Problem Set 4 Answers: Univariate Optimization and Solving for a Root
P. Dybvig

1. Convex ond nonconvex optimization: grid search versus binary search. Recall that an optimization problem is called convex if

- The feasible set is convex, and
- The objective function is concave if maximizing or convex if minimizing.

For more information about convex sets and functions, see Appendix A in the book.

A. Which of the following univariate optimization problems are convex optimizations and which are not? Justify your answers.

(i)
Choose $x \in \Re$ to
maximize $x - x^2$, subject to
$x \geq 1$ and
$x \leq 6$.

A convex problem: this is an optimization and the objective function is concave because the second derivative, -2, is negative and the feasible set is the interval $[1, 6]$, a convex set.

(ii)
Choose $y \in \Re$ to
maximize $y - \exp(y)$, subject to
$y^2 \geq 1/2$ and
$y^2 \leq 25$.

Not a convex problem: the objective function is okay because this is an optimization and the objective function is concave because the second derivative, $-\exp(y)$, is negative. However, the feasible set is $[\sqrt{1/2}, 5] \bigcup [-5, -\sqrt{1/2}]$, which is not convex. However, this is a convex problem on both intervals $[\sqrt{1/2}, 5]$ and $[-5, -\sqrt{1/2}]$ individually so we could solve the convex problem on each interval and the optimum for the nonconvex problem would be the optimum for the interval with the largest value.

(iii)
Choose $z \in \Re$ to
minimize $z \exp^{-z}$, subject to
$z \geq 1/2$ and
$z \leq 10$.

Not a convex problem: the feasible set is convex because it is an interval $[1/2, 10]$. However, this is a minimization and the objective function is not convex (or concave, for that matter) on the whole feasible set. The second derivative is $(x-2)exp(-x)$, which is negative for $x < 2$ and positive for $x > 2$.

B. Suppose it is important to find a global optimum. For which problems would you recommend using a grid search and for which would you recommend a binary search? (Assume you want to use the faster procedure if both are reliable.)

(i) is a convex problem, so I would definitely use a binary search, which should be much faster than a grid search and is guaranteed to find an optimum.

(ii) is not a convex problem, so a grid search would seem to be indicated to find the global optimum. However, I would do a binary search on each interval, guaranteed to find the optimum in each interval, and then pick the one that does better.

(iii) is not a convex problem, so a grid search would seem to be indicated to find the global optimum. However, it is possible to use binary search here as well (although it is more subtle than in case (ii)). On the inteval $[2, 10]$, the objective function is convex so binary search will find a global minimum there. On the inteval $(1/2, 2)$, the objective function is stricly concave so there cannot be either a local minimum or a global minimum there. Therefore, we can use binary search to find the minimum on $[2, 10]$ and compare the objective function at that local minimum with the value at $z = 1/2$. Whichever has smaller value is the global minimum.

Note that clever use of special information about the objective function allowed us to use binary search to find a global optimum in (ii) and (iii). In many cases in practice, we do not know enough about the objective function to do this kind of analysis, and even if we could it would not be worth the effort.

It is worth noting that all of these problems can be solved directly without any numerical optimization. (i) is a quadratic optimization whose objective has derivative $1-2x$ which is negative on the interval $[1,6]$, so the function is decreasing there and the optimum must be at $x=1$. (ii) has an objective function whose derivative $1-exp(y)$ is positive for $y \in [-5,-\sqrt{1/2}]$, so the objective function is maximized there at $y=-\sqrt{1/2}$, and is negative for $y \in [\sqrt{1/2},5]$, so the objective function is maximized in the interval at $y=\sqrt{1/2}$. Comparing the two values, we find the global maximum is at $y=-\sqrt{1/2}$. (iii) the objective of (iii) is decreasing on the interval $[2,10]$ (since its derivative $(1-z)exp(-z)$ is negative there), so the minimum on that interval is achieved at $z=10$. Therefore, the global minimum is at $z=10$, which has smaller value than $z=1/2$, which is the only other candidate as pointed out above. So, some applications do not require numerical optimization depending on how hard we want to work.

C. Follow the advice in B and solve each of the optimization problems.

The solutions are: (i) $x=1$
(ii) $y=-\sqrt{1/2}$
(iii) $z=10$.

2. Newton's method.

Consider finding a zero of the function

$$f(x)=\frac{x}{1+x^2}.$$

We know by inspection that $x=0$ is the only zero of the function; let us see how Newton's method performs.

A. Compute $f'(x)$. Compute the rule for a Newton step: $\mathcal{N}(x) = x - f(x)/f'(x)$.

$$\begin{aligned}
f'(x) &= \frac{1}{1+x^2} - \frac{2x \cdot x}{(1+x^2)^2} \\
&= \frac{1-x^2}{(1-x^2)^2} \\
\mathcal{N}(x) &= x - f(x)/f'(x) \\
&= x - \frac{x/(1+x^2)}{(1-x^2)/(1+x^2)^2} \\
&= x\left(1 - \frac{1+x^2}{1-x^2}\right) \\
&= x\left(\frac{1-x^2}{1-x^2} - \frac{1+x^2}{1-x^2}\right) \\
&= -\frac{2x^3}{1-x^2}
\end{aligned}$$

B. For what range of x does a sequence of normal Newton steps converge to the zero of f? (You can do this numerically or using algebra.)

