Χρηματοοικονομικά Παράγωγα

Βοντικάκης Βασίλειος

(Σημειώσεις)

ΕΙΣΑΓΩΓΗ ΣΤΑ ΣΤΟΧΑΣΤΙΚΑ ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΑ

Το ξεκίνημα των στοχαστικών χρηματοοικονομικών

Ο πρώτος που ασχολήθηκε με τη μαθηματική χρηματοοικονομική ήταν ο Bachelier το 1900, ο οποίος μελέτησε τις στοχαστικές διαδικασίες πάνω στη χρηματοοικονομική ως θέμα της διδακτορικής του διατριβής, η οποία είχε τίτλο “Theorie de la Speculation”. Ο Bachelier κατάφερε να κατασκευάσει ένα μοντέλο της κίνησης Brown ενώ προσπαθούσε να δημιουργήσει ένα μαθηματικό μοντέλο για τη μεταβολής της τιμής ενός χρηματοοικονομικού προϊόντος, πιο συγκεκριμένα προσπαθούσε να βρει τις πιθανότητες για παράγωγα options και futures που είχαν ως υποκείμενο τίτλο ένα γαλλικό κρατικό ομόλογο. Η εργασία του ήταν πολύ σημαντική τόσο για τα χρηματοοικονομικά όσο και για τη θεωρία πιθανοτήτων. Το μέρος της εργασίας του που αφορούσε τα χρηματοοικονομικά χάθηκε και εμφανίστηκε μετά από 64 χρόνια όταν ο Paul Samuelson ένας οικονομολόγος δημοσίευσε το 1964 δυο εργασίες για το πως μοντελοποιούνται οι τιμές των μετοχών και αυτό έχοντας λάβει υπόψην τον Bachelier.

Επόμενο σημαντικό βήμα στις στοχαστικές διαδικασίες οι οποίες είναι αλληλένδετες με τα χρηματοοικονομικά ήταν το 1923 όπου ο N. Wienner απέδειξε δυο σημαντικές ιδιότητες της κίνησης Brown που την συσχετίζουν με τη στοχαστική ολοκλήρωση. Η πρώτη σημαντική ιδιότητα είναι ότι οι τροχιές της κίνησης Brown έχουν μη μηδενική πεπερασμένη τετραγωνική μεταβολή έτσι ώστε σε ένα χρονικό (s,t) η τετραγωνική μεταβολή είναι (t-s). Η δεύτερη σημαντική ιδιότητα είναι ότι οι τροχιές της έχουν άπειρη μεταβολή σε συμπαγή διάστημα χρόνου. Ο ορισμός της κίνησης Brown είναι

ορισμός. Η κίνηση Brown είναι μια στοχαστική διαδικασία B_t , $t\in[0,\infty)$

με τις παρακάτω ιδιότητες

1) $B_0=0$

2) Οι προσαυξήσεις των τ.μ B_t είναι ανεξάρτητες στο χρονικό διάστημα

$0=t_0<t_1<\ldots<t_{n-1}<t_n=T$ δηλαδή οι $B_{t1}-B_{t0}, B_{t2}-B_{t1},\ldots$

3)για κάθε $0\leq s\leq t$ ισχύει $B_t-B_s\sim N(0,t-s)$

4) οι τροχιές της είναι συνεχείς, γιατί η $t\rightarrow B_t$ είναι συνεχής συνάρτηση

Αφού δώσαμε τον ορισμό της κίνησης Brown θα αποδείξουμε τη σημαντική ιδιότητα ότι η τετραγωνικής της μεταβολή είναι t-s. Ένας ορισμός για την τετραγωνική μεταβολή είναι ο παρακάτω

ορισμός. Έστω μια συνάρτηση $f:[a,b]\rightarrow\mathbb{R}$ τετραγωνική μεταβολή της συνάρτησης f στο διάστημα $[a,b]$ ορίζεται

$$\langle f\rangle_{a,b}^2=\lim_{\delta_n\rightarrow 0}\sum_{i=1}^{n}|\Delta f(t_i)|^2$$

όπου

$$\delta_n=\sup_i r|t_{i+1}-t_i|$$

στο

$$P_n=(t_{0,}t_{1,\ldots},t_n)=\{a=t_0<t_1<t_2<\ldots<t_n=b\}$$

οπότε η τετραγωνική μεταβολή της κίνησης brown σε ένα χρονικό διάστημα $[a,b]$ είναι $\langle B \rangle^2_{a,b} = a-b$ φαίνεται από το να αποδείξουμε ότι υπάρχει L^2 σύγκλιση δηλαδή ότι ισχύει

$$\lim_{n\to\infty \delta_n \to 0} E\left(\sum_{i=1}^{n} \Delta B_{t_i}^2 - (b-a)\right)^2 = 0$$

$$E\left(\sum_{i=1}^{n} \Delta B_{t_i}^2 - (b-a)\right)^2 = E\left(\sum_{i=1}^{n} (\Delta B_{t_i}^2 - \Delta t_i)\right)^2 = E\left(\sum_{i=1}^{n} (\Delta B_{t_i}^2 - \Delta t_i)^2\right) \overset{(1)}{=} \sum_{i=1}^{n} E(\Delta B_{t_i}^2 - \Delta t_i)^2$$

$$\sum_{i=1}^{n} E(\Delta B_{t_i})^4 - 2\Delta t\, E(\Delta B_{t_i})^2 + \Delta t_i^2 \overset{(2)}{=} \sum_{i=1}^{n} E\, 3(\Delta t_i)^2 - 2\Delta t_i + \Delta t_i^2 \leq 2\mathrm{t}\, max(\Delta t_i) \to 0$$

για $\delta_n \to 0$

(1) Λόγω ανεξαρτησίας της κίνησης Brown

(2) Η forth moment μιας τ.μ. με μέσω 0 και διακύμανση $t_i - t_{i-1}$ είναι $3\Delta t_i^2$

Η δεύτερη σημαντική ιδιότητα της κίνησης Brown είναι η ιδιότητα της μη φραγμένης κύμανσης σε συμπαγές διάστημα χρόνου.

$$\langle f \rangle^1_{a,b} = \sum_{i=1}^{n} |\Delta B_{t_i}| \to \infty \quad \text{σ.β.}$$

$$\sum_{i=1}^{n} |\Delta B_{t_i}|^2 \leq \max_{1\leq i\leq n} |\Delta B_{t_i}| \sum_{i=1}^{n} |\Delta B_{t_i}| \leq V(a,b) \max_{1\leq i\leq n} |\Delta B_{t_i}|$$

$$\max_{1\leq i\leq n} |\Delta B_{t_i}| \to 0 \quad \text{καθώς} \quad \delta_n \to 0$$

από το οποίο προκύπτει

$$\sum_{i=1}^{n} |\Delta B_{t_i}|^2 \to 0 \quad \text{σ.β.}$$

Επειδή η B είναι συνεχής σ.β. στο $[a,b]$ είναι επιπλέον ομοιόμορφα συνεχής στο $[a,b]$

Ενώ το μέρος της εργασία του Bachelier που αφορούσε τα χρηματοοικονομικά είχε αγνοηθεί μέχρι το 1964, στη περιοχή των πιθανοτήτων η εργασία του ήταν γνωστή στον A.N. Kolmogorov όπου έκανε αναφορές στο Bachelier για το πως κατασκεύασε την κίνηση Brown. Ο Kolmogorov κάνει πολύ σημαντικά βήματα στη θεωρία της στοχαστικών διαδικασιών όπου καταφέρνει να συσχετίσει τις κατανομές πιθανότητας στοχαστικών διαδικασιών με τις λύσεις μερικών διαφορικών εξισώσεων οι οποίες ονομάζονται “Εξισώσεις του Kolmogorov”. Η ανάλυση του βασίστηκε στης θεωρία των ημιομάδων (semi-groups) και του απειροστικού γεννήτορα αφού δεν είχε στη διάθεση του το ολοκλήρωμα του Ito.

Το επόμενο βήμα έρχεται από τον Kiyosi Ito όπου κατασκεύασε το στοχαστικό ολοκλήρωμα. Το πρόβλημα του Ito ήταν ότι με την κίνηση Brown ήταν ότι οι τροχιές της είναι μη φραγμένης διακύμανσης σχεδόν βεβαίως σε κάθε πεπερασμένο χρονικό διάστημα ο Ito ήξερε ότι δεν μπορούσε να ολοκληρώσει όλες τις συνεχείς στοχαστικές διαδικασίες. Οπότε αυτό που έκανε ήταν να περιόρισε το χώρο των ολοκληρώσεων σε αυτές τις διαδικασίες που ονομάζονταν non-anticipating, για αυτό το λόγω επιτρέπει ολοκληρώματα στοχαστικών διαδικασιών οι οποίες είναι προσαρμοσμένες σε filtrations δηλαδή ακολουθίες από σ-άλγεβρες οι οποίες δημιουργούνται από την κίνηση Brown

$$F_t = \sigma(B_s, s \le t)$$

Αυτό του επέτρεψε να χρησιμοποιήσει την ανεξαρτησία των αυξήσεων της κίνηση Brown και να καθιερώσει την L^2 ισομετρία. Μια σύντομη περιγραφεί αυτής της θεωρία είναι η παρακάτω, έστω $H^2[0,T]$ είναι το σύνολο συναρτήσεων για τις οποίες ισχύει ο παρακάτω περιορισμός.

$$\|f\|_{H^2} = E\left(\int_0^T |f|^2 \, dt\right) < \infty$$

Στη συνέχεια ορίζουμε το χώρο συναρτήσεων H_0^2 η οποία είναι υποσύνολο της κλάσης H^2 και θα περιλαμβάνει όλες τις συναρτήσεις από την κλάση H^2 οι οποίες μπορούν να γραφτούν με την παρακάτω σαν συναρτήσεις βήματος στο $[0,T]$ το οποίο διαμερίζεται σε $0 = t_0 < t_1 < \ldots \le t_n = T$.

$$f(t) = \sum_{i=0}^{n-1} a_i 1_{t_i < t \le t_{i+1}}$$

όπου οι a_i είναι F_{t_i} μετρήσιμες τυχαίες μεταβλητές δηλαδή $a_i \in F_{t_i}$ και τετραγωνικά ολοκληρώσιμες δηλαδή ισχύει $Ea_i^2 < \infty$. Στη συνέχεια θα ορίσουμε το ολοκλήρωμα των συναρτήσεων βήματος στον H_0^2 . Οπότε για ένα υποσύνολο $(t_1, t_2]$ του $[0,T]$ το ολοκλήρωμα συνάρτησης βήματος γράφεται ως

$$I(f) = \int_{t_1}^{t_2} f \, dB_t = B_{t_1} - B_{t_2}$$

Οπότε λαμβάνοντας υπόψη της προηγούμενη εξίσωση και το γεγονός ότι θέλουμε ολοκληρώσουμε σε όλο το $[0,T]$ προκύπτει ότι

$$I(f) = \int_0^T f \, dB_t = \sum_{i=0}^{n-1} a_i (B_{t_{i+1}} - B_{t_i})$$

Το παραπάνω είναι το στοχαστικό ολοκλήρωμα της συνάρτησης βήματος ισχύει η ισομετρία Ito H^2 .

