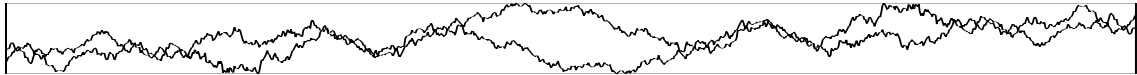




Programmierpraktikum Computational Finance Blatt 3



1 Einführung

Auf diesem Blatt wollen wir uns nun numerischen Quadraturverfahren zur Berechnung von Optionspreisen über den Martingal-Ansatz widmen. Im Gegensatz zu statistischen Verfahren, wie dem Monte-Carlo Verfahren, erlauben numerische Quadraturverfahren deterministische Fehlerschranken und im Falle von glatten Integranden höhere Konvergenzraten. Nachdem bei der Optionspreisbestimmung die Integranden wegen des Knicks in der Auszahlungsfunktion selbst nicht glatt sind, müssen hier Glattheitserhaltende Transformationen, wie bei Formel (25) auf Blatt 2 angewandt werden.

Allerdings hat man bei klassischen numerischen Verfahren in der Regel mit dem sogenannten Fluch der Dimension zu kämpfen. Das bedeutet, daß zum Beispiel bei einem Produktansatz der Aufwand (die Zahl der Funktionsauswertungen) um eine Genauigkeit ε zu erreichen, exponentiell mit der Dimension d ansteigt. Bei Optionen, die von einer endlichen Zahl von Zeitschritten abhängen (wie z.B. bei Asiatischen Optionen mit diskretem Mittel), entspricht die Dimension d der Zahl der Zeitschritte. Schon für wenige Zeitpunkte wird die Konvergenzrate dieser Verfahren geringer als bei Monte-Carlo und Quasi-Monte Carlo Verfahren. Je mehr Zeitschritte verwendet werden (z.B. um Konvergenz gegen das kontinuierliche Mittel zu erhalten), desto höher wird die Dimension und umso schlechter wird dabei die Konvergenz. Nichtsdestotrotz ist der klassische Produktansatz für niederdimensionale Probleme ($d = 1$ bis 4) gut geeignet.

Es gibt aber auch ein numerisches Verfahren, das diesen Fluch der Dimension, zumindest zu einem gewissen Grad, brechen kann, das sogenannte Dünngitter-Verfahren, das auf den russischen Mathematiker Smolyak (1963) zurückgeht. Dieses Verfahren basiert prinzipiell auch auf dem Produktansatz, jedoch werden zur Konstruktion von mehrdimensionalen Quadraturformeln nicht alle Produkt-Kombinationen verwendet, sondern nur ausgewählte. Auf diese Weise wird die Konvergenzrate nahezu unabhängig von der Dimension, wobei gleichzeitig die Glattheit des Integranden genutzt werden kann.

Im folgenden werden wir nun das Produkt-Verfahren und das Dünngitterverfahren genauer betrachten. Beide Verfahren können dabei auf verschiedenen eindimensionalen Quadraturformeln basieren. Die dabei entstehenden mehrdimensionalen Quadraturmethoden sollen dann zur Bewertung von Barrier und Lookback Optionen eingesetzt werden.

2 Eindimensionale Quadraturverfahren

Zunächst erinnern wir uns an die Definition einer Quadraturformel (Formel (11) von Blatt 2)

$$\int_0^1 f(x) dx \approx \sum_{i=1}^N w_i f(x_i). \quad (1)$$

Nun wollen wir einer Quadraturformel

$$Q_l^{(1)} f = \sum_{i=1}^{N_l} w_{li} f(x_{li}) \quad (2)$$

einen Level $l = 1, 2, \dots$ zuordnen, und die Zahl der Punkte der Quadraturformeln N_l von dem Level abhängig machen. Dabei soll sich die Zahl der Punkte von Level zu Level etwa verdoppeln (wie in den Aufgaben von Blatt 2), d.h.

$$N_l = O(2^l). \quad (3)$$

Da wir später mehrdimensionale Quadraturformeln betrachten, ist bei der Quadraturformel Q hochgestellt in Klammern die Dimension des Problems (hier 1) angegeben.

