Guiding center Fokker-Planck theory and Monte Carlo method

Taina Kurki-Suonio
Eero Hirvijoki
Simppa Äkäslompolo

Department of applied physics, Aalto University

August 24, 2014

Outline

1 Fokker-Planck theory and stochastic motion

2 Guiding center formalism

3 High Performance computing

4 Applying the formalism – if time permits

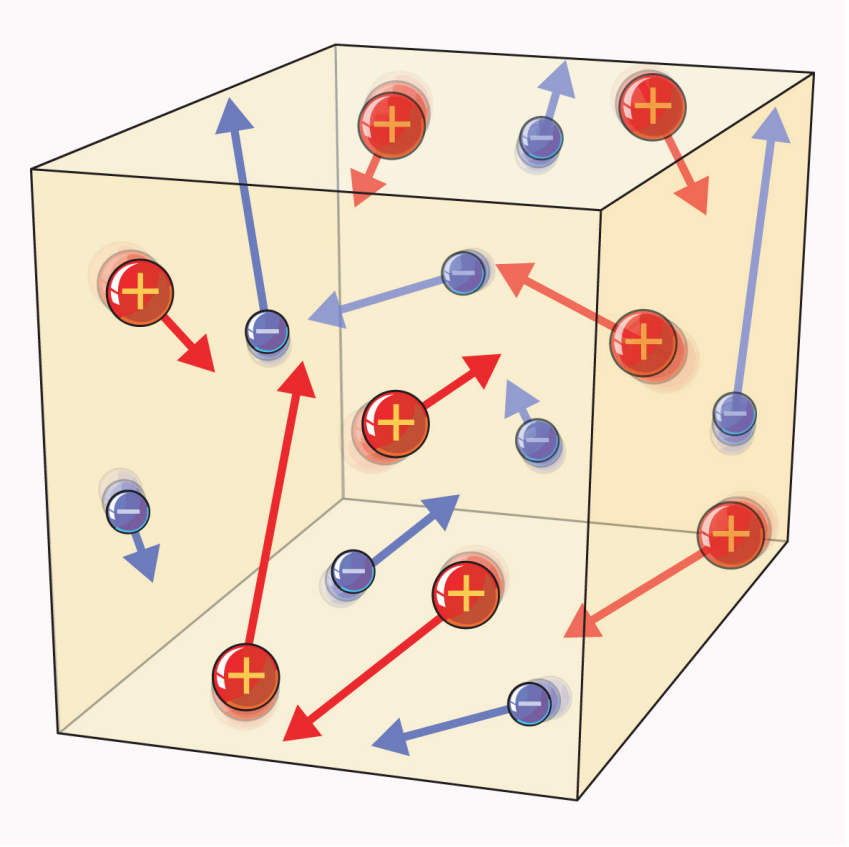
5 Summary

Outline

1 Fokker-Planck theory and stochastic motion
- Motivation
- Derivation of Fokker-Planck equation
- Deboning the FP equation, Part I: the RHS
- Deboning the FP equation, Part II: the LHS
- Equivalence to a stochastic differential equation

We need an efficient description for a plasma:

- Coulomb interaction has an infinite range
- Motion of each particle is connected to the motion of all other particles
- Number of particles in systems we are interested in is enormous



From individuals to society: statistical approach and distribution function

Fusion applications: single particle motion not interesting. Rather the evolution of macroscopic behaviour. This can be calculated if we know

$$f(\mathbf{r}, \mathbf{v}, t), \tag{1}$$

the *probability density* at time t to find a particle in phase space volume element (d^3r, d^3v).
Usually, $\int f d^3r d^3v = N$ and it is called the *distribution function*.
Each particle moves in a fluctuating field produced by the other particles
$\rightarrow$ small-angle collisions
$\rightarrow$ essentially Brownian motion
$\rightarrow$ probabilistic view w/ step-like Markov processes
$\rightarrow$ *Fokker-Planck equation*

Note: there are situations where the Fokker-Planck approach is not valid but one has to include strong head-on collisions as well. For example dense and cold plasmas in white dwarfs.

NOTE: The issues presented here involve not only some tedious mathematics but also several clever tricks that are not obvious. Most of these can be found in the dissertation of Eero Hirvijoki [EHD2014] and references therein.

Here we wish to only outline the procedures.

Outline

1 Fokker-Planck theory and stochastic motion
- Motivation
- Derivation of Fokker-Planck equation
- Deboning the FP equation, Part I: the RHS
- Deboning the FP equation, Part II: the LHS
- Equivalence to a stochastic differential equation

How to describe evolution of $f(\mathbf{z}, t)$ mathematically:

Consider a single particle:

- First, the particle is found at $\mathbf{z} = (\mathbf{r}, \mathbf{v})$
- An instant later, the particle is found at $\mathbf{z} + \Delta$
- Transition probability from $\mathbf{z}$ to $\mathbf{z} + \Delta$ during a time τ described by $W_\tau(\mathbf{z}, \Delta)$, with normalization $\int W_\tau(\mathbf{z}, \Delta) d\Delta = 1$

The probability density for finding the particle at $\mathbf{z}$ after time τ is

$$f(\mathbf{z}, t + \tau) = \int d\Delta \; f(\mathbf{z} - \Delta, t) W_\tau(\mathbf{z} - \Delta, \Delta), \tag{2}$$

Assume small steps due to small-angle Coulomb collisions:

1. Taylor expand f and W_τ around $\mathbf{z}$

$$f(\mathbf{z}, t+\tau) = \int d\Delta \left[f(\mathbf{z},t) W_\tau(\mathbf{z},\Delta) - \frac{\partial}{\partial \mathbf{z}} \left(f(\mathbf{z},t) W_\tau(\mathbf{z},\Delta) \right) \cdot \Delta \right.$$
$$\left. + \frac{1}{2} \frac{\partial}{\partial \mathbf{z}} \frac{\partial}{\partial \mathbf{z}} \left(f(\mathbf{z},t) W_\tau(\mathbf{z},\Delta) \right) : \Delta\Delta + \mathcal{O}(\Delta\Delta\Delta) \right].$$

2. Rearrange terms

$$\frac{f(\mathbf{z},t+\tau) - f(\mathbf{z},t)}{\tau} = - \frac{\partial}{\partial \mathbf{z}} \cdot \left(f(\mathbf{z},t) \frac{\langle \Delta \rangle}{\tau} \right)$$
$$+ \frac{1}{2} \frac{\partial}{\partial \mathbf{z}} \frac{\partial}{\partial \mathbf{z}} : \left(f(\mathbf{z},t) \frac{\langle \Delta\Delta \rangle}{\tau} \right) + \mathcal{O}\left(\frac{\langle \Delta\Delta\Delta \rangle}{\tau} \right),$$

Here the expectation value with respect to transition probability is

$$\langle \dots \rangle = \int d\Delta W_\tau(\mathbf{z},\Delta) \dots$$

Include only small-angle collisions:

In fusion plasmas, $\frac{\langle \Delta \rangle}{\tau}$ and $\frac{\langle \Delta\Delta \rangle}{\tau}$ dominate higher order terms by a factor $\ln \Lambda$ (# of particles within the Debye-sphere $>> 1$):

$$\frac{\partial}{\partial t} f(\mathbf{z}, t) = -\frac{\partial}{\partial \mathbf{z}} \cdot [(\dot{\mathbf{z}} + \mathbf{a}(\mathbf{z}, t)) f(\mathbf{z}, t)] + \frac{\partial}{\partial \mathbf{z}} \frac{\partial}{\partial \mathbf{z}} : [\mathbf{D}(\mathbf{z}, t) f(\mathbf{z}, t)], \tag{3}$$

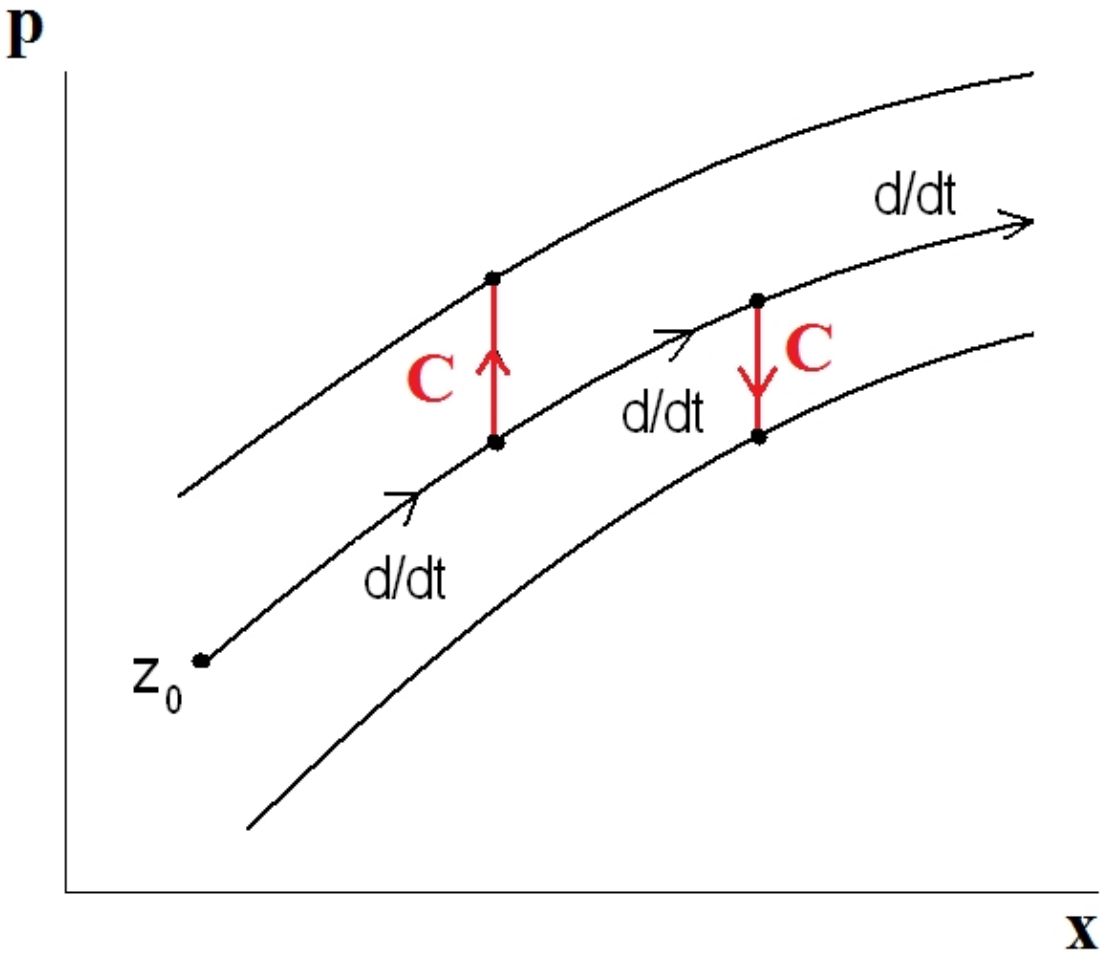
where the Hamiltonian part $\dot{\mathbf{z}}$ arises from the deterministic motion, $W_\tau = \delta(\dot{\mathbf{z}}\tau - \Delta)$, and the collisional friction (or drag) vector and diffusion tensor (the Fokker-Planck coefficients) are

$$\mathbf{a}(\mathbf{z}, t) = \lim_{\tau \to 0} \frac{\langle \Delta \rangle}{\tau}, \tag{4}$$

$$\mathbf{D}(\mathbf{z}, t) = \lim_{\tau \to 0} \frac{\langle \Delta\Delta \rangle}{2\tau}. \tag{5}$$

