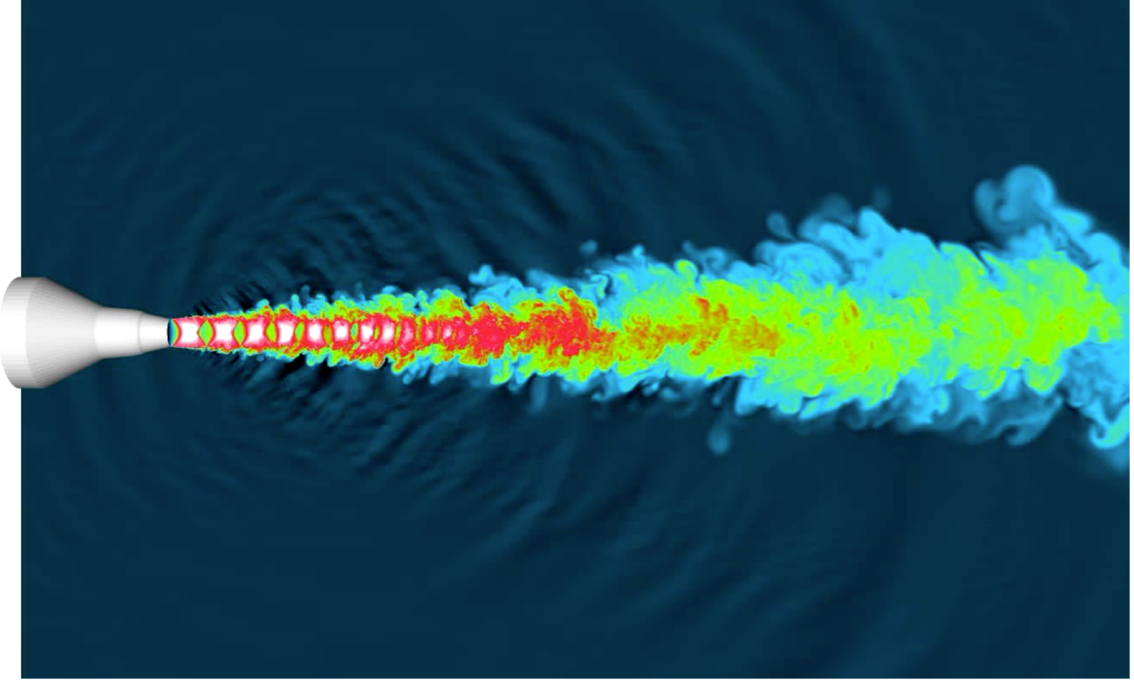
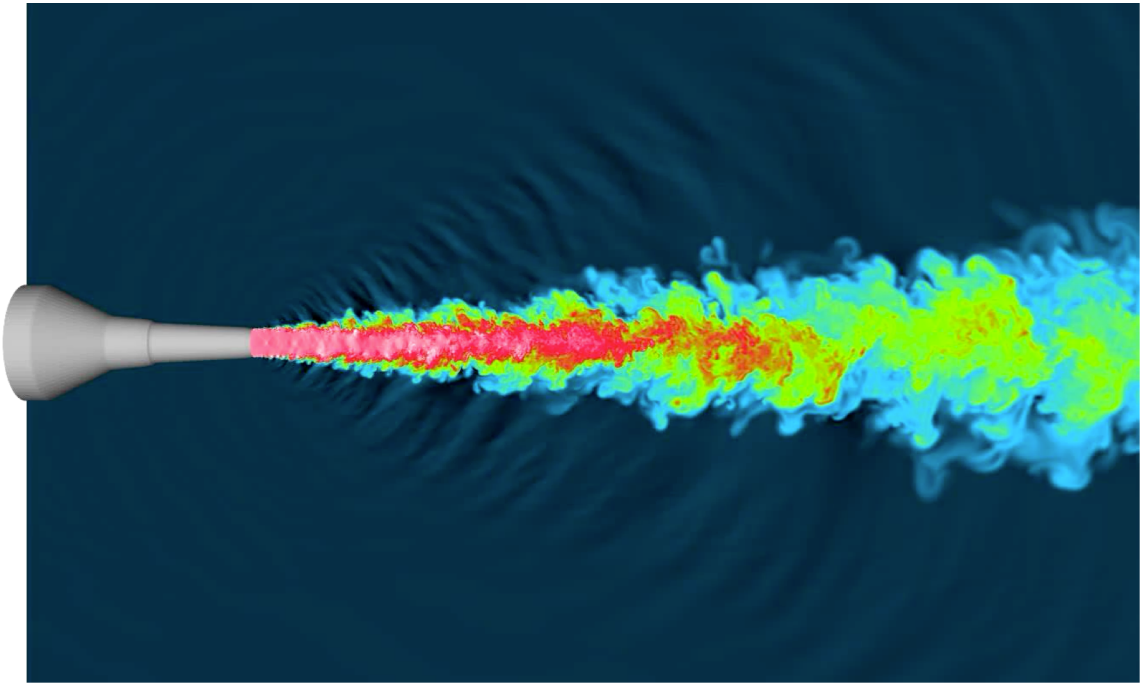
PHYD38 – Lecture 23

Turbulent jets

1. Examples and universal facts about jets
2. Similarities with instabilities in simplified dynamical systems
3. Physics of jets: entrainment of ambient fluid
4. Proof of the universal opening angle



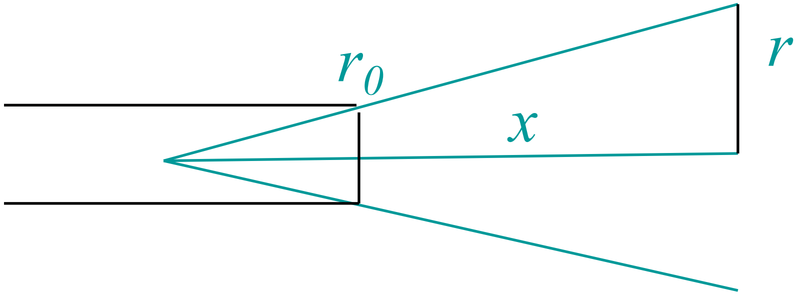


Cf. also part 4/5 of the presentations on turbulent jets

https://www.youtube.com/watch?v=1syjH7p2jyw

Basic facts about a turbulent jet:

- The u(r) cross-jet average profile of x-velocity component becomes self-similar a few nozzle diameters past the nozzle. Opening angle of the jet is constant.
- Momentum flux (momentum flowing through consecutive cross-sections, that is through different x=const. sections) is conserved, because the ambient fluid that is entrained initially lacks momentum.
- The shape is cylindrical cone with a tip being a virtual point inside the pipe/nozzle (the flow is still parallel there, but behaves *outside nozzle* as if it was coming from the tip of a cone.



$r = x/B$

$B \sim 6 = const.$

If B=6 then the full opening angle of the cone is always 20-22°

- Distance x is counted from that virtual source point.

Proof of B=const:

- A(x) = cross sectional area of the jet = πr^2
- ρ = const. density (assumed incompressibility of the fluid)
- u(x) = top speed at distance x, at axis ($r = 0$)
- Flux of momentum through section x is constant:

  $\rho A(x) u(x)^2 = \text{const.}$ Or: $r(x)\, u(x) = \text{const.} = r_0 u_0$
- Turbulent diffusion is widening the jet. It is a random walk process, and results in the radius r growing in time according to diff. equation: $r^2 = \nu t$.
- „nu” is the coefficient of diffusion and its units are m^2/s.

We will now assume that diffusion coefficient is constant, and show that, as a consequence, the jet has the empirically observed scaling B := x/r = const.

Proof:

For a constant diffusion coefficient ν, on average the jet half-width obeys $r^2 = (r_0 u_0 /u)^{-2} = \nu t$.

Proof (cont'd):

$$r = r_0 u_0 / u = (\nu\, t)^{1/2}$$

$$u = dx/dt = r_0 u_0 / (\nu\, t)^{1/2}$$

which integrates by separation of variables to

$$x(t) = (2\, r_0 u_0 / \nu^{1/2})\ t^{1/2}$$

Since both r(t) and x(t) grow as $\sim t^{1/2}$, their ratio is constant:

$$B = x/r = 2\, r_0 u_0 / \nu\ . \qquad (1)$$

But in theory of diffusion, $\nu = V\,L/3$, where L is the mixing length, a concept introduced by Ludwig Prandtl, denoting the average distance of turbulent transport. V is the aver. speed of transport. L is simply the size of a typical eddy, which Prandtl proposed to take equal to r in a jet, while V is L times the r-gradient of u, or using the estimate $\delta u/\delta r = u/r$, leading to an estimate V = u.

Thus the constant diffusion coeff. equals

$$\nu = u\, r/3 = u_0 r_0/3 \qquad \text{and} \quad u_0 r_0 = 3\,\nu.$$

Substituting into (1) we obtain B = 6.

The full opening angle of the cone is constant and equals

$$\phi = 2 \tan^{-1}(1/B) = 2 * 9.46^{o} \sim 19^{o}$$

The full opening angle ϕ of the cone is constant: $B = 6$. So:

$$\phi = 2 \tan^{-1} (1/B) = 2 * 9.46^{o} \sim 19^{o}$$

Constant diffision coefficient hypothesis works well! It explains the universal conical shape of the jet, with full opening angle of 19 degrees.

- Mass flux (dm/dt) is NOT constant, since clearly the ambient fluid is added, but at what rate? We have

  $u = u_0\, r_0 / r$ and hence

  $dm/dt\ (x) = \rho\, A(x)\, u(x) = \pi\, \rho\, u_0\, r_0\, r \sim r \sim x$

Mass transported in a jet grows linearly with distance from the virtual jet origin: at the distance from orifice = $B\, r_0$ = 3 diameters of the nozzle, it is already 2 x mass outflow rate from the nozzle: 50% of jet fluid and 50% of ambient, mixed-in fluid. At twice that distance, mass flowing in the jet is 3 x mass outflow, in the ratio 2:1 ambient:injected fluid. And so on. As a consequence, dilution and cooling of a warmer jet to the ambient temperature happens quickly.

PHYD38 – Lecture 24

Nonlinear astrophysical gas & particle dynamics. Supercomputing

Pawel Artymowicz

+ former UTSC undergrad & UofT graduate students:

- prof. Jeffrey Fung (Clemson U., in 2020)
- Fergus Horrobin (Tesla, car dynamics simulation div. leader, in 2022)

1. Exoplanets. Origin, migration.
2. Dust disk instabilities
3. Supercomputing
4. Other topics

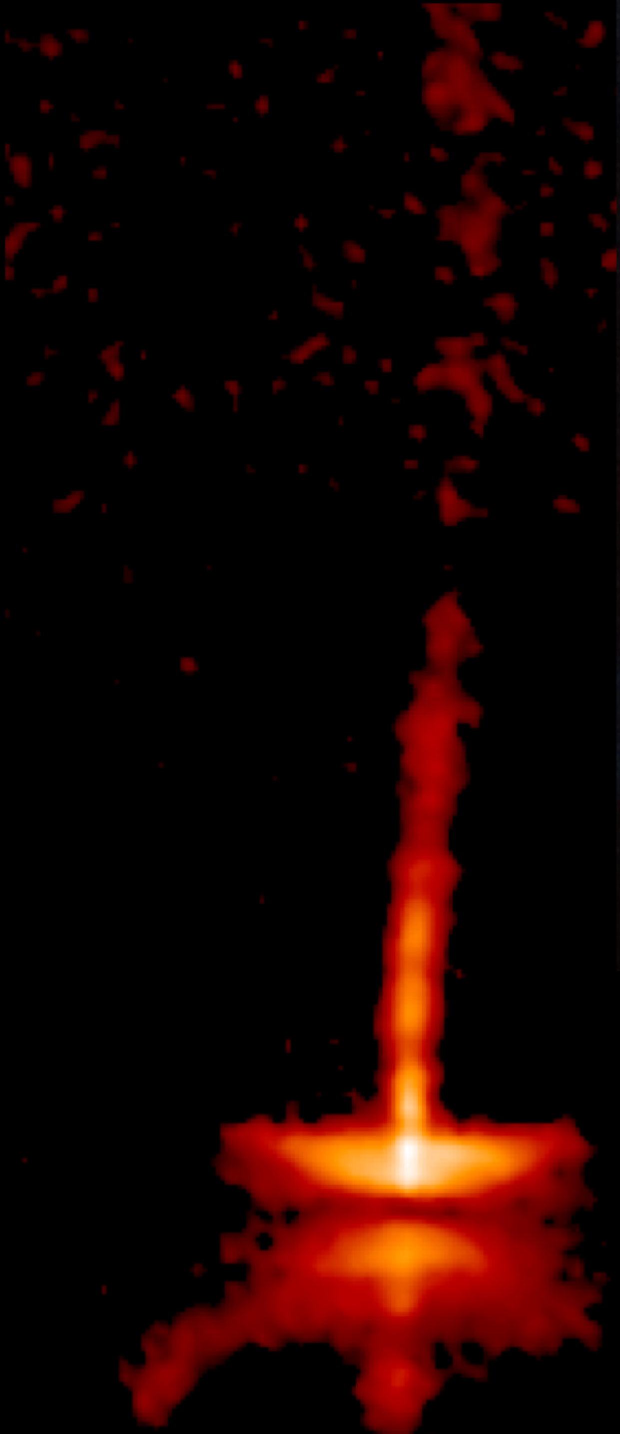
The worlds come into being as follows: many bodies of all sorts and shapes move from the infinite into a great void

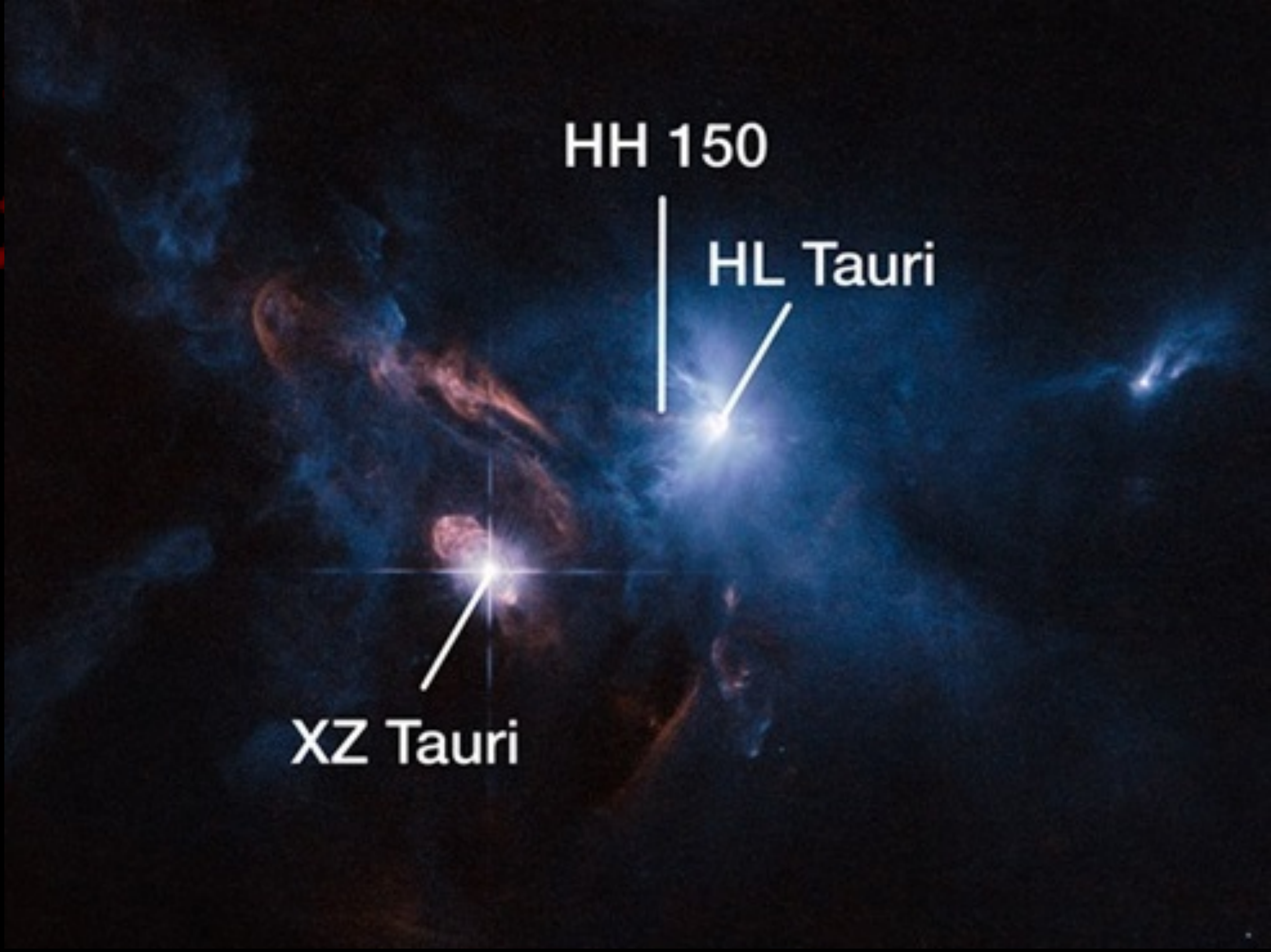
they come together there and produce a single whirl, in which, colliding with one another and revolving in all manner of ways

they begin to separate like to like.

Leucippos (480 - 420 BC),

cited by Diogenes Laertios (180 - 240 AD)





Disks in star-forming regions
produce stars & planets
(as by-product)

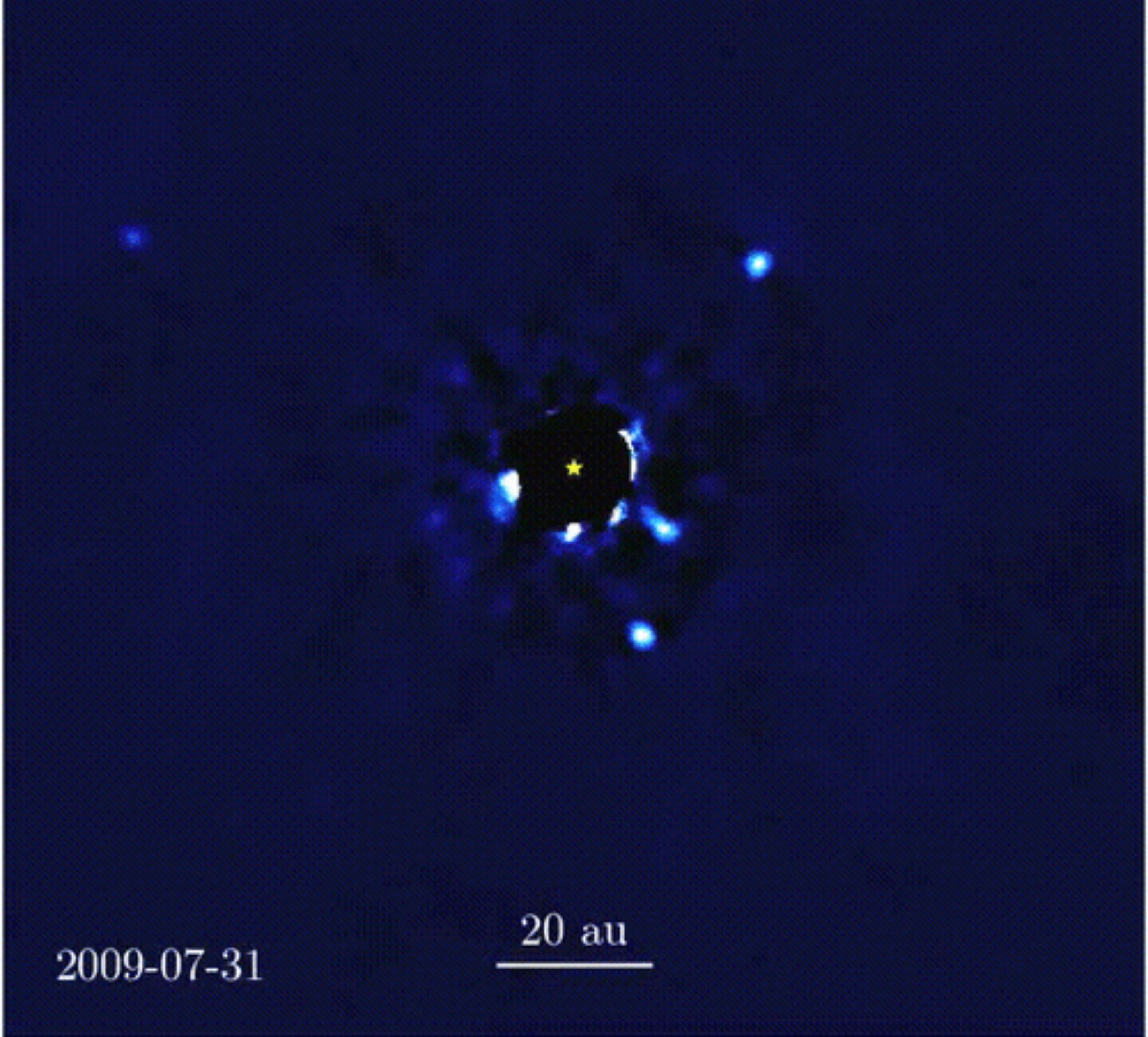
In some worlds there is no Sun and Moon, in others they are larger than in our world, and in others more numerous.
In some parts there are more worlds, in others fewer (...); in some parts they are arising,
in others failing.

