Particle-in-Cell methods in plasma physics

R. Hatzky and A. Bottino

Outline

- The Vlasov-Maxwell equations
- The PIC method
- Monte Carlo evaluation of integrals
- The gyrokinetic Vlasov equation
- Discretization of the PIC method
- Control variates as variance reduction method
- Summary

The Vlasov equation

The usual basis for kinetic treatments of a collisionless plasma is the Vlasov equation:

$$\frac{\partial \hat{f}_s}{\partial t} + \mathbf{v} \cdot \frac{\partial \hat{f}_s}{\partial \mathbf{x}} + \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial \hat{f}_s}{\partial \mathbf{v}} = 0$$

Here, $\hat{f}_s(\mathbf{x}, \mathbf{v}, t)$ is the distribution function of the sth species in six-dimensional phase space with the spatial coordinate $\mathbf{x}$ and the velocity coordinate $\mathbf{v}$.

The Vlasov equation can be written in the following form:

$$\frac{D\hat{f}_s}{Dt} \overset{\mathrm{def}}{=} \frac{\partial \hat{f}_s}{\partial t} + \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} \cdot \frac{\partial \hat{f}_s}{\partial \mathbf{x}} + \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} \cdot \frac{\partial \hat{f}_s}{\partial \mathbf{v}} = 0$$

where

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{v} \quad , \quad \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \qquad \text{equations of motion} \qquad (1)$$

Its short form $D\hat{f}_s/Dt = 0$ means that the total derivative vanishes along the characteristics given by the integration of Eqs. (1).

In physical terms:
If we follow the particles along their trajectories by integrating the equations of motion, Eqs. (1), in six-dimensional phase space, the initial value of $\hat{f}_s(\mathbf{x}(t_0), \mathbf{v}(t_0))$ will not change.

This method is well known by the name "method of characteristics" and can be used to evolve $\hat{f}_s$ in time (initial value problem).

The Vlasov-Maxwell equations

The self-consistent electric and magnetic fields $\mathbf{E}$ and $\mathbf{B}$ which appear in the force law are calculated from Poisson's equation and Ampère's law, two of the Maxwell equations:

$$\epsilon_0 \nabla \cdot \mathbf{E} = \rho \qquad\qquad \nabla \times B = \mu_0 \mathbf{j} + \frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t}$$

Here, the charge density ρ and current density $\mathbf{j}$ are to be obtained at each point in space from the appropriate moments of the distribution function itself:

$$\rho(\mathbf{x}, t) = \sum_s q \int \hat{f}_s \, \mathrm{d}^3 v \qquad\qquad \mathbf{j}(\mathbf{x}, t) = \sum_s q \int \mathbf{v} \hat{f}_s \, \mathrm{d}^3 v$$

where the summation is over the species of particles present in the plasma.

The PIC method

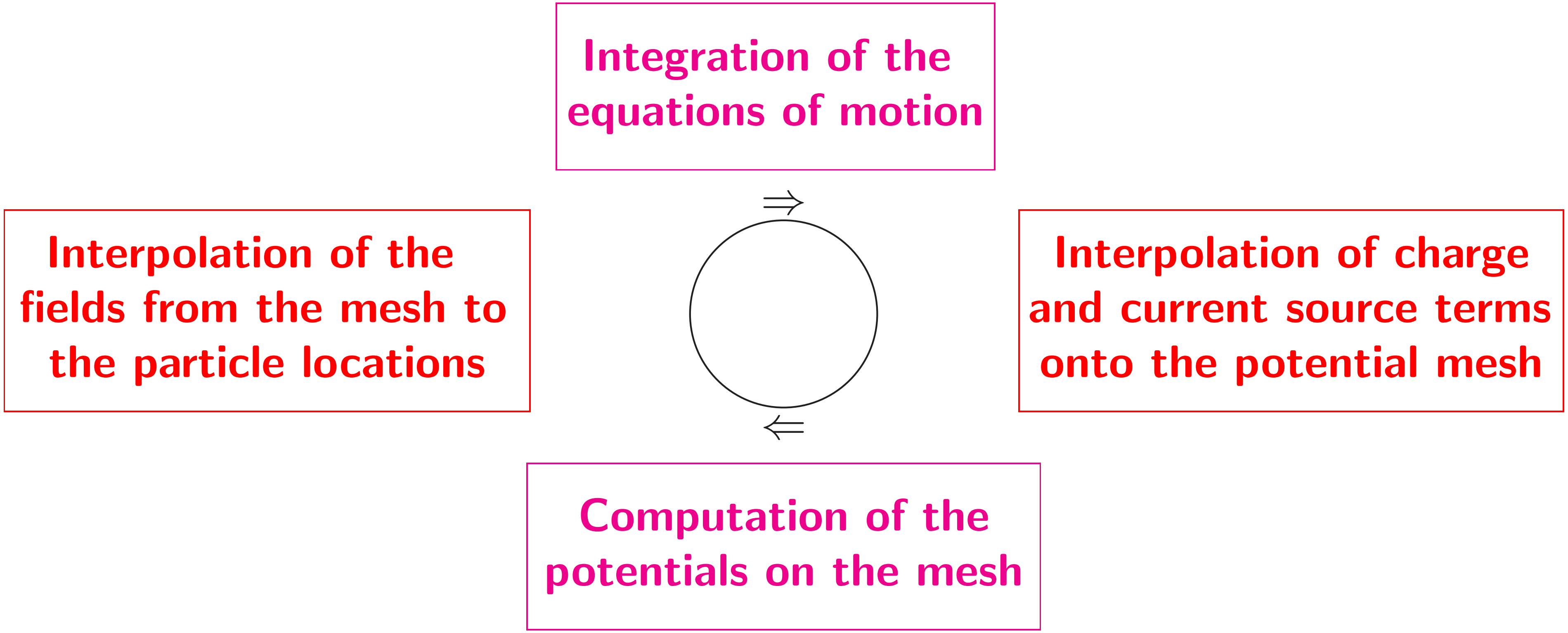
The Particle-In-Cell method (PIC) is a numerical technique used to solve a certain class of partial differential equations:

- Individual (macro) particles in a Lagrangian frame are traced in continuous phase space
- Moments of the distribution function are computed simultaneously on a Eulerian (stationary) mesh to solve the self-consistent field equations

The PIC method is a so-called Particle-Mesh (PM) method which includes interactions of particles only through the average fields.

Area of application in plasma physics:
laser-plasma interactions, electron acceleration and ion heating in the auroral ionosphere, magnetic reconnection, . . ., gyrokinetics

Schematic diagram of the PIC method



Integration of the equations of motion

Interpolation of charge and current source terms onto the potential mesh

Computation of the potentials on the mesh

Interpolation of the fields from the mesh to the particle locations

Monte Carlo evaluation of integrals

Of special interest is the evaluation of moments of the distribution function f over the phase space volume V, i.e. general integrals of the form

$$I(\Lambda) \overset{\text{def}}{=} \int_V \Lambda(\mathbf{z}) f(\mathbf{z}) \mathcal{J} \, \mathrm{d}\mathbf{z}$$

where $\Lambda(\mathbf{z})$ is a general function of the phase-space coordinates $\mathbf{z}$.
For example, $I(\Lambda)$ would be the number density in configuration space if $\Lambda = 1$, and the integral is evaluated over the velocity space.

The sampling distribution of our Monte-Carlo sampling points (marker) is done by a continuous probability density function $g(\mathbf{z})$ such that

$$\int_V g(\mathbf{z}) \mathcal{J} \, \mathrm{d}\mathbf{z} = 1$$

Now, the integral for $I(\Lambda)$ can be written in the following form

$$\mathcal{E}[\lambda(\mathbf{z})] \overset{\text{def}}{=} \int_V \lambda(\mathbf{z}) g(\mathbf{z})\,\mathrm{d}\mathbf{z} \qquad \textbf{where} \qquad \lambda(\mathbf{z}) \overset{\text{def}}{=} \frac{\Lambda(\mathbf{z}) f(\mathbf{z})}{g(\mathbf{z})}$$

and $\mathcal{E}[\lambda]$ is the expected value of the random variable λ. In addition, we define the variance of λ by

$$\sigma^2 \equiv \mathcal{V}[\lambda(\mathbf{z})] \overset{\text{def}}{=} \int_V \{\lambda(\mathbf{z}) - \mathcal{E}[\lambda(\mathbf{z})]\}^2 g(\mathbf{z})\,\mathrm{d}\mathbf{z}$$

The crude Monte-Carlo estimator for the integral $I(\Lambda)$ is given by the sum over the marker weights w_n

$$I(\Lambda) = \frac{1}{N}\sum_{n=1}^{N} \Lambda(\mathbf{z}_n) w_n + \epsilon \qquad \textbf{where} \qquad \epsilon \overset{\text{def}}{=} \frac{\sigma}{\sqrt{N}}, \qquad w_n \overset{\text{def}}{=} \frac{f(\mathbf{z}_n)}{g(\mathbf{z}_n)}$$

Liouville's theorem

A **phase space volume** $\Omega_p \overset{\rm def}{=} \Omega(\mathbf{R}_p, v_{\|p}, \mu_p)$ moving in phase space according to the equations of motion **does not change its volume**, although in general it can change its shape (**Liouville's theorem**):

$$\int_{\Omega_p} \mathrm{d}\mathbf{R}\,\mathrm{d}\mathbf{v} = \int_{\Omega_p} B^\star_\| \,\mathrm{d}\mathbf{R}\,\mathrm{d}v_\|\,\mathrm{d}\mu\,\mathrm{d}\alpha = \textbf{const.}$$

As the marker distribution and phase space volumes Ω_p are related by

$$g_p = \frac{\Omega}{\Omega_p} = \textbf{const.}$$

it follows for the marker weights from $f_p = \textbf{const.}$ that

$$w_p = \frac{f_p}{g_p} = \textbf{const.}$$

Importance sampling

The standard error ϵ can be reduced by e.g. increasing the number of markers N. However, the convergence is not very fast due to $\epsilon \propto 1/\sqrt{N}$.

In addition, one can **reduce the error ϵ by a reduction of the variance σ^2**.

The optimal choice is to set $g(\mathbf{z}) = |f(\mathbf{z})|$, i.e. allocate more markers in regions where $|f(\mathbf{z})|$ is large. This method is called **"importance sampling"**:

$$w_n = \frac{f(\mathbf{z}_n)}{g(\mathbf{z}_n)} = \frac{f(\mathbf{z}_n)}{|f(\mathbf{z}_n)|} = \textbf{const.}$$

Note, that at the initialization of the PIC method one knows $f(\mathbf{z}, t_0)$ analytically and can use it for an importance sampling.

