AA 279A - Space Mechanics
Winter 2017

Lecture 11 Notes

Wednesday, 15 February 2017
Prof. Andrew Barrows

Topics for today

AA 279B Preview
Orbit phasing
Synodic period
Phasing for non-coplanar transfers
State vector form of equations of motion
Numerical integration methods

Prof. Barrows' Remaining Office Hours:
(in Durand 359)

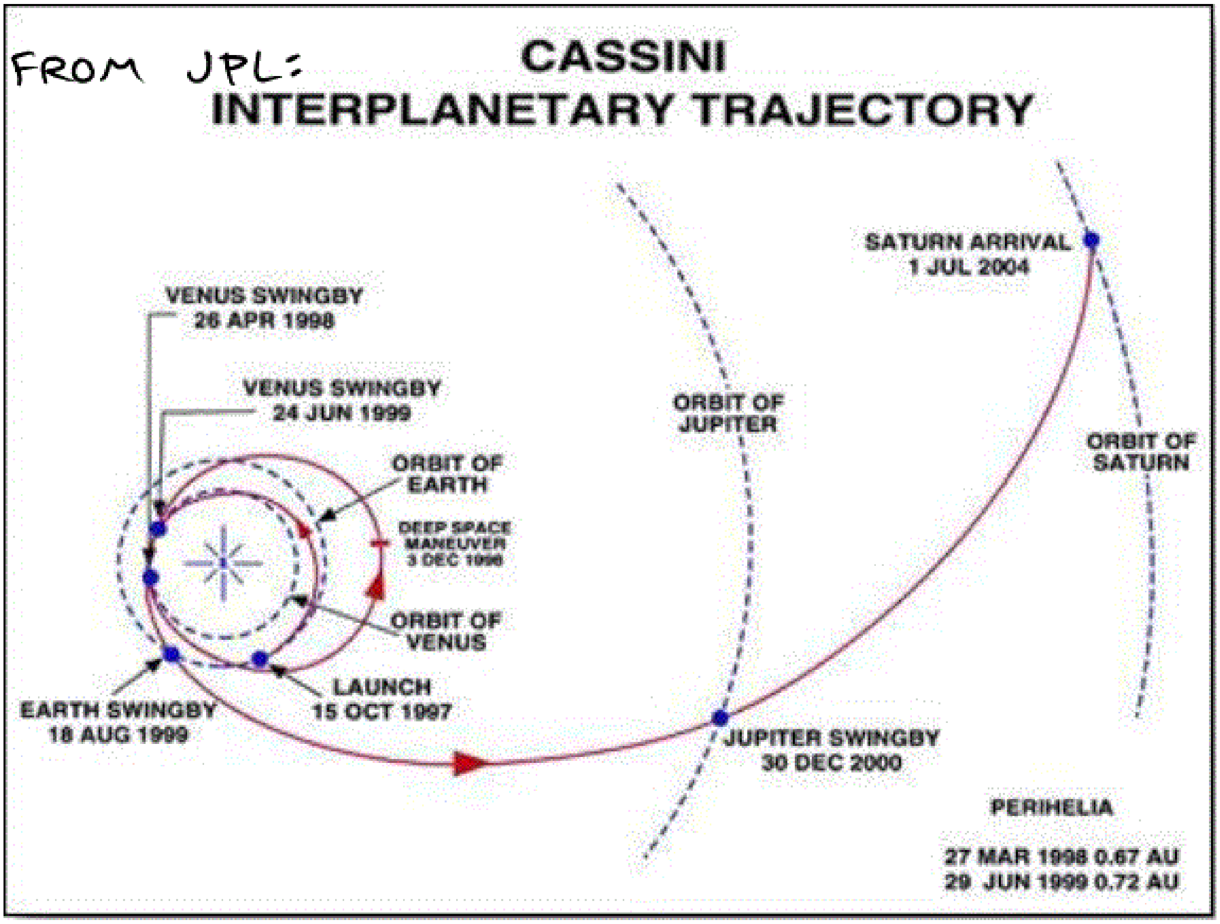
Wed 2/15 → 3:00 – 5:00
Thu 2/16 → 9:30 – 10:30
Problem Set 5 due Mon 2/20
Midterm Exam Wed 2/22
Mon 2/27 → 10:30 – 12:30
Problem Set 6 due Wed 3/1

INTERPLANETARY TRANSFER

Want to go from one planet to another in solar system.

EXAMPLE: Cassini mission to Saturn
(launched 1997, arrived at Saturn 2004)

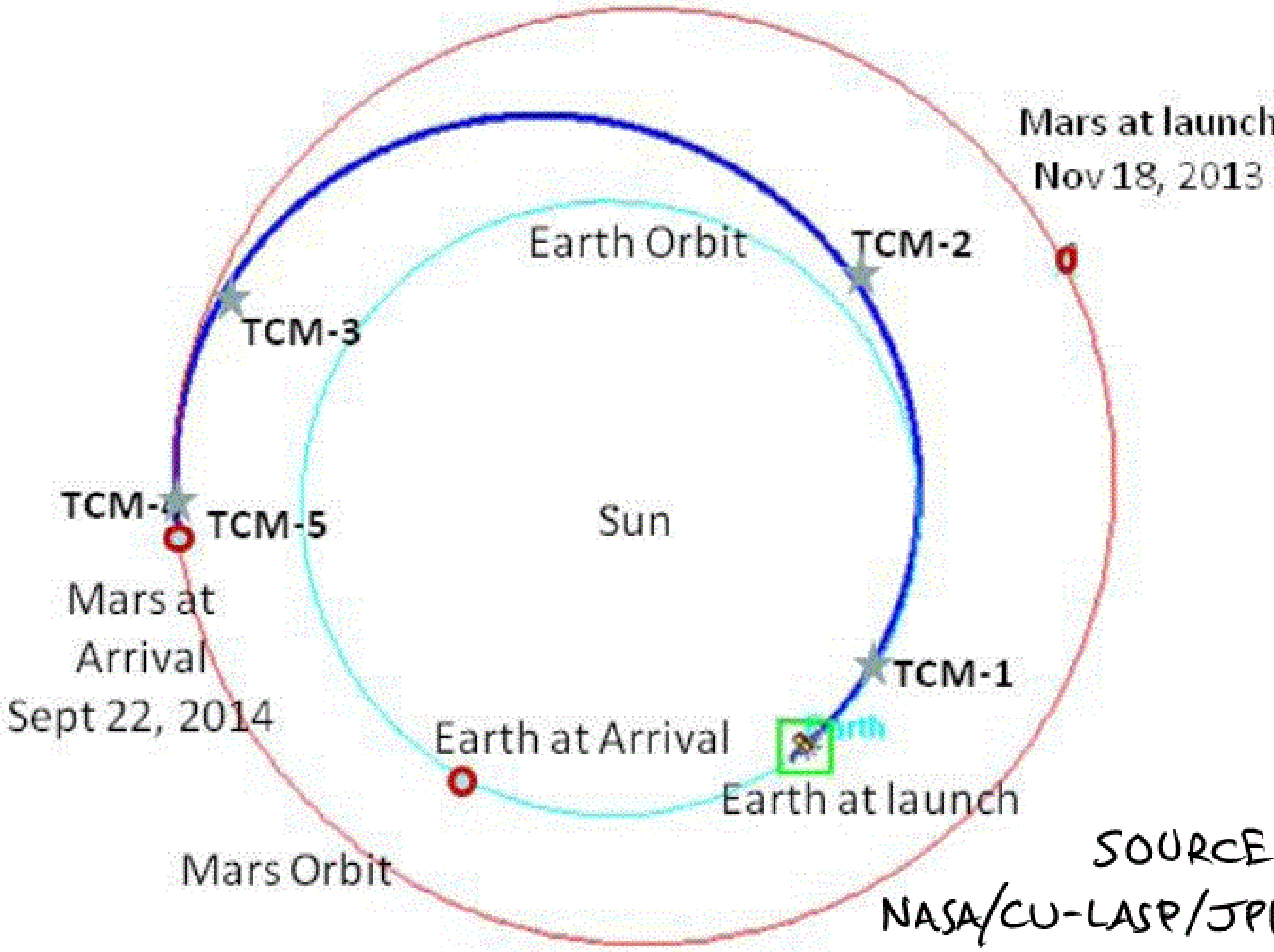
FROM JPL:



