Superfluid TDDFT for Fermi systems: TDSLDA - algorithms and their implementation

TDSLDA – Time-Dependent Superfluid Local Density Approximation

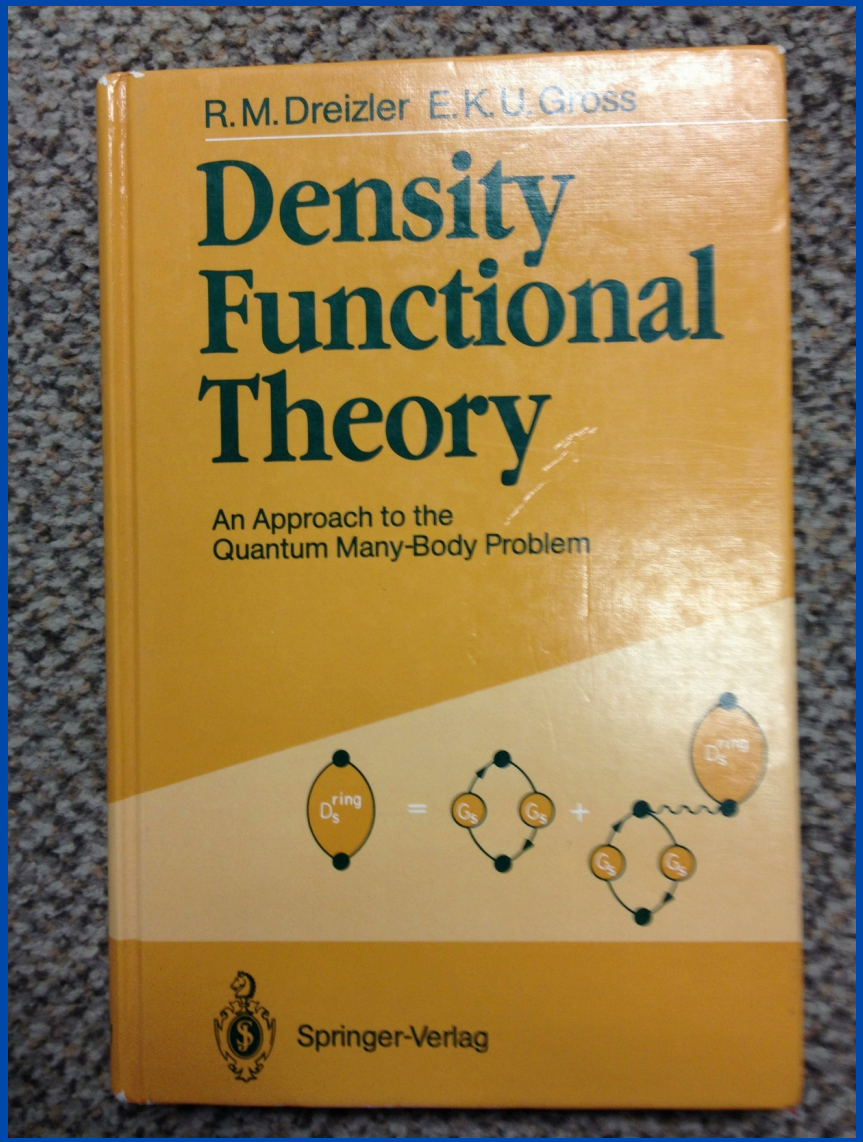
Aurel Bulgac
University of Washington, Seattle, WA

Collaborators:
Michael M. Forbes (Seattle -> Pullman)
Alan (Yuan-Lung) Luo (Seattle)
Piotr Magierski (Warsaw/Seattle)
Kenneth J. Roche (PNNL/Seattle)
Ionel Stetcu (UW -> LANL)
Gabriel Wlazlowski (Warsaw/Seattle)
Yongle Yu (Wuhan)

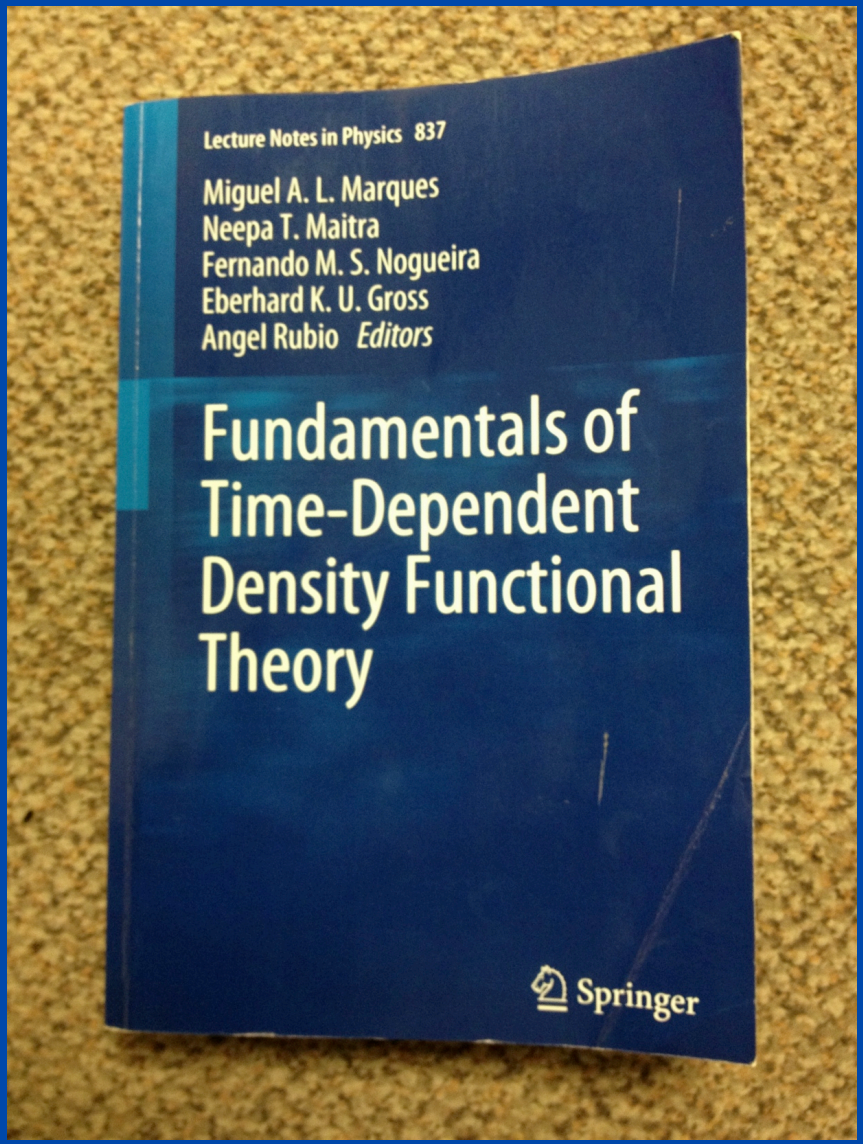
Funding: DOE, NSF, Polish Ministry of Science

The pptx version with movie can be found on speaker's webpage
http://www.phys.washington.edu/%7Ebulgac/talks.html#most_recent

Why Density Functional Theory (DFT)?



1990



2012

Near and long term goals:

To describe accurately the time-dependent evolution of externally perturbed Fermi superfluid systems

Tool: a DFT extension to superfluid systems and time-dependent phenomena (and subsequently we have to add quantum fluctuations and extend the theory to a stochastic incarnation)

Why TDDFT?
(not meant for debate during the talk, but for the afternoon and tomorrow discussions)

- **Unlike ATDHFB, TDSLDA does not requires introduction of hard to define collective degrees of freedom and there are no ambiguities arising from defining potential energy surfaces (PES) and inertias**
- **Currents are present by default, thus no ambiguity concerning their effects**
- **Interaction with basically any external probes (weak or strong) easy to implement and their effect on dynamics already included**
- **Accuracy of description in terms quantized trajectories with Maslov index corrections is at the same level of theoretical accuracy of a re-quantization of collective Hamiltonian**
- **One-body dissipation is automatically included in the formalism and this is a quantum approach (unlike diffusion models)**
- **TD equations are a consequence of the action minimum principle and thus the perils of using minimum of energy trajectories on PES and inertia are not encountered (see talk given by L. Robledo last week in which he demonstrated errors in time-lives up to $O(10^{30})$, thus errors up to ≈70 ħ in action)**
- **Overall computational effort in TDDSLDA is significantly less that in a ATHFB in a large collective space (five or more dimensions, which requires 5-9 million of configuration presently, and more in the future, see talk by A. Sierk last week) and more accurate numerically**
- **There are ways to include collective surface hopping and thus the effects of two-body collisions (surface hopping already studied for more than two decades in chemistry and now entering the condensed matter filed for normal systems)**

Physical systems and processes we are interested in:

- ✓ Collective states in nuclei
- ✓ Nuclear large amplitude collective motion (LACM)
  (Induced) nuclear fission
- ✓ Excitation of nuclei with gamma rays and neutrons
- ✓ Coulomb excitation of nuclei with relativistic heavy-ions
- ✓ Nuclear reactions, fusion between colliding heavy-ions
- ✓ Neutron star crust and dynamics of vortices and their pinning mechanism

- ✓ Dynamics of vortices, Anderson-Higgs Mode
- ✓ Vortex crossing and reconnection and the onset of quantum turbulence
- ✓ Domain wall solitons and shock waves in collision of fermionic superfluid atomic clouds

Outline:

- TDSLDA Equations
- Discrete Variable Representation of wave functions
- Calculation of derivatives and use of FFT
- Time propagation
- Calculation of Coulomb interaction
- Treatment of center of mass motion
- Gauge invariance, coupling to EM-field
- Construction of ground state:
  iterative methods vs time-dependent methods
- CPU vs GPU massively parallel implementation
  capability computing (not capacity computing)

Nuclear energy density functional

$$\varepsilon(\vec{r}) = \frac{\hbar^2}{2m_n}\tau_n(\vec{r}) - \Delta_n(\vec{r})\nu_n(\vec{r}) + \frac{\hbar^2}{2m_p}\tau_p(\vec{r}) - \Delta_p(\vec{r})\nu_p(\vec{r})$$

$$+ \sum_{T=0,1}\left[C_T^{\rho}\rho_T^2(\vec{r}) + C_T^{\Delta}\rho_T(\vec{r})\vec{\nabla}^2\rho_T(\vec{r}) + C_T^{\rho}\rho_0^{\gamma}(\vec{r})\rho_T^2(\vec{r}) \right]$$

$$+ \sum_{T=0,1}\left[C_T^{\tau}\left(\rho_T(\vec{r})\tau_T(\vec{r}) - \vec{j}_T^{\,2}(\vec{r})\right) + C_T^{\nabla J}\left(\rho_T(\vec{r})\vec{\nabla}\cdot\vec{J}(\vec{r}) + \vec{s}_T(\vec{r})\times\vec{j}_T(\vec{r})\right)\right]$$

$$+ \frac{e^2}{2}\int d^3r_1 \frac{\rho_p(\vec{r})\rho_p(\vec{r}_1)}{|\vec{r} - r_1|}$$

$$h(\vec{r}) = -\vec{\nabla}\cdot\left(\frac{\hbar^2}{2m(\vec{r})}\vec{\nabla}\right) + U(\vec{r}) + i\vec{\sigma}\cdot\vec{V}_{\sigma}(\vec{r}) + i\vec{\nabla}\cdot\vec{V}_{\nabla}(\vec{r}) + i\vec{W}(\vec{r})\cdot\vec{\sigma}\times\vec{\nabla}$$

Satisfies translational, rotational, isospin, and Galilean invariance, gauge covariance.
The coupling to electromagnetic field (not shown here) is gauge invariant minimal coupling.

Pairing part of the energy density functional

$$\varepsilon_S\left[\rho_n,\rho_p,\nu_p,\nu_n\right]=g(\rho_p,\rho_n)[|\nu_p|^2+|\nu_n|^2]$$

$$+f(\rho_p,\rho_n)[|\nu_p|^2-|\nu_n|^2]\quad\frac{\rho_p-\rho_n}{\rho_p+\rho_n}$$

where $\quad g(\rho_p,\rho_n)=g(\rho_n,\rho_p)$

and $\quad f(\rho_p,\rho_n)=f(\rho_n,\rho_p)$

TDSLDA equations

$$i\hbar\frac{\partial}{\partial t}\begin{pmatrix} u_{n\uparrow}(\vec{r},t) \\ u_{n\downarrow}(\vec{r},t) \\ v_{n\uparrow}(\vec{r},t) \\ v_{n\downarrow}(\vec{r},t) \end{pmatrix} = \begin{pmatrix} \hat{h}_{\uparrow\uparrow}(\vec{r},t)-\mu & \hat{h}_{\uparrow\downarrow}(\vec{r},t) & 0 & \Delta(\vec{r},t) \\ \hat{h}_{\downarrow\uparrow}(\vec{r},t) & \hat{h}_{\downarrow\downarrow}(\vec{r},t)-\mu & -\Delta(\vec{r},t) & 0 \\ 0 & -\Delta^*(\vec{r},t) & -\hat{h}^*_{\uparrow\uparrow}(\vec{r},t)+\mu & -\hat{h}^*_{\uparrow\downarrow}(\vec{r},t) \\ \Delta^*(\vec{r},t) & 0 & -\hat{h}^*_{\downarrow\uparrow}(\vec{r},t) & -\hat{h}^*_{\downarrow\downarrow}(\vec{r},t)+\mu \end{pmatrix} \begin{pmatrix} u_{n\uparrow}(\vec{r},t) \\ u_{n\downarrow}(\vec{r},t) \\ v_{n\uparrow}(\vec{r},t) \\ v_{n\downarrow}(\vec{r},t) \end{pmatrix}$$