Bei der Auswahl geeigneter Quadraturformeln ist jedoch Vorsicht geboten. Zum Beispiel werden bei Newton–Cotes Formeln, die äquidistante Stützstellen verwenden, ab $N = 8$ die Gewichte negativ und damit die Quadraturformeln numerisch instabil. Weiterhin werden durch die Anwendung der inversen Normalverteilung (siehe Formel (9) von Blatt 2) die Punkte 0 und 1 auf $-\infty$ und $+\infty$ abgebildet, wodurch zumindest bei $+\infty$ kein vernünftiger Wertpapierkurs bestimmt werden kann.

Daher wollen wir hier zwei sogenannte offene Quadraturformeln, welche Auswertungen am Rand des Integrationsgebiets vermeiden, die offene Trapezsumme und das offene Clenshaw–Curtis Verfahren einsetzen.

2.1 Trapezsumme

Die geschlossene Trapezsumme (die zusammengesetzte Trapezregel) verwendet äquidistante Stützstellen und identische Gewichte. Bei der offenen Trapezsumme werden die beiden äußersten Punkte bei 0 und 1 einfach weggelassen und stattdessen die beiden zweitäußersten Punkte mit einem Faktor $\frac{3}{2}$ gewichtet. Wir wollen nun

$$N_l = 2^l - 1 \quad (4)$$

wählen. Damit läßt sich die Trapezsumme schreiben als

$$Q_l^{(1)} f = \frac{1}{N_l + 1} \left(\frac{3}{2} \cdot f \left(\frac{1}{N_l + 1} \right) + \sum_{i=2}^{N_l-1} f \left(\frac{i}{N_l + 1} \right) + \frac{3}{2} \cdot f \left(\frac{N_l}{N_l + 1} \right) \right). \quad (5)$$

Die Konvergenzrate der Trapezsumme ist für einen Integrand $f \in C^2$

$$\varepsilon(N_l) = O(N_l^{-2}) \quad (6)$$

und damit um eine Ordnung besser als die des Quasi–Monte Carlo Verfahrens. Statt der Trapezsumme kann natürlich auch die Simpsonsumme oder höhere Newton–Cotes Summen mit entsprechenden Randmodifikationen (und mit entsprechend höheren Konvergenzraten) eingesetzt werden.

2.2 Clenshaw–Curtis

Es ist bekannt, daß Gauß–Quadraturformeln den höchsten polynomialen Exaktheitsgrad bei der numerischen Integration erzielen. Die Stützstellen und Gewichte der Gauß–Formeln sind jedoch aufwendig zu berechnen. Eine kostengünstigere Variante sind die sogenannten Clenshaw–Curtis Formeln, deren Stützstellen die Nullstellen oder die Extremwerte der Tschebyscheff–Polynome sind. Wir wollen hier die Extremwerte verwenden, da auf diese Weise eine Reihe geschachtelter Quadraturformeln entsteht, bei denen die Stützstellen eines Levels eine Teilmenge der Stützstellen des nächsten Levels sind.

Wir wählen nun N_l wie bei der offenen Trapezsumme (siehe (4)). Damit sind die Stützstellen der offenen Clenshaw–Curtis Formeln (auch Filippi Formeln genannt) gegeben als

$$x_{li} = \frac{1}{2} \left(1 - \cos \frac{\pi i}{N_l + 1} \right) \quad (7)$$

und die Gewichte als

$$w_{li} = \frac{2}{N_l + 1} \sin \frac{\pi i}{N_l + 1} \sum_{j=1}^{(N_l+1)/2} \frac{1}{2j-1} \sin \frac{(2j-1)\pi i}{N_l + 1}. \quad (8)$$

Der Exaktheitsgrad der Clenshaw–Curtis Formeln ist nur N_l im Vergleich zu $2N_l - 1$ bei den entsprechenden Gauß Formeln, was aber erheblich besser als 1 wie bei der Trapezsumme ist. Dadurch werden die Quadraturformeln aber interpolatorisch und somit hängt die Konvergenzrate bei den Clenshaw–Curtis Formeln auch von höherer Glattheit des Integranden ab und ist für $f \in C^r$

$$\varepsilon(N_l) = O(N_l^{-r}). \quad (9)$$

Bei nur stetigen Funktionen ist die Konvergenzrate immerhin 1, bei unendlich glatten Funktionen ist die Konvergenz aber exponentiell, also erheblich besser als für alle bisher besprochenen Verfahren.

