The Abdus Salam
International Centre for Theoretical Physics

United Nations
Educational, Scientific and
Cultural Organization

IAEA
International Atomic Energy Agency

2292-22

School and Conference on Analytical and Computational Astrophysics

14 - 25 November, 2011

Multiscale and Multiphysics Challenge in Modeling of Space Weather - 2

Giovanni Lapenta

Katholic University of Leuven, Centre for Plasma Astrophysics
Belgium

Strada Costiera 11, 34151 Trieste, Italy - Tel.+39 040 2240 111; Fax +39 040 224 163 - sci_info@ictp.it

Centrum voor Plasma-Astrofysica

KATHOLIEKE UNIVERSITEIT
LEUVEN

Multiscale and Multiphysics Challenge in Modeling of Space Weather

Giovanni Lapenta

Funding from:
BOF & GOA (KU Leuven),
EC (Swiff, Soteria),
NASA (MMS Mission),
Intel Exascience Lab.

Outline

KATHOLIEKE UNIVERSITEIT LEUVEN

Meeting the multiscale challenge

- PIC Method
- Implicit PIC
- Moment method

Application to space weather

- Local simulations
- Global simulations

Meeting the multiphysics challenge

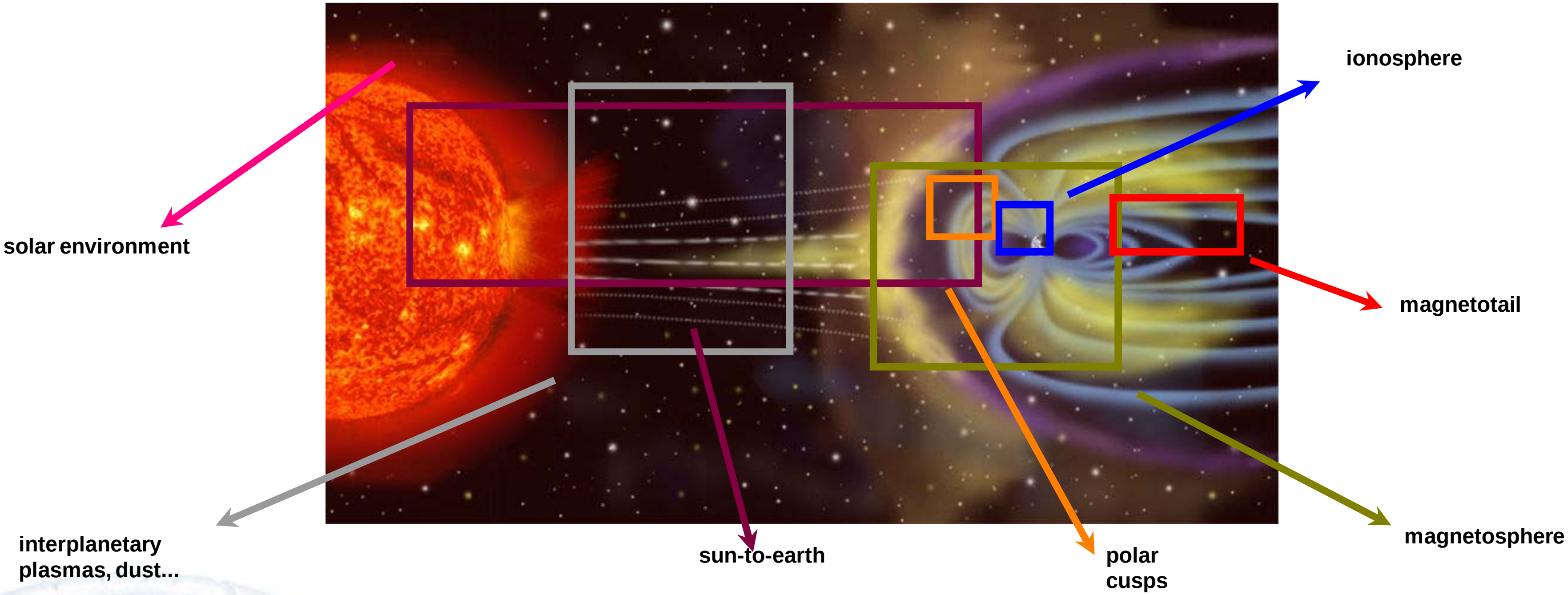
- Multi-level simulation
- AMR

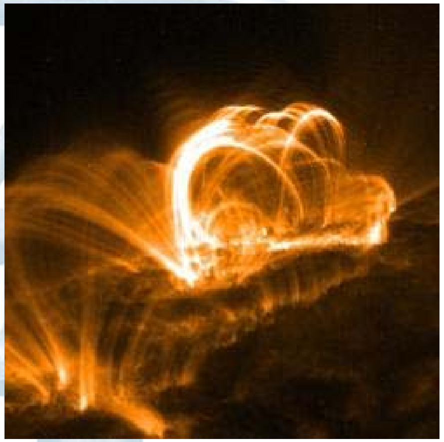
Meeting the exascale challenge

- Need for exascale
- Intel Exascience Lab

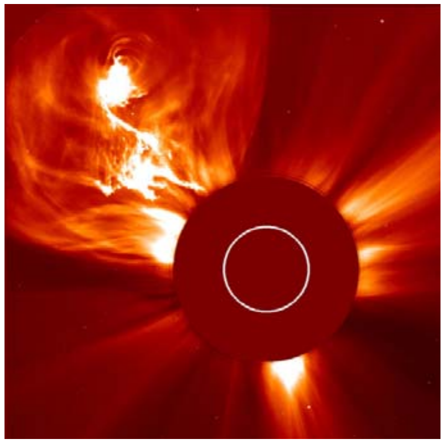
CHALLENGES IN MODELING SPACE WEATHER: MULTIPLE PHYSICS

KATHOLIEKE UNIVERSITEIT LEUVEN





TRACE September 2005



LASCO September 2002

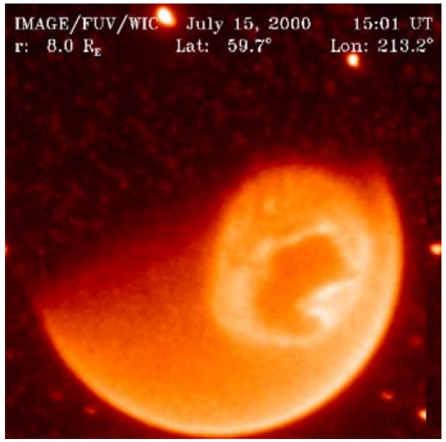
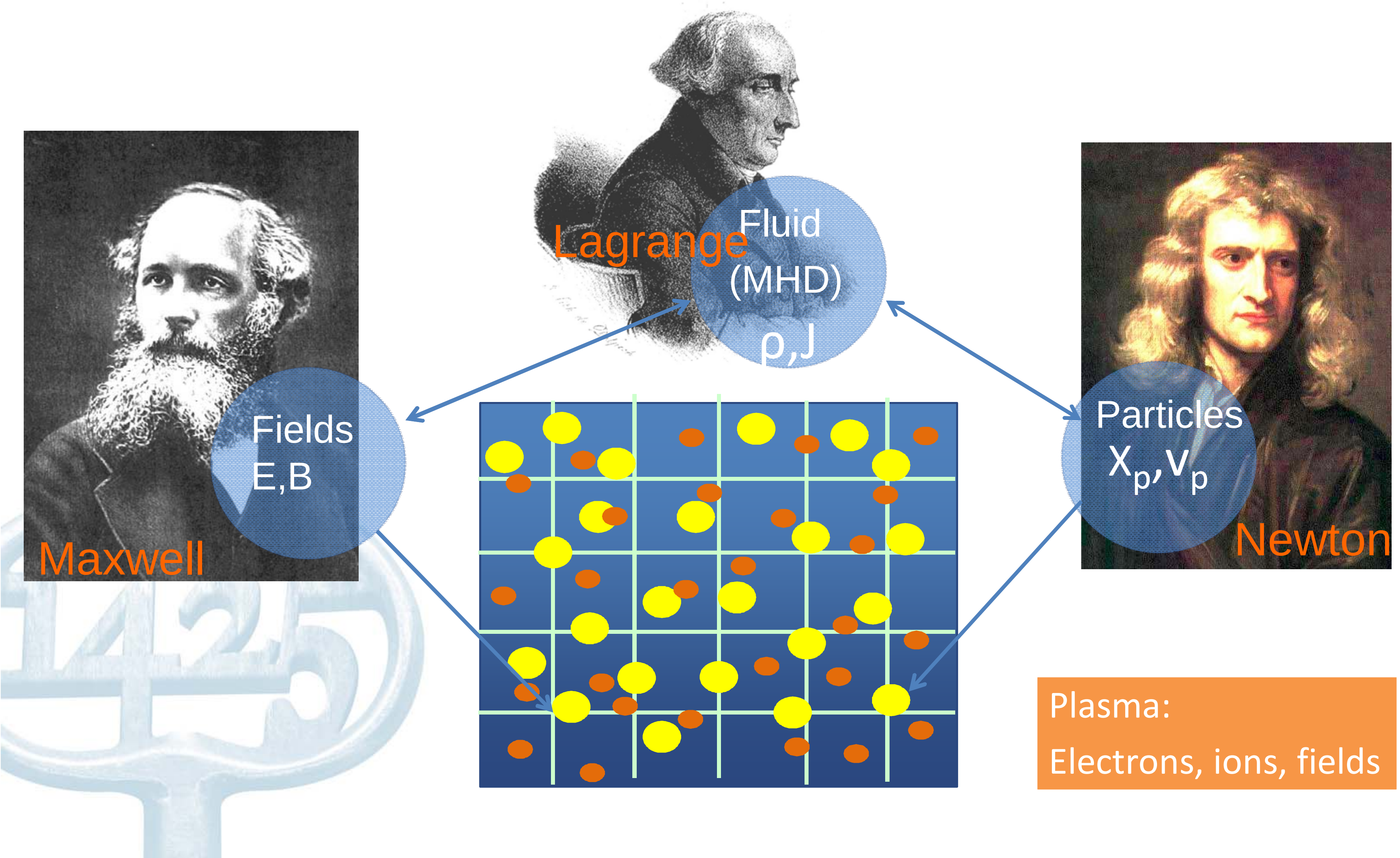


IMAGE July 2000

FLUID to KINETIC



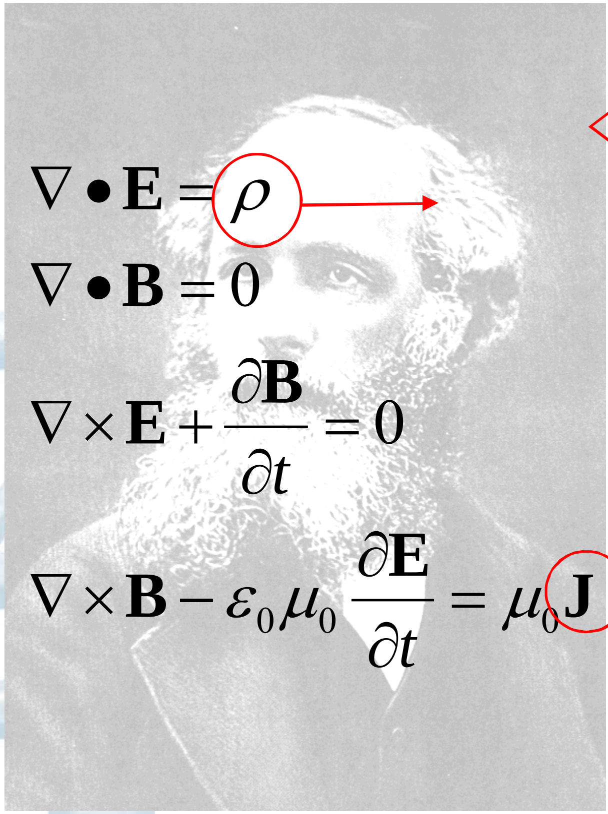
Plasma:

Electrons, ions, fields

Kinetic model: Fundamental Equations

- Boltzmann-Maxwell model

- Maxwell equations



$$\nabla \bullet \mathbf{E} = \rho$$

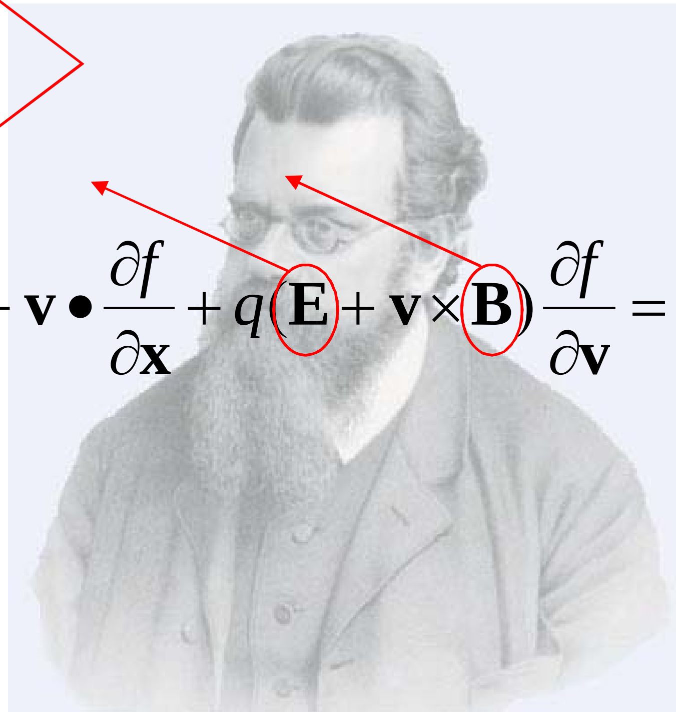
$$\nabla \bullet \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} = 0$$

$$\nabla \times \mathbf{B} - \varepsilon_0 \mu_0 \frac{\partial \mathbf{E}}{\partial t} = \mu_0 \mathbf{J}$$

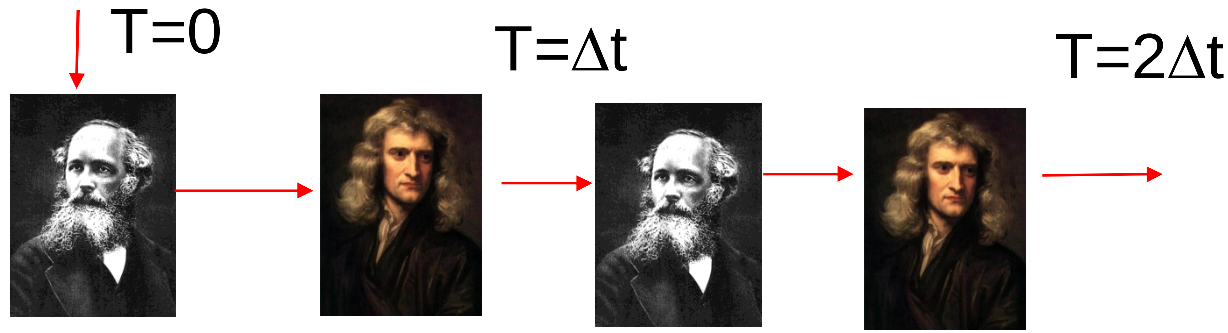
coupling

- Boltzmann equation



$$\frac{\partial f}{\partial t} + \mathbf{v} \bullet \frac{\partial f}{\partial \mathbf{x}} + q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \frac{\partial f}{\partial \mathbf{v}} = St(f)$$