Difference between the Eulerian and Monte Carlo method

Midpoint integration

For an equidistant mesh the sampling distance h is correlated with the number of sampling points N by

$$h \propto \frac{1}{N}$$

Accordingly, we have to distribute the sampling points over d dimensions by

$$h \propto \frac{1}{N^{\frac{1}{d}}}$$

The error term of the Midpoint integration is:

$$E_{\mathrm{M}} = \mathcal{O}(h^2) = \mathcal{O}\left(\frac{1}{N^{\frac{2}{d}}}\right)$$

Monte Carlo integration

The error term of the Monte-Carlo integration is:

$$E_{\mathrm{MC}} = \mathcal{O}\left(\frac{1}{N^{\frac{1}{2}}}\right)$$

We can compare the convergence rates of both integration methods

$$\mathcal{O}\left(\frac{1}{N^{\frac{1}{2}}}\right) < \mathcal{O}\left(\frac{1}{N^{\frac{2}{d}}}\right) \qquad \textbf{when} \qquad d > 4$$

to see that Monte-Carlo integration will converge faster for quintuple multiple integrals.

⇒ The Monte-Carlo method has its advantages in high dimensional spaces.

Quasi-Monte Carlo method

This method loads the sampling points (particles) as smoothly as possible in (phase) space without imposing strong correlations.

Two low-discrepancy sequences with their lower error bounds:

- the Sobol's quasirandom sequence

$$E_{\mathrm{S}} \leq \mathcal{O}\left[\frac{(\log_{10} N)^d}{N}\right]$$

- the bit-reversed numbers of Hammersley's sequence:

$$E_{\mathrm{H}} \leq \mathcal{O}\left[\frac{(\log_{10} N)^{d-1}}{N}\right]$$

⇒ Only helps at the start of the PIC simulation ("quiet start" conditions).

The gyrokinetic Vlasov equation (Hahm, 1988)

The average of the Vlasov equation over the fast gyro-motion leaves just the guiding center motion and thus reduces the dimensionality of the problem:

$$\frac{\partial f_s}{\partial t} + \frac{\mathrm{d}\mathbf{R}}{\mathrm{d}t} \cdot \frac{\partial f_s}{\partial \mathbf{R}} + \frac{\mathrm{d}v_\parallel}{\mathrm{d}t} \cdot \frac{\partial f_s}{\partial v_\parallel} = 0 \quad ; \quad \frac{\mathrm{d}\mu}{\mathrm{d}t} = 0 \quad , \quad \mu \overset{\rm def}{=} \frac{v_\perp^2}{2B}$$

Here, $f(\mathbf{R}, v_\parallel, \mu, t)$ is the guiding center distribution function of the sth species in the reduced five-dimensional phase space with the guiding center coordinate $\mathbf{R}$ and the parallel velocity coordinate $v_\parallel$ and the magnetic moment μ.

$$v_{\rm gc}(\langle\phi\rangle, \langle\mathbf{A}\rangle) \overset{\rm def}{=} \frac{\mathrm{d}\mathbf{R}}{\mathrm{d}t}, \qquad a_{\rm gc}(\langle\phi\rangle, \langle\mathbf{A}\rangle) \overset{\rm def}{=} \frac{\mathrm{d}v_\parallel}{\mathrm{d}t}$$

guiding center velocity/acceleration

with the gyro-averaged electrostatic and magnetic potentials $\langle\phi\rangle$ and $\langle\mathbf{A}\rangle$.

Differences in the numerical models

- **Local approximation vs. global model**
  - Covering just a local "flux-tube" or the whole physical domain
- **Adiabatic vs. kinetic electrons**
  - Taking the full kinetics of all species into account
- **Electrostatic vs. electromagnetic model**
  - Taking self generated currents and Ampère's law into account

So far none of the codes covers all physical aspects!

Numerical methods for solving the gyrokinetic equations

- Eulerian
  - Fixed grid in 5-dim phase space
  - Finite difference discretization of differential operators
- Semi-Lagrangian
  - Fixed grid in 5-dim phase space
  - Trace back of characteristics in time to evolve distribution function
- Lagrangian: Particle-in-Cell (PIC), i.e. Monte Carlo method
  - Follow particle trajectories in 5-dim phase space
  - Project particles to real space to compute charge and current density

All three methods need additional electrostatic and magnetic potential solvers.

The discretized f

The full-f PIC approximation is given by a sum of δ functions:

$$f(\mathbf{R}, v_\parallel, \mu, t) = \sum_{p=1}^{N} \frac{w_p}{\mathcal{J}_{\rm red}} \, \delta(\mathbf{R} - \mathbf{R}_p(t)) \, \delta(v_\parallel - v_{\parallel p}(t)) \, \delta(\mu - \mu_p(t_0))$$

with the Jacobian in the reduced phase space $\mathcal{J}_{\rm red} = 2\pi B_\parallel^\star$.

