

Statistical methods for the optimal planning of multi spacecraft missions to monitor space weather

G. Lapenta, S. Olshevsky, S. Bacchini,

M. E. Innocenti

Center for Mathematical Plasma Astrophysics, KULeuven, Leuven, Belgium

Space Science Institute, Boulder, USA

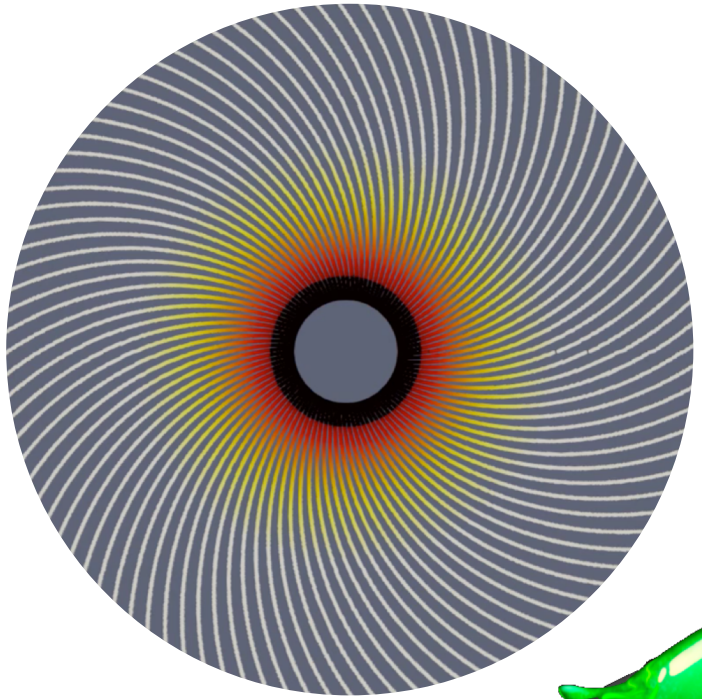
KU LEUVEN

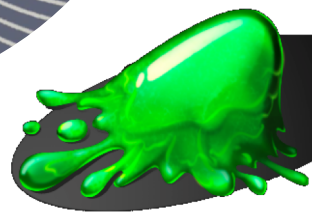
SPACE SCIENCE INSTITUTE

Outline

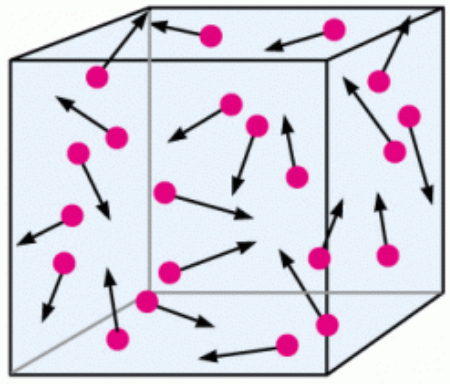
1. Lagrangian particle methods
    - Why we need the method
    - Implementation
    - Examples

2. Application to space need data assimilation
    - Kalman Filtering
    - Representer technique
    - Domain of influence in heliospheric problems.





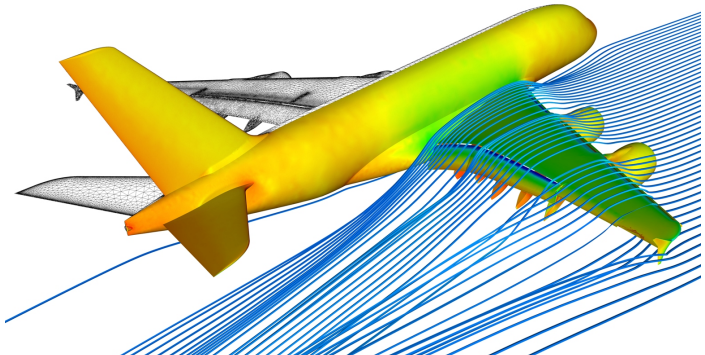
Kinetic versus Fluid



6D

Kinetic approach: study the distribution function (probability of finding a particle with a given velocity in a given point at a given time):

$$f(\mathbf{x},\mathbf{v},t)$$

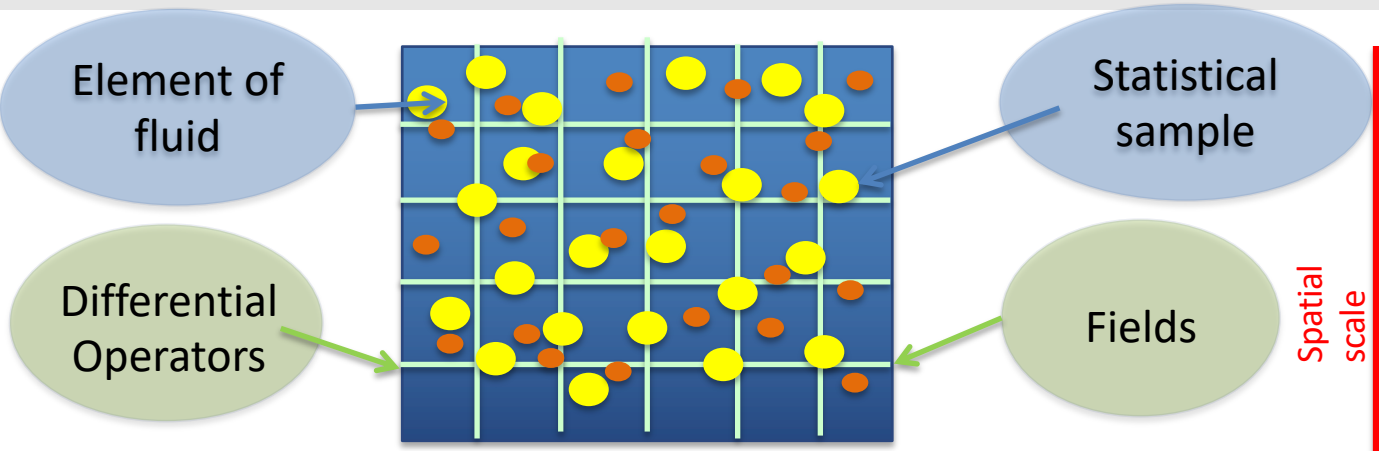


3D

Fluid approach: study local averages (density, average speed, temperature,….)

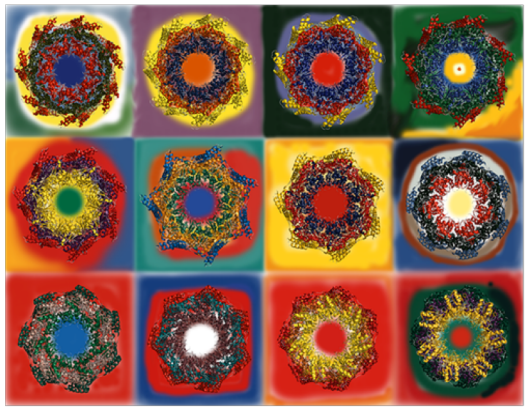
$$n(\mathbf{x},t),\mathbf{U}(\mathbf{x},t),T(\mathbf{x},t)$$

