BEPP 931 - Solution to Problem Set 6

Shasha Wang, Cung Truong Hoang, Jose Sidaoui

March 13, 2020

Question 1-4

Question 1 through question 4 are to solve a portfolio choice problem using different methods - Gauss-Hermite, QNWNORM, Monte Carlo, Quasi-Monte Carlo, and stochastic approximation. In short, "simulation" is not the only way to evaluate an integral. **Gauss-Hermite Quadrature and QNWNORM are the most stable, and QNWNORM is the easiest to code (given the distribution is normal). If distributions are other than normal, or of high dimensionality, it's easier to use the two Monte Carlo method, especially Quasi-Monte Carlo method, since equidistributed sequences are better for integration**.

The problem is that we have two assets, one save asset with return $R = 1.01$ and one risky asset with return $Z \sim N(1.06, 0.04)$. Same price for both assets.

We have one investor with utility function $u(c) = -e^{-c}$.

Hence the expected utility next period is

$$-E\left\{e^{-((1-\omega^*)R+\omega^* Z)}\right\} = -\int e^{-((1-\omega^*)R+\omega^* Z)} \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(\frac{z-\mu}{\sqrt{2}\sigma}\right)^2} dz$$

,

where $\mu = 1.06, \sigma = \sqrt{0.04},$ and $\omega^* = 1.25$.

Our objective is to maximize the next period expected utility

$$\max_{\omega} -E\left\{e^{-((1-\omega)R+\omega Z)}\right\}$$

.

Below are the five[1] methods to solve the integration optimization problem.

I change the order of the questions to follow Monte Carlo integration by Quasi-Monte Carlo integration.

Exercise 6-1 Gauss-Hermite Quadrature and QNWNORM

In this section we use Gauss-Hermite quadrature to solve for the integration. I also use qnwnorm in the compecon toolbox to compute nodes and weights for normal distribution and then solve for the integration.

[1] There are four sections and five methods, because I also use qnwnorm in addition to Gauss-Hermite in the first exercise.

Both methods yields the same result as shown in the tables below, but I prefer QNWNORM because it needs NOT change of variables and to multiply the utility function by the probability distribution function. Given the same level of accuracy, QNWNORM is more efficient at least in terms of coding time.

Table 1 tabulates the expected utility for $\omega = 1.25$.

# nodes	utility
3	-0.35301
5	-0.35301
10	-0.35301

Table 1. Integration by Gauss-Hermite Quadrature & QNWNORM given $\omega = 1.25$

Table 2 shows the optimization result.

# nodes	utility	portfolio
3	-0.35301	1.2502
5	-0.35301	1.25
10	-0.35301	1.25

Table 2. Gauss-Hermite Quadrature & QNWNORM Optimization

Exercise 6-2 Monte Carlo Integration

Monte Carlo integration takes on Law of Large Numbers, and is in my opinion as easy to code as QNWNORM. But since Monte Carlo suffers from problems of variance (Table 3), I prefer QNWNORM as long as the random variable follows normal distribution. If not, Monte Carlo should be the choice, and we could employ some variance reduction techniques as shown in the slide 7a.

	expected utility		optimal portfolio	
# draws	mean	std. dev.	mean	std. dev.
100	-0.35058	0.0098733	1.337	0.56359
1000	-0.35275	0.002848	1.2589	0.16533
10000	-0.3529	0.00082904	1.256	0.049235

Table 3. Monte Carlo Integration

Exercise 6-3 Stochastic Approximation

This question takes on stochastic gradient method: Iterate

$$x_{k+1} = x_k - \lambda_k g_x\left(x_k, z_k\right), \quad k = 1, 2, \dots$$

where $\{z_k\}_{k=1}^{\infty}$ is a sequence of draws (run data more than once by setting $i = (k - 1 \bmod N) + 1$) and $\{\lambda_k\}_{k=1}^{\infty}$ is a sequence of step lengths such that $\lambda_k \to 0, \sum_{k=1}^{\infty} \lambda_k = \infty$, and $\sum_{k=1}^{\infty} \bar{\lambda}_k^2 < \infty$ (Robbins-Monro conditions).

In this case, the iterations are

$$\omega_{k+1} = \omega_k + \lambda_k \left(-e^{-((1-\omega_k)R+\omega_k Z_k)}\left(R - Z_k\right)\right), \quad k = 1, 2, \dots$$

where $\omega_1 \sim U[0, 3]$. Note that the steepest descent direction is replaced by the steepest ascent direction. The left (right) panel sets $\lambda_k = \frac{1}{k}$ $\left(\lambda_k = \frac{10}{k}\right)$.

Table 4 shows the results of stochastic approximation. Note that even when the number of draws is 10000, the variance is still not non-trivial.

