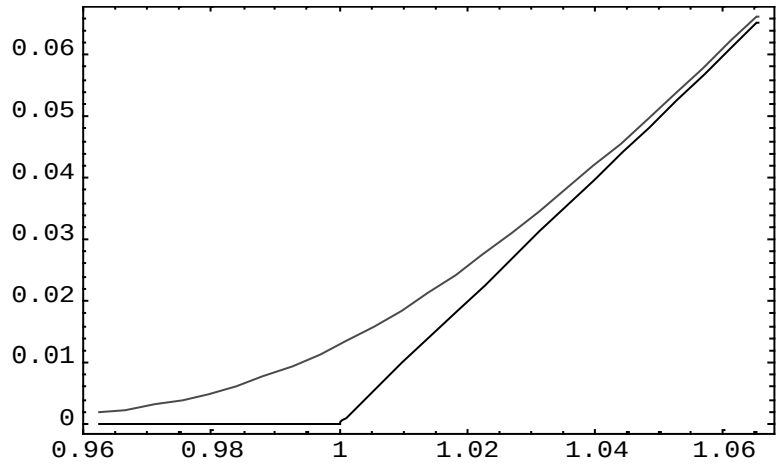
Dodatok C

Program na výpočet premenlivej volatility

V tomto dodatku uvedieme celý program na výpočet premenlivej volatility, závislej od ceny akcie $\sigma(S)$, aj s výsledkami jeho jednotlivých častí.

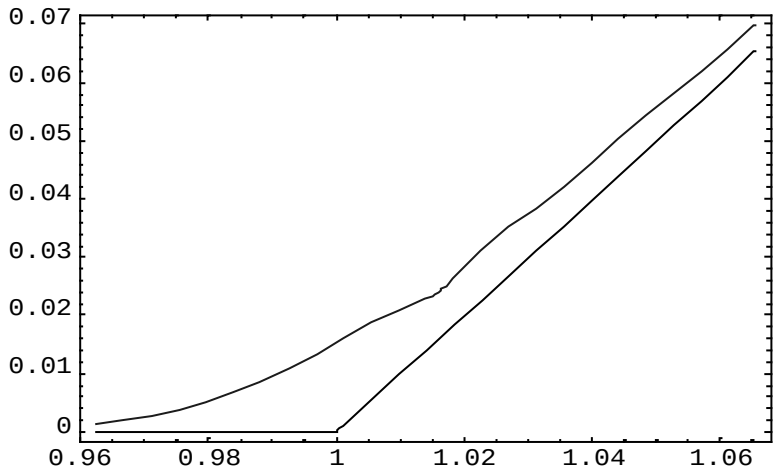
```
(* Nacitanie realnych hodnot cien opcii *)
realnedata= Reverse[ReadList["./sap12-11-98.txt", Number,RecordLists->True]];
minimumS=Min[ Table[realnedata[[i]][[1]], {i,1,Length[realnedata]}]];
maximumS=Max[ Table[realnedata[[i]][[1]], {i,1,Length[realnedata]}]];
minimumV=Min[ Table[realnedata[[i]][[2]], {i,1,Length[realnedata]}]];
maximumV=Max[ Table[realnedata[[i]][[2]], {i,1,Length[realnedata]}]];
```

```
(* Module Black Scholes +
   vykreslenie hodnot realnych dat pomocou Black Scholesa *)
BlackScholes[r_, EE_, D_, sigma_,T_, S_]:=
Module[{V,d1,d2},
        Nd[u_]:=(1.+Erf[u/Sqrt[2]])/2.;
        d1=(Log[S/EE] + (r-D+0.5*sigma^2)*T)/(sigma*Sqrt[T]);
        d2=d1-sigma*Sqrt[T];
        V=Exp[-D*T]*S*Nd[d1] - EE*Exp[-r*T]*Nd[d2];
V];

j2=Plot[{BlackScholes[r, EE, DD, sigmahist,T, S]}, {S,minimumS,maximumS},
   Frame->True, Axes->None,PlotStyle->RGBColor[1,0, 0]];
```



