Continuation of Periodic Orbits Around Lagrange Points and AUTO2000

The Three-Body Problem and Space Mission Design

Randy Paffenroth
Applied and Computational Mathematics
California Institute of Technology
redrod@acm.org

February 19, 2002

Collaborators

- Eusebius Doedel (Concordia University)
- Herb Keller (California Institute of Technology)
- Don Dichmann (Consultant, Torrence CA)
- André Vanderbauwhede (Gent)
- Jorge Galán (Sevilla)

Outline

- Introduction to AUTO2000
- Formulation of the Circular Restricted 3-Body Problem for Continuation Methods
- Periodic Orbits Emanating from L1 Computed Using AUTO2000
- Invariant Manifold Calculations
- Formulation of the Full 3-Body for Continuation Methods
- The Figure-8 Orbit of Montgomery and Chenciner
- Conclusions
- Acknowledgments

AUTO2000 Overview

- AUTO was originally written in 1980, but was recently redone in C and an interface based on the object oriented scripting language Python has been added.
- Computes parameter-dependent families of solutions to non-linear equations using numerical continuation.
- Used for the bifurcation analysis of algebraic as well as ordinary differential equations (ODEs).
- Has been used in many applications (fluid dynamics, cardiac electrophysiology, continuum models of large bio-molecules, flight trim analysis, and the gravitational N-body problem, to name but a few.)
- AUTO2000 is freely available from http://auto2000.sourceforge.net.

Basic Equation

The goal of AUTO2000 is to compute and visualize solution sets of equations of the form

$$F(\mathbf{x}) = \mathbf{0}, \;\; F : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n, \;\; n \in \mathbb{W}.$$

I.e. an under-determined system where you have one more unknown than equations.

Away from singularities the solution set of such a system is a 1-dimensional manifold embedded in $(n+1)$-dimensional space.

Many kinds of problems can be phrased in this way, including discretizations of ODE boundary value problems (which includes periodic solutions to the N-body problem).

Natural Parameter Continuation

Suppose we happen to know a solution to our problem and there is some component of the solution we wish to vary.

$$F(\mathbf{x}_0) = 0, \mathbf{x}_0 = (\mathbf{y}_0, \lambda_0), \mathbf{y}_0 \in \mathbb{R}^n, \lambda_0 \in \mathbb{R}$$

Natural Parameter Continuation

Suppose we happen to know a solution to our problem and there is some component of the solution we wish to vary.

$$F(\mathbf{x}_0) = 0, \mathbf{x}_0 = (\mathbf{y}_0, \lambda_0), \mathbf{y}_0 \in \mathbb{R}^n, \lambda_0 \in \mathbb{R}$$

Then it might be reasonable to assume that a small change in the parameter gives us a new point which is not a solution to our problem, but is close to a solution.

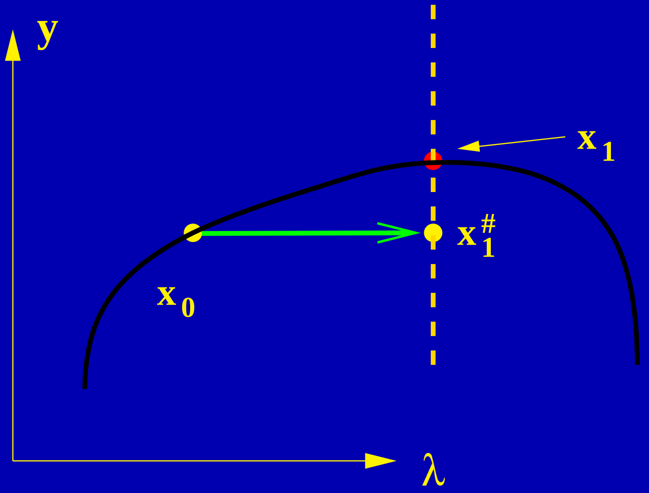
$$\begin{aligned} \mathbf{x}_1^{\#} &= (\mathbf{y}_0, \lambda_0 + \delta s), \\ F(\mathbf{x}_1^{\#}) &\neq 0, \\ F(\mathbf{x}_1^{\#}) &\ll 1 \end{aligned}$$

Natural Parameter Continuation 2

Given the initial guess $\mathbf{x}_1^{\#} = (\mathbf{y}_0, \lambda_0 + \delta s)$ we can then use Newton's method to compute the new solution to a desired accuracy of the augmented problem

$$\begin{aligned} F(\mathbf{x}) &= 0, \\ \lambda - (\lambda_0 + \delta s) &= 0. \end{aligned}$$

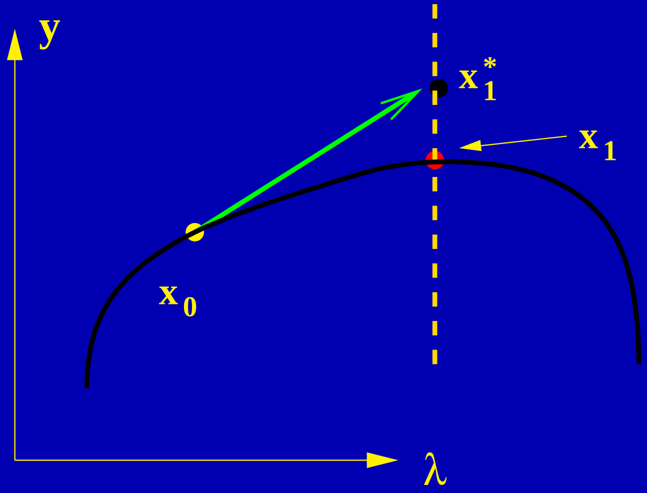
We add the additional equation to give Newton's method a unique solution to search for.



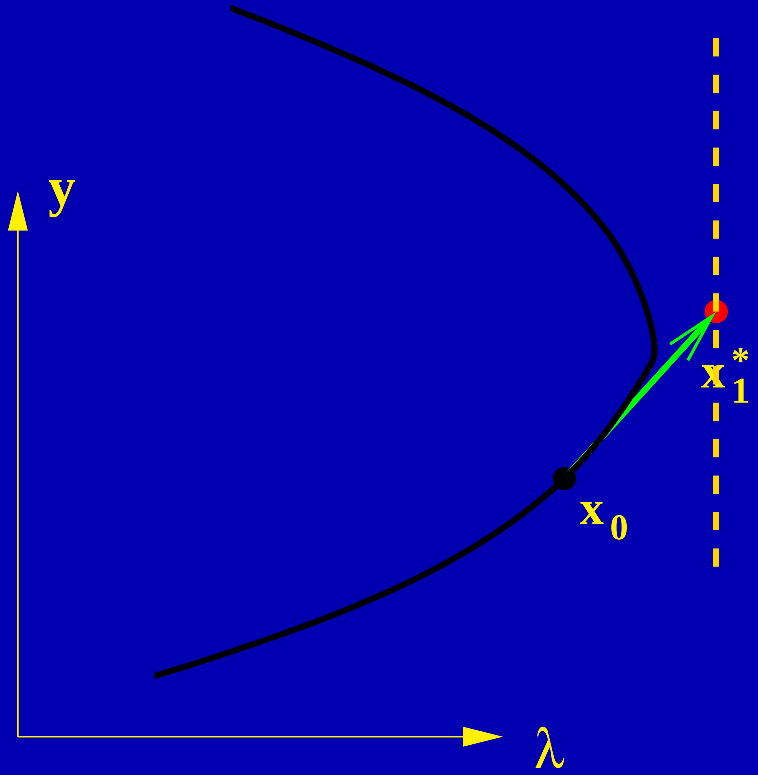
Tangent Continuation

A simple improvement to this method is to use a higher order initial guess $\mathbf{x}_1^*$, such as a tangent to the curve. We then use Newton's method to compute a new solution to the same augmented problem

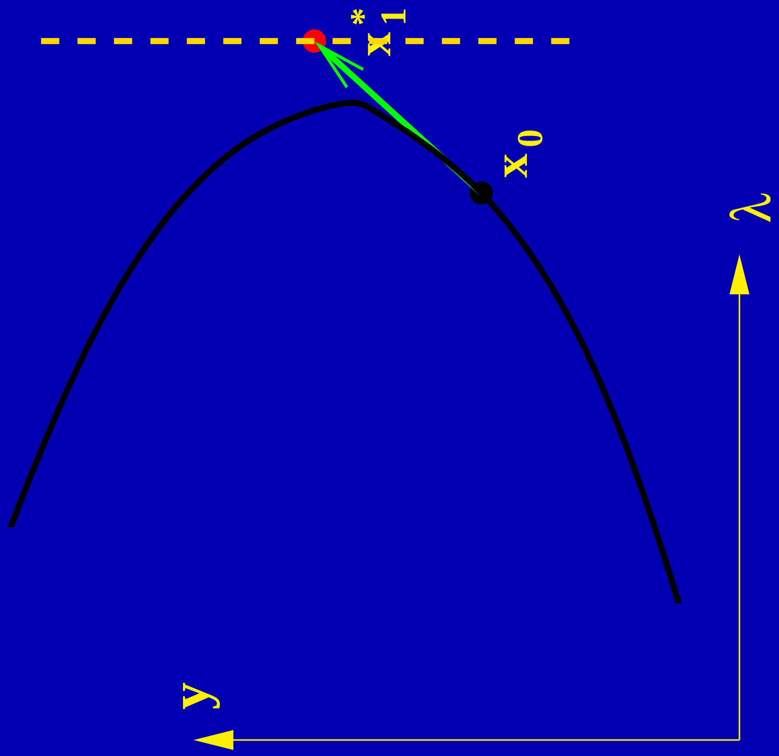
$$\begin{aligned} F(\mathbf{x}) &= 0, \\ \lambda - (\lambda_0 + \delta s) &= 0, \end{aligned}$$



But What Happens at a Fold!?



But What Happens at a Fold!?

