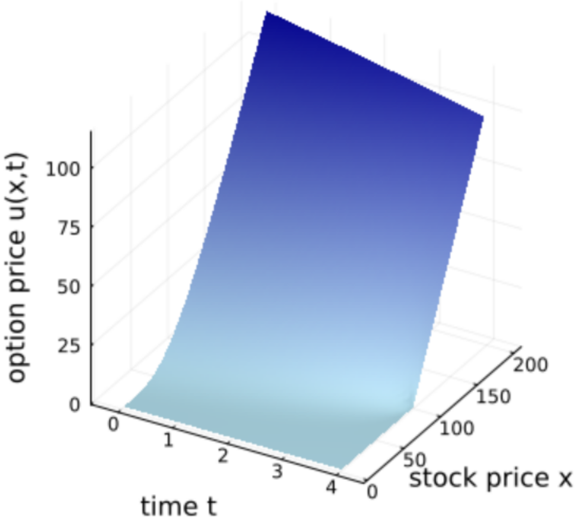
Q6 a)

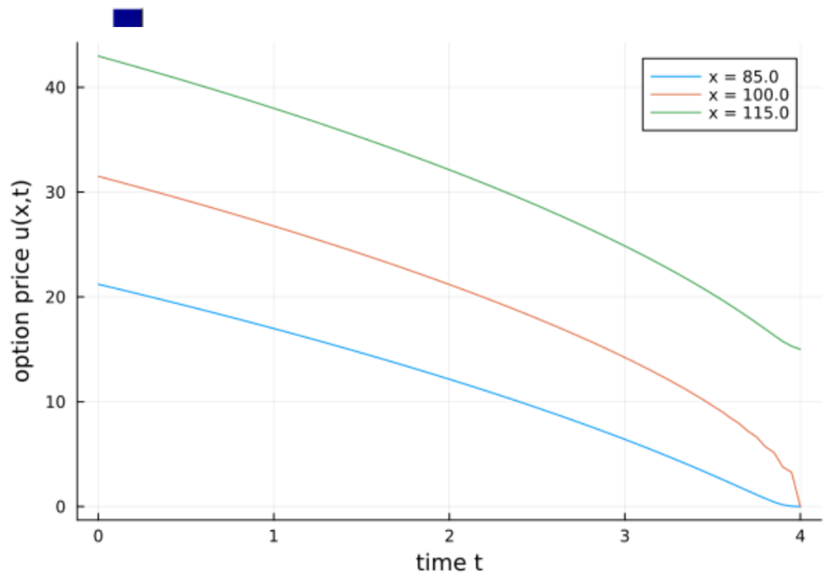
You, 2 days ago | 1 author (You)

```julia
#=================================================================================
Question 6

Sparse Arrays using Julia: https://docs.julialang.org/en/v1/stdlib/SparseArrays/
Sparse Linear Algebra using Julia: https://docs.julialang.org/en/v1/stdlib/SuiteSparse/
=================================================================================#

using SparseArrays, LinearAlgebra, Plots, Printf
# gr()
# plotlyjs()

function main(r, σ)
    # Constants
    N = 80
    x_max = 200.
    T = 4.
    Δx = x_max/N
    Δt = T/N
    K = 100.
    # r = 0.05
    # σ = 0.3
    x = range(0.0,x_max,N+1)
    times = range(0,T,N+1)

    #Initial and Boundary Conditions
    # u_x0 = spzeros(length(times))   # boundary condition
    u_xmax = x_max .- K*exp.(-r *(T .- times)) # boundary condition
    V = max.(x[2:end-1] .- K,0) # end condition

    # Crank-Nicholson Matrix scheme
    θ = 0.5
    A_multiplier = (0.5*σ^2*x[2:end-1].^2)/Δx^2
    A = spdiagm(N-1, N-1, -1 => A_multiplier[2:end].*ones(N-1 - 1), 0 => -2.0*A_multiplier.*ones(N-1), 1 => A_multiplier[1:end-1].*ones(N-1 - 1))
    B_multiplier = (0.5*r*x[2:end-1])/Δx
    B = spdiagm(N-1, N-1, -1 => -1.0*B_multiplier[2:end].*ones(N-1 - 1), 1 => B_multiplier[1:end-1].*ones(N-1 - 1))
    W = [sparsevec([N-1], [0.5*σ^2*x[N]^2 .*u_xmax[time_idx]/Δx^2 + 0.5*r*x[N]*u_xmax[time_idx]/Δx]) for time_idx in length(times)-1:-1:1]
    W1 = [sparsevec([N-1], [0.5*σ^2*x[N]^2 .*u_xmax[time_idx]/Δx^2 + 0.5*r*x[N]*u_xmax[time_idx]/Δx]) for time_idx in length(times):-1:2]
    Q = (A+B-r*I(N-1))
    lhs = (I - θ*Δt*Q)

    V_domain = V
    for iter = 1:1:length(W)
        # global V, V_domain
        rhs = (I + (1-θ)*Δt*Q)*V + (1-θ)*Δt*W1[iter] + θ*Δt*W[iter]
        V = lhs\collect(rhs)
        V_domain = hcat(V_domain, V)
    end

    # println(length(V))
    # println(length(x[2:end-1]))
    # println(length(times))
    # println(size(V_domain))

    return r, σ, times, x, V_domain
end

r, σ, times, x, V_domain = main(0.05, 0.3)

plotlyjs()
surface(show=true, reverse(times), x[2:end-1], V_domain, ylimits=[-5,210], xlimits=[-0.5,4.5], ylabel="stock price x", xlabel="time t", zlabel="option price u(x,t)", c=:blues)