$$E[|\int_0^T f_t dB_t|^2] = E(\int_0^T |f_t|^2 dt)$$

Στη συνέχεια θα μπορέσουμε να επεκτείνουμε το στοχαστικό ολοκλήρωμα που ορίζεται στις στοιχειώδης συναρτήσεις δηλαδή αυτές που βρίσκονται στο χώρο H_0^2 στον ευρύτερο χώρο συναρτήσεων δηλαδή το H^2 . Αυτό επιτυγχάνεται με το να προσεγγίσουμε μια συνάρτηση f η οποία ανήκει στο χώρο H^2 με μια ακολουθία από στοιχειώδεις συναρτήσεις δηλαδή τη f_n όπου αυτή η ακολουθία ανήκει στο χώρο H_0^2 δηλαδή

$$\|f - f_n\|_{H^2}^2 = (\int_a^b |f - f_n|^2 dt)^{\frac{1}{2}} \to 0 \quad \text{καθώς} \quad n \to \infty$$

Οπότε το ολοκλήρωμα του Ito για την $f \in H^2$ ορίζεται ορίζεται ως το όριο των ολοκληρωμάτων των $f_n \in H_0^2$

$$\int_a^b f(t) dB_t = \lim_{n \to \infty} \int_a^b f_n(t) dB_t$$

Με την ισομετρία Ito στο χώρο H^2 έχουμε κατασκευάσει μία γραμμική απεικόνιση του στοχαστικού ολοκληρώματος στον L^2 δηλαδή $I(f): H^2 \to L^2$. Οπότε το στοχαστικό ολοκλήρωμα είναι ισομετρία των δύο χώρων

$$E[|I(f)|^2] = E[|\int_a^b f_t dB_t|^2] = E(\int_a^b |f_t|^2 dt)$$

ή

$$\|I(f)\|_{L^2} = \|f\|_{H^2}$$

Το επόμενο βήμα είναι να δούμε το ολοκλήρωμα Ito σαν στοχαστική διαδικασία, οπότε χρειαζόμαστε μία απεικόνιση που παίρνει μια διαδικασία από τον H^2 σε διαδικασία και όχι σε τυχαία μεταβλητή. Αυτό επιτυγχάνεται με το να δείξουμε ότι το στοχαστικό ολοκλήρωμα είναι ένα συνεχές martingale.

Πρόταση. Οπότε για κάθε $f \in H^2[0, T]$ υπάρχει μία διαδικασία X_t η οποία ορίζεται ως $X_t = \int_0^t f(s) dB_s$ τότε η διαδικασία X_t ορίζεται ως μία συνεχής martingale

$$X_t = E(\int_0^t f(s) dB_s / F_t) \quad , \quad 0 \le t \le T$$

Ιδιότητες του ολοκληρώματος Ito στον χώρο H^2

1. $\int_{t1}^{t3} f(t) dB_t = \int_{t1}^{t2} f(t) dB_t + \int_{t2}^{t3} f(t) dB_t$
2. $\int_{t1}^{t2} (c_1 f(t) + c_2 g(t)) dB_t = c_1 \int_{t1}^{t2} f(t) dB_t + c_2 \int_{t1}^{t2} f(t) dB_t$
3. $E \int_{t1}^{t2} f(t) dB_t = 0$

Το ολοκλήρωμα Ito μπορεί να επεκταθεί ώστε να ολοκληρώνει διαδικασίες οι οποίες δεν ανήκουν στην κλάση $H^2[0,T]$. Οπότε το ολοκλήρωμα του Ito θα επεκταθεί σε μια ευρύτερη κλάση διαδικασιών την $H^2_{loc}[0,T]$ το οποίο επιτυγχάνεται με την τεχνική localization. Αυτή η κλάση θα περιλαμβάνει όλες τις διαδικασίες που ικανοποιούν το παρακάτω περιορισμό.

$$P\left(\int_0^T |f|^2 dt\right) = 1$$

Αρχικά θα ορίσουμε τι είναι μία localizing sequence, μία localizing είναι μία αύξουσα ακολουθία χρόνων διακοπής όπου κάνει localizing μία διαδικασία *f* που ανήκει στον $H^2[0,T]$ οπότε

$$f(t) 1_{t \le v_n} \in H^2[0,T]$$

Οπότε localization στο $H^2_{loc}[0,T]$ για μία διαδικασία $f \in H^2_{loc}[0,T]$ ορίζεται ως η ακολουθία χρόνων διακοπής για κάθε κάθε $n \ge 1$ υπάρχει $\tau_n \le \infty$ όπου για τ_n ισχύει

$$\tau_n = inf \left\{ \int_0^T t \le : 0 f^2(t) dt = n \right\}$$

είναι η ανταποκρινόμενη localizing sequence που ανήκει στον $H^2[0,T]$ και προσεγγίζει την *f* που ανήκει στην $H^2_{loc}[0,T]$.

Πρόταση. Για κάθε $f \in H^2_{loc}[0,T]$ και οποιαδήποτε localizing ακολουθία $\{\tau_n\}$ εάν η διαδικασία $\int_0^T f(s) 1_{s \le \tau_n} dB_s$ είναι συνεχές martingale τότε για κάθε $t \in [0,T]$ και $m \le n$ ισχύει ότι

$$\int_0^T f(s) 1_{s \le \tau_n} dB_s = \int_0^T f(s) 1_{s \le \tau_m} dB_s \qquad \forall t \le \tau_m$$

Πρόταση. Υπάρχει μία συνεχής διαδικασία $\int_0^T f(s)\,dB_s$ τέτοια ώστε

$$P\left(\int_0^T f(s)\,dB_s = \lim_{n\to\infty}\int_0^T f(s)\,1_{t\le\tau_n}\,dB_s\right)=1 \qquad \forall[0,T]$$

Αλλά παρόλα αυτά τα ολοκληρώματα Ito

$$\int_0^T f(s)\,1_{t\le\tau_n}\,dB_s$$

που συγκλίνουν με πιθανότητα 1 δεν είναι απαραίτητο να συγκλίνουν με την L^2 έννοια ούτε το όριο του να έχει πεπερασμένη μέση τιμή για αυτό το λόγο ορισμένα ολοκληρώματα στον $H^2_{loc}[0,T]$ δεν είναι απαραίτητο να είναι martingales οπότε για αυτό θα εισάγουμε την έννοια το local martingale.

Local martingale

Εάν μια διαδικασία X_t η οποία είναι προσαρμοσμένη σε μία filtration F_t για $0\le t<\infty$ τότε η X_t ονομάζεται local martingale εάν υπάρχει μία αύξουσα ακολουθία χρόνω διακοπής με $\tau_k\to\infty$ καθώς $k\to\infty$ με πιθανότητα 1 τέτοια ώστε για κάθε k η παρακάτω διαδικασία να είναι martingale.

$$X_t = X_{t\wedge\tau_k} - X_0$$

Για κάθε $f\in L^2_{loc}$ υπάρχει ένα συνεχές local martingale τέτοιο ώστε

$$P\left(X_t=\int_0^t f(s)\,dB_s\right)=1$$

Στοχαστική διαδικασία Ito

Έστω B_t είναι μία κίνηση Brown στο χ.π. (Ω, F, P) , στοχαστική διαδικασία Ito ονομάζεται η X_t η οποία έχει την παρακάτω μορφή.

$$X_t = X_0 + \int_0^t a(s)\,ds + \int_0^t b(s)\,dB_s \quad \text{(ip1)}$$

όπου για κάθε $t<\infty$

$$\int_0^t a(s)\,ds<\infty \quad \text{σ.β ή} \quad a\in H^1_{loc}$$

$$\int_0^t b^2(s)\,ds<\infty \quad \text{σ.β. Ή} \quad b\in H^2_{loc}$$

Και η διαδικασία Ito εκτός από την ολοκληρωτική μορφή πολλές γράφεται και σε διαφορική μορφή δηλαδή

$$dX_t=a(t)dt+b(t)dB_t \quad \text{με} \quad X(0)=X_0$$

Πρόταση. Μία στοχαστική διαδικασία Ito X_t όταν ικανοποιεί την *(ip1)* με $a\in H^1$ και $b\in H^2$ τότε η X_t είναι martingale αν και μόνο αν $a=0$ σ.β. γι α κάθε $t\geq 0$

Η Formula του Ito

Έστω η X_t είναι μια διαδικασία Ito της μορφής (ip1), τότε οποιαδήποτε συνάρτηση της X_t δηλαδή $f(t,X_t)$ η οποία έχει συνεχής δεύτερη παράγωγο, δηλαδή $f\in C^{1,2}$ μπορεί να γραφτεί στην παρακάτω μορφή

$$f(t,X_t)=f(0,X_0)+\int_0^t\left(\frac{\partial f}{\partial s}+a\frac{\partial f}{\partial x}+\frac{1}{2}b^2\frac{\partial^2 f}{\partial x^2}\right)dx+\int_0^t b\frac{\partial f}{\partial x}dB_s$$

η formula του Ito μπορεί να χρησιμοποιηθεί για τον υπολογισμό στοχαστικών ολοκληρωμάτων καθώς και για την επίλυση στοχαστικών διαφορικών εξισώσεων.

Προσχέδιο απόδειξης της formulas του Ito

Η απόδειξη θα γίνει ώστε να εντοπίσουμε την προσαύξηση της $f(X_t)$ στο χρονικό διάστημα $[0,t]$ η οποία σε μορφή προσαυξήσεων γράφεται

$$f(t,X_t)-f(0,X_0)=\sum_{i=0}^{n-1}\left[f(t_{i+1},X(t_{i+1}))-f(t_i,X(t_i))\right]$$

$$\sum_{i=0}^{n-1}\left[f(t_{i+1},X(t_{i+1}))-f(t_i,X(t_{i+1}))\right]-\sum_{i=0}^{n-1}\left[f(t_i,X(t_{i+1}))-f(t_i,X(t_i))\right]$$

Όπου η ποσότητα

$$\sum_{i=0}^{n-1}\left[f(t_{i+1},X(t_{i+1}))-f(t_i,X(t_{i+1}))\right] \quad \text{(ito1)}$$

θα αναπτυχθεί χρησιμοποιώντας τη formula του taylor ως προς τη μεταβλητή του χρόνου t δηλαδή $\Delta t_i=t_{i+1}-t_i$ προκύπτει

$$f(t_{i+1},X(t_{i+1}))-f(t_i,X(t_{i+1}))=\frac{\partial f}{\partial t}(t_i,X(t_{i+1}))\,\Delta t_i$$

και χρησιμοποιώντας την προηγούμενη στην (ito1) παίρνοντας το όριο της

$$\lim_{n\to\infty}\sum_{i=0}^{n-1}\left[\frac{\partial f}{\partial t}(t_i, X(t_{i+1}))\Delta t_i\right]=\int_0^t\frac{\partial f}{\partial s}(s, X(s))\,ds$$

στη συνέχεια εφαρμόζουμε στην παρακάτω ποσότητα

$$\sum_{i=0}^{n-1}\left[f(t_i, X(t_{i+1}))-f(t_i, X(t_i))\right]$$

το ανάπτυγμα taylor ως προς την μεταβλητή X, η formula taylor οπότε

$$f(t_i, X(t_{i+1}))-f(t_i, X(t_i))=\frac{\partial f}{\partial x}(t_i, X(t_i))\Delta X_i+\frac{1}{2}\frac{\partial^2 f}{\partial x^2}(t_i, X(t_i))(\Delta X_i)^2$$