Each marker (macro particle, tracer, $\ldots$) p is defined by:

- **its position in the 5-dim phase space $(\mathrm{R}_p, v_{\parallel p}, \mu_p)$**
- **its phase-space volume Ω_p assigned by the initial marker distribution $g(t_0)$**
- **its constant weight w_p**
- **its averaged value of $f_p = w_p/\Omega_p$ over the phase-space volume Ω_p**

Time integrators for the equations of motion

The time integrator should have a low storage and should be efficient.

- **Explicit time integrators are usually easy to implement but have a time step criterion which can be quite restrictive.**
- **Implicit time integrators are more complex and costly (e.g. due to solving of a matrix equation) but have no time step criterion.**

Quite popular for PIC are explicit Runge-Kutta schemes of $\mathcal{O}[(\Delta t)^4]$:

- **Classical Runge-Kutta, four stages, storage requirement of $4N$**
- **Low-storage Runge-Kutta, four stages, storage requirement of $3N$**
- **Very low-storage Runge-Kutta, six stages, storage requirement of $2N$**

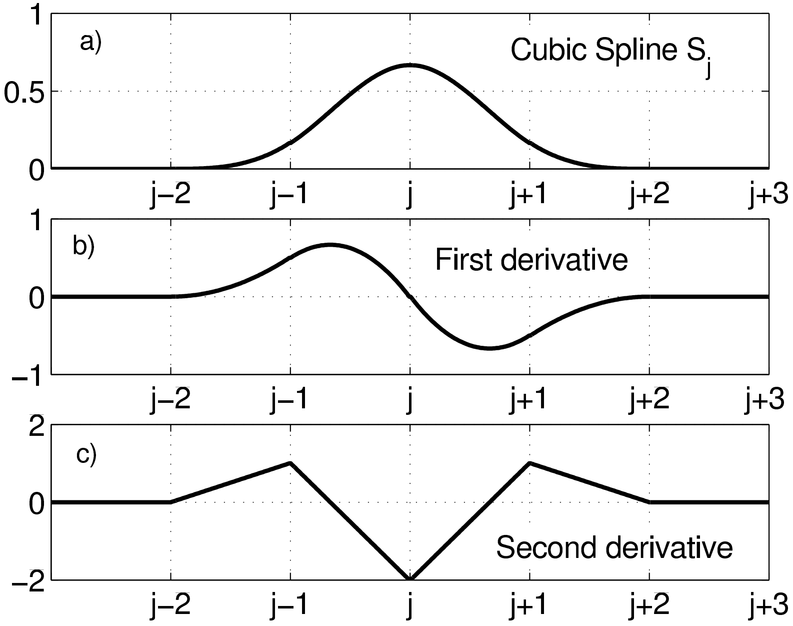
Discretization of the electrostatic potential equation

The Helmholtz type like electrostatic potential equation is discretization with a finite element method.

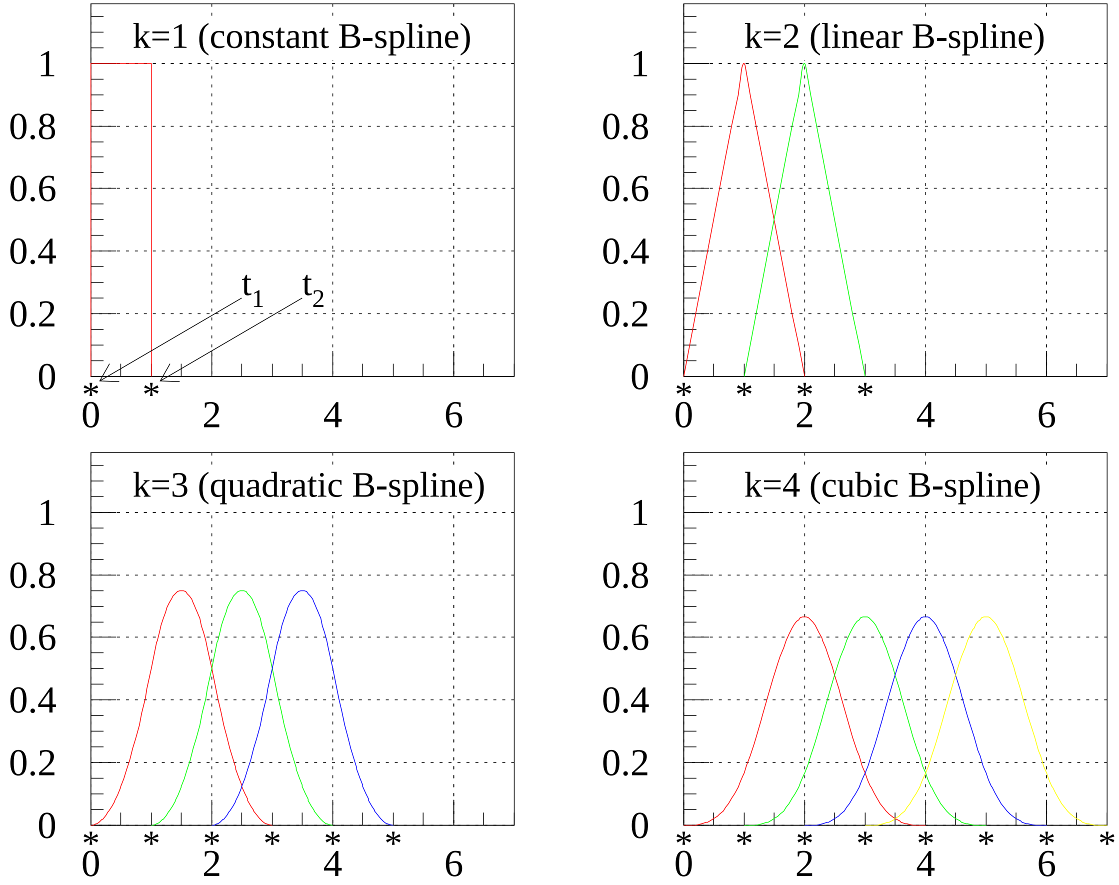
$$\phi(\mathbf{x},t)=\sum_{\nu}\phi_\nu(t)\Lambda_\nu(\mathbf{x})$$