Careful: $\ln \Lambda$ *determines the* **validity** *of the Fokker-Planck theory. Theory completely breaks down when* $\ln \Lambda \to 1$. **Error** *is proportional to* $1/\ln \Lambda$

The effect of collisions in particle space



(Courtesy of Alain Brizard)

Conventional form of the FP equation

The collisional terms, $\mathbf{a}$ and $\mathbf{D}$, are often written separately from the Hamiltonian contribution:

$$\frac{\partial}{\partial t} f + \frac{\partial}{\partial \mathbf{z}} \cdot (\dot{\mathbf{z}} f) = C[f], \tag{6}$$

and the right-hand-side, $C[f]$, is called the *collision operator*

$$C[f] = -\frac{\partial}{\partial \mathbf{z}} \cdot \left[\mathbf{a} f - \frac{\partial}{\partial \mathbf{z}} \cdot (\mathbf{D} f) \right] \equiv -\frac{\partial}{\partial \mathbf{z}} \cdot \mathbf{J}, \tag{7}$$

and $\mathbf{J}$ is the collisional flux density.

Note#1: This is *physically* well motivated
Note#2: The divergence form of the collision operator guarantees conservation of particles.

Outline

1 Fokker-Planck theory and stochastic motion
- Motivation
- Derivation of Fokker-Planck equation
- Deboning the FP equation, Part I: the RHS
- Deboning the FP equation, Part II: the LHS
- Equivalence to a stochastic differential equation

Particle Fokker-Planck coefficients $\mathbf{a}$ and $\mathbf{D}$

The *particle* phase space Coulomb collision operator acts only on $\mathbf{v}$:

$$C\left[f_a\right] = \sum_b C_{ab}\left[f_a, f_b\right] = \sum_b -\frac{\partial}{\partial \mathbf{v}} \cdot \left[\mathbf{a}_{ab} f_a - \frac{\partial}{\partial \mathbf{v}} \cdot \left(\mathbf{D}_{ab} f_a\right)\right]. \tag{8}$$

The friction and diffusion coefficients $\mathbf{a}$ and $\mathbf{D}$ are [e.g., Ichimaru's book]

$$\mathbf{a}_{ab} = \frac{c_{ab}}{m_a^2}\left(1 + \frac{m_a}{m_b}\right)\frac{\partial h_b}{\partial \mathbf{v}}, \quad \mathbf{D}_{ab} = \frac{1}{2}\frac{c_{ab}}{m_a^2}\frac{\partial}{\partial \mathbf{v}}\frac{\partial}{\partial \mathbf{v}} g_b, \tag{9}$$

where a and b refer to different species and $c_{ab} = q_a^2 q_b^2 \ln\Lambda/\epsilon_0$.
The *Rosenbluth potentials*, h_b and g_b, are defined

$$h_b(\mathbf{z}) = \int d\mathbf{z}' \delta(\mathbf{x} - \mathbf{x}') f_b(\mathbf{z}') \frac{1}{|\mathbf{v} - \mathbf{v}'|}, \tag{10}$$

$$g_b(\mathbf{z}) = \int d\mathbf{z}' \delta(\mathbf{x} - \mathbf{x}') f_b(\mathbf{z}') \left|\mathbf{v} - \mathbf{v}'\right|. \tag{11}$$

These coefficients have nice physical properties:

$$\dot{\mathbf{p}}_{ab} = -\dot{\mathbf{p}}_{ba} \tag{12}$$

$$\dot{\mathbf{p}}_{aa} = 0, \tag{13}$$

$$\dot{E}_{ab} + \dot{E}_{ba} = 0 \tag{14}$$

$$\dot{E}_{aa} = 0 \tag{15}$$

$$\tag{16}$$

FP coefficients in isotropic background plasma

Often the background plasma is assumed to have isotropic velocity dependence, $f_b(\mathbf{z}) = f_b(\mathbf{x}, v) \rightarrow$ coefficients become simpler:

$$\mathbf{a}_{ab} = -\left(1 + \frac{m_b}{m_a}\right) \nu_{ab}\, \mathbf{v}, \tag{17}$$

$$\mathbf{K}_{ab} = -\,\nu_{ab}\, \mathbf{v}, \tag{18}$$

$$\mathbf{D}_{ab} = D_{\parallel,ab} \frac{\mathbf{vv}}{v^2} + D_{\perp,ab} \left(\mathbf{I} - \frac{\mathbf{vv}}{v^2}\right), \tag{19}$$

where the scalar coefficients are defined

$$\nu_{ab} = -\,\frac{c_{ab}}{m_a^2} \frac{m_a}{m_b} \frac{1}{v} h_b'(v), \tag{20}$$

$$D_{\parallel,ab} = \frac{1}{2} \frac{c_{ab}}{m_a^2} g_b''(v), \tag{21}$$

$$D_{\perp,ab} = \frac{1}{2} \frac{c_{ab}}{m_a^2} \frac{1}{v} g_b'(v). \tag{22}$$

Symbols $\parallel$ and $\perp$ are wrt to the *local* magnetic field direction.

FP coefficients for Maxwellian plasma:

Background in thermal equilibrium, (typical of today's tokamaks):

$$f_b = \frac{n_b}{\pi^{3/2} v_b^3} \exp\left(-v^2/v_b^2\right), \tag{23}$$

the diffusion and friction coefficients become:

$$D_{\parallel,ab}(v) = \frac{1}{2}\frac{c_{ab}}{m_a^2}\frac{n_b}{v} G(v/v_b), \tag{24}$$

$$D_{\perp,ab}(v) = \frac{1}{2}\frac{c_{ab}}{m_a^2}\frac{n_b}{v}\left(\mathrm{erf}(v/v_b) - \frac{1}{2}G(v/v_b)\right), \tag{25}$$

$$\nu_{ab}(v) = \frac{c_{ab}}{m_a^2}\frac{m_a}{m_b}\frac{n_b}{v_b^2}\frac{G(v/v_b)}{v}. \tag{26}$$

where the Chandrasekhar function $G(x)$ is defined as

$$G(x) = \frac{\mathrm{erf}(x) - \frac{2x}{\sqrt{\pi}}\exp(-x^2)}{x^2}. \tag{27}$$

Outline

1 Fokker-Planck theory and stochastic motion
- Motivation
- Derivation of Fokker-Planck equation
- Deboning the FP equation, Part I: the RHS
- Deboning the FP equation, Part II: the LHS
- Equivalence to a stochastic differential equation

Hamiltonian part of the FP equation:

One could simply use the Lorenz force to write $\dot{\mathbf{z}}$.
However, with the guiding-center (GC) formalism in mind, we are better off starting rigorously with Hamiltonian formalism.

Recalling basic stuff from classical mechanics:

- the dynamics of a system is contained in the system *Lagrangian*, γ
- in 1D, $\gamma = p \cdot dq - H \cdot dt$
- the equations of motion are obtained by minimizing the *action integral*, $\int \gamma$

Path of least action in 3D:

$\mathbf{z} = (\mathbf{r}, \mathbf{v})$, with components z^α
Minimizing the action in 3D gives

$$\frac{\partial z^\alpha}{\partial t} = \{z^\alpha, H\} \tag{28}$$

where on the RHS we have the *Poisson bracket*,

$$\{f, g\} \equiv \frac{\partial f}{\partial z^\alpha} \Pi^{\alpha\beta} \frac{\partial g}{\partial z^\beta}. \tag{29}$$

and $\Pi^{\alpha\beta}$ is the *Poisson matrix* given as the inverse of the *Lagrange matrix*

$$\omega_{\alpha\beta} = \frac{1}{2}\left(\frac{\partial \gamma_\beta}{\partial z^\alpha} - \frac{\partial \gamma_\alpha}{\partial z^\beta}\right), \tag{30}$$

In the presence of electric and magnetic fields $E = -\nabla\Phi - \partial\mathbf{A}/\partial t$, $B = \nabla \times \mathbf{A}$, the Hamiltonian and Lagrangian are

$$H = \frac{1}{2}mv^2 + q\Phi \tag{31}$$

$$\gamma = (m\mathbf{v} + q\mathbf{A}) \cdot \mathrm{d}\mathbf{x} - H\mathrm{d}t, \tag{32}$$

and the Poisson bracket becomes

$$\{f, g\} = \frac{1}{m}\left(\nabla f \cdot \frac{\partial g}{\partial \mathbf{v}} - \frac{\partial f}{\partial \mathbf{v}} \cdot \nabla g\right) + \frac{q\mathbf{B}}{m^2} \cdot \frac{\partial f}{\partial \mathbf{v}} \times \frac{\partial g}{\partial \mathbf{v}}. \tag{33}$$

Then it is easy to show that

$$\frac{\partial z^\alpha}{\partial t} = \{z^\alpha, H\} \tag{34}$$

is equivalent to

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{v} \qquad \frac{\partial \mathbf{v}}{\partial t} = \frac{q}{m}(E + \mathbf{v} \times B) \tag{35}$$

Note

The Poisson brackets play a crucial role when the GC formalism is derived using Lie transformations. Thus the silly-appearing effort.

Outline

1 Fokker-Planck theory and stochastic motion
- Motivation
- Derivation of Fokker-Planck equation
- Deboning the FP equation, Part I: the RHS
- Deboning the FP equation, Part II: the LHS
- Equivalence to a stochastic differential equation

Solving the FP equation I: straightforward approach

Use finite difference, finite element (FEM) methods $\rightarrow$

- In the full 6D phase space, this leads to enormous matrices
- $\rightarrow$ mostly applicable when the # of coordinates can be brought down to 3 with various assumptions and approximations (axisymmetry, flux-surface averaging, orbit averaging, ...)

Solving the FP equation II: Monte Carlo method

Kolmogorov: FP equation is equivalent to a stochastic differential equation!