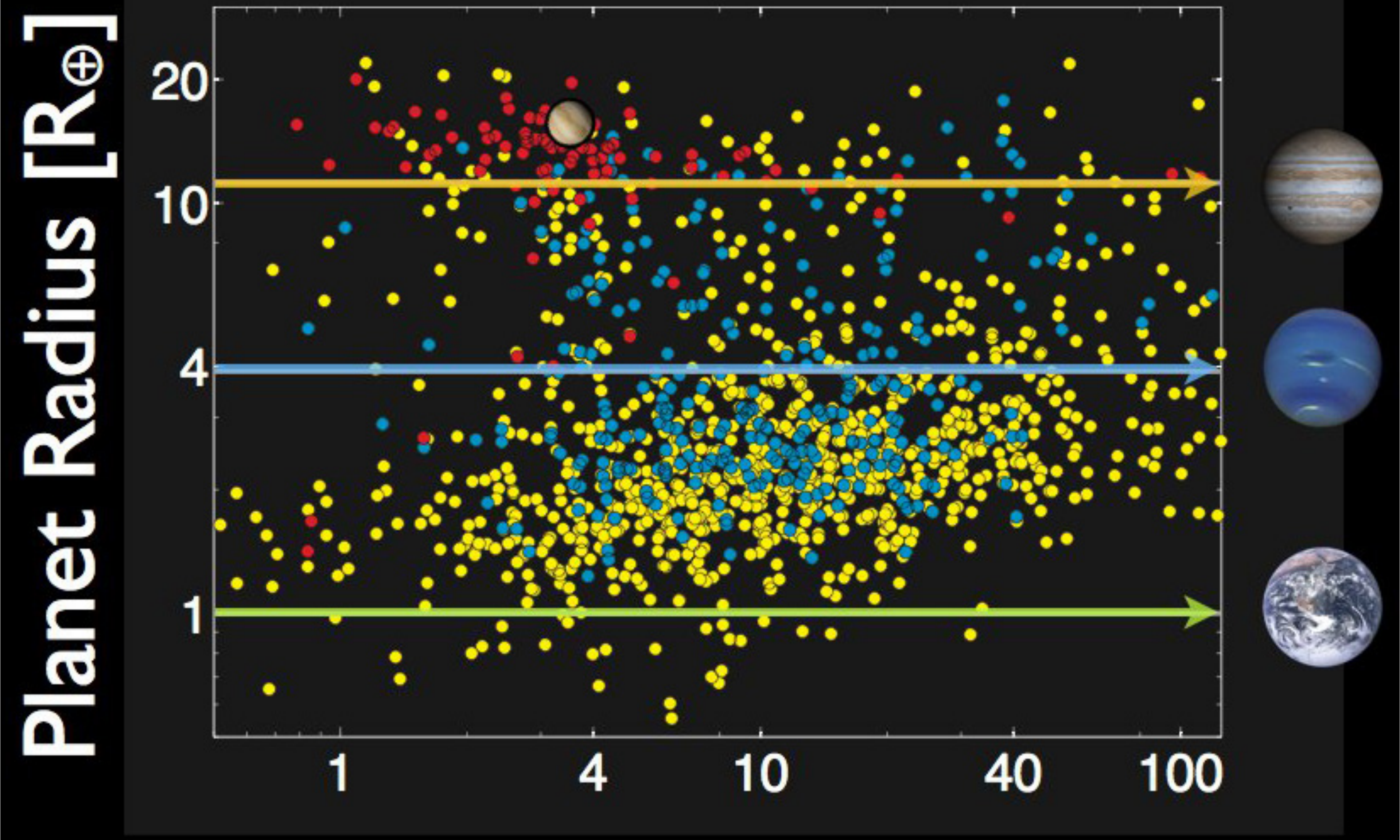
There are some worlds devoid of living creatures or plants or any moisture.

Democritus (ca. 460-370 B.C.)

In the last 30 years we've found a thousand proofs of this prescient thinking

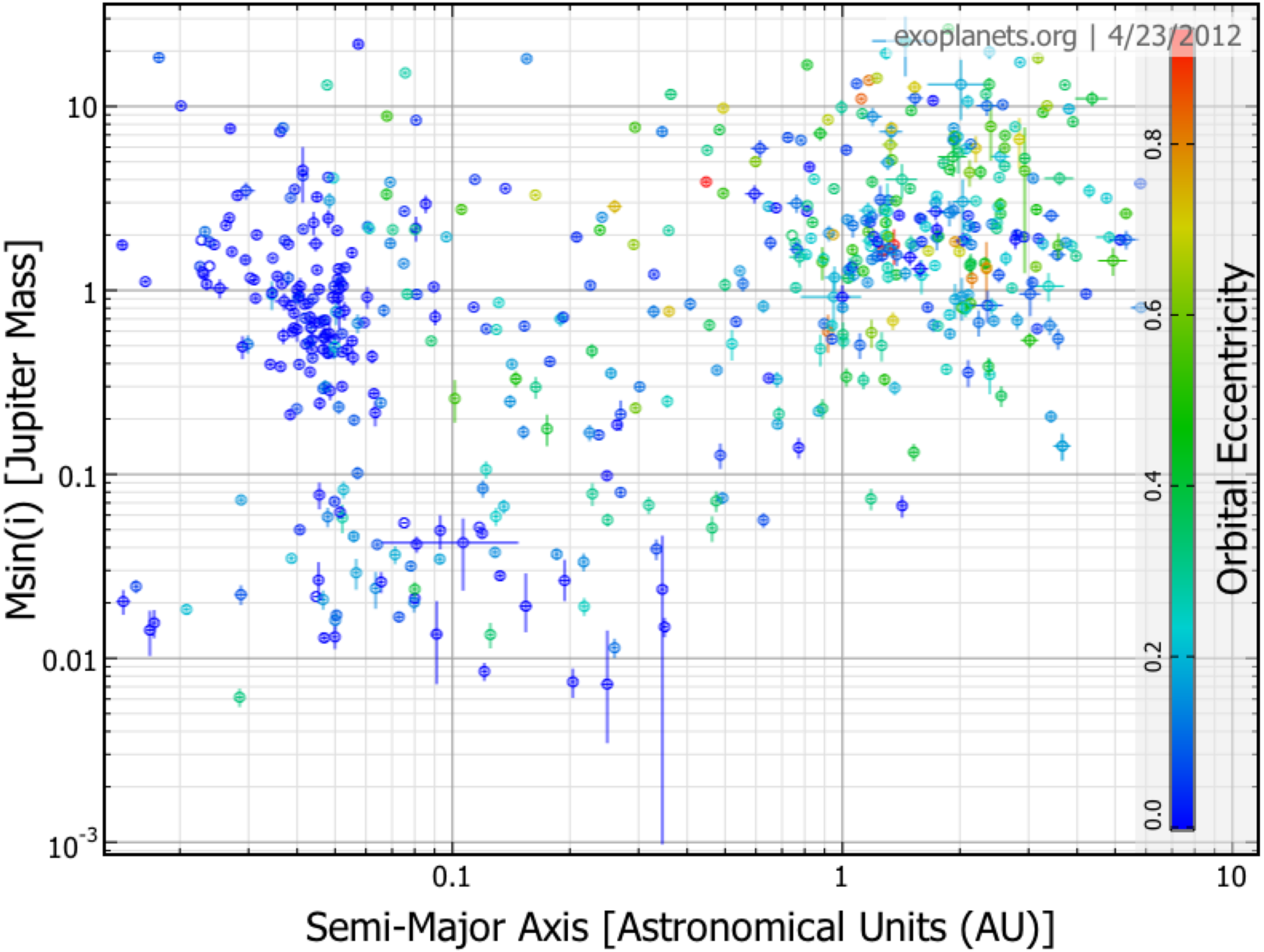
Imaging of exoplanets of HR 8799





Orbital Period [days]

We need to explain the ‘hot jupiters’, super-Earths etc.

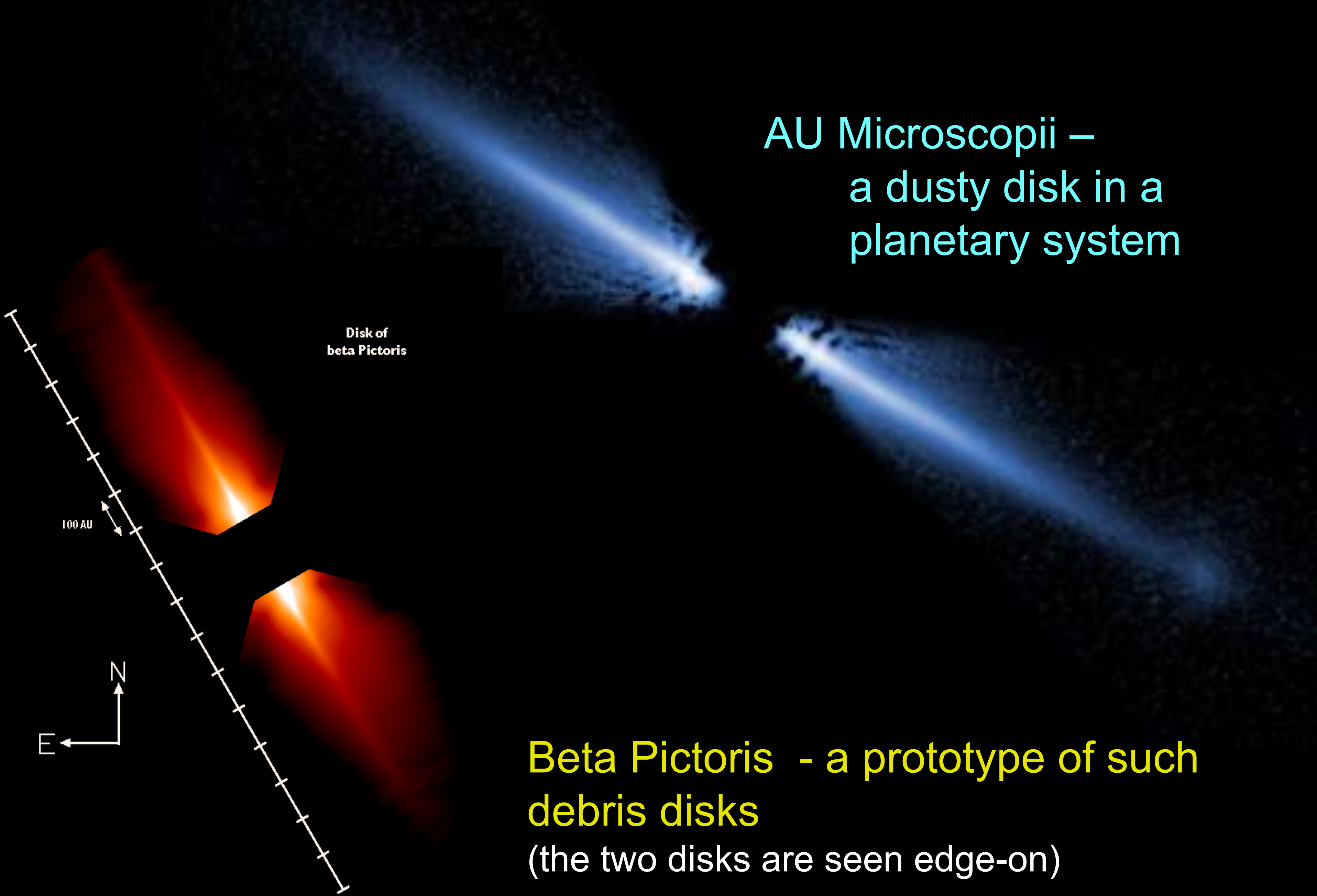


ALMA = Atacama
Large Millimeter Array

HL Tauri disk
0.5 Myr age

T Tau disks are primordial
– they have lots of H and He, which formed the star

We also need to understand the dusty disks around 1/3 of normal stars

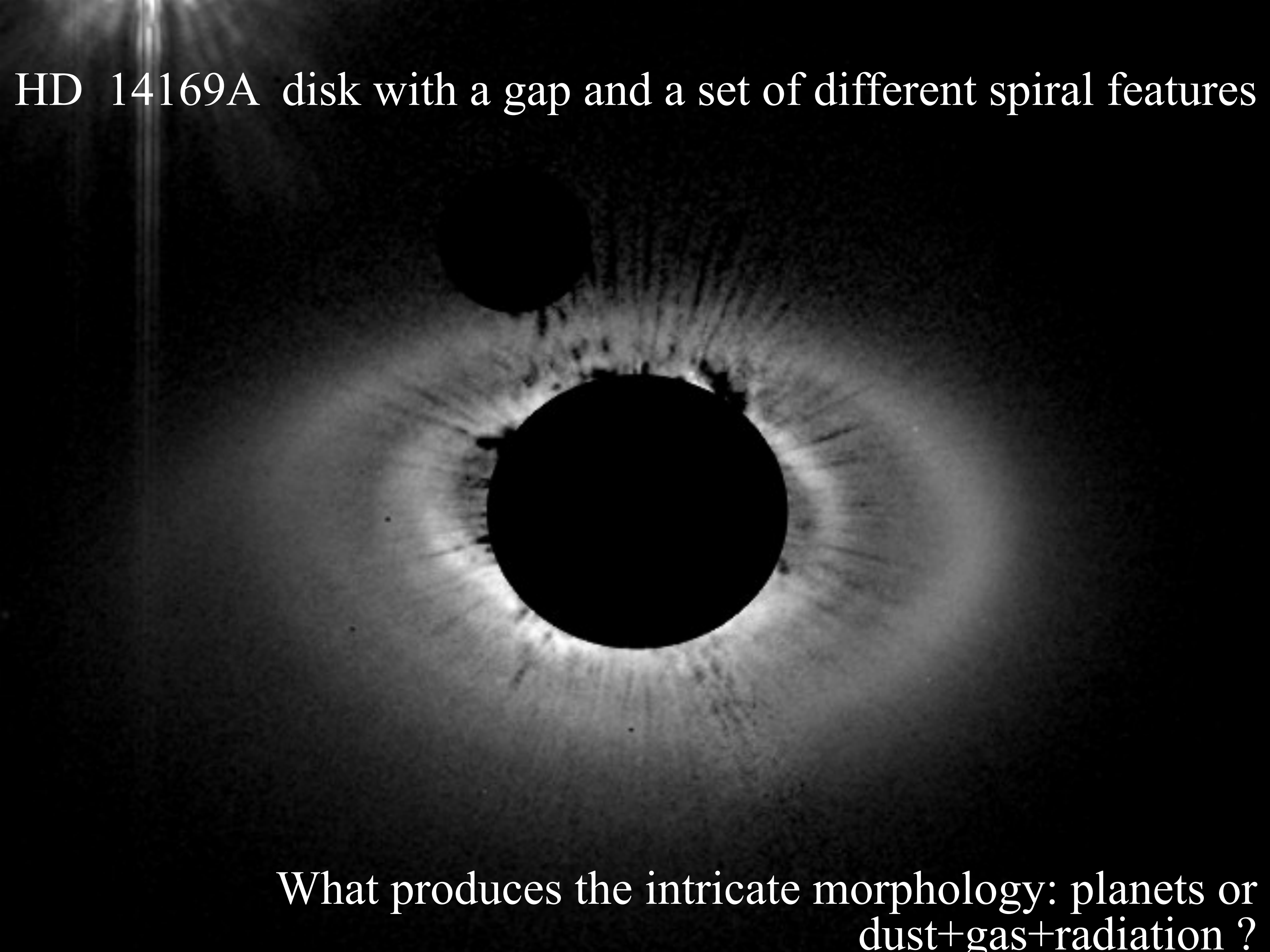


AU Microscopii –
a dusty disk in a
planetary system

Beta Pictoris - a prototype of such debris disks

(the two disks are seen edge-on)

HD 14169A disk with a gap and a set of different spiral features



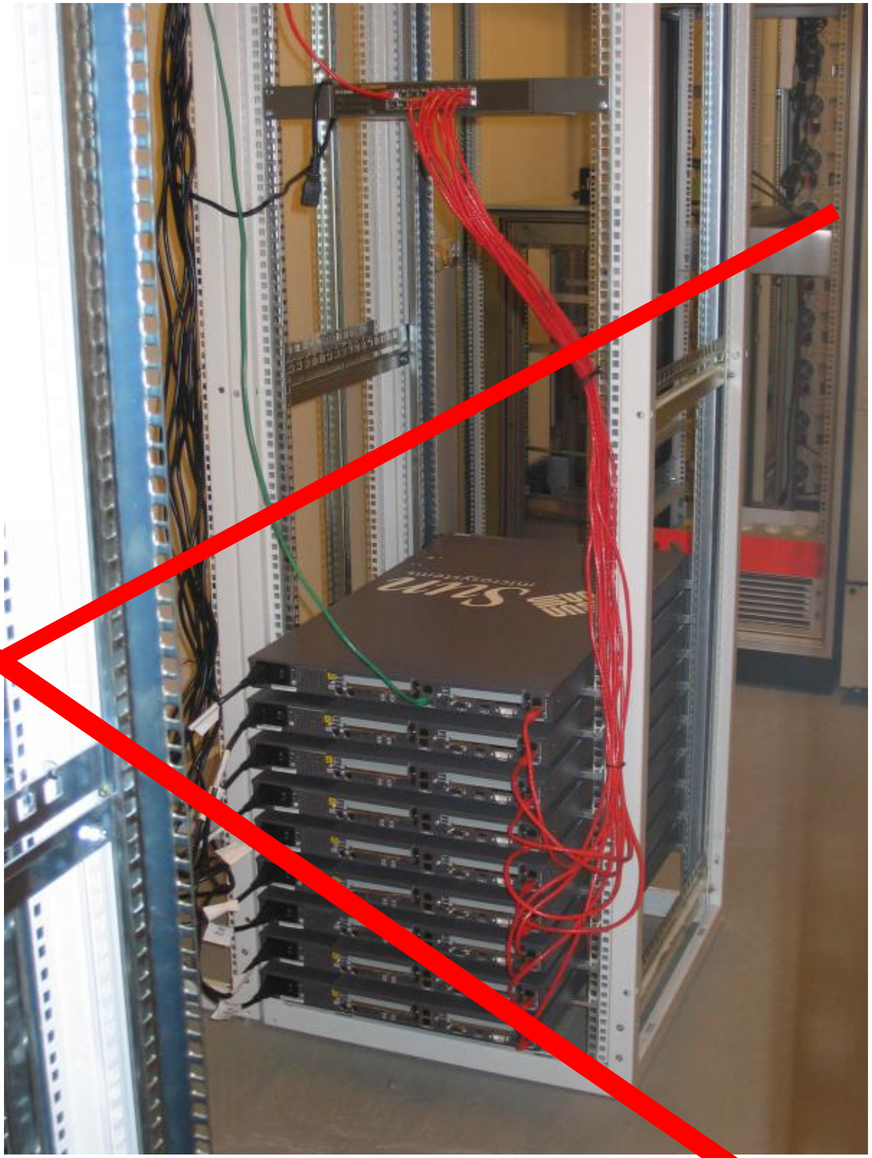
What produces the intricate morphology: planets or dust+gas+radiation ?

SUPERCOMPUTER = MANY WELL CONNECTED WORKSTATIONS

1990s and 2000s was the era of clusters

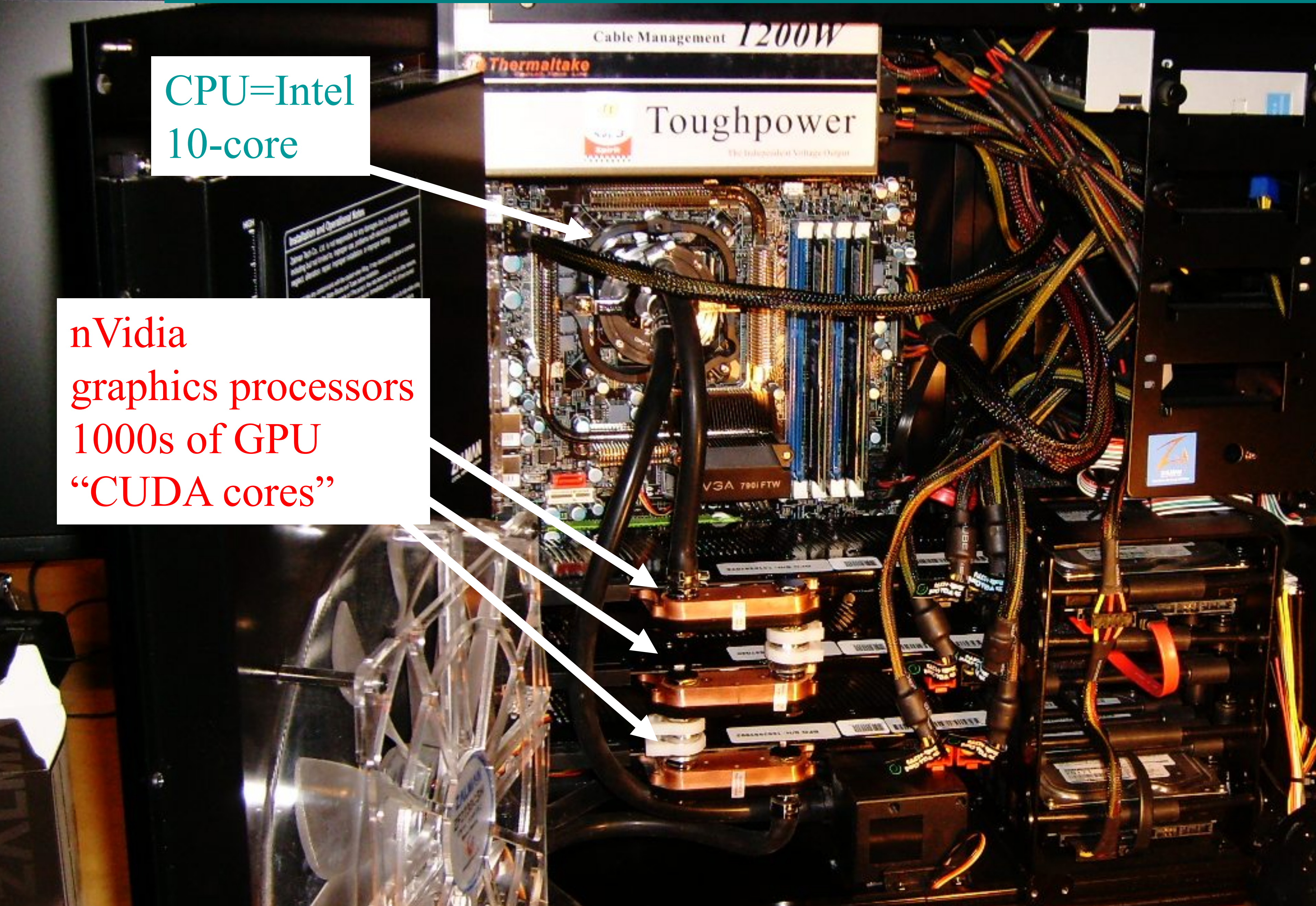
MPI (message passing interface) for parallelization

Then something happened & for some time years I thought that clusters will go away…



2007-2010 = beginning of an era of GPU or "graphics computing"

GPU-based PERSONAL SUPERCOMPUTE



CPU=Intel 10-core

nVidia
graphics processors
1000s of GPU
“CUDA cores”


ASUS
3-WAY SLI
GEFORCE GTX
GEFORCE GTX
GEFORCE GTX
asetek

Calculations on

(Nvidia) GPUs

…
GTX 970
GTX 1080ti
…

RTX 4090

Supercomputing ~2016

CPU **GPU** **MIC**

(Intel Xeon Phi)

3 different types of compute units:

CPU = Central Processor Unit

GPU = Graphics Processor Unit

MIC = Many Integrated Cores = like CPU but many more, simple cores

Supercomputing in 2024

CPU **GPU** **ASIC**

2 different types of compute units, since Xeon Phi line merged with Intel CPUs.

ASIC = application-specific integrated circuit (custom-designed)

CPU or GPU? In the last 10 years we thought the answer is simple: GPU

The graphics cards such as Nvidia GTX Titan can provide N-body or incompressible fluids simulation in 2D right out of the box.
1.5-3.2 TFLOPs per card, usually 3 of them can be persuaded to work 24/7 in one linux box, and exchange data w/o the help of CPU.