where $\Lambda_\mu(\mathbf{x})$ is a product of unidimensional B-splines S of order k:

$$\Lambda_\mu(\mathbf{x})=S_l^k(r)S_m^k(\chi)S_n^k(\varphi)$$



B-splines of different order



Some properties of B-splines

Partition of unity:

The sequence provides (B_{jk}) a positive and local partition of unity, that is, each (B_{jk}) is positive on $(t_j \ldots t_{j+k})$, is zero off $[t_j \ldots t_{j+k}]$, and

$$\sum_j B_{jk}(x) = 1$$

Differentiation:

$$\mathcal{D}\left(\sum_j \alpha_j B_{jk}\right) = (k-1)\sum_j \frac{\alpha_j - \alpha_{j-1}}{t_{j+k-1} - t_j} B_{j,k-1}$$

The first derivative of a spline function $\sum_j \alpha_j B_{jk}$ can be found simply by differencing its B-spline coefficients, thereby obtaining the B-spline coefficients of its first derivative, a spline one order lower.

Tensor product of B-splines:

B-splines can be extended to higher dimensions, e.g. three dimensions:

$$B_{lmn}(\mathbf{x}) \overset{\rm def}{=} B_l(x)\, B_m(y)\, B_n(z)$$

Monograph: Carl de Boor, A practical guide to splines, Revised edition, Springer-Verlag 2001.

Advantages of finite elements (B-splines):

- **Conservation laws, e.g. particle number and energy conservation for PIC are consistently preserved**
- **Complicated geometries and non-equidistant meshes are easy to implement**

Discretization using Galerkin's method

$$\frac{en_0}{k_{\mathrm{B}}T_{\mathrm{e}}}\phi - \nabla_\perp \cdot \left(\frac{n_0}{B\Omega_{\mathrm{i}}}\nabla_\perp \phi\right) = \langle n_{\mathrm{i}}\rangle \qquad \Rightarrow \qquad \sum_{\nu'} A_{\nu\nu'}\,\phi_{\nu'}(t) = b_\nu(t)$$

1. **Insert the discretized form of** $\phi(\mathbf{x},t) = \sum_{\nu'} \phi_{\nu'}(t)\Lambda_{\nu'}(\mathbf{x})$
2. **Multiply the equation by a test function** $\Lambda_\nu(\mathbf{x})$
3. **Integrate the whole equation over the entire plasma volume**

RHS:

$$\sum_{\nu'} \int \left(\frac{n_0}{B\Omega_{\mathrm{i}}}\nabla_\perp\Lambda_\nu \cdot \nabla_\perp\Lambda_{\nu'} + \frac{en_0}{k_{\mathrm{B}}T_{\mathrm{e}}}\Lambda_\nu\Lambda_{\nu'}\right) \mathrm{d}\mathbf{x}\ \phi_{\nu'}(t) \overset{\mathrm{def}}{=} \sum_{\nu'} A_{\nu\nu'}\,\phi_{\nu'}(t)$$

Polarization density (Laplacian operator): integrated by parts (weak form)
⇒ **Discretization consists of B-splines and B-spline first derivatives only**

LHS:

$$\sum_{p=1}^{N} w_p \frac{1}{2\pi}\int_0^{2\pi}\int \Lambda_\nu(\mathbf{x})\,\delta(\mathbf{R}_p+\rho_{\mathrm{i}p}-\mathbf{x})\,\mathrm{d}\mathbf{x}\,\mathrm{d}\alpha = \sum_{p=1}^{N} w_p \frac{1}{2\pi}\int_0^{2\pi}\Lambda_\nu(\mathbf{R}_p+\rho_{\mathrm{i}p})\,\mathrm{d}\alpha \overset{\mathrm{def}}{=} b_\nu(t)$$

using

1. **The definition of the gyro-averaged ion density which smears out the density along the gyro-ring of radius ρ_i:**

$$\langle n_\mathrm{i}\rangle \overset{\mathrm{def}}{=} \int f\,\delta(\mathbf{R}+\rho_\mathrm{i}-\mathbf{x})\,\mathrm{d}^6\mathbf{Z} = \int f\,\delta(\mathbf{R}+\rho_\mathrm{i}-\mathbf{x})\,B_\parallel^\star\,\mathrm{d}\mathbf{R}\,\mathrm{d}v_\parallel\,\mathrm{d}\mu\,\mathrm{d}\alpha$$

2. **The discretized f:**

$$f = \sum_{p=1}^{N} \frac{1}{2\pi B_\parallel^\star} w_p(t)\,\delta(\mathbf{R}-\mathbf{R}_p(t))\,\delta(v_\parallel - v_{\parallel p}(t))\,\delta(\mu-\mu_p(t_0))$$