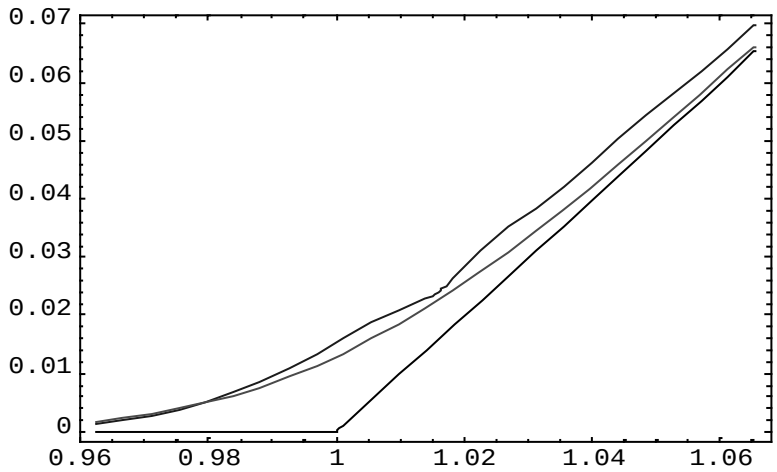
Graf vytvorený pomocou Black-Scholesovej formuly.

```
(* Vytvorenie globalne definovanej funkcie ceny *)
Vrealdata=Interpolation[ realnedata ];
Vreal[S_]:= If[S<minimumS , minimumV*S/minimumS,
                If[S>maximumS, maximumV+ S-maximumS,Vrealdata[S]]];
j1 =Plot[Vreal[S], {S,minimumS,maximumS}, Frame->True, Axes->None,
      PlotStyle->RGBColor[0,0,1]];
j11=Plot[Max[S-1,0], {S,minimumS,maximumS}, Frame->True, Axes->None];
j111=Show[j1,j11];
```



Graf vytvorený pomocou reálnych dát.

```
(* spojenie grafov realnych dat s Black Scholesovymi*)
j3=Plot[{Max[S-1,0]}, {S,minimumS,maximumS}, Frame->True, Axes->None];
Show[j1,j2,j3];
```



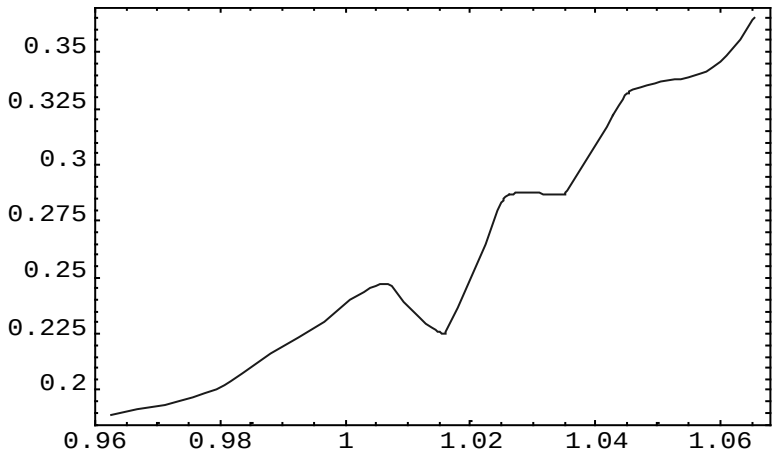
Graf porovnania reálnych dát s dátami vypočítanými pomocou Black-Scholesovej formuly.

```
ImpliedVolatility[r_, EE_, DDD_, S_,T_,Vac_]:=
  sig/.FindRoot[BlackScholes[r, EE, DD, sig, T, S]==Vac,{sig,0.2}]
Do [Print[" implikovana volatilita pre akciu ", realnedata[[i]][[1]]," je ",
     ImpliedVolatility[r,EE,DD,realnedata[[i]][[1]],T,realnedata[[i]][[2]]]]
    ,{ i,1,Length[realnedata]}]
```

implikovaná volatilita pre akciu 0.962473684 je 0.188941
implikovaná volatilita pre akciu 0.97099115 je 0.193546
implikovaná volatilita pre akciu 0.979660714 je 0.200901
implikovaná volatilita pre akciu 0.988486486 je 0.216868
implikovaná volatilita pre akciu 0.997472727 je 0.23227
implikovaná volatilita pre akciu 1.006623853 je 0.247571
implikovaná volatilita pre akciu 1.015944444 je 0.225335
implikovaná volatilita pre akciu 1.025439252 je 0.285718
implikovaná volatilita pre akciu 1.035113208 je 0.287014
implikovaná volatilita pre akciu 1.044971429 je 0.331462
implikovaná volatilita pre akciu 1.055019231 je 0.338771
implikovaná volatilita pre akciu 1.055019231 je 0.338771

```
(* Nacitanie hodnot implikovanej volatility *)
sigmaimplikovanadata=
Reverse[ReadList["./sap12-11-98-implvol.txt", Number,RecordLists->True]];

SigmaImpldata=Interpolation[  sigmaimplikovanadata];
SigmaImpl[S_]:= If[S<minimumS || S>maximumS ,
                        sigmahist, SigmaImpldata[S] ];
```

```
GrafSigmaImpl=
  Plot[SigmaImpl[S], {S,minimumS,maximumS}, Frame->True, Axes->None,
       PlotStyle->RGBColor[0,0,1]]
```