3 Mehrdimensionale Quadraturverfahren

Nun wollen wir aus den betrachteten eindimensionalen Quadraturformeln mehrdimensionale Quadraturformeln konstruieren. Mehrdimensionale Integrale treten, wie auf dem letzten Blatt gesehen, zum Beispiel bei pfadabhängigen Optionen, wie den Asiatischen Optionen, auf. Zwei weitere pfadabhängigen Optionstypen werden wir im nächsten Kapitel kennenlernen.

Zur Konstruktion mehrdimensionaler Quadraturformeln wollen wir nun den Produktansatz sowie den Dünngitter-Ansatz von der Theorie hier und im Hinblick auf die Umsetzung auf dem Rechner betrachten.

3.1 Produktformeln

Bei dem Produktansatz wird einfach das Tensorprodukt von eindimensionalen Quadraturformeln mit jeweils gleichem Level l zur Konstruktion mehrdimensionaler Quadraturformeln verwendet, d.h.

$$Q_l^{(d)} f = (Q_l^{(1)} \otimes \dots \otimes Q_l^{(1)}) f. \tag{10}$$

Das Tensorprodukt von d Quadraturformeln $(Q_{l_1}^{(1)} \otimes \dots \otimes Q_{l_d}^{(1)}) f$ (hier mit unterschiedlichen Leveln l_1 bis l_d , da diese allgemeinere Version gleich benötigt wird), ist dabei definiert als die Summe über alle Kombinationen, d.h.

$$(Q_{l_1}^{(1)} \otimes \dots \otimes Q_{l_d}^{(1)}) f = \sum_{i_1=1}^{N_{l_1}} \dots \sum_{i_d=1}^{N_{l_d}} w_{l_1 i_1} \cdot \dots \cdot w_{l_d i_d} \cdot f(x_{l_1 i_1}, \dots, x_{l_d i_d}). \tag{11}$$

Die Funktion wird also an einem regelmäßigen Produktgitter ausgewertet, wobei die entstehenden mehrdimensionalen Gewichte die Produkte der entsprechenden eindimensionalen Gewichte sind.

Bei der Implementierung stößt man an dieser Stelle auf eine unerwartete Schwierigkeit. Nachdem die Dimension d ein variabler Parameter ist, kann die Zahl der Summen (und damit die Zahl der benötigten Schleifen) nicht a priori angegeben werden. Hier kommen dann sogenannte Tröpfchen-Algorithmus zum Einsatz, wie sie auch zum Aufzählen von (z.B. Binär-) Zahlen verwendet werden:

```

für  $j = 1 \dots d$ 
  setze  $i_j = 1$ 
  setze  $p = 1$ 
  solange  $i_d \leq N_{l_d}$ 
    wenn  $i_p > N_{l_p}$ 
      setze  $i_p = 1$ 
      setze  $p = p + 1$ 
    sonst
      werte Funktion am Punkt  $x_{l_1 i_1}, \dots, x_{l_d i_d}$  aus
      multipliziere Funktionswert mit Gewicht  $w_{l_1 i_1} \dots w_{l_d i_d}$ 
      setze  $p = 1$ 
      setze  $i_p = i_p + 1$ 

```

Algorithmus 8: Mehrdimensionale Integration mit dem Produktansatz.

Wenn nun als eindimensionale Quadraturformel die Clenshaw-Curtis Formel verwendet wird, dann gilt für Integranden aus C^r (d.h. mit Funktionen mit beschränkten totalen Ableitungen bis zum Grad r) die Konvergenzordnung

$$\varepsilon(N) = O(N^{-r/d}). \tag{12}$$

Hierbei ist $N = (N_l)^d$ die Gesamtzahl der Funktionsauswertungen. An dieser Stelle sieht man den Fluch der Dimension deutlich in der Konvergenzrate. Je höher d wird, desto geringer wird die Konvergenz.

3.2 Dünngitter-Formeln

Bei dem sogenannten Dünngitter-Ansatz tritt diese Abhängigkeit von der Dimension in nur wesentlich geringerem Ausmaß auf. Die zugrundeliegende Idee hierbei ist eine Zerlegung der Quadraturformel in eine Teleskopsumme.