Def.: 1 TFLOP = 10^{12} floating point operations per sec = 1000000000000 op/s

Jeffrey Fung developed at UofT a GPU hydrocode PenGUIn. Method: PPM (Piecewise Parabolic Method of Woodward and Collela 1986)
He ran 2 & 3-d PPM simulations of planets embedded in disks and disk instab.

Benchmark 300^3 grid calculations makes 1 step in ~0.3s on 3-Titan machine, ~1s on one GPU. It is largely bandwidth-limited (~150 GB/s GPU-RAM).

(Jeffrey Fung says this is ~2x the performance of Berkeley dept. *cluster* of 128 nodes his collaborators were using)


ASUS
3-WAY SLI
GEFORCE GTX
GEFORCE GTX
GEFORCE GTX
asetek

CPU?

On the other hand, scientists are usually guilty of not squeezing the full power from their CPUs.

Many of us rely on compiler optimization switches & use MPI to connect the nodes of a cluster (to compute in parallel).
But, as a rule, we:
- don’t achieve a linear speedup on multi-core CPUs because we
- don’t do fully efficient openMP (multithreading) ←
- don’t spend time to optimize the code on the level of one thread
- don’t vectorize. We never bother to learn where and how to use freely and straightforwardly available AVX (advanced vector extensions on Intel proc’s). AVX descends from similar tools called MMX and SSE, SSE-2. Vectorization is sometimes called SIMD (single instruct., multiple data) processing.

CPU or GPU?

Speed comparisons:

A well optimized PPM hydrocode on a 300^3 grid,
w/o any hand-coded AVX instructions
on a machine with one of the best (in 2014) Intel CPU processors E5-2690v2
runs at 1.8s per step using gfortran GNU compiler, and only

0.76s/timestep using ifort Intel compiler with SIMD directives and OpenMP.
(This better vectorization by Intel is typical also of C/C++ compilers.)

On of the best GPUs (Titan) runs the problem at ~1s /step, i.e. at roughly equal speed, and certainly not 50x faster!

Thus, surprisingly, CPU ~ GPU.

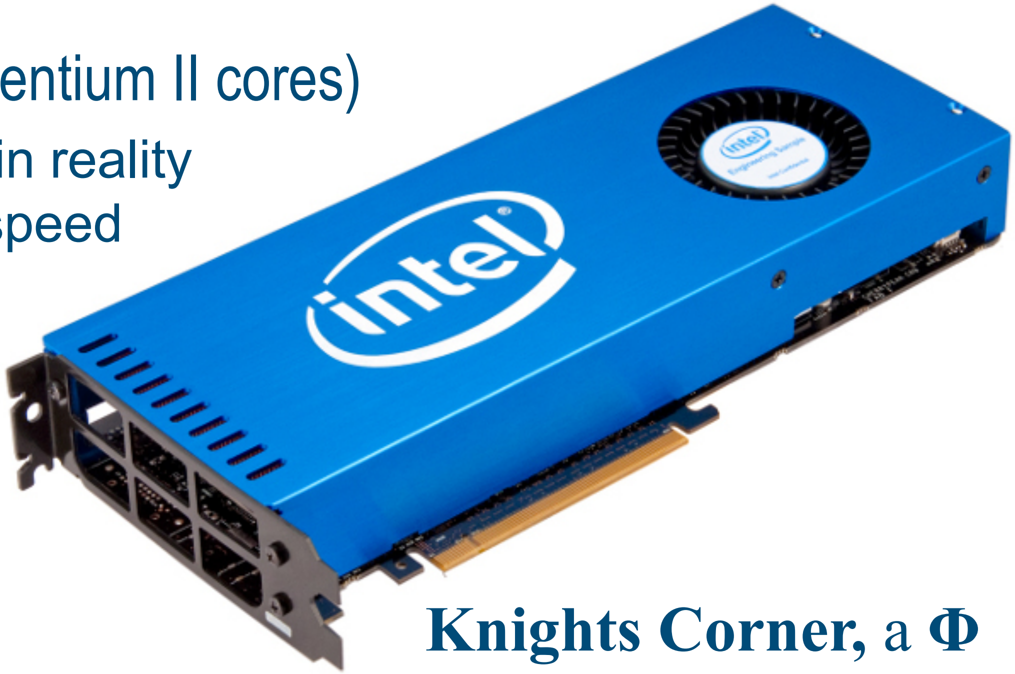
Yet another platform: Xeon Phi coprocessor, Many-Integrated-Core (MIC)

IXPs architectures: Knights Corner or KNC (*soon Knights Landing, 3x faster*)

- ■ Not unlike GPU, ~1 TFLOP theor. max throughput in double prec., ~2 TF sp.
- ■ Power consumption similar to GPU: 200-300W
  (200W for 400^3 grid CFD on Φ; while ~250W the same code on a GPU)
- ■ Similar physical format, cooling methods
- ■ Similar amount of DDR5 memory, 6GB on GPU vs. 8GB on Φ; similar bandwidth

☐ 57-60 Intel cores (more modern Pentium II cores)

☐ 1500 to 2600 cores" on GPU but in reality only 8-16 multiprocessors of clock speed ~1 GHz (= Phi)
(CUDA cores really do not exist)



☐ programing is very different:
CUDA on GPU is more complex;
there is no free CUDA Fortran, only CUDA-C/C++

Knights Corner, a **Φ**
a.k.a. IXP or MIC

What about Xeon Phi, Many-Integrated-Core (MIC) computing?

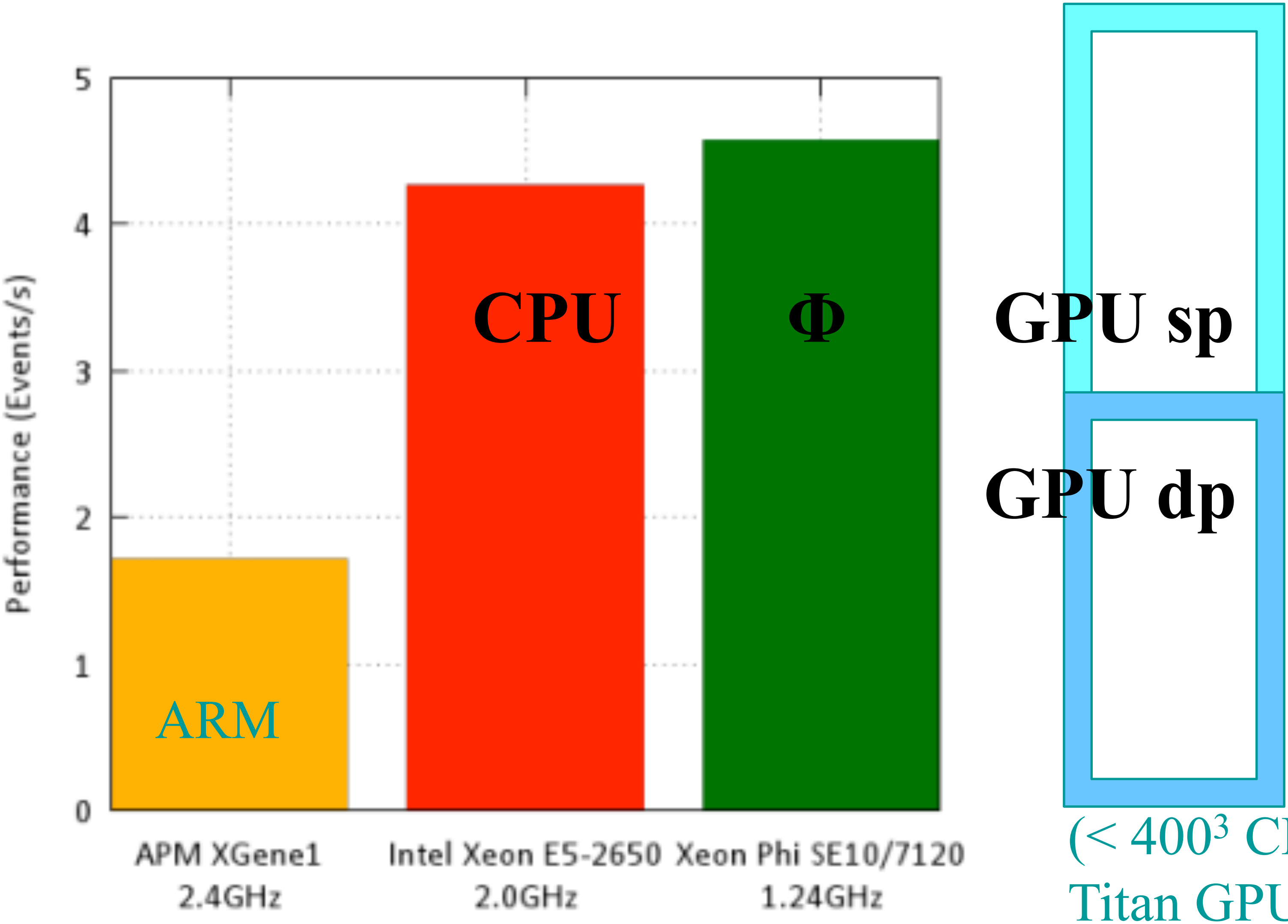
Xeon Phi processors (57-61 Intel CPU cores per card), are like mini clusters with their own IP address and own functional Linux system(!), but have no harddisks (only a virtual file system, volatile).

1 GHz clock is ~3x slower than a typical CPU clock, the amount of cache per core is much lower. Needs 4 threads per core to run efficiently →244 threads in practice, on KNC.

I have compared the platforms on CFD, on grids of order 400^3.
Results: hydrocodes on Phi turned out to compete favorably with CPUs and GPUs.
The ordering of compute power in **single prec. is: CPU ~ Phi < GPU**,
but in **double prec.   GPU < CPU ~ Phi**

This forced a substantial redesign of the UTSC supercomputer(s) toward Φ's:
Main, 16 x (CPU + GPU + Φ).   A 2nd cluster was added, based on 32 Φ's only.

In 2014, CERN Researchers considered which of the platforms makes the most sense for distributed Worldwide LHC Computing Grid, processing data for LHC experiments in 170 computing centers in 40 countries (incl. UofT)



($< 400^3$ CFD on Titan GPU)

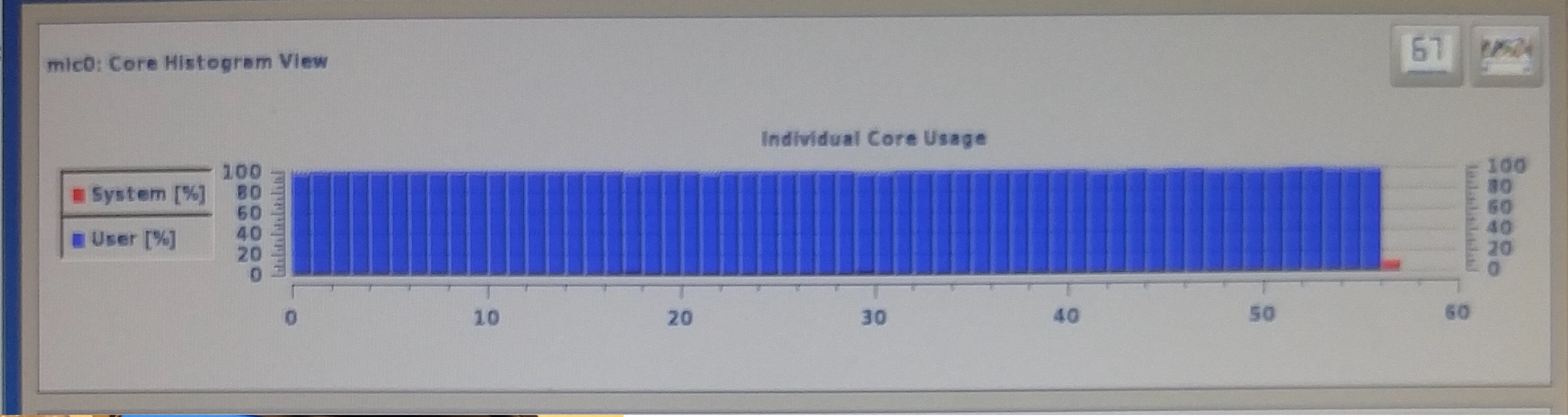
Comparisons:

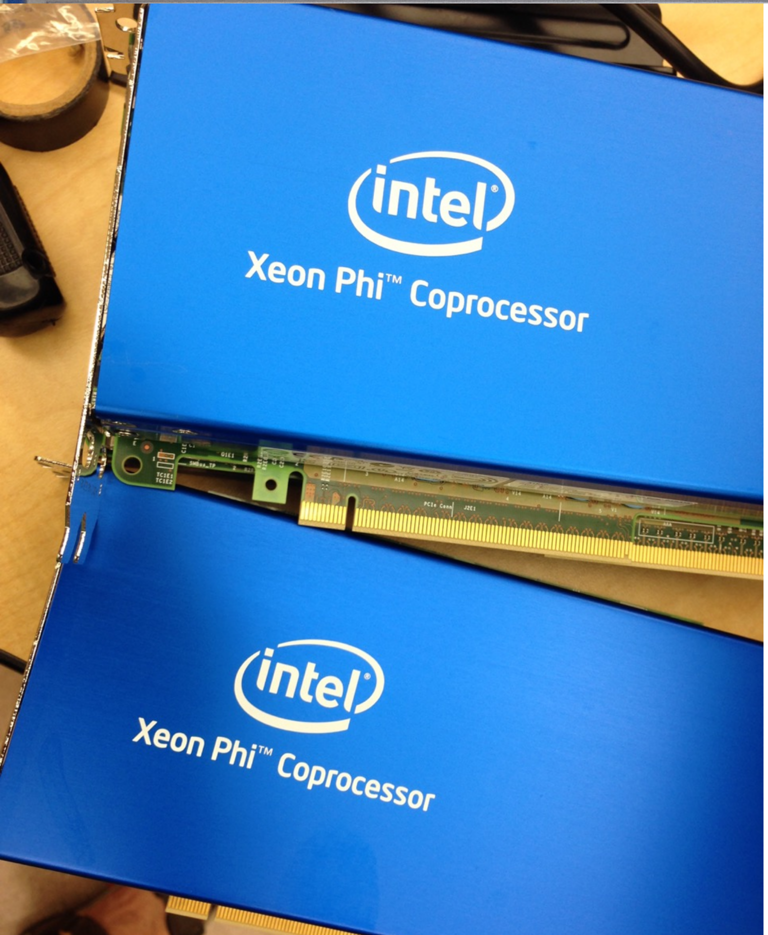
Somewhat surprising to a GPU-evangelist like me, the reason that GPUs seemed 10^2 times faster than CPU was that we typically *don't know how to program CPUs* well, not because of there advertised 1000s of CUDA-cores. In fact, # of those cores is a marketing ploy, as they do not exist physically, they are more like minimum # of threads.

Each modern GPU has 16-26 symmetric multiprocessors (SMPs), which play the role of Intel cores! But you won't find those numbers easily from GPU makers.

Intel Xeon Phi

Xeon Phi's powered the fastest supercomputer in the world Tianhe-2.
Their list price was high ($1.5-2.5k)
Around 2016 these processors became affordable because of....
geopolitics!





IXP
Intel Xeon Phi

about 57 to 60 cores,
CPU-like accelerator
~1 TFLOP dp

Adventures of Intel Xeon Phi

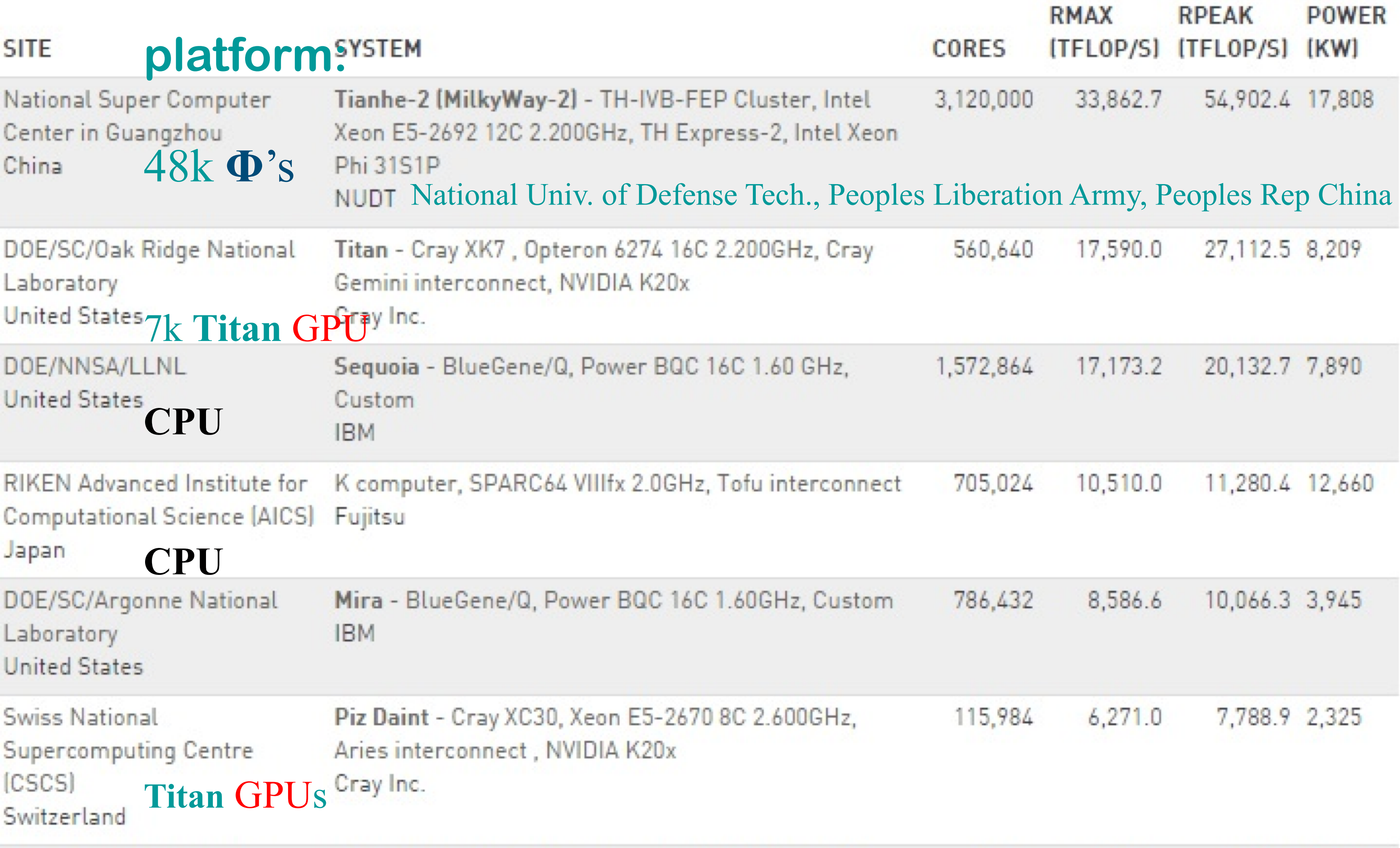
In 2007, Intel tried but failed to attack Nvidia & AMD GPUs market share with its won GPU. The Larrabee project was a many-core hybrid of CPU+GPU (at least 40 cores).

Year after year, the production was delayed and the chip redesigned. Finally, the funding of the project was cut after Intel lost an estimated $2 bln on it.

But Intel engineers had another idea in mind: use the many-core chips to conquer supercomputing markets. It worked for a while!

The Phi’s were the spin-offs of Larrabee project. The first massive batch of the chip 31S1P (special version not normally sold) was used to build the Tianhe-2 supercomputer at NUDT.

The list of fastest supercomputers in the world, **2015** edition.
The top platforms were: **Φ**, GPU & **CPU**, in that order.
Intel later merged CPU and **Φ** platforms.



SITE	platform:	SYSTEM	CORES	RMAX (TFLOP/S)	RPEAK (TFLOP/S)	POWER (KW)
National Super Computer Center in Guangzhou China	48k **Φ**'s	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT National Univ. of Defense Tech., Peoples Liberation Army, Peoples Rep China	3,120,000	33,862.7	54,902.4	17,808
DOE/SC/Oak Ridge National Laboratory United States	7k **Titan** GPU	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
DOE/NNSA/LLNL United States	**CPU**	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
RIKEN Advanced Institute for Computational Science (AICS) Japan	**CPU**	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4	12,660
DOE/SC/Argonne National Laboratory United States		**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786,432	8,586.6	10,066.3	3,945
Swiss National Supercomputing Centre (CSCS) Switzerland	**Titan** GPUs	**Piz Daint** - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Cray Inc.	115,984	6,271.0	7,788.9	2,325

UTSC adventures with Xeon Phi’s

U.S. GOV. ISSUES EXPORT BAN
In April 2015, the US administration (export committee) stopped all sales of Intel Xeon Phi coprocessors to Chinese supercomputer centers.
It is not known if the main motive was competitive fears or national security.