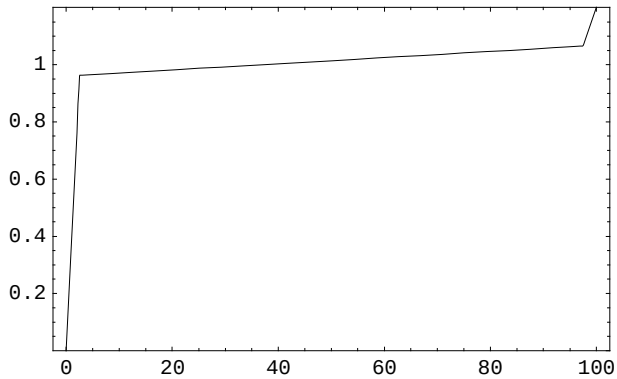
Graf implikovanej volatility pre transformované hodnoty údajov.

```
 (* Nacitanie balika sigma.m a zakladnych konstant *)
Get["./sigma.m"];
r = 0.04382548;
sigmahist=0.2;
EE = 1;
DD = 0.0145;
T = 0.024657534;

m = 20;
n = 100;
SM = 1.2;
MaxIter=30;
```

- *r* - bezriziková úroková miera
- *sigmahist* - hodnota volatility získanej z historických dát pre danú akciu
- *EE* - hodnota expiračnej ceny (v našom prípade pre transformované dáta je táto hodnota konštantná a rovná 1)
- *DD* - spojitý dividendový úrok
- *T* - čas do vypršania platnosti opcie (expiračný čas)
- *m* - počet časových delení
- *n* - počet priestorových delení
- *SM* - horná hranica intervalu

```
penalta=0.025*n;

MeshFunction[i_]:=
If[i < penalta,
        i*minimumS/penalta,
        If[i > n-penalta,
                        maximumS + (i- n +penalta)*(SM-maximumS)/penalta,
                        minimumS+(maximumS-minimumS)*(i- penalta)/(n-2*penalta)
                ]
        ];
```

```
(* Vykreslenie funnkcie delenia intervalu (0,SM) *)
Plot[MeshFunction[i], {i,0,n}, Frame->True,PlotRange->{0,SM}];
```



Mash funkcia vytvorená na delenie pracovného intervalu (0, SM).

```
(* Vytvorenie startovacej hodnoty funkcie sigma *)
sigma=Table[sigmahist, {i,1,n+1}];
```

```
(* Vytvorenie vektora realnych dat *)
V=ValueList[r, EE, DD, sigma, T, m, n, SM];
VRealneData=Table[
Max[
        Max[MeshFunction[i-1]-EE,0.],
 If[MeshFunction[i-1]<minimumS  || MeshFunction[i-1]>maximumS ,
                V[MeshFunction[i-1] ],Vreal[MeshFunction[i-1]]
                                        ]
        ],
{i,1,n+1}];
```