- The system is placed on a large 3D spatial lattice (adequate representation of continuum)
- Derivatives are computed with FFTW (this insures machine accuracy) and is very fast
- Fully self-consistent treatment with fundamental symmetries respected (isospin, gauge, Galilean, rotation, translation)
- Adams-Bashforth-Milne fifth order predictor-corrector-modifier integrator
- No symmetry restrictions
- Number of PDEs is of the order of the number of spatial lattice points
  - from 10,000s to a fraction of 1,000,000 determined by the dimension of the Hilbert space

$$\propto 4\left(\frac{2p_cL}{2\pi\hbar}\right)^3 = 4N_xN_yN_z$$

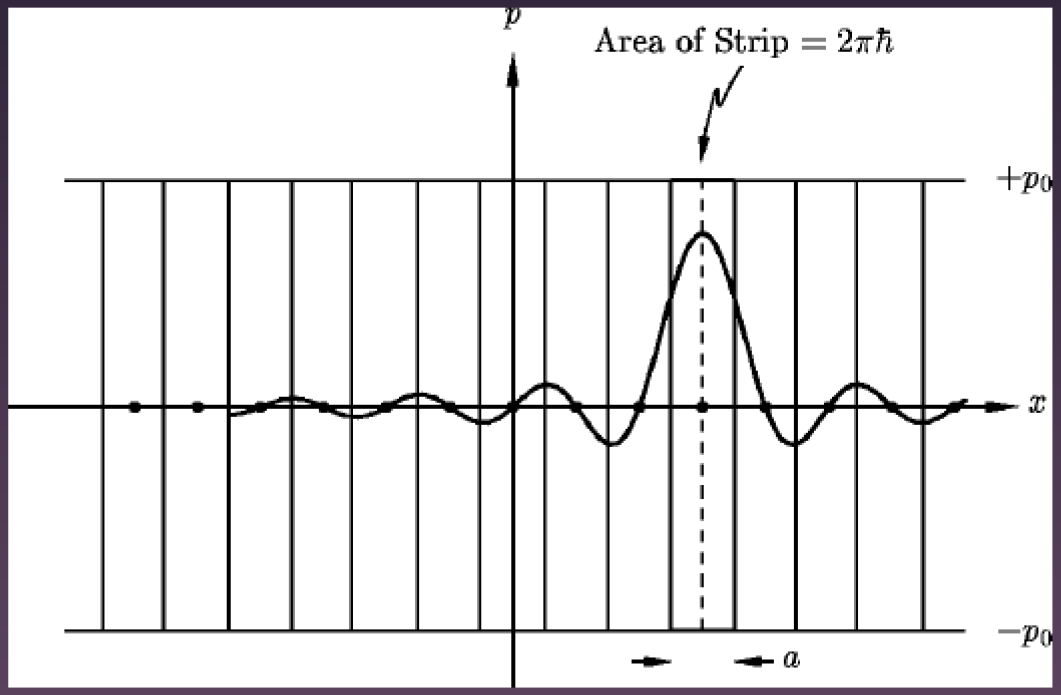
- Initial state is the ground state of the SLDA (formally like HFB/BdG)
- The code was implemented on Jaguar, Titan, Franklin, Hopper, Edison, Hyak, Athena
- Initially Fortran 90, 95, 2003 …, presently C, CUDA, and obviously MPI, pthreads, etc.

Discrete Variable Representation (DVR)

- Quasi-local (projected delta functions)
  - $F_n(x_m) \propto \delta_{mn}, \quad \langle F_m|V|F_n\rangle \approx \delta_{mn} V(x_n)$
- Analytic form for Kinetic Energy
- Exponential convergence
  - for appropriate potentials, boundary conditions etc.

All these dark slides are from an INT talk, given on April 8, 2013 by M.M. Forbes and based on our paper A. Bulgac and M.M. Forbes, Phys. Rev. C 87, 051301(R) (2013)

Phase-Space Coverage

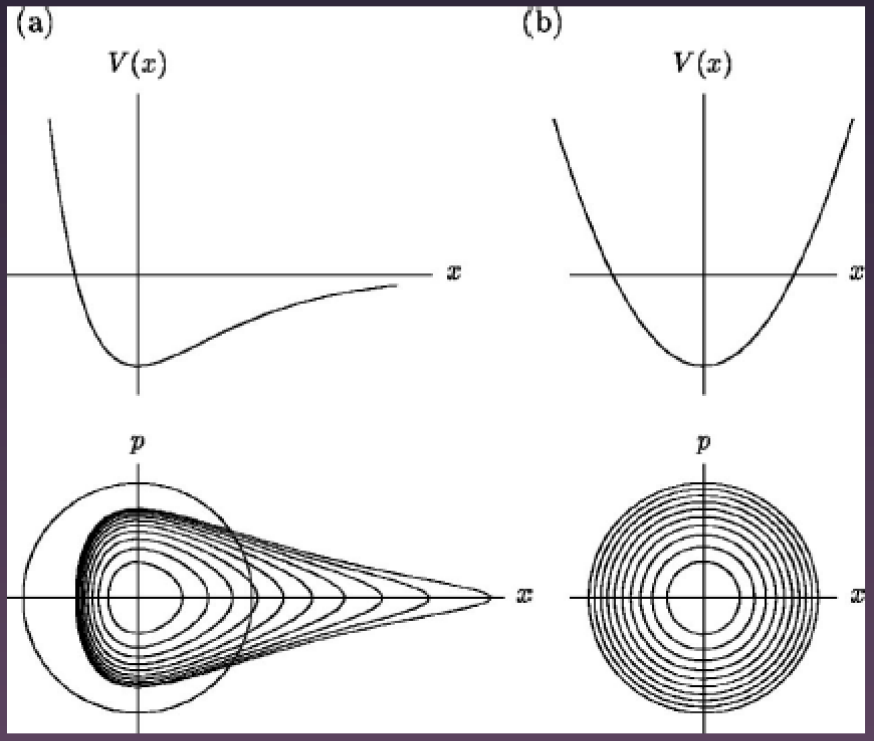


DVR basis slices phase space into strips

Efficient coverage of typical rectangular "model spaces" with simple IR and UV cutoffs

Littlejohn et al. J. Chem. Phys. 116 (2002) 8691

Phase-Space Coverage



Littlejohn et al. J. Chem. Phys. 116 (2002) 8691

For convergence:

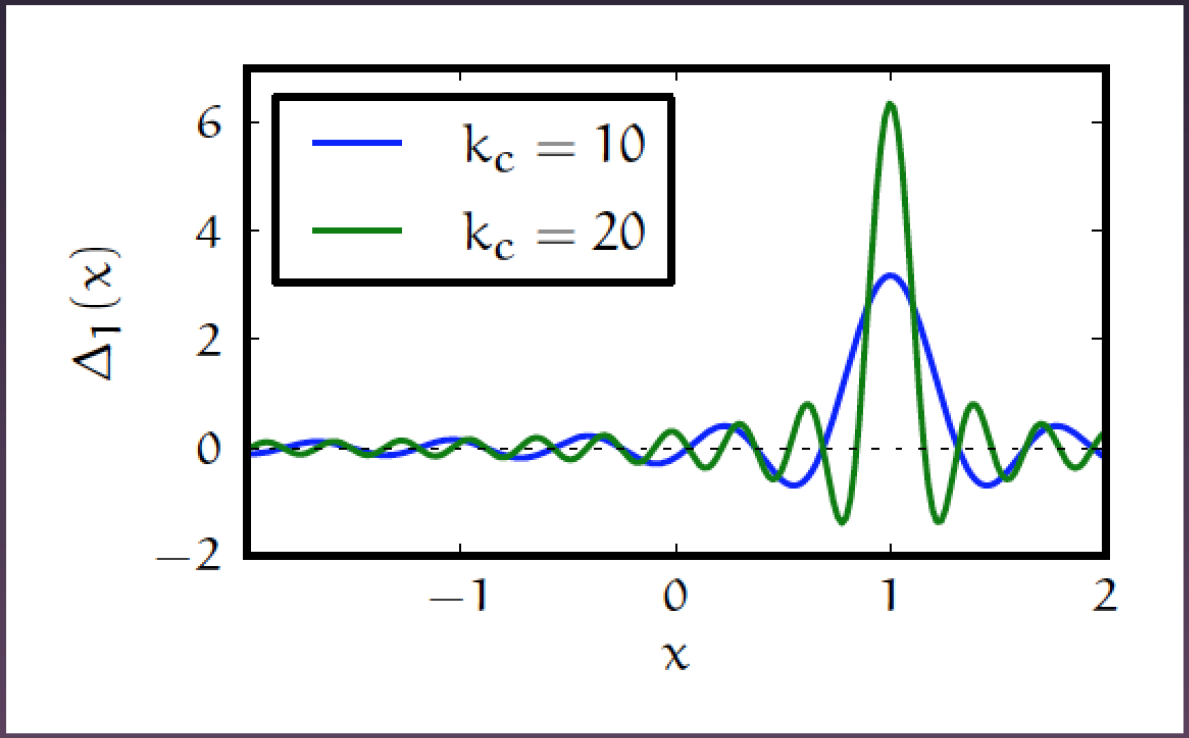
- Must cover same semi-classical phase space
- Consider modeling the Morse (left) potential with HO basis (right)

Standard DVR Basis

$$\Delta_n(x) \propto F_n(x) \propto L_n(x)$$

- Projected delta-functions: $\Delta_n(x)$
  - Let $\langle x|x_n\rangle = \delta(x-x_n)$, then $|\Delta_n\rangle = P|x_n\rangle$
- Interpolating functions: $L_n(x)$
  - $|f\rangle = \sum_n f(x_n)\ |L_n\rangle$
- Orthonormal basis functions: $F_n(x)$
  - $\langle F_m|F_n\rangle = \delta_{mn}$

Projected Delta Functions



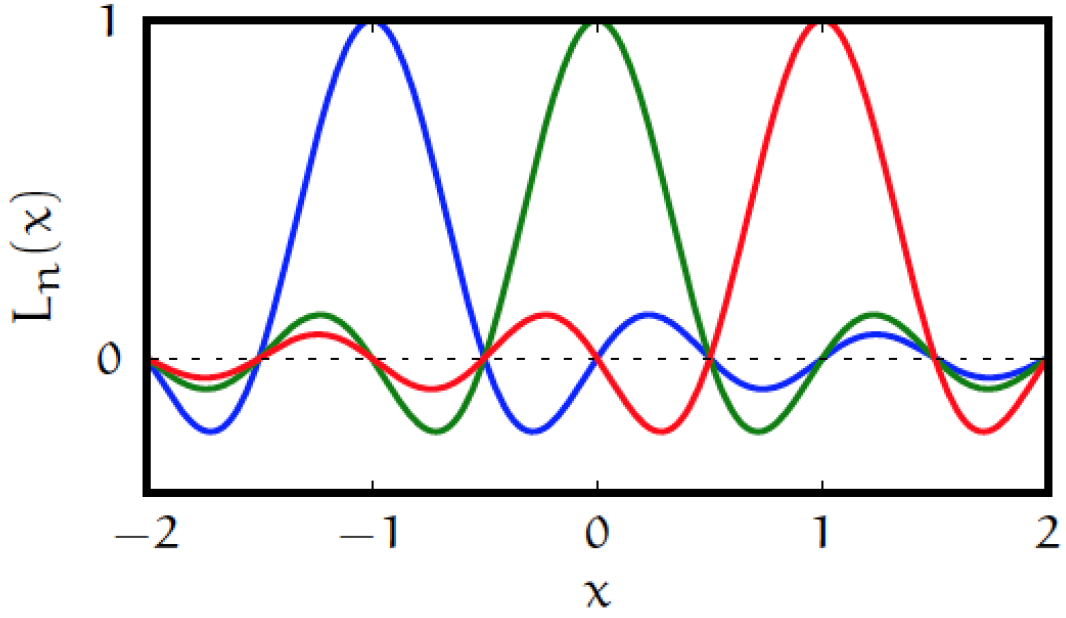
$$P = \sum_{k<k_c} |k\rangle \langle k|$$

$$\langle x|x_n\rangle = \delta(x - x_n)$$

$$|\Delta_n\rangle = P\,|x_n\rangle$$

$$|F_n\rangle = |\hat{\Delta}_n\rangle = \frac{|\Delta_n\rangle}{\sqrt{\langle\Delta_n|\Delta_n\rangle}}$$

Non-trivial Consistency of Abscissa

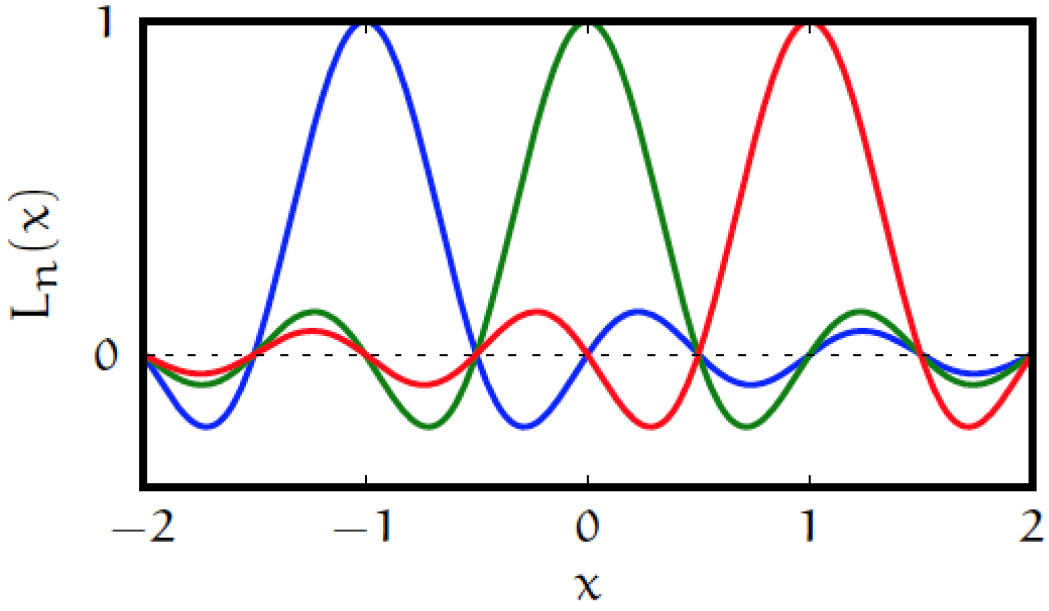


Abscissa must be nodes of $\Delta_m(x)$

$$\Delta_m(x_n) = \delta_{mn}/w_n$$

Associated with orthogonal polynomials

Interpolating Functions



Just evaluate $f(x_n)$ at the abscissa:

$$|f\rangle = \sum_n f_n |F_n\rangle$$
$$= \sum_n f(x_n) |L_n\rangle$$

Integration Weights

- $w_n = 1/\langle \Delta_n | \Delta_n \rangle = 1/\Delta_n(x_n)$
- $L_n(x) = w_n^{1/2} F_n(x) = w_n \Delta_n(x)$
- Gaussian quadrature wights for functions in basis:
  - $\langle f|g \rangle = \sum_n w_n f^*(x_n) g(x_n)$
- But... make sure to integrate functions in basis (or add more abscissa)