(The proof can be found in [EHD2014])

first: Simulate a large number of test particle trajectories according the stochastic differential equation

then: Construct the distribution function as a statistical average of the simulated trajectories

Langevin equation for a test particle

The Fokker-Planck equation

$$\frac{\partial}{\partial t} f(\mathbf{z}, t) = -\frac{\partial}{\partial \mathbf{z}} \cdot [(\dot{\mathbf{z}} + \mathbf{a}(\mathbf{z}, t)) f(\mathbf{z}, t)] + \frac{\partial}{\partial \mathbf{z}} \frac{\partial}{\partial \mathbf{z}} : [\mathbf{D}(\mathbf{z}, t) f(\mathbf{z}, t)] \,,$$

is equivalent to a stochastic differential equation, the *Langevin equation*

$$d\mathbf{z} \;=\; [\dot{\mathbf{z}} + \mathbf{a}(\mathbf{z}, t)]\, dt + \boldsymbol{\sigma} \cdot d\boldsymbol{\beta}, \tag{36}$$

where the matrix $\boldsymbol{\sigma}$ is defined via a decomposition of the diffusion tensor

$$2\mathbf{D} = \boldsymbol{\sigma}\boldsymbol{\sigma}^T. \tag{37}$$

and the stochastic differential $d\boldsymbol{\beta}$ denotes an infinitesimal change in the random variable $\boldsymbol{\beta}$ which has zero mean and variance t. (The upper index ^{T} denotes a transpose of a matrix.)

Getting ready to simulate **minority** test particles

Neglect self-collisions, $C_{aa}\left[f_a, f_a\right]$, and assume isotropic background populations $\rightarrow$ diffusion tensor straightforward to decompose for $\boldsymbol{\sigma} \rightarrow$

$$d\mathbf{v} = \left[\dot{\mathbf{v}} - \nu\mathbf{v}\right] dt + \sqrt{2D_{\parallel}}\frac{\mathbf{vv}}{v^2} + \sqrt{2D_{\perp}}\left(\mathbf{I} - \frac{\mathbf{vv}}{v^2}\right), \tag{38}$$

$$d\mathbf{x} = \dot{\mathbf{x}}dt, \tag{39}$$

where

- $\nu = \sum_b (1 + m_b/m_a)\nu_{ab}$, $D_{\parallel} = \sum_b D_{\parallel,ab}$, and $D_{\perp} = \sum_b D_{\perp,ab}$,
- and the Hamiltonian equations of motion are

$$\dot{\mathbf{x}} = \mathbf{v}, \quad \dot{\mathbf{v}} = (q/m)\left(\mathbf{E} + \mathbf{v} \times \mathbf{B}\right). \tag{40}$$

And we are done:

– **Simulate particles with the stochastic differential equation.**

– **Construct the distribution function from the trajectories.**

Integrating Langevin equation

For strongly peridodic motion (such as a particle gyrating around a magnetic field), Runge-Kutta (RK) is not an ideal choice since it accumulates error.

For ASCOT, leap-from Boris, familiar from PIC method, is used to follow gyro orbits for long times (slowing-down times for MeV-range particles)

Typically, the entire collision term is integrated with the simple Euler method. This is for historical reasons, but could be discussed:

- for consistency, all the deterministic terms (including friction) in the Langevin equation should be integrated with same accuracy?
- on the other hand, the collision term does not have the Hamiltonian nature ...

In ASCOT, the Wiener process simply applies the 'binary method', (-1,+1), but other choices exist.

Outline

1 Fokker-Planck theory and stochastic motion

2 Guiding center formalism

3 High Performance computing

4 Applying the formalism – if time permits

5 Summary

Outline

2 Guiding center formalism
- Motivation
- About Lie-transformations
- Guiding center Fokker-Planck equation

In many situations, magnetic field and background plasma can be considered constant across the Larmor orbit,

$$\rho_L << B/\nabla B, \tag{41}$$

$$\rho_L << n/\nabla n, T/\nabla T \tag{42}$$

Why waste computer resources on the trivial gyro motion???
$\rightarrow$ get rid of the gyration in the equations!

First approach by Alfvén, Nortrup and others before 1970's:

- expand the magnetic field in Taylor series around *guiding center*
- average the Lorenz force law over the gyro angle

Problems:

- *The equations of motion do not conserve energy...*
- *How to average out the larmor motion from the collision operator...?*

Guiding center vs. particle

- Guiding center often misunderstood as the average of the particle position over a single Larmor rotation, i.e., $\mathbf{X} \approx \int_0^{2\pi} \mathbf{x}\, d\zeta/(2\pi)$.
- However, rigorous definition is a **coordinate transformation**

$$\mathbf{x} = \mathbf{X} + \boldsymbol{\rho},$$

where $\boldsymbol{\rho}$ is the vector from the guiding center position to the particle position. Its length, ρ, is called the Larmor radius.

- In principle, for any function we have simple transformation rules

$$f(\mathbf{x}) = f(\mathbf{X} + \boldsymbol{\rho}) = \sum_n \frac{1}{n!}(\boldsymbol{\rho} \cdot \nabla)^n f(\mathbf{X}) = \exp(\boldsymbol{\rho} \cdot \nabla) f(\mathbf{X}) \equiv F(\mathbf{X}),$$

$$F(\mathbf{X}) = F(\mathbf{x} - \boldsymbol{\rho}) = \sum_n \frac{1}{n!}(-\boldsymbol{\rho} \cdot \nabla)^n F(\mathbf{x}) = \exp(-\boldsymbol{\rho} \cdot \nabla) F(\mathbf{x}) \equiv f(\mathbf{x})$$

But how to define $\boldsymbol{\rho}$, i.e., the transformation?

- In the **push-forward** (*pushes particle mappings to guiding center mappings*), $f(\mathbf{x}) = \exp(\boldsymbol{\rho} \cdot \nabla) f(\mathbf{X}) = F(\mathbf{X})$, we have $\boldsymbol{\rho}$ evaluated at the guiding center position $\mathbf{X}$.
- In the **pull-back** (*pulls guiding center mappings back to particle mappings*), $F(\mathbf{X}) = \exp(-\boldsymbol{\rho} \cdot \nabla) F(\mathbf{x}) = f(\mathbf{x})$, we have instead $\boldsymbol{\rho}$ evaluated at the particle position $\mathbf{x}$.
- So, do we define $\boldsymbol{\rho} = -m\mathbf{v} \times \mathbf{B}/(qB^2)$ and evaluate it in the proper place, regarding whether we are discussing pull-back or push-forward?

Result: *integrating the particle orbit in* **nonuniform** $\mathbf{B}$ *and evaluating the guiding center position accoding to* $\mathbf{X} = \mathbf{x} + m\mathbf{v} \times \mathbf{B}/(qB^2)$ *gives a point that oscillates at the same frequency as the particle orbit, i.e., the motion of* $\mathbf{X}$ *would not be* **independent of the gyrating motion**.

Outline

2 Guiding center formalism
- Motivation
- About Lie-transformations
- Guiding center Fokker-Planck equation

Dr. Lie comes to rescue!

The rapid gyro motion can be eliminated formally using the *Lie transform perturbation* method.

Basic idea:
Find a transformation of coordinates such that, in the new coordinates, the transformed Lagrangian is *independent* of the gyro angle.

Things to keep in mind:

- The gyro angle can still be resolved but it is not necessary.
- In the FP equation, the collisional part will still contain the gyro angle$\rightarrow$ has to be averaged

Here, the proceduce will only be outlined.
The math is tedious and *not* straightforward
$\rightarrow$ Those interested in details, see [EHD2014).

Basic concepts of the Lie Transformation

Lie transformation is defined by *pull-back* and *push-forward* operators:

- $\mathcal{T}_n = \exp(\epsilon^n \mathcal{L}_{G_n})$
- $\mathcal{T}_n^{-1} = \exp(-\epsilon^n \mathcal{L}_{G_n})$

where

- ϵ is a smallness parameter, giving the order of the *near-identity* transformation. Thus the term *'perturbation'*.
- $\mathcal{L}_{G_n}$ is the so-called *Lie derivative*, generated by a vector field G_n.
  - *Lie-derivative of a function reduces to* $\mathcal{L}_{G_n} F = G_n^\alpha \partial F / \partial Z^\alpha$ *so that* $\exp(\epsilon^n \mathcal{L}_{G_n}) F = \exp(\epsilon^n G_n^\alpha \partial / \partial Z^\alpha) F$. *A Lie-transformation is thus closely related to the original idea* $f(\mathbf{x}) = \exp(\boldsymbol{\rho} \cdot \nabla) f(\mathbf{X})$.

We start carrying out the transformation of both coordinates and the Lagrangian not knowing what the transformation is.

The generating functions are determined by requiring that the transformed Lagrangian is independent of the gyro angle.

Outline of the procedure

The transformation $\mathcal{T}_{gc} : z^\alpha \to Z^\alpha$ changes particle coordinates $z^\alpha = (x, v)$ to GC coordinates $Z^\alpha = (X, v_\parallel, \mu, \zeta)$, where

- X = coordinates of the guiding-center position
- $v_\parallel$ = the velocity parallel to the magnetic field
- ζ = the gyro angle

The *near-identity* transformation (ρ_L small) can be expanded

$$\mathcal{T}_{gc} = 1 + \epsilon\mathcal{L}_{G_1} + \epsilon^2(\mathcal{L}_{G_2} + \frac{1}{2}\mathcal{L}^2_{G_1}) \tag{43}$$

and the transformed Lagrangian Γ becomes

$$\Gamma(Z^\alpha) = \mathcal{T}^{-1}_{gc}(\epsilon^{-1}\gamma_0 + \gamma_1) + dS = (\epsilon^{-1}\Gamma_0 + \Gamma_1 + \epsilon\Gamma_2 + ...) \tag{44}$$

where dS is a *gauge function* for clean up.

Require that each Γ_n is independent of ζ
$\to$ $\mathbf{G_n}$'s determined!

Guiding-center Lagrangian & friends

After a fair amount of algebra and very clever thinking [EHD2014]:

$$\Gamma = \left(\epsilon^{-1} q\mathbf{A} + m v_{\parallel}\widehat{\mathbf{b}} - \epsilon\frac{m\mu}{q}\boldsymbol{R}^{\star}\right)\cdot \mathrm{d}\mathbf{X} + \epsilon\frac{m\mu}{q}\mathrm{d}\zeta - H_{gc}\mathrm{d}t, \qquad (45)$$

where $\mu = \frac{m v_{\perp}^2}{2B}$ is the magnetic moment, $H_{gc} = \frac{1}{2} m v_{\parallel}^2 + \mu B$ is the *guiding center Hamiltonian*, and $\boldsymbol{R}^{\star} = \boldsymbol{R} + (\tau/2)\widehat{\mathbf{b}}$, where $\boldsymbol{R}$ *Littlejohn's gyrogauge field* with $\tau = \widehat{\mathbf{b}}\cdot\nabla\times\widehat{\mathbf{b}}$ being the magnetic field-line twist.

The GC Lagrangian allows the following construction project:

- construct $\omega_{\alpha,\beta}$
- $\rightarrow \Pi^{\alpha,\beta}$
- $\rightarrow \{F, G\}_{gc}$
- $\rightarrow \dot{Z} = \{Z, H_{gc}\}_{gc}$, the equations of motion!