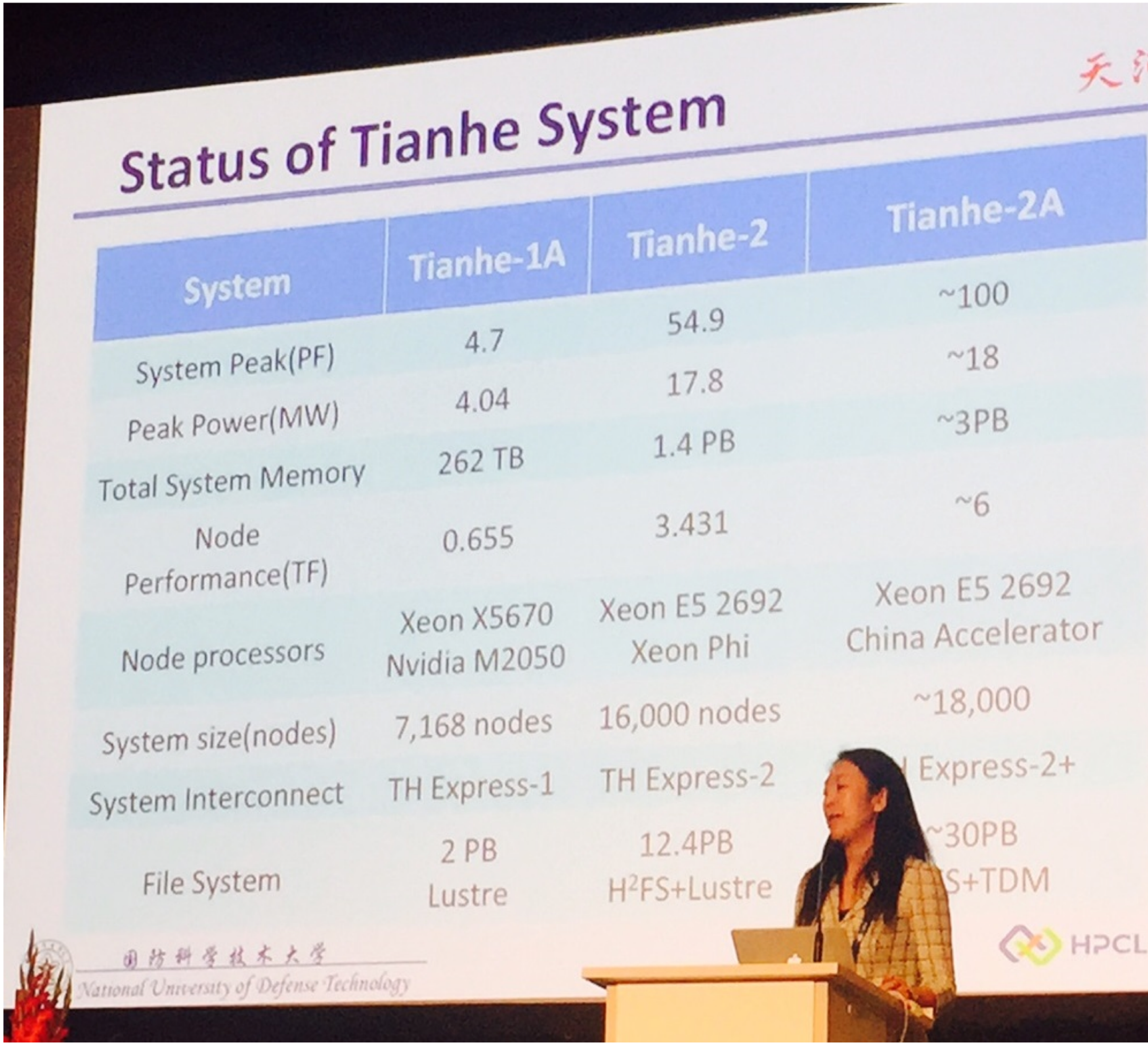
...AND IMMEDIATELY LOSES THE BATTLE
Despite this or perhaps, in the long run, because of the new restrictions,
Chinese centers will go ahead with their planned expansion,
replacing Intel chips with own coprocessors they call China Accelerators,
which are 64-bit DSP chips.

MEANWHILE, BACK IN CALIFORNIA,
in anticipation of the export ban, at the end of 2014 Intel started a quiet,
massive fire-sale of Tianhe-like Phi coprocessors.

In larger quantities, each 31S1P card was $125 USD. I’ve bought ~80 kg of Φs, or 52 pieces.

System designer for the multiple iterations of the Tianhe supercomputer, Dr. Yutong Lu, explains how China is still going to expand Tianhe to the 2A ver. with ~100 PFLOPs, despite the US export ban on Intel Xeon Phi processors.

In the long run, the US ruling will help Chinese chip designers, industry and military.



天

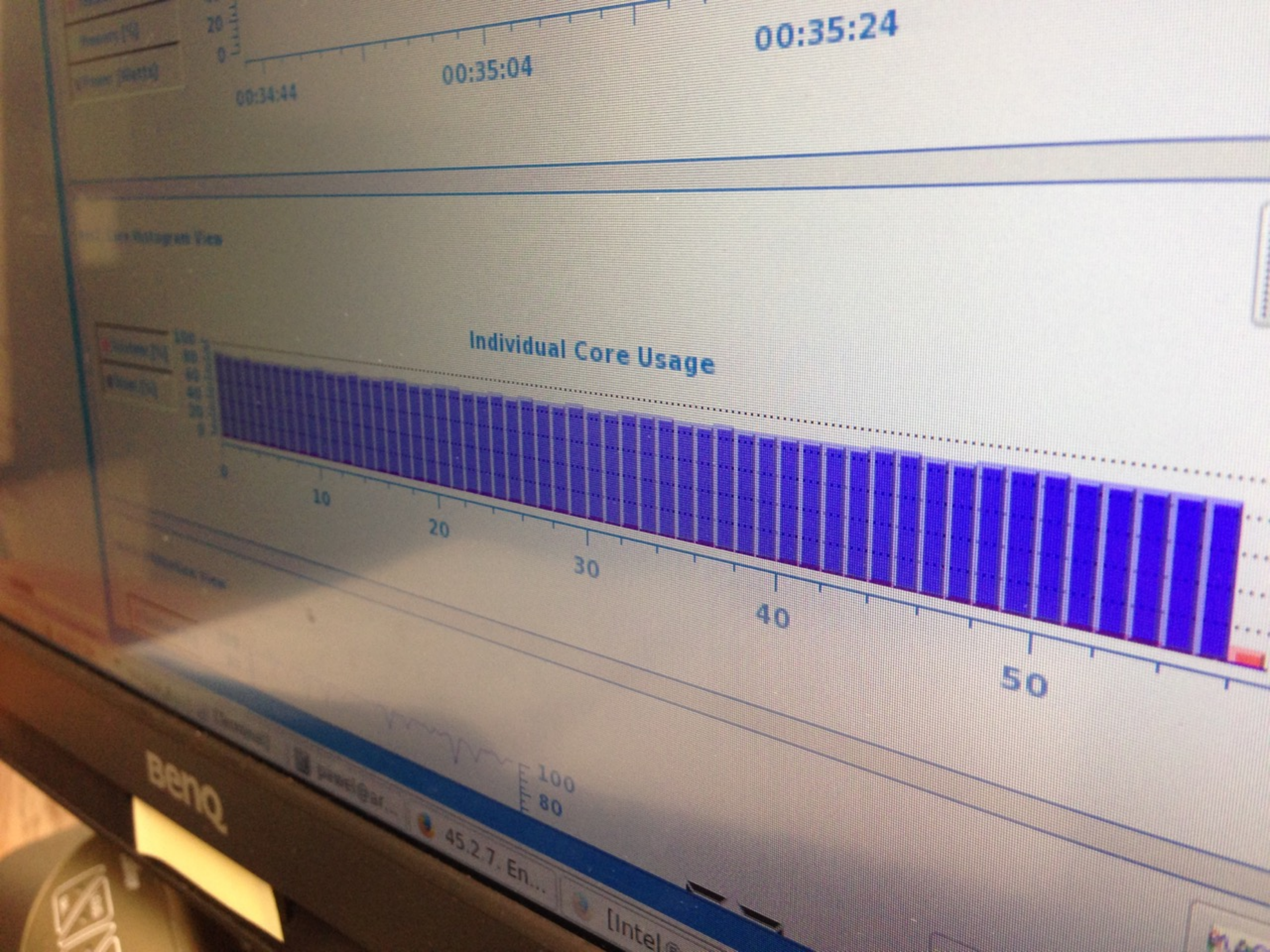
Status of Tianhe System

System	Tianhe-1A	Tianhe-2	Tianhe-2A
System Peak(PF)	4.7	54.9	~100
Peak Power(MW)	4.04	17.8	~18
Total System Memory	262 TB	1.4 PB	~3PB
Node Performance(TF)	0.655	3.431	~6
Node processors	Xeon X5670 Nvidia M2050	Xeon E5 2692 Xeon Phi	Xeon E5 2692 China Accelerator
System size(nodes)	7,168 nodes	16,000 nodes	~18,000
System Interconnect	TH Express-1	TH Express-2	Express-2+
File System	2 PB Lustre	12.4PB H²FS+Lustre	~30PB S+TDM

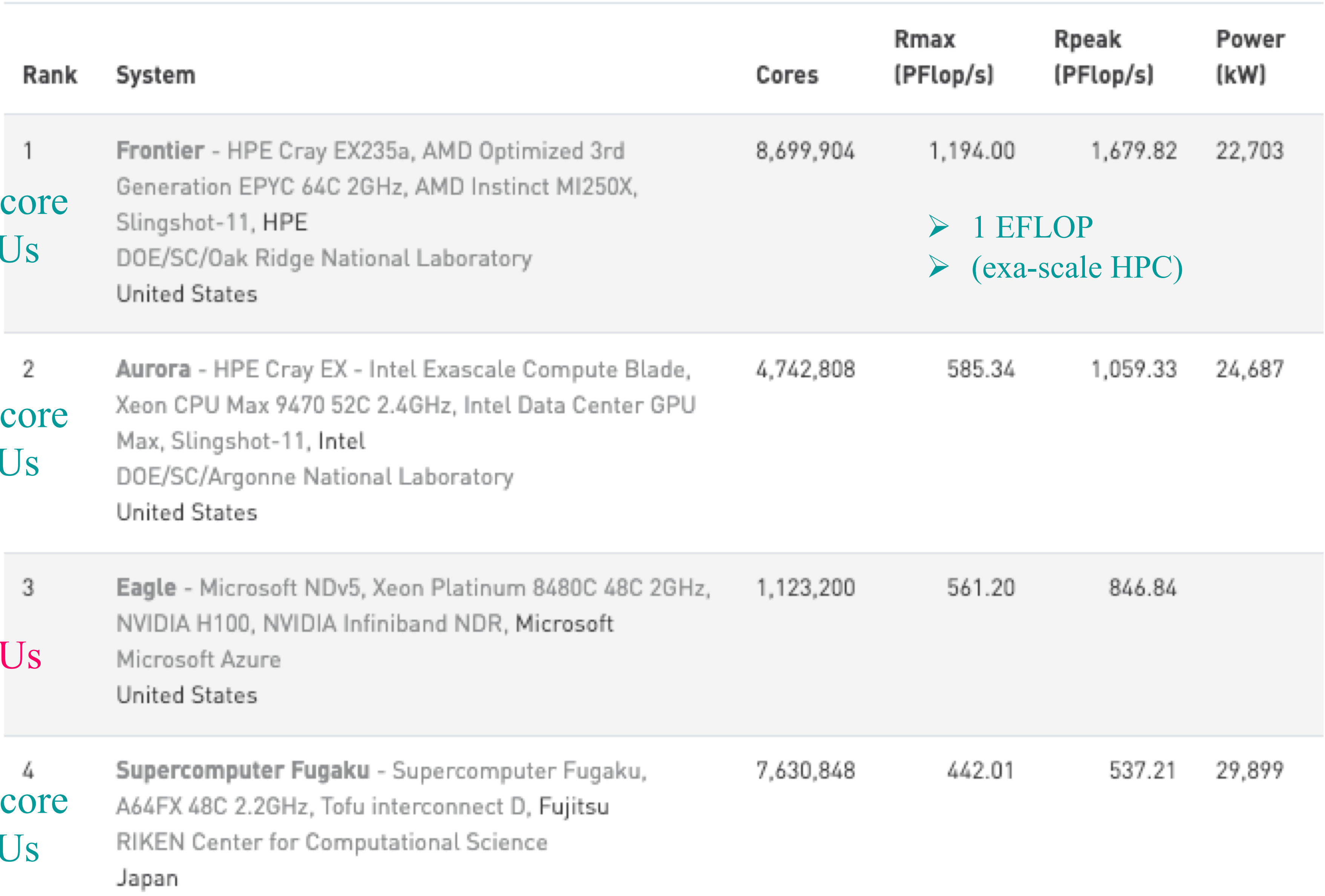
国防科学技术大学
National University of Defense Technology

HPCL


dd /dev/sda /dev/sdb bs=1M
1000³ grid = 1 GB, CFD
N-S
Cuda ← Fortran
UTSC
EVGA GTX 480
FedEx


20
0
00:34:44
00:35:04
00:35:24
Individual Core Usage
100
80
60
40
20
0
0
10
20
30
40
50
100
80
45.2.7. En...
[Intel
BenQ

The list of fastest supercomputers in the world, **Nov. 2023** edition.
The top platforms are: **CPU (=MIC) & GPU**, in that order.



Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,699,904	1,194.00	1,679.82	22,703
2	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	4,742,808	585.34	1,059.33	24,687
3	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Microsoft Azure United States	1,123,200	561.20	846.84	
4	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899

64-core CPUs

- 1 EFLOP
- (exa-scale HPC)

52-core CPUs

GPUs

48-core CPUs

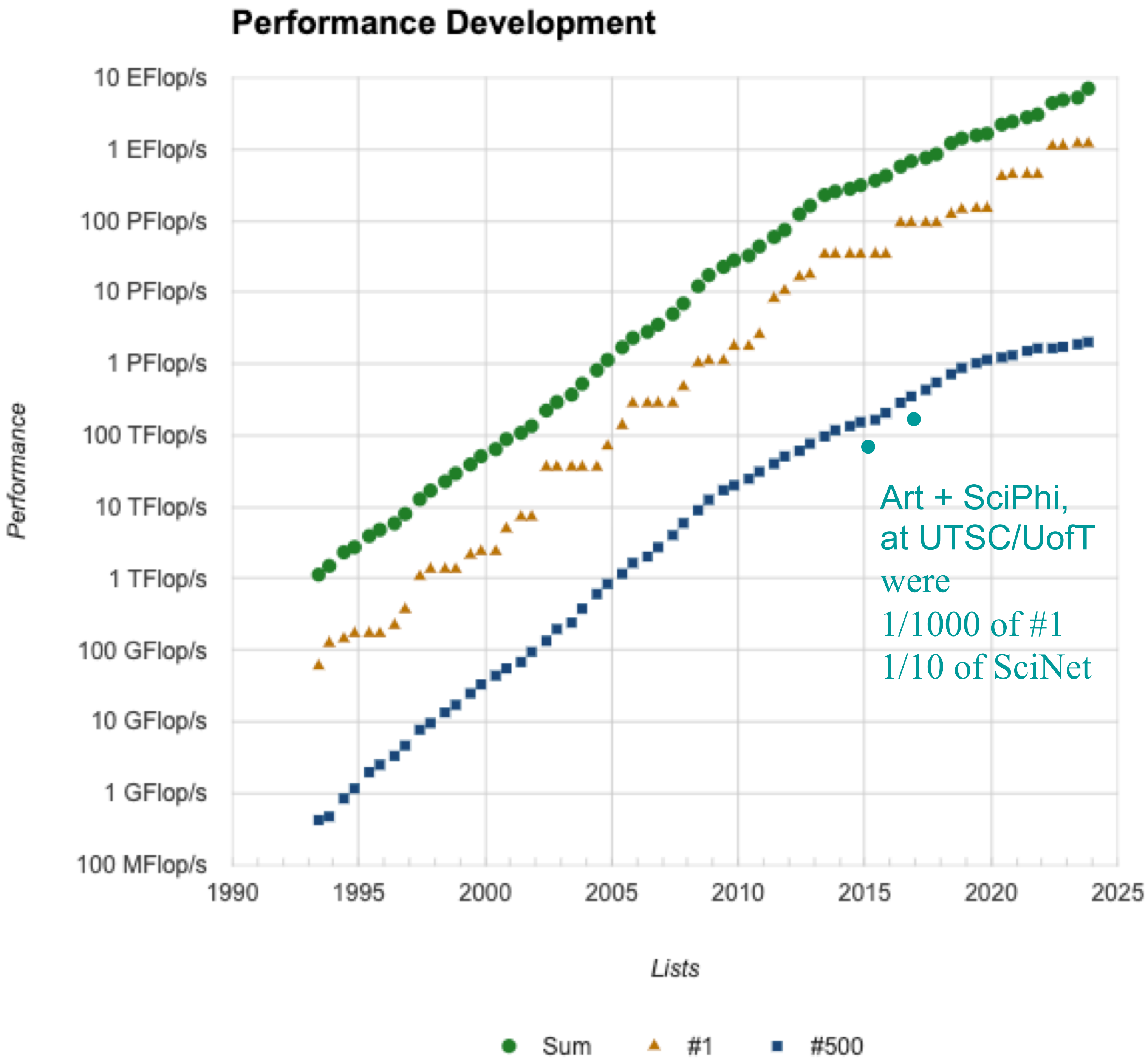
UTSC SUPERCOMPUTING

We have mastered the art of CUDA (GPU) programing, CPU (avx+openmp+mpi) and now the Xeon Phi programing. We have constructed our “theoretical telescopes”, starting from three 3-Titan linux boxes.

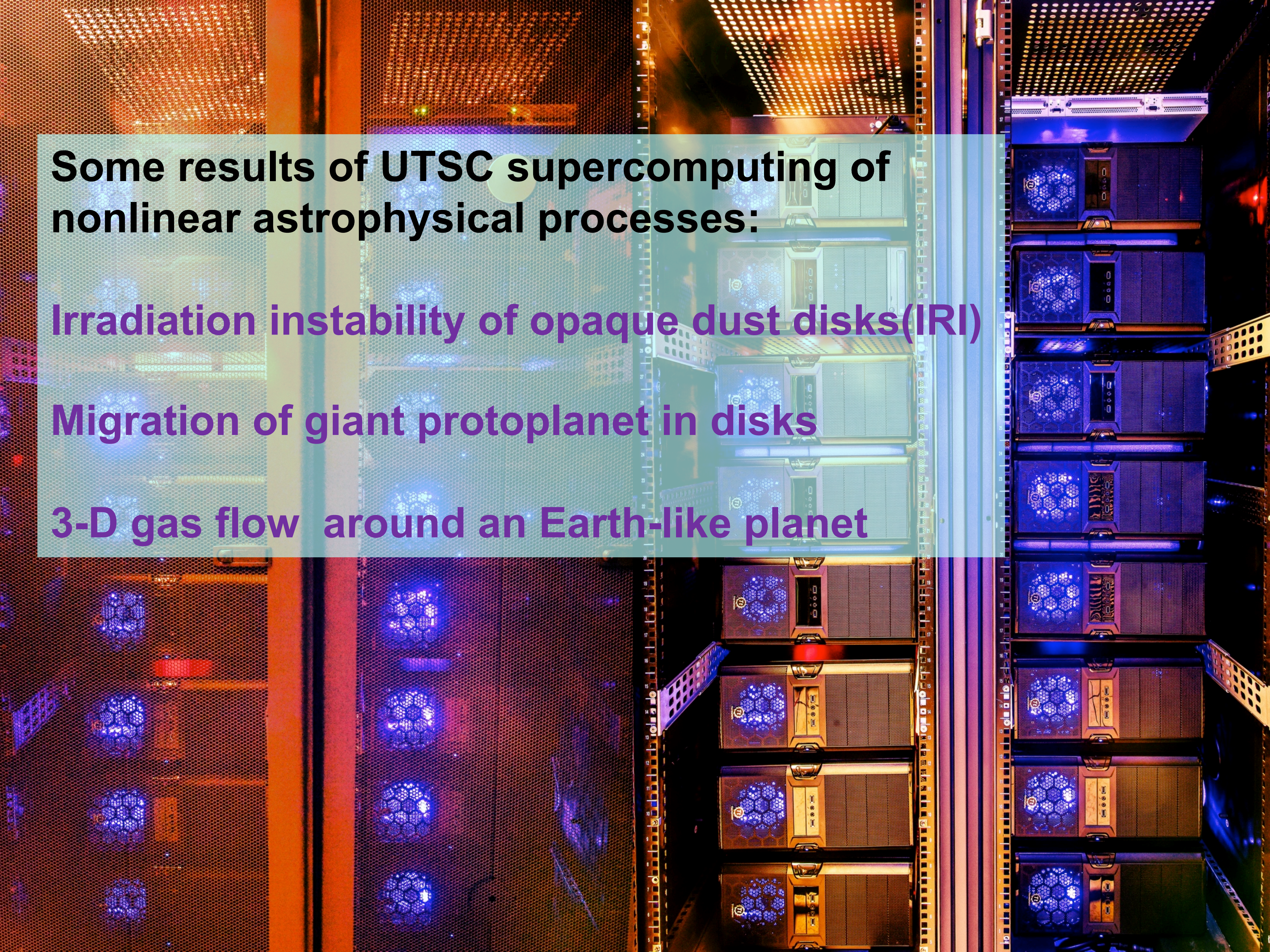
Two medium sized clusters, with a theoretical as well as practical performance equal to **64 TFLOP sp = 32 TF dp**
~1/10 of UofT’s **SciNet** at the time.

SciNet was the ~29th fastest computer in the world in ~2012.
Cost $50+ mln, continuously uses 1.5 MW of electric power,
(>$1.6 mln/yr electricity bill)

Our machines were 10 times more energy efficient (greener).
They cost 1000 less, and were 100 times more cost efficient
(price : performace 100 better than for SciNet).