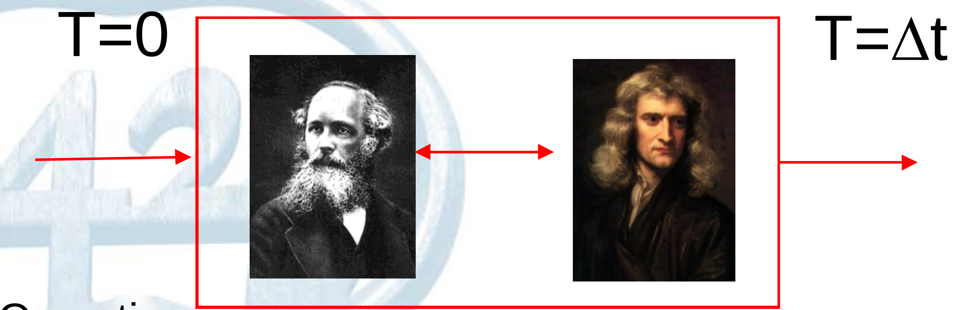
Explicit and implicit



EXPLICIT

Operations:

1. Solve Newton equations in previous electromagnetic fields
2. Solve Maxwell equations with previous particle positions

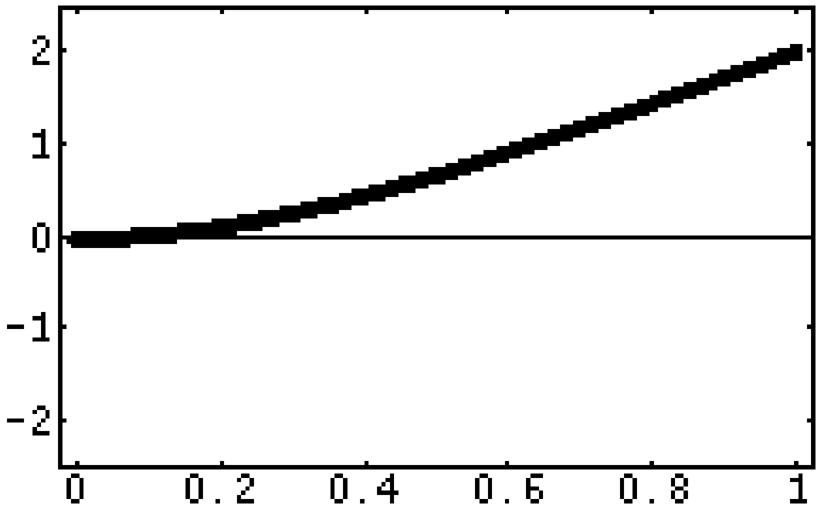


IMPLICIT

Operations:

Over each time step, iteratively solve the two coupled equations until convergence

Stability of a numerical scheme

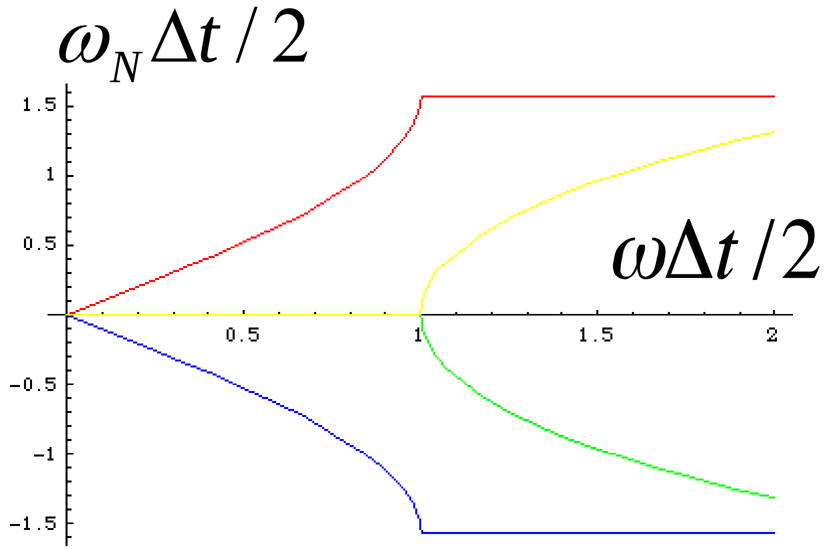


- Example: A cantilever (springboard)
- A perturbation makes it vibrate, but vibration amplitude does not grow in time

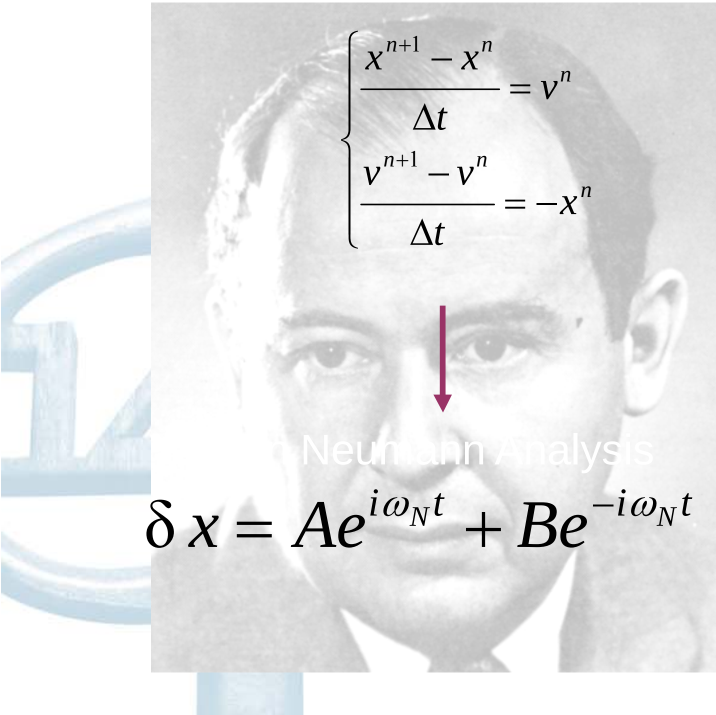
- **Stable system**: when perturbed its vibration amplitude does not grow
- **Unstable system**: when perturbed its vibration do grow

Stability of the explicit scheme: analogy with pendulum

$$\begin{cases} \dfrac{dx}{dt} = v \\ m\dfrac{dv}{dt} = -kx \end{cases} \longrightarrow \boxed{\begin{gathered} x = A\sin(\omega t) + B\cos(\omega t) \\ \omega = \sqrt{k/M} \end{gathered}}$$



$$\boxed{\omega\Delta t < 2}$$



$$\begin{cases} \dfrac{x^{n+1} - x^n}{\Delta t} = v^n \\ \dfrac{v^{n+1} - v^n}{\Delta t} = -x^n \end{cases}$$

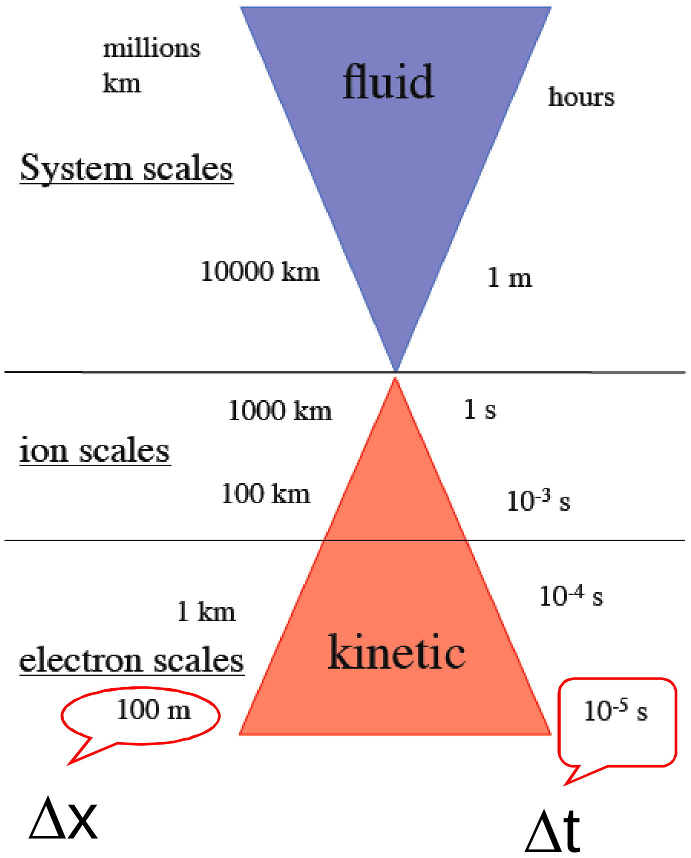
$$\delta x = Ae^{i\omega_N t} + Be^{-i\omega_N t} \longrightarrow \frac{\omega\Delta t}{2} = \sin\left(\frac{\omega_N \Delta t}{2}\right)$$

Limits of the explicit kinetic models

$$\omega_{fastest}\Delta t < 2$$

$$\Delta x < \lambda_{smallest}$$

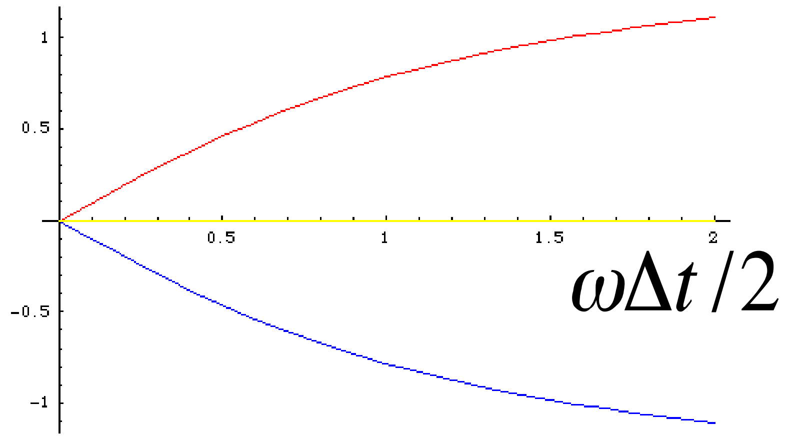
$$c\Delta t < \Delta x$$

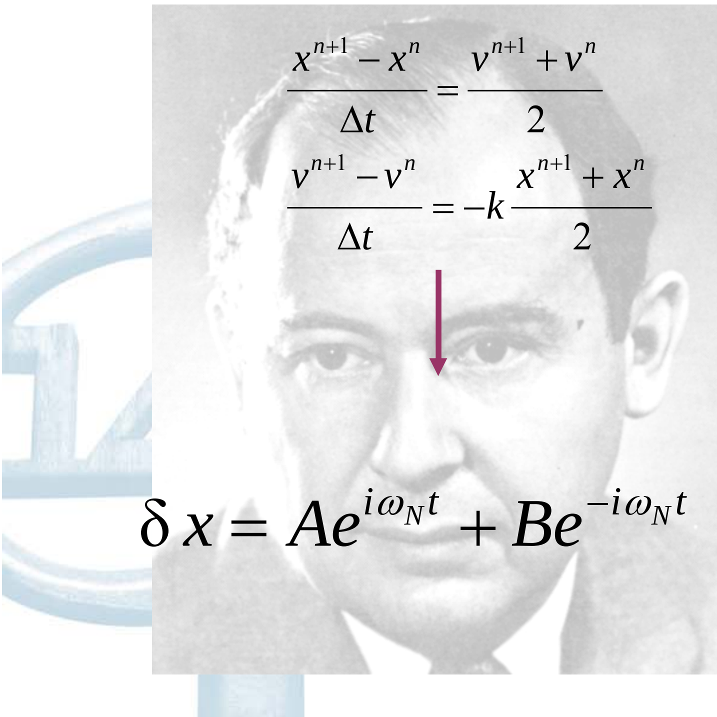


Stability of the implicit scheme

$$\boxed{\begin{gathered} x = A\sin(\omega t) + B\cos(\omega t) \\ \omega = \sqrt{k/M} \end{gathered}}$$

$$\frac{\omega \Delta t}{2} = \sin\left(\frac{\omega_N \Delta t}{2}\right)$$





$$\frac{x^{n+1} - x^n}{\Delta t} = \frac{v^{n+1} + v^n}{2}$$

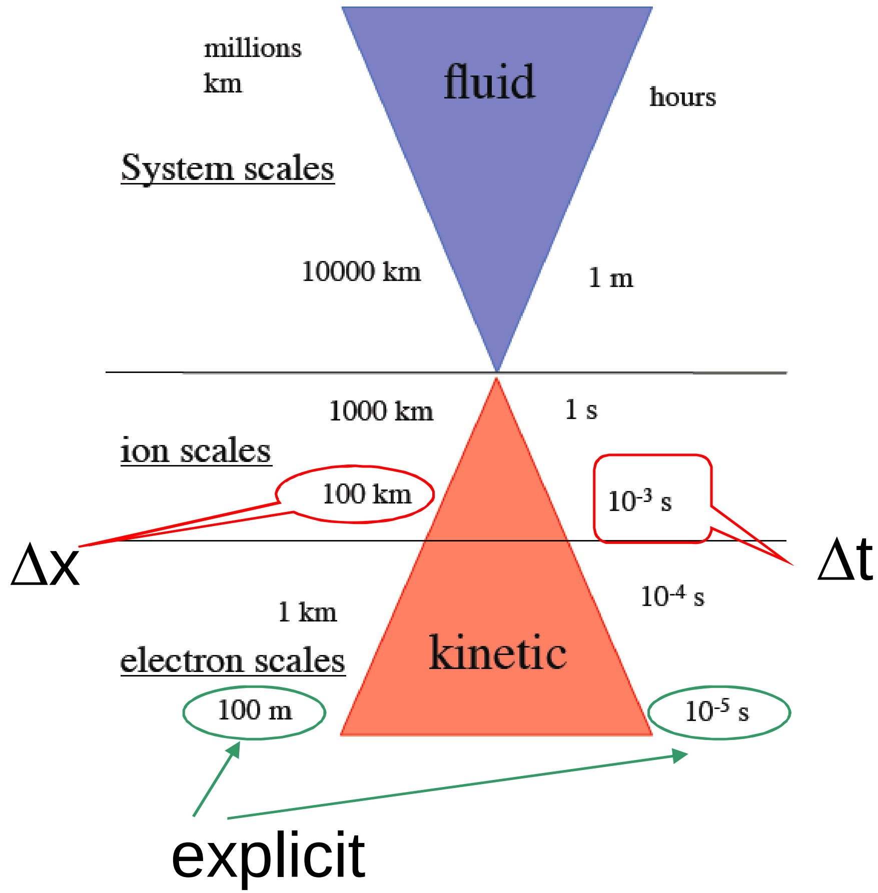
$$\frac{v^{n+1} - v^n}{\Delta t} = -k\frac{x^{n+1} + x^n}{2}$$

