AUTOMATIC MATHEMATICAL MODELING FOR
REAL TIME SIMULATION PROGRAM
(AI APPLICATION)

By Caroline Wang and Steve Purinton

Software and Data Management Division
Information and Electronic Systems Laboratory
Science and Engineering Directorate

February 1989
**Title and Subtitle**
Automatic Mathematical Modeling for Real Time Simulation Program

**Authors**
Caroline Wang and Steve Purinton

**Performing Organization**
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

**Sponsoring Agency**
National Aeronautics and Space Administration
Washington, D.C. 20546

**Abstract**
This report describes a methodology for automatic mathematical modeling and generating simulation models. The major objective was to create a user-friendly environment for engineers to design, maintain, and verify their models; to automatically convert the mathematical models into conventional code for computation; and finally, to document the model automatically.

**Keywords**
Automatic Modeling for Space Application Mathematical

**Distribution Statement**
Unclassified – Unlimited
### TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>II. INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>III. PURPOSE</td>
<td>2</td>
</tr>
<tr>
<td>IV. INTELLIGENT INTERFACES</td>
<td>2</td>
</tr>
<tr>
<td>V. AUTOMATIC MODELING</td>
<td>3</td>
</tr>
<tr>
<td>VI. SYSTEM OVERVIEW</td>
<td>3</td>
</tr>
<tr>
<td>- Flow Diagram</td>
<td>3</td>
</tr>
<tr>
<td>- Basic Software and Hardware Requirement</td>
<td></td>
</tr>
<tr>
<td>VII. MAINTAINING AND VERIFYING</td>
<td>9</td>
</tr>
<tr>
<td>VIII. GENERAL APPLICATION</td>
<td>9</td>
</tr>
<tr>
<td>IX. EXAMPLE APPLICATION</td>
<td>10</td>
</tr>
<tr>
<td>X. FUTURE PLANS</td>
<td>11</td>
</tr>
<tr>
<td>APPENDIX</td>
<td>13</td>
</tr>
<tr>
<td>- Sample Problem Solving for SSME Math Model</td>
<td>14</td>
</tr>
<tr>
<td>- Knowledge Base</td>
<td>14</td>
</tr>
<tr>
<td>- Generic Equations</td>
<td>15</td>
</tr>
<tr>
<td>- Information Base</td>
<td>21</td>
</tr>
<tr>
<td>- System Generated Equations</td>
<td>37</td>
</tr>
<tr>
<td>- System Generated FORTRAN Code</td>
<td>43</td>
</tr>
<tr>
<td>- System Generated Documentation</td>
<td>46</td>
</tr>
<tr>
<td>- Robot Manipulators</td>
<td>55</td>
</tr>
<tr>
<td>- System Generated Equations</td>
<td>57</td>
</tr>
</tbody>
</table>

**PRECEDING PAGE BLANK NOT FILMED**
This report describes a methodology for automatic mathematical modeling and generating simulation models. The models will be verified by running in a test environment using standard profiles with the results compared against known results. The major objective is to create a user-friendly environment for engineers to design, maintain, and verify their models; to automatically convert the mathematical models into conventional code for computation; and, finally, to document the model automatically.

The traditional method for simulation modeling is to define, derive, and organize the equations and then to develop the program for manual computation. If any modification or correction needs to be done in the design or mathematical equations, it will create a tremendous amount of work for the rest of the process. Generally, any model will require several modifications before it matches a real system. Historically, the modifications have been time consuming and a fertile source of errors.

Some of the available symbolic mathematics tools will help to derive the equation symbolically, and automatically produce the final equation and program code. The use of the LISP language can build the user-friendly interface and generate the knowledge base for the symbols mathematics tool to build equations and program codes. After the equations and codes have been stored in the files, the LISP program can then combine and organize them and create a complete compiled source code for any conventional language requirement. A demonstration program was designed for modeling the Real-Time Simulation system. It is written in LISP and MACSYMA and runs on a Symbolics 3670 LISP Machine. It contains an initial set of elements for the Real-Time Simulation system and a questionnaire that allows the engineer to answer a set of questions to specify a particular model. The system is then able to automatically generate the model and FORTRAN code. The future goal which is under development is to download the FORTRAN code to the VAX/VMS system for conventional computation. The mathematical model will be verified in the test environment and the solution compared with the real data profile. Control, display and ancillary systems will be developed which will allow the execution of a predefined profile (from a file) or interactive modification of the variables. Variables can be displayed in a text format or as a plot from the control and display system. An execution control module will be available to allow time and sequence control of the model and peripheral models. A recording module will be available for execution in series with the math model and will record (optional) interface variables transparently. This module will use packet definition variables and circular queues to determine what and when to record.

The use of Artificial Intelligence (AI) techniques has shown that the process of simulation modeling can be simplified.
II. INTRODUCTION

Mathematical modeling for real-time simulation programs is a very complicated process which includes analysis, design, and the generation of complex equations and programs. Generally, the model will require several modifications before it will match a real system. Historically, the modifications have been time consuming and a fertile source of errors.

The use of AI techniques has shown that this process can be simplified. Some of the AI software tools will allow us to create a user-friendly environment for engineers to design, maintain, and verify their model and also to automatically convert the mathematical model into a conventional programming language for execution.

III. PURPOSE

The major objective is to develop and create a very comfortable environment for engineers to design and maintain their model. The automatic Mathematical Modeling generates knowledge base, mathematical equations, and conventional program codes. It helps to simplify the process of the modeling, and cuts down the development time and errors.

The automatic generation of a math model in simulations will add confidence in the simulation. Using interfaces which are well defined and validated will reduce programming and debug time and will allow concentration on logic and equations.

This technique can be applied to many different science and engineering projects.

IV. INTELLIGENT INTERFACES

The intelligent interfaces include:

1) The user interface

2) The interface between knowledge base and automatic model generation

3) The interface between modules comprising the simulation.

Automatic modeling requires a very friendly work environment to collect the necessary information and to generate the knowledge base. Through the knowledge base information, the system automatically creates the mathematical equations and generic program codes for equations. Next, all the equation codes are generated and linked by another AI program to combine and organize the codes together and to build a completely compiled program.
An intermodule interface will become part of the automatic model generation process. Variables will be identified and an external interface will provide the variable and its type to external modules. These external modules will record, display, or modify the variable.

**V. AUTOMATIC MODELING**

The traditional method for simulation modeling is to manually define, derive, and organize the equations and then to develop the program for computation. If any modifications or corrections need to be done in the design or mathematical equations, it will create a tremendous amount of work for the rest of the process. Generally the model will require several modifications before it matches a real system. Historically, the modifications have been time consuming and a fertile source of errors.

Some of the available symbolic mathematics tools will help us to derive the equation symbolically and to automatically produce the final equation and program code. The use of LISP language can build the user-friendly interface and generate the knowledge base for the symbolics mathematics tool to build equations and program codes. After the equations and codes have been stored in the files, the LISP program can then combine and organize them and create a complete compiled source code for any conventional language you require.

The use of AI techniques has shown that the process of simulation modeling can be simplified.

Control, display, and ancillary systems will be developed which will allow the execution of a predefined profile (from a file) or interactive modification of the variables. Variables can be displayed in a text format or as a plot from the control and display system. An execution control module will be available to allow time and sequence control of the model and peripheral models. A recording module will be available for execution in series with the math model and will record (or not record) interface variables transparently. This module will use packet definition variables and circular queues to determine what and when to record.

**VI. SYSTEM OVERVIEW**

A. Flow Diagram
START

Define the problem

Define generic equations

Lisp program for user interface

Automatic knowledge base generation

Read knowledge Base and automatically generate the Macsyma code for Symbolic Math. tool "Macsyma" to recognize the information

Get in to Macsyma window to process the set of Macsyma code which has been generated and automatically create the equations and FORTRAN code for the model

Save the final symbolic equation and FORTRAN equation code into the disk files

Another set of Lisp function to read the equation code and build a compiled conventional program for computation

Verifying the result and maintain the equations or knowledge base and process the whole cycle again until correct
B. Basic Software and Hardware Requirements

The Basic software tools used were “MACSYMA” and “LISP.” The hardware was “Symbolics 3670” and “VAX.” We used a Symbolics for the equation and conventional code generation and then downloaded to the VAX system for computation.

There are a number of tools available for Symbolic Mathematical equation and conventional program code generation. Some examples are as follows:

<table>
<thead>
<tr>
<th>Symbolics Math Tool</th>
<th>Memory Requirements</th>
<th>Available Computers</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACSYMA</td>
<td>2 Megabytes</td>
<td>DEC 10 or 20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VAX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Honeywell 6000 Series</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Symbolics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LM2 or 3600</td>
</tr>
<tr>
<td>SMP</td>
<td>2 Megabytes</td>
<td>VAX 730, 750, or 780</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Using UNIX operating system</td>
</tr>
<tr>
<td>REDUCE</td>
<td>350 kilobytes</td>
<td>IBM-360 or 370</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DEC 10 or 20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VAX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UNIVAC 1100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CDC CYBER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRAY-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BURROUGHS 6700</td>
</tr>
<tr>
<td></td>
<td></td>
<td>APOLLO DOMAIN AND OTHERS</td>
</tr>
<tr>
<td>MuMATH</td>
<td>512 kilobytes</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>MAPLE</td>
<td>350 kilobytes</td>
<td>VAX Using UNIX 4.3</td>
</tr>
<tr>
<td>ALTRAN</td>
<td>270 kilobytes</td>
<td>1966 Standard FORTRAN</td>
</tr>
<tr>
<td>FORMAC</td>
<td>150 kilobytes</td>
<td>IBM-360 or compatible machine</td>
</tr>
<tr>
<td>SAC-2</td>
<td>120 kilobytes</td>
<td>1966 Standard Fortran</td>
</tr>
</tbody>
</table>

Hardware requirements are for the support of tasking and for the provision of adequate address space.
Features for the Most Popular Tools

MAC'S SYmbolic MANipulation (MACSYMA)

Features:

1. Solve equations, factor polynomials, expand functions in series
2. Equation simplification
3. Manipulate matrices, trigonometric equation
4. Integration and differentiation
5. Automatic code generation for FORTRAN program
6. File handling system
7. Interactive help facility
8. Control function and subroutine procedures
9. 3D graphics, automatic scaling, and different viewpoints
10. Interactive system
11. Excellent working environment
12. Ease of use

Available computers:

DEC 10 or 20
VAX
Honeywell 6000 Series
Symbolics LM2 or 3600

Memory requirements:

Two Megabytes
Symbolic Manipulation Program (SMP)

Features:

1. Solve equations, factor polynomials, expand functions in series
2. Equation simplification
3. Manipulate matrices, trigonometric equation
4. Integration and differentiation
5. Automatic code generation for FORTRAN or C program
6. File handling system
7. Interactive help facility
8. Control function and subroutine procedures
9. 3D graphics
10. Interactive system
11. Excellent working environment
12. Ease of use

Available computers:

VAX

Memory requirements:

Two megabytes

REDUCE II

Features:

1. Solve equations, factor polynomials
2. Equation simplification
3. Manipulate matrices
4. Integration and differentiation

5. Automatic code generation for FORTRAN program

6. File handling system

7. Control function and subroutine procedure

8. Interactive system

Available Computers:

- IBM-360 or 370
- DEC 10 or 20
- VAX
- UNIVAC 1100
- CDC CYBER
- CRAY-1
- BURROUGHS 6700
- APOLLO DOMAIN AND OTHERS

Memory requirements:

350 Kilobytes

**MuMath**

Features:

1. Solve equations, factor polynomials

2. Equation simplification

3. Manipulate matrices

4. Integration and differentiation

5. Interactive system

Available computers:

IBM PC or compatible system
VII. MAINTAINING AND VERIFYING

One of the most difficult problems in software today is the verification and maintenance of existing programs, especially programs built up over time with many programmers involved. This is particularly true for simulation programs. The traditional method of modifying simulation programs by rebuilding and recoding the model is a potential source of many errors.

The automatic system can eliminate most of these problems. If in error, the only place which need be considered to change is the knowledge base at the front, then the rest of the equation and program can automatically be changed. The procedure becomes much simpler, which makes it easy for the user to maintain and modify the model and program directly.

Improving maintenance and verification is a major goal for this type of model generation. Allowing maintenance to be concerned with equations and logic rather than the implementation is a goal for maintenance.

VIII. GENERAL APPLICATION

The AI technique for automatic mathematical modeling can apply to many problems which are required to be solved:

1) Complicated mathematical equations derivations that are hard or impossible for humans to do manually.