και ορίζοντας τη μεταβολή της ΔX_i ως μια διαδικασία Ito δηλαδή

$$\Delta X_i=a(t)\Delta t+b(t)\Delta B_t$$

και αντικαθιστώντας τη διαδικασία Ito στο ανάπτυγμα Taylor προκύπτει

$$f(t_i, X(t_{i+1}))-f(t_i, X(t_i))=\frac{\partial f}{\partial x}(t_i, X(t_i))\Delta X_i+\frac{1}{2}\frac{\partial^2 f}{\partial x^2}(t_i, X(t_i))\Delta X_i^2 \quad \text{ή}$$

$$f(t_i, X(t_{i+1}))-f(t_i, X(t_i))=\frac{\partial f}{\partial x}(t_i, X(t_i))a(t)\Delta t+\frac{\partial f}{\partial x}(t_i, X(t_i))b(t)\Delta B_t$$

$$+\frac{1}{2}\frac{\partial^2 f}{\partial x^2}(t_i, X(t_i))a^2\Delta t^2+\frac{1}{2}\frac{\partial^2 f}{\partial x^2}(t_i, X(t_i))b^2\Delta B_t^2+\frac{\partial^2 f}{\partial x^2}(t_i, X(t_i))\,a\,b\,\Delta t\,\Delta B_t$$

οι δύο τελευταίοι όροι της προηγούμενης εξίσωσης τείνουν στο μηδέν κατά L^2 όταν το $n\to\infty$

(Ελλειπής θα συμπληρωθεί η απόδειξη)

Θεώρημα Αναπαράστασης των Martingales

Έστω X_t μια συνεχής τοπική martingale η οποία είναι προσαρμοσμένη σε μία filtration η οποία παράγεται από μία κίνηση B_t τότε υπάρχει υπάρχει διαδικασία $f\in H^2[0,T]$ τέτοια ώστε

$$X_t=X_0+\int_0^t f(s)\,ds \qquad \forall[0,T]$$

Μία τοπική martigale μπορεί να αναπαρασταθεί ως ένα στοχαστικό ολοκλήρωμα

Στοχαστική διαφορική εξίσωση

Η στοχαστική διαφορική εξίσωση είναι μια ανέλιξη Ito στην οποία οι συντελεστής της a και b εκτός από το χρόνο *t,* εξαρτώνται και από την ίδια τη στοχαστική διαδικασία.

$$dX_t = a(t, X_t)dt + b(t, X_t)dB_t$$

επιπλέον ονομάζεται και στοχαστική ολοκληρωτική εξίσωση γιατί γράφεται ισοδύναμα και στη παρακάτω μορφή

$$X_t = X_0 + \int_0^t a(s, X_s)ds + \int_0^t b(s, X_s)dB_s$$

Οι υποθέσεις οι οποίες γίνονται για τους συντελεστές της παραπάνω εξίσωσης είναι ότι ισχύουν ισχύει η συνθήκη Lipschitz για τους συντελεστές δηλαδή

$$|a(t,x) - a(t,y)| \leq K|x-y|$$

$$|b(t,x) - b(t,y)| \leq K|x-y|$$

και η συνθήκη γραμμικής αύξησης των συντελεστών

$$|a(t,x)| + |b(t,x)| \leq K\sqrt{1+|x|^2}$$

Η αρχική τιμή είναι X_{t_0} , για $0 \leq t_0 \leq T$ είναι non anticipative $E X_{t_0}^2 < \infty$ κάνοντας τις παραπάνω υποθέσεις μπορεί να αποδειχθεί η ύπαρξη και μοναδικότητα λύσης για την παραπάνω στοχαστική διαφορική εξίσωση.

Μια από τις πιο γνωστές ΣΔΕ είναι η απλή γραμμική στοχαστική διαφορική εξίσωση

$$dS(t) = \mu S(t)dt + \sigma S(t)dB_t$$

όπου $S(t)$ είναι η τιμή μιας μετοχής τη χρονική στιγμή *t* και η παραπάνω εξίσωση περιγράφει τη δυναμική της στο χρόνο, ο συντελεστής *μ* είναι το drift της εξίσωσης και σ είναι η διακύμανση της τιμής της μετοχής. Η διακύμανση της μπορεί να εκτιμηθεί με historical volatility ή με πιο σύνθετες μεθόδους όπως το EWMA (Exponentially Weighting Moving Average) ή μοντέλα τύπου GARCH (Generalized Autoregressive Conditional Heteroskedasticity). Στη συνέχεια θα χρησιμοποιήσουμε τη formula του Ito για να την επιλύσουμε η παραπάνω εξίσωση θα έχει μία λύση τύπου $S(t) = f(t, B_t)$ και αυτό που θα προσπαθήσουμε να βρούμε είναι να βρούμε τη μορφή της συνάρτησης *f*, οπότε το πρώτο βήμα είναι να εφαρμόσουμε τη formula του Ito στην *f*

$$dS_t = \{\frac{\partial f}{\partial t}(t, B_t)dt + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(t, B_t)\}dt + \frac{\partial f}{\partial x}(t, B_t)dB_t$$

και αυτό που θα κάνουμε είναι να ταιριάξουμε τους συντελεστές της παραπάνω εξίσωσης με την απλή γραμμική ΣΔΕ προκύπτει το παρακάτω σύστημα με τις δύο εξισώσεις.

$$\mu f(t, x) = \frac{\partial f}{\partial t}(t, x)dt + \frac{1}{2}\frac{\partial^2 f}{\partial x}(t, x)$$

και

$$\sigma f(t, x) = \frac{\partial f}{\partial x}(t, x)$$

η δεύτερη εξίσωση έχει μία λύση τύπου

$$f(t, x) = e^{\sigma x + g(t)}$$

και αντικαθιστώντας την $f(t, x)$ στη αμέσως προηγούμενη και λύνοντας βλέπουμε ότι προκύπτει

$$\frac{\partial g}{\partial t}(t, x) = \mu - \frac{1}{2}\sigma^2$$

έχοντας τη λύση της g καταλήγουμε στη στοχαστική ανέλιξη

$$S(t) = e^{(\mu - \frac{1}{2}\sigma^2)t + \sigma B_t}$$

η οποία είναι γνωστή και με το όνομα γεωμετρική κίνηση Brown (Geometric Brownian Motion). Η μέθοδος που χρησιμοποιήσαμε για να λύσουμε τη απλή γραμμική ΣΔΕ ονομάζεται coefficient matching μέθοδος. Η μέθοδος coefficient matches μπορεί να λύσει αρκετές ΣΔΕ αλλά σε ορισμένες περιπτώσεις δεν είναι αποτελεσματική και υπάρχει μια πιο γενική μεθοδολογία και αυτό ωφείλεται στη μεγάλη ομάδα διαδικασιών Ito που έχουν την παρακάτω μορφή

$$X(t) = b(t)\{x + \int_0^t d(s)dB_s\} \quad \textbf{(cf)}$$

εφαρμόζοντας τη formula του Ito στην προηγούμενη προκύπτει

$$dX(t) = \frac{db(t)}{dt}\{x + \int_0^t d(s)dB_s\}dt + b(t)\cdot d(t)dB_t$$

αν κάνουμε τις υποθέσεις ότι $b(0) = 1$ και $b(t) > 0$ για κάθε $t \geq 0$ τότε η (cf) είναι η λύση της παρακάτω ΣΔΕ

$$dX(t) = \frac{(db(t)/dt)}{b(t)}X(t)dt + b(t)\cdot d(t)dB_t \qquad X(0) = x$$

μπορούμε να λύσουμε ΣΔΕ όπως την διαδικασία Ornstein-Uhlenbeck όπου με την απλή μέθοδο του coefficient matching αποτυγχάνουμε να βρούμε λύσεις. Η διαδικασία Ornstein-Uhlenbeck χρησιμοποιήθηκε από δύο φυσικούς τους Ornstein-Uhlenbeck για να μελετήσουν την κίνηση αερίων το 1931 και πρόσφατα χρησιμοποιήθηκε σε ορισμένα μοντέλα επιτοκίων στα χρηματοοικονομικά. Η εξίσωση είναι

$$dX(t) = -a(t)X(t)dt + \sigma(t)dB_t \quad \text{με} \quad X(0) = x$$

και χρησιμοποιώντας τη καινούργια μεθοδολογία έχουμε

$$\frac{(db(t)/dt)}{b(t)} = -\alpha(t) \quad \text{(cf1)}$$

$$b(t)\cdot d(t) = \sigma(t) \quad \text{(cf2)}$$

η cf1 είναι μια συνήθης διαφορικής εξίσωση όπου με τη συνθήκη $b(0)=1$ έχει ως λύση την εξίσωση

$$b(t) = e^{-\alpha\cdot t}$$

και αντικαθιστώντας την προηγούμενη στην (cf2) προκύπτει

$$d(t) = \sigma e^{\alpha\cdot t}$$

οπότε η λύση της ΣΔΕ Ornstein-Uhlenbeck είναι η στοχαστική διαδικασία

$$X(t) = e^{-\alpha\cdot t}\{x + \int_0^t \sigma e^{\alpha\cdot t} dB_s\} = e^{-\alpha\cdot t}x + \sigma\int_0^t e^{-\alpha\cdot(t-s)} dB_s$$

Μοντέλο Black Scholes

Το μοντέλο Black Scholes είναι ένα απλοποιημένο μοντέλο αγοράς στο οποίο μοντελοποιείται η τιμή ενός ακίνδυνου χρεογράφου όπως ένα T-bill και ενός χρεόγραφο το οποίο εμπεριέχει κίνδυνο όπως μία μετοχή και η δυναμική των τιμών αυτών των χρεωγράφων μοντελοποιείται από τις δυο παρακάτω εξισώσεις.

$$d\beta(t) = r(t)\beta(t)dt \quad \text{με} \quad \beta(0) = 1 \quad \text{(BS 1)}$$

$$dS(t) = \mu(t)S(t)dt + \sigma(t)S(t)dB(t) \quad \text{(BS 2)}$$

και το παραπάνω μοντέλο αγοράς συμβολίζεται με το διάνυσμα $S(t) = (\beta(t), S(t))$

Χαρτοφυλάκιο

Ένα χαρτοφυλάκιο είναι μια στοχαστική διαδικασία $\theta'(t)=(\theta_0(t),\theta_1(t))$ η οποία ουσιαστικά είναι τα βάρη του χαρτοφυλακίου. Δηλαδή η αξία ενός χαρτοφυλακίου σε ένα μοντέλο αγοράς Black Scholes μια χρονική στιγμή *t* είναι

$$V^{\theta}(t)=\theta_0(t)\cdot\beta(t)+\theta_1(t)\cdot S(t)$$

Αυτοχρηματοδοτούμενο χαρτοφυλάκιο

Ένα χαρτοφυλάκιο ονομάζεται αυτοχρηματοδοτούμενο όταν

$$dV(t)=\theta_0\cdot d\beta(t)+\theta_1\cdot dS(t)$$

Arbitrage Χαρτοφυλάκιο

Ένα χαρτοφυλάκιο θ ονομάζεται arbitrage όταν είναι αυτοχρηματοδοτούμενο η διαδικασία αξίας του χαρτοφυλακίου $V^{\theta}(t)$ ικανοποιεί τις παρακάτω συνθήκες $V^{\theta}(0)=0$ και $V^{\theta}(T)\geq 0$ με $P(V^{\theta}(T)>0)>0$

Contigent Claim

Δικαίωμα ανάκλησης είναι μια τυχαία μεταβλητή *X* η οποία ανήκει $X\in F_T$ και έχει τη μορφή $X=\Phi(S(T))$ όπου η Φ είναι η συνάρτηση συμβολαίου. Ανάλογα με το συνάρτηση συμβολαίου έχουμε το Contigent Claim παίρνει και την ανάλογη μορφή.

