Finite element phase space representation of zonal structures in ORB5, a.k.a how to calculate $f(r,v_{//},\mu)$ in ORB5*

*spin-off of ATEP activity on Phase Space Zonal Structures [M. Falessi]

A.Bottino

[1]Max Planck Institute for Plasma Physics,
Boltzmannstr. 2, 85748 Garching, Germany

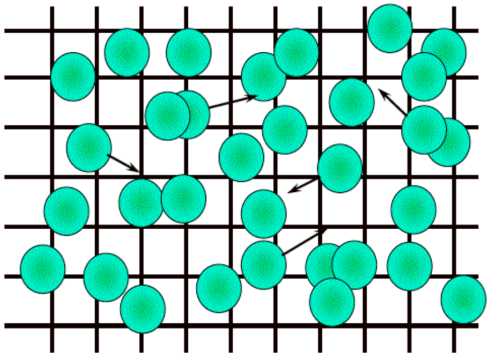
Acknowledgments:
Sergio Briguglio, Matteo Falessi, Thomas Hayward-Schneider,
Alexey Mishchenko and Xin Wang

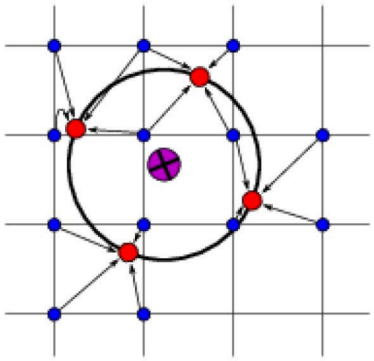
TSVV10, 23/02/2022

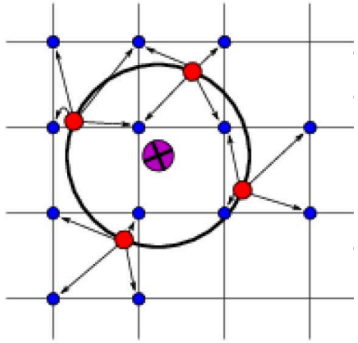
Particle-in-cell (PIC), physicist view

Why is the calculation of a simple quantity like $f(r,v_{//},\mu)$ a problem for ORB5?

- **In PIC, plasma is described by a small number of super-particles (SP), SP are objects with a weight (physical particles density) w_p a position (5D/6D) Z_p and a phase-space volume V_p associated with it.**
- **The Lagrangian motion of the SPs is straightforwardly described by the Newton-Maxwell equations, usually phase-space volumes are conserved along Lagrangian trajectories.**
- **The self-consistent fields (if needed) are calculated by projecting on a spatial grid charge and current associated with each SP. How the field equations are solved is not part of the PIC discretisation.**

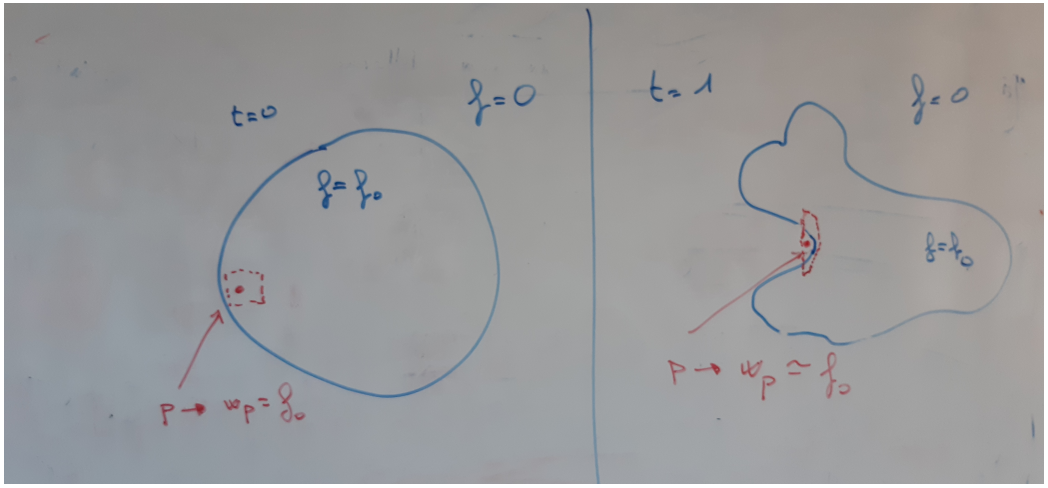






The PIC method does not calculate f

IMPORTANT: the "physics" interpretation of PIC is highly misleading, it hides the statistical nature of PIC. The weight of single particle IS NOT the value f in a point!



To represent f in a point you have to do the same operation you would do in nature (binning):

1) **Define a small 5D volume centred around the point with enough particles in it to diminish the error described above (many particles with small volumes).**
2) **Count how many physical particles are present in that volume (sum of the weights).**
3) **Divide the number of particles by the volume.**

Binning is noisy and expensive.

