Numerische Verfahren zur Optionspreisbewertung

Seminar
EconoPhysics

Sommersemester 1999

Tobias Galla

Ziel:

Pricing amerikanischer Optionen

dazu: Lösung der Diffusionsgleichung mit freien Randbedingungen

keine analytische Lösung bekannt

$\rightsquigarrow$ numerische Verfahren

Inhalt

1. Diskretisierungsverfahren am Beispiel europäischer Optionen (Diffusionsgleichung)

2. Wiederholung amerikanische Optionen

3. Numerische Behandlung amerikanischer Optionen

Ziel: Lösung der Diffusionsgleichung $\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2}$

Methode: Diskretisiere Variablen τ und x

$\rightsquigarrow$ Gitter

$u(x,\tau)$ nur an Gitterpunkten betrachten:

$$u_n^m = u(n\delta x, m\delta\tau)$$

Ableitungen auf dem Gitter

Kontinuum:

$$\frac{\partial u}{\partial \tau}(u,\tau) = \lim_{\delta\tau\to 0} \frac{u(x,\tau+\delta\tau) - u(x,\tau)}{\delta\tau}$$

Gitter:

$$\Delta_\tau^f u(x,\tau) = \frac{1}{\delta\tau}(u(x,\tau+\delta\tau) - u(x,\tau))$$

$$\Delta_\tau^b u(x,\tau) = \frac{1}{\delta\tau}(u(x,\tau) - u(x,\tau-\delta\tau))$$

Abweichung vom Kontinuum: $\mathcal{O}(\delta\tau)$

symmetrische Gitterableitung:

$$\Delta_\tau^{sym} u(x,\tau) = \frac{1}{2\delta\tau}(u(x,\tau+\delta\tau) - u(x,\tau-\delta\tau))$$

Abweichung: $\mathcal{O}((\delta\tau)^2)$

zweite Ableitung auf dem Gitter:

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

Explizites Iterationsverfahren

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

mit Rand- und Anfangsbedingungen:

$$\begin{aligned} u(x,\tau) &\approx u_{-\infty}(x,\tau) \text{ für } x \to \infty \\ u(x,\tau) &\approx u_{+\infty}(x,\tau) \text{ für } x \to \infty \\ u(x,0) &= u_0(x) \end{aligned}$$

Gitter:

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

$\rightsquigarrow$

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

mit $\alpha = \frac{\delta\tau}{(\delta x)^2}$.

u^m bekannt für ein $m \Rightarrow u^{m+1}$

daher: explizite, iterative Berechnung möglich.

endliche Raumdimension : $N^- \delta x \leq x \leq N^+ \delta x$

Randwerte:

$$\begin{aligned} u_{N^-}^m &= u_{-\infty}(N^- \delta x, m\delta\tau) \qquad \forall m \\ u_{N^+}^m &= u_{+\infty}(N^+ \delta x, m\delta\tau) \qquad \forall m \end{aligned}$$

Startwerte:

$$u_n^0 = u_0(n\delta x) \qquad N^- \leq n \leq N^+$$

Iteration liefert u_n^m für $0 < m \leq M$.

Problem:

Rundungsfehler $\Rightarrow$ Abweichungen (Instabilitäten)

- Stabilität $\Leftrightarrow$ $\alpha \leq \frac{1}{2}$
- Instabilität $\Leftrightarrow$ $\alpha > \frac{1}{2}$

dabei war

$$\alpha = \frac{\delta\tau}{(\delta x)^2}$$

Beispiel:

δx halbieren $\Rightarrow$ $\delta\tau$ vierteln

$\Rightarrow$ Laufzeit $\times 8$

Implizite Verfahren

diskretisiere

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

mit Rückwärts-Zeit-Ableitung $\rightsquigarrow$

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

jetzt: u^{m+1} implizit mit u^m verknüpft.
In Matrizen gesprochen:

$$\begin{pmatrix} 1+2\alpha & -\alpha & 0 & \cdots & 0 \\ -\alpha & 1+2\alpha & -\alpha & \cdots & 0 \\ 0 & -\alpha & & & \vdots \\ \vdots & & & & -\alpha \\ 0 & 0 & & -\alpha & 1+2\alpha \end{pmatrix} \begin{pmatrix} u_{N^-+1}^{m+1} \\ \vdots \\ u_0^{m+1} \\ \vdots \\ u_{N^+-1}^{m+1} \end{pmatrix}$$

