wRASP
Rocket Altitude Simulation Program for Windows

wRASP is a Windows program that simulates the flight of a rocket in one dimension (one degree-of-freedom).

Outline

1. Origin
2. Physics
3. Data Flow
4. Databases
5. DigiTrak
6. Software modules

1. Origin of wRASP

The original **RASP** was a BASIC program written by Harry Stine in 1979 and published in his ***Handbook of Model Rocketry***.

DOS **RASP** was converted to the C programming language by Mark Storin in 1989. New features were added by Kent Hoult in 1990 and Mark Johnson in 1990-94.

wRASP is an adaptation of that venerable RASP program for DOS that runs with a typical Windows graphical user interface (GUI). It was first made available to Compuserve’s Sport Rocketry forum in 1995.

wRASP v2.0 and up includes Larry Curcio's **DigiTrak**, and is based on his version 5.4. v2.0 also includes the NAR tracking algorithms, also originally available in BASIC.

2. Physics

Simulation programs typically define the level of sophistication of the program in "degrees of freedom" (DOF). These define how the airframe behaves in 3 dimensional space. The maximum degrees of freedom is six, three for movement (x,y,z) and three for rotation (roll,pitch,yaw). A full simulation is usually referred to as a "Six-DOF".

The simulation engine in wRASP is "raspsubs.c", derived directly from the main program in DOS RASP's "rasp.c". This engine has always been altitude only (specified as "x"), so they are both One-DOFs.

The simulation engine solves the differential equations for velocity and position by resolving the force summation on the rocket. The method is basic numerical methods for integrating acceleration and velocity:

$$a = \sum F/m \qquad \text{(from } F=ma\text{)}$$

$$v = \int a\,dt = a\,\Delta t + v_0$$

$$x = \int v\,dt = v\,\Delta t + x_0$$

where:
F = force in newtons
m = mass in kilograms
a = acceleration in meters/second^2
v = velocity in meters/second
x = distance in meters
dt = differential time in seconds
Δt = delta time in seconds = 0.001 second

The time step () was originally 0.1 second but was later changed to one millisecond (0.001 second). There are provisions for changing it but they are currently not implemented.

The simulation continually calculates these values and monitors for these events:

When altitude (x) reaches the top of the launch rod
Motor burn-out
Apogee
Motor delays
Ground impact

The force summation is the guts of these calculations:

$$\sum F = (th - drag)$$

where: th = thrust in Newtons
drag = drag force in Newtons

The thrust is determined by the motor(s)'s thrust-time curve and the time into the burn. The drag is based on:

$drag = c\,v^2$ (a form of $\frac{1}{2} \rho v^2$)

where: drag = drag force in Newtons (kg m/sec^2)
c = drag form factor
v = velocity in meters/sec

The drag form factor is defined as:

$$c = \frac{1}{2}\, r\ \pi\ C_D\ d^2 / 4$$

where: c = drag form factor in kilograms/meter
r = atmospheric constant in kilograms/meter^3
C_D = coefficient of drag
d = maximum body tube diameter in meters

The atmospheric constant is a function of the air density (ρ). The original RASP program used a constant value of:

$$r = 1.205\ kg/m^3$$

wRASP also allows the use of an atmospheric model to calculate the air density (ρ) at a specific altitude above mean sea level:

$$r = \rho\ 32.1725 / 2.204623 / 0.028317$$

where: ρ = air density in slugs/feet^3
rest = the constants to convert to kilograms/meter^3

Acceleration is then calculated:

```
a = (thrust - drag)/m – G           (upward motion)
a = drag/m – G                      (downward motion)
```

where: a = acceleration in meters/second^2
th = thrust in Newtons
drag = drag force in Newtons
m = mass in kilograms
G = acceleration of gravity in meters/second^2

