VI. NUMERICKÉ IMPLEMENTÁCIE

Súčasťou tejto práce je aj numerická implementácia niektorých modelov na počítači. Cieľom bolo vytvoriť nielen demonštračné algoritmy, ale aj prakticky použiteľný software. Algoritmy boli naprogramované v jazyku C++ v prostredí Microsoft Visual C++ 6 s použitím ActiveX Template Library ako tzv. COM objekty. COM (Component Object Model) je norma vyvinutá firmou Microsoft pre štandardizovanú komunikáciu medzi objektmi a aplikáciami. Obrovskou výhodou je, že COM objekty sú nezávislé na použitom programovacom jazyku. To znamená, že môžu byť napísané v ľubovoľnom jazyku podporujúcom COM a použité v ľubovoľnom inom jazyku tiež podporujúcom COM. Objekty, ktoré sú súčasťou tejto práce, je možné použiť napríklad v aplikáciách Microsoft Excel, Microsoft Word, z Visual Basicu, Delphi a iných. Taktiež bol vytvorený súbor pre Excel demonštrujúci použitie algoritmov.

COM objekty komunikujú medzi sebou pomocou interfacov, ktoré definuje programátor. Ľubovoľný objekt môže implementovať ľubovoľný počet rôznych interfacov. Z toho vyplýva ďalšia výhoda COM, ktorú je možné demonštrovať na okrajových podmienkach. Napríklad výpočet ceny európskych opcií sa pre jednotlivé typy derivátov líši iba použitím adekvátnych okrajových podmienok (medzi ktoré, samozrejme, patrí aj expiračná podmienka). V rámci software je napísaných niekoľko objektov reprezentujúcich rôzne okrajové podmienky (a tým aj rôzne deriváty; pre ich popis viď prílohu C), pričom všetky implementujú ten istý interface. Algoritmus výpočtu ceny je tiež implementovaný ako objekt (s iným interface). Ak potrebuje hodnotu okrajovej podmienky v nejakom bode, obráti sa na konkrétny objekt podmienky pomocou tohto spoločného interface. Algoritmus výpočtu je takto izolovaný od objektu okrajových podmienok, pozná len jeho interface. Teda vlastne ani „nevie“ s ktorým konkrétnym objektom okrajových podmienok pracuje a aký typ derivátu počíta (to určí programátor pri zavolaní algoritmu), pozná len ten interface. Jednotlivé objekty okrajových podmienok sú tzv. plug–in objekty. Je tiež možné kedykoľvek v budúcnosti dopísať nové objekty reprezentujúce okrajové podmienky nových typov derivátov a existujúce algoritmy ich budú schopné okamžite využiť. Taktiež algoritmus pre výpočet cien amerických opcií používa tieto objekty okrajových podmienok, čo šetrí množstvo nutného kódu. COM umožňuje i distribuované spracovanie, kedy aplikácia používajúca COM objekty beží na jednom počítači v sieti a samotné COM objekty na úplne inom. Laicky sa dá povedať, že tento software je určitá skladačka – zložením algoritmu s okrajovou podmienkou určíme typ opcie, ktorú chceme oceniť. Nevýhoda COM je jeho väzba na platformu Win32, aj keď firma DEC v súčasnej dobe portuje COM na Unix.

V nasledujúcich kapitolách sú matematicky rozobrané jednotlivé algoritmy. Programátorský popis COM objektov a ich zdrojový text je v prílohe C.

VI.1. Black–Scholesov model – implicitná schéma

Napriek tomu, že Black–Scholesov model má pre európsku put a call opciu explicitné riešenie, pomocou ktorého je možné presne určiť cenu týchto opcií (za predpokladu znalosti distribučnej funkcie normálneho rozdelenia s dostatočnou presnosťou), je výhodné implementovať riešenie BS PDR numericky. Výhoda vyplýva z toho, že PDR je nezávislá od okrajových podmienok a teda platná pre rôzne typy derivátov, ktorých cenu môžeme potom ľahko zistiť.

