Computational Methods in Economics and Finance
Exam 2

This test is due Sunday, April 18 at 12:00 noon. You should follow the guidelines discussed in the syllabus carefully when submitting your answer. All work should be your own and you should not discuss the exam with anyone but me. Collaborating and/or giving or receiving help on this exam is a violation of the university's honor code and will result in sanctions (see syllabus for more information). If you have questions about what is being asked or how to solve the problems, you should contact me (preferably by email).

2.1. Suppose that you want to integrate functions defined on the triangle with vertices $(x, y) = (0, 0)$, $(1, 0)$ and $(0, 1)$, i.e., you want to compute integrals of the form

$$\int_0^1 \int_0^{1-x} f(x, y) dy dx$$

An obvious set of nodes is to use the evenly spaced grid points on the unit square and throw away any for which $x + y > 1$.

These gridpoints are defined by $(i/m, j/m)$, for $i, j \in \{0, 1, \ldots m\}$, with $i + j \leq m$. Weights could be defined by that would exactly integrate

$$f(x, y) = x^i y^j$$

for all the above values of i and j. A bit of calculus shows that

$$\int_0^1 \int_0^{1-x} x^i y^j dy dx = \frac{i!j!}{(i + j + 2)!} = \frac{\Gamma(i + 1)\Gamma(j + 1)}{\Gamma(i + j + 3)}$$

Write a Matlab function that takes as input the value m and returns the nodal pairs (x, y) and the weights w. It should have the syntax

```
[x,w]=qnwtest(m)
```

where x is $(m + 1)(m + 2)/2 \times 2$ and w is $(m + 1)(m + 2)/2 \times 1$. Write a script that displays tables of nodes and weights for $m = 1$ to 4.

As an example, with $m = 1$ the three nodes are the corners of the triangle and each gets a weight of 1/6.

2.2. Consider the d-dimensional process described by

$$dx = \mu(x)dt + \omega(x)dW$$

Here $\mu : R^d \to R^d$ and $\omega : R^d \to R^{d\times d}$. The infinitesimal generator for this process is

$$\mathcal{L} = \sum_{k=1}^{d} \mu_k(x)\frac{\partial}{\partial x_k} + \frac{1}{2}\sum_{k=1}^{d}\sum_{l=1}^{d}\Omega_{kl}(x)\frac{\partial^2}{\partial x_k \partial x_l}$$

where $\Omega_{kl}(x) = \sum_{r=1}^{d} \omega_{kr}(x)\omega_{lr}(x)$. With this operator, the basic arbitrage equation in finance can be written

$$rV(x,t) = \mathcal{L}V(x,t) + V_t(x,t)$$

When we use the approximation $V(x,t) \approx \phi(x)c(t)$, the infinitesimal generator can be applied to $\phi(x)$:

$$\mathcal{L}\phi(x) = \sum_{k=1}^{d} \mu_k(x)\frac{\partial \phi(x)}{\partial x_k} + \frac{1}{2}\sum_{k=1}^{d}\sum_{l=1}^{d}\Omega_{kl}(x)\frac{\partial^2 \phi(x)}{\partial x_k \partial x_l}$$

If ϕ has n basis functions, this is a $1 \times n$ vector when evaluated at any point $x \in R^d$. If we evaluate it at N values of x it is an $N \times n$ matrix L.

Write a Matlab function that computes L. The function should have the syntax

```
L=infgen(fspace,mu,omega,x)
```

where `fspace` defines ϕ, `mu` is a function handle for a function that takes $N \times d$ matrices of values of x and returns an $N \times d$ matrix of values of $\mu(x)$, `omega` is a function handle for a function that takes $N \times d$ matrices of values of x and returns an $N \times d \times d$ array of values of $\omega(x)$ and `x` is an $N \times d$ matrix (if `x` is not passed to the function you should use the standard nodes for `fspace`).

Be sure to document the code.

Programming tip: use the next problem to test your code.

2.3. Consider an option that pays, at time T

$$V(S,T) = \max(0, K - \min(S_1, S_2))$$

where

$$dS_i = rS_i dt + \sigma_i S_i dW_i$$

with the correlation between the Browian motions equal to ρ (this is sometimes expressed as $dW_1 dW_2 = \rho dt$). In the notation of the previous problem

$$\mu(S) = [rS_1 \quad rS_2]$$

and

$$\omega(S) = \begin{bmatrix} \sigma_1 S_1 & 0 \\ \rho\sigma_2 S_2 & \sqrt{1-\rho^2}\sigma_2 S_2 \end{bmatrix}$$

Price this option in two ways: use FINSOLVE and do it using the form

$$[rI - L]c(t) = \Phi c'(t)$$

where you obtain L from the function you wrote in the previous problem (if you were unable to complete the previous problem, form L "by hand"). Solve this differential equation using the ODE solver of your choice.

The two answers should be essentially the same. Are they? Plot the contours (or make a `surface` plot) of the option with T periods until expiration.

Use the following parameter values: $r = 0.05$, $\sigma_1 = 0.5$, $\sigma_2 = 0.2$, $\rho = 0.5$, $T = 1$ and $K = 1$. Use the piecewise linear family of approximating functions.

Be sure your solution is intuitively reasonable. As a check, price the option that pays

$$V(S,T) = \max(0, K - \max(S_1, S_2))$$

(you should only have to change 1 line of code to accomplish this). Compare the values of the two options and discuss the intuition behind the results you get.

Programming tip: set $\rho = 0$ at first. When your code works well for this case, let ρ non-zero.

2.4. Using the four points (x_1, x_2), $(x_1+\delta, x_2)$, $(x_1, x_2+\delta)$ and $(x_1+\delta, x_2+\delta)$, can you define a set of weights to get an $O(\delta)$ approximation of the cross derivative at (x_1, x_2):

$$\frac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_2}$$

If this is possible, explain how you found the weights and demonstrate the use of the approach on an example function. If it cannot be done, explain why.