❑ LECTURE 8

- DISLIN graphics library. Volcano problem.

- Analysis of the Laplace stencil program in CUDA

- Massively parallel computation of Tetrahedron in a sphere problem (tetra-Dc) in Fortran [and tetra-Dg in CUDA Fortran]

- Supercomputing in astrophysics at UTSC

- CUDA C and CUDA Fortran

Literature: see our course home page, the refs subpage + the coding page linked to course page.

LECTURE 8

Programming and parallelism: C, Fortran, OpenMP (continued)

- Massively parallel processing of tetrahedrons (tetra-Dc) in Fortran [and tetra-Dg in CUDA Fortran]

- Analysis of the Laplace stencil program (art-2: ~/progD57) ifor-laplace3-dp.f90, -sp.f90. cudafor-laplace-sp.f90

Motivation for HPC: scientific calculations using massively parallel processors

Literature: see links on our course home page, the references page, and the coding page available from the course page.

Massively parallel integration on the **newest HPC platforms:** CPU, GPU and MIC

from a conference talk, 2017, discussing work by P. Artymowicz and F. Horrobin at UTSC

Concurrent simulation of 200 or 7000 planetary systems on CPUs or MIC

Collisionless gigaparticle disks. Interaction with binary system.

Hybrid algorithm (4^{th} order symplectic with collisions)
Implementation and optimization in Fortran90 on 1..32 MIC (Φ)
Migration problem
Tests and preliminary results
Fast migration in particle disks as type III CR-driven migration

1990s and 2000s was the era of clusters

MPI for parallelization.

Later, in 2000s, coprocessors appeared.... like

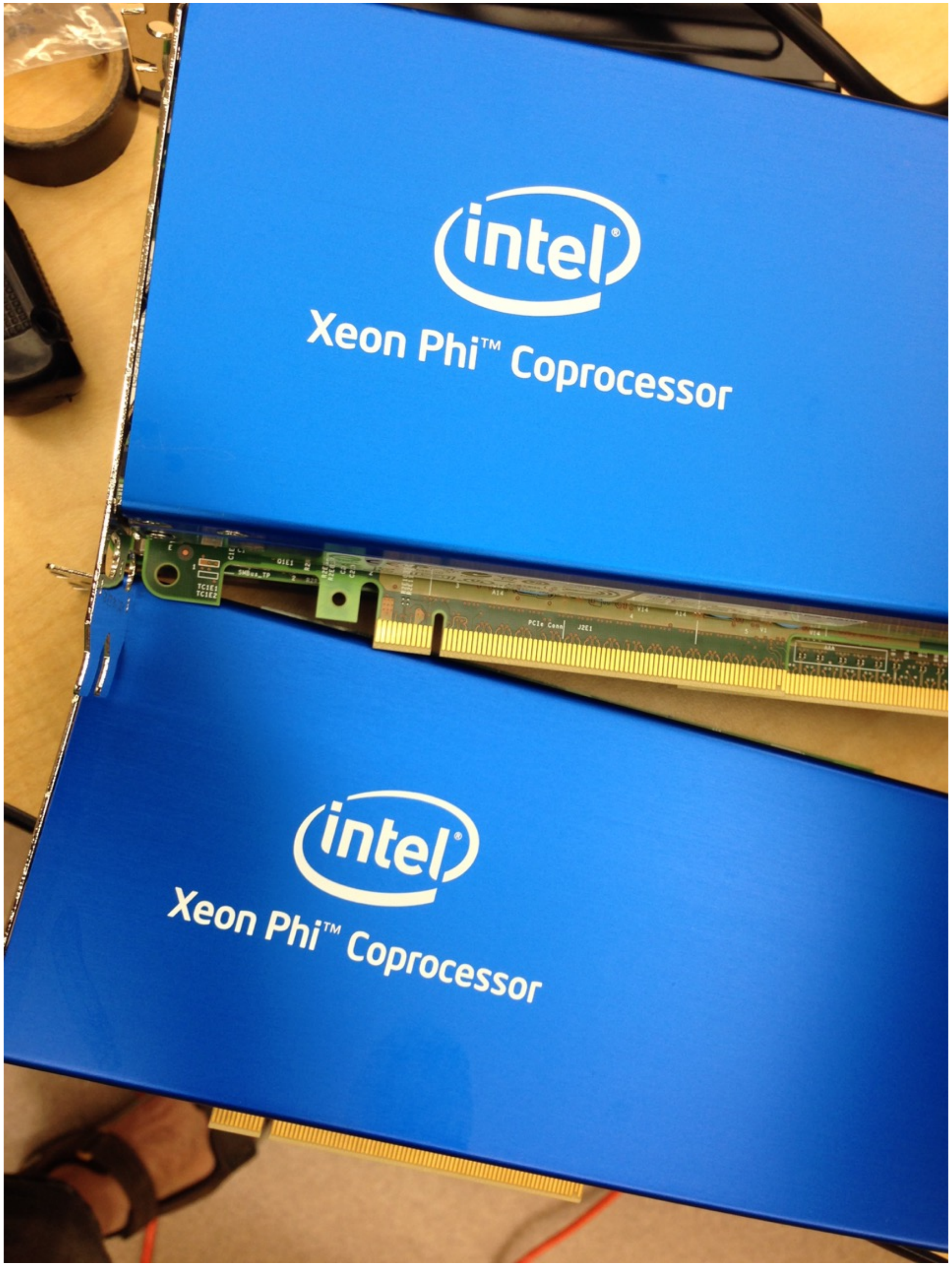
MIC

MIC = many integrated cores
(Intel’s term for many-core, massively parallel, CPU-like processors)

and GPU

GPU = Graphics Processing Unit (processor inside graphics card, actually more capable of quick computation than CPU).

It seemed that the we won’t bother to build clusters any more, but it wasn’t true.



MIC = many integrated CPU-like cores (~60)

Intel Xeon Phi accelerators

Knights Corner:
~1 TFLOP dp
~2 TFLOP sp

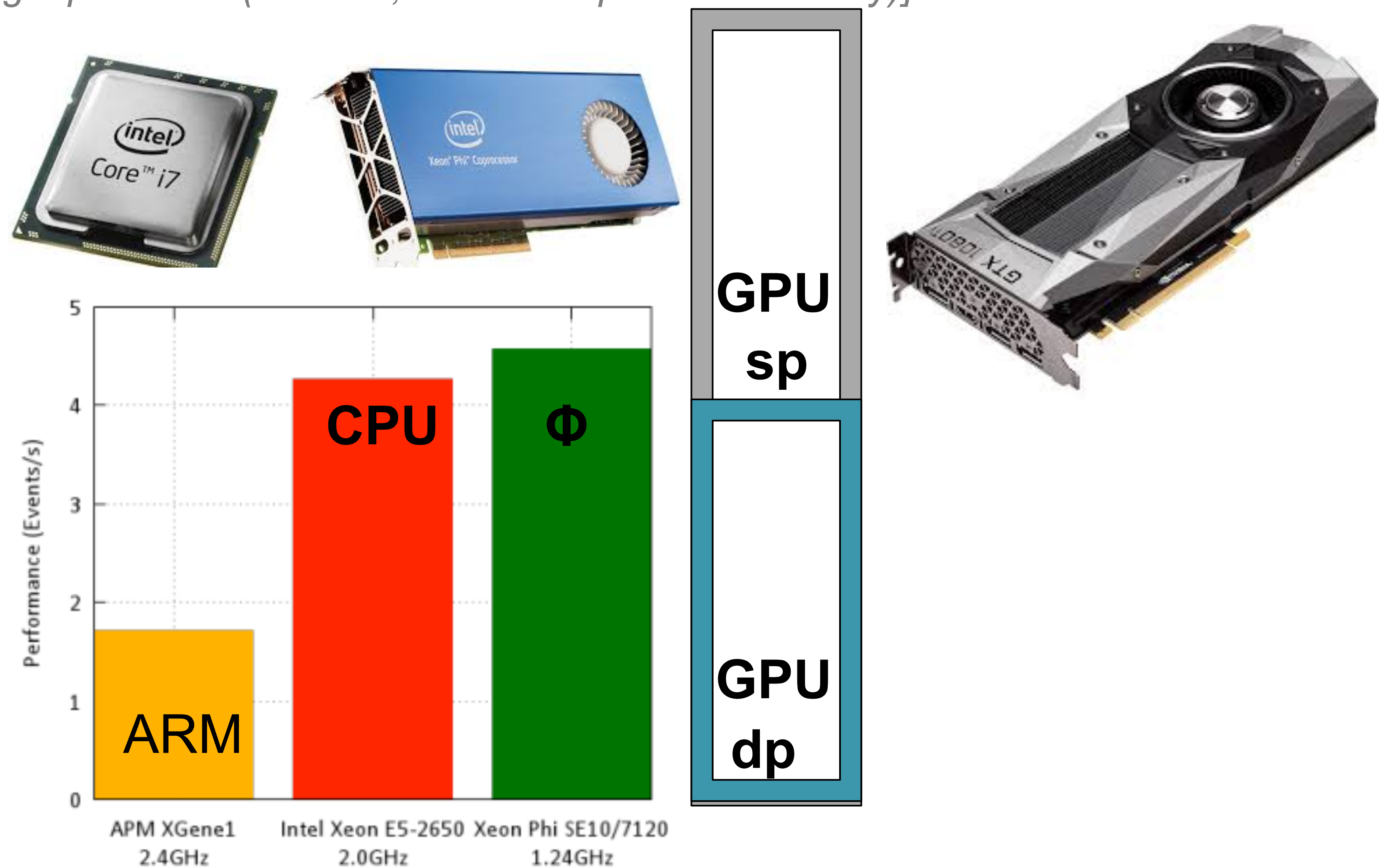
Knights Landing: ~3x more TFLOPs

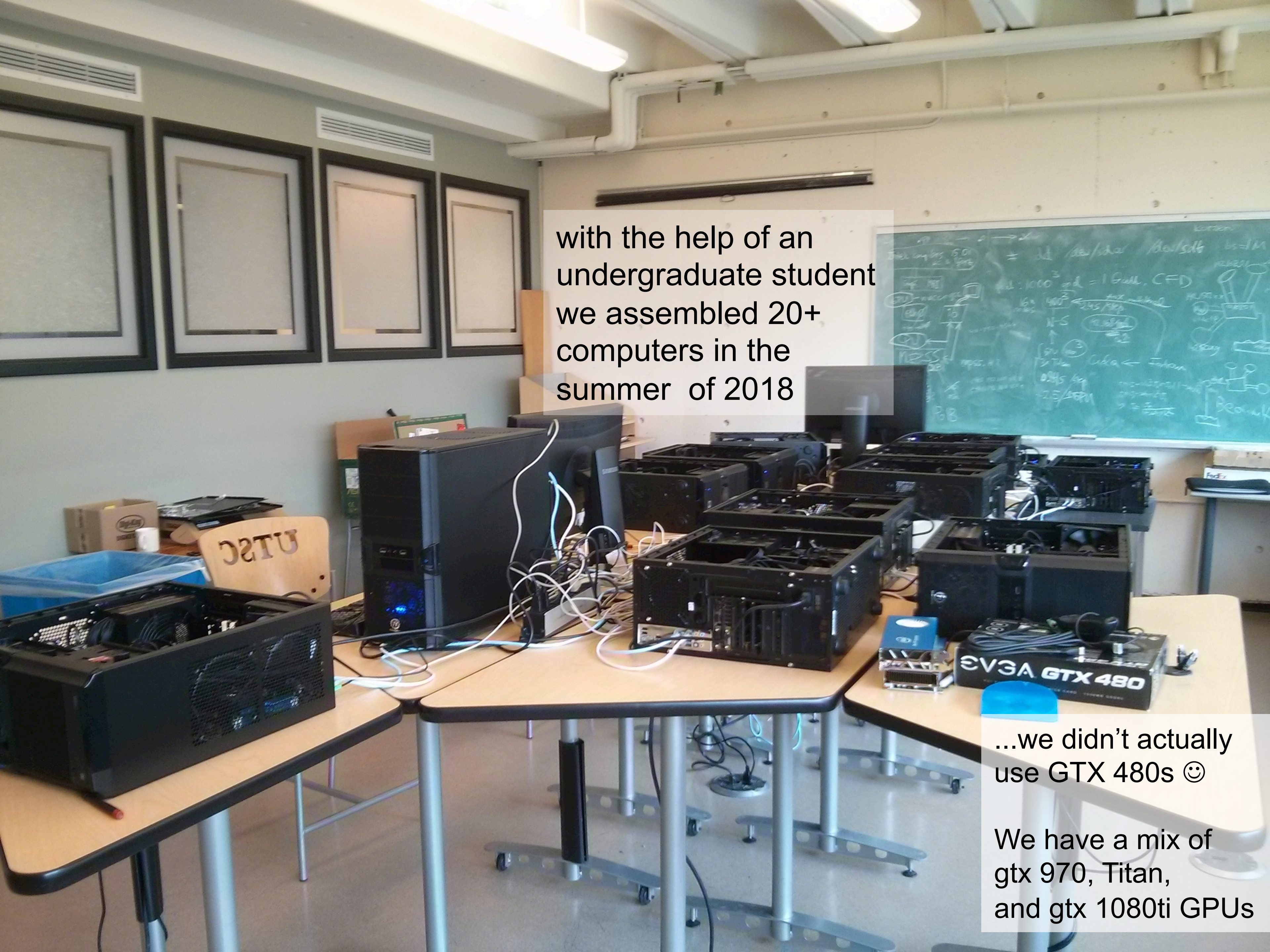
TFLOP = 1 T FLOP/s.

1 E = ~10^{18} exa
1 P = ~10^{15} peta
1 T = ~10^{12} tera
1 G = ~10^{9} giga
1 M = ~10^{6} mega
1 K = ~10^{3} kilo

In 2014, CERN Researchers considered which of the platforms makes the most sense for distributed Worldwide LHC Computing Grid, processing data for Large Hadron Collider experiments in 170 computing centers, in 40 countries (incl. UofT). The height of the bar is proportional to the estimated speed in CERN simulations with then-current hardware. .Nowadays GPU have somewhat more advantage over CPU & MIC

[dp = double precision (8B/float like in Python, 15 accurate decimal places),
sp = single precision (4B/float, 7 decimal places accuracy)]