Ευρωπαϊκό Δικαίωμα Αγοράς (European Call Option)

Για παράδειγμα ένα ευρωπαϊκό δικαίωμα αγοράς (European Call Option) η συνάρτηση συμβολαίου παίρνει τη παρακάτω μορφή

$$\Phi(x)=max(x-K,0)$$

και το οποίο εκτελείται στην περίπτωση που στη λήξη του δηλαδή τη χρονική στιγμή *T* η τιμή της μετοχής $S(T)$ είναι μεγαλύτερη της τιμή K (Strike Price) η οποία έχει καθοριστεί στο συμβόλαιο.

Look Back Call Option

Για ένα Look Back Call η συνάρτηση συμβολαίου παίρνει την παρακάτω μορφή

$$\Phi(x)=max(x-\min_{t\leq T}S(t),0)$$

δηλαδή το Look Back Call option θα έχει ως pay off την τιμή της μετοχής στη λήξη του συμβολαίου δηλαδή τη χρονική στιγμή *T* δηλαδή την $S(T)$ μείων τη μικρότερη δυνατή τιμή που είχε η τιμή της μετοχή S από τη χρονική στιγμή που αγοράστικέ το συμβόλαιο δηλαδή $t=0$ μέχρι την λήξη του δηλαδή $t=T$

Αναπαράγων Χαρτοφυλάκιο (Replicate Portfolio)

Ένα χαρτοφυλάκιο ονομάζεται αναπαράγων όταν τη χρονική στιγμή Τ η αξία του contigent claim μπορεί να αναπαραχθεί από ένα χαρτοφυλάκιο που περιέχει μια μετοχή και ένα ακίνδυνο χρεόγραφο δηλαδή ισχύει η συνθήκη για ένα χαρτοφυλακίου που αναπαράγει την αξία ενός Call Option

$$V(t)=max(S(T)-K,0)$$

Θεώρημα Girsanov

Εάν η B_t είναι μία κίνηση Brown στο μέτρο P και Q είναι το μέτρο στο χώρο των συνεχών συναρτήσεων που παράγεται από τη στοχαστική διαδικασία $S_t=B_t+\mu\cdot t$ τότε κάθε W μετρήσιμη συνάρτηση στο χώρο των συνεχών συναρτήσεων δηλαδή C[0,T] ικανοποιεί την παρακάτω σχέση

$$E^Q[W]=E^P[W\cdot M_T]$$

όπου το M_t είναι martingale στο μέτρο P

$$M_t=\mathrm{e}^{\mu B_t+\mu^2\frac{t}{2}}$$

Παραγωγής της Formulas των Black-Scholes με τη μέθοδο των Martingales

Στη συνέχεια αυτό που μας ενδιαφέρει είναι να έχουμε τις τιμές των ακίνδυνων χρεογράφων σε προεξοφλημένη αξία ως προς το ακίνδυνο χρεόγραφο οπότε το βασικό ενδιαφέρων μας είναι στις προεξοφλημένες τιμές δηλαδή στην ποσότητα

$$D(t)=\frac{S(t)}{\beta(t)}$$

Η οποία πρέπει να είναι martingale στο μέτρο Q. Στη συνέχεια έστω ότι ένα παράγωγο προϊόν πληρώνει (pay off) X στη λήξη του δηλαδή τη χρονική στιγμή T τότε η παρούσα αξία του τη χρονική στιγμή 0 είναι $X/\beta(T)$ οπότε η δίκαιη τιμή τη χρονική στιγμή t=0 για να αγοράσει αυτό το παράγωγο (contigent claim) είναι

$$V_0=E^Q[X/\beta_T]$$

Το pay off του contigent claim πρέπει να είναι τυχαία μεταβλητή η οποία είναι ολοκληρώσιμη και ισχύουν $X\geq 0$ και $E^Q[X^2]<\infty$. Η παραπάνω formula για οποιαδήποτε χρονική στιγμή t παίρνει τη μορφή

$$V_t = \beta_t E^Q[\frac{X}{B_T} / F_t]$$

Αν θέσουμε στη παραπάνω formula $t = T$ τότε ισχύει $V_T = X$, η παραπάνω formula για να χρησιμοποιηθεί πρέπει να ισχύει δύο συνθήκες , η πρώτης είναι ότι τα μέτρα P και Q πρέπει να είναι ισοδύναμα και η ποσότητα *D(t)* πρέπει να είναι martingale στο μέτρο πιθανότητας Q. Στη συνέχεια χρησιμοποιώντας τον παραπάνω τύπο θα εξάγουμε τη formula των Black Scholes για την τιμολόγηση ενός Call Option μέ την τεχνική των martingales. Οπότε για να βρούμε την τιμή του Call Option πρέπει να βρούμε να υπολογίσουμε την παρακάτω ποσότητα.

$$V_0 = e^{-rT} E^Q[max(S_T - K, 0)]$$

Το πρώτο πράγμα που πρέπει να κάνουμε είναι να βρούμε το μέτρο Q το οποίο κάνει την παρακάτω ποσότητα $D(t) = S(t) / \beta(t)$ martingale και για να το κάνουμε θα εκμεταλλευτούμε τη στοχαστική διαφορική εξίσωση για την κανονικοποιημένη τιμή D(t). Επείδη το όλο παράδειγμα γίνεται στα πλαίσια του μοντέλου Black Scholes για να βρούμε την D(t) θα διαιρέσουμε την λύση της BS 2 με την λύση της BS 1 και αυτό που θα προκύψει θα είναι

$$D(t) = \frac{S(t)}{\beta(t)} = \frac{S(0) e^{(\mu - \frac{1}{2}\sigma^2) t + \sigma B_t}}{e^{rt}} = \frac{S(t)}{\beta(t)} = S(0) e^{(\mu - r - \frac{1}{2}\sigma^2) t + \sigma B_t}$$

και χρησιμοποιώντας τη formula του Ito στην παραπάνω εξίσωση βρίσκουμε

$$dD(t) = (\mu - r) D(t) dt + \sigma D(t) dB_t$$

στην οποία αναδιατάσσοντας του όρους λαμβάνουμε

$$dD(t) = D(t) \sigma (\frac{(\mu - r)}{\sigma} dt + dB_t)$$

χρησιμοποιώντας το θεώρημα Girsanov βλέπουμε ότι

$$\tilde{B}_t = \frac{\mu - r}{\sigma} t + B_t$$

είναι μία κίνηση Brown στο μέτρο Q. Η στοχαστική διαφορική εξίσωση η οποία μοντελοποιεί τις προεξοφλημένες τιμές μπορεί να γραφτεί σε όρους του $\tilde{B}_t$ και γίνεται

$$dD(t) = D(t) \sigma \tilde{B}_t$$

Αφού έχουμε ορίσει το μέτρο Q με την παραπάνω διαδικασία το επόμενο βήμα είναι να εκφράσουμε του όρους της S_t σε όρους $\tilde{B}_t$ οπότε έχουμε

$$S(T)=S(0)e^{rT-\sigma^2\frac{T}{2}+\sigma\tilde{B}_t}$$

Αντικαθιστώντας στην αναμενόμενη τιμή ως προς το μέτρο Q παίρνουμε

$$V_0=e^{-rT}E^Q[max(S(0)e^{rT-\sigma^2\frac{T}{2}+\sigma\tilde{B}_t}-K,0)]$$

παρατηρώντας την ποσότητα $Y=-\sigma^2 T/2+\sigma\tilde{B}_t$ βλέπουμε ότι είνα μια τυχαία μεταβλητή που ακολουθεί την κανονική κατανομή με μέσω $-\sigma^2 T/2$ και διακύμανση $\sigma^2 T$ οπότε η αναμενόμενη τιμή μπορεί να γραφτεί ώς

$$V_0=e^{-rT}\int_{\log(K/S_0)-rT}^{\infty}(S(0)e^{rT+Y}-K)e^{-(Y+\sigma^2 T/2)^2/2\sigma^2 T}\frac{dy}{\sigma\sqrt{2\pi T}}$$

όπου

$$V_0=S(0)\int_{\log(K/S_0)-rT}^{\infty}e^Y e^{-(Y+\sigma^2 T/2)^2/2\sigma^2 T}\frac{dy}{\sigma\sqrt{2\pi T}}+e^{-rT}K\int_{\log(K/S_0)-rT}^{\infty}e^{-(Y+\sigma^2 T/2)^2/2\sigma^2 T}\frac{dy}{\sigma\sqrt{2\pi T}}$$

και καταλήγουμε στη γνωστή διάσημη formula των Black Scholes με την οποία μπορούμε να τιμολογίσουμε ένα Ευρωπαικό διακαίωμα Αγοράς.

$$C=S_0 N(d_1)-Ke^{-rt}N(d_2)$$

όπου

$$d_1=\frac{\ln(S_0/K)+(r+\sigma^2)T}{\sigma\sqrt{T}}$$

$$d_2=d_1-\sigma\sqrt{T}$$

Η απόδειξη για την formula των Black Scholes έγινε με την μέθοδο των martingales που όμως δεν ήταν αυτή η μέθοδος που χρησιμοποιήσαν οι Black και Scholes για να εξάγουν της Formula τους. Η μέθοδος την οποία χρησιμαοποίησαν ήταν να κατασκευάσουν μια μερική διαφορική εξίσωση που σήμερα ονομάζεται Black-Scholes PDE και να την επιλύσουν και αυτή η μέθοδος παρουσιάζεται παρακάτω.

Παραγωγή της Formulas των Black Scholes με τη μέθοδο της Μερικής Διαφορικής Εξισωσης

Οι Black και Scholes κατάφεραν να κατασκευάσουν τη μερική διαφορική εξίσωση με δυο διαφορετικούς τρόπους ο πρώτος τρόπος ήταν η παραγωγής της BS PDE από αντισταθμισμένο χαρτοφυλάκιο (Hedged Portfolio) και ο δεύτερος τρόπος από το γνωστό μοντέλο στα χρηματοοικονομικά CAPM.