Diagonal Potential Energy

$$\langle F_m | V | F_n \rangle \approx \delta_{mn} V(x_m)$$

- Not exact, but eigenvalues and eigenvectors still have exponential convergence
- No overlap integrals needed
- Trivial 3 and 4-body operators

Analytic Kinetic Energy

$$K_{mn} = \langle F_m | \frac{\hbar^2 \nabla^2}{2m} | F_n \rangle, \quad K_{mn} = \langle F_m | \frac{\hbar^2}{2m} \frac{d^2}{dr^2} - \frac{\nu^2 - \frac{1}{4}}{r^2} | F_n \rangle$$

- Include singularities here
  - They can spoil convergence

DVR Examples

- Fourier basis (rearrangement): use FFTW

$$L_n(x) = \frac{\sin k_c(x - x_n)}{N \sin \frac{k_c(x-x_n)}{N}} = \frac{1}{N} \sum_{m=-(N-1)/2}^{(N-1)/2} e^{ik_m(x-x_n)}$$

$$K_{m\neq n} = \frac{2\pi^2(-1)^{m-n}}{L^2} \frac{\cos \frac{k_c(x_m-x_n)}{N}}{\sin^2 \frac{k_c(x_m-x_n)}{N}},$$

$$K_{nn} = \frac{\pi^2}{3a^2}\left(1 - \frac{1}{N^2}\right).$$

DVR Examples

- Sinc function basis
- K_{mn} dense (but only in each dimension)

$$L_n(x) = \mathrm{sinc}\Big(k_c(x - x_n)\Big)$$

$$K_{m\neq n} = \frac{2(-1)^{m-n}}{(x_m - x_n)^2}, \qquad K_{nn} = \frac{\pi^2}{3a^2}.$$

In 3D the sparcity of the Hamiltonian matrix is ≈$1/N^2$, where N is the number of lattice points in 1D

DVR Examples

- Bessel Function Basis: Spherical/Cylindrical symmetry

$$J_\nu(z_{\nu n}) = 0, \quad w_n = \frac{2}{k_c z_{\nu n} J'_\nu(z_{\nu n})^2},$$

$$F_n(r) = (-1)^{n+1} \frac{k_c z_{\nu n} \sqrt{2r}}{k_c^2 r^2 - z_{\nu n}^2} J_\nu(k_c r),$$

$$K_{m \neq n} = \frac{8 k_c^2 (-1)^{m-n} z_{\nu n} z_{\nu m}}{(z_{\nu n}^2 - z_{\nu m}^2)^2},$$

$$K_{nn} = \frac{k_c^2}{3}\left[1 + \frac{2(\nu^2 - 1)}{z_{\nu n}^2}\right]$$

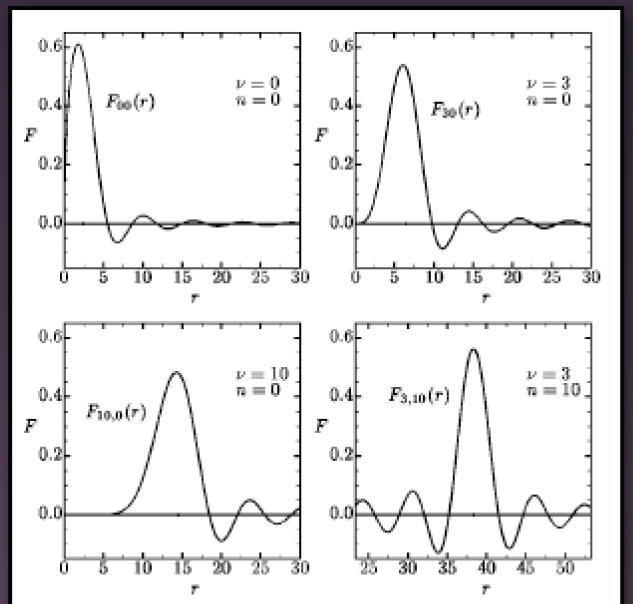


FIG. 2. Plots of the Bessel DVR functions $F_{\nu n}(r)$ for $K = 1$ and for selected values of ν and n.

DVR Examples

- Bessel Function Basis: Spherical/Cylindrical symmetry
- In principle: one basis for each l
- In practice (3D):
  - use l=0, for even l
  - use l=1, for odd l
- (may need extra point to represent densities etc.)

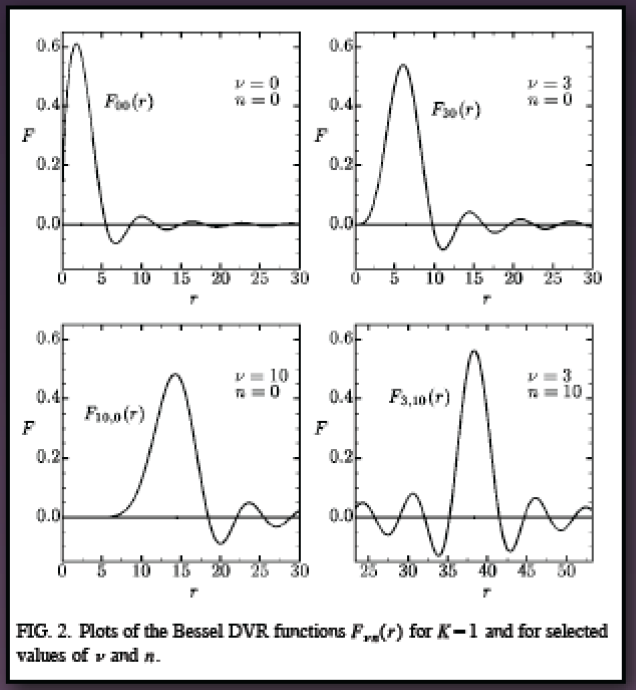


FIG. 2. Plots of the Bessel DVR functions $F_{\nu n}(r)$ for $K = 1$ and for selected values of ν and n.

Simple Implementation

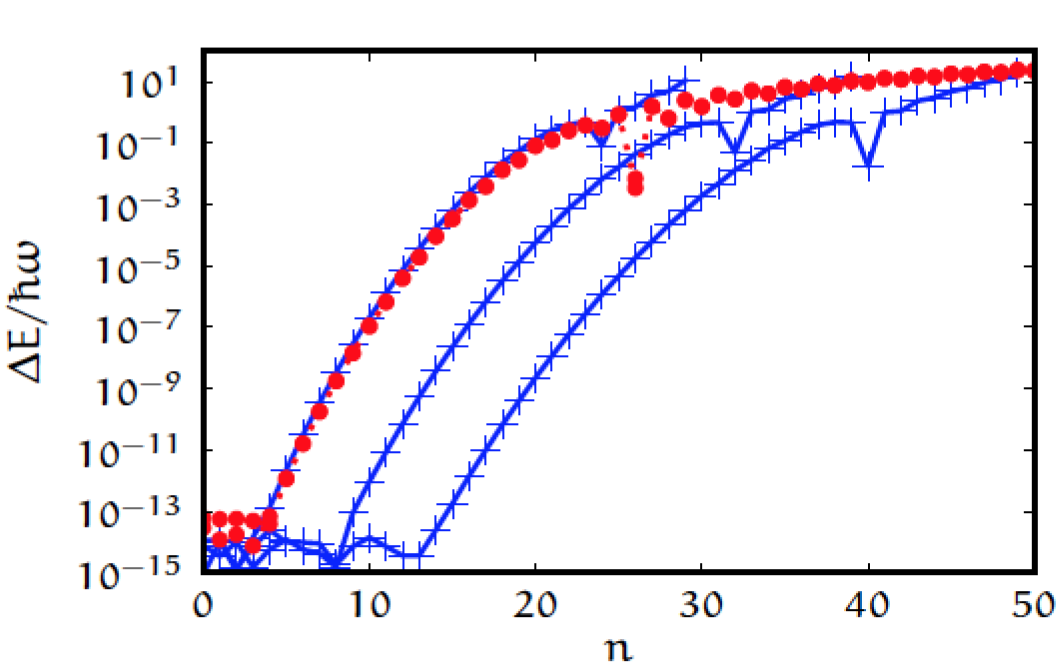
- MATLAB

```
N    = 32;
n    = (1:N);
[k,l] = meshgrid(n,n);
a    = 1;   %  lattice constant
x    = a*(-N/2:1:N/2-1)';
V    = x.^2/2;   %potential
p    = 2*pi/L*(-N/2:1:N/2-1);

Tk  = 2*(-1).^(k-l)./((sin(pi*(k-l)/N)).^2 + eps)/N^2;
Tk  = Tk - diag(diag(Tk));
Tk  = (Tk + eye(N)*(1+2/N^2)/3)*pi^2/a^2/2;
H   = Tk +diag(V);
energy  = eig(H);
```

- Python

```
class DVR1D(object):
    r"""Sinc function basis for non-periodic functions over
     an interval `xo +- L/2` with `N` points."""
    def __init__(self, N, L, xo=0.0):
        L = float(L)
        self.N = N
        self.L = L
        self.xo = xo
        self.a = L/N
        self.n = np.arange(N)
        self.x = self.xo + self.n*self.a - self.L/2.0 + self.a/2.0
        self.k_max = np.pi/self.a

    def H(self, V):
        """Return the Hamiltonian with the give potential."""
        _m = self.n[:, None]
        _n = self.n[None, :]
        K = 2.0*(-1)**(_m-_n)/(_m-_n)**2/self.a**2
        K[self.n, self.n] = np.pi**2/3/self.a**2
        K *= 0.5   # p^2/2/m
        V = np.diag(V(self.x))
        return K + V
```

HO Eigentstates with DVR basis



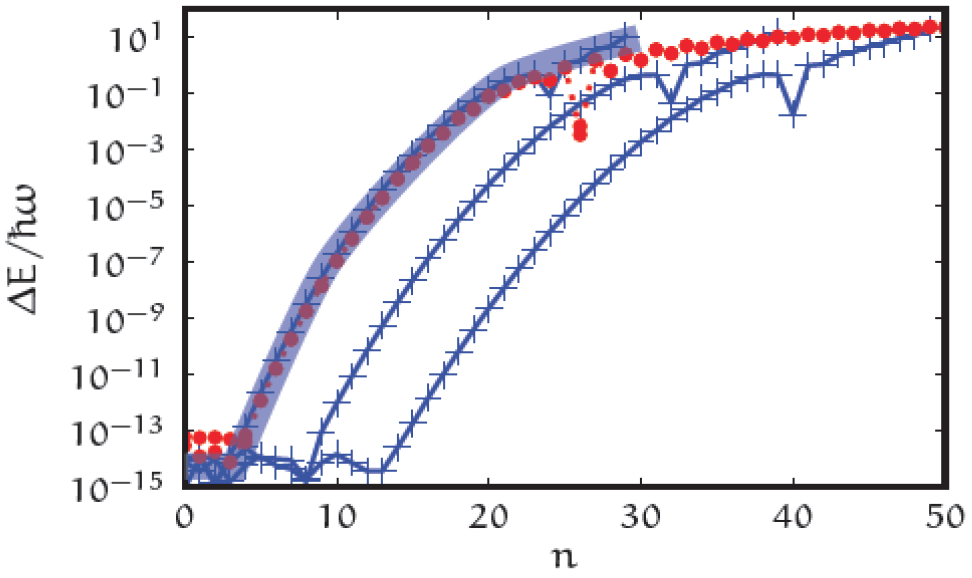
Ho potential with optimally tuned DVR basis

Bulgac & Forbes arXiv:1301.7354

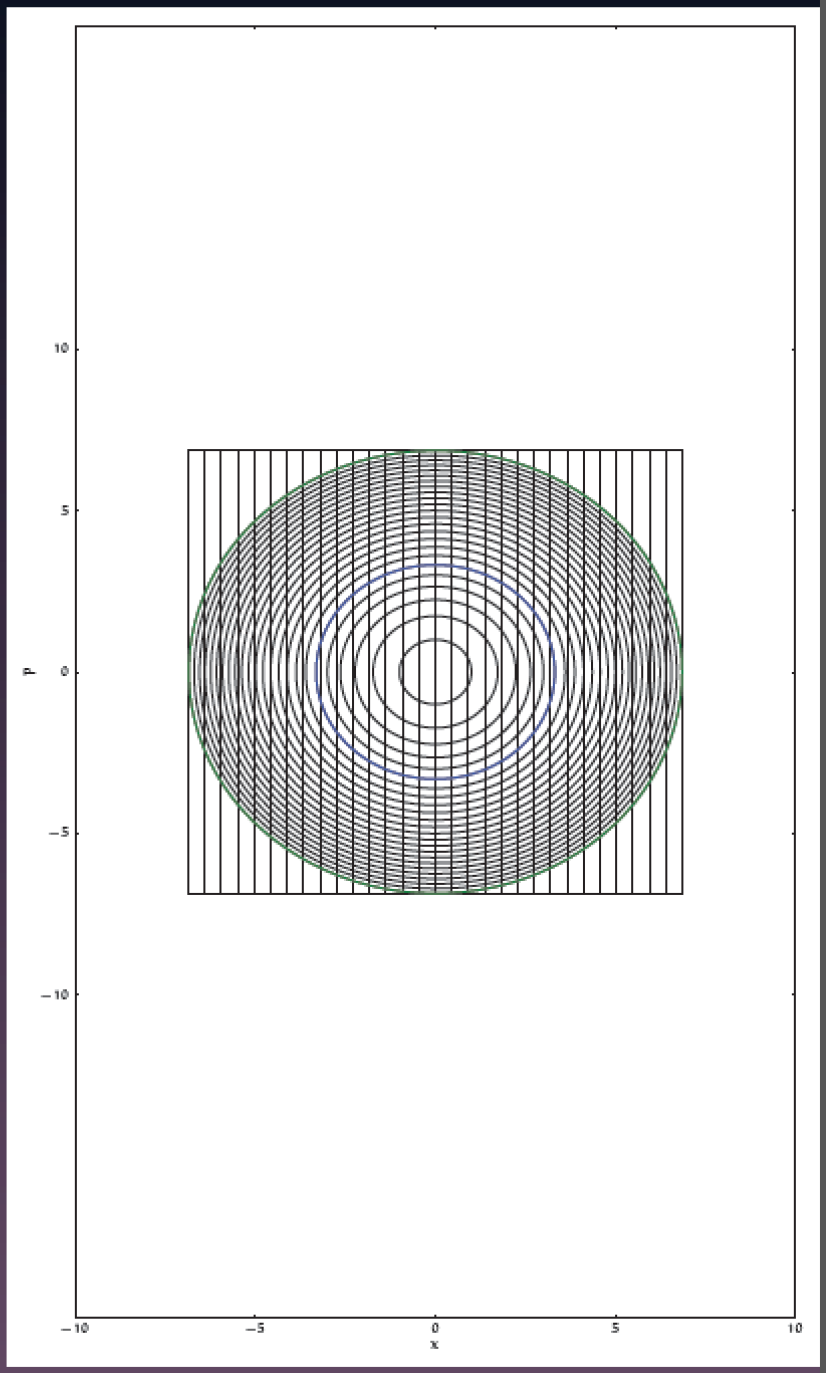
N=L=30

Optimal phase space coverage

- 5 energies to machine precision
- 24 reasonable energies (10%)