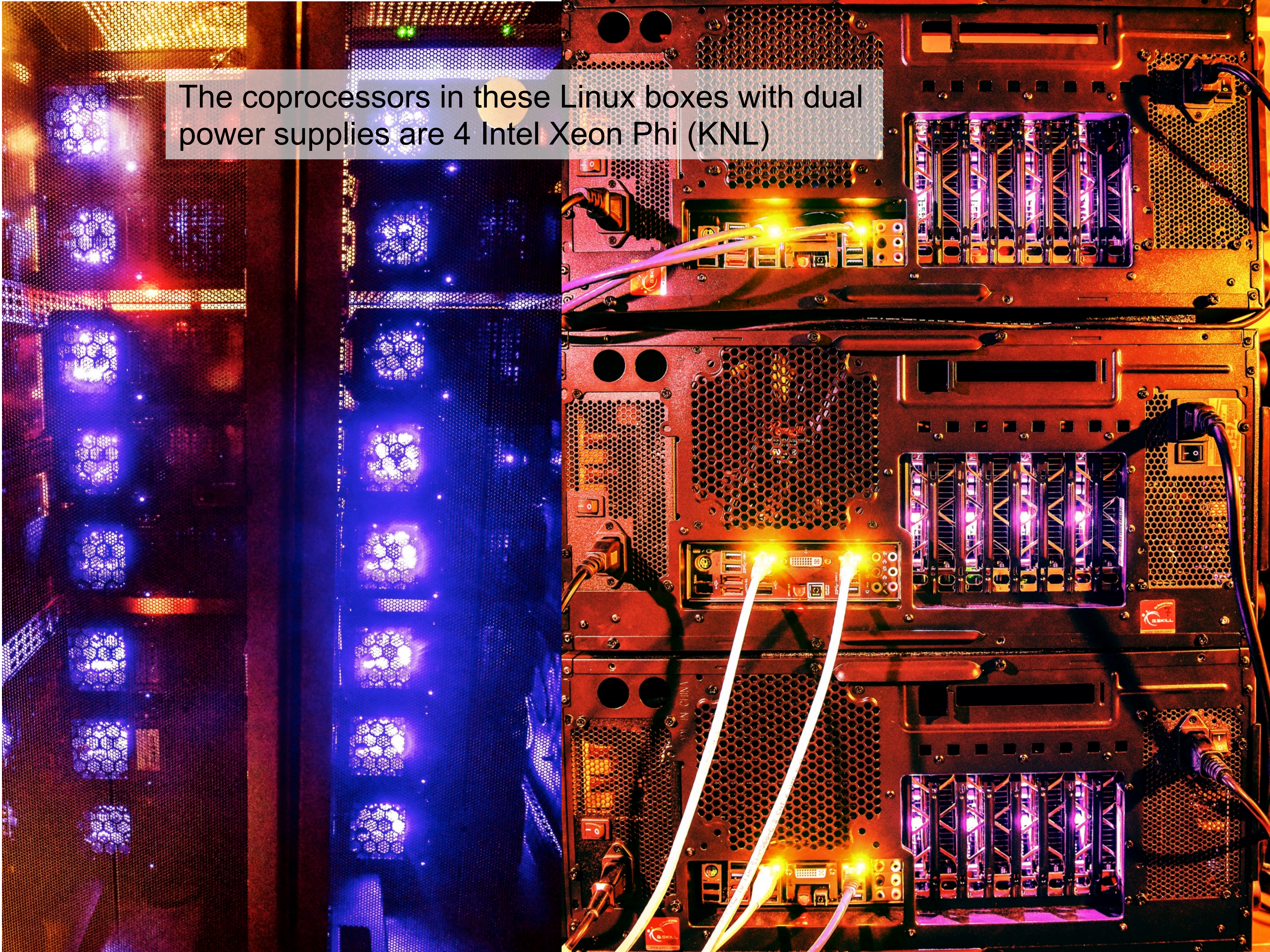




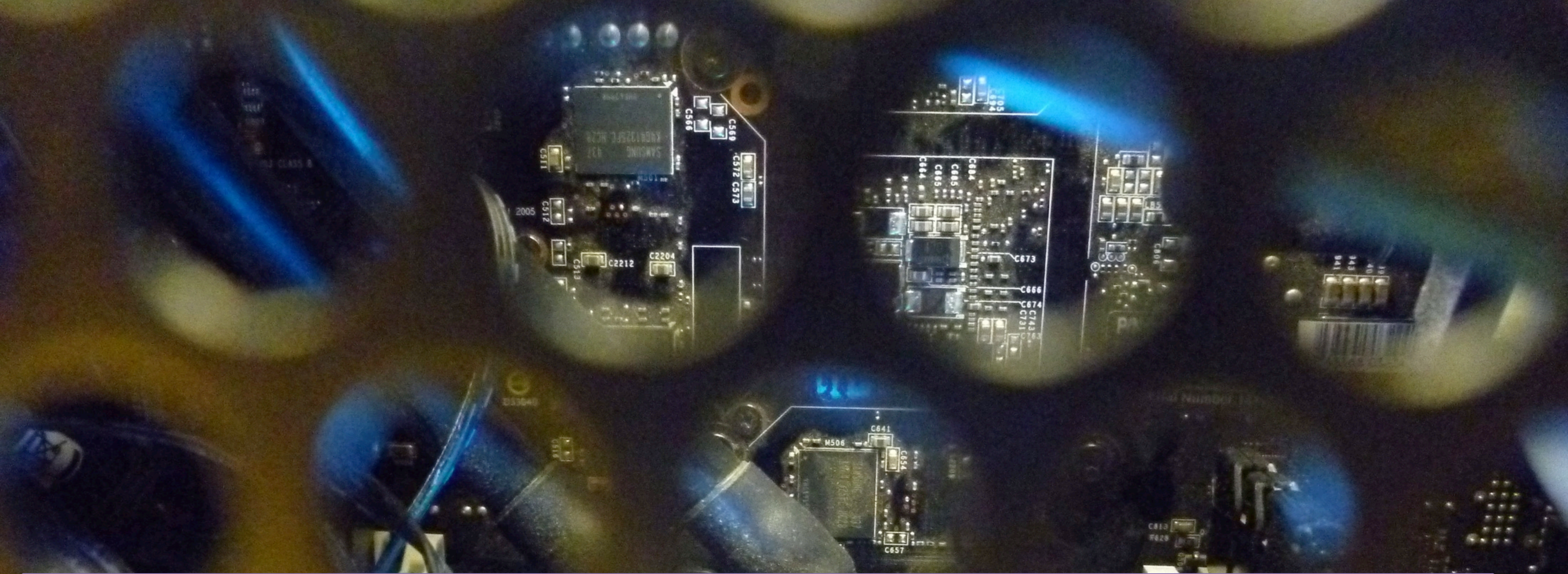
with the help of an undergraduate student we assembled 20+ computers in the summer of 2018

...we didn't actually use GTX 480s ☺

We have a mix of gtx 970, Titan, and gtx 1080ti GPUs



The coprocessors in these Linux boxes with dual power supplies are 4 Intel Xeon Phi (KNL)



In a small cluster, CPU, MIC and GPU are combined. They can work together or separately. We can simulate a galaxy’s inner part star by star (~4 G stars), and/or its gas disk at high resolution (2 Gcell) by exchanging data between linux nodes in every time step.

We can run 200 8-planet simulations very fast (1G periods simulated in a day at 360+ steps per orbit), or 7000 simulations, 5 times slower

Large N-body systems by direct summation

20 arithmetic operations per one pairwise grav. interaction

leapfrog (Fortran90) **leapfrog (CUDA C)**

N = 10 K ... 1 M

CPU (i7-5820K 4GHz)	**MIC** (KNC)	**GPU** (gtx 980, Titan)
0.28 TFLOP sp 14 G interac/s	1.33 TFLOP sp 67 G interac/s	3.5 TFLOP sp (gtx980) 190 G interac/s
0.09 TFLOP dp 4.5 G interac/s	0.51 TFLOP dp 25 G interac/s	0.81 TFLOP dp (Titan) 40 G interac/s

!
! on MIC the calculation is **2.8 times slower** than on GPU (sp)
! **1.6 times slower** than on GPU (dp)
! CPU (6c.) is **9..13 times slower** than GPU
!

note: this is a rare fully compute-bound calculation!

Concurrent 8-body systems by 4th order symplectic code

n8b-aug14.3.f90 Same double precision program. Compiled with ifort

platform	CPU	MIC
compiler flag	-xhost	-mmic
number of N-body systems per processor	12	224
N [#threads per sys.]	8 [1]	8 [1]
exec. time per step	0.871 μs	4.58 μs
steps per orbit	360	360
exec. time of 1 orbit	0.313 ms	1.65 ms
exec. time (1G orbits)	3.63 days	19.1 days
system clock	4 GHz	1.1 GHz
throughput	13.8 M sys-step/s	49 M sys-step/s
# concurrent systems (SciPhi cluster UTSC)	192	10752

Practical capabilities of processor platforms for dynamical astro-calculations. Single (co)processors
CPU ~ E5 and i7 ser. (Intel), MIC = Knights Corner (Intel 2013),
GPU = Nvidia GTX970..1080 (sp) and Titan (dp) run:

1. Gravit. **N-body problem** $\mathcal{O}(\sim N^2)$. N $\sim 10^6$ real-time (~1 fps)
   GPU > MIC ~ CPU (mostly comput. limited, > TFLOP)

2. Disks of **particles** (stars; asteroids, planetesimals, meteoroids and dust).
   ~ 10^9/s, ~10^8 in RAM, (~10 fps)
   GPU ~ MIC **>** CPU (bandwidth-limited to 150 GB/s)

3. Pure CFD = **fluids,** cells: ~10^8/s, ~10^8 in RAM
   GPU ~ MIC **~** CPU (mostly bandwidth limited) , (~1 fps)

GPU – some have decent double precision, most don't.
Somewhat difficult to program and optimize, compared to x86 platforms. Very fast on direct summation.

Collisionless gigaparticle disks can be simulated with 4th order symplectic algorithm

Algorithm: 4th Order Symplectic

Forest and Ruth (1990)

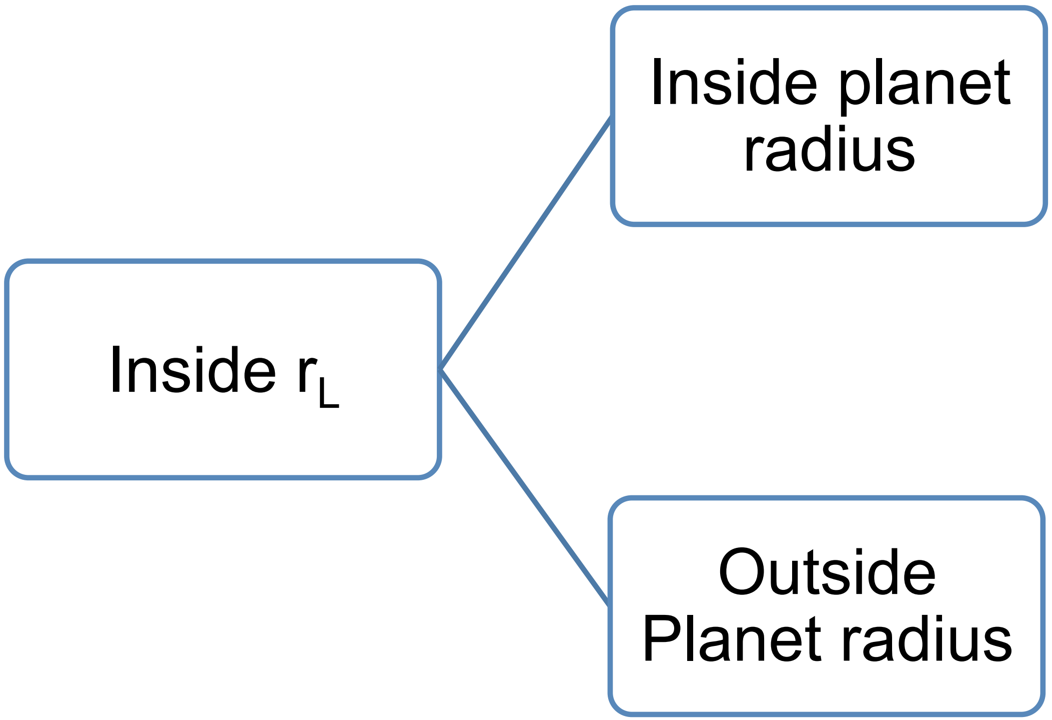
1. Push position: $x_2 = x_1 + c_1$*v
2. Calculate **force** (at updated position)
3. Kick velocity: $v_2 = v_1 + d_1$*a

4. Push position: $x_2 = x_1 + c_2$*v
5. Calculate **force** (at updated position)
6. Kick velocity: $v_2 = v_1 + d_2$*a

7. Push position: $x_2 = x_1 + c_3$*v
8. Calculate **force** (at updated position)
9. Kick velocity: $v_2 = v_1 + d_3$*a

10. Push position: $x_2 = x_1 + c_4$*v

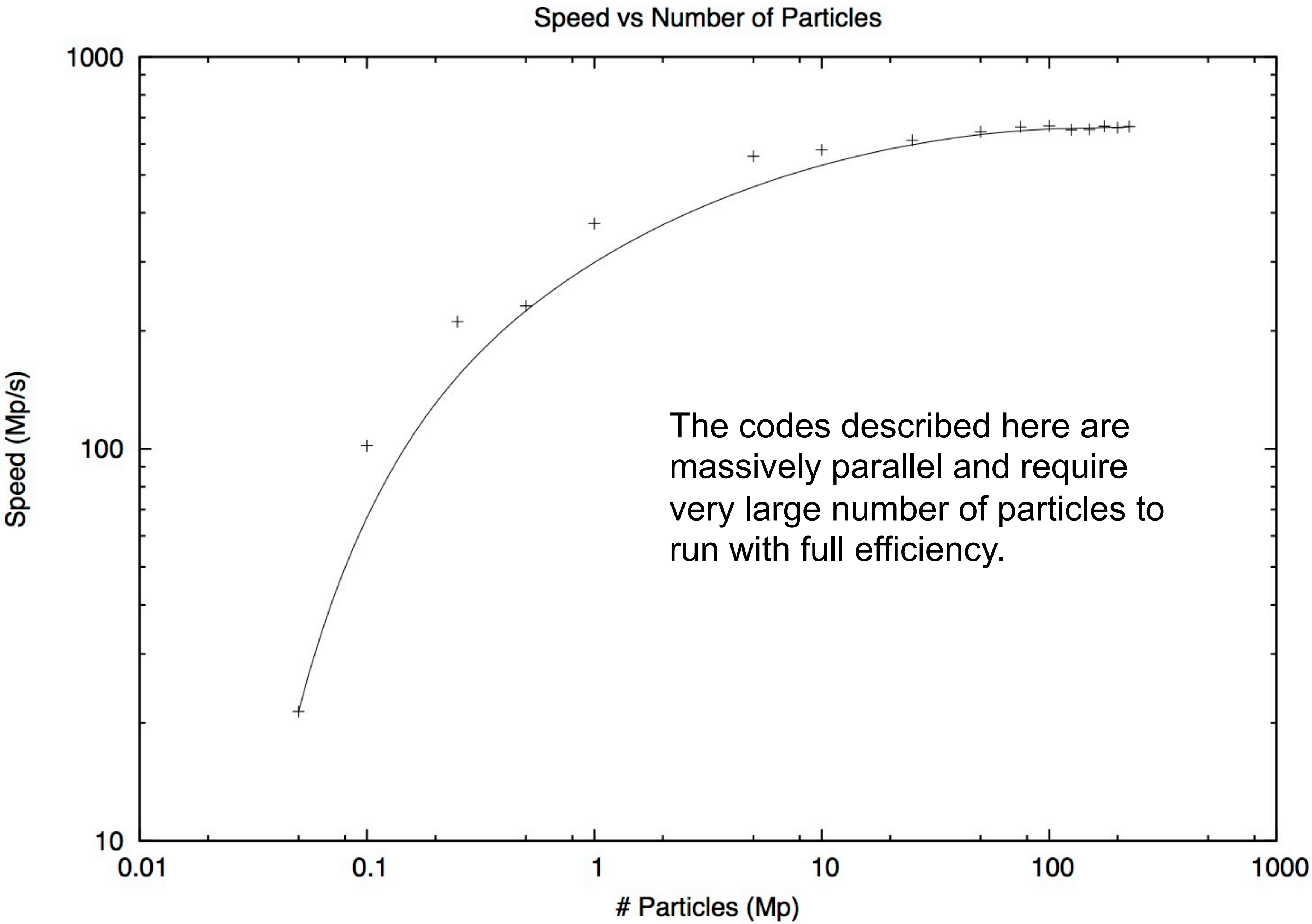
Collision with Binary and Variable dt



- Store particle and set to large r in main array
- Remove from array
- Transfer momentum and cm position
- Increase mass and spin