The PIC method in applied mathematics

The PIC method is a numerical technique (solid, fluids, cosmology…):

- **A continuous function, whose Lagrangian derivative is knows, is sampled using N individual, randomly chosen points (or fluid elements). Those points are tracked in continuous phase space via Euler-Lagrange equations. Up to 6N ODEs to solve (trivial).**
- **Moments of the distribution function are computed simultaneously on Eulerian (stationary) mesh points, charge assignment (less trivial, we use finite elements).**
- **It can be shown that the calculation of the moments is equivalent to a Monte-Carlo integration [Bottino and Sonnendruecker JPP 2015]. The "volume" V_p has now the meaning of "importance sampling".**
- **The closest we can be to calculate f in a point is ("binning"):**

$$f(\mathbf{x}) \simeq \frac{1}{V}\int_V \mathrm{d}\Omega f \qquad \text{Error} \propto \frac{\sigma}{\sqrt{N}}$$

This is clearly a statistical problem, with all the issues related to it.

The PIC/Finite element method in ORB5.

- Field solver and charge/current assignment: B-splines (finite elements).
- Example, Polarisation equation from a simple GK Lagrangian:

$$\frac{\delta L}{\delta \Phi}\cdot\delta\Phi = -\sum_{\mathrm{sp}}\int \mathrm{d}\Omega e J_0(\delta\Phi) f + \sum_{\mathrm{sp}}\int \mathrm{d}\Omega \frac{mc^2}{B^2} f_M \nabla_\perp \Phi\cdot\nabla_\perp\delta\Phi = 0 \quad \forall\delta\phi.$$

$$\sum_{\mathrm{sp}}\left(\int \mathrm{d}W e J_0^\dagger f + \nabla\cdot\frac{mn_0c^2}{B^2}\nabla_\perp\Phi\right) = 0$$

Finite element representation of potential: $\Phi_h(\mathbf{x},t) = \sum_{\mu=1}^{N_g}\Phi_\mu(t)\Lambda_\mu(\mathbf{x})$

Discrete Polarisation eq.:

$$\sum_{\mu=1}^{N_g}\Phi_\mu\sum_{\mathrm{sp}}\int \mathrm{d}\Omega\frac{f_M mc^2}{B^2}\nabla_\perp\Lambda_\nu\cdot\nabla_\perp\Lambda_\mu = \sum_{\mathrm{sp}}\frac{1}{N_p}\sum_{k=1}^{N_p} w_k(eJ_0\Lambda_\nu(\mathbf{R}_k)).$$

Set of linear equations to solve (LU decomposition):

$$\sum_\mu A_{\mu\nu}\Phi_\mu = b_\nu \qquad A_{\mu\nu} = \sum_{\mathrm{sp}}\int \mathrm{d}\Omega\frac{f_M mc^2}{B^2}\nabla_\perp\Lambda_\nu\cdot\nabla_\perp\Lambda_\mu \qquad b_\nu = \sum_{\mathrm{sp}}\frac{1}{N_p}\sum_{k=1}^{N_p} w_k(eJ_0\Lambda_\nu(\mathbf{R}_k)).$$

f(r,$v_{//}$,μ) using finite elements in ORB5

- **Define a new B-splines basis:**

$$\Lambda_\mu(\mathbf{x}) = \Lambda_{\mu 1}(r)\Lambda_{\mu 2}(v_\parallel)\Lambda_{\mu 3}(\mu)$$

- **Discrete f:**

$$f(r, v_\parallel, \mu, t) \simeq f_h(r, v_\parallel, \mu, t) = \sum_{\mu=1}^{N_g} f_\mu(t)\Lambda_\mu(\mathbf{x})$$

