj(NI(j))
Next j

RNT = Application.Transpose(RN)          ' transponacia matice N
RZB = Application.MMult(RZN, RNT)
RZB = Application.MMult(RZB, Application.MMult(RNEG2,
Application.MInverse(RB)))

For i = 1 To m
    RZK(BI(i)) = RZB(i)
Next i

For j = 1 To n - m
    RZK(NI(j)) = RZN(j)
Next j

'----- vypočet alfa -----
Ralfa = inf
For j = 1 To n
    If RZK(j) < 0 Then
        If (-X(j) / RZK(j)) < Ralfa Then
            Ralfa = (-X(j) / RZK(j))
        End If
    End If
Next j

```

```

'----- hľadanie minima -----
az = 0
q = 0
For i = 1 To 1
    az(i) = Ralfa * Rzk(i)
Next i

adr = Cells(14, 1).Address
For j = 1 To 1
    az(j).FormulaArray = "=" & Ralfak.Address & "*" &
Rzk(j).Address & ""
Next j

F.Name = "SF"
Ralfak.Name = "Salfak"
Ralfa.Name = "Salfa"

solverreset
SolverOk SetCell:="SF", MaxMinVal:=2, ValueOf:="0",
ByChange:="Salfak"
SolverAdd CellRef:="Salfak", Relation:=1, FormulaText:="Salfa"
SolverAdd CellRef:="Salfak", Relation:=3, FormulaText:="0"
SolverOk SetCell:="SF", MaxMinVal:=2, ValueOf:="0",
ByChange:="Salfak"
SolverSolve True

' ----- prepocet x -----
For i = 1 To n
    X(i) = X(i) + (Ralfak * Rzk(i))
Next i

