1991 NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

JOHN F. KENNEDY SPACE CENTER
UNIVERSITY OF CENTRAL FLORIDA

TRANSIENT STUDY OF A CRYOGENIC HYDROGEN FILLING SYSTEM

PREPARED BY:	Mr. Howard Schleier
ACADEMIC RANK:	Associate Professor
UNIVERSITY AND DEPARTMENT:	Norwalk State Technical College Mathematics/Science Department
NASA/KSC	
DIVISION:	Mechanical Engineering
BRANCH:	Special Projects
NASA COLLEAGUE:	Gary Lin Eric Thaxton
DATE:	August 21, 1991
CONTRACT NUMBER:	University of Central Florida NASA-NGT-60002 Supplement: 6

I

ACKNOWLEDGEMENTS

The author would like to thank his colleagues Gary Lin and Eric Thaxton for their leadership, guidance, and inspiration. It was their efforts that made this project possible. Sincere appreciation is also given Dr. E. Ramon Hosler and Ms. Kari Stiles, whose charismatic personalities insured the success of this project. The staff at DM MED at KSC, who were friendly and thoughtful, who accepted the author as one of them, and who never hesitated to attend to my needs are certainly worthy of the author's gratitude. Acknowledgement is also given to the staff at VAX Operations, and the librarians at KSC who were always ready to render first class service at the drop of a hat. Finally, mention is given to the remaining staff at KSC for their part in making this summer a memorable experience.

ABSTRACT

2.1 PREPARATION. An investigation was made as to producing a workable model for the transient analysis of a cryogenic hydrogen filling system. A series of programs and subprograms defining: the momentum, mass, and energy balances, the physical properties, the transport properties, and their interactions were devised.

2.2 TEST. The program was modified for a simple theoretical test fluid. Exhaustive runs and modifications were made and at this point no stability has been achieved except in trivial cases.

SUMMARY

3.1 SCOPE AND PURPOSE. This investigation was of a theoretical nature. A pressure driven, flow controlled cryogenic filing system was modeled to accept time dependent input. The system consists of an upstream tank (feedtank), upstream piping, a control element, downstream piping,and a downstream tank (external tank). The piping configuration is contained in block data, while certain initial conditions, ambient temperature, run time, and time interval, are entered as input.

3.2 RANGE OF VARIABLES. Though the algorithm is not limited in its variable range, some practical restrictions do exist. No inconsistency with the fundamental scientific constraints should be entered. Piping should be of order 1km, diameters 0.2m to 0.5m, and node lengths 1m to 10m. The time interval should be chosen such that it is no more than 10% of the residence time of the smallest node, however it should be long enough to avoid computational error. Pressures range from 1E5Pa to 5E5Pa, while temperatures between 20K and 300K are considered. Mass flows of course should be consistent with the 10% of the residence time restriction.

3.3 RESULTS. The transport and physical property routines used previously were reprogrammed. Most of them worked fairly well while others did not work at all. Several were improved upon. A transient cryogenic hydrogen filling process was programmed and tested with a fictitious fluid. Except in trivial cases, the program failed to reach convergence as of this date.

TABLE OF CONTENTS

Abbreviations and Acronyms List

VARIABLE	TYPE	DEFINITION	UNITS
CP	INTEGER*2	NODE # OF CONTROL POINT	LESS
ET	"	" " " EXTERNAL TANK	"
FRAME	"	TIME INTERVAL #	"
K	"	FITTING TYPE	"
NODE	"	NODE #	"
PUMP	"	NODE # OF PUMP	"
D	REAL*8	INSIDE DIAMETER	M
DELTAT	"	TIME INTERVAL	S
DO	"	OUTSIDE DIAMETER	M
E	"	U+V**2/2+G*Z	M**2/S**2
EPS	"	PIPE ROUGHNESS	M
H	"	ENTHALPY	M**2/S**2
H0	"	H+V**2/2+G*Z	M**2/S**2
H	"	HEAT TRANSFER COEFFICIENT	KG/S**3/K
F	"	FANNING FRICTION FACTOR	LESS
K	"	THERMAL CONDUCTIVITY	KG*M/S**3/K
KCO	"	TOTAL DISCHARGE COEFFICIENT	LESS
KELVIN	"	BULK TEMPERATURE	K
KELWAL	"	WALL TEMPERATURE	K
M	"	MASS OF NODE	M
MDOT	"	MASS FLOW	KG/S
MU	"	VISCOSITY	KG/M/S
P	"	PRESSURE	KG/M/S**2
PF	"	FRICTIONAL PRESSURE DROP	KG/M/S**2
PET	"	EXTERNAL TANK PRESSURE	KG/M/S**2
PFT	"	FEED TANK PRESSURE	KG/M/S**2
Q	"	HEAT FLUX	KG/S**3
QEXT	"	EXTERNAL HEAT FLUX	KG/S**3
QMAX	"	MAXIMUM HEAT FLUX	KG/S**3
REY	"	REYNOLDS NUMBER	LESS
RHO	"	DENSITY	KG/M**3
U	"	INTERNAL ENERGY	M**2/S**2
TAMB	"	AMBIENT TEMPERATURE	K
TIME	"	ELAPSED TIME	S
TFINAL	"	TOTAL ELAPSED TIME	S
V	"	VELOCITY	M/S
VET	"	EXTERNAL TANK VOLUME	M**3
X	"	QUALITY	LESS
Z	"	ELEVATION	M

VI

BODY OF TEXT

6.1 INTRODUCTION

6.1.1 SUBJECT. The subject of this project is the transient study of a cryogenic line. The intricate procedure involved in the filling of the external tank on the shuttle, coupled the large capacity of a long filling system, suggests a complicated transient. Predicting the nature of this transient might enable NASA to develop a more efficient filling procedure.

6.1.2 STATUS OF THE PROBLEM. NASA had developed some software to analyse transient flow of cryogenic fluids. However, all but one of these programs, fail to include the effects of heat transfer and two-phase flow. The program that includes these effects the TCTP(1) or "transient cryogenic transfer program". TCTP has reasonable success analyzing LOX systems, but is not as reliable for LH2-H2 modeling. TCTP is not well structured and is very poorly documented. after careful analysis, it was decided to start anew rather than modify TCTP.

6.1.3 SIGNIFICANCE OF WORK DONE. "H2FILL", a highly structured and rigidly documented FORTRAN code was developed to solve the problem. In the course of attempting to modify TCTP a set of subprograms defining the properties of hydrogen were developed. "H2FILL" is still being analyzed for run time stability at this point.

6.2 MAIN TEXT-DESCRIPTIVE INFORMATION

6.2.1 BACKGROUND. In the process of trying to decide whether to modify TCTP or start anew, the author investigated the possibility of locating software that might flowsheet the TCTP program and thereby facilitate the possibility of deciphering it. The "TAMU" code was located (2). It was available for $3500. After weighing the trade-offs between modification and initiation,it was decided to go with the latter.

6.2.2 CONFIGURATION. The configuration consists of the feedtank, the upstream piping, the control point (control valve), the downstream piping, and the external tank.

The parameters of this configuration are described in the BLOCK DATA statement "H2BLOCK". The time sensitive parameters are read in from an input list "H2LIS,LIS'.

6.2.3 PROGRAM. Program "H2FILL" the mainline program calls in four subroutines in sequence. These subroutines are: 1)"H2INPUT", 2)"H2INIT", 3)"H2MOMNRG", AND 4)"H2OUTPUT". The data between the subroutines is transferred by an unlabeled COMMON.

6.2.3.1 Input. Subprogram "H2INPUT" reads in the initial inventories in the tanks, the ambient temperature, and the initial pressures in the tanks. Then, the time of the the run and the time interval are read. The remaining data read in are the tank pressures and the mass flow at each non-zero time interval.

6.2.3.2 Initiation. Subprogram "H2INIT" initializes the conditions in the nodes. "H2INIT" also establishes the initial properties through "H2LSAT", "PROPHP",and "PROPPTGS".In addition it returns elevation from the inventory through "FEEDTANK" and "EXTANK".

6.2.3.3 Processing. "H2MOMNRG" simultaneously solves the momentum, continuity, and energy equations; the properties and flows at each node are updated. "TRANS" updates the transport coefficients and the wall temperature at each node from the properties and the design data.

6.2.3.4 Output. "H2OUTPUT" creates a file "H2DAT.DAT" on which it prints the data in spreadsheets. The first set is a node scan of each time. The second set is a time scan of each node.

