CADAC PRIMER

… building your own simulation

1. Computer Aided Design of Aerospace Concepts
2. Run GHAME3 test case to get started
3. Look at console display and plot output
4. Modify INPUT.ASC file
5. Define output in HEAD.ASC file
6. Plot output in 3 Dimensions
7. Modify Aerodynamic Module A1
8. Develop your own Module
9. Module assignments and sequencing
10. CADX3.FOR executes integration of state variables
11. Building HEAD.ASC in four steps
12. Building CADIN.ASC
13. Build your own SSTO simulation
14. Your SSTO output should look like this
15. Debugging aids for MODULE.FOR

1. Computer Aided Design of Aerospace Concepts

… simulating the flight dynamics of aerospace vehicles

History

1966 Litton Industry
1978 CADAC-Air Force
1998 CADAC Version 3.0
2000 CADAC Version 3.1

Run-Time Capabilities

Staging
Special functions
Random distributions
Multiple runs
Monte Carlo runs
Re-Initialization of runs
Automated envelope generation

CADAC Family of Simulations

CADAC 2 - 3 DoF, spherical earth, **GHAME3, ROCKET3**
CADAC 3 - 5 DoF, flat earth, air-to-ground, **CRUISE5**
CADAC 4 - 5 DoF, flat earth, air-to-air, **AIM5, SRAAM5**
CADAC 5 - 5 DoF, spherical earth
CADAC 6 - 6 DoF, flat earth, missiles **SRAAM6**
CADAC 7 - 6 DoF, flat earth, aircraft **FALCON6**
CADAC 8 - 6 DoF, elliptical earth, hypersonic **GHAME6**

Plotting and Analysis of Output

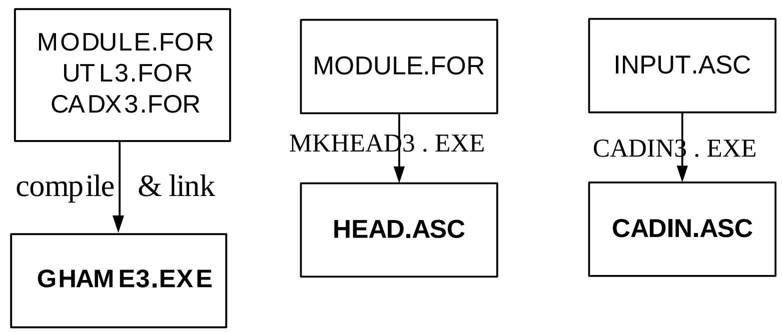
KPLOT
2-DIM, Strip Charts, 3-DIM, Globe,
Histograms, Bi-variate distributions
SWEEP
Launch envelopes
Footprints
MCAP
Monte Carlo analysis
QPRINT
Listing of variables

Compatibility

FORTRAN 77 with extensions
Compact Visual Fortran is preferred compiler
Platforms:
- All Windows platforms
- Adaptable to Silicon Graphics computers

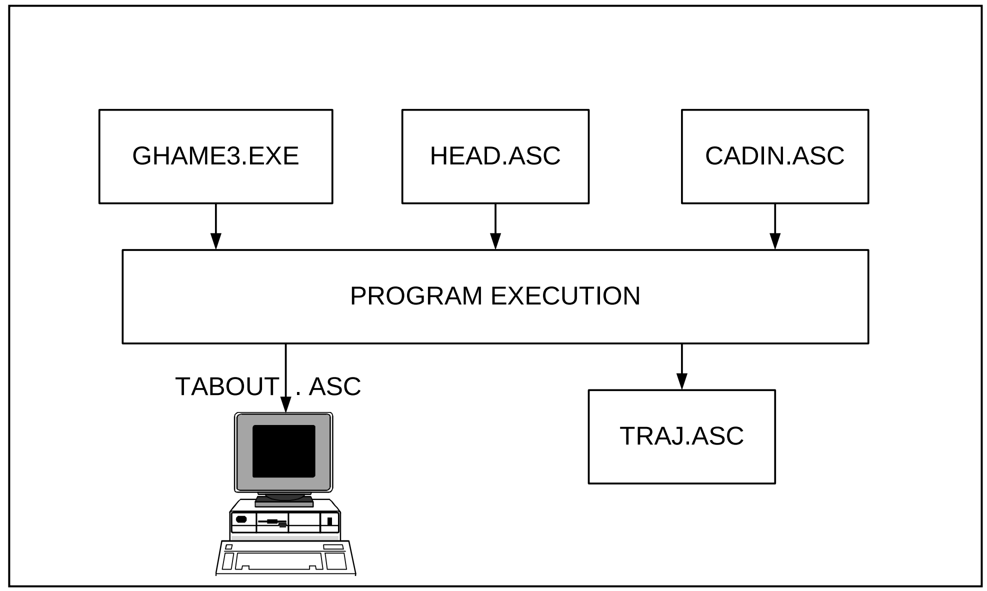
2. Run GHAME3 test case to get started

1. Build GHAME3.EXE
2. Place HEAD.ASC in project directory or build from MODULE.FOR
3. Build CADIN.ASC
4. Run GHAME3.EXE
5. Look at Screen output
6. Plot from TRAJ.ASC file