1) Αντισταθμισμένο Χαρτοφυλάκιο (Hedged Portfolio)

Η ιδέα του αντισταθμισμένου χαρτοφυλακίου βασιζόταν τη δημιουργία ενός χαρτοφυλακίου στο οποίο θα έπαιρναν long position σε μία μετοχή και για να κάνουν αντιστάθμιση θα έπαιρναν short position σε δικαιώματα προαίρεσης (options) όπου ο αριθμός τους θα άλλαζε συνεχώς για να αντισταθμίσει τη μεταβαλλόμενη της τιμής της μετοχής. Η αξία του χαρτοφυλακίου θα είναι

$$V^{\theta}(t) = S(t) \cdot 1 - \frac{f(t, S(t))}{\partial f / \partial x}$$

όπου $f(t, S(t))$ είναι η τιμή του δικαιώματος αγοράς και

$$\frac{1}{\partial f / \partial x}$$

είναι ο αριθμός των call option ο οποίος μεταβάλλεται για να αντισταθμίσει τη θέση του χαρτοφυλακίου. Η μεταβολή της αξίας του χαρτοφυλακίου στο χρόνο είναι

$$\Delta V^{\theta}(t) = \Delta S(t) \cdot 1 - \frac{\Delta f(t, S(t))}{\partial f / \partial x} \quad \text{(HP 1)}$$

Η επειδή η τιμή του option αλλάζει συνεχώς στη διάρκεια το χρόνου χρησιμοποιούν της formula του Ito

$$\Delta f(t, S(t)) = \frac{\partial f}{\partial t}(t, S_t)\Delta t + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(t, S(t))\sigma^2 S(t)^2 \Delta t + \frac{\partial f}{\partial x}(t, S(t))\Delta S_t \quad \text{(HP 2)}$$

και αντικαθιστώντας την (HP2) στην (HP1) σε συνδυασμό του επιχειρήματος ότι σε ένα αντισταθμισμένο χαρτοφυλάκιο σε ένα χρονικό διάστημα Δt έχει απόδοση όσο ένα ακίνδυνο χρεόγραφο στο διάστημα Δt προκύπτει η μερική διαφορική εξίσωση.

$$\left(S(t) \cdot 1 - \frac{f(t, S(t))}{\partial f / \partial x}\right) \cdot r \cdot \Delta t = -\left(\frac{\partial f}{\partial t}(t, S_t) + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(t, S(t))\sigma^2 S(t)^2\right)\Delta t / \partial f / \partial x$$

και διαγράφοντας το Δt προκύπτει η μερική διαφορική εξίσωση Black-Scholes.

$$\frac{\partial f}{\partial t}(t,S_t)=r\cdot f(t,S_t)-r\cdot S(t)\frac{\partial f}{\partial x}(t,S(t))-\frac{1}{2}\frac{\partial^2 f}{\partial x^2}(t,S(t))\sigma^2 S(t)^2$$

H Formula των Black Scholes

H Formula των Black Scholes που προέκυψε από τις παραπάνω μεθοδολογίες για την τιμολόγηση ενός Ευρωπαϊκού Δικαιώματος Αγοράς (European Call Option) παίρνει την παρακάτω μορφή

$$C=S_0 N(d_1)-Ke^{-rt}N(d_2)$$

όπου

$$d_1=\frac{\ln(S_0/K)+(r+\sigma^2)T}{\sigma\sqrt{T}} \qquad d_2=d_1-\sigma\sqrt{T}$$

Ένα κύριο χαρακτηριστικό της παραπάνω formulas εκτός από ότι δίνει μια αναλυτική λύση για το Ευρωπαϊκό δικαίωμα αγοράς είναι ότι δείχνει πως κατασκευάζεται ένα αναπαράγων χαρτοφυλάκιο και αυτό φαίνεται γιατί έχουμε θέση αγοράς στη μετοχή η οποία είναι

$$S_0 N(d_1)$$

και η ποσότητα $S_0 N(d_1)$ δείχνει την αξία των μετοχών που έχουμε αγοράσει στο χαρτοφυλάκιο όπου η ποσότητα $N(d_1)$ δείχνει το αριθμό των μετοχών το $N(d_1)$ είναι μικρότερο της μονάδα που σημαίνει ότι αναπαράγων χαρτοφυλάκιο για κάθε μονάδα Call Option αντιστοιχεί μία μετοχή και όχι περισσότερες. Το άλλο τμήμα της Formulas δηλαδή το

$$-Ke^{-rt}N(d_2)$$

δείχνει ότι το αναπαράγων χαρτοφυλάκιο του δικαιώματος παίρνει θέση πώλησης (short position) στο ακίνδυνο χρεόγραφο. Ένα παράδειγμα εφαρμογής της Formulas είναι το παρακάτω

Παράδειγμα

Για παράδειγμα έστω ότι η τιμή της μετοχής είναι στα 9 ευρώ δηλαδή $S_0=9$ ευρώ η strike price είναι $K=10$ ευρώ και έχει διάρκεια το συμβόλαιο 3 μήνες τότε το $T=0{,}25$ με risk free rate να έχει απόδοση $r=0.04$ και το volatility το οποίο μπορεί να υπολογιστεί με EWMA ή GARCH από τη χρονοσειρά της μετοχής.

$$d_1=\frac{\ln(9/10)+(0{,}04+0{,}3^2)T}{0{,}3\sqrt{0{,}25}}=-0{,}56$$

$$d_2=0{,}14-0{,}3\sqrt{(0{,}25)}=-0{,}71$$

$$C = 9\cdot N(-0{,}56) - 10\cdot e^{-0{,}04\cdot 0{,}25t} N(-0{,}71) = 0{,}2249$$

Σε βασικά χρηματοοικονομικά προϊόντα όπως είναι το European Call Option υπάρχει αναλυτική λύση όπως είναι η Formula των Black Scholes, αλλά σε πιο σύνθετα δικαιώματα χρειάζεται να καταφύγουμε σε υπολογιστικές μεθόδους όπως είναι αυτή της προσημείωσης.

Δ αντιστάθμιση (Delta Hedge)

Σε αυτό το σημείο θα παρουσιάσουμε πως γίνεται η δέλτα αντιστάθμιση σε ένα χαρτοφυλάκιο, θα χρησιμοποιήσουμε τα δεδομένα του προηγούμενου παραδείγματος, στο προηγούμενο παράδειγμα χρησιμοποιώντας τη Formula των Black Scholes η τιμή του Eυροpean Call που προέκυψε ήταν 0.2249 ευρώ. Στη δέλτα αντιστάθμιση σημαντικό ρόλο παίζει ο συντελεστής ευαισθήσιας Δ όπου είναι ένα από τα greeks για τα δεδομένα του προηγούμενο παραδείγματος αυτός ο συντελεστής παίρνει την τιμή

$$\Delta = \frac{\partial C}{\partial S} = N(d_1) = 0{,}29$$

Για τη δημιουργία ενός χαρτοφυλακίου το οποίο έχει δέλτα αντιστάθμιση μπορώ να πάρω θέση πώλησης (short position) για παράδειγμα σε 100 δικαιώματα αγοράς Ευρωπαϊκού τύπου και χρησιμοπιώντας το συντελεστής Δ θα πάρω θέση αγοράς (long position) σε 29 μετοχές. Οπότε έχω δημιουργήσει ένα χαρτοφυλάκιο του οποίου η αξία του είναι

$$V = 29\cdot 9 - 100\cdot 0.22 = 261 - 22 = 239$$

Στη συνέχεια μπορούμε να πάρουμε την περίπτωση όπου όπου η τιμή της μετοχής από τα 9 ευρώ στα οποία βρίσκεται πέφτει στα 8 ευρώ τότε η αξία των μετοχών που έχω αγοράσει στο χαρτοφυλάκιο πέφτει από τα 261 ευρώ στα $29\cdot 8 = 232$ ευρώ δηλαδή χάνω 29 ευρώ από την πτώση της μετοχής στη περίπτωση που δεν έχω κάνει hedge το χαρτοφυλάκιο. Όμως από την στιγμή που έχω πάρει θέση πώλησης στα call option κερδίζω από αυτά εξαιτίας του ότι πέφτει η τιμή από την πτώση τη μετοχής και χρησιμοποιειώντας τη formula των Black Scholes με τη διαφορά ότι αντι για $S = 9$ βάζουμε τη νέα τιμή της πτώση $S = 8$ η καινούργια τιμή των call option είναι $C = 0.05$. Οπότε ενώ από τις μετοχές χάνω 29 ευρώ από τα option κερδίζω $22 - 0.05\cdot 100 = 17$ οπότε η απώλειες του χαρτοφυλακίου αντί για 29 ευρώ θα είναι 12 ευρώ.

ΕΙΣΑΓΩΓΗ ΣΤΗ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C/C++

Πρώτη επαφή με την C/C++

Όλα τα προγράμματα στη C αποτελούνται από μία ή περισσότερες συναρτήσεις και η πιο βασική συνάρτηση που πρέπει να υπάρχει σε κάθε πρόγραμμα της C είναι η main(), και κάθε φορά που ξεκινάει η εκτέλεση ενός προγράμματος στη C η πρώτη συνάρτηση που καλείται είναι η main(). Το πιο απλό πρόγραμμα μπορεί να γραφτεί είναι το παρακάτω.

```
#include <stdio.h>

main()
{
   printf("Hello World");
   return 0;
}
```

Όπου η εντολή include στο παραπάνω πρόγραμμα φορτώνει στο πρόγραμμα μια εξωτερική βιβλιοθήκη με συναρτήσεις και πιο συγκεκριμένα την stdio.h η οποία περιέχει συναρτήσεις η οποίες είναι χρήσιμες για την εισαγωγή ή την εκτύπωση δεδομένων στη οθόνη. Η συνάρτηση printf() περιέχεται στη βιβλιοθήκη stdio.h και αυτό που κάνει είναι να παίρνει ως όρισμα ενα αλφαριθμιτικό και το τυπώνει στη οθόνη όταν εκτελεστεί το πρόγραμμα. Για την εκτέλεση το προγραμμάτων θα χρησιμοποιήσουμε το πρόγραμμα Dev-C++ που είναι δωρεάν πρόγραμμα όπου μπορούμε να εκτελέσουμε τα πρόγράμμα στη γλώσσα C και πιο συγκεκριμένο το Dev-C++ είναι ένα IDE (Intergrated Development Enviroment) στο οποίο είναι ενσωματωμένο ο compiler MinGW. Σε αυτό το περιβάλλον το πιο απλό πρόγραμμα στη C μπορεί να παίρνει την παρακάτω μορφή

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
   cout << "Hello World" << endl;
   cout << "This is the first programe" << endl;
   system("PAUSE");
   return EXIT_SUCCESS;
}
```

Στις δύο πρώτες γραμμές με την εντολή include έχουμε ως αποτέλεσμα να συμπεριλιφθούν οι βιβλιοθήκες συναρτήσεων cstdlib και iostream όπου η cstdlib είναι η Standard Library της C δηλαδή η βιβλιοθήκη που περιέχει τις συναρτήσεις εισαγωγής και εκτύπωσης δεδομένων στο πρόγραμμα, ενώ η iostream είναι η αντίστοιχη βιβλιοθήκη συναρτήσεων για την C++. Στο παραπάνω πρόγραμμα τυπώνουμε στην οθόνη τα αλφαριθμητικά χρησιμοποιώντας την εντολή cout η οποία εμπεριέχεται στη βιβλιοθήκη iostream αντί με την συνάρτηση printf() που περιέχεται στην cstdlib, και με την εντολή endl τελείωνει η γραμμή εκτύπωσης και το επόμενος χαρακτηρας ξεκινάει να τυπώνεται στην επόμενη γραμμή. Στο παραπάνω πρόγραμμα περιέχεται και η εντολή “using namespace” όπου η εντολή αυτή ομαδοποιεί ένα σύνολο συναρτήσεων, κλάσεων ή αντικοιμένων και αυτό για την standard library std και πιο συγκεκριμένα για να χρησιμοποιήσουμε την εντολή cout χωρίς το “namespace” το πρόγραμμα θα γραφόταν

```
#include <cstdlib>
#include <iostream>

int main(int argc, char *argv[])
{
   cout::std << "Hello World" << endl;
   system("PAUSE");
   return EXIT_SUCCESS;
}
```

Τύποι δεδομένων

Οι τύποι δεδομένων χρησιμοποιούνται για να δηλώσουν τι τύπου δεδομένα θα αποθηκευτούν στης μεταβλητές του προγράμματος ώστε να καθορίσουν το αριθμό το bytes που θα δεσμεύσουν σε κάθε μεταβλητή οι 5 κύριοι τύποι δεδομένων είναι *int* όπου χαρακτηρίζουμε μια μεταβλητής στην οποία αποθηκεύονται ακέραιες ποσότητες, είναι ο τύπος *char* όπου δηλώνουμε ότι στη μεταβλητή αποθηκεύονται αλφαριθμητικά, τύποι *float* και *double* για πραγματικούς αριθμούς και τέλος το void για να δειλώσει ότι μια συνάρτηση δεν έχει παραμέτρους.

Μεταβλητές

Μια μεταβλητή είναι μια τοποθεσία στη μνήμη του υπολογισμού που της έχουμε δώσει ένα όνομα και τη χρησιμοποιούμε για να αποθηκεύσουμε δεδομένα δηλαδή χρησιμοποιώντας μια μεταβλητή αναφερόμαστε σε μια συγκεκριμένη θέση που έχουν αποθηκευτεί δεδομένα στη μνήμη του υπολογιστή και πριν χρησιμοποιήσουμε μεταβλητή πρέπει πρώτα να τη δηλώσουμε. Ο τρόπος με το οποίο δηλώνουμε της μεταβλητές και εκχωρούμε σε αυτές δεδομένα είναι ο παρακάτω

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
   int s = 2;
   int f = 10;
   int r;

   r=f/s;

   cout << "the result is " << r << endl ;

   system("PAUSE");
   return EXIT_SUCCESS;
}
```

Όπου στο παραπάνω απλό πρόγραμμα έχουμε δειλώσει τρεις μεταβλητές τις s, f, r όπου και οι τρεις έχουμε δειλώσει ότι τα δεδομένα που θα αποθηκευτούν σε αυτά είναι ακέραιοι αριθμοί από το int και στις s, f έχουμε εκχωρήσει ως δεδομένα τους αριθμούς 2 και 10 αντίστοιχα ένα στιν r εκχωρείται το αποτέλεσμα από την διαίρεση των δύο άλλον μεταβλητών.