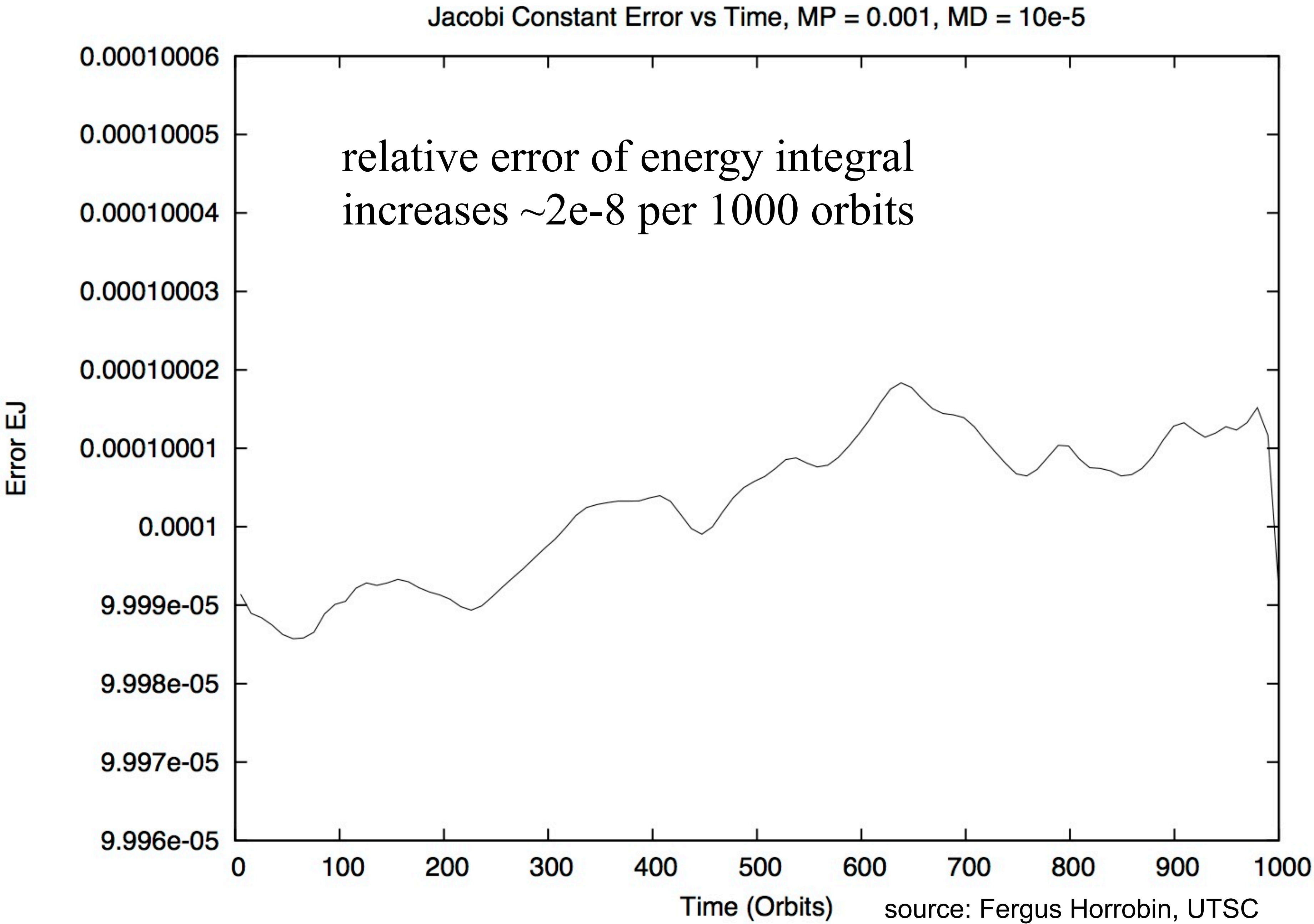
- Store particle and set to larger *r* in main array
- Perform same scheme but with variable dt
- Range 1e-8 – dt_1(0.004)

source: Fergus Horrobin, UTSC



The codes described here are massively parallel and require very large number of particles to run with full efficiency.

source: Fergus Horrobin, UTSC



source: Fergus Horrobin, UTSC

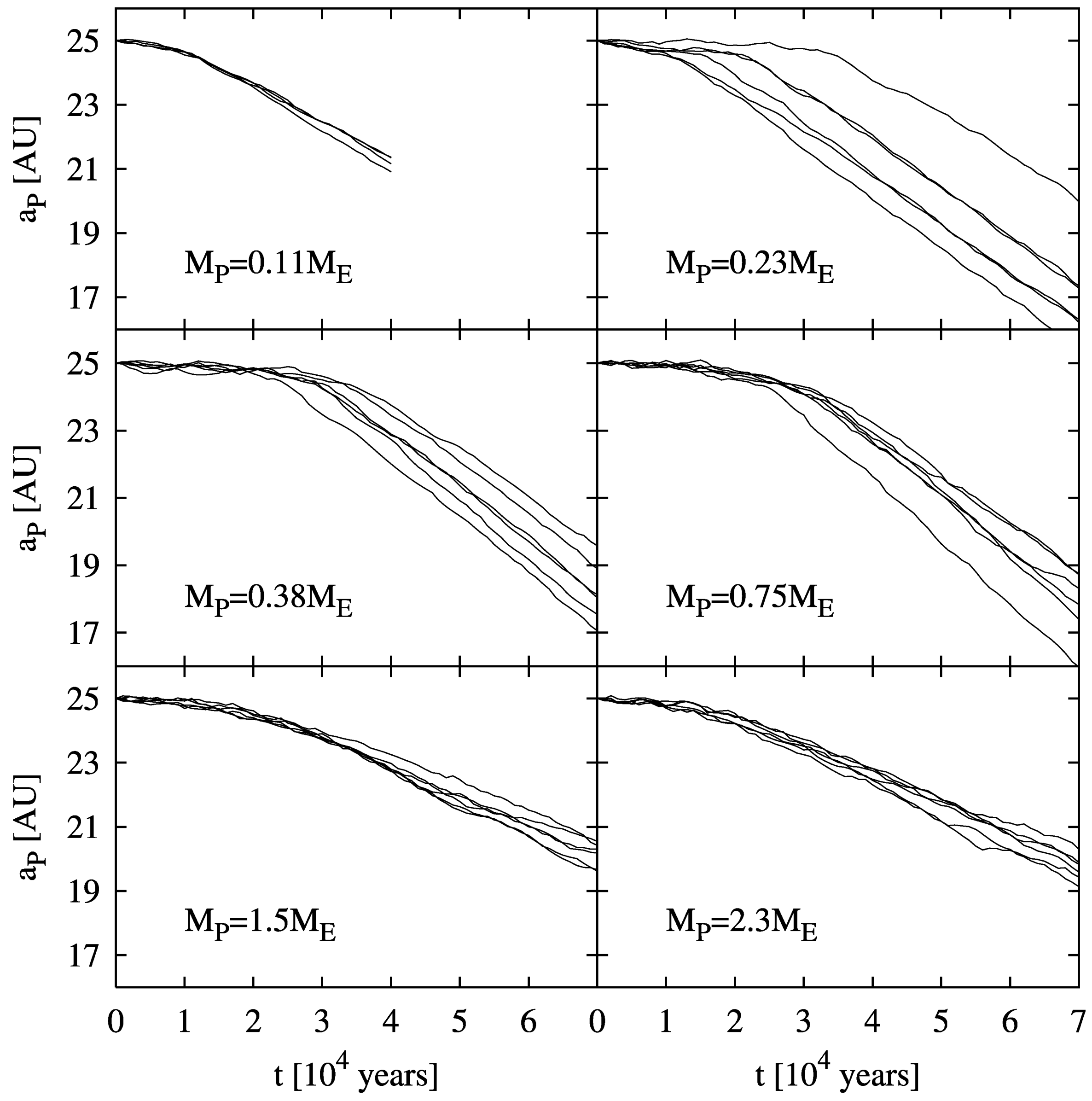
We study type III migration in Disks

- Very rapid migration in *gas* disks: 40-50 orbits timescale for Jupiter-mass planet in a solar nebula disk

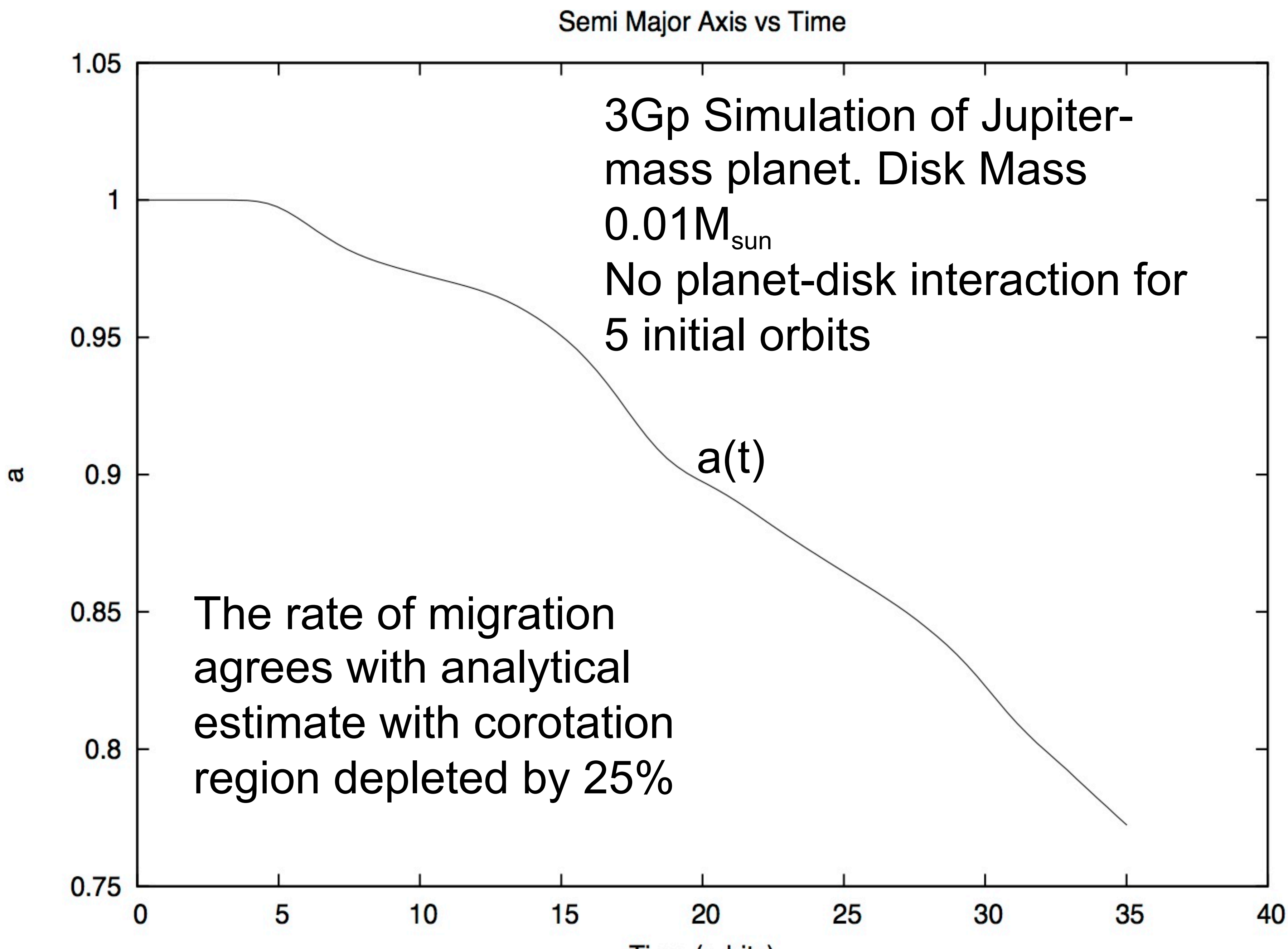
  (Papaloizou et al. in Protostars and Planets V, 2005)
- Rate does not depend on mass of planet
- Criterion compares disk (in CR = corotation region) and planet masses:
  - $M_p < M_{deficit}$. Difficult to satisfy by planetesimals...

source: Fergus Horrobin, UTSC

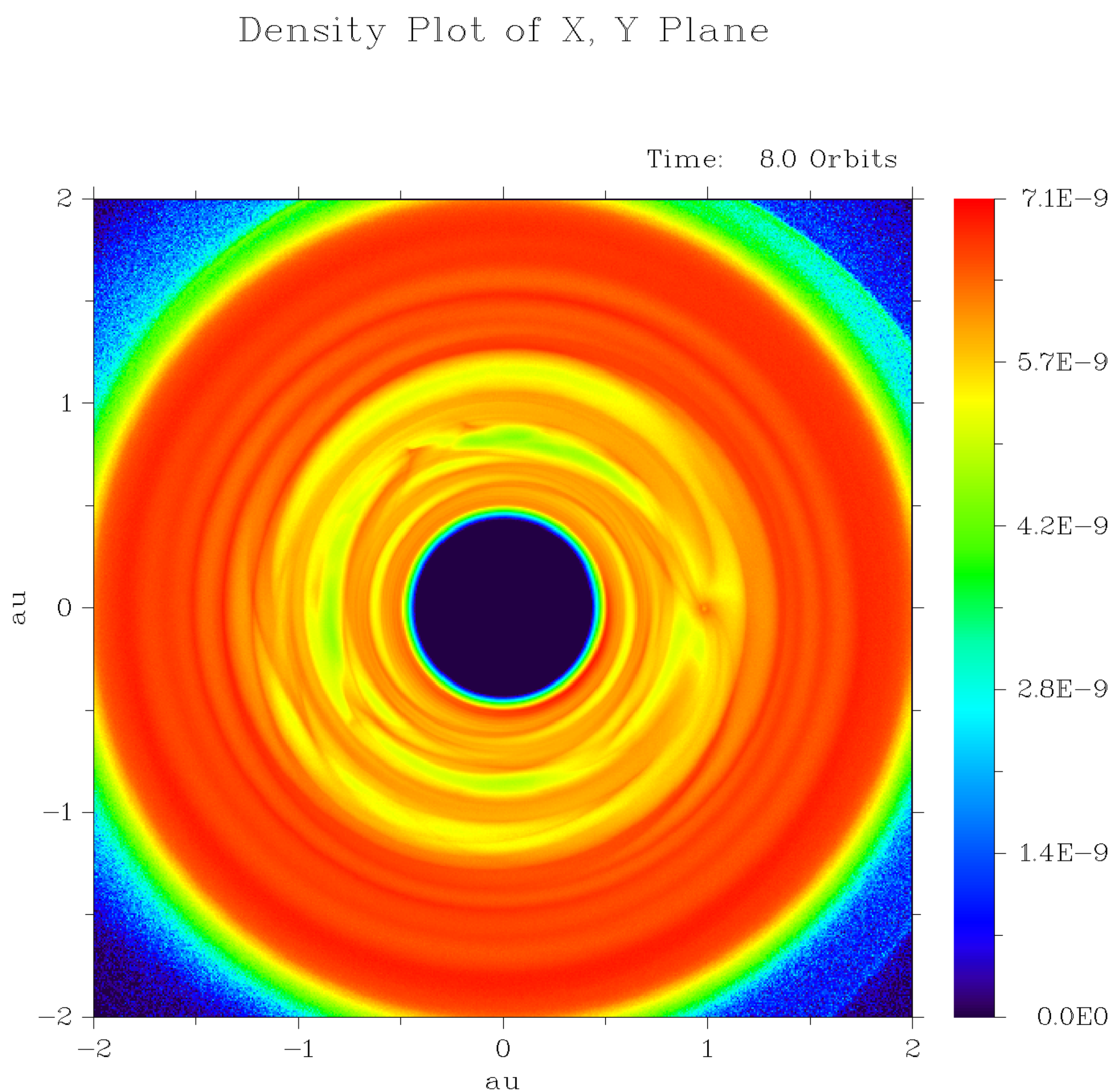
Previous results: Kirsh et al. 2009 identified the fast migration and offered an explanation [without noticing a connection with type III migration, e.g. as reviewed by Papaloizou et al. 2006, PP V]



