Spring 2017: Numerical Analysis
Assignment 6 (due April 25, 2017)

2 extra credit points will be given for cleanly plotted and labeled figures (see also rules on the first assignment). Use a legend and different line styles to label multiple graphs in one plot (no colors needed). Do not export figures using raster graphics (.jpg, .png) but use vector graphics (.eps, .pdf, .dxf) that do not mess up lines. Label axes and use titles. Use `help plot`, `help legend`, `help xlabel` to better understand MATLAB's plotting capabilities.

1. **[Composite trapezoidal and Simpson sum, 4+2pt]** Write codes[1] to approximate integrals of the form

$$I(f) = \int_a^b f(t)\,dt$$

using the trapezoidal and Simpson's rule on the sub-intervals $[x_{i-1}, x_i]$, $i = 1, \ldots, m$, where $x_i = a + ih$, $i = 0, \ldots, m$ with $h = (b-a)/m$.[2]

   (a) Hand in listings of your codes, and use them to approximate the integrals

$$\int_0^1 \cos(3\pi x)^4\,dx, \qquad \int_{0.1}^1 \sqrt{x}\,dx.$$

   Compare the numerical errors $\mathcal{E}$ for both quadrature rules (the exact values of the integrals are $3/8$ and $2/3$, respectively). Try different m (e.g., $m = 10, 20, 40, 80, \ldots$) and plot the quadrature errors versus m in a double-logarithmic plot.

   (b) To numerically study how the errors $\mathcal{E}$ decrease with m, we assume that the errors behaves like Cm^κ, with to-be-determined $C, \kappa \in \mathbb{R}$. Applying the logarithm to $\mathcal{E} = Cm^\kappa$ results in

$$\log(\mathcal{E}) = D + \kappa \log(m), \qquad (1)$$

   where $D = \log(C)$. Use the values for m and $\log(\mathcal{E})$ you computed in (a) to find the best-fitting values for D and κ in (1) by solving a least squares problem. Compare your findings for κ with the theoretical estimates for the composite trapezoidal and Simpson's rules.[3]

2. **[Orthogonal polynomials, 2+2pt]** Remember that a function f is called *even* if $f(-x) = f(x)$ and *odd* if $f(-x) = -f(x)$ for all x in its domain. Let w be an even weight function on the interval $(-a, a)$ and $\{\varphi_0, \varphi_1, .., \varphi_n\}$ be a system of orthogonal polynomials on $(-a, a)$ with respect to w, constructed using the Gram-Schmidt-Orthogonalization.

[1] Ideally, you write functions `trapez(f,a,b,m)` and `simpson(f,a,b,m)`, where f is a function handle (see http://www.mathworks.com/help/matlab/matlab_prog/creating-a-function-handle.html if you are not familiar with that concept) or f is the vector $(f(x_0), \ldots, f(x_m))$.

[2] For these composite rules, see Definitions 7.1 and 7.2 in the book.

[3] Compare with (7.16) and (7.18) in the book. You can ignore the constants, just compare κ, the exponent of m, with the theoretical results.

(a) Show that, if j is even, then φ_j is an even function and if j is odd, then φ_j is an odd function.

(b) Let $f : [-a, a] \to \mathbb{R}$ and $p_n(x) = \gamma_0\varphi_0(x) + \ldots + \gamma_n\varphi_n(x)$ its best polynomial approximation of degree n with respect to the weighted 2-norm. Show that if f is an even function, then all the odd coefficients γ_{2j-1} are zero and if f is an odd function, then all the even coefficients γ_{2j} are zero.

3. **[Newton-Cotes vs. Gauss Quadrature, 2+2+2+1pt]** We discussed two methods to integrate functions numerically, namely the Newton-Cotes formulas and Gauss quadrature.

   (a) Recall that we calculated the first three orthogonal polynimals with respect to $w \equiv 1$ on $(0, 1)$ in class to be $\{\varphi_0, \varphi_1, \varphi_2\} = \{1, x - 1/2, x^2 - x + 1/6\}$. Calculate $\varphi_3(x)$ using the ansatz $\varphi_3(x) = x^3 - a_2\varphi_2(x) - a_1\varphi_1(x) - a_0\varphi_0(x)$, with appropriately computed $a_2, a_1, a_0 \in \mathbb{R}$.

   (b) Derive the Gaussian Quadrature formula for $n = 2$, i.e., calculate both the quadrature points x_0, x_1, x_2 (these are the roots of φ_3 and the corresponding weights W_0, W_1, W_2.[4]

   (c) Now we want to compare Gaussian quadrature derived in (b) with the Simpson's Rule. Use both methods to numerically find

$$I_k = \int_0^1 x^k\, dx, \qquad \text{for} \quad k = 0, \ldots, 7.$$

   Plot the errors arising in each method as a function of k. Note that to find the error, you will need to calcualte the exact values for I_k (by hand).

   (d) Explain your findings using the results on the exact integration for polynomials up to certain degrees discussed in class.

4. **[Orthogonal polynomials on $[0, \infty)$, 3+2+2pt]**

   (a) Find orthogonal polynomials l_0, l_1, l_2, l_3 for the unbounded interval $[0, \infty)$ with the weight function $\omega(x) = \exp(-x)$.[5] Plot these polynomials (they are called *Laguerre polynomials*).

   (b) As these are orthogonal polynomials, they correspond to a quadrature rule for weighted integrals on $[0, \infty)$. The resulting quadrature points and weight are given in Table 1. Verify that for $n = 2$, $n = 3$, the quadrature nodes x_i are the roots of the polynomials $l_2(x), l_3(x)$ (up to round-off).

   (c) Use the quadrature rules from Table 1 to approximate the integrals

$$\int_0^\infty \exp(-x)\exp(-x)\, dx \quad \text{and} \quad \int_0^\infty \exp(-x^2)\, dx.$$

[4] See equation (10.7) in the book.

[5] Feel free to look up the values for the indefinite integrals $\int_0^\infty \exp(-t)t^k\, dx$ $(k = 0, 1, 2, 3)$—I use Wolfram Alpha for looking up things like that: http://www.wolframalpha.com/.

Table 1: Gauss quadrature points and weights for quadrature on $[0, \infty)$.

n	x_i	W_i
2	0.585786	0.853553
	3.41421	0.146447
3	0.415775	0.711093
	2.29428	0.278518
	6.28995	0.0103893
4	0.322548	0.603154
	1.74576	0.357419
	4.53662	0.0388879
	9.39507	0.000539295

Note that, to take into account the weight $\omega(x) = \exp(-x)$, for the first integral $f(x) = \exp(-x)$ and for the second $f(x) = \exp(-x^2 + x)$. Report the errors for $n = 2, 3, 4$ using that the exact values for the integrals are $1/2$ and $\sqrt{\pi}/2$.