$$\delta x = Ae^{i\omega_N t} + Be^{-i\omega_N t}$$

$$\frac{\omega \Delta t}{2} = \tan\left(\frac{\omega_N \Delta t}{2}\right)$$

Summary of the complete Stability Analysis

$$\frac{\Delta t\,\omega_{fastest}}{\Delta x / \lambda_{smallest}} < 1$$



Numerical Stability Analysis

Explicit stability constraints

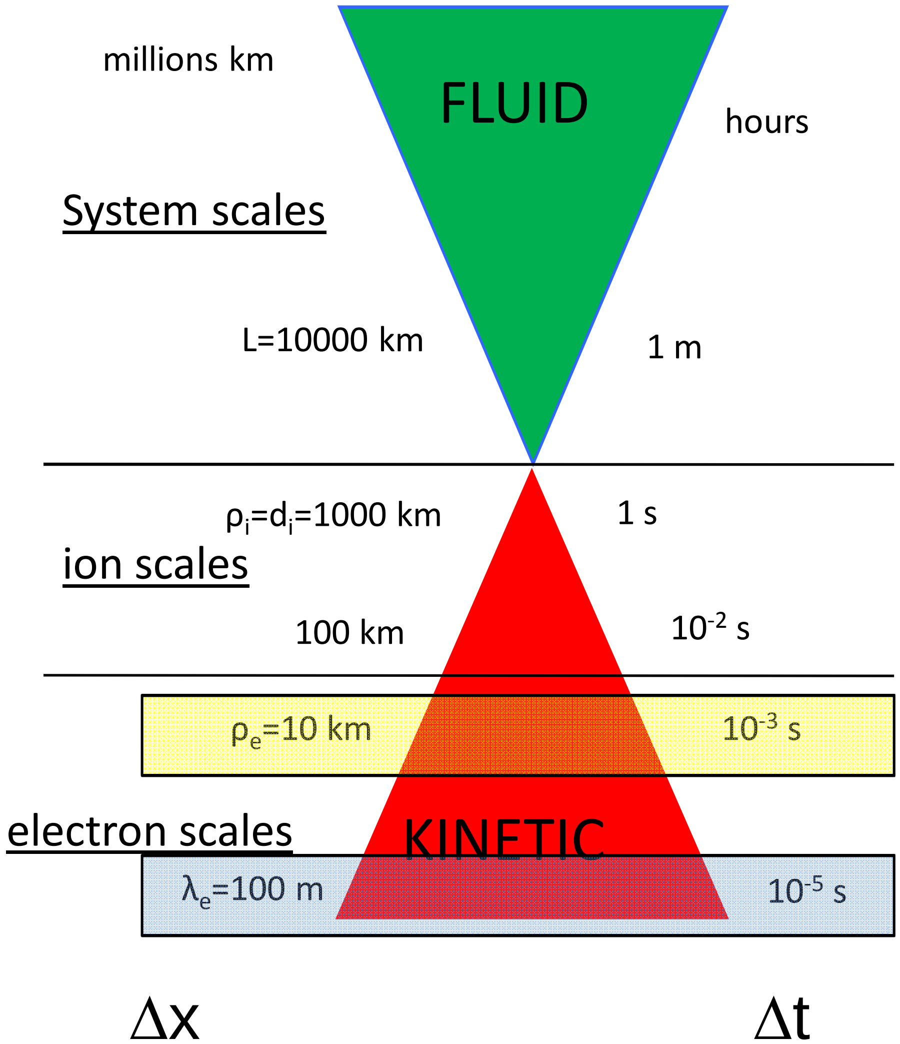
$$\omega_{fastest}\Delta t < 2$$

$$\Delta x < \lambda_{smallest}$$

$$c\Delta t < \Delta x$$

Implicit stability constraints

$$\frac{\Delta t\,\omega_{fastest}}{\Delta x / \lambda_{smallest}} < 1$$



Implicit formulation of Vlasov-Maxwell

- Maxwell equations: implicit second order formulation for the field E
- Newton equations: implicit form

$$\psi^{n+\theta} = \theta\psi^{n+1} + (1-\theta)\psi^{n}$$

- Solvers:
  - Coupled
  - Non-linear

Particle mover

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^{n} + \mathbf{v}_p^{n+1/2}\Delta t$$

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^{n+1} + \frac{q_s\Delta t}{m_s} \ldots {}^{+1/2}) + \mathbf{v}_p^{n+1/2} \times \ldots {}^{/2}))$$

Field Solver

$$\nabla \times \mathbf{E}^{n+\theta} + \frac{1}{c}\frac{\mathbf{B}^{n+1} - \mathbf{B}^{n}}{\Delta t} = 0$$

$$\nabla \times \mathbf{B}^{n+\theta} - \frac{1}{c}\frac{\mathbf{E}^{n+1} - \mathbf{E}^{n}}{\Delta t} = \frac{4\pi}{c} \ldots$$

$$\nabla \cdot \mathbf{E}^{n+\theta} = 4\pi \ldots$$

$$\nabla \cdot \mathbf{B}^{n} = \nabla \cdot \mathbf{B}^{n+1} = 0,$$

Example: Electrostatic Implicit Solver

Lapenta, Markidis, PoP, 18, 072101 (2011);
Markidis, Lapenta, JCP, 230, 7037 (2011)

Coupled Equations

$$\mathbf{u}_p^{n+1} = \mathbf{u}_p^n + q_s/m_s \Delta t (\bar{\mathbf{E}}_p + \bar{\mathbf{v}}_p \times \bar{\mathbf{B}}_p)$$

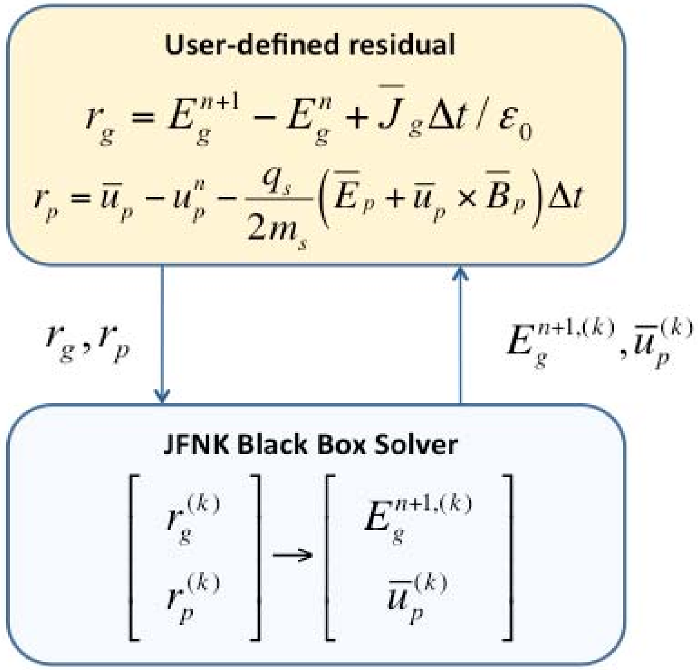
$$\mathbf{E}_g^{n+1} - \mathbf{E}_g^n = -\bar{\mathbf{J}}_g \Delta t/\epsilon_0$$

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \bar{\mathbf{v}}\Delta t$$

$$\bar{\mathbf{J}}_g = \sum_p q_p \bar{\mathbf{v}}_p W(\mathbf{x}_g - \bar{\mathbf{x}}_p)/V_g$$

$$\mathbf{f}(\mathbf{w}) = 0$$

$$\mathbf{f}(\mathbf{w}^{(k)}) + \frac{\mathbf{f}(\mathbf{w}^{(k)} + \epsilon\delta\mathbf{w}) - \mathbf{f}(\mathbf{w}^{(k)})}{\epsilon} = 0$$



Implicit Computational Cycle

Extremely simple program in MATLAB

http://swiff.eu/codes/matlab-ec/

```
% Energy Conserving PIC code

for it = 1:NT
    % start computational cycle
    xkrylov = [v0; E0];
    % Newton Krylov GMRes solver
    [sol, it_hist, ierr] = nsolgm(xkrylov, 'residueEC', tol);
    v_average = sol(1:N);
    % update particle positions and velocities
    v0 = 2*v_average-v0;
    x0 = x0 + v_average * DT;
    % check if particle are out of the periodic boundaries
    out = (x0 < 0); x0(out) = x0(out) + L;
    out = (x0 >= L); x0(out) = x0(out)-L;
    % new electric field
    E0 = sol((N + 1): (N + NG));
    % calculate the total energy
    Etot = 0.5 * abs(Q) * sum(v0.^2) + 0.5 * sum(E0.^2)*dx;
    % save the total energy
    histEnergy = [histEnergy Etot];
    % end computational cycle
end
```

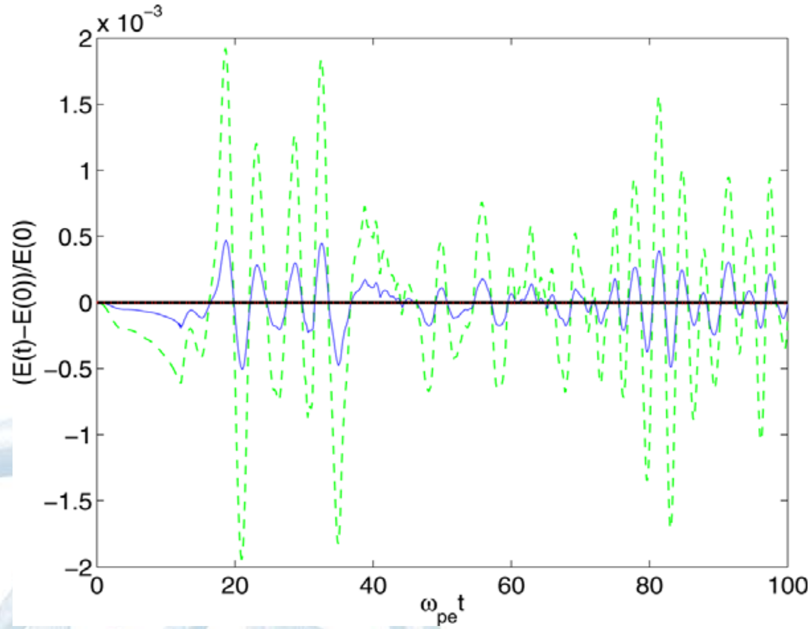
```
%residual calculation for the EC PIC
function res = residueEC(xkrylov)

% calculate the x at n + 1/2 time level
x_average = x0 + xkrylov(1:N) * DT/2;
% check if particle are out of the periodic boundaries
out = (x_average < 0); x_average(out) = x_average(out) + L;
out = (x_average >= L); x_average(out) = x_average(out)-L;
% interpolation
p = 1:N; p = [p p]; g1 = floor(x_average/dx-.5) + 1; g = [g1; g1 + 1];
fraz1 = 1-abs(x_average(1:N)/dx-g1 + .5); fraz = [(fraz1); 1-fraz1];
out = (g < 1); g(out) = g(out) + NG;
out = (g>NG); g(out) = g(out)-NG;
mat = sparse(p, g, fraz, N, NG);

res = zeros(N + NG, 1);
% average velocity residual
res(1:N, 1) = xkrylov(1:N)-v0-.5* mat * QM * (E0 + xkrylov((N + 1):(N + NG))) * DT;

% calculate the average J
fraz = [(fraz1). * xkrylov(1:N); (1-fraz1). * xkrylov(1:N)];
mat = sparse(p, g, fraz, N, NG);
J = full((Q/dx) * sum(mat))';

% electric field residual
res((N + 1):(N+NG)) = xkrylov((N + 1):(N + NG))-E0 + J * DT;
```

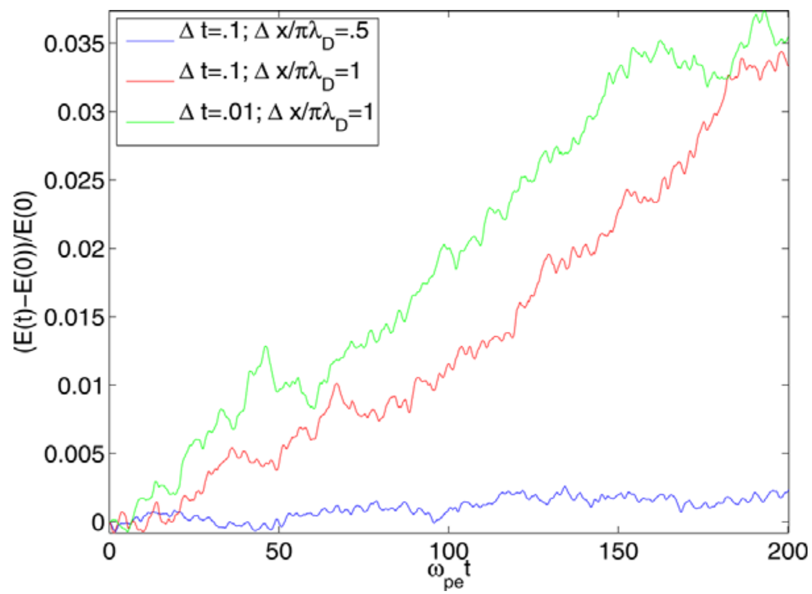
Full listing on Markidis, Lapenta, JCP, 230, 7037 (2011)

Exact Energy Conservation

Lapenta, Markidis, PoP, 18, 072101 (2011)

2-stream instability	Thermal plasma





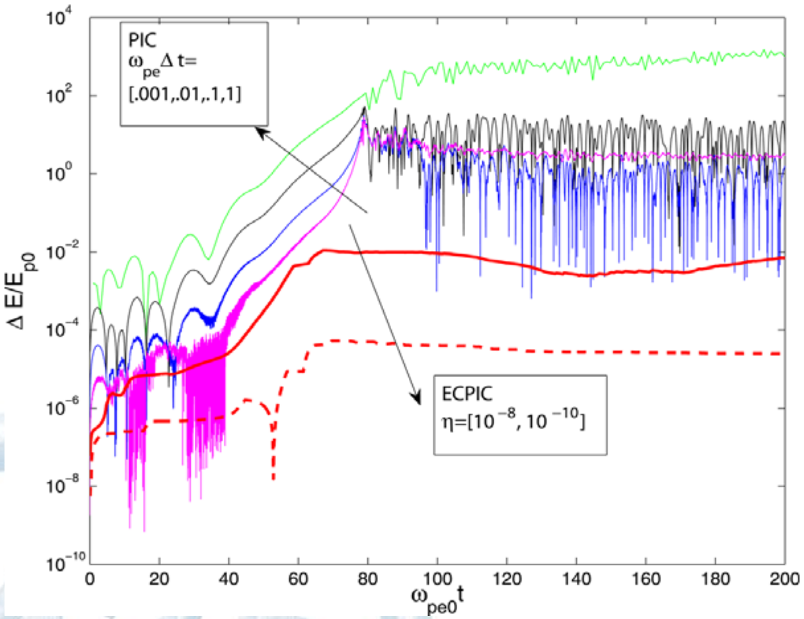
ECPIC=1.32 10^{-8}

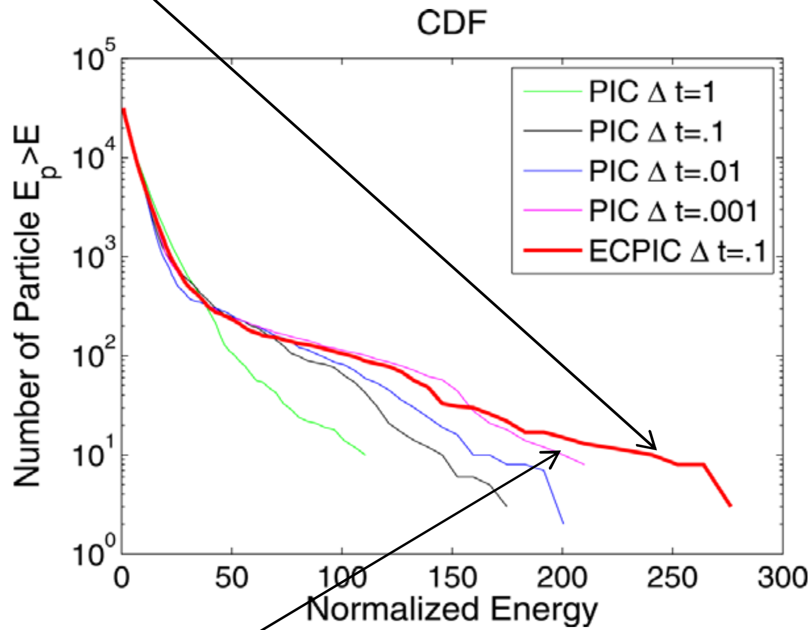
ECPIC=1.75 10^{-8}

Effects of error in energy conservation

Lapenta, Markidis, PoP, 18, 072101 (2011)

Implicit run: 498 s



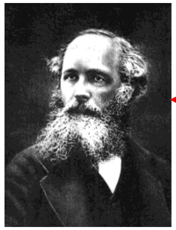


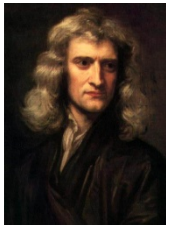
Explicit run: 3164 s

IMPLICIT MOMENT METHOD

J.U. Brackbill et al., **JCP**, *46*, 271, 1982; G. Lapenta, et al, **Phys. Plasmas**, *13*, 055904, 2006.