Particle-based simulationmethods can treat all levels of description

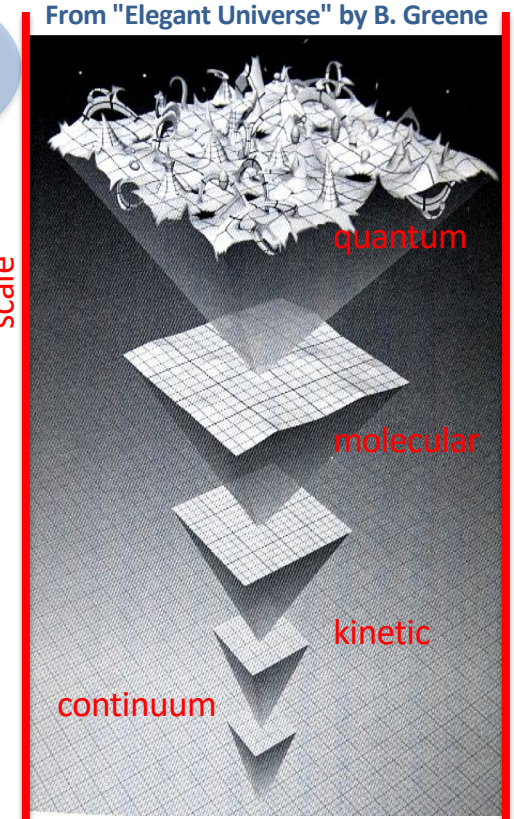




Wassily Kandinsky: Squares and Concentric Rings

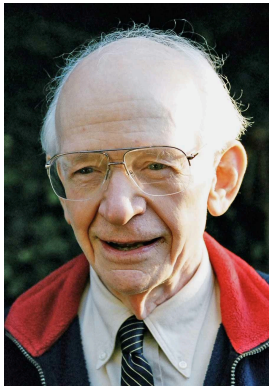


Yang, et al. (2009) *PLoS Comput Biol* ***5(4)****: e1000360.*



The idea for Fluid PIC

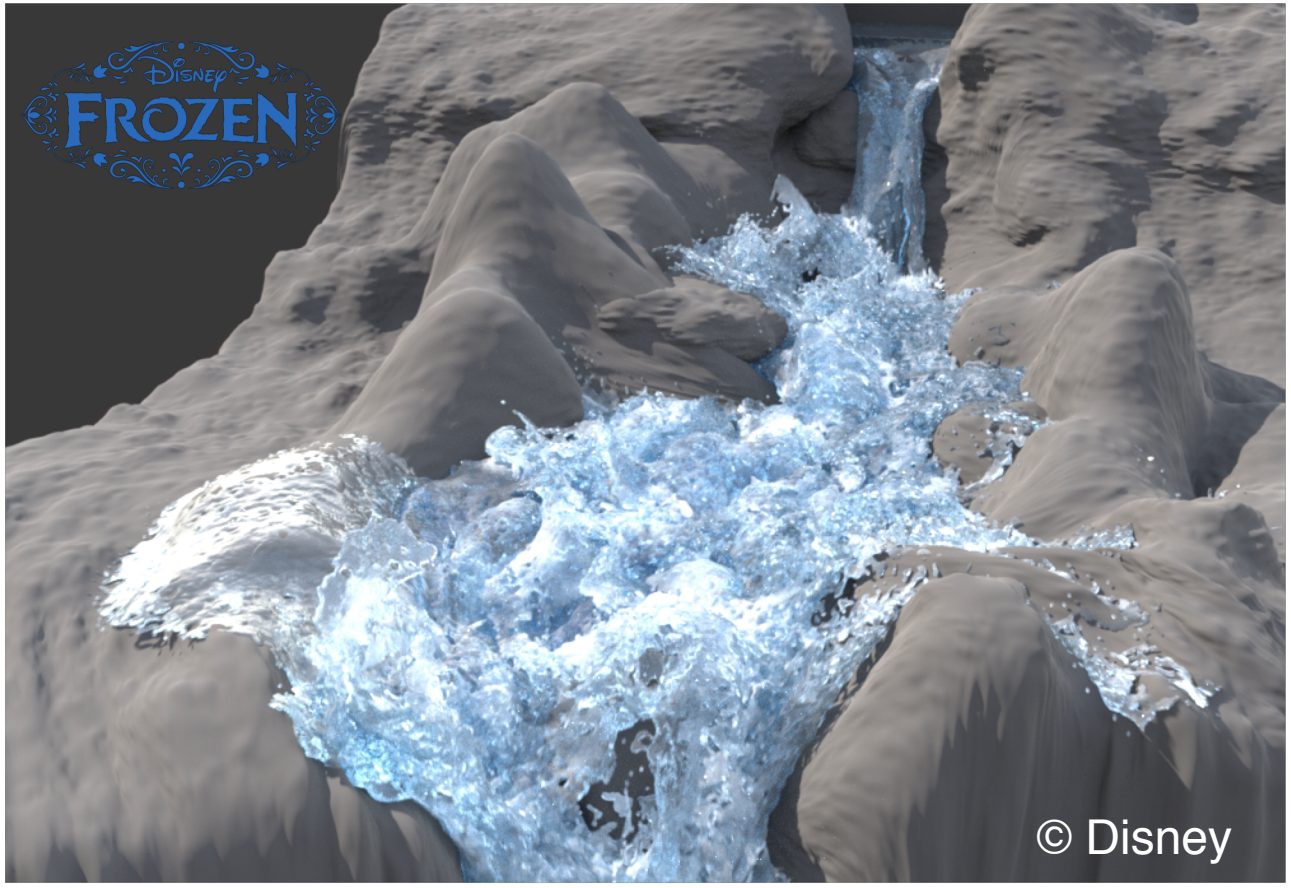

Frank Harlow


Ned Birdsall

PIC is NOT just a kinetic model!

- **Fluid Particle-in-cell** (PIC) method for compressible fluids was invented by F. Harlow (1959), at LANL. It combines particles to follow material motion with a grid to solve the equations of motion. *Each particle is a blob of fluid.* See Lapenta and Brackbill 2008 for a review.
- **Kinetic Particle-in-cell** (PIC) was invented in the same period (Buneman, Dawson, Hockney, Birdsall, Morse and others) for plasma simulation. It combines Vlasov + Maxwell (Brackbill 2005, Lapenta 2012 for a review). *Each computational particle is a cloud in phase-space.*

PIC Fluid methods chosen by Disney for the effects in the movie Frozen



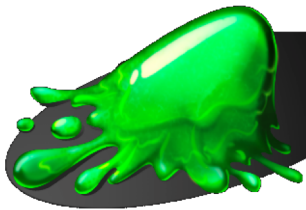


Part 1: Numerical modeling approaches

- Pure grid-based **Eulerian** – suffers from numerical dissipation
- **Lagrangian** – suffer from grid deformations and stretching
- **Arbitrary Lagrangian-Eulerian (ALE)** – highly dissipative
- Pure particle: **smoothed particle hydrodynamics (SPH)** – expensive, tricky to reproduce key instabilities

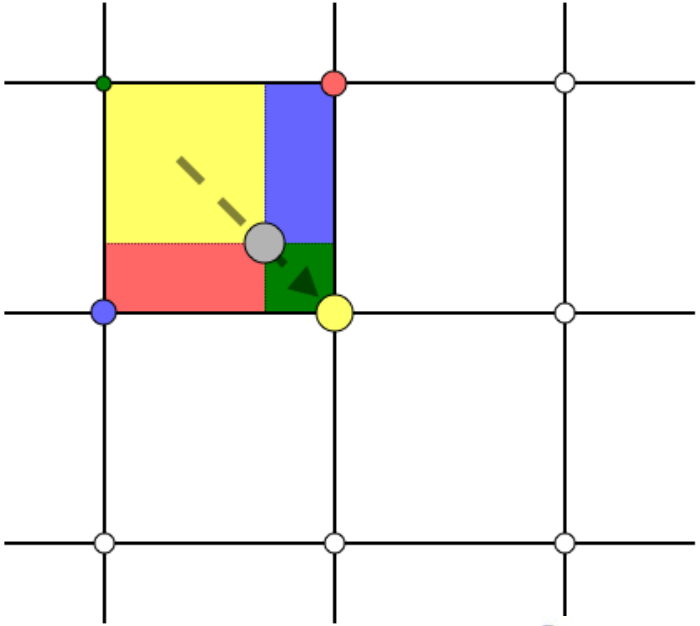
Slurm uses

Lagrangian particles + fixed grid = **Particle-in-cell**

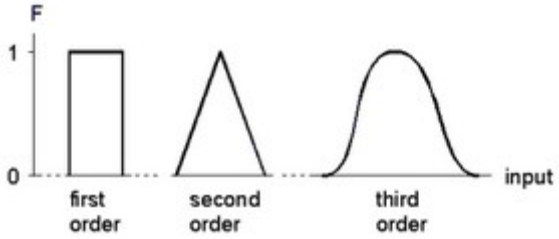


Bacchini et al. (*Computer Physics Communications*, 2017)

Interpolation between particles and grid



https://www.particleincell.com/



Interpolation Weight

Density on the grid

$$\rho_C = \sum_p m_p S_p(x_C - x_p)$$

Shape function S_p

$$\int_{-\infty}^{\infty} S_\xi(\xi - \xi_p)d\xi = 1$$

B-splines (basic spline)

$$b_0(x) = \begin{cases} 1 \text{ if } |x| < 0.5 \\ 0 \text{ otherwise} \end{cases}$$

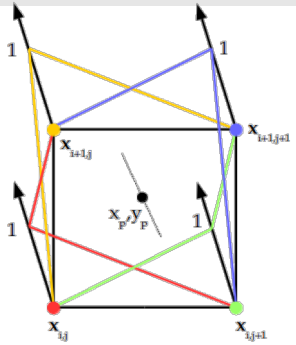
$$b_{n+1} = \int_{-\infty}^{\infty} dx' b_0(x - x') b_n(x')$$

Fluid Particle In Cell: computational cycle

$$\boldsymbol{x}_p^n, \boldsymbol{u}_p^n, e_p^n, V_p^n, \boldsymbol{A}_p^n$$

$$\downarrow$$

$$\boldsymbol{x}_p^{n+1}, \boldsymbol{u}_p^{n+1}, e_p^{n+1}, V_p^{n+1}, \boldsymbol{A}_p^{n+1}$$

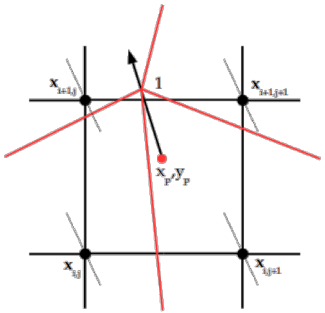


Project Particles – grid

Advance grid

Project grid – particles

Advance particles



$$\frac{1}{\rho}\frac{\mathrm{d}\rho}{\mathrm{d}t} = -\nabla \cdot \mathbf{u}$$

$$\rho\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = -\nabla\left(p + \frac{\mathbf{B}^2}{2\mu_0}\right) + \frac{(\mathbf{B}\cdot\nabla)\mathbf{B}}{\mu_0}$$

$$\rho\frac{\mathrm{d}e}{\mathrm{d}t} = -p\nabla\cdot\mathbf{u}$$

$$\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}t} = \boldsymbol{u}\times\boldsymbol{B} + (\mathbf{u}\cdot\nabla)\mathbf{A}$$

