UNIVERSITY OF TORONTO

CPS | CENTRE FOR PLANETARY SCIENCES UNIVERSITY OF TORONTO

Some recent work on long term integrations of planetary systems

Hanno Rein, Dan Tamayo, Garett Brown

Conclusions

Determining the stability of planetary systems is a very old problem. Analytic solution cannot answer all question.

Our new machine-learning classifier SPOCK can accurately predict the stability of compact planetary systems over billions of years in seconds (10^5 times faster than direct N-body).

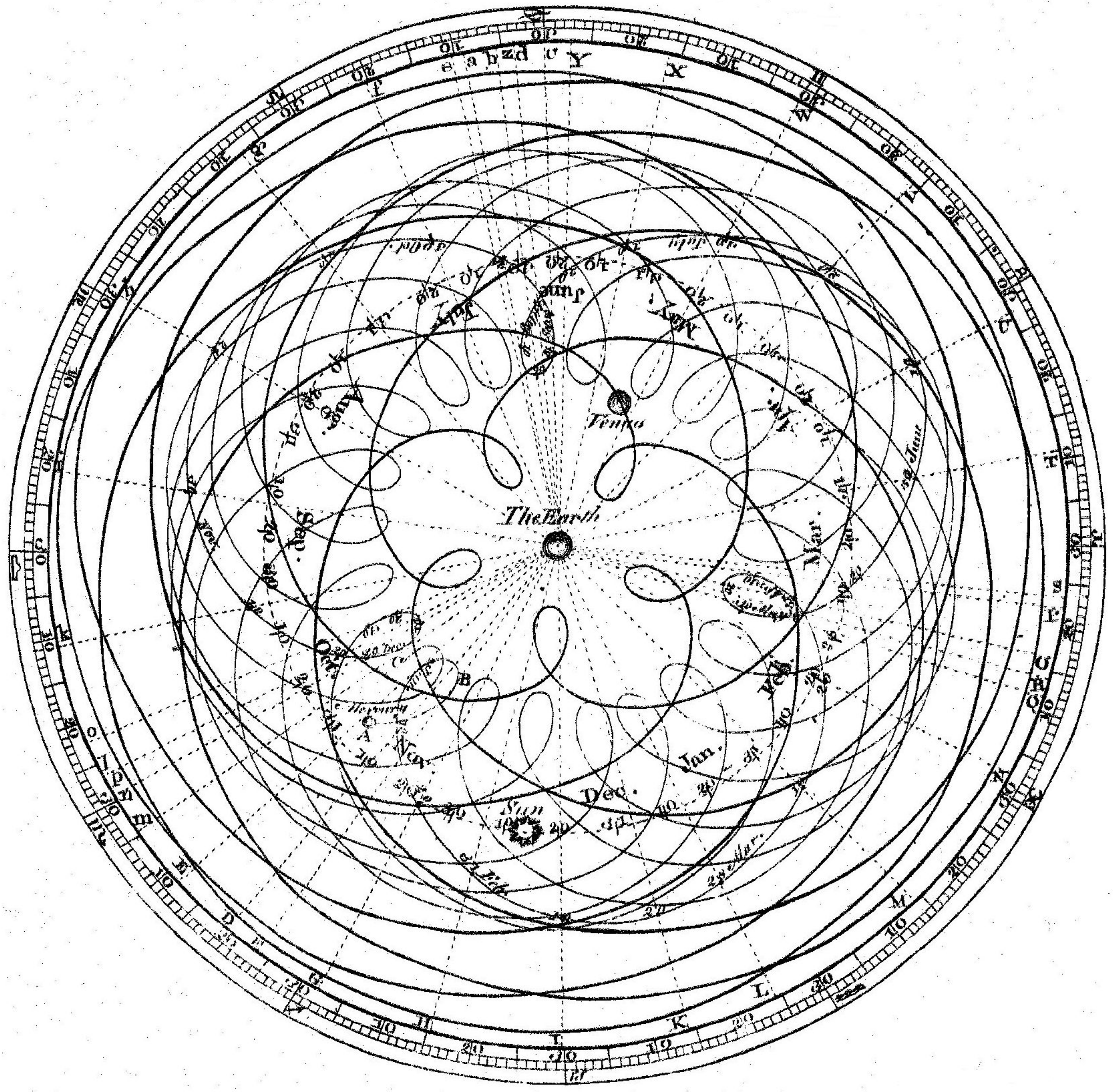
Workhorse for numerical N-body simulations are operator splitting methods, especially the Wisdom-Holman integrator.

Embedded Operator Splitting methods (EOS) are very easy to implement as they do not require a Kepler solver. An EOS method can be configured to be equivalent to: leap-frog, Wisdom-Holman, Mercury, SYMBA, and many new methods.

The History of the N-body problem (Solar System)

Laskar (*Lagrange et la stabilité du Systéme solaire*, 2006), Laskar (2013)

Epicycles


The Earth
Venus
Sun
Mercury
Jan.
Feb.
Mar.
Ap.
May.
June
July
Aug.
Sep.
Oct.
Nov.
Dec.

Newton (1687)

$$\ddot{\mathbf{r}}_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} m_j \frac{\mathbf{r}_j - \mathbf{r}_i}{\left|\mathbf{r}_j - \mathbf{r}_i\right|^3}$$

Newton (Opticks 1717,1730)

And to show that I do not take Gravity for an essential Property of Bodies, I have added one Question concerning its Cause, chosing to propose it by way of a Question, because I am not yet satisfied about it for want of Experiments. […]

For while comets move in very excentrick orbs in all manner of positions, blind fate could never make all the planets move one and the same way in orbs concentrick, some inconsiderable irregularities excepted, which may have risen from the mutual actions of comets and planets upon one another, and which will be apt to increase, till this system wants a reformation.

Evidence for irregularity/instability

On March 1st, 228 BC, at 4:23 am, mean Paris time, Saturn was observed two fingers under Gamma in Virgo.

+

Observations from 1590 and 1650.

=

Six million years ago Jupiter and Saturn were at the same distance from the Sun.

Explanations for the irregularities?

Euler was twice awarded a prize in 1748 and 1752 related to this problem by the Paris Academy of Sciences.

Lagrange thought that Euler’s calculations were wrong and did his own.

Laplace (1776)

Mr. Euler, in his second piece on the irregularities of Jupiter and Saturn, find it equal for both these planets. According to Mr. de Lagrange, on the contrary, […] it is very different for these two bodies. […] I have some reasons to believe, however, that the formula is still not accurate. The one which I obtain is quite different. […] by substituting these values in the formula of the secular equation, I found absolutely zero, from which I conclude the alteration of the mean motion of Jupiter, if it exists, does not result from the action of Saturn.

Lagrange (in a letter to d’Alembert, 1775)

I am ready to give a complete theory for the variations of the elements of the planets under their mutual action. That Mr. de la Place did on this subject I liked, and I flatter myself that he will not be offended if I do not hold the kind of promise that I made to completely abandon this subject to him; I could not resist to the desire to look into it again, but I am no less charmed that he is also working on it on his side; I am even very eager to read his subsequent research on this topic, but I do ask him not to send me any manuscript and send them to me only in printed form; I would be obliged that you tell him, with a thousand compliments from my side.

Secular Dynamics, Lagrange (1774)

1) Averaging over short time scales.

2) Perturbation theory.

No semi-major axis changes to first and also second order (Poisson, Haretu and Poincaré) in the expansion.

This still contradicts Ptolemy's observations from antiquity.

	Lagrange	**Brown & Rein** **(in prep)**
s_1	5.98	5.59
s_2	6.31	7.05
s_3	19.80	18.84
s_4	18.31	17.74
s_5	0	0
s_6	25.34	26.35
s_7	-	2.99
s_8	-	0.69

Laplace (1785)

Simple energy argument implies:

$$\frac{m_J}{a_J} + \frac{m_S}{a_S} = const$$

Thus, can be confident that the change in orbits must be due to mutual interactions.

He’s also shown, no secular terms. Hence must be short period.

Near 5:2 mean motion resonance. Period of 900 years.

Why is 900 years important?

Demo

Stability

Le Verrier (1840, 1841)

Follows up on the work of Lagrange and Laplace

Goes to higher order.

Discovered small divisor problem: third order could be larger than second order terms

Poincaré (1897)

The terms of these series, in fact, decrease first very quickly and then begin to grow, but as the Astronomers's stop after the first terms of the series, and well before these terms have stop to decrease, the approximation is sufficient for the practical use. The divergence of these expansions would have some disadvantages only if one wanted to use them to rigorously establish some specific results, as the stability of the Solar System.