T=0





T=Δt

IMPLICIT

IMPLICIT MOMENT

$$\nabla \bullet \mathbf{E} = \rho$$
$$\nabla \bullet \mathbf{B} = 0$$
$$\nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} = 0$$
$$\nabla \times \mathbf{B} - \varepsilon_0 \mu_0 \frac{\partial \mathbf{E}}{\partial t} = \mu_0 \mathbf{J}$$

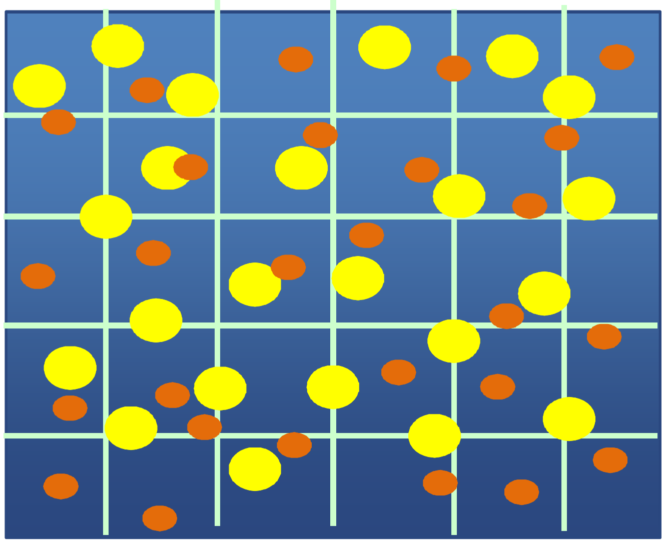
$$\rho = M_0$$
$$J = M_1$$

Moment equations from Taylor expansion to make solver feasible

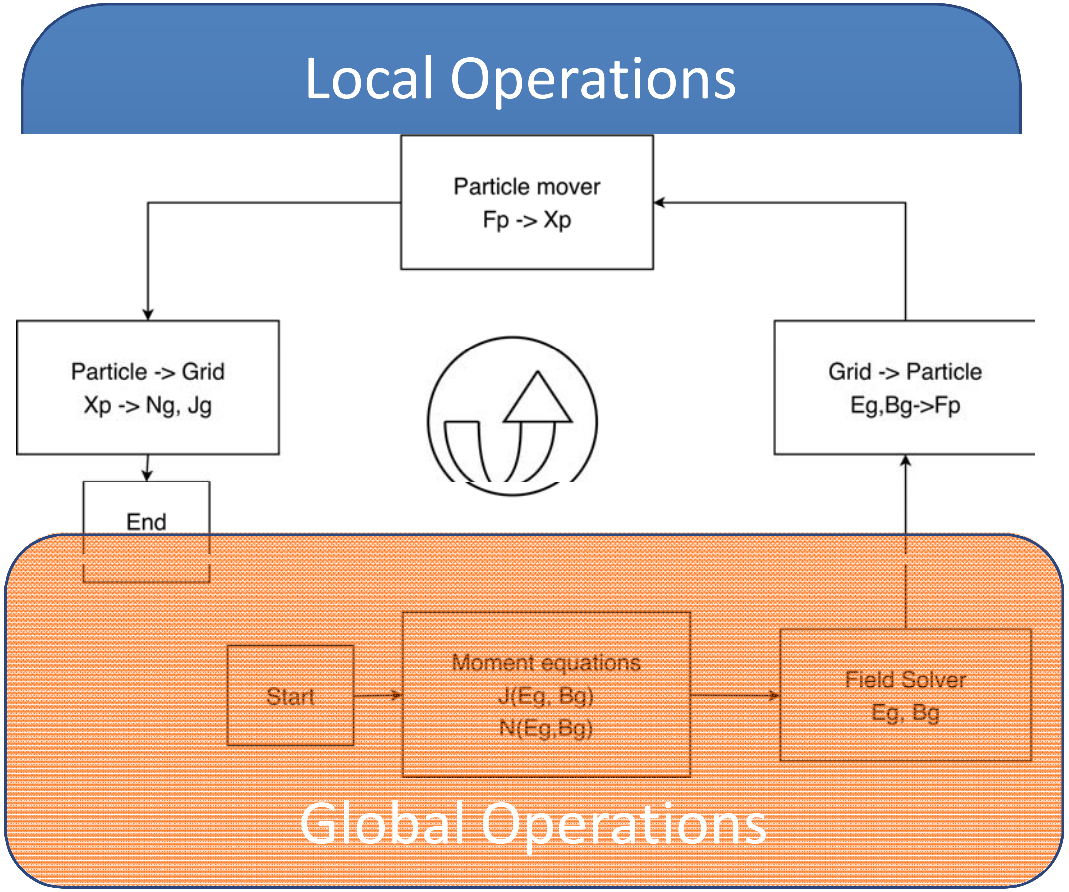
$$M_n = \sum_p q_p v_p^n W(x - x_p)$$

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}$$
$$m\frac{d\mathbf{v}}{dt} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B})$$

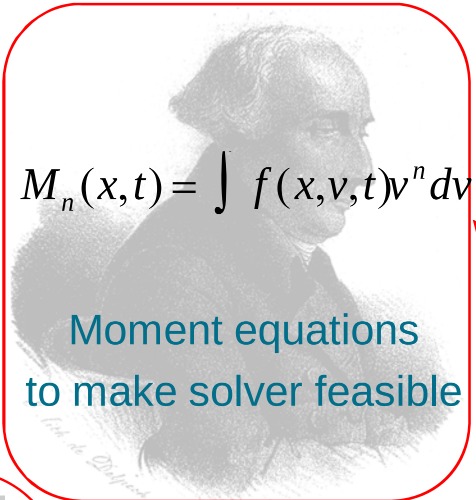
Operations needed



- Hundreds of particles per cell
- Data locality

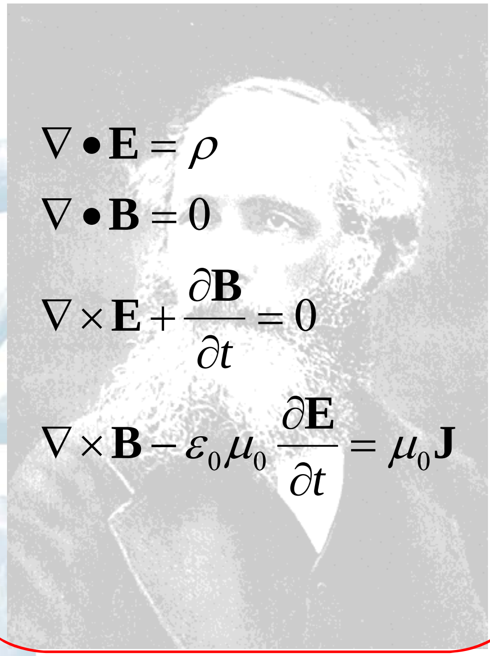


Implicit moment method



$$M_n(x,t) = \int f(x,v,t) v^n \, dv$$

Moment equations
to make solver feasible

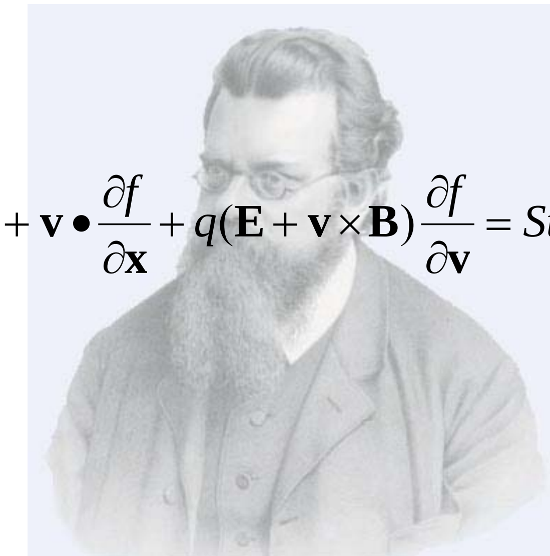


$$\nabla \bullet \mathbf{E} = \rho$$

$$\nabla \bullet \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} = 0$$

$$\nabla \times \mathbf{B} - \varepsilon_0 \mu_0 \frac{\partial \mathbf{E}}{\partial t} = \mu_0 \mathbf{J}$$



$$\frac{\partial f}{\partial t} + \mathbf{v} \bullet \frac{\partial f}{\partial \mathbf{x}} + q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \frac{\partial f}{\partial \mathbf{v}} = St(f)$$

Outline

Meeting the multiscale challenge

- PIC Method
- Implicit PIC
- Moment method

Application to space weather

- Local simulations
- Global simulations

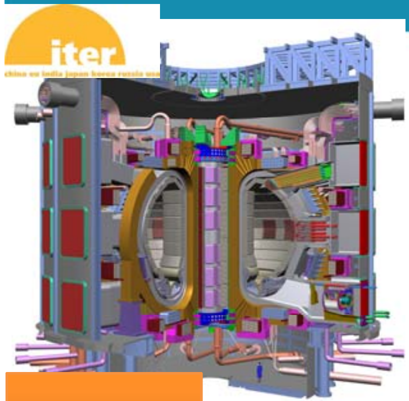
Meeting the multiphysics challenge

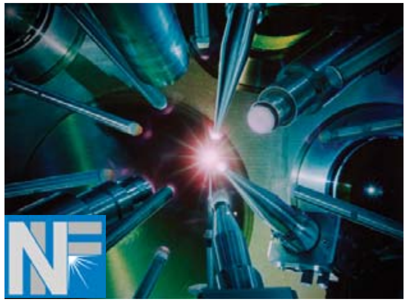
- Multi-level simulation
- AMR

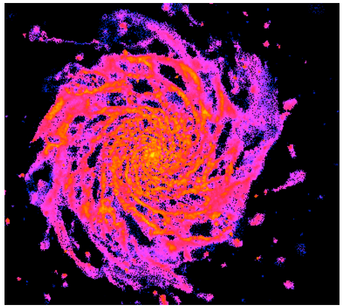
Meeting the exascale challenge

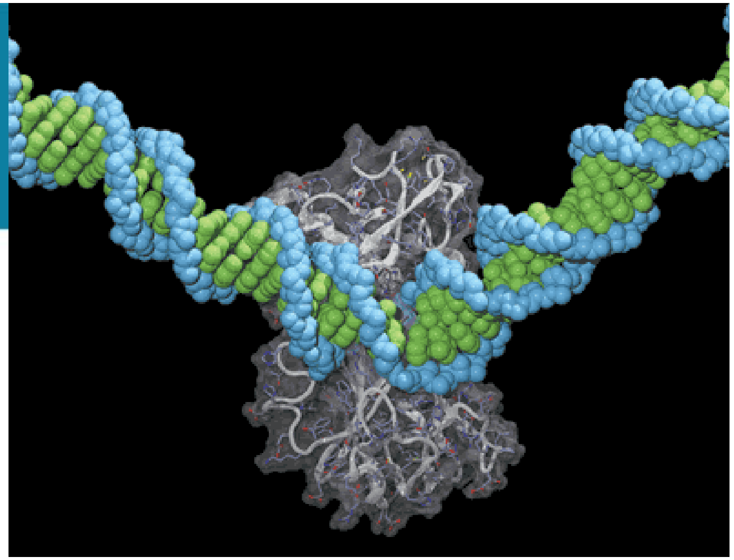
- Need for exascale
- Intel Exascience Lab

Wide Applicability

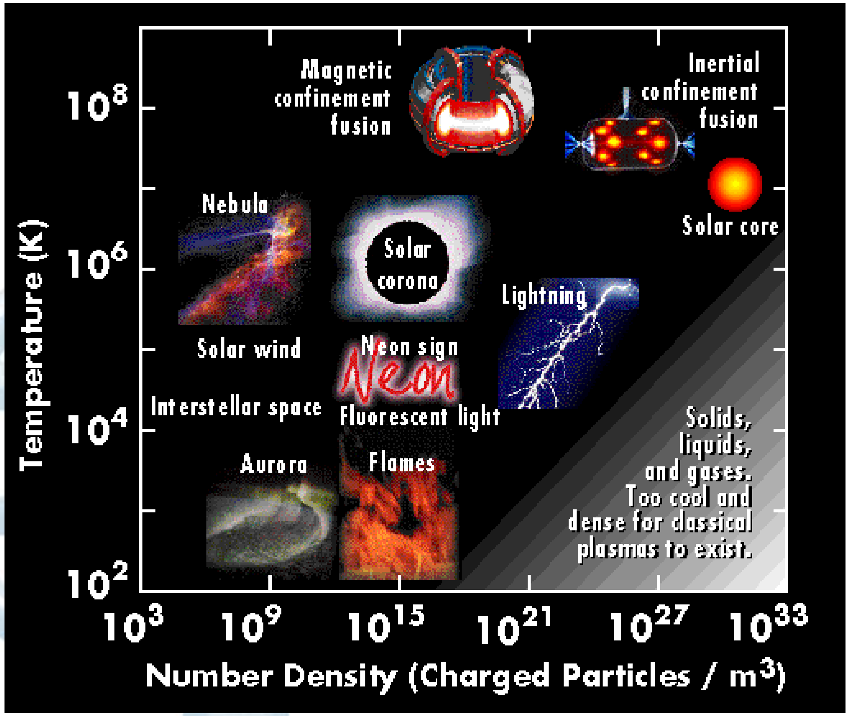


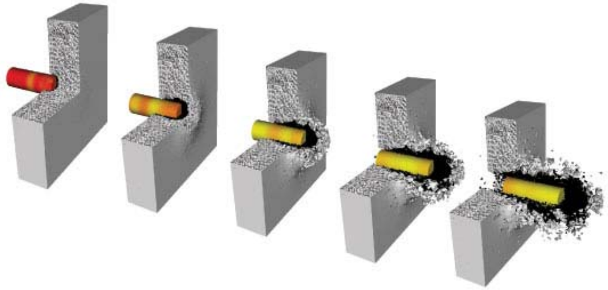




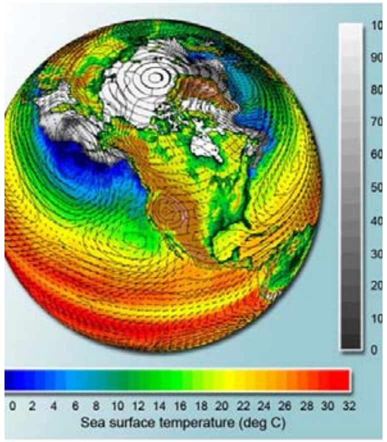


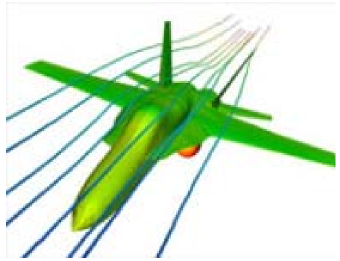
Plasma





Particles





Moments

Beyond the State of the ART

Brackbill, Forslund (LANL)