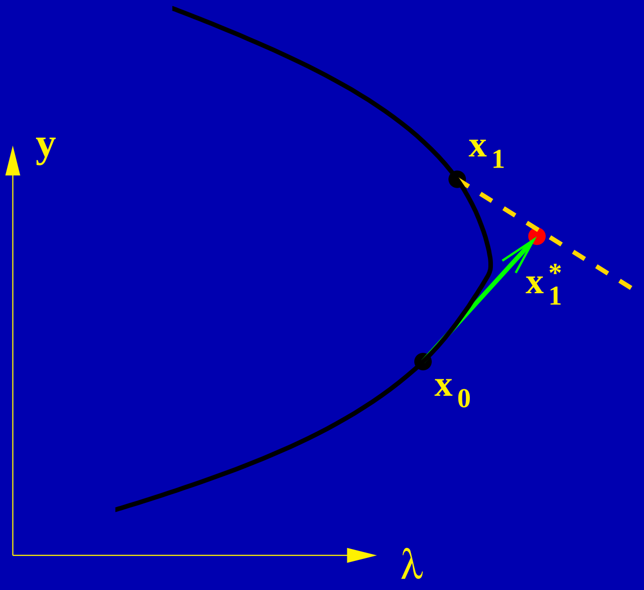


Pseudo-arclength Continuation

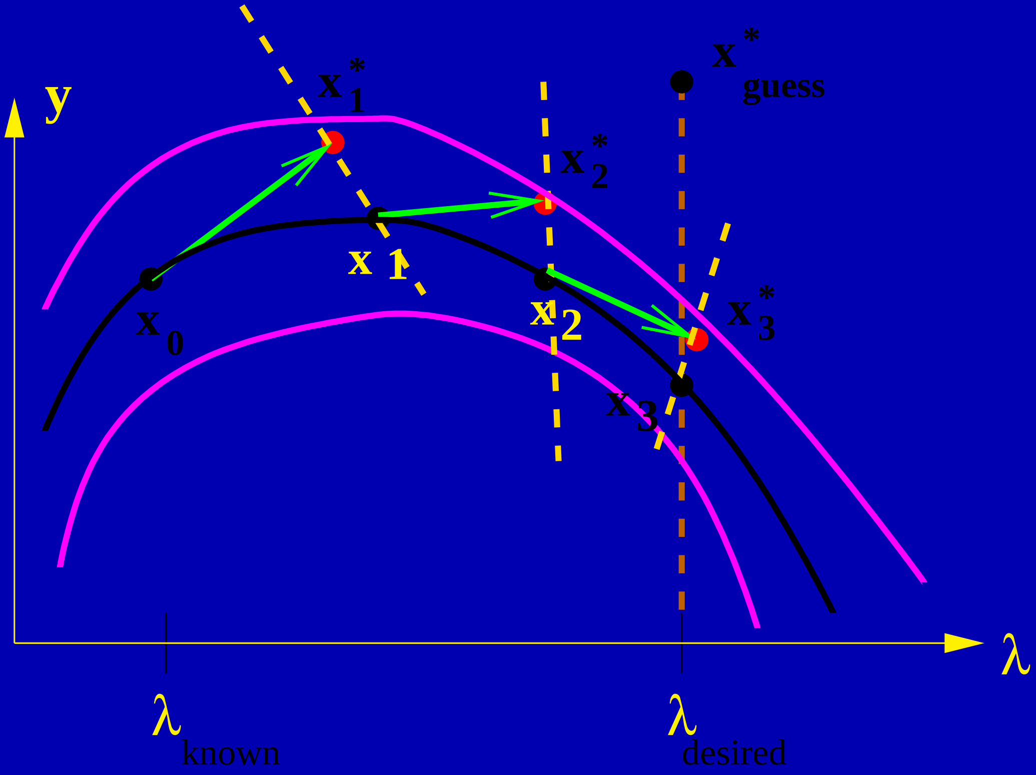
The problem with a fold is that the “parameter” we have chosen cannot be used to “parameterize” the curve. So, we change the system so that we parameterize by something which doesn’t have this problem, namely arclength along the curve.

Specifically, we can change the problem we are solving to be

$$\begin{aligned} F(\mathbf{x}) &= 0, \\ \dot{F}(\mathbf{x}_0) - \mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0 - \delta s) &= 0 \end{aligned}$$



Point of Numerical Continuation



Randy Paffenroth

redrod@acm.org

Software Issues:
Conversion from Fortran to C

AUTO was originally written in Fortran, but AUTO2000 has been translated into C. There are several reasons for the translation, the main ones being better access to external libraries, and ready access to dynamic memory allocation. The original translation was performed using f2c, but the code has been extensively modified.

Python interface

Previous versions of AUTO2000 had a good interface for expert users. Unfortunately, this interface was somewhat difficult for beginners.

Old interaction paradigm:

- Edit a file of control parameters.
- Run AUTO to compute a single family of solutions, a computation taking anywhere from a few second to several days.
- Look at the generated solutions and manually decide what further calculations to preform.
- Repeat.

Python interface (cont)

Accordingly, one question which arose in the modernization of AUTO2000 was "how do we leverage the interface that we already have to create a friendlier one and, at the same time, make it more powerful?" In addition, it was desired to maintain the original AUTO functionality for the pre-existing user base.

Python interface (cont)

Accordingly, one question which arose in the modernization of AUTO2000 was “how do we leverage the interface that we already have to create a friendlier one and, at the same time, make it more powerful?” In addition, it was desired to maintain the original AUTO functionality for the pre-existing user base.

To achieve these goals we decided to implement a wrapper for the old interface using the object oriented scripting language Python. The main idea is to have an interface of the flavor of Matlab, but geared toward continuation methods.

Example Control Script

Just like Matlab, AUTO2000 can be used in two modes. First, it can be used interactively. Second, it is a programming language in which you can express complex computations.:

```
copydemo(’bvp’)

ld(’bvp’)
run()
sv(’bvp’)
ld(s=’bvp’)
data = sl(’bvp’)
ch("NTST",50)
for solution in data:
    if solution["Type"] == ’BP’:
        ch("IRS", solution["Label"])
        ch("ISW", -1)
        # Compute back
        ch("DS",-pr("DS"))
        run()
        ap(’bvp’)
plot(’bvp’)
```

Parallelization

The parallel version of AUTO2000 has been developed to be both portable and efficient on a wide range of parallel architectures, and the code has been written to allow the incorporation of a variety of low level communication protocols. The two currently implemented protocols are explicit shared memory multi-threading (using the standard Pthreads library) and message passing (using MPI).

Parallelization

The parallel version of AUTO2000 has been developed to be both portable and efficient on a wide range of parallel architectures, and the code has been written to allow the incorporation of a variety of low level communication protocols. The two currently implemented protocols are explicit shared memory multi-threading (using the standard Pthreads library) and message passing (using MPI).

Both implementations have been tested, and Pthreads version has been optimized, on a variety of parallel architectures, including:

- Dell bus based symmetric multi-processor (SMP)
- Sun crossbar based SMP
- HP/Convex Exemplar cache-coherent non-uniform memory access computer (ccNUMA)
- SGI Onyx ccNUMA.

Circular Restricted 3-Body Problem

We will now apply the methods from the previous section to the gravitational Circular Restricted 3-Body Problem (CR3BP)

- Two primary masses move in circular orbits about their barycenter,
- Third body with negligible mass influenced by the primaries,
- Coordinates in barycentric frame rotating with the primaries.

$$\begin{aligned} x'' &= 2y' + x - (1-\mu)(x+\mu)r_1^{-3} - \mu(x-1+\mu)r_2^{-3}, \\ y'' &= -2x' + y - (1-\mu)yr_1^{-3} - \mu y r_2^{-3}, \\ z'' &= -(1-\mu)zr_1^{-3} - \mu z r_2^{-3}, \end{aligned}$$

$$r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}, \qquad r_2 = \sqrt{(x-1+\mu)^2 + y^2 + z^2},$$

From this set of coupled ODEs we compute fixed points and periodic orbits. μ is the ratio of the masses of the two primaries, and for the calculations that follow we use 0.01215 which is the ratio for the earth moon system.

Problem Formulation

To apply numerical continuation methods we first phrase the computation of a periodic orbit as a two-point boundary value problem by adding appropriate boundary conditions

$$\begin{array}{lll} x(1) = x(0) & y(1) = y(0) & z(1) = z(0) \\ v_x(1) = v_x(0) & v_y(1) = v_y(0) & v_z(1) = v_z(0) \end{array}$$

to impose unit periodicity and solve for the unknown period as part of the numerical continuation procedure.

Problem Formulation

To apply numerical continuation methods we first phrase the computation of a periodic orbit as a two-point boundary value problem by adding appropriate boundary conditions

$$\begin{array}{lll} x(1) = x(0) & y(1) = y(0) & z(1) = z(0) \\ v_x(1) = v_x(0) & v_y(1) = v_y(0) & v_z(1) = v_z(0) \end{array}$$

to impose unit periodicity and solve for the unknown period as part of the numerical continuation procedure.

To solve the two-point boundary value problem we do not use a shooting method, but instead discretized the system (in this case using orthogonal collocation) and use Newton's method to solve for the discretized solution.

Problem Formulation

To apply numerical continuation methods we first phrase the computation of a periodic orbit as a two-point boundary value problem by adding appropriate boundary conditions

$$\begin{array}{lll} x(1) = x(0) & y(1) = y(0) & z(1) = z(0) \\ v_x(1) = v_x(0) & v_y(1) = v_y(0) & v_z(1) = v_z(0) \end{array}$$

to impose unit periodicity and solve for the unknown period as part of the numerical continuation procedure.

To solve the two-point boundary value problem we do not use a shooting method, but instead discretized the system (in this case using orthogonal collocation) and use Newton's method to solve for the discretized solution.

Finally, we add an unfolding term to the equations to allow us to parameterize the family of periodic orbits implied by the Cylinder Theorem...