Near $x = 0$, $\mathcal{N}(x) \approx -2x^3$ and the sequence x_1, x_2, ... where $x_{n+1} = \mathcal{N}(x_n)$ converges very rapidly to 0 (as guaranteed by $f'(0) \neq 0$). For $|x|$ large, on the other hand, $|\mathcal{N}(x)| > 2|x|$, and the sequence diverges rapidly. It can be proven that $|\mathcal{N}(x)| < |x|$ for $|x| < \sqrt{1/3} \approx 0.5773503$ and $|\mathcal{N}(x)| > |x|$ for $|x| > \sqrt{1/3}$ (except when $|x| = 1$ when $\mathcal{N}(x)$ is undefined due to division by 0). Therefore, the sequence explodes for $|x_1| > \sqrt{1/3}$ and converges for $|x_1| < \sqrt{1/3}$. Convergence or explosion becomes slower and slower as x_1 approaches $\pm\sqrt{1/3}$. When $|x_1| = \sqrt{1/3}$, the sequence alternates between $\sqrt{1/3}$ and $-\sqrt{1/3}$ and the sequence neither converges nor explodes.

3. Portfolio Insurance. You are working for a big international investment bank, and you have a client that is interested in using portfolio insurance or some variant for a pension fund with \$1 billion in assets. (This is large

compared with the entire size of some smaller firms, but only a small piece of the pension money at some of the largest firms.) Traditional portfolio insurance follows a dynamic trading strategy (computed using option pricing techniques) to generate a payoff at the end that is the larger of a guarantee G (often chosen as $G = W_0$) and some constant k times the value M at the end of investing \$1 in a stock index:

$$\text{payoff} = \begin{cases} G & kM \leq G \\ kM & G < kM \end{cases}$$

Sometimes a program is given a pool of money to invest in the stock (so $k = W_0$) and the put represented by the insurance (the option to forfeit the stock portfolio at the end in exchange for W_0) is paid for separately. More often, the price of the insurance is built into the cost of the program and k is chosen less than W_0 so the portfolio insurance payoff is worth W_0. That is, k solves the nonlinear equation that says k plus the value of the put equals W_0.

Recall that the the Black-Scholes call option pricing formula says the call price is

$$C(S, B, v) \quad = \quad SN(\frac{\log(S/B)}{v} + \frac{v}{2}) - BN(\frac{\log(S/B)}{v} - \frac{v}{2}),$$

where S is the stock price, B is the price of the discount bond paying X at the maturity of the option, and v is the square root of the remaining variance in $\log(S/B)$ between now and the maturity of the option. In the original Black-Scholes model, the relative stock price variance was a constant σ^2 per unit time, the interest rate was a constant r, and the strike price was X. Therefore, if the time-to-maturity is T, we can write $B = Xe^{-rT}$ and $v = \sqrt{\sigma^2 T}$. Recall that $\partial C(S, B, v)/\partial S = N(\frac{\log(S/B)}{v} + \frac{v}{2})$: all the terms from the chain rule having $\partial \log(S/B)/\partial S$ inside $N(\cdot)$ cancel (although this is a challenge to prove).

You should use the Black-Scholes model to solve for k in the portfolio insurance contract, assuming that the price of insurance is built-in.

A. Use put-call parity to write down the Black-Scholes put price + stock price in terms of the stock and the call.

$$P + S = B + C$$

B. Write down the nonlinear equation to be solved by k. Note that the stock price to be used in the option formula will depend on k. The strike price of the option is G, so $B = G\exp(-rT)$, so we have

$$W_0 = G\exp(-rT) + C(k, G\exp(-rT), v)$$

C. Solve for the appropriate value for k. Assume $W_0 = \$1,000,000,000$, $\sigma = 0.2$, $T = 5$ years, and $r = 5\%/\text{year}$. Solve for k five times, for G taking on five different values ranging from $0.5W_0$ to $1.5W_0$. Diagnose and to the extent possible correct for any failures in the numerical solution. Use Solver if you like or use Newton's method and do it yourself.

Here are the values I get:

G ($ billions)	k ($ billions)
0.500	0.998
0.750	0.980
1.000	0.906
1.250	0.613
1.500	failed

In the last case, there is no solution because even when $k = 0$ the present value of the guarantee ($1.168 billion) is already bigger than initial wealth. In this case, solver reduces k until $k < 0$, and at this point $\log(k/B)$ is undefined.

There are a lot of potential pitfalls in this example. This is similar to optimization in practice, which usually requires some hand-holding. Here are some things I noticed:

- If we start at $k = 0$, we have a problem because $\log(k/B)$ is undefined and therefore the call price and the test are undefined. In this case, Solver fails from the outset.

- If we start with a very small value of k to avoid $\log(0)$, then the call is way out of the money and the Solver computes $f'(k) = 0$ and concludes there is no solution (because k does not seem to affect the value of the formula it is trying to move to 0). Starting with $k = W_0$ always seems to work.

- I thought scaling might be a problem, but Solver seemed to do okay with this (I suspect it uses double precision arithmetic). In general, if you have a number like \$1 billion = \$1,000,000,000, I suggest entering it as a number like 1 (units=billions of dollars) or 1,000 (units=millions of dollars). Leaving it in billions is asking for trouble.

- As noted above, if G is large enough, there is no solution. Specifically, the payoff is always at least G, a claim that is worth $G\exp(-rT)$ today. Therefore, if $G\exp(-rT) > W_0$, the claim is too expensive whatever the value of k, and there is no solution.