Numerical Analysis

Math 370 — MWF 2:30-3:25
Spring 1997 — 201 Fowler

(03.31.97) Numerical Differentiation and Integration

Warm-Up Let's repeat the exercise from the previous class which illustrates that we can never forget we do calculations on actual machines and this means round-off error is ever present. We'll use centered-differences to repeatedly approximate $f'(1)$ where $f(x) = e^x$. Your $h = 10^{-k}$ for $k = 1, 2, 3, \ldots$.

h	$f(1+h)$	$f'(1) \approx$	Error $= f'(1) - e$

Does it appear as if there an optimal value of h to use?

Explanation of Optimal h

Consider that whenever you make a function evaluation there is really some round-off error associated with it, so that $f(x) = \tilde{f}(x) + e(x)$. The $\tilde{f}$ is the machine number approximation to the exact f that you are trying to compute and e is the round-off error that the computer produces due to way it represents numbers in memory.

If we look at the forward difference formula we can obtain an expression for the error involved in approximating the derivative due to round-off error, $E(h)$

$E(h) =$ $\qquad\qquad\qquad\qquad \leq$

Let's assume that M is a bound for f'' and that ϵ is a bound on the round-off error. Then we can get an expression for $E(h)$ and we can find out what value of h **minimizes** it.

$E(h) =$

Optimal value of h
Let's use the optimal value of h and compute the approximation to $f'(1.8)$ using the forward difference approximation.

Can we always find this optimal value of h?

Numerical Integration

One of the most famous problems in mathematics is one of **quadrature**. That is, finding a square which has the same area as some other curvilinear shape. The standard problem is the problem of "squaring the circle."

From Calculus you know that the area under a curvlinear shape given by $y = f(x)$ on an interval [a,b] is given by:

Thus the problem of quadrature becomes one of approximating an integral. The method we will use is known as numerical quadrature and involves using function values $f(x_i)$ which are known at $N+1$ distinct nodes $a = x_0 < x_1 < \ldots < x_N = b$. We will try and come up with formulas such that

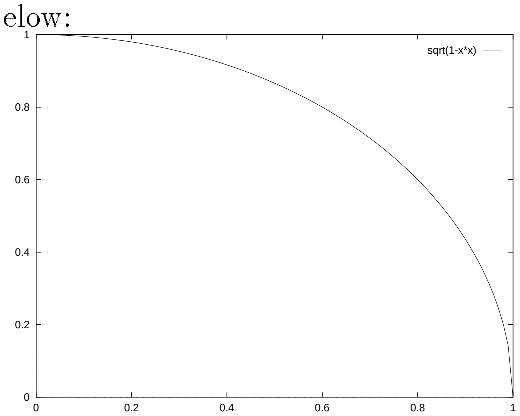
$$\int_a^b f(x)dx \approx \sum_{i=o}^{N} a_i f(x_i)$$

One way to derive these formulas is to again apply our knowledge of Lagrange interpolation to approximate $f(x)$ and then integrate this formula and simplify.

Before we illustrate how this general formula can be applied to give us many useful formulae let's recall what we already know about numerical quadrature formulae.
What methods do you know of approximating the area under a curve?

Example

Let's try and approximate the area of a unit circle using these methods of numerical quadrature you already know.
We will do this by estimating the area under the curve $f(x) = \sqrt{1 - x^2}$ from 0 to 1, depicted below:



How close did you get to the exact value of the area of the circle?
What order do you think the quadrature methods you used are? i.e. $O(h)$? $O(h^2)$? Something else?

2-point Quadrature Rule

Let's assume that we only have 2 data points $(x_0, f(x_0))$ and $(x_1, f(x_1))$. We can use a first degree Lagrange polynomial to interpolate $f(x)$, integrate it and thus derive a quadrature rule we call __________________