Pro and Contras

Pro

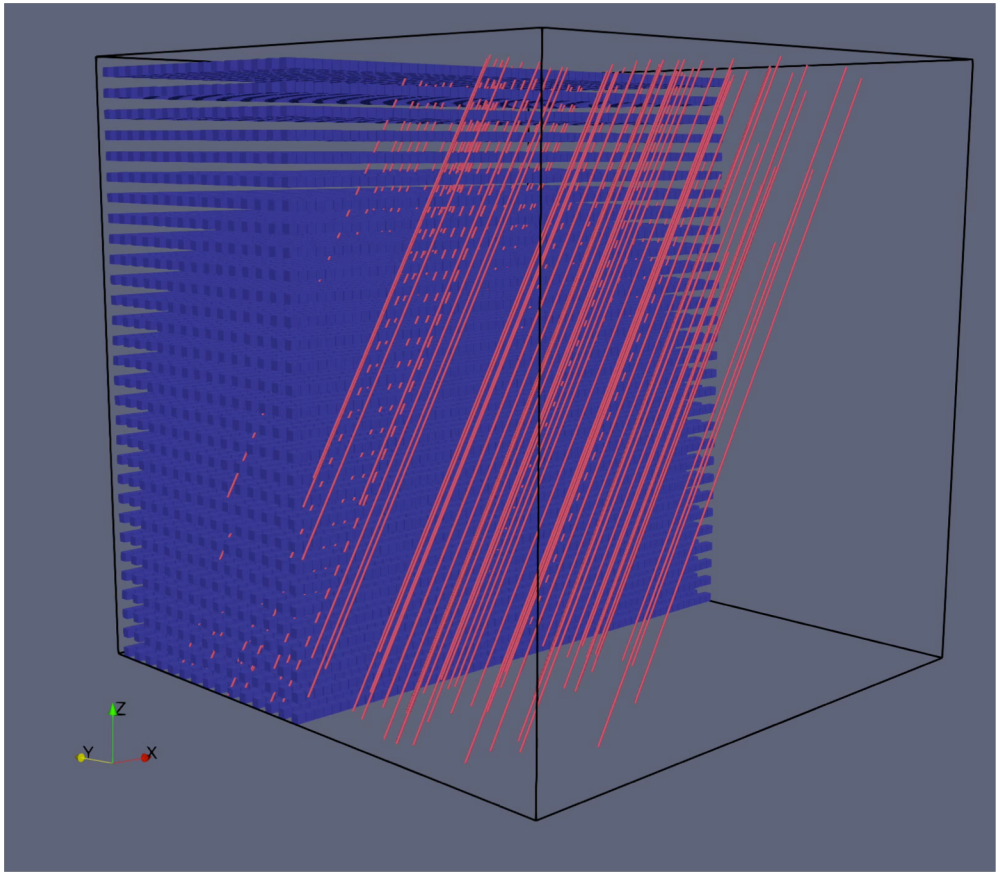
- Superbly parallelizable
- GPU-friendly
- Exact Advection
- Exact Material interfaces
- Sharp Discontinuities
- It replaces the CFL condition with a weaker one
- Suitable to:
  - High speed, super-, ultra- sonic flows
  - Solid materials (MPM), granular materials, solid or liquid interface, surface tension

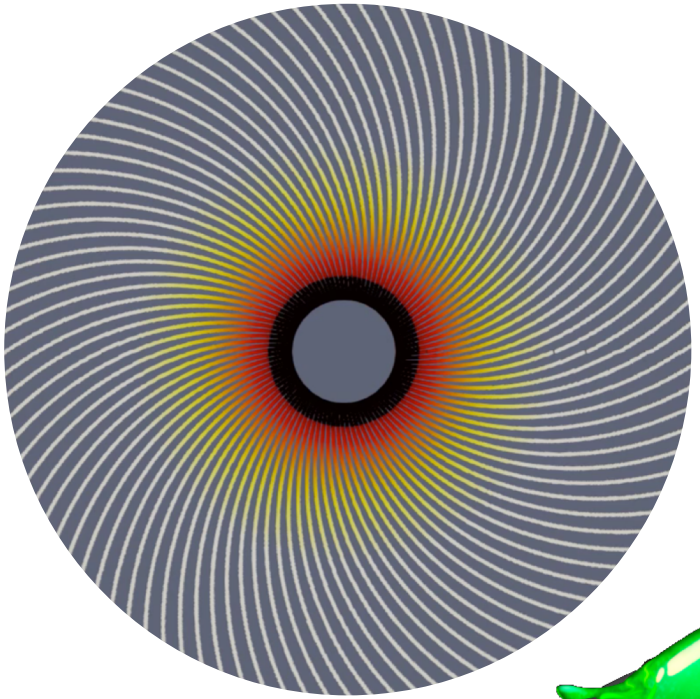
Needed work

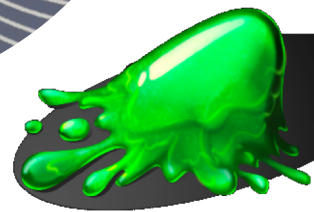
- Early on memory requirements were an issue
- Three main numerical issues for going to MHD, solved in our research:

1. Implementation in modern computers 👍
2. Aliasing between particles and cells (ringing instability) 👍
3. Account for action at a distance coming from electromagnetic field (peculiar to MHD) 👍

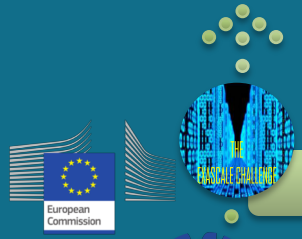
Advantages for the Bz project







Kinetic Models



xPic

Lapenta et al, CPC, 2017.

Hybrid computing



ECSIM

Lapenta, JCP, 2017;





Distributed computing

iPic3D

Markidis, Lapenta, MCS, 2010





Workstations

Celeste

Lapenta, Ricci, Brackbill, PoP, 2005



Vector Computing

Venus

Brackbill, Forslund, JCP, 1985



Exactly energy Conserving PIC (support form AFSOR)



GMRES Solver for non-symmetric matrices

Fluid (MHD) Models

Evolving geometric tensor from particles (support form AFSOR)