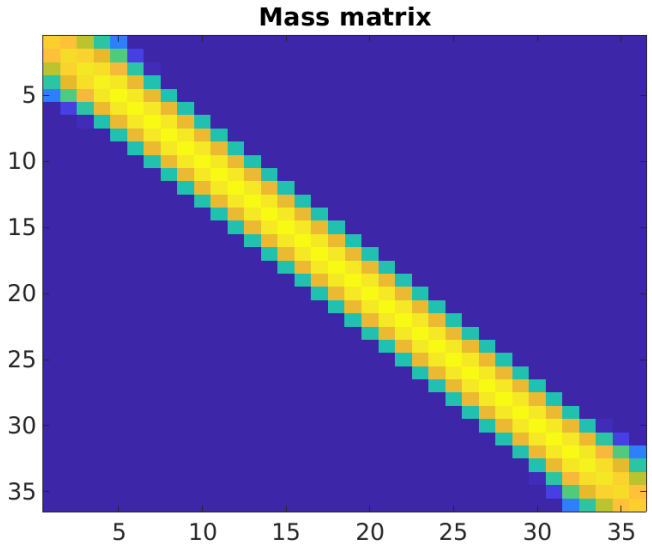
- **Equations to solve:**

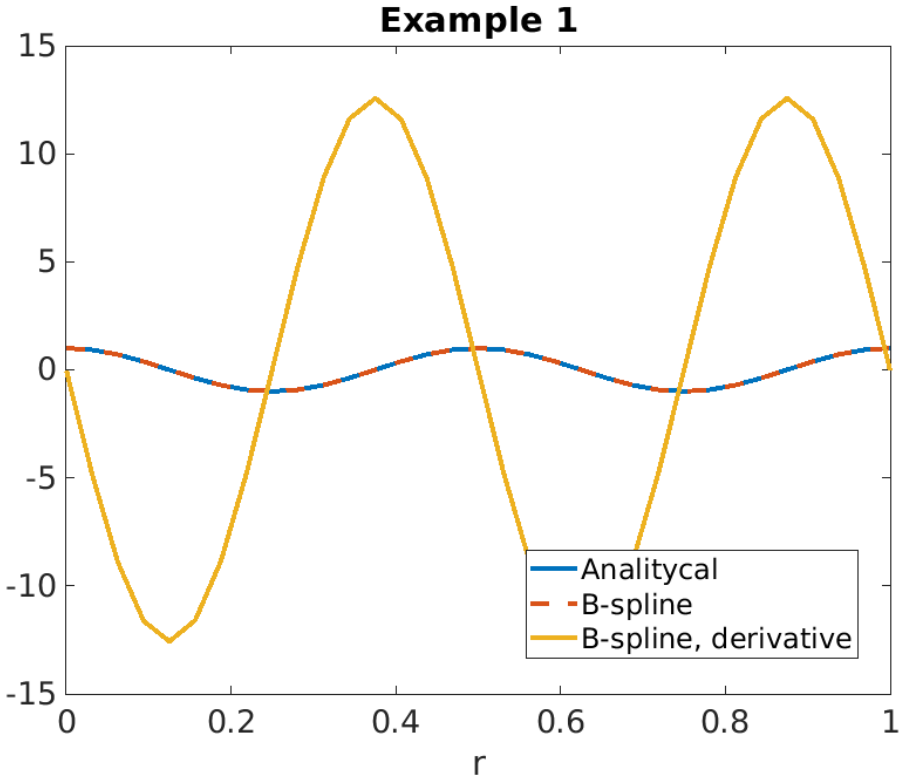
$$\sum_\mu A_{\mu\nu} f_\mu = b_\nu$$

$$A_{\mu\nu} = \int \mathrm{d}\Omega \Lambda_\nu \Lambda_\mu \qquad b_\nu = \frac{1}{N_p} \sum_{k=1}^{N_p} \left(w_k + f_0(r_k, v_{\parallel k}, \mu_k) V_k\right) \Lambda_\nu(r_k, v_{\parallel k}, \mu_k).$$

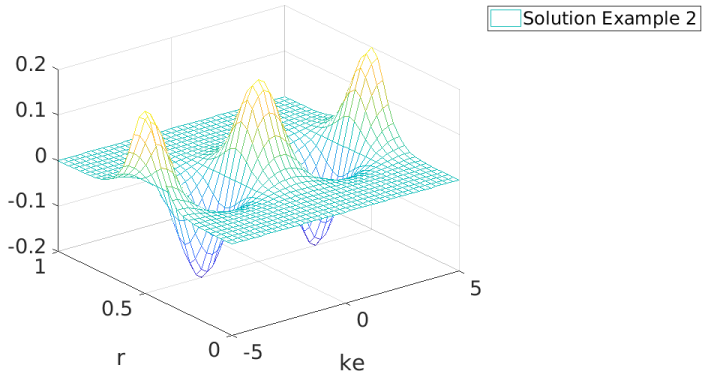
- **Caveat**: Matrix construction requires knowledge of the 5D metric (**Jacobian**). *Note: some approximations in the present version of ORB5.*
- New module in ORB5: pszs.f90: 1D, 2D and 3D projections on different coordinates (r,$v_{//}$,μ), (P_{phi},E_k,μ).., solver for **non periodic** coordinates.