Simple Example of Unfolding Term

Consider the conservative system

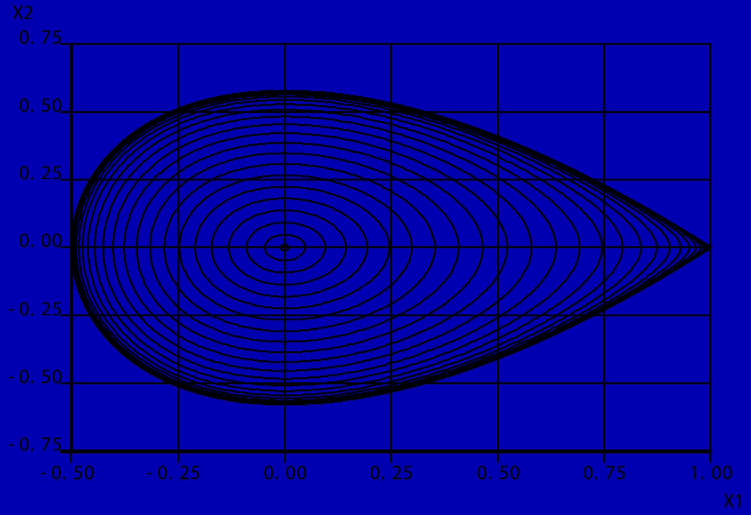
$$\begin{aligned} x_1' &= x_2, \\ x_2' &= -x_1(1-x_1). \end{aligned}$$

which has one integral, the Hamiltonian

$$H(x_1, x_2) = \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2 - \frac{1}{3}x_1^3 \ .$$

Family of Periodic Orbits, but No Parameter

This system has a family of periodic orbits, but no parameter.



Randy Paffenroth

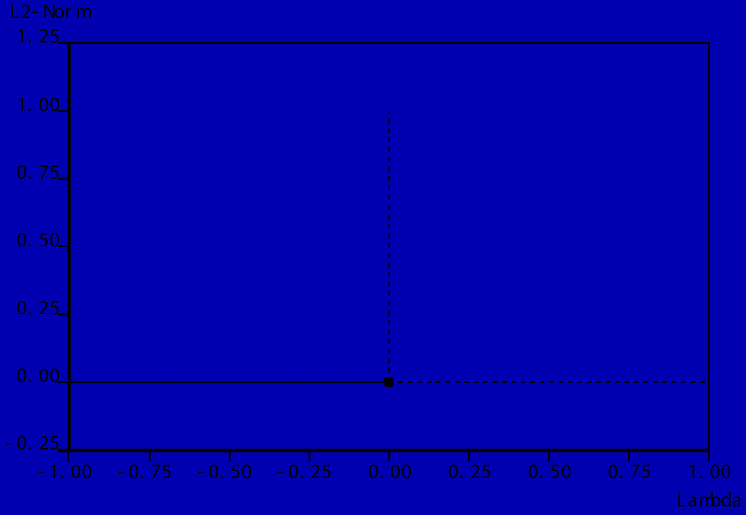
redrod@acm.org

Just Add a Parameter

The solution is to add a “damping parameter” (also called an “unfolding parameter”) which gives rise to a “vertical” Hopf bifurcation.

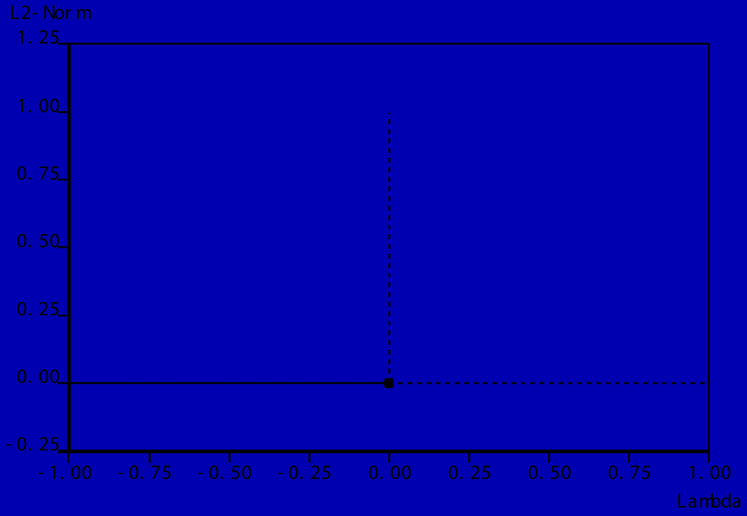
$$x_1' = x_2,$$

$$x_2' = -x_1(1 - x_1) + \lambda x_2.$$



What is the continuation parameter?

All of the CR3BP bifurcation diagrams I will show in this talk are computed by doing continuation in the “unfolding parameter”. Though it may seem a bit counter-intuitive, they are all vertical branches on which the continuation parameter is identically zero (to within numerical precision).



Fortunately, pseudo-arclength continuation has no problem computing such families of solutions.

Application to the CR3BP

The equations of motion for the CR3BP

$$\begin{aligned} x'' &= 2y' + x - (1-\mu)(x+\mu)r_1^{-3} - \mu(x-1+\mu)r_2^{-3}, \\ y'' &= -2x' + y - (1-\mu)yr_1^{-3} - \mu yr_2^{-3}, \\ z'' &= -(1-\mu)zr_1^{-3} - \mu zr_2^{-3}, \end{aligned}$$

have a single integral

$$E = (v_x^2 + v_y^2 + v_z^2)/2 - U(x,y,z) - \mu(1-\mu)/2.$$

The Equations for the CR3BP

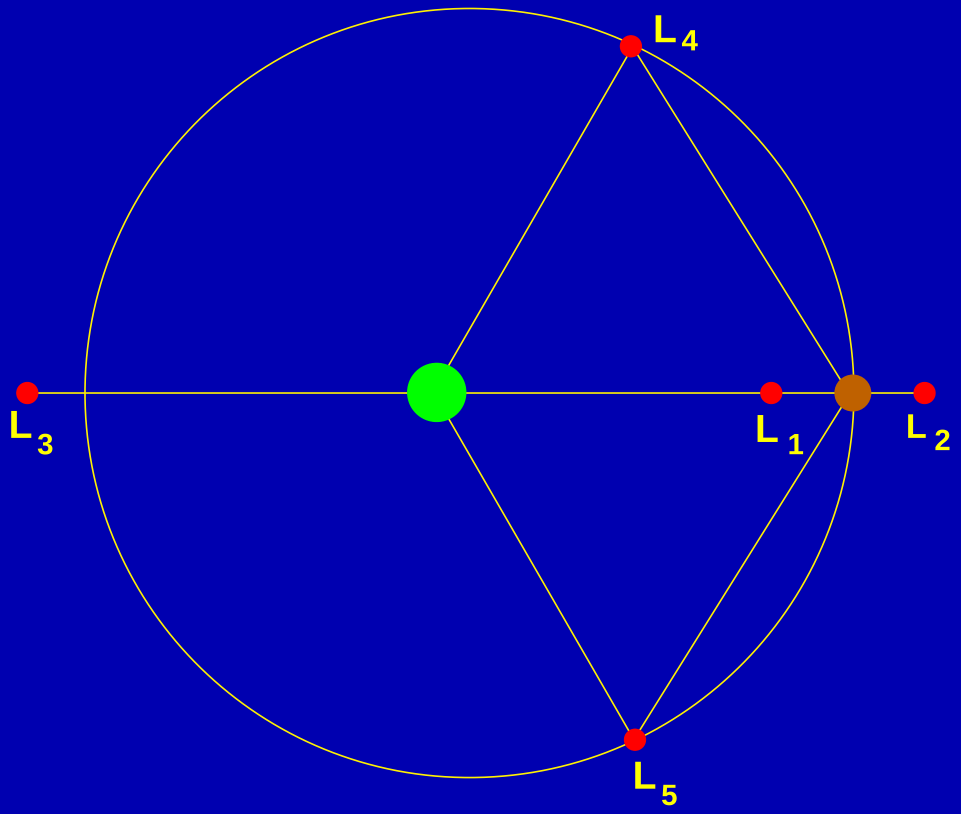
Accordingly, the equations we solve have an appropriate unfolding parameter and unfolding terms.

$$\begin{aligned}
x' &= Tv_x + \lambda E_x, \\
y' &= Tv_y + \lambda E_y, \\
z' &= Tv_z + \lambda E_z, \\
v_x' &= T[2v_y + x - (1-\mu)(x+\mu)r_1^{-3} - \mu(x-1+\mu)r_2^{-3}] + \lambda E_{v_x}, \\
v_y' &= T[-2v_x + y - (1-\mu)yr_1^{-3} - \mu y r_2^{-3}] + \lambda E_{v_y}, \\
v_z' &= T[-(1-\mu)zr_1^{-3} - \mu z r_2^{-3}] + \lambda E_{v_z}.
\end{aligned}$$

Where T is the period of the periodic solution.

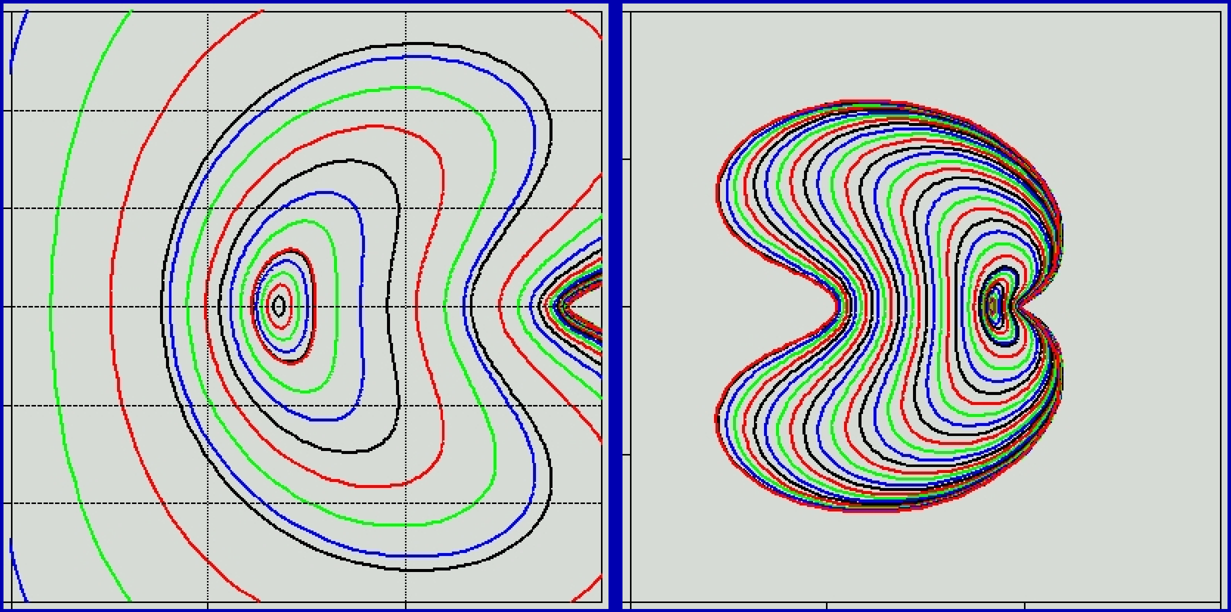
Libration Points

Our computational focus will be the periodic orbits which emanate from the so called "Libration Points".