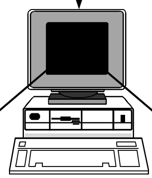
File Names

MODULE.FOR	Vehicle subsystems (modules)
UTL3.FOR	Utility matrix routines (V.3)
CADX3.FOR	CADAC executive (V.3)
INPUT.ASC	Free format input file
CADIN.ASC	Fixed format Fortran input
HEAD.ASC	Defines output to TRAJ.ASC
TRAJ.ASC	Output file for plotting
TABOUT.ASC	Output scrolled to screen
GHAME3.EXE	Compiled Fortran program
MKHEAD3.EXE	Utility program (V.3)
CADIN3.EXE	Utility program (V.3)



3. Look at console display and plot output

TABOUT.ASC

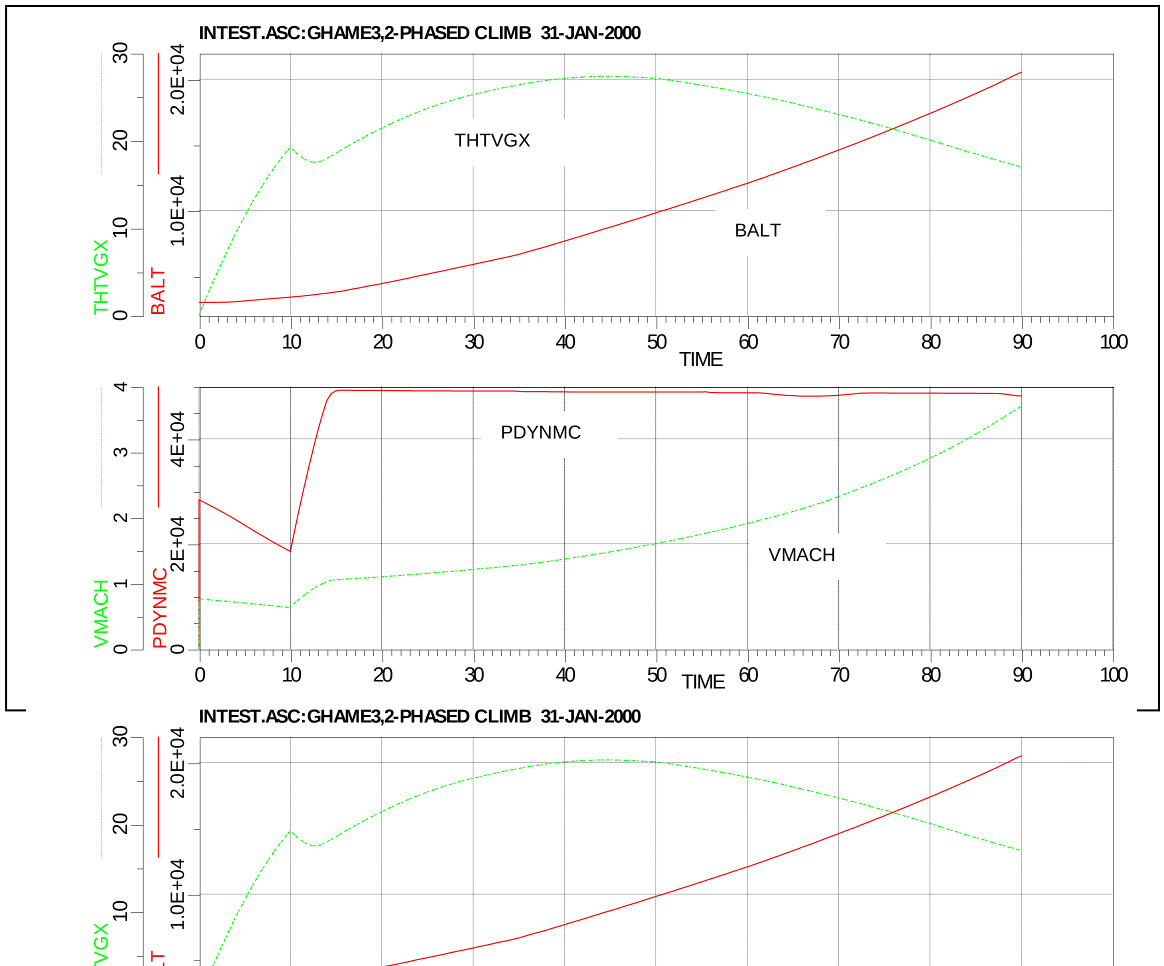


INTEST.ASC:GHAME3,2-PHASED CLIMB 31-JAN

TIME	BALT	VMACH	PDYNMC	THTVGX
0.0	3000.	0.00	0.	0.0
0.0	3000.	0.76	28401.	0.0
2.0	3013.	0.74	26588.	4.9
4.0	3057.	0.71	24632.	9.3
6.0	3177.	0.68	22509.	13.2
8.0	3294.	0.66	20467.	16.5
10.0	3408.	0.63	18534.	19.3
12.0	3542.	0.87	34435.	17.7
14.0	3709.	1.03	47289.	17.9
16.0	3910.	1.06	49242.	19.2
TIME	BALT	VMACH	PDYNMC	THTVGX
18.0	4175.	1.08	49217.	20.3
20.0	4440.	1.10	49197.	21.4
22.0	4719.	1.12	49153.	22.4
24.0	5012.	1.14	49133.	23.3
26.0	5305.	1.16	49123.	24.0
28.0	5601.	1.19	49110.	24.7
30.0	5898.	1.21	49101.	25.2
32.0	6197.	1.24	49094.	25.7
34.0	6498.	1.26	49091.	26.2
36.0	6862.	1.29	48976.	26.5

