All the matlab codes engineering essay

Engineering

All the Matlab codes which we have used to work put the results are given.

A1. 1 European Call

function call_price= EuroCall(S, E, r, sigma, time)S = 10; E = 5; time = 0. 25; r = 0. 06; sigma = 0. 3; time_sqrt = sqrt(time); d1 = (log(S/E)+r*time)/(sigma*time_sqrt)+0. 5*sigma*time_sqrt; d2 = d1-(sigma*time_sqrt); call_price = S*normcdf(d1)-E*exp(-r*time)*normcdf(d2);

A1. 2 European Put

function put_price= EuroPut(S, E, r, sigma, time)S = 10; E = 5; time = 0. 25; r = 0. 06; sigma = 0. 3; d1=(log(S/E)+(r+sigma^2/2)*time)/(sigma*sqrt(time)); d2= d1-sigma*sqrt(time); put_price= E*exp(-r*time)*normcdf(-d2)-S*normcdf(-d1);

A1. 3 European call with dividends

function call_price= EuroCallWithDivs(S, E, r, sigma, time, D)S = 30; E = 20; time = 0. 5; r = 0. 06; sigma = 0. 3; D= 0. 04; time_sqrt = sqrt(time); d1 = (log(S/E)+(r-D)*time)/(sigma*time_sqrt)+0. 5*sigma*time_sqrt; d2 = d1-(sigma*time_sqrt); call_price= S.*exp(-D.*time).*normcdf(d1, 0, 1)-E.*exp(-r.*time).*normcdf(d2, 0, 1);

A1. 4 European put with dividends

function put_price= EuroPutWithDivs(S, E, r, sigma, time, D)S = 30; E = 20; time = 0. 5; r = 0. 06; sigma = 0. 3; D= 0. 04; d1=(log(S/E)+((r-D) +sigma^2/2)*time)/(sigma*sqrt(time)); d2= d1-sigma*sqrt(time); put_price=-S.*exp(-D.*time).*normcdf(-d1, 0, 1)+E.*exp(-r.*time).*normcdf(-d2, 0, 1);

A2 Finite Difference Methods

A2. 1 Explicit Method

function AmerPutPrice= AmerPutEM(S0, E, r, sigma, T, M, N)T = 0. 5; S0 = 30; E = 20; sigma = 0. 3; r = 0. 06; N = 2430; M = 2430; dt = T/N; mu = r - sigma^2/2; dx = sigma*sqrt(3*dt); pu = dt*(sigma^2/2/dx^2 + mu/2/dx); pm = 1 - dt*sigma^2/dx^2 - r*dt; pd = dt*(sigma^2/2/dx^2 - mu/2/dx); S = zeros(2*M+1, N+1); V = zeros(2*M+1, N+1); I = [0: 1: N]; J = [M:-1:-M]'; S = S0*exp(J.*dx); V(:, end) = max(E - S, 0); for j= N:-1: 1for i= 2: 2*MV(i, j) = pu*V(i-1, j+1) + pm*V(i, j+1) + pd*V(i+1, j+1); endV(2*M+1, j) = V(2*M, j) + (S(2*M) - S(2*M+1)); V(1, j) = V(2, j); for i= 1: 2*M+1V(i, j) = max(E - S(i), V(i, j)); endendAmerPutPrice = V(M+1, 1)

A2. 2a Fully Implicit Method

function ImplicitPrice= AmerPutIM(S0, E, r, sigma, T, M, N)T = 0. 5; S0 = 30; E = 20; sigma = 0. 3; r = 0. 06; PutCall = 'P'; EuroAmer = 'A'; N = 2430; M = 2430; dt = T/N; mu = r - sigma^2/2; dx = sigma*sqrt(3*dt); pu = -dt/2*(sigma^2/dx^2 + mu/dx); pm = 1 + dt*sigma^2/dx^2 + r*dt; pd = -dt/2*(sigma^2/dx^2 - mu/dx); S = zeros(2*M+1, N+1); V = zeros(2*M+1, N+1); J = [M:-1:-M]'; S = S0*exp(J.*dx); clear Jif strcmp(PutCall,'P')V(:, end) = max(E - S, 0); elseV(:, end) = max(S - E, 0); endif strcmp(PutCall,'P')lambda_L = max(0, E - S(2*M+1)); lambda_U = 0; elseif strcmp(PutCall,'C')lambda_L = 0; lambda_U = max(0, S(1) - E); endfor j= N:-1: 1C = [lambda_U; V(2: 2*M, j+1); lambda_L]; V(:, j) = SolveTriangular(C, pu, pm, pd, lambda_L, lambda_U); if strcmp(EuroAmer,'A')for i= 1: 2*M+1; switch PutCallcase 'P'V(i, j) = max(V(i, j), E - S(i)); case 'C'V(i, j) = max(V(i, j), S(i) - E); endendendendclear i jImplicitPrice = V(M+1, 1);