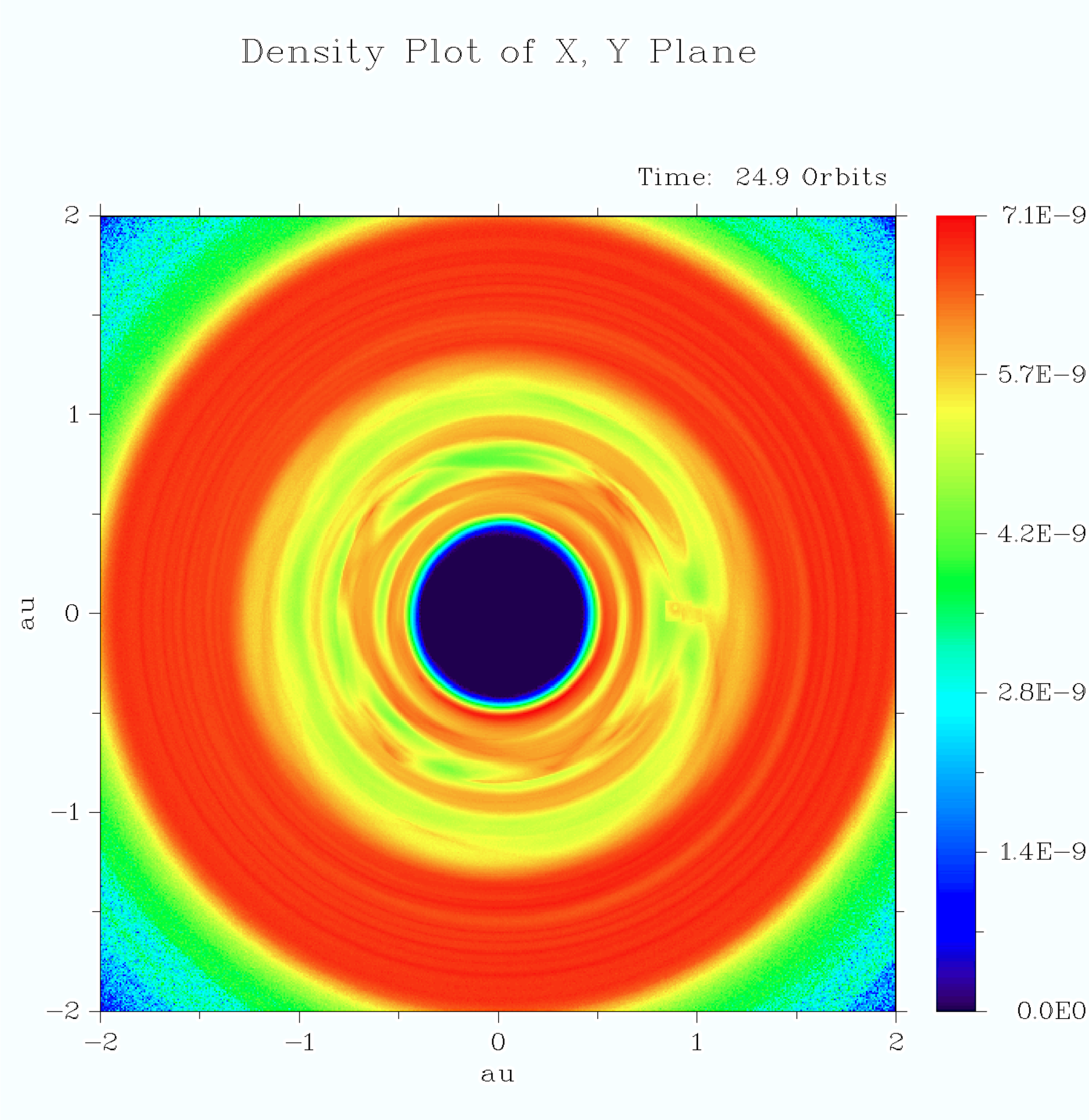
Much slower migration by mean-motion resonant scattering (w/similarly v. massive disks) proposed by Murray et al (1998).



source: Fergus Horrobin, UTSC



source: Fergus Horrobin, UTSC



source: Fergus Horrobin, UTSC

Conclusions of Fergus Horrobin's summer research in 2017

For large-scale particle integrations in non-collisional disks, codes can run v. fast on MIC cluster (Xeon Phi)

- 3+ billion particles (150M per MIC), timestep ~0.2 s
- Hybrid parallelization method combining OpenMP and MPI seems best for this type of platform
- We've implemented 4th order symplectic integrator.
- Though deeper analysis must be made, we see similarities between gas and particle disks in the context of rapid migrations

N-body simulations of the Universe

- https://www.youtube.com/watch?v=YjUICiYICYE
- Millenium – 10+G particles Gadget code,
- kept the main supercomp at MPI Inst of Astronomy in Garching, Germany, busy for a month in 2004
- (700 MPc)^3

- https://www.youtube.com/watch?v=32qqEzBG9OI

(350Mpc)^3, 5e4 galaxies, 12G particles, 8k CPUs

- Millenium XXL

N-body simulations of the Universe

- https://www.youtube.com/watch?v=YjUICiYICYE
- https://www.youtube.com/watch?v=32qqEzBG9OI

$(350\text{Mpc})^3 = (1\text{ billion ly})^3$, 50K galaxies, 12G particles

- Simulation name: Bolshoi
- Run on Pleiades cluster (supercomputer) at NASA Ames Research Center in Mountainview, California.

N-body simulations of the Universe

12G particles create 50000 galaxies,   gas: Adapt. Mesh Refinement grid,   8k CPUs used for Bolshoi-Planck simulation

Pleiades has theor. peak performance 7.3 PFLOPS

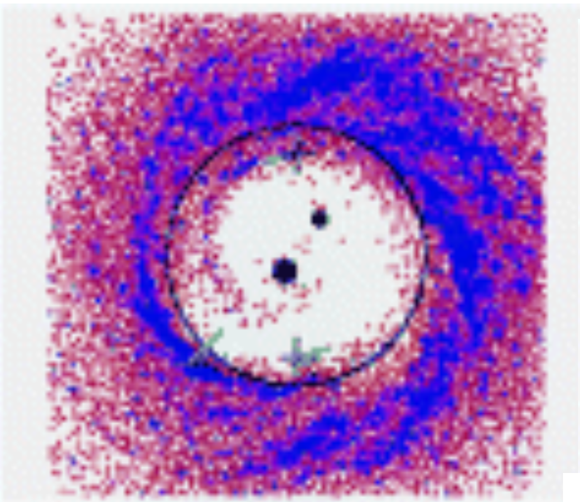
1. Astrophysical problems for CPU and GPU calc’s:

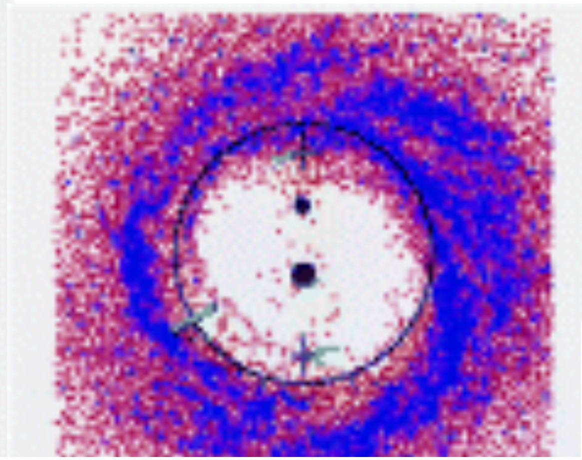
- Disk-planet interaction and migration
- Disks with structure: IRI (irradiation instability in particle and gas disks)
- Flow of gas around Super-Earth (5 M_E)

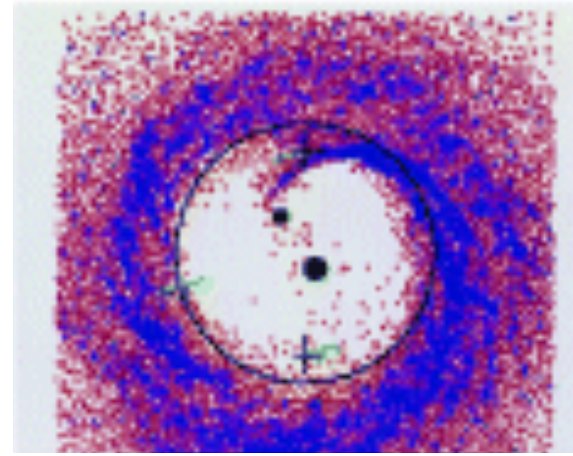
2. Massively parallel numerics on mini-supercomputers:

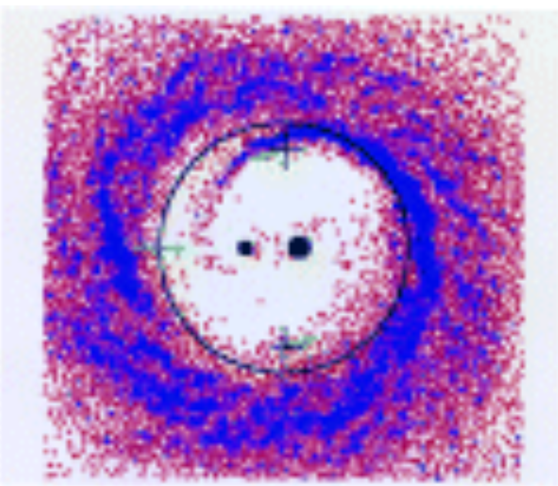
- Comparison of HPC platforms: CPU, GPU, and MIC (Φ)
- UTSC clusters

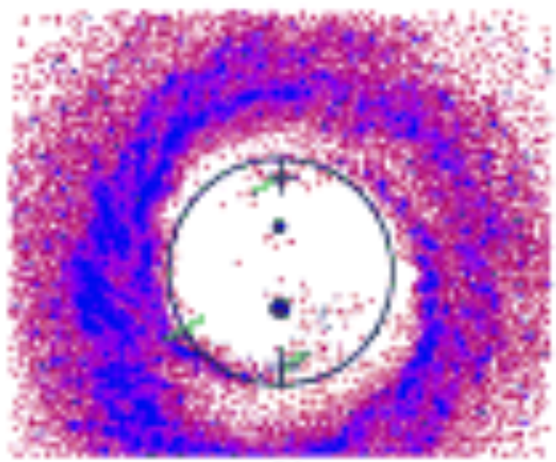
Binary-disk interaction

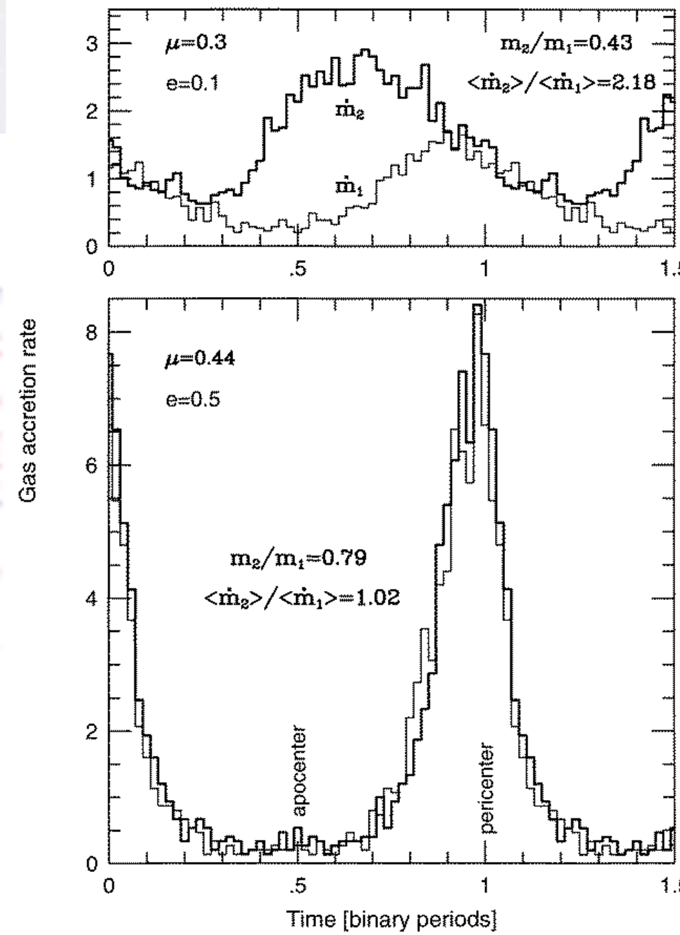










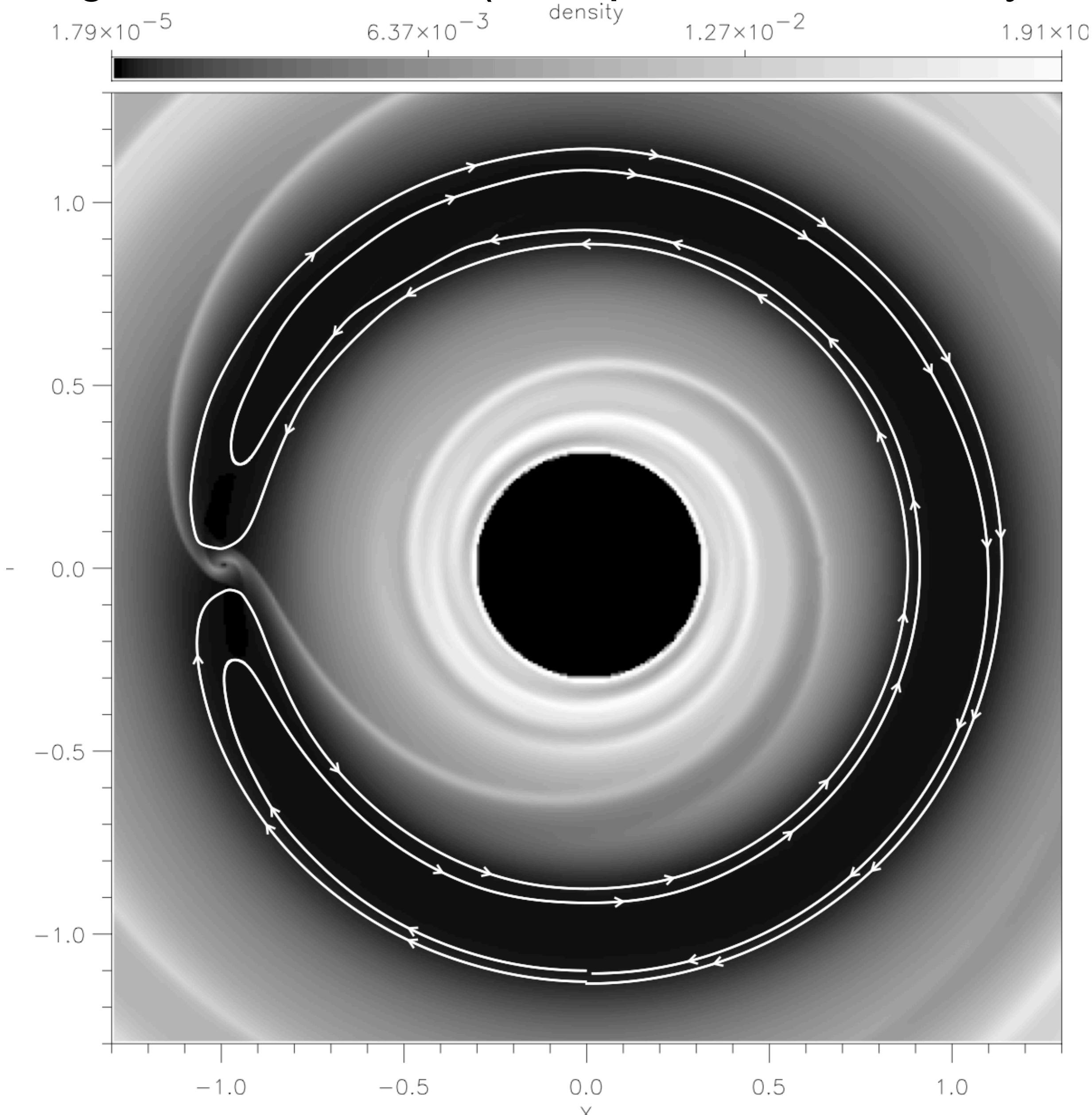


SPH = smoothed hydrodynamics: cf. wiki

Artymowicz and Lubow (1996)

Binary-disk interaction

method: grid-based CFD (Computational fluid dynamics)