TRAJ.ASC

KPLOT 2-DIM



4. Modify INPUT.ASC file

Module calling sequence

Stage at time > 10 s

Execute program

Fortran Stop

```
TITLE INPUT.ASC:GHAME3,2-PHASED CLIMB
MODULES
 G2   GEOPHYSICS
 A2   PROPULSION
 A1   AERODYNAMICS
 A3   FORCES
 D1   NEWTONS LAW
END
OPTMET = 1.0000    ! E Units of measure 1 = metric; 0 = English
! ** CONTROL **
ALPHAX = 7.0000    ! D Angle of attack - deg
PHIMVX = 0.00000   ! D Bank angle of maneuver plane wrt vertical - deg
! ** PROPULSION **
MPROP = INT(1)     ! D =0:Motor off,=1:throttle command,=2:autothrottle
THRTL = 0.20000    ! D Throttle, controlling the fuel-air ratio - ND
THRTLMAX = 2.0000  ! D Maximum throttle setting permissible - ND
! ** VEHICLE DATA **
SREF = 557.42      ! D Aerodynamic reference area - m^2
! *** LAUNCH COND **
BLON = 4.8770      ! I/G Vehicle longitude - rad
BLAT = 0.49620     ! I/G Vehicle latitude - rad
BALT = 3000.0      ! I/O Vehicle altitude = m
DVBE = 250.00      ! I/G Geographic speed - m/s
PSIVGX = 90.000    ! I Heading angle from north - deg
THTVGX = 0.00000   ! I Flight path angle from horizontal - deg
! ** EXECUTIVE **
PPP = 0.50000      ! E Time interval writing to TRAJ.BIN or TRAJ.ASC - sec
CPP = 2.0000       ! E Time interval writing to Screen or TABOUT.ASC - sec
DER = 0.01000      ! E Integration interval - sec
! ##1 STAGE ##
IF TIME > 10.000   ! E  Trajectory time - s
MPROP = INT(2)     ! D =0:Motor off,=1:throttle command,=2:autothrottle
QHOLD = 50000.     ! D Dynamic pressure hold command - Pa
ALPHAX = 2.500     ! D Angle of attack - deg
! ##2 STAGE ##
IF TIME > 50.000   ! E  Trajectory time - s
ALPHAX = 2.4000    ! D Angle of attack - deg
! ##3 STAGE STOP ##
IF TIME > 90.00    ! E  Trajectory time - s
RUN
STOP
```

Change to:
ALPHAX = 3.0

5. Define output in HEAD.ASC file

Enables writing TRAJ.ASC

SCROLL to screen

Messages written to screen

Files written to local directory

Integer in third column specifies decimals

Written to Screen

Separator in first column

Comment lines from here on down

Written to TRAJ.ASC

```
SCROLL NOECHOIN NOINTMSG NOSTGMSG NORANVAR NOTRAJBIN TRAJASC NOSTATBIN NOSTATASC NOTABOUT
  1 2000    TIME
    1606    BALT          I Vehicle altitude = m
  1 1603    THTUGX        I Flight path angle from horiz. - deg
  2  206    VMACH         O Mach number of rocket
     207    PDYNMC        G Dynamic Pressure - Pa
*
* EXECUTIVE
      51    REARTH        E Radius of Earth - m
      52    CRAD          E Conversion from rad to deg = 57.298
      54    AGRAV
      58    WEII3         E Earth rotation - rad/sec
* ENVIRONMENT
     201    TEMPK         G Atmospheric temperature - degK
     202    PRESS         O Atmospheric pressure - Pa
     203    RHO           O Atmospheric density - kg/m^3
     204    VSOUND        G Sonic speed - m/sec
     205    GRAV          O Gravity acceleration - m/s^2
 *   206    VMACH         O Mach number of rocket - ND
 *   207    PDYNMC        G Dynamic pressure - Pa
              …
              …
              …
```