2) Equations which share the generic theory.

3) Models which need to be changed considerably.

Robotics, contact dynamics, and other problems which require a lot of complicated and long equation derivation in Matrix Multiplication can very easily be solved symbolically.

The Space Shuttle Main Engine Simulation contains several equations where there are sets of equations sharing the generic theory. The engine model is required to change when the engine design in hardware is modified. This is the case used for the Automatic Modeling Example.

There are many simulation models which need to be changed under different conditions. If they are done manually, it will cost a tremendous amount time to modify the model, change the program code, and verify the result before it can be used. The AI technique totally improves the environment, and the only place which needs to be modified is the knowledge base; the rest of the product can be automatically generated.

Spacelab training applications where several physical models will be developed for a single flight. Automatic generation of individual models from equations and logic will eliminate much of the repetitive system integration.
IX. EXAMPLE APPLICATION

A demonstration program, called Propulsion System Automatic Modeling (PSAM), was designed for modeling the Space Shuttle Main Engine Simulation Mathematic Model. PSAM is written in LISP and MACSYMA and runs on a Symbolics 3670 LISP machine.

The design goals for PSAM were to develop automatic modeling skills for propulsion systems and other scientific and engineering applications. The old Engine Model was used as an example to study.

PSAM includes the following features:

1) User-friendly interface

2) Automatic Knowledge Base generation

3) Automatic Equation and Coding generation

4) Automatic documentation generation.

The Space Shuttle Main Engine Simulation model was built up from the component process elements and their combination into the subprograms.

The component process elements are pump, hot gas turbine, hydraulic turbine, turbopump, combustor, valve, incompressible propellant flow, injector volume with priming for start, hot gas heat transfer, and regenerative cooling flow. The subprograms are fuel, oxidizer, and hot gas.

There are two types of information for a PSAM knowledge base. One is the component process elements generic equations, and the other is the information base for the combination of the Space Shuttle Main Engine model subprograms and component process elements.

The system collects the detailed requirement and generates a set of specific equations for the component process elements and subprograms.

PSAM has the ability to:

1) Create or maintain the knowledge base

2) Load different knowledge

3) Automatically generate equations

4) Output generated equation or FORTRAN code to disk file or option for printout of the Laser Printer

5) Automatic documentation.
The FORTRAN code is in generic conventional program format only for the equations. Another part of the LISP program will combine all the subprograms and component process elements equations and append the header program codes to become a whole compiled program. The coding format for the equations are the same for many available conventional software languages such as FORTRAN, C, or ADA, etc. So the final program can be built on whichever conventional software the user requires by changing only the header requirements and I/O functions.

Several versions and configurations of SSME will be generated and run with a version of the engine controller software.

Another example program was designed for automated kinematic equation generation for robot manipulators. Any manipulator can be considered to consist of a series of links connected together by joints. Historically, the homogeneous transformation describing the relation between one link to the next has been called an A matrix. The position and orientation of the nth link in base coordinates are given by the matrix product

\[
T_n = A_1 * A_2 * A_3 * \ldots A_n
\]

These products of A matrices have been called T matrices. An n-link manipulator can have n degrees of freedom, one for each link, and can be positioned and oriented arbitrarily within its range of freedom. Most robot controllers equations were developed manually. The complicated matrix multiplication is inverse with trigonometric functions. It is tedious and difficult. A small, careless mistake can cost major modification of the design. Using the AI technique to derive and simplify the equations symbolically and automatically can reduce a tremendous amount of effort.

These are the proof of concept examples which lead to future projects.

X. FUTURE PLANS

Currently the existing project is being used to test the method. In the future this technique can be applied to numerous new projects. The goal is to build the simulation model and maintain it automatically.

The knowledge base will also contain detailed information for automatic documentation.

The future plans for PSAM are to generate models based upon logic and flow rather than equations, and to build control and display modules which will complement the models.
APPENDIX

Sample Problem Solving for

- Space Shuttle Main Engine Math Model
- Robot Manipulator
SSME Math Model
Knowledge Base
Generic Equations
Information Base
Generic Equations

(D105) A1: >PSAM>Macsyma-kb>psam-generic-equation.out.1

(C106) BATCH("a1: >psam>macsyma-kb>psam-generic-equation.kb");

(C107) "**********PUMP**********$

(C108) "Pump Flow Variable"

(C109) PFV(F, B, DW, S):=
Block ( F = B * (DW / S));

(D109)

(C110) "Pump Total Pressure"

(C111) PTP(PP, P, B, S, T):=
Block ( PP = P + B * (S)^2 * T);

(D111)

(C112) "Pump Torque"

(C113) PT(R, B, S, T):=
Block ( R = B * (S)^2 * T);

(D113)

(C114) "**********TURBINE**********$

(C115) "Turbine Torque"

(C116) TT(R, B, P, T):=
Block ( R = B * P * T);

(D116)

(C117) "Turbine Speed Parameter"

(C118) TSP(M, B, S, T):=
Block ( M = B * S / (T^2));

(D118)

(C119) "Turbine Weight Flowrate"

(C120) TWF(DW, B, dP, r):=
Block ( DW = (B * (dP) * r)^(1/2));

(D120)

(C121) "**********TURBOPUMP SPEED**********$

(C122) "TurboPump Speed"

(C123) TPS(S, B, Tq, SO):=
Block ( S = B * 'integrate ( Tq, T) + SO);

(D123)

(C124) "**********VALVE**********$

(C125) "Valve Area"

(C126) VA(A, T, X):=
Block ( A = T * X);

(D126)

(C127) "**********FLOW**********$

(C128) "Fuel Flow"

(C129) FF(DWF, Bi, P, P, B, R, DWO):=
Block ( DWF: DWF(t), DWO:DWO(t), DWF = B * 'integrate (((P9 - P - (Bi / R)) * DWF^2), t) + DWO);

(D129)
\[
\text{DWO}_{2} = 2 \quad \text{DWO} = \text{BK} \left( \text{PPOS} - \text{P} \right) \\
\text{A} \\
\text{AB}
\]

(C132) "Variable in injector Priming function, Dimensionless"

\[
\text{VIPF}(E, \text{BH}, \text{DWO}, \text{DWI}, \text{EO}) := \text{Block}(\text{DWO} : \text{DWO}(t), \text{DWI} : \text{DWI}(t), E = \text{BH} \int (\text{DWO} - \text{DWI}, t) \cdot \text{EO});
\]

(C133) "Weight Flowrate Injector"

\[
\text{WFI}(\text{DWI}, \text{DWO}, \text{EO}) := \text{Block}(\text{DWI} = \text{DWO} \cdot \text{EO});
\]

(C134) "Combuseter Total Pressure"

\[
\text{CTP}(P, \text{BI}, \text{DWI}, \text{DW2}, \text{BJ}, \text{DW3}, \text{PO}) := \text{Block}(\text{DWI} : \text{DWI}(t), \text{DW2} : \text{DW2}(t), \text{DW3} : \text{DW3}(t), P = \text{BI} \int (\text{DWI} + \text{DW2} - \text{BJ} \cdot \text{DW3}, t) \cdot \text{PO});
\]

(C135) "Combuseter Fuel Weight Flowrate"

\[
\text{WFF}(F, \text{DW2}, \text{DW}) := \text{Block}(F = \text{DW2} / (\text{DW1} + \text{DW2}));
\]

(C136) "Combuseter Temperature"

\[
\text{CT}(T, \text{TT}, \text{BK}, \text{T9}) := \text{Block}(T = \text{TT} + \text{BK} \cdot \text{T9});
\]

(C137) "Cooling Element Total Pressure"

\[
\text{CETP}(P, \text{B}, \text{DQW1}, \text{DQW2}, \text{H3}, \text{DWI}, \text{H}, \text{DW}, \text{PO}) := \text{Block}(\text{DQW1} : \text{DQW1}(t), \text{DQW2} : \text{DQW2}(t), \text{DWI} : \text{DWI}(t), \text{DW} : \text{DW}(t), P = \text{B} \int (\text{DQW1} + \text{DQW2} + \text{H3} \cdot \text{DWI} - \text{H} \cdot \text{DW}, t) \cdot \text{PO});
\]
Cooling Element Specific Enthalpy

CESE(H,B,T) := Block(H = B ? T); CESE(H, B, T) := Block(H = B ? T)

Cooling Element Weight Flowrate Main Chamber Heat Exchanger

CEWFM(DW,Bi,P,P5,Bj,R5,DW,DWO) := Block(DW = DW(t), DW = Bi ? integrate((P - P5 - (Bj / R5) * DW^2),t) + DWO);  

CEWFM(DW, Bi, P, P5, Bj, R5, DW, DWO) := Block(DW = DW(t), DW = Bi ? (P - P5 - ------ ) dT + DWO)

Cooling Element Density

CED(R,B,DWi,DW,RO) := Block(DWi = DWi(t), DW = DW(t), R = B ? integrate((DWi - DW),t) + RO);  

CED(R, B, DWi, DW, RO) := Block(DWi = DWi(T), DW = DW(T), R = B ? (DWi - DW) dT + RO)

Cooling Element Temperature

CET(T, B, P,R) := Block(T = B ? (P / R)); CET(T, B, P, R) := Block(T = ---) R

Cooling Element Heat Transfer Rate Hot Gas Wall

HT(DQw,B,T,Tw,DWi) := Block( DQw = B ? (1.0 + 0.002 * T) * (Tw-T) * (DWi)^0.8);  

HT(DQw, B, T, Tw, DWi) := Block(DQw = B (1.0 + 0.002 T) (TW - T) DWI )

Cooling Element Heat Transfer Rate TC

HTTC(DQtc,B,Tc,Twl,DWcn) := Block(DQtc = B ? (Tc - Twl) * DWcn^0.8);  

HTTC(DQTC, 8, TC, TW1, DWCN) := Block(DQTC = B (TC - TW1) DWCN)

Cooling Element Hot Gas Wall Temperature

HGT(TT,B,DQwZ,TO) := Block( DQwZ = DQwZ(t), TT = B ? integrate((-DQwZ),t) + TO);  

HGT(TT, B, DQWZ, TO) := Block(DQWZ(DQWZ(T), TT = B ? (-DQWZ) dT + TO)

Cooling Element Ambient thrust Chamber Temperature

ATT(TT,B,DQw2,T0) := Block( DQw2 = DQw2(t), TT = B ? integrate((-DQw2),t) + T0);  

ATT(TT, B, DQW2, T0) := Block(DQW2(DQW2(T), TT = B ? (-DQW2) dT + T0)

Flowmeter

VF(Q,C1,T,C2,C3,C4,DW,C5,S) := Block(Q = [1.0 - C1 * (T - C2)]* C3 + C4 * DW + C5 * S);  

VF(Q, C1, T, C2, C3, C4, DW, C5, S) := Block(Q = [1.0 - C1 (T - C2)] [C3 + C4 DW
"Total Pressure"$
\text{TP}(P, Pf, RF, DW, R) := \text{BLOCK}(P = Pf - RF \cdot DW^2) / R;

"Flowmeter Temperature"$
\text{FT}(T, Ti, S, Coe) := \text{BLOCK}(T = Ti + Coe \cdot S);

"Preburner Fuel Supply Duct"$
\text{SDTP}(P, B, DW1, DW2, DW3, DW4, PO) := \text{BLOCK}(DW1(t), DW2(t), DW3(t), DW4(t), P = B \cdot \int (DW1 - DW2 - DW3 + DW4) \cdot dt + PO);

"Supply Duct Total Pressure"$
\text{SDT}(T, B, P, R) := \text{BLOCK}(T = B \cdot \frac{P}{R});

"Supply Duct Density"$
\text{SDD}(R, DW1, DW2, RR, B) := \text{BLOCK}(R = \frac{[DW1 + DW2]}{[DW2/RR + B \cdot DW1]});

"High Pressure Oxidizer Pump Preburner Boost Stage"$
\text{HPOPFW}(P3, B, S, TT, F) := \text{BLOCK}(P3 = B \cdot S \cdot TT * F);

"High Pressure Oxidizer Boostpump Pressure"$
\text{HPOPFP}(P3, B, S, TT, F, P2) := \text{BLOCK}(P3 = B \cdot S \cdot TT * F + P2);