Also: Messenger mission to Mercury
(launched 2004, arrived at Mercury 2011)

Unfortunately, N-body problem of spacecraft going from earth to some planet while attracted by sun and other planets has no analytical solution.

NASA MAVEN TRAJECTORY

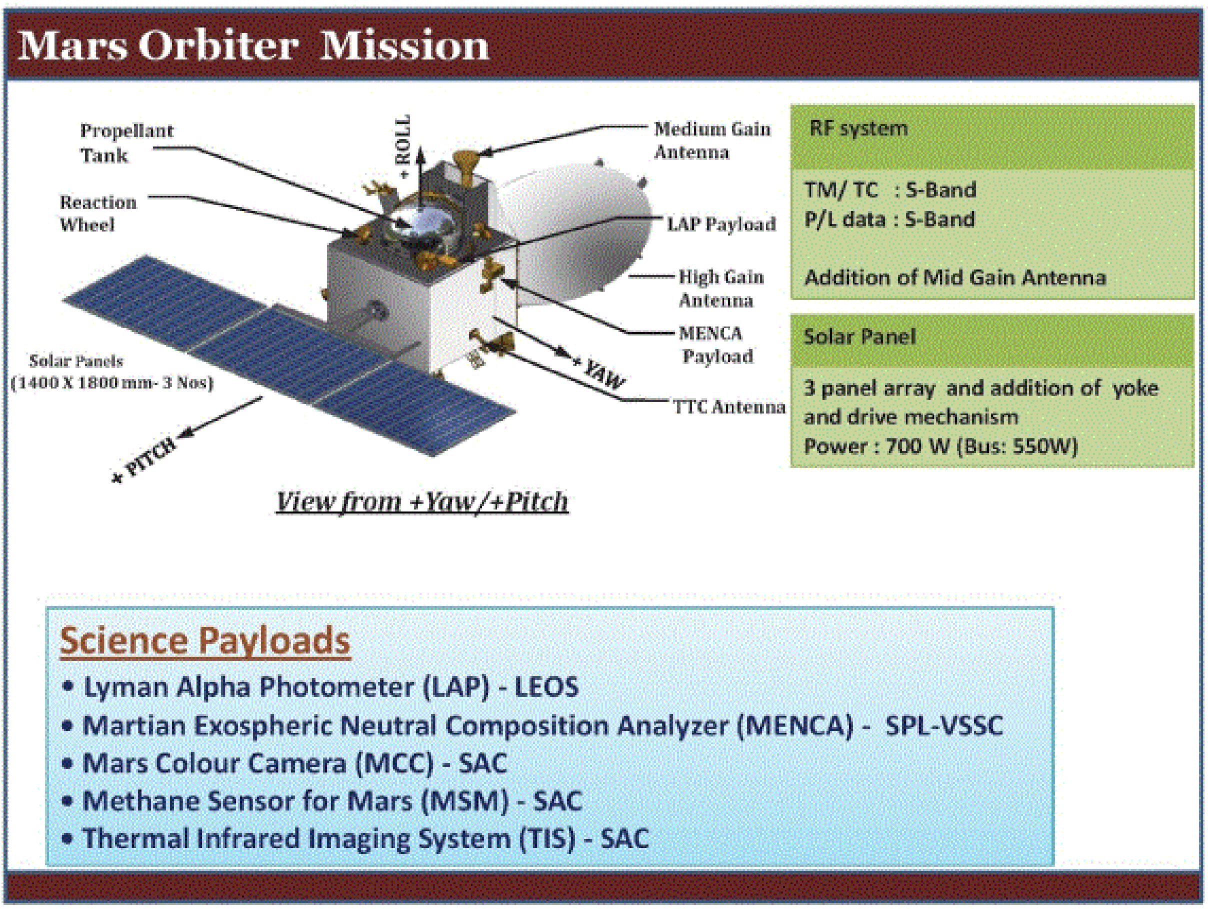


SOURCE:
NASA/CU-LASP/JPL

(Powerful Centaur upper stage enabled straight forward earth departure trajectory.)