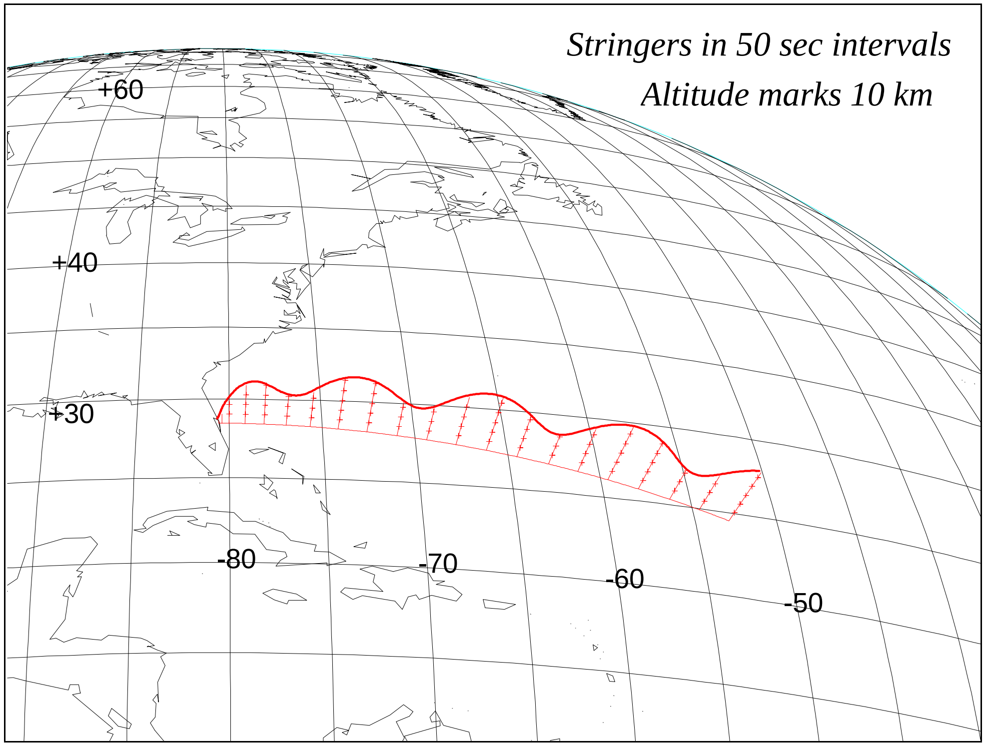
COMMON location

Variable name

No TABs Permitted
Turn-on ‘white space display’
CTRL+SHIFT+8 and remove any tabs(>>)

6. Plot output in 3 Dimensions

Use **KPLOT-GLOBE** and plot BLON, BLAT, BALT traces



Challenge: Modify INPUT.ASC by modulating angle of attack to duplicate this trajectory of 1000 sec duration

7. Modify Aerodynamic Module A1

Problem

Increase drag by a factor specified in INPUT.ASC

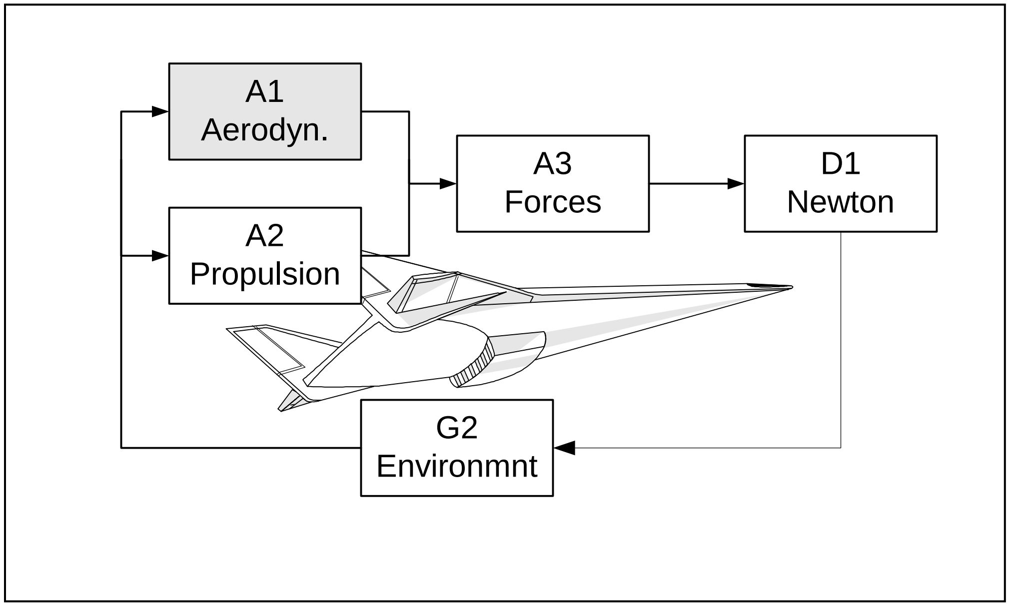
Modify A1 Module source code

Include FACTCD in HEAD.ASC

Insert FACTCD into INPUT.ASC

Build CADIN.ASC

Run GHAME3.EXE




```
C*********************************
      SUBROUTINE A1
C*********************************
C
      COMMON C(3510)
C
C*** INPUT DATA
C
      EQUIVALENCE (C(1209),FACTCD)
C
C*** OUTPUT TO OTHER MODULES
C
      EQUIVALENCE (C(1201),CD)
      EQUIVALENCE (C(1202),CL)
C
C CD = O Drag coefficient - ND
C CL = O Lift coefficient - ND
C
C
C*** CALCULATE THE DRAG FORCE
C
      CL=CLA0+CLA*ALPHAX
      CD=CL0+CKK*(CL-CL0)**2
      CD=CD*(1.+FACTCD)
C
      RETURN
      END
```

