Density Potential Functional Theory in position and momentum space and its implementation using PyTorch

Ding Ruiqi

Supervisor: Prof. Berge Englert

March 2020

Table of Contents

Summary

- **Topic:** Density Potential Functional Theory
- **Application:** Spin polarized Fermi gas with magnetic dipole-dipole interaction
- **Contributions:**
  - Derived various energy functionals
  - Developed my own DPFT package: PyDPFT
  - Simulated the dipole system
  - New idea to filter Thomas Ferimi density
- **Highlights:** 1D to 3D, position and momentum space, multi-GPU acceleration
- **Results:** Agrees well with physical expectation

Density functional theory

- **Wave function to density:** $\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2, ...\boldsymbol{r}_N)$ to $n(\boldsymbol{r})$ means $\boldsymbol{R}^{3N} \to \boldsymbol{C}$ to $\boldsymbol{R}^3 \to R$
- **Orbital free:** faster but less accurate
  - **Thomas-Fermi density:** Integrating the phase space density $\eta(\mu - H(\boldsymbol{p}, \boldsymbol{r}))$ (Wigner function).
    $n_{\mathrm{TF}}(\boldsymbol{r}) = 2 \int d\boldsymbol{p}\, \eta(\mu - H(\boldsymbol{p}, \boldsymbol{r})) \frac{1}{(2\pi\hbar)^3} = \frac{1}{3\pi^2} k_F^3(\boldsymbol{r})$
    where η: Heaviside step function. μ: chemical potential.
  - **Kinetic energy:** $T_{\mathrm{TF}}[n] = \int d\boldsymbol{r} d\boldsymbol{p}\, \frac{\boldsymbol{p}^2}{2m} = C_{\mathrm{TF}} \int d\boldsymbol{r}\, n^{5/3}$
  - **Corrections:** Gradient expansions. E.g. TF-Weizsacker functional: $T_{\mathrm{TFW}}[n] = T_{\mathrm{TF}}[n] + \frac{1}{8} \int d\boldsymbol{r}\, \frac{|\nabla n|^2}{n}$

Density functional theory

- **Kohn Sham:** slower but more accurate
  - **Kohn Sham orbitals:**

    $\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2, ...\boldsymbol{r}_N) \quad \text{to} \quad \{\psi_1(\boldsymbol{r}), \psi_2(\boldsymbol{r}), ...\psi_N(\boldsymbol{r})\}$

    $\boldsymbol{R}^{3N} \rightarrow \boldsymbol{C} \quad \text{to} \quad \boldsymbol{R}^3 \rightarrow N \times \boldsymbol{C}$
  - **Single particle Schrodinger equation:**

    $\left[\frac{(i\nabla)^2}{2} + V_{\text{ext}}[n] + V_{\text{int}}[n]\right] \psi_i(\boldsymbol{r}) = \epsilon_i \psi_i(\boldsymbol{r})$
  - **Density:** $n(\boldsymbol{r}) = \sum_i^N |\psi_i(\boldsymbol{r})|^2$
  - **Kinetic energy:** $T_{\text{kin}}[\psi_i(\boldsymbol{r})]$

Density potential functional theory (DPFT)

- By Julian Schwinger and Berge Englert
- **Energy functional:**

  $$E[n,\mu] = E_{\text{kin}} + \int d\boldsymbol{r}\; V_{\text{ext}} n + E_{\text{int}}[n] + \mu(N - \int d\boldsymbol{r}\; n)$$

- **Conjugate variable:** $V := \mu - \frac{\delta E_{\text{kin}}}{\delta n}$
- **Legendre transform:**

  $$E_{\text{kin}}^{\text{LGD}} := E_{\text{kin}} - \int d\boldsymbol{r}\; n\; \frac{\delta E_{\text{kin}}}{\delta n} = E_{\text{kin}} + \int d\boldsymbol{r}\; n \cdot (V - \mu)$$

- **Rewrite Energy functional:**

  $$E = E_{\text{kin}}^{\text{LGD}}[V - \mu] + \int d\boldsymbol{r}\; (V_{\text{ext}} - V)n + E_{\text{int}}[n] + \mu N$$

- **Self consistent equations:**

  $$\frac{\delta E[n,\mu,V]}{\delta n} = 0 = V_{\text{ext}} - V + \frac{\delta E_{\text{int}}[n]}{\delta n}$$
  $$\frac{\delta E[n,\mu,V]}{\delta \mu} = 0 = \frac{\delta E_{\text{kin}}^{\text{LGD}}[V-\mu]}{\delta \mu} + N$$
  $$\frac{\delta E[n,\mu,V]}{\delta V} = 0 = \frac{\delta E_{\text{kin}}^{\text{LGD}}[V-\mu]}{\delta V} - n$$

DPFT: Momentum space

- **Energy functional:**

  $$E[n,\mu] = \int d\boldsymbol{p}\, T_{\text{kin}} n + E_{\text{ext}} + E_{\text{int}}[n] + \mu(N - \int d\boldsymbol{p}\, n)$$

- **Conjugate variable:** $T := \mu - \frac{\delta E_{\text{ext}}}{\delta n}$

- **Legendre transform:**

  $$E_{\text{ext}}^{\text{LGD}} := E_{\text{ext}} - \int d\boldsymbol{p}\, n\, \frac{\delta E_{\text{ext}}}{\delta n} = E_{\text{ext}} + \int d\boldsymbol{p}\, n \cdot (T - \mu)$$

- **Rewrite Energy functional:**

  $$E = \int d\boldsymbol{p}\, (T_{\text{kin}} - T) n + E_{\text{ext}}^{\text{LGD}} + E_{\text{int}}[n] + \mu N$$