A2. 2b Fully Implicit Method

function y = SolveTriangular(C, pu, pm, pd, lambda_L, lambda_U); M = (length(C)-1)/2; pmp(2*M) = pm + pd; pp(2*M) = C(2*M) + pd*lambda_L; for j= 2*M-1:-1: 2pmp(j) = pm - pu*pd/pmp(j+1); pp(j) = C(j) - pp(j+1)*pd/pmp(j+1); endy(1) = (pp(2) + pmp(2)*lambda_U)/(pu + pmp(2)); y(2) = y(1) - lambda_U; for j= 3: 2*M; y(j) = (pp(j) - pu*y(j-1))/pmp(j); endy(2*M+1) = y(2*M) - lambda_L;

A2. 3 Crank Nicolson Method

function AmerPutPrice= AmerPutCN(S0, E, r, sigma, T, M, N)T = 0. 5; S0 = 30; E = 20; sigma= 0. 3; r = 0. 06; N = 2430; M = 2430; dt = T/N; mu = r-sigma^2/2; dx = sigma*sqrt(3*dt); pu = -1/4*dt*(sigma^2/dx^2 + mu/dx); pm = 1 + dt*sigma^2/2/dx^2 + r*dt/2; pd = -1/4*dt*(sigma^2/dx^2 - mu/dx); S = zeros(2*M+1, N+1); V = zeros(2*M+1, N+1); J = [M:-1:-M]'; S = S0*exp(J.*dx); V(:, end) = max(E - S, 0); pmp = zeros(2*M+1, N+1); pp = zeros(2*M+1, N+1); C = zeros(2*M+1, N+1); for j= N+1:-1: 2pmp(2*M, j) = pd+pm; for i= 2*M-1:-1: 2pmp(i, j) = pm - pd/pmp(i+1, j)*pu; endendlambda_L = S(2*M+1) - S(2*M); lambda_U = 0; for j= N+1:-1: 2; for i= 2*M:-1: 2if i== 2*Mpp(i, j) = -pu*V(i-1, j) - (pm-2)*V(i, j) - pd*V(i+1, j) + pd*lambda_L; elsepp(i, j) = -pu*V(i-1, j) - (pm-2)*V(i, j) - pd*V(i+1, j) - pd/pmp(i+1, j)*pp(i+1, j); endendj= j-1; for i= 1: 2*M+1if i== 1C(i, j) = (pp(i+1, j+1) + pmp(i+1, j+1)*lambda_U) / (pmp(i+1, j+1) + pu); V(i, j) = max(E - S(i), C(i, j)); elseif i <2*M+1C(i, j) = (pp(i, j+1) - pu*C(i-1, j))/pmp(i, j+1); V(i, j) = max(E - S(i), C(i, j)); elseC(i, j) = C(i-1, j) - lambda_L; V(i, j) = max(E - S(i), C(i, j)); endendj= j+1; endVAmerPutPrice = V(M+1, 1)

A3 Binomial Method

A3. 1 Binomial Method European Call

function call_price= european_call_bin(S, E, r, sigma, t, M)S= 45; E= 40; r= 0. 1; sigma= 0. 25; t= 0. 5; M= 10; R = exp(r*(t/M)); Rinv = 1. 0/R; u = exp(sigma*sqrt(t/M)); uu = u*u; d = 1. 0/u; p_up = (R-d)/(u-d); p_down = 1. 0-p_up; prices= zeros(M+1, 1); prices(1) = S*(d^M); for (i= 2: (M+1))prices(i) = uu*prices(i-1); endcall_values= zeros(M+1, 1); call_values = max(0, (prices-E)); for (M= M:-1: 1)for (i= 1: 1:(M))call_values(i) = (p_up*call_values(i+1)+p_down*call_values(i))*Rinv; endendcall_price = call_values(1); For the European Put you just have to replace max(0, (prices-E)) with max(0, (E-prices))