Lapenta, Brackbill (LANL)

Markidis, Lapenta (Leuven)

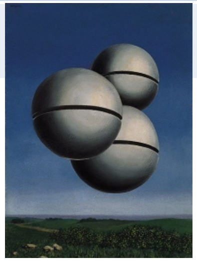
New work of the SWIFF And Exascience Lab

Venus	Celeste	Parsek/iPIC	iPic-AMR
80's	90's - now	2005-now	NEW

Venere
(Sandro Botticelli)

Starry Night
(Vincent Van Gogh)

Several Circles
(Vasily Kandinsky)

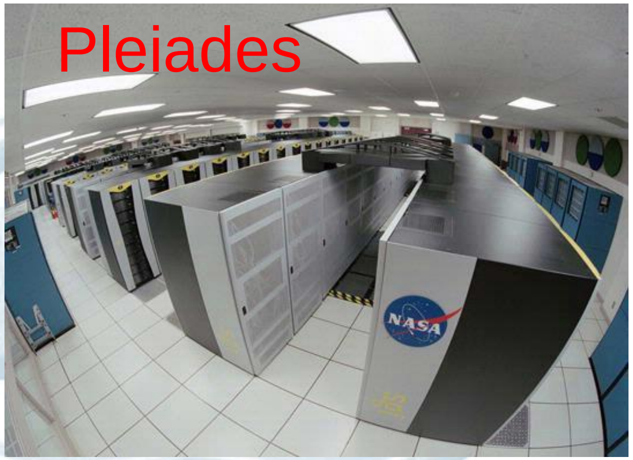


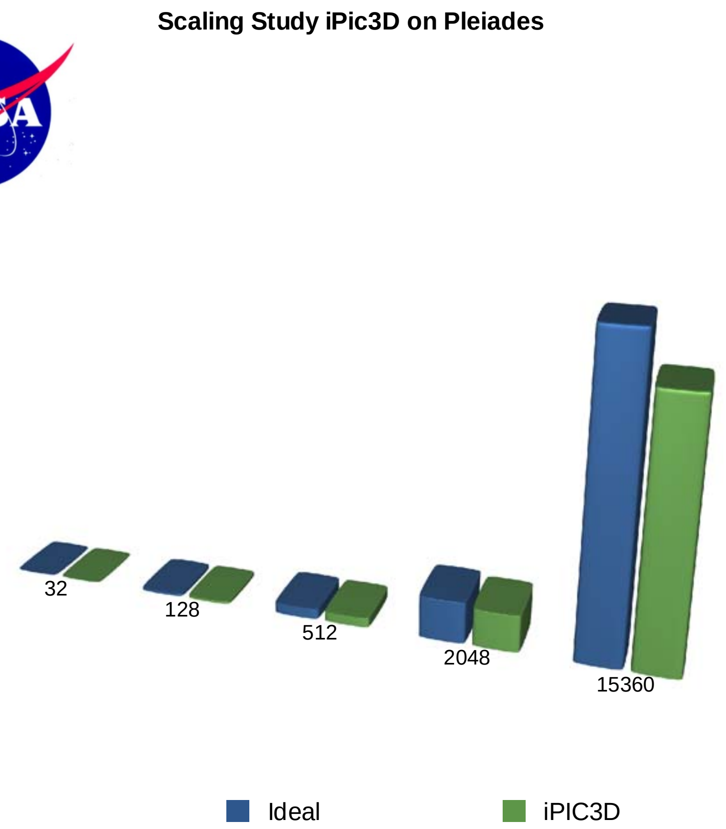
The Voice of Space
(René Magritte)

http://code.google.com/p/celeste/

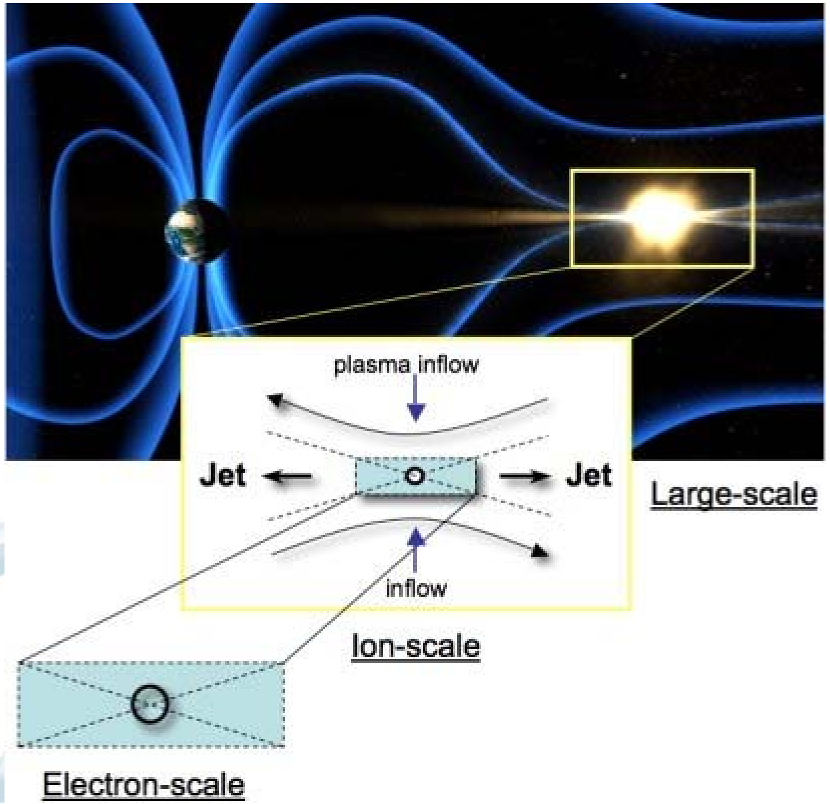
http://code.google.com/p/ipic3d/

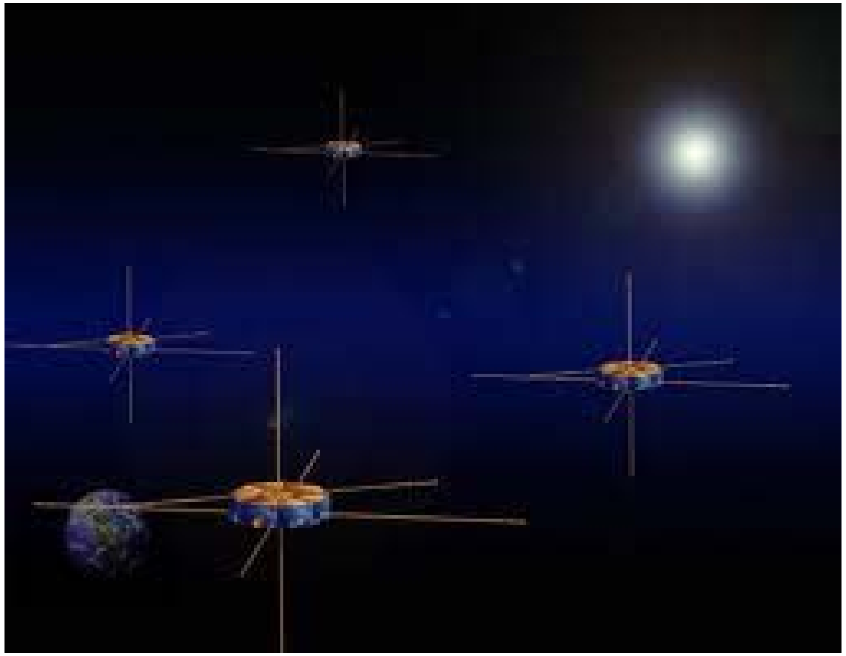
Performances of iPIC3D



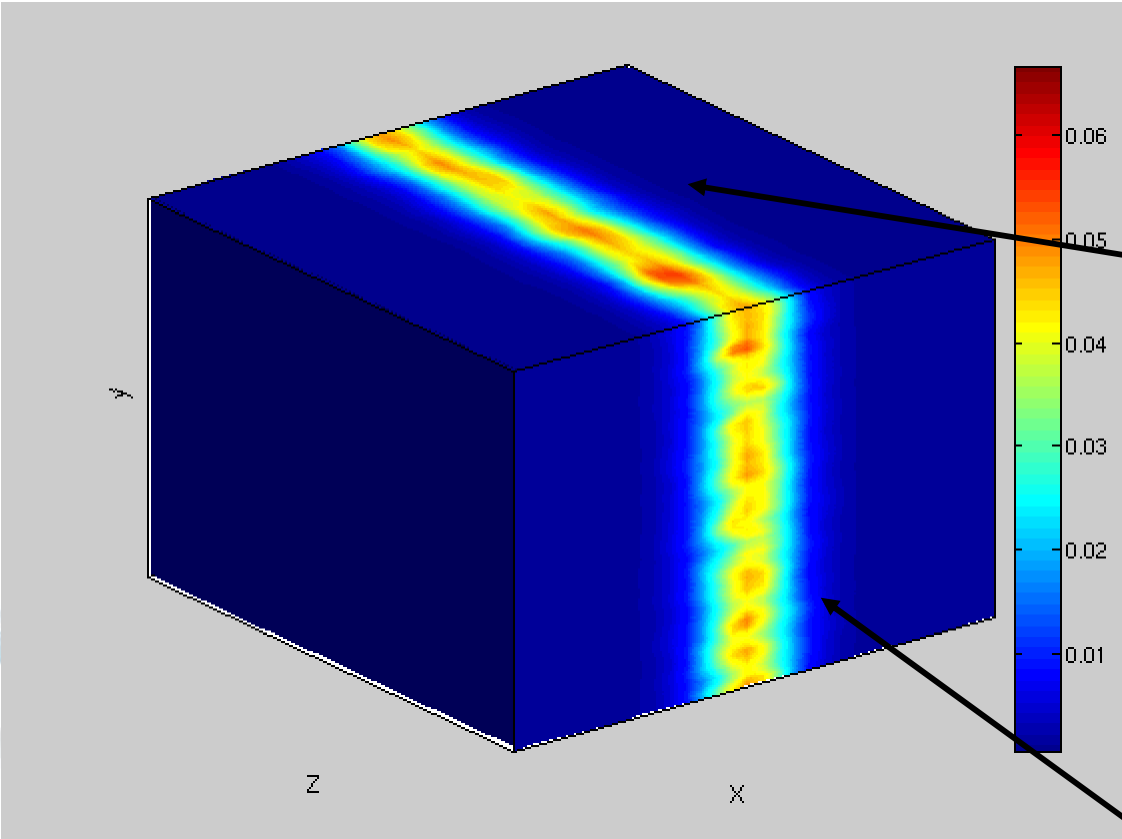


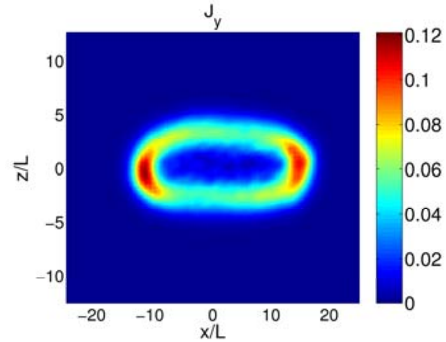
An example: kinetic simulation of a 3D current layer

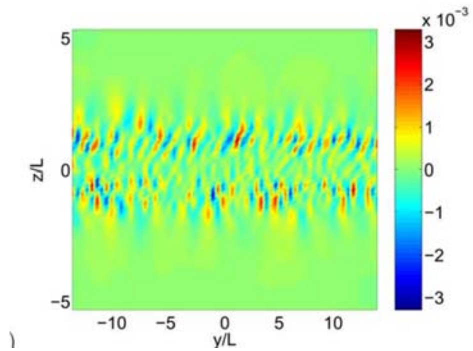




3D micro-macro coupling:
a typical space weather problem

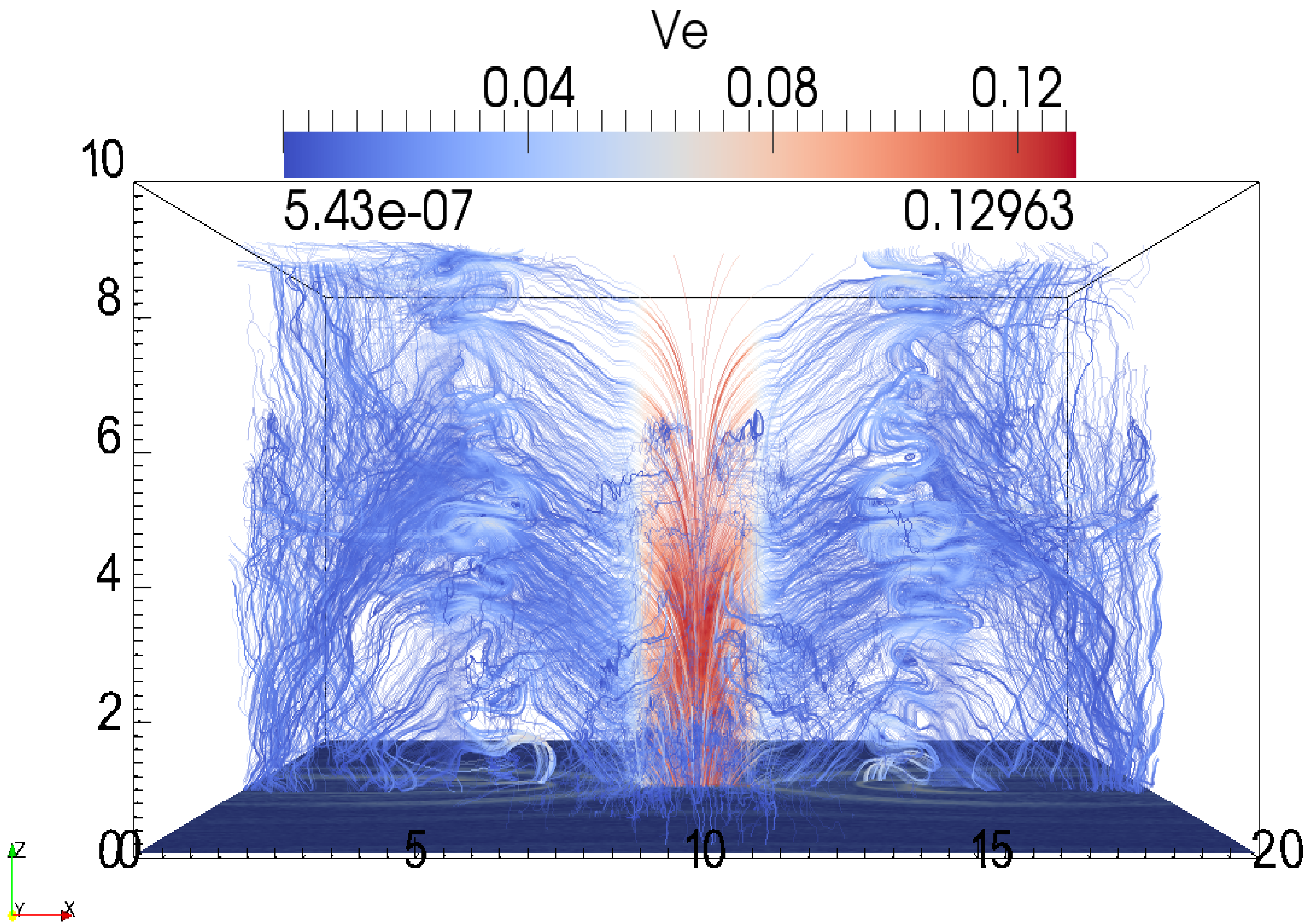






small/large scale coupling captured

The 3D Electron Flow



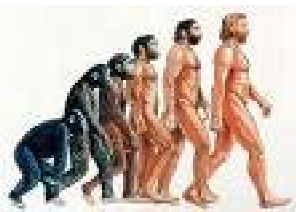
Timing considerations for 3D fully kinetic simulations - Implicit vs Explicit