- **Self consistent equations:**

  $$\frac{\delta E[n,\mu,T]}{\delta n} = 0 = T_{\text{kin}} - T + \frac{\delta E_{\text{int}}[n]}{\delta n}$$
  $$\frac{\delta E[n,\mu,T]}{\delta \mu} = 0 = \frac{\delta E_{\text{ext}}^{\text{LGD}}[T-\mu]}{\delta \mu} + N$$
  $$\frac{\delta E[n,\mu,T]}{\delta T} = 0 = \frac{\delta E_{\text{ext}}^{\text{LGD}}[T-\mu]}{\delta T} - n$$

Wigner function and densities

- **Wigner function :** $\nu(\boldsymbol{r}, \boldsymbol{p})$
- **Spatial one particle density matrix:** $^{(1)}$ for one particle. $_1$ for position
  $n^{(1)}(\boldsymbol{r}_1; \boldsymbol{r}_2) = \int \frac{d\boldsymbol{p}}{(2\pi\hbar)^3} \, \nu(\frac{\boldsymbol{r}_1+\boldsymbol{r}_2}{2}, \boldsymbol{p}) e^{i\frac{\boldsymbol{p}}{\hbar}\cdot(\boldsymbol{r}_1-\boldsymbol{r}_2)}$
- **Spatial one particle density:**
  $n(\boldsymbol{r}) = n^{(1)}(\boldsymbol{r}; \boldsymbol{r}) = \int \frac{d\boldsymbol{p}}{(2\pi\hbar)^3} \, \nu(\boldsymbol{r}, \boldsymbol{p})$
- **Momentum space:** $\rho(\boldsymbol{p}) = \int \frac{d\boldsymbol{r}}{(2\pi\hbar)^3} \, \nu(\boldsymbol{r}, \boldsymbol{p})$

Approximations

- **Dirac's approximation for two particle density matrix:** $_{1a}$ for position 1 of particle (a)

  $n^{(2)}(\boldsymbol{r}_{1a}, \boldsymbol{r}_{1b}; \boldsymbol{r}_{2a}, \boldsymbol{r}_{2b}) =$

  $n^{(1)}(\boldsymbol{r}_{1a}; \boldsymbol{r}_{2a})\; n^{(1)}(\boldsymbol{r}_{1b}; \boldsymbol{r}_{2b}) - n^{(1)}(\boldsymbol{r}_{1a}; \boldsymbol{r}_{1b})\; n^{(1)}(\boldsymbol{r}_{2b}; \boldsymbol{r}_{2a})$

- **Physics:** Effect of exchanging positions of the two particles
- **Our system:** Particle can only be at one place $\boldsymbol{r}_{1a} = \boldsymbol{r}_{2a}$:

  $n^{(2)}(\boldsymbol{r}_a, \boldsymbol{r}_b; \boldsymbol{r}_a, \boldsymbol{r}_b) = \underbrace{n(\boldsymbol{r}_a)\; n(\boldsymbol{r}_b)}_{\text{direct term}} - \underbrace{n^{(1)}(\boldsymbol{r}_a; \boldsymbol{r}_b)\; n^{(1)}(\boldsymbol{r}_b; \boldsymbol{r}_a)}_{\text{exchange term}}$

- **Thomas Fermi approximation:**

  $\underbrace{\nu(\boldsymbol{r}, \boldsymbol{p}) = \eta(\hbar[6\pi^2 n(\boldsymbol{r})]^{1/3} - p)}_{\text{position space}} \quad \underbrace{\nu(\boldsymbol{r}, \boldsymbol{p}) = \eta(\hbar[6\pi^2 \rho(\boldsymbol{p})]^{1/3} - r)}_{\text{momentum space}}$

Spin polarized Fermi gas with magnetic dipole-dipole interaction

- **Kinetic energy:** $E_{\text{kin}} = \int d\boldsymbol{p}\ \frac{\boldsymbol{p}^2}{2M}\rho(\boldsymbol{p})$

  $\frac{\delta E_{\text{kin}}}{\delta \rho} = \frac{\boldsymbol{p}^2}{2M}$
- **External potential:** Harmonic trap

$$E_{\text{ext}} = \int d\boldsymbol{r}\ \frac{1}{2}M\omega^2 r^2 n(\boldsymbol{r}) \tag{1}$$

$$= \ldots = \int \frac{d\boldsymbol{p}}{20\pi^2}\ M(\hbar\omega)^2[6\pi^2\rho(\boldsymbol{p})]^{5/3} \tag{2}$$

$\frac{\delta E_{\text{ext}}}{\delta \rho} = \frac{1}{2}M(\hbar\omega)^2[6\pi^2\rho(\boldsymbol{p})]^{2/3}$

- **Density:** $T := \mu - \frac{\delta E_{\text{ext}}}{\delta \rho} \Rightarrow \rho(\boldsymbol{p}) = \frac{1}{6\pi^2}\left[\frac{2(\mu - T)}{M(\hbar\omega)^2}\right]^{3/2}$

Spin polarized Fermi gas with magnetic dipole-dipole interaction

- **Dipole-dipole interaction:**

$$E_{\rm dd} = \frac{1}{2}\int d\boldsymbol{r}_a d\boldsymbol{r}_b\, U_{\rm dd}(\boldsymbol{r}_a - \boldsymbol{r}_b) n^{(2)}(\boldsymbol{r}_a, \boldsymbol{r}_b; \boldsymbol{r}_a, \boldsymbol{r}_b) \qquad (3)$$

where $U_{\rm dd}(\boldsymbol{r}) = \frac{\mu_0}{4\pi}\left[\frac{\boldsymbol{\mu}^2}{r^3} - 3\frac{(\boldsymbol{\mu}\cdot\boldsymbol{r})^2}{r^5} - \frac{8\pi}{3}\boldsymbol{\mu}^2\delta(\boldsymbol{r})\right]$

