Final Report

Finance PDE Project

Math 458

Professor Ricardo Mancera

Yuchen Shi

Wei Zhang

Ruiqi Wang

11/30/2019

Financial Background:

a). Put Option vs Call Option:

- Call options are financial contracts that give the option buyer the right, but not the obligation, to **buy** a stock, bond, commodity or other asset or instrument at a specified price (strike price) within a specific time period. A call buyer profits when the underlying asset **increases** in price.

- Put option are financial contract that give the option buyer the right, but not the obligation, to **sell** a stock, bond, commodity or other asset or instrument at a specified price (strike price) within a specific time period. A put buyer profits when the underlying asset **decreases** in price.

b). American Option vs European Option:

- American options allow holders to exercise the option rights at any time before and including the day of expiration.

- European options allow execution only on the day of expiration.

Objective:

Price a 5-month American put option, when the stock price $S0$ = \$50, the strike price K = \$50, the risk-free interest rate r = 0.10 and the volatility σ = 0.4.

a) Use implicit finite differences method to solve Black-Scholes PDE directly

The Black-Sholes PDE is:

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

Where f is the price of the put option, and is a function of time t and stock price S.

Specifically,

$$f(S,T) = \max(K-S, 0)$$

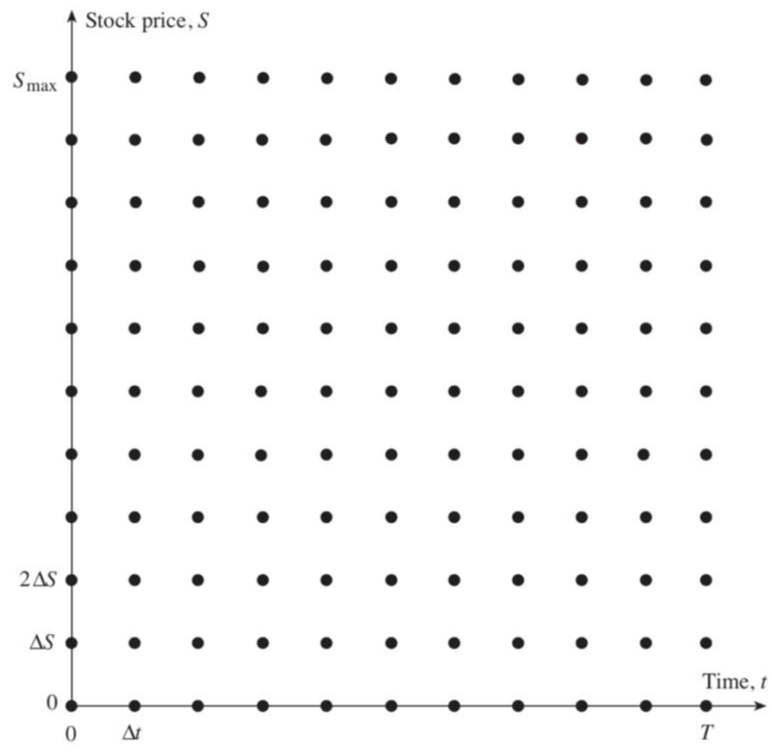
Since it is a put option, if the stock price is smaller than the strike price, investors would exercise the option, and sell the stock at the strike price to option writers, so the value of the option would be K (strike price) – S (stock price). If the stock price is larger than the strike price, investors would not exercise the option, so the value of the option would be 0.

In finite difference methods, we value a derivative by solving the differential equation that the derivative satisfies. The differential equation is converted into a set of difference equations, and the difference equations are solved iteratively.

Suppose the life of the option is T. We divide this into N equally spaced intervals of length of length dt=T/N. There will be a total of N+1 times. We choose N to be 10 and T equals to 0.4167 (5 months, 5/12) for this project.

Suppose that Smax is a stock price sufficiently high that, when it is reached, the put option virtually has no value. We divide Smax into M equally spaced stock prices. The difference between each stock price is ds=Smax/M. There will be a total of M+1 stock prices. We choose M to be 20 and Smax to be 100 for this project.

The time points and stock price points define a grid consisting of a total of (M+1) * (N+1) points, as shown in figure below.



The point (i, j) on the grid is the point that corresponds to time i * dt and stock price j * ds. We will use the variable f(i,j) to denote the value of the option at the point (i,j).

Implicit Finite Difference Method

For each point on the grid, we have:

$$\frac{\partial f}{\partial S} = \frac{f_{i,j+1} - f_{i,j}}{\Delta S}$$

And

$$\frac{\partial f}{\partial S} = \frac{f_{i,j} - f_{i,j-1}}{\Delta S}$$

In order to have a better approximation, we use the a more symmetric form by averaging the two:

$$\frac{\partial f}{\partial S} = \frac{f_{i,j+1} - f_{i,j-1}}{2\,\Delta S}$$

We also have

$$\frac{\partial f}{\partial t} = \frac{f_{i+1,j} - f_{i,j}}{\Delta t}$$

And

$$\frac{\partial^2 f}{\partial S^2} = \left(\frac{f_{i,j+1} - f_{i,j}}{\Delta S} - \frac{f_{i,j} - f_{i,j-1}}{\Delta S}\right) \Big/ \Delta S$$

$$\frac{\partial^2 f}{\partial S^2} = \frac{f_{i,j+1} + f_{i,j-1} - 2 f_{i,j}}{\Delta S^2}$$

The second derivative is just another finite difference approximation of the first derivative.