Ροές ελέγχου

Οι ροές ελέγχου είναι το if το οποίο επιλέγει μια κατάσταση σε ένα κομμάτι κώδικα ανάλογα με τις συνθήκες παράδειγμα

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
   int f = 10;

   if(f >= 10){
      f = f+1;
   }else{
         f = f-1;
   }

   cout << "the value of f is " << f << endl ;

   system("PAUSE");
   return EXIT_SUCCESS;
}
```

Όπου στο παραπάνω πρόγραμμα έχουμε δηλώσει τη μεταβλητή f και τις έχουμε εκχωρήσει την τιμή 10 στη συνέχεια με το if κάνουμε μια σύγκριση και βλέπου αν οι τιμή f είναι μεγαλύτερη ή ίση του 10 τότε θα προστεθεί συν 1 στη μεταβλητή f αλλιώς να αφαιρεθεί μια μονάδα από την τιμή, στην προκειμένη περίπτωση θα προστίθενται μια μονάδα. Μιά άλλη ροή είναι ο βρόγχος επανάληψης for με τον οποίο μπορούμε να επαναλάβουμε πολλές φορές ένα κομμάτι κώδικα όπως για παράδειγμα

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{

   int i,f,sum;
   f=0;
```

```
    for(i=0;i<=10;i++){
      f += 2;
    }

    cout << "the value of f is " << f << endl ;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Ο βρόγχος loop παίρνει τρία ορισμάτα όπου καθορίζουν πόσες φορές θα επαναληφθεί σαν πρώτο όρισμα παίρνει τη μεταβλητή i όπου της εκχωρείται σαν αρχική τιμή το 0, το δεύτερο όρισμα του βρόγχου δηλώνει ότι το κομμάτι κώδικα που βρίσκεται μέσα στο βρόγχο θα επαναλαμβάνεται μέχρι η μεταβλητή i να πάρει την τιμή 10, και το τρίτο όρισμα του δείχνει ότι σε κάθε επανάληψη του βρόγχου η μεταβλητή i θα αυξάνεται κατά μία μονάδα. Η μεταβλητή f έχει αρχικοποιηθεί με την τιμή μηδέν σε κάθε επανάληψη του βρόγχου όπου συνολικά κάνει 11 επαναλήψεις προσθέτεται συν δύο μονάδες, το αποτέλεσμα είναι ότι όταν τελειώσει ο βρόγχος η μεταβλητής θα έχει αποθηκευμένο τον αριθμό 22.

Πίνακες (Arrays)

Ένα πίνακας στο C είναι μια μεταβλητή η οποία περιέχει πολλαπλά στοιχεία του τα οποία έχουν το ίδιο τύπο δεδομένων

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{

    int i,j;
    int random_matrix[4][3];

    for(i=0;i<4;i++){
        for(j=0;j<3;j++){
            random_matrix[i][j] = (rand() % 10) + 1;
            cout << random_matrix[i][j] << "   ";
        }
      cout <<  endl;
    }

    system("PAUSE");
    return EXIT_SUCCESS;

}
```

Το παραπάνω πρόγραμμα ένα πίνακα έχουμε δειλώσει ένα πίνακα δύο διαστάσεων όπου τον έχει ονομαστεί random_matrix πριν το όνομα έχουμε δώσει την εντολή int που σημαίνει ότι τα δεδομένα που θα αποθηκευτούν στο array θα είναι ακέραιοι αριθμοί, πίνακα είναι δύο διαστάσεων με τέσσερις γραμμές και τρεις στήλες. Ένα τρόπος για να εισάγουμε δεδομένα ή να διαβάσουμε δεδομένα από τον πίνακα είναι με τον βρόγχο for, όπου στο συγκεκριμένο πρόγραμμα χρησιμοποιούμε τη μεταβλητή i για τις γραμμές του πίνακα τη μεταβλητή j για τις στίλες του πίνακα και οι οποίες αρχικοποιούνται μέσα στους βρόγχους δηλαδή παίρνουν την τιμή 0. Αυτό που κάνει το πρόγραμμα είναι να εισάγει σε κάθε κελί του πίνακα ένα τυχαίο ακέραιο αριθμό από το 0 εώς το 10 χρησιμοποιώντας τη συνάρτηση rand() η οποία εμπεριέχεται στη βιβλιοθήκη cstlib.

Διανύσματα (Vectors)

Μια άλλη χρήσιμη λειτουργία της C++ και όχι τις C είναι η κλάση διανυσμάτων vector της Standard Template Library όπου με την οποία μπορούν να δημιουργηθούν δυναμικοί πίνακες arrays, κατά τη διάρκεια που τρέχει ένα πρόγραμμα οι πίνακες δεν μπορούν να αλλάξουν μέγεθος και να προσαρμοστούν στις συνθήκες του προγράμματος, ενώ ένα vector μπορεί να το κάνει και να αλλάξει δυναμικά το μέγεθος του δεσμεύοντας την απαραίτητη μνήμη του υπολογιστή.

ΥΠΟΛΟΓΙΣΤΙΚΈΣ ΜΈΘΟΔΟΙ ΣΤΗΝ ΤΙΜΟΛΟΓΙΣΗ ΔΙΚΑΙΩΜΑΤΩΝ (OPTIONS)

Multiperiod Binomial Pricing Model

Στο διωνυμικό μοντέλο τιμολόγησης δικαιωμάτων το υποκείμενο του δικαιώματος δηλαδή η μετοχή τη χρονική στιγμή *t* παίρνει την τιμή S_t , στη πρώτη χρονική στιγμή δηλαδή t=0 η μετοχή έχει ως αρχική τιμή S_0 ενώ στην στη μεταβολή του χρόνου δηλαδή την επόμενη χρονική στιγμή t=1 η τιμή της μπορεί να βρεθεί σε δύο πιθανές καταστάσεις, να ανέβει και να πάρει την τιμή $S_1^{up} > S_0$ ή να κατέβει και να πάρει την τιμή $S_1^{down} < S_0$, το ποσοστό της ανόδου της μετοχής εξαρτάται από την μεταβλητή *u* για την οποία ισχύει $u > 1$ δηλαδή $S_1^{up} = u \cdot S_0$ ενώ το ποσοστό της καθόδου εξαρτάται από την μεταβλητή $d < 1$ με $S_{t+1}^{down} = d \cdot S_0$, η πιθανότητα να ανέβει μπορεί να υπολογιστεί εξής

$$p \cdot S_1^{up} + (1-p) \cdot S_1^{down} = S_0 e^{r \cdot \Delta t}$$

ή

$$p \cdot u \cdot S_0 + (1-p) \cdot d \cdot S_0 = S_0 e^{r \cdot \Delta t}$$

και γενικεύοντας για οποιαδήποτε χρονική στιγμή η παραπάνω σχέση μπορεί να γραφτεί για οποιαδήποτε S σε οποιαδήποτε χρονική στιγμή δηλαδή

$$p \cdot u \cdot S + (1-p) \cdot d \cdot S = S e^{r \cdot \Delta t}$$

όπου λύνοντας ως προς *p* δηλαδή την πιθανότητα ανόδου προκύπτει

$$p = \frac{e^{r \cdot \Delta t} - d}{u - d} \quad \text{(bion1)}$$

ενώ η πιθανότητα καθόδου της μετοχής είναι

$$1 - p = \frac{u - e^{r \cdot \Delta t}}{u - d}$$

οπότε τη χρονική στιγμή *t=1* δημιουργήθηκαν δύο πιθανές καταστάσεις της μετοχής,, στην επόμενη μεταβολή του χρόνου δηλαδή από την *t=1* στην *t=2* ο κάθε κόμβος μπορεί να βρεθεί σε άλλες δύο πιθανές καταστάσεις όπως φαίνεται στο παρακάτω σχήμα και μάλιστα η κάθοδος από την κατάσταση S_1^{up} συμπίπτει με την άνοδο από την κατάσταση S_1^{down} και η διαδικασία μπορεί να συνεχιστεί πέρα από τη χρονική στιγμή *t=2* για *t=N*. Πιο συγκεκριμένα στο πλαίσιο ανάλυση διωνυμικών δέντρων των Cox-Ross-Rubinstein για τις μεταβλητές u και d ισχύει η παρακάτω ταυτότητα

$$u=\frac{1}{d}$$

και

$$u=e^{\sigma\cdot\sqrt{\Delta t}} \qquad d=e^{-\sigma\cdot\sqrt{\Delta t}}$$

0	1	2
		$S_2^{up}=u\cdot u\cdot S_0$ $C_{uu}=max(u\cdot u\cdot S_0-K,0)$
	$S_1^{up}=u\cdot S_0$ $C_u=e^{-r}(q\cdot C_{uu}+(1-q)\cdot C_{ud})$	
S_0 $C_t=e^{-r}(q\cdot C_u+(1-q)\cdot C_d)$		$\left(S_2^{up}\, ή\, S_2^{down}\right)=u\cdot d\cdot S_0$ $C_{ud}=max(u\cdot d\cdot S_0-K,0)$
	$S_1^{down}=d\cdot S_0$ $C_d=e^{-r}(q\cdot C_{ud}+(1-q)\cdot C_{dd})$	
		$S_2^{down}=d\cdot d\cdot S_0$ $C_{dd}=max(d\cdot d\cdot S_0-K,0)$