ISRO MARS ORBITER MISSION

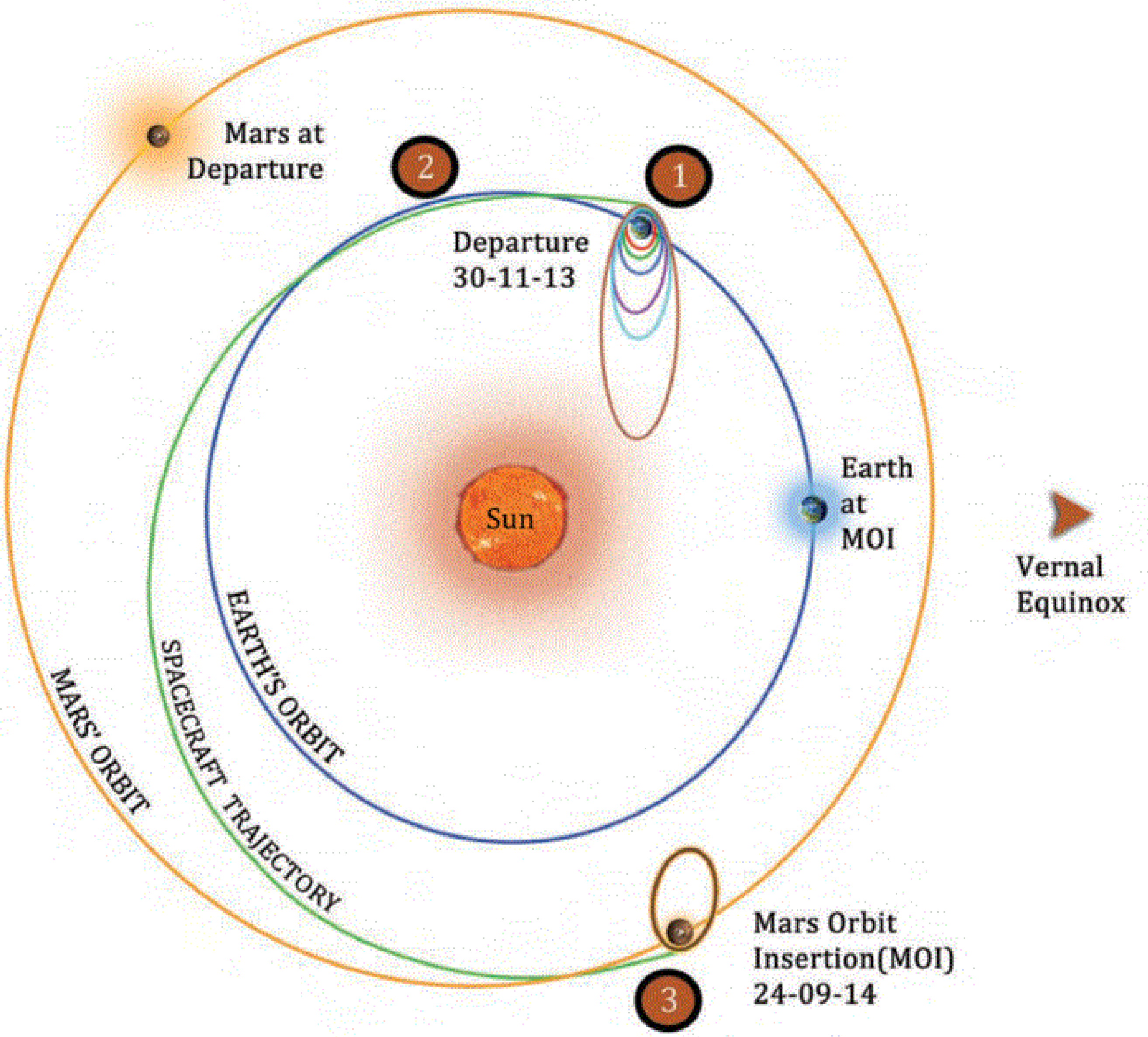
- Launched 5 November 2013
- Polar Satellite Launch Vehicle (PLSV) rocket
- 1x 440 N rocket on spacecraft
- ~ $70M
- Technology demonstrator



SOURCE: ISRO

1,350 [kg] at launch
~ 1.5 [m] cube

MARS ORBITER MISSION TRAJECTORY



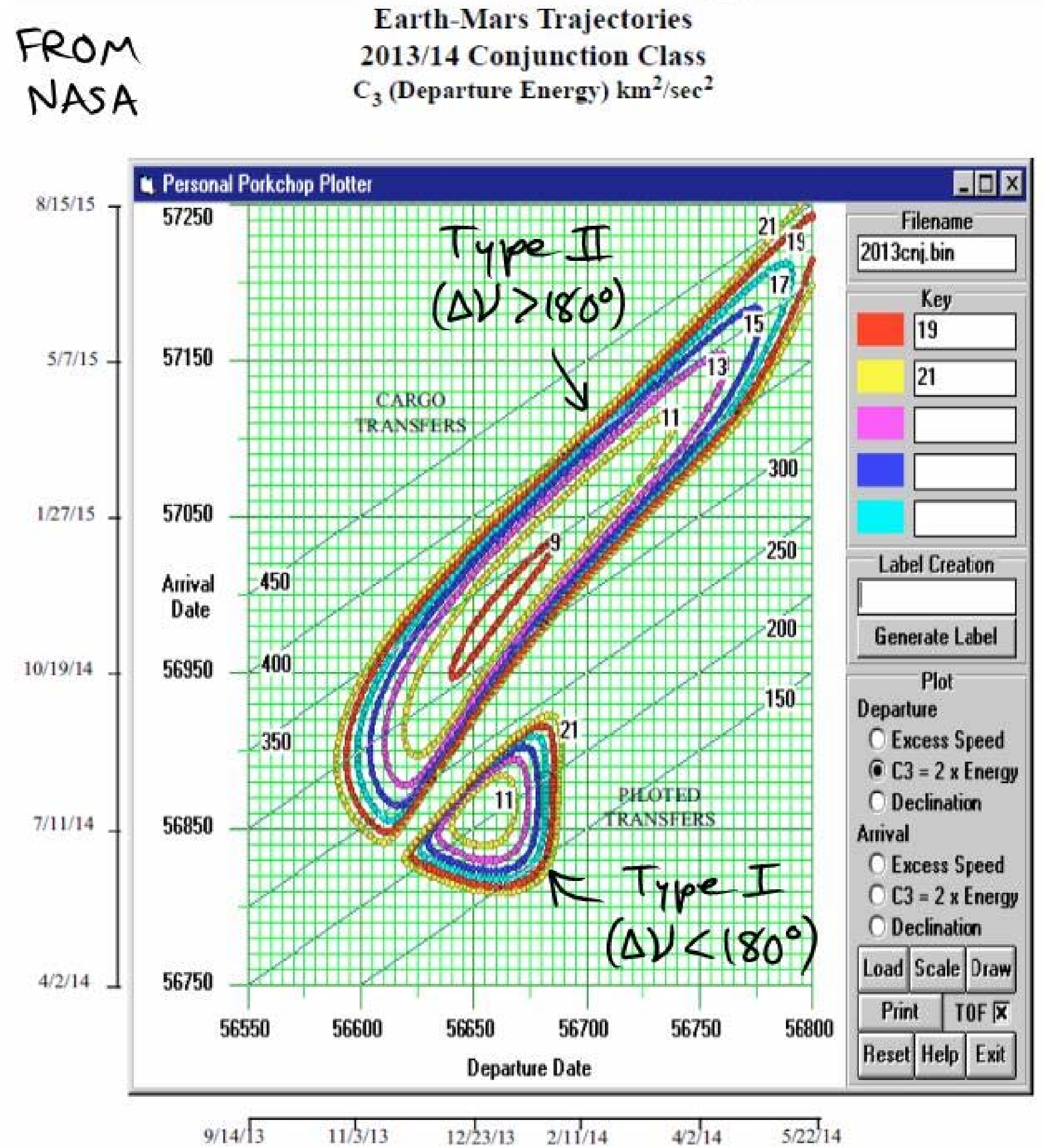
Trajectory Design

SOURCE: ISRO

(25 days of apogee-raising maneuvers were required to achieve energy required to depart Earth.)