Substituting these equations into the Black-Sholes PDE, we would get

$$\frac{f_{i+1,j} - f_{i,j}}{\Delta t} + (r-q)j\,\Delta S\,\frac{f_{i,j+1} - f_{i,j-1}}{2\,\Delta S} + \tfrac{1}{2}\sigma^2 j^2 \Delta S^2\,\frac{f_{i,j+1} + f_{i,j-1} - 2f_{i,j}}{\Delta S^2} = r f_{i,j}$$

Rearranging terms to get the variable at time i to one side and variables at time i+1 to the other side, we would get:

$$a_j f_{i,j-1} + b_j f_{i,j} + c_j f_{i,j+1} = f_{i+1,j}$$

$$a_j = \tfrac{1}{2}(r-q)j\,\Delta t - \tfrac{1}{2}\sigma^2 j^2 \Delta t$$

$$b_j = 1 + \sigma^2 j^2 \Delta t + r\,\Delta t$$

$$c_j = -\tfrac{1}{2}(r-q)j\,\Delta t - \tfrac{1}{2}\sigma^2 j^2 \Delta t$$

Where j = 1, 2 … M-1 and i = 0, 1 … N-1

a_j, b_j and c_j could all be calculated.

Now, we have a system of M-1 equations to solve for option price at time i from the option price at time i+1.

In matrix form, this corresponds to a tridiagonal matrix, where the main diagonal is the b_j, the diagonal below the main diagonal is the a_j, and the diagonal above the main diagonal is the c_j. The right hand side is just f(i+1,j+1), except for the first element, which is -a_1 * f(i,0) and the last element, which is $-c_{M-1}$ * f(i,M).

The matrix is not changing, and its value for this project is shown here:

1.0108	-0.005	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-0.009	1.0308	-0.018	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-0.024	1.0642	-0.036	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	-0.045	1.1108	-0.062	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	-0.073	1.1708	-0.094	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	-0.108	1.2442	-0.133	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	-0.149	1.3308	-0.178	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	-0.197	1.4308	-0.23	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	-0.251	1.5442	-0.289	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	-0.313	1.6708	-0.354	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	-0.38	1.8108	-0.426	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	-0.455	1.9642	-0.505	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	-0.536	2.1308	-0.59	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	-0.624	2.3108	-0.683	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	-0.719	2.5042	-0.781	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	-0.82	2.7108	-0.887	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-0.928	2.9308	-0.999	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1.043	3.1642	-1.118
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1.164	3.4108

Which is a 19*19 matrix, since we choose M equals to 20.

Initial condition:

Since at the time of expiration (when i = N), we know the value of the put option,

$$f_{N,j} = \max(K - j\,\Delta S,\ 0), \quad j = 0, 1, \ldots, M$$

Boundary conditions:

We also know that at any time, if the stock price is zero, the option price would just be K.

If the stock price reaches the Smax we defined above, the option price would just be 0.

$$f_{i,0} = K, \quad i = 0, 1, \ldots, N$$

$$f_{i,M} = 0, \quad i = 0, 1, \ldots, N$$

After we set up the initial and boundary conditions, the last column, the top row and the last row of the grid all have values. We could then use the system of equations to solve for option prices at each point of the grid backward, starting with time i = N-1. Iteratively, we could calculate the value of option price today (i=0).

Since it is an American put option, option holders could exercise the option earlier as they willingness. When we are iterating backward, we also need to check whether early exercise is optimal. At each time i = N-1, N-2 … 0, if the value of the option is less than the value that it could have if it is exercised today, early exercise is optimal. We should set the price to the price as it would be if it is exercised today.

Finally, given the stock price today, we could be able to estimate the price of a American put option with the given characteristics.

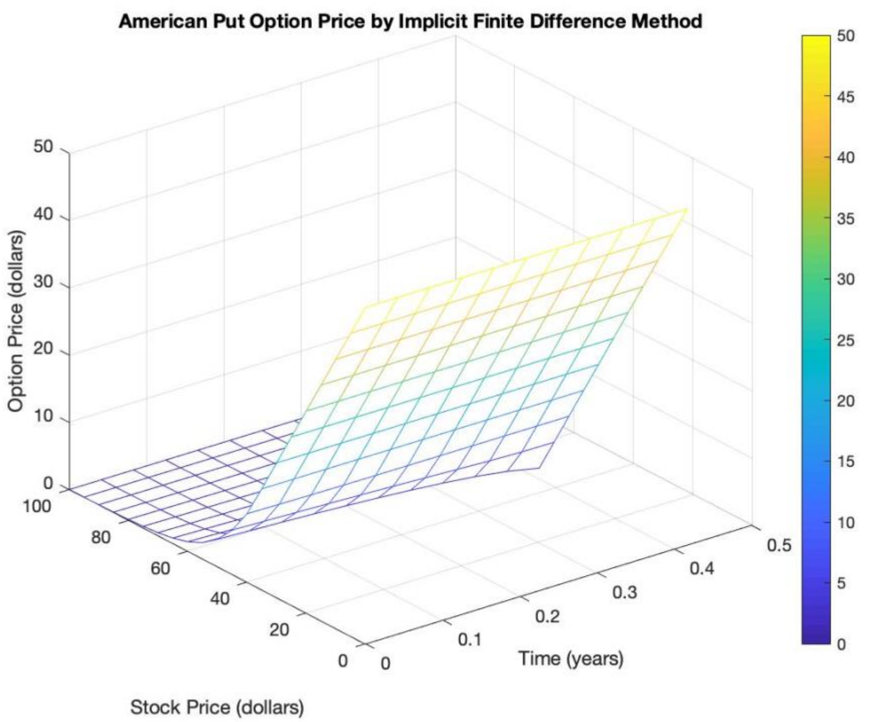
Result:

The table below shows the result of using implicit finite difference method to price an American put option. The grid is constructed at $5 stock price intervals between $0 and $100 and at half month time intervals throughout the life of the option.