A3. 2 Binomial Method American Call

function call_price= american_call_bin(S, E, r, sigma, t, M)S= 45; E= 40; r= 0. 1; sigma= 0. 25; t= 0. 5; M= 10; R = exp(r*(t/M)); Rinv = 1. 0/R; u = exp(sigma*sqrt(t/M)); d = 1/u; p_up = (R-d)/(u-d); p_down = 1-p_up; prices = zeros(M+1); prices(1) = S*(d^M); uu = u*u; for i= 2: M+1prices(i) = uu*prices(i-1); endcall_values = max(0, (prices-E)); for M= M:-1: 1for i= 1: M+1call_values(i) = (p_up*call_values(i+1)+p_down*call_values(i))*Rinv; prices(i) = d*prices(i+1); call_values(i) = max(call_values(i), prices(i)-E); endendcall_price= call_values(1); For the American Put you just have to replace max(0, (prices-E)) with max(0, (E-prices)) and also replace max(call_values(i), prices(i)-E) with max(call_values(i), E-prices(i)).

A3. 3 Binomial Method for European and American Options with Dividends

function BioWoithDivs= AmerEuroDivsBin(S0, E, r, sigma, T, M, N, D)BSCall = inline('s*exp(-D*T)*normcdf((log(s/E) + (r-D+sigma^2/2)*T)/sigma/sqrt(T)) - E*exp(-r*T)*normcdf((log(s/E) + (r-D+sigma^2/2)*T)/sigma/sqrt(T) - sigma*sqrt(T))',...'s','E','r','D','sigma','T'); BSPut = inline('E*exp(-r*T)*normcdf(-(log(s/E) + (r-D+sigma^2/2)*T)/sigma/sqrt(T) + sigma*sqrt(T)) - s*exp(-D*T)*normcdf(-(log(s/E) + (r-D+sigma^2/2)*T)/sigma/sqrt(T))',...'s','E','r','D','sigma','T'); S0 = 30; r = 0. 06; D = 0. 04; sigma = 0. 3; N = 10; E = 20; T = 0. 5; PutCall = 'P'; EuroAmer = 'A'; dt = T/N; u = exp(sigma*sqrt(dt)); d = 1/u; p = (exp((r-D)*dt)-d)/(u-d); S = zeros(2*N+1, N+1); S(N+1, 1) = S0; for j= 2: N+1for i= N-j+2: 2: N+jS(i, j) = S0*u^(N+1-i); endendV = zeros(2*N+1, N+1); switch PutCallcase 'C'V(:, N+1) = max(S(:, N+1) - E, 0); case 'P'V(:, N+1) = max(E - S(:, N+1), 0); endfor j= N:-1: 1for i= N-j+2: 2: N+jswitch EuroAmercase 'A'if strcmp(PutCall, 'C')V(i, j) = max(S(i, j) - E, exp(-r*dt)*(p*V(i-1, j+1) + (1-p)*V(i+1, j+1))); elseV(i, j) = max(E - S(i, j), exp(-r*dt)*(p*V(i-1, j+1) + (1-p)*V(i+1, j+1))); endcase 'E'V(i, j) = exp(-r*dt)*(p*V(i-1, j+1) + (1-p)*V(i+1, j+1)); endendendTreePrice = V(N+1, 1)if strcmp(PutCall, 'C')EuroPrice = BSCall(S0, E, r, D, sigma, T)elseEuroPrice = BSPut(S0, E, r, D, sigma, T)endif strcmp(PutCall, 'P') & strcmp(EuroAmer, 'A')ExercisePremium = TreePrice - EuroPriceend

Project Plan

Weeks 3-8During week 3 I will be given my dissertation topic and know what dissertation I will be doing. I will try and see my supervisor on a weekly basis

so that I have regular feedback on my dissertation. Early during this period I will do some background research on, Options, Black-Scholes and Asset price model. To do this research I will need to look for books, which will help me for my background information and during the dissertation. During week 8, the background, project plan and Gantt chart will be due in. Weeks 9-12During this 4 week period I will do more research on Options, specifically American style options and look at upper and lower boundary conditions for both American and European style options, and do the write up for these topics. Weeks 13-14During these two weeks, which are holidays, I will start looking into detail the Binomial Model, and see what effect it has on valuing options. Also I will start doing research into Finite Difference Methods. Weeks 15-19I will start doing in depth research in to Finite Difference Methods. Also during these 5 weeks I will start implementing on Matlab. Weeks 20-24In these weeks, I will analyse the results I get from Matlab and complete my final write up. The dissertation is due in on week 25, but I??? ve left an extra week for anything I??? ve left out or any final checks. Weeks 25-29Week 29 is when the oral presentation is, so in these weeks I will prepare my presentation and rehearse for it, also go over my dissertation as I will be getting asked questions.