Art + SciPhi,
at UTSC/UofT
were
1/1000 of #1
1/10 of SciNet



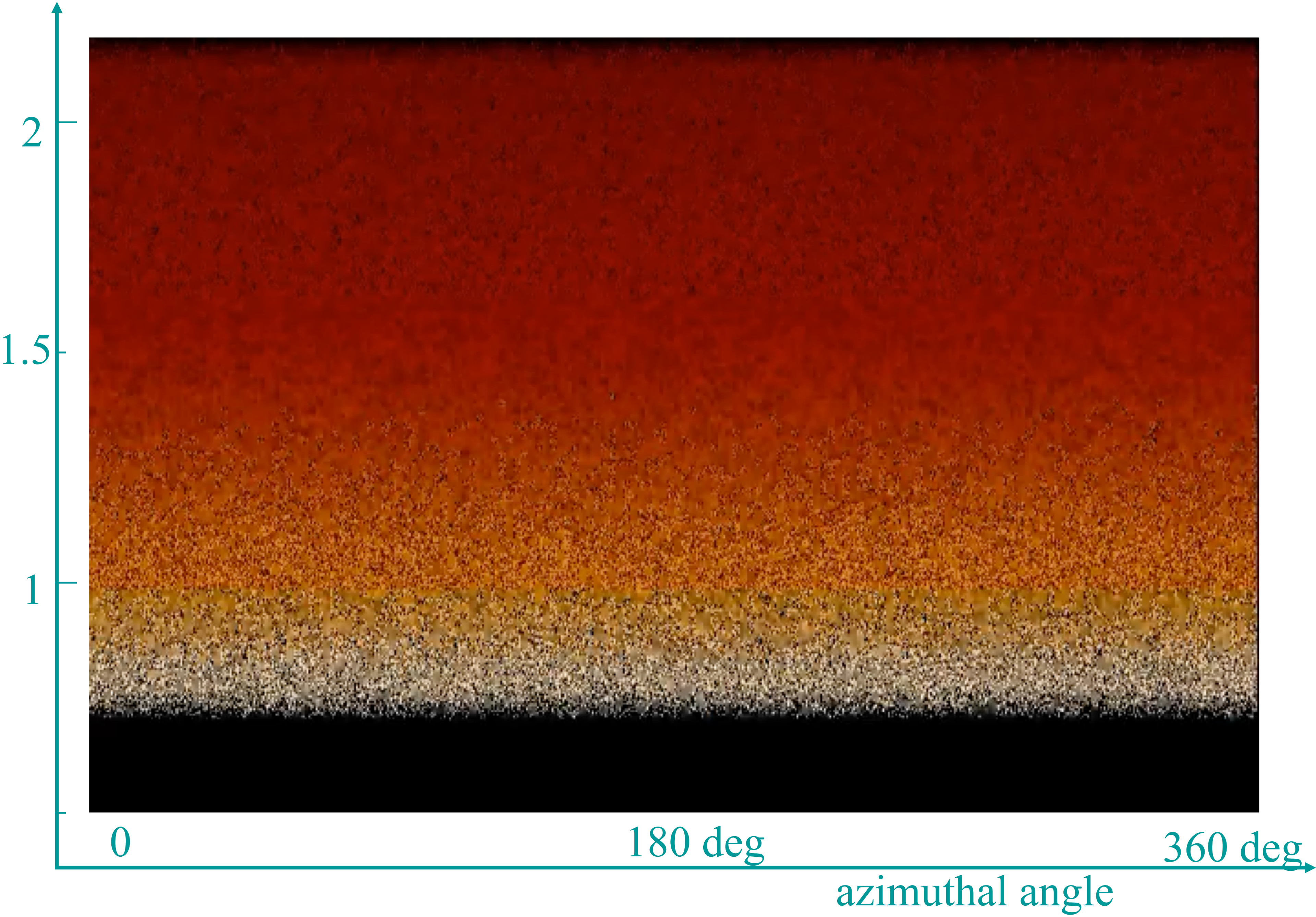
Some results of UTSC supercomputing of nonlinear astrophysical processes:

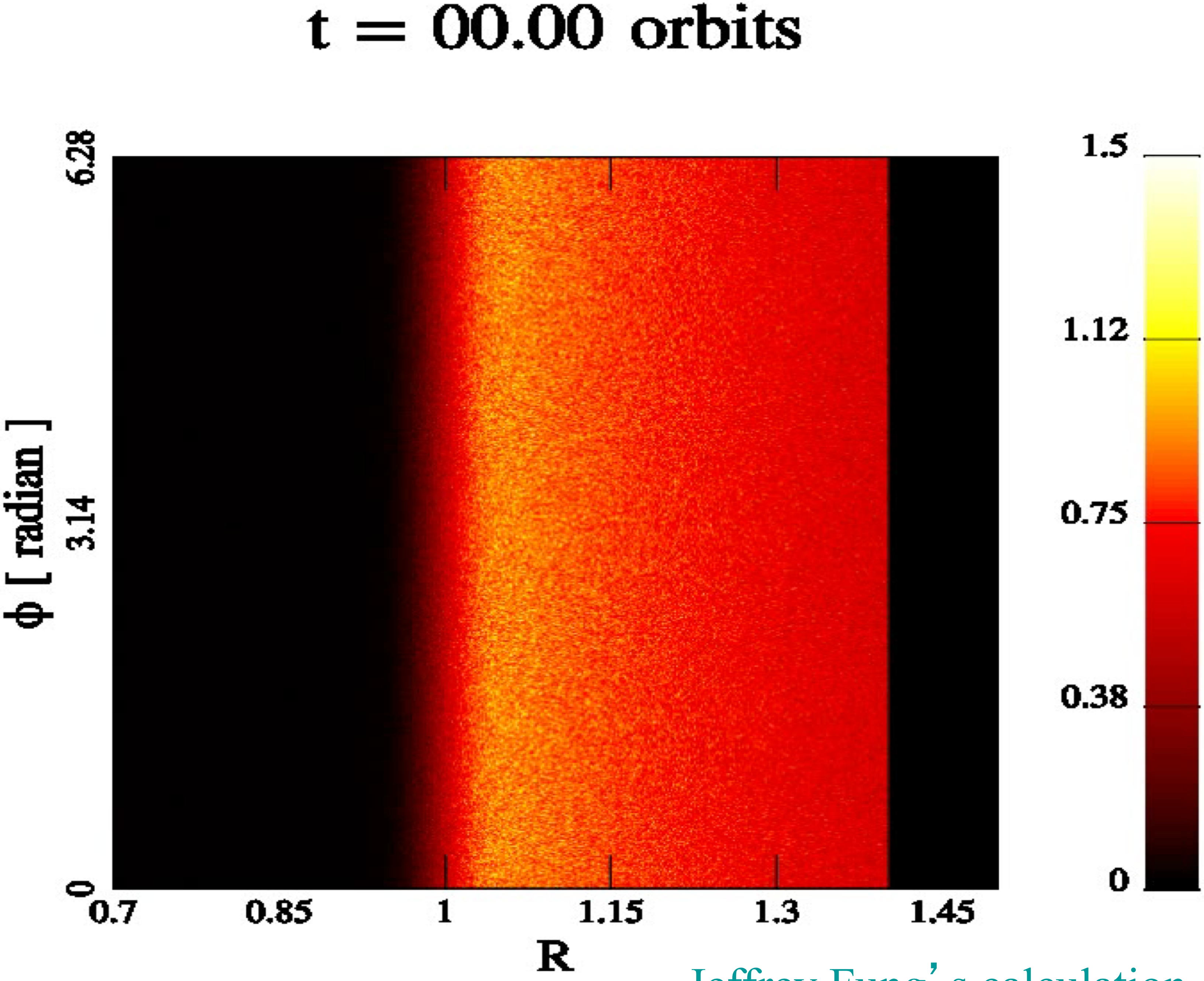
Irradiation instability of opaque dust disks(IRI)

Migration of giant protoplanet in disks

3-D gas flow around an Earth-like planet

Free particles casting shadows video **$\tau = 4$, $\beta = 0.2$**

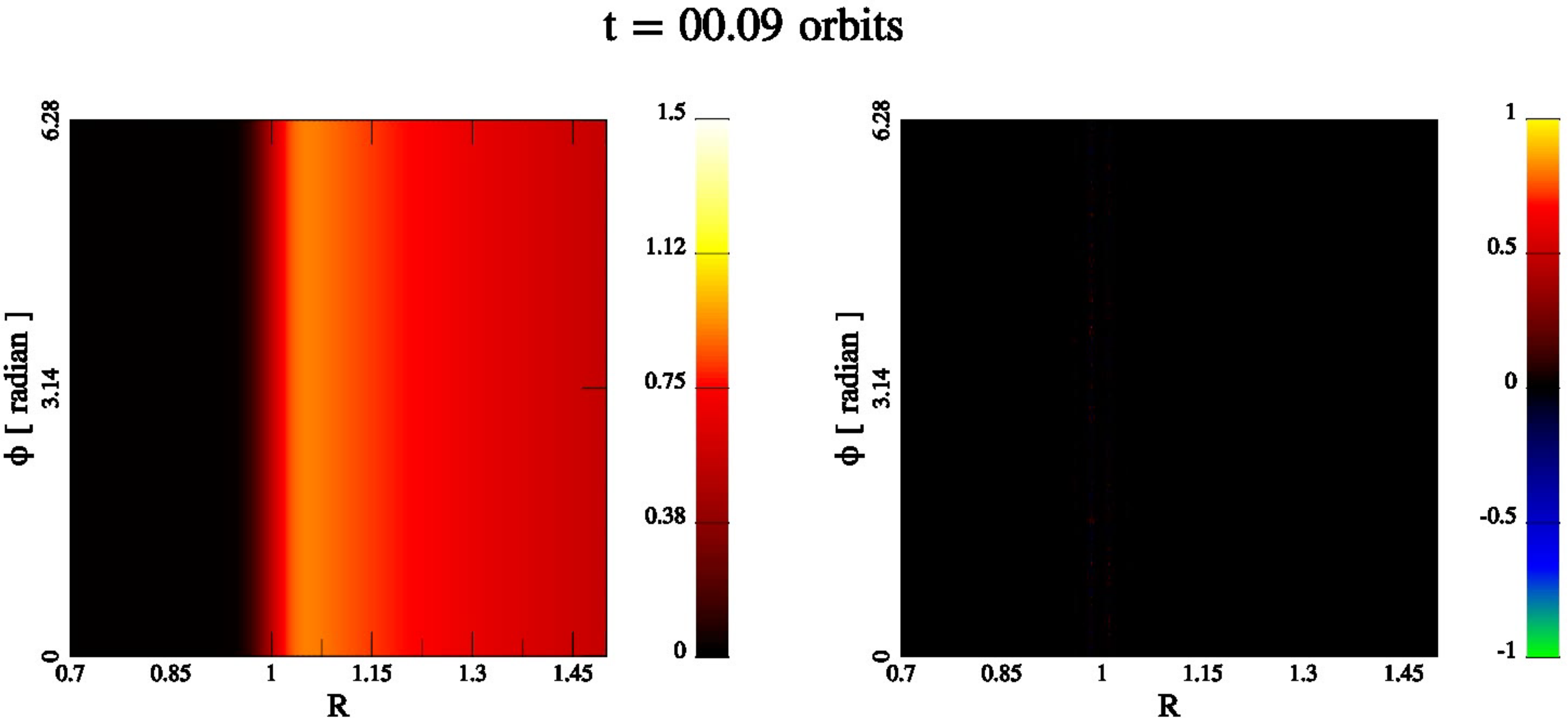




Jeffrey Fung’s calculation

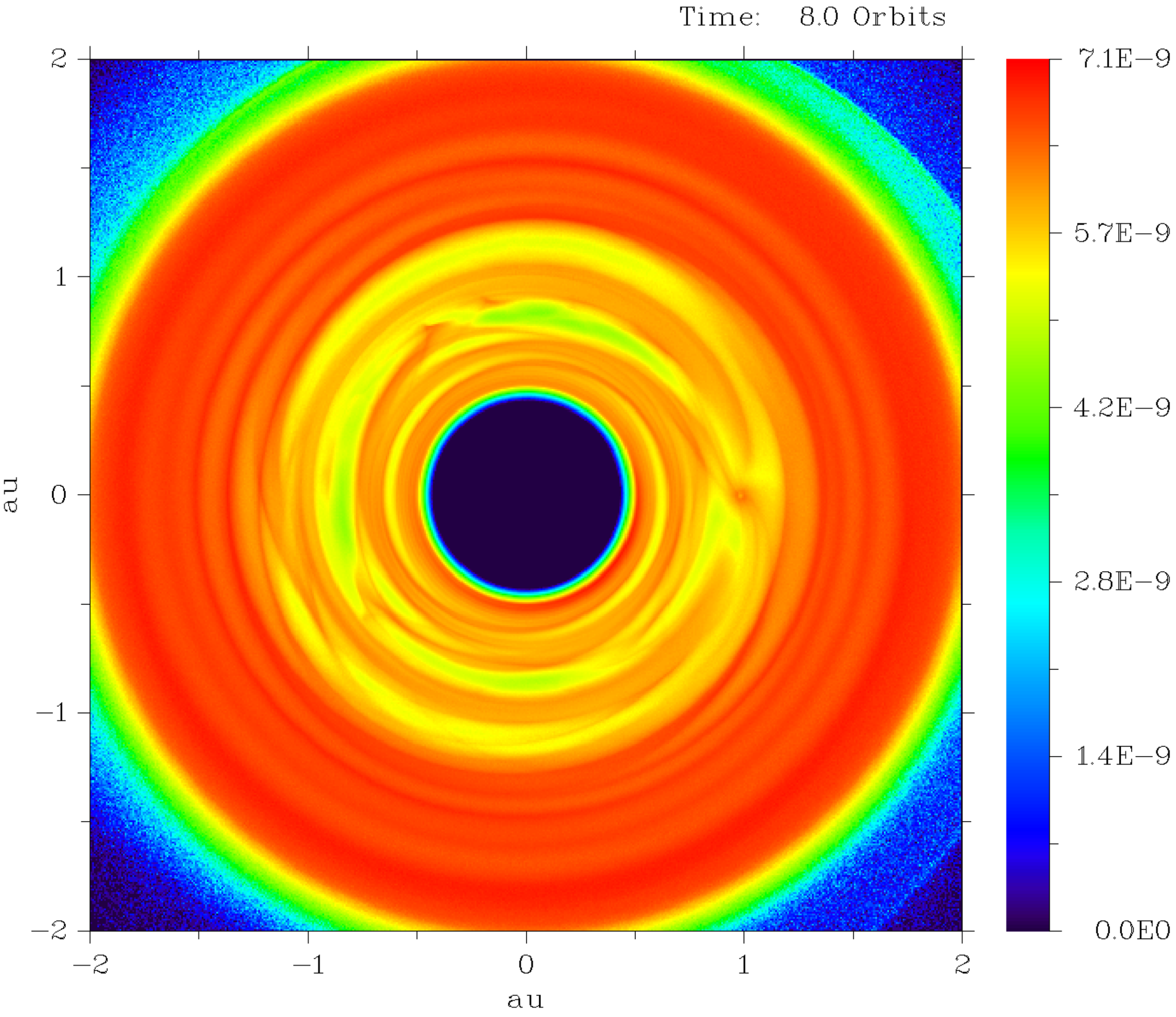
GAS DISK HYDRODYNAMICAL SIMULATION (PPM method)

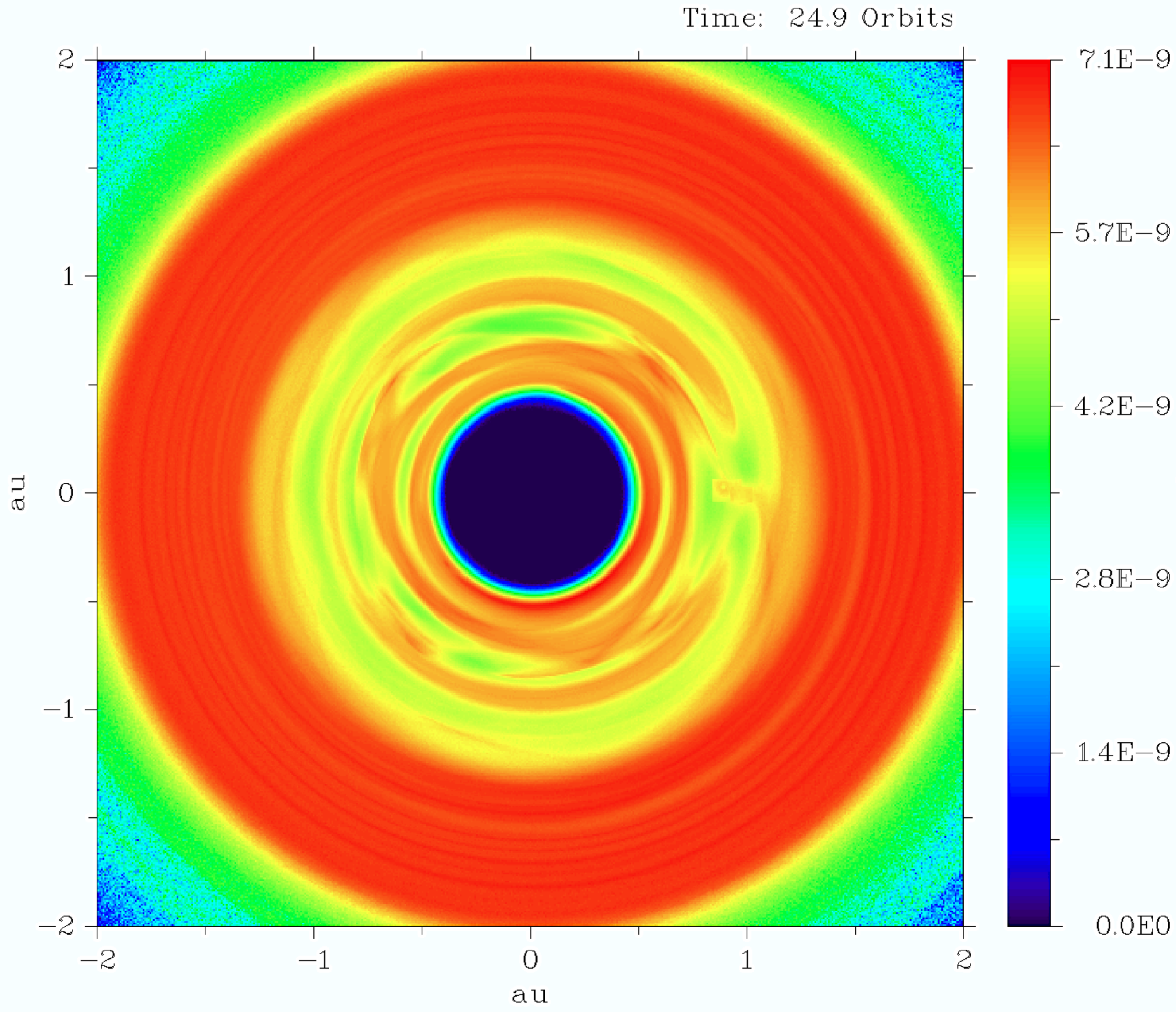
The r.h.s. shows a background-removed picture of density variations in growing modes. They are predicted analytically, and their growth rates are in agreement with calculations.



Thus opaque disks are unstable under illumination by the central object

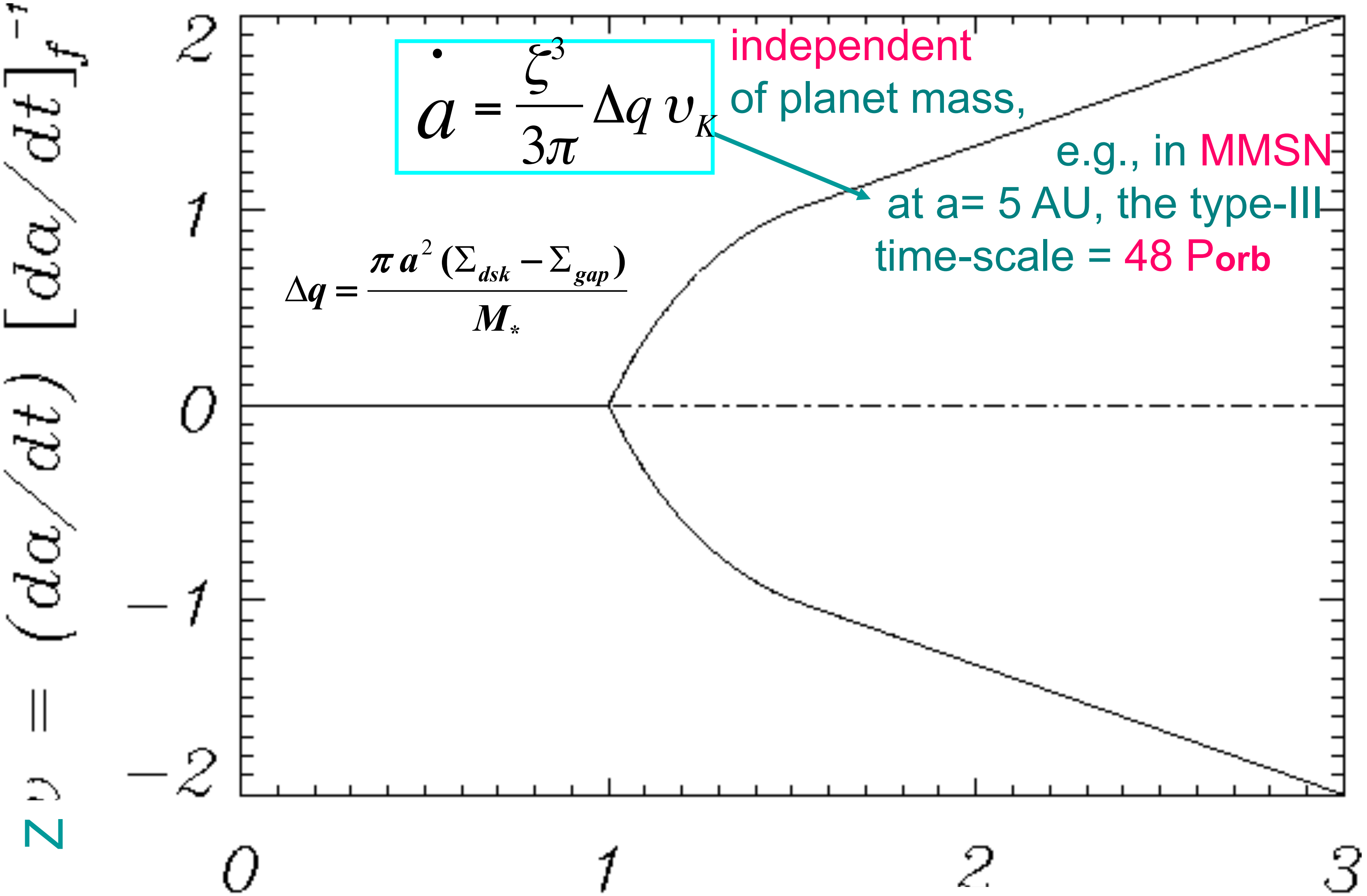
Fergus Horrobin (2017). Simulation of collisionless disk of 1 billion particles perturbed by Jupiter