An example of the Guiding center Poisson bracket

In the absence of time varying fields using phase space coordinates $Z^\alpha = (\mathbf{X}, v_\parallel, \mu, \theta)$, with $v_\parallel$ the guiding center velocity parallel to the magnetic field, μ the magnetic moment, and ζ the gyro-angle:

- Poisson bracket:

$$\{F, G\}_{gc} = \epsilon^{-1} \frac{q}{m} \left(\frac{\partial F}{\partial \zeta} \frac{\partial G}{\partial \mu} - \frac{\partial F}{\partial \mu} \frac{\partial G}{\partial \zeta} \right) + \frac{\mathbf{B}^\star}{m B_\parallel^\star} \cdot \left(\nabla F \frac{\partial G}{\partial v_\parallel} - \frac{\partial F}{\partial v_\parallel} \nabla G \right) - \epsilon \frac{\widehat{\mathbf{b}}}{q B_\parallel^\star} \cdot (\nabla F \times \nabla G) .$$

- Hamiltonian:

$$\mathcal{H}_{gc} = m v_\parallel^2 / 2 + \mu B + q\Phi$$

Here $\mathbf{B}^\star = \mathbf{B} + \epsilon (m v_\parallel / q) \nabla \times \widehat{\mathbf{b}}$, $B_\parallel^\star = \mathbf{B}^\star \cdot \widehat{\mathbf{b}}$, and $\widehat{\mathbf{b}}$ is the magnetic field unit vector.

GC equations of motion

The equations in form ready to be programmed are

$$\dot{\mathbf{X}} = \{\mathbf{X}, H_{gc}\}_{gc} = v_{\parallel} \frac{\mathbf{B}^{\star}}{B^{\star}_{\parallel}} + \epsilon \frac{\widehat{\mathbf{b}}}{qB^{\star}_{\parallel}} \times \mu \nabla B, \tag{46}$$

$$\dot{v_{\parallel}} = \{v_{\parallel}, H_{gc}\}_{gc} = -\frac{\mu}{m} \frac{\mathbf{B}^{\star}}{B^{\star}_{\parallel}} \cdot \nabla B, \tag{47}$$

$$\dot{\mu} = \{\mu, H_{gc}\}_{gc} = 0, \tag{48}$$

$$\dot{\zeta} = \{\zeta, H_{gc}\}_{gc} = \epsilon^{-1}\Omega + \dot{\mathbf{X}} \cdot \boldsymbol{R}^{\star}, \tag{49}$$

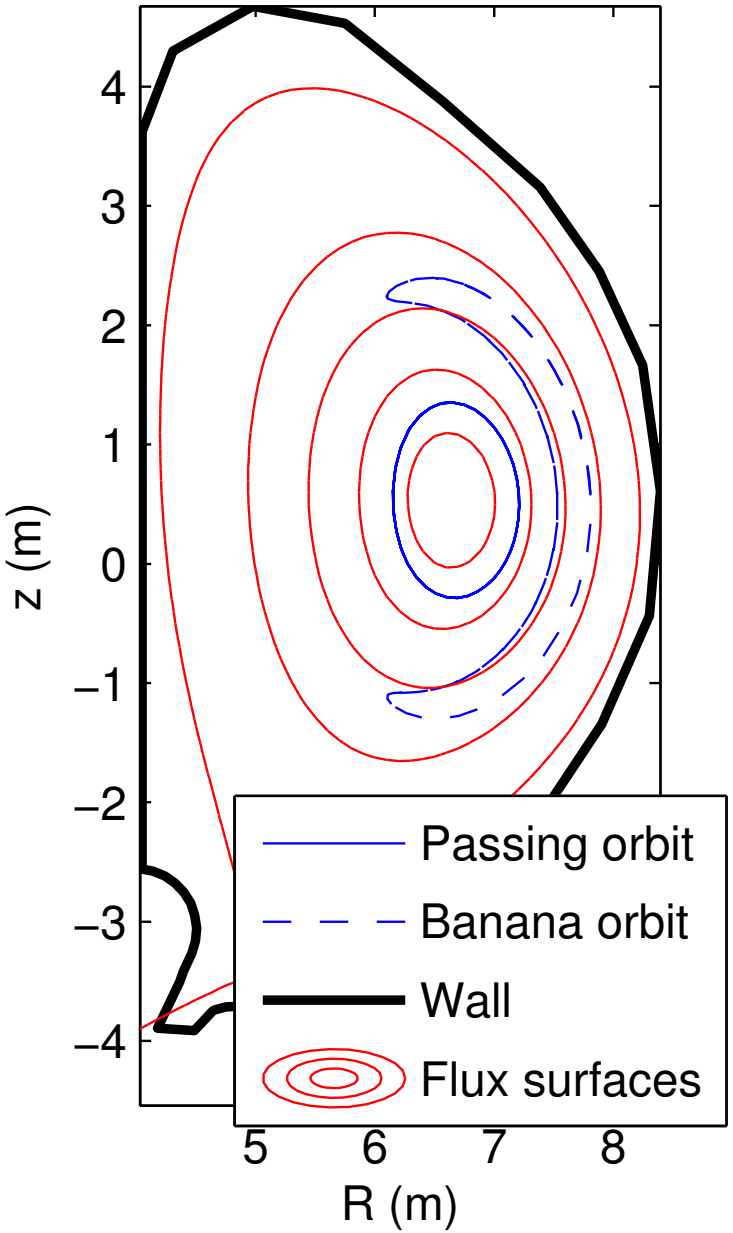
Note:

- $X, v_{\|}, \mu$ do not depend on ζ
- μ is a constant of motion in GC formalism
- we haven't lost any information. The gyro angle ζ can still be followed if so desired.

Drift orbits

We now have a powerful tool that allows:

- to follow the *drift orbits*, traced by the guiding centers of a test particle.



Outline

2 Guiding center formalism
- Motivation
- About Lie-transformations
- Guiding center Fokker-Planck equation

Transforming the FP equation

So far only LHS = the Hamiltonian part of the FP equation transformed. How to transform the entire FP equation, including the collision operator? Answer: *the Poisson brackets*!

The velocity derivative $\partial/\partial v^i$ can be written in terms of Poisson brackets:

$$\frac{\partial f}{\partial v^i} = \frac{1}{m}\{x^i, \mathcal{H}\} \tag{50}$$

$\rightarrow$ Write the FP equation in terms of Poisson brackets and use the known transformation $\{f, g\} \rightarrow \{F, G\}_{gc}$

$$\frac{\partial f}{\partial t} + \{f, \mathcal{H}\} = \{x^i, mK^i f - m^2 D^{ij}\{x^j, f\}\}. \tag{51}$$

Here we have used $\mathbf{K} = \mathbf{a} - \frac{\partial}{\partial \mathbf{v}} \cdot \mathbf{D}$ for practical reasons.

Carrying out the GC transformation:

$$\frac{\partial F}{\partial t} + \{F, H_{gc}\}_{gc} = \{T_{gc}^{-1} x^i, m(T_{gc}^{-1} K^i)F - m^2 (T_{gc}^{-1} D^{ij})\{T_{gc}^{-1} x^j, F\}_{gc}\}_{gc}, \tag{52}$$

where where $F(Z^\alpha)$ is the transformed distribution function in transformed coordinates.

Towards Langevin equation in GC coordinates

To find the corresponding Langevin equation, use the divergence form:

$$\frac{\partial F}{\partial t} + \dot{Z}^{\alpha}\frac{\partial F}{\partial Z^{\alpha}} = -\frac{1}{\mathcal{J}}\frac{\partial}{\partial Z^{\alpha}}\left[\mathcal{J}\left(m\mathcal{K}^{\alpha}F - m^{2}\mathcal{D}^{\alpha\beta}\frac{\partial F}{\partial Z^{\beta}}\right)\right] = C_{gc}[F], \quad (53)$$

where the GC friction and diffusion coefficients, $\mathcal{K}^{\alpha}$ and $\mathcal{D}^{\alpha\beta}$, are

$$\mathcal{K}^{\alpha} = (\mathcal{T}_{gc}^{-1}\mathbf{K})\cdot\mathbf{\Delta}^{\alpha}, \quad (54)$$

$$\mathcal{D}^{\alpha\beta} = (\mathbf{\Delta}^{\alpha})^{\dagger}\cdot(\mathcal{T}_{gc}^{-1}\mathbf{D})\cdot\mathbf{\Delta}^{\beta}, \quad (55)$$

and $\mathbf{\Delta}^{\alpha}$ are so-called projection vectors [EHD2014].

Eliminating the last traces of ζ

The GC distribution function $F(Z^\alpha)$ and the friction and diffusion coefficients still have ζ dependence $\rightarrow$ FP has to be averaged over ζ. This tedious job can be found in [EHD2014], yielding

$$\frac{\partial \mathcal{F}}{\partial t} = -\frac{1}{\mathcal{J}}\frac{\partial}{\partial \mathcal{Z}^\alpha}\left(\mathcal{J}\mathcal{A}^\alpha \mathcal{F}\right) + \frac{1}{\mathcal{J}}\frac{\partial^2}{\partial \mathcal{Z}^\alpha \partial \mathcal{Z}^\beta}\left(\mathcal{J}\mathcal{D}^{\alpha\beta}\mathcal{F}\right), \tag{56}$$

where the diffusion term was split into two with the coefficient

$$\mathcal{A}^\alpha = \dot{\mathcal{Z}}^\alpha + \mathcal{K}^\alpha + \frac{1}{\mathcal{J}}\frac{\partial}{\partial \mathcal{Z}^\beta}(\mathcal{J}\mathcal{D}^{\alpha\beta}). \tag{57}$$

containing both the Hamiltonian contribution and the deterministic part of the collision operator.