$$= \begin{pmatrix} u_{N^-+1}^{m} \\ \vdots \\ u_0^{m} \\ \vdots \\ u_{N^+-1}^{m} \end{pmatrix} + \alpha \begin{pmatrix} u_{N^-}^{m+1} \\ 0 \\ \vdots \\ 0 \\ u_{N^+}^{m+1} \end{pmatrix} =: \begin{pmatrix} b_{N^-+1}^{m} \\ \vdots \\ b_0^{m} \\ \vdots \\ b_{N^+-1}^{m} \end{pmatrix}$$

Haben also:

$$Mu^{m+1} = b^m$$

Man kann zeigen: M ist für $\alpha \geq 0$ invertierbar
$\rightsquigarrow$

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

Also Algorithmus:

- aus u^m und Randbedingung ermittele b^m
- berechne $u^{m+1} = M^{-1} b^m$
- starte dabei mit u^0

Problem: Wie kommt man gut an $M^{-1} b^m$?

LR-Zerlegung

Idee: Zerlege $M = L \cdot R$ mit

- L linker unterer Dreiecksmatrix
- R rechter oberer Dreiecksmatrix

Löse dann wie folgt:

$$\begin{aligned} Mu^{m+1} &= b^m \\ \Leftrightarrow L\,\underbrace{(Ru^{m+1})}_{q^m} &= b^m \\ \Leftrightarrow Lq^m &= b^m \text{ und } Ru^{m+1} = q^m \end{aligned}$$

Dies geschieht durch Vorwärts- bzw. Rückwärtssubstitution !

beachte: Die LR-Zerlegung muß nur einmal durchgeführt werden.

SOR-Methode

SOR = *successive over-relaxation*
Hatten:

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

Idee:

- Annahme: u^m schon berechnet
- Erfinde einen Startwert $u^{m+1,(0)}$
- Setze $u^{m+1,(k)}$ in die rechte Seite ein, erhalte $u^{m+1,(k+1)}$
- Iteriere bis Abbruchbedingung erfüllt.

Abbruchbedingung:

$$\|u^{m+1,(k+1)} - u^{m+1,(k)}\| < \varepsilon$$

Variante (Gauss-Seidel-Verfahren)

Wenn $u_n^{m+1,(k+1)}$ berechnet wird, ist $u_{n-1}^{m+1,(k+1)}$ schon bekannt

$\rightsquigarrow$

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

Vorteil

alter Vektor $u_n^{m+1,(k)}$ kann direkt überschrieben werden $\rightsquigarrow$ Speicherersparnis

Konvergenz ?

für alle $\alpha > 0$

2. Variante: SOR

beachte:

$$u_n^{m+1,(k+1)} = u_n^{m+1,k} + (u_n^{m+1,(k+1)} - u_n^{m+1,k})$$

$\rightsquigarrow$ $(u_n^{m+1,(k+1)} - u_n^{m+1,k})$ als Korrekturterm verstehbar

Idee: over-correct the sequence !

Setze:

$$\begin{aligned} & y_n^{m+1,(k+1)} \\ = \; & \frac{1}{1+2\alpha}\left(b_n^m + \alpha\left(u_{n-1}^{m+1,(k+1)} + u_{n+1}^{m+1,(k)}\right)\right) \end{aligned}$$

und dann

$$\begin{aligned} u_n^{m+1,(k+1)} &= u_n^{m+1,(k)} \\ &\quad +\omega\left(y_n^{m+1,(k+1)} - u_n^{m+1,(k)}\right) \end{aligned}$$

Dabei:

- $\omega = 1$: Gauss-Seidel !
- $\omega > 1$: over-relaxation
- $\omega < 1$: under-relaxation

Konvergenz ? Ja, für $0 < \omega < 2$.

