Software engineering for validating finite-temperature XC-functional

Lokamani · m.lokamani@hzdr.de · FWCC-HZDR · www.hzdr.de
**XC – eXchange Correlation**

- Main ingredient of **DFT** → exchange-correlation **XC**
- **DFT** → **Density Functional Theory** at ground state (ambient conditions)
- Study properties of matter e.g. conductivity, bulk modulus
- Density as input for evaluating **XC** contributions

*Source: https://doi.org/10.1016/j.susc.2021.121877*
Matter under extreme conditions

- **DFT** can be used to study matter under extreme conditions
- But we need explicit temperature dependent **XC** to improve accuracy

- Gold-standard → **Path Integral Monte Carlo** (PIMC), very expensive and smaller systems
- Form of accurate **tXC** are known → implementation in legacy codes, not trustworthy
Initial discussion on implementing an explicit temperature dependent \( \text{XC} \) functional \( \text{tPBE} \)

Automated validation and first step toward democratizing temperature dependent \( \text{XC} \) for the scientific community

Steps toward an efficient implementation

Density functional theory

Hard-coded implementation of \( \text{tPBE} \)

Theory Department
Max Planck Institute of Microstructure Physics

Hackathon at MPI in Halle

Challenges while reverse engineering maple definition of \( \text{XC} \) in LIBXC & modifying interfaces to various ab-initio codes
Implementing thermal PBE in elk

Advantages

- Code easily accessible/readable
- found the $X$ and the derivative in `x_pbe.f90`
- found the $C$ and the derivative in `c_pbe.f90`
- Quick validation

$$\theta = \frac{T}{T_F} \quad r_s = \left(\frac{3}{4\pi n}\right)^{1/3}$$

Challenges

- Accurate derivative with respect to $r_s$ needed
- Hidden dependencies $\rightarrow \theta$ depends on $T_F$ which in turn depend on $r_s$
- Cumbersome and potential source of error if higher orders of derivatives are required
Implementing thermal PBE in LIBXC

Advantages

- Libxc use symbolic definitions for \( \text{XC} \)

\[
J_{\text{XC}}^{\text{unif}}(r_s, \theta) = -\frac{1}{r_s} \frac{a(\theta) + b(\theta)r_s^{1/2} + c(\theta)r_s}{1 + d(\theta)r_s^{1/2} + e(\theta)r_s}
\]

\[\text{fxc} := \text{omega}, \text{b}, \text{c}, \text{d}, \text{e}, \text{rs}, \text{t} \rightarrow \]
\[-(\text{omega}\ast\text{aa}(\text{t}) + \text{bb}(\text{b}, \text{t})\ast\sqrt{\text{rs}} + \text{cc}(\text{c}, \text{e}, \text{t})\ast\text{rs})/(\text{rs}\ast(1 + \text{dd}(\text{d}, \text{t})\ast\sqrt{\text{rs}} + \text{ee}(\text{e}, \text{t})\ast\text{rs})):\]

- automated evaluation of derivatives to arbitrary order
- Interface present in many ab-initio code
- First step toward democratizing temperature dependent \( \text{XC} \) functionals
  - no restriction to proprietary ab-initio codes or legacy codes
Results

- Visualizing thermal $\mathbf{XC}$ effects on electron density
- Relative differences between $t\text{PBE}$ and $\text{PBE}$
- 32 Hydrogen atoms at density $r_s = 4$

Fermi temperature with pronounced thermal $\mathbf{XC}$ effect

Thermal $\mathbf{XC}$ effects diminish with increasing temperature
Results

- Ab-initio quantum molecular dynamics
- Implementation more accurate than pure XC without temperature effects
- Efficient

<table>
<thead>
<tr>
<th>XC</th>
<th>Method</th>
<th>time/step (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBE</td>
<td>Internal</td>
<td>8.04</td>
</tr>
<tr>
<td>PBE</td>
<td>LIBXC</td>
<td>9.13</td>
</tr>
<tr>
<td>tPBE</td>
<td>LIBXC</td>
<td>9.35</td>
</tr>
</tbody>
</table>

tPBE efficiency comparable with PBE in LIBXC
Steps toward an efficient implementation

Density functional theory

Initial discussion on implementing an explicit temperature dependent $XC$ functional $t_{PBE}$

Automated validation and first step toward democratizing temperature dependent $XC$ for the scientific community

Hard-coded implementation of $t_{PBE}$

Challenges while reverse engineering maple definition of $XC$ in LIBXC & modifying interfaces to various ab-initio codes
Once you have clear documentation about how to set projects up, translating it into a GitLab CI pipeline is straightforward.
**Jacamar project**

- HPC focused CI/CD driver using GitLab's custom executor model
- GitLab Runner at HZDR
- Work-in-progress
  - Prototype implementation
  - Tackling issues regarding user management, workspace, permission and isolation

- [https://codebase.helmholtz.cloud/fwcc/gitlab-hpc-driver](https://codebase.helmholtz.cloud/fwcc/gitlab-hpc-driver)

(see [https://ecp-ci.gitlab.io/docs/admin/jacamar/introduction.html](https://ecp-ci.gitlab.io/docs/admin/jacamar/introduction.html))
Collaborative community-led structure for evaluating, verifying and improving the quality of research software and code
Collaborative community-led structure for evaluating, verifying and improving the quality of research software and code.

Thank you for your attention.
Implementing thermal PBE in LIBXC

- Refactoring maple recipe for tPBE
Implementing thermal PBE in LIBXC

- Modify CP2K interface to pass the temperature parameter to LIBXC