2013 EARTH → MARS OPPORTUNITY

FROM NASA



2013 launch opportunity favored Type II trajectories.

AA 279B PREVIEW

ADVANCED SPACE MECHANICS

TYPICALLY OFFERED SPRING QUARTER
→ BUT, NOT THIS SPRING 2017 ☹

Three Body Problem
  Lagrange points

Relative Motion } may change
  HCW equations }

Satellite Communications
  Link budgets, data rate vs. distance

Lambert's Problem
  Intercept and rendezvous

Interplanetary Mission Design
  Full 3D analysis
  Real-world examples

Satellite Constellations
  Navigation (e.g. GPS), communications

Space Debris

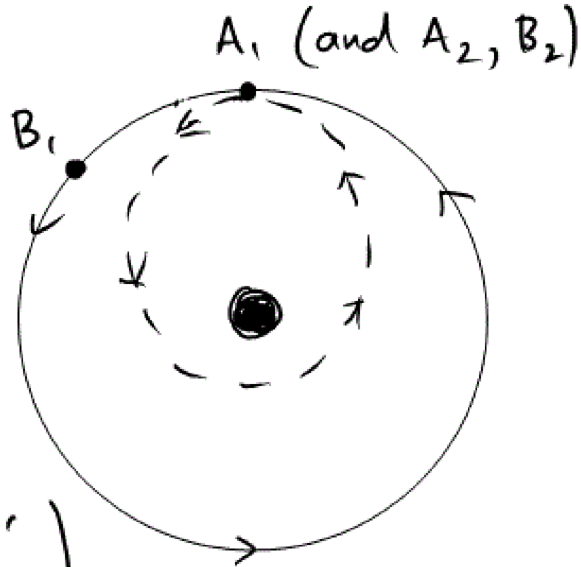
COURSE PROJECT with in-class presentations at end of quarter

MIDTERM EXAM ONLY, no Final Exam

ORBIT PHASING

These are orbital maneuvers used to get to "the right place at the right time."

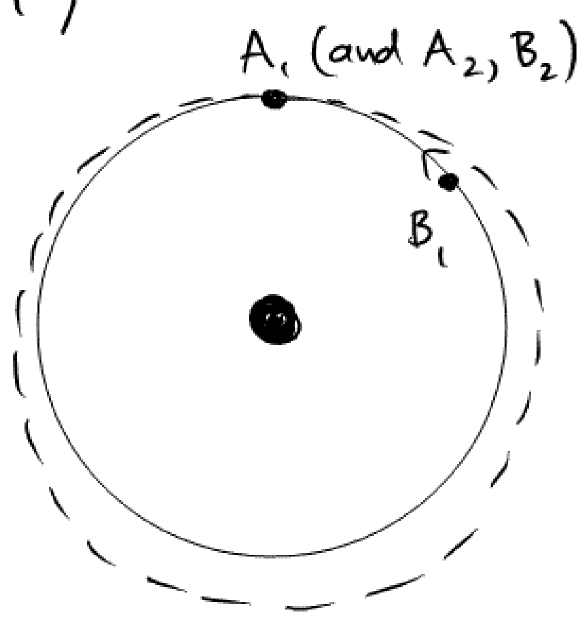
PHASING WITHIN A SINGLE CIRCULAR ORBIT



At t_1, A wants to 'catch up' to meet B.

A does ΔV, takes a faster ellipse, and meets back up with B at t_2.

(Additional $\Delta V \rightarrow$ 'RENDEZVOUS'
No extra $\Delta V \rightarrow$ 'INTERCEPT')



At t_1, A wants to 'fall behind' to meet B.

A does ΔV, takes a slower ellipse, and meets back up with B at t_2.

Math is straightforward.