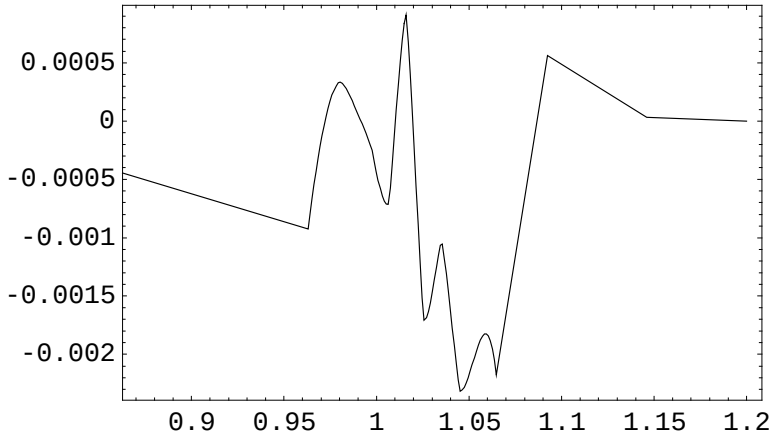
```
(* Zistenie minimalneho a maximalneho platneho indexu *)
Do[ If[  MeshFunction[i-1]<minimumS,imin=i +1 , ]
        ,{i,1,n+1}];
Do[ If[  MeshFunction[i-1]>maximumS,imax=i -1 , ]
        ,{i,n+1,1,-1}];
Print[" Minimalny realny index ", imin];
Print[" Maximalny realny index ", imax];
```

```
" Minimalny realny index "   4
" Maximalny realny index "   98
```

```
(* Vlastne jadro programu *)

(* Prirastok v delta *)
deltasigma=2/EE;

(* Iteracny cyklus na postupne zlepsovanie sigma *)
Do[
Print["Iteracia ", iter];
V=ValueList[r, EE, DD, sigma, T, m, n, SM];

Do[
        novesigma = sigma[[i]] - (V[MeshFunction[i-1]] - VRealneData[[i]])*deltasigma;
        novesigma =Max[0.*sigmahist, novesigma];
  sigma=ReplacePart[sigma, novesigma, i]
,{i,imin,imax}];

Do[ sigma=ReplacePart[sigma, sigmahist, i] ,
                {i,1,imin-1}];
Do[ sigma=ReplacePart[sigma, sigmahist, i] ,{i,imax+1,n+1}];

        L1Chyba=0.;LNekChyba=0.;
        Do[
                L1Chyba=
    L1Chyba+ Abs[  V[MeshFunction[i-1]] -  VRealneData[[i]] ]/(imax-imin);
                LNekChyba=Max[LNekChyba, Abs[  V[MeshFunction[i-1]] -  VRealneData[[i]] ] ];

,{i,imin,imax}];

Print["L1 Chyba  ", L1Chyba,"  L-nekonecno  Chyba  ",
LNekChyba, "  MaxSigma = ", Max[sigma]];

                ,{iter,1,MaxIter}];
```