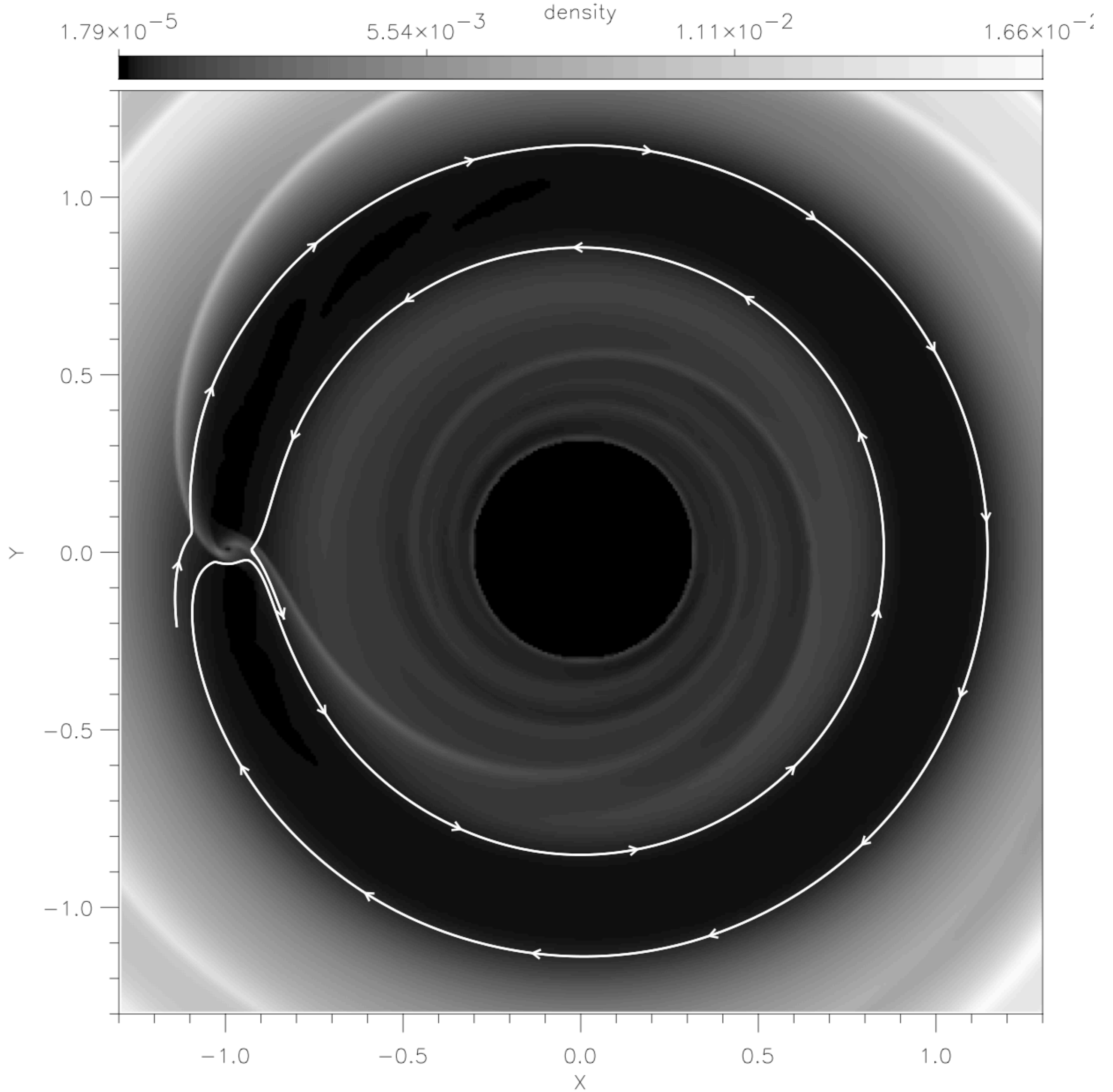


CPU 2-d

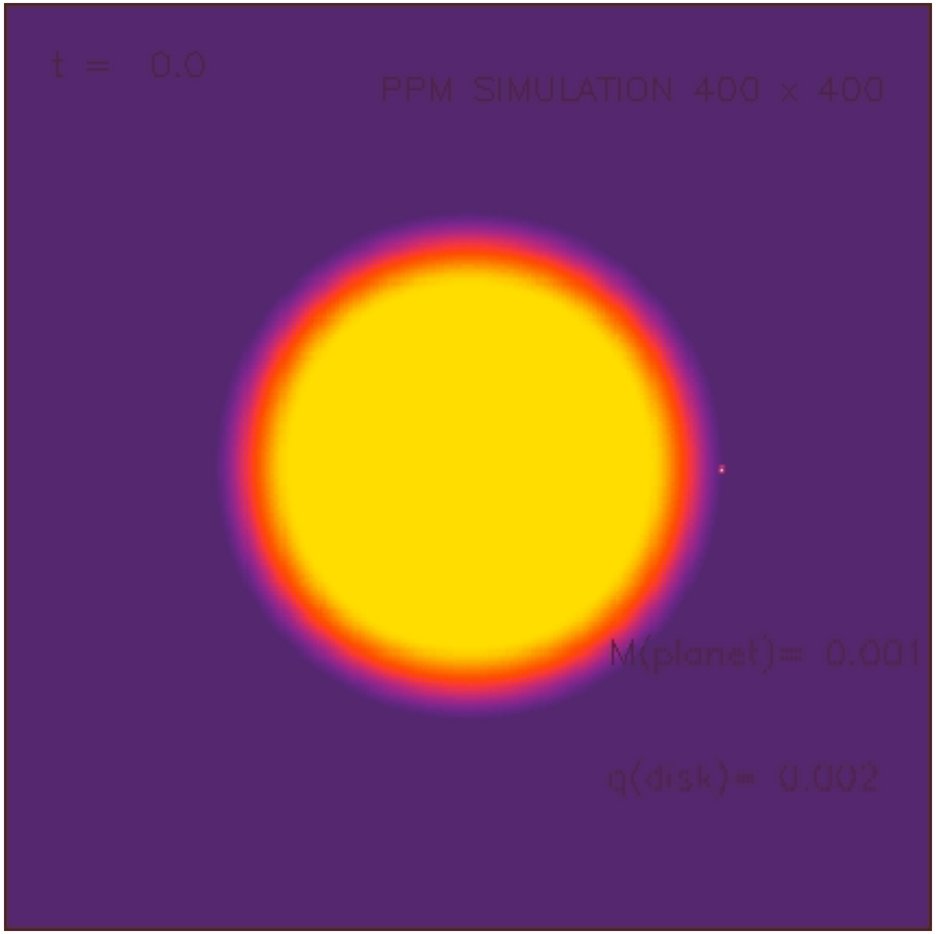
2nd order ZEUS hydro

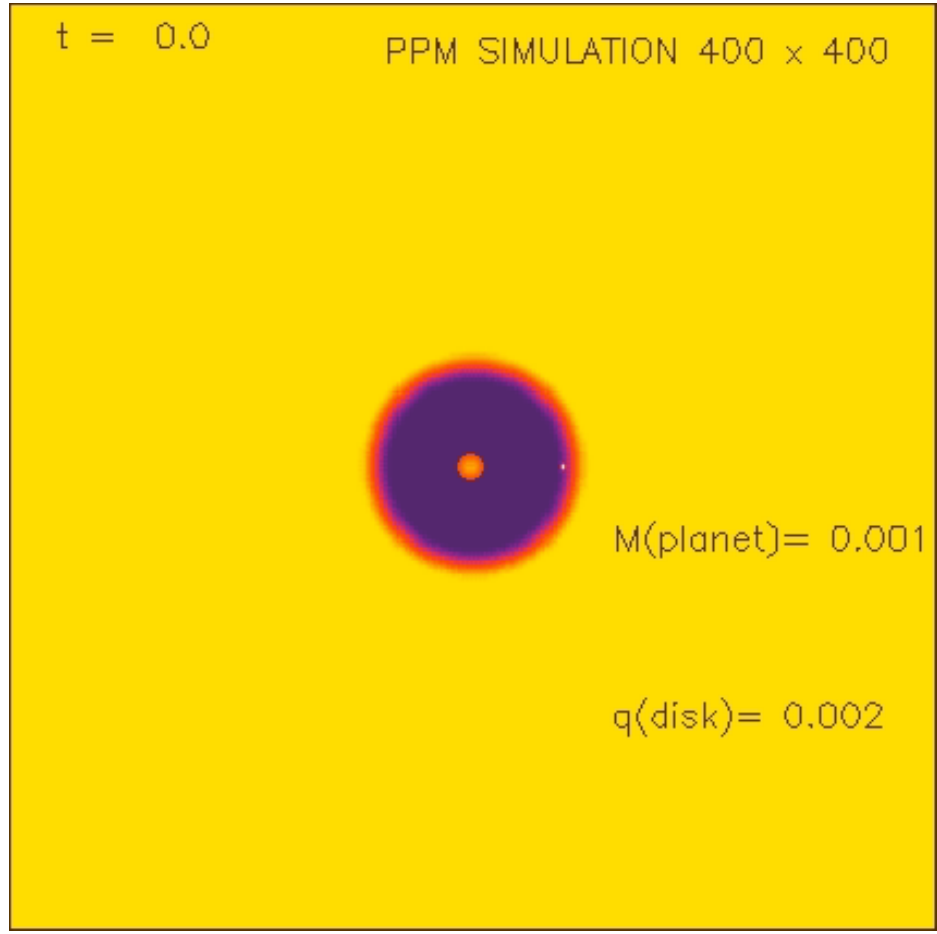
notice mass flow through gap

LUBOW, SEIBERT, & ARTYMOWICZ



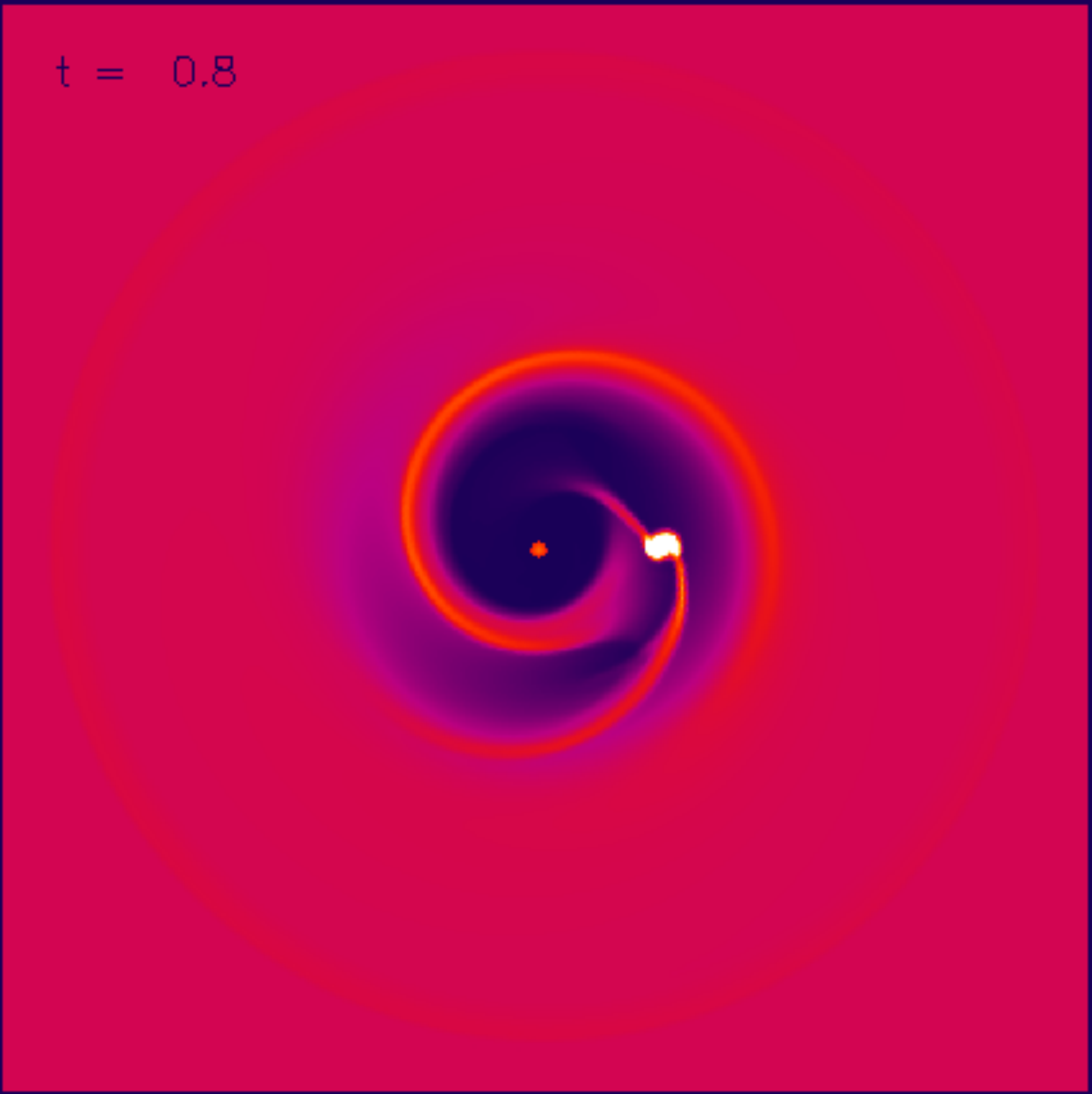
One-sided disk (**inner/outer disk only**). The rapid inward migration is OPPOSITE to the expectation based on shepherding (Lindblad resonances).