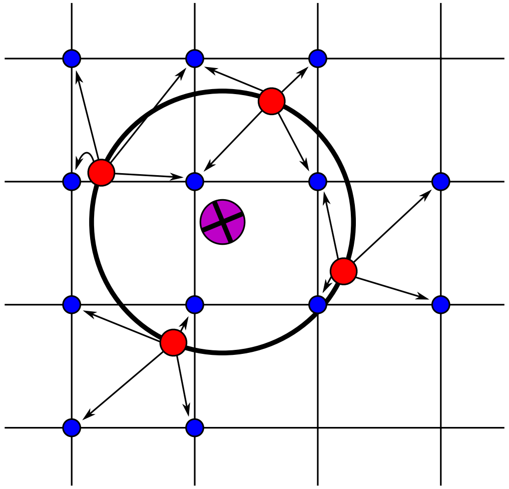
The interpolation of the density onto the grid

The construction of the RHS is the so-called charge assignment:

$$b_\nu(t) \overset{\text{def}}{=} \sum_{p=1}^{N} w_p \frac{1}{2\pi} \int_0^{2\pi} \Lambda_\nu(\mathbf{R}_p + \rho_{\text{ip}})\, \mathrm{d}\alpha$$

Projection of the weights w_p in the form of gyro-rings onto the B-spline basis.

The charge assignment is a scatter operation, e.g. each sample point contributes to 64 grid points for cubic B-splines in 3-dim.

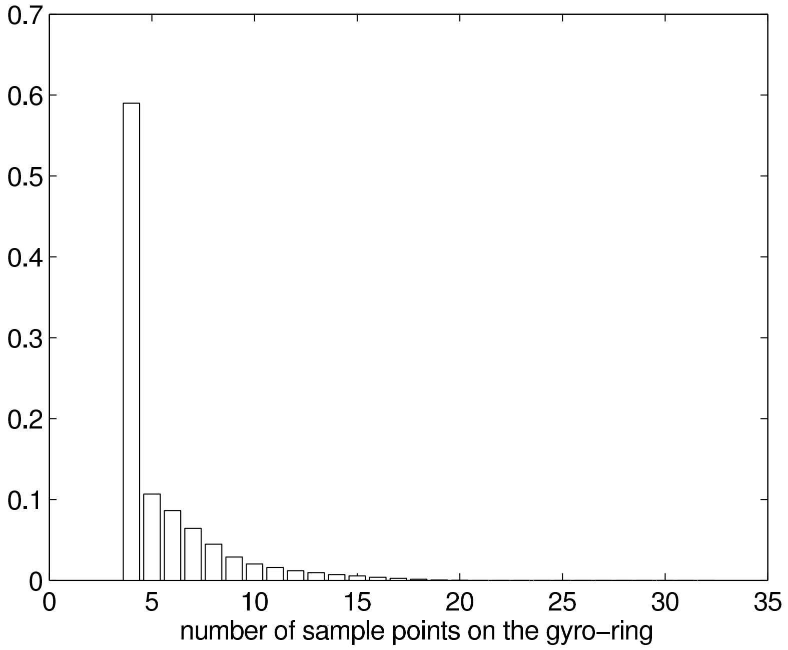


The figure shows linear interpolation for linear B-splines in 2-dim.

Adaptive gyro-average (Hatzky, et al., 2002)

The number of the sample points on the gyro-ring is linearly increased from a minimum of 4 to a maximum of 32 according to the size of the gyro-radius. The equidistantly distributed points are rotated for each marker by a random angle.

The figure shows the relative frequency of the total number of sample points on all gyro-rings.



⇒ Better approximation of the integral over the gyro-ring

Solving the electrostatic potential equation

The following **matrix equation** has to be solved:

$$\sum_{\nu'} A_{\nu\nu'}\,\phi_{\nu'}(t) = b_\nu(t)$$

where the matrix A is symmetric, positive definite and time-independent.

For 2-dim problems direct solvers are quite useful as the Cholesky decomposition has to be performed only once at the initialization. At each time step the very efficient backsolve can be applied.
⇒ direct (parallel) sparse matrix packages, e.g. IBM WSMP

For 3-dim problems iterative solvers become mandatory. The conjugate gradient method is very efficient and can be combined with appropriate preconditioners.
⇒ iterative parallel sparse matrix packages, e.g. PETSc

The interpolation of the electric field from the grid

The gyro-averaged electrostatic potential $\langle\phi\rangle$ is defined by:

$$\langle\phi\rangle \overset{\rm def}{=} \frac{1}{2\pi}\int_0^{2\pi}\int \phi(\mathbf{x})\,\delta(\mathbf{R}+\rho_{\rm i}-\mathbf{x})\,{\rm d}\mathbf{x}\,{\rm d}\alpha = \frac{1}{2\pi}\int_0^{2\pi}\phi(\mathbf{R}+\rho_{\rm i})\,{\rm d}\alpha$$

The gyro-averaged electric field $\langle{\rm E}\rangle$ is defined by:

$$\langle\mathbf{E}\rangle \overset{\rm def}{=} -\nabla_{\rm R}\langle\phi\rangle = \frac{1}{2\pi}\int_0^{2\pi} -\nabla_{\mathbf{x}}\phi(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{R}+\rho_{\rm i}}\,{\rm d}\alpha + \mathcal{O}(\epsilon_{\rm B})$$