6.3 MATHEMATICAL PRESENTATION

6.3.1 MOMENTUM From Newton's second law (1)
F=d(M*V)/dTIME+(MDOT*V)out-(MDOT*V)in
where F is the net force.

The parameters of this configuration are described in the
and

```
M=RHO*PI/4*D**2*L                                              (2)
```

for a cylinder.Also note that

```
dF=-PI/4*D**2*(dP+dPF+RHO*G*dZ)                                (3)
```

and

```
MDOT=RHO*V*PI/4*V*D**2                                         (4)
```

combining (1), (2) (3), and (4) and integrating we get

```
(P(NODE,FRAME-1)-P(NODE-1,FRAME-1)+
(RHO(NODE,FRAME-1)*V(NODE-1,FRAME-1)**2-
RHO(NODE-1,FRAME-1)*V(NODE-1,FRAME-1)**2)+
PF(NODE,FRAME-1)+G*(Z(NODE)-Z(NODE-1))=L(NODE)/DELTAT*
                    (RHO(NODE,FRAME-1)*(V(NODE,FRAME-1)-
                    RHO(NODE,FRAME)*V(NODE,FRAME))             (5)
```

Substituting (4) in (5) and rearranging the following appears:

```
MDOT(NODE,FRAME)=MDOT(NODE,FRAME-1)-
                 PI/4*D(NODE)**2*DELTAT/L(NODE)*
                 (P(NODE,FRAME-1)-P(NODE-1,FRAME-1)+
                 RHO(NODE,FRAME-1)*V(NODE,FRAME-1)**2-
                 RHO(NODE-1,FRAME-1)*V(NODE-1,FRAME-1)**2
                 +PF(NODE,FRAME-1)+RHO(NODE,FRAME-1)*
                 G*(Z(N0DE)-Z(NODE-1)))                        (6)
```

which updates the mass flow.

6.3.2 CONTINUITY. For mass conversation one sees that

```
dM/dTIME=MDOTin-MDOTout                                        (7)
```

integrating we get

```
M(NODE,FRAME)=M(NODE,FRAME-1)+DELTAT*
              MDOT(NODE-1,FRAME)-MDOT(NODE,FRAME))             (8)
```

and

```
RHO(NODE,FRAME)=M(NODE,FRAME)/
                (PI/4*D(NODE)**2*L(NODE))                      (9)
```

also

```
V(NODE,FRAME)=MDOT(NODE,FRAME)/
              (RHO(NODE,FRAME)*PI/4*D(NODE,FRAME)**2)          (10)
```

hence the momentum and mass conservation laws has allowed us to update MDOT, M, RHO, and V. However, in in order to ascertain the state of the system we need another independent variable. Energy conservation will satisfy this requirement.

6.3.3 ENERGY. For the energy conservation (1) through (8)

```
d(MDOT*E)/dTIME=(MDOT*(U+V**2/2+G*Z))in
                -MDOT*(U+V**2/2+G*Z))out
                +(heat)in-(work)out,(7)                        (11)
where (work)out=(MDOT*P/RHO)out-(MDOT*P/RHO)in,(1)             (12)
and (heat)in=HC*PI*D*L*(KELWAL-KELVIN),(1)                     (13)
```

combining (11), (12), and (13) and noting the definition of H0 we obtain:

```
d(M*E)/dTIME=(MDOT*H0)in-(MDOT*H0)out+
             HC*PI*D*L*(KELWAL-KELVIN),(7).                (14)
```

Integrating and solving for the update of E gives

```
E(NODE,FRAME)=(((MDOT(NODE-1,FRAME-1)*H0(NODE-1,FRAME-1)-
                MDOT(NODE,FRAME)*H0(NODE,FRAME)+
                HC(NODE,FRAME-1)*PI*D(NODE)*L(N0DE)*
                (KELWAL(NODE,FRAME)-KELVIN(NODE,FRAME)))*
                DELTAT+M(NODE,FRAME-1)*E(NODE,FRAME-1)/
                M(NODE,FRAME)                              (15)
```

then we update U

```
U(NODE,FRAME)=E(NODE,FRAME)-V(NODE,FRAME)**2/2-G*Z(NODE)(16)
```

With U and RHO determined the state of the system is also determined.Therefore P, H, KELVIN, MU, K and X can all be updated. A combination of physical properties an the flow conditions then can produce F, HC, and eventually KELWAL, PF, and H0. Hence the update would be complete, and the next iteration can proceed.

6.4 DISCUSSION

6.4.1 HISTORY. NASA has several programs that simulate the cryogenic fueling systems at the Kennedy Space Center. However, only one of those algorithms accounts for both heat transfer and two-phase flow. The program with both of these attributes is the TCTP code. TCTP seems to be satisfactory for oxygen, but fails in describing hydrogen fueling. The authors' suspicion is that the hydrogen properties are "stiff", and are not converging with the numerical techniques used in TCTP.

6.4.2 AGENDA. In order to produce a program that would successfully model the hydrogen fueling system, the question of whether to modify the existing TCTP, or to start anew had to be contemplated. Ordinarily, the easier path would seem to be to attempt modification of the existing code. However, TCTP turned out to be very poorly structured, and hardly documented at all. Due to the above, and the fact that something might be gained from a new approach, it was decided that a new approach would be the way to go.

6.4.3 PROJECT. A highly structured intricately documented FORTRAN code was started and developed for numerically solving the momentum, continuity, and energy equations.

VII

CONCLUDING SECTION

7.1 CONCLUDING REMARKS

7.1.1 PHYSICAL PROPERTIES. A battery of subprograms describing the properties of cryogenic hydrogen were developed. The mode was made compatible to the SI system of units. Most of the data agreed reasonably well with NBS data, while did not agree well at all. On several points the author improved on the existing algorithms. Extensive documentation and perfect structure was used at all times.

7.1.2 TRANSPORT PROPERTIES. An improved algorithm was produced for flow conditions and the thermal equations were rewritten.

7.1.3 TRANSIENT CRYOGENIC FLOW ALGORITHM. A highly structured, extensively documented program was written. A ficticious fluid was introduced and at this point convergence to reasonable numbers has not as yet been achieved.

7.1.4 RECOMMENDATIONS. The author that some minor revision would stabilize the run time condition of H2FILL. The author also believes that when actual cryogenic hydrogen properties are evaluated the previous troubles experienced with TCTP will reappear. A more novel method of presenting the equation of state, might alleviate the problem. Differential techniques with smoothing might provide the answer. Other alternatives in the numerical analysis of the equation of state might also be tried. One might also note that proper adjustment of the input parameters can also bring about convergence.

VIII

APPENDIX

8.8 APPENDIX A

8.8.1 TIME SCAN OF NODES

```
FEED TANK AT NODE =  0
CONTROL POINT AT NODE =  4
EXTERNAL TANK AT NODE = 35
NODE #  =  3
```

TIME	QUALITY	PRESSURE	TEMPERATURE	WALL TEMP.	DENSITY	VELOCITY	MASS	MASS FLOW
0.000000D+00	0.000000D+00	0.499421D+06	0.203967D+02	0.203900D+02	0.589015D+01	0.000000D+00	0.185044D+01	0.000000D+00
0.150000D-01	0.000000D+00	0.499421D+06	0.203967D+02	0.203900D+02	0.589015D+01	0.540411D-11	0.185044D+01	0.100000D-11
0.300000D-01	0.000000D+00	0.499421D+06	0.203967D+02	0.203900D+02	0.589015D+01	0.540411D-11	0.185044D+01	0.100000D-11
0.450000D-01	0.000000D+00	0.499421D+06	0.203967D+02	0.203900D+02	0.589015D+01	0.540411D-11	0.185044D+01	0.100000D-11
0.600000D-01	0.000000D+00	0.499421D+06	0.203967D+02	0.203900D+02	0.589015D+01	0.540411D-11	0.185044D+01	0.100000D-11
0.750000D-01	0.000000D+00	0.499421D+06	0.203967D+02	0.203900D+02	0.589015D+01	0.540411D-11	0.185044D+01	0.100000D-11
0.900000D-01	0.000000D+00	0.499421D+06	0.203967D+02	0.203900D+02	0.589015D+01	0.540411D-11	0.185044D+01	0.100000D-11
0.105000D+00	0.000000D+00	0.499421D+06	0.203967D+02	0.203900D+02	0.589015D+01	0.540411D-11	0.185044D+01	0.100000D-11
0.120000D+00	0.000000D+00	0.499421D+06	0.203967D+02	0.203900D+02	0.589015D+01	0.540411D-11	0.185044D+01	0.100000D-11
0.135000D+00	0.000000D+00	0.499421D+06	0.203967D+02	0.203900D+02	0.589015D+01	0.540411D-11	0.185044D+01	0.100000D-11
0.150000D+00	0.000000D+00	0.499421D+06	0.203967D+02	0.203900D+02	0.589015D+01	0.540411D-11	0.185044D+01	0.100000D-11
0.165000D+00	0.000000D+00	0.499417D+06	0.203966D+02	0.203900D+02	0.589015D+01	0.540411D-11	0.185044D+01	0.100000D-11
0.180000D+00	0.000000D+00	0.499426D+06	0.203969D+02	0.203900D+02	0.589015D+01	0.540411D-11	0.185044D+01	0.100000D-11
0.195000D+00	0.000000D+00	0.499696D+06	0.204078D+02	0.203900D+02	0.589018D+01	0.540407D-11	0.185046D+01	0.100000D-11
0.210000D+00	0.000000D+00	0.499607D+06	0.204042D+02	0.203900D+02	0.589017D+01	0.540408D-11	0.185045D+01	0.100000D-11
0.225000D+00	0.000000D+00	0.482641D+06	0.197145D+02	0.203900D+02	0.586803D+01	0.540605D-11	0.184978D+01	0.100000D-11
0.240000D+00	0.000000D+00	0.453474D+06	0.185376D+02	0.203900D+02	0.582462D+01	0.540918D-11	0.184671D+01	0.100000D-11
0.255000D+00	0.000000D+00	0.168173D+07	0.672791D+02	0.203900D+02	0.601307D+01	0.529363D-11	0.188906D+01	0.100000D-11
0.270000D+00	0.000000D+00	-0.126012D+08	-0.467474D+03	0.203900D+02	0.648448D+01	0.490880D-11	0.203716D+01	0.100000D-11
0.285000D+00	0.000000D+00	0.239541D+10	-0.461194D+06	0.203900D+02	-0.871501D+00	-0.365243D-10	-0.273790D+00	0.100000D-11
0.300000D+00	0.000000D+00	-0.794325D+11	0.432888D+07	0.203900D+02	-0.441410D+01	-0.721121D-11	-0.138673D+01	0.100000D-11