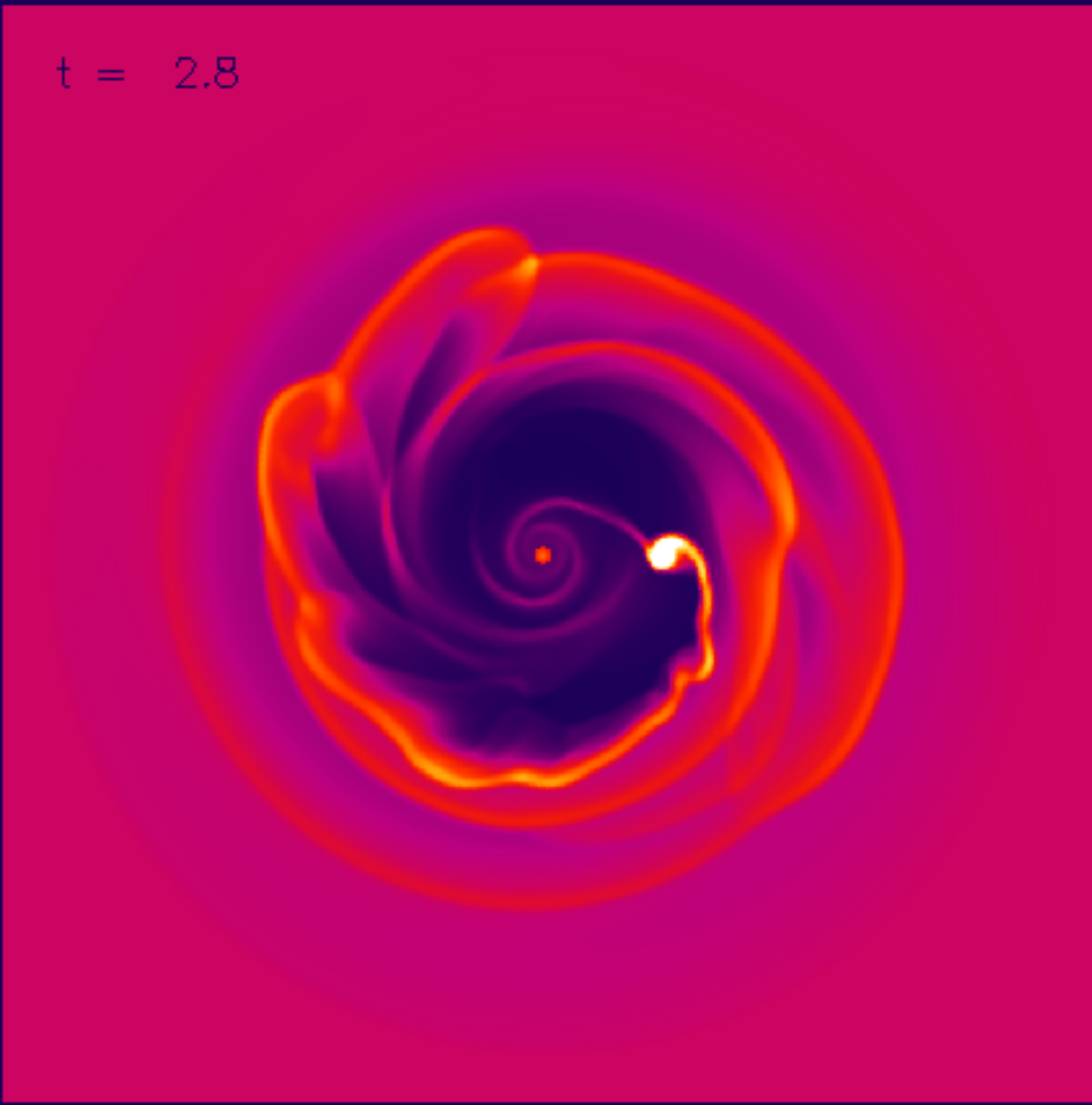


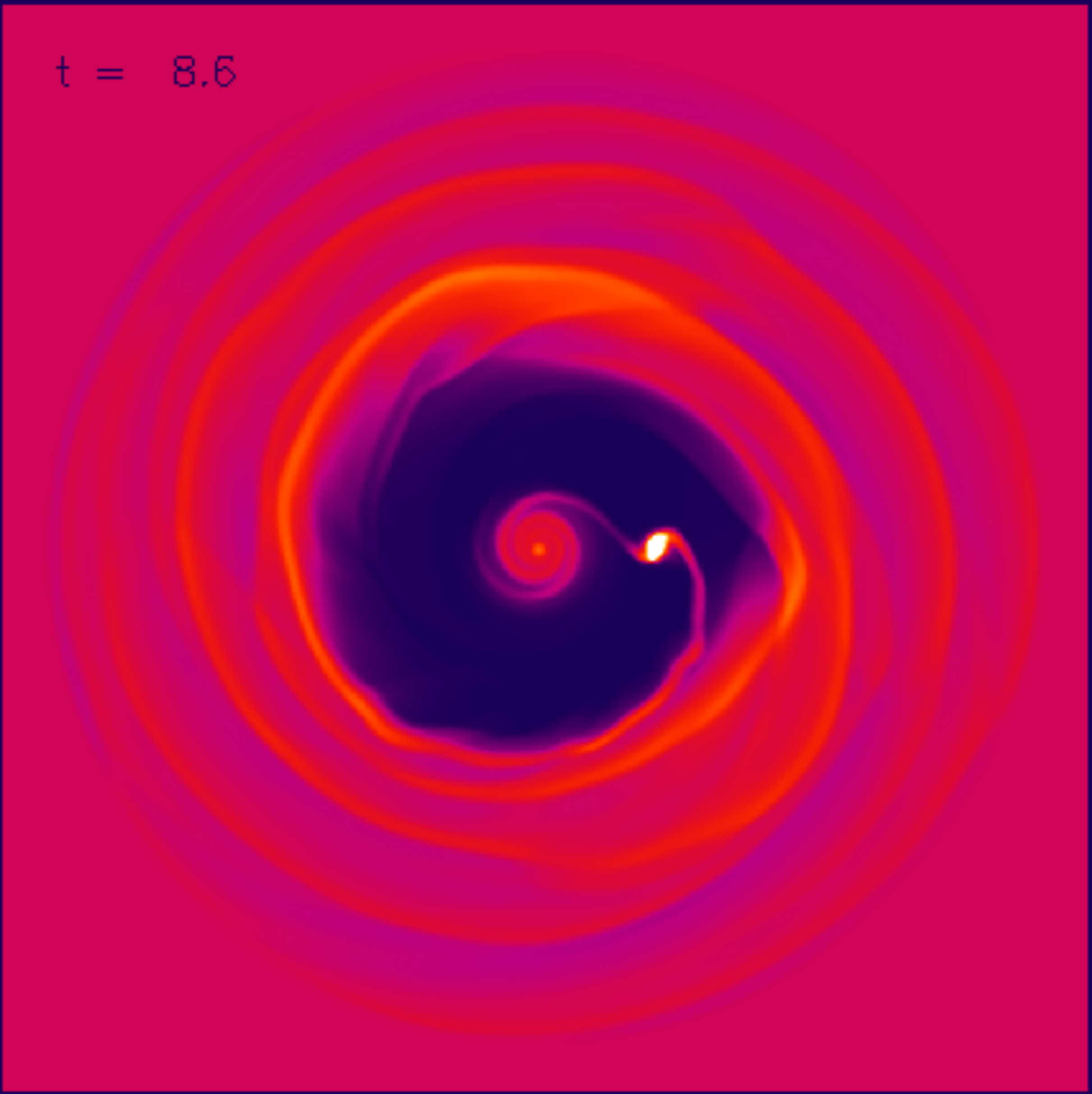


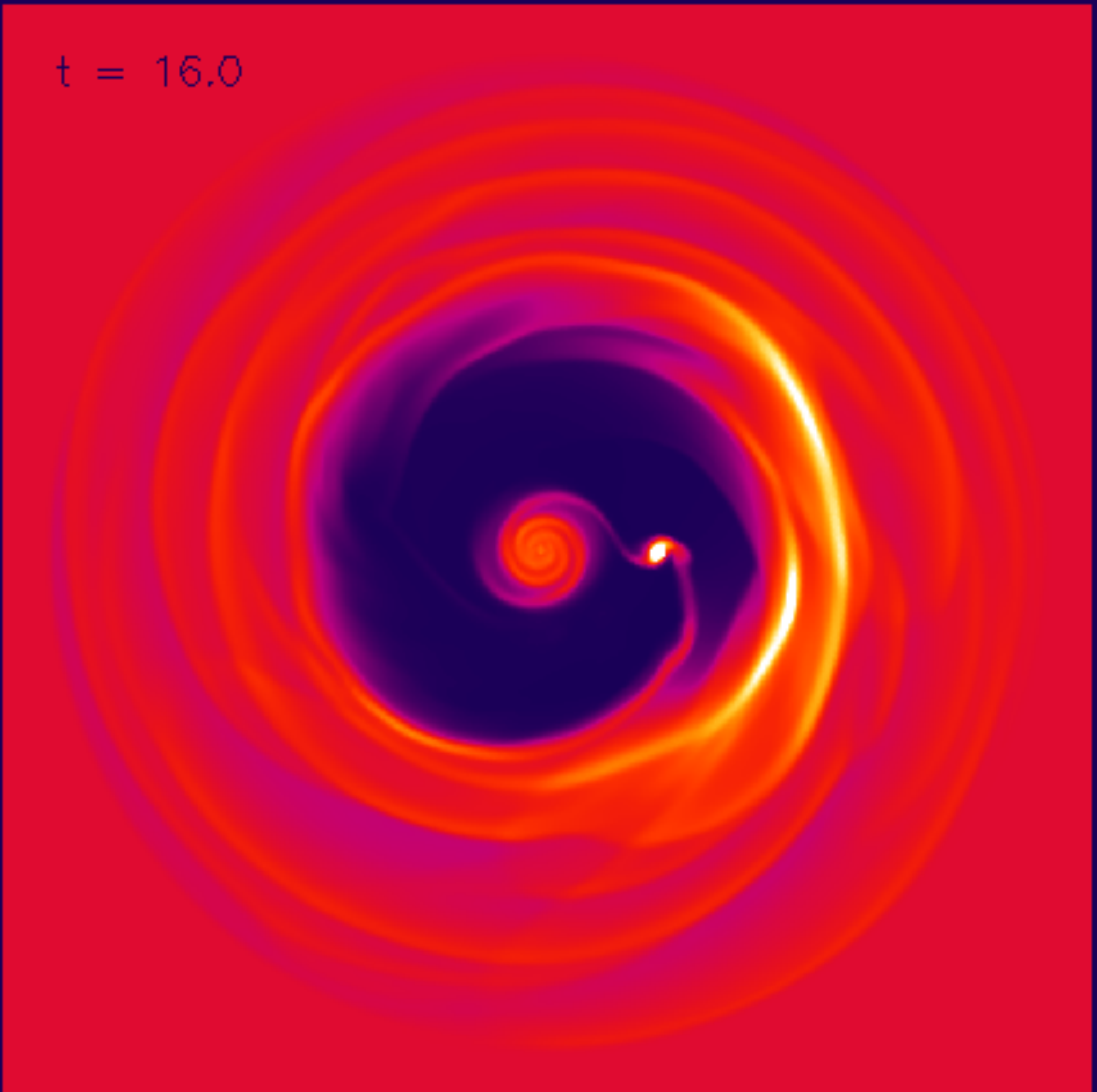
Like in the well-known problem of “sinking satellites” (small satellite galaxies merging with the target disk galaxies),
Corotational torques cause rapid inward sinking.

A few snapshots from a 2-D simulation of a brown dwarf circling a
interacting with the circum-binary disk.   Density of gas is color-co

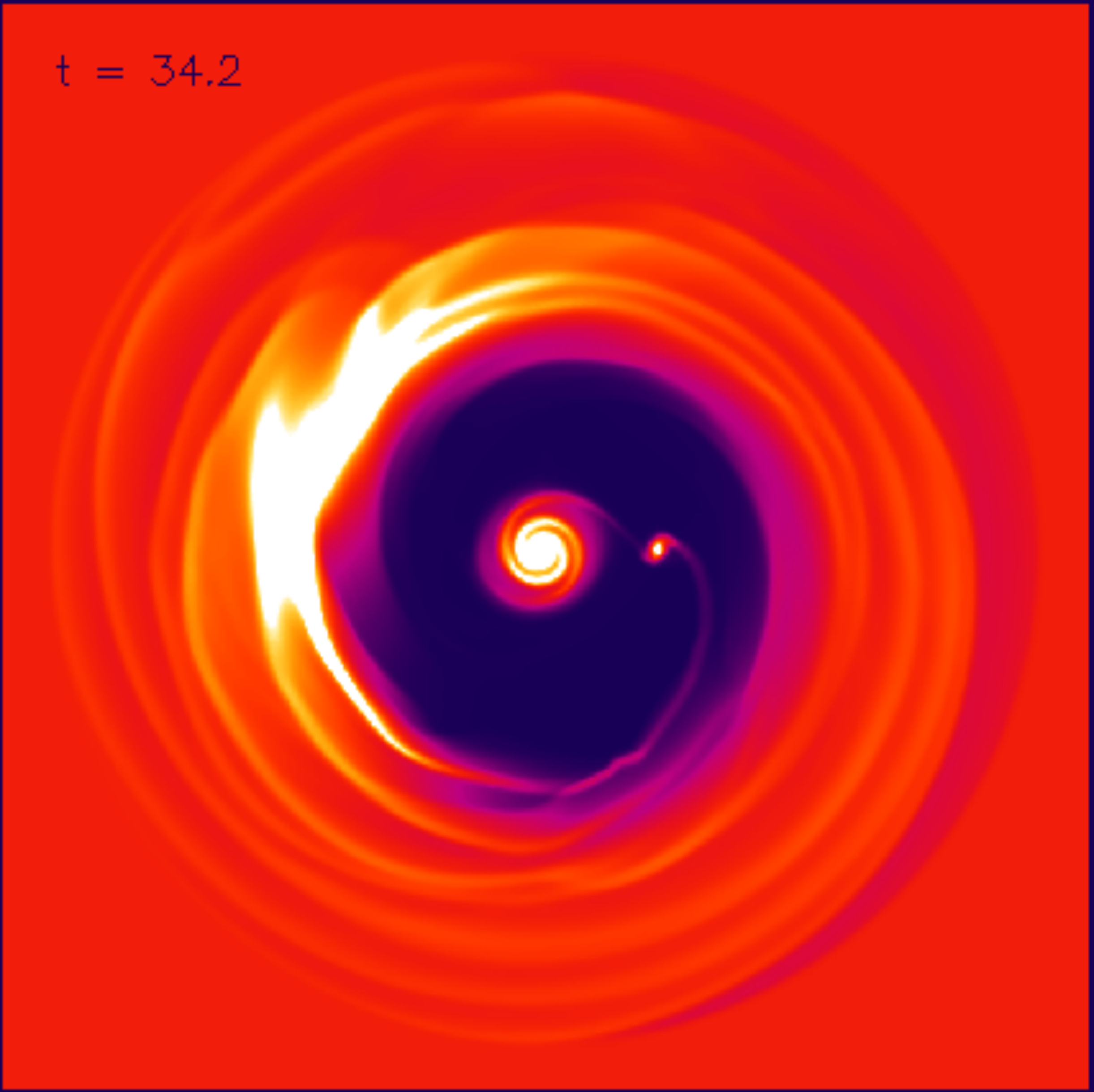



t = 2.8

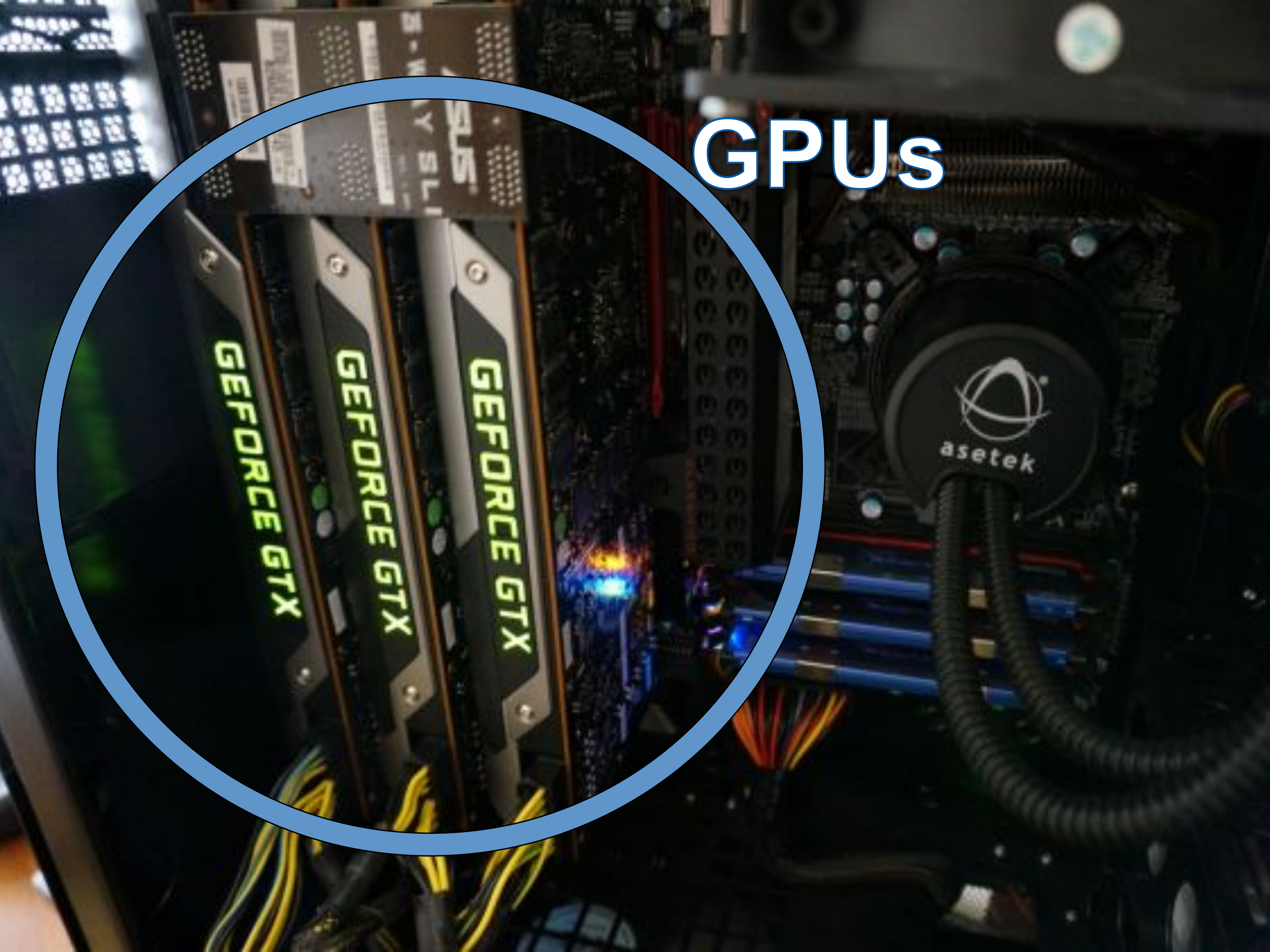

t = 8.6


t = 16.0

An edge mode of spiral density waves appears, grows non-linear, a
forms a vortex-like structure in disk. Density of gas is color-coded



This computation used C
Fortran hydrocode algo
PPM (Piecewise Parabo


GPUs
GEFORCE GTX
GEFORCE GTX
GEFORCE GTX
asetek

The 3-D flow around a small, embedded planet

J. Fung, P. Artymowicz and Y. Wu (ApJ, Nov 2015)

Code: PenGUIn.

CUDA C++. Processes up to ~20 Mcells/s (dp), ~40 Mcell/s (sp)

for comparison, Xeon Phi can run the same size problems at ~30 Mcell/s (sp)

and a modern 6-core CPU does ~28 Mcell/s.

These codes are bandwidth-bound. GPU > MIC ~ CPU

We have found big differences between 2-D and 3-D flow pattern of gas from a protoplanetary disk around a planet. These differences may influence the way planets form and migrate in disks.