Kolmogorov (1954), Arnold (1963), Moser (1962)

Kolmogorov showed that convergent perturbation series exists

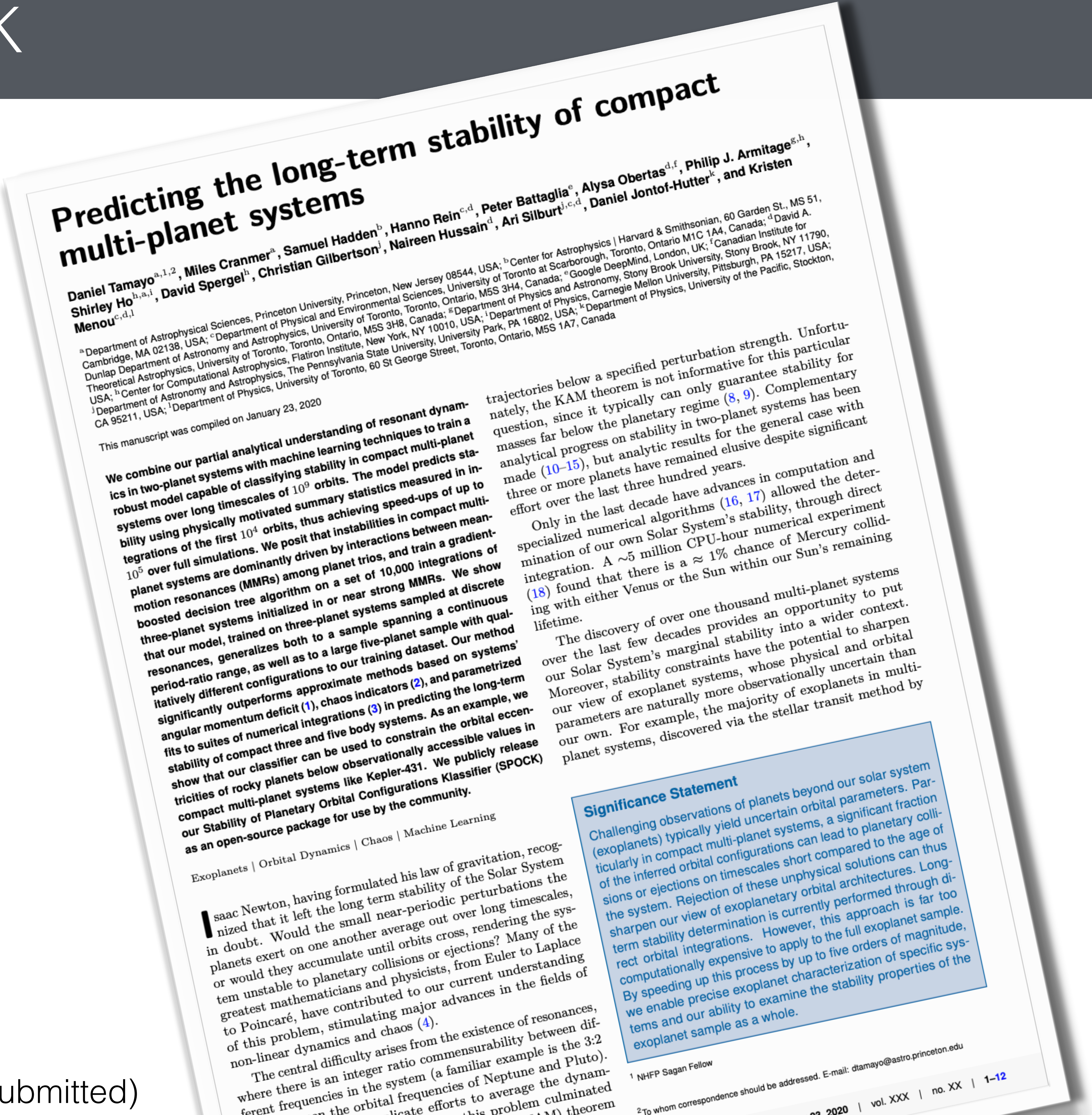
Skipping many subtleties (degeneracy, small masses, slow Arnold diffusion)

In short: not useful for determining the stability of our Solar System with a very specific set of initial conditions

Predicting the stability of planetary systems with machine learning

Tamayo et al. (submitted)

SPOCK



Predicting the long-term stability of compact multi-planet systems

Daniel Tamayo[a,1,2], Miles Cranmer[a], Samuel Hadden[b], Hanno Rein[c,d], Peter Battaglia[e], Alysa Obertas[d,f], Philip J. Armitage[g,h], Shirley Ho[h,a,i], David Spergel[h], Christian Gilbertson[j], Naireen Hussain[d], Ari Silburt[j,c,d], Daniel Jontof-Hutter[k], and Kristen Menou[c,d,l]

[a]Department of Astrophysical Sciences, Princeton University, Princeton, New Jersey 08544, USA; [b]Center for Astrophysics | Harvard & Smithsonian, 60 Garden St., MS 51, Cambridge, MA 02138, USA; [c]Department of Physical and Environmental Sciences, University of Toronto at Scarborough, Toronto, Ontario M1C 1A4, Canada; [d]David A. Dunlap Department of Astronomy and Astrophysics, University of Toronto, Toronto, Ontario, M5S 3H4, Canada; [e]Google DeepMind, London, UK; [f]Canadian Institute for Theoretical Astrophysics, University of Toronto, Toronto, Ontario, M5S 3H8, Canada; [g]Department of Physics and Astronomy, Stony Brook University, Stony Brook, NY 11790, USA; [h]Center for Computational Astrophysics, Flatiron Institute, New York, NY 10010, USA; [i]Department of Physics, Carnegie Mellon University, Pittsburgh, PA 15217, USA; [j]Department of Astronomy and Astrophysics, The Pennsylvania State University, University Park, PA 16802, USA; [k]Department of Physics, University of the Pacific, Stockton, CA 95211, USA; [l]Department of Physics, University of Toronto, 60 St George Street, Toronto, Ontario, M5S 1A7, Canada

This manuscript was compiled on January 23, 2020

We combine our partial analytical understanding of resonant dynamics in two-planet systems with machine learning techniques to train a robust model capable of classifying stability in compact multi-planet systems over long timescales of 10^9 orbits. The model predicts stability using physically motivated summary statistics measured in integrations of the first 10^4 orbits, thus achieving speed-ups of up to 10^5 over full simulations. We posit that instabilities in compact multi-planet systems are dominantly driven by interactions between mean-motion resonances (MMRs) among planet trios, and train a gradient-boosted decision tree algorithm on a set of 10,000 integrations of three-planet systems initialized in or near strong MMRs. We show that our model, trained on three-planet systems sampled at discrete resonances, generalizes both to a sample spanning a continuous period-ratio range, as well as to a large five-planet sample with qualitatively different configurations to our training dataset. Our method significantly outperforms approximate methods based on systems' angular momentum deficit (1), chaos indicators (2), and parametrized fits to suites of numerical integrations (3) in predicting the long-term stability of compact three and five body systems. As an example, we show that our classifier can be used to constrain the orbital eccentricities of rocky planets below observationally accessible values in compact multi-planet systems like Kepler-431. We publicly release our Stability of Planetary Orbital Configurations Klassifier (SPOCK) as an open-source package for use by the community.

Exoplanets | Orbital Dynamics | Chaos | Machine Learning

Isaac Newton, having formulated his law of gravitation, recognized that it left the long term stability of the Solar System in doubt. Would the small near-periodic perturbations the planets exert on one another average out over long timescales, or would they accumulate until orbits cross, rendering the system unstable to planetary collisions or ejections? Many of the greatest mathematicians and physicists, from Euler to Laplace to Poincaré, have contributed to our current understanding of this problem, stimulating major advances in the fields of non-linear dynamics and chaos (4).

The central difficulty arises from the existence of resonances, where there is an integer ratio commensurability between different frequencies in the system (a familiar example is the 3:2 n the orbital frequencies of Neptune and Pluto). licate efforts to average the dynam- this problem culminated AM) theorem

trajectories below a specified perturbation strength. Unfortunately, the KAM theorem is not informative for this particular question, since it typically can only guarantee stability for masses far below the planetary regime (8, 9). Complementary analytical progress on stability in two-planet systems has been made (10–15), but analytic results for the general case with three or more planets have remained elusive despite significant effort over the last three hundred years.

Only in the last decade have advances in computation and specialized numerical algorithms (16, 17) allowed the determination of our own Solar System's stability, through direct integration. A ~5 million CPU-hour numerical experiment (18) found that there is a ≈ 1% chance of Mercury colliding with either Venus or the Sun within our Sun's remaining lifetime.

The discovery of over one thousand multi-planet systems over the last few decades provides an opportunity to put our Solar System's marginal stability into a wider context. Moreover, stability constraints have the potential to sharpen our view of exoplanet systems, whose physical and orbital parameters are naturally more observationally uncertain than our own. For example, the majority of exoplanets in multi-planet systems, discovered via the stellar transit method by

Significance Statement

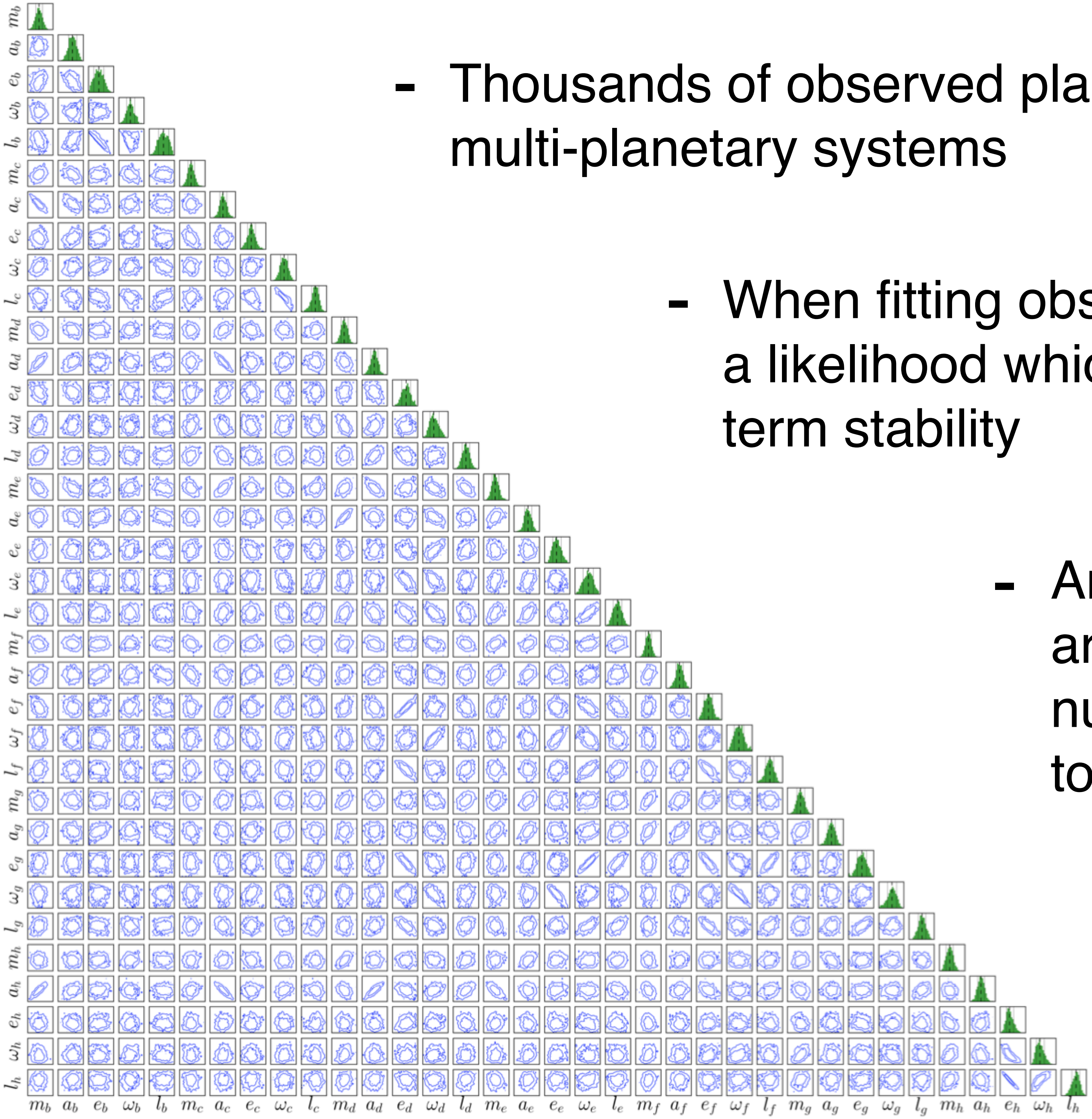
Challenging observations of planets beyond our solar system (exoplanets) typically yield uncertain orbital parameters. Particularly in compact multi-planet systems, a significant fraction of the inferred orbital configurations can lead to planetary collisions or ejections on timescales short compared to the age of the system. Rejection of these unphysical solutions can thus sharpen our view of exoplanetary orbital architectures. Long-term stability determination is currently performed through direct orbital integrations. However, this approach is far too computationally expensive to apply to the full exoplanet sample. By speeding up this process by up to five orders of magnitude, we enable precise exoplanet characterization of specific systems and our ability to examine the stability properties of the exoplanet sample as a whole.