Density Plot of X, Y Plane

Migration type III, neglecting LRs & viscous disk flow



$$\dot{a} = \frac{\zeta^3}{3\pi} \Delta q\, \upsilon_K$$

independent of planet mass,

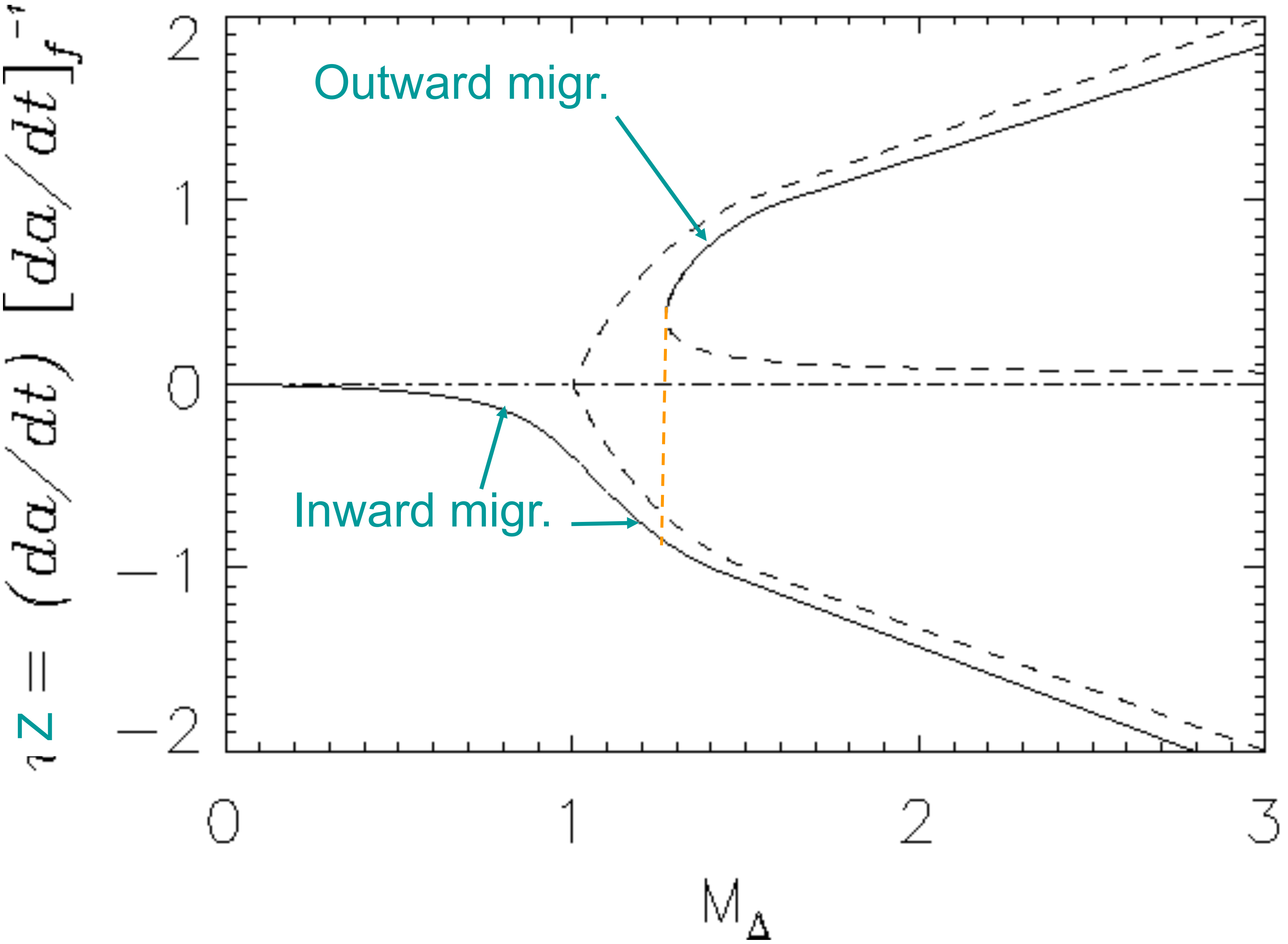
e.g., in MMSN at a= 5 AU, the type-III time-scale = 48 P**orb**

$$\Delta q = \frac{\pi a^2 (\Sigma_{dsk} - \Sigma_{gap})}{M_*}$$

$$M_\Delta = \frac{4 x_{CR}}{\mu} \Delta q \quad \sim \mu^{-2/3}\, \Delta q$$

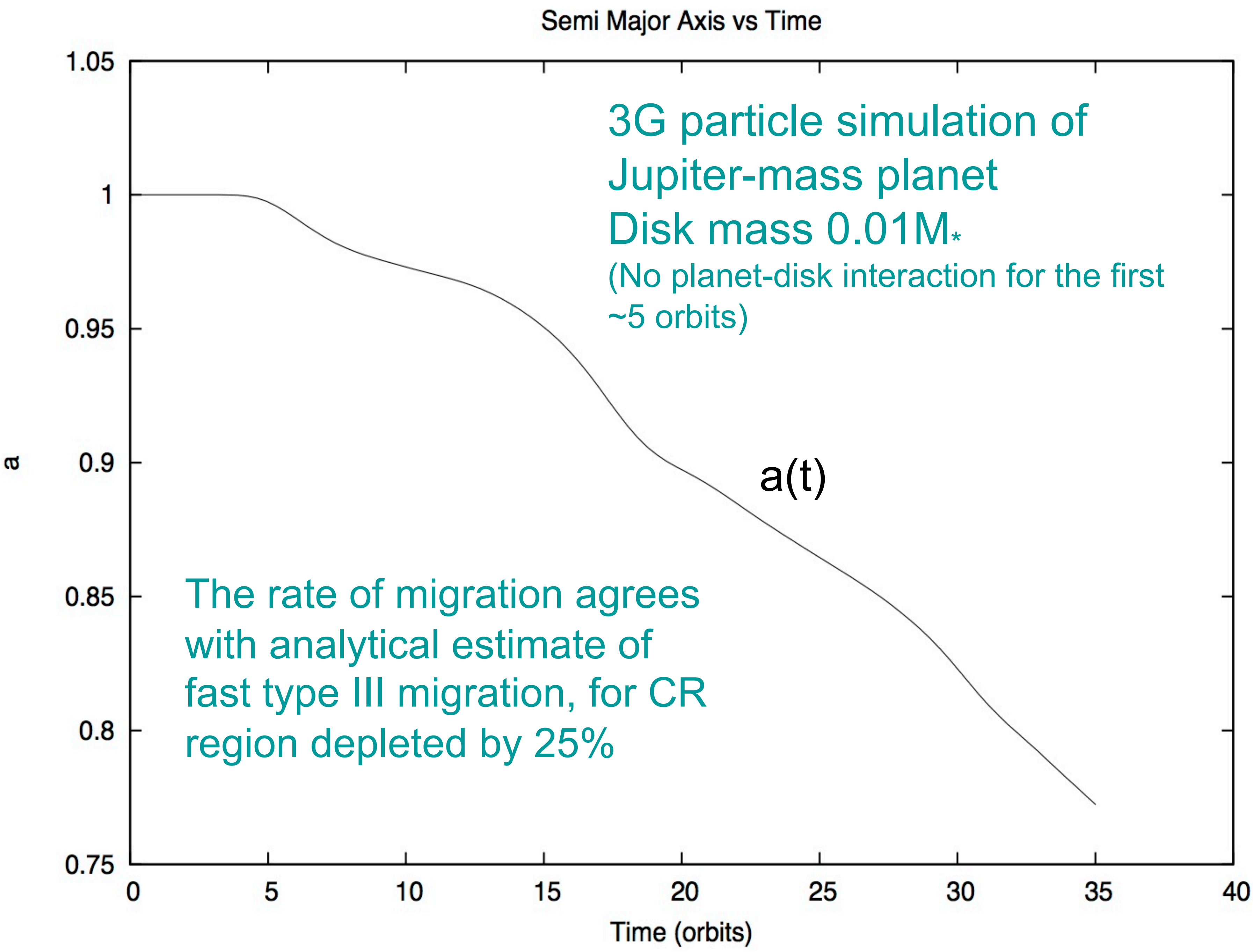
from:
Papaloizou, Nelson, Artymowicz PP V 2006

Migration speed including all torques (CR+LR+viscous), LRs bias toward inward migration



Outward migr.

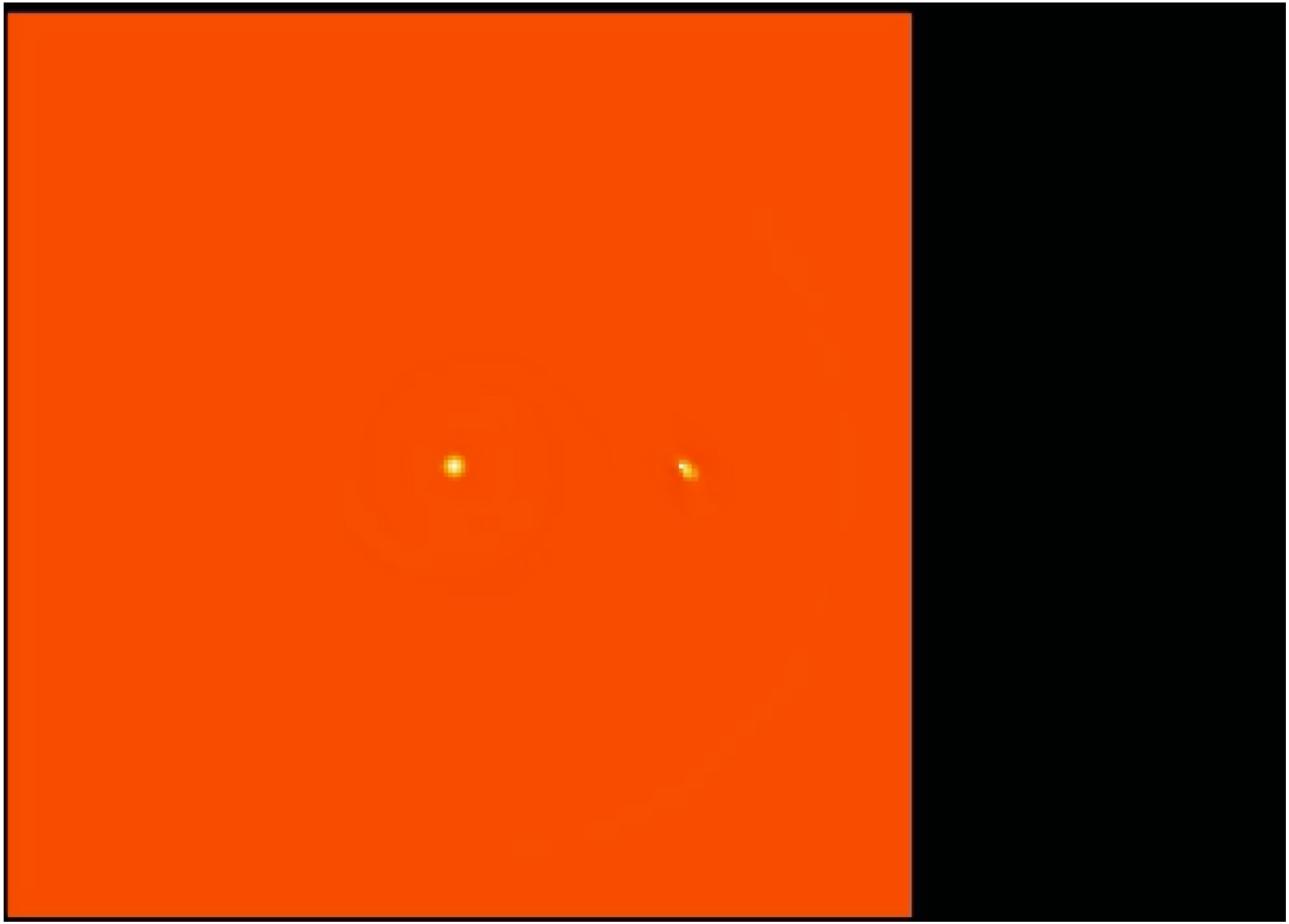
Inward migr.



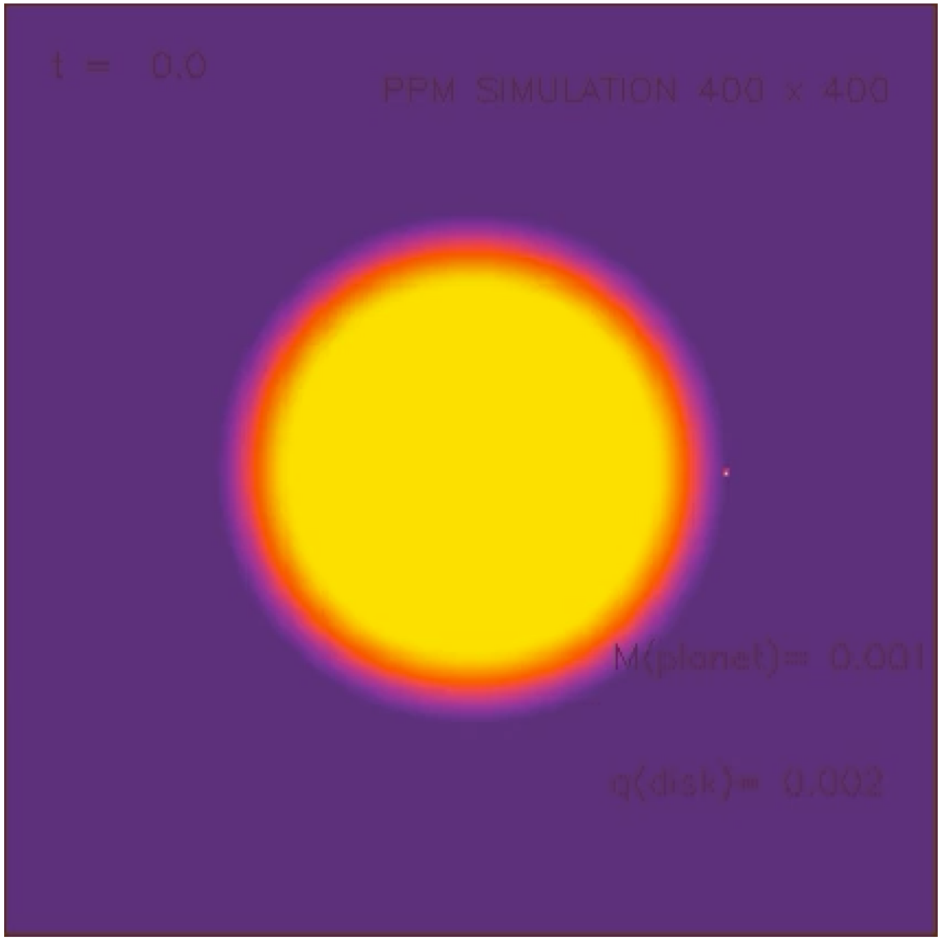
3G particle simulation of
Jupiter-mass planet
Disk mass $0.01M_*$
(No planet-disk interaction for the first ~5 orbits)

The rate of migration agrees with analytical estimate of fast type III migration, for CR region depleted by 25%

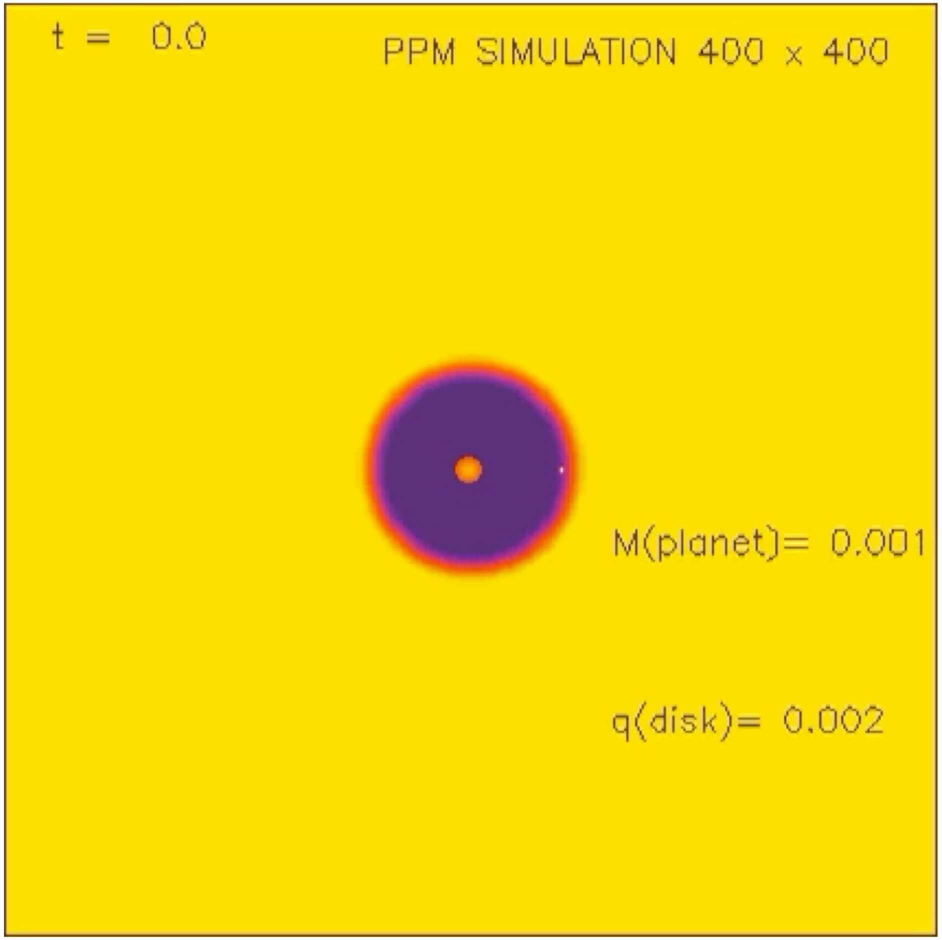
Artymowicz (2000) - protojupiter migrating inward in protoplanetary disk



The rapid inward/outward migration in the direction opposite to standard theory of the tidal disk-planet interaction (via Lindblad resonances)

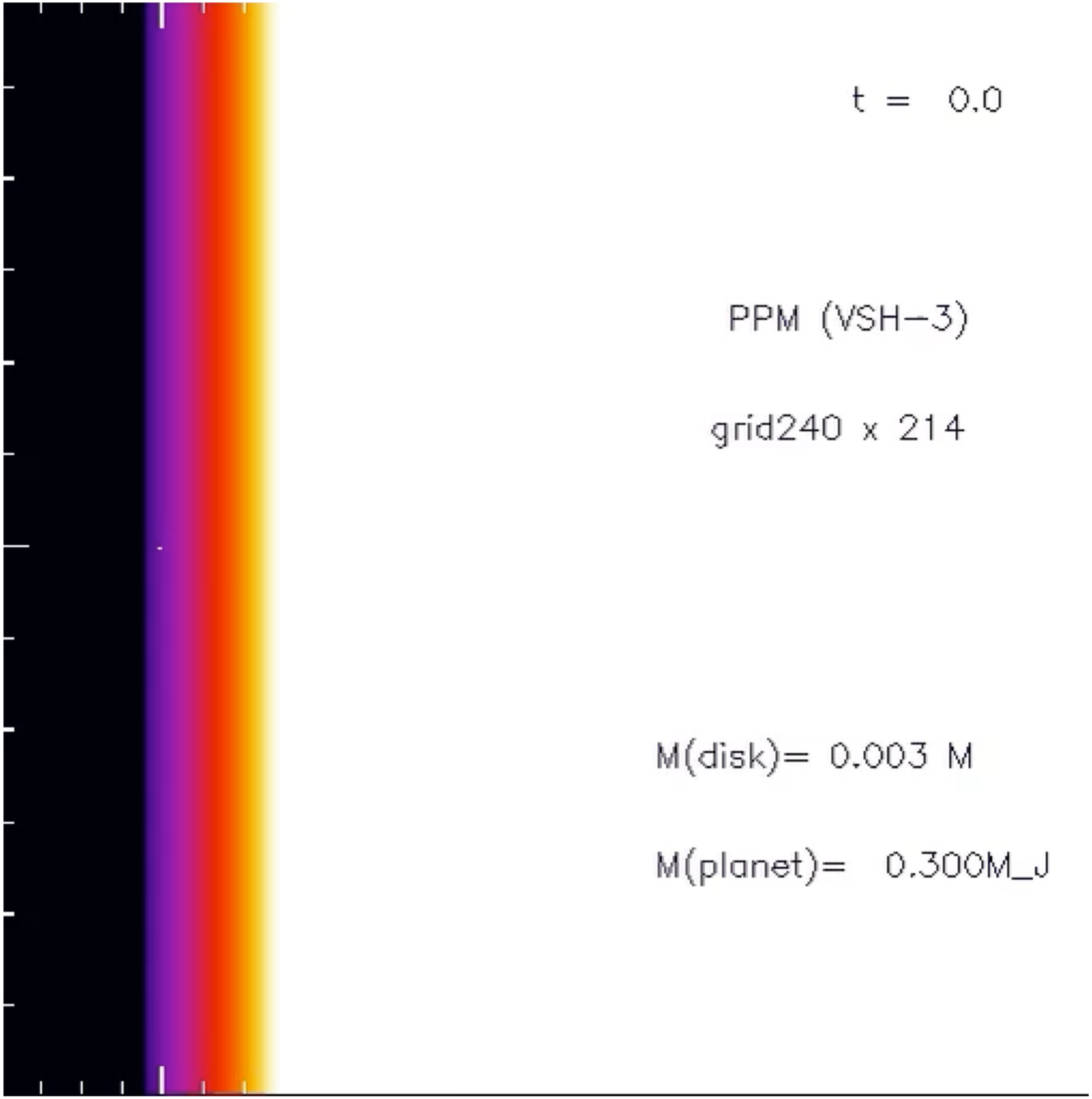


Initially a jupiter-mass planet outside the disk
Migrates inward



Initially inside a disk gap
Migrates outward

The so-called type III migration is very rapid and can create hot jupiters



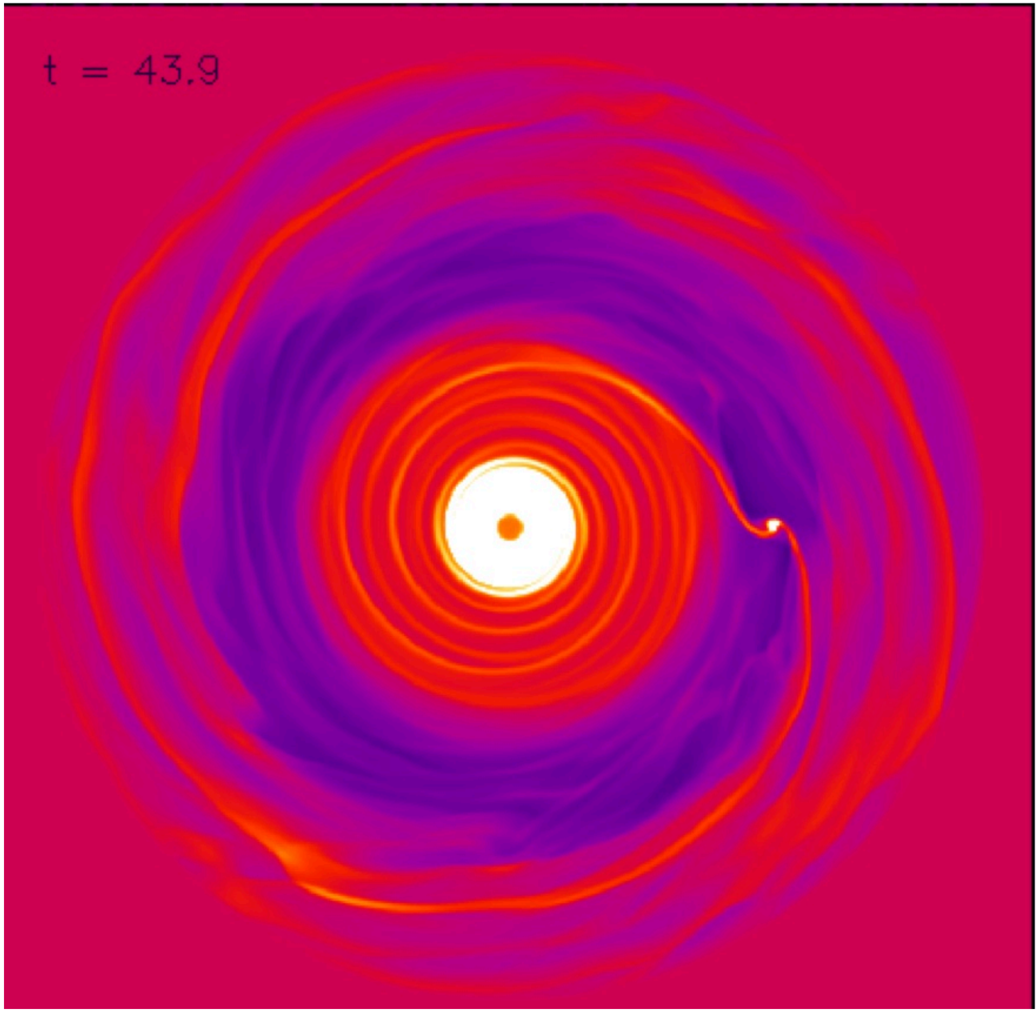
Simulation of Proto-Saturn in a primordial disk

(r,phi) view

Full range (2π) of azimuth angles shown on vertical axis.
Horizontal axis shows radius from r = 1 to r = 3

Previously, only 2-D simulations were possible.