"High Pressure Oxidizer Boostpump Weight Flowrate"$
\text{HPOP}((DW3, B1, P3, P, B2, DW0) := \text{BLOCK}(DW3:DW3(t), DW3 = B1 \cdot \int (P3 - P - B2 \cdot DW3 - DW0), t) + DW0);
[D184] \[ \text{DPWM}(\text{DW3}, \text{B1}, \text{P3}, \text{B2}, \text{DW0}) := \text{BLOCK}(\text{DW3} : \text{DW3}(t), \text{DW3} = \text{B1} \ (\text{P3} - \text{P} - \text{B2} \ 	ext{DW3} \ 	ext{DW3}) / \text{d}t + \text{DW0}) \]

[C185] "***************Main Chamber Injector, Fuel and Hot Gas ***************"

[C186] "Main Chamber Injector Total Pressure"

[C187] \[ \text{MCITP}(\text{P}, \text{B1}, \text{DW1}, \text{DW2}, \text{DW3}, \text{B2}, \text{DW4}, \text{PO}) := \text{BLOCK}(\text{DW1} : \text{DW1}(t), \text{DW2} : \text{DW2}(t), \text{DW3} : \text{DW3}(t), \text{DW4} : \text{DW4}(t), \text{P} = \text{B1} \ '\text{integrate}((\text{DW1} + \text{DW2} \ + \text{DW3} \ - \text{B2} \ \text{DW4}), \text{t}) + \text{PO}); \]

[C188] "Main Chamber Injector Weight Flowrate"

[C189] \[ \text{MCIW}(\text{DW}, \text{B}, \text{P1}, \text{T}, \text{PC}, \text{P2}, \text{TT}) := \text{BLOCK}(\text{DW} = \text{B} \ \text{P1} \ ((\text{T} \ \text{PC}) / \text{P2}) / \text{TT}^0.5); \]

[C189] \[ \text{MCIW}(\text{DW}, \text{B}, \text{P1}, \text{T}, \text{PC}, \text{P2}, \text{TT}) := \text{BLOCK}(\text{DW} = \text{B} \ \text{P1} \ ((\text{T} \ \text{PC}) / \text{P2}) / \text{TT}^0.5); \]

[C189] \[ \text{MCIW}(\text{DW}, \text{B}, \text{P1}, \text{T}, \text{PC}, \text{P2}, \text{TT}) := \text{BLOCK}(\text{DW} = \text{B} \ \text{P1} \ ((\text{T} \ \text{PC}) / \text{P2}) / \text{TT}^0.5); \]

[C190] "Main Chamber Injector Temperature"

[C191] \[ \text{MCIT}(\text{TF1}, \text{B1}, \text{TFP}, \text{B2}, \text{TOP}, \text{B3}, \text{T5}):= \text{BLOCK}(\text{TF1} = \text{B1} \ \text{TFP} + \text{B2} \ \text{TOP} + \text{B3} \ \text{T5}); \]

[C192] "***************Main Chamber Injector, Oxidizer ***************"

[C193] "Main Chamber Injector, Oxidizer Weight Flowrate"

[C194] \[ \text{MCOW}(\text{DW01}, \text{DW}, \text{E}) := \text{BLOCK}(\text{DW01} = \text{DW} \ \text{E}) \]

[C195] "Main Chamber Injector, Oxidizer Weight"

[C196] \[ \text{MCW}(\text{W}, \text{DW1}, \text{DW2}, \text{WO}) := \text{BLOCK}(\text{DW1} : \text{DW1}(t), \text{DW2} : \text{DW2}(t), \text{W} = \ '\text{integrate}((\text{DW1} - \text{DW2}), \text{t}) + \text{WO}); \]

[C197] "Main Chamber injector, Oxidizer Variable in injector Priming function"

[C198] \[ \text{MCV}(\text{E}, \text{W}) := \text{BLOCK}(\text{E} = \text{W}) \]

[C199] "***************Main Thrust Chamber ***************"

[C200] "Combuster Injector end static Total Pressure"

[C201] \[ \text{CITP}(\text{P}, \text{B}, \text{P}) := \text{BLOCK}(\text{P} = \text{B} \ \text{P}) \]

[C202] "Main thrust Chamber Total Pressure"

[C203] \[ \text{MCITP}(\text{P}, \text{B1}, \text{B2}, \text{DW1}, \text{DW2}, \text{DW3}, \text{PO}) := \text{BLOCK}(\text{DW1} : \text{DW1}(t), \text{DW2} : \text{DW2}(t), \text{DW3} : \text{DW3}(t), \text{P} = \text{B1} \ '\text{integrate}((\text{B2} \ \text{DW1} + \text{DW2} - \text{DW3}), \text{t}) + \text{PO}); \]
(D203) \( \text{MCTCP}(P, B_1, B_2, DW_1, DW_2, DW_3, P_0) := \text{BLOCK}(DW_1 : \text{DW}_1(t), \text{DW}_2 : \text{DW}_2(t), \text{DW}_3 : \text{DW}_3(t), P = B_1 I \left( B_2 \frac{\text{DW}_1 + \text{DW}_2 - \text{DW}_3}{\text{d}t + P_0} \right) \) \\

(C204) "**************************Nozzle Bypass Element**************************"$

(C205) "Nozzle Heat exchanger bypass Weight Flowrate"$

(C206) \( \text{NHWF}(D_W, B_1, P, P_9, B_2, D_W_1, A, D_W_0) := \text{BLOCK}(D_W_1 : D_W_1(t), D_W = B_1 \int (P - P_9 - B_2 \frac{D_W_1}{A}) dt + D_W_0) \) \\

(C207) "Coolant Control Valve"$

(C208) \( \text{CCV}(D_W, D_W_1) := \text{BLOCK}(D_W = D_W_1) \) \\

(C209) \( \text{CLOSEFILE}() ; \)
(Deframe Pop3)
(Class BOOST-STAGE)
(Unit "Flow-Variable")
(Nomenclature "High Pressure Oxidizer boostpump Flow variable")
(Input-parameter B43 DWop3 S02))

(Deframe Top3)
(Class BOOST-STAGE)
(Unit "Torque")
(Nomenclature "High Pressure Oxidizer Boostpump Torque")
(Input-parameter B44 So2 Trop3 Pop3))

(Deframe Pod3)
(Class BOOST-STAGE)
(Unit "Pressure")
(Nomenclature "High Pressure Oxidizer Boostpump Pressure")
(Input-parameter B45 So2 Tpop3 Pop3 Pod2))

(Deframe DWop3)
(Class BOOST-STAGE)
(Unit "Weight-Flowrate")
(Nomenclature "High Pressure Oxidizer Boostpump Weight Flowrate")
(Input-parameter B46 Pod3 Ppos B104 DWop30))

(Deframe Trop3)
(Class BOOST-STAGE)
(Unit "Table-Function")
(Nomenclature "High Pressure Oxidizer Boostpump Torque")
(Input-parameter ))

(Deframe Tpop3)
(Class BOOST-STAGE)
(Unit "Table-Function")
(Nomenclature "High Pressure Oxidizer boostpump Pressure Rise Character")
(Input-parameter ))

(Deframe DWop30)
(Class BOOST-STAGE)
(Unit "Initial-Condition")
(Nomenclature "DWop3 when t=0")
(Input-parameter 105.43))
(Deframe P5
  (Class COOLING)
  (Unit "Total-Pressure")
  (Nomenclature "Main Chamber Heat Exchanger Total Pressure")
  (Input-parameter 1 DQwl5 DQw25 H3 Dwmc H5 DW5 P50))

(Deframe P4
  (Class COOLING)
  (Unit "Total-Pressure")
  (Nomenclature "Nozzle Regenerative Exchanger Element Total Pressure")
  (Input-parameter b84 DQw14 DQw24 H4 D4 H3 Dwfn P40))

(Deframe H5
  (Class COOLING)
  (Unit "Specific-Enthalpy")
  (Nomenclature "Main Chamber Heat Exchanger Specific Enthalpy")
  (Input-parameter Bl01 T5))

(Deframe H4
  (Class COOLING)
  (Unit "Specific-Enthalpy")
  (Nomenclature "Nozzle Regenerative Cooling Element Specific Enthalpy")
  (Input-parameter Bl02 T4))

(Deframe Dw5
  (Class COOLING)
  (Unit "Weight-Flowrate")
  (Nomenclature "Main Chamber Heat Weight Flowrate")
  (Input-parameter B75 Pmfvd P5 B76 R5 Dwmc Dwmc0))

(Deframe Dw4
  (Class COOLING)
  (Unit "Weight-Flowrate")
  (Nomenclature "Nozzle Regenerative Cooling Weight Flowrate")
  (Input-parameter B94 P4 P9 B95 R4 Dw4 Dw40))

(Deframe R5
  (Class COOLING)
  (Unit "Density")
  (Nomenclature "Main Chamber Element Density")
  (Input-parameter B77 Dwmc Dw5 R50))

(Deframe R4
  (Class COOLING)
  (Unit "Density")
  (Nomenclature "Nozzle Regenerative Element Density")
  (Input-parameter B87 Dwfn Dw4 R40))

(Deframe T5
  (Class COOLING)
  (Unit "Temperature")
  (Nomenclature "Main Chamber Heat Exchanger Temperature")
  (Input-parameter B78 P5 R5))

(Deframe T4
  (Class COOLING)
  (Unit "Temperature")
  (Nomenclature "Nozzle Regenerative Cooling Temperature")
  (Input-parameter B88 P4 R4))

(Deframe Dw15
  (Class COOLING)
  (Unit "Heat-Transfer")
  (Nomenclature "Main Chamber Hot Gas Heat Transfer Rate")
  (Input-parameter B79 T5 TW15 Dwmc))

(Deframe Dw25
  (Class COOLING)
  (Unit "Heat-Transfer")
  (Nomenclature "Main Chamber Ambient Thrust Chamber Heat Transfer Rate")
  (Input-parameter B80 T5 TW25 Dwmc))

(Deframe Dw14
  (Class COOLING)
  (Unit "Heat-Transfer")
  (Nomenclature "Nozzle Regenerative Hot Gas Heat Transfer Rate")
  (Input-parameter B89 T4 TW14 Dwfn))

(Deframe Dw24
  (Class COOLING)
  (Unit "Heat-Transfer")
  (Nomenclature "Nozzle Regenerative Ambient Thrust Heat Transfer Rate")
  (Input-parameter B90 T4 TW24 Dwfn))

(Deframe Dwtc5
  (Class COOLING)
  (Unit "Heat-Transfer-TC")
  (Nomenclature "Main Chamber Cooling Element Heat Transfer Rate TC")
  (Input-parameter B81 To Tw15 Dwcn))

(Deframe Dwtc4
  (Class COOLING)
  (Unit "Heat-Transfer-TC")
  (Nomenclature "Nozzle Regenerative Heat Transfer Rate TC")

23
(Input-parameter B91 Tw14 Dwcn)

(Deframe Tw15
(Class COOLING)
(Unit "Hot-Gas-Temperature")
(Nomenclature "Hot Gas Temperature")
(Input-parameter B82 Dqtc5 Dqwl5 Tw150))

(Deframe Tw25
(Class COOLING)
(Unit "Ambient-Thrust-Temperature")
(Nomenclature "Ambient thrust chamber Temperature")
(Input-parameter B83 Dqww25 Tw250))

(Deframe Tw14
(Class COOLING)
(Unit "Hot-Gas-Temperature")
(Nomenclature "Hot Gas temperature")
(Input-parameter B92 Dqtc4 Dqwl4 Tw140))

(Deframe Tw24
(Class COOLING)
(Unit "Ambient-Thrust-Temperature")
(Nomenclature "Ambient Thrust Temperature")
(Input-parameter B93 Dqww24 Tw240))

(Deframe P50
(Class COOLING)
(Unit "Initial-Condition")
(Nomenclature "P5 when t=0")
(Input-parameter 5143.0))

(Deframe P40
(Class COOLING)
(Unit "Initial-Condition")
(Nomenclature "P4 when t=0")
(Input-parameter 6794.0))

(Deframe Dwmc0
(Class COOLING)
(Unit "Initial-Condition")
(Nomenclature "DWmc when t=0")
(Input-parameter 35.24))

(Deframe Dw40
(Class COOLING)
(Unit "Initial-Condition")
(Nomenclature "DW4 when t=0")
(Input-parameter 41.98))

(Deframe R50
(Class COOLING)
(Unit "Initial-Condition")
(Nomenclature "R5 when t=0")
(Input-parameter 0.001402))

(Deframe R40
(Class COOLING)
(Unit "Initial-Condition")
(Nomenclature "R4 when t=0")
(Input-parameter 6650.0))

(Deframe Tw150
(Class COOLING)
(Unit "Initial-Condition")
(Nomenclature "TW15 when t=0")
(Input-parameter 1330.0))