Add code

8. Develop your own Module

```
C*******************************************
      SUBROUTINE XXI
C*******************************************
      COMMON C(3510)
C*** INITIALIZATION
       EQUIVALENCE (C(1210),IX1)
C IX1 = I placeholder for table look-up - ND
       IX1=1
       RETURN
       END
C******************************************
      SUBROUTINE XX
C******************************************
      COMMON C(3510)
C***  INPUT DATA
      EQUIVALENCE (C(1203),ALPHAX)
C ALPHAX = D Angle of attack - deg
C*** INITIALIZATION
       EQUIVALENCE (C(1210),IX1)
C*** INPUT FROM EXECUTIVE
      EQUIVALENCE (C(0052),CRAD)
C*** INPUT FROM OTHER MODULES
      EQUIVALENCE (C(0206),VMACH)
C VMACH= O Mach number of rocket - ND
C*** OUTPUT TO OTHER MODULES
      EQUIVALENCE (C(1201),CD)
      EQUIVALENCE (C(1202),CL)
C CD = O Drag coefficient - ND
C CL = O Lift coefficient - ND
C*** DIAGNOSTICS
      EQUIVALENCE (C(1204),CD0)
      EQUIVALENCE (C(1205),CL0)
      EQUIVALENCE (C(1206),CKK)
      EQUIVALENCE (C(1207),CLA)
>>>>>>> CODE <<<<<<<<

      RETURN
      END
```

Initialization module XXI is called once
- Initializes variables
- Identifies state variables to be integrated

Module XX is called twice for every integration step (Euler predictor/corrector)
- Calculates the derivatives of the state variables
- Executes all other computations
- Calls utility subroutines MATyyy, VECyyy, TABLy,TABLPy
- Calls one lower level of subroutines XXyyyy
- Talks to other modules by EQUIVALENCEing to COMMON(3510)
  - INPUT DATA: D
  - INITIALIZATION: I
  - INPUT FROM EXECUTIVE: E
  - INPUT FROM OTHER MODULES
  - STATE VARIABLES: S
  - OUTPUT TO OTHER MODULES: O
  - DIAGNOSTICS: G

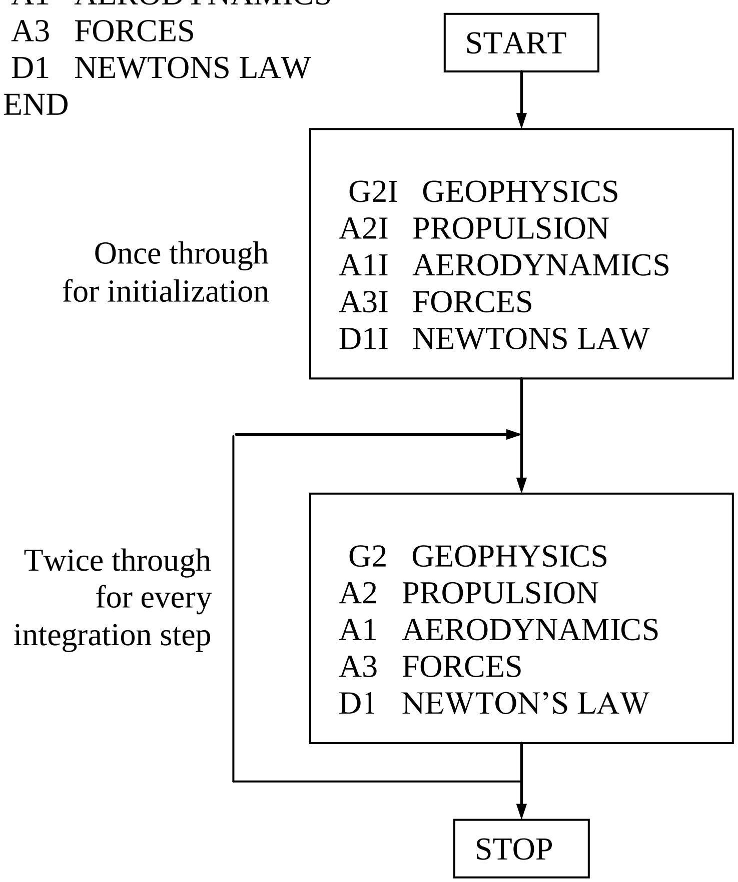
- Avoid:
  - labeled COMMON
  - CALLs to subroutines of other modules