We’ve recently simulated a small, embedded planet of 5 Earth masses in a protoplanetary disk in 3-D.



The results show many new phenomena, such as:

1. Columnar flow resembling Taylor-Proudman columns in rapidly rotating fluids, and
2. Wake vorticity genaration by the planet (4 counter-rotating vortices)

Fung, Artymowicz, and Wu (2015, Astroph. J.)

2-D

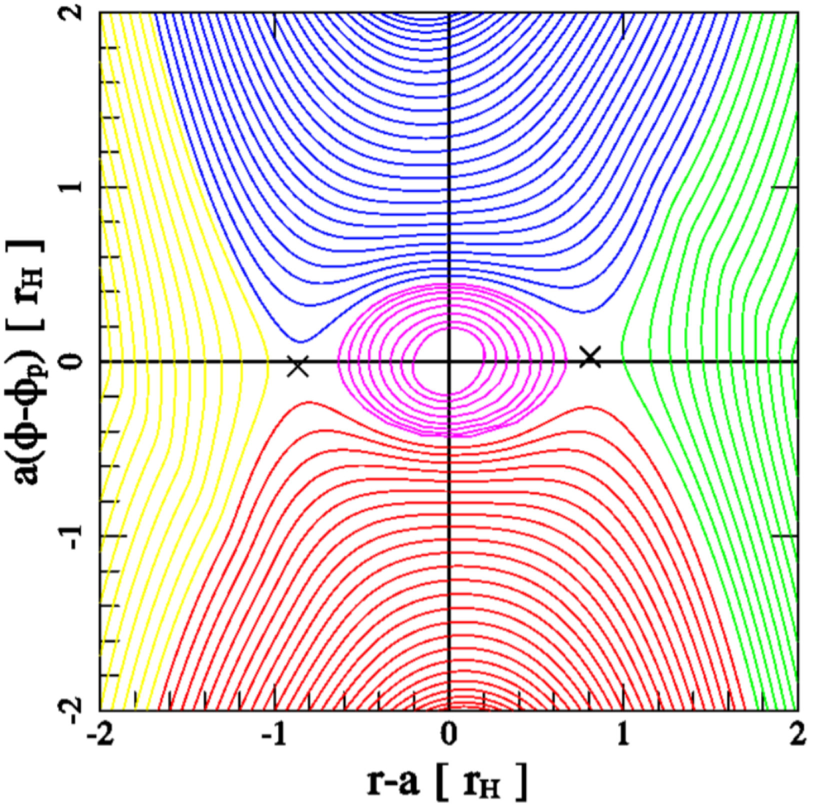


FIG. 1.— Streamlines around a planet in 2D, plotted in the corotating frame

3-D

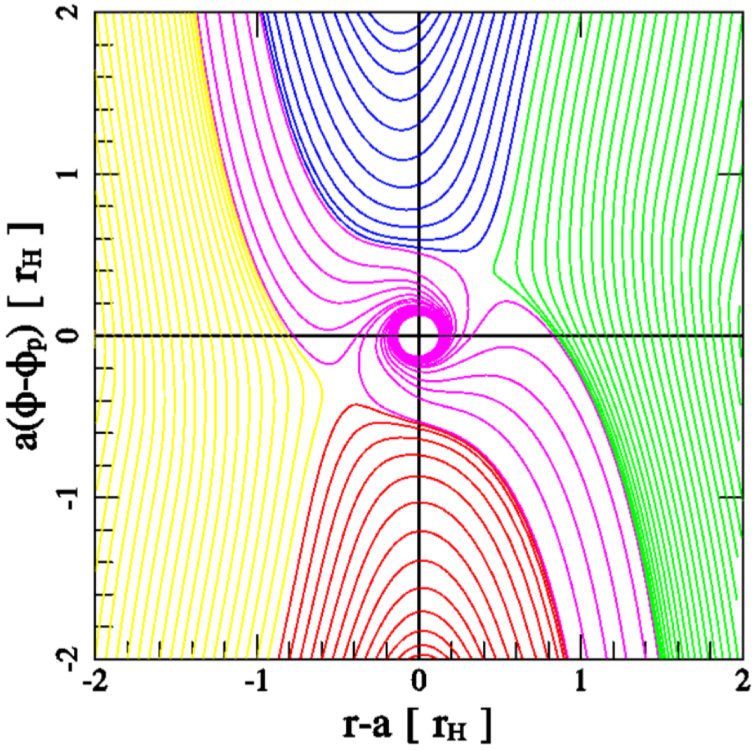
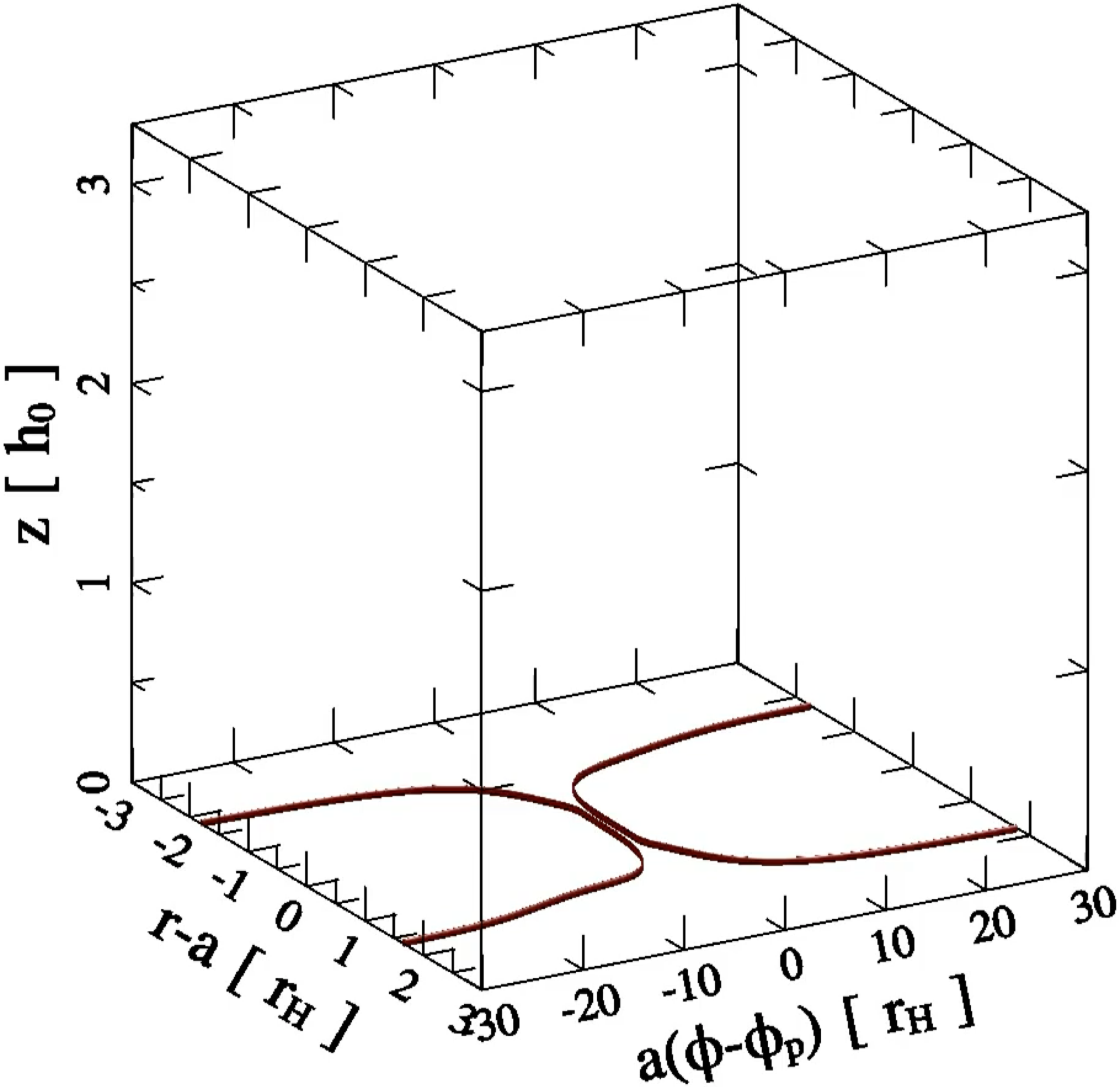
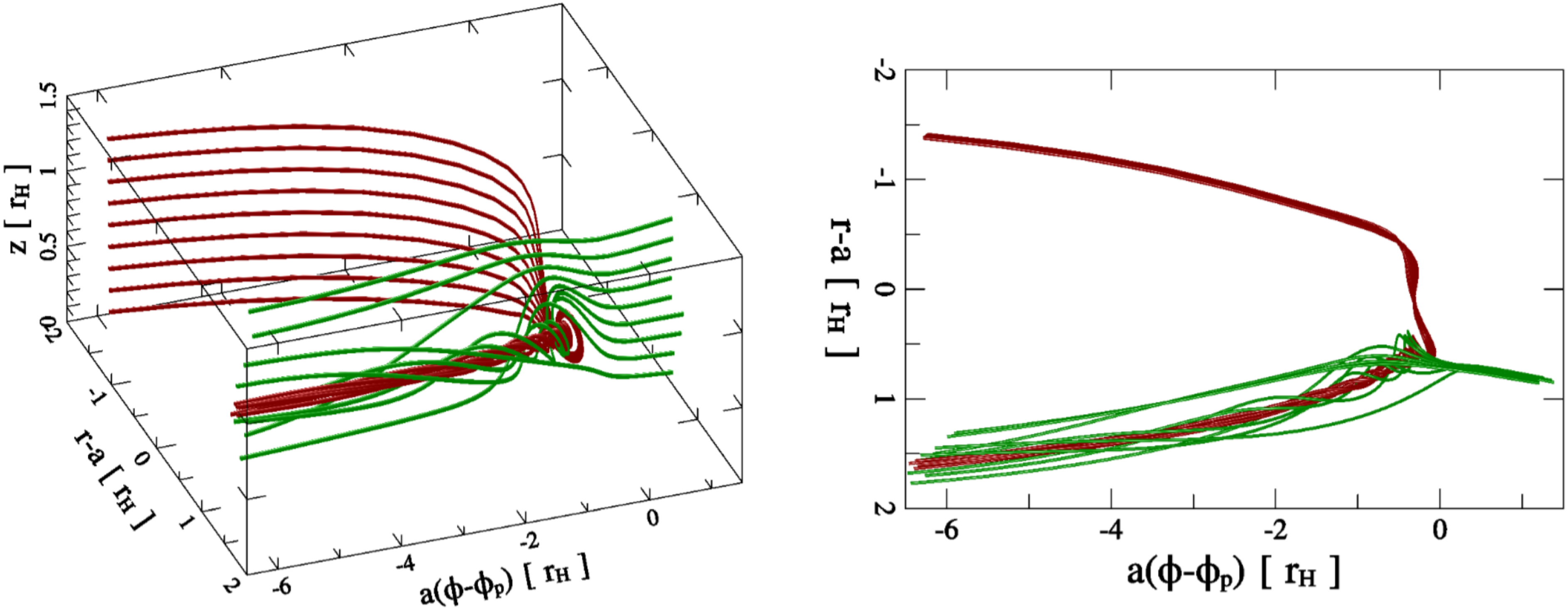
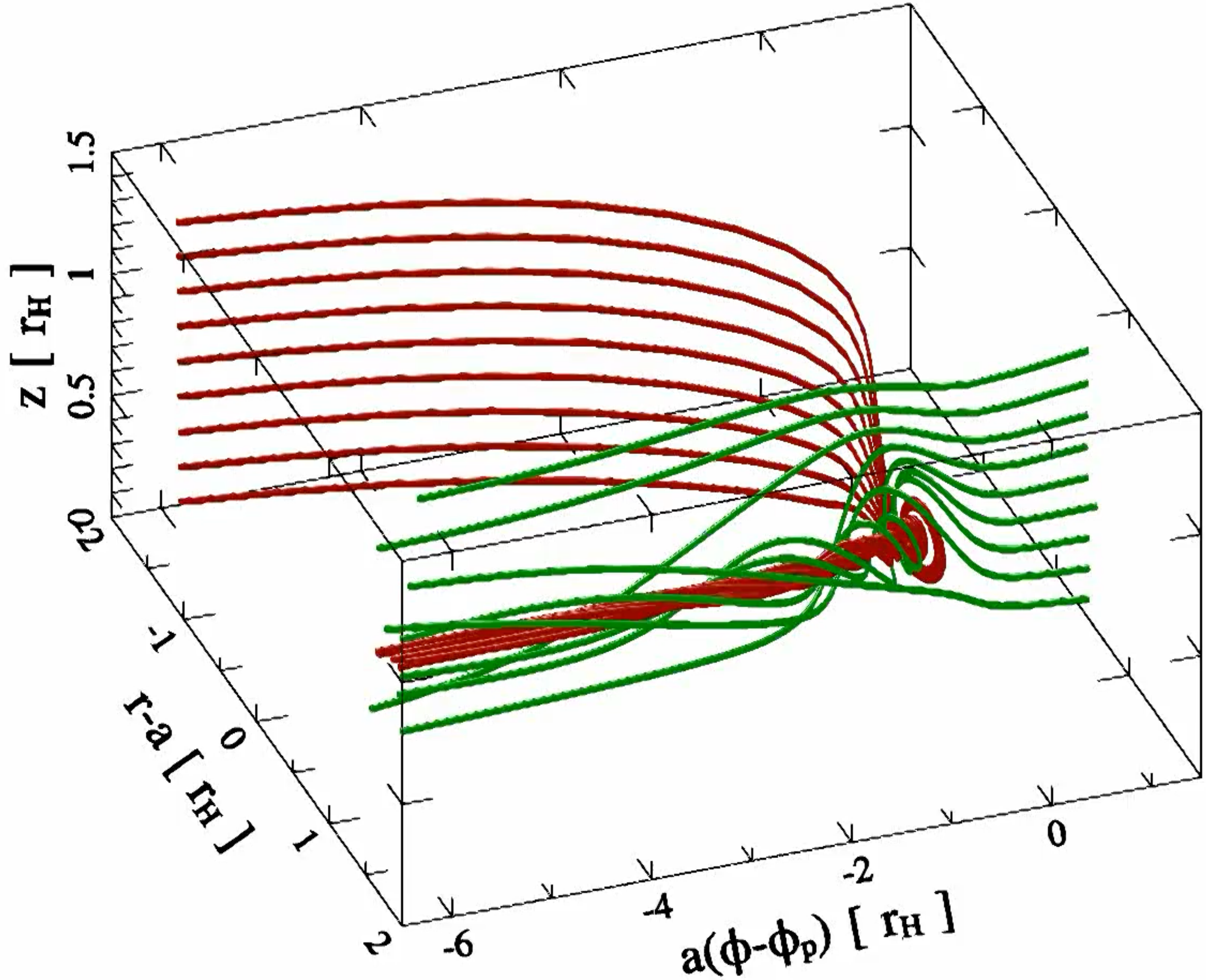


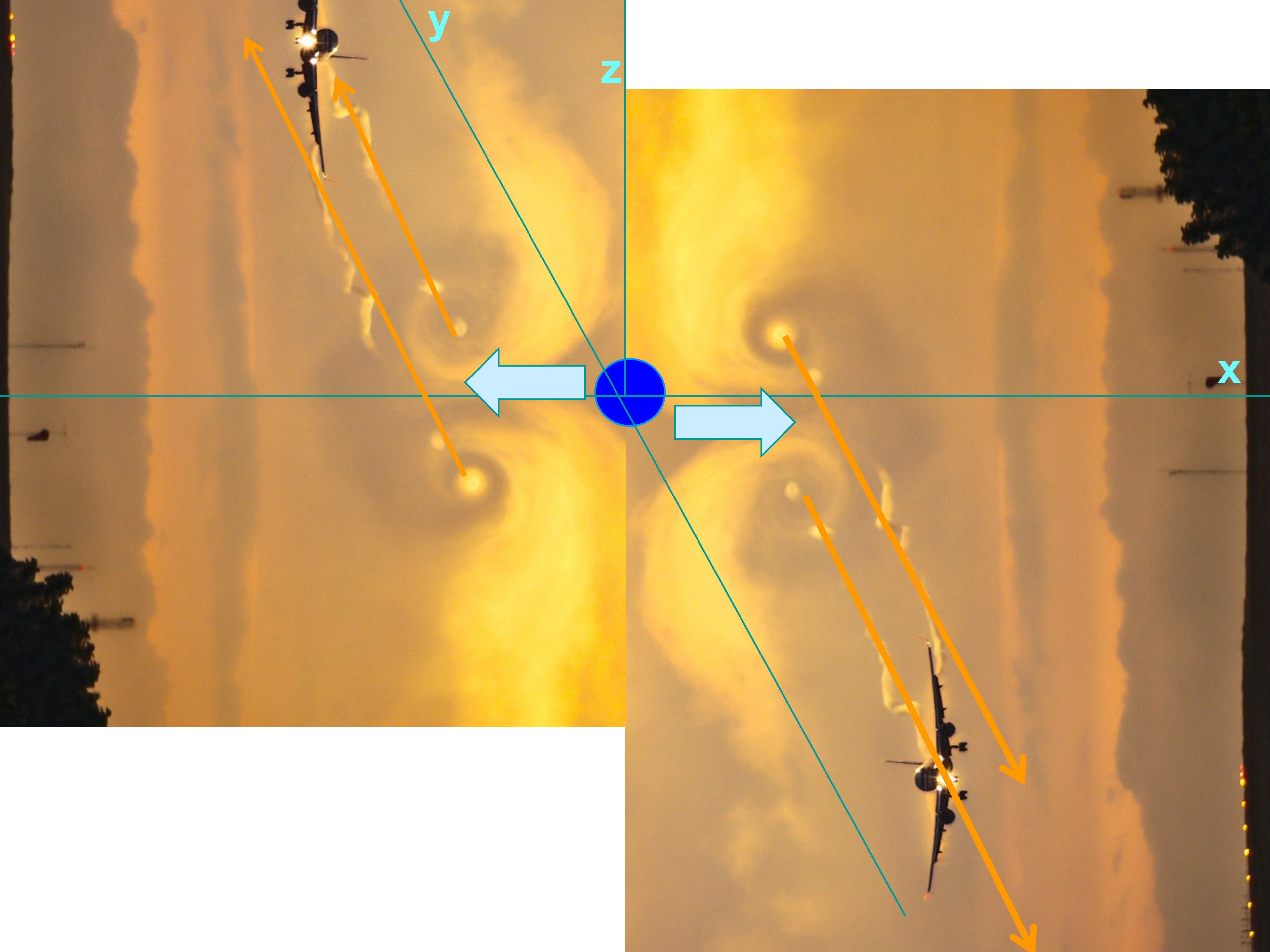
FIG. 8.— Streamlines in the disk midplane. Compare with Figure 1 for differences between 2D and 3D flow. Yellow, red, green, and blue streamlines are assigned in the same manner as Figure 1. Unlike Figure 1, magenta lines are outflows away from the planet, pulled down from initially higher altitudes.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

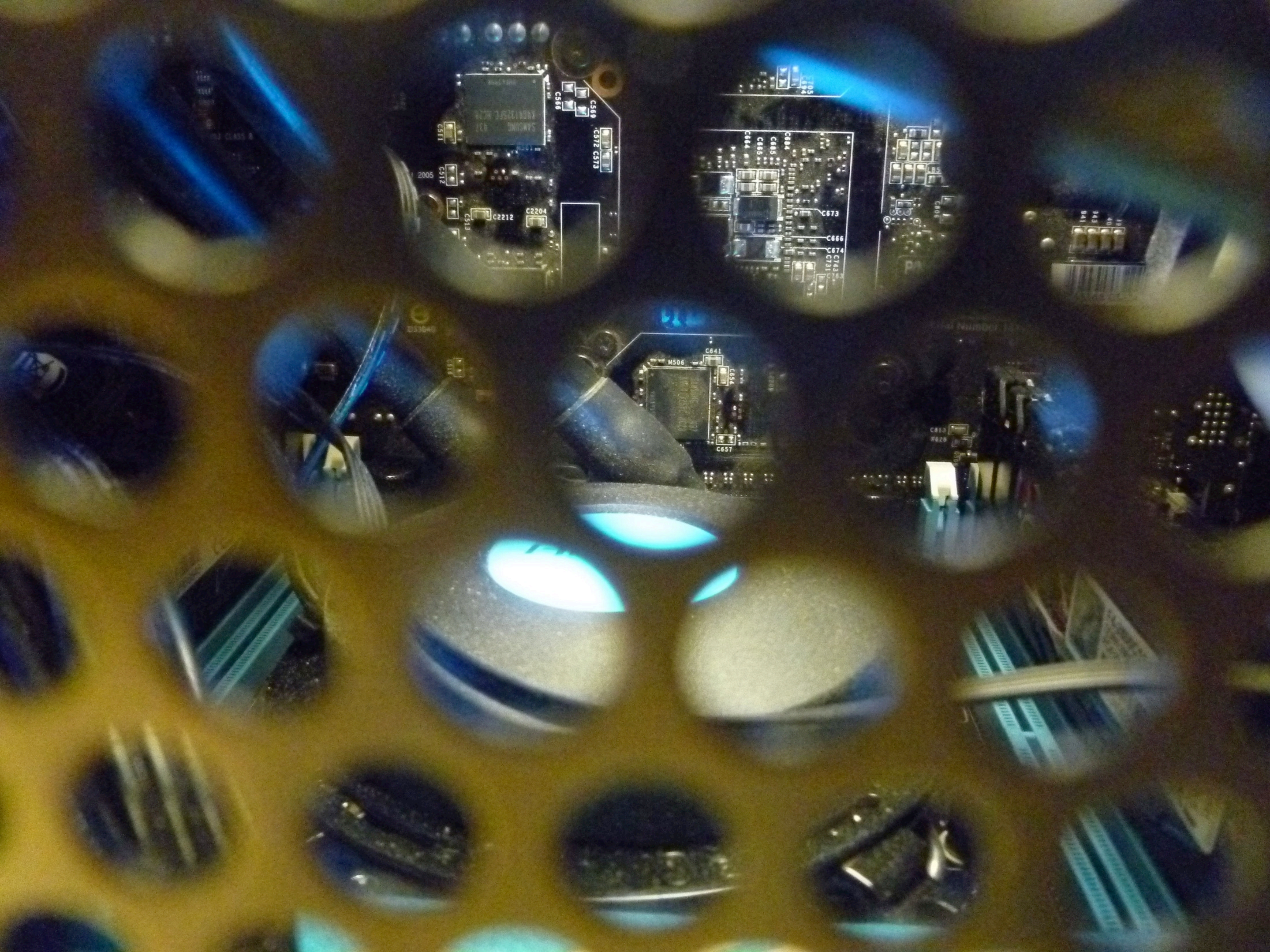

z [h_0]
0
1
2
3
r-a [r_H]
-3
-2
-1
0
1
2
3
a(ϕ-ϕ$_P$) [r_H]
-30
-20
-10
0
10
20
30