Vo všeobecnosti ide o problém nájdenia riešenia parciálnej diferenciálnej rovnice pre dané okrajové podmienky. Tento problém sa rieši tzv. diskretizáciou PDR, kedy sa derivácie tvoriace rovnicu nahradia ich lineárnymi aproximáciami. Aby toto bolo možné, je nutné obmedziť nekonečné intervaly, na ktorých počítame riešenie, na dostatočne veľké konečné a tieto potom rozdeliť na malé dieliky. Tým dostaneme sieť bodov, v ktorých spočítame hodnoty neznámej funkcie. Pokiaľ je potrebná hodnota medzi bodmi siete, môžeme použiť interpoláciu. Táto metóda sa nazýva *metóda konečných diferencií.*

Zavedením známej substitúcie premenných:

$$S = Ee^{x},$$

$$t = T - \frac{\tau}{\frac{1}{2}\sigma^2},$$

$$V = Ee^{-\frac{1}{2}(k_2-1)x-\left(\frac{1}{4}(k_2-1)^2+k_1\right)\tau}u(x,\tau),$$

$$k_1 = \frac{r}{\frac{1}{2}\sigma^2},\ \ k_2 = \frac{r-D}{\frac{1}{2}\sigma^2},\ \ x \in (-\infty,\infty),\ \ \tau \in \left\langle 0, \frac{1}{2}\sigma^2 T\right\rangle,$$

Zjednodušíme BS PRD na tvar:

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2}$$

s patričnými okrajovými podmienkami. Je výhodné numericky implementovať radšej transformovaný problém než pôvodný, pretože výsledný algoritmus bude jednoduchší a efektívnejší. Z definície parciálnej derivácie u podľa τ

$$\frac{\partial u}{\partial \tau} \approx \frac{u(x,\tau+\Delta\tau)-u(x,\tau)}{\Delta\tau},$$

kde $\Delta\tau$ je malý interval. Táto aproximácia sa nazýva dopredná a vedie k explicitnej metóde konečných diferencií. Explicitná metóda má však určité problémy s konvergenciou a stabilitou pri nevhodne zvolených deleniach intervalov. Naopak jej výhodou je to, že hodnoty hľadanej funkcie v danom časovom okamihu sa dajú vyjadriť ako funkcia hodnôt tej istej funkcie z predchádzajúceho časového okamihu. Ak deriváciu aproximujeme výrazom

$$\frac{\partial u}{\partial \tau} \approx \frac{u(x,\tau)-u(x,\tau-\Delta\tau)}{\Delta\tau}$$

(dozadná aproximácia), dôjdeme k implicitnej metóde konečných diferencií, ktorá je bezpodmienečne stabilná, ale vyžaduje pre každý časový bod riešenie sústavy lineárnych rovníc. V našom prípade sa ukáže, že daná sústava sa dá veľmi efektívne riešiť LU rozkladom a keďže implicitná metóda nevyžaduje toľko časových krokov, koľko explicitná, v konečnom dôsledku môže byť ešte efektívnejšia.

Preferovaným spôsobom aproximácie druhej derivácie podľa priestorovej premennej je centrálna diferencia:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u(x+\Delta x,\tau)-2u(x,\tau)+u(x-\Delta x,\tau)}{(\Delta x)^2}.$$

Pomocou Taylorovho rozvoja sa dá ukázať, že aproximácia prvej derivácie má presnosť $o(\Delta\tau)$, aproximácia druhej derivácie $o\left((\Delta x)^2\right)$. Rozdeľme časový interval $\left\langle 0,\frac{1}{2}\sigma^2 T\right\rangle$ na dieliky veľkosti $\Delta\tau$ a nech ich počet je M. Keďže $x\in(-\infty,\infty)$, musíme tento interval obmedziť na (x^l,x^r) pre $|x^l|$ a $|x^r|$ dosť veľké. Tento interval rozdeľme na N dielikov veľkosti Δx. Tým dostávame časopriestorovú sieť, v bodoch ktorej potom vypočítame približne hodnoty funkcie $u(x,\tau)$. Označme jednotlivé body $u_n^m := u(n\cdot\Delta x, m\cdot\Delta\tau)$.

Aproximovaním derivácií diferenciami (pričom použijeme dozadnú aproximáciu) dostávame:

$$\frac{u_n^m - u_n^{m-1}}{\Delta\tau} = \frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{(\Delta x)^2},$$

po úprave:

$$u_n^m - \alpha\left(u_{n+1}^m - 2u_n^m + u_{n-1}^m\right) = u_n^{m-1},$$

kde $\alpha = \frac{\Delta\tau}{(\Delta x)^2}$, u_n^0 vieme určiť z počiatočnej podmienky a u_{-N}^m, u_N^m z okrajových podmienok. V kompaktnejšom maticovom tvare môžeme problém zapísať takto:

$$\mathbf{M}\mathbf{u}^{\mathbf{m}} = \mathbf{u}^{\mathbf{m-1}} + \mathbf{b}^{\mathbf{m}},$$

kde:

$$\mathbf{M} = \begin{pmatrix} 1+2\alpha & -\alpha & 0 & \cdots & 0 \\ -\alpha & \cdot & \cdot & & \vdots \\ 0 & \cdot & \cdot & \cdot & 0 \\ \vdots & & \cdot & \cdot & -\alpha \\ 0 & \cdots & 0 & -\alpha & 1+2\alpha \end{pmatrix}, \mathbf{u}^{\mathbf{m}} = \begin{pmatrix} u_{N-1}^m \\ \vdots \\ u_0^m \\ \vdots \\ u_{-N+1}^m \end{pmatrix}, \mathbf{b}^{\mathbf{m}} = \begin{pmatrix} u_N^m \\ 0 \\ \vdots \\ 0 \\ u_{-N}^m \end{pmatrix}.$$

Keďže platí $|1+2\alpha| > |2\alpha|$, matica **M** je striktne regulárna (aj striktne diagonálne dominantná) a teda môžeme riešenie sústavy lineárnych rovníc vyjadriť priamo: $\mathbf{v}^{\mathbf{m}} = \mathbf{M}^{-1}\left(\mathbf{v}^{\mathbf{m-1}} + \mathbf{b}^{\mathbf{m}}\right)$. Prakticky je výhodnejšie danú sústavu lineárnych rovníc riešiť pomocou LU rozkladu. Matica **M** je trojdiagonálna matica rozmerov $(2N-2)\times(2N-2)$, preto na jej reprezentáciu v pamäti potrebujeme $6N-8$ čísel (napríklad ak $N = 1000$, číslo je typu double, čo znamená 8 bajtov pamäte, potrebujeme rádovo 48kB). Inverzná matica vo všeobecnosti nie je trojdiagonálna a na jej reprezentáciu v pamäti potrebujeme $(2N-2)^2$ čísel (teda pre $N = 1000$ je to rádovo 32MB, čo je podstatný rozdiel). Okrem toho pre trojdiagonálnu maticu existuje veľmi efektívny spôsob nájdenia LU rozkladu, čo urýchľuje výpočet.

Metóda LU rozkladu spočíva v tom, že maticu **M** rozložíme na súčin dvoch trojuholníkových matíc. Sústavu lineárnych rovníc s trojuholníkovou maticou vieme vyriešiť priamym dosadením. Nech teda matica **M** má vo všeobecnosti tvar:

$$\mathbf{M} = \begin{pmatrix} a_1 & c_1 & 0 & \cdots & 0 \\ b_1 & a_2 & \cdot & & \vdots \\ 0 & \cdot & \cdot & \cdot & 0 \\ \vdots & & \cdot & \cdot & c_{n-1} \\ 0 & \cdots & 0 & b_{n-1} & a_n \end{pmatrix}$$

a nech je silne regulárna. Potom pre takúto maticu existuje jednoznačný LU rozklad s maticami **L** a **U** v tvare:

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_1 & 1 & \cdot & & \vdots \\ 0 & \cdot & \cdot & \cdot & 0 \\ \vdots & & \cdot & \cdot & 0 \\ 0 & \cdots & 0 & l_{n-1} & 1 \end{pmatrix}, \mathbf{U} = \begin{pmatrix} u_1 & z_1 & 0 & \cdots & 0 \\ 0 & \cdot & \cdot & & \vdots \\ 0 & \cdot & \cdot & \cdot & 0 \\ \vdots & & \cdot & \cdot & z_{n-1} \\ 0 & \cdots & 0 & 0 & u_n \end{pmatrix}.$$

Jednotlivé prvky týchto matíc sú dané vzťahmi:

$$u_1 = a_1;\ u_i = a_i - \frac{c_{i-1}b_{i-1}}{u_{i-1}} \quad 2 \le i \le n;\ z_i = c_i,\ l_i = \frac{b_i}{u_i} \quad 1 \le i \le n-1.$$

Riešenie sústavy rovníc $\mathbf{Mx} = \mathbf{q}$ je ekvivalentné riešeniu sústavy $\mathbf{LUx} = \mathbf{q}$. Ak si označíme $\mathbf{y} = \mathbf{Ux}$, riešime následne dve sústavy lin. rovníc: $\mathbf{Ly} = \mathbf{q}$ a $\mathbf{Ux} = \mathbf{y}$, pričom obe majú trojuholníkové matice a vieme ich vyriešiť dosadením:

$$y_1 = q_1;\ y_i = q_i - \frac{b_{i-1}y_{i-1}}{u_{i-1}} \quad 2 \le i \le n;\ x_n = \frac{y_n}{u_n};\ x_n = \frac{y_n - c_n x_{n+1}}{u_n} \quad n-1 \ge i \ge 1.$$

Implicitná schéma je stabilná pre $\alpha > 0$ a konverguje k presnému riešeniu, ak $\Delta\tau \to 0$ a $\Delta x \to 0$.