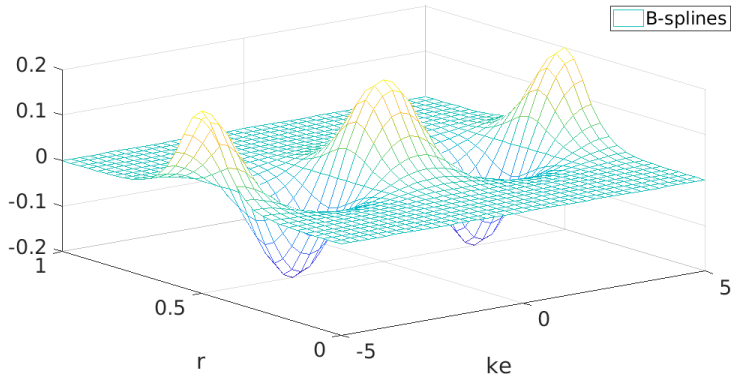
1D Example: nr=32; spline order=4

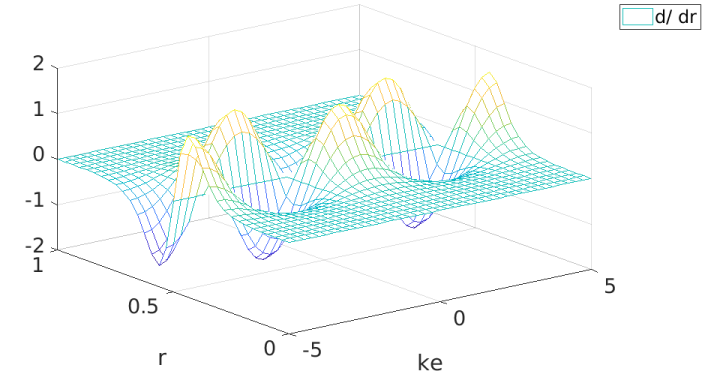


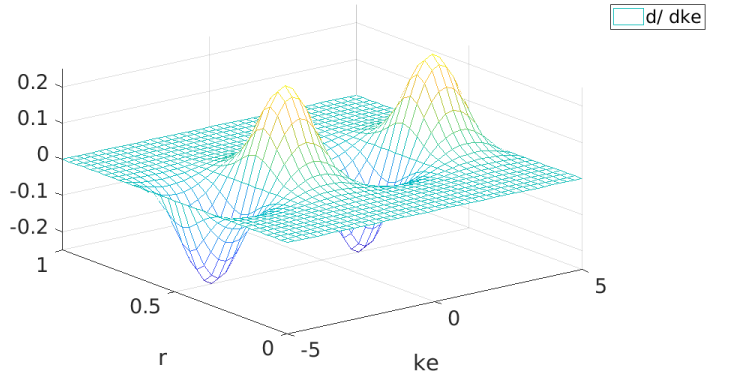


2D Example: nr=32, $nv_{//}$=40; spline order=4

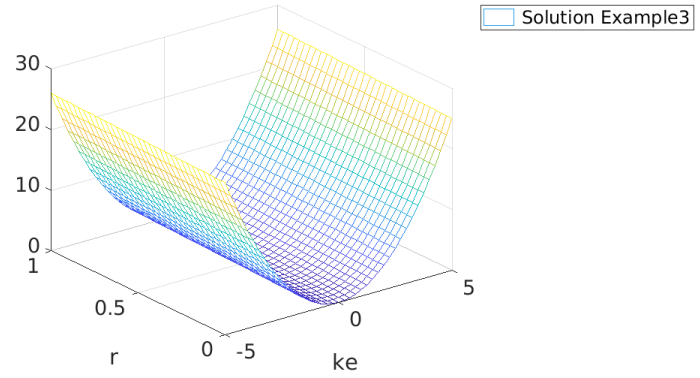


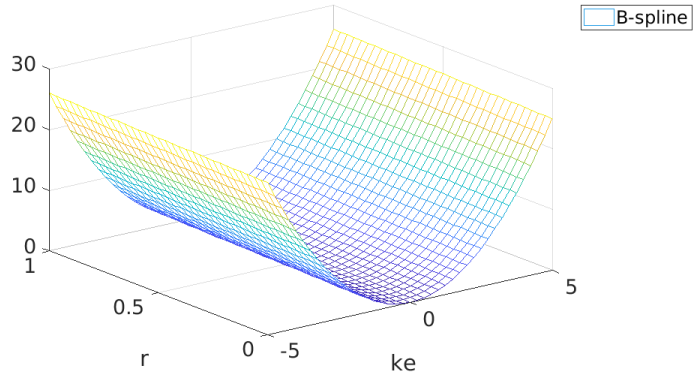


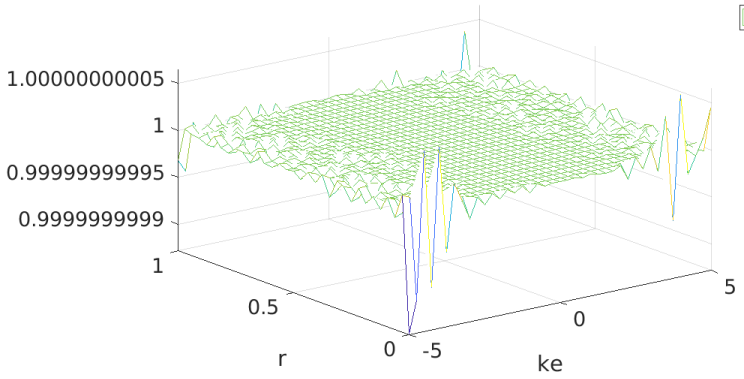