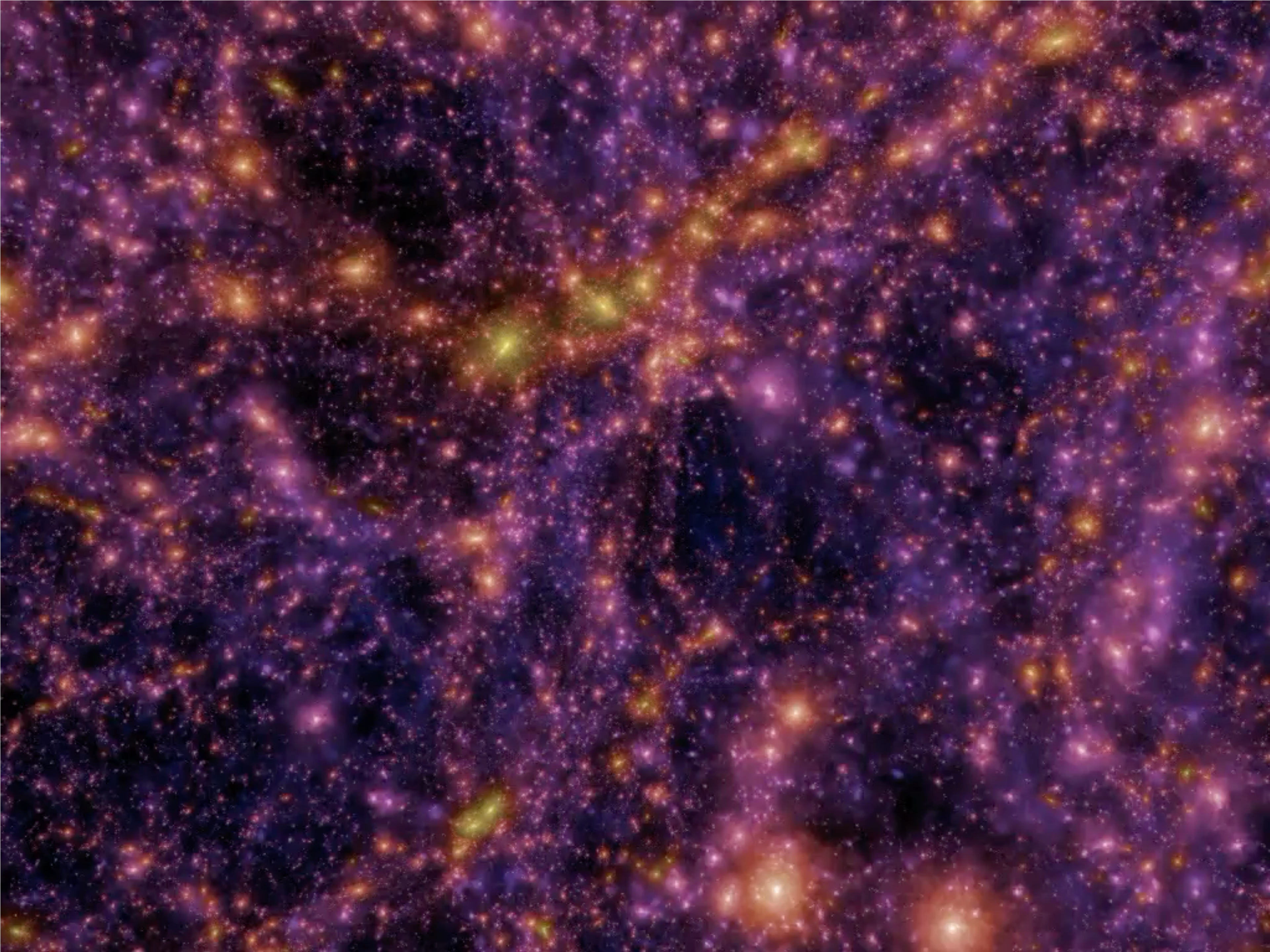
A total of 4 such vortices are shed by an embedded planet

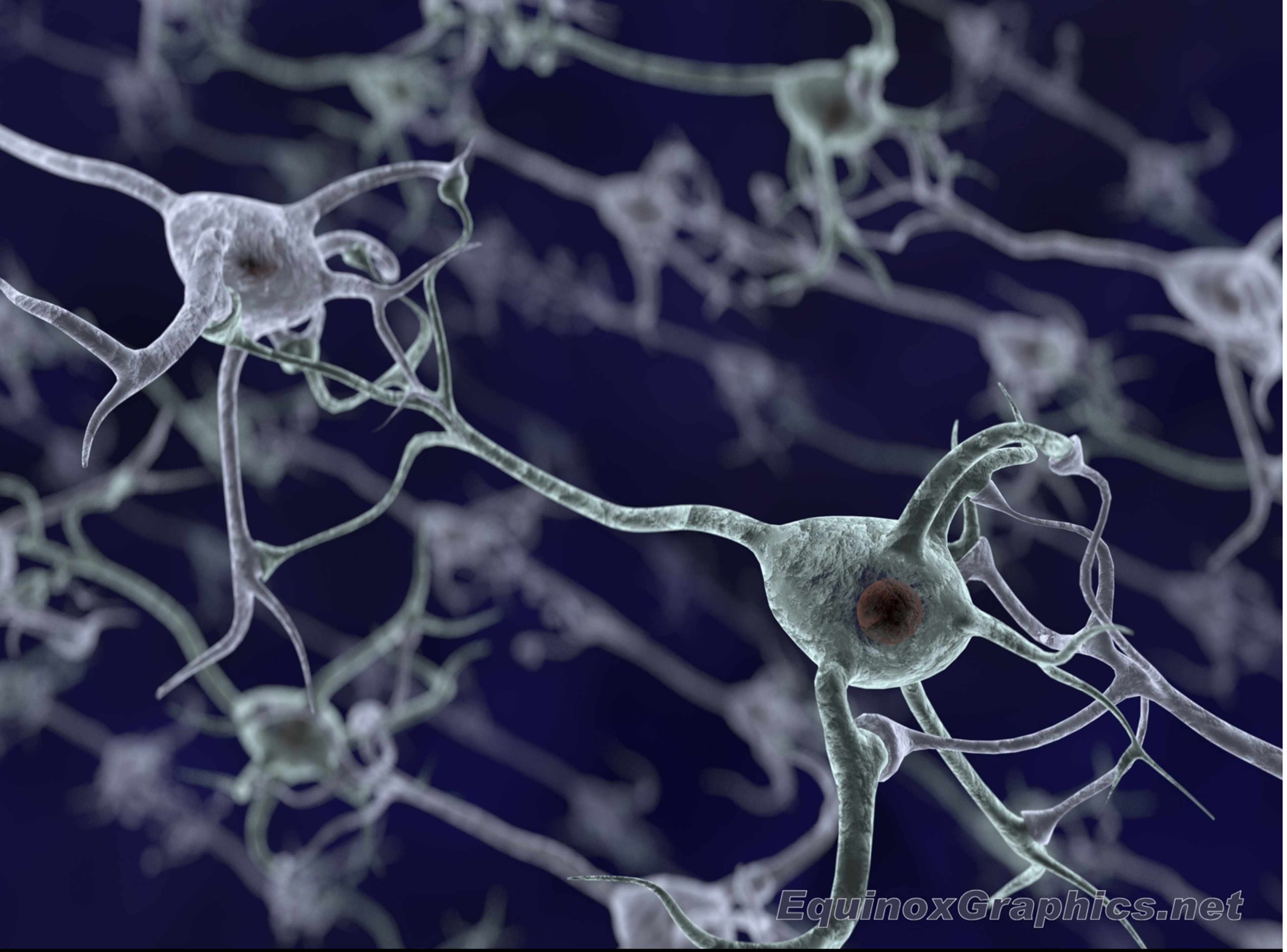



z [r_H]
1.5
1
0.5
0
-2
-1
0
1
2
r-a [r_H]
-6
-4
-2
0
a(ϕ-ϕ_p) [r_H]

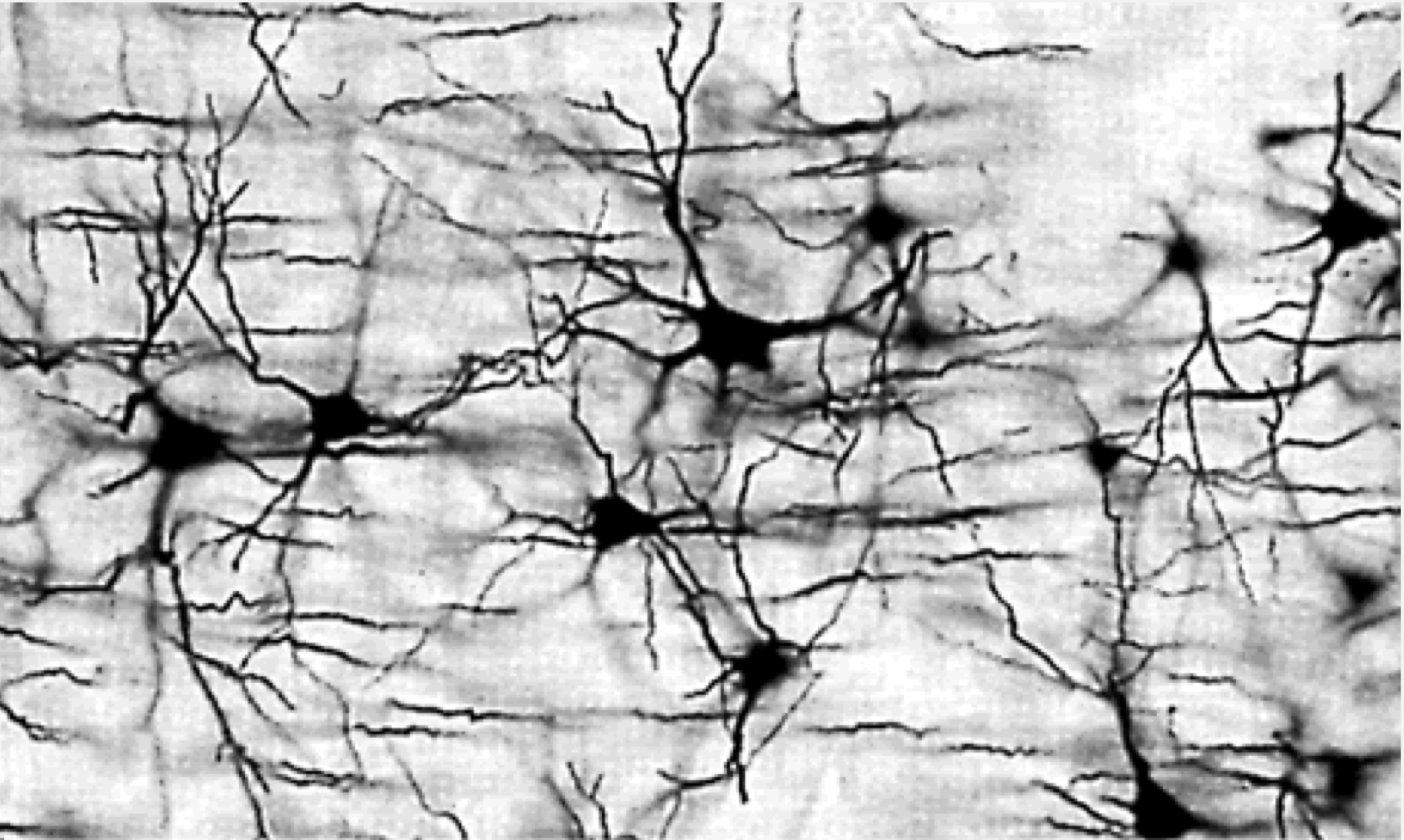

y
z
x


SAMSUNG
C566
C569
C572
C573
C511
C512
2005
C2212
C2204
C673
C666
C674
M506
C641
C654
C657
C813

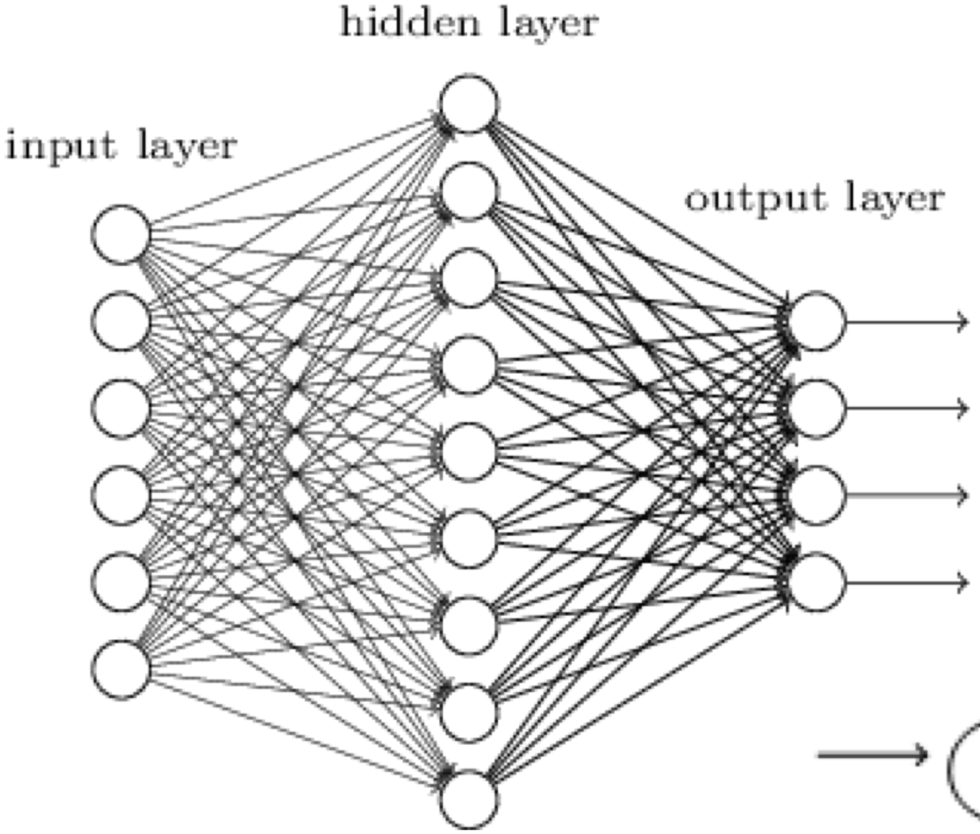


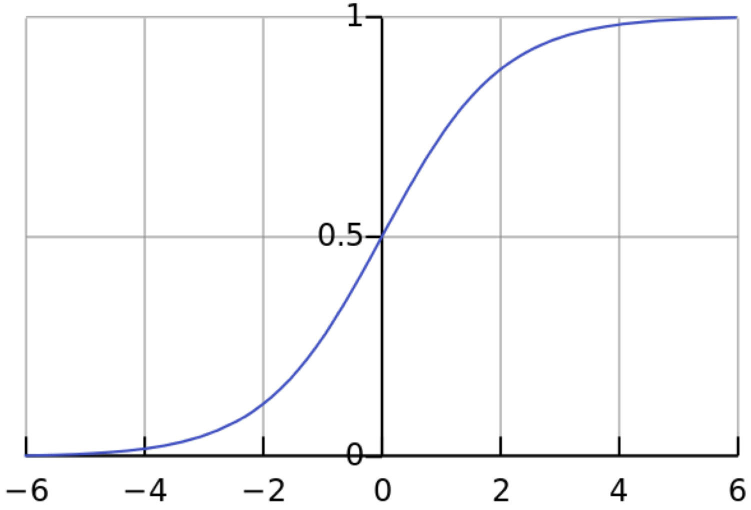


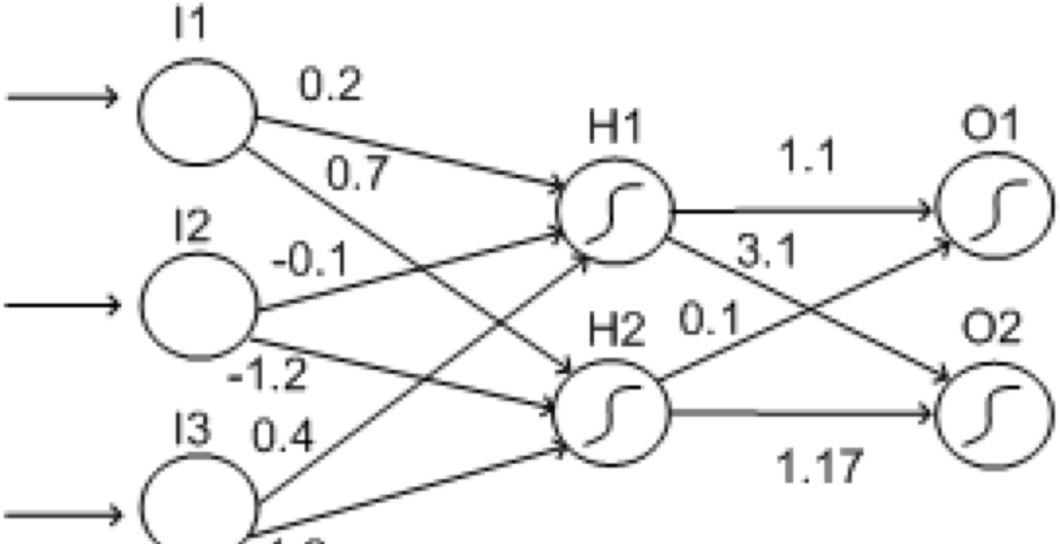
Can machines be taught to think?



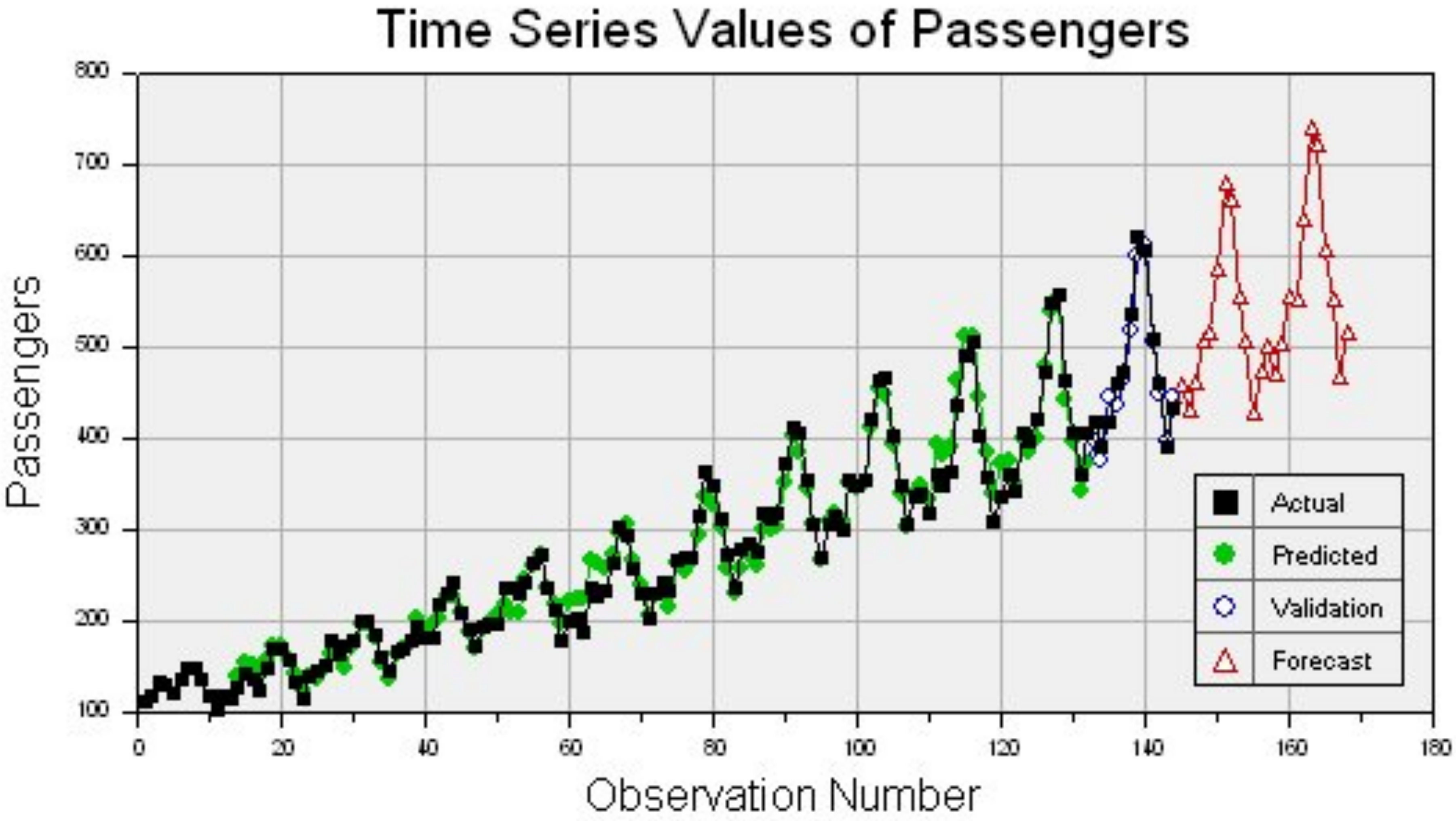
Artificial Neural Networks – computer simulates biologically-inspired layers of neurons which process information in parallel







AI can extrapolate some regular behavior into the future



Real uses of Artificial Intelligence:

- controls robots
- does voice recognition, e.g. in
- telephone menus
- trades stocks
- does data mining, e.g. Google
- designs chips
- detects fraud
- helps in scientific calculations
- image recognition, classification
- text analysis, auto-correction
- generates music
- drives cars
- plays chess, Go, better than us
- may one day be in your robot or friend, or your doctor

Computer intelligence
and data mining will be featured
in advanced computing (D-level) course at UTSC