Mast 683 Assignment 4

Professor: *Richard Hall*
Instructions: *Answer one question only. Your solution should include a careful exposition of the mathematics and program design, along with suitable specific examples to illustrate the results, using graphics where possible. Submit hard copy of all computer code and executable files on a diskette. The problem descriptions are not complete specifications: the student should discuss the chosen problem with me before starting serious work.*
Due Date: *28th November 2000.*

(4.1) Consider the numerical integration of the differential equation $y'(x) = f(x,y), \quad y(t_0) = y_0$ by the (classical) Runge-Kutta method of order 4.

(i) Define the class `rk(fun2 *f)` which has the member functions `void init(double xi, double yi), void seth(double hi), double getx(), double gety()` and `void iterate()`. We may call an instance of this class a Runge-Kutta 'engine'.

(ii) Construct an engine for the case $f(x,y) = ay\cos(bx+cy)$ and find $y(10)$ given that $a = 1, \ b = 2, \ c = 0.2,$ and $y(0) = -1$.

(iii) Now add graphics. There are many ways to do this. One idea might be simply to 'add' some member functions such as: `void rk::xplot(gwin &w, double y0)`; *this* rk::xplot plots the solution $y(x)$ satisfying $y(x_0) = y_0$, where x_0 is the first argument of the call to `gwin::scale(...)`. Use the augmented class rk to study the behaviour of the solutions of the non-linear differential equation in (ii) with respect to the parameters a, b, c. You may wish to add a further method `rk::xplots(gwin &w, double y1, double y2, int n = 10)` which plots a 'field' of n solutions with different initial y_0 values.

(4.2) Consider the quantum-mechanical eigenvalue problem

$$H\psi_n = E_n\psi_n, \quad \text{where} \quad H = -\Delta + vf(x),$$

the 'coupling parameter' $v > 0, \ n = 0, 1, 2, \ldots,$ and the 'wave function' ψ_n is in a suitable domain in the Hilbert space $L^2[\mathbb{R}]$. For example, in the case of the harmonic oscillator $f(x) = x^2$ it is known that

$$E_n(v) = v^{\frac{1}{2}}(2n+1). \tag{1}$$

If we assume that the potential 'shape' $f(x)$ has a finite minimum point and that it is monotone increasing in directions away from this point, then the WKB approximation $\mathcal{E}_n(v)$ for the eigenvalues is given by solving the following equation for $\mathcal{E}$

$$(n+\frac{1}{2})\pi = \int_{x_1}^{x_2} [\mathcal{E} - vf(x)]^{\frac{1}{2}} dx, \tag{2}$$

where x_1 and x_2 are the zeros of the integrand.

(i) Design a function called `energy(...)` which allows one to 'set' v and n and compute the corresponding approximation $\mathcal{E}_n(v)$. Test this for the harmonic oscillator $f(x) = x^2, \quad v = 1, \quad n = 0, 1, 2, \ldots, 9,$ for which the exact solution is given by (1).

(ii) Plot a graph of the unsymmetrical potential $g(x) = |x|$ for $x < 0$ and $g(x) = e^{-x^2} - 1$ for $x > 0$ on $[-10, 10] \times [0, 50]$; and repeat the calculation of (i) for this case.

(iii) Plot graphs in $[0, 10] \times [0, 20]$ of the (approximate) 'energy trajectories' $\mathcal{E}_n(v)$, $n = 0, 1, 2, 3, 4,$ for the unsymmetrical potential $g(x)$ of (ii).