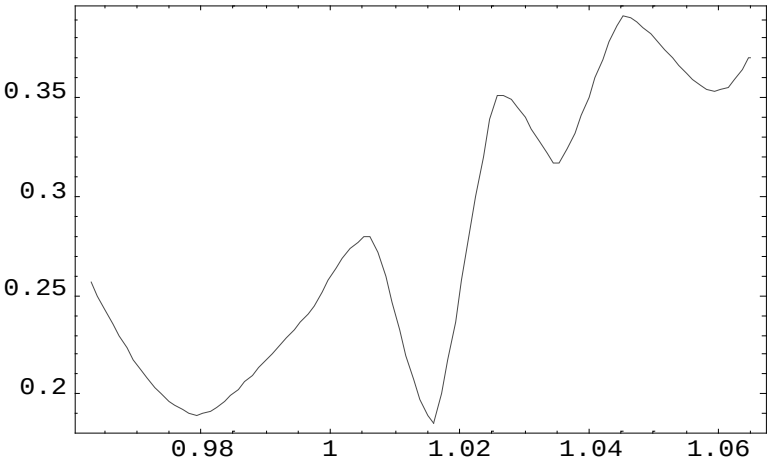
```
Iteracia  1
L1 Chyba   0.00261524    L-nekonecno  Chyba   0.00461431      MaxSigma =  0.209229
Iteracia  2
L1 Chyba   0.00244547    L-nekonecno  Chyba   0.00447085      MaxSigma =  0.21817
Iteracia  3
L1 Chyba   0.00229026    L-nekonecno  Chyba   0.00433133      MaxSigma =  0.226833
Iteracia  4
L1 Chyba   0.00214836    L-nekonecno  Chyba   0.00419658      MaxSigma =  0.235226
Iteracia  5
L1 Chyba   0.00201864    L-nekonecno  Chyba   0.0040671       MaxSigma =  0.24336
Iteracia  6
L1 Chyba   0.00190003    L-nekonecno  Chyba   0.00394322      MaxSigma =  0.251247
Iteracia  7
L1 Chyba   0.00179292    L-nekonecno  Chyba   0.00382505      MaxSigma =  0.258897
Iteracia  8
L1 Chyba   0.00169928    L-nekonecno  Chyba   0.00371262      MaxSigma =  0.266322
Iteracia  9
L1 Chyba   0.00161557    L-nekonecno  Chyba   0.00360584      MaxSigma =  0.273534
Iteracia  10
L1 Chyba   0.00154212    L-nekonecno  Chyba   0.00350459      MaxSigma =  0.280543
Iteracia  11
L1 Chyba   0.00147831    L-nekonecno  Chyba    0.00340867     MaxSigma =   0.28736
Iteracia  12
L1 Chyba   0.00142212    L-nekonecno  Chyba    0.0033179      MaxSigma =   0.293996
Iteracia  13
L1 Chyba   0.00137228    L-nekonecno  Chyba    0.00323204     MaxSigma =   0.30046
Iteracia  14
L1 Chyba   0.00132792    L-nekonecno  Chyba    0.00315085     MaxSigma =   0.306762
Iteracia  15
L1 Chyba   0.00128744    L-nekonecno  Chyba    0.00307411     MaxSigma =   0.31291
Iteracia  16
L1 Chyba   0.00125047    L-nekonecno  Chyba    0.00300158     MaxSigma =   0.318913
Iteracia  17
L1 Chyba   0.00121749    L-nekonecno  Chyba    0.00293302     MaxSigma =   0.324779
Iteracia  18
L1 Chyba   0.00118757    L-nekonecno  Chyba    0.00286821     MaxSigma =   0.330516
Iteracia  19
L1 Chyba   0.00115954    L-nekonecno  Chyba    0.00280694     MaxSigma =   0.33613
Iteracia  20
L1 Chyba   0.00113344    L-nekonecno  Chyba    0.002749       MaxSigma =   0.341628
Iteracia  21
L1 Chyba   0.00110909    L-nekonecno  Chyba    0.0026942      MaxSigma =   0.347016
Iteracia  22
L1 Chyba   0.00108622    L-nekonecno  Chyba    0.00264234     MaxSigma =   0.352301
Iteracia  23
L1 Chyba   0.00106481    L-nekonecno  Chyba    0.00259325     MaxSigma =   0.357487
Iteracia  24
L1 Chyba   0.00104522    L-nekonecno  Chyba    0.00254677     MaxSigma =   0.362581
Iteracia  25
L1 Chyba   0.00102719    L-nekonecno  Chyba    0.00250274     MaxSigma =   0.367586
Iteracia  26
L1 Chyba   0.00101014    L-nekonecno  Chyba    0.00246101     MaxSigma =   0.372508
Iteracia  27
L1 Chyba   0.000994004   L-nekonecno  Chyba    0.00242144     MaxSigma =   0.377351
Iteracia  28
L1 Chyba   0.000978728   L-nekonecno  Chyba    0.00238391     MaxSigma =   0.382119
Iteracia  29
L1 Chyba   0.000964259   L-nekonecno  Chyba    0.00234829     MaxSigma =   0.386816
Iteracia  30
L1 Chyba   0.000950549   L-nekonecno  Chyba    0.00231448     MaxSigma =   0.391444
```

```
(* Zobrazenie odchylky vypocitaneho riesenia od realnych hodnot *)
ListPlot[Table[{MeshFunction[i-1],V[MeshFunction[i-1]] -  VRealneData[[i]]},
{i,1,n+1}], PlotJoined->True,  Frame->True,Axes->None];
```



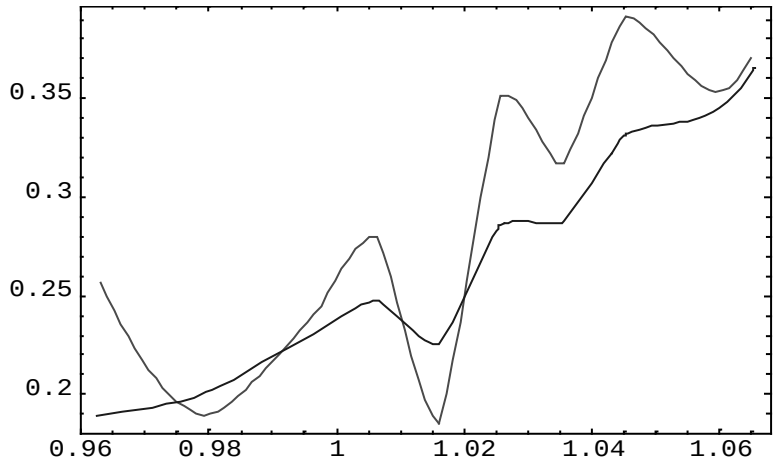
Graf odchyliek premenlivej volatility od implikovanej volatility.

```
 (* Zobrazenie grafov vypocitanej sigma a implikovanej volatility *)
GrafSigma=ListPlot[Table[{MeshFunction[i-1],sigma[[i]]},
{i,imin,imax}], PlotJoined->True,
Frame->True,Axes->None, PlotStyle->RGBColor[1,0,0]];
```



Graf premenlivej implikovanej volatility $\sigma(S)$.

```
Show[GrafSigma,GrafSigmaImpl];
```



Graf porovnania premenlivej implikovanej volatility $\sigma(S)$
s implikovanou volatilitou.