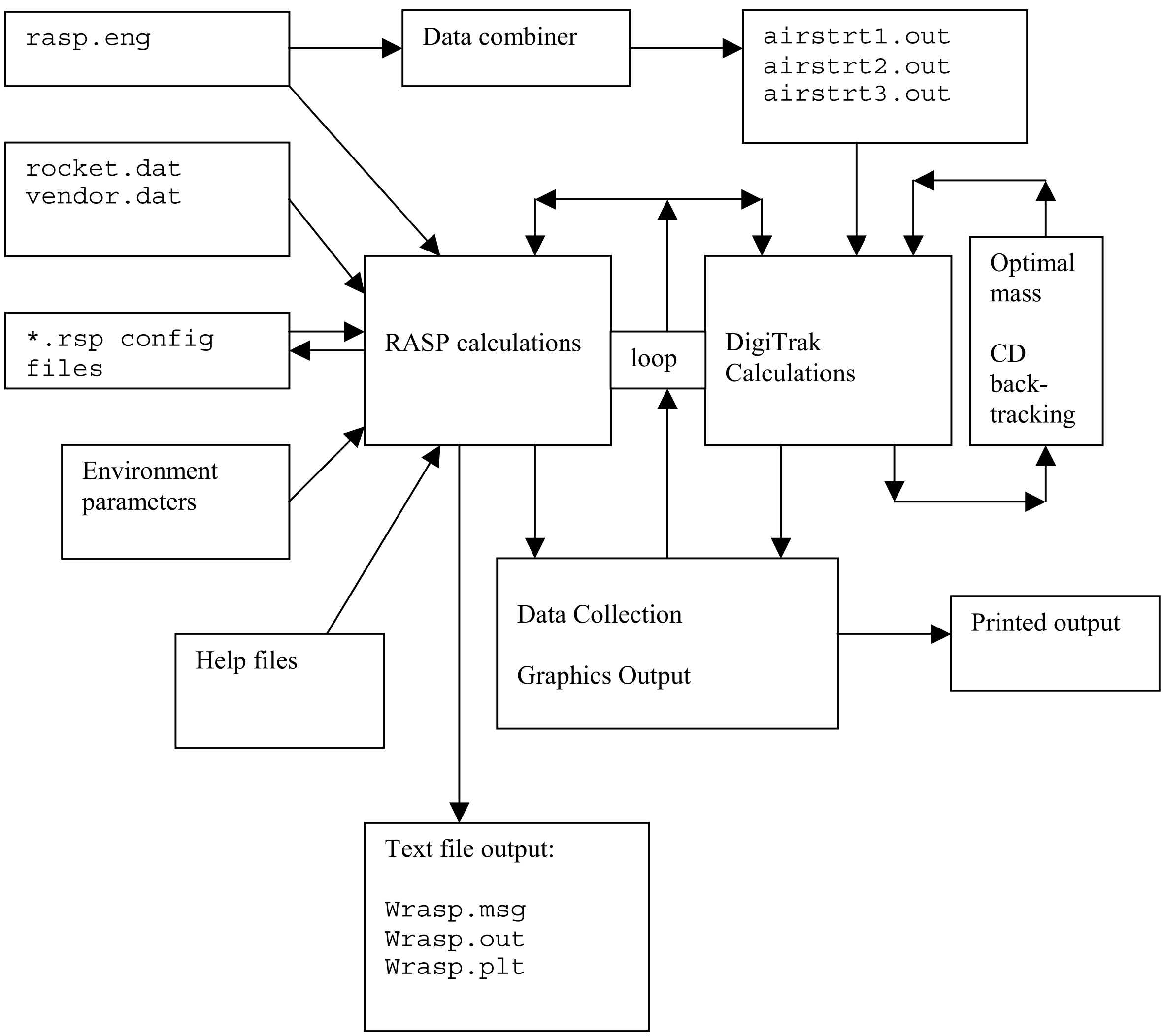
Mass varies with motor burn and staging. It is assumed that 100% of the propellant weight specified in the motor data file is consumed from the beginning to the end of the motor burn time. This can be done two ways:

m = total motor mass – (burn time)/(total burn time)*(propellant mass)

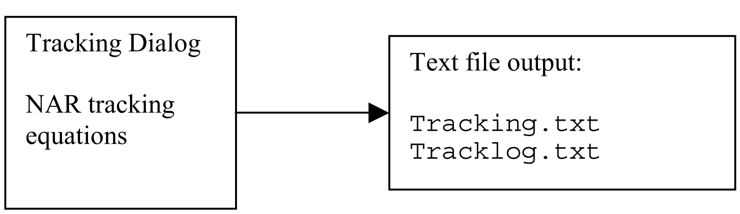
m = motor mass – (expended impulse)/(total impulse)*(propellant mass)

wRASP uses the first method – linear reduction of mass during the burn. The second is probably more accurate – this may be added as an option in the future.

3. Data Flow



rasp.eng
Data combiner
airstrt1.out
airstrt2.out
airstrt3.out
rocket.dat
vendor.dat
*.rsp config files
RASP calculations
loop
DigiTrak Calculations
Optimal mass
CD back-tracking
Environment parameters
Help files
Data Collection
Graphics Output
Printed output
Text file output:
Wrasp.msg
Wrasp.out
Wrasp.plt



Tracking Dialog
NAR tracking equations
Text file output:
Tracking.txt
Tracklog.txt

4. Databases

rasp.eng – motor data

; semi-colon in the first column followed by text is a comment and is not processed.
Type diameter length delays propellant_wt total_wt mfg notes
time1 thrust1
time2 thrust2
...
timeN thrustN

Internally, prior to wRASP v1.61, each motor was stored in blocks of 64 values – a maximum of 32 time/thrust pairs were allowed.

From v1.61 and later, motor data values were stored in a "pool" of 16384 double precision values. There is no maximum for an individual motor.

rocket.dat – rocket kit database

mfg name [; alias#1] [; alias#2]...
weight length diameter max_alt
catalog#, motor list (catalog date) price

Useful if the data is complete. Many manufacturers do not include a typical rocket empty weight.

vendor.dat – manufacturer data

abbreviation full_name [; abbrev alias#1][...]