- **Results:** after long derivation

$V_{\rm dd}^{\rm mom} := \frac{\delta E_{\rm dd}}{\delta\rho(\boldsymbol{p}_a)} =$
$\frac{\mu_0}{2}\int \frac{d\boldsymbol{p}_b}{(2\pi\hbar)^3}\left[\frac{(\boldsymbol{\mu}\cdot\boldsymbol{k})^2}{\boldsymbol{k}^2} - \frac{1}{3}\boldsymbol{\mu}^2\right]_{\boldsymbol{k}=\frac{\boldsymbol{p}_a-\boldsymbol{p}_b}{\hbar}} \eta[\rho(\boldsymbol{p}_b) - \rho(\boldsymbol{p}_a)]$

$V_{\rm dd}^{\rm pos} := \frac{\delta E_{\rm dd}}{\delta\rho(\boldsymbol{r}_a)} = \frac{\mu_0}{4\pi}\int d\boldsymbol{r}_b\, \frac{1}{\boldsymbol{r}^3}\left[-3\frac{(\boldsymbol{\mu}\cdot\boldsymbol{r})^2}{\boldsymbol{r}^2} + \boldsymbol{\mu}^2\right]_{\boldsymbol{r}=\boldsymbol{r}_a-\boldsymbol{r}_b}$

Analytical attempt

- **The integral equation:** $\frac{\delta E}{\delta \rho} = 0$
- **Isotropic assumption:** $\eta[\rho(\boldsymbol{p} - \hbar\boldsymbol{k}) - \rho(\boldsymbol{p})] = 1$ and integrate over a ball
- **Integration trick:** $\boldsymbol{p} = \Big(0, 0, p\Big) \quad \hat{\boldsymbol{\mu}} = \Big(\sin\alpha, 0, \cos\alpha\Big) \quad \boldsymbol{q} = q\Big(\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta\Big)$
- **Anisotropic result:** $\rho(\boldsymbol{p}) =$

$$\frac{1}{6\pi^2}\left(\frac{1}{2}M(\hbar\omega)^2\right)^{-3/2}\left[\underbrace{\mu}_{V_{\text{chem}}}\underbrace{-\frac{\boldsymbol{p}^2}{2M}}_{V_{\text{kin}}}\underbrace{+\frac{\mu_0\boldsymbol{\mu}^2p^3}{144\,\hbar^3\pi^2}(3\cos(2\alpha)+1)}_{V_{\text{dd}}^{\text{mom,iso}}}\right]^{3/2}$$

My own DPFT code development: PyDPFT

- All Members of my group use Dr. Martin Trappe's code
- His code runs on **CPU, which is slow**
- I want to develop my code with **GPU parallelization**
- I used PyTorch (state of the art Machine Learning library) to achieve this
- Orbital free. Published as a **python package**. **GitHub:** `github.com/tesla-cat/PyDPFT`

Figure: My Python package: orbital free PyDPFT

PyDPFT: Compare with Kohn Sham

- **The interaction:** Exchange energy under Dirac approximation and the Hartree (Coulomb) energy

  $E_H[n(\boldsymbol{r})] = \frac{1}{2}\int d\boldsymbol{r}d\boldsymbol{r}'\ \frac{n(\boldsymbol{r})n(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|+\epsilon}$

  $E_{\mathrm{x}}^{\mathrm{LDA}}[n] = -\frac{3}{4}(\frac{3}{\pi})^{1/3}\int d\boldsymbol{r}\ n^{4/3}$

- **Kohn Sham density:** Express the Laplacian as matrix to solve single particle shrodinger equation:

  $\nabla^2\psi = \frac{d^2\psi}{dx^2} = \frac{\psi_{i+1}-2\psi_i+\psi_{i-1}}{h^2} \equiv M\psi$

- **PyDPFT Thomas Fermi density:**

  $E_{\mathrm{kin}}^{\mathrm{LGD}}[V-\mu] = \int \frac{d\boldsymbol{r}d\boldsymbol{p}}{(2\pi\hbar)^3}[\frac{\boldsymbol{p}^2}{2m}+V(\boldsymbol{r})-\mu]\ \eta(\mu-\frac{\boldsymbol{p}^2}{2m}-V)$

  $n = \frac{\delta E_{\mathrm{kin}}^{\mathrm{LGD}}[V-\mu]}{\delta V} = \frac{1}{6\pi^2\hbar^3}P^3\big|_{P=\sqrt{2m(\mu-V)}}$