	$\lambda_k = \frac{1}{k}$ optimal portfolio		$\lambda_k = \frac{10}{k}$ optimal portfolio	
# draws	mean	std. dev.	mean	std. dev.
100	1.4784	0.74873	1.4461	0.60564
1000	1.4135	0.81175	1.3087	0.49708
10000	1.4833	0.81663	1.36	0.55157

Table 4. Stochastic Approximation - Computed using 100 Replications

Exercise 6-4 Quasi-Monte Carlo Integration

For this question, I build on QNWEQUI in compecon toolbox to write a qnwequi_WHNBR function to include the Baker method.

Quasi-Monte Carlo integration takes on pseudorandom number generation, whose goal is to construct sequences that are well-suited for integration, not statistics.

In this section we compare between the performances of random number generator and pseudo random number generator, the latter of which takes on four equidistributed sequences - Weyl, Haber, Niederreiter, and Baker, as shown below,

sequence	definition
Weyl	$\left(\left\{jp_1^{\frac{1}{2}}\right\},\ldots,\left\{j_d^{\frac{1}{2}}\right\}\right)$
Haber	$\left(\left\{\frac{j(j+1)}{2}p_1^{\frac{1}{2}}\right\},\ldots,\left\{\frac{j(j+1)}{2}p_d^{\frac{1}{2}}\right\}\right)$
Niederreiter	$\left(\left\{j2^{\frac{1}{d+1}}\right\},\ldots,\left\{j2^{\frac{-1}{d+1}}\right\}\right)$
Baker	$\left(\{je^{r_1}\},\ldots,\{je^{r_{id}}\}\right)$

where $\{x\}$ is the fractional part of x and $p_1,\ldots,p_d\,(r_1,\ldots,r_d)$ are d distinct prime (rational) numbers.

Table 5 and 6 show the results of expected utility and optimal portfolio for each of the five methods.

By comparison, Haber and Uniform do not as better as the other methods which is no surprise given Figure

	expected utility				
#draws	Weyl	Haber	Niederreiter	Baker	Uniform
100	-0.34814	-0.51355	-0.34814	-0.36809	-0.39311
1000	-0.35274	-0.34488	-0.35274	-0.3547	-0.39066
10000	-0.35307	-0.35105	-0.35307	-0.35305	-0.368

Table 5. Expected Utility - Quasi-Monte Carlo Integration

	optimal portfolio				
#draws	Weyl	Haber	Niederreiter	Baker	Uniform
100	1.2654	1.1039	1.2654	1.5575	1.478
1000	1.2522	1.0975	1.2522	1.2501	1.2042
10000	1.2477	1.1953	1.2477	1.2531	1.2249

Table 6. Optimal Portfolio - Quasi-Monte Carlo Integration

5.1 in Miranda, M. Fackler, P. (2002), in which it shows Weyl and Niederreiter perform the best in terms of equidistributedness.