Planar Lyapunov Orbits

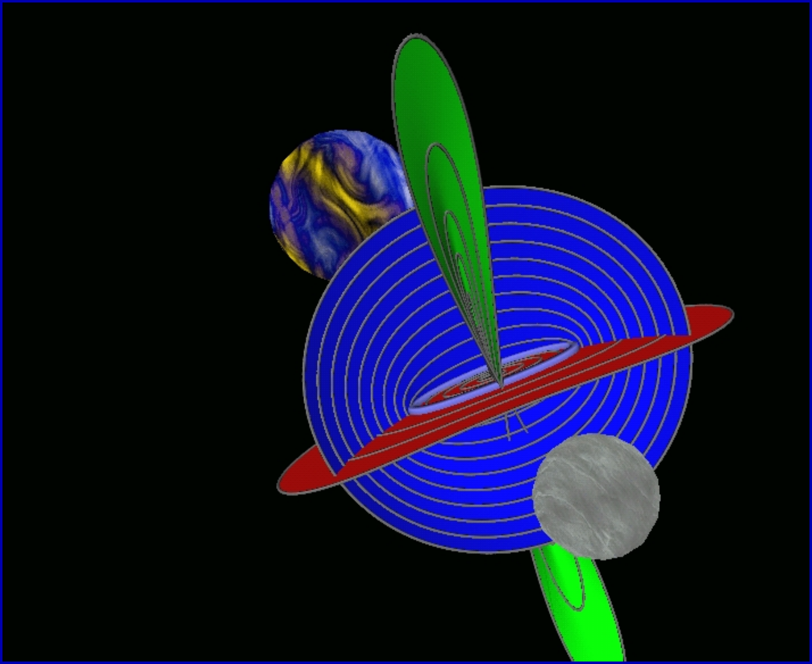
As a first example we show the planar Lyapunov orbits about L1.



The initial solution close to the L1 Hopf bifurcation is found analytically and the rest of the solutions are computed using continuation. We stop near collision, at which point the system becomes singular.

Periodic Orbits Near L1

The red plane consists of Lyapunov orbits and the green sheet consists of Vertical orbits. The blue sheet consists of Halo orbits which bifurcate from the Lyapunov orbits at the thick blue curve.



Randy Paffenroth

redrod@acm.org

Stability of Solutions

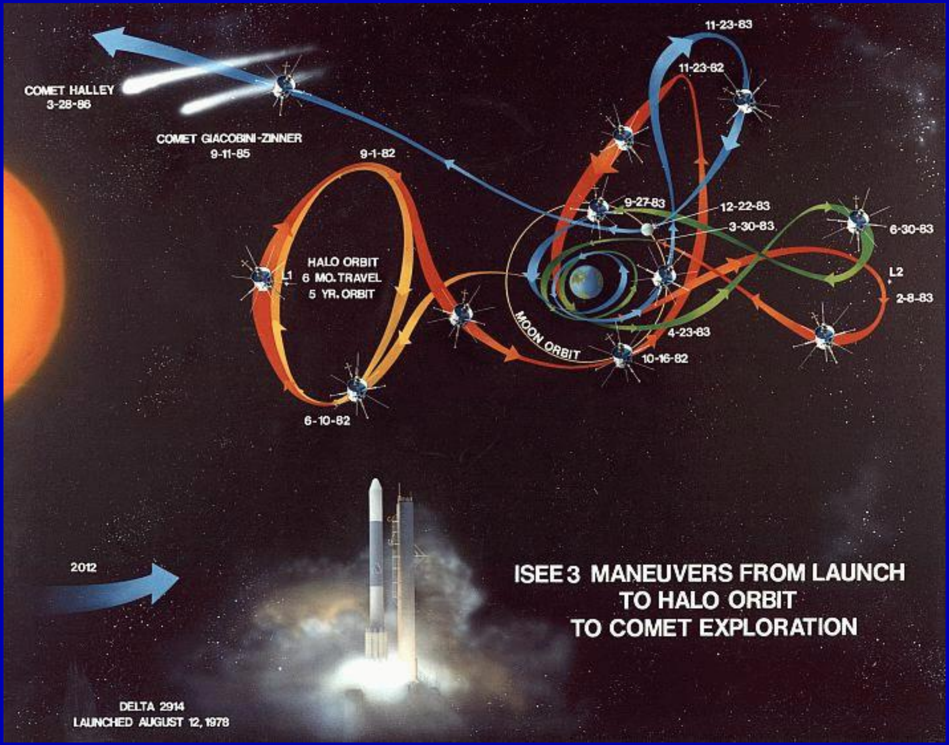
"That sounds like a neat problem, but are the solutions stable?"
– David Amundsen - On the way to the gym

"The instability is purely mathematical."
– Martin Lo, JPL - Lagrange Points and the Exploration of Space Meeting

The families of solutions that we have computed are not generally stable, but for mission planning the stability of the solutions is not paramount. The satellites have active control, and the amount of fuel required for the small corrections to stay near an unstable orbit are very small compared to the fuel to change trajectories.

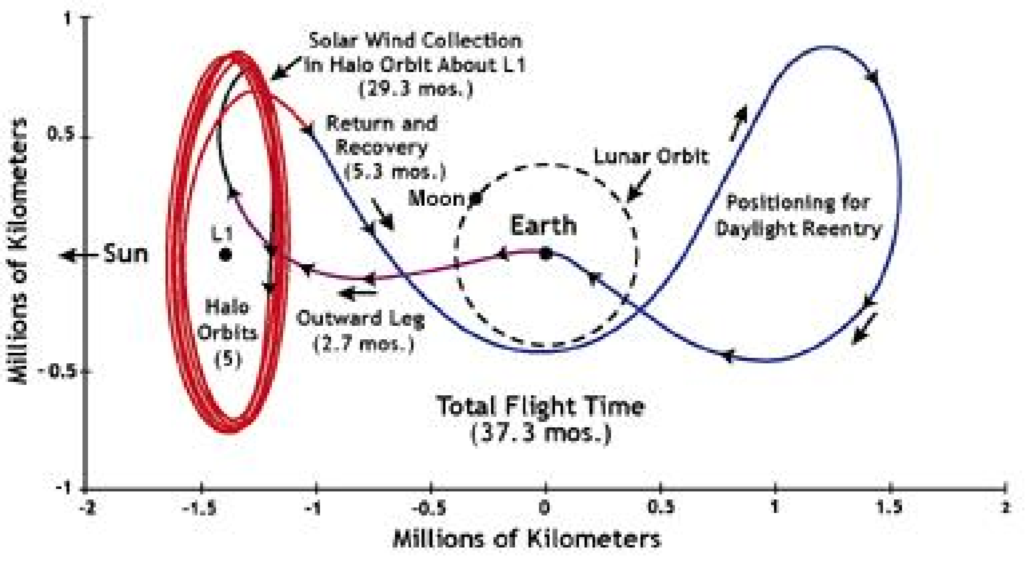
In fact, the instability of a Libration point orbit can actually be useful, since a satellite can approach it on its stable manifold and then leave it (for small fuel cost) on its unstable manifold.

ISEE3/ICE



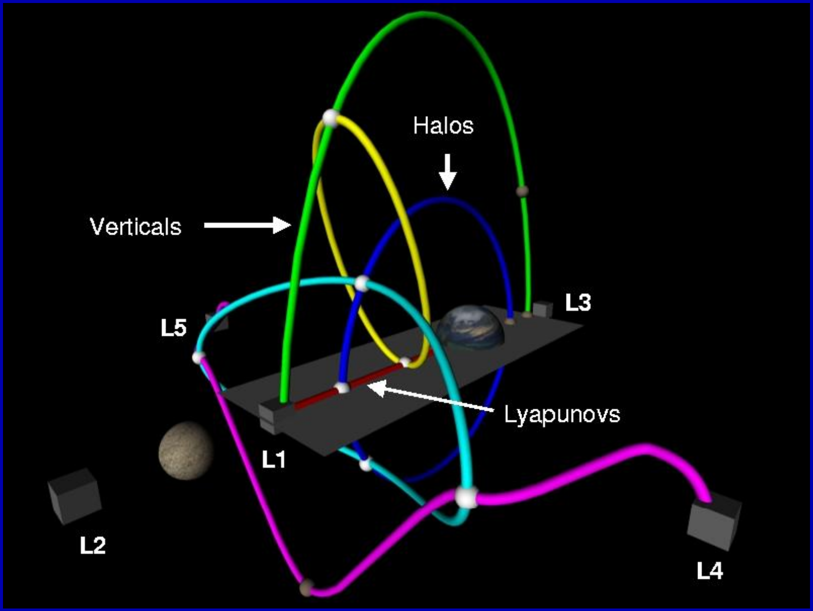
ISEE 3 MANEUVERS FROM LAUNCH
TO HALO ORBIT
TO COMET EXPLORATION

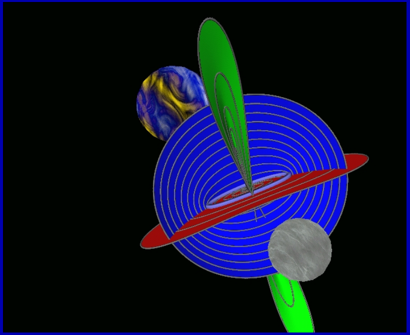
Genesis


Solar Wind Collection
In Halo Orbit About L1
(29.3 mos.)
Return and
Recovery
(5.3 mos.)
Lunar Orbit
Moon
Earth
Positioning for
Daylight Reentry
Sun
L1
Halo
Orbits
(5)
Outward Leg
(2.7 mos.)
Total Flight Time
(37.3 mos.)
Millions of Kilometers
Millions of Kilometers
1
0.5
-0.5
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Bifurcation Diagram for L1