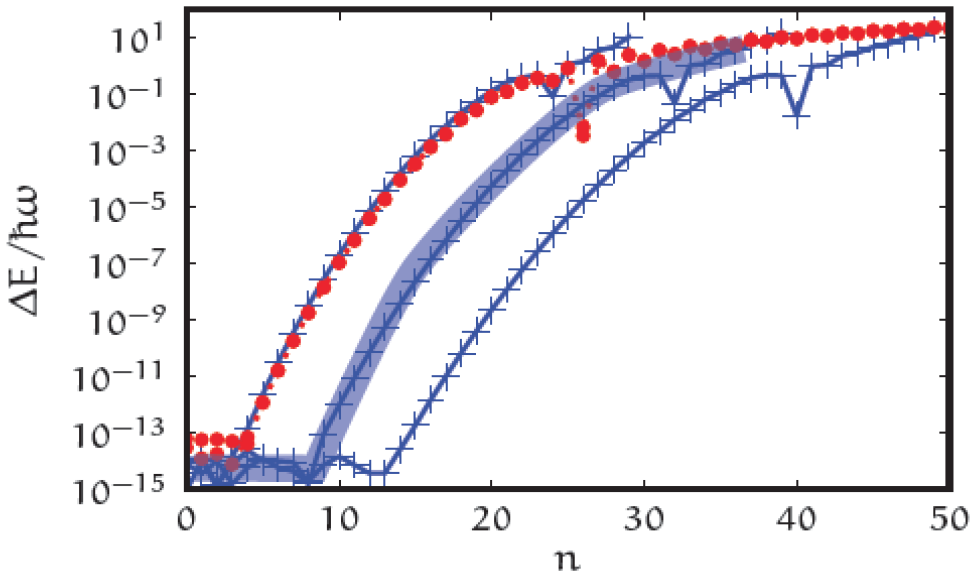
Bulgac & Forbes arXiv:1301.7354



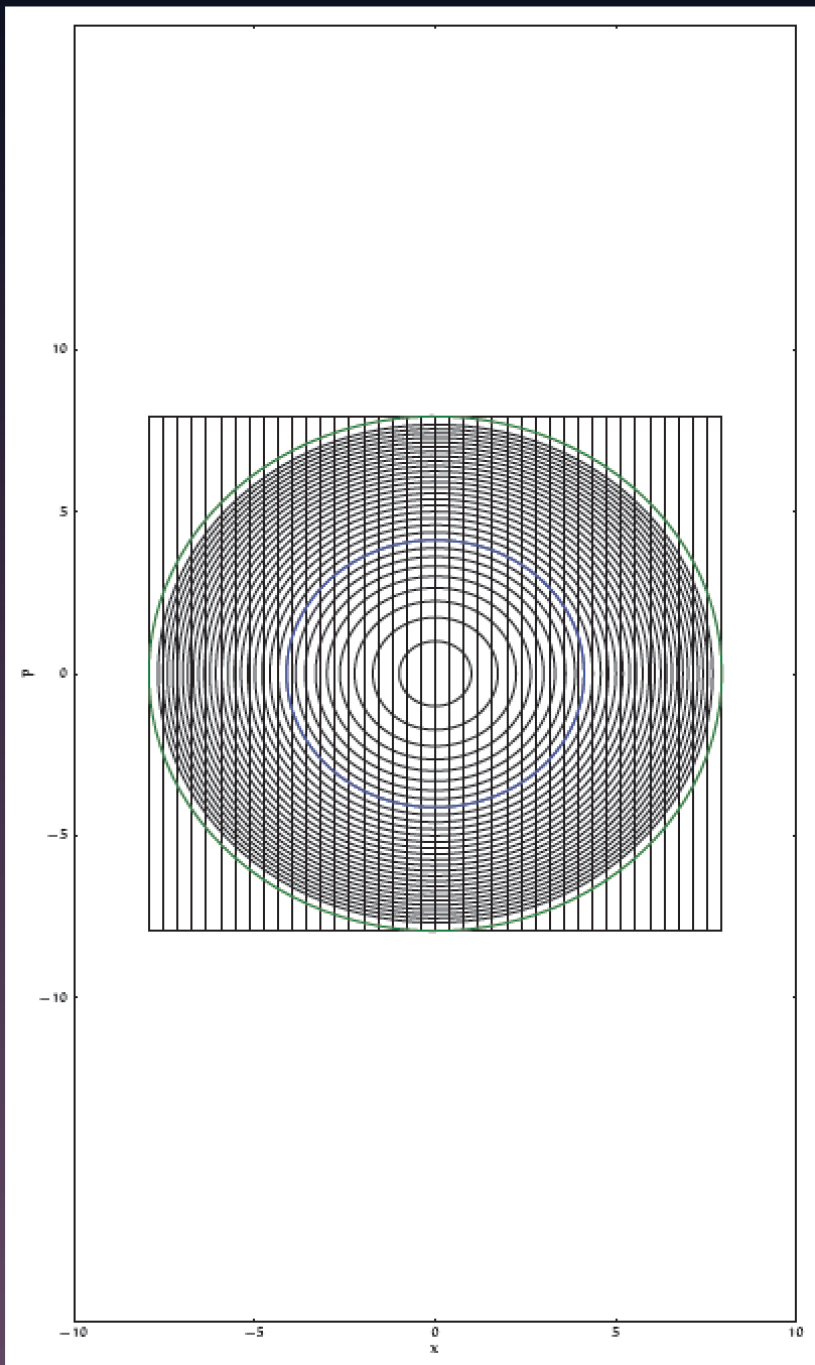
N=L=40

Optimal phase space coverage

- 8 energies to machine precision
- 32 reasonable energies (10%)



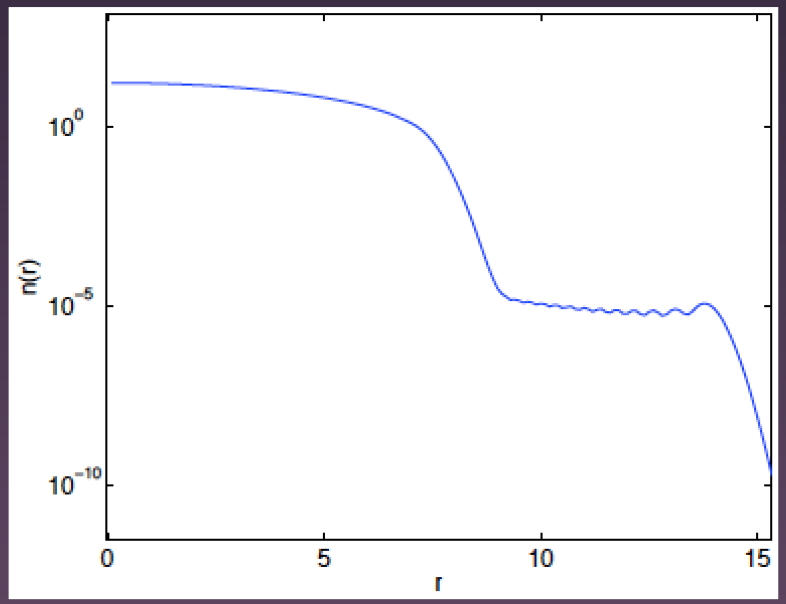
Bulgac & Forbes arXiv:1301.7354



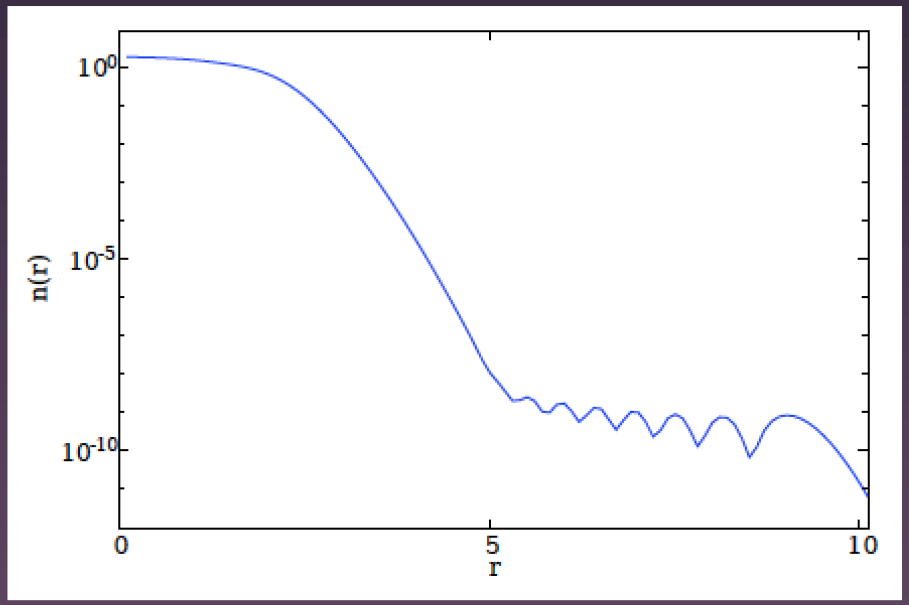
Difficulties with HO Basis

- Large Radius of HO wavefunctions introduce artifacts
- Need large number of states to correct
- (Requires HO wavefunction to high precision!)

Grasso and Urban, BCS Code



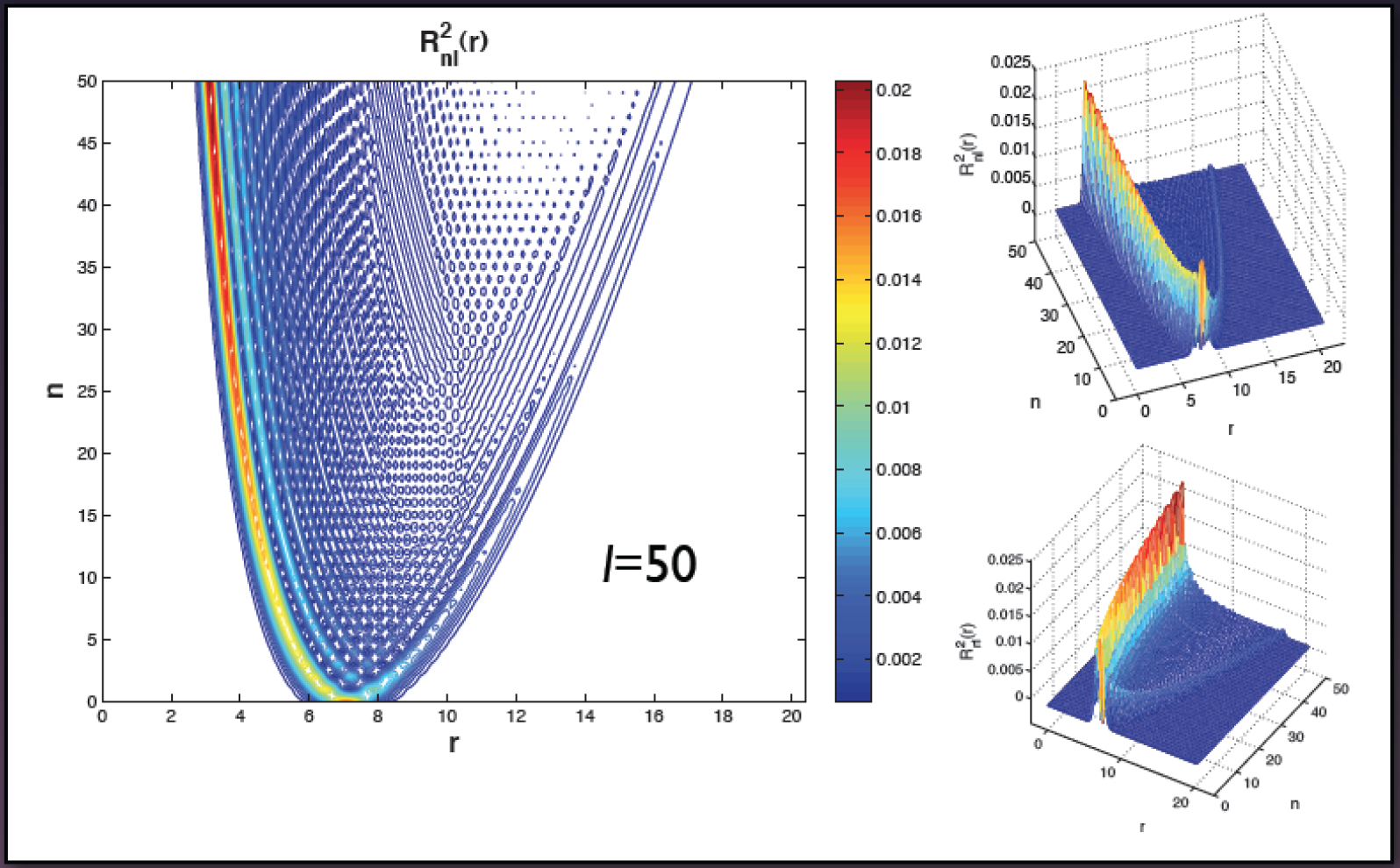
Our Unitary HO Basis Code



Grasso and Urban, PRA, 68, 033610 (2003)