[1] NHFP Sagan Fellow

[2] To whom correspondence should be addressed. E-mail: dtamayo@astro.princeton.edu

Tamayo et al. (submitted)

Applications



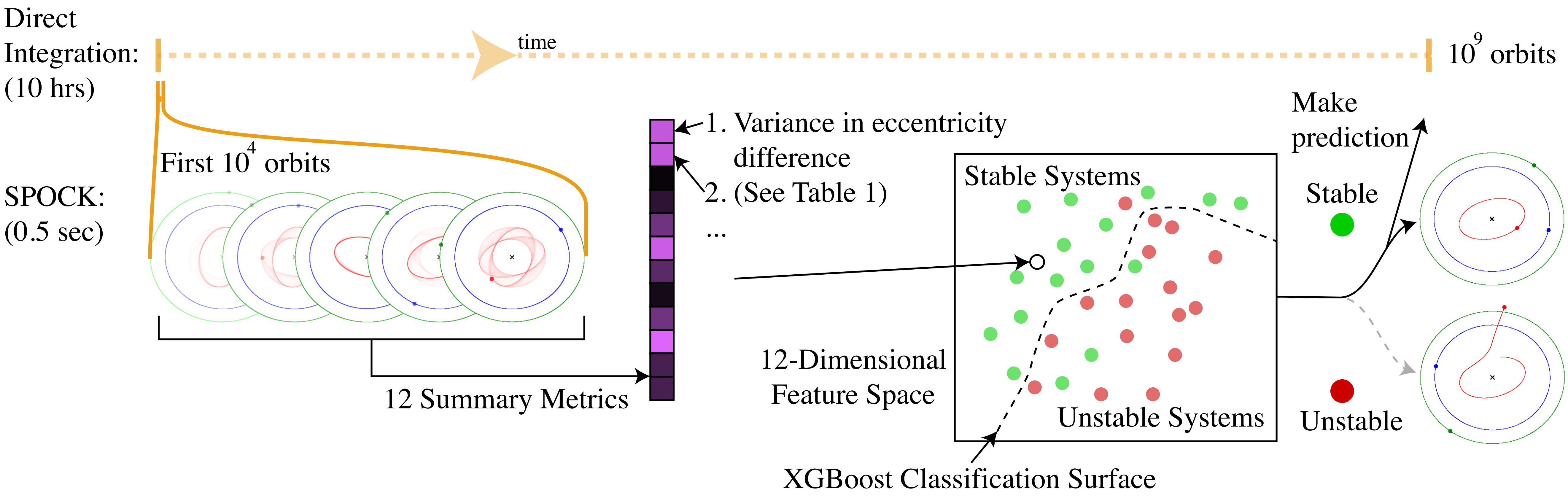
Corner plot of Trappist-1 posterior, Grimm et al (2018)

- Thousands of observed planets, many in multi-planetary systems

- When fitting observations, should use a likelihood which incorporates long term stability

- Analytical estimates alone are not good enough, direct numerical simulation way too slow

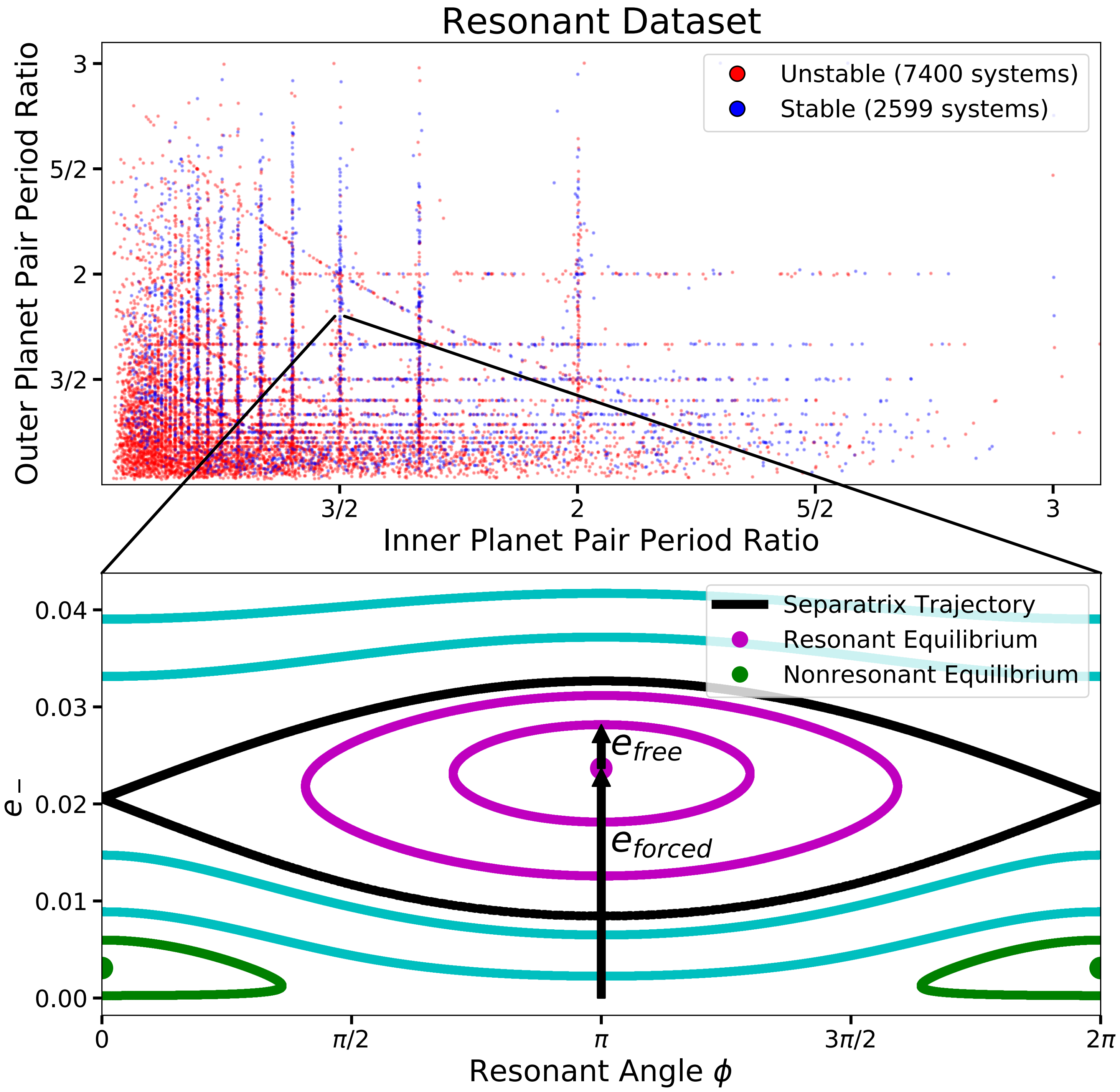
SPOCK



Tamayo et al. (submitted)