8.8.2 NODE SCAN OF TIME

FEED TANK AT NODE = 0
CONTROL POINT AT NODE = 4
EXTERNAL TANK AT NODE = 35
ELAPSED TIME = 0.0000000D+00

NODE	QUALITY	PRESSURE	TEMPERATURE	WALL TEMP.	DENSITY	VELOCITY	MASS	MASS FLOW
0	0.0000000D+00	0.5000000D+06	0.2039000D+02	0.0000000D+00	0.5198492D+01	0.0000000D+00	0.1000000D+03	0.0000000D+00
1	0.0000000D+00	0.4894210D+06	0.2039467D+02	0.2039467D+02	0.5196150D+01	0.0000000D+00	0.1000000D+03	0.0000000D+00
2	0.0000000D+00	0.4894210D+06	0.2039467D+02	0.2039000E+02	0.5196150D+01	0.0000000E+00	0.1850464D+01	0.0000000D+00
3	0.0000000D+00	0.4954210D+06	0.2039467D+02	0.2039000D+02	0.5196150D+01	0.0000000D+00	0.1850464D+01	0.0000000D+00
4	0.0000000D+00	0.2000467D+06	0.1000260E+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.0000000D+00	0.1810460D-11
5	0.0000000D+00	0.2000467D+06	0.1000260E+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
6	0.0000000D+00	0.2000467D+06	0.1000260E+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
7	0.0000000E+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
8	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
9	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
10	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
11	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
12	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
13	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
14	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
15	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
16	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
17	0.0000000D+00	0.2000467D+06	0.1000260E+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
18	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
19	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
20	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
21	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
22	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000E+00	0.1259377D+01	0.0000000D+00
23	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
24	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
25	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
26	0.0000000E+00	0.2000467D+06	0.1000260E+01	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000E+00
27	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
28	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
29	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
30	0.0000000D+00	0.2000467D+06	0.1000260D+01	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
31	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
32	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
33	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
34	0.0000000D+00	0.2000467D+06	0.1000260D+03	0.1000260D+03	0.1663396D+00	0.0000000D+00	0.1259377D+01	0.0000000D+00
35	0.0000000D+00	0.2000000D+06	0.1000000D+03	0.0000000D+00	0.1668372D+00	0.0000000D+00	0.1000000E+03	0.0000000E+00