VI.2. Americké opcie – PSOR

Problém riešenia ceny amerických opcií bol implementovaný pre zápis v tvare lineárnej komplementarity pomocou projektovaného SOR algoritmu. Zasa nie je riešený originálny problém, ale jeho transformovaná verzia s použitím zhodnej transformácie, ako v predchádzajúcej kapitole. Máme teda transformovaný problém zapísaný v tvare lineárnej komplementarity (viď tiež kapitolu III.3):

- $\left(\frac{\partial u}{\partial \tau} - \frac{\partial^2 u}{\partial x^2}\right)\left(u(x,\tau) - g(x,\tau)\right) = 0 \quad \forall x \in \left\langle -x^l, x^p \right\rangle \forall \tau \in \left\langle 0, \frac{1}{2}\sigma^2 T \right\rangle$
- $\left(\frac{\partial u}{\partial \tau} - \frac{\partial^2 u}{\partial x^2}\right) \ge 0 \quad \forall x \in \left\langle -x^l, x^p \right\rangle \forall \tau \in \left\langle 0, \frac{1}{2}\sigma^2 T \right\rangle$
- $\left(u(x,\tau) - g(x,\tau)\right) \ge 0 \quad \forall x \in \left\langle -x^l, x^p \right\rangle \forall \tau \in \left\langle 0, \frac{1}{2}\sigma^2 T \right\rangle.$

Vidíme, že v tejto formulácii priamo nevystupuje voľná hranica a túto je možné zistiť a posteriori z riešenia. Problém najskôr diskretizujeme:

$$\frac{\partial u}{\partial \tau} \approx \frac{u_n^{m+1} - u_n^m}{\Delta\tau},$$

$$\frac{\partial^2 u}{\partial x^2} \approx \theta\left(\frac{u_{n+1}^{m+1} - 2u_n^{m+1} + u_{n-1}^{m+1}}{(\Delta x)^2}\right) + (1-\theta)\left(\frac{u_{n+1}^{m} - 2u_n^{m} + u_{n-1}^{m}}{(\Delta x)^2}\right),$$

pričom $\theta \in \langle 0,1 \rangle$. Pre voľbu $\theta = 1$ dostávame implicitnú schému, pre $\theta = 0$ explicitnú a pre $\theta = 0.5$ Crank–Nicolsonovú. Podmienku $u \geq g$ aproximujeme nerovnicou $u_n^m \geq g_n^m$ pre $m \geq 1$, zo začiatočnej podmienky a okrajových podmienok máme krajné hodnoty. Podmienke $\left(\frac{\partial u}{\partial \tau} - \frac{\partial^2 u}{\partial x^2}\right) \geq 0$ zodpovedá nerovnica:

$$u_n^{m+1} - \alpha\theta\left(u_{n+1}^{m+1} - 2u_n^{m+1} + u_{n-1}^{m+1}\right) \geq u_n^m + \alpha(1-\theta)\left(u_{n+1}^{m} - 2u_n^{m} + u_{n-1}^{m}\right).$$

Definujme:

$$b_n^m = u_n^m + \alpha(1-\theta)\left(u_{n+1}^{m} - 2u_n^{m} + u_{n-1}^{m}\right).$$

Nerovnica sa tým zjednoduší na tvar:

$$u_n^{m+1} - \alpha\theta\left(u_{n+1}^{m+1} - 2u_n^{m+1} + u_{n-1}^{m+1}\right) \geq b_n^m,$$

pričom hodnotu pravej strany nerovnosti b_n^m v čase $m + 1$ vieme určiť. Nakoniec z podmienky lineárnej komplementarity dostávame rovnicu:

$$\left(u_n^{m+1} - \alpha\theta\left(u_{n+1}^{m+1} - 2u_n^{m+1} + u_{n-1}^{m+1}\right) - b_n^m\right)\left(u_n^{m+1} - g_n^{m+1}\right) = 0$$