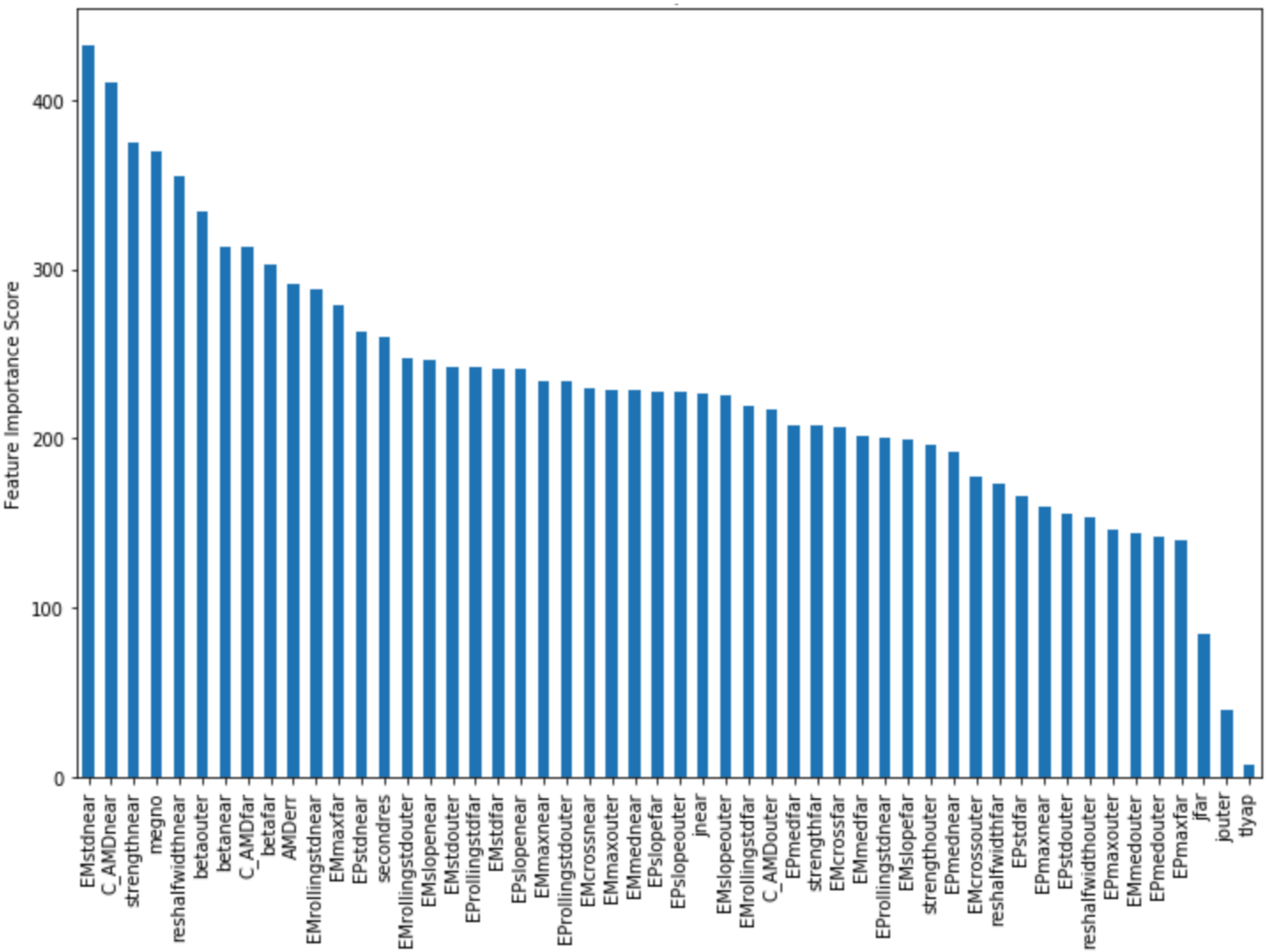
Training dataset

- Training dataset is not random
- Many systems in or near mean motion resonances
- Analytic framework to setup training dataset
- Also use analytic framework to calculate features



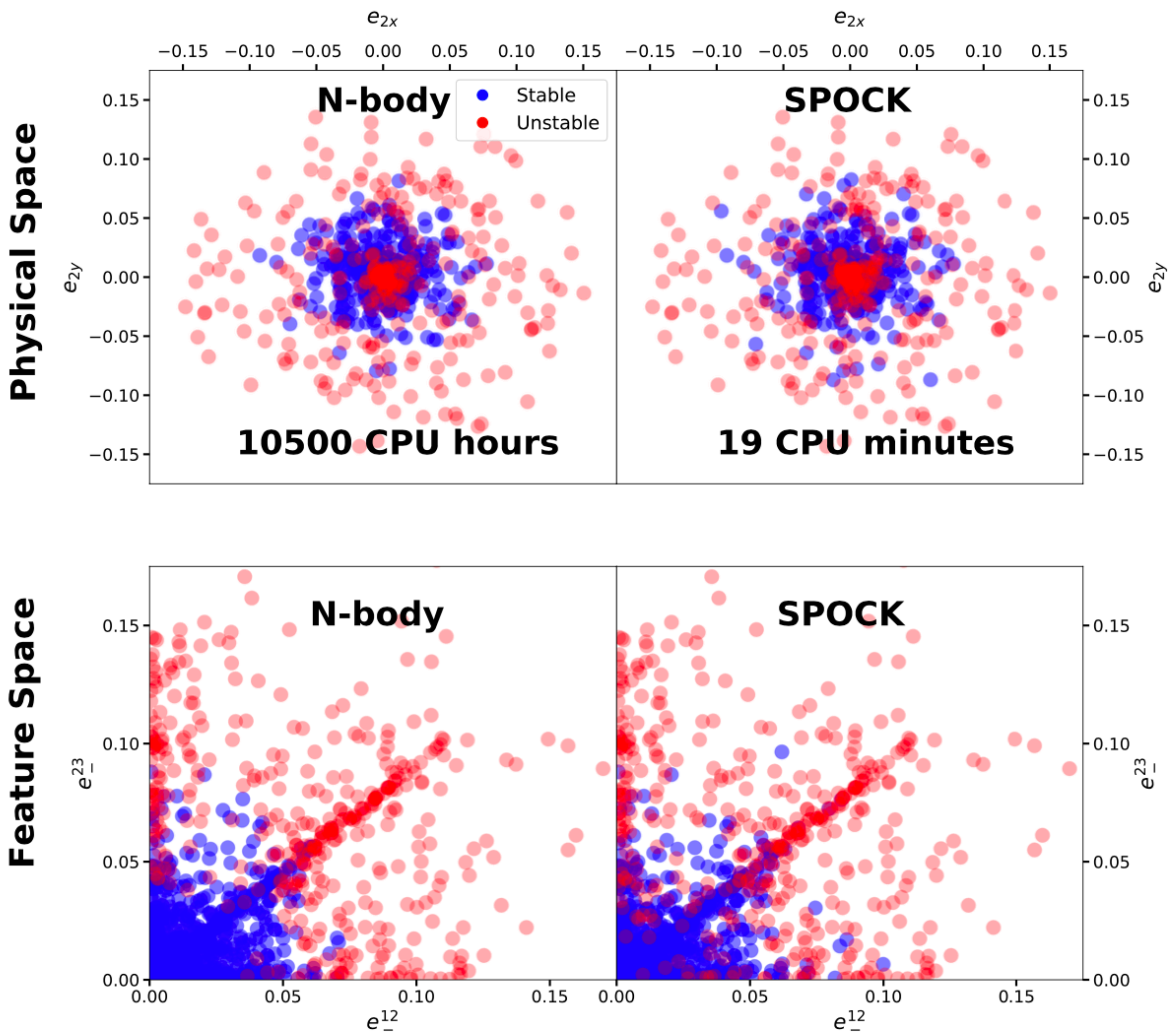
Tamayo et al. (submitted)

Feature importance



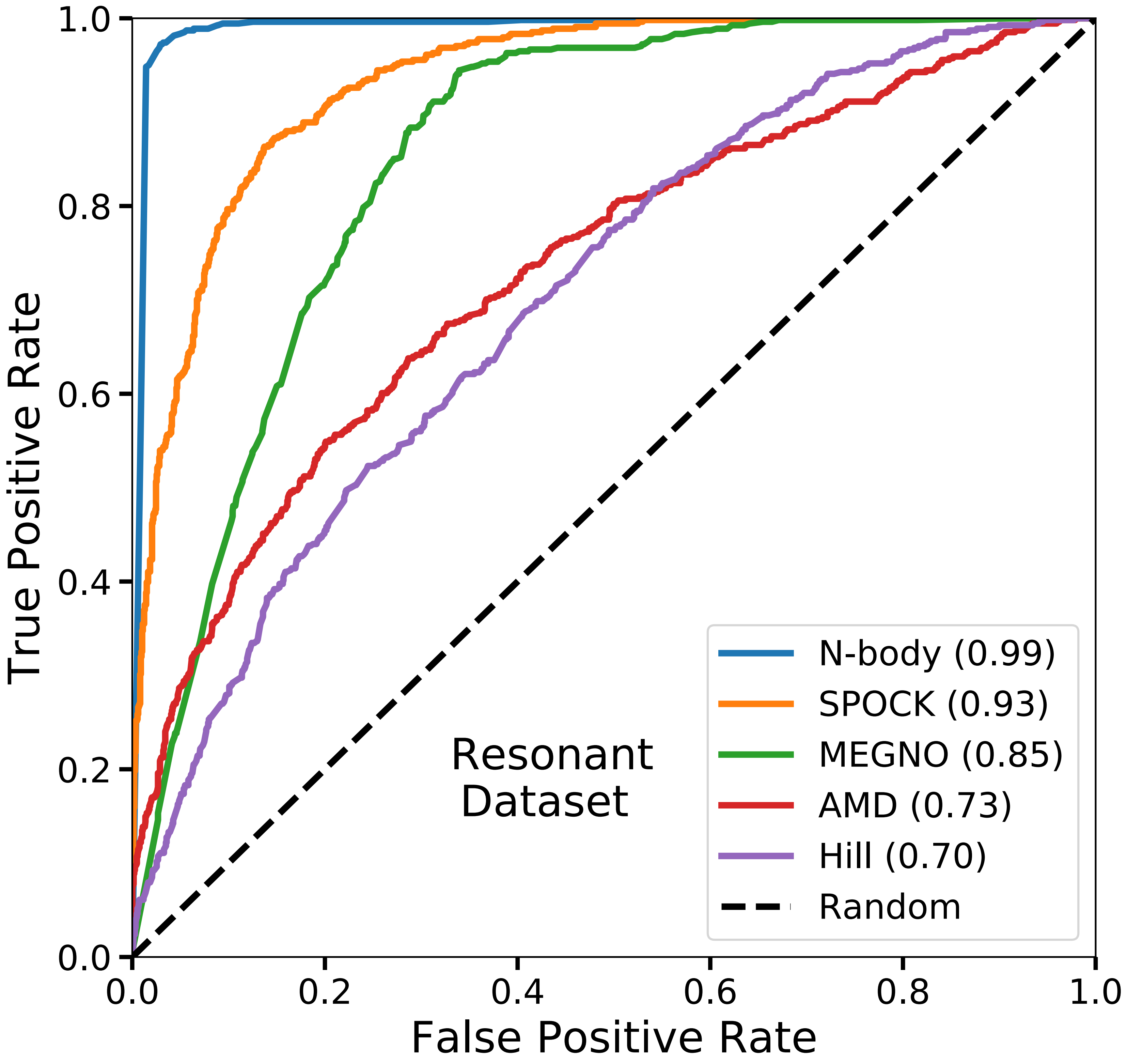
Tamayo et al. (submitted)