8.9 APPENDIX B

8.9.1 FORTRAN PROGRAM H2FILL

```
      program h2fill
     integer*2 cp,et,frame,kc(5,0:99),node,pump
     real*8 cpr(0:99,0:99),cvo(0:99,0:99),
    &       d(0:99),deltat,do(0:99),e(0:99,0:99),eps(0:99),
    &       h(0:99,0:99),hc(0:99,0:99),
    &       h0(0:99,0:99),f(0:99,0:99),k(0:99,0:99),
    &       kco(0:99,0:99),kelvin(0:99,0:99),kelwal(0:99,
    &       0:99),l(0:99),m(0:99,0:99),mdot(0:99,0:99),
    &       mu(0:99,0:99),p(0:99,0:99),pf(0:99,0:99),
    &       pet(0:99),pft(0:99),q(0:99,0:99),
    &       qext(0:99,0:99),
    &       qmax(0:99),rey(0:99,0:99),
    &       rho(0:99,0:99),tamb,tfinal,time(0:99),
    &       u(0:99,0:99),v(0:99,0:99),
    &       vet,x(0:99,0:99),z(0:99)
      common
    &       cp,et,frame,kc,node,pump,
    &       cpr,cvo,
    &       d,deltat,do,e,eps,
    &       h,hc,h0,f,k,
    &       kco,kelvin,kelwal,
    &       l,m,mdot,
    &       mu,p,pf,pet,
    &       pft,q,qext,qmax,
    &       rey,rho,tamb,tfinal,
    &       time,u,
    &       v,vet,x,z
C*********************************************************************
C     H2FILL CALCULATES THE TRANSIENT OF A PRESSURE DRIVEN H2 FILL  *
C     SYSTEM. THE PROGRAM REQUIRES THAT INITIAL CONDITION IN THE    *
C     TWO TANKS,THE MASS FLOW AT THE CONTROL POINT,AND THE AMBIENT  *
C     TEMPERATURE BE GIVEN. TIME INDEPENDENT DATA ARE INCLUDED IN A *
C     BLOCK DATA STATEMENT. A SINGLE UNLABLED COMMON IS USED IN ALL *
C     SUBPROGRAMS FOR TRANSFER OF CONTROL. BOUNDARY CONDITIONS RE-  *
C     QUIRED ARE THE PRESSURES AT THE TOP OF THE TANKS,AND THE MASS *
C     FLOW AT EACH TIME INTERVAL.                                   *
C*********************************************************************
C*********************************************************************
C     ALL DIMENSIONED VARIABLES ARE IN THE SI SYSTEM INTERNALLY.    *
C     ON I/O NON-SI DATA ARE CONVERTED TO/FROM SI DATA AT THE  I/O  *
C     OR BLOCK DATA INTERFACE.                                      *
C*********************************************************************
C
C*********************************************************************
C     VARIABLE    TYPE          DEFINITION              UNITS       *
C     -------- ---------- -------------------------   ----------    *
C     CP       INTEGER*2  NODE # OF CONTROL POINT      LESS         *
C     ET          "          "  "  "  EXTERNAL TANK      "          *
C     FRAME       "       TIME INTERVAL #               "          *
C     KC          "       FITTING TYPE                  "          *
C     NODE        "       NODE #                        "          *
C     PUMP        "       NODE # OF PUMP                "          *
C     D        REAL*8     INSIDE DIAMETER             M            *
C     DELTAT      "       TIME INTERVAL               S            *
C     DO          "       OUTSIDE DIAMETER            M            *
C     E           "       U+V**2/2+G*Z                M**2/S**2    *
C     EPS         "       PIPE ROUGHNESS              M            *
C     H           "       ENTHALPY                    M**2/S**2    *
C     H0          "       H+V**2/2+G*Z                M**2/S**2    *
C     HC          "       HEAT TRANSFER COEFFICIENT   KG/S**3/K    *
C     F           "       FANNING FRICTION FACTOR     LESS         *
C     K           "       THERMAL CONDUCTIVITY        KG*M/S**3/K  *
```

```
C     KCO          "       TOTAL DISCHARGE COEFFICIENT  LESS          *
C     KELVIN       "       BULK TEMPERATURE             K             *
C     KELWAL       "       WALL  TEMPERATURE            K             *
C     M            "       MASS OF NODE                 M             *
C     MDOT         "       MASS FLOW                    KG/S          *
C     MU           "       VISCOSITY                    KG/M/S        *
C     P            "       PRESSURE                     KG/M/S**2     *
C     PF           "       FRICTIONAL PRESSURE DROP     KG/M/S**2     *
C     PET          "       EXTERNAL TANK PRESSURE       KG/M/S**2     *
C     PFT          "       FEED TANK PRESSURE           KG/M/S**2     *
C     Q            "       HEAT FLUX                    KG/S**3       *
C     QEXT         "       EXTERNAL HEAT FLUX           KG/S**3       *
C     QMAX         "       MAXIMUM HEAT FLUX            KG/S**3       *
C     REY          "       REYNOLDS NUMBER              LESS          *
C     RHO          "       DENSITY                      KG/M**3       *
C     U            "       INTERNAL ENERGY              M**2/S**2     *
C     TAMB         "       AMBIENT TEMPERATURE          K             *
C     TIME         "       ELAPSED TIME                 S             *
C     TFINAL       "       TOTAL ELAPSED TIME           S             *
C     V            "       VELOCITY                     M/S           *
C     VET          "       EXTERNAL TANK VOLUME         M**3          *
C     X            "       QUALITY                      LESS          *
C     Z            "       ELEVATION                    M             *
C*************************************************************************
C
C*************************************************************************
C     INPUT REQUIRED:                                                   *
C                                H2LIS.LIS                              *
C     OUTPUT GENERATED:                                                 *
C                                H2DAT.DAT                              *
C     SUBROUTINES CALLED:                                               *
C                                H2INPUT                                *
C                                H2INIT                                 *
C                                h2MOMNRG                               *
C                                H2OUTPUT                               *
C*************************************************************************
C
C*************************************************************************
C     INFORMATION CONCERNING THE PIPING CONFIGURATION IS CONTAINED      *
C     IN THE BLOCK DATA STATEMENT "H2BLOCK.BLK".THE INITIAL AND         *
C     BOUNDARY CONDITIONS ARE READ IN ON DEVICE #20 FROM FILE           *
C     "H2LIST.LIS".                                                     *
C*************************************************************************
C
C*************************************************************************
C     H2INPUT IS THE INPUT SUBROUTINE.                                  *
C*************************************************************************
      call h2input
C*************************************************************************
C     H2INIT IS THE INITIALIZATION SUBROUTINE                           *
C*************************************************************************
      call h2init
C*************************************************************************
C     h2MOMNRG IS THE PROCESSING SUBROUTINE                             *
C*************************************************************************
      call h2momnrg
C*************************************************************************
C     H2OUTPUT IS THE OUTPUT SUBROUTINE                                 *
C*************************************************************************
      call h2output
C*************************************************************************
C     H2OUTPUT CREATES A REPORT ON A NEW FILE "H2DAT.DAT" ON DE-        *
```

```
C     VICE #21.                                                         *
C***********************************************************************
C
C***********************************************************************
C     THEN                                                              *
C***********************************************************************
      stop
C***********************************************************************
C     AND                                                               *
C***********************************************************************
      end
      subroutine h2input
C***********************************************************************
C     H2INPUT READS "H2LIS.LIS". THE FIRST RECORD CONSISTS OF THE       *
C     MASS IN THE FEED TANK, THE AMBIENT TEMPERATURE,THE INITIAL        *
C     PRESSURE IN THE FEED TANK, AND THE INITIAL PRESSURE IN THE        *
C     EXTERNAL TANK. THE SECOND RECORD CONTAINS THE TIME INTERVAL       *
C     AND THE THE TOTAL TIME OF THE RUN. ALL OF THE                     *
C     REMAINING RECORDS CONSIST OF THE PRESSURE AT THE FEED TANK        *
C     THE MASS FLOW AT THE CONTROL POINT, AND THE PRESSURE AT THE       *
C     EXTERNAL TANK AT ALL TIMES. THE THE RECORDS CONSIST OF THREE      *
C     FLOATING POINT NUMBERS EACH SEPERATED BY A COMMA. THE             *
C     INITIAL MASS FLOW WILL BE SET TO 0.0. DURING INITIATION.          *
C***********************************************************************
      integer*2 cp,et,frame,kc(5,0:99),node,pump
      real*8 cpr(0:99,0:99),cvo(0:99,0:99),
     &       d(0:99),deltat,do(0:99),e(0:99,0:99),eps(0:99),
     &       h(0:99,0:99),hc(0:99,0:99),
     &       h0(0:99,0:99),f(0:99,0:99),k(0:99,0:99),
     &       kco(0:99,0:99),kelvin(0:99,0:99),kelwal(0:99,
     &       0:99),l(0:99),m(0:99,0:99),mdot(0:99,0:99),
     &       mu(0:99,0:99),p(0:99,0:99),pf(0:99,0:99),
     &       pet(0:99),pft(0:99),q(0:99,0:99),
     &       qext(0:99,0:99),
     &       qmax(0:99),rey(0:99,0:99),
     &       rho(0:99,0:99),tamb,tfinal,time(0:99),
     &       u(0:99,0:99),v(0:99,0:99),
     &       vet,x(0:99,0:99),z(0:99)
      common
     &       cp,et,frame,kc,node,pump,
     &       cpr,cvo,
     &       d,deltat,do,e,eps,
     &       h,hc,h0,f,k,
     &       kco,kelvin,kelwal,
     &       l,m,mdot,
     &       mu,p,pf,pet,
     &       pft,q,qext,qmax,
     &       rey,rho,tamb,tfinal,
     &       time,u,
     &       v,vet,x,z
C***********************************************************************
C     OPEN "H2LIS.LIS" AND CONNECT TO DEVICE # 20                       *
C***********************************************************************
      open (file='h2lis.lis', unit=20, status='old')
C***********************************************************************