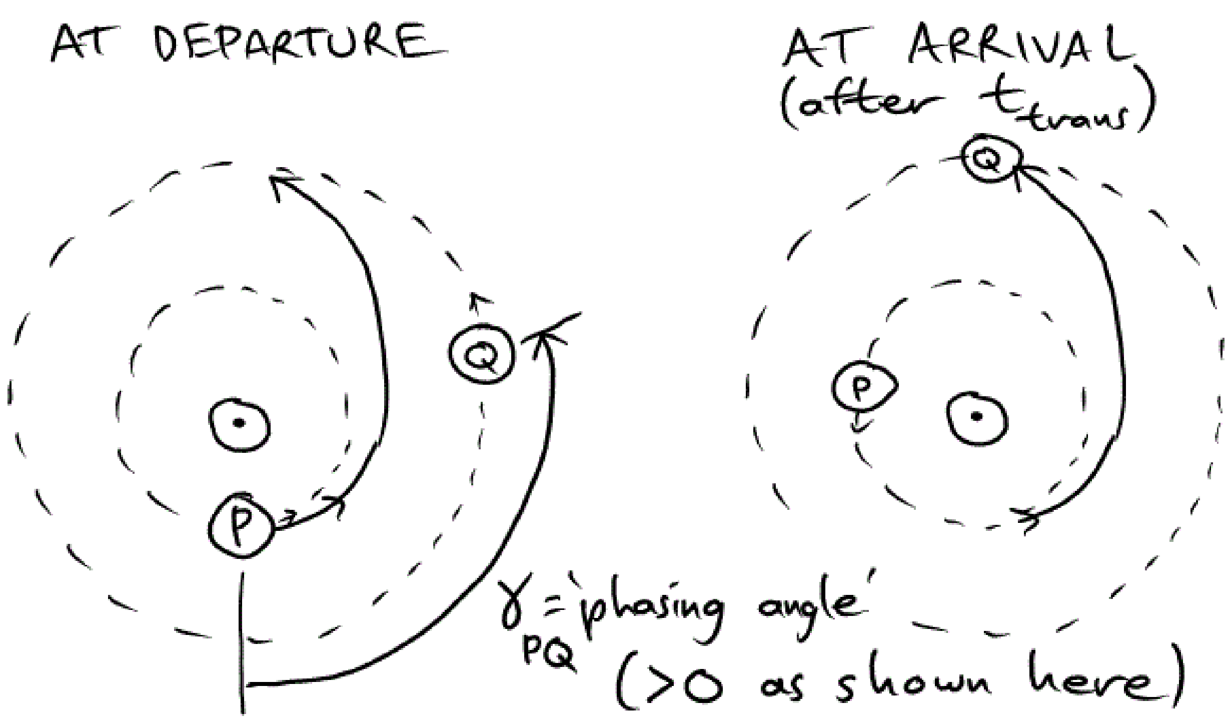
HOHMANN TRANSFER PHASING

Imagine the sun (⊙) is the central body, and we want to follow a Hohmann transfer from P to Q (which are in circular orbits) with correct timing. In fact, this won't happen by accident; we must plan the departure time.

For Hohmann transfer between P and Q (typically planets), transfer time is

$$t_{trans} = \frac{1}{2} 2\pi \sqrt{\frac{a_t^3}{\mu_\odot}} = \pi \sqrt{\frac{({}^\odot r^P + {}^\odot r^Q)^3}{8\mu_\odot}}$$

Opportunities to use Hohmann transfer only occur with P and Q phased correctly:



We want expression for phasing angle γ_{PQ} so we know when to launch. Since planet Q moves through angle of $n_Q t_{trans}$ during transfer,

$$\gamma_{PQ} + n_Q t_{trans} = \pi$$

$$\gamma_{PQ} = \pi - n_Q t_{trans}$$

SYNODIC PERIOD

We can show that this phasing angle γ_{PQ} occurs periodically, with occurances spaced at the '<u>synodic</u> <u>period</u>':

$$\tau_{syn} = \frac{2\pi}{|n_P - n_Q|}$$

The closer ${}^{\odot}r^{P}$ and ${}^{\odot}r^{Q}$ are (and therefore the closer n_P and n_Q are), the longer we'll have to wait between these special alignments!

EXAMPLE

Table of values of transfer time t_{trans} and synodic period τ_{syn} for Hohmann transfers from earth to other planets

	t_{trans} [days] (years)	τ_{syn} [days]	
Mercury	106	117	dominated by planet's mean motion
Venus	146	584	
Mars	259	780	
Jupiter	1,001 (2.74)	399	dominated by earth's mean motion
Saturn	2,206 (6.04)	378	
Uranus	5,902 (16.16)	369	
Neptune	11,242 (30.78)	368	

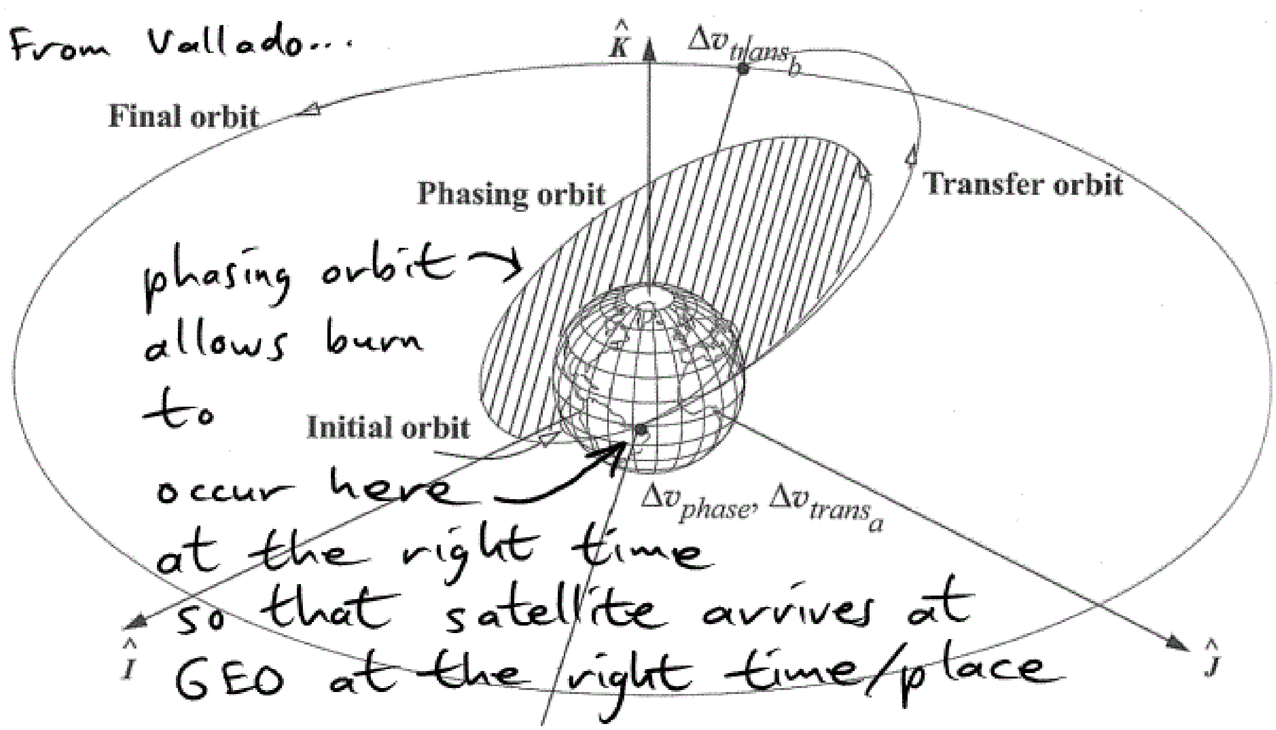
launch windows to Mars ~2.1 years apart

PHASING ORBITS / NON-COPLANAR

Orbit used to "kill time" while waiting for the right instant to do an orbital maneuver.

Can be especially useful for non-coplanar transfers. Since we may need to do burns at a node (e.g. inclination change) we can't choose just any time for burn as in coplanar case.

→ Insert "phasing orbit" with period τ that gives desired timing. Often doesn't require any extra ΔV.