Comparison



Tamayo et al. (submitted)

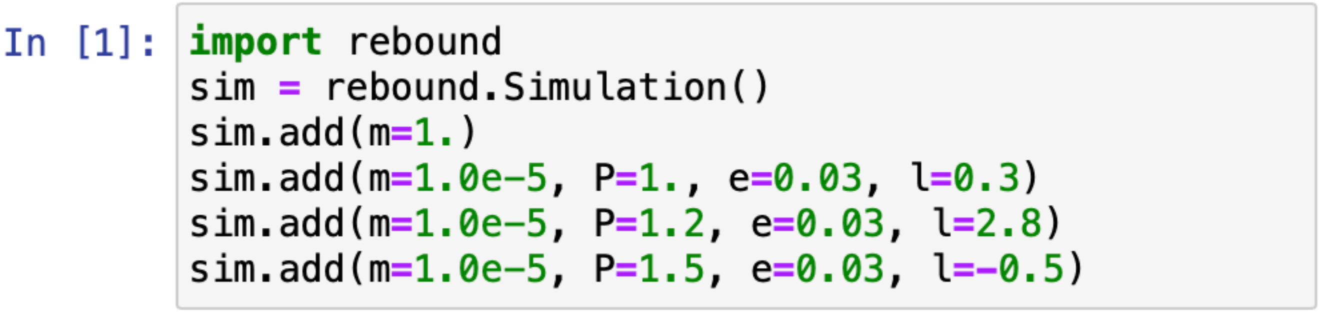
Performance



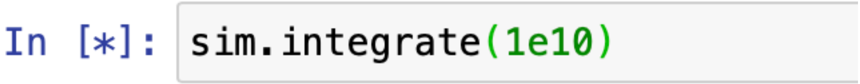
Tamayo et al. (submitted)

Easy to use implementation

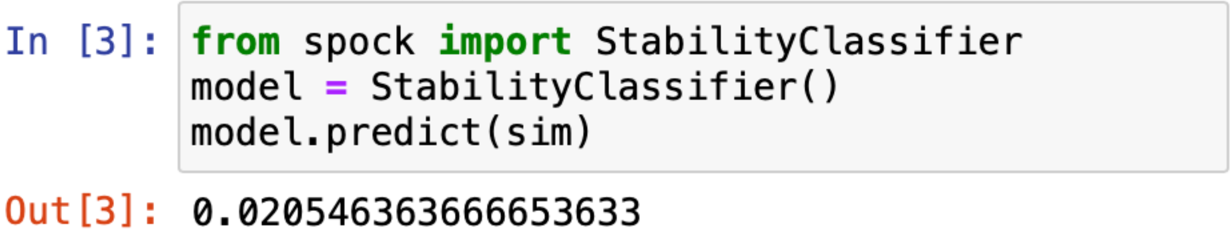
Setup simulation in REBOUND



```
In [1]: import rebound
        sim = rebound.Simulation()
        sim.add(m=1.)
        sim.add(m=1.0e-5, P=1., e=0.03, l=0.3)
        sim.add(m=1.0e-5, P=1.2, e=0.03, l=2.8)
        sim.add(m=1.0e-5, P=1.5, e=0.03, l=-0.5)
```



```
In [*]: sim.integrate(1e10)
```



```
In [3]: from spock import StabilityClassifier
        model = StabilityClassifier()
        model.predict(sim)

Out[3]: 0.020546363666653633
```

Direct N-body simulation with REBOUND

May take days

Prediction with SPOCK

Takes one second

Tamayo et al. (submitted)

Numerics:
Operator splitting methods

Fundamentals

Equations of motions

$$\ddot{\mathbf{r}}_i = \sum_{\substack{j=1\\ j\neq i}}^{N} m_j \frac{\mathbf{r}_j - \mathbf{r}_i}{\left|\mathbf{r}_j - \mathbf{r}_i\right|^3}$$

↕

Hamiltonian

$$H = \overset{\substack{\textbf{kinetic term}\\ \swarrow}}{T} + \overset{\substack{\textbf{potential term}\\ \swarrow}}{U}$$

Fundamentals

Poisson Bracket of two functions f, g of the canonical coordinates q, p.

$$\{g, h\} = \sum_i \left(\frac{\partial g}{\partial q_i} \frac{\partial h}{\partial p_i} - \frac{\partial g}{\partial p_i} \frac{\partial h}{\partial q_i} \right)$$

Allows us to write Hamilton’s equations as

$$\dot{q}_i = \frac{\partial H}{\partial p_i} = \{q_i, H\} \qquad \text{and} \qquad \dot{p}_i = -\frac{\partial H}{\partial q_i} = \{p_i, H\}$$

Allows us to write the time derivate of *any* function of q and p as

$$\dot{g} = \{g, H\}$$

Fundamentals

Introduce

$$y(t) = (q_1(t), \ldots, q_N(t), p_1(t), \ldots, p_N(t))$$

The differential equation for the N-body problem is then

$$\dot{y}(t) = \{y, H\}$$

A bit more notations. Lie derivative:

$$\mathcal{L}_H = \{\cdot, H\}$$

Define a solution operator:

$$\varphi_t^{[H]}(y_0)$$

Notation roughly follows Hairer, Lubich & Wanner 2006

Fundamentals

The differential equation we're trying to solve

$$\dot{y}(t) = \mathcal{L}_H \, y(t)$$

A formal way to write down the solution operator

$$\begin{aligned} \varphi_t^{[H]}(y_0) &= \exp\left(t\mathcal{L}_H\right)\mathrm{Id}(y_0) \\ &= \left(\mathrm{Id} + t\mathcal{L}_H\mathrm{Id} + \frac{1}{2}t^2\mathcal{L}_H\mathcal{L}_H\mathrm{Id} + \ldots\right)(y_0) \end{aligned}$$

Only a formal solution!

Notation roughly follows Hairer, Lubich & Wanner 2006

Fundamentals

$$\ddot{\mathbf{r}}_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} m_j \frac{\mathbf{r}_j - \mathbf{r}_i}{\left|\mathbf{r}_j - \mathbf{r}_i\right|^3}$$

Splitting method

$$\dot{y} = \mathcal{L}_H(y)$$

$$H = A + B$$

$$\dot{y} = \left(\mathcal{L}_A + \mathcal{L}_B\right)(y) = \mathcal{L}_A(y) + \mathcal{L}_B(y).$$

We can now consider two new differential equations

$$\dot{y} = \mathcal{L}_A(y) \quad \text{and} \quad \dot{y} = \mathcal{L}_B(y).$$

Example!

$$H = \frac{1}{2}p^2 + U(q) \qquad\qquad y(t) = (q(t), p(t))$$

Let’s split the Hamiltonian:

$$A = \frac{1}{2}p^2 \qquad B = U(q)$$

$$\dot{y} = \mathcal{L}_A\, y \qquad \dot{y} = \mathcal{L}_B\, y$$

$$\dot{y} = \{y, A\} \quad \dot{y} = \{y, B\}$$

$$\dot{q} = p \qquad\qquad \dot{q} = 0$$

$$\dot{p} = 0 \qquad\qquad \dot{p} = -\nabla U(q)$$

Solutions are trivial!

Example!

$$\begin{aligned}\dot{q} &= p\\ \dot{p} &= 0\end{aligned} \qquad \longrightarrow \qquad \varphi_t^{[A]}(q_0, p_0) = (q_0 + t \cdot p_0, p_0)$$

$$\begin{aligned}\dot{q} &= 0\\ \dot{p} &= -\nabla U(q)\end{aligned} \qquad \longrightarrow \qquad \varphi_t^{[B]}(q_0, p_0) = (q_0, p_0 - t \cdot \nabla U(q))$$

Idea of a splitting scheme is:

$$\varphi_t^{[H]}(y_0) \approx \varphi_t^{[B]} \varphi_t^{[A]}(y_0)$$

Example!

$$\varphi_t^{[A]}(q_0, p_0) = (q_0 + t \cdot p_0, p_0)$$

$$\varphi_t^{[B]}(q_0, p_0) = (q_0, p_0 - t \cdot \nabla U(q))$$

$$\exp\left(t\mathcal{L}_A\right)\exp\left(t\mathcal{L}_B\right) = \exp\left(t\mathcal{L}_A + t\mathcal{L}_B + \frac{1}{2}t^2\left[\mathcal{L}_A, \mathcal{L}_B\right] + \mathcal{O}(t^3)\right)$$

BCH

Can show that

$$[\mathcal{L}_A, \mathcal{L}_B] = \mathcal{L}_{\{A,B\}}$$

So, we're solving the EoM for the following Hamiltonian *exactly*

$$H' = A + B + \frac{1}{2}t\{A, B\} + \mathcal{O}(t^2)$$

Example!

We’re solving the EoM for the following Hamiltonian *exactly*

$$H' = A + B + \frac{1}{2}t\{A, B\} + \mathcal{O}(t^2)$$

for our specific example:

$$H' = \frac{1}{2}p^2 + U(q) + \frac{1}{2}t\left\{\frac{1}{2}p^2, U(q)\right\} + \mathcal{O}(t^2)$$

$$= H - \frac{t}{2}p\nabla U(q) + \mathcal{O}(t^2)$$

Works for small t, thus

$$t \to dt$$

and repeat many times.

