Stockholm School of Economics
Adrien d'Avernas and Pablo Balsinde
Spring 2025

ASSET PRICING THEORY

Problem Set 4

The goal of this problem set is to develop a foundation in numerical techniques, including optimization, numerical integration, and simulation. While built-in functions exist for these methods, you are required to implement the algorithms yourself. You may use any standard programming language. Be sure to include a random seed in your code. Attach your codes in your submission. If you receive the help of an LLM, please submit your conversations with it.

1 Optimization

The first-order condition that characterizes the solution to an optimization problem states that the derivative of the objective function must be equal to zero. Because of this, we often have to find the points where equations equal zero. Lets denote the derivative of the objective function as $f(x)$, where the solution to the optimization problem satisfies $f(x^*) = 0$.

The Newton-Raphson method is one of the main methods to find the roots of a function. It is iterative and requires an initial guess x_0. It also requires knowing the derivative of the function to be solved.[1] Detailed notes on the method can be easily found online; the intuition for it comes from the first-order Taylor expansion of the function $f(x)$ around the initial guess x_0:

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0).$$

To find the root x^* where $f(x^*) = 0$, we set the approximation to zero:

$$0 \approx f(x_0) + f'(x_0)(x - x_0).$$

Solving for x:

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

[1] The method can still work if this derivative is approximated; this modified version of the method is called the Secant method.

The resulting x will not be the x^* we are after because we used a first-order expansion. However it gives us an iterative updating rule:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Starting from our initial guess. We will compute the above expression until it has converged (when $|x_{n+1} - x_n|$ is below a given tolerance level). Intuitively, each step approximates $f(x)$ by its tangent line at x_n. The next x_{n+1} is where this tangent crosses the x-axis. The method converges quadratically if x_n is close to the actual root x^*. For a visual interpretation of the method, see the first 10 minutes of 3Blue1Brown's video on a related topic.

1.1 A polynomial:

Consider the function $f(x) = -x^6 + 2x^3 + 3x^2$.

- Can you find its global maximum with our method?
- Define a fine grid of initial guesses in the interval $[-3, 3]$. Run the algorithm for each initial guess, and for each initial guess store the solution obtained through the algorithm as well as the number of iterations.
- Plot the obtained solution and the number of iterations against the initial guess. Is the algorithm robust? What share of initial guesses lead you to the global maximum? What happens when we make a guess close to a local minimum or maximum?

1.2 The sigmoid function:

The sigmoid function is special in that it is bounded between 0 and 1. It is central in logistic regressions and choice models, where it allows one to model probabilities of events occurring (e.g., the probability that a consumer will choose a product or that a borrower will default). It takes the form

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$

Suppose we want to solve the following modified version of it:

$$\frac{1}{1 + e^{-x}} - 0.5 = 0$$

Implement the same steps as in the prior section. Why does the procedure fail for very inaccurate guesses? Analyze what happens to the update steps, it might be helpful to record the average update step for different initial guesses. Discuss the strengths and weaknesses of the Newton-Raphson method.

1.3 The bisection algorithm (optional)

Lets consider now the bisection method. It is more robust, does not require derivatives, and guarantees convergence under mild conditions. However it is typically slower. The method is based on the Intermediate Value Theorem: if a function $f(x)$ is continuous and $f(a)f(b) < 0$, then there exists a root in (a, b). It works as follows:

1. Start with an interval (a, b) such that $f(a)f(b) < 0$.
2. Compute the midpoint:
$$c = \frac{a+b}{2}$$
3. Evaluate $f(c)$. If $f(c) = 0$, stop (root found).
4. If $f(a)f(c) < 0$, update $b = c$; otherwise, update $a = c$.
5. Repeat until $|b - a|$ is sufficiently small.

Implement the bisection algorithm for the polynomial in part 1.1. You will have to define different initial intervals to be able to find all of the roots. How many iterations does it take? How does it compare to Newton-Raphson?

Numerical integrals

We often have to evaluate integrals that do not have simple anti-derivatives. We will explore three numerical integration methods to do so. The first two are the simplest of the many Closed Newton-Coates Quadratures: the idea behind them is to fit a simple function to the target function and to integrate over that simple function. The last is a stochastic method, which can be useful to solve deterministic integrals if these are high-dimensional, as quadrature methods become intractable.