PyDPFT: Compare with Kohn Sham

- **Result:** agrees well with each other

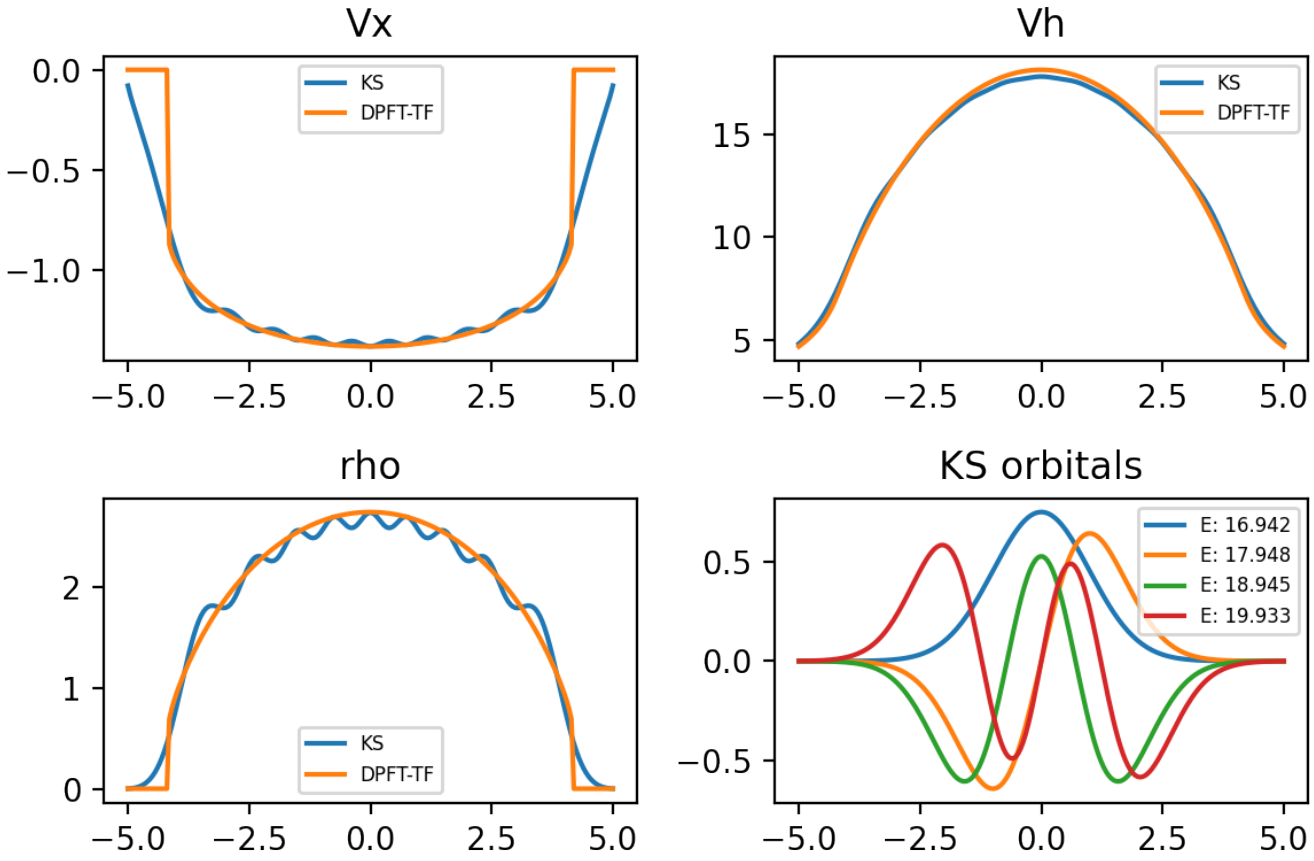
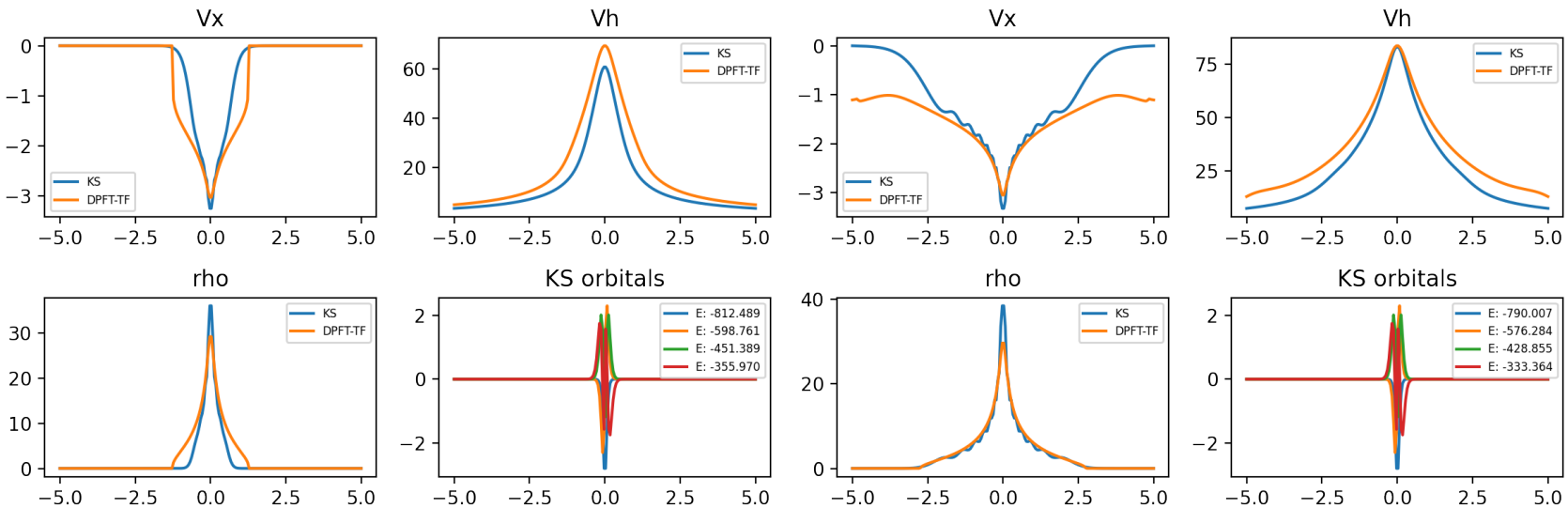


Figure: Kohn Sham vs DPFT-TF in 1D: $V_{\text{ext}} = r^2$, $N_{\text{particle}} = 18$