SLURM



Bacchini, Olshevsky, Poedts, Lapenta, CPC, 2017

Hybrid computing

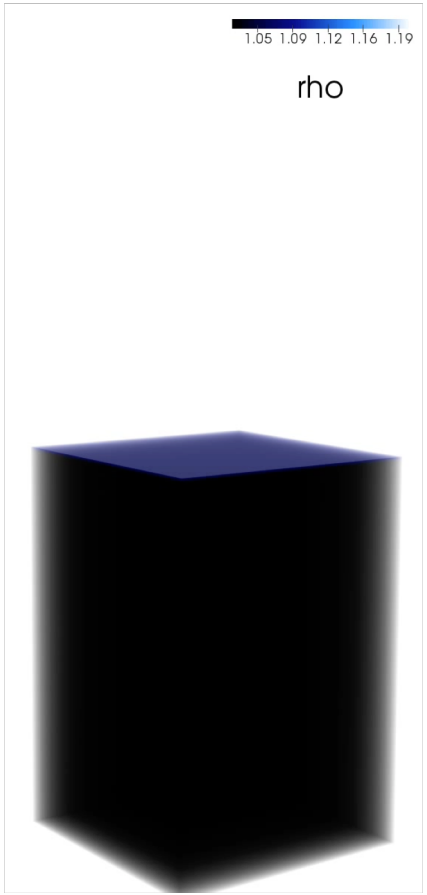


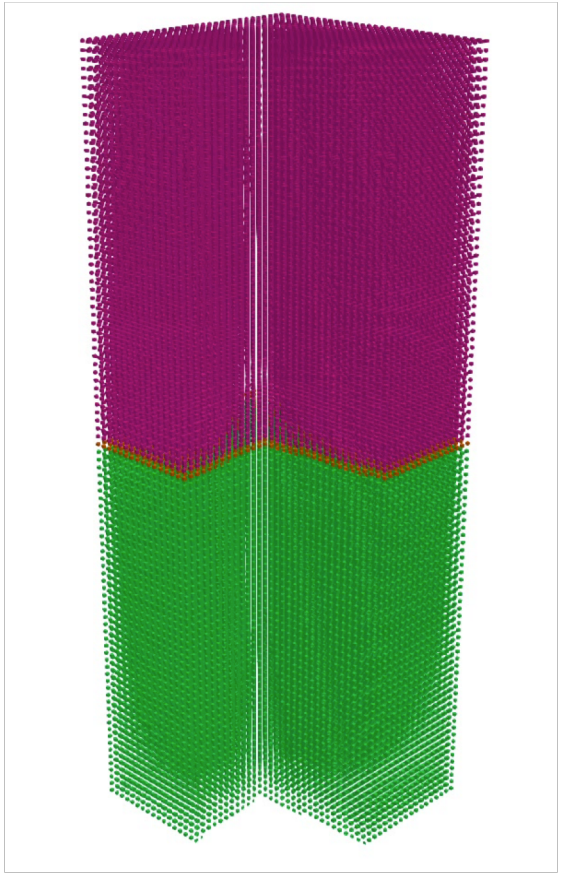


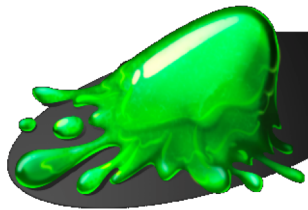
Flip3DMHD

Brackbill, JCP, 1991

Examples of Slurm: Rayleigh-Taylor instability and Expanding Solar Wind

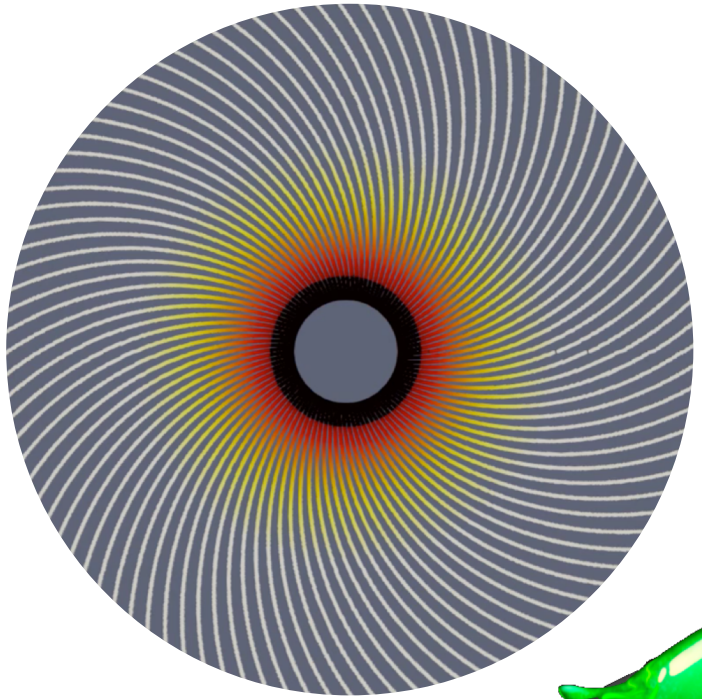


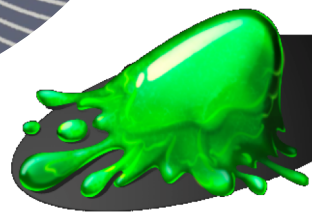




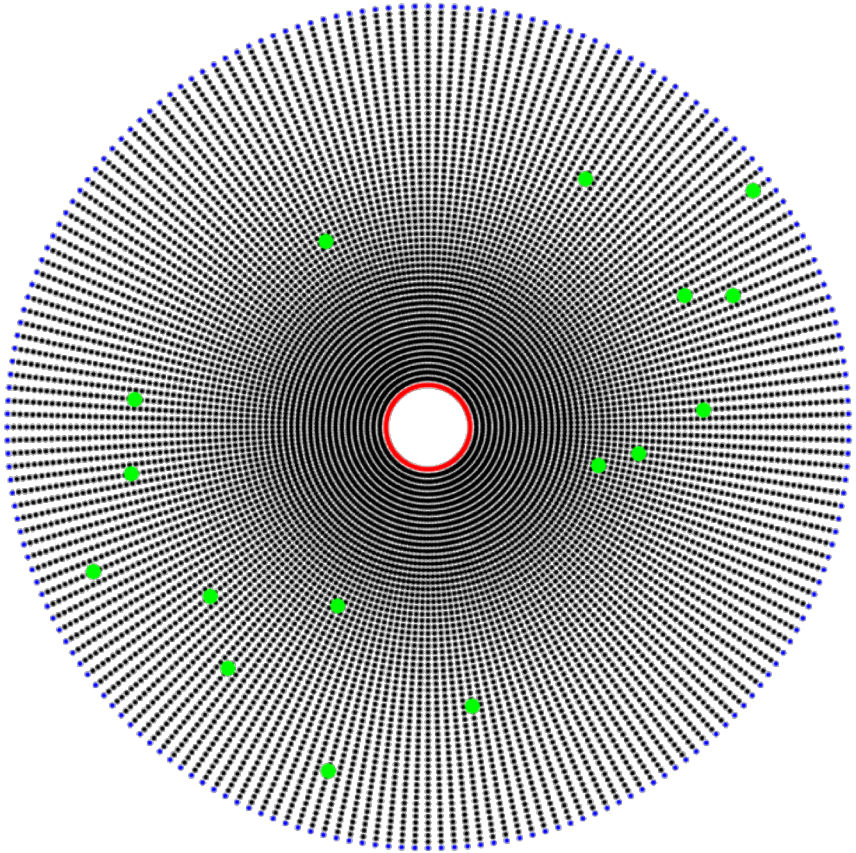
Our goal: SLURM ENLIL-like with Particles

- Import data from magnetograms and solar wind origin
- Expand it out as Lagrangian elements
- No dissipation of advection
- Sharp discontinuities and shocks (key to particle acceleration models)





Data Assimilation form multiple missions



Part 2: Kalman Filtering

1. Some Concepts of Bayesian Statistics
2. Kalman Filtering for Heliospheric Modelling
3. Domain on Influence and Multispacecraft Missions

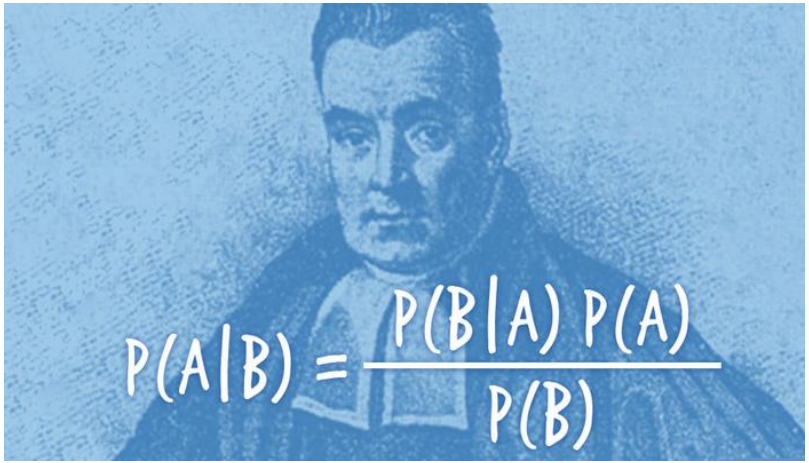


Figure: Thomas Bayes (1701–1761)



Figure: Rudolf Emil Kálmán (1930–2016)

Bayesian Statistic

Frequentist interpretation: probability measures a fraction of outcomes.

e.g.: How many of the next 1000 Earth-Directed CMEs will be of the Carrignton strength?

Bayesian (epistemological) interpretation: probability measures a degree of belief.

e.g.: What degree of confidence will we have that an Earth-Directed CME is a Carrington-type event after carrying a protocol of measurements?

Applications

- The first approach helps us to obtain a probabilistic description of **risk**.
- The second approach helps us to obtain a level of confidence in **forecasts**.

Introductory concepts

Bayes Theorem

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

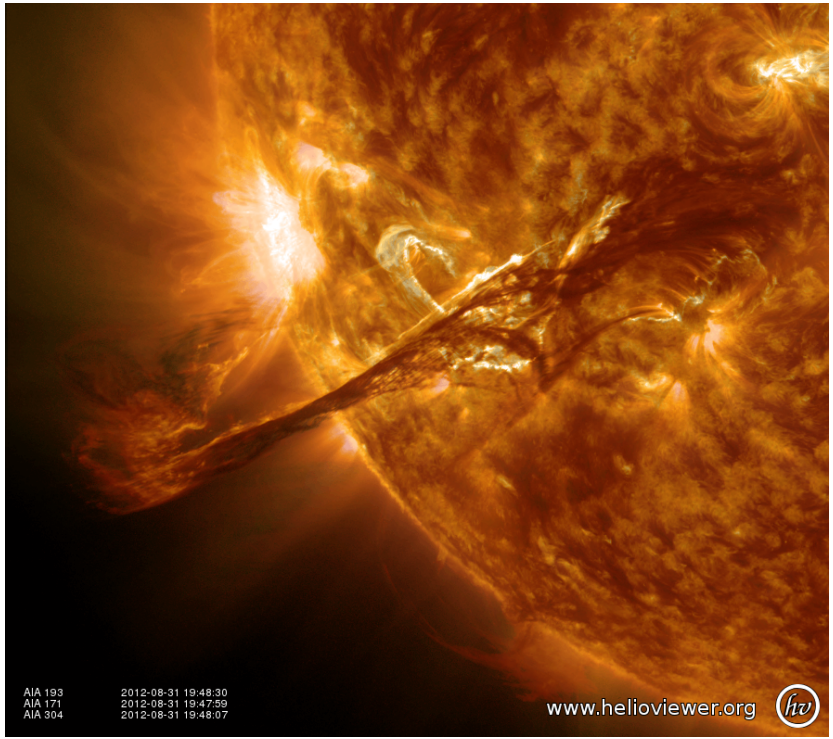
where A and B are events and$P(B) \neq 0$.

- $P(A \mid B)$ is a conditional probability: the likelihood of event A occurring given that B is true.
- $P(B \mid A)$ is also a conditional probability: the likelihood of event B occurring given that A is true.
- $P(A)$ and $P(B)$ are the probabilities of observing A and B independently of each other; this is known as the *marginal probability*.

Cassandra's Curse: predicting rare events

Let us consider an application of Bayes theorem: The 1 in a 1000 superstorm.

- Suppose our detection system for the initiation of a superstorm on the Sun is **99% sensitive**, i.e. it detects 99% of superstorms.
- Suppose also our system is also **99% selective**, i.e. it screens out 99% of non superstorms (there is just 1% probability of mistaking a regular storm for a superstorm).
- But let us also imagine **only 1 in 1,000 of the CMEs starting in the visual field becomes superstorms**, the other will not reach the Earth or are not geoffective.



Cassandra's Curse: we can't predict rare events

Let us consider an application of Bayes theorem: The superstorm

- Superstorm Sensitivity: 99%: 99 true positives for every 100 real superstorms
- Superstorm Specificity: 99%: 99 true negatives for every 100 non superstorms
- Superstorm frequency: 0.1%

Let us Apply Bayes Theorem:

$$\mathrm{P}(\text{superstorm} \mid \text{detect}) = \frac{\mathrm{P}(\text{detect} \mid \text{superstorm})\mathrm{P}(\text{superstorm})}{\mathrm{P}(\text{detect})} = \frac{0.99 \times 0.001}{0.99 \times 0.001 + 0.01 \times 0.999} = 9\%$$

$\mathrm{P}(\text{detect} \mid \text{superstorm}) = 0.99$
$\mathrm{P}(\text{superstorm}) = 0.001$
$\mathrm{P}(\text{detect}) = \mathrm{P}(\text{detect} \mid \text{superstorm})\mathrm{P}(\text{superstorm}) + \mathrm{P}(\text{detect} \mid \text{non} - \text{superstorm})\mathrm{P}(\text{non} - \text{superstorm}) =$
$0.99 \times 0.001 + 0.01 \times 0.999$

The percentage of **false positive** is 91% despite an efficiency and a sensitivity of 99%.

Kalman Filtering

- A priory forecast: $\mathbf{x}^f(t_i) \in \mathbb{R}^N$
- Measurements: $\mathbf{y}^o(t_i) \in \mathbb{R}^M$
- Typically $N \gg M$
- Kalman filtering uses Bayesian statistics to improve the *a priori estimate* to produce an *a posteriori estimate* improved by the measurements: $\mathbf{x}^a(t_i) \in \mathbb{R}^N$
- The *observation operator* $H(t_i)$ projects the model space $\mathbb{R}^N$ to the observation space $\mathbb{R}^M$.