1. Trapezoidal Rule: involves dividing the interval of the function to be integrated into smaller subintervals and approximating the function within each subinterval with a linear (first-order) function. The integral is then approximated by summing up the areas of the resulting trapezoids:

$$I_{Trap} = h\left[\frac{f(a)+f(b)}{2} + \sum_{i=1}^{N-1} f\left(x_i\right)\right]$$

N, h, x_i are left undefined. Any standard textbook will allow you to figure them out. The error bound for the Trapezoidal rule follows:[2]

$$|E_{Trap}| \leq \frac{(b-a)h^2}{12} \max_{x\in[a,b]} |f''(x)|$$

2. Simpson's 1/3 Rule: involves dividing the interval of the function to be integrated into an even number of subintervals and approximating the function over pairs of subintervals using a quadratic (second-order) polynomial. The integral is then approximated by summing up the areas under the resulting parabolic segments.

$$I_{S1/3} = \frac{h}{3}\left[f(a) + 4\sum_{\substack{i=1\\ \text{odd}}}^{N-1} f(x_i) + 2\sum_{\substack{i=2\\ \text{even}}}^{N-2} f(x_i) + f(b)\right]$$

Its error bound is:

$$|E_{S1/3}| \leq \frac{(b-a)h^4}{180} \max_{x\in[a,b]} |f^{(4)}(x)|.$$

[2]You are encouraged to review a proof of this error bound. For example see Bender (2008).

3. Monte Carlo Integration: involves generating N independent random samples $U_1, U_2, \ldots, U_N$ from the uniform distribution over the interval $[a, b]$. The Monte Carlo estimator for the integral is:

$$\hat{I}_{MC} = \frac{(b-a)}{N} \sum_{i=1}^{N} f(U_i).$$

$\hat{I}_{MC}$ is an unbiased estimator of I. Its standard error follows:

$$\text{SE} = (b-a)\sqrt{\frac{\mathbb{V}[f(U)]}{N}},$$

Exercises

For this exercise we will consider a one-dimensional integral that can actually be solved analytically:

$$I = \int_{-1}^{1.57474} f(x)dx = -x^6 + 2x^3 + 3x^2$$

1. Algebraically, how many intervals will you need under the first two methods to get the error bounds below 10^{-4}?

2. Write a program to implement the three integration methods. Obtain estimates for the integral for different values of N. Here is a suggestion: $[6, 10, 100, 500, 1000, 5000, 10000]$

3. Make a plot of the evolution of the absolute errors, the error bounds, and the standard errors against N (log scales may help). Remember we can obtain the actual errors because we know the value of the integral.

4. Which orders for the decline rate of the errors do you find? Could you have inferred the orders of convergence based on the expressions for the error bounds and standard errors?[3]

5. Discuss the tradeoffs and limitations of the different methods. It may also be helpful to keep track of the number of function evaluations required and the run time.

[3] Hint: Take the log of the errors and fit a straight line over the series you produced, the convergence rate will be the slope.

Portfolio selection and Value at Risk (optional)

Consider an economy with three risky assets. Their returns follow:

$$r = \begin{bmatrix} 0.12 \\ 0.14 \\ 0.16 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 0.04 & 0.01 & 0.02 \\ 0.01 & 0.09 & 0.03 \\ 0.02 & 0.03 & 0.16 \end{bmatrix}$$

Suppose you are Alice, a portfolio manager. Construct a grid of target returns in the interval $\mu_{\text{target}} \in [0.10, 0.20]$. As in Chapter 2, find the portfolio weights that will deliver the target returns and each portfolio's volatility.

Now suppose you are Bob, another portfolio manager. Alice proposes to Bob that he invest following the portfolios she has obtained. Bob knows the distribution of the three asset returns, and he knows the shape (the picture) of the efficient frontier. However he does not have the asset pricing knowledge to confirm whether Alice's portfolios are on it.

He decides to take a Monte Carlo approach to test Alice's portfolios. He decides to simulate $M = 10,000$ draws of returns following the known multivariate distribution and apply them to the different portfolios. His plan is to build a confidence interval of the expected returns of the portfolios, and check whether he can reject that Alice's portfolios lie on the frontier. Implement Bob's plan; you can consider the following expression for the confidence intervals:

$$\text{CI}_\mu = \mu_{\text{realized}} \pm 1.96 \frac{\sigma_{\text{realized}}}{\sqrt{M}}$$

What are you assuming when building the CIs this way? If you see fit, consider percentile based CIs or bootstrap methods to build them. Can you (Bob) reject that Alice's portfolios were optimally chosen? Plot the efficient frontier (with the true standard deviations and expected returns), along with the confidence intervals you obtained.

The firm now imposes a VaR constraint on its portfolio managers with a confidence level $\lambda = 0.70$ and a maximum loss $\bar{V} = 0.075$. Alice applies her theoretical knowledge of Chapter 2.7 to classify the portfolios, what is the upper bound on the volatility she finds? [4]

She informs Bob of which portfolios are VaR-compliant, but Bob again turns to simulations. Implement Bob's plan: is it true that the theoretically VaR-compliant portfolios also meet the constraint in the simulations?[5] If you would like, you can display a single plot of the frontier with the CIs, marking the portfolios that do not satisfy the constraint.

[4] You can check that the derivation of the constraint is similar without a risk-free asset: $\mu_{\text{target}} \geq \Phi^{-1}(\lambda)\sigma_{\text{theoretical}} - \bar{V}$

[5] It is possible that you not be able to verify this for the portfolios around the threshold due to sampling error.