		Time to Maturity (months)										
		5	4.5	4	3.5	3	2.5	2	1.5	1	0.5	0
Stock Prices (dollars)	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	95	0.02	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	90	0.05	0.04	0.03	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00
	85	0.09	0.07	0.05	0.03	0.02	0.01	0.01	0.00	0.00	0.00	0.00
	80	0.16	0.12	0.09	0.07	0.04	0.03	0.02	0.01	0.00	0.00	0.00
	75	0.27	0.22	0.17	0.13	0.09	0.06	0.03	0.02	0.01	0.00	0.00
	70	0.47	0.39	0.32	0.25	0.18	0.13	0.08	0.04	0.02	0.00	0.00
	65	0.82	0.71	0.60	0.49	0.38	0.28	0.19	0.11	0.05	0.02	0.00
	60	1.42	1.27	1.11	0.95	0.78	0.62	0.45	0.30	0.16	0.05	0.00
	55	2.43	2.24	2.05	1.83	1.61	1.36	1.09	0.81	0.51	0.22	0.00
	50	4.07	3.88	3.67	3.45	3.19	2.91	2.57	2.17	1.66	0.99	0.00
	45	6.58	6.44	6.29	6.13	5.96	5.77	5.57	5.36	5.17	5.02	5.00
	40	10.15	10.10	10.05	10.01	10.00	10.00	10.00	10.00	10.00	10.00	10.00
	35	15.00	15.00	15.00	15.00	15.00	15.00	15.00	15.00	15.00	15.00	15.00
	30	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
	25	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00
	20	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00
	15	35.00	35.00	35.00	35.00	35.00	35.00	35.00	35.00	35.00	35.00	35.00
	10	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00
	5	45.00	45.00	45.00	45.00	45.00	45.00	45.00	45.00	45.00	45.00	45.00
	0	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

From the table, we could see that if the stock price is 50 today, the price of an American put option is $4.07.

A 3D plot of the option price against time and stock prices is shown here:



b) Modify your program in part a) to find the price of a European put option with the same data as above. Compare your results with the ones obtained using B-S formulas.

For a European put option, the only difference is that the investors could only exercise the option at the exact date of expiration. There is no early exercise. As a result, when we are working backward to solve the system of equations at each time i = N-1, N-2 … 0, we would not compare the value of the option with the value it would have if it is exercised today. And then the rest is the same as in the case of an American put option.

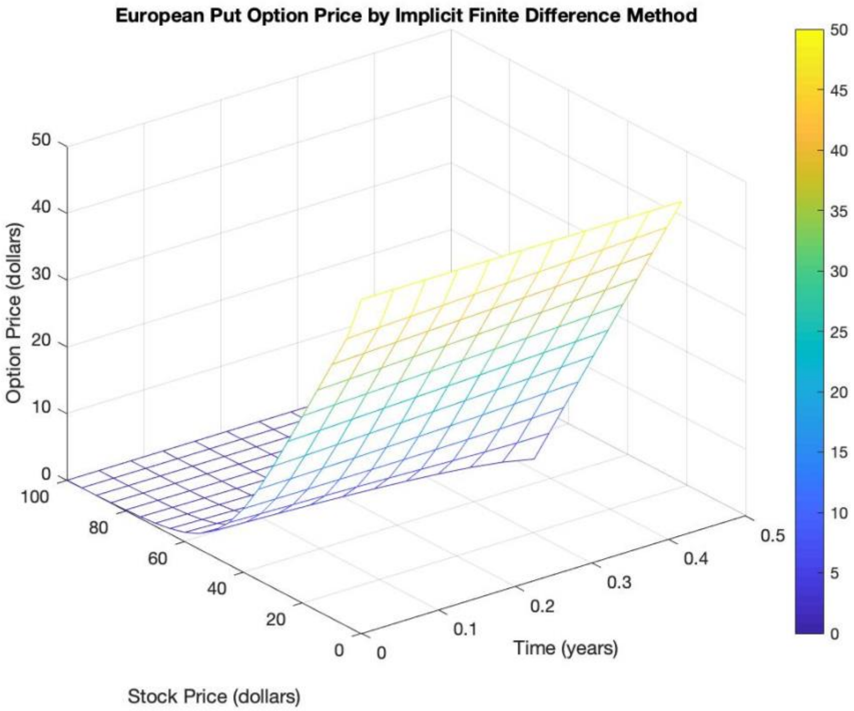
Result:

The table below shows the result of using implicit finite difference method to price a European put option. The grid is constructed at $5 stock price intervals between $0 and $100 and at half month time intervals throughout the life of the option.

	Time to Maturity (months)										
Stock Prices (dollars)	5	4.5	4	3.5	3	2.5	2	1.5	1	0.5	0
100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
95	0.02	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
90	0.05	0.04	0.03	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00
85	0.09	0.07	0.05	0.03	0.02	0.01	0.01	0.00	0.00	0.00	0.00
80	0.15	0.12	0.09	0.07	0.04	0.03	0.02	0.01	0.00	0.00	0.00
75	0.27	0.22	0.17	0.13	0.09	0.06	0.03	0.02	0.01	0.00	0.00
70	0.46	0.39	0.31	0.25	0.18	0.13	0.08	0.04	0.02	0.00	0.00
65	0.80	0.69	0.59	0.48	0.37	0.28	0.19	0.11	0.05	0.02	0.00
60	1.38	1.24	1.09	0.93	0.77	0.61	0.45	0.30	0.16	0.05	0.00
55	2.35	2.18	2.00	1.80	1.58	1.34	1.08	0.80	0.51	0.22	0.00
50	3.91	3.74	3.56	3.35	3.12	2.86	2.54	2.15	1.66	0.99	0.00
45	6.27	6.16	6.04	5.91	5.77	5.62	5.46	5.30	5.14	5.02	5.00
40	9.51	9.50	9.50	9.50	9.51	9.52	9.56	9.62	9.70	9.83	10.00
35	13.55	13.64	13.75	13.86	13.98	14.11	14.26	14.43	14.61	14.80	15.00
30	18.14	18.30	18.46	18.63	18.81	19.00	19.19	19.39	19.59	19.79	20.00
25	23.00	23.19	23.38	23.58	23.78	23.98	24.18	24.38	24.59	24.79	25.00
20	27.97	28.17	28.37	28.57	28.77	28.97	29.18	29.38	29.59	29.79	30.00
15	32.96	33.16	33.36	33.57	33.77	33.97	34.18	34.38	34.59	34.79	35.00
10	37.96	38.16	38.36	38.57	38.77	38.97	39.18	39.38	39.59	39.79	40.00
5	42.98	43.17	43.37	43.57	43.77	43.98	44.18	44.38	44.59	44.79	45.00
0	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