Difficulties with HO Basis

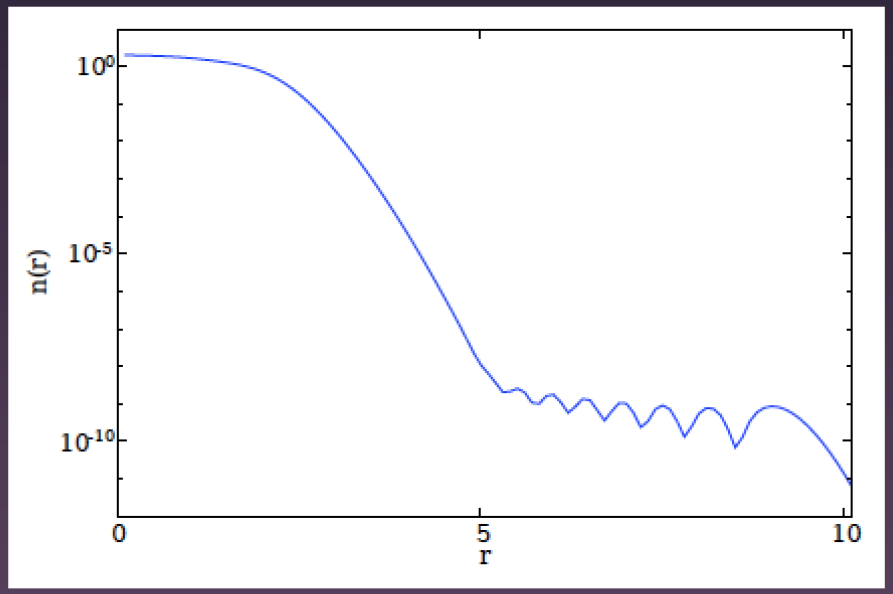
- Tails (turning points) spoil large r behaviour



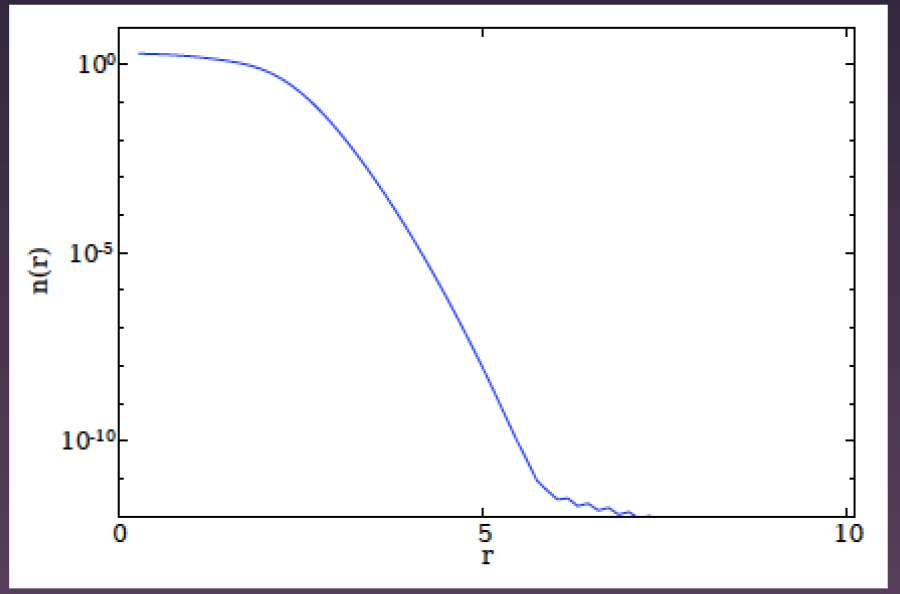
DVR Solves the Problem

- Tails spoil large r behaviour

Our code with HO Basis



Our code with DVR Basis



Difficulties with HO Basis Complex Convergence

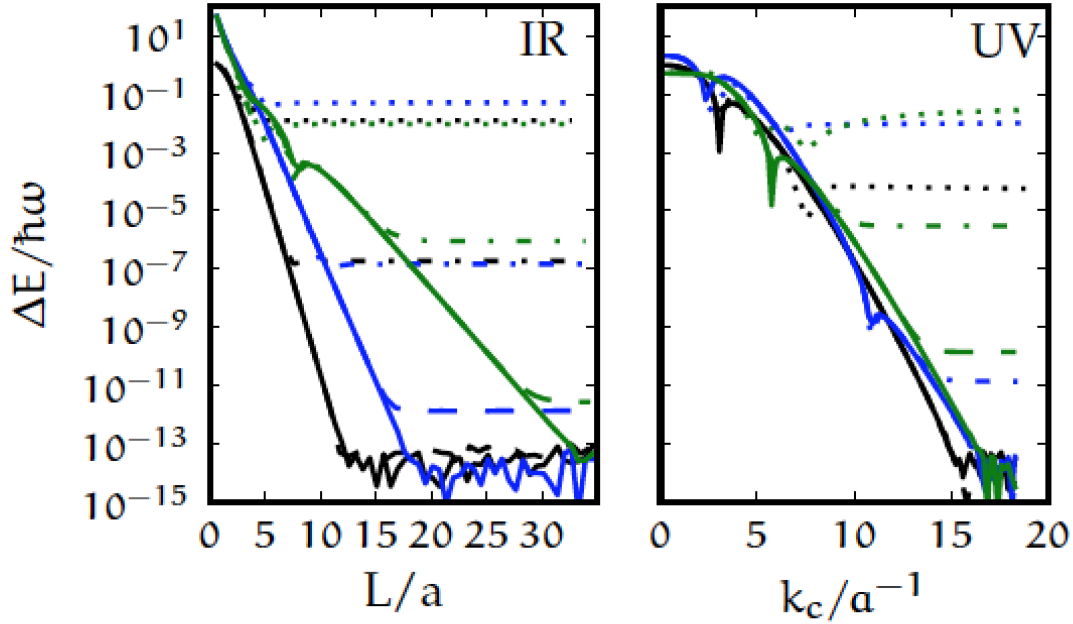
- Subtle convergence issues:
  - IR needs subtle properties of HO wavefunctions

    Furnstahl, Hagen, & Papenbrock PRC 86 (2012) 031301(R)

    More, Ekstrom, Furnstahl, Hagen, & Papenbrock arXiv:1302.3815
  - UV convergence?
    - Emperical: $E(\Lambda_{uv}) = E_\infty + A_0 \exp(-2\Lambda^2_{uv}/\lambda^2)$

      Furnstahl, Hagen, & Papenbrock PRC 86 (2012) 031301(R)
    - Where does this Gaussian behaviour come from?

Simple Convergence



IR convergence:
- Periodic Box (images)
- Lowest many-body threshold
- Band theory

UV convergence:
- Fourier analysis

Both are simple exponentials

Bulgac & Forbes arXiv:1301.7354

IR Convergence

- Band theory
- Exponential (think "tunneling") with scale set by lowest many-body dissociation threshold
  - e.g. s-wave two-body threshold

$$E(L) = E_\infty + \frac{A \exp(-2\sqrt{2MQ(L)})L/\hbar}{L^2}$$

UV Convergence

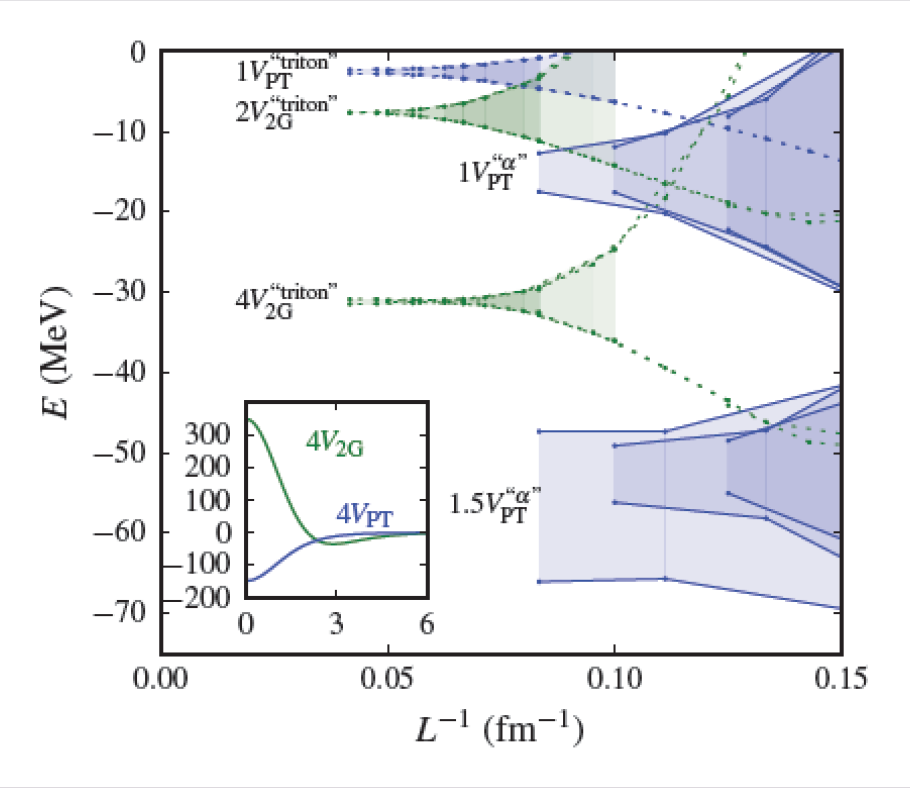
- Follows from Fourier analysis

$$E(k_c) = E_\infty + A \exp(-2k_c r_0)$$

- Exponential (not Gaussian)
  - Recall "emperical" formula for HO basis:

$$E(\Lambda_{UV}) = E_\infty + A \exp(-2\Lambda_{UV}^2/\lambda^2)$$

Exact Diagonalization ("Triton" and "Alpha")



Bulgac & Forbes arXiv:1301.7354

Use DVR for relative coords.

Directly solve 6D and 9D Schrödinger Eq.

Lanczos iterations

- No matrices $O(N \ln N)$

Several minutes on laptop

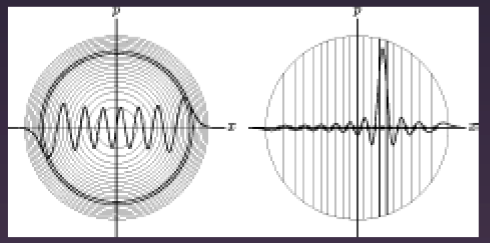
Hilbert space to $8^9 = 10^8$

- $a = 0.5$ to 1.5 fm
- $\Lambda = 300$ to 930 MeV/c

MATLAB code about 200 lines

DVR: an Efficient basis

- Quasi-local
  - $\langle F_m|V|F_n\rangle \approx \delta_{mn} V(x_n)$
  - $f_n = f(x_n)/w_n$
- Good phase-space coverage
- Easy to implement
- Straightforward convergence properties
- An efficient alternative to HO basis?



Why we use FFTW to compute derivatives?

Why it is practical:
The amazing speed of FFT

Try:
(a) FFT a three dimensional array $A(i,j,k)$
(b) Multiply two arrays $C(i,j,k) = A(i,j,k)B(i,j,k)$

Hardware	System	FFT	mesh	$\frac{T_{fft}}{T_{mul}}$	mesh	$\frac{T_{fft}}{T_{mul}}$
P-III, 750 MHz	Linux	FFTW	64^3	2.4	128^3	3.1
Origin 3800	Irix	SCSL	64^3	2.8	128^3	3.2
Origin 3800	Irix	FFTW	64^3	3.2	128^3	4.7
Itanium II	Altix	SCSL	64^3	5.1	128^3	6.3
Itanium II	Altix	FFTW	64^3	7.0	128^3	7.9

From a talk given by E. Krotscheck

Time propagation

Equation(s) to be solved numerically: $\frac{dy(t)}{dt} = f(t)$

y(t) - stands for a vector representing the values of all qpwfs at all points in space

Discretize time $t_1, t_2, t_3, \ldots$

$y_n \equiv y(t_n), f_n \equiv f(t_n), \ldots$

Adams-Bashforth-Milne method

(effectively 6th order, minimizes discretization and roundoff errors)

$$p_{n+1} = \frac{y_n + y_{n-1}}{2} + \frac{\Delta t}{48}\left(119 f_n - 99 f_{n-1} + 69 f_{n-2} - 17 f_{n-3}\right)$$

$$m_{n+1} = p_{n+1} - \frac{161}{170}\left(p_n - c_n\right)$$

$$c_{n+1} = \frac{y_n + y_{n-1}}{2} + \frac{\Delta t}{48}\left(17 m_{n+1} + 55 f_n + 3 f_{n-1} + f_{n-2}\right)$$

$$y_{n+1} = c_{n+1} + \frac{9}{170}\left(p_{n+1} - c_{n+1}\right)$$

Only 2 evaluations (shown in red) of the right-hand side per step

Characteristics of the 3D spatial lattice

$$\text{Number of quantum states} \propto 4\left(\frac{2p_c L}{2\pi\hbar}\right)^3 \approx 800{,}000$$