Pomocný program "sigma.m"

Tento program je určený na riešenie Black-Scholesovej rovnice s premenlivou volatilitou závislou na cene akcie $\sigma(S)$.

```
(* Balik sigma.m                    *)
(* na riesenie BS rovnice s uvazovanou *)
(* premenlivou volatilitou          *)

(* Autor: Daniel Sevcovic 1999      *)
(* Datum: 7.11.1999                 *)


Needs["LinearAlgebra`Tridiagonal`"];

penalta=0.025*n;

MeshFunction[i_]:=
If[i < penalta,
        i*minimumS/penalta,
        If[i > n-penalta,
                        maximumS + (i- n +penalta)*(SM-maximumS)/penalta,
                        minimumS+(maximumS-minimumS)*(i- penalta)/(n-2*penalta)
                ]
        ];

SM*(i/n)^(1/4);
```

```
ValueList[r_, EE_, D_, sigma_, T_, m_, n_, SM_]:=
Module[{V,G,a,b,c,adiag,bdiag,cdiag,SMinus,SPlus,S,TStep,VOld, VNew},

(* Module                       *)
(* r  - urokova miera            *)
(* EE - strike price            *)
(* D  - miera dividend           *)
(* T  - expiracny cas            *)
(* sigma - n+1 rozmerny vektor volatility *)
(* m  - pocet casovych deleni    *)
(* n  - pocet priestorovych deleni  *)
(* SM - horna hranica intervalu     *)

(* Payoff funkcia pre Call opciu    *)
PayOffFunction[S_]:=Max[S-EE,0.];

(* Definicia rozdelenia intervalu (0,SM) na delacie body *)


VOld=Table[PayOffFunction[MeshFunction[i]], {i,0,n}];

TStep= T/m;

(* Vypocet tridiagonalnej matice *)
Do[
        SMinus=MeshFunction[i-1];
        S    =MeshFunction[i];
        SPlus =MeshFunction[i+1];

        G[i]=2.*(
                (VOld[[i+2]]-VOld[[i+1]])/(SPlus-S) -
                (VOld[[i+1]]-VOld[[i]])/(S-SMinus)
                )/(SPlus-SMinus);

        c[i] =   - TStep*(sigma[[i]]^2*S^2/(SPlus-S)  + (r-D)*S )/(SPlus-SMinus);
        b[i] = 1 + TStep*(sigma[[i]]^2*S^2/((SPlus-S)*(S-SMinus)) +r ) ;
        a[i] =   - TStep*(sigma[[i]]^2*S^2/(S-SMinus) - (r-D)*S )/(SPlus-SMinus);


 ,{i,1,n-1}];



(* Stanovenie okrajovych podmienok *)
SMinus=MeshFunction[n-1];
S    =MeshFunction[n];
a[n]=  - TStep*(  - (r-D)*S/(S-SMinus) );
b[n]=1.+ TStep*(r - (r-D)*S/(S-SMinus) );

b[0]=1.;
c[0]=0.;


(* Vytvorenie diagonal tridiagonalnej matice *)
adiag=Table[a[i],   {i,1,n}];
bdiag=Table[b[i-1], {i,1,n+1}];
cdiag=Table[c[i-1], {i,1,n}];
```

```
(* Riesenie casovych diskretizacnych krokov *)
        Do[
        VNew=TridiagonalSolve[ adiag, bdiag, cdiag, VOld];

(* Americka  opcia *)
        VOld=Table[Max[
                PayOffFunction[MeshFunction[i-1]],
                VNew[[i]] ],  {i,1,n+1}];

(* Europska opcia *)
(* VOld=VNew; *)

        ,{j,1,m}];

(* Vystup do tvaru interpolacnej funkcie *)
V=Interpolation[Table[{ MeshFunction[i],VOld[[i+1]]  }  ,{i,0,n}] ];
V
(* Koniec modulu *)
];
```