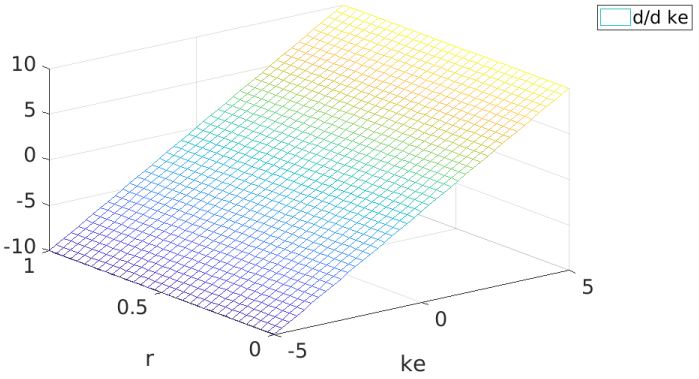


2D Example: nr=32, $nv_{//}$=40; spline order=4

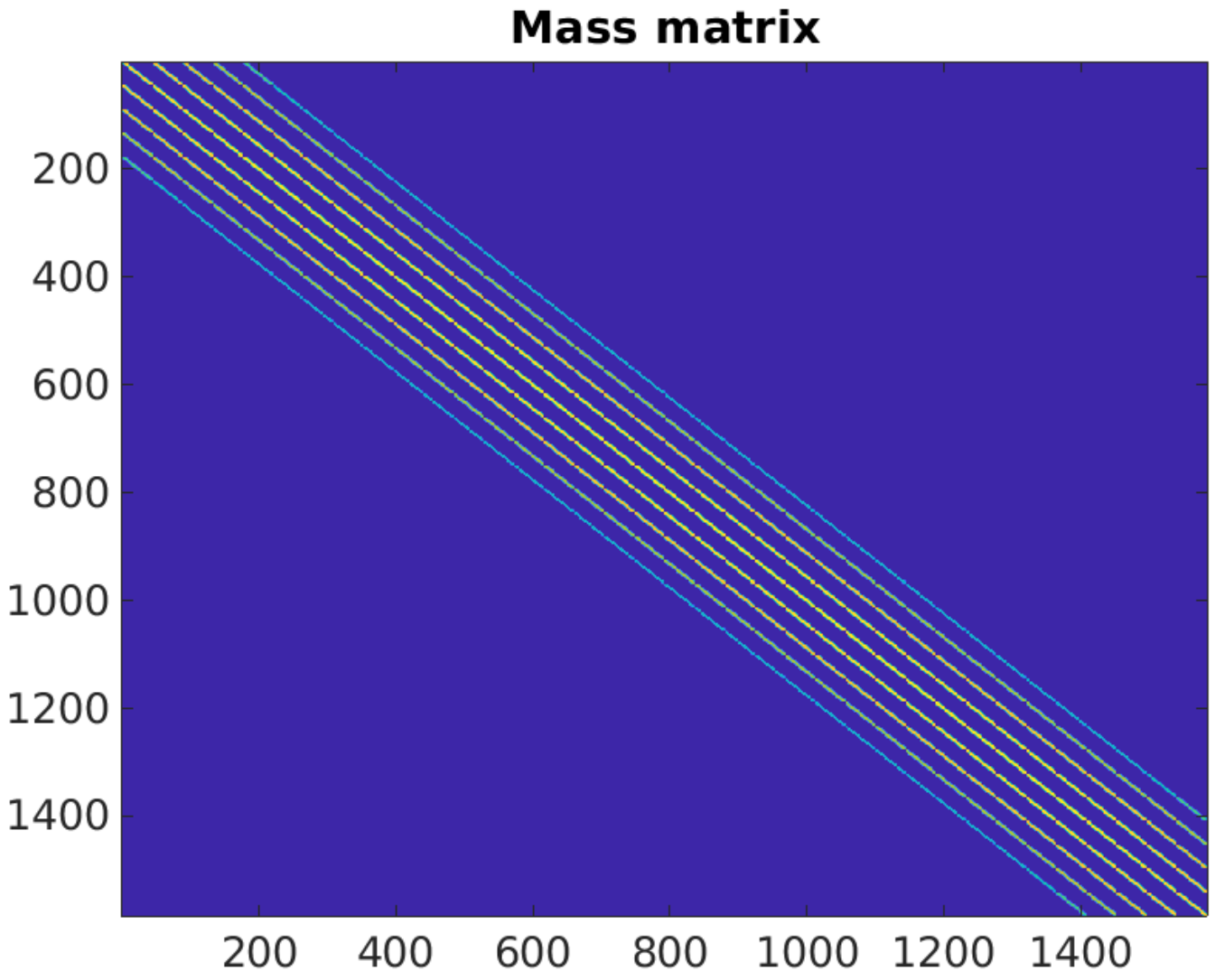




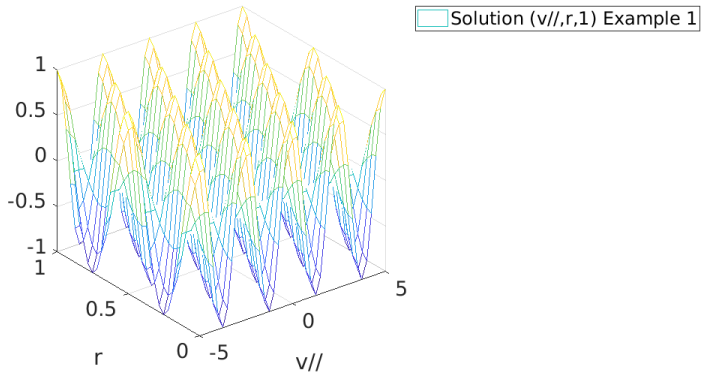


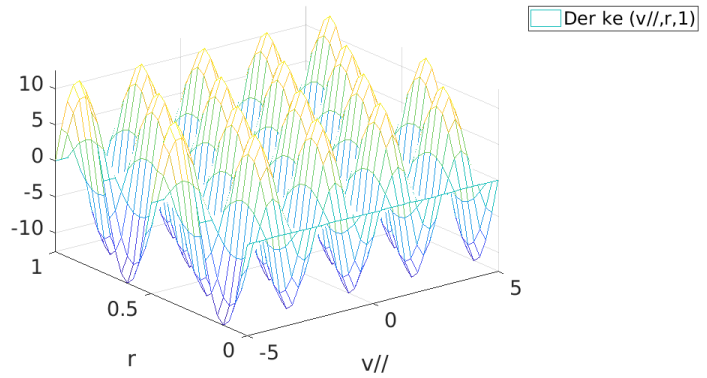


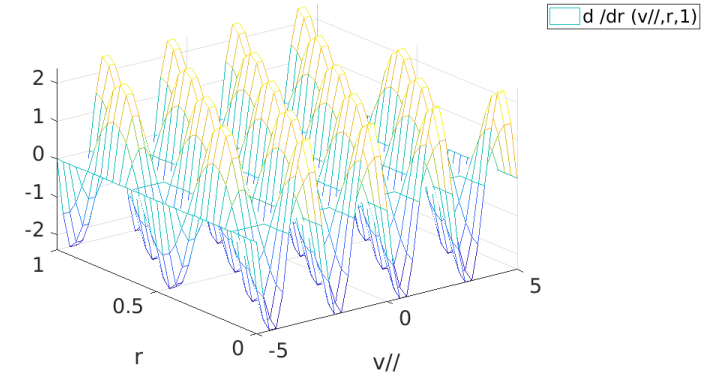
2D Example: nr=32, $nv_{//}$=40; spline order=4