Ένα γενικός τύπος για την εύρεση τη τιμής τη μετοχής σε κάθε κόμβο του διωνυμικού δένδρου έχοντας ως δεδομένα την τιμή της μετοχής τη χρονική στιγμή *t* καθώς και της μεταβλητές u και d είναι

$$S_{i,j}=S_t\cdot u^j\cdot d^{i-j} \quad \text{(bion2)}$$

όπου με *(i,j)* συμβολίζουμε το κάθε κόμβο του δένδρου όπου *i* είναι η κάθε χρονική περίοδος δηλαδή ο αριθμός χρονικών βημάτων στο δέντρο όπου παίρνει τιμές στο $i=0,1,2,\ldots,N$ και *j* ο αριθμός των κόμβων που υπάρχει σε κάθε περίοδο και οποίο παίρνει τιμές στο διάστημα $j=0,1,2,\ldots,i$. Οπότε στη πρώτη φάση θα πάρουμε αν θέλουμε να δημιουργήσουμε ένα διωνυμικό δένδρο δύο περιόδων θα ξεκινήσουμε από την t=0 όπου είναι γνωστή η αρχική τιμή της μετοχής S_0 και χρησιμοποιώντας τον παραπάνω τύπο (bion2) θα συμπληρώσουμε τους κόμβους του δένδρου. Μόλις γίνει αυτό θα πάνε στην τελευταία περίοδο αφού έχουμε πάρει δένδρο δύο περιόδων την t=2 και θα βρούμε τις τιμές του δικαιωμάτων σε αυτήν δηλαδή της C_{uu} , C_{ud} C_{dd} και έχοντας βρει αυτές τις τιμές στη συνέχεια θα το μόνο που έχουμε να κάνουμε να τρέξουμε προς τα πίσω το αλγόριθμος του δυναμικού προγραμματισμού σε κόμβο του δέντρου όπου είναι

$$C_{i,j}=e^{-r\Delta t}\left(p\cdot C_{i+1,j+1}+(1-p)\cdot C_{i+1,j}\right)$$

όπου p είναι η σχέση (bion1). Η παραπάνω μεθοδολογία για την τιμολόγηση ενός Call Option υλοποιείται στο παρακάτω πρόγραμμα που είναι γραμμένο στην C

```
#include <cstdlib>
#include <iostream>
#include <cmath>

using namespace std;

double multiperiod_binomial_American_call_pricing(const double& St,
                                    const double& K,
                                    const double& r,
                                    const double& sigma,
                                    const double& T,
                                    const int& Num_of_Periods)
{

int i,j;
double S[200][200] = {0.0};
double C[200][200] = {0.0};
double up = 0.0;
double down = 0.0;
double a;
double p;
double dt;


dt=T/Num_of_Periods;
up = exp(sigma*sqrt(dt));
down = 1/up;
a = exp(r*dt);
p = (a-down) / (up - down);


for(i=0;i<=Num_of_Periods;i++){
   for(j=0;j<=i;j++){
      S[i][j] = St*(pow(up,j)*pow(down,i-j));
      C[i][j] = 0.0;
   }
}

for(j=0;j<=i;j++){
   C[Num_of_Periods][j] = max(S[Num_of_Periods][j]-K,0.0);
}

for (i = Num_of_Periods-1; i >= 0; i--){
   for (j = i; j >= 0; j--){
      C[i][j] = exp(-r*dt)*(p*(C[i+1][j+1]) + (1- p)*(C[i+1][j]));
   }
}
 return C[0][0];
}

int main(int argc, char *argv[])
{
```

```
cout << multiperiod_binomial_American_call_pricing(33.75,35,0.055,0.15,0.75,4) <<
endl ;



   system("PAUSE");
   return EXIT_SUCCESS;
}
```

Το παραπάνω πρόγραμμα μπορεί να χρησιμοποιηθεί και για την τιμολόγηση ενός Αμερικάνικου δικαιώματος αγοράς με τη διαφορά ότι πρέπει να γίνουν κάποιες τροποποιήσεις δηλαδή η σχέση (bion1) πλέον θα πρέπει να υπάρχει και όρος των μερισμάτων

$$p=\frac{e^{(r-q)\cdot\Delta t}-d}{u-d}$$

και καθώς θα πρέπει να συμπεριλάβουμε και ένα αλγόριθμου δυναμικού προγραμματισμού που θα ελέγχει αν είναι βέλτιστη η εξάσκηση του δικαιώματος σε κάθε κόμβο

$$C_{i,j}=max\left(S_{i,j},e^{-r\Delta t}\left(p\cdot C_{i+1,j+1}+(1-p)\cdot C_{i+1,j}\right)\right)$$

όπου $S_{i,j}$ είναι η σχέση (bion2)

Προσημείωση Monte Carlo για τιμολόγηση European Call Option

Μια μεθοδολογία για την τιμολόγηση ενός προϊόντος με τη μέθοδο monte carlo, σε πρώτη φάση αφού έχουμε υποθέσει ότι οι τιμές της μετοχής ακολουθούν κανονική κατανομή θα χρειαστούμε τυχαίες μεταβλητές που θα παράγονται από την κανονική κατανομή. Για να το πετύχουμε αυτό θα χρησιμοποιήσουμε τη polar method του marsaglia σύμφωνα με την οποία θα πρέπει να παράγουμε δύο τυχαία μεταβλητές από την ομοιόμορφη κατανομή στο διάστημα $(0,1]$ δηλαδή

$$u_1\sim U(0,1] \quad \text{(mc1)}$$
$$u_2\sim U(0,1] \quad \text{(mc2)}$$

Στη συνέχεια υπολογίζουμε τις μεταβλητές

$$v_1=2\cdot u_1-1 \quad \text{(mc3)}$$
$$v_2=2\cdot u_2-1 \quad \text{(mc4)}$$

και κρατάμε εκείνες τις τυχαίες μεταβλητές για τις οποίες ισχύει η συνθήκη

$$s = v_1^2 + v_1^2 < 1 \quad \text{(mc5)}$$

στη συνέχεια υπολογίζουμε

$$x_1 = v_1 \sqrt{\frac{-2 \cdot \log s}{s}}$$

$$x_2 = v_2 \sqrt{\frac{-2 \cdot \log s}{s}}$$

όπου οι τυχαίες μεταβλητές x_1, x_2 κατανέμονται σύμφωνα με την τυποποιημένη κανονική κατανομή.

$$x_1 \sim N(0,1)$$

$$x_2 \sim N(0,1)$$

Ο βασικός τύπος που θα χρησιμοποιήσουμε για την τιμολόγηση ενός European Call Option

$$V_0 = e^{-rT} E^Q[max(S(T) - K, 0)]$$

τον οποίο τον συναντήσαμε στη μέθοδο των martingales με contract function αφού είναι European Call Option $max(S(T) - K, 0)$, η τιμή της μετοχής ακολουθεί τη γεωμετρική κίνηση Brown κάτω από το ισοδύναμο μέτρο Q.

$$S(T) = S(0) e^{rT - \sigma^2 \frac{T}{2} + \sigma \tilde{B}_t}$$

Η αναμενόμενη τιμή ενός δικαιώματος αγοράς θα προέρχεται από τον παρακάτω τύπο

$$\tilde{c} = e^{-rT} E^Q[\sum_{i=1}^{n} max(S_i(T) - K, 0)]$$

Στη συνέχεια χρησιμοποιώντας τα ίδια νούμερα όπως στο προηγούμενο μέθοδο μπορούμε να υλοποιήσουμε το πρόγραμμα σε τη παραπάνω μέθοδο προσομοίωσης στη γλώσσα C χρησιμοποιώντας τον compiler Dev C++. Το πρόγραμμα είναι από τις numerical recipes in C++ for finance προσαρμοσμένο στο Compiler της Dev C++ και θα αναλυθεί βήμα προς βήμα

```
#include <cstdlib>
#include <iostream>
#include <cmath>

using namespace std;
```

Η λειτουργία της παρακάτω συνάρτησης random_uniform είναι ότι εξάγει τυχαία μεταβλητή από την ομοιόμορφη κατανομή στο διάστημα [0,1]. Για να συμβεί αυτό χρησιμοποιήθηκε η συνάρτηση rand() η οποία εμπεριέχεται στη βιβλιοθήκη cmath την οποία την έχουμε συμπεριλάβει στο πρόγραμμα με την εντολή #include <cmath>. Η rand() επιστρέφει ένα αριθμό στο διάστημα [0,RAND_MAX] όπου για το compiler της Dev C++ ο αριθμός RAND_MAX είναι 32767. Διαιρώντας τη rand() με το RAND_MAX λαμβάνουμε ένα πραγματικό αριθμό στο διάστημα [0,1]

```
double random_uniform(void){
   return double(rand())/double(RAND_MAX);
};
```

Οπότε αφού πλέον μπορούμε να παράγουμε τυχαία μεταβλητές στο διάστημα [0,1] μπορούμε να χρησιμοποιήσουμε την polar method για να παράγουμε τυχαίες μεταβλητές από την κανονική κατανομή. Αυτό το πετυχαίνουμε την συνάρτηση normal random

```
double random_normal(void){
   double U1, U2, V1, V2;
   double S=2;
   while (S>=1) {
```

Σε αυτό το βήμα εκχωρούμε στης μεταβλητές U1 και U2 τους τυχαίους αριθμούς από την ομοιόμορφη κατανομή όπως βλέπουμε στου τύπους (mc1) και (mc2).

```
                  U1 = random_uniform();
                  U2 = random_uniform();
```

Στη συνέχεια υπολογίζουμε της μεταβλητές V1 και V2 σύμφωνα με τους τύπους (mc3) και (mc4)

```
                  V1 = 2.0*U1-1.0;
                  V2 = 2.0*U2-1.0;
```

Σύμφωνα με τον τύπο (mc5) η μεταβλητή S πρέπει να είνα μικρότερη του 1 οπότε τρέχει το πρόγραμμα και οι μεταβλητές V1 και V2 βγάλουν τιμή S σύμφωνα με τον παρακάτω τύπό μεγαλύτερη του 1 ο βρόγχος while ξανακάνει loop όταν S<1 τότε ικανοποιείται η polar method και εκχωρείται στην X1 η τιμή μιας τυποποιημένης κανονικά κατεμημένης τυχαίας μεταβλητής.

```
                  S = pow(V1,2)+pow(V2,2);
   };
   double X1=V1*sqrt((-2.0*log(S))/S);
   return X1;
};
```

Παρακάτω είναι η συνάρτηση που τιμολογεί το European Call Option

```
double option_price_call_european_simulated( const double& S,
                                             const double& K,
                                             const double& r,
                                             const double&
```

```
sigma,
                                                                   const double&
time,
                                                                   const int&
no_sims){
   double R = (r - 0.5 * pow(sigma,2))*time;
   double SD = sigma * sqrt(time);
   double sum_payoffs = 0.0;
   for (int n=1; n<=no_sims; n++) {
                 double S_T = S* exp(R + SD * random_normal());
                 sum_payoffs += max(0.0, S_T-K);
   };
   return exp(-r*time) * (sum_payoffs/double(no_sims));
};


int main(int argc, char *argv[])
{

                 cout    <<"The    price    of    The    European    Call    Option    is
"<<option_price_call_european_simulated(9,10,0.04,0.3,0.25,100000) <<endl;
   system("PAUSE");
   return EXIT_SUCCESS;
}
```

Η παραπάνω προσημείωση μπορεί να γίνει και για ένα European put option και αντικαθιστώντας στο παραπάνω πρόγραμμα όπου αντι για max(0.0,S_T-K) με max(0.0,K-S_T) οπότε έτσι θα λαμβάναμε σαν αποτέλεσμα τη τιμή του put Option. Φυσικά η μέθοδος της προσομοίωσης μπορεί να βελτιωθεί ώστε να λαμβάνουμε καλύτερα αποτελέσματα και αυτό μπορεί να γίνει με τις antithetic variates

Antithetic Variates

American Put Option

Η παρακάτω εξίσωση είναι η γνωστή μερική διαφορική εξίσωση των Black Scholes με τη διαφορά ότι περιέχει ένα επιπλέον όρο το q το οποίο είναι το μέρισμα της μετοχής

$$\frac{\partial f}{\partial t}+(r-q)\cdot S\frac{\partial f}{\partial S}+\frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2=r\cdot f$$