C     READ THE FIRST RECORD (INITIAL CONDITIONS)                        *
C***********************************************************************
      read(20,*)m(1,0),tamb,pft(0),pet(0)
C***********************************************************************
C     READ THE SECOND RECORD (TIME PARAMETERS)                          *
C***********************************************************************
      read(20,*)deltat,tfinal
```

```
C*********************************************************************
C     READ THE REMAINING RECORDS (BOUNDARY CONDITIONS)                *
C*********************************************************************
      read(20,*)(pft(frame),mdot(cp,frame),
     &          pet(frame),
     &          frame=1,tfinal/deltat)
C*********************************************************************
C     DISCONNECT AND CLOSE THE FILE                                   *
C*********************************************************************
      close(unit=20,status='keep')
C*********************************************************************
C     THEN                                                            *
C*********************************************************************
      return
C*********************************************************************
C     AND                                                             *
C*********************************************************************
      end
      subroutine h2init
C*********************************************************************
C     SUBROUTINE "H2INIT.FOR" SETS THE INITIAL CONDITIONS FOR         *
C     H2FILL. THE INITIAL CONDITION IS CONSIDERED TO BE STATIC.       *
C     THE PRESSURE IS DETERMINED HYDROSTATICALLY, TAKING INTO         *
C     CONSIDERATION THAT H0 IS CONSERVED IN THIS INSTANCE, H IS       *
C     DETERMINED, FORCING THE DETERMINATION OF THE OTHER PROP-        *
C     ERTIES AT THE NODE. THE FEED TANK IS CONSIDERED TO BE           *
C     PARTIALLY FILLED WITH SATURATED LIQUID, WHILE THE EXTERNAL      *
C     TANK IS GAS AT AMBIENT TEMPERATURE.                             *
C*********************************************************************
C
C*********************************************************************
C     SUBROUTINES CALLED:                                             *
C                         FEEDTANK                                    *
C                         H2LSAT                                      *
C                         PROPHP                                      *
C                         PROPPTGS                                    *
C*********************************************************************
      integer*2 cp,et,frame,kc(5,0:99),node,pump
      real*8 cpr(0:99,0:99),cvo(0:99,0:99),
     &       d(0:99),deltat,do(0:99),e(0:99,0:99),eps(0:99),
     &       h(0:99,0:99),hc(0:99,0:99),
     &       h0(0:99,0:99),f(0:99,0:99),k(0:99,0:99),
     &       kco(0:99,0:99),kelvin(0:99,0:99),kelwal(0:99,
     &       0:99),l(0:99),m(0:99,0:99),mdot(0:99,0:99),
     &       mu(0:99,0:99),p(0:99,0:99),pf(0:99,0:99),
     &       pet(0:99),pft(0:99),q(0:99,0:99),
     &       qext(0:99,0:99),
     &       qmax(0:99),rey(0:99,0:99),
     &       rho(0:99,0:99),tamb,tfinal,time(0:99),
     &       u(0:99,0:99),v(0:99,0:99),
     &       vet,x(0:99,0:99),z(0:99)
      common
     &       cp,et,frame,kc,node,pump,
     &       cpr,cvo,
     &       d,deltat,do,e,eps,
     &       h,hc,h0,f,k,
     &       kco,kelvin,kelwal,
     &       l,m,mdot,
     &       mu,p,pf,pet,
     &       pft,q,qext,qmax,
     &       rey,rho,tamb,tfinal,
     &       time,u,
```

```
     &       v,vet,x,z
C*********************************************************************
C     SIX VARIABLES ARE INITIALIZED                                  *
C*********************************************************************
      p(0,0)=pft(0)
      p(et,0)=pet(0)
      kelvin(0,0)=20.39d0
      kelvin(et,0)=tamb
      time(0)=0.
      m(0,0)=m(1,0)
      frame=0
      node=0
C*********************************************************************
C     "H2LSAT" RETURNS THE PROPERTIES OF THE SATURATED LIQUID        *
C*********************************************************************
      call h2lsat
C*********************************************************************
C     "FEEDTANK" RETURNS THE ELEVATION OF THE LIQUID LEVEL           *
C*********************************************************************
      call feedtank
C*********************************************************************
C     FOR THE INITIAL CONDITION H0 IS INVARIENT AND IT IS            *
C     CALCULATED FOR THE TOP OF THE LIQUID LEVEL.                    *
C*********************************************************************
      h0(0,0)=h(0,0)+9.81d0*z(0)
C*********************************************************************
C     FROM THE OUTLET OF THE FEED TANK TO THE INLET OF THE           *
C     CONTROL POINT INITIAL PROPERTIES ARE CALCULATED.               *
C*********************************************************************
      do node=1,cp-1
C*********************************************************************
C     THE PRESSURE IS CALCULATED HYDROSTATICALLY.                    *
C*********************************************************************
       p(node,0)=rho(node-1,0)
     &           *9.81d0*(z(node)-z(node-1))
     &           +p(node-1,0)
C*********************************************************************
C     H IS DETERMINED FROM H0.                                       *
C*********************************************************************
       h0(node,0)=h0(0,0)
       h(node,0)=h0(node,0)-9.81d0*z(node)
C*********************************************************************
C     "PROPHP" DETERMINES THE PROPERTIES FROM H AND P                *
C*********************************************************************
       call prophp
C*********************************************************************
C     FIVE MORE VARIABLES ARE INITIALIZED                            *
C*********************************************************************
       mdot(node,0)=0.
       e(node,0)=h0(node,0)-p(node,0)/rho(node,0)
       hc(node,0)=0.
       pf(node,0)=0.
       kelwal(node,0)=kelvin(node,0)
      end do
C*********************************************************************
C     THE REMAINING MASSES OF THE UPSTREAM NODES ARE ESTABLISHED.    *
C*********************************************************************
      do node=2,cp-1
       m(node,0)=rho(node,0)*datan(1.d0)
     &           *d(node)**2*l(node)
      end do
C*********************************************************************
```

```
C     THE INITIAL CONDITION IS NOW ESTABLISHED FROM THE CONTROL   *
C     POINT TO THE EXTERNAL TANK.                                 *
C***************************************************************
      node=et
C***************************************************************
C     "PROPPTGS" RETURNS THE PROPERTIES OF A GAS GIVEN PRESSURE   *
C     TEMPERATURE INPUT.                                          *
C***************************************************************
      kelvin(node,0)=tamb
      call h2lsat
C***************************************************************
C     THE MASS AND H0 FOR THE EXTERNAL TANK ARE ESTABLISHED.      *
C***************************************************************
      m(et,0)=rho(et,0)*vet
C***************************************************************
C     FOR THE INITIAL CONDITION H0 IS INVARIENT AND IT IS         *
C     CALCULATED FOR THE TOP OF THE EXTERNAL TANK.                *
C***************************************************************
      h0(et,0)=h(et,0)+9.81d0*z(et)
C***************************************************************
C     FROM THE INLET OF THE EXTERNAL TANK TO THE OUTLET OF THE    *
C     CONTROL POINT THE INITIAL PROPERTIES ARE CALCULATED.        *
C***************************************************************
      do node=et-1,cp,-1
C***************************************************************
C     THE PRESSURE IS CALCULATED HYDROSTATICALLY.                 *
C***************************************************************
       p(node,0)=rho(node+1,0)
     &           *9.81d0*(z(node+1)-z(node))
     &           +p(node+1,0)
C***************************************************************
C     H IS DETERMINED FROM H0.                                    *
C***************************************************************
       h0(node,0)=h0(et,0)
       h(node,0)=h0(node,0)-9.81d0*z(node)
C***************************************************************
C     "PROPHP" DETERMINES THE PROPERTIES FROM H AND P             *
C***************************************************************
       call prophp
C***************************************************************
C     FIVE MORE VARIABLES ARE INITIALIZED                         *
C***************************************************************
       mdot(node,0)=0.
       e(node,0)=h0(node,0)-p(node,0)/rho(node,0)
       hc(node,0)=0.
       pf(node,0)=0.
       kelwal(node,0)=kelvin(node,0)
      end do
C***************************************************************
C     THE REMAINING MASSES OF THE DOWNSTREAM NODES ARE            *
C     ESTABLISHED.                                                *
C***************************************************************
      do node=cp+1,et-1
       m(node,0)=rho(node,0)*4.d0*datan(1.d0)
     &           *d(node)*l(node)
      end do
      do frame=1,tfinal/deltat
       p(0,frame)=pft(frame)
       node=0
       call h2lsat
       p(et,frame)=pet(frame)
       node=et
```

```
       call h2lsat
      end do
      do node=0,et
       mdot(node,0)=mdot(node,1)
      end do
C*********************************************************************
C     THEN                                                           *
C*********************************************************************
      return
C*********************************************************************
C     AND                                                            *
C*********************************************************************
      end
