Assignment 3

Due. March 31, Monday, 9:00.

1. Suppose one is given n points (x_i, y_i) in the Euclidean plane $(i = 1, ..., n)$ and wants to put a smooth closed curve through them in order. One method is to draw a closed polygon joining the points in the same order and to let t stand for the arc length along the polygon $(0 \leq t \leq T)$, so that the vertices of the polygon occur as

$$0 < t_1 < t_2 < ... < t_n = T.$$

Next, one fits the data (t_i, x_i) $(i = 1, ..., n)$ by a periodic cubic spline $x(t)$ with period T. That is, in each interval $t_i \leq t \leq t_{i+1}$, $x(t)$ is a cubic polynomial, and

$$x(0) = x(T), \quad x'(0) = x'(T), \quad \text{and} \quad x''(0) = x''(T).$$

Similarly, one fits the data (t_i, y_i) with a periodic cubic spline $y(t)$.

The desired curve is then given parametrically by $x(t)$ and $y(t)$.

Your task is to modify the function `ncspline` to compute the periodic cubic splines $x(t)$ and $y(t)$. You may assume equally spaced knots and modify your tridiagonal solver in the previous assignment. Test the program by using as initial data the vertices of a regular polygon of n vertices for $n = 3$ and 4. See how close your solutions come to being circles.

2. A common problem of applied mathematics is that of solving the integral equation

$$f(x) + \int_a^b K(x,t)y(t)dt = y(x),$$

where the functions $f(x)$ and $K(x,t)$ are given and the problem is to compute $y(x)$.

If we approximate the integral by the quadrature formula

$$\int_a^b K(x,t)y(t)dt \approx \sum_{i=1}^{n} \alpha_i K(x, t_i) y(t_i),$$

then the integral equation becomes a system of linear algebraic equations:

$$f(t_j) + \sum_{i=1}^{n} \alpha_i K(t_j, t_i) y(t_i) = y(t_j), \quad j = 1, 2, ..., n.$$

The solution $y(t_i)$, $i = 1, ..., n$, is the desired discretized approximation to the function $y(t)$.

Using Simpson's rule, find an approximate solution of the integral equation

$$\frac{4x^3 + 5x^2 - 2x + 5}{8(x+1)^2} + \int_0^1 \left(\frac{1}{1+t} - x \right) y(t)dt = y(x).$$

Write a MATLAB/Octave program to solve the above equation. The main program (script file) calls a general integral equation solver

```
y = inteqn(t, kernel, fun, coef)
```

where `t` is the vector of partition points, `kernel` is the kernel function K, `fun` is the function f, and `coef` is the vector of coefficients in the quadrature rule (Simpson's rule in this case). Your program should be general enough to take any coefficient vector $[\alpha_1, ..., \alpha_n]^{\mathrm{T}}$. So, you should first generate the coefficient vector for the Simpson's rule before calling `inteqn`.

Fit a spline function to the discretized approximation of the function $y(t)$. Compare the resulting spline function with the true solution

$$y(x) = (1 + x)^{-2}$$

at various points in the interval $[0,\ 1]$.

3. A vehicle of mass M, suspended by a lumped spring-dashpot system as shown, travels with constant horizontal velocity. At time $t = 0$ the vehicle is travelling with its center of gravity on the ground and with no vertical velocity. For subsequent times, the vertical displacement of the road above the ground is given by a function $x_0(t)$.

   Suppose the spring is linear with a constant of proportionality k and that the damping coefficient of the dashpot r is a nonlinear function of the relative velocities of the two ends of the dashpot:

   $$r = r_0\left(1 + c\left|\frac{dx}{dt} - \frac{dx_0}{dt}\right|\right).$$

   It is easily shown that the displacement of the center of gravity of the vehicle $x(t)$ is the solution of the second-order ordinary differential equation

   $$M\frac{d^2x}{dt^2} = -k(x - x_0) - r\left(\frac{dx}{dt} - \frac{dx_0}{dt}\right)$$

   with the initial conditions

   $$\begin{aligned} x(0) &= 0, \\ \left.\frac{dx}{dt}\right|_{t=0} &= 0. \end{aligned}$$

   Write a MATLAB/Octave program calling `ode45`/`lsode` to calculate $x(t)$, $0 \le t \le t_{\max}$ for a road contour given by

   $$x_0(t) = A(1 - \cos\omega t),$$

   where $2A$ is the maximum road displacement above the ground.

   Note that for the linear case ($c = 0$) the underdamped, critically damped, and overdampted cases correspond to

   $$\xi = \frac{r}{2\sqrt{kM}},$$

   being less than, equal to, and greater than unity, respectively.

   The program should read values for M, r_0, c, k, A, w, and $t_{\max}$, where w corresponds to ω, and plot x, dx/dt, d^2x/dt^2, and ξ.

   Suggested test data:

   $$M = 10, \quad A = 2, \quad w = 7, \quad t_{\max} = 5, \quad k = 640.$$

   Investigate values of

r_0	c
80	0
160	1
240	10