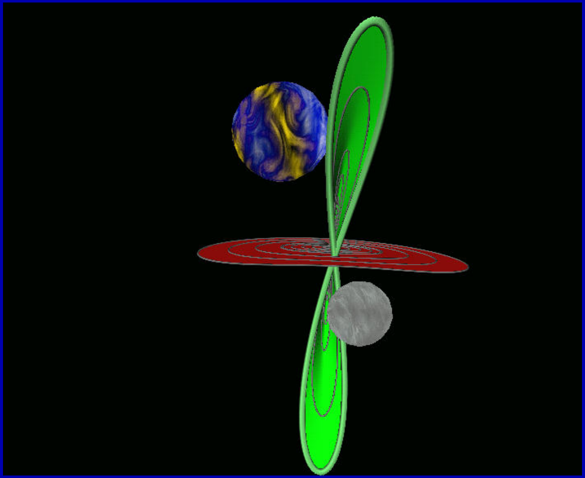
To organize the data from our computations we use a bifurcation diagram. The cubes are the Lagrange points, the spheres are simple bifurcation points, the gray plane signifies planar orbits, and the colored curves represent families of periodic orbits.

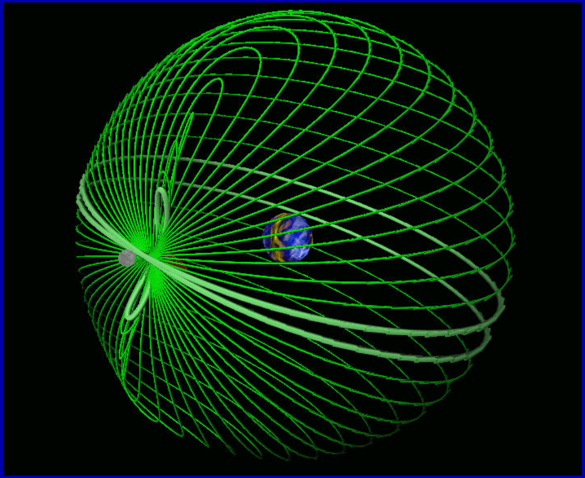


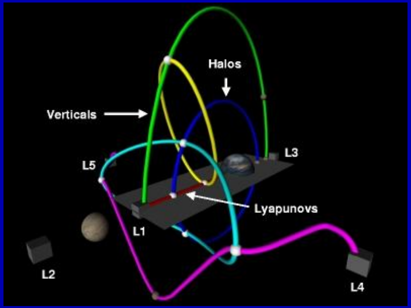


Vertical Orbits for L1

The vertical orbits for L1. These "Doubly Symmetric" orbits have been known since at least Bray and Goudas, 1966.

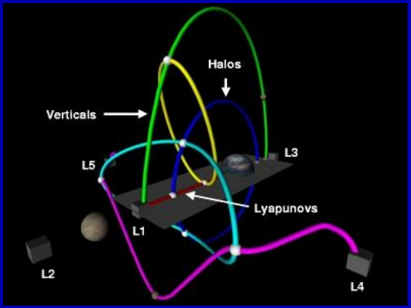


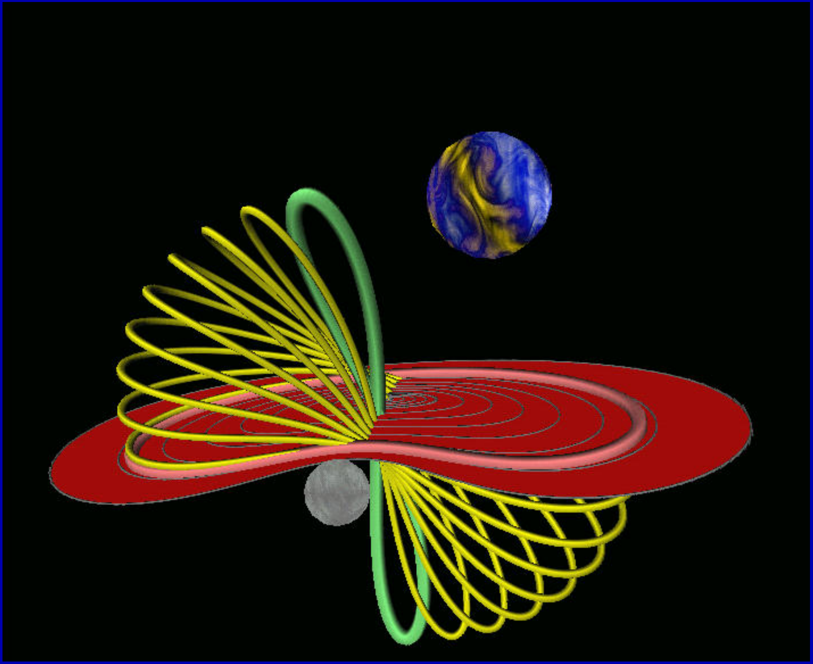




Connecting Verticals and Lyapunovs

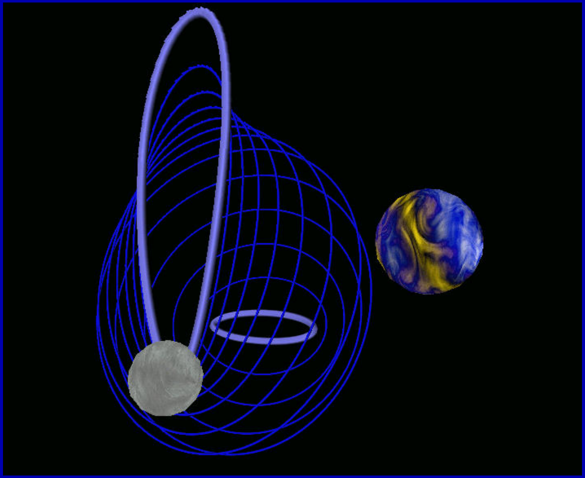
A family of orbits connecting the second bifurcation point on the Lyapunovs to the first bifurcation point on the Verticals. Similar “Axisymmetric” orbits are mentioned in Zagouras et al., 1979.

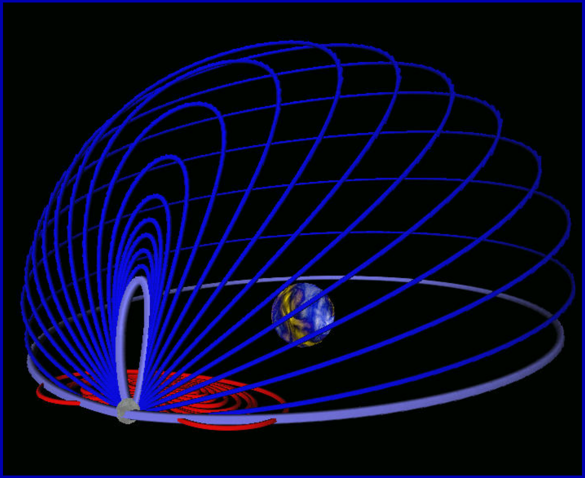


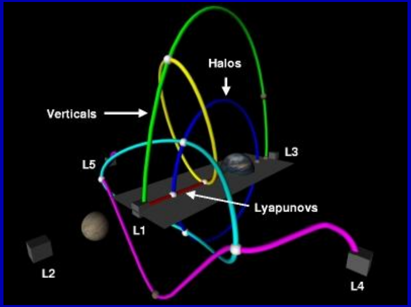


Northern Halo orbits for L1

The Northern Halo orbits for L1. Such orbits have been studied extensively by Farquhar, Howell, Simo, and others.

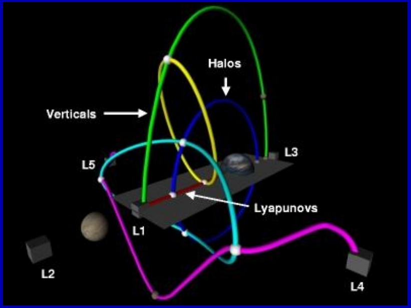


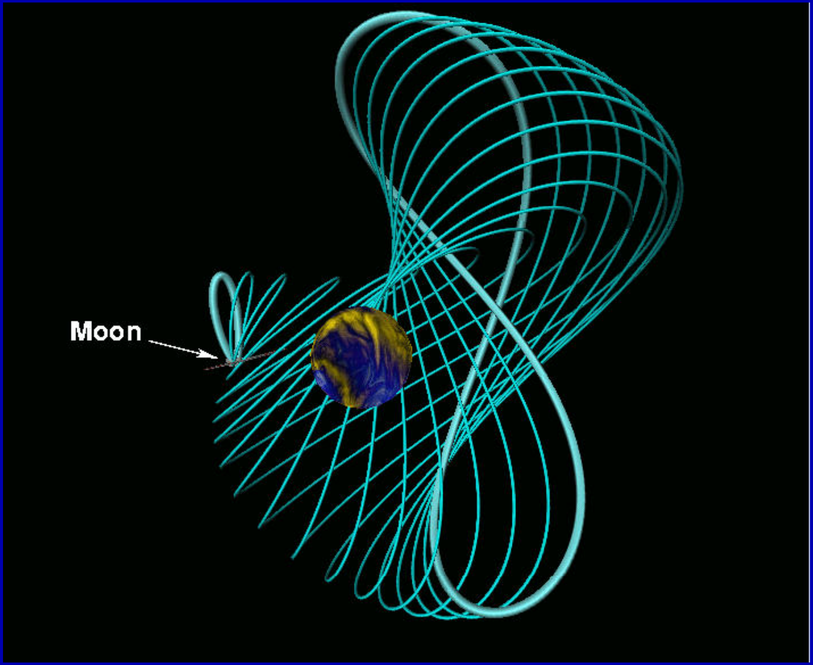




Connection between L1 and L4/L5

These orbits bifurcate from the Halo orbits.

