MATHCALC Dr. David Jagerman AT&T Bell Laboratories

ABSTRACT

This paper presents a number of programs which have been found useful by the author and which have been developed over the past years in response to the needs which have arisen. Some of these needs included the investigation of queueing models with finite buffers, the inversion of generating functions arising from semi-Markov processes - used in the study of error rates for the transmission of data, and in the study of communication protocols - and in the solution of difference equations arising from some models of computer delay. The code for each program is written in the form of individual applications since they were originally so used, however some of them were completely reworked by W. Obi for use as applications and library functions using C and FORTRAN and to be accessed under the UNIX environment; notably, VIEQ, LAPINV, GEN, CI, and certain of the quadrature programs.

Many of the quadrature programs employ methods which may be considered unusual since they do not depend on the traditional numerical analysis procedures constructed from algebraic polynomials, they are rooted instead in the theory of cardinal series interpolation and quadrature. A basic form is established for quadrature on the entire real line, subsequently mapping is used as the tool to effect quadrature on subsets.

An introductory discussion of the theory underlying cardinal series quadrature is given before the main discussions of the individual programs. Each program is discussed systematically and at least one example of its use is given. One of the quadrature programs, however, INTO-OO, is based on the Norlund form of the Euler-Maclaurin expansion and utilizes properties of the eigenvalue of the homogeneous Norlund sum. This may be considered to be a departure from the usual mode of applying the Euler-Maclaurin series. Another program containing a variation from the usual is VIEQ (Volterra integral equation solver) which uses a specially designed quadrature algorithm of greater accuracy than the simple trapezoidal rule but possessing a robustness almost as great. Certain of the traditional procedures are also given - Newton's method for finding zeros, Simpson's rule of quadrature, Runge-Kutta solution of differential equations, and a method of computing the Fourier coefficients of a periodic function. Somewhat unusual is the implementation of Tartaglia's explicit solution of the cubic equation in program CUBIC rather than reliance on iteration procedures.

The programming language which the author uses is PC/FORTH (Laboratory Microsystems Inc.). This language is very convenient for the development of scientific applications; thus, the code for all programs given in this paper are in PC/FORTH which serves as a model for program description. However, in order that these programs be accessible to the UNIX system, a C version of the code is being prepared by the author and will also be given in this paper. Functions which are required in many of the C programs should be written in a separate file to be linked with the main program. For those programs using complex number routines, a separate file is given containing those routines and which is also to be linked to the main program calling them.

The Thunderbird (Mayan Eagle) used as a motif introducing the FORTH code is intended to symbolize the *speed and power* of FORTH while the Unicorn motif introducing the C code symbolizes the *grace and elegance* of C.