From the table, we could see that if the stock price is $50 today, the price of the European put option is $3.91.

A 3D plot of the option price against time and stock prices is shown here:



Since it is a European put option, we could also use the Black-Scholes-Merton formula to get an analytical valuation of its price. The formula is:

$$p = Ke^{-rT}N(-d_2) - S_0N(-d_1)$$

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

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

The function N(x) is the cumulative probability distribution function for a standardized normal distribution. P is the price of the European put option, and the remaining variables have the same definitions as before.

The result of European put option price we get from this B-S formula is $4.08. The difference between the numerical solution and analytical solution is 4.08-3.91=0.17.

c) **Solve the Heat Problem with Crank-Nicolson Method and compare with the exact solution**

1. **Set-up of the Heat Problem**

The heat problem is given by:

$$\frac{\partial f}{\partial t} = \frac{\partial^2 f}{\partial t^2}$$
$$f(x,0) = sin(\pi x), \quad 0 \le x \le 1$$
$$f(0,t) = 0, \quad 0 \le t \le 1$$
$$f(1,t) = 0, \quad 0 \le t \le 1$$

The exact solution for the heat equation is given by:

$$f(x,t) = e^{-\pi^2 t} sin(\pi x)$$

2. **Solving with Crank-Nicolson Method**

a) Discretize the time T and the Space S.

For the Space S, we define S=S_{max}/M. Then, we set M =200 (200 equally spaced position) and S_{max} =1, so we divide S into M + 1 nodes with step-size equal to 0.005.
For the time T, we define t=T_{max}/N. Then, we set N =200 (200 equally spaced time period) and T_{max} =1, so we divide t into N+1 nodes with step-size equal to 0.005.

The initial and boundary conditions are:

$$f(x,0) = sin(\pi x), \quad 0 \le x \le 1$$
$$f(0,t) = 0, \quad 0 \le t \le 1$$
$$f(1,t) = 0, \quad 0 \le t \le 1$$

b) Derive the Crank-Nicolson Method

The following equations are three finite difference methods, including forward Euler, backward Euler, and Crank-Nicolson method which is the average of the first two. In this part, we derive the simultaneous equation for solving the heat equation using Crank-Nicolson method.

i : space j : time

Forward Euler:

$$\frac{f_{i,j+1} - f_{i,j}}{\Delta t} = \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j+1}}{\Delta s^2}$$

Backward Euler:

$$\frac{f_{i,j+1} - f_{i,j}}{\Delta t} = \frac{f_{i+1,j+1} - 2f_{i,j+1} + f_{i-1,j+1}}{\Delta s^2}$$

Crank-Nicolson:

$$\frac{f_{i,j+1} - f_{i,j}}{\Delta t} = \frac{1}{2\Delta S^2}(f_{i+1,j} - 2f_{i,j} + f_{i-1,j} + f_{i+1,j+1} - 2f_{i,j+1} + f_{i-1,j+1})$$

Rearrange to make the terms with j and $j+1$ together:

$$-\frac{1}{2\Delta S^2}f_{i+1,j+1} + (\frac{1}{\Delta t} + \frac{2}{2\Delta S^2})f_{i,j+1} - \frac{1}{2\Delta S^2}f_{i-1,j+1} = \frac{1}{2\Delta S^2}f_{i+1,j} + (\frac{1}{\Delta t} + \frac{2}{2\Delta S^2})f_{i,j} + \frac{1}{2\Delta S^2}f_{i-1,j}$$

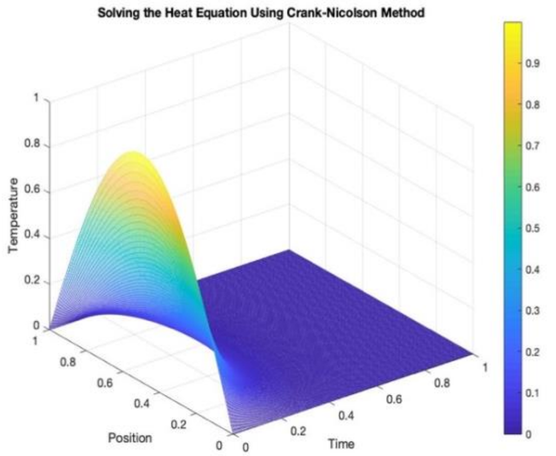
Multiple both sides by Δt, Set $r = \frac{\Delta t}{2\Delta S^2}$:

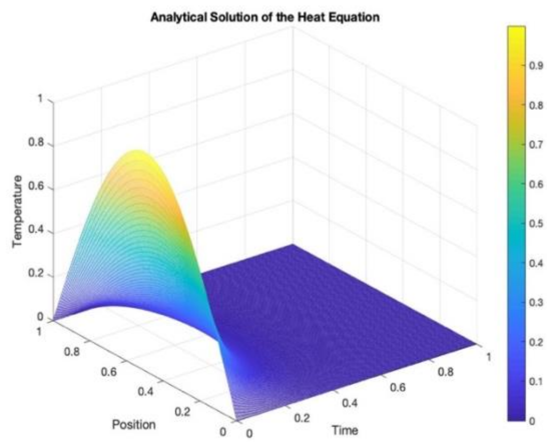
$$-rf_{i+1,j+1} + (1+2r)f_{i,j+1} - rf_{i-1,j+1} = rf_{i+1,j} + (1-2r)f_{i,j} + rf_{i-1,j}$$

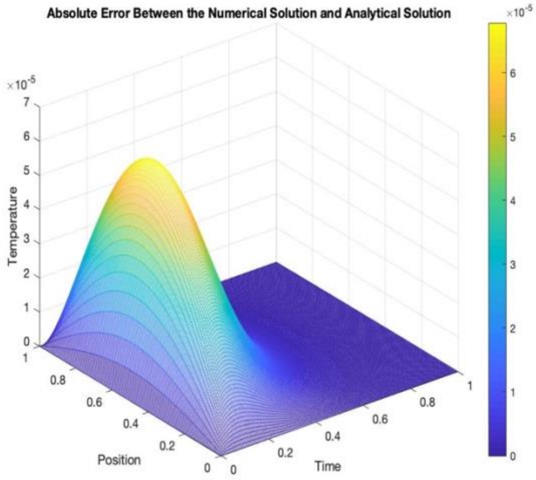
c) Solving the heat equation

The information for the right part of the equation is known by the initial condition, we could construct a tridiagonal matrix to solve the heat problem. We will go forwards, and use the simultaneous equations shown above to solve the value of f at time j+1 by its value at time j.

d) The graph for the numerical solution and analytical solution are shown as below.







d) Transform the B-S equation into the Heat Equation

Objectives: Use change of variable to transform B-S equation to the heat equation. The B-S equation is given by:

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}s^2S^2\frac{\partial^2 y}{\partial S^2} = rf$$

$$f(S,T) = \max(K - S, 0)$$

$$f(0,t) = K$$

Process: Variables→Derivatives→Replace

1. Introduce two new variables x and τ:

$$S = e^x,\ t = T - \tfrac{2\tau}{\sigma^2}$$

   Then transform the equation as:

$$f(S,t) = v(x,\tau) = v\left(\ln S, \frac{\sigma^2}{2}(T - t)\right)$$

2. Find the partial derivatives of functions of v in terms of x and τ:

$$\frac{\partial f}{\partial t} = -\frac{\sigma^2}{2}\frac{\partial v}{\partial \tau}$$

$$\frac{\partial f}{\partial S} = -\frac{1}{S}\frac{\partial v}{\partial x}$$

$$\frac{\partial^2 f}{\partial S^2} = -\frac{1}{S^2}\frac{\partial v}{\partial x} + \frac{1}{S^2}\frac{\partial^2 v}{\partial x^2}$$

3. Substitute into the B-S equation and simplify, we get:

$$\frac{\partial v}{\partial \tau} = \frac{\partial^2 v}{\partial x^2} + \left(\frac{2r}{\sigma^2} - 1\right)\frac{\partial v}{\partial x} - \frac{2r}{\sigma^2} v$$

4. Let $\rho = \frac{2r}{\sigma^2}$, then:

$$\frac{\partial v}{\partial \tau} = \frac{\partial^2 v}{\partial x^2} + (\rho - 1)\frac{\partial v}{\partial x} - \rho v, \;\; -\infty < x < \infty, 0 \le \tau \le \frac{\sigma^2}{2} T$$

Then the initial condition becomes:

$$f(e^x, T) = v(x, 0), -\infty < x < \infty$$

The boundary condition becomes:

$$f(0, t) = v(ln0, \tau)$$

5. Introduce variables α and β :

$$v(x, \tau) = e^{\alpha x + \beta \tau} u(x, \tau) = \phi u$$

6. Take partial derivative of u in terms of x and τ:

$$\frac{\partial v}{\partial \tau} = \beta \phi u + \phi \frac{\partial v}{\partial \tau}$$
$$\frac{\partial v}{\partial x} = \alpha \phi u + \phi \frac{\partial u}{\partial \tau}$$
$$\frac{\partial v}{\partial x} = \alpha^2 \phi u + 2\alpha\phi \frac{\partial u}{\partial x} + \phi \frac{\partial^2 u}{\partial x^2}$$

7. Set $\alpha = -\frac{1}{2}(\rho - 1) = \frac{\sigma^2 - 2r}{2\sigma^2}$ $, and\ \beta = -\frac{1}{4}(\rho + 1)^2 = -\left(\frac{\sigma^2 + 2r}{2\sigma^2}\right)^2$
8. Then we have:

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 v}{\partial x^2}, -\infty < x < \infty, 0 \le \tau \le \frac{\sigma^2}{2} T$$

$$f(x, 0) = v(x, 0) = \frac{1}{e^{-\alpha x}} u(x, 0)\ , -\infty < x < \infty$$

$$f(0, t) = v(ln0, \pi) = \phi u(ln0, \tau)$$

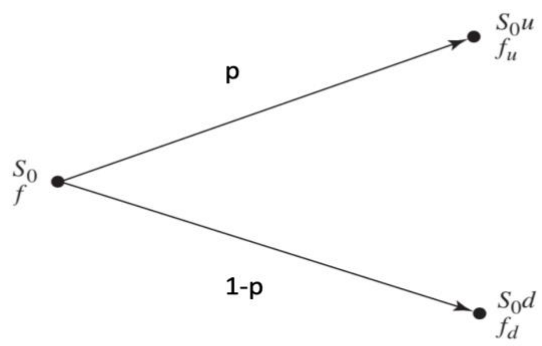
e) Use the binomial tree approach to estimate the price of an to price a 5 month American put option when the stock price So= \$50, the strike price K = \$50, the risk-free interest rate r = 0.10 and the volatility σ = 0.4

Binomial Tree Method:

Binomial tree method is another popular way to price an option. It relies on the risk neutral principle, which consists of two important procedures:

- Assume the expected return for all traded assets is the risk-free rate
- Value payoff from the derivative by calculating their expected values and discounting at the risk-free interest rate.