EQUATIONS OF MOTION IN STATE VECTOR FORM

Until now, we've used analytic solutions of Fundamental Orbital Differential Equation to describe orbits.

$$\frac{{}^{i}d^2}{dt^2}\vec{r} = -\mu\frac{\vec{r}}{|\vec{r}|^3}$$

But, we got to this because we ignored movement of the central body; third, etc. bodies; and other forces like drag, solar radiation pressure, and other perturbations. In fact,

$$\frac{{}^{i}d^2}{dt^2}\vec{r}_j = -\sum_{\substack{k=1\\k\neq j}}^{N}\mu_k\frac{\vec{r}_{kj}}{|\vec{r}_{kj}|^3} + \overrightarrow{\text{other forces}}$$

We'll numerically integrate this to predict orbits and understand these perturbations.

First, we'll rewrite equation of motion in state vector form:

state vector for body j is

$$\vec{y}_j = \begin{bmatrix} \vec{r}_j \\ {}^i\vec{v}_j \end{bmatrix} \quad \text{so that}$$

$$\frac{{}^i d\vec{y}_j}{dt} = \begin{bmatrix} \frac{{}^i d}{dt}\vec{r}_j \\ \frac{{}^i d}{dt}{}^i\vec{v}_j \end{bmatrix} = \begin{bmatrix} {}^i\vec{v}_j \\ \frac{{}^i d^2}{dt^2}\vec{r}_j \end{bmatrix} = \begin{bmatrix} {}^i\vec{v}_j \\ \sum \overrightarrow{\frac{\text{forces on } j}{m_j}} \end{bmatrix}$$

Move from vector form to matrix math form in some inertial coordinate system. (Since time derivatives also i-frame, we can now just use scalar derivatives.)

$$\begin{bmatrix} \dot{\underline{r}}_j \\ --- \\ \dot{\underline{v}}_j \end{bmatrix} = \begin{bmatrix} \underline{v}_j \\ --- \\ \sum \frac{\underline{F}_j}{m_j} \end{bmatrix} = \dot{y}_j = f(y_j)$$

← 6 × 1 matrix

Integrating these 6 1^{ST}-order DEs is the same as solving the 3 2^{ND}-order DEs.

We can 'stack' these formulas over N bodies:

$$
\begin{bmatrix} \dot{\underline{r}}_1 \\ \dot{\underline{r}}_2 \\ \downarrow \\ \dot{\underline{r}}_N \\ \hline \dot{\underline{V}}_1 \\ \dot{\underline{V}}_2 \\ \downarrow \\ \dot{\underline{V}}_N \end{bmatrix} = \begin{bmatrix} \underline{V}_1 \\ \underline{V}_2 \\ \downarrow \\ \underline{V}_N \\ \hline \frac{1}{m_1}\Sigma \underline{F}_1 \\ \frac{1}{m_2}\Sigma \underline{F}_2 \\ \downarrow \\ \frac{1}{m_N}\Sigma \underline{F}_N \end{bmatrix} = \dot{y} = f(y)
$$

$\leftarrow$ $6N \times 1$ matrix

... in order to run large-scale simulations.

Given $\dot{y} = f(y)$ and initial conditions y_0, MATLAB can integrate this for us.

(Our job becomes to write the function $f(y)$ that returns the state derivative given the state.)

NUMERICAL INTEGRATION TECHNIQUES

Generally based on Taylor series to find $y(t) = y(t_0 + h)$ given $y(t_0)$, $\dot{y}(t_0) \triangleq f(t_0)$, and h (the "timestep").

$$y(t_0 + h) = y(t_0) + f(t_0)\,h + \frac{\dot{f}(t_0)\,h^2}{2} + \text{H.O.T.}$$

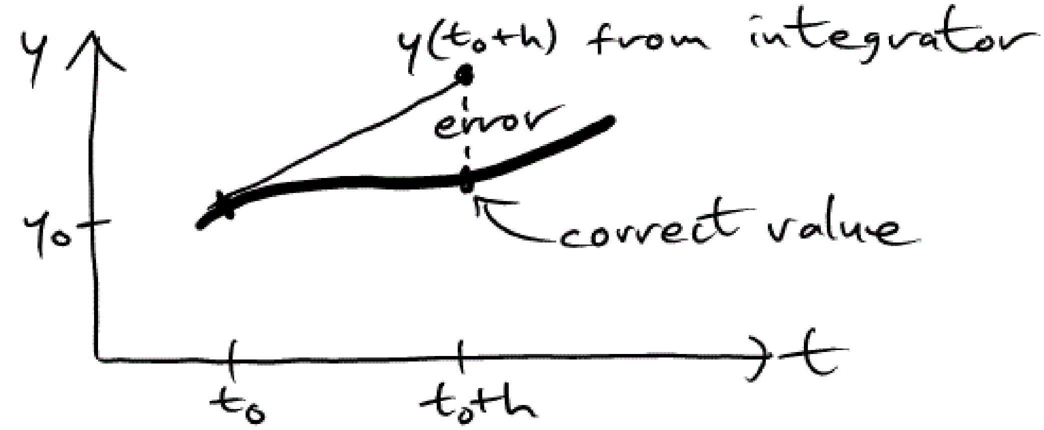
$\dot{y}$ (know this) $\qquad$ $\ddot{y}$ (don't know this)

Simplest integration scheme is the...

EULER INTEGRATOR

$$y(t_0 + h) = y(t_0) + h f(t_0)$$

slope



Simple, but...

- overall error proportional to h ('first order')
- can be numerically unstable

RUNGE-KUTTA METHODS

These try to estimate the higher-order terms by evaluating $\dot{\underline{y}} = f(t, \underline{y})$ at multiple points within a given timestep of duration h.

(note that $\dot{y}$ is most generally a function of y and t)

'FOURTH ORDER RUNGE KUTTA'