GC Langevin equation

Now the 'Kolmogorov connection' can be used $\rightarrow$

$$d\mathcal{Z}^{\alpha} = \mathcal{A}^{\alpha}_{gc} dt + \Sigma^{\alpha\beta}_{gc} d\mathcal{W}^{\beta}, \tag{58}$$

where $d\mathcal{W}^{\alpha}$ is again a Wiener process with zero mean and variance t, and $\Sigma^{\alpha\beta}_{gc}$ can be calculated from (NO LONGER EASY! [EHD2014])

$$\mathcal{D}^{\alpha\beta}_{gc} = \frac{1}{2} \Sigma^{\alpha\gamma}_{gc} \Sigma^{\beta\gamma}_{gc}, \tag{59}$$

GC collision operators for numerical implementation: Friction (drag)

$$\langle \mathcal{K}^{\mathbf{X}} \rangle = \epsilon\nu \frac{\widehat{\mathbf{b}}}{\Omega^{\star}_{\|}} \times \dot{\mathbf{X}} + \mathcal{O}(\epsilon^3), \tag{60}$$

$$\langle \mathcal{K}^{v_{\|}} \rangle = -\nu v_{\|} - \epsilon\lambda \frac{\mu B}{m v_{\|}} \nu + \mathcal{O}(\epsilon^2), \tag{61}$$

$$\langle \mathcal{K}^{\mu} \rangle = -(2 - \epsilon\lambda)\nu\mu + \mathcal{O}(\epsilon^2), \tag{62}$$

where $\lambda = v_{\|} \widehat{\mathbf{b}} \cdot \nabla \times \widehat{\mathbf{b}} / \Omega$

GC collision operators for numerical implementation: Diffusion

$$\langle \mathcal{D}^{\mathbf{XX}} \rangle = \epsilon^2 \left[(D_\parallel - D_\perp) \frac{\mu B}{2\mathcal{E}} + D_\perp \right] \frac{\mathbf{I} - \widehat{\mathbf{b}}\widehat{\mathbf{b}}}{(m\Omega_\parallel^\star)^2} + \mathcal{O}(\epsilon^3), \tag{63}$$

$$\langle \mathcal{D}^{v_\parallel v_\parallel} \rangle = \frac{D_\parallel}{m^2} + (1 - \epsilon\lambda) \frac{D_\perp - D_\parallel}{m^2} \frac{\mu B}{\mathcal{E}} + \mathcal{O}(\epsilon^2), \tag{64}$$

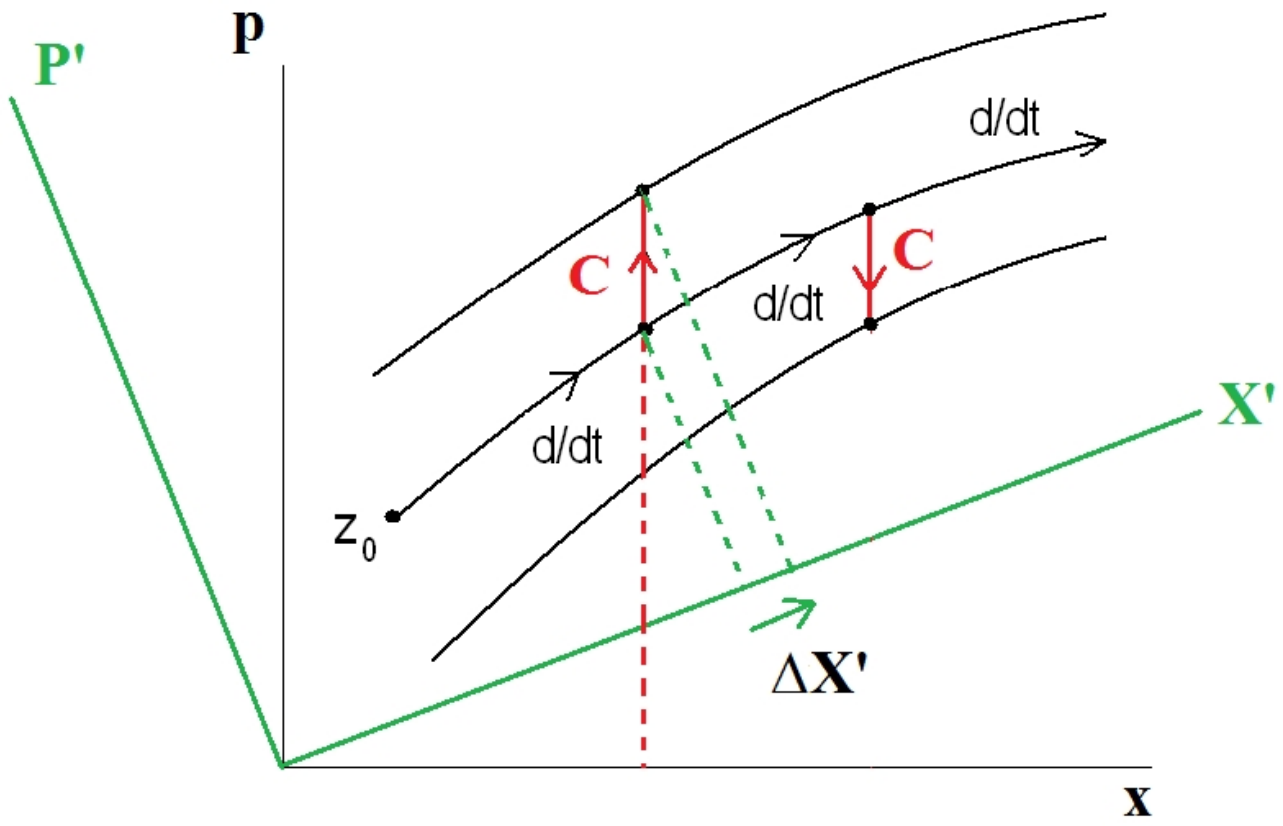
$$\langle \mathcal{D}^{\mu\mu} \rangle = (1 - \epsilon\lambda) \frac{2\mu}{mB} \left[(D_\parallel - D_\perp) \frac{\mu B}{\mathcal{E}} + D_\perp \right] + \mathcal{O}(\epsilon^2), \tag{65}$$

$$\begin{aligned} \langle \mathcal{D}^{\mathbf{X}v_\parallel} \rangle =& \epsilon^2 \frac{v_\parallel}{(m\Omega_\parallel^\star)^2} (D_\parallel - D_\perp) \frac{\mu B}{2\mathcal{E}} \nabla_\perp \ln B \\ &+ \epsilon^2 \frac{v_\parallel}{(m\Omega_\parallel^\star)^2} \left[D_\parallel + \frac{\mu B}{2\mathcal{E}} (D_\perp - D_\parallel) \right] \widehat{\mathbf{b}} \cdot \nabla \widehat{\mathbf{b}} + \mathcal{O}(\epsilon^3), \end{aligned} \tag{66}$$

$$\langle \mathcal{D}^{\mathbf{X}\mu} \rangle = -\epsilon \frac{\mu}{2m\mathcal{E}} (D_\parallel - D_\perp) \frac{\widehat{\mathbf{b}}}{\Omega_\parallel^\star} \times \dot{\mathbf{X}} + \mathcal{O}(\epsilon^3), \tag{67}$$

$$\langle \mathcal{D}^{\mu v_\parallel} \rangle = (1 - \epsilon\lambda) \frac{\mu v_\parallel}{m\mathcal{E}} (D_\parallel - D_\perp) + \epsilon\lambda \frac{\mu}{v_\parallel m^2} D_\parallel + \mathcal{O}(\epsilon^2), \tag{68}$$

Does your guiding center code include the spatial effects?

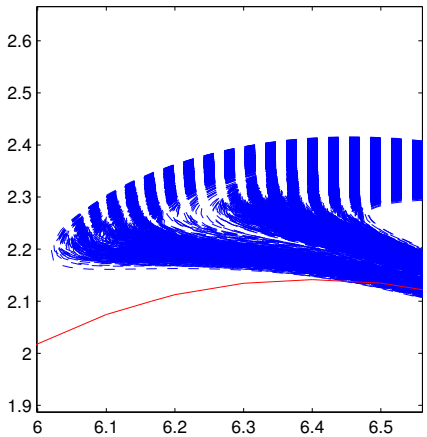


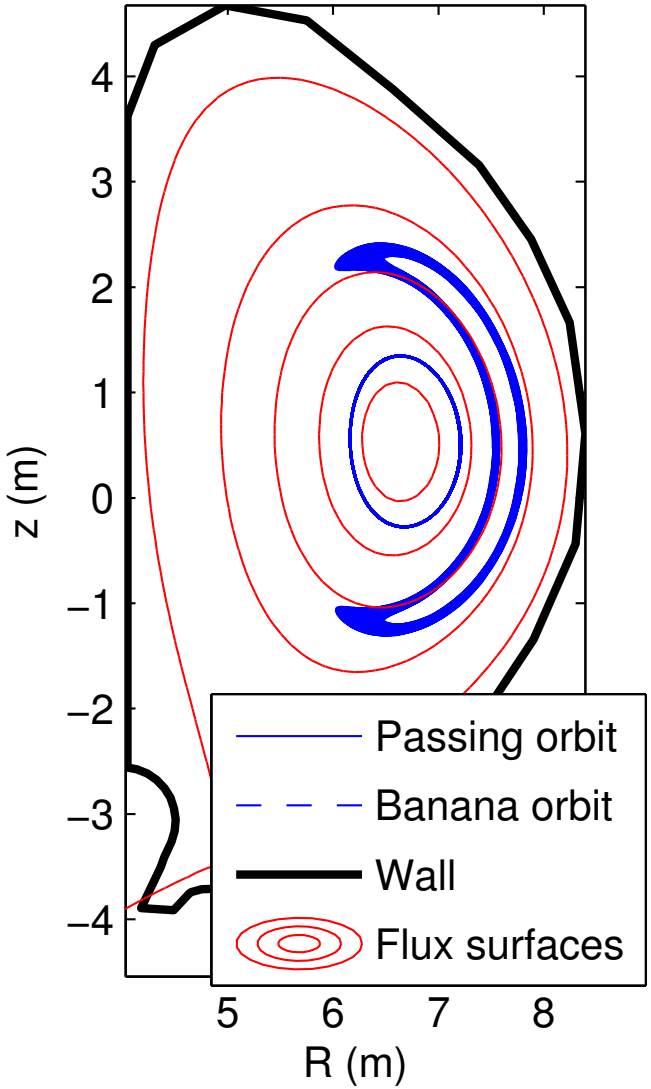
- Apparently, all GC codes today use particle phase space $C[f]$
- Proper GC collision operator includes the *spatial* drag and diffusion! (Recall: particle collisions only in $\mathbf{v}$)

Drift orbits and neoclassical transport

We now have a powerful tool that allows:

- to follow the *drift orbits*, traced by the guiding centers of a test particle.
- to simulate the *neoclassical transport*, caused by the combined effect of toroidal geometry and Coulomb collisions, of the test particles.





On the choice of integrator

Tokamak geometry exhibits a host of non-trivial orbits $\rightarrow$ the integrator should be reasonably accurate

Typical choice: 4^{th} order Runge-Kutta with 5^{th} order error checking.

Again, (in our opinion) the deterministic part = Hamiltonian motion + deterministic part of the collision operator, should be integrated at the same accuracy.

The stochastic part is integrated with simple Euler method.