2-D

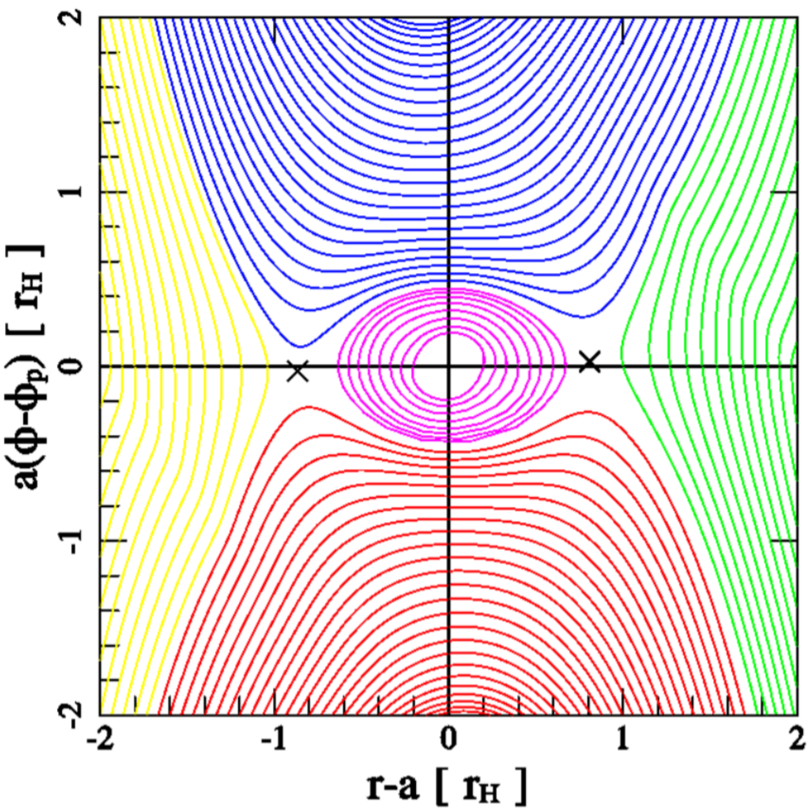


FIG. 1.— Streamlines around a planet in 2D, plotted in the corotating frame

3-D

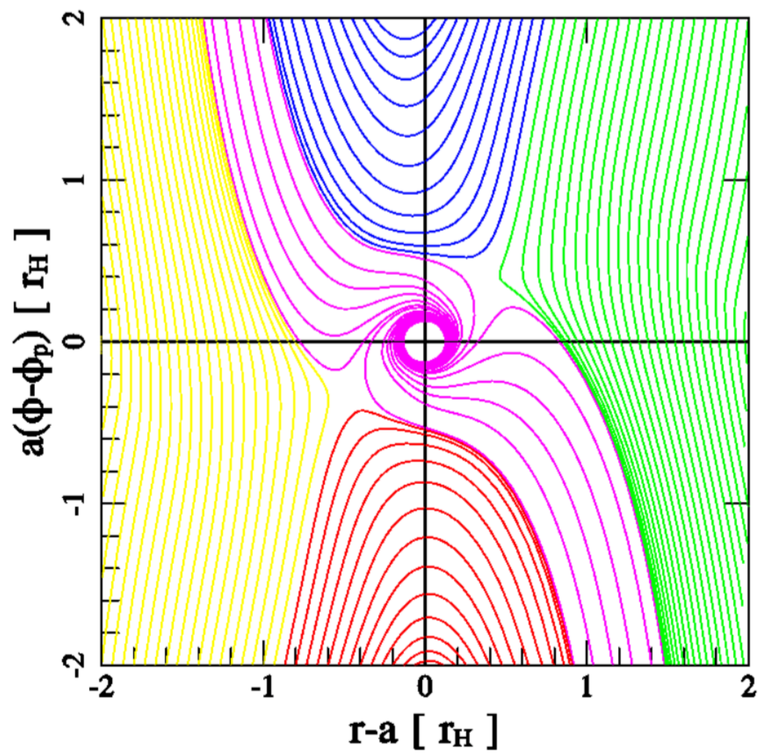
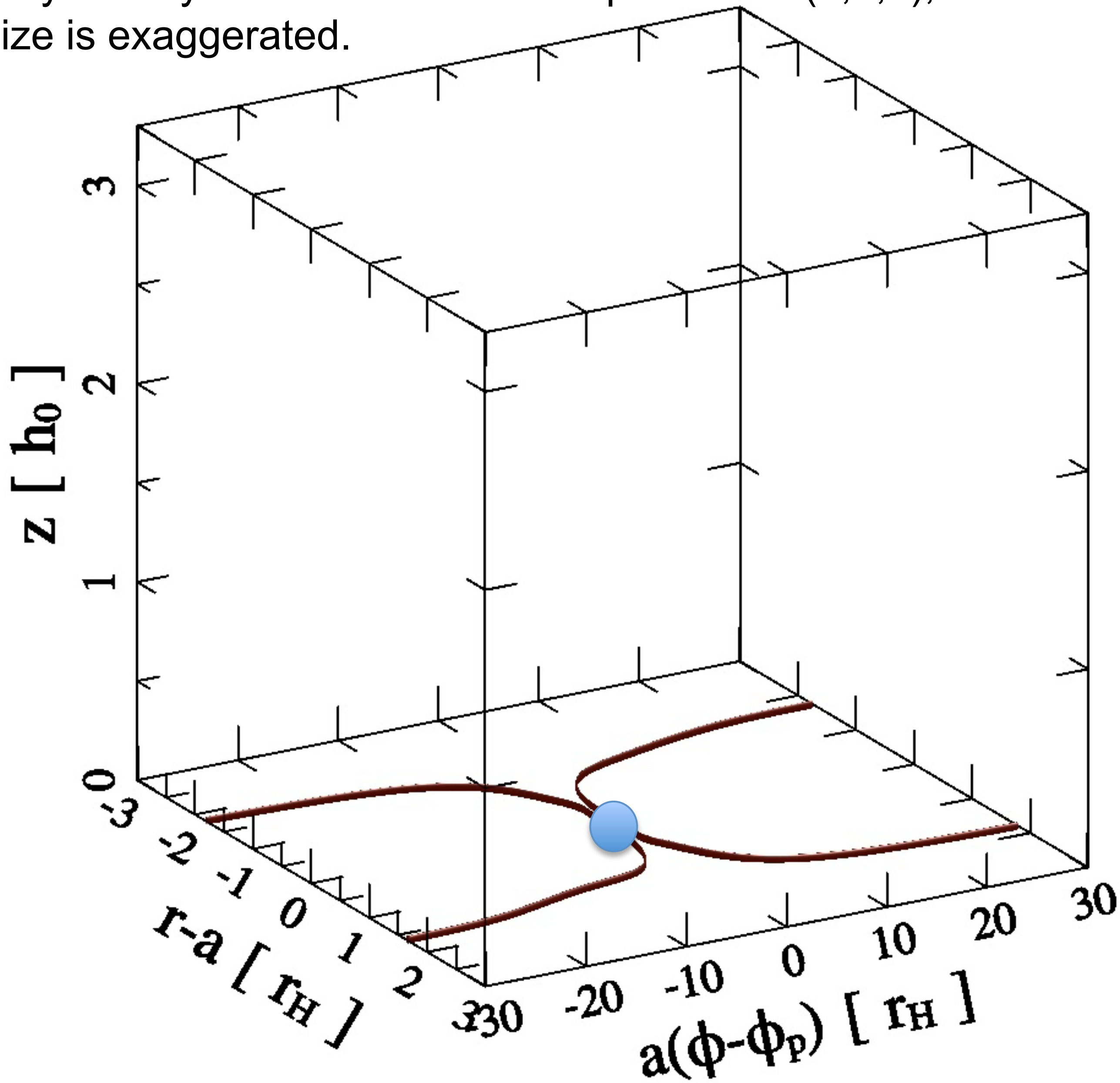
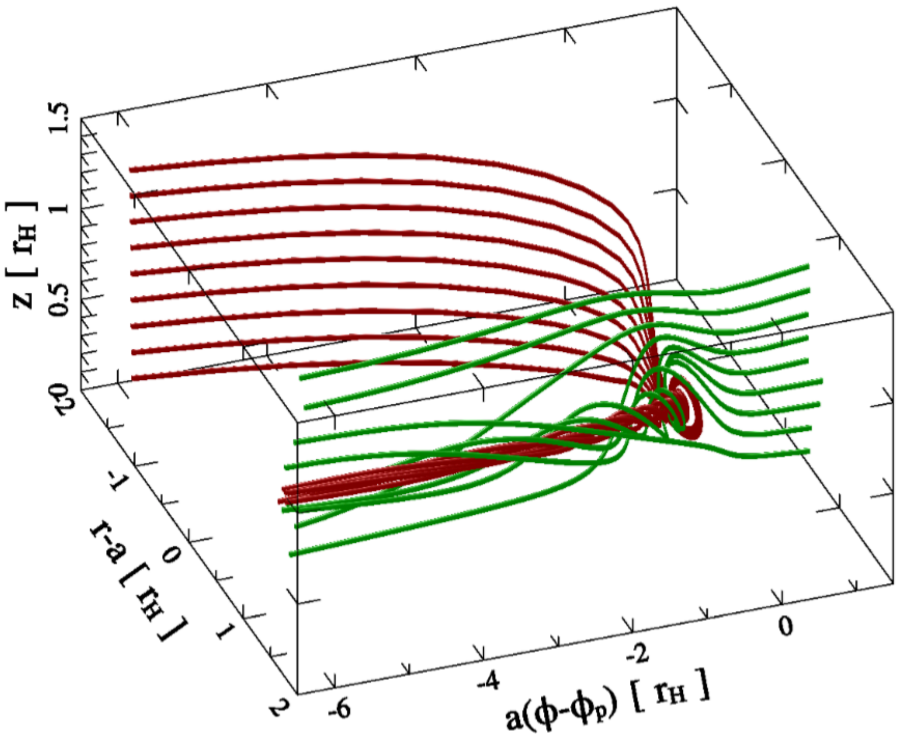


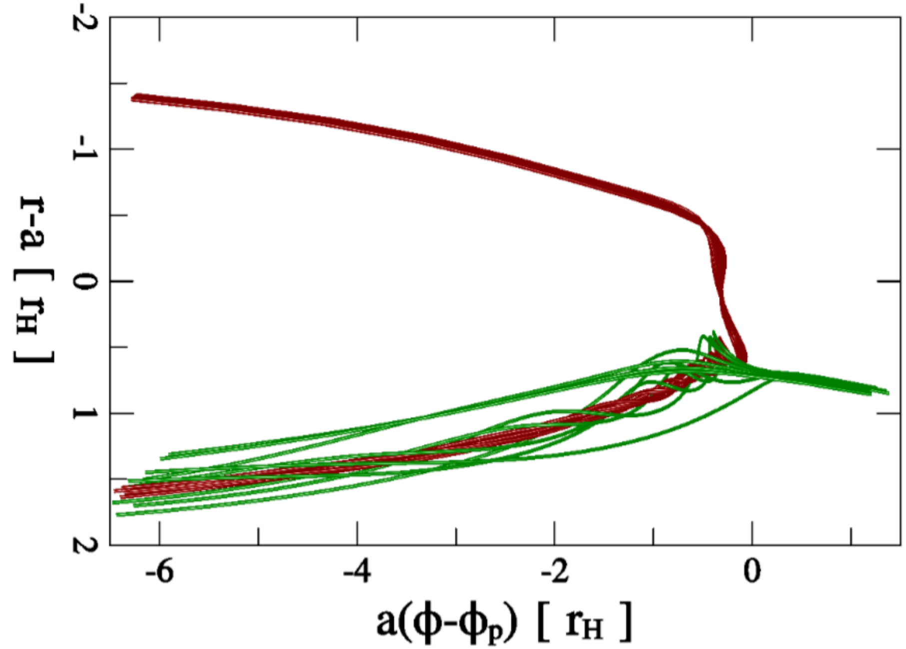
FIG. 8.— Streamlines in the disk midplane. Compare with Figure 1 for differences between 2D and 3D flow. Yellow, red, green, and blue streamlines are assigned in the same manner as Figure 1. Unlike Figure 1, magenta lines are outflows away from the planet, pulled down from initially higher altitudes. They reach as close as $1.5 r_s$ from the planet and are unbound.

Computational box, the bottom part of which lies in the disk midplane.
Up-down symmetry is assumed. Planet's position is (0,0,0), indicated by a circle)
but the size is exaggerated.



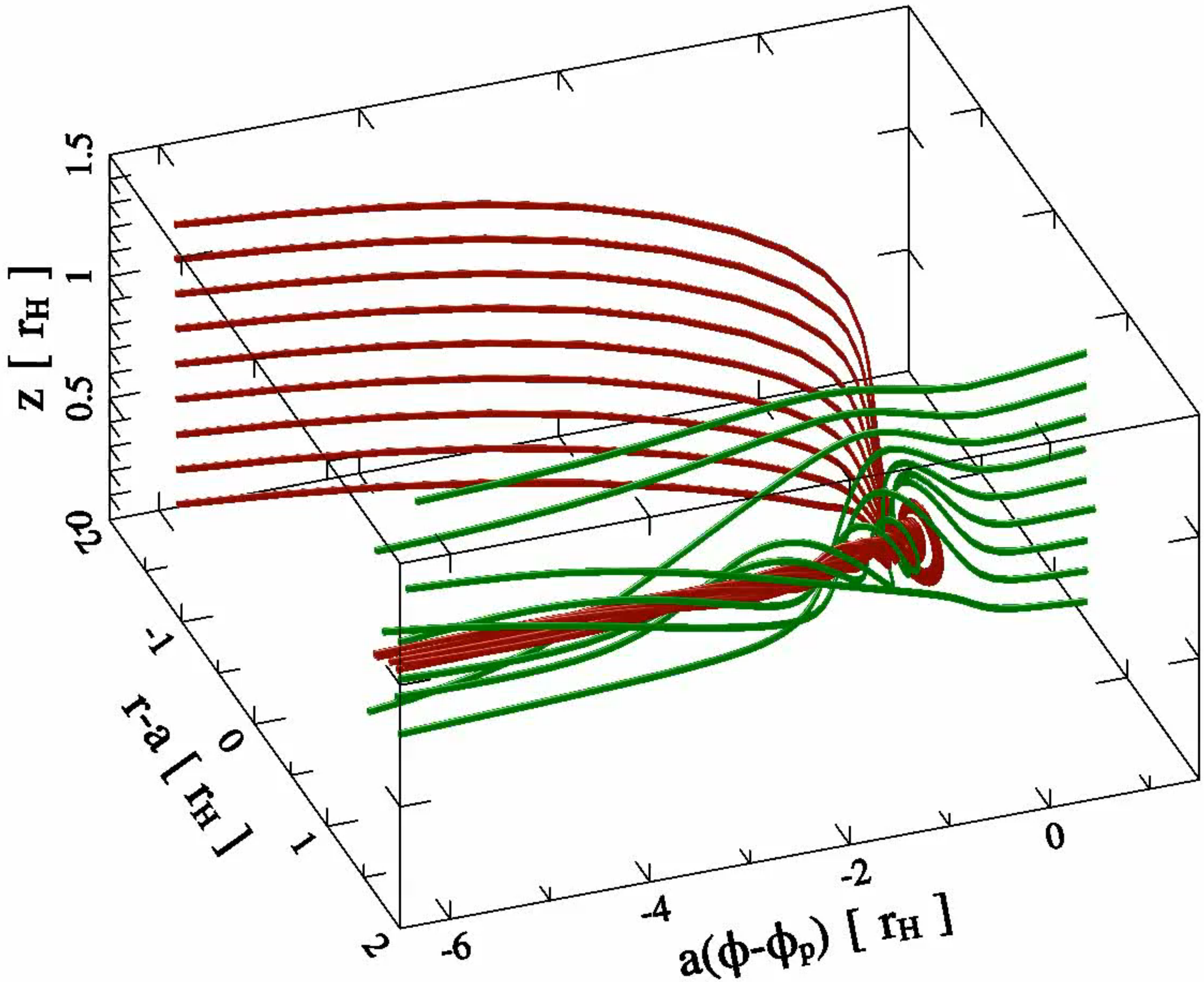
Gas flow approches planet on one side of the disk (say, further from the star than the orbit of the planet) and after curious vertical compression (into a vortex) departs on the other side (closer to the star than planet).