9. Module assignments and sequencing

MODULE XX	MODULE NAME	COMMON
A1	Aero Coefficients	1200 - 1299
A2	Propulsion	1300 - 1399
A3	Forces & Moments	1400 - 1499
A4	Free	1500 - 1599
C1	Guidance	800 - 899
C2	Autopilot	900 - 999
C3	TVC	1000 - 1099
C4	Actuator	1100 - 1199
D1	Newton Eqs.	1600 - 1699
D2	Euler Eqs.	1700 - 1749
D4	Free	1900 - 1999
G1	Target	100-199
G2	Geophysics	200-299
G3	Kinematics	300-399
G4	Intercept	1750-1799
S1	Seeker	400-499
S2	Radar	500-599
S3	NAV Filter	600-699
S4	INS	700-799
SWEEP MODULES	Sweep Methodology	1800 - 1824
CADAC EXECUTIVE	Controlling Methodology	1 - 99 2000 - 2999
Unassigned Locations		3000 - 3510

MODULES Module call-sequence established in INPUT.ASC:

G2 ENVIRONMENT
A2 PROPULSION
A1 AERODYNAMICS
A3 FORCES
D1 NEWTONS LAW
END



10. CADX3.FOR executes integration

- Initialization Module stores derivative and state C-locations in IPL(100) and IPLV(100), respectively

- Integration subroutine AMRK gets derivative value from Module and predicts state value at next time step.

- AMRK corrects prediction by a second pass and returns the value of the state

```
C*****************************************************
      SUBROUTINE A2I
C*****************************************************
C
C*** INPUT FROM EXECUTIVE
C
      EQUIVALENCE (C(2561),NIP)
      EQUIVALENCE (C(2562),IPL(1))
      EQUIVALENCE (C(2867),IPLV(1))
C
C NIP = E Number of variables to integrate
C IPL(100) = E Start of derivative c-array locations
C IPLV(100) = E Start of state C-array locations
C
C*** Initialization of integration variable
C
      IPL(NIP)=1304
      IPLV(NIP)=1305
      NIP=NIP+1
C
      RETURN
      END
C*****************************************************
      SUBROUTINE A2
C*****************************************************
C
C*** STATE VARIABLES
C
      EQUIVALENCE (C(1304),FMASSED)
      EQUIVALENCE (C(1305),FMASSE)
C
C FMASSED = S Derivative of fuel mass expended - kg/sec
C FMASSE = S Fuel mass expended - kg
C
         IF(SPI.NE.0.) FMASSED=THRUST/(SPI*AGRAV)
         VMASS=VMASS0-FMASSE
C
      RETURN
```

11. Building HEAD.ASC in four steps

1. Merge Modules into MODULE.FOR
(Window PC use DOS COPY command)

MKHEAD3.EXE

2. Builds columns of C-Locations
Error checking

HEAD.ASC

DFHEAD3.EXE

3. Inserts variable definitions taken from MODULE.FOR

HEAD.ASC

4. Insert variables for scroll list and asterisks for variables to be written to TRAJ.ASC

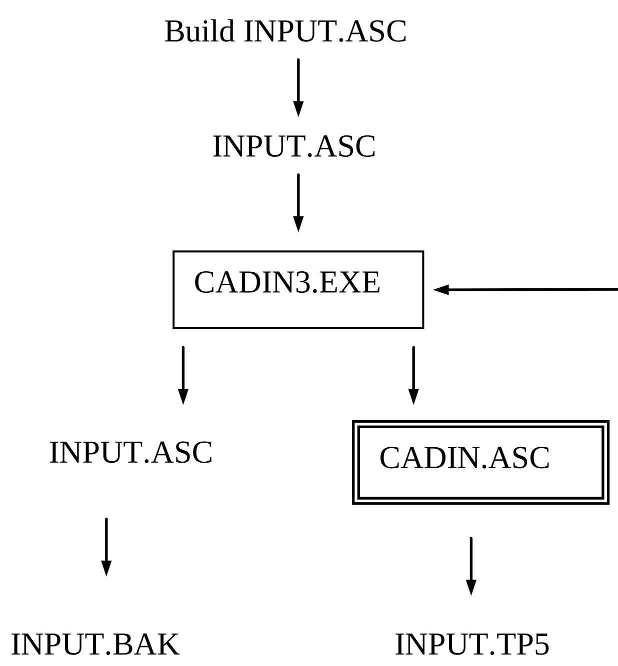
HEAD.ASC

```
C*******************************************
      SUBROUTINE A1
C*******************************************
C***  INPUT DATA
      EQUIVALENCE (C(1203),ALPHAX)
C ALPHAX = D Angle of attack - deg
C*** INITIALIZATION
       EQUIVALENCE (C(1210),IX1)
C*** INPUT FROM EXECUTIVE
      EQUIVALENCE (C(0052),CRAD)
C*** INPUT FROM OTHER MODULES
      EQUIVALENCE (C(0206),VMACH)
C VMACH= O Mach number of rocket - ND
C*** OUTPUT TO OTHER MODULES
      EQUIVALENCE (C(1201),CD)
      EQUIVALENCE (C(1202),CL)
C CD = O Drag coefficient - ND
C CL = O Lift coefficient - ND
C*** DIAGNOSTICS
      EQUIVALENCE (C(1204),CD0)
C CD0 = G Zero lift drag coefficient - ND
```

```
SCROLL NOECHOIN NOINTMSG NOSTGMSG NORANVAR NOTRAJBIN TRAJASC NOSTATBIN
                                          NOSTATASC NOTABOUT NOSWEEP
  1 2000    TIME
    1606    BALT         I Vehicle altitude = m
  1 1603    THTUGX       I Flight path angle from horizontal - deg
  2  206    VMACH        O Mach number of rocket
     207    PDYNMC       G Dynamic Pressure - Pa
*
* EXECUTIVE
      51    REARTH       E Radius of Earth - m
      52    CRAD         E Conversion from radians to degrees = 57.298
* ENVIRONMENT
     201    TEMPK        G Atmospheric temperature - degK
     202    PRESS        O Atmospheric pressure - Pa
     203    RHO          O Atmospheric density - kg/m^3
     204    VSOUND       G Sonic speed - m/sec
     205    GRAV         O Gravity acceleration - m/s^2
     206    VMACH        O Mach number of rocket - ND
```