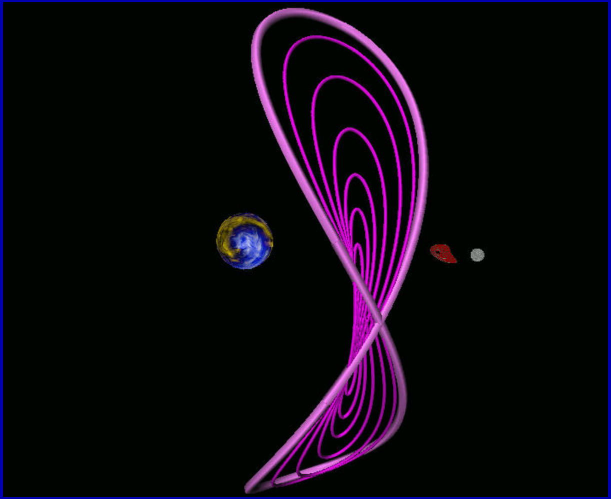


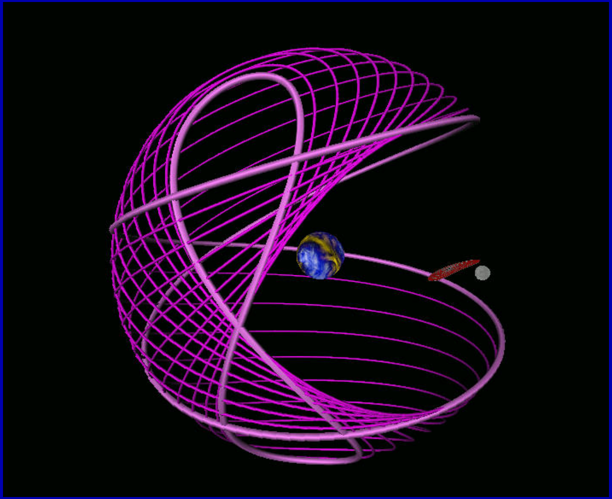


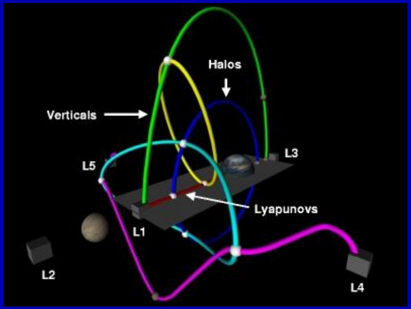
Note that these orbits are not symmetric, but that our method makes no assumptions about the symmetry properties of the solution.

Connection between L4 and L5

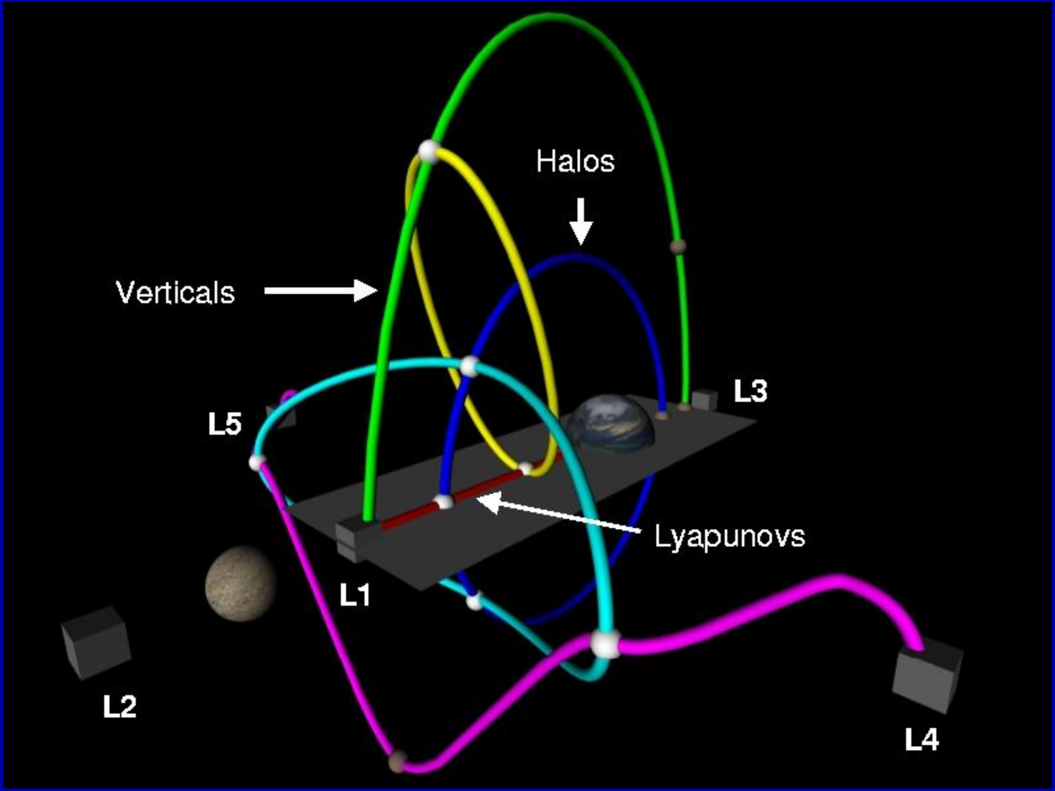
These orbits connect the Verticals of L4 and L5.







But wait, there's more!



Randy Paffenroth

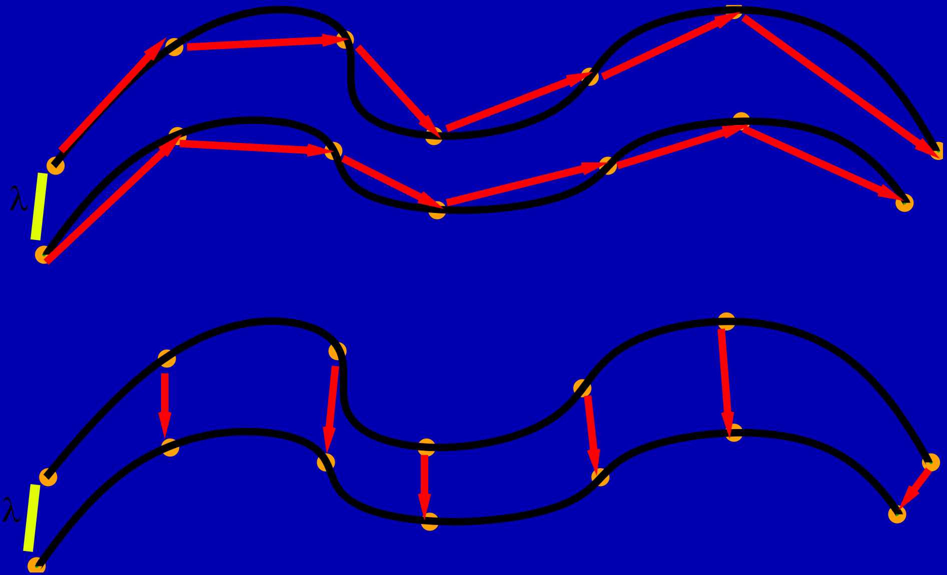
redrod@acm.org

Connecting Orbits

Using a BVP solver to solve IVPs

"When all you have is a hammer, everything begins to look like a nail." –Unknown

We will now use AUTO, a BVP solver, to solve a collection of IVPs. Now, while this may seem strange at first, it does have some advantages.



Formulation of method

These can also be computed by continuation:

$$x'(t) = Tf(x(t), \lambda), x(0) = x_0(0) + \epsilon v(0),$$

where

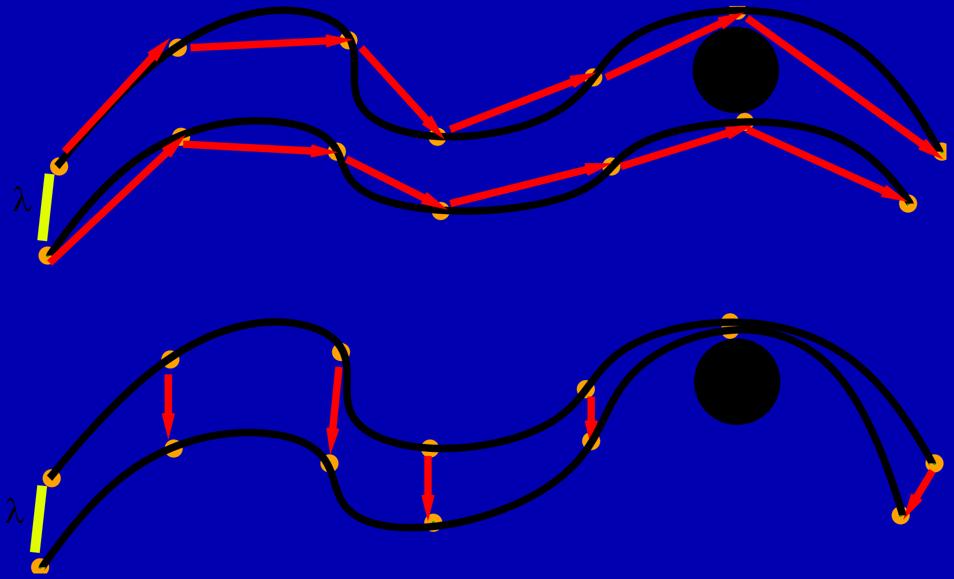
the independent t-variable has been scaled to vary in $[0, 1]$.
T is the total integration time.
x_0 is a Lyapunov orbit.
v is a Floquet eigenfunction of the Lyapunov orbit.
ϵ is the "starting-distance" into the stable or unstable manifold.

Suitable endpoint conditions (including "none") can be added.

