Econ 561a
Spring 2010
Yale University
Prof. Tony Smith

# HOMEWORK #3

**1.** Write a program (in a language of your choosing) that uses bicubic spline interpolation to approximate the function $f(x,y) = \log(x + y^3)$ for $(x,y) \in [0.1,1] \times [0.1,1]$. How does the approximation error compare to bilinear interpolation?

**2.** Write a program in that uses Gauss-Hermite quadrature to compute $E[e^X]$, where $X \sim N(\mu, \sigma^2)$. Set $\mu = 1$ and $\sigma = 1,2,3$. How does your answer change as you vary the number of quadrature points from 2 to 10? (Note: You can check your numerical answer against the analytical formula $E[e^X] = e^{\mu+\sigma^2/2}$. To obtain the Gauss-Hermite weights and abscissas, you can use the program `gauher` in Chapter 4.5 of *Numerical Recipes*.)

**3.** Write a program that computes the expectation in the second problem using Monte Carlo integration, first with 1,000 and then with 4,000 draws. How do your estimates compare to the analytical formula? What are the standard errors of your estimates? (To generate standard normal random numbers, you can use the Fortran code available on the course web site at www.econ.yale.edu/smith/econ561a.)

**4.** Write a program that uses Chebyshev interpolation to approximate the function $f(x) = \log(x)$ on the interval $[0.1, 1]$. How does the approximation error compare to linear and cubic spline interpolation?

**5.** Write a program to solve the (deterministic) neoclassical growth model using Chebyshev collocation on the interval $[0.8\bar{k}, 1.2\bar{k}]$, where $\bar{k}$ is the steady-state capital stock. That is, express the optimal decision rule as a linear combination of Chebyshev polynomials up to order $n$ (i.e., degree less than or equal to $n-1$) and then choose the $n$ unknown coefficients so that the Euler equation errors are set to zero at the (appropriately scaled) roots of the $(n+1)$th-order Chebyshev polynomial. Graph the Euler equation errors as a function of capital for $n = 2,3,4$. (Note: To do this problem, you will need to write a program that uses Newton's method to find the roots of a set of $n$ nonlinear equations.)