In this method, stock price follows a random walk. Within each time step Δ t, the probability of an up movement is denoted as p, and the probability of a down movement is $1 - p$.



So: current stock price

f: current option price

$So * u$: stock price if the price goes up, u > 1

$So * d$: stock price if the price goes down, 0<d<1

fu, fd: corresponding payoff from the option

Since we are working in a risk-natural world, $expected\ return\ =\ risk\ free\ rate$. We can determine parameter p, u ,and d by the formulas:

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

Suppose the lifetime of American put option has been divided into N subintervals with length Δ t. We use i to count the time interval where $0 \leq i \leq N$. j is the number of times the stock takes up movement. At time $i\Delta t$, we have $i + 1$ different stock prices. These are,

$$S_0 u^j d^{i-j}, \quad j = 0, 1, \ldots, i$$

Given the stock price, we could use the following 3 steps to we get the option values after N periods of time.

- Step 1: Given the stock price and strike price, we obtain the last column of option price by equation $P = \max\,(K - Sn, 0)$ .Step1: Given the stock price and strike price, we obtain the last column of option price by equation $P = \max(K - Sn, 0)$.

- Step 2: Work backwards through the tree.
  Refer to the Define fi, j as the value of the option at the (i, j) node

$$f_{i,j} = e^{-r\Delta t}[p f_{i+1,j+1} + (1-p) f_{i+1,j}]$$

- Step 3: Compare
  Since American put option can be exercised at any time during its lifetime, which is any node that is profitable. Therefore, we have to compare the value of early exercised and value calculated from binomial tree model. Take the one with higher profit.

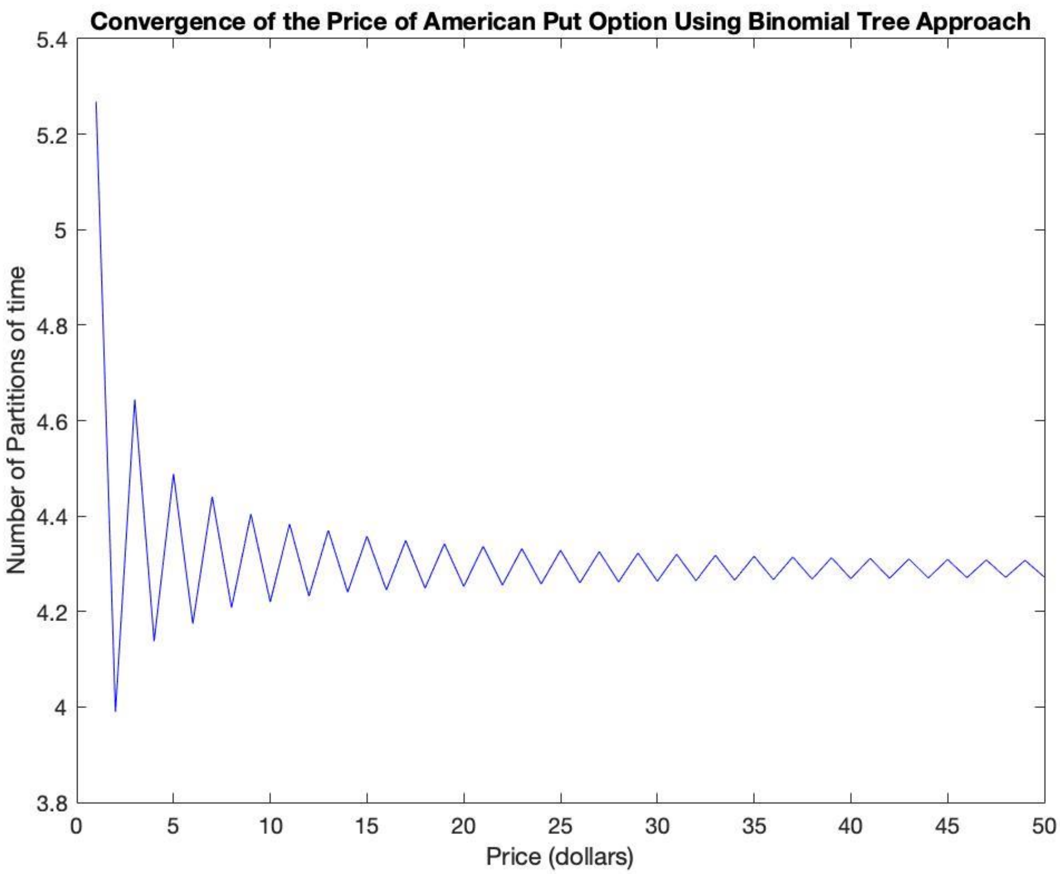
$$f_{i,j} = \max\{K - S_0 u^j d^{i-j},\ e^{-r\Delta t}[p f_{i+1,j+1} + (1-p) f_{i+1,j}]\}$$

As we keep working backwards, the first node is the price of the option, which is $4.49 as the grid shows.