(Deframe Tw250
(Class COOLING)
(Unit "Initial-Condition")
(Nomenclature "TW25 when t=0")
(Input-parameter 544.5))

(Deframe Tw140
(Class COOLING)
(Unit "Initial-Condition")
(Nomenclature "TW14 when t=0")
(Input-parameter 1260.0))

(Deframe Tw240
(Class COOLING)
(Unit "Initial-Condition")
(Nomenclature "TW24 when t=0")
(Input-parameter 679.3))
(Deframe DWfpf
  (Class FLOW)
  (Unit "Fuel-Flow")
  (Nomenclature "Fuel Preburner Fuel Flow")
  (Input-parameter B50 P9 Pfp B51 R9 DWfpf0))

(Deframe DWopf
  (Class FLOW)
  (Unit "Fuel-Flow")
  (Nomenclature "Oxidizer Preburner Fuel Flow")
  (Input-parameter B57 P9 Pop B58 R9 DWopf0))

(Deframe DWfpo
  (Class FLOW)
  (Unit "Fuel-Flow")
  (Nomenclature "Fuel Preburner Oxidizer Weight Flowrate")
  (Input-parameter B52 Ppos Pfp B53 A Ab B54 DWfpoi DWfpo0))

(Deframe DWopo
  (Class FLOW)
  (Unit "Oxidizer-Flow")
  (Nomenclature "Oxidizer Preburner Oxidizer Weight Flowrate")
  (Input-parameter B59 Ppos Pop B61 A Ab B60 DWopo DWopo0))

(Deframe Efp0
  (Class FLOW)
  (Unit "Variable-in-injector-Priming")
  (Nomenclature "Fuel Preburner Oxidizer Variable in injector Priming")
  (Input-parameter B55 DWfpo DWfpo0 Efpo0))

(Deframe Eopo
  (Class FLOW)
  (Unit "Variable-in-injector-Priming")
  (Nomenclature "Oxidizer Preburner Oxidizer Variable in injector Priming")
  (Input-parameter B62 DWopo DWopo0 Eopo0))

(Deframe DWfpoi
  (Class FLOW)
  (Unit "Weight-Flowrate-Injector")
  (Nomenclature "Fuel Preburner Oxidizer Injector Weight Flowrate")
  (Input-parameter DWfpo Efpo))

(Deframe DWfpo
  (Class FLOW)
  (Unit "Weight-Flowrate-Injector")
  (Nomenclature "Oxidizer Preburner Oxidizer Injector Weight Flowrate")
  (Input-parameter DWfpoi DWfpo Efpo))

(Deframe DWfpf0
  (Class PUMP)
  (Unit "Initial-Condition")
  (Nomenclature "DWfpf when t=0")
  (Input-parameter 83.15))

(Deframe DWopf0
  (Class PUMP)
  (Unit "Initial-Condition")
  (Nomenclature "DWopf when t=0")
  (Input-parameter 38.93))

(Deframe DWfpo0
  (Class PUMP)
  (Unit "Initial-Condition")
  (Nomenclature "DWfpo when t=0")
  (Input-parameter 79.30))

(Deframe DWopo0
  (Class PUMP)
  (Unit "Initial-Condition")
  (Nomenclature "DWopo when t=0")
  (Input-parameter 26.13))

(Deframe Efpo0
  (Class PUMP)
  (Unit "Initial-Condition")
  (Nomenclature "Efpo when t=0")
  (Input-parameter 1.0))

(Deframe Eopo0
  (Class PUMP)
  (Unit "Initial-Condition")
  (Nomenclature "Eopo when t=0")
  (Input-parameter 1.0))
(Deframe Pfi)
(Class MAIN-CHAMBER)
(Unit "Pressure")
(Nomenclature "Main Chamber Injector, fuel and Hot Gas Total Pressure")
(Input-parameter B64 DWfi1 DWot2 DWft2 B108 DWfi PfiO))

(Deframe DWfi)
(Class MAIN-CHAMBER)
(Unit "Weight-Flowrate")
(Nomenclature "Main Chamber injector weight flowrate")
(Input-parameter B64 Pfi Tpr Pc Ppi Tfi))

(Deframe Tfi)
(Class MAIN-CHAMBER)
(Unit "Temperature")
(Nomenclature "Main Chamber Temperature")
(Input-parameter B66 Tfp B67 Top B68 T5))

(Deframe DWoi)
(Class MAIN-CHAMBER)
(Unit "Oxidizer-Weight-Flowrate")
(Nomenclature "Main Chamber injector, Oxidizer Weight Flowrate")
(Input-parameter DWmov Eoi))

(Deframe Woi)
(Class MAIN-CHAMBER)
(Unit "Oxidizer-Weight")
(Nomenclature "Main Chamber injector, Oxidizer Weight")
(Input-parameter DWmov DWOi Woi))

(Deframe Eoi)
(Class MAIN-CHAMBER)
(Unit "Oxidizer-Variable")
(Nomenclature "Main Chamber injector, Oxidizer variable in injector priming function")
(Input-parameter TEOi Woi))

(Deframe PfiO)
(Class MAIN-CHAMBER)
(Unit "Initial-Condition")
(Nomenclature "Pfi when t=0")
(Input-parameter 3662.0))
(Deframe Pci6s)
(Class MAIN-THRUST-CHAMBER)
(Unit "Combustor-Total-Pressure")
(Nomenclature "Combuster Injector end static Total Pressure")
(Input-parameter B114 Pc))

(Deframe Pc
(Class MAIN-THRUST-CHAMBER)
(Unit "Main-thrust-Total-Pressure")
(Nomenclature "Main thrust chamber Total Pressure")
(Input-parameter B72 B109 DWfi DWoi DWon Pc0))

(Deframe Pc0
(Class MAIN-THRUST-CHAMBER)
(Unit "Initial-Condition")
(Nomenclature "Pc when t=0")
(Input-parameter 3330.0))
(Deframe DWfnbp)
  (Class NOZZLE-BYPASS)
  (Unit "Nozzle-Heat-Weight-Flowrate")
  (Nomenclature "Nozzle Heat Exchanger bypass weight flowrate")
  (Input-parameter B98 Pmfvd P9 B99 DWfnbp Accv DWfnbp0)

(Deframe DWccv)
  (Class NOZZLE-BYPASS)
  (Unit "Coolant-Control-Weight-Flowrate")
  (Nomenclature "Coolant Control Valve")
  (Input-parameter DWfnbp)

(Deframe DWfnbp0)
  (Class NOZZLE-BYPASS)
  (Unit "Initial-Condition")
  (Nomenclature "DWfnbp when t=0")
  (Input-parameter 0.19)
(Deframe P9
(Class PREBURNER-SUPPLY-DUCT)
(Unit "Supply-Duct-Total-Pressure")
(Nomenclature "Preburner Fuel Supply Element Total Pressure")
(Input-parameter B96 DW4 DWopf DWfpf DWfnbp P90))

(Deframe Ppos
(Class PREBURNER-SUPPLY-DUCT)
(Unit "Supply-Duct-Total-Pressure")
(Nomenclature "Preburner Oxidizer Supply Total Pressure")
(Input-parameter B49 DWop3 DWfpo Dwopo DW PposO))

(Deframe T9
(Class PREBURNER-SUPPLY-DUCT)
(Unit "Supply-Duct-Temperature")
(Nomenclature "Preburner Fuel Supply Element Temperature")
(Input-parameter B100 P9 R9))

(Deframe R9
(Class PREBURNER-SUPPLY-DUCT)
(Unit "Supply-Duct-Density")
(Nomenclature "Preburner Fuel Supply Element Density")
(Input-parameter DWfnbp DW4 R4 B105))

(Deframe P90
(Class PREBURNER-SUPPLY-DUCT)
(Unit "Initial-Condition")
(Nomenclature "P9 when t=0")
(Input-parameter 6650.0))

(Deframe PposO
(Class PREBURNER-SUPPLY-DUCT)
(Unit "Initial-Condition")
(Nomenclature "Ppos when t=0")
(Input-parameter 8457.0))
(Deframe Trfpl)
(Class PUMP)
(Unit "Table-Function")
(Nomenclature "Low Pressure Fuel Pump Pump Torque Characteristics")
(Input-parameter )

(Deframe Trfp2)
(Class PUMP)
(Unit "Table-Function")
(Nomenclature "High Pressure Fuel Pump Pump Pump Torque Characteristics")
(Input-parameter )

(Deframe Tropl)
(Class PUMP)
(Unit "Table-Function")
(Nomenclature "Low Pressure Oxidizer Pump Pump Pump Torque Characteristics")
(Input-parameter )

(Deframe Torque)
(Class PUMP)
(Unit "Torque")
(Nomenclature "High Pressure Oxidizer Pump \"Torque\")
(Input-parameter B41 So2 Trop2)

(Deframe Trop2)
(Class PUMP)
(Unit "Table-Function")
(Nomenclature "High Pressure Oxidizer Pump Pump Torque Characteristics")
(Input-parameter )
(Deframe \text{Sf1})
(Class TURBOPUMP)
(Unit "Rotational-Speed")
(Nomenclature "Low Pressure Fuel Turbopump Rotational Speed")
(Input-parameter B14 TQft1-TQfp1 Sf10))

(Deframe \text{Sf2})
(Class TURBOPUMP)
(Unit "Rotational-Speed")
(Nomenclature "High Pressure Fuel Turbopump Rotational Speed")
(Input-parameter B23 TQft2-TQfp2 Sf20))

(Deframe \text{Sol})
(Class TURBOPUMP)
(Unit "Rotational-Speed")
(Nomenclature "Low Pressure Oxidizer Turbopump Rotational Speed")
(Input-parameter B35 TQo1-TQop1 Sol0))

(Deframe \text{So2})
(Class TURBOPUMP)
(Unit "Rotational-Speed")
(Nomenclature "High Pressure Oxidizer Turbopump Rotational Speed")
(Input-parameter B42 TQo2-TQop2-TQop3 So20))

(Deframe \text{Sf10})
(Class TURBOPUMP)
(Unit "Initial-Condition")
(Nomenclature "Sf1 when t=0")
(Input-parameter 1638.0))

(Deframe \text{Sf20})
(Class TURBOPUMP)
(Unit "Initial-Condition")
(Nomenclature "Sf2 when t=0")
(Input-parameter 3922.0))

(Deframe \text{Sol0})
(Class TURBOPUMP)
(Unit "Initial-Condition")
(Nomenclature "Sol when t=0")
(Input-parameter 570.9))

(Deframe \text{So20})
(Class TURBOPUMP)
(Unit "Initial-Condition")
(Nomenclature "So2 when t=0")
(Input-parameter 3262.0))
(Deframe Amfv
  (Class VALVE)
  (Unit "Area")
  (Nomenclature "Main Fuel Valve Area")
  (Input-parameter TQmfv Xmfv))

(Deframe Amov
  (Class VALVE)
  (Unit "Area")
  (Nomenclature "Main Oxidizer Valve Area")
  (Input-parameter TQmov Xmov))

(Deframe Afpv
  (Class VALVE)
  (Unit "Area")
  (Nomenclature "Fuel Preburner Oxidizer Valve Area")
  (Input-parameter TQfpv Xfpv))

(Deframe Aopv
  (Class VALVE)
  (Unit "Area")
  (Nomenclature "Oxidizer Preburner Oxidizer Valve Area")
  (Input-parameter TQopv Xopv))
System Generated Equations
Component Processor Element Pump

\[
\begin{align*}
SO_2 &= B_{42} T (TQOT_2 - TQOP_3 - TQOP_2) + SO_20 \\
SO_1 &= B_{35} T (TQOT_1 - TQOP_1) + SO_10 \\
SF_2 &= B_{23} T (TQFT_2 - TQFP_2) + SF_20 \\
SF_1 &= B_{14} T (TQFT_1 - TQFP_1) + SF_10
\end{align*}
\]

Component Processor Element Turbine

\[
\begin{align*}
B_{48} \text{ POP TPR} \\
DWO_T_2 &= \sqrt{\text{TOP}} \\
POD_2 &= POD_1 \\
DWO_T_1 &= \sqrt{\text{ROT}_1 + B_{37}} \\
DWF_T_2 &= \sqrt{B_{24} PFP \text{ TPR}} \\
DWF_T_1 &= \sqrt{B_{17} (P_5 - PFI) R_3} \\
B_{47} \text{ SO}_2 \\
UOT_2 &= \sqrt{\text{ROP TOP}} \\
B_{38} \text{ SO}_1 \\
FOT_1 &= \sqrt{\text{DWO}_T_1} \\
B_{25} \text{ SF}_2 \\
UFT_2 &= \sqrt{\text{TFP}} \\
B_{15} \text{ SF}_1 \\
UFT_1 &= \sqrt{T_5} \\
TQOT_2 &= B_{29} \text{ POP TROT}_2 \\
2TQOT_1 &= B_{36} \text{ DWO}_T_1 \text{ TROT}_1 \\
TQFT_2 &= B_{26} \text{ PFP TRFT}_2 \\
TQFT_1 &= B_{16} P_5 \text{ TRFT}_1
\end{align*}
\]