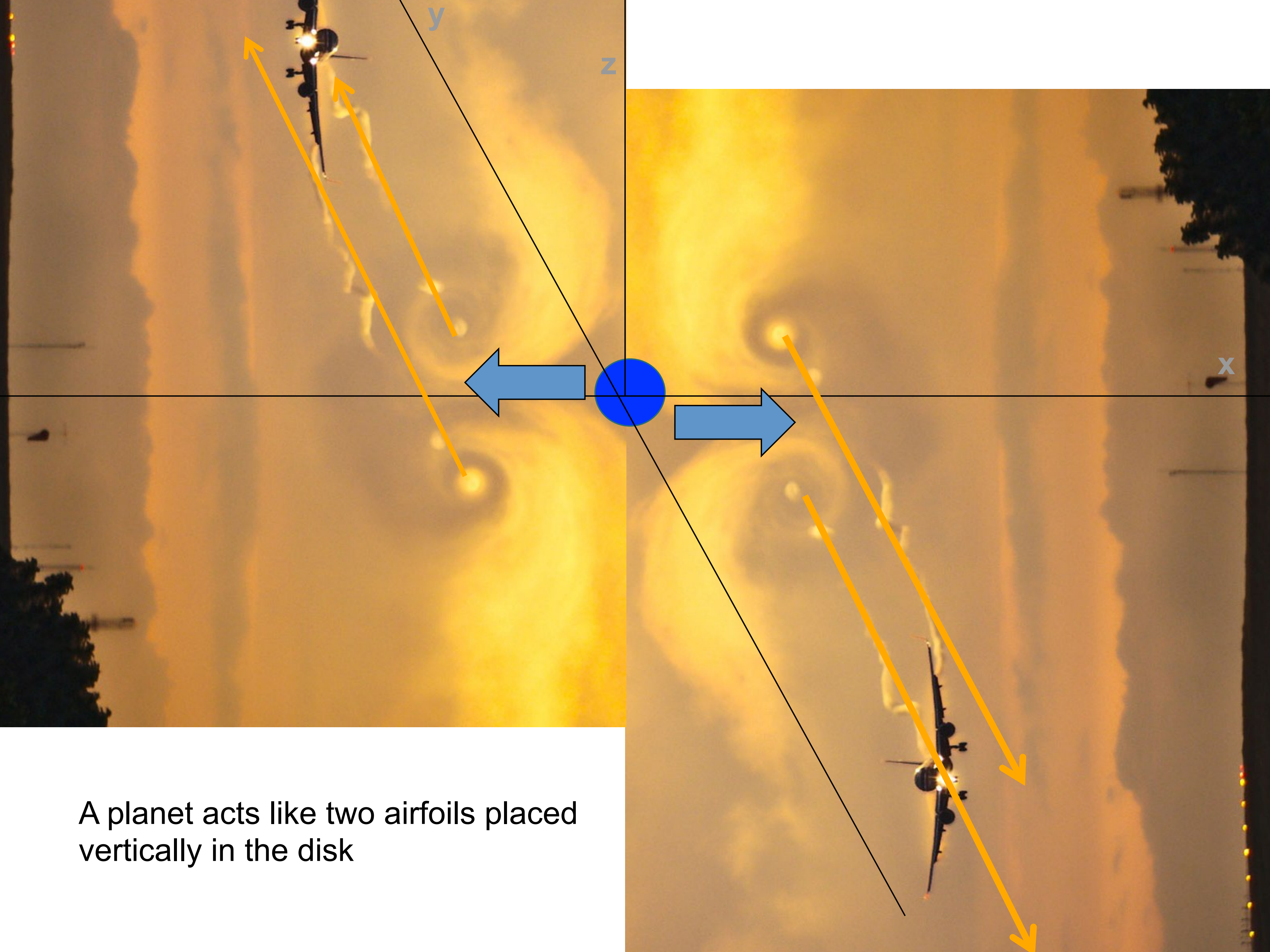




top view

Such vortex, because of up-down and far-near symmetry is found near the protoplanet in 4 copies (2 counterrotating pairs). A planet sheds vortices familiar to flows patterns in aerodynamics, where there are called wingtip vortices.





A planet acts like two airfoils placed vertically in the disk

Novel results from 3D simulaions

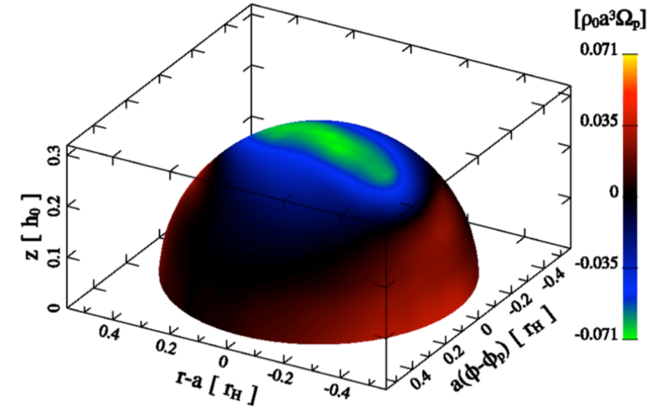


FIG. 9.— Mass flux across the surface of a sphere centered on the planet. The sphere has a radius of $0.5r_B$. Blue and green indicate influx; red and yellow are outflux. The speed of the downward flow is about $0.7c_s$ in this plot, while the two radial outward flows in the midplane (one not visible from this viewing angle) each has a speed of $\sim 0.2c_s$, as is explained in Appendix A. Match this figure with Figure 8 for a more complete view of the flow topology near the midplane.

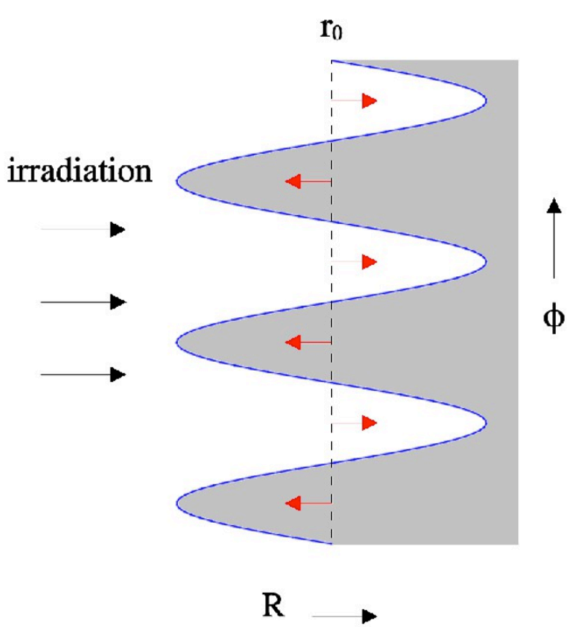
New 3D phenomena, absent in 2D flows, including new columnar topology

vorticity generation mechanism around a small planet, have a potential to resolve the long-standing problems in planet formation theory:

migration and cooling/contraction of the growing planet, occasional transmutation into a giant gaseous planet.

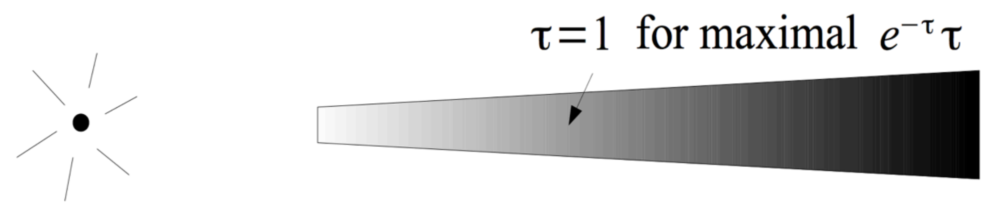
DUST/RADIATION PRESSURE-RELATED INSTABILITIES
including the IRI = IrRadiation Instability

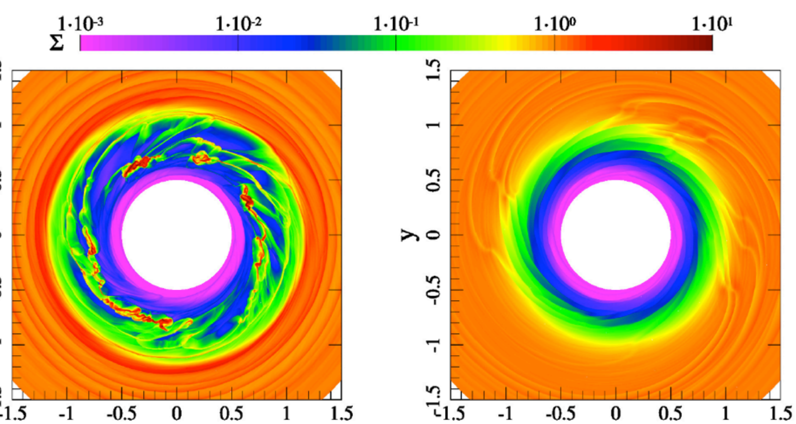
IRI in 2D



Jeffrey Fung (UC Berkeley)
used workstations at UofT with 3 GPUs
for parallel computations

IRI in 2D





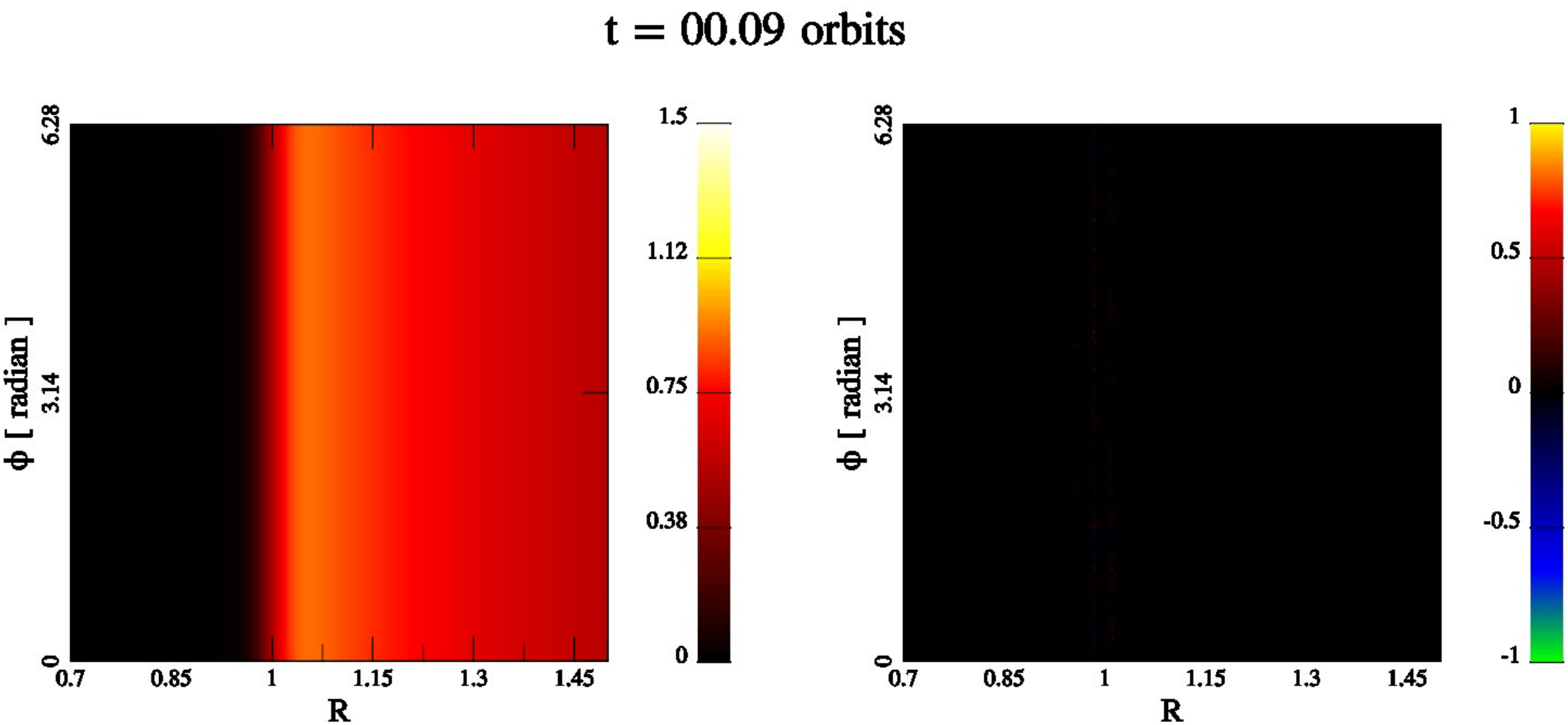
Criterion for instability: $\beta e^{-\tau}\tau \frac{d\ln[r\Re]}{d\ln r} > 1$

$$\Re \equiv \frac{\Sigma\Omega_k \beta e^{-\tau}}{\kappa^2}$$

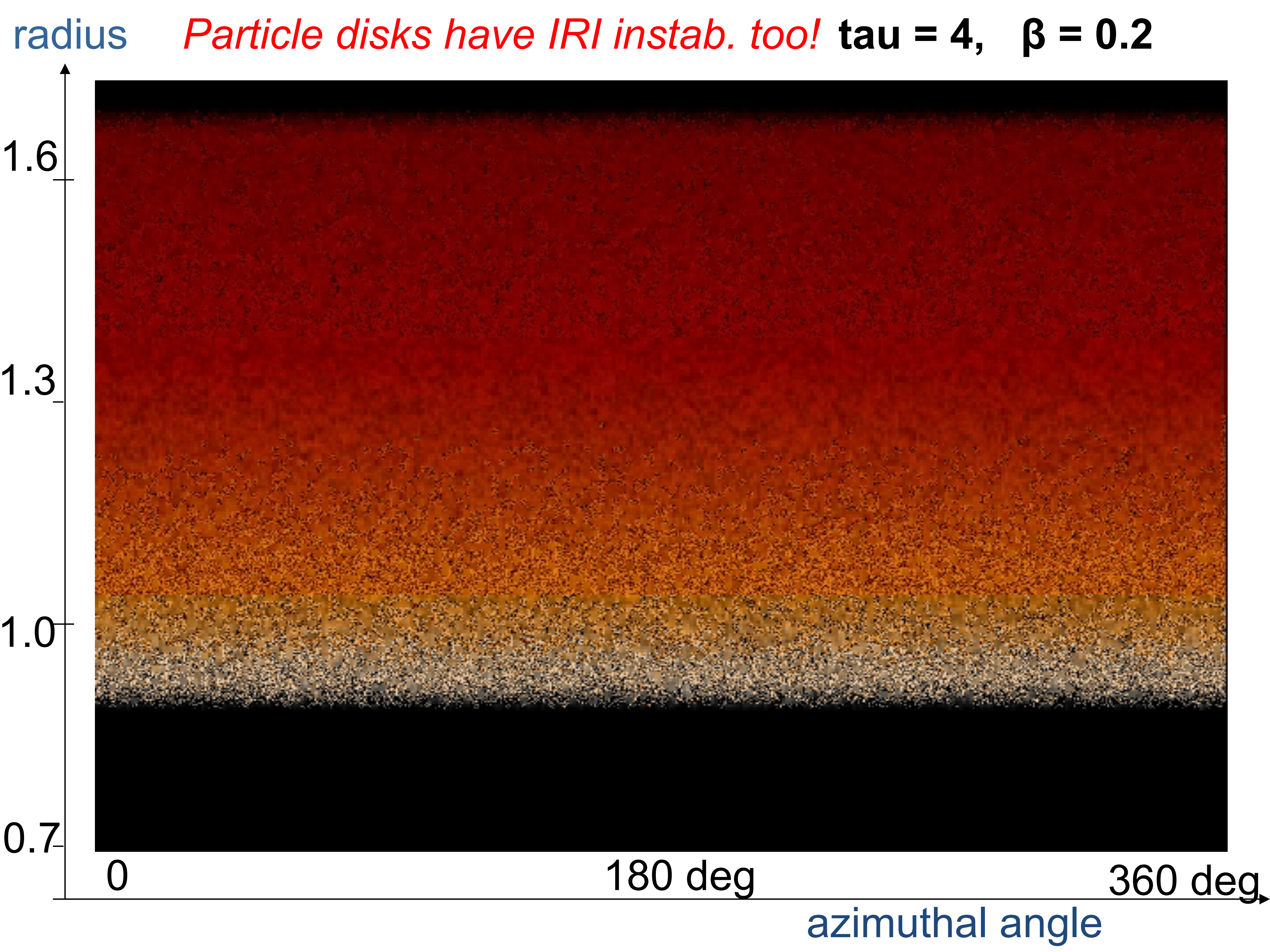
See Fung & Artymowicz (2014) for details

GAS DISK HYDRODYNAMICAL SIMULATION (PPM method, 2-D)

R.h.s. shows a background-removed picture of density of growing modes.
Analytical predictions are in agreement with calculations.
Models of disks were running faster on 3 GPUs than on UCB 128-cpu cluster.



Opaque disks are unstable under illumination by the central object


radius
Particle disks have IRI instab. too!
tau = 4,   β = 0.2
1.6
1.3
1.0
0.7
0
180 deg
360 deg
azimuthal angle