		Time to Maturity (months)					
		5	4	3	2	1	0
Number of Up Movement	0	4.49	6.96	10.36	14.64	18.50	21.93
	1	0.00	2.16	3.77	6.38	10.31	14.64
	2	0.00	0.00	0.64	1.30	2.66	5.45
	3	0.00	0.00	0.00	0.00	0.00	0.00
	4	0.00	0.00	0.00	0.00	0.00	0.00
	5	0.00	0.00	0.00	0.00	0.00	0.00

f) Investigate the convergence rate of the binomial method as a function of the number of partitions of your time to expiration. Draw a graph.

By increasing N to 150, we obtain a plot which shows the relationship between price of the American put option and the number of partitions of time:



As a result, the put option price converge to \$4.28. From the plot, we can see when N increases to 50, the put price becomes stable.

g) Use the binomial tree approach to estimate the price of a February 2020 European call option, where r = 0.02, Sn = \$326, K = \$320

Since European option can only be exercised at the expiration time. We do not need to compare with the case of early exercised. In addition, we will the call function instead of the put function.

$C = \max(Sn - K, 0)$.

By the binomial tree model, using 50 time intervals, we get the option price of \$24.80.

Since the grid for 50 time intervals is very big, we just show the grid for 5 time intervals here, as a comparison of the previous grid for the American put option.

		Time to Maturity (months)					
		5	4	3	2	1	0
Number of Up Movement	0	25.64	12.02	3.52	0.00	0.00	0.00
	1	0.00	40.05	21.02	7.25	0.00	0.00
	2	0.00	0.00	60.20	35.59	14.92	0.00
	3	0.00	0.00	0.00	86.28	57.48	30.70
	4	0.00	0.00	0.00	0.00	116.81	85.85
	5	0.00	0.00	0.00	0.00	0.00	149.68

h) Use B-S formula to calculate the price of a February 2020 European call option, where r = 0.02, Sn = \$326, K = \$320. Use the put-call parity relation to calculate the price of a corresponding put option.

The Black-Sholes-Merton formula for pricing a European call option is:

$$c = S_0 N(d_1) - Ke^{-rT} N(d_2)$$

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

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

Again, the function N(x) is the cumulative probability distribution function for a standardized normal distribution. c is the price of the European put option, and the remaining variables have the same definitions as before.

The result of European call option price we get from this B-S formula is \$24.71.

An important relationship between the prices of European put and call options that have the same strike price and time to maturity is called the put-call parity.

Consider two portfolios:

Portfolio A has one European call option plus a zero-coupon bond that provides a payoff of K at time T.

Portfolio C has one European put option plus one share of stock.

The value of the two portfolios at time T is shown below:

		$S_T > K$	$S_T < K$
Portfolio A	Call option	$S_T - K$	0
	Zero-coupon bond	K	K
	Total	S_T	K
Portfolio C	Put Option	0	$K - S_T$
	Share	S_T	S_T
	Total	S_T	K

If the price of stock at expiration (S_T) is larger than the strike price (K), both portfolios are worth S_T. If the price of stock at expiration is smaller than the strike price, both portfolios are worth K. As a result, the two portfolios always have the same value. In other words, both are worth

$$\max(S_T, K)$$

Since none of the options could be exercised prior to time T, and the portfolios have identical values at time T, they must also have identical values today; otherwise, there will be arbitrage opportunities.

The components of portfolio A are worth c and K * e^{-rT} today, and the components of portfolio C are worth p and S0 today. Hence,

$$c + Ke^{-rT} = p + S_0$$

Using this put-call parity, we could then evaluate the price of a corresponding European put option with the same strike price and time to maturity. The result we get is $17.64.