- Analysis step:

$$\mathbf{x}^a(t_i) = \mathbf{x}^f(t_i) + K(t_i)\left[\mathbf{y}^o(t_i) - H(t_i)\mathbf{x}^f(t_i)\right]$$

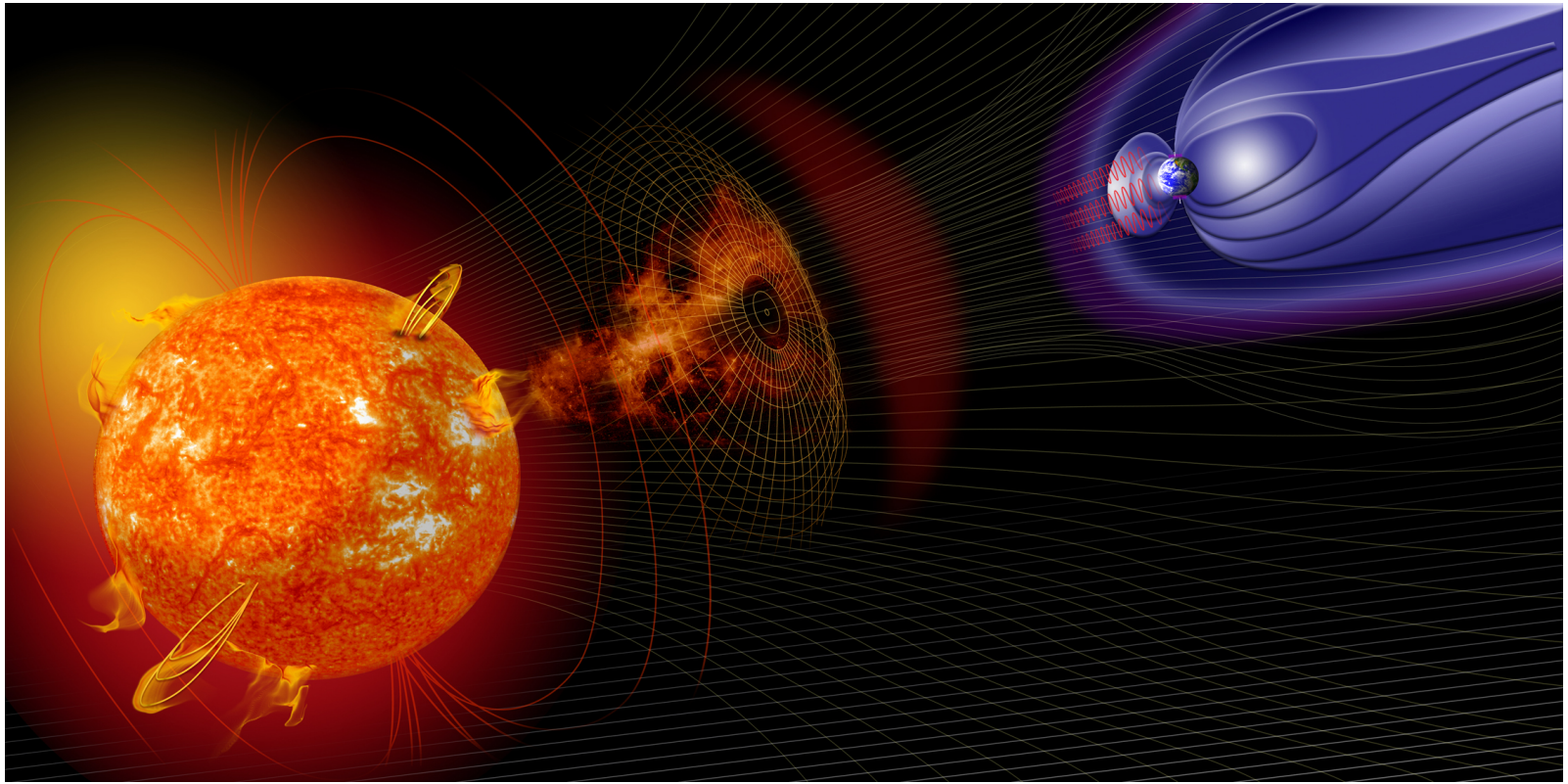
- The Kalman gain matrix is obtained by minimising an error cost function

Kalman Gain

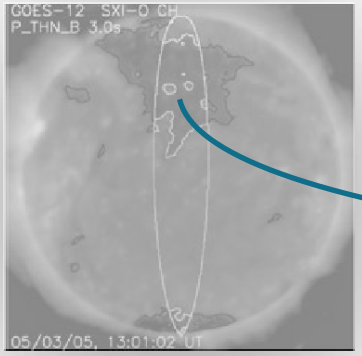
$$K(t_i) = P^f(t_i)H^T(t_i)\left[H(t_i)P^f(t_i)H^T(t_i) + R(t_i)\right]^{-1}$$

- Computed using:
  - Observation operator H.
  - Forecast error covariance matrix P^f.
  - Observational error covariance matrix R.
- The **Ensemble Kalman Filtering** method uses an ensemble of models to compute the forecast error covariance P^f.

Application to Space Weather



Non-standard Kalman filtering applied to an empirical solar wind forecasting model



% of the meridional slice of the Sun covered by Coronal Holes

empirical solar wind forecasting model

VRSNAK 2007

improved forecast of selected solar wind parameters

$\tilde{x}_k$

$z_{k+\Delta t_1}$

P_k^-

$K\ F$

INNOCENTI 2011

$\hat{x}_k$

P

$f_{k+\Delta t_2}$

P_k

Σ

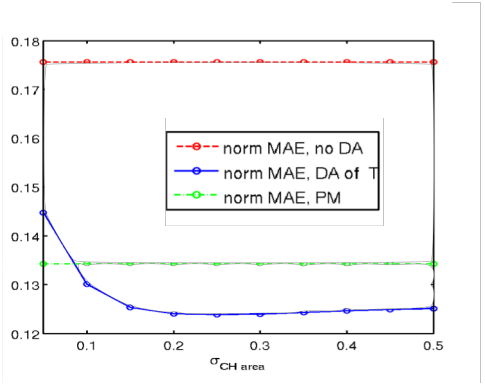
Q

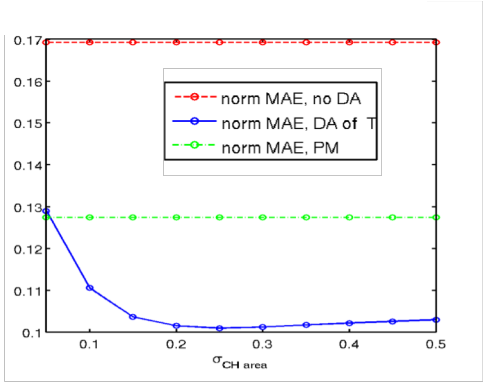
tunable parameter: process noise covariance

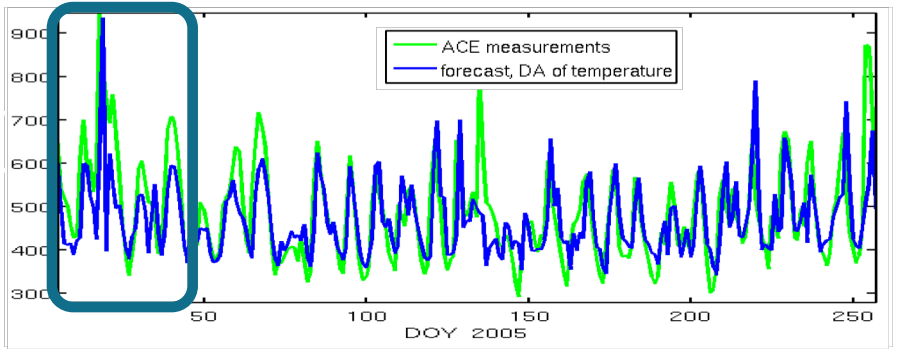
ACE observations of selected solar wind variables