Zaveďme vektorové označenie:

$$\mathbf{u^m} = \begin{pmatrix} u_{N-1}^m \\ \vdots \\ u_0^m \\ \vdots \\ u_{-N+1}^m \end{pmatrix}, \quad \mathbf{b^m} = \begin{pmatrix} b_{N-1}^m \\ 0 \\ \vdots \\ 0 \\ b_{-N+1}^m \end{pmatrix}, \quad \mathbf{g^m} = \begin{pmatrix} g_{N-1}^m \\ 0 \\ \vdots \\ 0 \\ g_{-N+1}^m \end{pmatrix},$$

kde b_n^m sú dané predchádzajúcim vzťahom okrem:

$$b_{-N+1}^m = u_{-N+1}^m + \alpha(1-\theta)\left(g_{-N}^m - 2u_{-N+1}^m + u_{-N+2}^m\right) + \alpha\theta g_{-N}^{m+1},$$

$$b_{N-1}^m = u_{N-1}^m + \alpha(1-\theta)\left(g_{N}^m - 2u_{N-1}^m + u_{N-2}^m\right) + \alpha\theta g_{N}^{m+1}.$$

Definujme maticu **C**:

$$\mathbf{C} = \begin{pmatrix} 1+2\alpha\theta & -\alpha\theta & 0 & \cdots & 0 \\ -\alpha\theta & \cdot & \cdot & & \vdots \\ 0 & \cdot & \cdot & \cdot & 0 \\ \vdots & & \cdot & \cdot & -\alpha\theta \\ 0 & \cdots & 0 & -\alpha\theta & 1+2\alpha\theta \end{pmatrix}$$

rozmerov $(2N-2)\times(2N-2)$. Potom celý problém v maticovom zápise vyzerá nasledovne:

$$\left(\mathbf{C}\mathbf{v}^{m+1}-\mathbf{b}^{m}\right)\geq 0\,,\ \left(\mathbf{v}^{m+1}-\mathbf{g}^{m+1}\right)\geq 0\,,\ \left(\mathbf{v}^{m+1}-\mathbf{g}^{m+1}\right)^{T}\left(\mathbf{C}\mathbf{v}^{m+1}-\mathbf{b}^{m}\right)=0\,.$$

Označenie $\mathbf{v}\geq 0$ znamená, že každá položka vektora $\mathbf{v}$ je nezáporná. Všimnime si, že rovnicu komplementarity sme zapísali v tvare skalárneho súčinu, t.j. súčet jednotlivých rovníc má byť nulový, nie každá. Keďže máme podmienku nezápornosti pre každý činiteľ a teda aj pre každý sčítanec v skalárnom súčine, ich súčet je nulový práve vtedy, keď sú všetky nulové.

Na riešenie problémov typu nájsť vektor $\mathbf{x}$, pre ktorý platí: $\mathbf{A}\mathbf{x}\geq\mathbf{b}$, $\mathbf{x}\geq\mathbf{c}$, $(\mathbf{x}-\mathbf{c})^{T}(\mathbf{A}\mathbf{x}-\mathbf{b})=0$ sa používa projektovaný SOR (Successive Over Relaxation) algoritmus. Ak je matica $\mathbf{A}$ kladne definitná, dá sa ukázať, že problém má práve jedno riešenie, ktoré existuje. Algoritmus je iteračný. Ako $\mathbf{x}^{0}$ sa zvolí nejaký „odhad“ $\mathbf{x}^{0}\geq\mathbf{c}$ (pre $\mathbf{x}^{0}<\mathbf{c}$ nemusí algoritmus konvergovať). Počas každej iterácie sa vytvára nový vektor $\mathbf{x}^{k+1}$ z aktuálneho vektora $\mathbf{x}^{k}$ v dvoch krokoch pre každé i:

a) sekvenčne vytvoríme pomocnú premennú $y_i^{k+1}=\dfrac{1}{A_{ii}}\left(b_i-\sum_{j=1}^{i-1}A_{ij}x_j^{k+1}-\sum_{j=i+1}^{n}A_{ij}x_j^{k}\right)$,

b) $x_i^{k+1}=\max\left\{c_i,x_i^{k}+\omega\left(y_i^{k+1}-x_i^{k}\right)\right\}$.

Konštanta ω sa nazýva relaxačný parameter a pre $\omega\in(0,2)$ algoritmus konverguje. Iteračný proces sa zastaví, ak $\left\|\mathbf{x}^{k+1}-\mathbf{x}^{k}\right\|<\varepsilon$ (zvolená presnosť). Potom $\mathbf{x}^{k+1}$ reprezentuje aproximáciu riešenia.

V tomto konkrétnom prípade vidieť, že $\mathbf{C}$ je zasa trojdiagonálna, je kladne definitná (ľahko sa dá overiť napríklad metódou rohových determinantov), teda jednak zasa klesá pamäťová náročnosť a zároveň sa PSOR zjednoduší. Pre konkrétnu implementáciu viď prílohu C.