Outline

1 Fokker-Planck theory and stochastic motion

2 Guiding center formalism

3 High Performance computing

4 Applying the formalism – if time permits

5 Summary

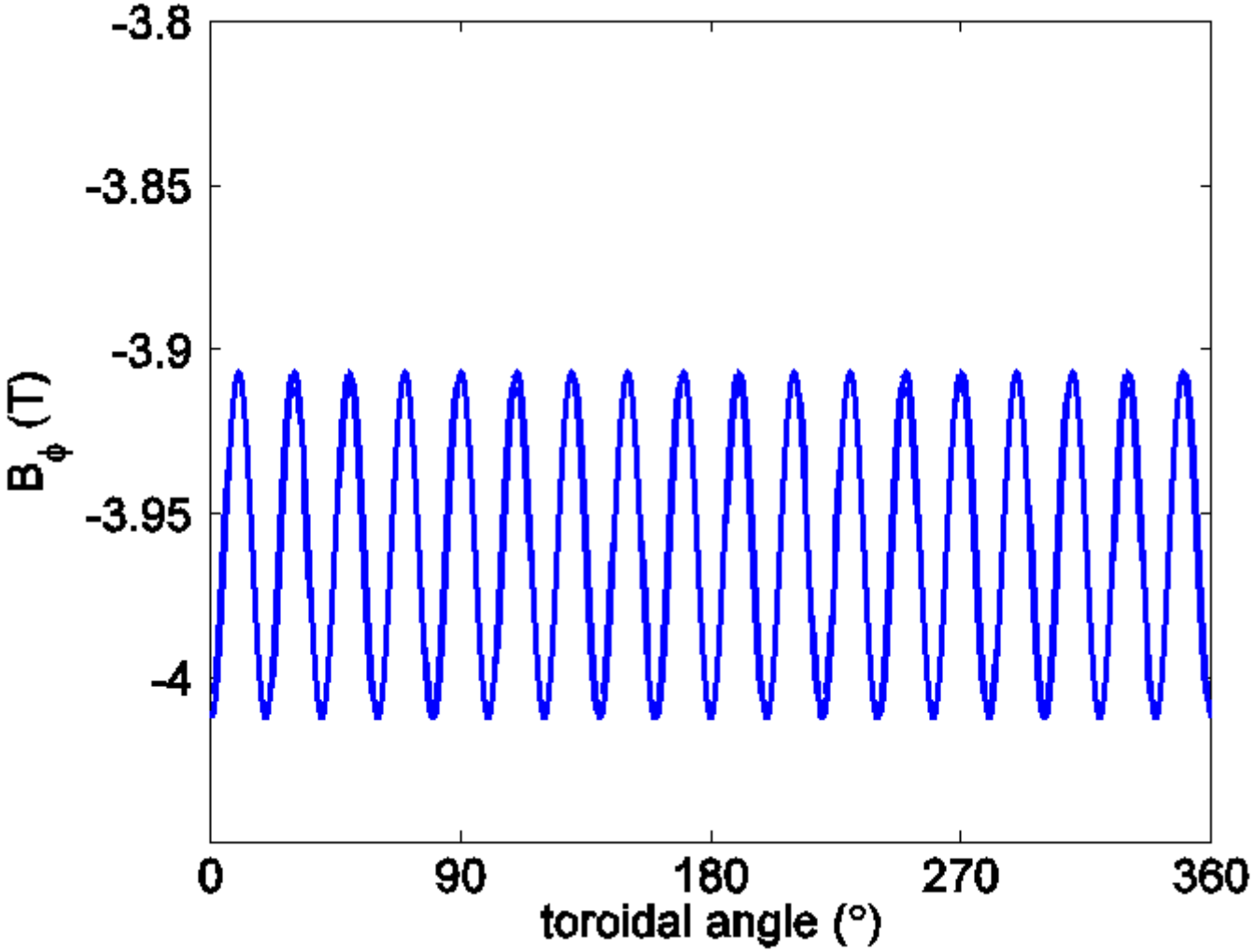
Test particles in tokamak magnetic field

In high performance plasmas, *magnetic field* dominates the test particle behaviour. How do we get the values of magnetic field and its derivative, needed for the equations of motion?

- t < 1990's: assume circularly symmetric plasma with simple limiter $\rightarrow$ analytical expression for $\mathbf{B}$.
  - very fast
  - very accurate
  - has very little to do with the reality of shaped and diverted plasmas
- 1990's – 2000's: introduce diverted geometry with non-circular cross section (ASCOT got experimental backgrounds in $\sim$1997.)
  - allows simulations in the SOL and divertor region
  - calculation slows down because 2D interpolation needed
  - choice of interpolation routine not trivial
  - still assumes axisymmetry

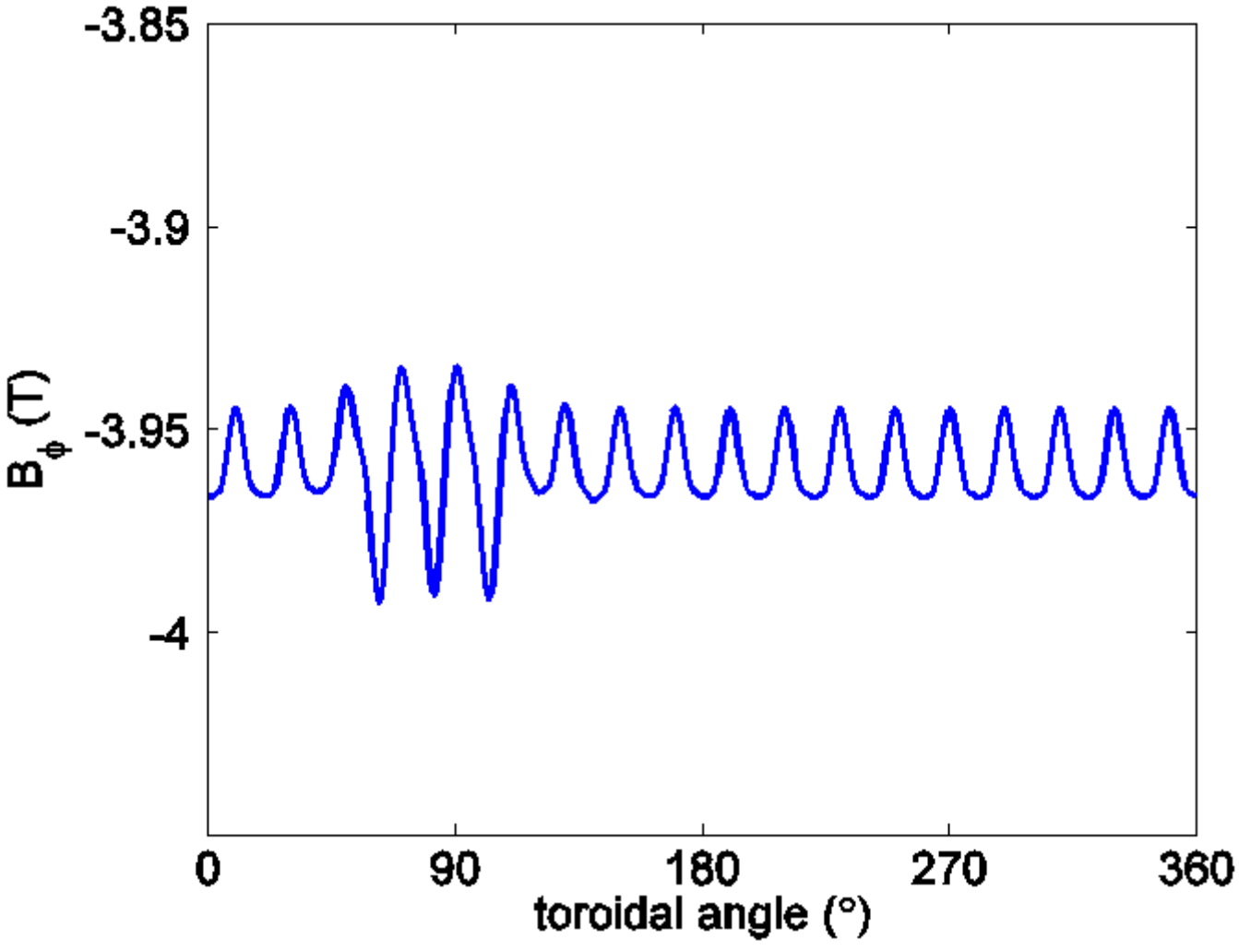
How about the still prevailing assumption of axisymmetry???

Magnetic field strength along ITER OMP separatrix



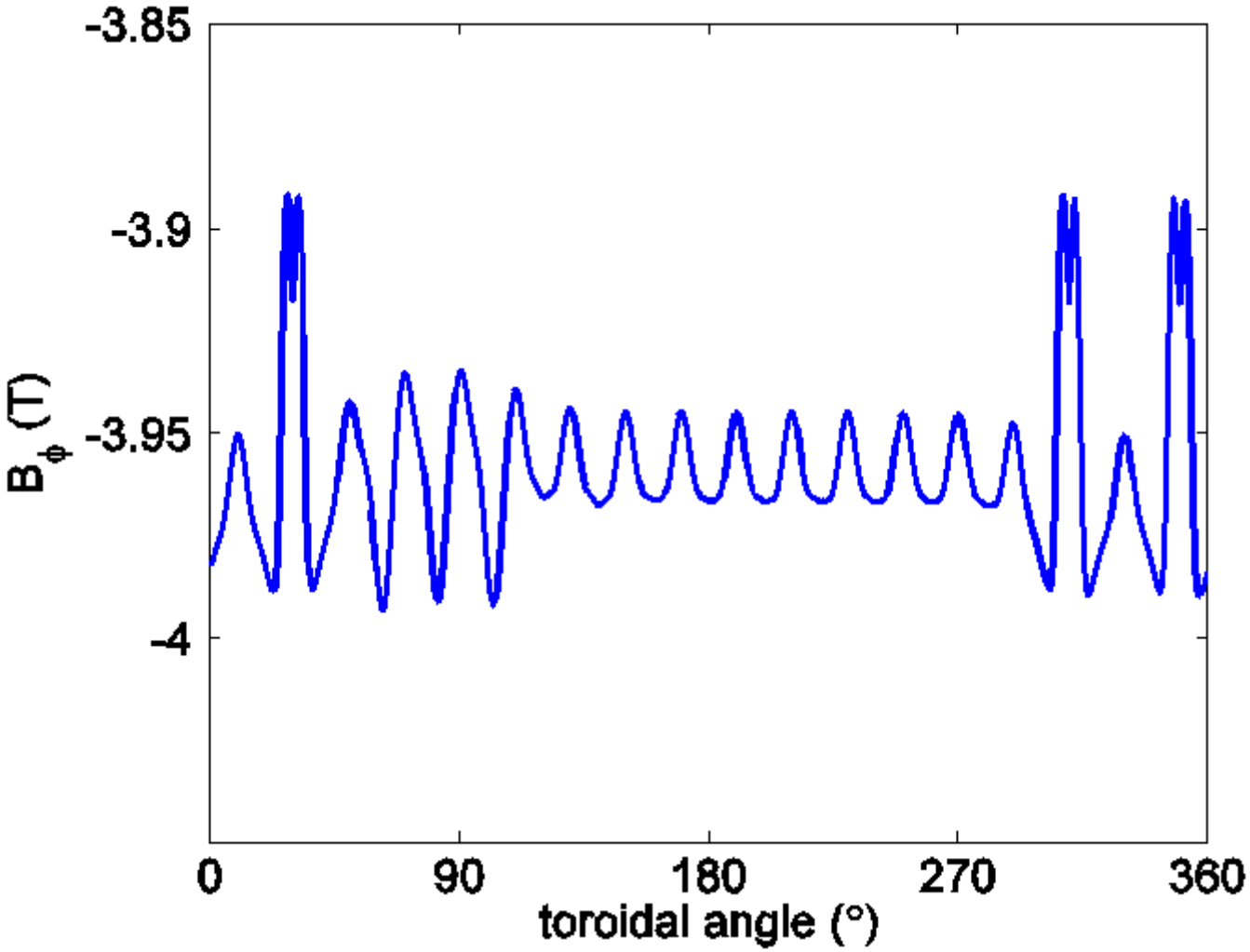
Effect of the finite # ($= 18$) of toroidal field coils