$$\text{for } L \approx 60\, fm$$

Lattice constant

$$p_c = \hbar k_c = \frac{\hbar\pi}{\Delta x} \approx 600\, MeV/c$$

$$\Delta x \approx 1\, fm$$

$$E_c = \frac{p_c^2}{2m} = \frac{\hbar^2\pi^2}{2m\Delta x^2} \approx 200\, MeV$$

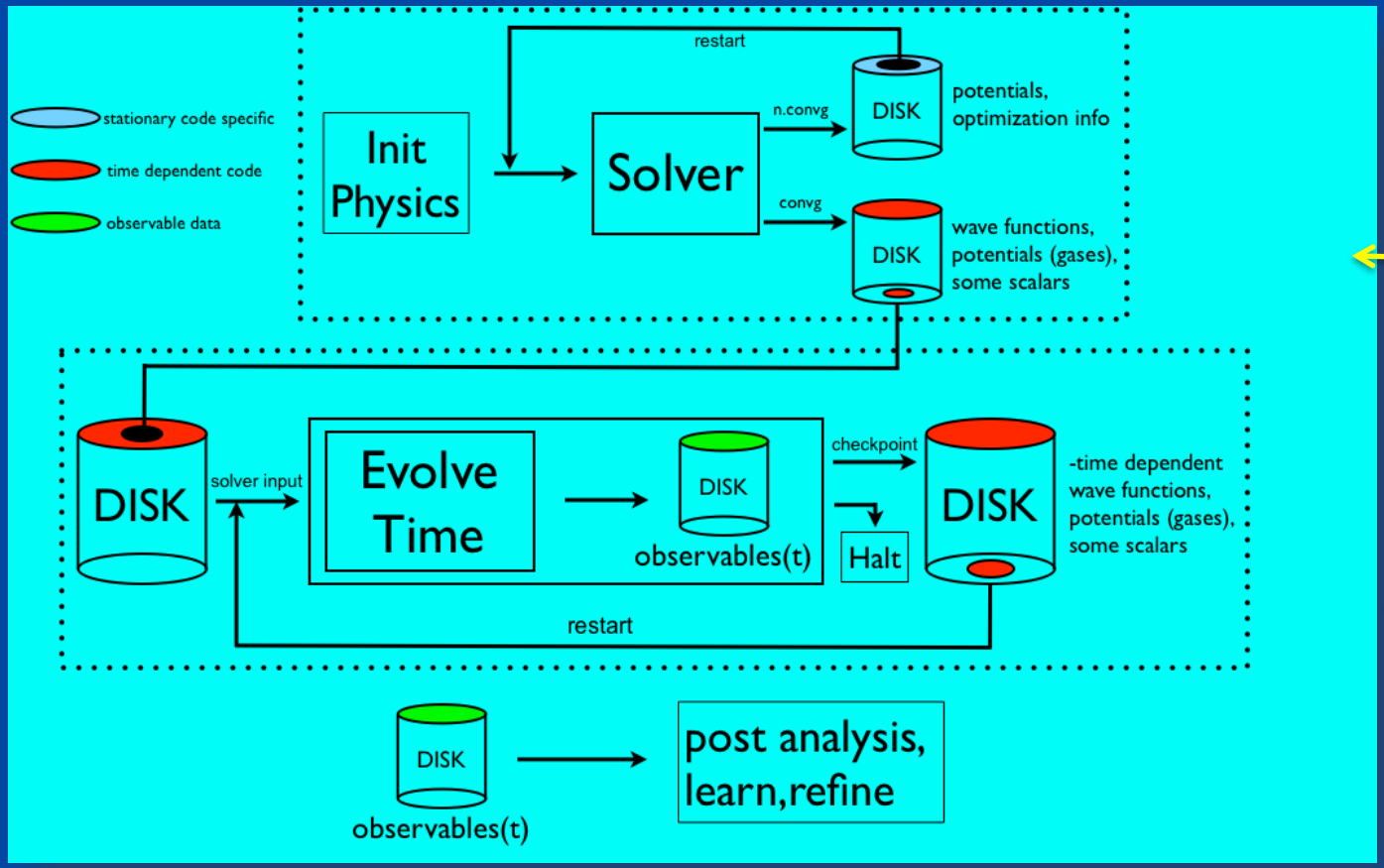
Time-step

$$\text{Error of finite difference formulas} \propto \left[f^{(5)}(x)\Delta x\right]^5 \Rightarrow \left(\frac{E_c \Delta t}{\hbar}\right)^5 = 10^{-6}\ldots 10^{-10}$$

$$\Rightarrow \Delta t \approx 0.07\cdots 0.01\, fm/c$$

Number of time steps

$$N_t \approx \frac{10^{-19}\,\text{sec.}}{\Delta t} \approx 400{,}000$$



Static

Dynamic

- ✓ full diagonalization of Hermitian matrices 409,600x409,600 on JaguarPF (for ^{238}U, 40x40x64 lattice) (required about 5.1 wall-time hours on 97% of the machine for one iteration)
- ✓ Performance on Jaguar:

Static: ins/wall-time = 1.37e19/18,393= 7.45e14
flops/wall-time=9.42e16/18,393= 5.12e12, PEs = 217,800

TD: ins/wall-time = 7.11e17/2,031= 3.50e14
flops/wall-time= 1.89e16/2,031= 9.3e12, PEs = 136,628

- ✓ excellent weak and strong scaling
- ✓ very fast I/O and checkpoint/restart capabilities
- ✓ number of coupled nonlinear time-dependent 3D PDEs for ^{238}U = 546,512

Our old code on Jaguar in 2010

24408	*48816*	*97632*	*195250*	*PEs*
8	*4*	*2*	*1*	*WF/PE*
14.52	*8.59*	*5.83*	*4.16*	*T[s]/TS*
6.99E+14	*8.72E+14*	*1.32E+15*	*2.02E+15*	*INS/TS*
2.33E+14	*2.53E+14*	*2.94E+14*	*3.74E+14*	*FP_OP/TS*
	84%	*62%*	*44%*	
0.76	*0.80*	*0.89*	*0.96*	*ins / cyc / pe*
0.33	*0.29*	*0.22*	*0.19*	*fp_op / ins*
3.00	*3.44*	*4.50*	*5.40*	*()^-1*

Figure 8. $N_x=N_y=N_z=72$ TD-SLDA Strong Scaling Example. The floating point efficiency goes down as a function of increasing processes. However, a speed-up is measured in each case.

Size of the nuclear problem (present):

Spatial lattice size $N_xN_yN_z \approx 32^3 \ldots 64^3$ (on GPUs powers of 2 are preferable)
4-component quasiparticle (complex) wave functions
Number of quasiparticle wave functions $\approx N_xN_yN_z/2$
Number of bytes per time step to represent qpwfs $\approx 2 \times 10^{12}$
total memory required (wfs, TD derivatives, potentials, etc.) $\approx 50 \times 10^{12}$
Number of time-steps $\approx O(10^6)$ (typically an order of magnitude or more less than the number of configurations needed for a PES in ATDHFB)

This is one of the largest Direct Numerical Simulation problems ever attempted and requires capability computing (HFB with constraints is a typical example of capacity computing)

From Wikipedia:

Capability vs capacity [edit]

Supercomputers generally aim for the maximum in *capability computing* rather than *capacity computing*. Capability computing is typically thought of as using the maximum computing power to solve a single large problem in the shortest amount of time. Often a capability system is able to solve a problem of a size or complexity that no other computer can, e.g. a very complex weather simulation application.[68]

Capacity computing in contrast is typically thought of as using efficient cost-effective computing power to solve a small number of somewhat large problems or a large number of small problems, e.g. many user access requests to a database or a web site.[68] Architectures that lend themselves to supporting many users for routine everyday tasks may have a lot of capacity but are not typically considered supercomputers, given that they do not solve a single very complex problem.[68]

CPU vs GPU on Titan ⟶ ≈ 15 speed-up

(likely an additional factor of 4 possible)

64^3 spatial lattice

2-component quasiparticle wave functions (no spin-orbit coupling)

- Slightly imbalanced version:

  137062 2-component wave functions

  - CPU version - 27.90 sec for 10 time steps

    4096 nodes = 16x4096 = 65,536 PEs (2 and 3 qpwfs per PEs)

  - GPU version – 1.84 sec for 10 time steps

    4096 nodes = 4096 PEs (one per node) + 4096 GPUs

- Balanced version

  131072 2-component wave functions

  - CPU version - 23.96 sec for 10 time steps

    4096 nodes = 16x4096 = 65,536 PEs (exactly 2 qpwfs per PEs)

  - GPU version – 1.65 sec for 10 time steps

    4096 nodes = 4096 PEs (one per node) + 4096 GPUs

GPUs are less prone to suffer from load imbalanced

A new method to construct the ground state which eschews big matrix diagonalization:

adiabatic switching with quantum friction

$$i\hbar\dot{\Psi}(x,t) = \left[H(x,t)+U(x,t)\right]\Psi(x,t)$$

$$E = \left\langle\Psi\middle|H\middle|\Psi\right\rangle$$

$$\dot{E} = \left\langle\Psi\middle|\dot{H}\middle|\Psi\right\rangle + \frac{2}{\hbar}\mathrm{Im}\left\langle\Psi\middle|HU\middle|\Psi\right\rangle$$

$$\text{if } U \propto -\hbar\vec{\nabla}\cdot\vec{j} = \hbar\dot{\rho} \;\Rightarrow\; \dot{E} \le \left\langle\Psi\middle|\dot{H}\middle|\Psi\right\rangle$$

$$\text{We choose } U = -\beta\frac{\hbar\vec{\nabla}\cdot\vec{j}}{\rho}$$

$$\vec{j}(\vec{r}) = \frac{\hbar}{m}\mathrm{Im}\sum_n \psi_n^*(\vec{r},t)\vec{\nabla}\psi_n(\vec{r},t)$$

Main advantage:
Replace iterative procedure which requires $O(N^3)$ operations for diagonalization with time evolution which requires only $O(N^2 \ln(N))$ operations per time step.

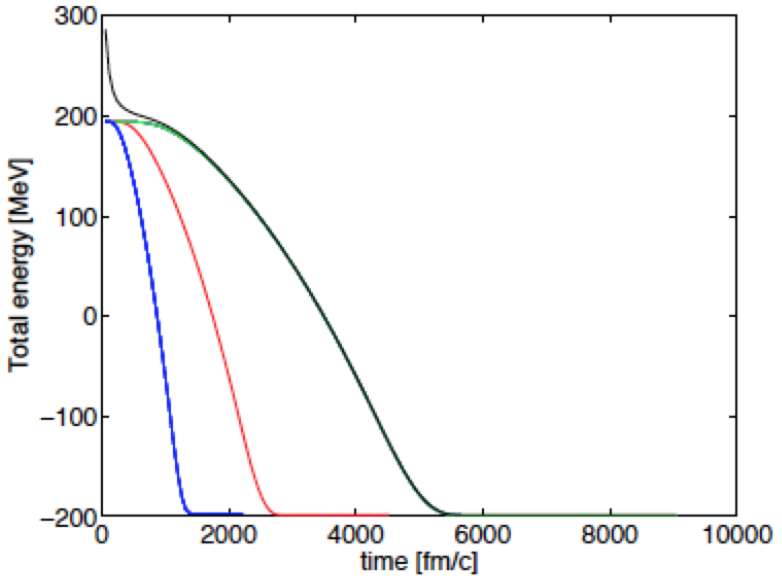


FIG. 2. (Color online) The total instantaneous energy of a system of twenty non-interacting neutrons evolving from an initial 3D harmonic oscillator potential to a final symmetrized Woods-Saxon potential. The curves correspond to quasi-adiabatic evolution with friction $(1-s_t)H_0 + s_t H_1 + U_t$ for various switching periods T (two-thirds of the simulation time) and just friction $H_1 + U_t$ for the remaining third of the simulation. That the energy is constant during this time demonstrates that the ground state has been reached. Note: there are three curves for the longest T corresponding to different simulations with $\{24^3, 32^3, 40^3\}$ lattices of 1 fm spacing: this demonstrates the infrared (IR) convergence.

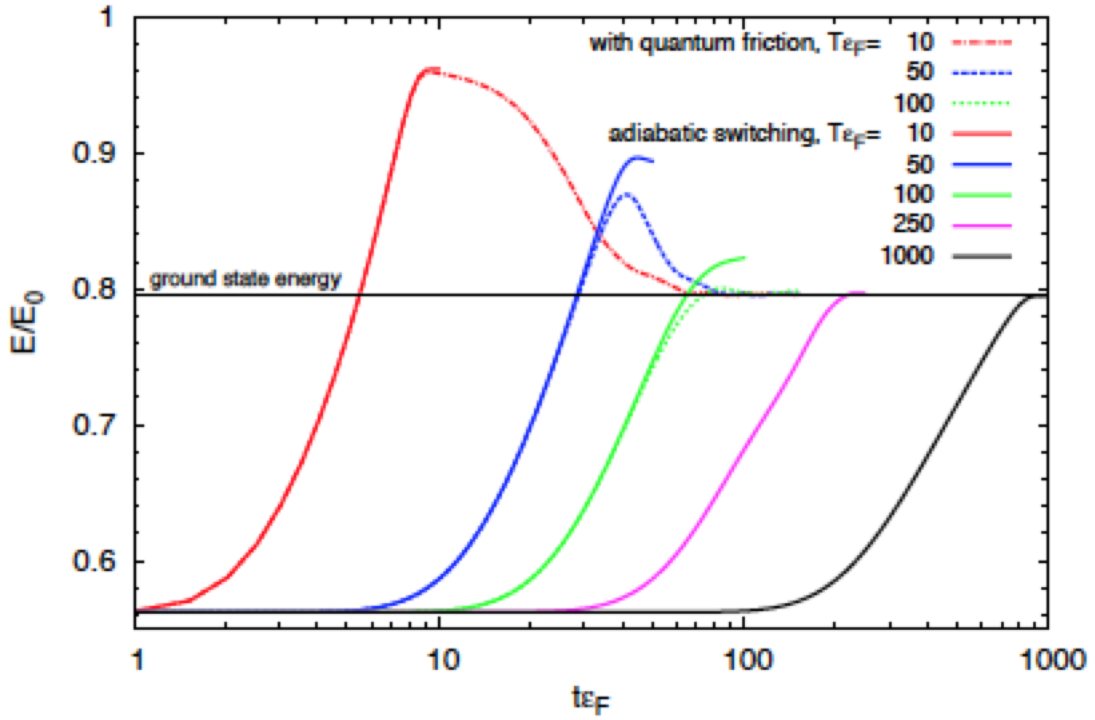


FIG. 3. (Color online) Ground state preparation of a trapped three-dimensional UFG gas from a translationally invariant (effectively two-dimensional) solution. The ground state can be prepared purely using adiabatic switching (solid lines) but the switching period can be dramatically reduced with local quantum friction. $E_0 = \frac{3}{5}N\varepsilon_F$, where N is number of particles and ε_F is Fermi energy in the center of initial solution. (See also the movie in [24].)

A nucleus under the influence of an external projectile can move outside the simulation box.