Component Processor Element Turbopump

\[
\begin{align*}
SO_2(T) &= B_{42} \int (TROT_2 - TROP_3 - TROP_2) + SO_2(T) \\
G(T)
\end{align*}
\]

\[
\begin{align*}
SO_1(T) &= B_{35} \int (TROT_1 - TROP_1) + SO_1(T) \\
G(T)
\end{align*}
\]

\[
\begin{align*}
SF_2(T) &= B_{23} \int (TRFT_2 - TRFP_2) + SF_2(T) \\
G(T)
\end{align*}
\]

\[
\begin{align*}
SF_1(T) &= B_{14} \int (TRFT_1 - TRFP_1) + SF_1(T) \\
G(T)
\end{align*}
\]

ORIGINAL PAGE IS OF POOR QUALITY
Component Processor Element Valve

\[ \text{AOPV} = \text{TQOPV} \times \text{XOPV} \]
\[ \text{AFPV} = \text{TQFPV} \times \text{XFPV} \]
\[ \text{AMOV} = \text{TQMOV} \times \text{XMOV} \]
\[ \text{AMFV} = \text{TQMFV} \times \text{XMFV} \]

Component Processor Element Flow

\[ \text{DWPO} = \text{DWPO} \times \text{EOPO} \]
\[ \text{DWFPO} = \text{DWFPO} \times \text{EFPO} \]

\[ \text{DWO}(T) = B59 \left( - B60 \text{DWPO} (T) - \frac{\text{PPOS} - \text{POP}}{2} \right) dT + \text{DWCP} \]

\[ \text{DWFPO}(T) = B52 \left( - B54 \text{DFP} (T) - \frac{\text{PPOS} - \text{PFP}}{2} \right) dT + \text{DWPFP} \]

Component Processor Element Combustor

\[ \text{TOP} = \text{TFOP} + B113 T9 \]
\[ \text{TFP} = \text{TFFP} + B112 T9 \]

\[ \text{DWPPO} \]
\[ \text{PO} = \frac{\text{DWPPO} + \text{DW1}}{} \]
\[ \text{DWPFOI} \]
\[ \text{FFP} = \frac{\text{DWPFOI} + \text{DW1}}{} \]

\[ \text{PO} = B63 \left( - B107 \text{DWP2} (T) + \text{DWPPO} (T) + \text{DWPFP} (T) \right) dT + \text{POP} \]

\[ \text{FFP} = B56 \left( - B111 \text{DFP2} (T) + \text{DWPFOI} (T) + \text{DWPFP} (T) \right) dT + \text{PFP} \]

Component Processor Element Cooling

\[ \text{TW} = \text{TW24} - B93 \left( - \text{DQW2} (T) \right) dT \]
TW14 = TW140 + B92 \int (DQTC4(T) - DQW14(T)) dT

TW25 = TW250 - B83 \int DQW25(T) dT

TW15 = TW150 + B82 \int (DQTC5(T) - DQW15(T)) dT

DQTC4 = B91 DWCN (TC - TW14)

DQTC5 = B81 DWCN (TC - TW15)

DQW24 = B90 DWFN (0.002 T4 + 1.0) (TW24 - T4)

DQW14 = B89 DWFN (0.002 T4 + 1.0) (TW14 - T4)

DQW25 = B80 DWMC (0.002 T5 + 1.0) (TW25 - T5)

DQW15 = B79 DWMC (0.002 T5 + 1.0) (TW15 - T5)

B88 P4
T4 = ------
R4

B78 P5
T5 = ------
R5

R4 = B87 \int (DWFN(T) - DW4(T)) dT + R40

R5 = B77 \int (DWMC(T) - DW5(T)) dT + R50

DW4(T) = B94 \int (- \frac{P9 + P4}{R4}) dT + DW40

DWMC(T) = B75 \int (- \frac{PMFVD - P5}{R5}) dT + DWMC0

H4 = B102 T4
H5 = B101 T5

P4 = B84 \int (- P3 DWFN(T) + P1 DW4(T) + DQW24(T) + DQW15(T)) dT + P40
Component Processor Element Flowmeter

\[ P_5 = \int \left( H_3 \, D\!WMC(T) - H_5 \, D\!W5(T) + D\!W25(T) + D\!W15(T) \right) \, dT + P_{50} \]

\[ P_{OFM} = P_{OD2} - \frac{1}{2} \, S_{O2} \left( D\!W \!O \!T \!P \!R + D\!W \!O \!P \!3 + D\!W \!M \!O \!V \right) \, R_{FD2} \]

\[ P_{OFM} = \left( 0.02099 \, O_S - 164.0 \right) \, T_{O2} \]

\[ T_{FD2} = T_{FS} + 0.00547 \, S_{O2} \]

\[ T_{FD1} = T_{FS} + 0.0015789 \, S_{F1} \]

\[ 2 \times \left( D\!W \!O \!T \!P \!R + D\!W \!O \!P \!3 + D\!W \!M \!O \!V \right) \, R_{FD2} \]

\[ Q_{FM} = \left( 0.20206 \, S_{F1} + 101.323 \, D\!W \!D2 - 204.852 \right) \, (1.0 - 0.002789 \, T_{0.008783} \, (T_{F3} - 37.0)) \]

\[ Q_{F} = \left( 0.20206 \, S_{F1} + 101.323 \, D\!W \!D2 - 204.852 \right) \, (1.0 - 0.002789 \, T_{0.008783} \, (T_{F3} - 37.0)) \]

Component Processor Element Preburner Supply Duct

\[ R_9 = \left( D\!W \!F \!N \!B \!P + D\!W \!4 \right) \]

\[ D\!W \!4 = \frac{B_{105} \, D\!W \!F \!N \!B \!P}{R_{4}} \]

\[ B_{100} \, P_{9} \]

\[ T_9 = \frac{R_{9}}{2} \]

\[ P_{POS} = B_{49} \int \left( - D\!W \!O \!P \!O(T) + D\!W \!O \!P \!3(T) - D\!W \!F \!P \!O(T) + D\!W(T) \right) \, dT + P_{POS0} \]

\[ P_{9} = B_{96} \int \left( - D\!W \!O \!P \!F(T) - D\!W \!F \!P \!F(T) + 2 \, D\!W \!F \!N \!B \!P(T) \right) \, dT + P_{90} \]

Component Processor Element High Pressure Oxidizer Pump Preburner Boost-stage

\[ R_{D0} \]

\[ D\!W \!O \!P \!3(T) = B_{1} \int \left( - R_{2} \, D\!W \!O \!P \!3(T) + P_{3} - P \right) \, dT + D\!W_{0} \]

\[ P_{OD3} = B \, F \, S \, T \left( T_{T} + P_{2} \right) \]

\[ 2 \times \left( B_{44} \, F \, S \, T_{T} \right) \]

\[ B \, D\!W \]

\[ F_{0} = F \, T \]

\[ B \, D\!W \]

Component Processor Element Main Chamber Injector, Fuel, Hot Gas and Oxidizer

\[ E_{GI} = T_{E0} \, W_{0} \]
\[
\begin{align*}
\text{WOCI} &= \text{WOI} + I \left( \text{DWMOV}(T) - \text{DWOI}(T) \right) dT \\
\text{DWOI} &= \text{DWMOV} \text{EOI} \\
\text{TFI} &= B67 \text{ TOP} + B66 \text{ TFP} + B68 \text{ T5} \\
B64 \text{ PC PFI TPR} \\
\text{DWFI} &= \text{------------} \\
0.5 \\
\text{PFI TFI} \\
\text{PFI} &= B64 I \left( \text{DWOT2}(T) + \text{DWF2}(T) + \text{DWF1}(T) - B108 \text{DWFI}(T) \right) dT + \text{PFI0} \\
\end{align*}
\]

Component Processor Element Main Thrust Chamber

\[
\begin{align*}
\text{PC} &= B72 I \left( \text{DWOI}(T) + B109 \text{DWFI}(T) - \text{DCI}(T) \right) dT + \text{PC0} \\
\end{align*}
\]

Component Processor Element Nozzle Bypass Element

\[
\begin{align*}
\text{DWCCV} &= \text{DWFNB} \\
\text{DWFNB} &= B98 I \left( \text{-------------} + \text{PMFVD} - \text{P9} \right) dT + \text{DWFNB0} \\
\end{align*}
\]

\[
\begin{align*}
\text{ACCV} \\
\end{align*}
\]
System Generated FORTRAN Code
Component Processor Element Pump

\[
\begin{align*}
\text{TORQUE} &= B41*S02**2*TROP2 \\
\text{ROP1} &= B34*S01**2*TROP1 \\
\text{RFP2} &= B20*SF2**2*TRFP2 \\
\text{RFP1} &= B13*SF1**2*TRFP1 \\
\text{POD2} &= B40*S02**2*TPOP2 + POD1 \\
\text{POD1} &= B28*S01**2*TPOP1 + POS \\
\text{PFD2} &= B19*SF2**2*TFPP2 + PFD1 \\
\text{PFD1} &= B12*SF1**2*TFPP1 + PFS \\
\text{FOP2} &= B39*(DWOT1 + DWOP3 + DWMOV)/SO2 \\
\text{FDP1} &= B27*(DWOP3 + DWMOV)/SO1 \\
\text{FFP2} &= B18*DWFD2/SF2 \\
\text{FPF1} &= B11*DWFD1/SF1
\end{align*}
\]

Component Processor Element Turbine

\[
\begin{align*}
\text{DWOT2} &= \sqrt{B48*POP*TPR/TOP} \\
\text{DWOT1} &= \sqrt{(POD2-POD1)/(ROT1+B37)} \\
\text{DWFT2} &= \sqrt{B24*PFP*TPR} \\
\text{DWFT1} &= \sqrt{(B17*(F5-PFI)*R5)} \\
\text{UOT2} &= B47*SO2/\sqrt{ROP*TOP} \\
\text{UOT1} &= B38*SO1/\sqrt{DWOT1} \\
\text{UFT2} &= B25*SF2/\sqrt{TFP} \\
\text{UFT1} &= B15*SF1/\sqrt{T5} \\
\text{TOT2} &= B29*POP*TROT2 \\
\text{TOT1} &= B36*DWOT1**2*TROT1 \\
\text{TQFT2} &= B26*PFP*TRFT2 \\
\text{TQFT1} &= B16*P5*TRFT1
\end{align*}
\]

Component Processor Element Turbopump

\[
\begin{align*}
\text{SO2} &= B42*T*(TQOT2-TQOP3-TQOP2)+SO20 \\
\text{SO1} &= B35*T*(TQOT1-TQOP1)+SO10 \\
\text{SF2} &= B23*T*(TQFT2-TQFP2)+SF20 \\
\text{SF1} &= B14*T*(TQFT1-TQFP1)+SF10
\end{align*}
\]

Component Processor Element Valve

\[
\begin{align*}
\text{AOPV} &= TQOPV*XOPV \\
\text{AFPV} &= TQFPV*XFPV \\
\text{AMOV} &= TQMOV*XMOV \\
\text{AMFV} &= TQMFV*XMFV
\end{align*}
\]

Component Processor Element Flow

\[
\begin{align*}
\text{DWOPOI} &= DWPOI*EOPO \\
\text{DWFPFI} &= DWFPFI*EFPO \\
\text{DWPO1} &= B59*' \text{INTEGRATE} \left(-B60*DWPOI(T)*2-AB**2*B61*DWPOI(T)*2/1 \right) \\
\text{DWPOF1} &= B52*' \text{INTEGRATE} \left(-B54*DWPOFI(T)*2-AB**2*B53*DWPOFI(T)*2/1 \right) \\
\text{DWOPF} &= B*(-B57/R9-POP+P9)**\text{INTEGRATE} \left(DWOPF(T)**2, T \right) + DWPOO(T) \\
\text{DWFPF} &= B*(-B50/R9-PFP+P9)**\text{INTEGRATE} \left(DWFPF(T)**2, T \right) + DWPOO(T)
\end{align*}
\]