Magnetic field strength along ITER OMP separatrix



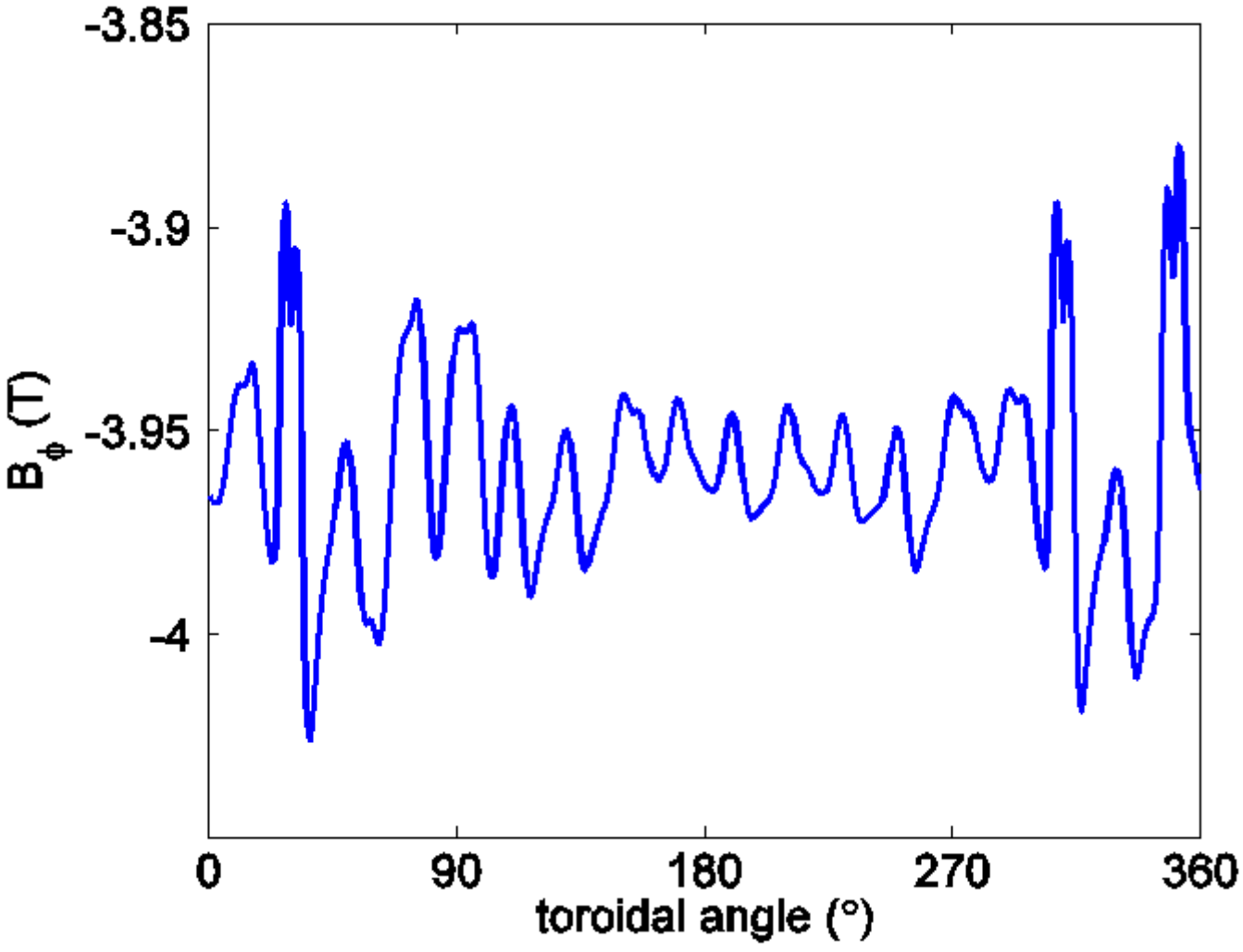
Effect of the ferritic inserts

Magnetic field strength along ITER OMP separatrix



Effect of the TBMs (Test Blanket Modules)

Magnetic field strength along ITER OMP separatrix



Effect of the RMP coils

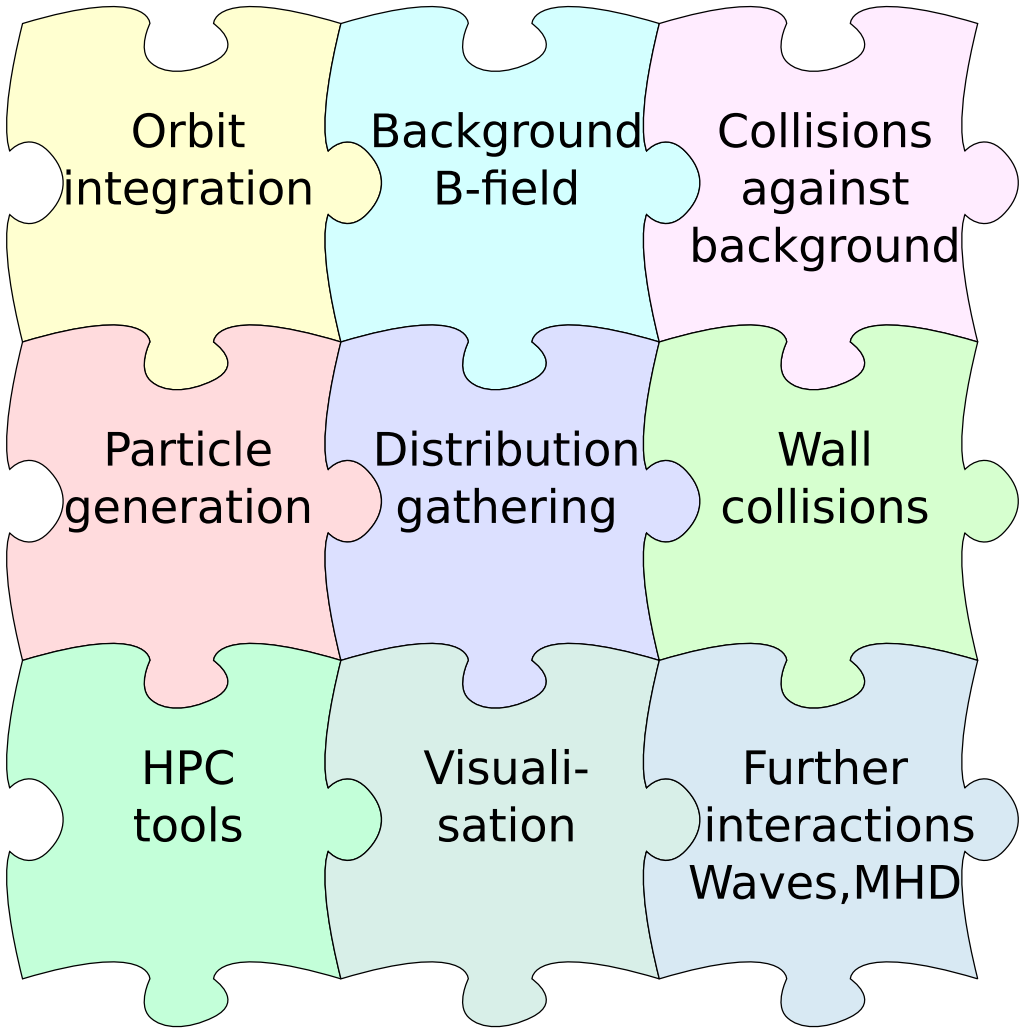
HPC: super computers needed to simulate fast ions

A 3.5 MeV alpha-particle in ITER slows down in $\sim$1 s. Simulating it takes (with realistic 3D B field and wall and collisions etc)

- $\sim$1 min in GC approach as a guiding centre or an
- $\sim$1 h in full orbit approach

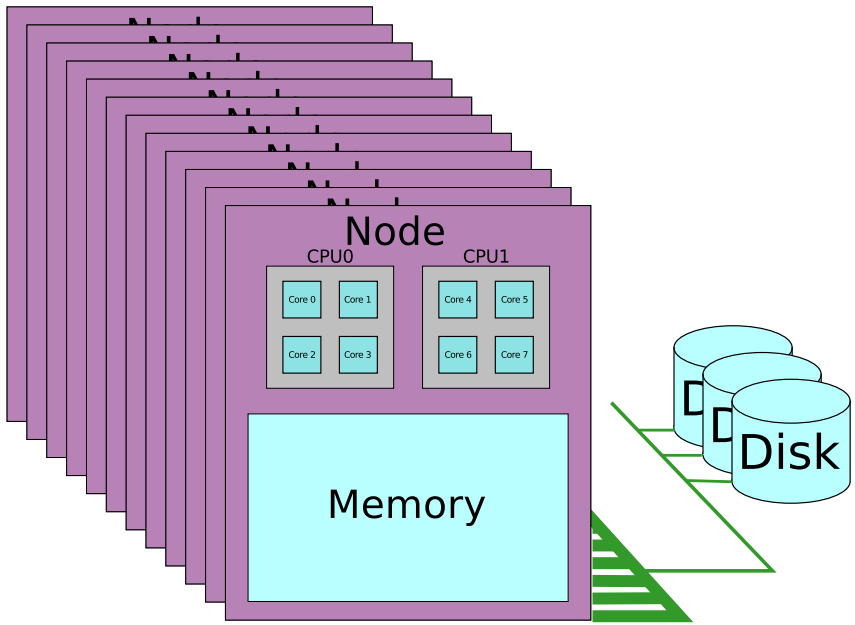
One needs from thousands to millions of alphas, depending on the quantity of interest $\rightarrow$ super computers are a good idea.

Building blocks for a MC-FP simulation tool



HPC: super computers are complicated beasts with many layers and connections of variable speed

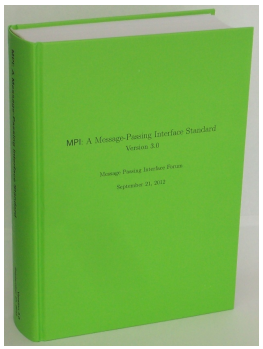
- First: make an efficient program for one CPU on a desktop. Then the code can be made to be run massively parallel.
- What follows applies to codes that are run with some thousands of CPUs. Codes requiring more CPUs would use more advanced tools.