HINTS There are many different ways of doing these jobs. The following are some suggestions. The classes `ker` and `wkb` are each derived classes of the class `fun`. In mathematical terms `ker`, the *kernel*, is a function of x, and `wkb` is the function of $\mathcal{E}$ whose zero is the value of the energy sought. Thus mathematically we would write

$$ker(x) = [\mathcal{E} - vf(x)]^{\frac{1}{2}}, \tag{3}$$

and

$$wkb(\mathcal{E}) = (n + \frac{1}{2})\pi - \int_{x_1}^{x_2} ker(x)dx. \tag{4}$$

To complete the toolbox for this task one might define the function `double energy(wkb *ww, int ni, double vi, double e1, double e2, double etol = 1e-6)`

for purely numerical results, and, for graphics, one could define the class `ev:fun` an instance of which, say `eg`, is, in mathematical terms, the function $eg(v) = \mathcal{E}_n(v)$. For plotting the energy trajectories corresponding to $g(x)$ I suggest that you `scale(0,10,0,20,50)` and `xplotd(eg, 0.1,10,50)` in a suitable loop: very small values of v may defeat the routine for finding the zeros x_1 and x_2.

(4.3) As in the WKB problem, consider the lowest discrete eigenvalue E_1 of the Schrödinger operator $H = -\Delta + vf(x)$, where the potential shape $f(x)$ is symmetric. Use a family of 'trial functions' with fixed shape $\phi(xs)$ and variable scale factor s. Form the *Rayleigh Quotient* given by

$$\mathcal{E}(s) = \frac{(\phi, H\phi)}{(\phi, \phi)}.$$

The best possible estimate for the Gaussian family of trial functions (and for a given v) is provided by minimizing $\mathcal{E}(s)$ with respect to s. Construct a general variational 'engine' called `ray` for this task. The constructor will set the functions f and ϕ and have a method to set the coupling parameter $v > 0$. The methods of `ray` will then compute the Rayleigh Quotient and minimize it with respect to the scale s. As a test example you can consider the potential family $f(x) = ax^2 + bx^4$ and the Gaussian 'trial function' $\phi(x) = e^{-x^2}$. This has the advantage that for $a = 1$ and $b = 0$, we know the exact answer $E_0 = \mathcal{E}(\hat{s}) = v^{\frac{1}{2}}$, where $\hat{s}$ is the optimized scale. Now estimate E_0 for

the case $a = b = 1$ with $v = 1$ (the quartic an-harmonic oscillator). Can you plot a graph of your approximation for $E_0(v)$ with $v \in [0, 10]$?

(4.4) Design a vector class `vec` for arrays of type double. Include a variety of constructors such as `vec(int n, double a, double b)` which generates a *mesh*, that is to say a set of n equally spaced points on the interval $[a, b]$; and `vec(int n, double a, double b, fptype1 f)` creates a vector in which the function f has been applied to a mesh. Now design a class called `lpoly` which is derived from vec and from fun. The constructor `lpoly(vec vx, vec vy)` receives the x and y values as vectors and creates a private vector of the coefficients for a Lagrange polynomial by using Newton's divided-difference method; meanwhile the virtual function `double lpoly::f(double)` becomes the corresponding Lagrange polynomial (arranged optimally for computations). Thus an instance of the class lpoly has two very distinct aspects. Now apply your class lpoly in a program that plots some polynomial approximations for $|x|$ and $|x|^3$, illustrating that neither of these functions is 'like' a polynomial.

(4.5) Design a class called surf which is derived from gwin. The class surf includes new data and methods which allow for the scaling of a three-dimensional block, which is then projected on to the plane (without perspective). The constructor simply creates an instance of gwin. Possible methods (returning void) might be called brot (which rotates and tilts the block and sets some private variables), bscale, bframe, bline, projx, projy, curve(f,g,h,t1,t2,n), which plots a curve in space, and various surface plotters. An ambitious goal might be to define a method called perhaps `surf2h(fptype2 f,...)` which plots a surface with 'hidden lines'.

