DEMONSTRATION FORTRAN PROGRAMS.

DIRECTORY 903FORTRAN

There are twelve demos:

DEMO1.DAT: Table of Squares based on Hunter ALGOL DEMO1 using the FORTRAN Load and Go (LG) system.

DEMO2.DAT: Computes EXP(X) for a range of values of X using Taylor's series and compares with the result of the EXP function supplied by 903 FORTRAN. Run using the two-pass system in batch mode. (This demo is the FORTRAN equivalent to DEMO2 in 920ALGOL.)

DEMO3.DAT: Computes the roots of the equation f(x) = cosh x + cos x -3 = 0 using the Newton-Raphson method. Run using the two-pass system in relocatable mode. (This demo is the equivalent to DEMO1 in 903ALGOL, and DEMO5 in 905FORTRAN.)

DEMO4.DAT: Computes sines for input angles using Taylor's series and compared against the SIN function supplied by 903 FORTRAN. Run using the large program system. Note the significant errors that arise for large angles. (This demo is the FORTRAN II equivalent to DEMO2 in 903ALGOL and DEMO6 in 905FORTRAN).

DEMO5.DAT: Computes three digit numbers that are equal to the sum of the cubes of their digits (The "Stickler Problem" from Forsythe, Keenan, Organick and Stenberg, FORTRAN Language Supplement, page 112. Run using the load and go system.

DEMO6.DAT: Solves sets of simultaneous linear equations using the Gauss-Seidel method. Run using the large program system. This demo is the FORTRAN II version of DEMO11 in 905FORTRAN.

DEMO7.DAT: Graphs a decaying oscillating electric circuit on the teletype. Run using the load and go system. This demo is the FORTRAN II version of DEMO12 in 905FORTRAN.

DEMO8.DAT: Graphs solutions to the Laplace heat transfer equation for a pipe carrying a hot liquid through an ice bath. The output is a contour map of heat ranges. The size of the array (59 * 59) is the largest that can be used without provoking a store full error. Run using the large program

system. This demo is the FORTRAN II version of DEMO13 in
905FORTRAN.

DEMO9.DAT: Prints out solutions to input quadratic equations.
Demonstrates use of subroutines and GLOBAL statement. Run
using load and go system. This is the FORTRAN II version of
905FORTRAN DEMO14.

DEMO10: An example of the use of code sections. The function
sub-program DSUMSQ calculates the sum of the squares of the
elements on the diagonal of a square matrix using SIR rather
than FORTRAN. DSUMSQ is called with local, common and formal
array parameters to illustrate the techniques discussed in the
notes on FORTRAN code sections. An equivalent demonstration
for 905 FORTRAN is in the file DEMO16.DAT in 905FORTRAN.

DEMO11: this program calculates the constant pi to up to 300
decimal places using Machin's formula. It illustrates how to
use arrays for multi-precision arithmetic, FORTRAN functions
and procedures and the use of COMMON storage. It takes a real
903 about one hour to calculate pi to 300 decimal places.
Under the simulator it takes about 10 seconds!

DEMO12: this program shows the use of the MTFORTRAN library
for using magnetic tapes. It is a modified version of DEMO6
(Solving simultaneous equations using the Gauss-Siedel
method). The DEMO12 version is intended to work on very large
sets of equations with up to 400 unknowns and therefore the
matrix of equation coefficients is too large for an 8K or 16K
store). Instead each row of the matric is written as a block
on tape and read in when needed. The MTFORTRAN routines are
essentially the same as those of MTALGOL, modified to work
with the FORTRAN runtime system. The most notable difference
is the handling of tape names by MTOPEN. If the call to
MTOPEN is followed by a FORMAT statement containing an initial
Hollerith string, that string is taken to be the name of the
tape file to open.

The script starts by initializing a tape using MTINIT and then
runs the main program. The program first opens the tape and
the coefficients of the equations to solve are input and
copied to tape. The input format is as for DEMO6, but
requires that the data be presented in row order (column order
within rows can be arbitrary).

Once the data has been read, the program executes the Gauss-
Siedel iteration using MTREAD to read in each block as
required. Since tape blocks are numbered, starting at block 1

for the header, data block (n+1) will correspond to row (n) of the matrix of coefficients. By asking MTREAD to find a numbered block, the tape will be automatically rewound after every iteration.

When the iteration converges, the results are output and the tape rewound.

Note that MTFORTRAN does not include equivalent routines to the ALGOl raget and raput procedures to allow real numbers to be transferred to and from tape buffers. Instead the tape buffer is declared as real and put in COMMON so that simple wrapper routines can be provided that overlay integer arrays over the same COMMON area, suitable for use by MTOPEN, MTREAD and MTWRITE.

Note all the demos specify use of the 903 instruction architecture – this is because the 16K versions of FORTRAN (load and go, large program versions) require the precise behaviour of the way the 903 enables initial instructions rather than the simplified behaviour implemented for the 900 instruction architecture.

System files.

903FORT16KLG(ISS5).BIN - "903 FORTRAN, 16K LG ISS 5 Copy 502": The load and go system paper tape in binary format suitable for loading by initial instructions.

903FORT16KLP(ISS4).BIN - "903 FORTRAN 16K LP, ISS 4 Copy 502". The large program system paper tape in binary format suitable for loading by initial instructions

903FORT1(ISS4C).BIN - "903 FORTRAN TAPE 1, ISSUE 4C". The FORTRAN translator paper tape in binary format suitable for load by initial instructions.

903FORT2(ISS4A).BIN instructions "903 FORTRAN TAPE 2, ISS 4A Copy 502": The FORTRAN runtime, in binary format suitable for load by initial instructions.

MTFORTRAN(ISS3).900 - "MTFORTRAN ISS 3 Copy 259": The SIR source of the MTFORTRAN routines.