$\exists$ optimales ω, abhängig vom Spektralradius der Matrix

Die Crank-Nicholson-Methode

Diffusionsgleichung mit Vorwärtsableitungen:

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

mit Rückwärtsableitungen:

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

Jeweils mit $\frac{1}{2}$ gewichten und addieren $\rightsquigarrow$

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

$$= \underbrace{u_n^m + \frac{1}{2}\alpha\left(u_{n-1}^m - 2u_n^m + u_{n+1}^m\right)}_{Z_n^m}$$

Führt auf:

$$\begin{pmatrix} 1+\alpha & -\alpha & 0 & \cdots & 0 \\ -\frac{1}{2}\alpha & 1+\alpha & -\frac{1}{2}\alpha & \cdots & 0 \\ 0 & -\frac{1}{2}\alpha & & & \vdots \\ \vdots & & & & -\frac{1}{2}\alpha \\ 0 & 0 & & -\frac{1}{2}\alpha & 1+\alpha \end{pmatrix} \begin{pmatrix} u_{N^-+1}^{m+1} \\ \vdots \\ u_0^{m+1} \\ \vdots \\ u_{N^+-1}^{m+1} \end{pmatrix}$$

$$= \begin{pmatrix} Z_{N^-+1}^m \\ \vdots \\ Z_0^m \\ \vdots \\ Z_{N^+-1}^m \end{pmatrix} + \frac{1}{2}\alpha \begin{pmatrix} u_{N^-}^{m+1} \\ 0 \\ \vdots \\ 0 \\ u_{N^+}^{m+1} \end{pmatrix} =: b^m$$

Fazit: genau wie o.g. implizites Verfahren, lediglich $\alpha \longrightarrow \frac{1}{2}\alpha$

Algorithmus:

- b^m aus bekannten Größen berechnen
- LGS lösen via LR oder SOR

Vorteil: Genauigkeit !

Wiederholung: Optionen

europäische Call-Optionen:
Recht, zu festem Zeitpunkt T Aktie zum Preis E zu kaufen

Black-Scholes-Differentialgleichung

$$\frac{\partial P}{\partial t} + rS\frac{\partial P}{\partial S} + \frac{1}{2}\sigma_0^2 S^2 \frac{\partial^2 P}{\partial S^2} - rP = 0$$

Anfangsbedingung:

$$G(S,T) = \max(S(T) - E, 0)$$

amerikanische Put-Optionen
Recht, bis zum Zeitpunkt T Aktie zum Preis P zu verkaufen
$\rightsquigarrow$ zusätzliche freie Randbedingung:

$S(t) < S_f(t)$: Einlösen
$S(t) > S_f(t)$: Option behalten

Problem: $S_f(t)$ unbekannt

betrachte 2 Portfolios:

- A: eine Put-Option und eine Aktie
- B: bond, der bei T den Wert E hat

Annahme: Option bei $t < T$ eingelöst $\rightsquigarrow$

Wert des Portfolios A: E
Wert des Portfolios B: $Ee^{-r(T-t)}$

Annahme: Option bis zum Ende gehalten $\rightsquigarrow$

Wert des Portfolios A: $max\{E, S_T\}$
Wert des Portfolios B: E

In jedem Fall

$$P(t) + S_t \;\geq\; Ee^{-r(T-t)}$$

$$\Rightarrow P(t) \;\geq\; E - S_t$$

$\rightsquigarrow$ Wert der amerikanischen Put-Option $P(t)$ ist stets mindestens so groß wie der Pay-Off zum Zeitpunkt $E - S_t$

Außerdem:

Black-Scholes-Gleichung folgt aus Arbitrage-argumenten, hier nur noch Ungleichung möglich:

Black-Scholes-DifferentialUNgleichung

$$\frac{\partial P}{\partial t} + rS\frac{\partial P}{\partial S} + \frac{1}{2}\sigma_0^2 S^2 \frac{\partial^2 P}{\partial S^2} - rP \leq 0$$

1.Fall: $S(t) < S_f(t)$ **Einlösen optimal**

$$P = E - S$$

und

$$\frac{\partial P}{\partial t} + rS\frac{\partial P}{\partial S} + \frac{1}{2}\sigma_0^2 S^2 \frac{\partial^2 P}{\partial S^2} - rP < 0$$