PyDPFT: Compare with Kohn Sham

- **Result:** Thomas Fermi works better at large N_{particle}



(a) $N_{\text{particle}} = 18$ (b) $N_{\text{particle}} = 36$

Figure: Kohn Sham vs DPFT-TF in 1D: $V_{\text{ext}} = -\frac{Z}{r+\epsilon}$

PyDPFT: 2D dipole dipole

- **PyDPFT:** very simple to use

2D Dipole-dipole interaction in momentum (p) space

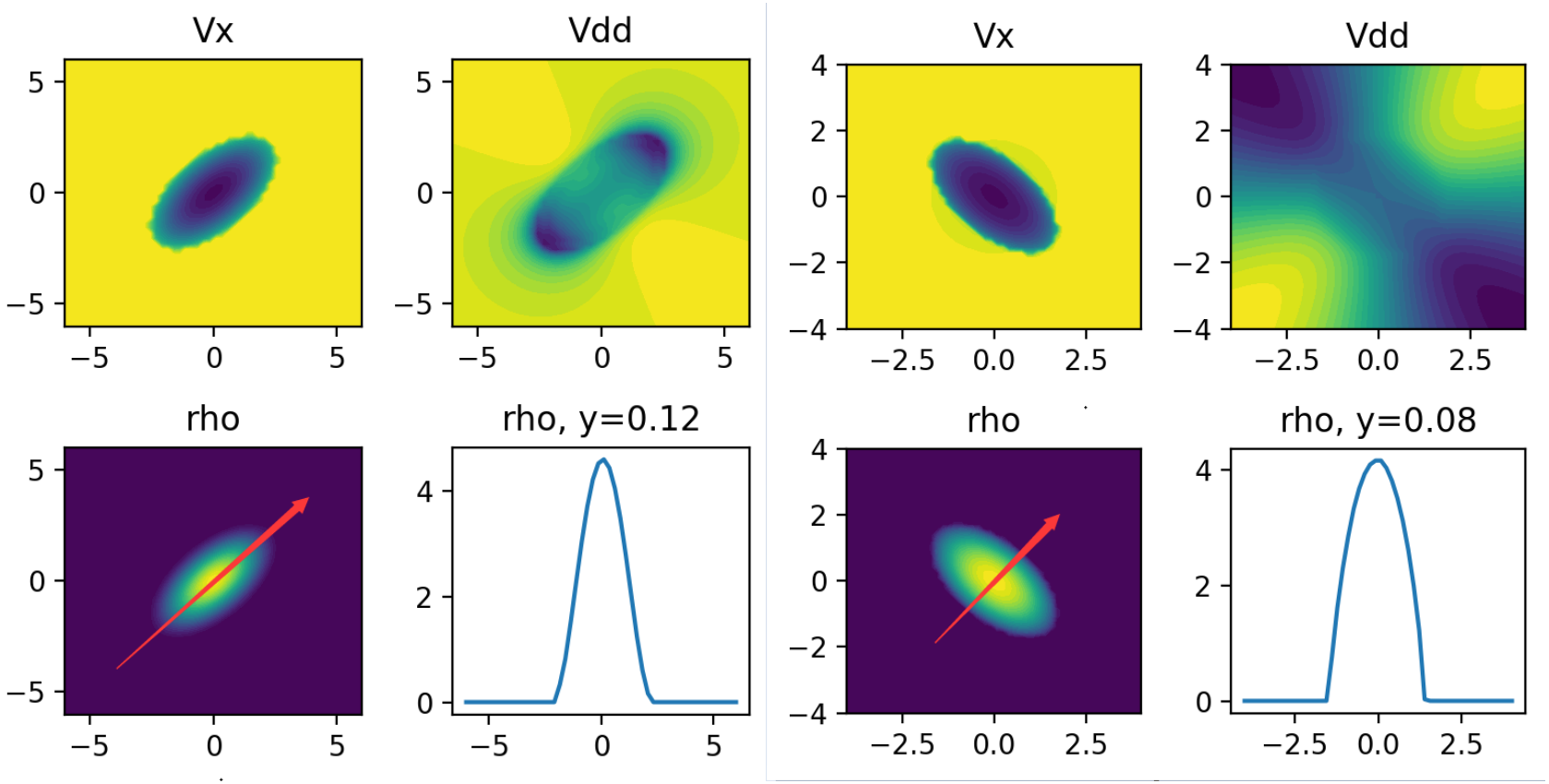
```
config = {
    'space':{'x':[-5,5,50],'y':[-5,5,50]},
    'loop':{'Imax':1000,'precision':1e-6,'mix':0.05},
    'const':{'epsilon':1e-2,'mu':[0.7, 0.7]},
    'rho':{'N':32},
    'Vint':{'name':'Dipole-p','coef':.1},
}

dpft = PyDPFT(config)
Vext = dpft.xx**2 + dpft.yy**2
Vx,Vint,rho,N = dpft(Vext)
plot(dpft,Vx,Vint,rho)
```

```
PyDPFT: Written by Ding Ruiqi from NUS for his bachelor thesis
PyDPFT: Detected dim = 2
PyDPFT: Using 1 GPUs !
PyDPFT: Starting the self consistent loop
PyDPFT: Converged after 219 iterations in 3.8995468616485596 seconds!
```

Figure: PyDPFT: very simple to use, 2D

PyDPFT: 2D dipole dipole



(a) position space

(b) momentum space

Figure: Position space density is stretched along $\boldsymbol{\mu} = (0.7,\ 0.7)$, momentum space density is squeezed

PyDPFT: 2D dipole dipole explanation

- **Position space:** Density stretched along $\boldsymbol{\mu}$
  - Magnets tend to align head to tail with one another.
- **Momentum space:** Density squeezed along $\boldsymbol{\mu}$
  - **Wave view:** $k = \frac{2\pi}{\lambda}$, thus large λ profile along some direction in $\boldsymbol{r}$ space corresponds to small k profile in $\boldsymbol{p}$ space
  - **Particle view:** easier to move perpendicular to $\boldsymbol{\mu}$ than along $\boldsymbol{\mu}$ due to attraction and repulsion from neighbouring atoms
  - **Math view:** $V_{\mathrm{dd}}^{\mathrm{pos}}$ is basically the negative of $V_{\mathrm{dd}}^{\mathrm{mom}}$ with an extra $\frac{1}{\boldsymbol{r}^3}$

$$V_{\mathrm{dd}}^{\mathrm{mom}} = \frac{\mu_0}{2} \int \frac{d\boldsymbol{p}_b}{(2\pi\hbar)^3} \left[\frac{(\boldsymbol{\mu}\cdot\boldsymbol{k})^2}{\boldsymbol{k}^2} - \frac{1}{3}\boldsymbol{\mu}^2 \right]_{\boldsymbol{k}=\frac{\boldsymbol{p}_a-\boldsymbol{p}_b}{\hbar}} \eta[\rho(\boldsymbol{p}_b) - \rho(\boldsymbol{p}_a)]$$

$$V_{\mathrm{dd}}^{\mathrm{pos}} = \frac{\mu_0}{4\pi} \int d\boldsymbol{r}_b \, \frac{1}{\boldsymbol{r}^3} \left[-3\frac{(\boldsymbol{\mu}\cdot\boldsymbol{r})^2}{\boldsymbol{r}^2} + \boldsymbol{\mu}^2 \right]_{\boldsymbol{r}=\boldsymbol{r}_a-\boldsymbol{r}_b}$$