Component Processor Element Combuster

\[
\begin{align*}
\text{TOP} &= TTFOP+B113*T9 \\
\text{TFP} &= TTFPP+B112*T9 \\
\text{POP} &= DWOP1/(DWOP1+DW1) \\
\text{FPF} &= DWPOI/(DWPOI+DW1) \\
\text{POP} &= B63*' \text{INTEGRATE} \left(-B107*DWOT2(T)+DWPOI(T)+DWPOF(T)+T \right) + POP0 \\
\text{PFP} &= B56*' \text{INTEGRATE} \left(-B111*DWFT2(T)+DWPOI(T)+DWPOF(T)+T \right) + PFP0
\end{align*}
\]

Component Processor Element Cooling

\[
\text{TW24} = TW240-B93**\text{INTEGRATE} \left(DQW24(T), T \right)
\]
TW14 = TW140+B92*'INTEGRATE(DQTC4(T)-DQW14(T),T)
TW25 = TW250-B83*'INTEGRATE(DQW15(T),T)
TW15 = TW150+B82*'INTEGRATE(DQTC5(T)-DQW15(T),T)
DQTC4 = B91*DWCN**0.8*(TC-TW14)
DQTC5 = B81*DWCN**0.8*(TC-TW15)
DQW24 = B90*DWFN**0.8*(0.002*T4+1.0)*(TW24-T4)
DQW14 = B89*DWFN**0.8*(0.002*T4+1.0)*(TW14-T4)
DQW25 = B80*DWMC**0.8*(0.002*T5+1.0)*(TW25-T5)
DQW15 = B79*DWMC**0.8*(0.002*T5+1.0)*(TW15-T5)
T4 = B88*P4/R4
T5 = B78*P5/R5
R4 = B87*'INTEGRATE(DWFN(T)-DW4(T),T)+R40
R5 = B77*'INTEGRATE(DWMC(T)-DW5(T),T)+R50
DW4(T) = B94*'INTEGRATE(-B95*DW4(T)**2/R4-P9+P4,T)+DW40
DWMC(T) = B75*'INTEGRATE(-B76*DWMC(T)**2/R5+PMFVD-P5,T)+DWMC0
H4 = B102*T4
H5 = B101*T5
P4 = B84*'INTEGRATE(-H3*DWFN(T)+H4*DW4(T)+DQW24(T)+DQW14(T),T)+P40
P5 = 'INTEGRATE(H3*DWMC(T)-H5*DW5(T)+DQW25(T)+DQW15(T),T)+P50

Component Processor Element flowmeter

TOD2 = TOS+0.00547*SO2
TFP1 = TFS+0.0015789*SF1
POFM = POD2-(DWOTPR+DWOP3+DWMOV)**2*RFOD2/ROD2
PFFM = PFD1-DWFD2**2*RFCCD/RFD1
QOFM = [(0.0209912*SO2+6.19891*(DWOTPR+DWOP3+DWMOV)+17.5339)*(1.0-
1 0.002789*(TOS-164.0))]
QF = [(0.20206*SF1+101.323*DWFD2-204.852)*(1.0-0.008783*(TFS-37.0)
1 )

Component Processor Element Preburner Supply Duct

R9 = [(DWFBP+DW4)/(DW4/R4+B105*DWFBP)]
T9 = B100*P9/R9
PPOS = B49*'INTEGRATE(-DWPO(T)+DWOP3(T)-DWFP(T)+DW(T),T)+PPOS0
P9 = B96*'INTEGRATE(-DWOPF(T)-DWFPF(T)+2*DWFBP(T),T)+P90

Component Processor Element High Pressure Oxidizer Pump Preburner Boost-stage

DWOP3(T) = B1*'INTEGRATE(-B2*DWOP3(T)**2+P3-P,T)+DW0
POD3 = B*F*S**2*TT+P2
TOP3 = B44*F*S**2*TT
FOP3 = B*DW/S

Component Processor Element Main Chamber Injector, Fuel, Hot Gas and Oxidizer

EOI = TEOI*WOI
WOI = WOI+'INTEGRATE(DMOV(T)-DWOI(T),T)
DWOI = DMOV*EOI
TFI = B67*TOP+B66*TFP+B68*T5
DWFI = B64*PC*FPI*TR/(PFI*TFI**0.5)
PFI = B64*'INTEGRATE(DWOT2(T)+DWFT2(T)+DWFI(T)-B108*DWFI(T),T)+PF
1 10

Component Processor Element Main Thrust Chamber

PC = B72*'INTEGRATE(DWOI(T)+B109*DWFI(T)-DWCN(T),T)+PC0
PC = B114*PC

Component Processor Element Nozzle Bypass Element

DWCCV = DWFBP
DWFBP = B98*'INTEGRATE(-B99*DWFBP(T)/ACCV**2+PMFVD-P9,T)+DWFBP0

ORIGINAL PAGE IS OF POOR QUALITY
System Generated Documentation
Documentation for Component Processure PUMP

1. Low Pressure Fuel Pump
2. High Pressure Fuel Pump
3. Low Pressure Oxidizer Pump
4. High Pressure Oxidizer Pump

Generic Equations

\[ F = B_i \left[ \frac{DW}{S} \right] \quad ; \quad \text{Flow variable}, \ in^3 \]
\[ P = P_s + B_j \left( S \right)^2 \times T_p(F) \quad ; \quad \text{Total pressure}, \ Psia \]
\[ R = B_k \left( S \right)^2 \times T_r(F) \quad ; \quad \text{Torque}, \ \text{in-lb} \]

EOPO0 : Eopo when \( t=0 \)
EFPO0 : EFpo when \( t=0 \)
DWOP00 : DWopo when \( t=0 \)
DWFPO0 : Dwfpo when \( t=0 \)
DWOPF0 : DWopf when \( t=0 \)
DWFPF0 : Dwfpf when \( t=0 \)

TROP2 : High Pressure Oxidizer Pump Pump Torque Characteristics
TORQUE : High Pressure Oxidizer Pump "Torque"
TROP1 : Low Pressure Oxidizer Pump Pump Torque Characteristics
TRFP2 : High Pressure Fuel Pump Pump Torque Characteristics
TRFP1 : Low Pressure Fuel Pump Pump Torque Characteristics
TPOP2 : High Pressure Oxidizer Pump Pump Pressure Rise Characteristics
TPOP1 : Low Pressure Oxidizer Pump
TPFP2 : High Pressure Fuel Pump Pump Pressure Rise Characteristics
TPFP1 : Low Pressure Fuel Pump Pump Pressure Rise Characteristics
ROP1 : Low Pressure Oxidizer Pump
RFP2 : High Pressure Fuel Pump
RFP1 : Low Pressure Fuel Pump
POD2 : High Pressure Oxidizer Pump Total Pressure
POD1 : Low Pressure Oxidizer Pump Total Pressure
PFD2 : High Pressure Fuel Pump Total Pressure
PFD1 : Low Pressure Fuel Pump Discharge Total pressure
FOP2 : High Pressure Oxidizer Pump Flow variable
FPD1 : Low Pressure Oxidizer Pump Flow variable
FFP2 : High Pressure Fuel Pump Flow Variable
FFPl : Low Pressure Fuel Pump Flow variable

---------------------------------------------------------------------

Documentation for TURBINE section

1. Low Pressure Fuel Turbine
2. High Pressure Fuel Turbine
3. Low Pressure Oxidizer Turbine
4. High Pressure Oxidizer Turbine

Generic Equations

\[ R = B_i \times P \times Tr \quad ; \ Torque \quad \text{in-lb f} \]
\[ T_s = B_j \times S / (T^{(1/2)}) \quad ; \ Speed \ Parameter \]
\[ R = B_k \times (S)^2 \times Tr(F) \quad ; \ Weight \ Flowrate \quad \text{lb m/sec} \]

TPR : Pressure Ratio

TROT2 : Low Pressure Oxidizer Turbine Turbine Torque Characteristics
TROT1 : High Pressure Oxidizer Turbine Turbine Torque Characteristics
TRFT2 : High Pressure Fuel Turbine Turbine Torque Characteristics
TRFT1 : Low Pressure Fuel Turbine Turbine Torque Characteristics

DWOT2 : High Pressure Oxidizer Turbine Weight Flowrate
DWOT1 : Low Pressure Oxidizer Turbine Weight Flowrate
DWFT2 : High Pressure Fuel Turbine Weight Flowrate
DWFT1 : Low Pressure Fuel Turbine Weight Flowrate

UOT2 : High Pressure Oxidizer Turbine Speed Parameter
FOT1 : Low Pressure Oxidizer Turbine Flow Variable

UFT2 : High Pressure Fuel Turbine Speed Parameter
UFT1 : Low Pressure Fuel Turbine Speed Parameter

TQOT2 : High Pressure Oxidizer Turbine
TQOT1 : Low Pressure Oxidizer Turbine Torque

TQFT2 : High Pressure Fuel Turbine Torque
TQFT1 : Low Pressure Fuel Turbine Torque

---------------------------------------------------------------------

Documentation for TURBOPUMP section

1. Low Pressure Fuel Turbopump
2. High Pressure Fuel Turbopump
3. Low Pressure Oxidizer Turbopump
4. High Pressure Oxidizer Turbopump

Generic Equations

\[ S = (1/G) \times \text{Integrate} (Tq*temp(t),t) + SO \ ; \ Speed \]
Create Knowledge Base for VALVE section

1. Main Fuel Valve
2. Main Oxidizer Valve
3. Fuel Preburner Oxidizer Valve
4. Oxidizer Preburner Oxidizer Valve

Generic Equations

\[ A = Ab \times (T \times Th \times Thb) \]
Input-parameter: \( Ab \), \( T \), \( Th \), \( Thb \)
\( Ab \): Average of area, \( T \times Th \times Thb \) = Valve Effective Area

\[ Pit = Pi - RF \times DW^2 \times RHO \]
Input-parameter: \( Pi \), \( RF \), \( DW \), \( RHO \)
\( Pit \): Inlet Total Pressure, \( Pi \): Inlet Pressure
\( RF \): Resistance to Flow, \( DW \): Weight Flowrate, \( RHO \): Density

\[ Pis = Pit - DW^2 \times [(772.8) \times (Al^2) \times RHO] \]
Input-parameter: \( Pit \), \( DW \), \( Al \), \( RHO \)
\( Pis \): Inlet Total Pressure, \( Pit \): Inlet Pressure

\[ dP = B \times (RHO_b \times RHO) \times (DW \times (A/Ab)^2 - RF \times DW \times RHO) \]
Input-parameter: \( B \), \( RHO_b \), \( RHO \), \( DW \), \( A \), \( Ab \), \( RF \)

AOPV : Oxidizer Preburner Oxidizer Valve Area

AFPV : Fuel Preburner Oxidizer Valve Area

AMOV : Main Oxidizer Valve Area

AMFV : Main Fuel Valve Area

Create Knowledge Base for FLOW section

1. Fuel Preburner Fuel Flow
2. Fuel Preburner Oxidizer Flow
3. Oxidizer Preburner Fuel Flow
4. Oxidizer Preburner Oxidizer Flow

Generic Equations

\[ DWf = B_1 \int [P,pos - P - Be(DWo/(A/A'))^2 - Bm(DWi)^2]dt + [DWf]t=0 \]; Fuel Flow

\[ DWo = B_k \int [P,pos - P - Be(DWo/(A/A'))^2 - Bm(DWi)^2]dt + [DWo]t=0 \]; Oxidizer Flow

\[ Eo = B_n \int (DWo - DWi)dt + [Eo]t=0 \]; Variable in injector Priming function

\[ DWi = DWo \times [Eo] \]

WOPOI : Oxidizer Preburner Oxidizer Injector Weight Flowrate
Create Knowledge Base for COMBUSTER section

1. Fuel Preburner COMBUSTER
2. Oxidizer Preburner COMBUSTER

Generic Equations

\[ P = B_1 \int [DW1 + DW2 - Bj \cdot DW3],t + P_0 \quad \text{Total Pressure, Psia} \]

\[ F = DW2/(DW1+DW2) \quad \text{Fraction} \]

\[ T = Tqt(F) + B_k \cdot T_9 \quad \text{Temperature} \]