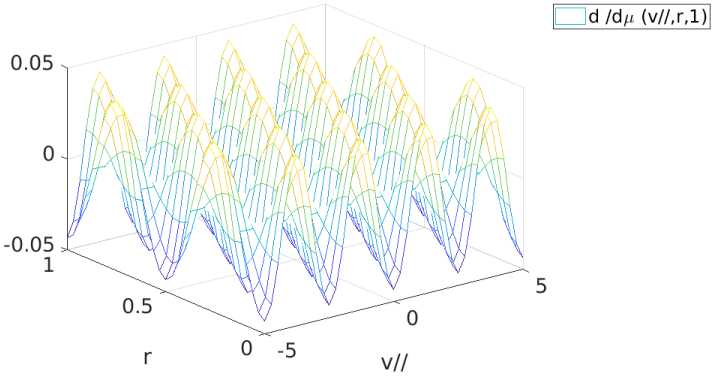


3D Example: nr=32, $nv_{//}$=40, nμ=12; spline order=4

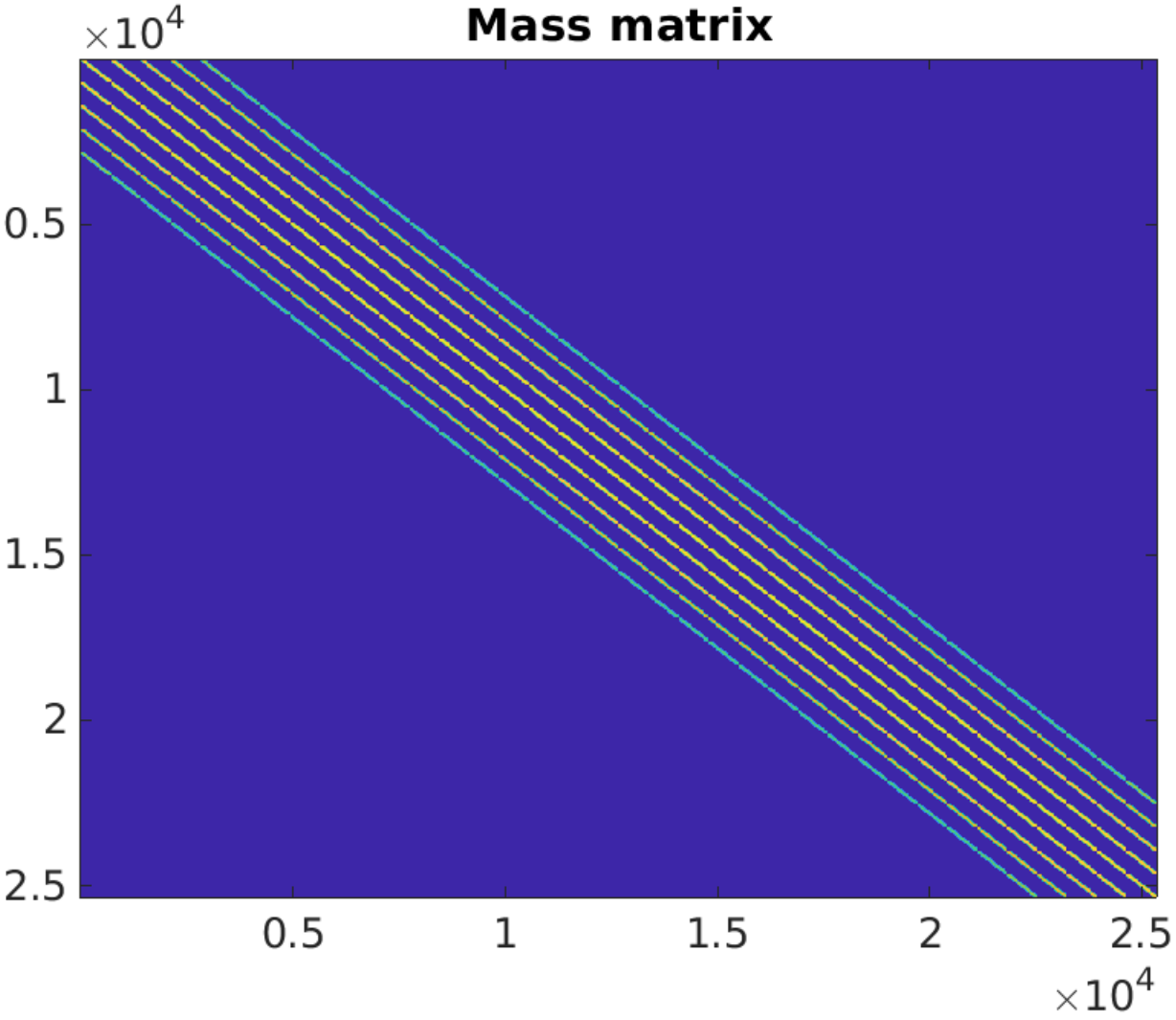








3D Example: nr=32, $nv_{//}$=40, nμ=12; spline order=4

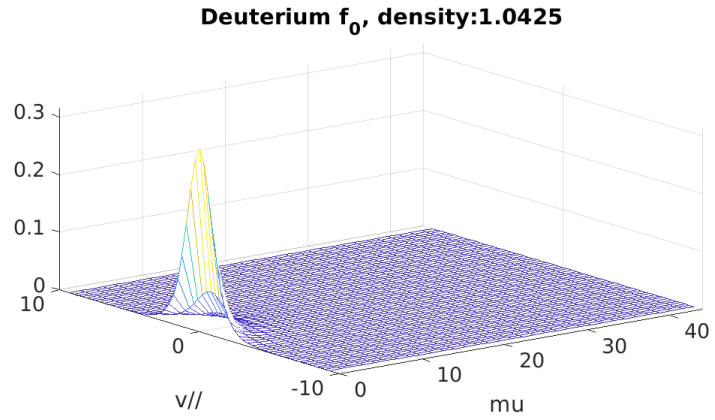


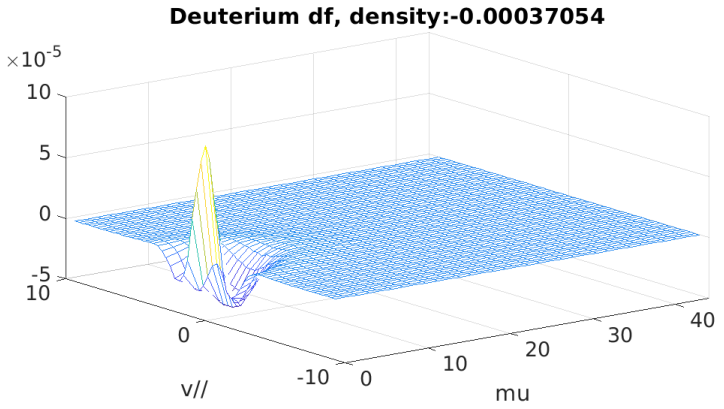
ORB5 defaults

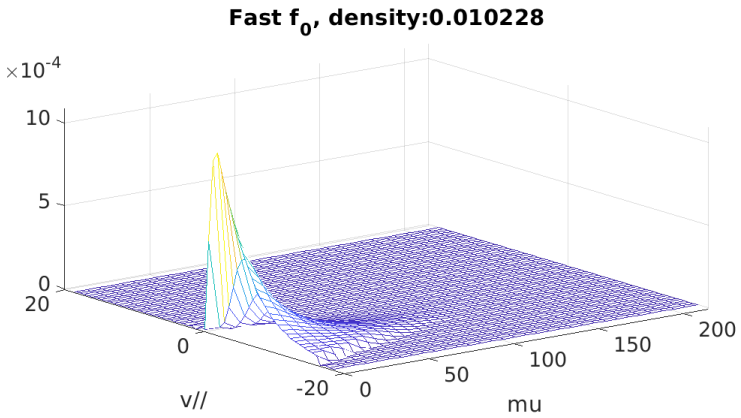
- 1D f(r) in hdf5 output on a grid of the size of number of splines.
- 2D f($v_{//}$,μ) in hdf5 output on a grid of the size of number of splines.
- 3D B-spline coefficients stored (no matrix construction).
- pszs3d can be used to construct the matrix, solve the linear algebra problem and construct f (r,$v_{//}$,μ) on a new grid of any size.
- to compile pszs3d:
  > make OPENMP=TRUE pszs3d.
- Tested up to: nr=80, nv//=100, nmu= 40 on a Raven node.