PyDPFT: 3D dipole dipole

3D Dipole-dipole interaction in position (x) space

```
config = {
    'space':{'x':[-5,5,20],'y':[-5,5,20],'z':[-5,5,20]},
    'loop':{'Imax':1000,'precision':1e-6,'mix':0.05},
    'const':{'epsilon':1e-2,'mu':[0.7, 0.7, 0]},
    'rho':{'N':32},
    'Vint':{'name':'Dipole-x','coef':5},
}

dpft = PyDPFT(config)
Vext = dpft.xx**2 + dpft.yy**2 + dpft.zz**2
Vx,Vint,rho,N = dpft(Vext)
plot(dpft,Vx,Vint,rho)
```

```
PyDPFT: Written by Ding Ruiqi from NUS for his bachelor thesis
PyDPFT: Detected dim = 3
PyDPFT: Using 1 GPUs !
PyDPFT: Starting the self consistent loop
PyDPFT: Converged after 142 iterations in 3.1674556732177734 seconds!
```

Figure: PyDPFT: very simple to use, 3D

PyDPFT: 3D dipole dipole - momentum

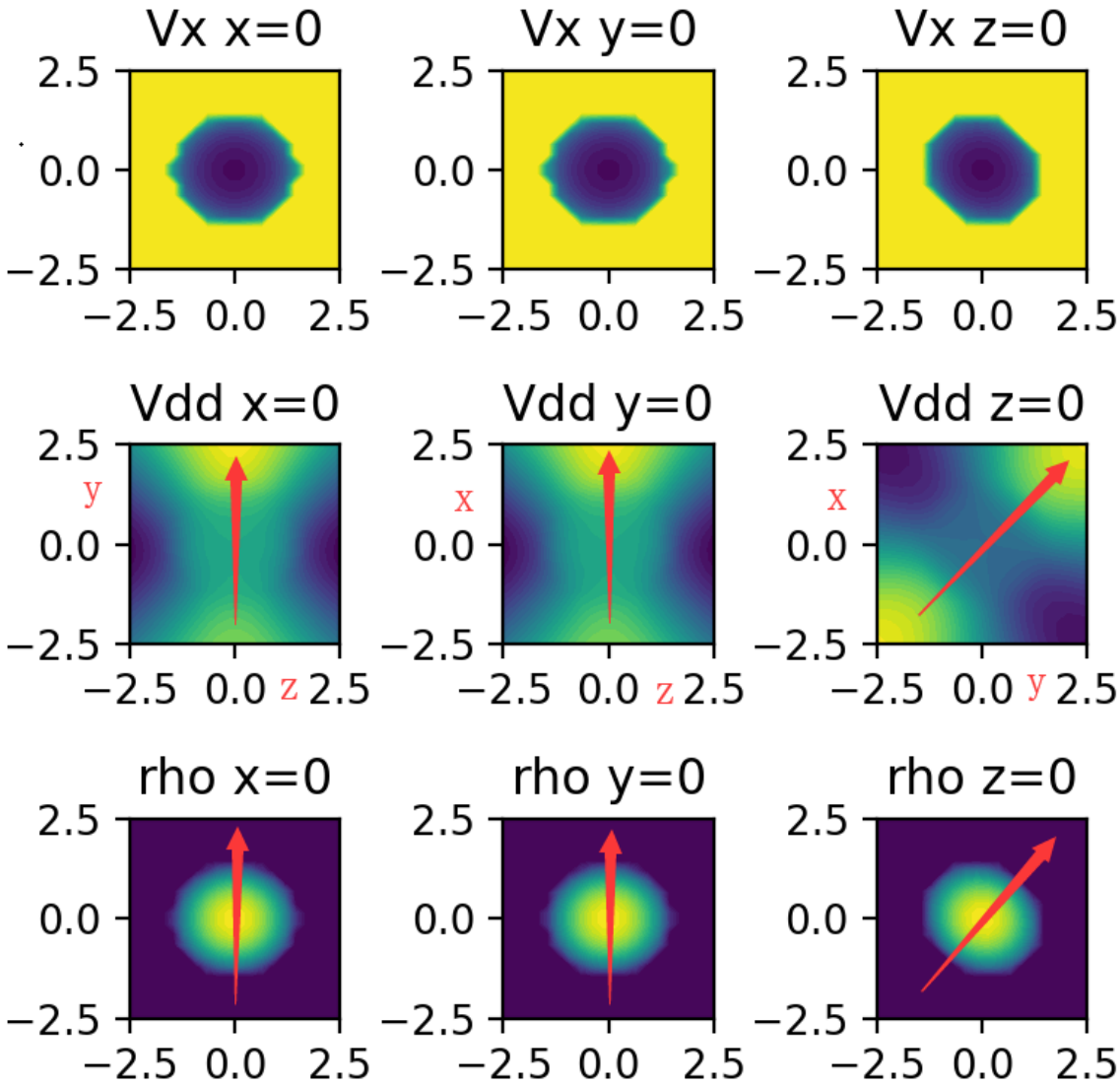


Figure: The result in 3D agrees well with 2D: momentum space

PyDPFT: 3D dipole dipole - position

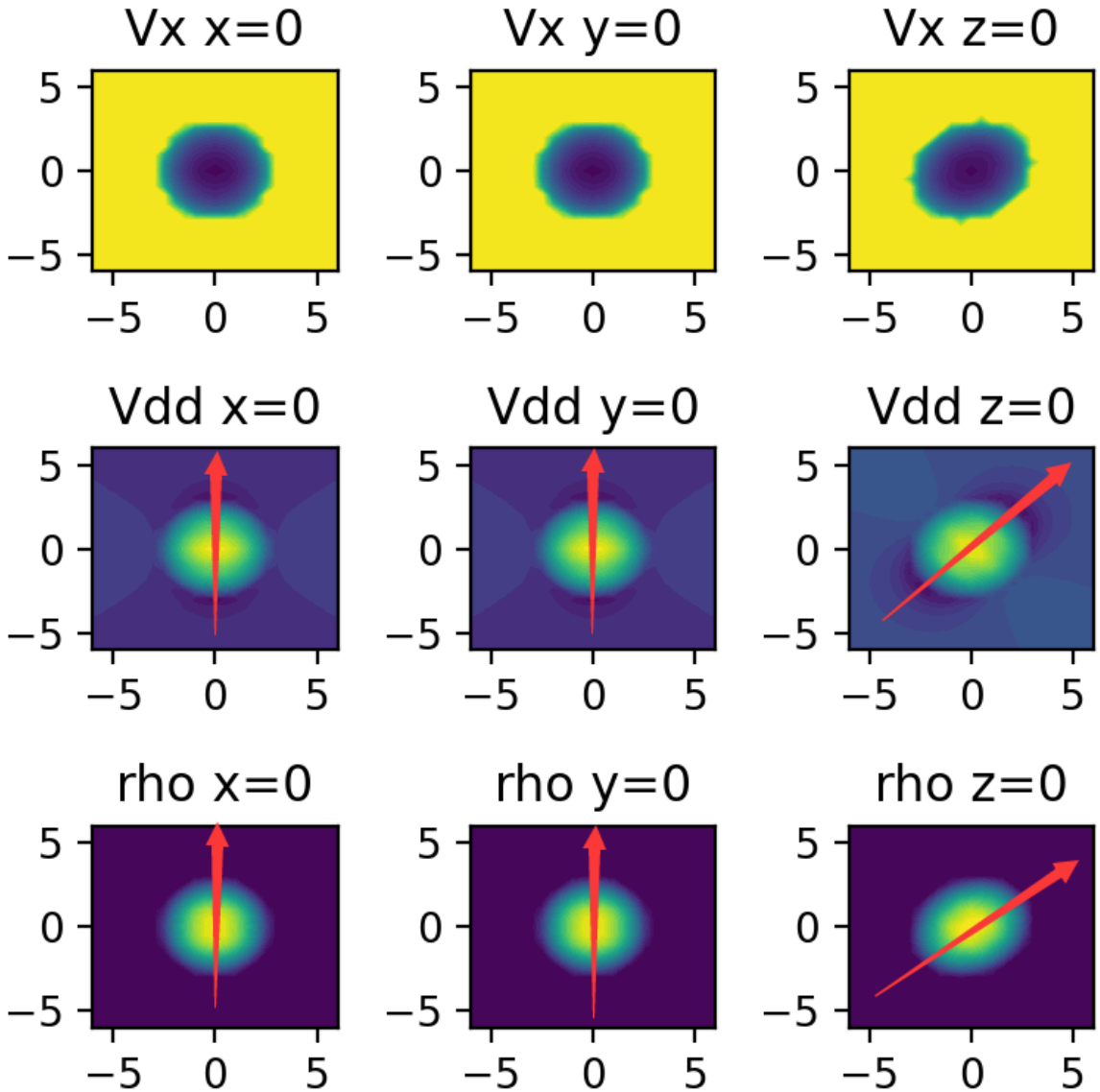