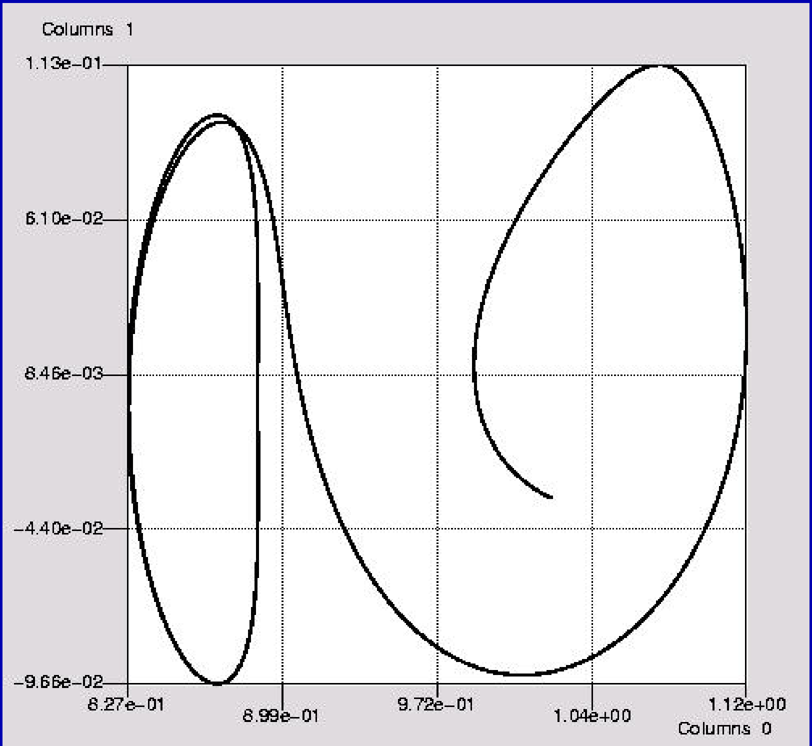
For more detail on the theory of such methods one can see, e.g.: Friedman MJ and Doedel EJ Numerical Computation and Continuation of Invariant-Manifolds Connecting Fixed-Points SIAM J Numerical Analysis 28: (3) 789-808 Jun 1991

Basic Idea of Method

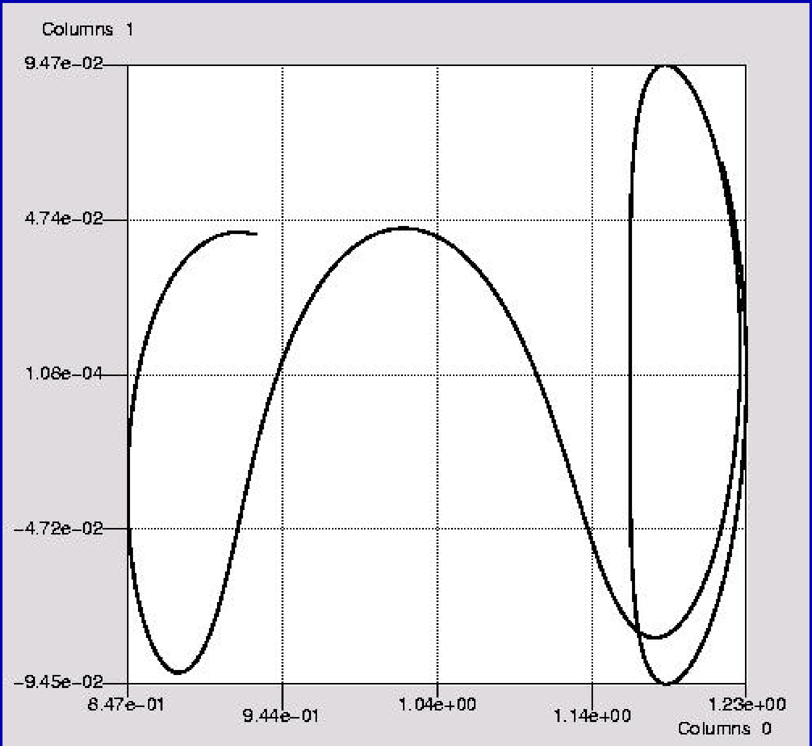
In very general terms the idea of the method is that when one uses an IVP solver to integrate a family of solutions it is easy to miss invariant sets in the stable/unstable manifold. On the other hand, when using a BVP solver (with step size adaptation) one can detect such sets based on the convergence of all points along the solution.



Homoclinic Connection from L1 to L1



Heteroclinic Connection from L2 to L1



Randy Paffenroth

redrod@acm.org

The Full 3-Body problem

We now turn our attention to the full 3-Body problem.

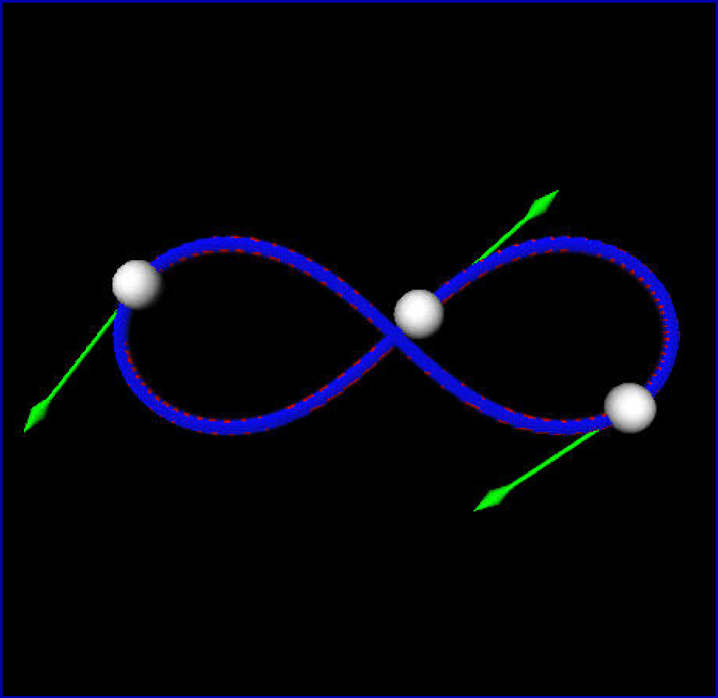
$$\begin{aligned}
x_1'' &= -m_2\tfrac{x_1-x_2}{|x_1-x_2|^3} - m_3\tfrac{x_1-x_3}{|x_1-x_3|^3},\\
x_2'' &= -m_1\tfrac{x_2-x_1}{|x_1-x_2|^3} - m_3\tfrac{x_2-x_3}{|x_2-x_3|^3},\\
x_3'' &= -m_2\tfrac{x_3-x_2}{|x_3-x_2|^3} - m_1\tfrac{x_3-x_1}{|x_1-x_3|^3}.
\end{aligned}$$

where

$$x_i \equiv \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}.$$

The Figure-eight orbit of Montgomery and Chenciner

Our goal is now to perform continuation on the figure-eight orbit of Montgomery and Chenciner, by using the mass m_1 as our continuation parameter.



The Basic Formulation

The basic problem formulation is the same as in the CR3BP case. We rewrite the system as first order, add boundary conditions to impose periodicity, and solve for the unknown period T,

$$x_1' = Tv_1, \qquad x_2' = Tv_2, \qquad x_3' = Tv_3,$$

$$\begin{aligned}
v_1' &= -T\{m_2\tfrac{x_1-x_2}{|x_1-x_2|^3} + m_3\tfrac{x_1-x_3}{|x_1-x_3|^3}\},\\
v_2' &= -T\{m_1\tfrac{x_2-x_1}{|x_1-x_2|^3} + m_3\tfrac{x_2-x_3}{|x_2-x_3|^3}\},\\
v_3' &= -T\{m_2\tfrac{x_3-x_2}{|x_3-x_2|^3} + m_1\tfrac{x_3-x_1}{|x_1-x_3|^3}\},
\end{aligned}$$

$$x_1(0) - x_1(1) = 0, \qquad x_2(0) - x_2(1) = 0, \qquad x_3(0) - x_3(1) = 0,$$

$$v_1(0) - v_1(1) = 0, \qquad v_2(0) - v_2(1) = 0, \qquad v_3(0) - v_3(1) = 0,$$

but in this case there is an important difference...

More Integrals of Motion

For the full 3-Body problem the Hamiltonian is not the only integral of motion. There are six other independent integrals of motion and associated symmetries of the solution.

First, the Center of Mass of the system is constant, which is associated with the symmetry of invariance under translations.

Second, the Angular Momentum vector of the system is fixed, which is associated with the symmetry of invariance under rotations.

The symmetries imply non-uniqueness of the solutions...

Add Constraints

Three translation constraints:

$$\sum_{i=1}^{n} \int_0^1 x_i(\tau) - \hat{x}_i(\tau)\ d\tau = 0\ .$$

Three rotations constraints:

$$\sum_{i=1}^{n} \int_0^1 [y_i(\tau) - \hat{y}_i(\tau)]z_i(\tau) - [z_i(\tau) - \hat{z}_i(\tau)]y_i(\tau)\ d\tau = 0\ .$$

$$\sum_{i=1}^{n} \int_0^1 -[x_i(\tau) - \hat{x}_i(\tau)]z_i(\tau) + [z_i(\tau) - \hat{z}_i(\tau)]x_i(\tau)\ d\tau = 0\ .$$

$$\sum_{i=1}^{n} \int_0^1 [x_i(\tau) - \hat{x}_i(\tau)]y_i(\tau) - [y_i(\tau) - \hat{y}_i(\tau)]x_i(\tau)\ d\tau = 0\ .$$

Too many equations

If we add the above constraints to the full 3-Body problem, we end up with a system of 18 unknowns with 24 constraints (18 boundary conditions to enforce periodicity and 6 integral constraints to factor out the symmetries in translation and rotation).

We don't have enough unknowns!