Sample of Results



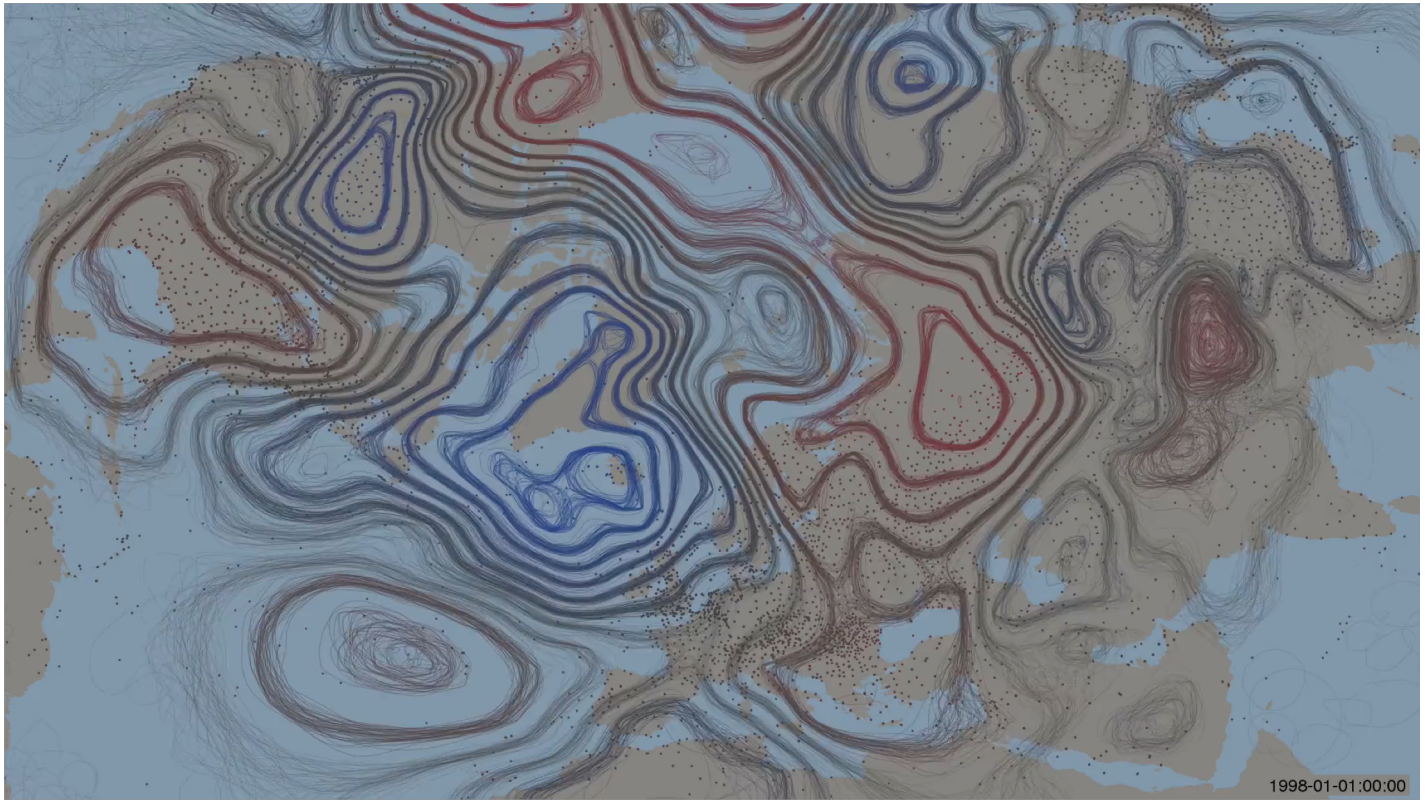




INNOCENTI et al. 2011

- statistical techniques (Data Assimilation with modified Kalman filtering) improve the overall quality of the forecasts,
- especially in periods of geogmagnetic activity (info on the ongoing g.a. is brought into the forecast by the assimilation)

Example of Kalman filtering used on Earth



Representer (Influence Function)

The method is based on rewriting the analysis step:

$$\mathbf{x}^a(t_i) = \mathbf{x}^f(t_i) + K(t_i)\left[\mathbf{y}^o(t_i) - H(t_i)\mathbf{x}^f(t_i)\right]$$

interpreted as a correction produced by a combination of representers:

$$\mathbf{x}^a(t_i) = \mathbf{x}^f(t_i) + \sum_{j=1}^{M} \boldsymbol{\rho}_j(t_i) b_j(t_i)$$

- The representer $\boldsymbol{\rho}_j(t_i) \in \mathbb{R}^N$ has the dimensions of the state vector and represents the effect of the observation j on improving the forecast.
- The representers form a matrix defined by: $\{\boldsymbol{\rho}_j\} = P^f(t_i)H^T(t_i)$
- The scalar coefficients $\{b_j\}$associated with the observation are given by:

$$\left[H(t_i)P^f(t_i)H^T(t_i) + R(t_i)\right]^{-1}\left[\mathbf{y}^o(t_i) - H^T(t_i)x^f(t_i)\right]$$

Domain of Influence

The correction due to a specific observable k can then be written as:

$$\mathbf{x}^a(t_i) = \mathbf{x}^f(t_i) + b_k(t_i)\boldsymbol{\rho}_k(t_i)$$

- Using the *Ensemble Kalman filtering* this term is expressed via the correlations and covariances within the ensemble.
- The correction on the state variable i (assuming that each observation refer to a state variable) due to measurement k is:

$$\delta\mathbf{x}_i = \underbrace{\boxed{corr(\epsilon_i^f, \epsilon_k^f)}}_{\text{domain of influence}} \underbrace{\boxed{\frac{\sqrt{var(\epsilon_i^f)/var(\epsilon_k^f)}}{1 + \frac{var(\epsilon_k^o)}{var(\epsilon_k^f)}}}}_{\text{modulation factor}} \underbrace{\boxed{\delta\mathbf{y}_k}}_{\text{innovation}}$$

where ϵ^f (ϵ^o) are the model (observational) forecast errors, estimated by ensemble variances

Ensemble generation

The ensemble is generated by initialising many (100) simulations with different boundary conditions, obtained from **a base state** with a time-dependent **perturbation generated from a Gaussian** distribution with μ=0, σ= 0.05

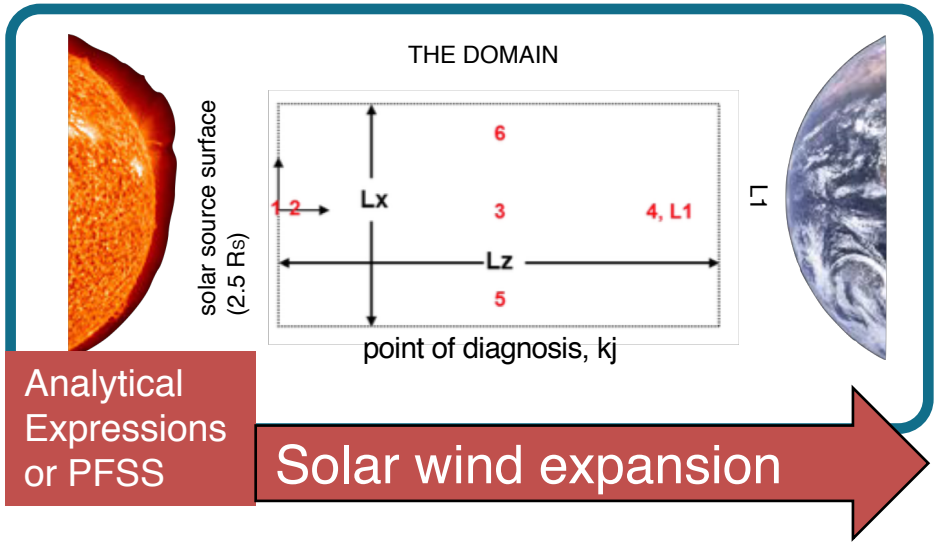
THE CODE

SLURM

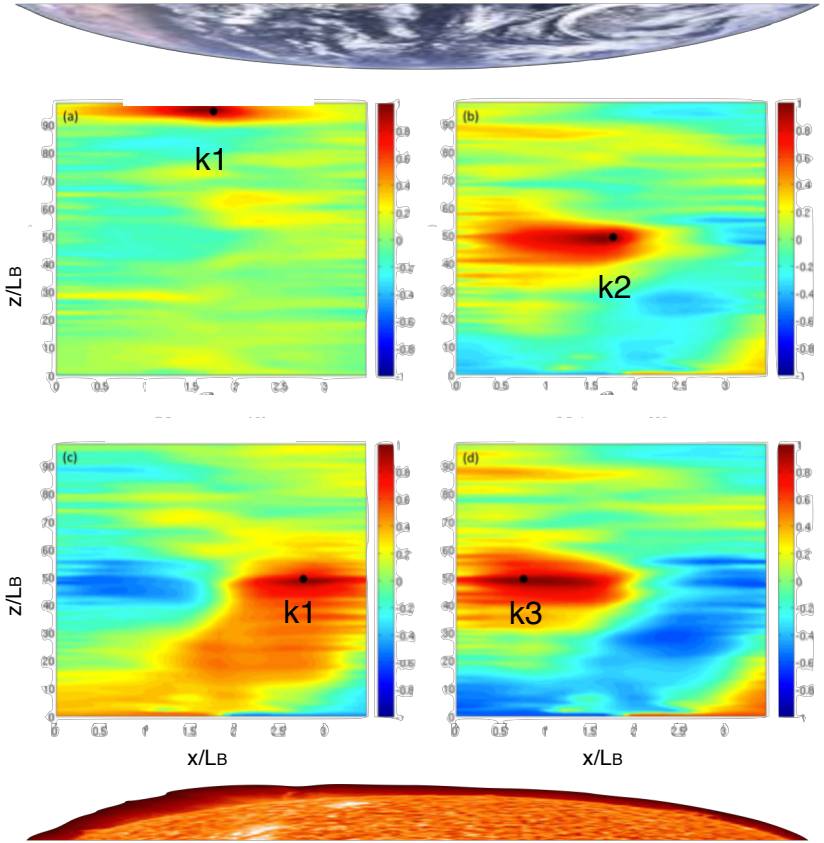
THE CODE

FlipMHD

BRACKBILL 1991



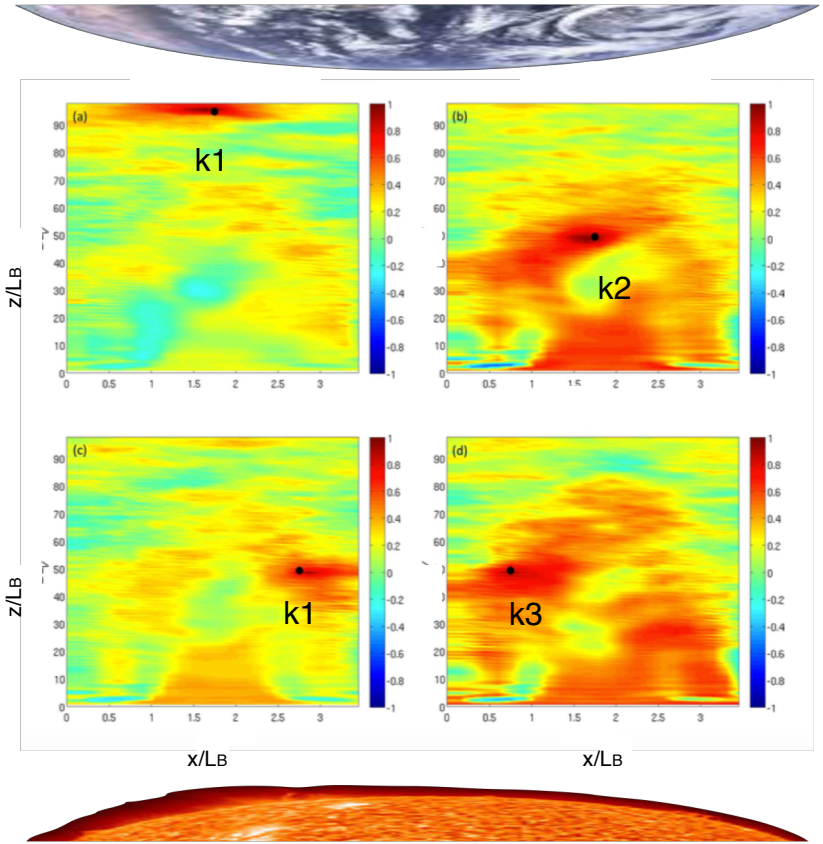
Domain of influence – Bz Magnetic Field



- Areas of positive and negative correlation, which change according to the position k of the observation point
- Missions closer to Sun, with respect to missions closer to Earth, increase the areas of both the regions which benefit and which are damaged by assimilation