$$y(t_0+h) = y(t_0) + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

where

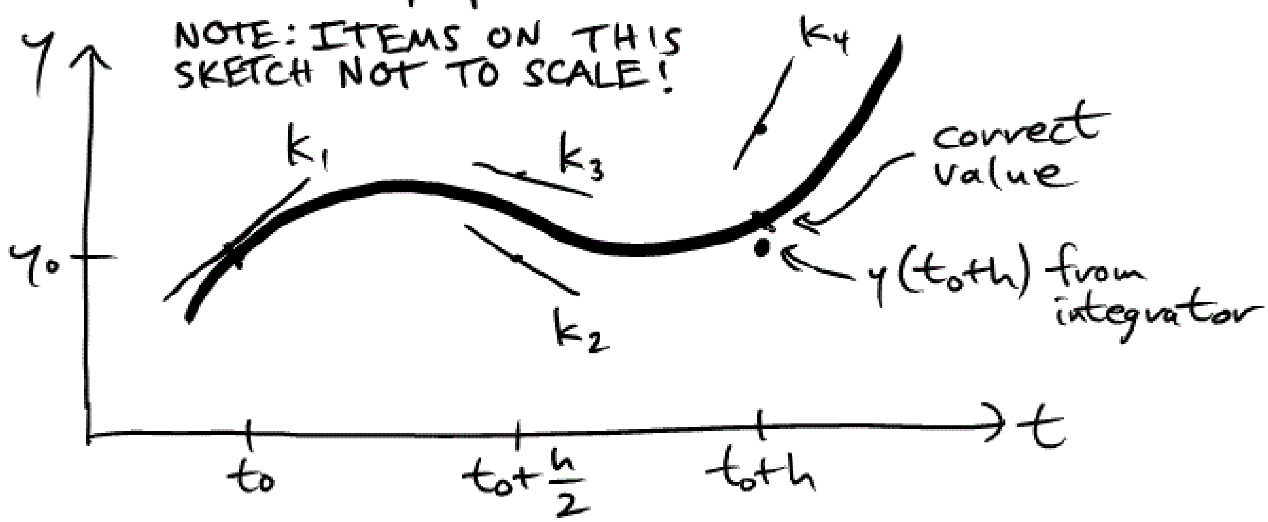
$$k_1 = f(t_0, y_0)$$

$$k_2 = f\left(t_0 + \frac{h}{2}, y_0 + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(t_0 + \frac{h}{2}, y_0 + \frac{h}{2}k_2\right)$$

$$k_4 = f\left(t_0 + h, y_0 + hk_3\right)$$

- overall error proportional to h^5 ('fourth order')



ADAMS-BASHFORTH-MOULTON METHOD

'Multistep' method - saves information from previous timesteps to improve current timestep. Also a 'predictor-corrector' method - first it makes a prediction:

$$y_{n+1}^{p} = y_n + \frac{h}{24}\left(55\dot{y}_n - 59\underbrace{\dot{y}_{n-1} + 37\dot{y}_{n-2} - 9\dot{y}_{n-3}}_{\text{saved from previous timesteps}}\right)$$

then a correction:

$$y_{n+1} = y_n + \frac{h}{24}\left(9\dot{y}_{n+1}^{p} + 19\dot{y}_n - 5\dot{y}_{n-1} + \dot{y}_{n-2}\right)$$

Compare y_{n+1}^{p} to y_{n+1} to assess accuracy and stability.

SUMMARY

A wide variety of methods is available.
Choosing is part art, part science.
Rules of thumb

- Want at least 100 points per orbit.
- Runge Kutta (4,5) (MATLAB function ode45) is always a good first try.
- Experiment in MATLAB!

NUMERICAL INTEGRATION HINTS FOR MATLAB

STATE VECTOR DERIVATIVE FUNCTION

(You write this)

You must provide a function that finds

$\dot{y} = f(t, y)$ your function must accept t as an input, even if it ignores t!

VECTOR OF TIMES WHERE YOU WANT DATA

You provide a vector of times at which you want MATLAB to output data, e.g. [10.0 : 2.0 : 20.0]' is at 10, 12 ..., 18, 20.

INTEGRATION FUNCTIONS

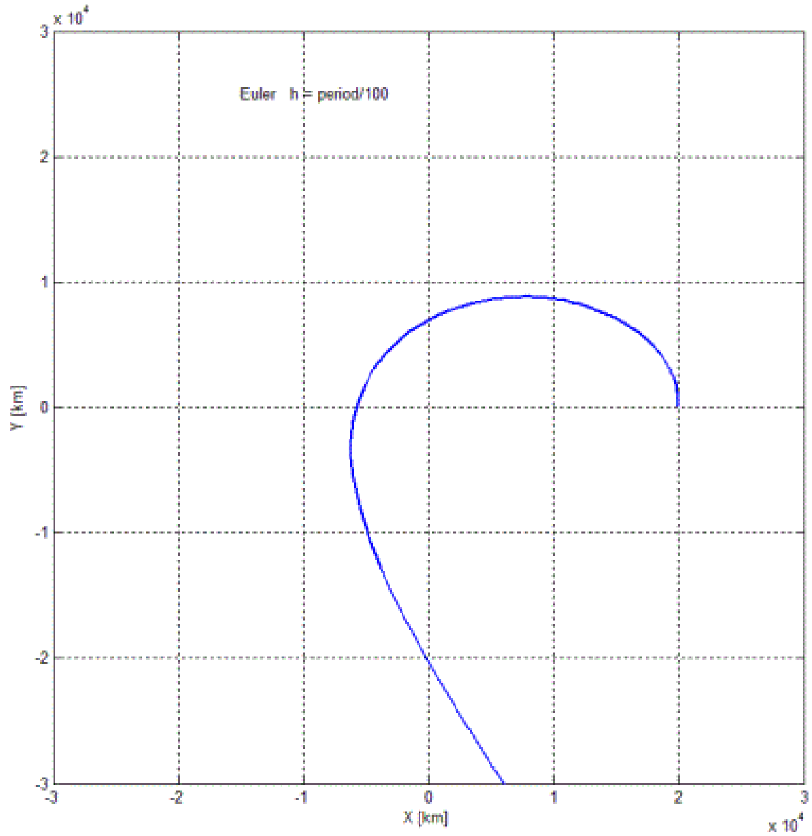
(Matlab supplies these)