TTFOP : Oxidizer Preburner Hot Gas Temperature Function

TTFFP : Fuel Preburner Hot Gas Temperature Function

POPO : Pop when t=0

PPFO : Pfp when t=0

TOP : Oxidizer Preburner Combuster Temperature

TFP : Fuel Preburner Combuster Temperature

FOP : Oxidizer Preburner Flowrate Fraction

FFP : Fuel Preburner Flowrate Fraction

POP : Oxidizer Preburner Total Pressure

PFPO : Pfp when t=0

Create Knowledge Base for COOLING section

1. Main Chamber Regenerative Cooling Element
2. Nozzle Regenerative Cooling Element

Generic Equations

\[ P = B \int [DQw1 + DQw2 + H3 \cdot DW1 - H \cdot DW],t + P_0 \quad \text{Total Pressure} \]

\[ H = b \cdot T \quad \text{Specific Enthalpy} \]

\[ DW = B_1 \int [P - P_5 - (Bj/R_5) \cdot DW^2],t + DW_0 \quad \text{Weight Flowrate} \]

\[ R = B \int [DW1 - DW],t + RO \quad \text{Density} \]

\[ T_5 = B \cdot (P / R) \quad \text{Temperature} \]

\[ DQw1 = b \cdot [1.0 + 0.002 \cdot T_5] \cdot [T_{w15} - T_5] \cdot D_{Wmc}^{0.8} \quad \text{Heat Transfer Rate} \]

\[ DQw2 = B \cdot [1.0 + 0.002 \cdot T_5] \cdot [T_{w25} - T_5] \cdot D_{Wmc}^{0.8} \quad \text{Heat Transfer Rate} \]
DQtc = B * [Tc - Tw15] *[DWcn]^0.8
Tw1 = B * integrate ([DQtc - DW1,t] + Tw10 ; Hot Gas Wall Temperature
Tw2 = B * integrate ([-DQw2],t) + Tw20 ; Ambient thrust chamber Temperature
TW240 : Tw24 when t=0
TW140 : Tw14 when t=0
TW250 : Tw25 when t=0
TW150 : Tw15 when t=0
R40 : R4 when t=0
R50 : R5 when t=0
DW40 : DW4 when t=0
DWMC0 : DWmc when t=0
P40 : P4 when t=0
P50 : P5 when t=0
TW24 : Ambient Thrust Temperature
TW14 : Hot Gas temperature
TW25 : Ambient thrust chamber Temperature
TW15 : Hot Gas Temperature
DQTC4 : Nozzle Regenerative Heat Transfer Rate TC
DQTC5 : Main Chamber Cooling Element Heat Transfer Rate TC
DQW24 : Nozzle Regenerative Ambient Thrust Heat Transfer Rate
DQW14 : Nozzle Regenerative Hot Gas Heat Transfer Rate
DQW25 : Main Chamber Ambient Thrust Chamber Heat Transfer Rate
DQW15 : Main Chamber Hot Gas Heat Transfer Rate
T4 : Nozzle Regenerative Cooling Temperature
T5 : Main Chamber Heat Exchanger Temperature
R4 : Nozzle Regenerative Element Density
R5 : Main Chamber Element Density
DW4 : Nozzle Regenerative Cooling Weight Flowrate
DW5 : Main Chamber Heat Weight Flowrate
H4 : Nozzle Regenerative Cooling Element Specific Enthalpy
H5 : Main Chamber Heat Exchanger Specific Enthalpy
P4 : Nozzle Regenerative Exchanger Element Total Pressure
P5 : Main Chamber Heat Exchanger Total Pressure
Create Knowledge Base for FLOWMETER section

1. Fuel Flowmeter
2. Oxidizer Flowmeter

Generic Equations

\[ Q = \frac{P - (RF \times DW^2)}{Rb}; \text{Volumetric Flowrate, GPM} \]
\[ P = Pf - \frac{(RF \times DW^2)}{R}; \text{Total Pressure, Psia} \]
\[ T = T_i + B \times S; \text{Temperature} \]

TOD2 : Low pressure Oxidizer pump Flowmeter Temperature

TFP1 : Low pressure Fuel Pump Flowmeter Temperature

PFFM : Fuel Flowmeter Total Pressure

QF : Fuel Volumetric Flowrate

Create Knowledge Base for PREBURNER-SUPPLY-DUCT section

1. Preburner Fuel Supply Duct
2. Preburner Oxidizer Supply Duct

Generic Equations

\[ P = B \times \text{integrate}([DW], t) + P_0; \text{Total Pressure, Psia} \]
\[ T = B \times \frac{[P/R]}{T^2}; \text{Temperature} \]
\[ R = \frac{[DW_1 + DW_2]((DW_2/R) + B \times DW_3)}{\text{Density}} \]

PPOS0 : Ppos when t=0

P90 : P9 when t=0

R9 : Preburner Fuel Supply Element Density

T9 : Preburner Fuel Supply Element Temperature

PPOS : Preburner Oxidizer Supply Total Pressure

P9 : Preburner Fuel Supply Element Total Pressure

Create Knowledge Base for BOOST-STAGE section

1. High Pressure Oxidizer boostpump Flow variable
2. High Pressure Oxidizer boostpump rotational Torque
3. High Pressure Oxidizer Boostpump pressure
4. High Pressure Oxidizer Boostpump weight flowrate

Generic Equations

\[ F = B \times \left(\frac{DW}{S}\right)\text{;} \text{Flow variable, in}^3\]
\[ T = B \times (S^2) \times TT \times (F)\text{;} \text{Torque, in-lb} \]
\[ P3 = B \times (S^2) \times TT \times F \times P2\text{;} \text{Pressure} \]
\[ DW3 = B_1 \times \text{integrate} \left(\frac{(P3 - P - B_2^2)}{t} + DW_0\right)\text{;} \text{Weight Flowrate, lbm/sec} \]

DWOP30 : DWop3 when t=0

TPOP3 : high Pressure Oxidizer boostpump Pressure Rise Character
TROP3 : High Pressure Oxidizer Boostpump Torque

DWOP3 : High Pressure Oxidizer Boostpump Weight Flowrate

POD3 : High Pressure Oxidizer Boostpump Pressure

TOP3 : High Pressure Oxidizer Boostpump Torque

FOP3 : High Pressure Oxidizer boostpump Flow variable

Create Knowledge Base for BOOST-STAGE section

1. High Pressure Oxidizer boostpump Flow variable
2. High Pressure Oxidizer boostpump rotational Torque
3. High Pressure Oxidizer Boostpump pressure
4. High Pressure Oxidizer Boostpump weight flowrate

Generic Equations

\[ F = B \times \left( \frac{DW}{S} \right) \]; Flow variable, in^3

\[ T = B \times \left( S^2 \times TT \times (F) \right) \]; Torque, in-lb

\[ P3 = B \times \left( S^2 \times TT \times (F + P2) \right) \]; Pressure

\[ DW3 = B1 \times \int \left[ (P3 - P - B2^2), t \right] + DWO \]; Weight Flowrate, lbm/sec

Create Knowledge Base for MAIN-THRUST-CHAMBER section

1. Combustor Injector end static Total Pressure
2. Main thrust Chamber Total Pressure

Generic Equations

\[ P = b \times P \]; Combustor Injection Total pressure, Psia

\[ P = B1 \times \int \left[ (B2 \times DW1 + DW2 - DW3), t \right] + P0 \]; Main Thrust Chamber Total Pressure, Psia

Create Knowledge Base for NOZZLE-BYPASS section

PC : Main thrust chamber Total Pressure

PCIES : Combustor Injector end static Total Pressure
1. Nozzle Heat exchanger bypass Weight Flowrate
2. Coolant Control valve

Generic Equations

\[ DW = B_1 \cdot \int (P - P_9 - B_2 \cdot \frac{DW_1}{(A/Ab)^2}, t) + DW_0; \text{ Nozzle Heate weight flowrate, lbm/sec} \]

\[ DW = DW_1; \text{ Coolant Control Valve, lbm/sec} \]

DWFNBP0 : DWFnbp when t=0

DWCCV : Coolant Control Valve

DWFNBP : Nozzle Heat Exchanger bypass weight flowrate
Example for the Robot Manipulators

Example steps to solve the end position.

1. Define rules. \( \sin^2 \theta_i = 1 - \cos^2 \theta_i \)

   \(SS_i = \sin \theta_i\) \(i\) is from 1 to 6

   \(CC_i = \cos \theta_i\) \(i\) is from 1 to 6

2. Define matrices \(A_1, A_2, A_3, A_4, A_5, A_6, T_6\).

3. \(T_6 = A_1 A_2 A_3 A_4 A_5 A_6\)

   \(T_6\) is the end position and \(A_i\) is the matrix for each joint.

   \(T_6\) is the product of \(A_1 \ldots A_6\)

4. \(A_1^{-1} T_6 = A_2 A_3 A_4 A_5 A_6\)

   \(A_1^{-1} A_2^{-1} T_6 = A_3 A_4 A_5 A_6\)

   \(A_1^{-1} A_2^{-1} \ldots A_5^{-1} T_6 = A_6\)

5. Compare the above equations and solve

   \[
   T_6 = \begin{bmatrix}
   n_x & o_x & a_x & p_x \\
   n_y & o_y & a_y & p_y \\
   n_z & o_z & a_z & p_z \\
   0 & 0 & 0 & 1
   \end{bmatrix}
   \]

   \[
   A_6 = \begin{bmatrix}
   CC_6 & -SS_6 & 0 & 0 \\
   SS_6 & CC_6 & 0 & 0 \\
   0 & 0 & 1 & 0 \\
   0 & 0 & 0 & 1
   \end{bmatrix}
   \]
6. The end position is a product of the $A$ matrices

\[ T_6 = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6 \]

7. Matrix inverse, matrix multiplication

\[ T_{16} = A_1^{-1} T_6 = A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6 \]
\[ T_{26} = A_1^{-1} A_2^{-1} T_6 = A_3 \cdot A_4 \cdot A_5 \cdot A_6 \]
\[ T_{36} = A_1^{-1} A_2^{-1} A_3^{-1} T_6 = A_4 \cdot A_5 \cdot A_6 \]
\[ T_{46} = A_1^{-1} A_2^{-1} A_3^{-1} A_4^{-1} T_6 = A_5 \cdot A_6 \]
\[ T_{56} = A_1^{-1} A_2^{-1} A_3^{-1} A_4^{-1} A_5^{-1} T_6 = A_6 \]

8. Evaluating equations to obtain the end position

\[ n_x, n_y, n_z, o_x, o_y, o_z, a_x, a_y, a_z, p_x, p_y, p_z \]
A1: demo demo-stanford.out.1

(C5) BATCH("A1:DEMO-DEMO-STANFORD.DEMO");

(C6) tellsimp(ss1^2,1-cc1^2)$

(C7) tellsimp(ss2^2,1-cc2^2)$

(C8) tellsimp(ss3^2,1-cc3^2)$

(C9) tellsimp(ss4^2,1-cc4^2)$

(C10) tellsimp(ss5^2,1-cc5^2)$

(C11) tellsimp(ss6^2,1-cc6^2)$

(C12) A1:MATRIX([CC1,0,-SS1,0],[SS1,0,CC1,0],[0,-1,0,0],[0,0,0,1]);
    [ CC1  0  - SS1  0 ]
    [ ]
    [ SS1  0  CC1  0 ]
    [ ]
    [ 0  -1  0  0 ]
    [ ]
    [ 0  0  0  1 ]

(D12)

(C13) A2:MATRIX([CC2,0,SS2,0],[SS2,0,-CC2,0],[0,1,0,DD2],[0,0,0,1]);
    [ CC2  0  SS2  0 ]
    [ ]
    [ SS2  0  -CC2  0 ]
    [ ]
    [ 0  1  0  DD2 ]
    [ ]
    [ 0  0  0  1 ]

(D13)

(C14) A3:MATRIX([1,0,0,0],[0,1,0,0],[0,0,1,DD3],[0,0,0,1]);
    [ 1  0  0  0 ]
    [ ]
    [ 0  1  0  0 ]
    [ ]
    [ 0  0  1  DD3 ]
    [ ]
    [ 0  0  0  1 ]

(D14)

(C15) A4:MATRIX([CC4,0,-SS4,0],[SS4,0,CC4,0],[0,-1,0,0],[0,0,0,1]);
    [ CC4  0  - SS4  0 ]
    [ ]
    [ SS4  0  CC4  0 ]
    [ ]
    [ 0  -1  0  0 ]
    [ ]
    [ 0  0  0  1 ]

(D15)

(C16) A5:MATRIX([CC5,0,SS5,0],[SS5,0,-CC5,0],[0,1,0,0],[0,0,0,1]);
    [ CC5  0  SS5  0 ]
    [ ]
    [ SS5  0  -CC5  0 ]
    [ ]
(D16) [ 0 1 0 0 ]
[ ]
[ ]
[ 0 0 0 1 ]