Inserting the discretized form of $\phi(\mathbf{x},t)=\sum_\nu \phi_\nu(t)\Lambda_\nu(\mathbf{x})$ gives:

$$\langle\mathbf{E}\rangle = -\sum_\nu \frac{\phi_\nu}{2\pi}\int_0^{2\pi}\nabla\Lambda(\mathbf{R}+\rho_{\rm i})\,{\rm d}\alpha + \mathcal{O}(\epsilon_{\rm B})$$

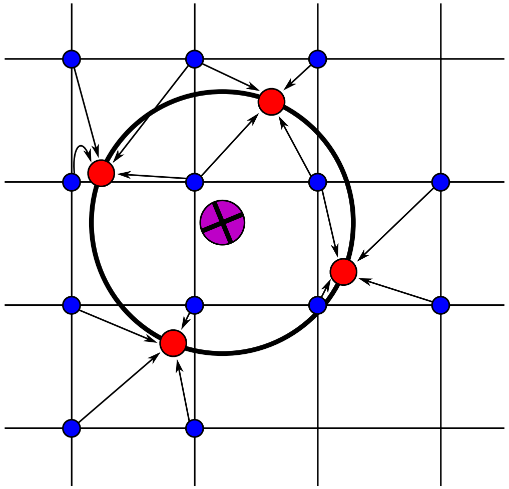
The gyro-averaged electric field $\langle \mathrm{E} \rangle$ is an analytic differential of the potential represented by the B-splines, i.e., the gradient is computed exactly using:

$$\nabla \Lambda_\nu(s, \vartheta, \varphi) = \frac{\partial \Lambda_\nu}{\partial s} \nabla s + \frac{\partial \Lambda_\nu}{\partial \vartheta} \nabla \vartheta + \frac{\partial \Lambda_\nu}{\partial \varphi} \nabla \varphi$$

The N_{av} field vectors E_n on the gyro-ring are calculated from the B-spline representation of the potential and then averaged.

The E_n calculation is a gather operation, e.g. each sample point is assembled from 64 grid points for cubic B-splines in 3-dim.

The figure shows linear interpolation for linear B-splines in 2-dim.



Control variates as variance reduction method

One tries to utilize (strong) correlation between the observed variable X and some auxiliary variable Y, the so called control variable whose expected value $\mathcal{E}[Y] = \nu$ has to be known analytically.

The task is to estimate the expected value $\mathcal{E}[X] = \mu$ with a preferably smaller standard deviation than $\mathcal{V}[X]$.

Hence, we define the variable Z which has the same expected value as $\mathcal{E}[X]$ by

$$Z \overset{\text{def}}{=} X - \alpha(Y - \nu) = \tilde{Z} + \alpha\nu \qquad \textbf{where} \qquad \tilde{Z} \overset{\text{def}}{=} X - \alpha Y$$

The parameter α will be used to further optimize the variance reduction property of the control variate.

The expected value of Z is as predicted

$$\mathcal{E}[Z] = \mathcal{E}[\tilde{Z}] + \alpha\nu = \mathcal{E}[X] - \alpha(\mathcal{E}[Y] - \nu) = \mathcal{E}[X].$$

**Only the variable $\tilde{Z}$ will be discretized by our control variate schemes as the expected value ν is known analytically and can be added at any time.
This holds also in the general case where μ is replaced by a multidimensional function f_0 in phase space.**

The variance of Z is

$$\mathcal{V}[Z] = \mathcal{V}[\tilde{Z}] = \mathcal{V}[X - \alpha Y] = \mathcal{V}[X] - 2\alpha \mathrm{Cov}[X, Y] + \alpha^2 \mathcal{V}[Y]$$

where the covariance is defined by

$$\mathrm{Cov}[X, Y] \overset{\text{def}}{=} E[(X - \mu)(Y - \nu)].$$

Effective control variates

The auxiliary variable Y is an effective control variate if the correlation is strong enough, i.e.

$$\frac{\mathrm{Cov}[X,Y]}{\alpha\mathcal{V}[Y]} > \frac{1}{2} \qquad \Rightarrow \qquad \mathcal{V}[\tilde{Z}] < \mathcal{V}[X].$$

For PIC simulations one can use the knowledge about the initial state $f(t_0)$ of the system to construct an effective control variate as long as the system does not evolve too far from its initial state.

For such situations the usage of a control variate is a valuable enhancement of the full-f PIC method which has naturally problems to resolve relatively small changes of the system.

The standard error ϵ (statistical noise) can be reduced in some cases drastically.

Construction of an effective control variates

A maximal reduction of the variance $\mathcal{V}[Z]$ is given by the optimal parameter

$$\alpha^\star \overset{\text{def}}{=} \frac{\text{Cov}[X,Y]}{\mathcal{V}[Y]}$$

which leads always to an optimal control variate. In practice, $\alpha^\star$ has to be estimated with a sufficiently small statistical error.