Dann wird auf die Teleskopsummenglieder wieder ein Produktansatz angewandt, aber von allen entstehenden Kombinationen wird nur ein Teil ausgewählt. Diese Auswahl geschieht dadurch, daß Aufwand und Genauigkeit balanciert werden.

Zunächst startet man mit einer Teleskopsumme, also man definiert sich Differenzen

$$\Delta_k^{(1)} f = (Q_k^{(1)} - Q_{k-1}^{(1)}) f \quad \text{wobei} \quad (13)$$

$$Q_0^{(1)} f = 0. \quad (14)$$

Hierbei sind die Differenzen $\Delta_k^{(1)} f$ wieder (eindimensionale) Quadraturformeln. Für den Fall, daß die Ausgangsformeln $Q_k^{(1)} f$ geschachtelt sind (wie für die Trapezsumme und die Clenshaw–Curtis Formeln der Fall), dann verwenden die Differenzformeln die gleichen Gitterpunkte, nur mit anderen Gewichten. Die neuen Gewichte sind dabei nichts anderes als die Differenzen der Gewichte der Ausgangsformeln zwischen zwei aufeinanderfolgenden Levels.

Nun hat die Dünngitter–Konstruktion folgende Darstellung

$$Q_l^{(d)} f = \sum_{k_1 + \dots + k_d \leq l + d - 1} (\Delta_{k_1}^{(1)} \otimes \dots \otimes \Delta_{k_d}^{(1)}) f. \quad (15)$$

Es werden also alle möglichen Tensorprodukte der Differenzformeln betrachtet. Von diesen Möglichkeiten werden jedoch nur diejenigen verwendet, deren Summe der Einzelindizes kleiner als eine Konstante (hier $l + d - 1$) ist. Der Produktansatz vom vorigen Abschnitt ist übrigens dadurch charakterisiert, daß alle gültigen Indizes verwendet werden, also

$$Q_l^{(d)} f = \sum_{\max\{k_1, \dots, k_d\} \leq l} (\Delta_{k_1}^{(1)} \otimes \dots \otimes \Delta_{k_d}^{(1)}) f. \quad (16)$$

Bildlich gesprochen entspricht der Produktansatz einer Summation über den Hyperwürfel mit Kantenlänge l , also $\max\{k_1, \dots, k_d\} \leq l$ während dem Dünngitter–Verfahren der Simplex mit Kantenlänge l , also $k_1 + \dots + k_d \leq l + d - 1$ zugrunde liegt. In der folgenden Abbildung wird die Konstruktion an einem zweidimensionalen Beispiel gezeigt.

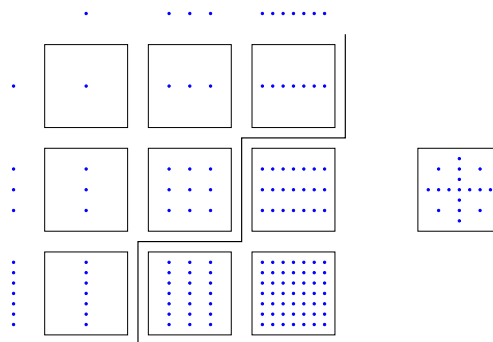


Abbildung 1: Links sieht man die Gitterpunkte der Trapezsumme für $l = 1, 2, 3$ in x - und y -Richtung sowie die entstehenden Produktgitter $\Delta_{k_1} \otimes \Delta_{k_2}$ für $1 \leq k_1, k_2 \leq 3$. Rechts ist das entsprechende dünne Gitter $Q_3^{(2)}$ abgebildet.

In der folgenden Abbildung sieht man ein Produkt- und ein dünnes Gitter im Vergleich.

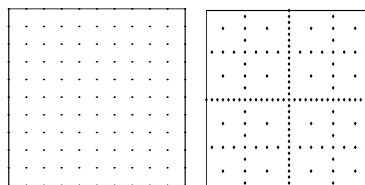


Abbildung 2: Produktgitter und entsprechendes dünnes Gitter für $l = 4$

Während die Programmierung der einzelnen Tensorprodukte in Formel (15) in genau der gleichen Weise wie für den Produktansatz (siehe Algorithmus 8) durchgeführt werden kann, bereitet die Programmierung der Summe