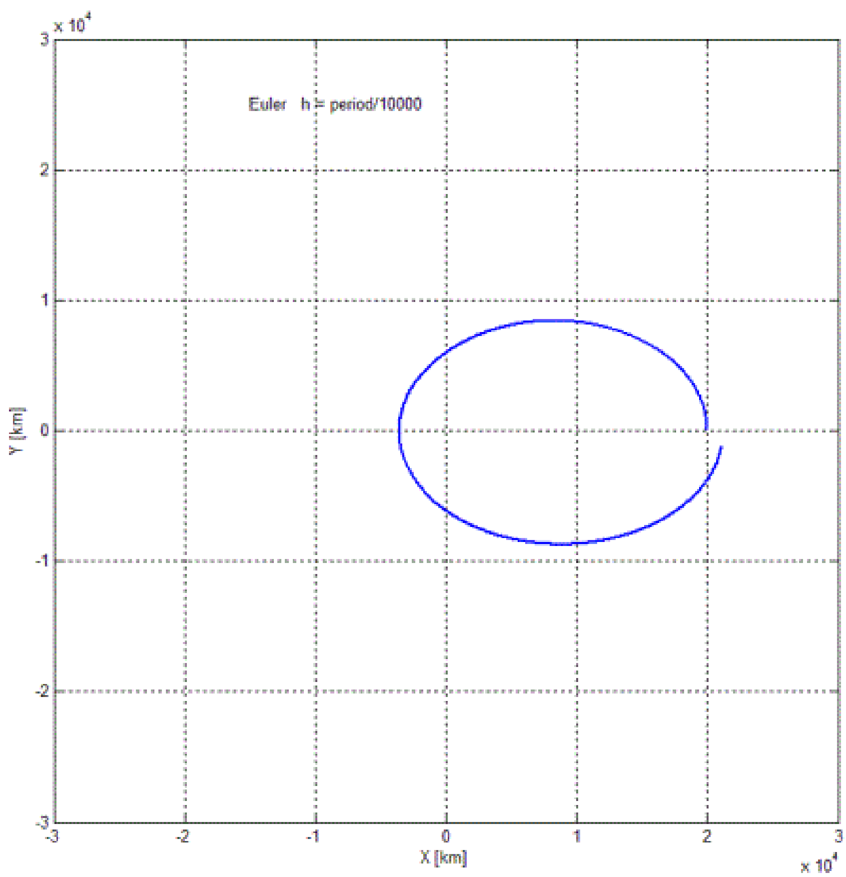
ode45 : A Runge-Kutta variant

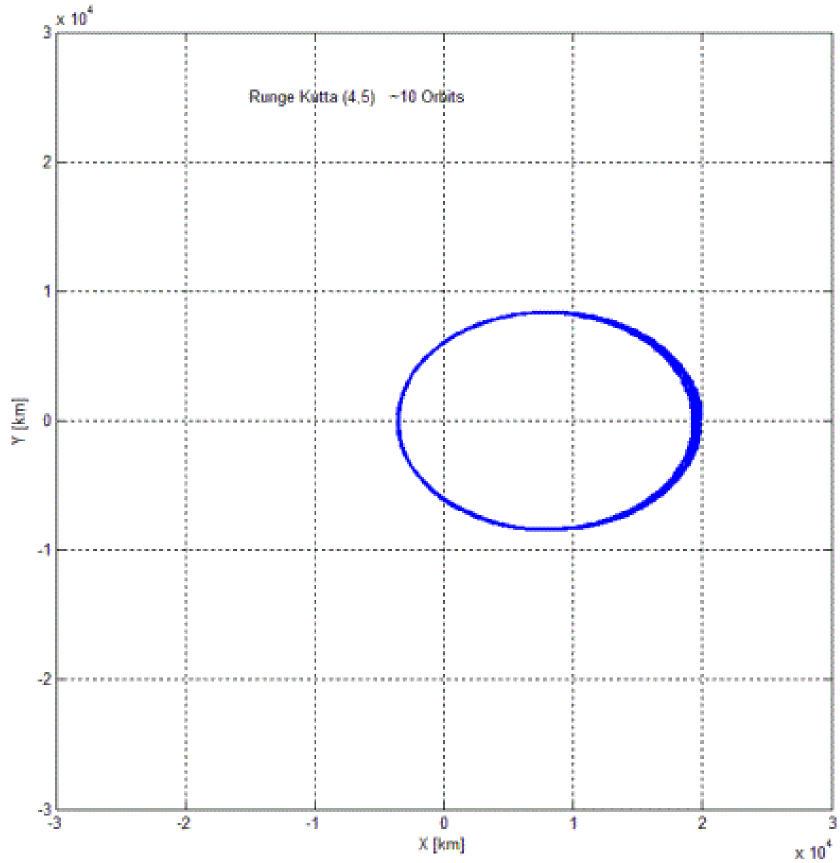
ode113 : Adams-Bashforth-Moulton

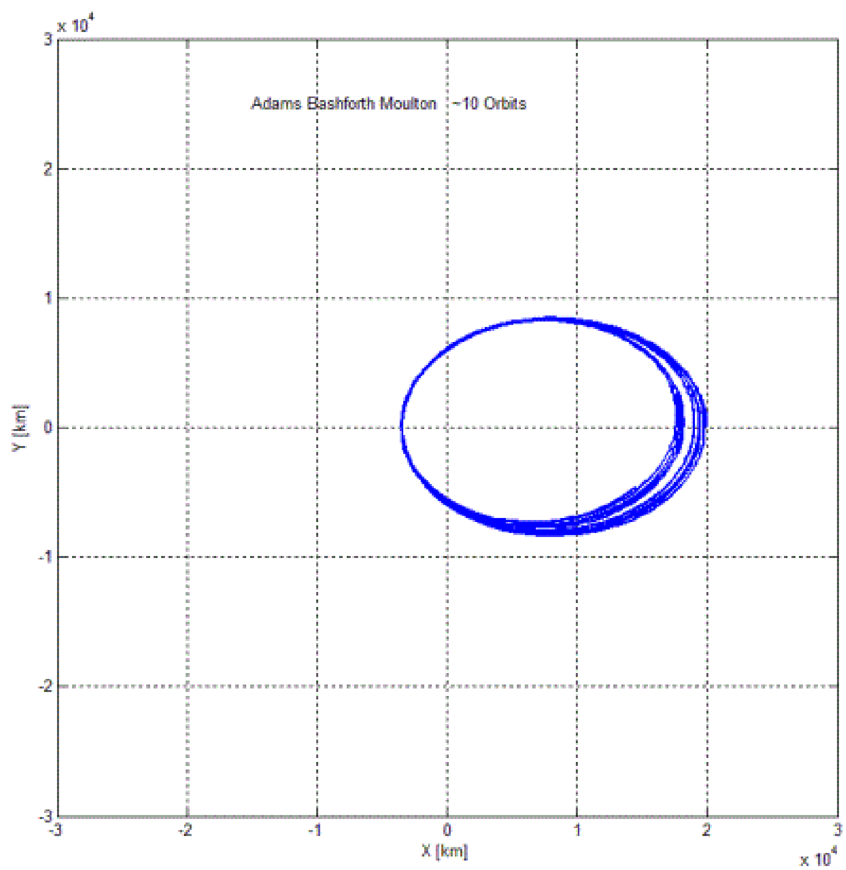
→ These both choose (and dynamically vary) their timesteps internally. They then output data at the times you requested.

→ Both can internally monitor tolerances (accuracies that you can modify via 'odeset' function.)


x 10^4
Euler h = period/100
Y [km]
X [km]
x 10^4


x 10^4
Euler h = period/10000
Y [km]
X [km]
x 10^4


x 10^4
Runge Kutta (4,5) ~10 Orbits
Y [km]
X [km]
x 10^4
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3


x 10^4
Adams Bashforth Moulton ~10 Orbits
Y [km]
X [km]
x 10^4
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3

INCLINED ORBIT WITH OBLATE GRAVITY FIELD