(C17) A6: MATRIX([CC6, SS6, SS6, CC6], [0, 0, 0, 0], [0, 0, 0, 1]);
[ CC6 - SS6 0 0 ]
[ ]
[ SS6 CC6 0 0 ]
[ ]
[ 0 0 1 0 ]
[ ]
[ 0 0 0 1 ]

(C18) T6: MATRIX([Nx, Ox, Ax, Px], [Ny, Oy, Ay, Py], [Nz, Oz, Az, Pz], [0, 0, 0, 1]);
[ NX OX AX PX ]
[ ]
[ NY OY AY PY ]
[ ]
[ NZ OZ AZ PZ ]
[ ]
[ 0 0 0 1 ]

(C19) A11: INVERT(A1);
A1: demo-stanford.out 1
A1: cl-macysma>share>invert.bin.14 being loaded.
[ CC1 SS1 0 0 ]
[ ]
[ 0 0 -1 0 ]
[ ]
[ - SS1 CC1 0 0 ]
[ ]
[ 0 0 0 1 ]

(C20) A11: RATSIMP(A11);
[ CC1 SS1 0 0 ]
[ ]
[ 0 0 -1 0 ]
[ ]
[ - SS1 CC1 0 0 ]
[ ]
[ 0 0 0 1 ]

[ NY SS1 + CC1 NX OY SS1 + CC1 OX AY SS1 + AX CC1 PY SS1 + CC1 PX ]
[ ]
[ - NZ - OZ - AZ - PZ ]
[ ]
[ CC1 NY - NX SS1 CC1 OY - OX SS1 AY CC1 - AX SS1 CC1 PY - PX SS1 ]
[ ]
[ 0 0 0 1 ]

[ CC2 (CC4 CC5 CC6 - SS4 SS6) - CC6 SS2 SS5 CC2 (- CC4 CC5 SS6 - CC6 SS4) + SS2 SS5 SS6 CC2 CC4 SS5 + CC5 SS2 DD3 SS2 ]
[ ]
[ SS2 (CC4 CC5 CC6 - SS4 SS6) + CC2 CC6 SS5 SS2 (- CC4 CC5 SS6 - CC6 SS4) - CC2 SS5 SS6 CC4 SS2 SS5 - CC2 CC5 - CC2 DD3 ]
(D22) [ ]
\[\begin{align*}
(C23) & \text{a21:invert(a2);} \\
& \begin{bmatrix}
CC2 & SS2 & 0 & 0 \\
0 & 0 & 1 - CC2 & DD2 - (1 - CC2) DD2 \\
SS2 - CC2 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\end{align*}
\]

\[\begin{align*}
(D23) & \text{a21:ratsimp(a2);} \\
& \begin{bmatrix}
CC2 & SS2 & 0 & 0 \\
0 & 0 & 1 - DD2 \\
SS2 - CC2 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\end{align*}
\]

\[\begin{align*}
(C24) & \text{a21:ratsimp(a2);} \\
& \begin{bmatrix}
CC2 & SS2 & 0 & 0 \\
0 & 0 & 1 - DD2 \\
SS2 - CC2 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\end{align*}
\]

\[\begin{align*}
(C25) & \text{a216:a21.a11.t6;} \\
& \begin{bmatrix}
CC2 \ (NY \ SS1 + CC1 \ NX) - NZ \ SS2 & CC2 \ (OY \ SS1 + CC1 \ OX) - OZ \ SS2 & CC2 \ (AY \ SS1 + AX \ CC1) - AZ \ SS2 & CC2 \ (PY \ SS1 + CC1 \ PX) - PZ \ SS2 \\
CC1 \ NY - NX \ SS1 & CC1 \ OY - OX \ SS1 & AY \ CC1 + AX \ SS1 & - PX \ SS1 + CC1 \ PY - DD2 \\
(Y \ SS1 + CC1 \ NX) \ SS2 + CC2 \ NX & (OY \ SS1 + CC1 \ OX) \ SS2 + CC2 \ OZ & (AY \ SS1 + AX \ CC1) \ SS2 + AZ \ CC2 & (PY \ SS1 + CC1 \ PX) \ SS2 + CC2 \ PZ \\
0 & 0 & 0 & 1
\end{bmatrix}
\end{align*}
\]

\[\begin{align*}
(D26) & \text{t26:a3.a4.a5.a6;} \\
& \begin{bmatrix}
CC4 & CC5 & CC6 - SS4 \ SS6 & CC4 \ CC5 \ SS6 & CC4 \ CC6 - CC5 \ SS6 & CC4 \ SS5 & 0 \\
CC4 \ SS6 + CC5 \ CC6 \ SS4 & CC4 \ CC6 - CC5 \ SS4 \ SS6 & SS4 \ SS5 & 0 \\
- CC6 \ SS5 & SS5 \ SS6 & CC5 & DD3 \\
0 & 0 & 0 & 1
\end{bmatrix}
\end{align*}
\]

\[\begin{align*}
(C27) & \text{a3l:invert(a3);} \\
& \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\end{align*}
\]

\[\begin{align*}
(D27) & \text{a3l:ratsimp(a3);} \\
& \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\end{align*}
\]

\[\begin{align*}
(C28) & \text{a3l:ratsimp(a3);} \\
& \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\end{align*}
\]

\[\begin{align*}
(D28) & \text{a3l:ratsimp(a3);} \\
& \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\end{align*}
\]
A1: demo>demo-stanford.out.1

(C29) \( a3i6: a3i. sa2i. sa1i. t6; \)

(D29)

\[
\begin{align*}
\text{CC2} & (NY SS1 + CC1 NX) - NZ SS2 \quad \text{CC2} (OY SS1 + C1 CX) - OZ SS2 \quad \text{CC2} (AY SS1 + AX CC1) - AZ SS2 \\
\text{CC2} (PY SS1 + CC1 PX) - PZ SS2 \\
\text{CC1} NY - NX SS1 \quad \text{CC1 OY - OX SS1} \quad \text{AY CC1 - AX SS1} \quad - PX SS1 + CC1 PY - DD2 \\
\text{NY SS1 + CC1 NX} SS2 + CC2 NZ & (OY SS1 + CC1 OX) SS2 + CC2 OZ \quad \text{(AY SS1 + AX CC1)} SS2 + AZ CC2 \quad (PY SS1 + CC1 PX) SS2 + CC2 PZ - DD3 \\
\end{align*}
\]

(C30) \( t36: a4. a5. a6; \)

\[
\begin{align*}
\text{CC4 CC5} CC6 - SS4 SS6 - CC4 CC5 SS6 - CC6 SS4 & CC4 SS5 0 \\
\text{CC4} SS6 + CC5 CC6 SS4 & \text{CC4 CC6 - CC5 SS4 SS6} SS4 SS5 0 \\
\text{SS4} SS5 & SS5 SS6 CC5 0 \\
\end{align*}
\]

(D30)

(C31) \( a4i: invert(a4); \)

\[
\begin{align*}
\text{CC4} SS4 0 0 & 0 \\
0 0 -1 0 & \\
-SS4 CC4 0 0 & \\
0 0 0 1 & \\
\end{align*}
\]

(D31)

(C32) \( sa4i: ratsimp(a4i); \)

\[
\begin{align*}
\text{CC4} SS4 0 0 & 0 \\
0 0 -1 0 & \\
-SS4 CC4 0 0 & \\
0 0 0 1 & \\
\end{align*}
\]

(D32)

(C33) \( a4i6: sa4i. sa3i. sa2i. sa1i. t6; \)

\[
\begin{align*}
\text{(CC1 NY - NX SS1) SS4} + CC4 (CC2 (NY SS1 + CC1 NX) - NZ SS2) & \text{(CC4 CC2 (NY SS1 + CC1 NX) - NZ SS4)} \\
\text{CC2 (NY SS1 + CC1 NX) SS2} & \text{CC2 (NY SS1 + CC1 OX) - OZ SS2} \\
\text{CC1 OY - OX SS1} SS4 + CC4 (CC2 (OY SS1 + CC1 OX) - OZ SS2) & \\
\text{CC4 (CC1 OY - OX SS1) - (CC2 (OY SS1 + CC1 OX) - OZ SS2)} SS4 \\
\end{align*}
\]

(D33) \( \text{Col 1} = \)

\[
\begin{align*}
\text{CC4 (CC1 NY - NX SS1)} - (CC2 (NY SS1 + CC1 NX) - NZ SS2) SS4 & \\
\text{0} & \\
\end{align*}
\]

Col 2 =

\[
\begin{align*}
\text{CC4 (CC1 OY - OX SS1)} & - (CC2 (OY SS1 + CC1 OX) - OZ SS2) SS4 \\
\end{align*}
\]
Col 3 =

[ (AY CC1 - AX SS1) SS4 + CC4 (CC2 (AY SS1 + AX CC1) - AZ SS2) ]
[ ]
[ - (AY SS1 + AX CC1) SS2 - AZ CC2 ]
[ ]
[ CC4 (AY CC1 - AX SS1) - (CC2 (AY SS1 + AX CC1) - AZ SS2) SS4 ]
[ ]
[ D ]
[ ]
[ (- PX SS1 + CC1 PY - DD2) SS4 + CC4 (CC2 (PY SS1 + CC1 PX) - PZ SS2) ]
[ ]
[ - (PY SS1 + CC1 PX) SS2 - CC2 PZ + DD3 ]
[ ]
[ CC4 (- PX SS1 + CC1 PY - DD2) - (CC2 (PY SS1 + CC1 PX) - PZ SS2) SS4 ]
[ ]
[ 1 ]

(C34) t46:a5.a6;

[ CC5 CC6 - CC5 SS6 SS5 0 ]
[ ]
[ CC6 SS5 - SS5 SS6 - CC5 0 ]
[ ]
[ SS6 CC6 0 0 ]
[ ]
[ 0 0 0 1 ]

(D34)

(C35) a5i:invert(a5);

[ CC5 SS5 0 0 ]
[ ]
[ 0 0 1 0 ]
[ ]
[ SS5 - CC5 0 0 ]
[ ]
[ 0 0 0 1 ]

(D35)

(C36) sa5i:ratsimp(a5i);

[ CC5 SS5 0 0 ]
[ ]
[ 0 0 1 0 ]
[ ]
[ SS5 - CC5 0 0 ]
[ ]
[ 0 0 0 1 ]

(D36)

(C37) a5i6:sa51.sa41.sa31.sa21.sali.t6f

(D38) col(a5i6,1);

[ - (NY SS1 + CC1 NX) SS2 - CC2 NZ) SS5 + CC5 ((CC1 NY - NX SS1) SS4 + CC4 (CC2 (NY SS1 + CC1 NX) - NZ SS2)) ]
[ ]
[ CC4 (CC1 NY - NX SS1) - (CC2 (NY SS1 + CC1 NX) - NZ SS2) SS4 ]
[ ]
[ ((CC1 NY - NX SS1) SS4 + CC4 (CC2 (NY SS1 + CC1 NX) - NZ SS2)) SS5 - CC5 (- (NY SS1 + CC1 NX) SS2 - CC2 NZ) ]
[ ]
[ 0 ]
(C39) col(a516,2);  
[ (- (OY SS1 + CC1 OX) SS2 - CC2 OZ) SS5 + CC5 ((CC1 OY - OX SS1) SS4 + CC4 (CC2 (OY SS1 + CC1 OX) - OZ SS2)) ]  
[ ]  
(C40) col(a516,3);  
[ (- (AY SS1 + AX CC1) SS2 - AZ CC2) SS5 + CC5 ((AY CC1 - AX SS1) SS4 + CC4 (CC2 (AY SS1 + AX CC1) - AZ SS2)) ]  
[ ]  
(C41) col(a516,4);  
[ (- (PY SS1 + CC1 PX) SS2 - CC2 PZ + DD3) SS5 + CC5 ((- PX SS1 + CC1 PY - DD2) SS4 + CC4 (CC2 (PY SS1 + CC1 PX) - PZ SS2)) ]  
[ ]  
(C42) t56:a6;  
[ ]  
(C43) CLOSEFILE();
REFERENCES


AUTOMATIC MATHEMATICAL MODELING FOR
REAL TIME SIMULATION PROGRAM

By Caroline Wang and Steve Purinton

The information in this report has been reviewed for technical content. Review
of any information concerning Department of Defense or nuclear energy activities or
programs has been made by the MSFC Security Classification Officer. This report,
in its entirety, has been determined to be unclassified.

WILLIAM B. CHUBB
Director, Information and Electronic
Systems Laboratory