Order

First order scheme

$$\varphi_t^{[A]}\varphi_t^{[B]}$$

$$E \sim \mathcal{O}(dt)$$

Standard second order leapfrog:

$$\varphi_{0.5t}^{[A]}\varphi_t^{[B]}\varphi_{0.5t}^{[A]}$$

$$E \sim \mathcal{O}(dt^2)$$

Fourth order leapfrog:

$$\varphi_{\alpha t}^{[A]}\varphi_{2\alpha t}^{[B]}\varphi_{(0.5-\alpha)t}^{[A]}\varphi_{(1-4\alpha)t}^{[B]}\varphi_{(0.5-\alpha)t}^{[A]}\varphi_{2\alpha t}^{[B]}\varphi_{\alpha t}^{[A]}$$

$$\alpha = 0.675603\ldots$$

$$E \sim \mathcal{O}(dt^4)$$

Embedded Operator Splitting Methods (EOS)

Rein, Brown & Tamayo (2019), Rein, Tamayo & Brown (2019), Rein (2020)

N-body Hamiltonian

kinetic term

potential term

$$H = T + U$$

$$T = \sum_{i=0}^{N-1} T_i$$

$$U = \sum_{i=0}^{N-1} \sum_{j=i+1}^{N-1} U_{ij}$$

$$T_i = \frac{p_i^2}{2m_i}$$

$$U_{ij} = -\frac{G m_i m_j}{|r_i - r_j|}$$

Leap frog integrator

$$H = \overset{\text{kinetic term}}{T} + \overset{\text{potential term}}{U}$$

A $$T = \sum_{i=0}^{N-1} T_i$$

B $$U = \sum_{i=0}^{N-1} \sum_{j=i+1}^{N-1} U_{ij}$$

Both solutions are trivial again!

$$\varphi_t^{[A]}(y_0) = (q_0 + t \cdot p_0, p_0)$$

$$\varphi_t^{[B]}(y_0) = (q_0, p_0 - t \cdot \nabla U(q))$$

Wisdom-Holman integrator

$$H = A + B$$

A

$$\sum_{i=0}^{N-1} T_i + \sum_{i=1}^{N-1} U_{i0}$$

Dominant part of motion

B

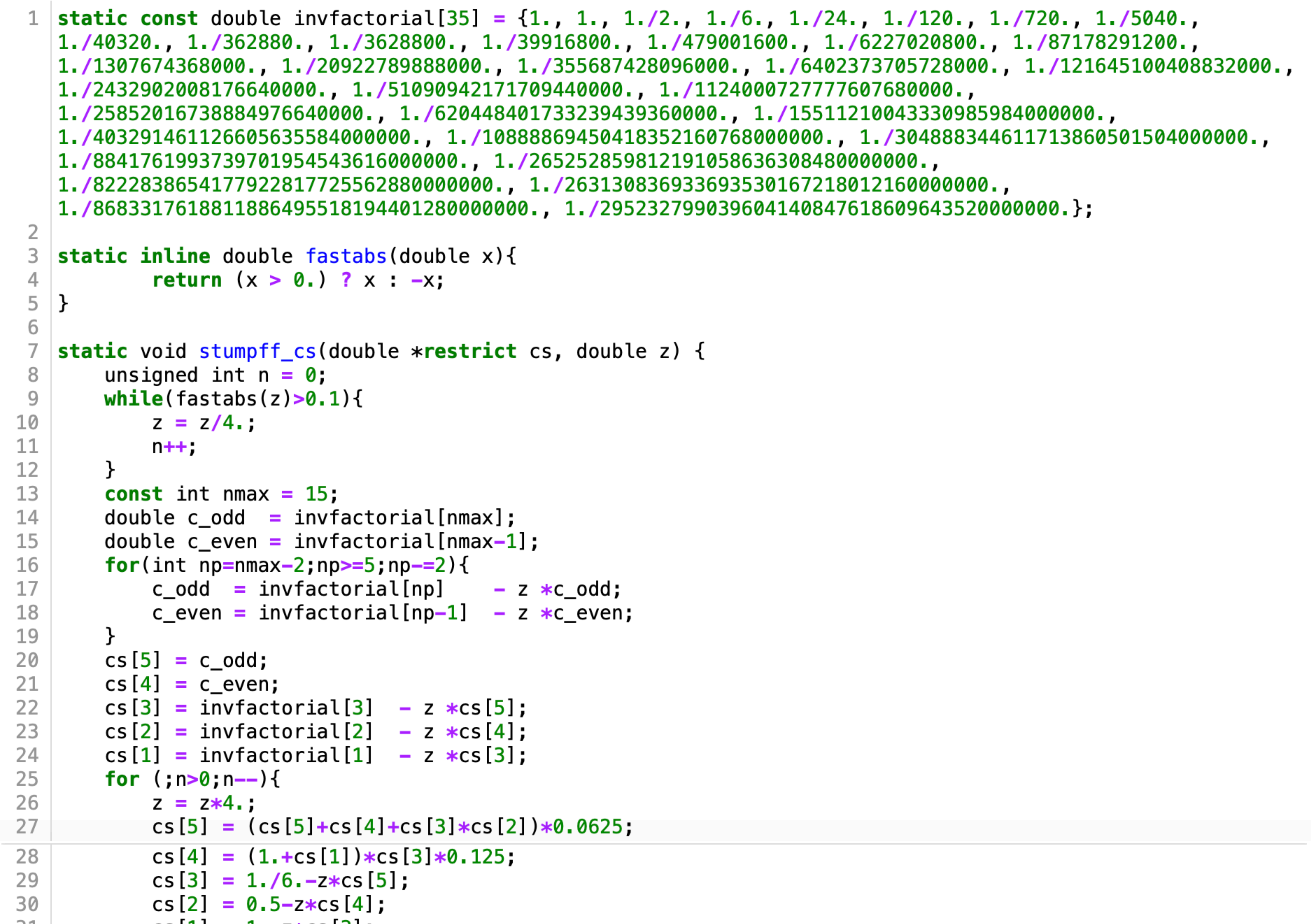
$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N-1} U_{ij}$$

Perturbation

Solution for B is still trivial.

Solution for A is more complicated. We need a “Kepler solver”.

Kepler Solver (WHFast)



```c
static const double invfactorial[35] = {1., 1., 1./2., 1./6., 1./24., 1./120., 1./720., 1./5040.,
1./40320., 1./362880., 1./3628800., 1./39916800., 1./479001600., 1./6227020800., 1./87178291200.,
1./1307674368000., 1./20922789888000., 1./355687428096000., 1./6402373705728000., 1./121645100408832000.,
1./2432902008176640000., 1./51090942171709440000., 1./1124000727777607680000.,
1./25852016738884976640000., 1./620448401733239439360000., 1./15511210043330985984000000.,
1./403291461126605635584000000., 1./10888869450418352160768000000., 1./304888344611713860501504000000.,
1./8841761993739701954543616000000., 1./265252859812191058636308480000000.,
1./8222838654177922817725562880000000., 1./263130836933693530167218012160000000.,
1./8683317618811886495518194401280000000., 1./295232799039604140847618609643520000000.};

static inline double fastabs(double x){
        return (x > 0.) ? x : -x;
}

static void stumpff_cs(double *restrict cs, double z) {
    unsigned int n = 0;
    while(fastabs(z)>0.1){
        z = z/4.;
        n++;
    }
    const int nmax = 15;
    double c_odd  = invfactorial[nmax];
    double c_even = invfactorial[nmax-1];
    for(int np=nmax-2;np>=5;np-=2){
        c_odd  = invfactorial[np]    - z *c_odd;
        c_even = invfactorial[np-1]  - z *c_even;
    }
    cs[5] = c_odd;
    cs[4] = c_even;
    cs[3] = invfactorial[3]  - z *cs[5];
    cs[2] = invfactorial[2]  - z *cs[4];
    cs[1] = invfactorial[1]  - z *cs[3];
    for (;n>0;n--){
        z = z*4.;
        cs[5] = (cs[5]+cs[4]+cs[3]*cs[2])*0.0625;
        cs[4] = (1.+cs[1])*cs[3]*0.125;
        cs[3] = 1./6.-z*cs[5];
        cs[2] = 0.5-z*cs[4];
        cs[1] = 1.-z*cs[3];
```

Embedded Operator Splitting Method (EOS)

$$H = A + B$$

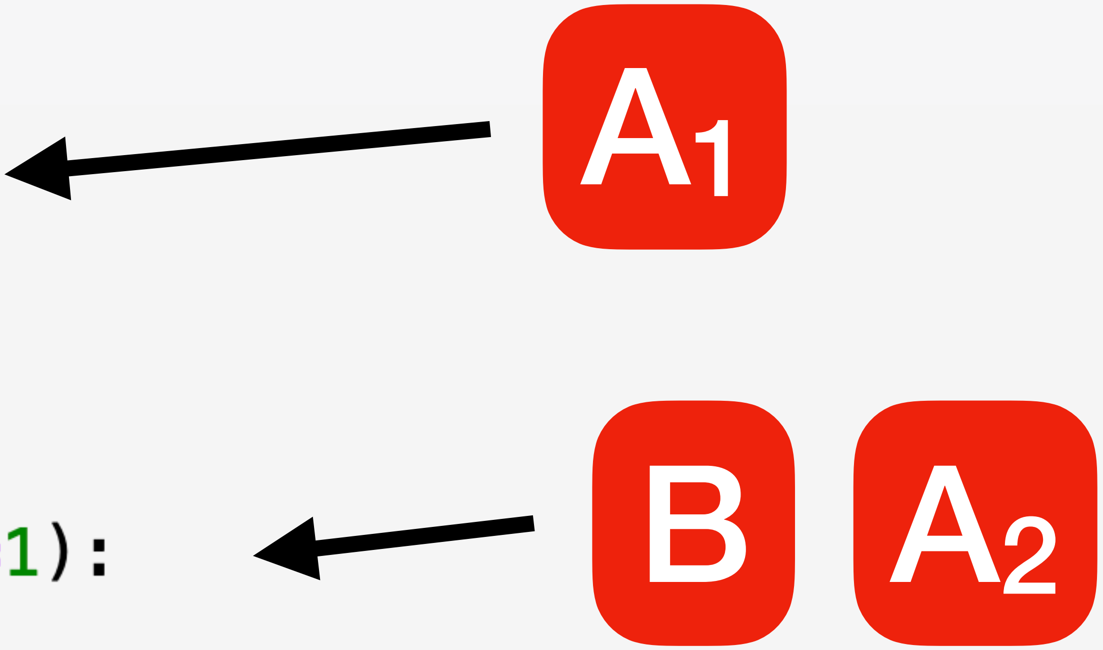
A $\sum_{i=0}^{N-1} T_i + \sum_{i=1}^{N-1} U_{i0}$