über den Simplex auf den ersten Blick für allgemeines d Schwierigkeiten. Dieses Problem kann jedoch durch einen analogen Tröpfchenalgorithmus gelöst werden, hierzu verwendet man einen zusätzlichen Vektor \hat{k} , welcher die momentan gültigen maximalen Werte enthält und der mit der Zeit aktualisiert wird:

```

für  $j = 1 \dots d$ 
  setze  $k_j = 1$ 
  setze  $\hat{k}_j = l$ 
setze  $q = 1$ 
setze  $k_1 = l$ 
solange  $k_d \leq l$ 
  wenn  $k_q > \hat{k}_q$ 
    setze  $k_q = 1$ 
    setze  $q = q + 1$ 
  sonst
    für  $r = 1 \dots q - 1$ 
      setze  $\hat{k}_r = \hat{k}_q - k_q + 1$ 
    setze  $k_1 = \hat{k}_1$ 
    rufe Algorithmus 8 mit  $l_1 = k_1, \dots, l_d = k_d$  auf
    setze  $q = 1$ 
  setze  $k_q = k_q + 1$ 

```

Algorithmus 9: Mehrdimensionale Integration mit dem Dünngitter-Verfahren.

Für die richtige Zusammensetzung der verwendeten Punkte sorgt die Kombinationstechnik. Dieser Algorithmus liefert dann auch den Optionspreis mit dem Dünngitter-Verfahren.

```

setze  $V = 0$ 
für  $i = 1 \dots l$ 
  setze  $|k| = i + d - 1$ 
  falls  $|k| \geq l$  und  $|k| \leq l + d - 1$ 
    rufe Algorithmus 9 mit  $l = i$  und Dimension  $d$  auf
    setze das Ergebnis  $V^*$  in
     $V = V + (-1)^{l+d-|k|-1} \cdot \binom{d-1}{|k|-l} \cdot V^*$  ein

```

Algorithmus 10: Kombinationstechnik.

Den Vorteil des Dünngitterverfahrens sieht man an der Konvergenzrate, welche für Funktionen mit beschränkten gemischten Ableitungen bis zum Grad r

$$\varepsilon(N) = O(N^{-r}(\log N)^{(d-1)(r+1)}) \quad (17)$$

ist. Dabei ist die Gesamtzahl der Punkte nur $N = O(N_l \log N_l^{d-1})$. In der Zahl der Punkte und in der Konvergenzrate tritt also die Dimension d nur noch in einem logarithmischen Faktor auf. Auf diese Weise wird der Fluch der Dimension (zumindest zu einem gewissen Grad) gebrochen und auch hochdimensionale Integrationsprobleme können so unter Ausnutzung der Glattheit r des Integranden gelöst werden.

4 Pfadabhängige Optionen

Wir wollen nun zwei weitere Beispiele für pfadabhängige Optionen betrachten, die stärker als Asiatische Optionen auch an Aktienmärkten (in Deutschland über Zertifikate) gehandelt werden, sogenannte Barrier und Lookback Optionen.

4.1 Barrier Optionen

Barrier Optionen sind Optionen bei denen, sobald das zugrundeliegende Wertpapier eine Schranke (Barrier) durchschreitet, die Option wertlos wird bzw. erst einen Wert erlangt. Bei einer Knock-Out Option wird die Option

wertlos, sobald die Schranke übertritt (Up-Out) oder unterschreitet (Down-Out). Knock-In Optionen sind solange wertlos bis die Schranke übertreten (Up-In) oder unterschritten (Down-In) wird. In allen vier Fällen gibt es entsprechende Call und Put Optionen. Barrier Optionen werden gerne gehandelt, da durch das zusätzliche Riskiko die Optionsscheine noch billiger werden (Beispiele sind sogenannte Turbo-Optionsscheine).

Wir wollen nun die Auszahlungsfunktion einer Down-Out Call Option mit Barrier H betrachten. Sie lautet einfach

$$V(S, T) = \begin{cases} (S(T) - K)^+ & \text{falls } S(t) > H \text{ für } 0 \leq t \leq T \\ 0 & \text{sonst} \end{cases} \quad (18)$$