Figure: The result in 3D agrees well with 2D: position space

An initial exploration: filtering the TF density

- Discontinuity in TF density due to step function $\eta()$

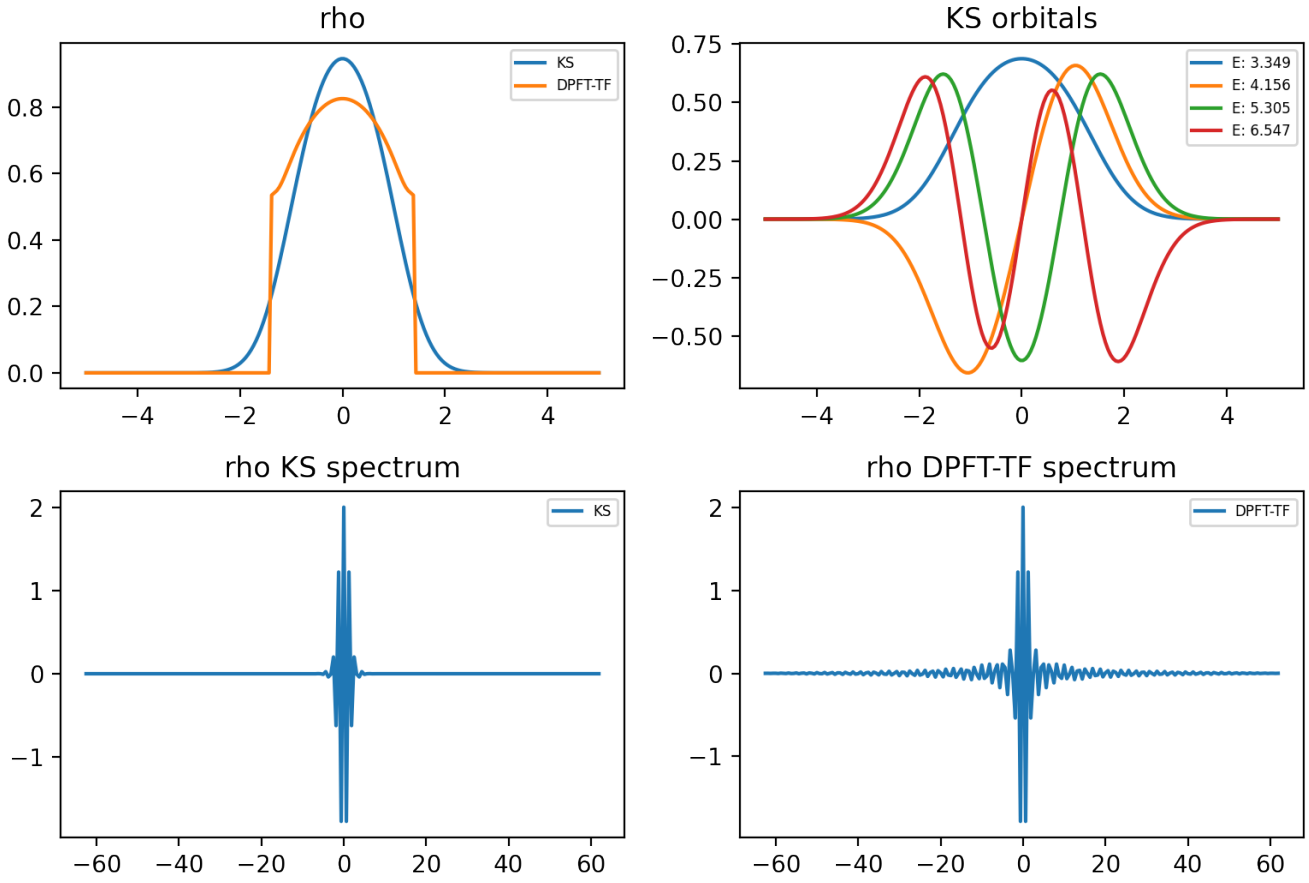


Figure: Original

An initial exploration: filtering the TF density

- Discontinuity in $\boldsymbol{r}$ or t space leads to high frequency responses in $\boldsymbol{k}$ or ω space

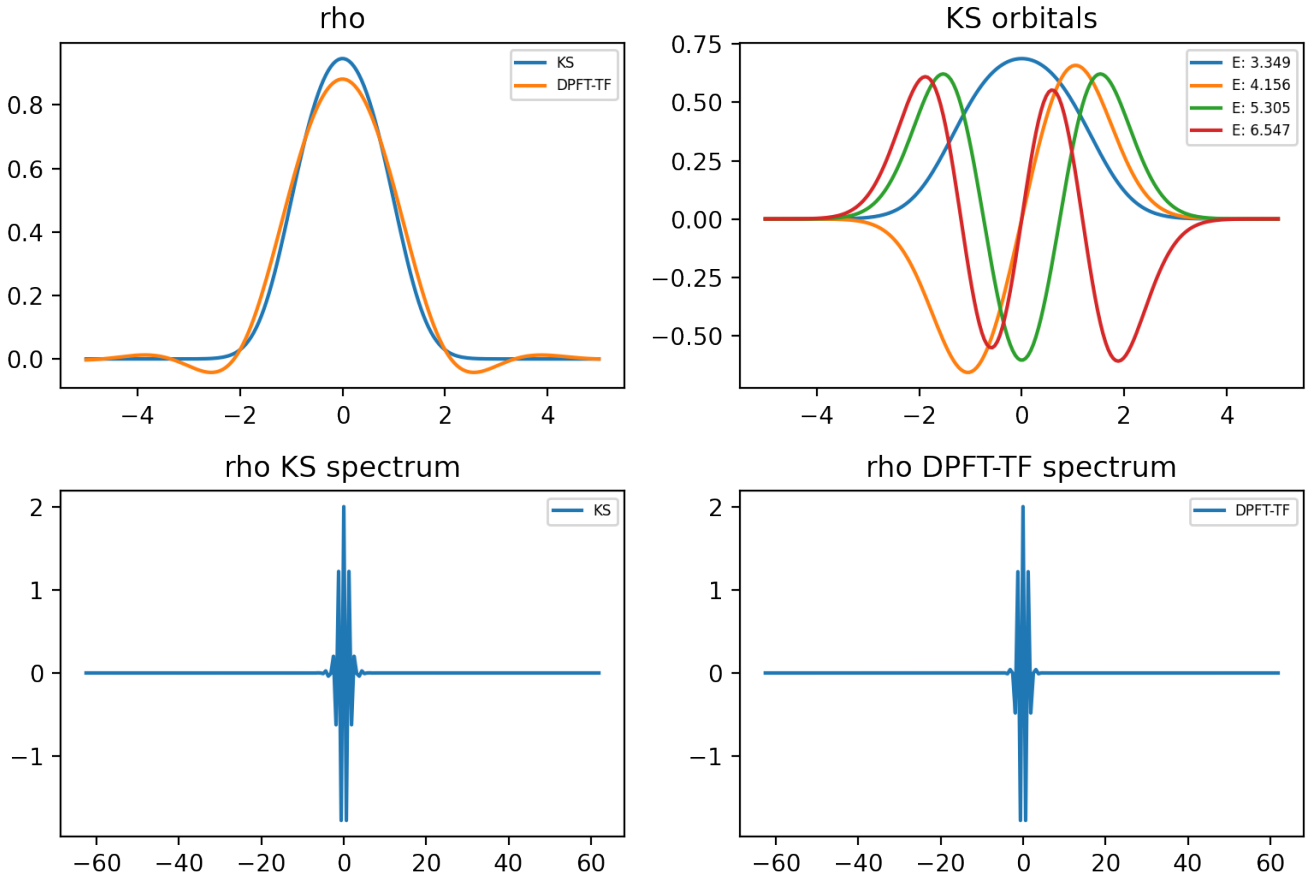


Figure: Filtered