Add Unfolding Terms

The solution is very similar to the one we used in the case where we had but one integral, we add unfolding terms.

$$x_1' = Tv_1, \qquad x_2' = Tv_2, \qquad x_3' = Tv_3,$$

$$v_1' = -T\{m_2 \frac{x_1 - x_2}{|x_1 - x_2|^3} + m_3 \frac{x_1 - x_3}{|x_1 - x_3|^3}\} \ + \ G(x_1, v_1, \lambda),$$

$$v_2' = -T\{m_1 \frac{x_2 - x_1}{|x_1 - x_2|^3} + m_3 \frac{x_2 - x_3}{|x_2 - x_3|^3}\} \ + \ G(x_2, v_2, \lambda),$$

$$v_3' = -T\{m_2 \frac{x_3 - x_2}{|x_3 - x_2|^3} + m_1 \frac{x_3 - x_1}{|x_1 - x_3|^3}\} \ + \ G(x_3, v_3, \lambda),$$

where

$$G(x, v, \lambda) = \lambda_1 v + \begin{pmatrix} \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{pmatrix} + \lambda_5 \begin{pmatrix} 0 \\ z \\ -y \end{pmatrix} + \lambda_6 \begin{pmatrix} -z \\ 0 \\ x \end{pmatrix} + \lambda_7 \begin{pmatrix} y \\ -x \\ 0 \end{pmatrix} \ .$$

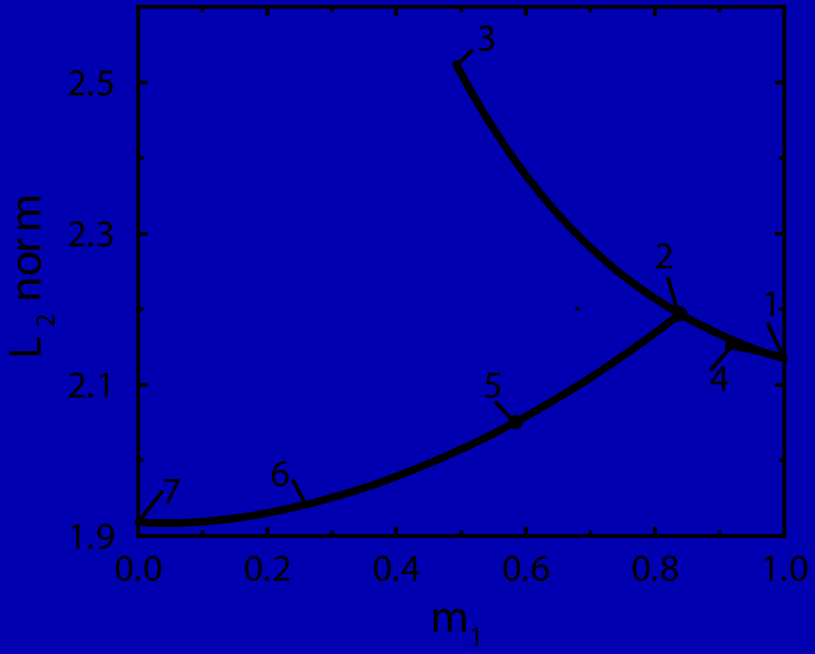
Comments

- The equations have been written as a first order system.
- The independent time-variable t has been scaled.
- Periodic boundary conditions (period 1) are added.

Special (for this conservative system):

- The unfolding term $G(x, v, \lambda)$ regularizes the continuation.
- Integral constraints have been added to factor out the symmetries.
- $\lambda \in \mathbb{R}^7$ will be "zero", once solved for.
- The period T is fixed.

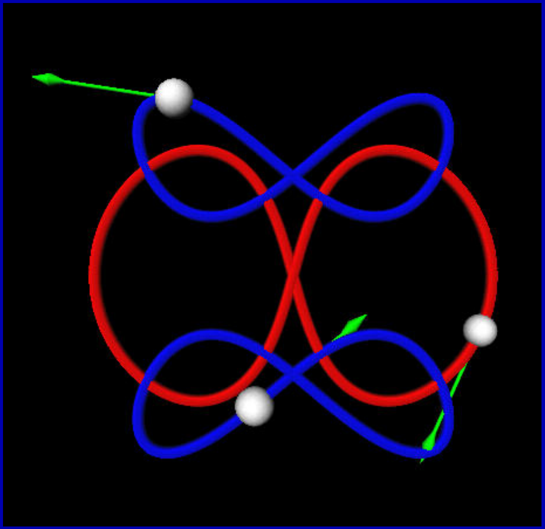
Bifurcation Diagram

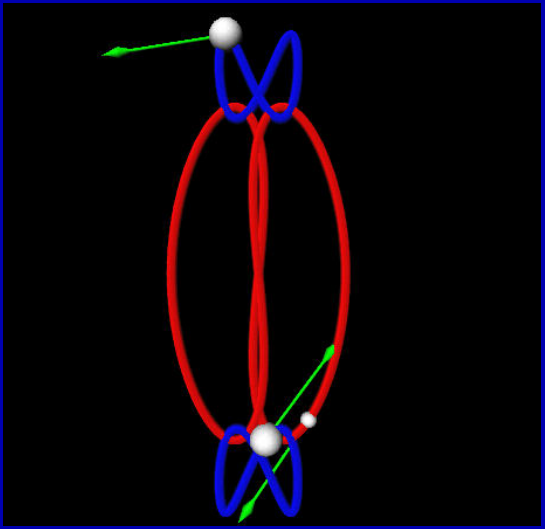


Randy Paffenroth

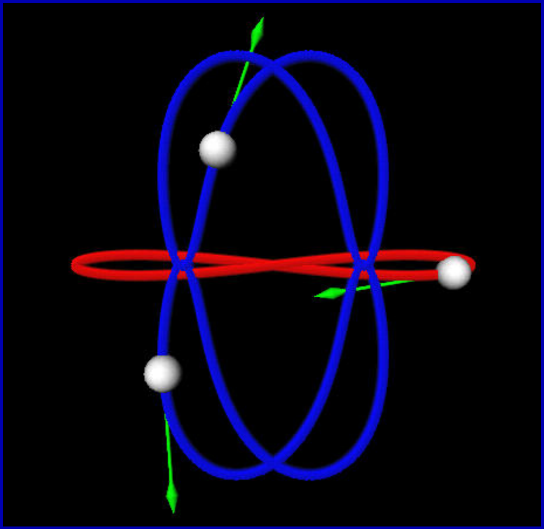
redrod@acm.org

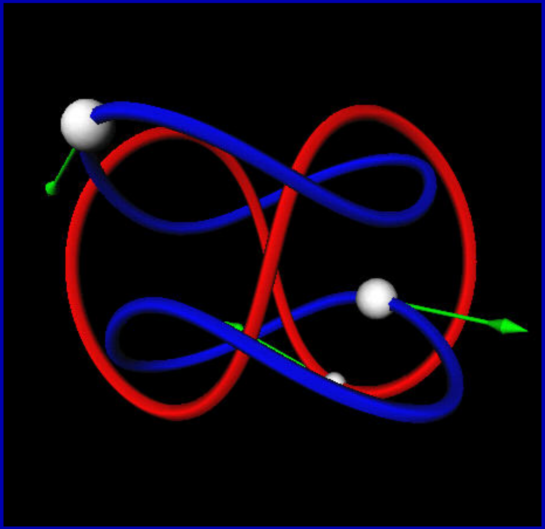
Solutions 1



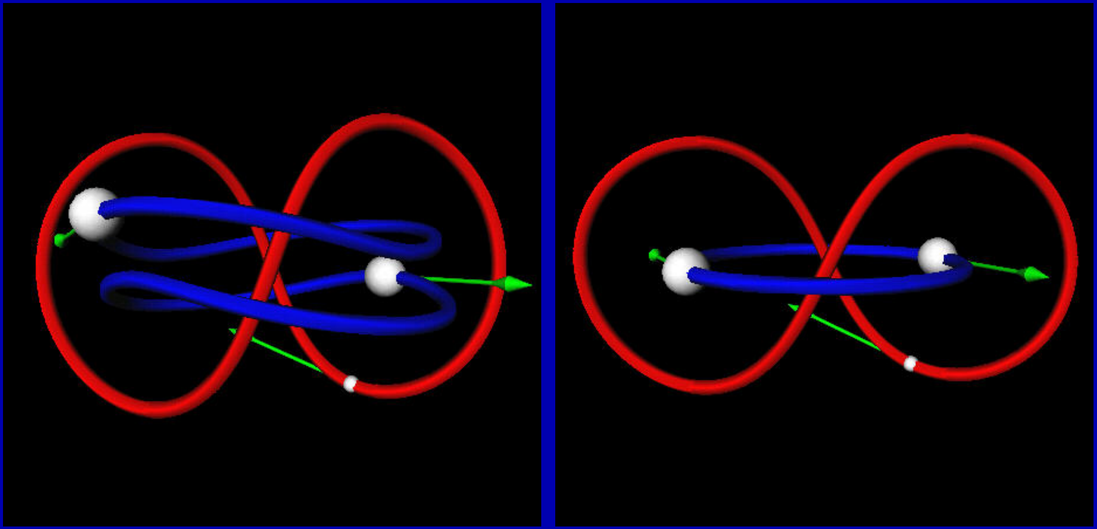


Solutions 2





Solutions 3



Conclusions and future work

- The AUTO2000 continuation procedure is robust and can easily be automated using the Python based command language. In fact, the whole calculation shown in the previous slide can be done in batch mode.
- We would be very interested in taking the data we compute for the CR3BP and putting it into a high-fidelity model to check how the "stranger" orbits perturb.
- We are planning on doing continuation in μ to follow the various bifurcation points and gain an understanding of the form of the bifurcation diagram for all μ values.
- AUTO2000 is freely available from http://auto2000.sourceforge.net

Acknowledgments

I would like to thank Christine Thorn for generating several of the (nicer) figures in this talk.

I would especially like to thank Herb Keller for initially getting us interested in this problem and for his support of this work.

In addition, this work has been supported by the National Science Foundation through the grant KDI/NCC Molecular information and computer modeling in electrophysiology, SBR-9873173.

I would also like to thank Oscar Bruno, California Institute of Technology, for his current support of this work.

The End

Typical Problem Sizes

The size of dimension n can vary greatly, depending on the kind of problem:

- 10^1 to 10^2 for intrinsically discrete, algebraic problems
- 10^2 **to** 10^3 **for discretizations of ordinary differential equation boundary value problems**
- 10^3 to 10^k for discretizations of partial differential equation boundary value problems