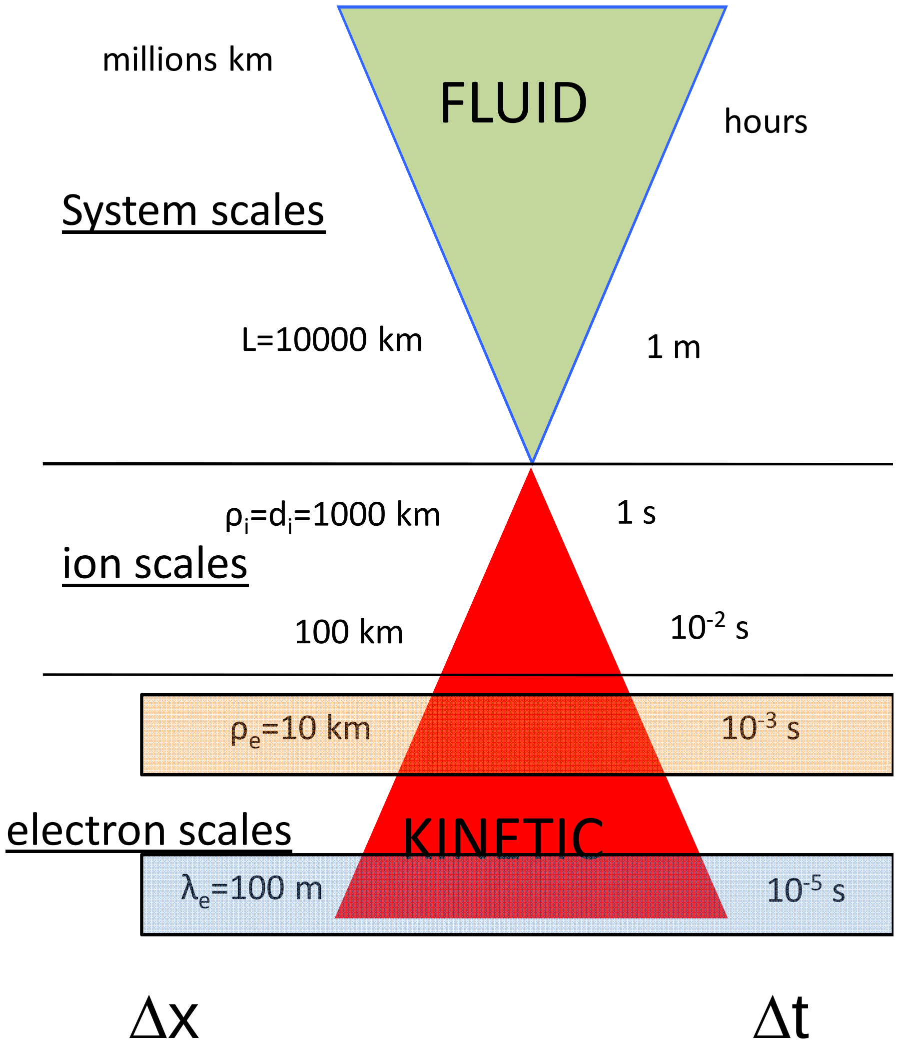
	Explicit	Implicit	Gain
Dx	λ_{De}=100 m	d_e=10 Km	100
Dy	λ_{De}=100 m	d_e=10 Km	100
Dz	λ_{De}=100 m	d_e=10 Km	100
Dt	$\omega_{pe}\Delta t$=0.1 or 10^{-5}s	$\omega_{pe}\Delta t$=100 or 10^{-3}s	1000
Tot			10^9

An implicit run that takes **1 day** would take:

2,800,000 year

n an





Outline

Meeting the multiscale challenge

- PIC Method
- Implicit PIC
- Moment method

Application to space weather

- Local simulations
- Global simulations

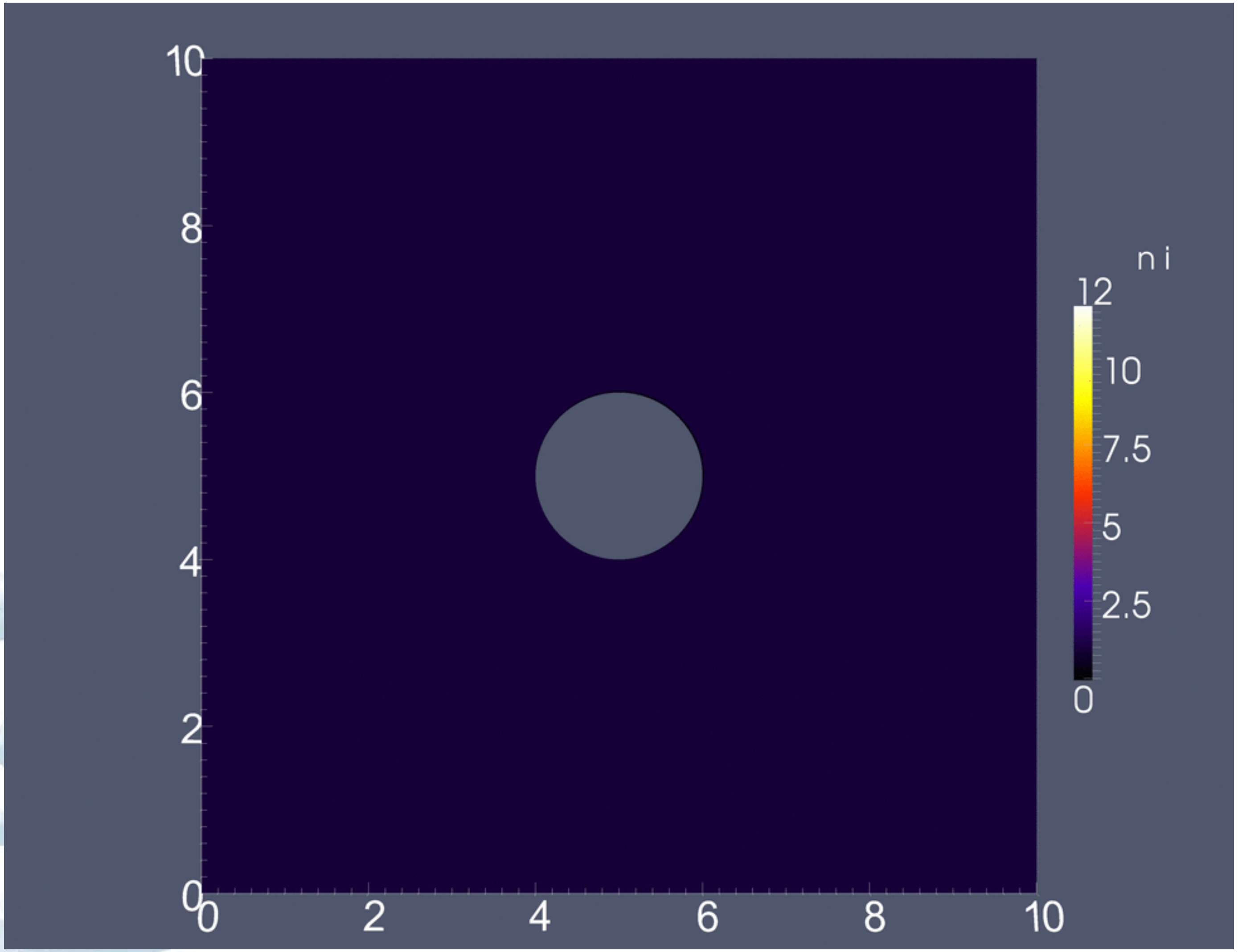
Meeting the multiphysics challenge

- Multi-level simulation
- AMR

Meeting the exascale challenge

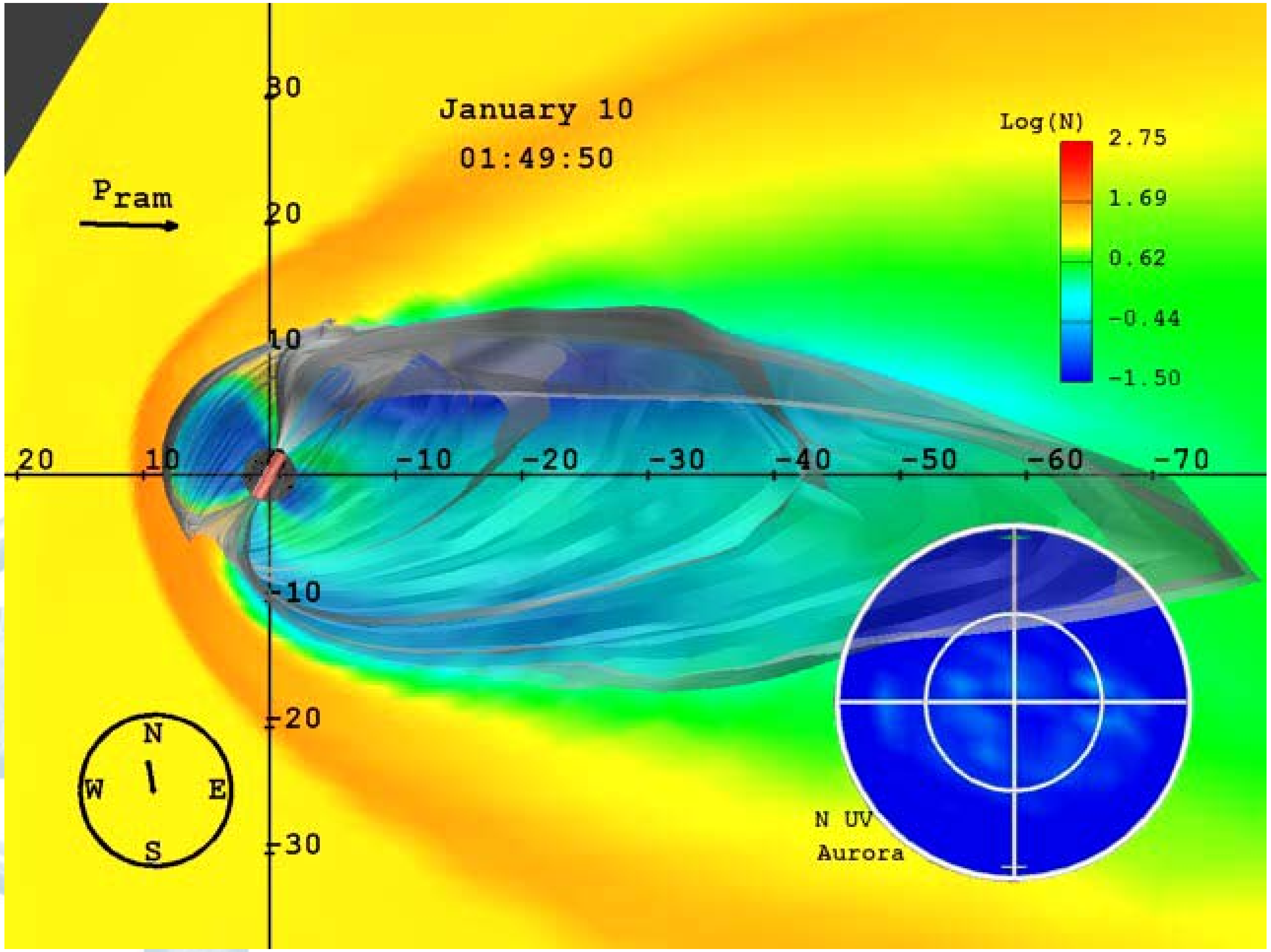
- Need for exascale
- Intel Exascience Lab

Mercury Magnetosphere
iPic3D



Reference case: Earth Environment

State of the art now - CCMC



Typical needs

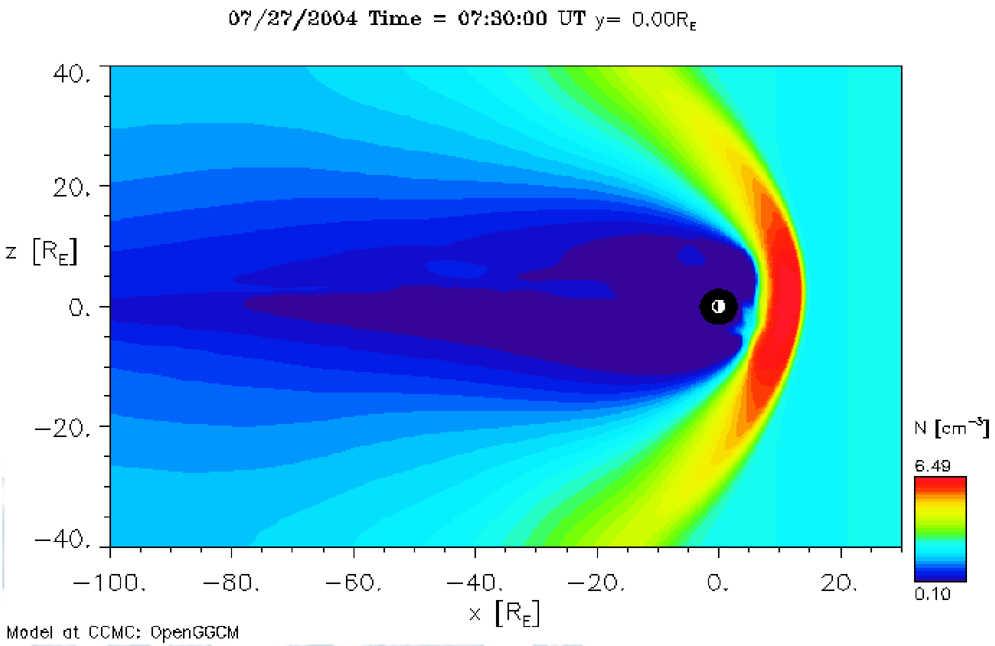
- Box:

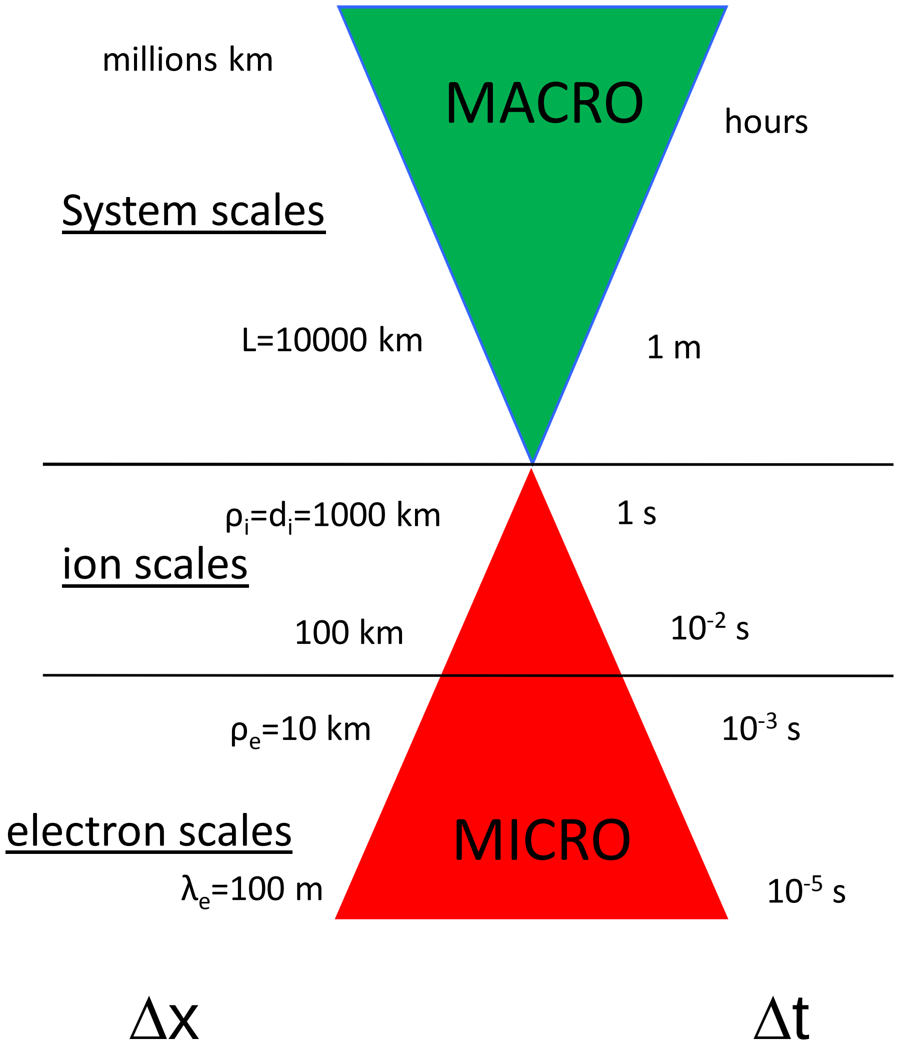
100 R_E x 100 R_E x 100 R_E

- Max load per processor:
  - 16x16x16 cells
  - 144 + 144 particles per cell(electron populations)
  - 144 +144 particles per cell (ion populations)

- Coupled with heliospheric models and observations

Scales to be Resolved





State of the Art

KATHOLIEKE UNIVERSITEIT LEUVEN

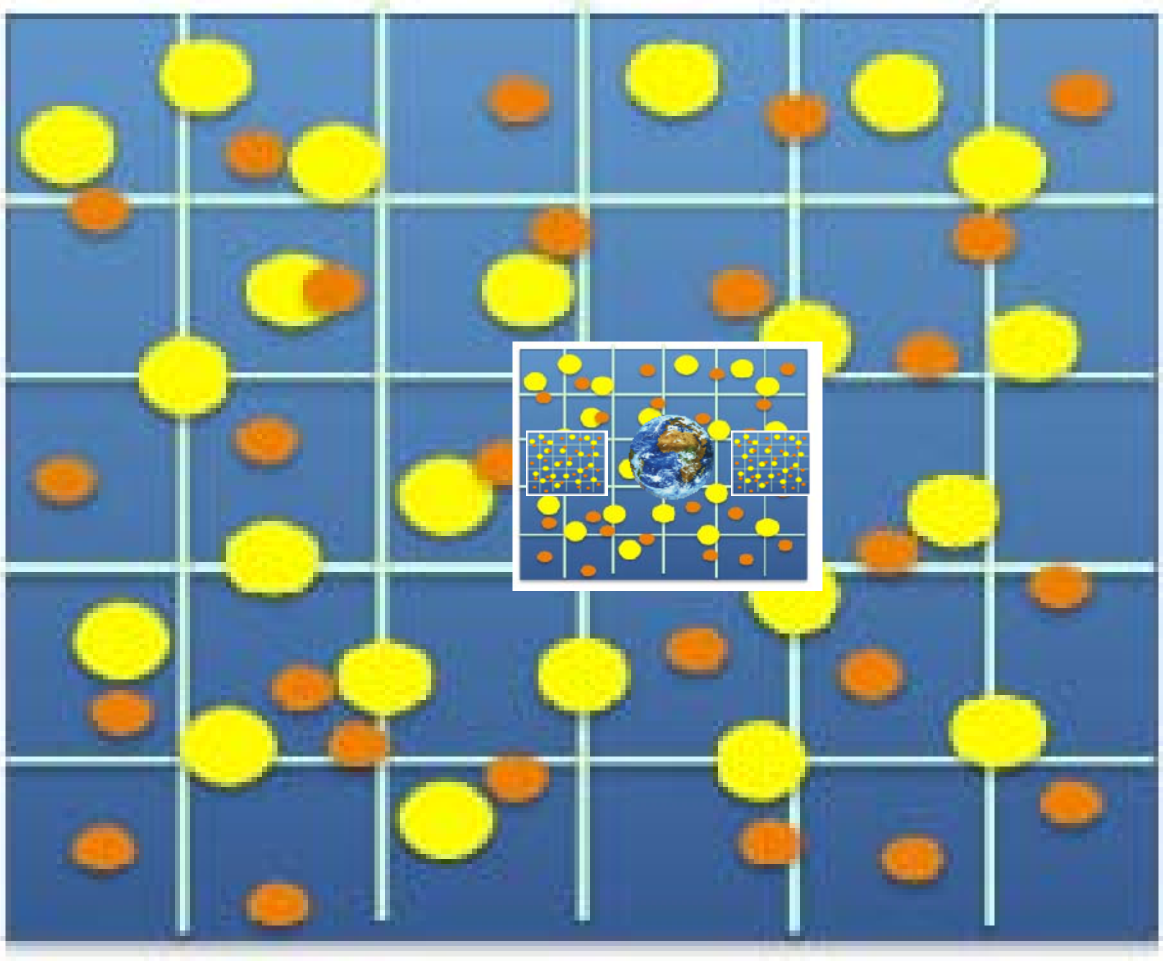
Explicit Formulation

- Resolution needed: electrostatic processes at electron scales: 100m (Debye length)
- Cells per dimension: 6,353,000
- Total processors needed: 6.2600e+16
- 63 million billions

Moment Implicit Formulation

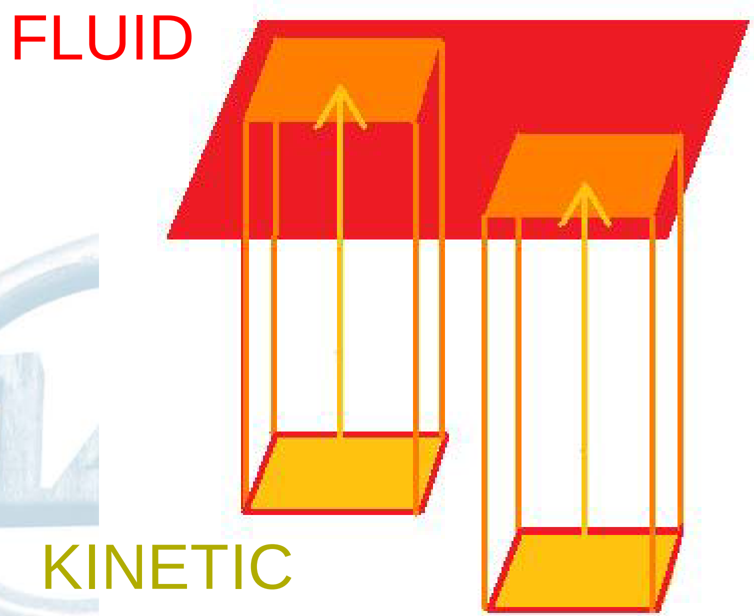
- Resolution needed: electromagnetic processes at electron scales: 10Km (inertial length)
- Cells per dimension: 63,530
- Total processors needed: 6.2600e+10
- 63 billions