Next t

End Sub

```

Príloha 2 – príklad

Zadanie príkladu

Uvažujme úlohu nelineárneho programovania s dvoma lineárnymi ohraničeniami a nelineárnou účelovou funkciou

$$\min f(x) = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2$$

$$x_1 + x_2 + x_3 = 2$$

$$x_1 + 5x_2 + x_4 = 5$$

$$x_1, x_2, x_3, x_4 \geq 0.$$

Ako počiatočný východiskový bod zvolíme $x^0 = (0, 0, 2, 5)$ a vypočítame gradient účelovej funkcie $\nabla f(x) = (4x_1 - 2x_2 - 4, 4x_2 - 2x_1 - 6, 0, 0)^T$. Takto pripravenú úlohu vložíme do hárku MS Excelu tak, ako je to ukázané na Obr. P2.1 a nadefinujeme vzťahy medzi bunkami.. Zavoláme procedúru *redgrad* a zadáme vstupné parametre nasledovne:

F = "B3"
 DF = "B6:E6"
 A = "B9:E10"
 B = "F9:F10"
 X = "B2:E2"
 az = "B12:C12"
 vyp = "A19"
 ite = 1

	A	B	C	D	E	F	G
1		x1	x2	x3	x4		
2	x_x^T	0	0	2	5		
3							
4	UF:	0					
5							
6	df^T	-4.00	-6.00	0.00	0.00		
7							
8		A				b	
9	dg1	1	1	1	0	2	
10	dg2	1	5	0	1	5	
11							
12	az						
13							

Obr. P2.1

Po dokončení procedúry dostaneme výstup, vyobrazený na *Obr. P2.3*. Okrem iného môžeme pozorovať, že smer posunu v prvej iterácii $z^1 = (4, 6, -10, -34)$ a dĺžka kroku $\alpha_1^* = 0,147058824$. Nové riešenie $x^1 = (0,588; 0,882; 0,529; 0)$ je možno vidieť na *Obr. P2.2*, ako aj novú hodnotu účelovej funkcie $f(x^1) = -10,45$. Opätovným zavolaním procedúry *redgrad* s parametrom *ite* = 1, alebo zadaním parametru *ite* = 2 pri pôvodnej úlohe dosiahneme výsledky vyobrazené na *Obr. P2.4* a *Obr. P2.5*.

V prípade, že pri volaní procedúry zadáme počet iterácií *ite* = 3 alebo viac, algoritmus skončí dopočítaním konečného riešenia v tretej iterácii s výstupom, ktorý môžeme vidieť na *Obr. P2.6* a *Obr. P2.7*. Tu môžeme vidieť, že všetky prvky z_j sú nulové, algoritmus teda dospel ku konečnému riešeniu a nemôže pokračovať. V platnosti teda zostávajú výsledky z druhej iterácie $x^2 = (1,129; 0,774; 0,097; 0)$ a $f(x^2) = -6,436$.

Prvá iterácia

	A	B	C	D	E	F	G
1		x1	x2	x3	x4		
2	x_x^T	0.588235294	0.882352941	0.529411765	0		
3							
4	UF:	-10.449827					
5							
6	df^T	-3.41	-3.65	0.00	0.00		
7							
8		A				b	
9	dg1	1	1	1	0	2	
10	dg2	1	5	0	1	5	
11							
12	az	0.588235294	0.882352941				
13							

Obr. P2.2

	A	B	C	D	E	F	G
19	k	1					
20							
21	B	1	0				
22		0	1				
23							
24	N	1	1				
25		1	5				
26							
27	BDF	0	0				
28							
29	r	-4	-6	0	0		
30							
31	zj	4	6	0	0		
32							
33	zN	4	6				
34							
35	zB	-10	-34				
36							
37	zk	4	6	-10	-34		
38							
39	alfa	0.147058824					
40							
41	alfak	0.147058824					
42							

Obr. P2.3

Druhá iterácia

	A	B	C	D	E	F	G
1		x1	x2	x3	x4		
2	x_x^T	1.129032258	0.774193548	0.096774194	0		
3							
4	UF:	-6.43598616					
5							
6	df^T	-1.03	-5.16	0.00	0.00		
7							
8		A				b	
9	dg1	1	1	1	0	2	
10	dg2	1	5	0	1	5	
11							
12	az	0.540796964	-0.10815939				
13							

Obr. P2.4

	A	B	C	D	E	F	G
19	k	2					
20							
21	B	1	1				
22		1	5				
23							
24	N	1	0				
25		0	1				
26							
27	BDF	-3.41176471	-3.64705882				
28							
29	r	0	0	3.352941176	0.058823529		
30							
31	zj	0	0	-1.77508651	0		
32							
33	zN	-1.77508651	0				
34							
35	zB	2.218858131	-0.44377163				
36							
37	zk	2.218858131	-0.44377163	-1.77508651	0		
38							
39	alfa	0.298245614					
40							
41	alfak	0.243727599					
42							

Obr. P2.5

Tretia iterácia

	A	B	C	D	E	F	G	H
1		x1	x2	x3	x4			
2	x_x^T	1.129032258	0.774193548	0.096774194	0			
3								
4	UF:	#HODNOTA!						
5								
6	df^T	-1.03	-5.16	0.00	0.00			
7								
8		A				b		
9	dg1	1	1	1	0	2		
10	dg2	1	5	0	1	5		
11								
12	az	N/A	N/A					
13								

Obr. P2.6

	A	B	C	D	E	F	G
19	k	3					
20							
21	B	1	1				
22		1	5				
23							
24	N	1	0				
25		0	1				
26							
27	BDF	-1.03225806	-5.16129032				
28							
29	r	0	0	-5.107E-15	1.032258065		
30							
31	zj	0	0	5.10703E-15	0		
32							
33	zN	N/A	N/A				
34							
35	zB	N/A	N/A				
36							
37	zk	N/A	N/A	N/A	N/A		
38							
39	alfa	N/A					
40							
41	alfak	N/A					
42							

Obr. P2.7