(4.6) This problem is about the FFT. Consider the heat-flow (diffusion) equation in one spatial dimension. We let $u(x, t)$ represent the temperature of a uniform bar of length ℓ at the point x at time t. The bar is insulated on the sides and initially the temperature of the bar is given by the expression $u(x, 0) = f(x),\ 0 \le x \le \ell$. As the time increases from $t = 0$ the temperature $u(x, t)$ varies in such a way that it satisfies the diffusion equation:

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial u}{\partial x^2},$$

where a is a constant depending on the substance of the bar. For copper $a = 1.14$, for glass $a = 0.006$ $\mathrm{cm}^2\mathrm{sec}^{-1}$. For the special case where the boundary conditions are $u(0, t) = u(\ell, t) = 0$, we have the following general solution:

$$u(x,t) = \sum_{n=1}^{\infty} c_n sin(\frac{n\pi x}{\ell}) e^{-(\frac{n\pi a}{\ell})^2 t}, \qquad (a)$$

where

$$u(x,0) = f(x) = \sum_{n=1}^{\infty} c_n sin(\frac{n\pi x}{\ell}). \qquad (b)$$

Thus the procedure for finding the solution of this problem is first to find the Fourier coefficients $\{c_n\}$ of $f(x)$ from (b); then, for each t, multiply by the exponential time

factors, and transform back to find $u(x,t)$, as shown in (a). This task is in general computationally very difficult. Develop the Fast Fourier Transform algorithm FFT in the form of a class fft (an FFT engine) with constructor `fft(int m)`, where the dimension n of the discrete Fourier representation is given by $n = 2^m$. Once an instance of the engine has been constructed its functions can be used to transform a complex array of dimension n in either direction: a 'message' can be sent to the fft to change the direction of the transform. This apparatus should first be tested numerically and then it should be applied to solve the diffusion equation in the case that $\ell = \pi$ and the initial temperature is given by

$$u(x,0) = f(x) = \begin{cases} \left(\frac{x}{b}\right)^2, & \text{if } 0 \le x \le b, \\ \left(\frac{x-\pi}{\pi-b}\right)^2, & \text{if } b \le x \le \pi, \end{cases}$$

where $b = 2.5$. Write a program that plots a series of 20 graphs showing how the temperature profile $u(x,t)$ evolves in time for the case of glass. Although the diffusion problem is real, it is not necessary to generate a new fft for this special case: use the general complex one. Prove mathematically that for insulated ends, $u_t(0,t) = u_t(\ell,t) = 0$, the 'total heat' $\propto \int_0^\ell u(x,t)dx$ is invariant with respect to t. If time permits, it would be interesting to illustrate this case too.

(4.7) Solve the problem of finding the shape of a vibrating circular drum. You will first need a Runge-Kutta engine `rk2d` for the initial-value problem $y'' = f(x,y,y')$. Your program should solve the `ode` for the radial Bessel function, and provide a graphical illustration of this. The program should also provide a graphical illustration of the shape of the drum surface at a given time and for given values of the radial and angular node numbers $m \ge 0$ and $n \ge 0$.

(4.8) Consider the continuous functions $f : [0,1] \to [0,1]$ which satisfy $f(0) = f(1) = 0$ and are unimodal. Suppose the max of $f(x)$ is $\lambda \le 1$. An example of such a function is $f_1(x) = 4\lambda x(1-x)$. Write a graphics program that allows the user to explore the iterations $\{x_n = f \circ f \circ \ldots f(x_0)\}$ of such a function, starting at the initial value $x = x_0$. For a given f the sequence $\{x_n\}_{n=0}^\infty$ depends on the initial point x_0 and on λ. Explore this with your program and comment on the dependence of the sequence on λ (for 'most' x_0).

(4.9) Write a program to explore prime numbers. The program should have a tabular area that would generate a table of prime numbers in a given range, and a graphical area that would exhibit a graph of $\pi(x)$, the number of primes $\le x$. The graphical area would also show the values of one or more approximations to $\pi(x)$. The generation of prime numbers and the design of approximations to $\pi(x)$ are two minor 'industries' within number theory. The program should accommodate numbers up to the largest `long integer`. This may mean that, for some lines of code, that temporary `double` approximations to integers will have to be used. A more serious program would, of course, have to include a new 'very large' integer type in order to transcend the arbitrary limitations of `long integer`.