A simple coordinate transformation and change in EoM maintains the nucleus at all times in the center of the box:

$$i\hbar\dot{\Psi}(\vec{r},t) = H(\vec{r},t)\Psi(\vec{r},t)$$

$$\vec{R}(t) = \int d^3r \left|\Psi(\vec{r},t)\right|^2 \vec{r}$$

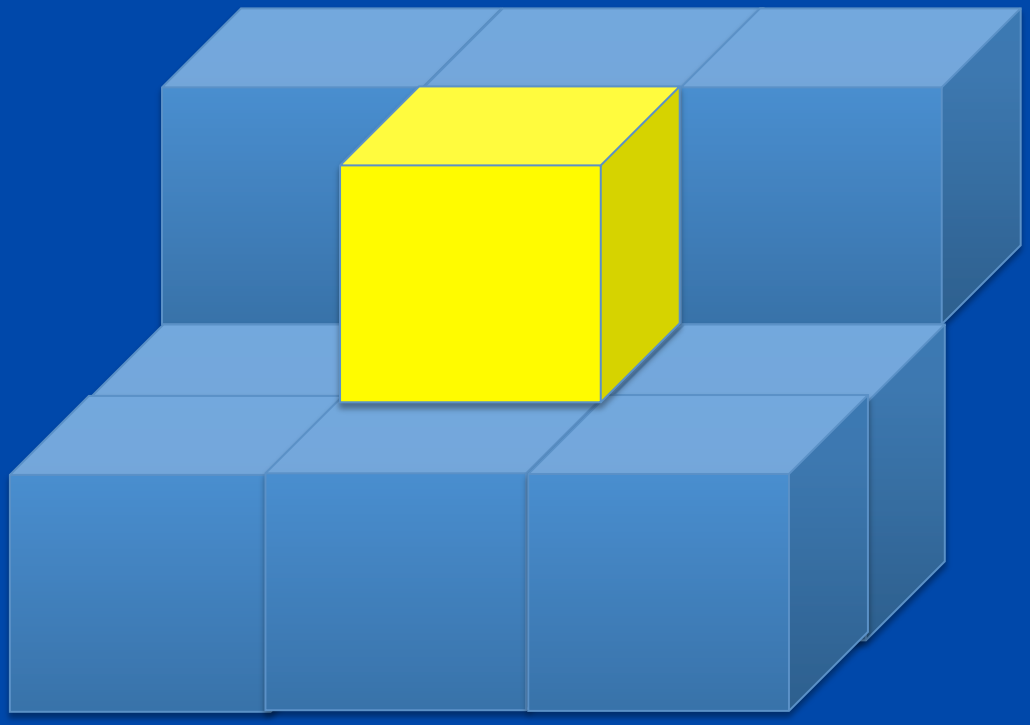
$$\Phi(\vec{r},t) = \exp\left(i\vec{R}(t)\cdot\vec{p}\right)\Psi(\vec{r},t) = \Psi(\vec{r}+\vec{R}(t),t)$$

$$\int d^3r \left|\Phi(\vec{r},t)\right|^2 \vec{r} = 0$$

$$i\hbar\dot{\Phi}(\vec{r},t) = H(\vec{r},t)\Phi(\vec{r},t) - \frac{d\vec{R}(t)}{dt}\cdot\vec{p}\ \Phi(\vec{r},t)$$

Calculation of the Coulomb potential

Problem: In a box with periodic boundary conditions one should avoid the contribution from image charges. Here the simulation box is yellow, a few images are blue.



Define a charge distribution which is non-vanishing only in the yellow box, and vanishing in blue boxes.

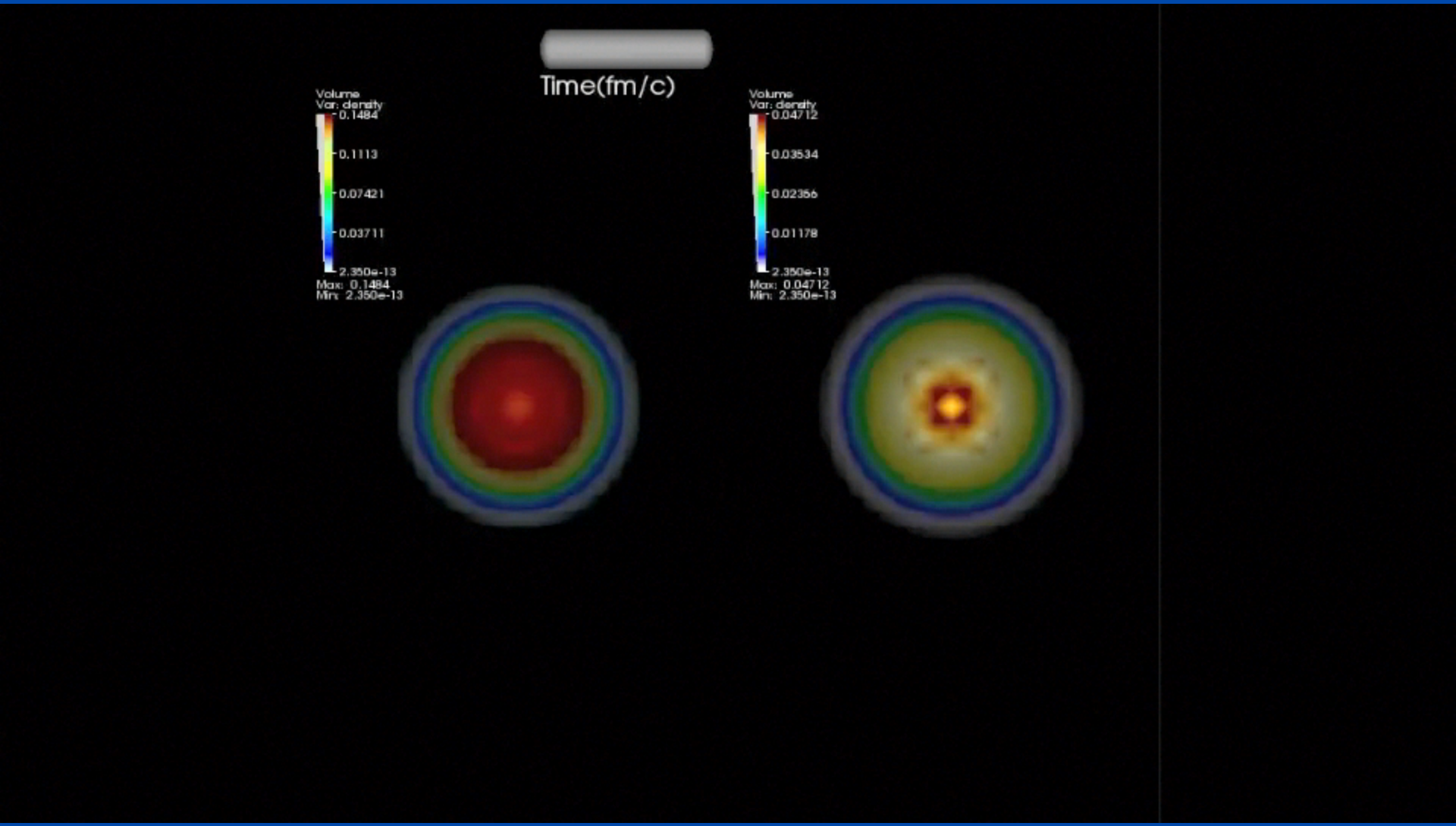
Replace the Coulomb interaction with

Plus a couple of small details to speed-up Calculations.

$$\frac{1}{|\vec{r}_1-\vec{r}_2|} \Rightarrow \begin{cases} \dfrac{1}{|\vec{r}_1-\vec{r}_2|} & \text{if } |\vec{r}_1-\vec{r}_2| \le \sqrt{3}\,L \\ 0 & \text{otherwise} \end{cases}$$

$$\int d^3r \frac{\exp(i\vec{k}\cdot\vec{r})}{r} = 4\pi \frac{1-\cos(\sqrt{3}kL)}{k^2} \times F_p^2(k)$$

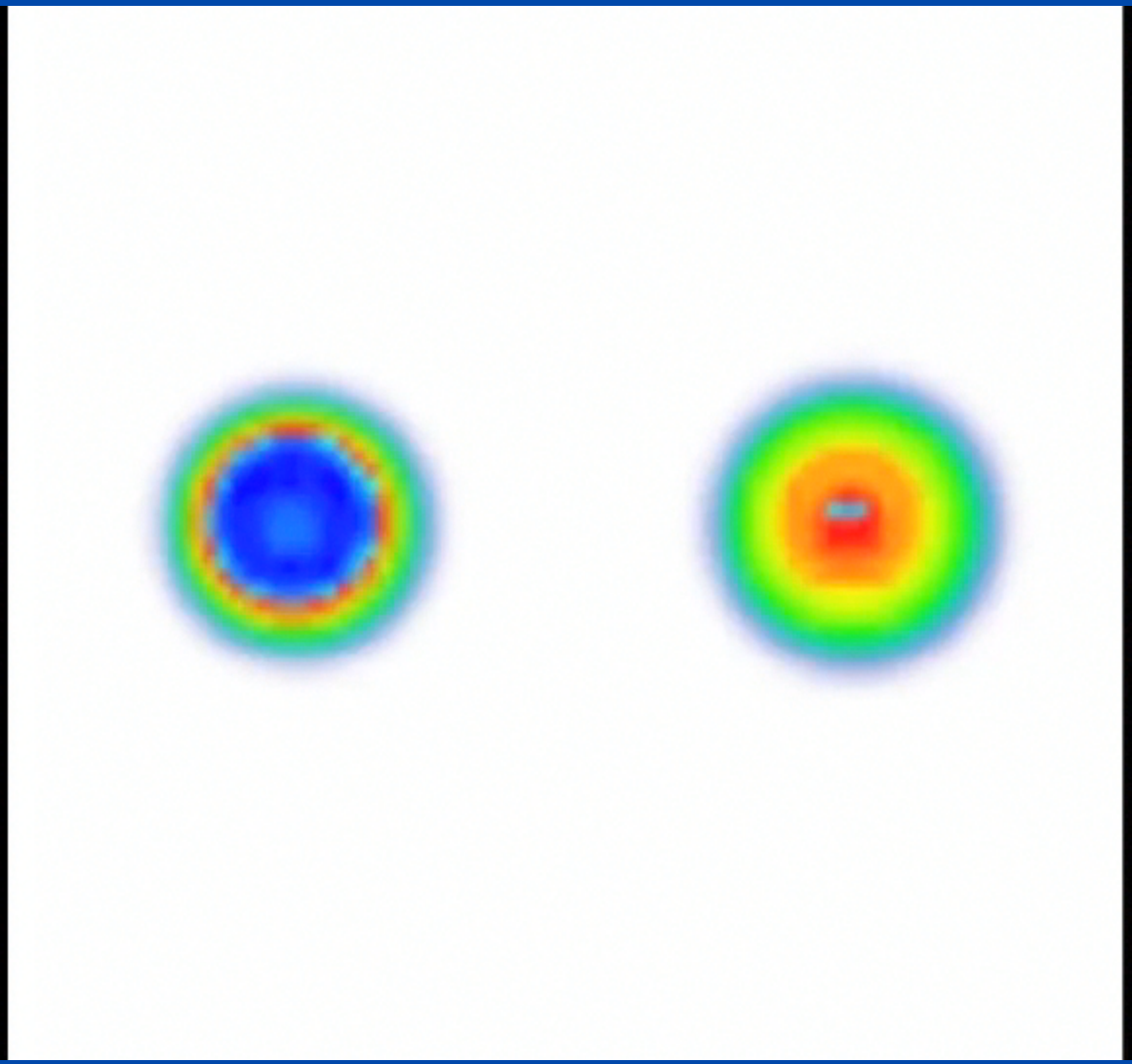
Proton charge formfactor



Coulomb excitation of GDR with a relativistic heavy-ion computed in TDSLDA
External EM field created by the uranium projectile field very strong.