gr()
surface_plot = surface(reverse(times), x[2:end-1], V_domain, ylimits=[-5,210], xlimits=[-0.5,4.5], ylabel="stock price x", xlabel="time t", zlabel="option price u(x,t)", c=:blues)
savefig(surface_plot, "surface_plot.png")
plot(reverse(times), transpose(V_domain[findall(.==(85.), x).-1, :]), label="x = 85.0")
plot!(reverse(times), transpose(V_domain[findall(.==(100.), x).-1, :]), label="x = 100.0")
plot!(reverse(times), transpose(V_domain[findall(.==(115.), x).-1, :]), label="x = 115.0")
contour_plot = plot!(xlabel="time t", ylabel="option price u(x,t)")
savefig(contour_plot, "contour_plot.png")

r_vec = [0.03; 0.05; 0.07]
σ_vec = [0.2; 0.3; 0.4]
table = zeros(3,3)
```

```julia
for r_iter = 1:3
    for σ_iter = 1:3
        global r_vec, σ_vec, table
        _, _, _, x1, V_domain1 = main(r_vec[r_iter], σ_vec[σ_iter])
        table[r_iter, σ_iter] = V_domain1[findall(.==(85.), x1).-1, end][1]
    end
end

@printf "          σ = %.2f    σ = %.2f    σ = %.2f\n" σ_vec[1] σ_vec[2] σ_vec[3]
@printf "r = %.2f    %.2f       %.2f       %.2f\n" r_vec[1] table[1,1] table[1,2] table[1,3]
@printf "r = %.2f    %.2f       %.2f       %.2f\n" r_vec[2] table[2,1] table[2,2] table[2,3]
@printf "r = %.2f    %.2f       %.2f       %.2f\n" r_vec[3] table[3,1] table[3,2] table[3,3]
```

Q6b)





Q6 c)

	σ = 0.20	σ = 0.30	σ = 0.40
r = 0.03	11.99	18.61	24.58
r = 0.05	14.84	21.21	26.98
r = 0.07	17.97	23.93	29.44

As the risk free growth rate increases so does the option value. This is because stocks with higher growth rates will likely grow in value at a much faster rate than those with lower growth rates. Thus the option to buy the stock at the strike price (which is more likely to be overtaken by the stock price at the expiry time) becomes more valuable.

As the volatility of the underlying stock increases the harder it is to predict where it will land at the expiry time. Thus the option to by it becomes more valuable since the holder of the option can choose to buy the stock at the expiry time only if it is higher than the strike price. In the case that the stock price is lower than the strike price, then the holder of the option can simply choose not to excersize it. This protects from the risk associated with the underlying stock.