Do not use TAB's

12. Building CADIN.ASC



INPUT.ASC

```
TITLE INPUT.ASC:GHAME3,Climb
MODULES
 G2 GEOPHYSICS
 A2 PROPULSION
 A1 AERODYNAMICS
 A3 FORCES
 D1 NEWTONS LAW
END
! *** Launch Conditions ***
PSIVGX = 90         ! I Heading angle from north - deg
THTVGX = 0          ! I Flight path angle from horizontal - deg
DVBE = 400          ! I/G Geographic speed - m/s
RUN
STOP
```

HEAD.ASC

```
SCROLL NOECHOIN NOINTMSG NOSTGMSG NORANVAR NOTRAJBIN TRAJASC NOSTATBIN

  1 2000    TIME
    1606    BALT         I Vehicle altitude = m
  1 1603    THTUGX       I Flight path angle from horizontal - deg
  2  206    VMACH        O Mach number of rocket
     207    PDYNMC       G Dynamic Pressure - Pa
*
* EXECUTIVE
      51    REARTH       E Radius of Earth - m
      52    CRAD         E Conversion from radians to degrees = 57.298
* NEWTON'S LAW
 *  1602    PSIVGX       I Heading angle from north - deg
 *  1603    THTVGX       I Flight path angle from horizontal - deg
 *  1613    DVBE         I/G Geographic speed - m/s
```

CADIN.ASC

```
INPUT.ASC:GHAME3,CLIMB
01 OUTPUT 2,3         0003
01 STAGE 2,3          0004
02 G2  ENVIRONMENT    0023
02 A2  PROPULSION     0003
02 A1  AERODYNAMICS   0002
02 A3  FORCES         0004
02 D1  NEWTONS LAW    0017
04*** LAUNCH CONDITIONS
03 DVBE               1613      400.00
03 PSIVGX             1602      90.000
03 THTVGX             1603      0.00000
06
13
```

13. Build your own SSTO simulation

A2 MODULE Propulsion

Liquid throttlable rocket motors
Thrust is acting parallel to the body x-axis.

Maximum Thrust = 1.51×10^6 N
Nozzle Exit Velocity = 4860 m/s
Launch Mass = 181,437 kg
Fuel Mass = 156194 kg
Vehicle Mass = 25243 kg (no fuel)

Test Case
Initial Conditions
Cape Canaveral, Altitude 12 km,
Geographic Speed 253 m/s, easterly direction
Control Sequence

t sec	α deg	*throttle*
< 200	22.93	.9
200 -400	5.73	.9
> 400	5.73	.5

A1 MODULE Aerodynamics

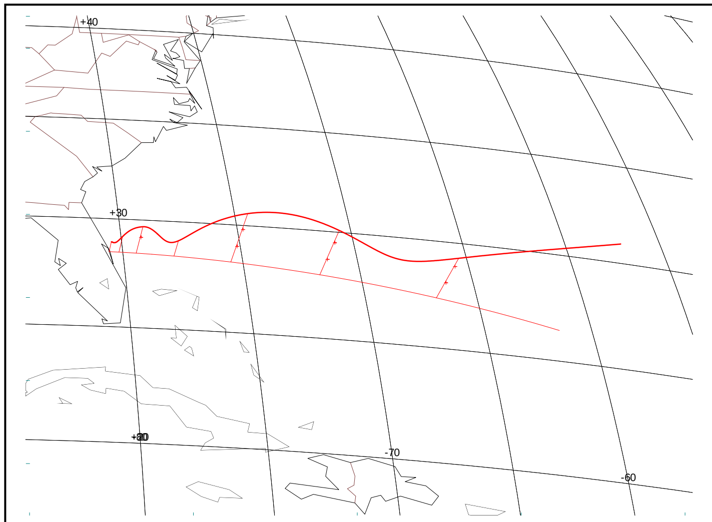
The aerodynamics are modeled by a symmetric drag polar of the form