Αυτή η μερική διαφορική εξίσωση θα επιλυθεί με αριθμητική μέθοδος και πιο συγκεκριμένα με περασμένες διαφορές . Στο πρώτο στάδιο αυτό που πρέπει να γίνει είναι να απλοποιηθεί η παραπάνω εξίσωση και αυτό θα επιτευχθεί κάνοντας τη μετατροπή $x=\ln(S)$ οπότε η BS PDE παίρνει την παρακάτω μορφή

$$\frac{\partial f}{\partial t}+(r-q)\frac{\partial f}{\partial x}+\frac{1}{2}\sigma^2\frac{\partial^2 f}{\partial x^2}=r\cdot f$$

και στο δεύτερο στάδιο είναι να απαλλαγούμε από τον δεξιό όρο $r\cdot f$ εφαρμόζοντας τον παρακάτω μετασχηματισμό

$$u(t,x)=e^{r(T-t)}\cdot f(e^x,t)$$

και $\mu=(r-q)$ πλέον έχουμε την παρακάτω μορφή

$$\frac{\partial u}{\partial t}+\mu\frac{\partial u}{\partial x}+\frac{1}{2}\sigma^2\frac{\partial^2 u}{\partial x^2}=0 \quad \text{(MBS)}$$

όπου *u* είναι η forward τιμή του δικαιώματος και η οποία εξαρτάται από τη μεταβλητή του χρόνου *t* και από την μεταβλητή x που είναι ο φυσικός λογάριθμος της τιμής της μετοχής. Οπότε η λύση της της PDE θα βρίσκεται στο χώρο (t,x) οπότε αυτό που ουσιαστικά θα γίνει είναι να δημιουργήσουμε ένα πλέγμα σε αυτό το χώρο όπου στον οριζόντια άξονα θα είναι η μεταβλητή του χρόνου t και στον κάθετο η λογαριθμική τιμή της μετοχής. Ο χρόνος *t* θα παίρνει τιμές $t_0,t_1,\ldots,t_n$ στον οριζόντιο άξονα ενώ η μεταβλητή x στον κάθετο άξονα θα παίρνει τιμές στο διάστημα $-\infty<x<+\infty$ όπου όμως θα φράσσεται από την *M* δηλαδή $x_{-M},\ldots,x_0,\ldots,x_M$.Οπότε ένα σημείο του πλέγματος θα το συμβολίζουμε με (t_i,x_j) και η τιμή του option σε αυτό του σημείο θα είναι $u(t_i,x_j)$ ή $u_{i,j}$. Από το σημείο (t_i,x_j) μπορούμε να φτάσουμε σε τέσσερα άλλα σημεία του πλέγματος τα οποία είναι το $t_i,x_j+\Delta x$ πρός τα επάνω u_{up} , προς τα δεξιά $t_i+\Delta t,x_j$ u_{rt} , προς τα κάτω $t_i,x_j-\Delta x$ u_{down} , προς τα αριστερά $t_i-\Delta t,x_j$ ως u_{lf} . Οπότε αυτό που θα γίνει είναι να υπολογίσουμε τις τιμές του option σε κάθε σημείο του πλέγματος και αλγόριθμος θα επιστρέψει την τιμή του στο σημείο $u_{0,0}$. Η λύση σε αυτό το χώρο θα βρεθεί λύνοντας αριθμιτικά BS PDE με explicit μέθοδο των πεπερασμένων διαφορών. Δηλαδή το επόμενο βήμα είναι να προσεγγίσουμε αριθμητικά της παραγώγους της PDE η παράγωγος ως προς το χώρο έχοντας κύριο σημείο το $u_{i+1,j}$ προσεγγίζεται

$$\frac{\partial u}{\partial x}(t_{i+1},x_j)\approx\frac{u_{up}-u_{down}}{2\Delta x}\approx\frac{u_{i+1,j+1}-u_{i+1,j-1}}{2\Delta x}$$

η δεύτερη παράγωγος

$$\frac{\partial^2 u}{\partial x^2}(t_{i+1},x_j)\approx\frac{u_{up}-2\cdot u+u_{down}}{(\Delta x)^2}\approx\frac{u_{i+1,j+1}-2\cdot u_{i+1,j}+u_{i+1,j-1}}{(\Delta x)^2}$$

επειδή η μέθοδος είναι explicit η παράγωγος ως προς το χρόνο θα είναι

$$\frac{\partial u}{\partial t}(t_{i+1}, x_j) \approx \frac{u - u_{\alpha\rho\iota\sigma\tau\varepsilon\rho\acute{\alpha}}}{\Delta t} \approx \frac{u_{i+1,j} - u_{i,j}}{\Delta t}$$

και αντικαθιστώντας τις προσεγγίσεις των παραγώγων στη (MBS)

$$\frac{u_{i+1,j} - u_{i,j}}{\Delta t} + \mu \frac{u_{i+1,j+1} - u_{i+1,j-1}}{2\,\Delta x} + \frac{1}{2}\sigma^2 \frac{u_{i+1,j+1} - 2\cdot u_{i+1,j} + u_{i+1,j-1}}{(\Delta x)^2} = 0$$

και λύνοντας την παραπάνω ως προς την $u_{i,j}$

$$u_{i,j} = a\cdot u_{i+1,j+1} + b\cdot u_{i+1,j} + b\cdot u_{i+1,1} + c\,u_{i+1,j-1}$$

όπου

$$a = \frac{\sigma^2 \Delta t}{2\,\Delta x^2} + \frac{\mu \Delta t}{2\,\Delta x}$$

$$b = 1 - \frac{\sigma^2 \Delta t}{\Delta x^2}$$

$$c = \frac{\sigma^2 \Delta t}{2\,\Delta x^2} + \frac{\mu \Delta t}{2\,\Delta x}$$

Η παραπάνω μεθοδολογία για την τιμολόγηση του American Put Option θα τιμολογηθει με το παρακάτω πρόγραμμα

```
#include <cstdlib>
#include <iostream>
#include <cmath>


using namespace std;

double americaputexlicitdefferences(const double&  price,

const double&  strike,

const double&  rate,

const double&  sigma,

const double&  T,
```

```
const int& M,

const int& N,
                        const double& div
                        ){


   int i,j;
   double dt = T/N;
   double m = rate - div - 0.5*(sigma*sigma);
   double dx = sigma*sqrt(3*dt/2);
   double a, b, c;
   double P[150][150] = {0.0}; // stores option prices
   double S[150][150] = {0.0}; // stores asset prices

   a = (sigma*sigma*dt)/(2*dx*dx) + (m*dt)/(2*dx);
   b = 1.0 - (sigma*sigma*dt)/(dx*dx);
   c = (sigma*sigma*dt)/(2*dx*dx) - (m*dt)/(2*dx);



   for (j = -M; j <= M; j++)
   {
      S[N][j] = price*exp(j*dx);
   }

   for (i = 0; i < N; i++)
   {
      P[i][0] = strike;
      P[i][M] = 0;
   }
   for (j = -M; j <= M; j++)
   {
      P[N][j] = max(strike - S[N][j],0.0);
   }
   for (j = -M; j <= M; j++)
   {
      P[N][j] = max(strike - S[N][j],0.0);
   }

   for (i = 0; i < N; i++)
   {


         P[i][-M] = strike;
         P[i][M] = max(0.0,strike - S[N][j]);



   }
   for (i = N-1; i >= 0; i--)
   {
      for (j = M-1; j >= -M; j--)
      {
```

```
            P[i][j] = a*P[i+1][j+1] + b*P[i+1][j] + c*P[i+1][j-1];
            P[i][j] = max(strike - S[N][j], P[i][j]);
        }
    }

return P[0][0];
}



int main(int argc, char *argv[])
{
    /*
    double S = 50.0;
    double K = 50.0;
    double r = 0.06;
    double sigma = 0.2;
    double time=1;
    int no_S_steps=5;
    int no_t_steps=4;
    double div=0.03;
    */
    double S = 8.0;
    double K = 10.0;
    double r = 0.1;
    double sigma = 0.4;
    double time=0.25;
    int no_S_steps=5;
    int no_t_steps=4;
    double div=0.0;

    cout << " explicit finite differences, american put price = ";
    cout << americaputexlicitdefferences(S,K,r,sigma,time,no_S_steps,no_t_steps,div)
    << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Look Back Options

Εκτός βέβαια από τα European Option υπάρχουν και διαφορετικά είδη δικαιωμάτων όπως είναι τα Look Back Option τα οποία ανήκουν στην κατηγορία των Exotic Option. Το payoff ενός European Call Option είναι είναι της υποκείμενης μετοχής στη λήξη του μείων το K strike η διαφορά του Look Back Call Option floating Strike είναι για μια σταθερή τιμή εκτέλεσης K η οποία είνα σταθερή και καθορίζεται στην αρχή του συμβολαίου η τιμή εκτέλεσεις είναι η μικρότερη τιμή που παίρνει η μετοχή σε όλη τη χρονική διάρκεια του συμβολαίου.

Lookback Call Option (floating Strike)

$$C_T^{LB} = max(S(T) - \min_{t \leq T} S(t), 0)$$

Lookback Put Option (floating Strike)

$$P_T^{LB} = max(\max_{t \leq T} S(t) - S(T), 0)$$

Η τιμολόγηση ενός Lookback Call Option μπορεί να γίνει με την μέθοδο Monte Carlo και πιο συγκεκριμένα η τιμολόγηση για ένα Lookback Call Option μπορεί να γίνει με το παρακάτω πρόγραμμα

```
#include <cstdlib>
#include <iostream>
#include <cmath>
#include <vector>


using namespace std;


double random_uniform(void){
    return double(rand())/double(RAND_MAX); // this uses the C library random
number generator.
};

double random_normal(void){
   double U1, U2, V1, V2;
   double S=2;
   while (S>=1) {
            U1 = random_uniform();
            U2 = random_uniform();
            V1 = 2.0*U1-1.0;
            V2 = 2.0*U2-1.0;
            S = pow(V1,2)+pow(V2,2);
   };
   double X1=V1*sqrt((-2.0*log(S))/S);
   return X1;
};

vector<double> simulate_lognormally_distributed_sequence(const double& S,
                                                          const double& r,
                                                            const   double&
sigma,
                                                            const   double&
time,  // time to final date
                                                              const     int&
no_steps){  // number of steps
   vector<double> prices(no_steps);
   double delta_t = time/no_steps;
```

```
    double R = (r-0.5*pow(sigma,2))*delta_t;
    double SD = sigma * sqrt(delta_t);
    double S_t = S;                 // initialize at current price
    for (int i=0;  i<no_steps; ++i) {
                 S_t = S_t * exp(R + SD * random_normal());
    prices[i]=S_t;
    };
    return prices;
};

double payoff_lookback_call(const vector<double>& prices) {
   double m = *min_element(prices.begin(),prices.end());
   return prices.back()-m; // always positive or zero
};

double payoff_lookback_put(const vector<double>& prices) {
    double m = *max_element(prices.begin(),prices.end());
    return m-prices.back(); // max is always larger or equal.
};


int main(int argc, char *argv[])
{
    double lookback_call_option, lookback_put_option;

                                                    vector<double>stock_prices            =
simulate_lognormally_distributed_sequence(9,0.04,0.3,0.25,90);

    lookback_call_option = payoff_lookback_call(stock_prices);
    lookback_put_option = payoff_lookback_put(stock_prices);

    cout << "the look back call option price is" << lookback_call_option << endl;
    cout << "the look back call option price is" << lookback_put_option << endl;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```