The corresponding optimal variance is given as

$$\mathcal{V}[Z] = \mathcal{V}[X] - \frac{\text{Cov}^2[X,Y]}{\mathcal{V}[Y]} = \mathcal{V}[X](1 - \text{Corr}^2[X,Y])$$

where the correlation coefficient is defined by

$$\text{Corr}[X,Y] \overset{\text{def}}{=} \frac{\text{Cov}[X,Y]}{\sqrt{\mathcal{V}[X]}\sqrt{\mathcal{V}[Y]}} \qquad \textbf{with} \qquad -1 \le \text{Corr}[X,Y] \le 1$$

- The quality of the auxiliary variable Y as a control variate depends on the correlation between the variables X and Y.
- As long as X and Y are correlated the optimal parameter $\alpha^\star$ always reduces the variance of $\mathcal{V}[Z]$ even if X and Y are negatively correlated.
- In practice, the optimal parameter $\alpha^\star$ can be estimated from the Monte Carlo data.
- The performance does not depend strongly on α when α is close to $\alpha^\star$, where the derivative $\mathrm{d}\mathcal{V}[Z]/\mathrm{d}\alpha$ is zero.
- For very strong correlation between the variables X and Y the optimal parameter $\alpha^\star$ can be approximated by one.

Charge assignment with a control variate

To simplify matters we consider the charge assignment without gyro-average:

$$b(t) = \sum_{p=1}^{N} \Omega_p \underbrace{\left[f_p - \alpha f_0(\mathbf{R}_p(t), \mathbf{v}_p(t))\right]}_{\tilde{Z}=X-\alpha Y} \Lambda(\mathbf{R}_p) + \alpha \underbrace{\int f_0(\mathbf{R}, \mathbf{v}) \Lambda(\mathbf{R}) \, \mathrm{d}\mathbf{R} \, \mathrm{d}\mathbf{v}}_{\mathcal{E}[Y]=\nu}$$

$$= \sum_{p=1}^{N} [w_p - \alpha \Omega_p f_{0p}] \Lambda(\mathbf{R}_p) + \alpha \hat{b} \qquad \textbf{where} \qquad \hat{b} \overset{\mathrm{def}}{=} \int f_0(\mathbf{R}, \mathbf{v}) \Lambda(\mathbf{R}) \, \mathrm{d}\mathbf{R} \, \mathrm{d}\mathbf{v}$$

- **The B-spline coefficient vector $\hat{\mathrm{b}}$ is the result of the analytic projection of the control variate f_0 onto the B-spline basis.**
- **As long as the control variate is time independent the B-spline coefficient vector $\hat{\mathrm{b}}$ has to be calculated only once at the initialization.**

The δf method

The popular δf method chooses the following ansatz:

$$\delta f = f - f_0$$

For PIC simulations it is only used as an efficient noise reduction method if

$$\delta f \ll f_0$$

It can be interpreted as a control variate method which has set $\alpha = 1$:

$$b(t) = \sum_{p=1}^{N} \Omega_p [f_p - f_{0p}(t)] \Lambda(\mathbf{R}_p) + \hat{b} = \sum_{p=1}^{N} \Omega_p \delta f_p(t)\, \Lambda(\mathbf{R}_p) + \hat{b}$$

It usually integrates an unnecessary evolution equation to derive $\delta f_p(t)$.

Optimized marker loading (Hatzky, et al., 2002)

When using a control variate only the part $f - \alpha f_0$ is disretized by the markers.

The optimal marker distribution would be $g(\mathbf{z}, t) = |f(\mathbf{z}, t) - \alpha f_0(\mathbf{z})|$, i.e. allocate more markers in regions where $|f - \alpha f_0|$ is large ("importance sampling").

Difficulties:

- The "ad hoc" marker distribution $g(\mathbf{z}, t_n)$ is not known.
- It is very complicated to redistribute the markers during the simulation.

Solution: Iterative treatment

- Use a precursor run to derive information about a better suited marker distribution for the whole simulation interval.
- Subsequently, modify the initial marker distribution of an optimized run.

Summary

- PIC simulations are widely-used in plasma physics, e.g. in gyrokinetics.
- The PIC method is relatively intuitive and straightforward to implement.
- PIC simulations can consist of more than 10^{10} markers which makes even 3-dimensional simulation domains with full velocity space possible.
- PIC simulations do not evolve “numerical diffusion”.
- The PIC method is easy to parallelize on computers.

Disadvantage:

- The PIC method suffers from “statistical noise”.

References

- **C.K. Birdsall and A.B. Langdon:** *Plasma physics via computer simulation*, **in Plasma Physics Series edited by E. W. Laing (Institute of Physics Publishing, Bristol, 1995).**
- **R.W. Hockney and J.W. Eastwood:** *Computer simulation using particles* **(Adam Hilger, Bristol, 1989).**
- **T.S. Hahm,** *Nonlinear gyrokinetic equations for tokamak microturbulence*, **Phys. Fluids** 31**, p. 2670–2673 (1988).**
- **A.Y. Aydemir,** *A unified Monte Carlo interpretation of particle simulations and applications to non-neutral plasmas*, **Physics of Plasmas** 1**, p. 822–831 (1994).**
- **R. Hatzky, T.M. Tran, A. Könies, R. Kleiber, and S.J. Allfrey,** *Energy Conservation in a Nonlinear Gyrokinetic Particle-in-cell Code for Ion-Temperature-Gradient-driven (ITG) Modes in* θ*-Pinch Geometry*, **Physics of Plasmas,** 9**: p. 898–912 (2002).**
- **URL: www.efda-hlst.eu**