The rocket database is only used to browse the kit list. The vendor list is used to identify the manufacturer of a motor or kit.

5. **DigiTrak** (by Larry Curcio)

Special Features: optimal mass calculation
C_D backtracking (time or altitude based)

DOS DigiTrak used a separate program, AIRSTART, to combine the thrust curves of multiple motors. wRASP does this automatically and outputs the result in airstrt1.out for stage 1, airstrt2.out for stage 2, and airstrt3.out for stage 3.

See the “help” file for additional DigiTrak information and tips on backtracking. Larry’s original discussions from his DigiTrak version are included there.

Newton’s method of slope (first derivative) is used to adjust the C_D while backtracking. The C_D is not allowed to go below 0.015.

6. Software Modules

Group	File	function(s)	purpose(s)
General	readme	-	Revision history
	track.bas	-	Original tracking BASIC code
Windows	wrasp.def	-	".exe" definition file
	wrasp.rh	-	windows resource header
	wrasp.rc	-	windows resource description
	wrasp.res	-	windows resources (compiled)
	wrasp.mak	-	make file for Borland C/C++
Help	topics.rtf	-	Help topics (outline)
	wrasp.rtf	-	Help body
	wrasp.hpj	-	Help project definition file
	wrasp.hlp	-	Compiled help file
Main	wrasp.h	-	Main header file
	prowrasp.h	-	Function prototypes
	rasp.h	-	RASP constants and structures
	stats.h	-	More constants and structures
	wrasp.c	upd_rkt	Update rocket data
		sortfn	General purpose sort function
		init_efile	Initialize and load motor database
		del_efile	Delete current motor data in memory
		ld_alpha	Load alpha rocket when motor DB changed
		WinMain	Main program
		Wnd_procM	Window procedure – motor data window
		Wnd_proc	Window procedure
		AboutDlgProc	Help (About) dialog
		InfoDlgProc	Help (System Info) dialog
	raspsubs.c	calc	Physics routine
		prt_full	Print full flight
		prt_summary	Print summary
		prt_plot	Output plot file
	engine.c	init_engine	Initialize engine data block
		load_engine	Load engine data file
	fltclass.c	fltclass	Calculate flight class
		motorclass	Calculate motor class
	files.c	ld_config	Load wRASP configuration file
		sv_config	Save wRASP configuration file
		get_ini	Get wrasp.ini data values
		set_ini	Save wrasp.ini data values
	atmos.c	powf	Floating point pow()
		logf	Floating point log()
		sqrtf	Floating point sqrt()
		atmos	Atmosphere model
	rocket.c	rocketdat	Load rocket DB file
	vendor.c	get_abbrev	Get vendor abbreviation
		vendordat	Load vendor DB file
	stage.c	init_stage	Initialize stage structure
		Stage1DlgProc	Stage data dialog (all stages)
		ComplexDlgProc	Complex Stage dialog
		NStageDlgProc	General rocket data dialog
Dialogs	browse.c	BrowseDlgProc	Rocket database browse dialog
	environ.c	EnvDlgProc	Environment dialog
	option.c	OptionsDlgProc	Options dialog

		UpdateDlgProc CalcDlgProc DTDlgProc TrkDlgProc	Graphics update options dialog Calculations dialog DigiTrak options dialog NAR Tracking dialog
DigiTrak	digitrak.h digitrak.c dt_orig.c as_orig.c dt_conv.txt dt_eng.txt	- dt_stage get_mach1 machmaker set_next_values prt_Dtsummary FM_coast_time FM_coast_alt boost coast trajectory optmass newton backtrack digitrak - - - -	DigiTrak header Builds stage motor data (from airstart) Find mach 1 from temperature Calculate mach effects on drag Next value set Print stage summary Backtracking function Backtracking function Boost phase Coast phase Combines boost and coast Optical mass calculations Newton's method for backtracking Backtracking Top level control Original DigiTrak source Original Airstart source Conversion notes DigiTrak engine data notes
Graphics	wrasp1.c adj.c out_d.c out_i.c out_spc.c out_t.c	wrasp1 motor1 adj adjf out_d* out_I out_spc out_t*	Graphics and printer output Motor data output Screen parameter adjustment Screen parameter adjustment Decimal text output Integer text output Special text output String text output
Support	gp.c Conv.c	getline tluppd tlupf getfn remfn getext remext C_to_F F_to_C	Binary file line input Table look-up, double precision Table look-up, single precision Extract file name from string Remove file name from string Get file extension Remove file extension Covert Celsius to Fahrenheit Convert Fahrenheit to Celsius

7. Resources

The latest versions and source will appear here:

http://www.wrasp.com

A very new version for X Windows (wxRASP) by Matthew Dockrey:

http://www.cs.washington.edu/homes/mrd/wxrasp.html

General rocketry calculations:

http://www.execpc.com/~culp/rockets/rckt_eqn.html

http://www.execpc.com/~culp/space/space.html