Für den Preis von Barrier Optionen lassen sich entsprechende Black-Scholes Formeln angeben. Für die Down-Out Call Option lautet sie

$$V(S, T) = V_{bs}(S, \bar{H}) - Z \cdot V_{bs}(H^2/S, \bar{H}) + (\bar{H} - K)e^{-rT}(N(d(S, \bar{H})) - Z \cdot N(d(H^2/S, \bar{H}))) \quad (19)$$

mit

$$Z = \left(\frac{H}{S}\right)^{\frac{2r}{\sigma^2} - \frac{1}{2}} \quad (20)$$

und

$$\bar{H} = \max\{H, K\}. \quad (21)$$

Hierbei ist $V_{bs}(S, K)$ der Black-Scholes Preis für eine Europäische Call Option zum Kurs S und Ausübungspreis K , also wie in Formel (11)–(13) auf Blatt 1

$$V_{bs}(S, K) = SN(d(S, K) + \sigma\sqrt{T}) - Ke^{-rT}N(d(S, K)) \quad (22)$$

mit entsprechend

$$d(S, K) = \frac{\ln \frac{S}{K} + (r - \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}. \quad (23)$$

Allerdings läßt sich für komplexer gebaute Barrier Optionen (z.B. für Amerikanische Barrier Optionen oder gleitende Barriers) keine geschlossene Lösung mehr angeben.

4.2 Lookback Optionen

Bei Lookback Optionen liegt die Überlegung zugrunde, wie schon bei Asiatischen Optionen, daß zu plötzliche Schwankungen gegen Ende der Laufzeit den Optionspreis extrem stark beeinflussen können. Bei Asiatischen Optionen wird daher das zeitliche Mittel des Wertpapierkurses statt des Wertpapierkurses zum Ausübungszeitpunkt verwendet. Bei Lookback Optionen wird das Maximum bzw. Minimum der Kurse betrachtet, was für den Anleger den Vorteil hat, daß so der maximale Gewinn bezüglich des Verlauf des Wertpapiers ausgeschöpft werden kann. Der Nachteil ist, daß diese Optionen dadurch relativ teuer im Vergleich zu anderen Optionstypen werden.

Man unterscheidet Lookback Optionen mit festem und variablem Ausübungspreis. Bei festem Ausübungspreis lautet die Auszahlungsfunktion

$$V(S, T) = \left(\max_{0 \leq t \leq T} S(t) - K\right)^+ \quad (24)$$

während bei variablen Ausübungspreis die Auszahlungsfunktion als

$$V(S, T) = \left(S(T) - \min_{0 \leq t \leq T} S(t)\right)^+ \quad (25)$$

gestaltet ist. In beiden Fällen existieren Black-Scholes Formeln, welche lauten

$$V(S, T) = e^{-rT}(S - K) + V_{bs}(S, S) + \frac{S\sigma^2}{2r} \left(N(d_{bs}(S, S) + \sigma\sqrt{T}) - e^{-rT}N(-d_{bs}(S, S))\right) \quad (26)$$

für Lookback Optionen mit festem Ausübungspreis und

$$V(S, T) = V_{bs}(S, S) + \frac{S\sigma^2}{2r} \left(e^{-rT}N(d_{bs}(S, S) + \sigma\sqrt{T}) - N(-d_{bs}(S, S))\right) \quad (27)$$

für variablen Ausübungspreis. Aufgrund der Ähnlichkeit der Bewertungsformeln sieht man schon, daß kein allzu-großer Unterschied in der Charakteristik von Lookback Optionen mit festem und variablen Ausübungspreis besteht. Deswegen werden wir uns in den Beispielen auf die (populärere) Variante mit festem Ausübungspreis beschränken. Aber in beiden Fällen gilt auch hier, daß für komplexere Strategien oder für Lookback Optionen vom Amerikanischen Typ keine exakte Bewertungsformel mehr möglich ist.

5 Implementierung

Nachdem wir nun mehrere Optionen, die sich vor allem in der Auszahlungsfunktion unterscheiden, mehrere Integrationsverfahren und mehrere Diskretisierungsverfahren kennengelernt haben, sollten diese Elemente (wenn nicht schon geschehen) möglichst modular gehalten werden. Eine Möglichkeit besteht in der Verwendung folgender Struktur:

- `PAYOFF(M, S[M], K, Type, H)`: Die Auszahlungsfunktion für Europäische (`Type=0`), Asiatische (`Type=1`), Down-Out Barrier (`Type=2`), sowie Lookback Optionen mit festem Ausübungspreis (`Type=3`). Bei der Eingabe wird `H` nur im Fall einer Barrier Option benötigt.
- `DISCRETIZE(M, S0, sigma, r, x[M], Walk)`: Die Diskretisierung des Wiener Prozesses mit Random Walk (`Walk=1`) und Brownscher Brücke (`Walk=0`). Hierbei werden die zu verwendenden Koordinaten in dem Feld `x[M]` übergeben und die simulierten Wertpapierkurse in einem Feld der Größe `M` zurückgegeben.
- `INTEGRATE(N, M, S0, K, T, sigma, r, Int, Walk, Type, H)`: Diese Routine ruft die entsprechenden Integrationsverfahren auf, also Monte Carlo (`Int=0`), Quasi-Monte Carlo (`Int=1`), den Dünngitteransatz mit Trapezsumme (`Int=2`), sowie den Dünngitteransatz mit Clenshaw-Curtis (`Int=3`).

Nachdem Sie Ihre Programme entsprechend umgestellt haben, fehlen natürlich noch die auf diesem Blatt neu zu programmierenden Funktionen.

- `TRAPEZSUMME(1)`: Liefert die Stützstellen und Gewichte der eindimensionalen Trapezsumme, nach Formel (4)-(6).
- `CLENSHAW_CURTIS(1)`: Berechnet entsprechend die Stützstellen und Gewichte der eindimensionalen Clenshaw-Curtis Formel nach (7)-(8).
- `DUENN(1, M, S0, K, T, sigma, r, Int, Walk, Type, H)`: Berechnet den jeweiligen Optionspreis mit dem Dünngitter-Verfahren. Es wird dazu Algorithmus 10 verwendet. Dieser ruft dann Algorithmus 9 auf, welcher wiederum Algorithmus 8 verwendet.

Entsprechend müssen die `MONTE_CARLO` und `QUASI_MONTE_CARLO` Routinen abgeändert werden, um in der Funktion `INTEGRATE` verwendet werden zu können. Nun fehlen nur noch die beiden neuen Optionstypen Barrier (`Type=2`) und Lookback Optionen (`Type=3`) in der Auszahlungsfunktion `PAYOFF`.

6 Beispiele

Alle Aufgabenlösungen müssen wieder schriftlich in elektronischer Form abgegeben werden. Sie können die Lösung wieder per e-mail einschicken (noll@math.uni-frankfurt.de).

Führen Sie mit den Programmen folgende Beispielrechnungen durch:

- Berechnen Sie den Optionspreis für eine Barrier Option mit $S_0 = 24$, $K = 24$, $\sigma = 0.2$, $r = 0.02$, $T = 1$ und $H = 30$ für $M = 4$ Zeitschritte mit den vier verschiedenen Integrationsverfahren für $l = 1 \dots 8$. Für die Monte-Carlo-artigen Verfahren wählen Sie dabei $N_l = 512 \cdot 2^l$. Bestimmen Sie jeweils die Konvergenzrate und plotten ihre Ergebnisse.
- Berechnen Sie mit den vier verschiedenen Integrationsverfahren die Optionspreise für eine Lookback Option mit festem Ausübungspreis für $M = 4$ Zeitschritte mit den gleichen Parametern wie für obige Barrier Option. Bestimmen Sie auch hier jeweils die Konvergenzrate und plotten ihre Ergebnisse.
- Berechnen Sie mit den vier verschiedenen Integrationsverfahren ebenfalls die Optionspreise für eine europäische Option mit den gleichen Parametern wie zuvor. Bestimmen Sie auch hier jeweils die Konvergenzrate, plotten Sie ihre Ergebnisse und interpretieren Sie diese im Vergleich zu den Aufgabenteilen a) und b).

Literatur

- [1] C. Clenshaw, A. Curtis. A method for numerical integration on an automatic computer, *Numerische Mathematik*, 2:197–205, 1960.
- [2] P. Davis, P. Rabinowitz. *Methods of Numerical Integration*, Academic Press, 1975.
- [3] T. Gerstner, M. Griebel. Numerical Integration using Sparse Grids, *Numerical Algorithms*, 18:209–232, 1998.
- [4] E. Novak, K. Ritter. High dimensional integration of smooth functions over cubes, *Numerische Mathematik*, 75:79–98, 1996.
- [5] P. Zhang. *Exotic Options*, 2nd Edition, World Scientific, 1998.