      subroutine h2momnrg
C*********************************************************************
C     "H2MOMNRG" BASICALLY SOLVES THE TRANSIENT MOMENTUM AND         *
C     ENERGY EQUATIONS IN ORDER TO UPDATE M, MDOT, RHO, V, E, U,     *
C     AND THEN THE PHYSICAL PROPERTIES. AFTER THE ABOVE HAS          *
C     BEEN ESTABLISHED, THE TRANSPORT COEFFICIENTS HC, F, AND KCO    *
C     ARE DETERMINED.                                                *
C*********************************************************************
C
C*********************************************************************
C     SUBROUTINES CALLED:                                            *
C                           EXTANK                                   *
C                           FEEDTANK                                 *
C                           PROPURHO                                 *
C                           TRANS                                    *
C*********************************************************************
      integer*2 cp,et,frame,kc(5,0:99),node,pump
      real*8 cpr(0:99,0:99),cvo(0:99,0:99),
     &       d(0:99),deltat,do(0:99),e(0:99,0:99),eps(0:99),
     &       h(0:99,0:99),hc(0:99,0:99),
     &       h0(0:99,0:99),f(0:99,0:99),k(0:99,0:99),
     &       kco(0:99,0:99),kelvin(0:99,0:99),kelwal(0:99,
     &       0:99),l(0:99),m(0:99,0:99),mdot(0:99,0:99),
     &       mu(0:99,0:99),p(0:99,0:99),pf(0:99,0:99),
     &       pet(0:99),pft(0:99),q(0:99,0:99),
     &       qext(0:99,0:99),
     &       qmax(0:99),rey(0:99,0:99),
     &       rho(0:99,0:99),tamb,tfinal,time(0:99),
     &       u(0:99,0:99),v(0:99,0:99),
     &       vet,x(0:99,0:99),z(0:99)
      common
     &       cp,et,frame,kc,node,pump,
     &       cpr,cvo,
     &       d,deltat,do,e,eps,
     &       h,hc,h0,f,k,
     &       kco,kelvin,kelwal,
     &       l,m,mdot,
     &       mu,p,pf,pet,
     &       pft,q,qext,qmax,
     &       rey,rho,tamb,tfinal,
     &       time,u,
     &       v,vet,x,z
C*********************************************************************
C     START THE CLOCK.                                               *
C*********************************************************************
      do frame=1,tfinal/deltat
C*********************************************************************
C     RECORD THE TIME.                                               *
C*********************************************************************
```

```
       time(frame)=frame*deltat
C*********************************************************************
C     NO ACCUMULATION OCCURS WITHIN THE CONTROL POINT NODE.          *
C*********************************************************************
       mdot(cp-1,frame)=mdot(cp,frame)
C*********************************************************************
C     THE FEED TANK PRESSURE IS ZEROED IN ORDER TO ENTER THE DO      *
C     LOOP. IT IS LATER UPDATED TO WITHIN 1% OF PFT IN AN            *
C     ITERATIVE PROCESS.                                             *
C*********************************************************************
       p(cp-1,frame)=p(0,frame)*9.d-1
C        do while
C     &           ((abs(p(cp-1,frame)/p(cp-1,frame-1))
C     &            .lt. .99d0) .or.
C     &           (abs(p(cp-1,frame)/p(cp-1,frame-1))
C     &            .gt. 1.01d0))
C*********************************************************************
C     FOR THE INTERMEDIATE PIPING UPSTREAM OF THE CONTROL POINT.     *
C*********************************************************************
         do node=cp-2,1,-1
C*********************************************************************
C     CALCULATE THE CROSS SECTIONAL AREA.                            *
C*********************************************************************
          s=datan(1.d0)*d(node+1)**2
C*********************************************************************
C     UPDATE MDOT                                                    *
C*********************************************************************
          mdot(node,frame)=mdot(node,frame-1)-s*deltat/l(node+1)*
     &                     (p(node+1,frame-1)-p(node,frame-1)+
     &                     v(node+1,frame-1)**2*
     &                     rho(node+1,frame-1)-
     &                     v(node,frame-1)**2*rho(node,
     &                     frame-1)+rho(node+1,frame-1)
     &                     *9.81d0*(z(node+1)-z(node))+pf(
     &                     node+1,frame-1))
C*********************************************************************
C     UPDATE M, RHO, AND V                                           *
C*********************************************************************
         end do
         do node=cp-1,2,-1
          s=datan(1.d0)*d(node)**2
          m(node,frame)=(mdot(node-1,frame-1)-mdot(node,frame-1))
     &                  *deltat+m(node,frame-1)
          rho(node,frame)=m(node,frame)/s/l(node)
          v(node,frame)=mdot(node,frame)/s/rho(node,frame)
         end do
C*********************************************************************
C     UPDATE MDOT FOR THE FEED TANK                                  *
C*********************************************************************
         mdot(0,frame)=mdot(1,frame)
         m(0,frame)=m(0,frame-1)-mdot(0,frame-1)*deltat
C*********************************************************************
C     UPDATE M, RHO, AND V FOR THE TERMINAL UPSTREAM NODES           *
C*********************************************************************
         m(1,frame)=m(0,frame)
C.... THIS IS A COURSE APPROXIMATION FOR AN IDEAL GAS ONLY
C.... A BARAMETRIC SUBROUTINE WOULD BE DEVELOPED FOR THE
C.... GENERAL CASE
         rho(1,frame)=rho(2,frame)
         v(1,frame)=v(2,frame)
C*********************************************************************
C       UPDATE E,U,H,H0, AND PF FOR ALL THE UPSTREAM PIPE NODES       *
```

```
C*********************************************************************
        do node=cp-1,1,-1
         if (node .ne. 1)then
          e(node,frame)=((mdot(node-1,frame-1)*h0(node-1,
     &                  frame-1)-mdot(node,frame-1)*h0
     &                  (node,frame-1))+hc(node,frame-1)
     &                  *(4.d0*datan(1.d0)*d(node)*l(node))*
     &                  (kelwal(node,frame-1)-kelvin(node,
     &                  frame-1))*deltat+m(node,frame-1)*
     &                  e(node,frame-1))/m(node,frame)
         else
          e(node,frame)=-mdot(node,frame-1)*h0(node,frame-1)
     &                 +e(node,frame-1)*m(node,frame-1)/
     &                  m(node,frame)
         endif
         u(node,frame)=e(node,frame)
     &                -v(node,frame)**2/2-9.81d0*z(node)
C*********************************************************************
C        "PROPURHO" RETURNS THE PHYSICAL PROPERTIES FROM AN         *
C        INPUT OF U AND RHO.                                        *
C*********************************************************************
          call propurho
          h0(node,frame)=h(node,frame)+v(node,frame)**2/2
     &                  +9.81d0*z(node)
C*********************************************************************
C         "TRANS" RETURNS THE TRANSPORT PROPERTIES F,KCO,AND HC      *
C         FROM THE PHYSICAL PROPERTIES AND CERTAIN FLOW              *
C         PARAMETERS THAT HAVE ALREADY BEEN ESTABLISHED. TW IS       *
C         ALSO RETURNED.                                             *
C*********************************************************************
           if(node .ne. 1)then
            call trans
            pf(node,frame)=(4.d0*f(node,frame)+kco(node,frame))*
     &                     rho(node,frame)*
     &                     v(node,frame)**2/2*
     &                     l(node)/d(node)
         endif
        end do
C*********************************************************************
C     "FEEDTANK" RETURNS THE ELEVATION OF THE LIQUID LEVEL IN THE    *
C     FEED TANK FROM THE FEED TANK MASS.                             *
C*********************************************************************
        call feedtank
C*********************************************************************
C     UPDATE E,H,AND H0 FOR THE TOP OF THE FEED TANK                 *
C*********************************************************************
        e(0,frame)=(-mdot(1,frame-1)*h0(1,frame-1)*
     &              deltat+m(1,frame-1)*e(0,frame-1))
     &              /m(1,frame)
        u(0,frame)=e(0,frame)-
     &             9.81d0*z(0)
        node=0
        call propurho
        h0(0,frame)=h(0,frame)+
     &              9.81d0*z(0)
C*********************************************************************
C     THIS IS THE THROTTLING CONDITION.                             *
C*********************************************************************
       h0(cp,frame)=h0(cp-1,frame)
C*********************************************************************
C     INITALIZE P INORDER TO ENTER LOOP. AN ITERATIVE PROCESS WILL*
C     THEN BRING UPDATE IT TO WITHIN 1% OF PET.                     *
```

```
C*********************************************************************
        p(et,frame)=0.
        do while
     &          (abs(pet(frame)-p(et,frame))
     &          .gt. 0.01d0*pet(frame))
C*********************************************************************
C     EVALUATE MDOT, RHO, V, M, E, H0, U, H, PLUS THE PHYSICAL     *
C     AND TRANSP0RT PROPERTIES FOR THE INTERMEDIATE NODES. NOTE    *
C     THAT TW IS ALSO RETURNED.                                    *
C*********************************************************************
         do node=et-1,cp+1,-1
          s=datan(1.d0)*d(node)**2
          mdot(node,frame)=mdot(node,frame-1)-s*deltat/l(node)*
     &                    (p(node,frame-1)-p(node-1,frame-1)+
     &                    v(node,frame-1)**2*
     &                    rho(node,frame-1)-
     &                    v(node-1,frame-1)**2*rho(node-1,
     &                    frame-1)+rho(node-1,frame-1)
     &                   *9.81d0*(z(node)-z(node-1))+pf(
     &                    node,frame-1))
          m(node,frame)=(mdot(node-1,frame-1)-mdot(node,frame-1))*deltat
     &                  +m(node,frame-1)
          rho(node,frame)=m(node,frame)/s/l(node)
          v(node,frame)=mdot(node,frame)/s/rho(node,frame)