Utilizing many nodes and CPUs

Our problem is *Embarassingly Parallel*: just distribute the particles to different CPUs for calculation. How to do that?

- *Message Passing Interface (MPI)* offers a relatively simple way to distribute the tasks to the various nodes. In fact, each CPU on each node can run a separate task. Messages are passed only when combining final results from all tasks. MPI is good for cluster environments (such as our HELIOS in Japan).
- *High Troughput Computing (HTC)*: we have implemented support for *CPU scavenging* with HTCondor enabled workstations in our department in Finland. This works well because the markers are independent.
- In principle also *Grids* would work, but we haven't needed that option – yet.



HTCondor
High Throughput Computing

Using the (slow) disk efficiently

In any supercomputer the computing nodes outnumber the nodes handling the shared filesystems. Therefore file reading and writing needs attention. Here are some solutions:

- One computing task can do the file reading and writing, then MPI is used to send the data as messages. (We use this approach.)
- Using compressed binary file formats can reduce reading and writing times and enhance user experience. (We use HDF5.)
- There are parallellized high level file access libraries, such as parallel HDF5 and parallel NetCDF. They use lower level libraries to access parallelism in the computer.

Making do with the memory on a node

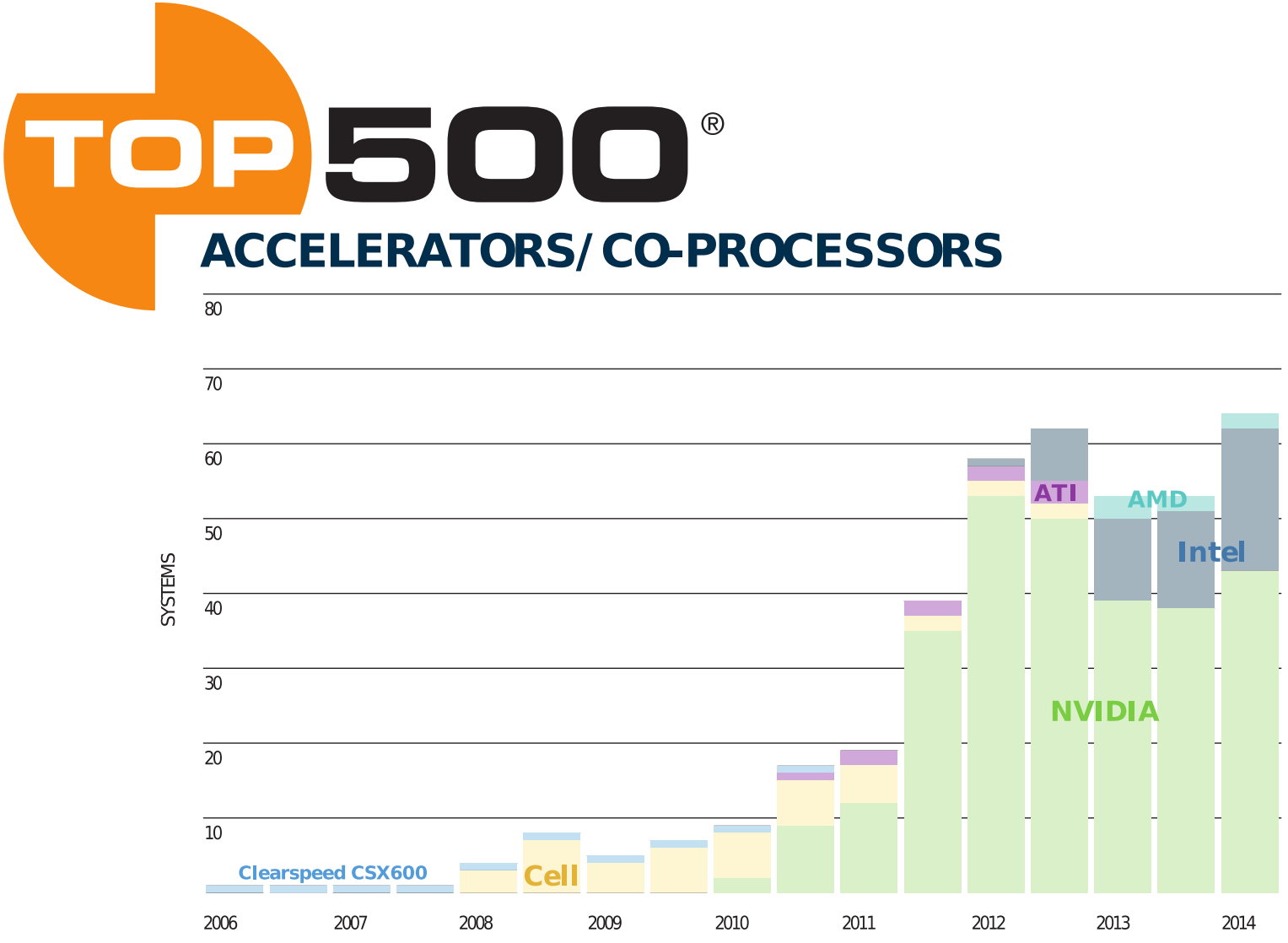
All the CPUs on a single node share the same memoryspaces.
What do we need the memory for:

- Background magnetic field is a largish (a few GB)
- The particle density histogram is a multidimensional object, often at least 4D (at least several MB, grows quickly with number of slots)

Luckily until now the amount of memory per CPU has remained on the level of several GB / CPU for years (Helios: 16 cores sharing 58GB).
If, in the future, the memory between processes on each node has to be shared $\rightarrow$ use *Open Multi-Processing (openMP)*

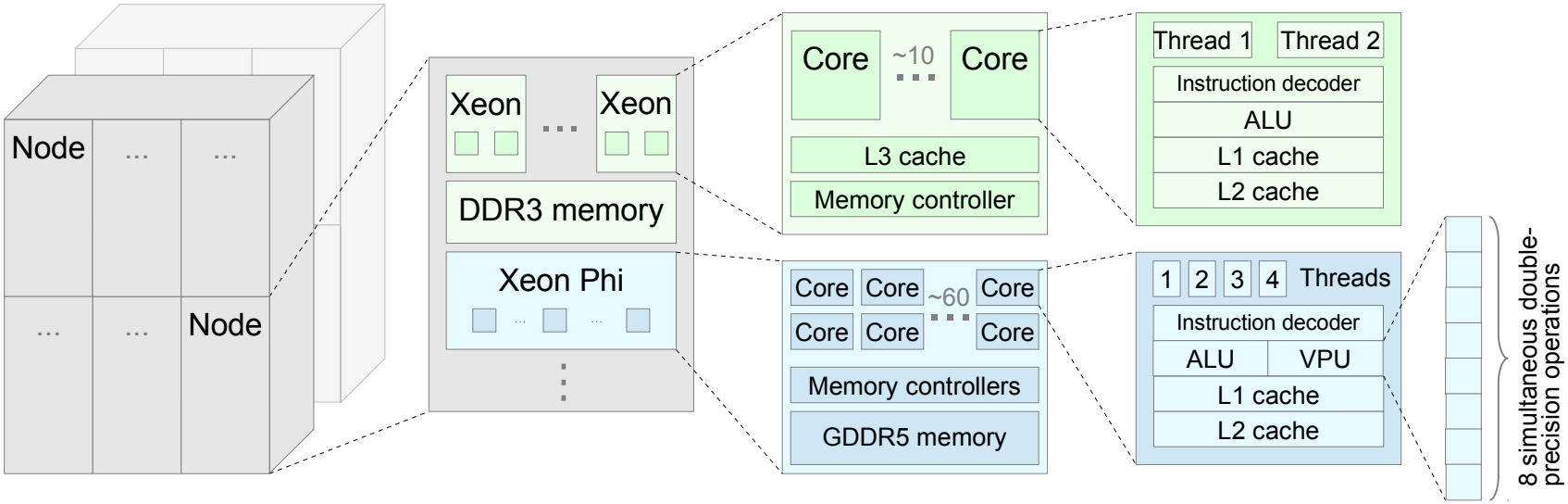
- Typically each node would house a single MPI task
- Each task would then be divided with openMP to multiple threads, typically one per CPU core on each node.
- Threads would share the memory. This means that the programmer needs to make sure the threads do not corrupt the data e.g. by concurrently writing to it.

Future: GPGPU / Accelerators



June 2014

Future: Many many cores, same memory



- Intel's Xeon Phi is an accelerator card. Some 250 threads sharing 15 GB of memory.
- The cores are not very fast, they are wide: Single Instruction Multiple Data (SIMD) means code needs to be well vectorized.
- Our demonstration code runs thousands of particles in parallel per Xeon Phi and shares memory. We are projecting 50x speed up on Helios Xeon Phi nodes.
- Check out also LOCUST-GPU: Production ready code that runs on NVIDIA's GPGPU cards.

Outline

1 Fokker-Planck theory and stochastic motion

2 Guiding center formalism

3 High Performance computing

4 Applying the formalism – if time permits

5 Summary

Example of using Langevin equation to solve FP equation

- ASCOT – race track for fusion ions
- Fusion alphas and ITER first wall
- TBM mock-up experiments at DIII-D
- Impurity injection experiments at AUG

Outline

1 Fokker-Planck theory and stochastic motion

2 Guiding center formalism

3 High Performance computing

4 Applying the formalism – if time permits

5 Summary

Pros and Cons of Langevin equation with MC

Solving the kinetic equation (here: FP) using the Monte Carlo method is very easy and flexible.

HOWEVER,

in careless use, it suffers from the *garbage-in, garbage-out* syndrome. Therefore, one should keep in mind at least the following:

Fokker-Planck theory

- Solving FP equation in *particle picture* is easy to formulate but computationally very expensive – and, in most cases, solving the gyro motion does not yield additional information
- Solving FP equation in *GC picture* is computationally efficient but comes at the price of mathematical complexity
- In some cases, a *hybrid picture* is beneficial: e.g., when approaching material surfaces, follow particles in both pictures and drop full orbit when receding from the surface.
- When introducing new operators in the FP equation, one should always make sure that they can be written in divergence form

High Performance computing

Your results can only be as good and reliable as your input!
Therefore, before carrying out the simulations check the following

- The quality of your magnetic background: is the field divergence-free and smooth? (Crazy ∇B -drifts are guaranteed to give crazy results)
- Choose your interpolation routine carefully: you have to navigate between requirements for accuracy and speed.
- Check you orbit-following routine so that it conserves energy and collisionless orbits in axisymmetric case (otherwise you get numerical transport)
- When moving to HPC platforms, check scalability and optimize the code as well as I/O for that particular platform.