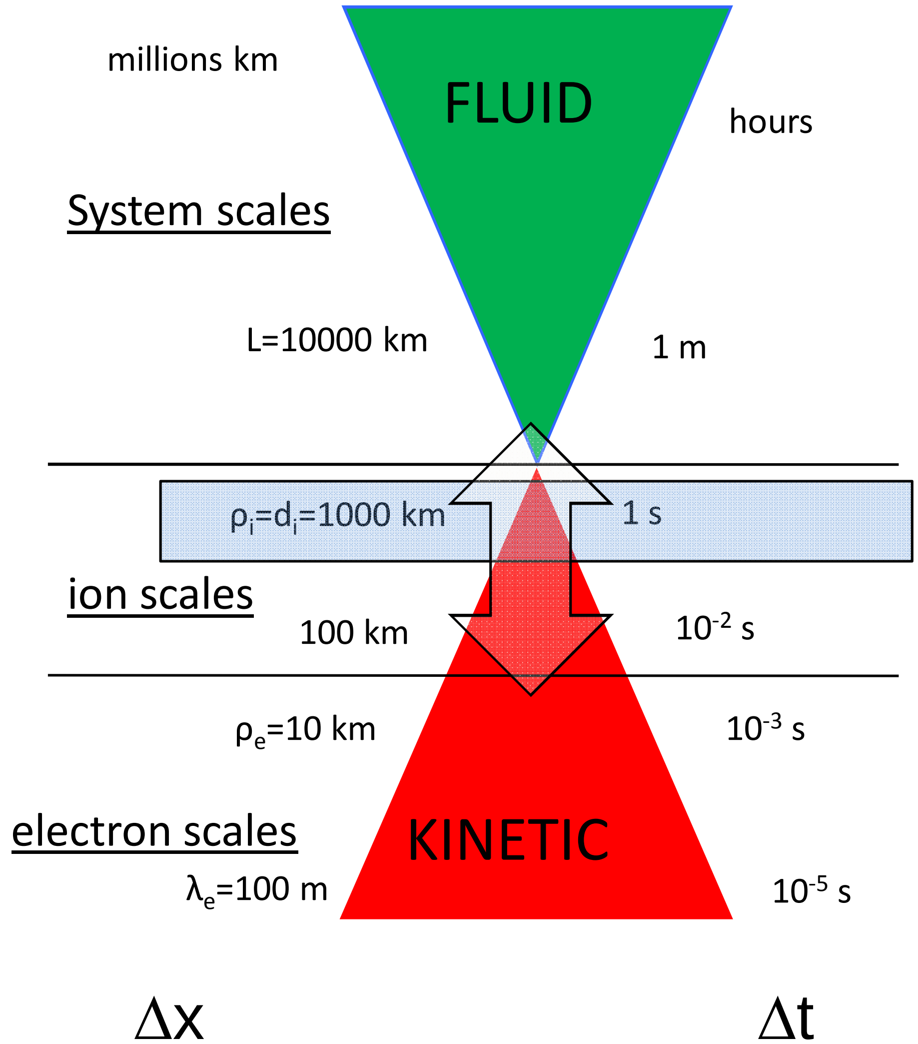
Adaptive Multiphysics approach



Multilevel and Multidomain

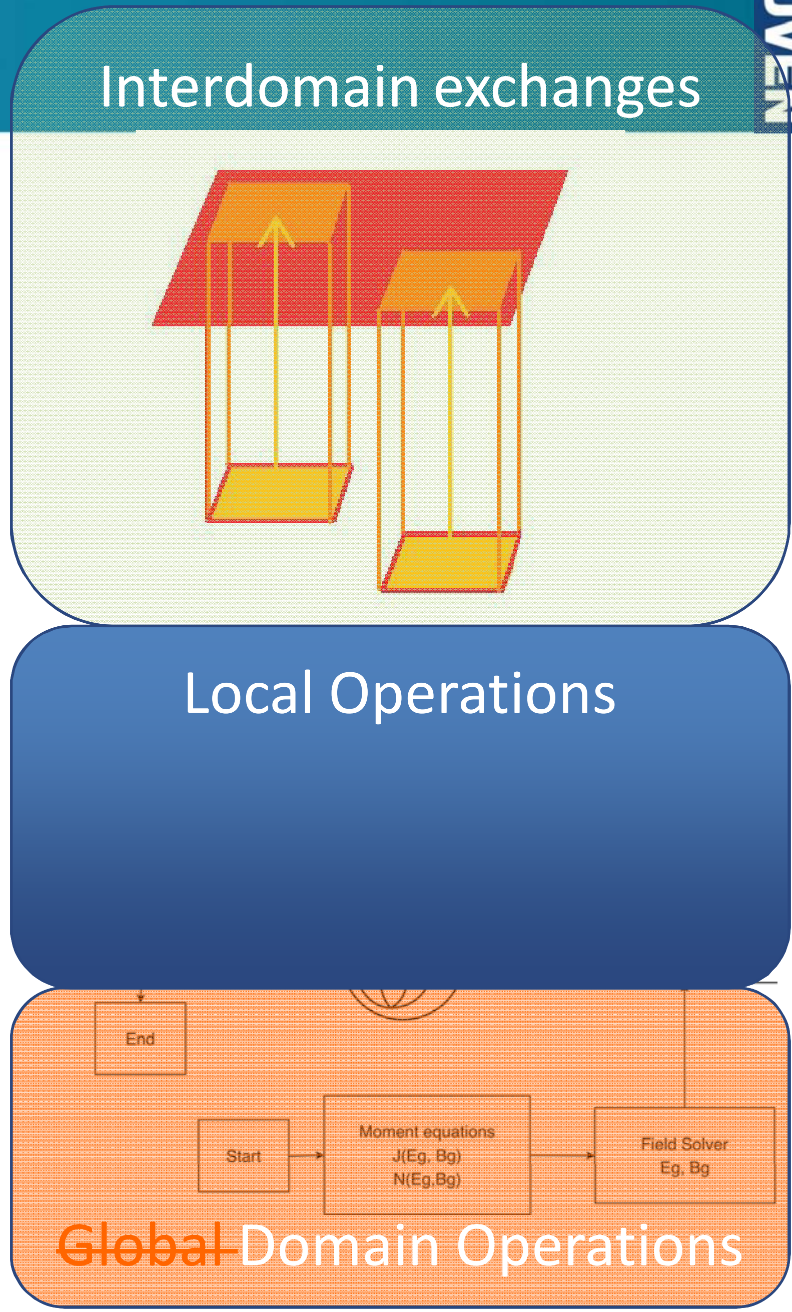
Each domain operates at its



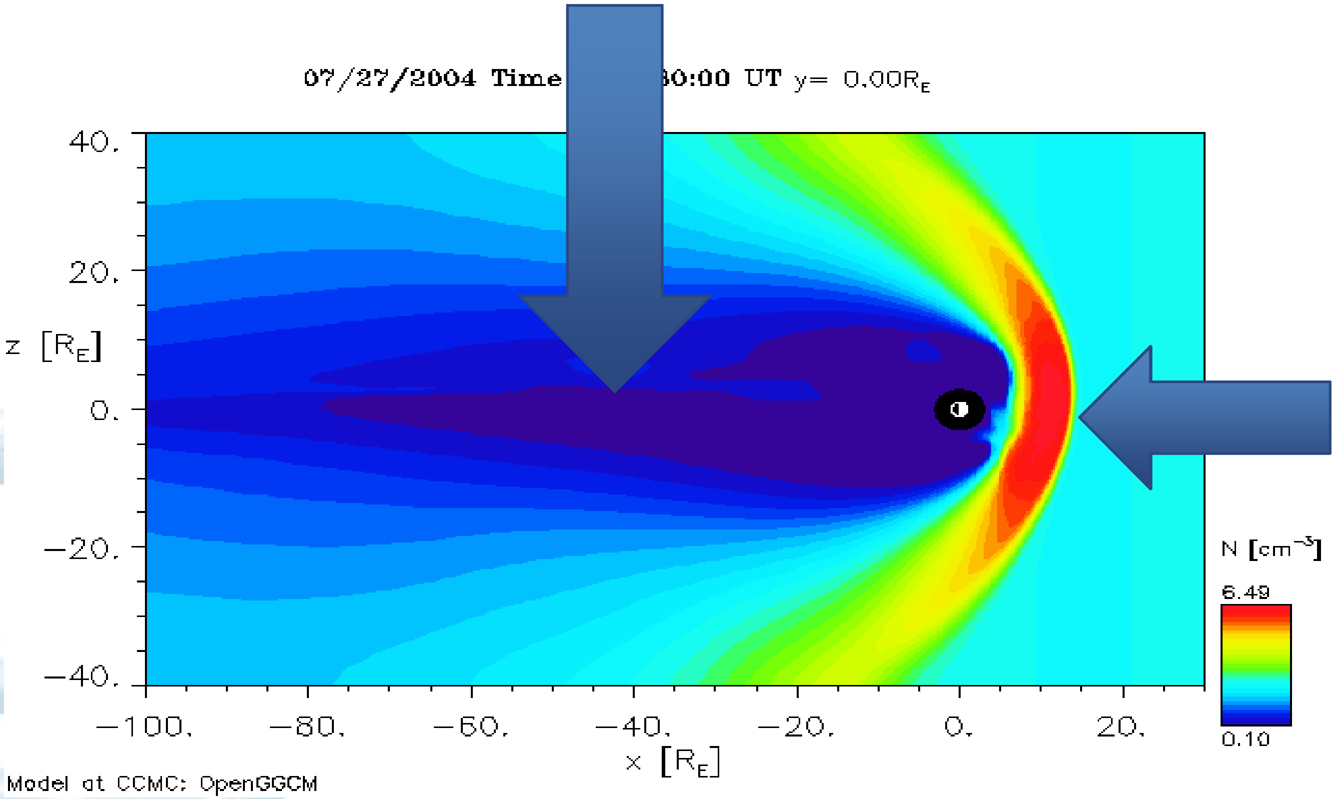


Overlapping Multidomain

- Each domain is an exact replica of the others, but scaled.
- The operations are identical for the same physics module
- Algorithm designed to minimize exchange of information
- No global operation, all solvers operate on each domain
- 1D and 2D versions implemented on massively parallel computers



Resolution needed only in small areas



Multidomain approach

Coarse Level

- Resolution needed: electromagnetic processes at ion scales: 1000Km
- Cells per dimension: 635
- Total processors needed: 6.2600e+04
- 62 thousand – petascale

Finer levels

- Resolution needed: electromagnetic processes at electron scales: 10Km
- Needed only on a thin crust on small surfaces.
- Estimated processors: 1.9812e+06
- Total processors needed:

  2 millions - exascale needed but sufficient

Outline

Meeting the multiscale challenge

- PIC Method
- Implicit PIC
- Moment method

Application to space weather

- Local simulations
- Global simulations

Meeting the multiphysics challenge

- Multi-level simulation
- AMR

Meeting the exascale challenge

- Need for exascale
- Intel Exascience Lab

Support at KU Leuven

KATHOLIEKE UNIVERSITEIT LEUVEN

- First ever EC funded project on space weather
- Involving 16 centers in 13 countries
- Focus on space weather and Earth impact
- Observation (some simulation)

Soteria

- Multiphysics modelling of space science applied to space weather
- Focus on simulation and theory
- To run till 2014
- Involving 7 centers in 5 countries

Swiff

- Being negotiated.
- Scheduled to start in march 2012
- Continuation of Soteria with emphasis on space exploration instead of Earth impact.
- Possibility of interaction with Boulder Lunar Science.

eHeroes

DEEP

Intel-based hybrid architectures

ExaScience Lab
Intel Labs Europe
EXASCALE COMPUTING

Co-Design of space simulation on exascale

SWIFF: Space Weather Integrated Modelling Framework

Space Weather Integrated Forecasting Framework

Coordinator: Giovanni Lapenta

Katholieke Universiteit Leuven

Collaborative Project

FP7- Space

- Create a mathematical-physical framework to integrate multiple-physics (fluid with kinetic)
- Focus on method and software development, rather than reuse of existing codes
- Physics-based forecasting
- Focus on coupling small-large scales
- Based on implicit methods and AMR

Science Lead	Participant organisation name	Country
Coordinator: G. Lapenta	Katholieke Universiteit Leuven	Belgium
V. Pierrard	Belgian Institute for Space Aeronomy	Belgium
F. Califano	Università di Pisa	Italy
A. Nordlund	Københavns Universitet	Denmark
A. Bemporad	Astronomical Observatory Turin - Istituto Nazionale di Astrofisica	Italy
P. Travnicek	Astronomical Institute, Academy of Sciences of the Czech Republic	Czech Republic
C. Parnell	University of St Andrews	UK

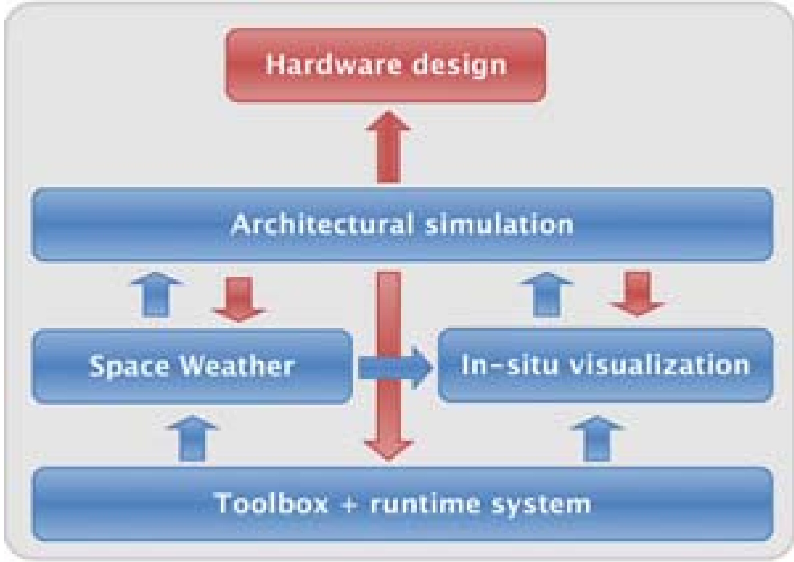
Space Weather and High Performance Computing

KATHOLIEKE UNIVERSITEIT LEUVEN

ExaScience Lab
Intel Labs Europe
EXASCALE COMPUTING

ExaScience Lab to Develop Space Weather Prediction as Driver for Intel's Future Exascale Supercomputers

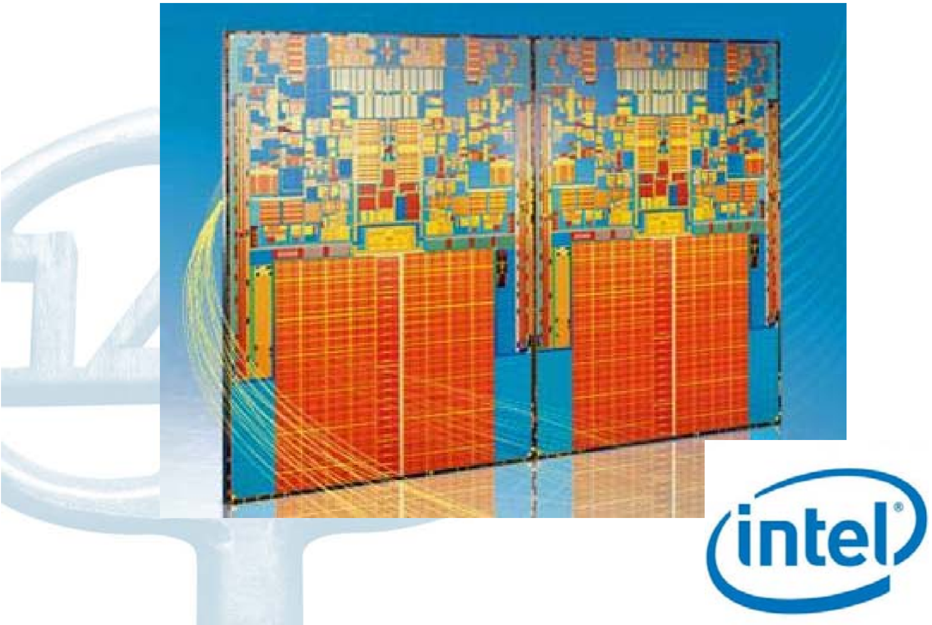
- **Intel**
- **Imec**
- **Five Flemish Universities**



exascience.com

KATHOLIEKE UNIVERSITEIT LEUVEN · UNIVERSITEIT GENT · Vrije Universiteit Brussel · Universiteit Antwerpen · universiteit hasselt · Vlaamse overheid · imec · intel

Space Weather and High Performance Computing



Dynamical Exascale Entry Platform

Coordinator: Thomas Lippert ,

Forschungszentrum Juelich GmbH

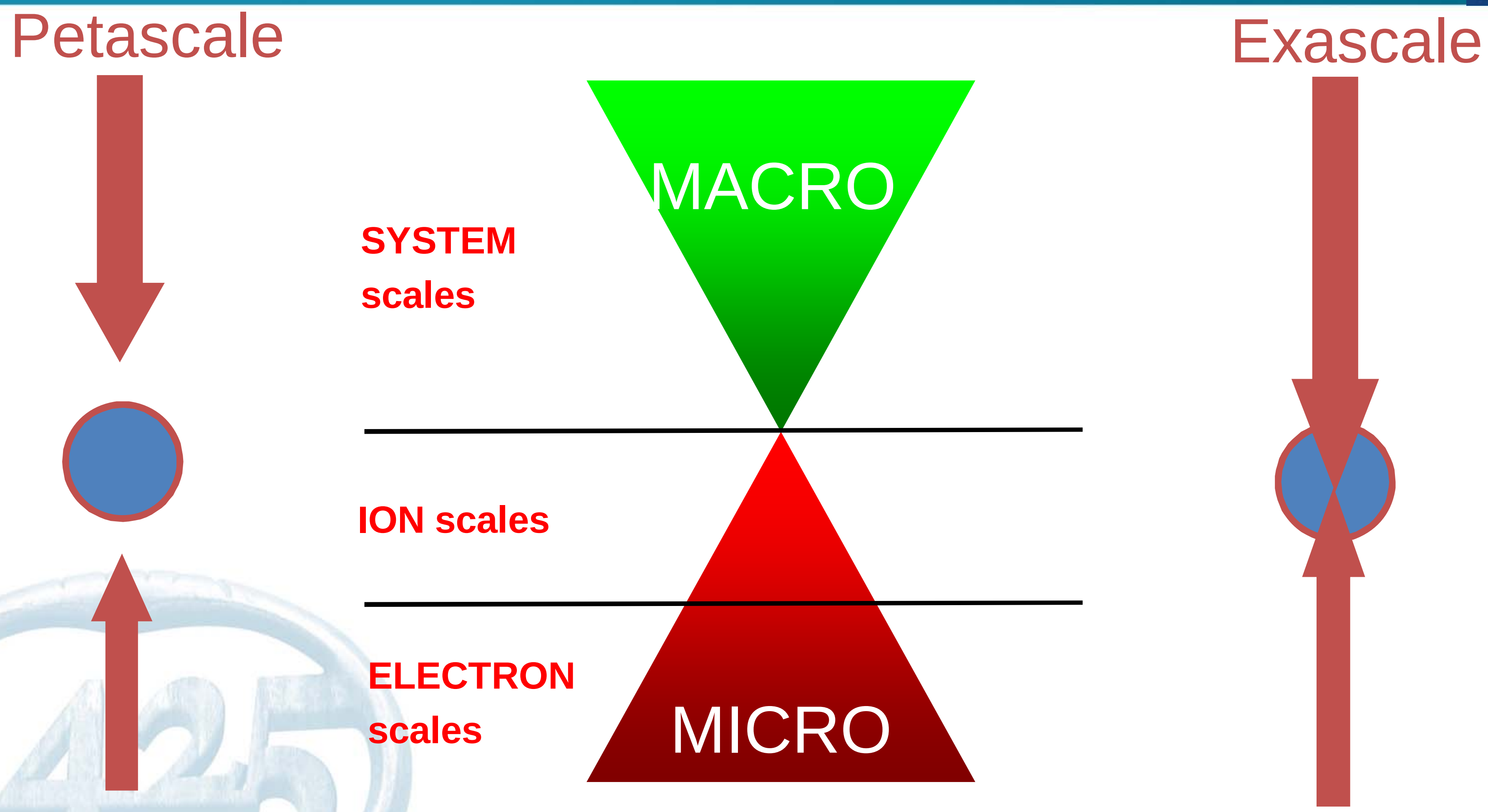
Large-scale integrating project (IP)

Towards Exascale with application to:

- Detailed brain simulation
- Space Weather
- Climate simulation
- Computational fluid engineering
- High temperature superconductivity
- Seismic imaging

Job: postdoc to develop and test implicit PIC on GPUs

Goals of SWIFF and Exascale Lab

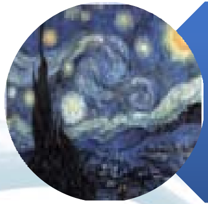


EXASCALE allows to bridge the micro-macro gap by increasing size and resolution by the needed 3 orders of magnitude

Codes and support material

http://www.glapenta.net/

http://swiff.eu/codes/matlab-ec/



http://code.google.com/p/celeste/

http://code.google.com/p/ipic3d/