          e(node,frame)=((mdot(node-1,frame-1)*h0(node-1,
     &                   frame-1)-mdot(node,frame-1)*h0
     &                   (node,frame-1))+hc(node,frame-1)
     &                   *(4.d0*datan(1.d0)*d(node)*l(node))*
     &                   (kelwal(node,frame-1)-kelvin(node,
     &                   frame-1))*deltat+m(node,frame-1)*
     &                   e(node,frame-1))/m(node,frame)
          u(node,frame)=e(node,frame)
     &                 -v(node,frame)**2/2-9.81d0*z(node)
          call propurho
          h(node,frame)=u(node,frame)+p(node,frame)
     &                  /rho(node,frame)
          h0(node,frame)=h(node,frame)+v(node,frame)**2/2
     &                   +9.81d0*z(node)
          call trans
          pf(node,frame)=(4.d0*f(node,frame)+kco(node,frame))*
     &                   rho(node,frame)*
     &                   v(node,frame)**2/2*
     &                   l(node)/d(node)
         end do
C*********************************************************************
C     UPDATE THE DOWNSTREAM PROPERTIES FOR THE CONTROL ELEMENT     *
C     AND THE TOP OF THE EXTERNAL TANK.                            *
C*********************************************************************
        rho(cp,frame)=rho(cp+1,frame)
        v(cp,frame)=mdot(cp,frame)/rho(cp,frame)
     &              /datan(1.d0)/d(node)**2
        h(cp,frame)=h0(cp,frame)
     &             -v(cp,frame)**2/2
     &             -9.81d0*z(frame)
        node=cp
C*********************************************************************
C       "PROPHRHO" RETURNS THE PHYSICAL PROPERTIES FROM AN          *
C       INPUT OF H AND RHO.                                         *
C*********************************************************************
        call prophrho
        e(cp,frame)=h0(cp,frame)-
     &              p(cp,frame)/
```

```
     &              rho(cp,frame)
        u(cp,frame)=h(cp,frame)-
     &              p(cp,frame)/
     &              rho(cp,frame)
C*********************************************************************
C     "EXTANK" RETURNS THE ELEVATION OF THE FLUID LEVEL FROM THE    *
C     MASS AND QUALITY IN THE EXTERNAL TANK.                        *
C*********************************************************************
        call extank
        e(et,frame)=mdot(et-1,frame-1)*h0(et,frame-1)
     &              *deltat+m(et,frame-1)*e(et,frame-1)
     &              /m(et,frame)
        u(et,frame)=e(et,frame)-
     &              9.81d0*z(et)
        node=et
        call propurho
        h0(et,frame)=e(et,frame)+
     &               p(et,frame)/
     &               rho(et,frame)
        h(et,frame)=u(et,frame)+
     &              p(et,frame)/
     &              rho(et,frame)
       end do
C*********************************************************************
C     RESET THE CLOCK                                               *
C*********************************************************************
      end do
C*********************************************************************
C     THEN                                                          *
C*********************************************************************
      return
C*********************************************************************
C     AND                                                           *
C*********************************************************************
      end
      subroutine h2output
C*********************************************************************
C     "H2OUTPUT" CREATES THE FILE "H2DAT.DAT" ON DEVICE # 21. IT    *
C     THEN PRINTS A TABULAR OUTPUT OF THE TIME AT VARIOUS NODES,    *
C     FOLLOWED BY A REARRANGEMENT OF THE DATA FOR THE NODES AT      *
C     VARIOUS TIMES.                                                *
C*********************************************************************
      integer*2 cp,et,frame,kc(5,0:99),node,pump
      real*8 cpr(0:99,0:99),cvo(0:99,0:99),
     &       d(0:99),deltat,do(0:99),e(0:99,0:99),eps(0:99),
     &       h(0:99,0:99),hc(0:99,0:99),
     &       h0(0:99,0:99),f(0:99,0:99),k(0:99,0:99),
     &       kco(0:99,0:99),kelvin(0:99,0:99),kelwal(0:99,
     &       0:99),l(0:99),m(0:99,0:99),mdot(0:99,0:99),
     &       mu(0:99,0:99),p(0:99,0:99),pf(0:99,0:99),
     &       pet(0:99),pft(0:99),q(0:99,0:99),
     &       qext(0:99,0:99),
     &       qmax(0:99),rey(0:99,0:99),
     &       rho(0:99,0:99),tamb,tfinal,time(0:99),
     &       u(0:99,0:99),v(0:99,0:99),
     &       vet,x(0:99,0:99),z(0:99)
      common
     &       cp,et,frame,kc,node,pump,
     &       cpr,cvo,
     &       d,deltat,do,e,eps,
     &       h,hc,h0,f,k,
     &       kco,kelvin,kelwal,
```

```
     &        l,m,mdot,
     &        mu,p,pf,pet,
     &        pft,q,qext,qmax,
     &        rey,rho,tamb,tfinal,
     &        time,u,
     &        v,vet,x,z
C*********************************************************************
C     CREATE OUTPUT FILE "H2DAT.DAT" AND CONNECT TO DEVICE # 21.    *
C*********************************************************************
      open (file='h2dat.dat', unit=21, status='new')
C*********************************************************************
C     START THE CLOCK.                                              *
C*********************************************************************
      do frame=0,tfinal/deltat
C*********************************************************************
C     PRINT THE HEADING.                                            *
C*********************************************************************
       write(21,100) cp,et,time(frame)
 100   format(1x,'FEED TANK AT NODE =  0',/,
     &        1x,'CONTROL POINT AT NODE = ',i2,/,
     &        1x,'EXTERNAL TANK AT NODE = ',i2,/,
     &        1x,'ELAPSED TIME = ',d13.6,/)
       write(21,101)
 101   format(1x,5x,'NODE',5x,
     &           4x,'QUALITY'3x,
     &           3x,'PRESSURE',3x,
     &           2x,'TEMPERATURE',1x,
     &           2x,'WALL TEMP.',2x,
     &           4x,'DENSITY',3x,
     &           3x,'VELOCITY',3x,
     &           5x,'MASS',5x,
     &           3x,'MASS FLOW',/)
C*********************************************************************
C      SCAN THE NODES.                                              *
C*********************************************************************
       do node=0,et
        write(21,102)node,x(node,frame),p(node,frame),
     &               kelvin(node,frame),kelwal(node,frame),
     &               rho(node,frame),v(node,frame),
     &               m(node,frame),mdot(node,frame)
 102   format(1x,6x,i2,6x,8(1x,d13.6))
C*********************************************************************
C     NEXT NODE                                                     *
C*********************************************************************
       end do
C*********************************************************************
C     RESET THE CLOCK                                               *
C*********************************************************************
      end do
C*********************************************************************
C     TOP OF THE PAGE FOR THE TIME SCAN OF THE NODES.               *
C*********************************************************************
      write(21,103)
 103   format('1')
C*********************************************************************
C     SCAN THE NODES.                                               *
C*********************************************************************
      do node=0,et
C*********************************************************************
C     PRINT A HEADING.                                              *
C*********************************************************************
       write(21,104) cp,et,node
```

```
 104  format(1x,'FEED TANK AT NODE =  0',/,
     &       1x,'CONTROL POINT AT NODE = ',i2,/,
     &       1x,'EXTERNAL TANK AT NODE = ',i2,/,
     &       1x,'NODE #  = ',i2,/)
       write(21,105)
 105  format(1x,5x,'TIME',5x,
     &          4x,'QUALITY'3x,
     &          3x,'PRESSURE',3x,
     &          2x,'TEMPERATURE',1x,
     &          2x,'WALL TEMP.',2x,
     &          4x,'DENSITY',3x,
     &          3x,'VELOCITY',3x,
     &          5x,'MASS',5x,
     &          3x,'MASS FLOW',/)
C*********************************************************************
C      START THE CLOCK                                               *
C*********************************************************************
       do frame=0,tfinal/deltat
C*********************************************************************
C      PRINT THE DATA.                                               *
C*********************************************************************
        write(21,106)time(frame),x(node,frame),p(node,frame),
     &               kelvin(node,frame),kelwal(node,frame),
     &               rho(node,frame),v(node,frame),
     &               m(node,frame),mdot(node,frame)
 106  format(1x,9(1x,d13.6))
C*********************************************************************
C     RESET THE CLOCK.                                               *
C*********************************************************************
       end do
C*********************************************************************
C     NEXT NODE.                                                     *
C*********************************************************************
       end do
C*********************************************************************
C     CLOSE AND DISCONNECT FILE "H2DAT.DAT".                         *
C*********************************************************************
       close(unit=21)
C*********************************************************************
C     THEN                                                           *
C*********************************************************************
       return
C*********************************************************************
C     AND                                                            *
C*********************************************************************
       end
       subroutine h2lsat
       integer*2 cp,et,frame,kc(5,0:99),node,pump
       real*8 cpr(0:99,0:99),cvo(0:99,0:99),
     &        d(0:99),deltat,do(0:99),e(0:99,0:99),eps(0:99),