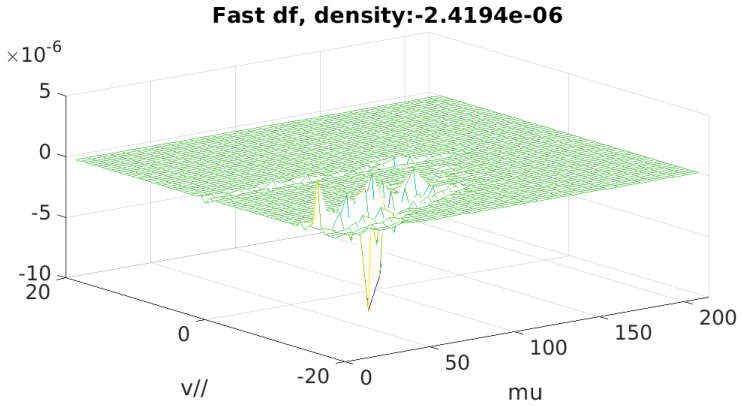
NLED-AUG: nr=32, $nv_{//}$=60, nμ=32; spline order=3

1) EP distribution: Anisotropic Slowing-down [Hayward-Schneider, Rettino]



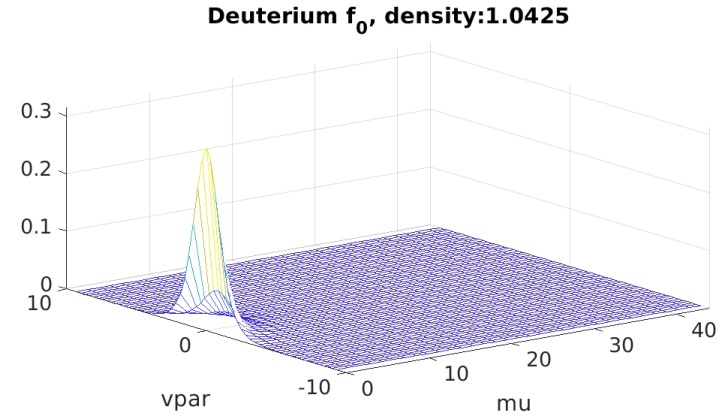


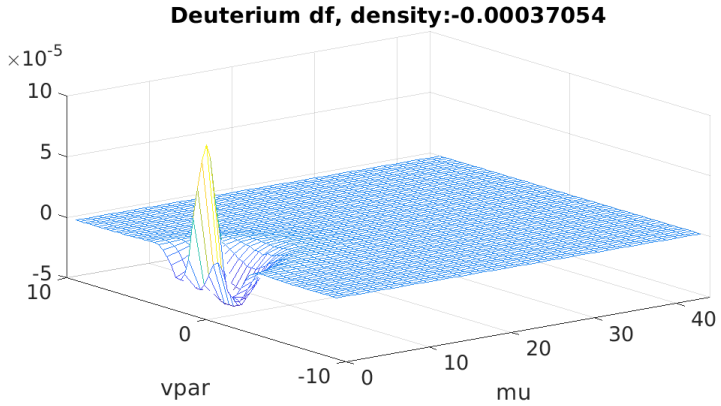


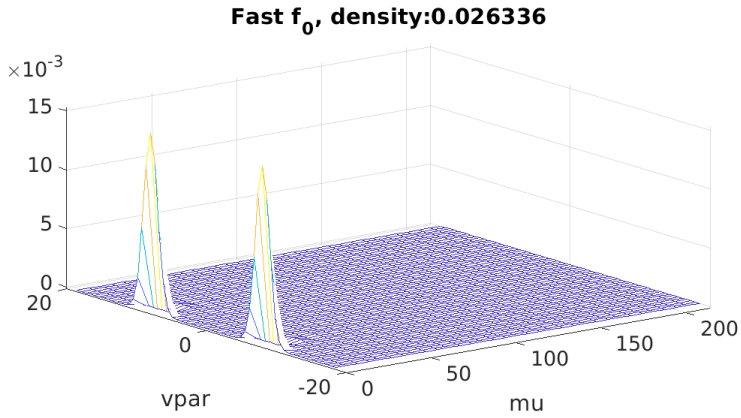


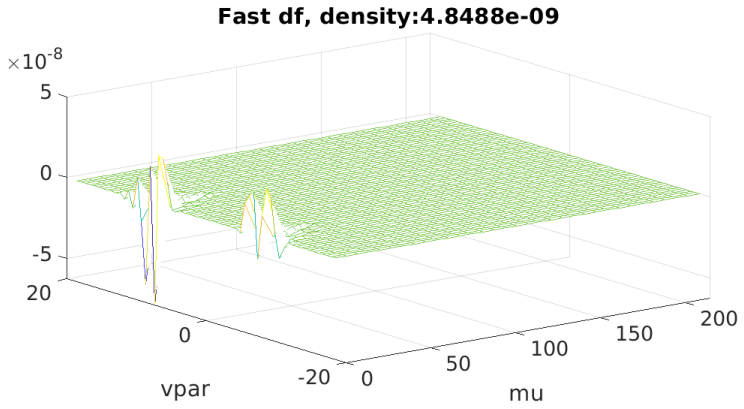
NLED-AUG: nr=32, $nv_{//}$=60, nμ=32; spline order=3

1) EP distribution: Bump-on-tail [Novikau, Vannini]









Outlook and to do list

- Check and improve the calculation of Jacobian in pszs3d.
- Extend the diagnostics to other possibly useful zonal quantities as power balance….
- Physics applications (ATEP).
- Compare with other codes…