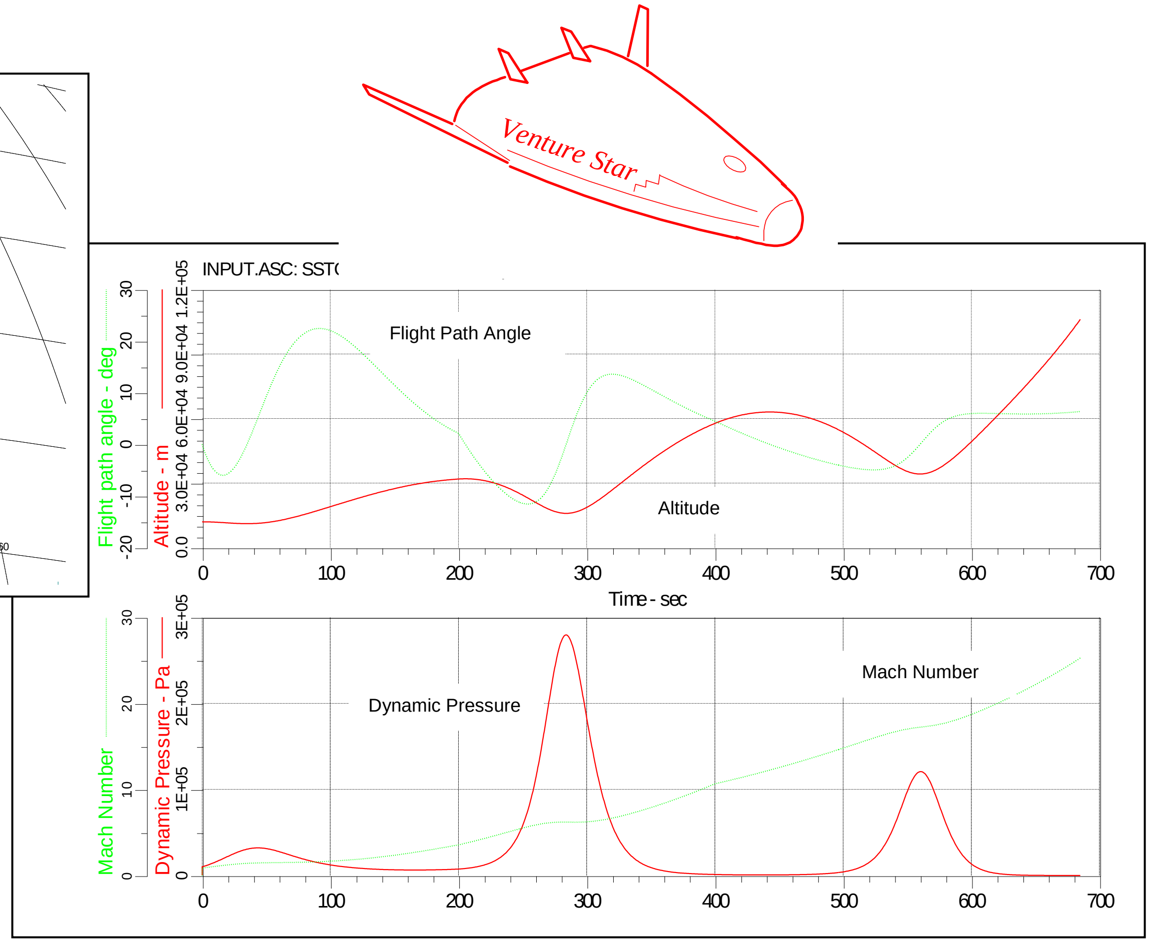
$$C_D = C_{D_0}(M) + k(M)C_{L_\alpha}^2(M)\alpha^2 = C_{D_0}(M) + \overline{C}_{L_{\alpha^2}}(M)\alpha^2$$

Reference Area = 102 m^2

Mach	C_{D_0}	$C_{L_\alpha}(rad^{-1})$	$\overline{C}_{L_{\alpha^2}}(rad^{-2})$
0.2	.0417	1.569	0.815
1.2	.0850	1.482	1.185
5.0	.0400	1.115	1.135
10.0	.0290	1.063	1.040
20.0	.0320	1.033	1.022

14. Your SSTO output should look like this





15. Debugging aids for MODULE.FOR

MODULE.FOR → **MKHEAD3.EXE** flags the following errors: → HEAD.ASC

Variables equivalenced to different C-locations → ‘letter’ code in HEAD.ASC
C-location equivalenced to several variables → ‘*’ code in HEAD.ASC
Mismatch between EQUIVALENCEd and DIMENSIONed arrays size
Overlap of arrayed C-locations

MODULE.FOR, HEAD.ASC → **EQMAP3.EXE** maps the usage of all EQUIVALENCEd variables IN → EQMAP.ASC

Subroutine where variable is defined
Subroutine where variable is used
Subroutine call where variable is in argument list

MODULE.FOR, GHAME3.EXE → **Compiler/RunTime Diagnostics** (Digital’s Visual Fortran 6.0) → Screen Output, Run-Time Out

To check for spelling errors during compile:
Project/Settings/Fortran/Category/Compilation Diagnostics: ∨ Un-Initialized Variables, ∨ Unused Variables
Project/Settings/Fortran/Category/Optimization: *Full Optimization
To check for array dimensioning errors during run-time:
Project/Settings/Fortran/Category/Run Time: ∨ General Traceback Information, ∨ Array & String Bounds