     &        h(0:99,0:99),hc(0:99,0:99),
     &        h0(0:99,0:99),f(0:99,0:99),k(0:99,0:99),
     &        kco(0:99,0:99),kelvin(0:99,0:99),kelwal(0:99,
     &        0:99),l(0:99),m(0:99,0:99),mdot(0:99,0:99),
     &        mu(0:99,0:99),p(0:99,0:99),pf(0:99,0:99),
     &        pet(0:99),pft(0:99),q(0:99,0:99),
     &        qext(0:99,0:99),
     &        qmax(0:99),rey(0:99,0:99),
     &        rho(0:99,0:99),tamb,tfinal,time(0:99),
     &        u(0:99,0:99),v(0:99,0:99),
     &        vet,x(0:99,0:99),z(0:99)
       common
```

```
&       cp,et,frame,kc,node,pump,
&       cpr,cvo,
&       d,deltat,do,e,eps,
&       h,hc,h0,f,k,
&       kco,kelvin,kelwal,
&       l,m,mdot,
&       mu,p,pf,pet,
&       pft,q,qext,qmax,
&       rey,rho,tamb,tfinal,
&       time,u,
&       v,vet,x,z
 call propptgs
 return
 end
 subroutine prophrho
 integer*2 cp,et,frame,kc(5,0:99),node,pump
 real*8 cpr(0:99,0:99),cvo(0:99,0:99),
&       d(0:99),deltat,do(0:99),e(0:99,0:99),eps(0:99),
&       h(0:99,0:99),hc(0:99,0:99),
&       h0(0:99,0:99),f(0:99,0:99),k(0:99,0:99),
&       kco(0:99,0:99),kelvin(0:99,0:99),kelwal(0:99,
&       0:99),l(0:99),m(0:99,0:99),mdot(0:99,0:99),
&       mu(0:99,0:99),p(0:99,0:99),pf(0:99,0:99),
&       pet(0:99),pft(0:99),q(0:99,0:99),
&       qext(0:99,0:99),
&       qmax(0:99),rey(0:99,0:99),
&       rho(0:99,0:99),tamb,tfinal,time(0:99),
&       u(0:99,0:99),v(0:99,0:99),
&       vet,x(0:99,0:99),z(0:99)
 common
&       cp,et,frame,kc,node,pump,
&       cpr,cvo,
&       d;deltat,do,e,eps,
&       h,hc,h0,f,k,
&       kco,kelvin,kelwal,
&       l,m,mdot,
&       mu,p,pf,pet,
&       pft,q,qext,qmax,
&       rey,rho,tamb,tfinal,
&       time,u,
&       v,vet,x,z
 u(node,frame)=h(node,frame)/1.4d0
 call propurho
 return
 end
 subroutine propptgs
 integer*2 cp,et,frame,kc(5,0:99),node,pump
 real*8 cpr(0:99,0:99),cvo(0:99,0:99),
&       d(0:99),deltat,do(0:99),e(0:99,0:99),eps(0:99),
&       h(0:99,0:99),hc(0:99,0:99),
&       h0(0:99,0:99),f(0:99,0:99),k(0:99,0:99),
&       kco(0:99,0:99),kelvin(0:99,0:99),kelwal(0:99,
&       0:99),l(0:99),m(0:99,0:99),mdot(0:99,0:99),
&       mu(0:99,0:99),p(0:99,0:99),pf(0:99,0:99),
&       pet(0:99),pft(0:99),q(0:99,0:99),
&       qext(0:99,0:99),
&       qmax(0:99),rey(0:99,0:99),
&       rho(0:99,0:99),tamb,tfinal,time(0:99),
&       u(0:99,0:99),v(0:99,0:99),
&       vet,x(0:99,0:99),z(0:99)
 common
&       cp,et,frame,kc,node,pump,
```

```
&       cpr,cvo,
&       d,deltat,do,e,eps,
&       h,hc,h0,f,k,
&       kco,kelvin,kelwal,
&       l,m,mdot,
&       mu,p,pf,pet,
&       pft,q,qext,qmax,
&       rey,rho,tamb,tfinal,
&       time,u,
&       v,vet,x,z
 if((frame .eq. 0) .or.
&   (node .eq. 0) .or.
&   (node .eq. et)) then
  if(node .lt. cp)kelvin(node,frame)=20.39d0
  if(node .ge. cp)kelvin(node,frame)=tamb
 endif
 rho(node,frame)=p(node,frame)/
&                4157.d0/kelvin(node,frame)
 u(node,frame)=2.5d0*4157.d0*kelvin(node,frame)
 call propurho
 return
 end
 subroutine propurho
 integer*2 cp,et,frame,kc(5,0:99),node,pump
 real*8 cpr(0:99,0:99),cvo(0:99,0:99),
&       d(0:99),deltat,do(0:99),e(0:99,0:99),eps(0:99),
&       h(0:99,0:99),hc(0:99,0:99),
&       h0(0:99,0:99),f(0:99,0:99),k(0:99,0:99),
&       kco(0:99,0:99),kelvin(0:99,0:99),kelwal(0:99,
&       0:99),l(0:99),m(0:99,0:99),mdot(0:99,0:99),
&       mu(0:99,0:99),p(0:99,0:99),pf(0:99,0:99),
&       pet(0:99),pft(0:99),q(0:99,0:99),
&       qext(0:99,0:99),
&       qmax(0:99),rey(0:99,0:99),
&       rho(0:99,0:99),tamb,tfinal,time(0:99),
&       u(0:99,0:99),v(0:99,0:99),
&       vet,x(0:99,0:99),z(0:99)
 common
&       cp,et,frame,kc,node,pump,
&       cpr,cvo,
&       d,deltat,do,e,eps,
&       h,hc,h0,f,k,
&       kco,kelvin,kelwal,
&       l,m,mdot,
&       mu,p,pf,pet,
&       pft,q,qext,qmax,
&       rey,rho,tamb,tfinal,
&       time,u,
&       v,vet,x,z
 kelvin(node,frame)=u(node,frame)/2.5d0/4157.d0
 p(node,frame)=rho(node,frame)*4157.d0*
&              kelvin(node,frame)
 h(node,frame)=1.4d0*u(node,frame)
 mu(node,frame)=2.d-6
 k(node,frame)=0.05d0
 return
 end
 block data h2block
 integer*2 cp,et,frame,kc(5,0:99),node,pump
 real*8 cpr(0:99,0:99),cvo(0:99,0:99),
&       d(0:99),deltat,do(0:99),e(0:99,0:99),eps(0:99),
&       h(0:99,0:99),hc(0:99,0:99),
```

```
&       h0(0:99,0:99),f(0:99,0:99),k(0:99,0:99),
&       kco(0:99,0:99),kelvin(0:99,0:99),kelwal(0:99,
&       0:99),l(0:99),m(0:99,0:99),mdot(0:99,0:99),
&       mu(0:99,0:99),p(0:99,0:99),pf(0:99,0:99),
&       pet(0:99),pft(0:99),q(0:99,0:99),
&       qext(0:99,0:99),
&       qmax(0:99),rey(0:99,0:99),
&       rho(0:99,0:99),tamb,tfinal,time(0:99),
&       u(0:99,0:99),v(0:99,0:99),
&       vet,x(0:99,0:99),z(0:99)
 common
&       cp,et,frame,kc,node,pump,
&       cpr,cvo,
&       d,deltat,do,e,eps,
&       h,hc,h0,f,k,
&       kco,kelvin,kelwal,
&       l,m,mdot,
&       mu,p,pf,pet,
&       pft,q,qext,qmax,
&       rey,rho,tamb,tfinal,
&       time,u,
&       v,vet,x,z
 data cp,et/4,35/
 data d/
&       4*0.2d0,32*0.25d0,64*0.d0/
 data do/
&       4*0.22d0,32*0.27d0,64*0.d0/
 data eps/
&        100*0.0d0/
 data l/
&      100*10.d0/
 data qmax/
&        '100*1500.d0/
 data z/
&      10.d0,34*0.d0,30.d0,64*0.d0/
 end


 subroutine trans
 integer*2 cp,et,frame,kc(5,0:99),node,pump
 real*8 cpr(0:99,0:99),cvo(0:99,0:99),
&       d(0:99),deltat,do(0:99),e(0:99,0:99),eps(0:99),
&       h(0:99,0:99),hc(0:99,0:99),
&       h0(0:99,0:99),f(0:99,0:99),k(0:99,0:99),
&       kco(0:99,0:99),kelvin(0:99,0:99),kelwal(0:99,
&       0:99),l(0:99),m(0:99,0:99),mdot(0:99,0:99),
&       mu(0:99,0:99),p(0:99,0:99),pf(0:99,0:99),
&       pet(0:99),pft(0:99),q(0:99,0:99),
&       qext(0:99,0:99),
&       qmax(0:99),rey(0:99,0:99),
&       rho(0:99,0:99),tamb,tfinal,time(0:99),
&       u(0:99,0:99),v(0:99,0:99),
&       vet,x(0:99,0:99),z(0:99)
 common
&       cp,et,frame,kc,node,pump,
&       cpr,cvo,
&       d,deltat,do,e,eps,
&       h,hc,h0,f,k,
&       kco,kelvin,kelwal,
&       l,m,mdot,
&       mu,p,pf,pet,
&       pft,q,qext,qmax,
```

```
&       rey,rho,tamb,tfinal,
&       time,u,
&       v,vet,x,z
 f(node,frame)=0.002d0
 hc(node,0)=0.d0
 hc(node,frame)=1360.d0
 if(node .lt. cp)kelwal(node,0)=20.39d0
 if(node .ge. cp)kelwal(node,0)=tamb
 qext(node,frame)=qmax(node)
&                 *(tamb-kelwal(node,frame-1))
&                 /(tamb-20.39d0)
 kelwal(node,frame)=kelwal(node,frame-1)
&                  +qext(node,frame-1)
&                  /419/7800/datan(1.d0)
&                  /(do(node)**2
&                  -d(node)**2)/l(node)
&                  *deltat-hc(node,frame-1)
&                  *4.d0*datan(1.d0)*d(node)*l(node)
&                  *(kelwal(node,frame-1)
&                  -kelvin(node,frame-1))
&                  *deltat
&                  /419.d0/7800.d0/datan(1.d0)
&                  /(do(node)**2
&                  -d(node)**2)/l(node)
 return
 end
 subroutine prophp
 integer*2 cp,et,frame,kc(5,0:99),node,pump
 real*8 cpr(0:99,0:99),cvo(0:99,0:99),
&       d(0:99),deltat,do(0:99),e(0:99,0:99),eps(0:99),
&       h(0:99,0:99),hc(0:99,0:99),
&       h0(0:99,0:99),f(0:99,0:99),k(0:99,0:99),
&       kco(0:99,0:99),kelvin(0:99,0:99),kelwal(0:99,
&       0:99),l(0:99),m(0:99,0:99),mdot(0:99,0:99),
&       mu(0:99,0:99),p(0:99,0:99),pf(0:99,0:99),
&       pet(0:99),pft(0:99),q(0:99,0:99),
&       qext(0:99,0:99),
&       qmax(0:99),rey(0:99,0:99),
&       rho(0:99,0:99),tamb,tfinal,time(0:99),
&       u(0:99,0:99),v(0:99,0:99),
&       vet,x(0:99,0:99),z(0:99)
 common
&       cp,et,frame,kc,node,pump,
&       cpr,cvo,
&       d,deltat,do,e,eps,
&       h,hc,h0,f,k,
&       kco,kelvin,kelwal,
&       l,m,mdot,
&       mu,p,pf,pet,
&       pft,q,qext,qmax,
&       rey,rho,tamb,tfinal,
&       time,u,
&       v,vet,x,z
 u(node,frame)=h(node,frame)/1.4d0
 rho(node,frame)=p(node,frame)/(h(node,frame)
&                -u(node,frame))
 call propurho
 return
 end
 subroutine feedtank
 integer*2 cp,et,frame,kc(5,0:99),node,pump
 real*8 cpr(0:99,0:99),cvo(0:99,0:99),
```

```
&       d(0:99),deltat,do(0:99),e(0:99,0:99),eps(0:99),
&       h(0:99,0:99),hc(0:99,0:99),
&       h0(0:99,0:99),f(0:99,0:99),k(0:99,0:99),
&       kco(0:99,0:99),kelvin(0:99,0:99),kelwal(0:99,
&       0:99),l(0:99),m(0:99,0:99),mdot(0:99,0:99),
&       mu(0:99,0:99),p(0:99,0:99),pf(0:99,0:99),
&       pet(0:99),pft(0:99),q(0:99,0:99),
&       qext(0:99,0:99),
&       qmax(0:99),rey(0:99,0:99),
&       rho(0:99,0:99),tamb,tfinal,time(0:99),
&       u(0:99,0:99),v(0:99,0:99),
&       vet,x(0:99,0:99),z(0:99)
 common
&       cp,et,frame,kc,node,pump,
&       cpr,cvo,
&       d,deltat,do,e,eps,
&       h,hc,h0,f,k,
&       kco,kelvin,kelwal,
&       l,m,mdot,
&       mu,p,pf,pet,
&       pft,q,qext,qmax,
&       rey,rho,tamb,tfinal,
&       time,u,
&       v,vet,x,z
        z(0)=z(0)
        return
        end
 subroutine extank
 integer*2 cp,et,frame,kc(5,0:99),node,pump
 real*8 cpr(0:99,0:99),cvo(0:99,0:99),
&       d(0:99),deltat,do(0:99),e(0:99,0:99),eps(0:99),
&       h(0:99,0:99),hc(0:99,0:99),
&       h0(0:99,0:99),f(0:99,0:99),k(0:99,0:99),
&       kco(0:99,0:99),kelvin(0:99,0:99),kelwal(0:99,
&       0:99),l(0:99),m(0:99,0:99),mdot(0:99,0:99),
&       mu(0:99,0:99),p(0:99,0:99),pf(0:99,0:99),
&       pet(0:99),pft(0:99),q(0:99,0:99),
&       qext(0:99,0:99),
&       qmax(0:99),rey(0:99,0:99),
&       rho(0:99,0:99),tamb,tfinal,time(0:99),
&       u(0:99,0:99),v(0:99,0:99),
&       vet,x(0:99,0:99),z(0:99)
 common
&       cp,et,frame,kc,node,pump,
&       cpr,cvo,
&       d,deltat,do,e,eps,
&       h,hc,h0,f,k,
&       kco,kelvin,kelwal,
&       l,m,mdot,
&       mu,p,pf,pet,
&       pft,q,qext,qmax,
&       rey,rho,tamb,tfinal,
&       time,u,
&       v,vet,x,z
        z(et)=z(et)
        return
        end
```

REFERENCES

1.Lin, T.Y.

Technical summary and user's guide for the transient cryogenic transfer program.

KSC-DD-988 (1984)

2.Coulter Jr., B.M.

Superflow.

Fluid Research Publishing, Melbourne,FL (1987)

3.Cosmic software catalogue

NASA-CR-187366 (1991)

4.Perry, R.H.

Perry's chemical engineer's handbook

6th edition

McGraw-Hill (1984)

5.Scott, R.B.

Cryogenic engineering

1963 edition

Met-Chem Research Inc., Boulder,CO (1988)

6.McCarty, R.D.
Selected properties of hydrogen
U.S. G.P.O. (1981)

7.Bejan, A.
Advanced engineering thermodynamics
Wiley (1988)

8.Bejan, A.
Convective heat transfer
Wiley (1984)