2.Fall: $S(t) > S_f(t)$ **Halten optimal**

$$P > E - S$$

und

$$\frac{\partial P}{\partial t} + rS\frac{\partial P}{\partial S} + \frac{1}{2}\sigma_0^2 S^2 \frac{\partial^2 P}{\partial S^2} - rP = 0$$

Alles zusammen ergibt:

Linear Complementarity Problem:

$$\left(\frac{\partial u}{\partial \tau} - \frac{\partial^2 u}{\partial x^2}\right)(u(x,\tau) - g(x,\tau)) = 0$$

$$\left(\frac{\partial u}{\partial \tau} - \frac{\partial^2 u}{\partial x^2}\right) \geq 0$$

$$(u(x,\tau) - g(x,\tau)) \geq 0$$

mit Anfangs- und Randbedingungen:

$$\begin{aligned} u(x,0) &= g(x,0) \\ u(-x^-,\tau) &= g(-x^-,\tau) \\ u(x^+,\tau) &= g(x^+,\tau) = 0 \end{aligned}$$

Vorteil:

freie Randbedingung $S_f(t)$ taucht nicht explizit auf

$S_f(t)$ kann im Nachhinein aus numerischer Lösung bestimmt werden

Numerische Methoden für Amerikanische Optionen

hier nur: **Crank-Nicholson-Verfahren**

Black-Scholes-Ungleichung $\rightsquigarrow$

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

$$\geq \underbrace{u_n^m + \frac{1}{2}\alpha\left(u_{n-1}^m - 2u_n^m + u_{n+1}^m\right)}_{Z_n^m}$$

Rand- und Anfangsbedingungen

$$\begin{array}{rcl} u_n^m & \geq & g_n^m \\ u_{N^-}^m & = & g_{N^-}^m \\ u_{N^+}^m & = & g_{N^+}^m \\ u_n^0 & = & g_n^0 \end{array}$$

diskretisiertes Komplementaritätsproblem:

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

In Matrixschreibweise:

$$\begin{aligned} Cu^{m+1} &\geq b^m \\ u^{m+1} &\geq g^{m+1} \\ (u^{m+1} - g^{m+1})(Cu^{m+1} - b^m) &= 0 \end{aligned}$$

$\rightsquigarrow$ **projected SOR**

SOR:

$$
\begin{aligned}
& y_n^{m+1,(k+1)} \\
= \; & \frac{1}{1+2\alpha}\left(b_n^m + \alpha\left(u_{n-1}^{m+1,(k+1)} + u_{n+1}^{m+1,(k)}\right)\right)
\end{aligned}
$$

und

$$
\begin{aligned}
u_n^{m+1,(k+1)} &= u_n^{m+1,(k)} \\
&\quad +\omega\left(y_n^{m+1,(k+1)} - u_n^{m+1,(k)}\right)
\end{aligned}
$$

liefert:

$$Cu^{m+1} = b^m$$

<u>hier:</u> constraint $u^{m+1} \geq g^{m+1} \rightsquigarrow$

$$
\begin{aligned}
u_n^{m+1,(k+1)} &= max\Big\{g_n^{m+1}, u_n^{m+1,(k)} + \\
&\quad +\omega\Big(y_n^{m+1,(k+1)} - u_n^{m+1,(k)}\Big)\Big\}
\end{aligned}
$$

Lösung erfüllt:

$$u_n^{m+1} = g_n^{m+1} \qquad \text{oder} \qquad Cu^{m+1} = b^m$$

$$u^{m+1} \geq g^{m+1}$$

$$Cu^{m+1} \geq b^m$$

Letzteres ist nicht offensichtlich, folgt aber aus Eigenschaften der beteiligten Matrix (positiv definit).

Zusammenfassung: Algorithmus

- gegeben u^m, berechne b^m
- rate ein $u^{m+1,(0)} = max\{u^m, g^{m+1}\}$
- berechne $y_n^{m+1,(k+1)}$
- setze
  $u^{(k+1)} \sim max\{g, u^{(k)} + \omega(y^{(k+1)} - u^{(k)})\}$
- überprüfe Konvergenzbedingung
  $\|u^{m+1,(k+1)} - u^{m+1,(k)}\| \leq \epsilon$
- von vorn oder $u^{m+1} := u^{m+1,(k+1)}$