SKANDRANI et al. 2014

Domain of influence – Radial Velocity



- Correlation is predominantly positive, notwithstanding the position k of the observation point
- Missions closer to Sun, with respect to missions closer to Earth, greatly increase the areas of positive correlation
- the domain of influence analysis can help “placing” future missions with the aim of providing optimal assimilation point;
- it can also estimate a radius around the assimilation point beyond which no improvement in the state knowledge can be expected from the assimilation

SKANDRANI et al. 2014

Conclusions

- SLURM has solved all three of the issues relative to MHD-PIC:
  - Parallel implementation
  - Finite Grid Instability
  - Formulation of the action at a distance via vector potential and current formulation
- Several test passed:
  - Standard tests (KH, RT, OT, Brio and Wu shocks)
  - Flux emergence , Flux rope CME initiation and Solar wind propagation
- Towards Bayesian methods for data assimilation:
  - Bayes' theorem indicates the need to have a good balance between constraints and uncertain variable.
  - Ensemble Kalman filters provide a method to assess the covariance matrices
  - The representer technique gives as a tool to evaluate the efficacy of Kalman filtering based on observations in different regions
- **Proposal** for a **future operational** development of Slurm/AI with Data Assimilation

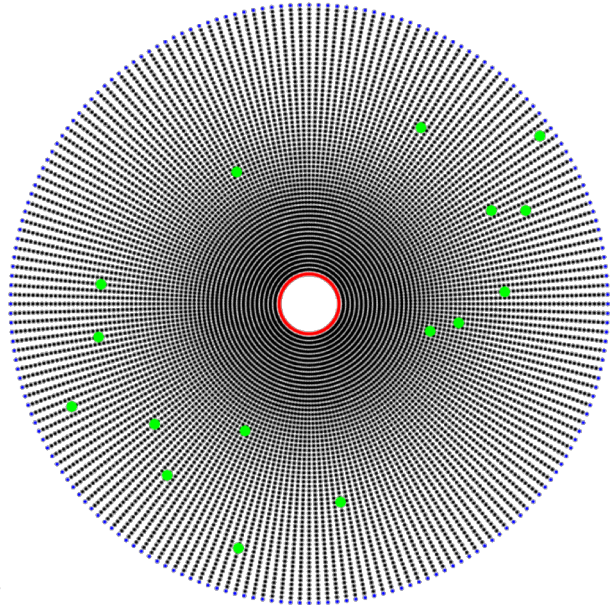
Idea for the future

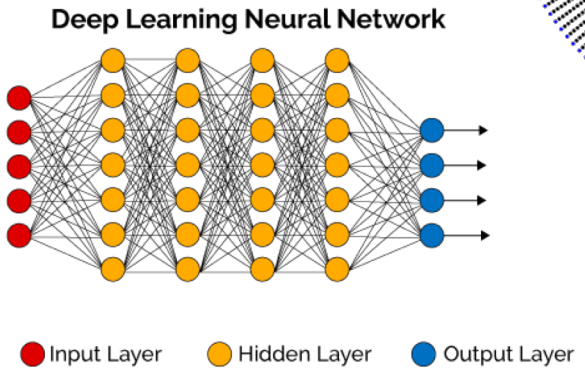
Input: Photospheric information

Kalman Filtering with:

- L1 spacecraft
- Kp and other Earth indexes
- Future: more missions - L5 - solar orbiter - solar probe

Slurm





Workshop Goals (1)

How can we better combine the various remote-sensing techniques (and modelling) for enhancing our understanding of the Heliophysical System?

How can we better utilise heliospheric remote-sensing observations for space-weather forecasting?

We are focusing on data assimilation via Ensemble Kalman filters and Deep Neural Networks

What additional/new data are needed for heliospheric models (those current, and those in development)?

What are the modelling gaps/needs in forecasting space weather throughout the Heliophysical System?

Our research points strongly to the need to go closer to the Sun: SO and PSP will teach us a lot on this.

Workshop Goals (2)

What steps needs to be thought through and what should be implemented/developed with remote-sensing observations to support the Solar Orbiter and Parker Solar Probe missions (particularly science planning for SO and collaborative/complementary observations for both)?

How do we get the wider Heliophysics Community to use and better engage with the available radio remote-sensing observations and models with currently utilise them?

We are planning to create a suite of python-based tools for data assimilation (of in situ missions at L1, PSP and SO and remote imaging) into Slurm-like heliospheric models and geospace coupled codes (e.g. openGGCM).

We are developing also python-tools based on Intel's tensorflow for Deep Learning of solar and geospace information (applicable to all sources of data including radio

Will we ever get reliable B_z ahead of the Sun-Earth L1 Point, and if so, how, when, and how reliable will it be?

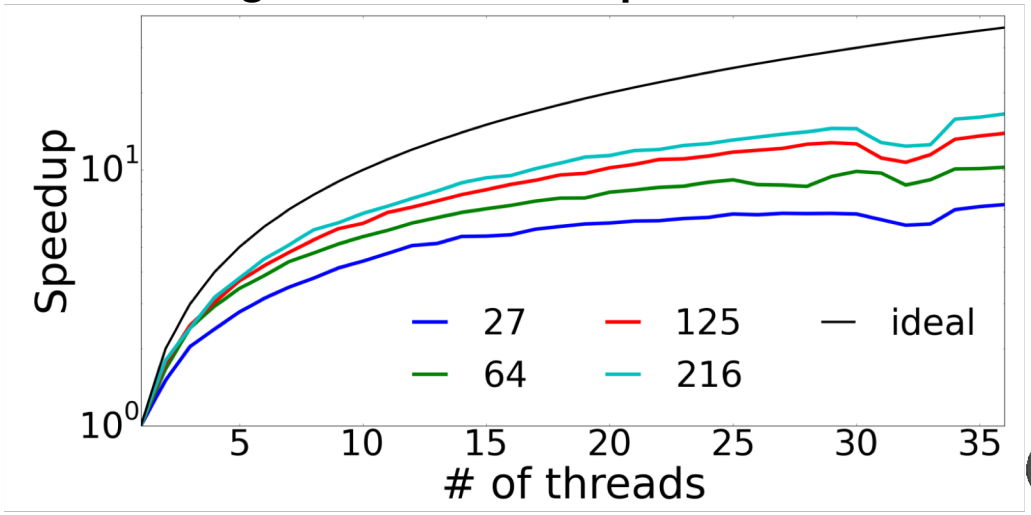
The representer method shown above shows great promise for missions around 0.5AU

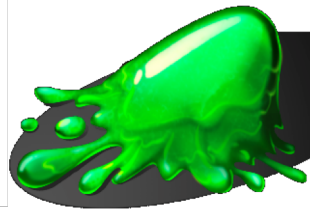
Extras

Slurm Parallel Implementation

- Cross-platform (C++ with CMake, only dependency: Boost)
- Boost arrays and lists as primary data structures
- Efficient parallelization on shared-memory architectures (OpenMP)
- Clusters of MICs or GPUs
- Output to VTKx

Scaling on Intel® Xeon® processor E5-2600 v4 (Broadwell)





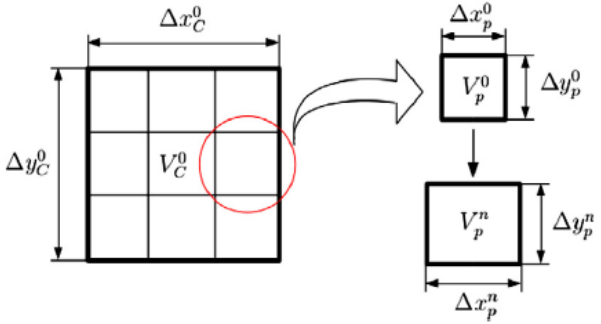
Particle volume evolution: Kelvin-Helmholtz