B $\sum_{i=1}^{N-1} \sum_{j=i+1}^{N-1} U_{ij}$

Rein 2020

Example EOS Implementation in Python

```python
def drift(particles, dt):
    for p in particles:
        p.x += dt*p.vx
        p.y += dt*p.vy
        p.z += dt*p.vz
def kick(particles, dt, shell=1):
    for p1 in particles:
        for p2 in particles:
            if p1!=p2:
                if shell==0 and (p1.m==1. or  p2.m==1.): continue
                if shell==1 and (p1.m!=1. and p2.m!=1.): continue
                dx, dy, dz = p1.x-p2.x, p1.y-p2.y, p1.z-p2.z
                dr3 = pow(dx**2 + dy**2 + dz**2, 3./2.)
                p1.vx -= dt*p2.m*dx/dr3
                p1.vy -= dt*p2.m*dy/dr3
                p1.vz -= dt*p2.m*dz/dr3
```



Ideal for GPUs!

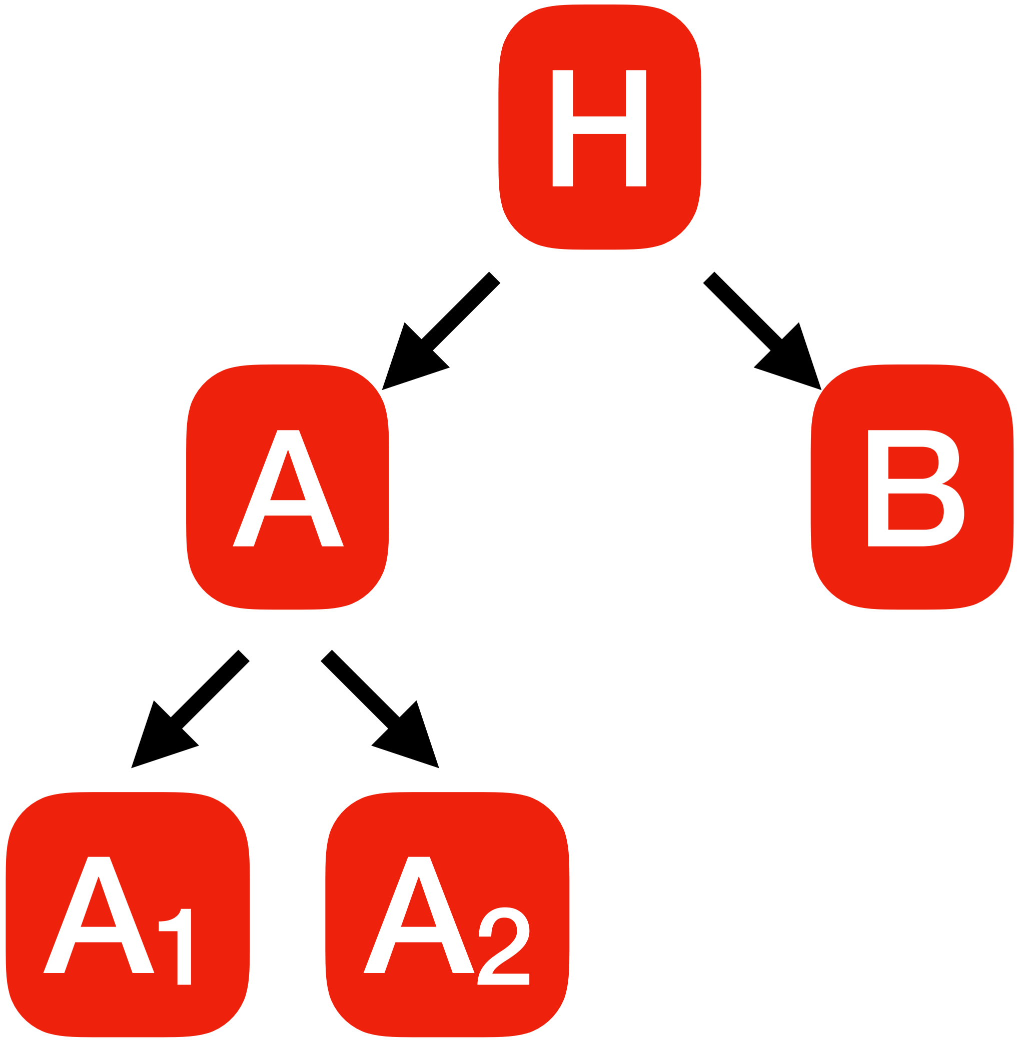
Rein 2020

A lot of choice

Full Hamiltonian

1st splitting

2nd splitting



At each splitting, can choose:

- How to split Hamiltonian into two parts
- Which splitting method to use
- Timestep

Embedded Operator Splitting Method (EOS)

First order method:

$$\varphi_t^{[A]} \approx \varphi_t^{[A_1]} \varphi_t^{[A_2]}$$

But can also choose any arbitrary operator splitting method:

$$\varphi_t^{[A]} \approx \varphi_{0.5t}^{[A_1]} \varphi_t^{[A_2]} \varphi_{0.5t}^{[A_1]}$$

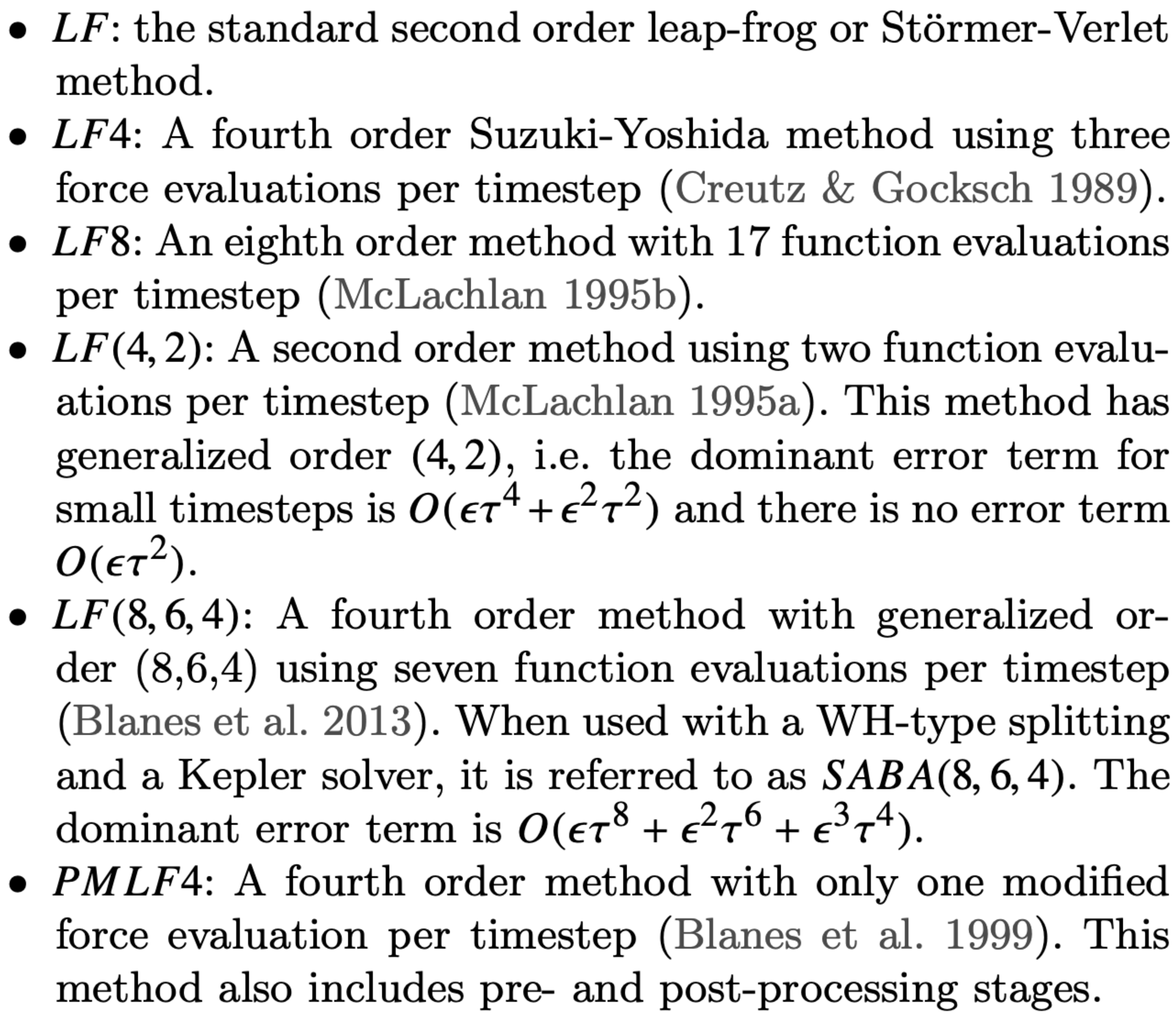
$$\varphi_t^{[A]} \approx \varphi_{\alpha t}^{[A_a]} \varphi_{2\alpha t}^{[A_2]} \varphi_{(0.5-\alpha)t}^{[A_1]} \varphi_{(1-4\alpha)t}^{[A_2]} \varphi_{(0.5-\alpha)t}^{[A_1]} \varphi_{2\alpha t}^{[A_2]} \varphi_{\alpha t}^{[A_1]}$$