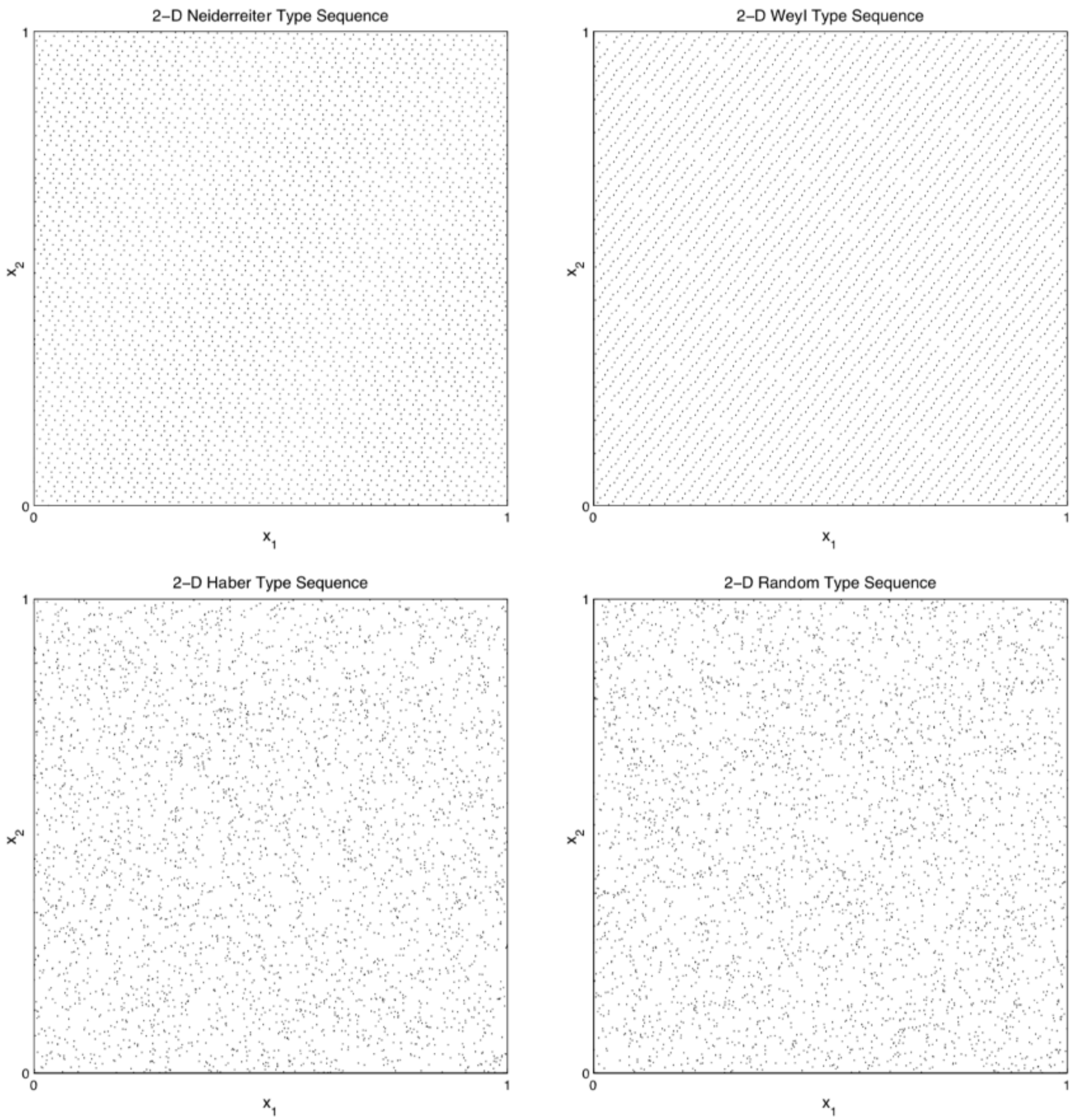


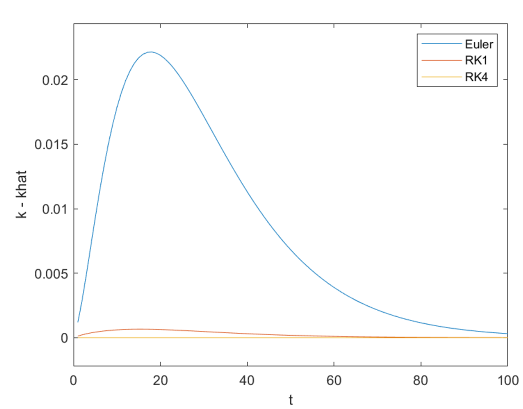
Figure 1. Alternative Equidistributed Sequences - (a) Niederreiter, (b) Weyl, (c) Haber, and (d) uniform. Source: Miranda, M. Fackler, P. (2002), Figure 5.1.

Exercise 6-5 Euler and Runge-Kutta

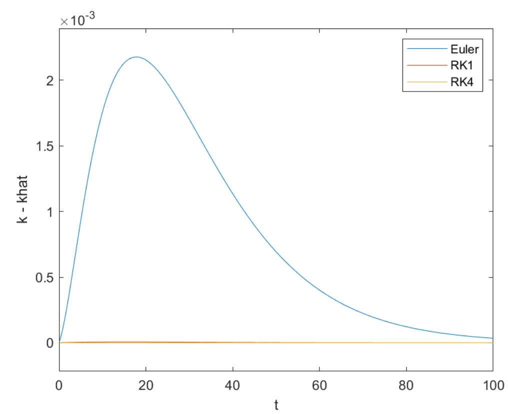
For the Solow Model, we compare the numerical solution of the Euler, Runge-Kutta order 1 and Runge-Kutta oder 4 method with the true solution for the evolution of the capital stock k. For every step size h, we create a grid t ranging from 0 to 100 with equal spacing of size h. The table below compares the maximum approximation error across the three methods for different values of h. We see that Euler converges linearly at rate h, RK1 at rate h^2 and RK4 at rate h^4. In the figures below we plot the approximation error $k - \hat{k}$ against t. We notice that for smaller values of h the approximation improves for every method. Secondly, the approximation errors converge to zero as t increases and as we reach the steady state value k^* for the capital stock.

Table 7. Maximum Approximation Error for t $\in [0, 100]$

h	Euler	RK1	RK4
1	2.21(-2)	6.63(-4)	3.16(-7)
0.1	2.18(-3)	6.34(-6)	3.27(-11)
0.01	2.17(-4)	6.31(-8)	6.66(-15)



(a) Approximation Errors for h = 1



(b) Approximation Errors for h = 0.1

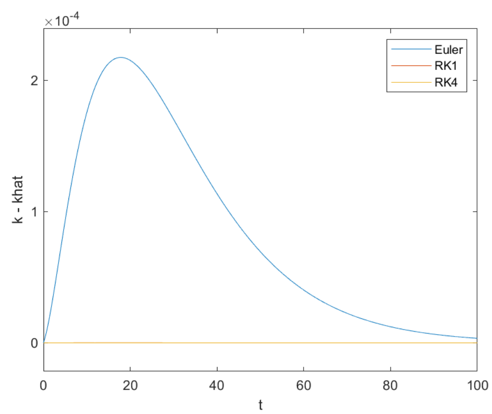


Figure 3. Approximation Errors for h = 0.01