Instead of

$$\rho_c = \frac{1}{\Delta x \Delta y \Delta z} \sum_p m_p W_{Cp}$$

Use

$$\rho_c = \frac{\sum_p m_p W_{Cp}}{\sum_p V_p W_{Cp}}$$



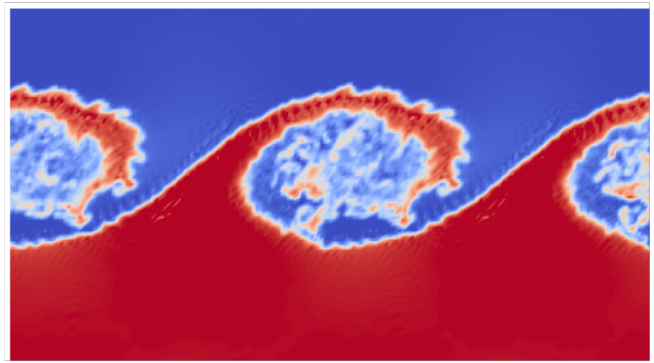
Volume evolution

$$V_p^{n+1} = V_p^0 J_p^{n+1}$$

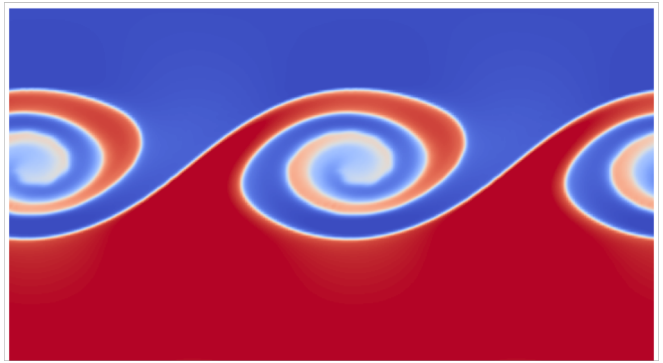
Particle deformation gradient

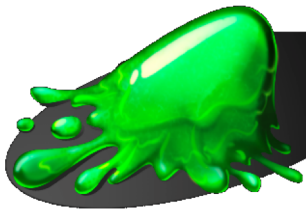
$$J_p^{n+1} = \left(\hat{\mathbf{I}} + \Delta t \nabla \boldsymbol{u}_p^{n+1}\right) J_p^n$$

Ringing instability



With volume evolution





Formulation of the Magnetic Field equations

Past proposed methods:

1. Assign a **particle magnetization** that can be used to compute the magnetic field [Brackbill, 1991]: need to clean div(B)
2. Advect electromagnetic **vector potential** $\mathbf{A}$ (carried by particles) [Bacchini, 2015]: *need to clean the gauge condition*
3. Both perform well but are requring to interpolate the full information and not the changes

New method:

1. Track the **current** in each element of fluid represented by the particles.
2. Evolve the particle current
3. Formulate the evolution equation for the current
4. Interpolate only the updates to the particle currents

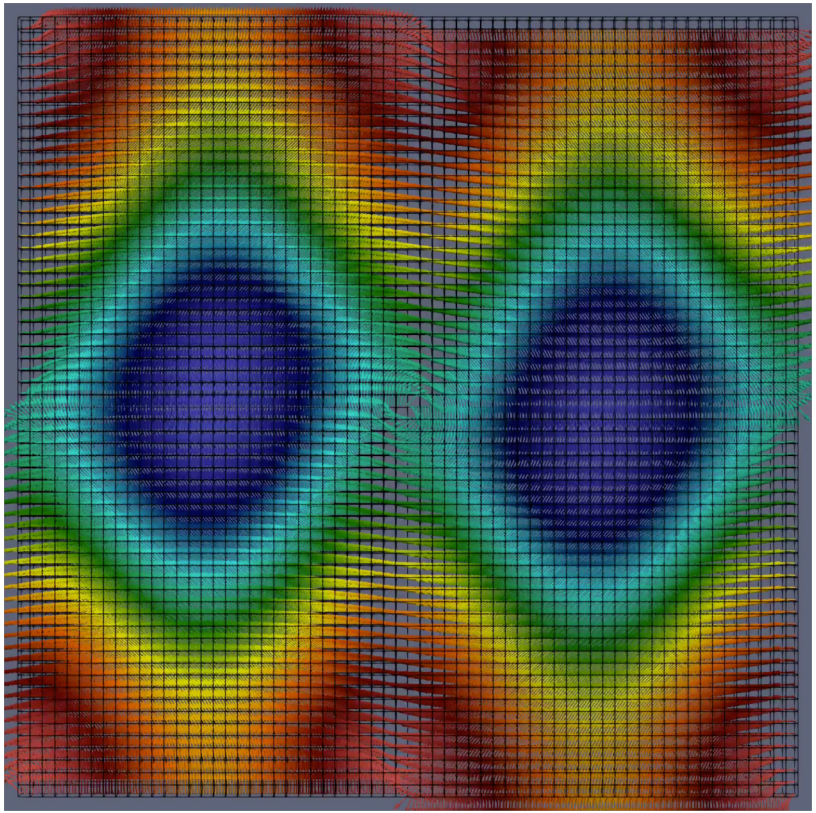
The new method is more efficient computationally and more directly linked to physical intuition.

MHD shocks: Orszang-Tang

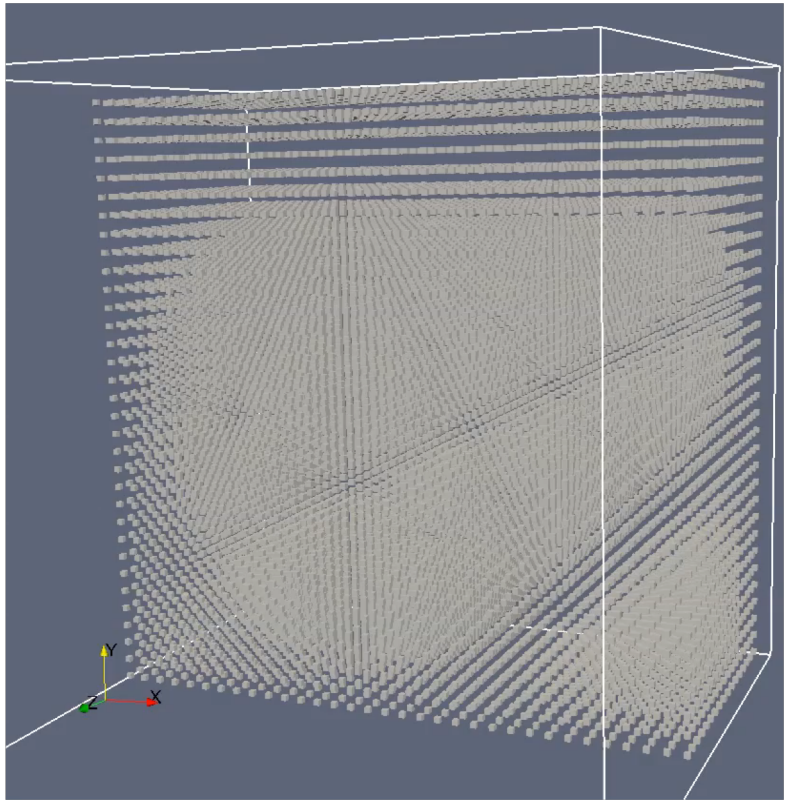
Pressure



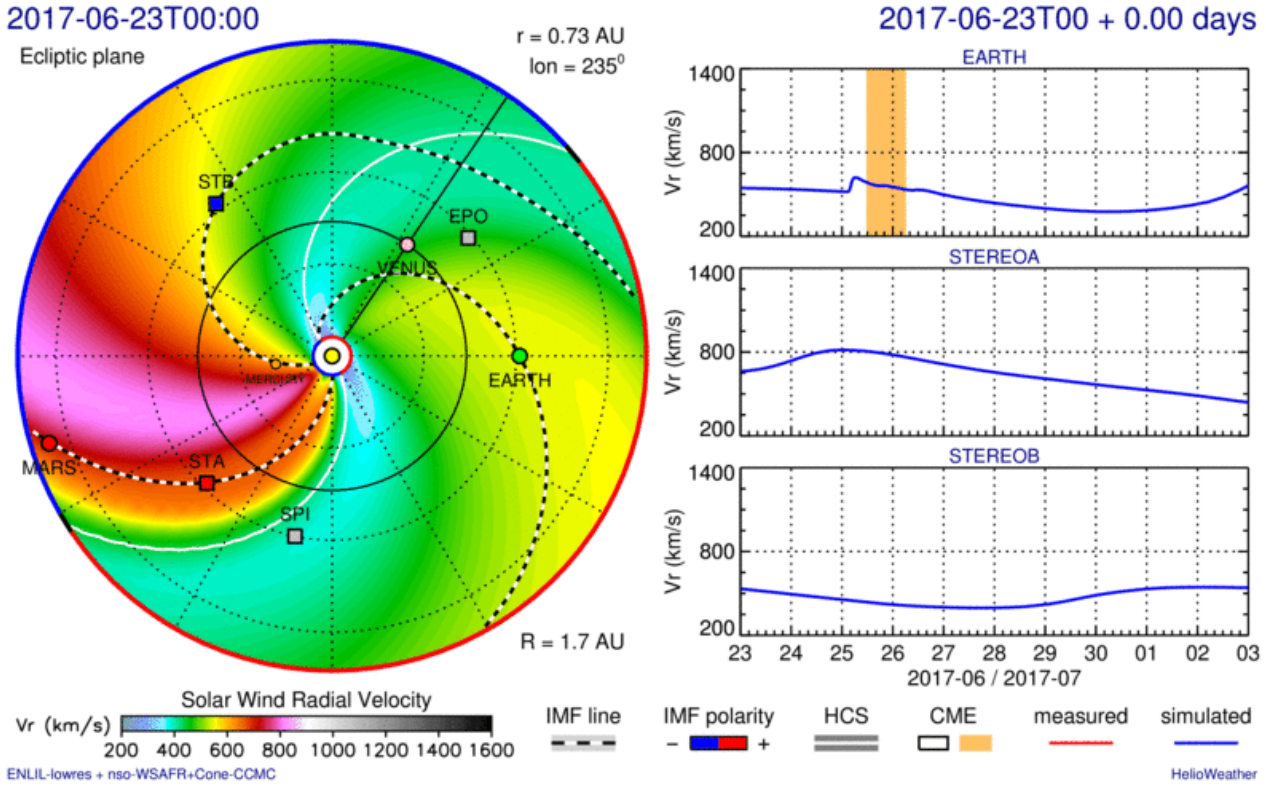
Az



SLURM example of application: Blast Wave



Future direction: Data Assimilation form multiple missions



CCMC: ENLIL Simulation: Olli_Tormanen_111017_SH_1