Alternatively, reduce timestep in embedded method:

$$\varphi_t^{[A]} \approx \left(\varphi_{t/n}^{[A_1]} \varphi_{t/n}^{[A_2]} \right)^n$$

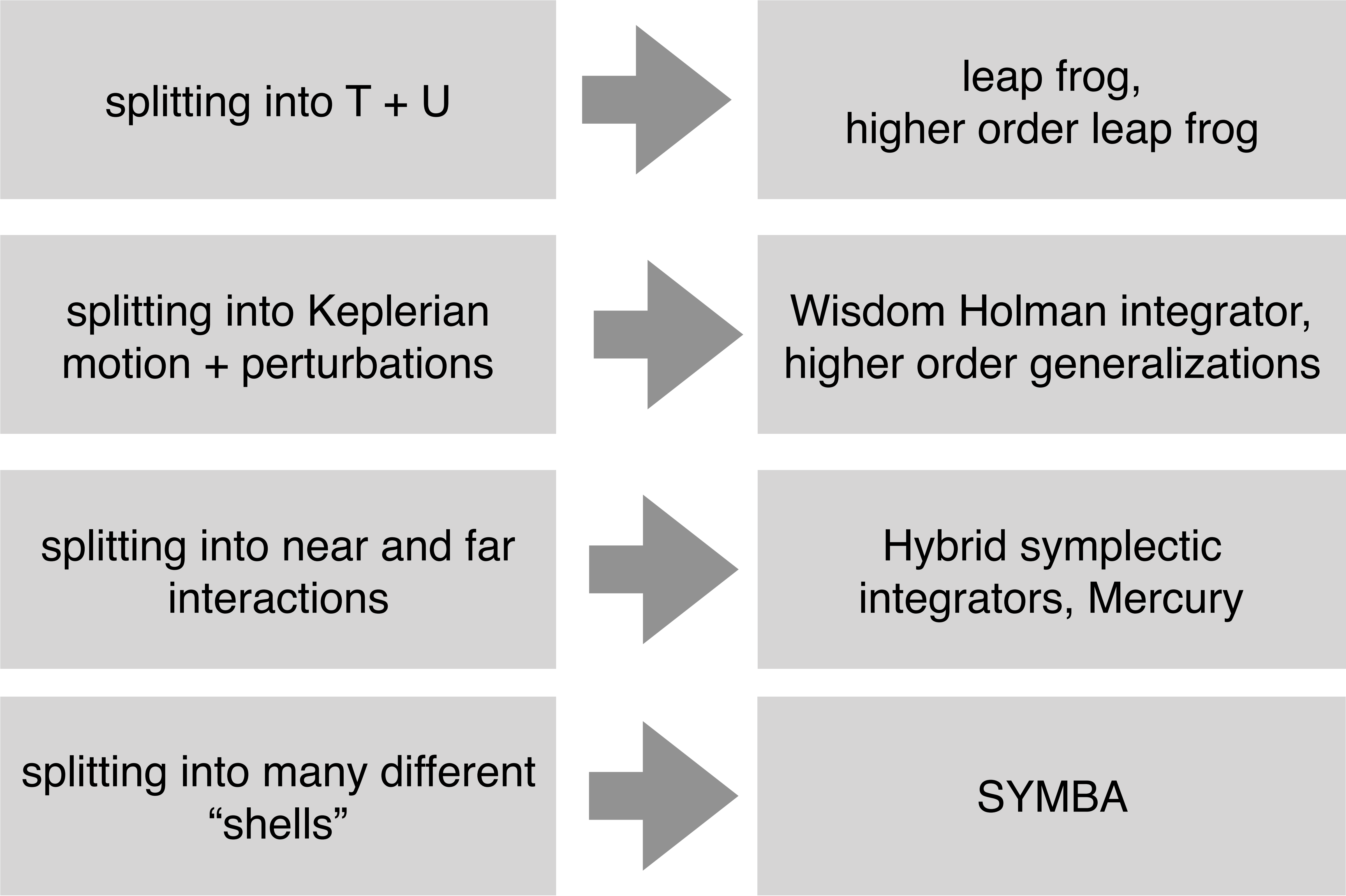
Rein 2020

Splitting methods



- LF: the standard second order leap-frog or Störmer-Verlet method.
- $LF4$: A fourth order Suzuki-Yoshida method using three force evaluations per timestep (Creutz & Gocksch 1989).
- $LF8$: An eighth order method with 17 function evaluations per timestep (McLachlan 1995b).
- $LF(4,2)$: A second order method using two function evaluations per timestep (McLachlan 1995a). This method has generalized order (4,2), i.e. the dominant error term for small timesteps is $O(\epsilon\tau^4+\epsilon^2\tau^2)$ and there is no error term $O(\epsilon\tau^2)$.
- $LF(8,6,4)$: A fourth order method with generalized order (8,6,4) using seven function evaluations per timestep (Blanes et al. 2013). When used with a WH-type splitting and a Kepler solver, it is referred to as $SABA(8,6,4)$. The dominant error term is $O(\epsilon\tau^8 + \epsilon^2\tau^6 + \epsilon^3\tau^4)$.
- $PMLF4$: A fourth order method with only one modified force evaluation per timestep (Blanes et al. 1999). This method also includes pre- and post-processing stages.

EOS methods are extremely flexible



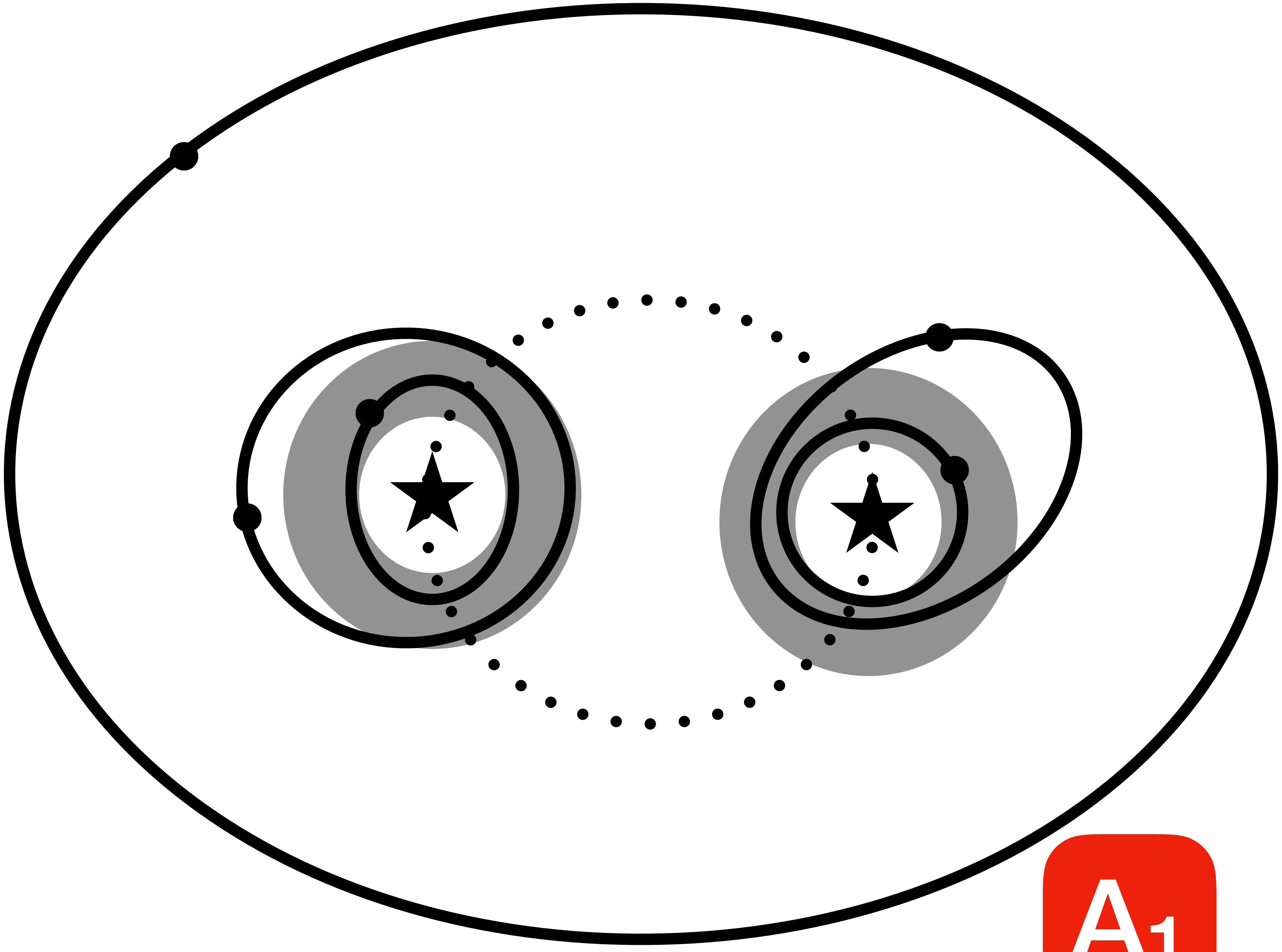
splitting into T + U → leap frog, higher order leap frog

splitting into Keplerian motion + perturbations → Wisdom Holman integrator, higher order generalizations

splitting into near and far interactions → Hybrid symplectic integrators, Mercury

splitting into many different “shells” → SYMBA

Example: complicated hierarchical systems



A_1 kinetic term (drift)

B planet-planet interactions (kick)

A_2 planet-star and star-star interactions (kick)

Demo

Conclusions

Determining the stability of planetary systems is a very old problem. Analytic solution cannot answer all question.

Workhorse for numerical N-body simulations are operator splitting methods, especially the Wisdom-Holman integrator.

Embedded Operator Splitting methods (EOS) are very easy to implement as they do not require a Kepler solver. An EOS method can be configured to be equivalent to: leap-frog, Wisdom-Holman, Mercury, SYMBA, and many new methods.

Our new machine-learning classifier SPOCK can accurately predict the stability of compact planetary systems over billions of years in seconds (10^5 times faster than direct N-body).