I. Stetcu *et al.*

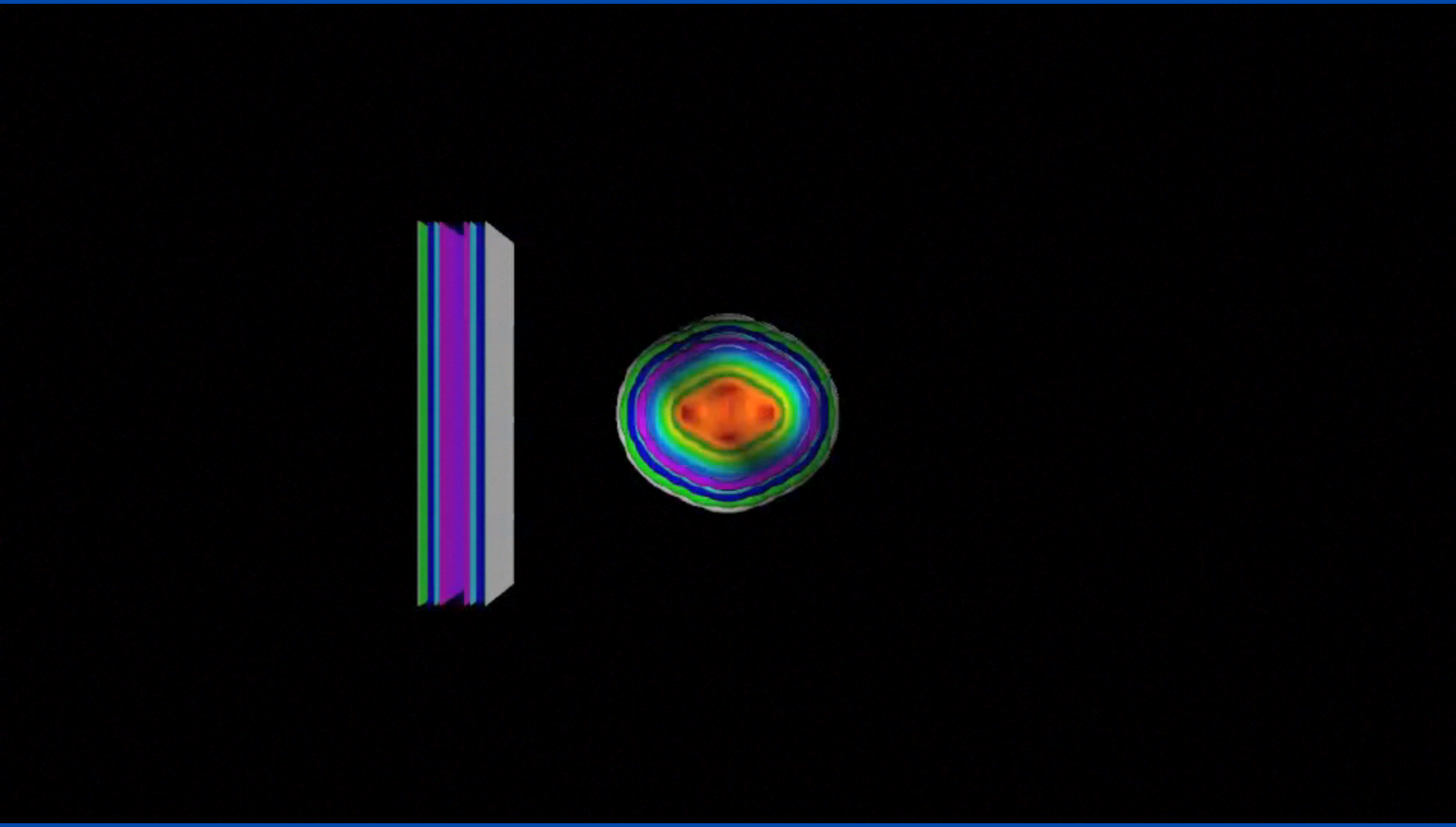
Movie



Coulomb excitation of GDR with relativistic heavy-ions computed in TDSLDA

Movie

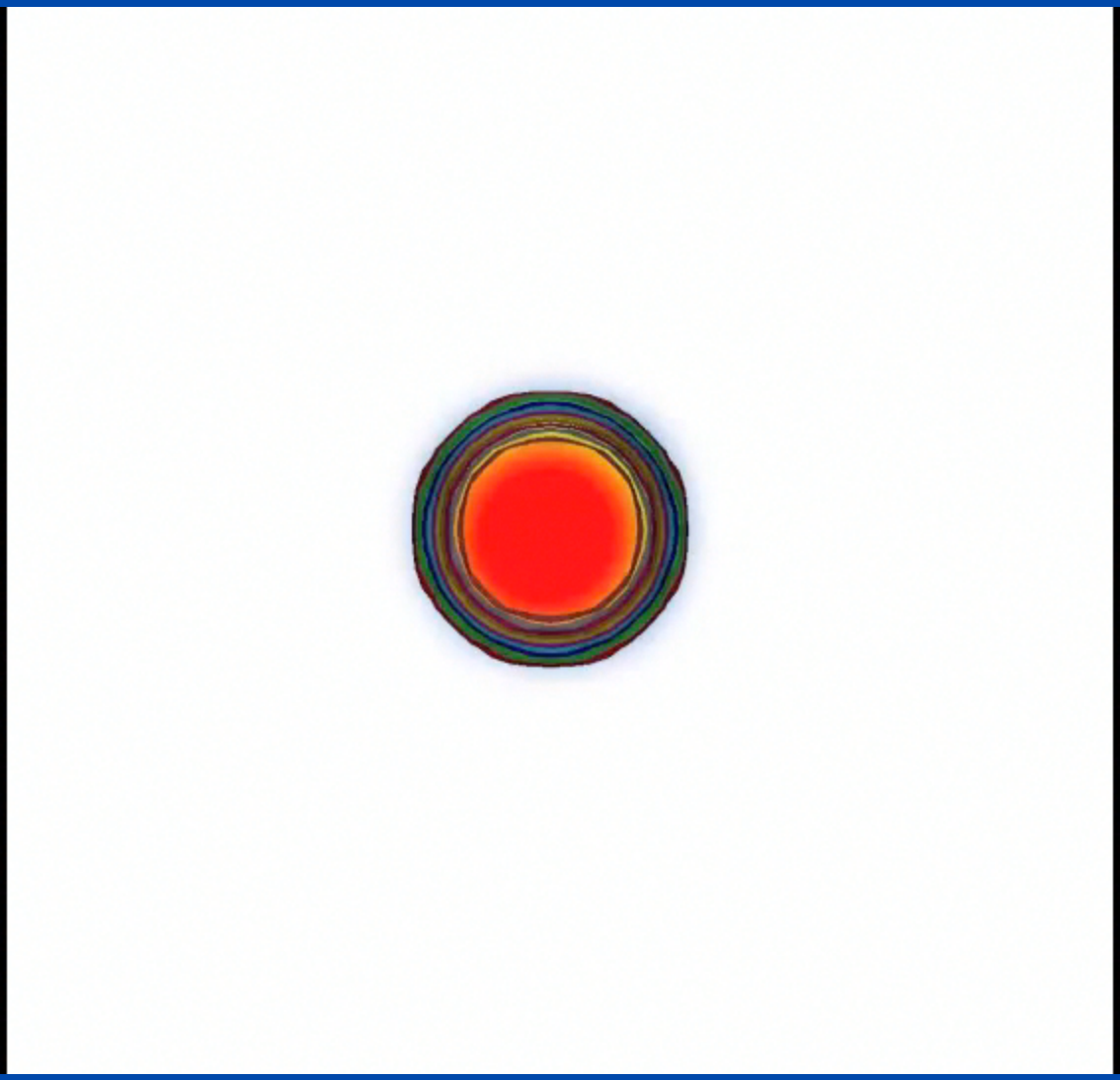
I. Stetcu *et al.*



Neutron scattering of ^{238}U computed in TDSLDA with absorbing boundary conditions

Movie

I. Stetcu *et al.*

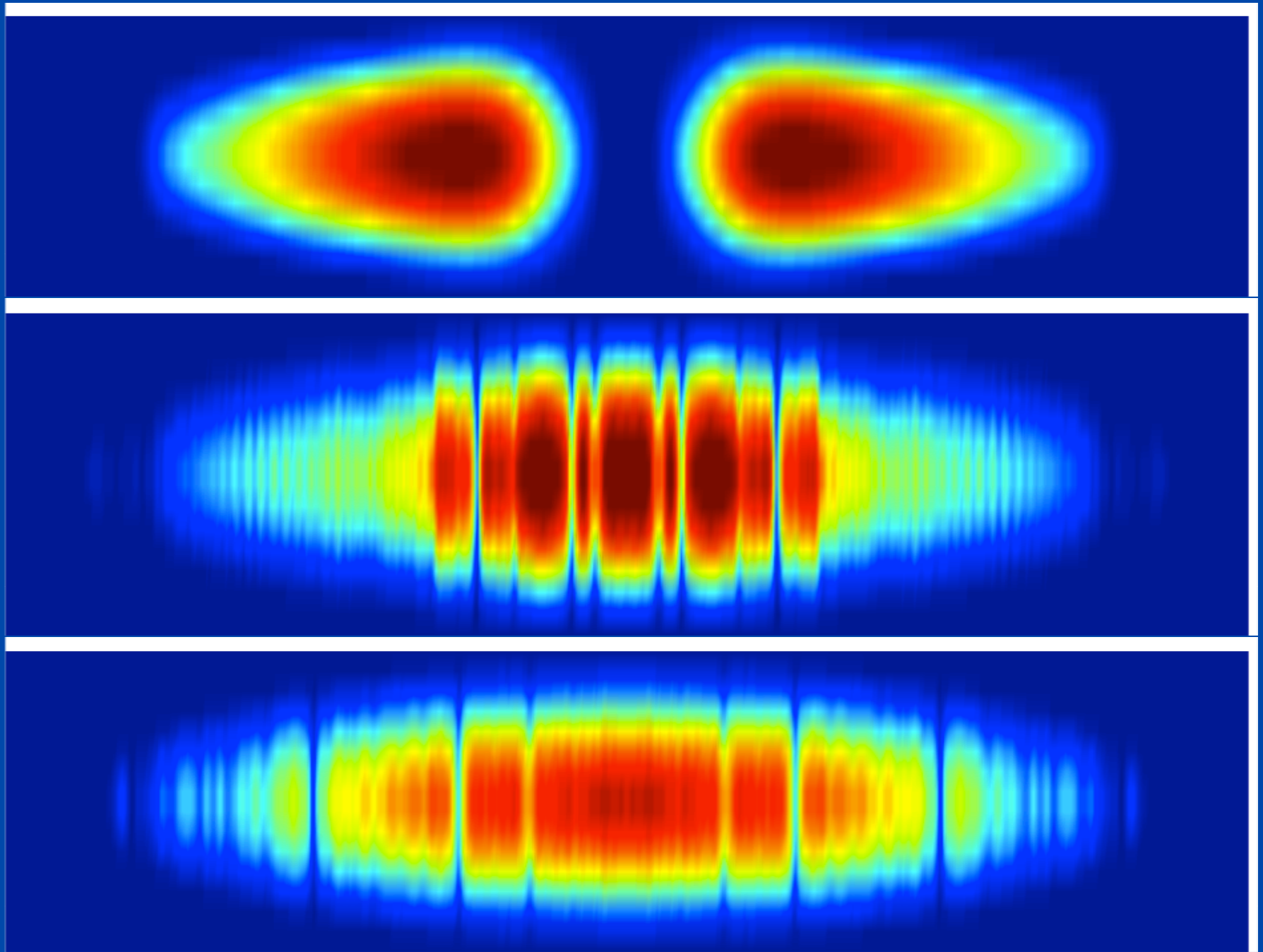


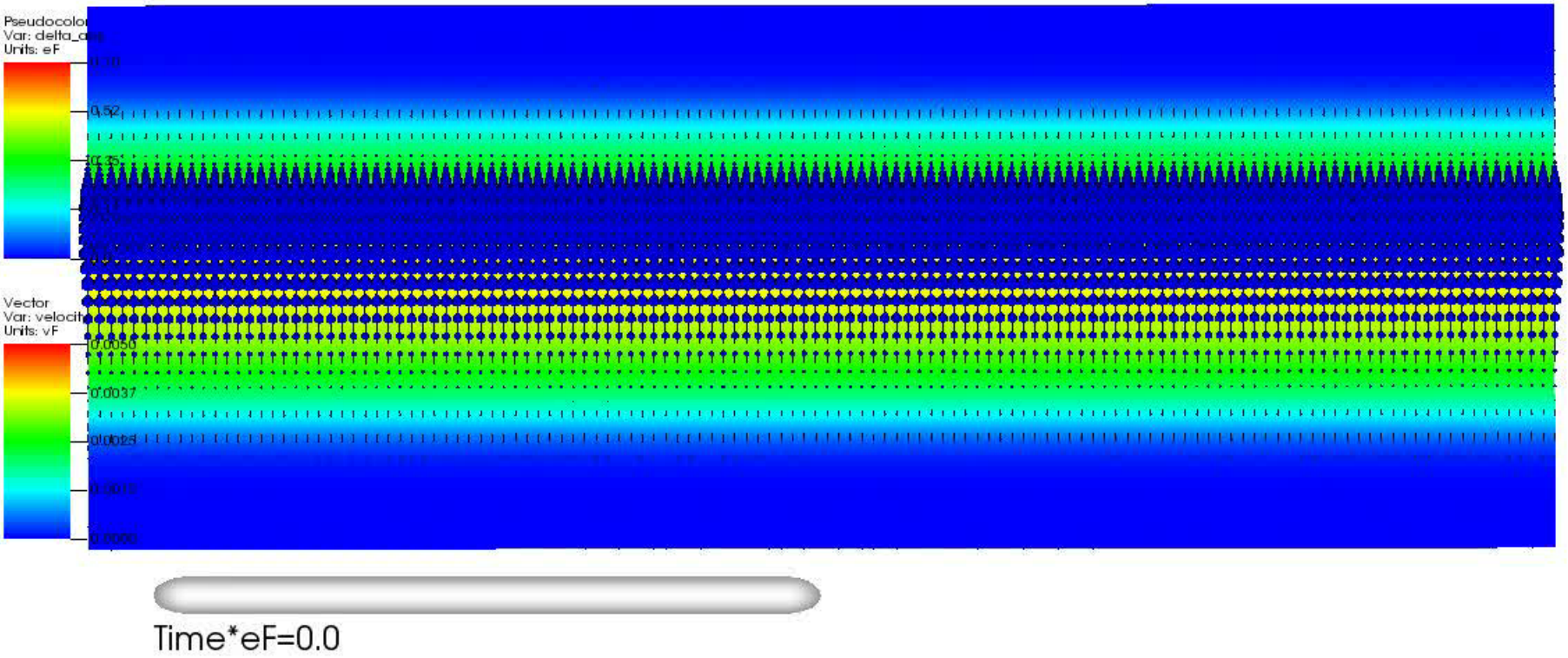
Real-time induced fission of ^{280}Cf computed in TDSLDA

I. Stetcu *et al.*

Movie

Collision of two superfluid clouds, cca 720 fermions (order parameter)





Construction of ground state (adiabatic switching with quantum friction), generation of a domain wall using an optical knife, followed by the spontaneous formation of a vortex ring. Aproximately 1270 fermions on a 48x48x128 spatial lattice, ≈ 260,000 complex PDEs, ≈ 309,000 time-steps, 2048 GPUs on Titan, 27.25 hours of wall time (initial code)

Papers we published so far on SLDA and TDSLDA
(stars indicate papers with significant nuclear physics content):

arXiv:1306.4266
* arXiv:1305.6891
* Phys. Rev. Lett. 110, 241102 (2013)
* Phys. Rev. C 87 051301(R) (2013)
* Ann. Rev. Nucl. Part. Phys. 63, 97 (2013)
* Phys. Rev. C 84, 051309(R) (2011)
Phys. Rev. Lett. 108, 150401 (2012)
Science, 332, 1288 (2011)
J. Phys. G: Nucl. Phys. 37, 064006 (2010)
Phys. Rev. Lett. 102, 085302 (2009)
Phys. Rev. Lett. 101, 215301 (2008)
* J.Phys. Conf. Ser. 125, 012064 (2008)
arXiv:1008.3933 chapter 9 in Lect. Notes Phys. vol. 836
Phys. Rev. A 76, 040502(R) (2007)
* Int. J. Mod. Phys. E 13, 147 (2004)
Phys. Rev. Lett. 91, 190404 (2003)
* Phys. Rev. Lett. 90, 222501 (2003)
* Phys. Rev. Lett. 90, 161101 (2003)
* Phys. Rev. C 65,051305(R) (2002)
* Phys. Rev. Lett. 88, 042504 (2002)
Plus a few other chapters in various books.