Appendix (Code):

```
% part a
clear

r=0.1; % risk-free interest rate
sigma=0.4; % volatility
k=50; % strike price
S0=50; % current stock price
T=5/12; % time to maturity
Smax=100;
M=20; % number of equally spaced stock prices
N=10; % number of equally spaced time intervals
ds=Smax/M;
dt=T/N;

jlist=0:ds:Smax; % list of stock prices
ilist=0:dt:T; % list of time intervals

grid=zeros(M+1, N+1); % initialize the grid
grid(1,:)=k; % boundary condition for stock price=0
grid(end,:)=0; % boundary condition for stock price=Smax
grid(:,end)=max(0,k-jlist); % boundary condition for t=T

% construct the tridiagonal matrix to solve the option price at each
iteration
j=1:M-1;
alpha=0.5*r*dt*j-0.5*sigma^2*dt*j.^2;
beta=1+sigma^2*dt*j.^2+r*dt;
gamma=-0.5*r*dt*j-0.5*sigma^2*dt*j.^2;
A=diag(beta)+diag(alpha(2:end),-1)+diag(gamma(1:end-1),1);
% iterate backwards
for int=N:-1:1
    b=grid(2:end-1,int+1); % construct the right hand side
    b(1)=b(1)-alpha(1)*k;
    grid(2:end-1,int)=A\b; % update grid by the computed option price
    for count=2:M
        if grid(count,int)<k-(count-1)*ds % determine whether early
exercise is optimal
            grid(count,int)=k-(count-1)*ds; % update grid if early
exercise is optimal
        end
    end
end

% 3D plot of option prices
mesh(ilist,jlist,grid);colorbar();title('American Put Option Price by
Implicit Finite Difference Method')
xlabel('Time (years)');ylabel('Stock Price (dollars)');zlabel('Option
Price (dollars)')
```

```
% part b
clear
% the part is the same as part a except we do not check whether early
% exercise is optimal because it is an European put option
r=0.1;
sigma=0.4;
k=50;
S0=50;
T=5/12;
Smax=100;
M=20;
N=10;
ds=Smax/M;
dt=T/N;

jlist=0:ds:Smax;
ilist=0:dt:T;

grid=zeros(M+1, N+1);
grid(1,:)=k;
grid(end,:)=0;
grid(:,end)=max(0,k-jlist);

j=1:M-1;
alpha=0.5*r*dt*j-0.5*sigma^2*dt*j.^2;
beta=1+sigma^2*dt*j.^2+r*dt;
gamma=-0.5*r*dt*j-0.5*sigma^2*dt*j.^2;
A=diag(beta)+diag(alpha(2:end),-1)+diag(gamma(1:end-1),1);

for int=N:-1:1
    b=grid(2:end-1,int+1);
    b(1)=b(1)-alpha(1)*k;
    grid(2:end-1,int)=A\b;
    % we no longer update grid by the early exercise
end

% Black-Scholes-Merton formula
d1=(log(S0/k)+(r+sigma^2/2)*T)/(sigma*sqrt(T));
d2=d1-sigma*sqrt(T);
p=k*exp(-r*T)*normcdf(-d2)-S0*normcdf(-d1); % European put option
price

% 3D plot of option prices
mesh(ilist,jlist,grid);colorbar();title('European Put Option Price by
Implicit Finite Difference Method')
xlabel('Time (years)');ylabel('Stock Price (dollars)');zlabel('Option
Price (dollars)')


% partc
```

```
clear

M=200; % 200 equally spaced position
N=200; % 200 equally spaced time period
x=linspace(0,1,M);
t=linspace(0,1,N);
dx=1/(M-1);
dt=1/(N-1);

grid=zeros(M,N); % initialize the grid
grid(:,1)=sin(pi*x); % boundary condition for t=0
grid(1,:)=0; % boundary condition for x=0
grid(end,:)=0; % boundary condition for x=1

% construct the tridiagonal matrix for solving the heat problem
r=dt/(2*dx^2);
one=ones(M-2,1);
alpha=one*2*r+1;
beta=one*-r;
A=diag(alpha)+diag(beta(2:end),-1)+diag(beta(2:end),1);
b=zeros(M-2,1);

for j=2:N
    for i=2:M-1
        b(i-1)=r*grid(i-1,j-1)+(1-2*r)*grid(i,j-1)+r*grid(i+1,j-1); %
construct the right hand side
    end
    grid(2:end-1,j)=A\b; % update grid
end

% the heat computed at each grid position through the exact function
true_grid=zeros(M,N);
for i=1:M
    for j=1:N
        true_grid(i,j)=f_heat(x(i),t(j));
    end
end

% compare the error at each grid position
error_grid=abs(grid-true_grid);

% 3D plot
mesh(t,x,grid);colorbar();title('Solving the Heat Equation Using
Crank-Nicolson Method')
xlabel('Time');ylabel('Position');zlabel('Temperature')
figure;mesh(t,x,true_grid);colorbar();title('Analytical Solution of
the Heat Equation')
xlabel('Time');ylabel('Position');zlabel('Temperature')
figure;mesh(t,x,error_grid);colorbar();title('Absolute Error Between
the Numerical Solution and Analytical Solution')
xlabel('Time');ylabel('Position');zlabel('Temperature')
```

```
% part e
clear

r=0.1; % risk-free interest rate
sigma=0.4; % volatility
k=50; % strike price
S0=50; % current stock price
T=5/12; % time to maturity
N=50; % number of partitions of time

price = binomial_american_put(r,sigma,k,S0,T,N);


function price = binomial_american_put(r,sigma,k,S0,T,N)

dt=T/N;
u=exp(sigma*sqrt(dt));
d=exp(-sigma*sqrt(dt));
a=exp(r*dt);
p=(a-d)/(u-d);
grid=zeros;

for j=0:N
    grid(j+1,N+1)=max(k-S0*u^j*d^(N-j), 0);
end

for i=N-1:-1:0
    for j=0:i
        grid(j+1,i+1)=max(k-S0*u^j*d^(i-j), exp(-
r*dt)*(p*grid(j+2,i+2)+(1-p)*grid(j+1,i+2)));
    end
end
price=grid(1,1);


% part f
clear

r=0.1; % risk-free interest rate
sigma=0.4; % volatility
k=50; % strike price
S0=50; % current stock price
T=5/12; % time to maturity

prices=zeros(1,50);
for N=1:50
    prices(N)=binomial_american_put(r,sigma,k,S0,T,N);
end
N=1:50;
```

```
plot(N,prices,'-b')
title('Convergence of the Price of American Put Option Using Binomial Tree
Approach')
xlabel('Price (dollars)');ylabel('Number of Partitions of time')




% part g
clear

r=0.02; % risk-free interest rate
S0=326; % current stock price
k=320; % strike price
T=2/12; % time to maturity
sigma=0.4; % volatility
N=50; % number of partitions of time

price = binomial_european_call(r,sigma,k,S0,T,N);


function price = binomial_european_call(r,sigma,k,S0,T,N)

dt=T/N;
u=exp(sigma*sqrt(dt));
d=exp(-sigma*sqrt(dt));
a=exp(r*dt);
p=(a-d)/(u-d);
grid=zeros;

for j=0:N
    grid(j+1,N+1)=max(S0*u^j*d^(N-j)-k, 0);
end

for i=N-1:-1:0
    for j=0:i
        grid(j+1,i+1)=exp(-r*dt)*(p*grid(j+2,i+2)+(1-
p)*grid(j+1,i+2));
    end
end
price=grid(1,1);


% part h
clear

r=0.02; % risk-free interest rate
S0=326; % current stock price
k=320; % strike price
T=2/12; % time to maturity
```

```
sigma=0.4; % volatility
% Black-Scholes-Merton formula
d1=(log(S0/k)+(r+sigma^2/2)*T)/(sigma*sqrt(T));
d2=d1-sigma*sqrt(T);
c=S0*normcdf(d1)-k*exp(-r*T)*normcdf(d2);
% Put-call parity
p=c+k*exp(-r*T)-S0;
```