

# CDFTPY: A python package for performing classical density functional theory calculations for molecular liquids ☆, ☆☆



Marat Valiev<sup>a,\*</sup>, Gennady N. Chuev<sup>b,\*</sup>, Marina V. Fedotova<sup>c</sup>

<sup>a</sup> Environmental Molecular Sciences Laboratory, PNNL, Richland, 99352, WA, USA

<sup>b</sup> Institute of Theoretical and Experimental Biophysics, Russian Academy of Science, Pushchino, 142290, Moscow Region, Russia

<sup>c</sup> G.A. Krestov Institute of Solution Chemistry, Russian Academy of Sciences, Ivanovo, 153045, Russia

## ARTICLE INFO

### Article history:

Received 26 January 2022

Accepted 7 March 2022

Available online 10 March 2022

Dataset link: [10.17632/p8dsgz5n4g.1](https://doi.org/10.17632/p8dsgz5n4g.1)

### Keywords:

Molecular liquids

Classical density functional theory

Solvated particle

Site density

Reference interaction site model

Renormalized site density functional theory

## ABSTRACT

Classical density functional theory (CDFT) provides a rigorous theoretical framework for the statistical mechanics based analysis of many-body systems. This approach has proven to be successful in simulations of mono-atomic, i.e. simple liquids, and there is an ongoing theoretical effort in extending it to more complex polyatomic, molecular liquid systems. Sharing these developments in the form of open-source and easily accessible codes could greatly benefit these efforts. In this work, we present python-based CDFT code that contains both conventional Reference Interaction Site Model (RISM) and recently developed renormalized site density theory (RSDFT) approach. The current implementation is focused on ion solvation - the problem of both fundamental and practical importance. It allows the calculation of individual ions as well as comparative analysis across a range of interaction parameters.

### Program summary

Program Title: cdfpty

CPC Library link to program files: <https://doi.org/10.17632/p8dsgz5n4g.1>

Developer's repository link: <https://github.com/opencdft>

Licensing provisions: GPLv3

Programming language: python 3.9+

**Nature of problem:** Computational modeling of molecular liquids at the atomistic level of resolution is an important capability across many scientific areas. Classical density functional theory (CDFT) approaches this problem by building statistical mechanics model of the system in terms of its average atomic (site) density. Such an approach can provide orders of magnitudes improvements in efficiency compared to conventional molecular dynamics simulations, but requires special treatment of multi-scale interactions in a molecular liquid. A practical utility of our open-source python based implementation of CDFT is the study the problem solvation of ions or Lennard-Jones particles.

**Solution method:** Python package developed in this work provides two CDFT implementations for molecular liquids - renormalized site density functional theory and reference interaction site model. It enables calculations of thermodynamic and structural properties related to solvation of spherical solutes. The nonlinear integral equations associated with the two methods are solved iteratively, utilizing Fast Fourier Transform (FFT) for the calculation of the numerically intensive convolution integrals. The resulting code provides near instantaneous evaluation of the solvated properties of individual solutes and high-throughput screening across the range of different solute parameters.

© 2021 Elsevier B.V. All rights reserved.

## 1. Introduction

Computational modeling of inhomogeneous molecular liquids at the atomistic level of resolution is a problem of great practical importance to many scientific research efforts in energy, environment, and biology domains. The current solution to this problem involves primarily molecular dynamics (MD) or Monte-Carlo (MC) methods, which operate on the full configurational space of the

☆ The review of this paper was arranged by Prof. N.S. Scott.

☆☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

\* Corresponding authors.

E-mail addresses: [marat.valiev@pnnl.gov](mailto:marat.valiev@pnnl.gov) (M. Valiev), [genchuev@rambler.ru](mailto:genchuev@rambler.ru) (G.N. Chuev).

system. While such a view certainly provides the most detailed description of the system, it incurs enormous computational complexity at large length scales.

Classical density functional theory (CDFT) [1–4] takes a different approach to this problem, observing that from a thermodynamics standpoint the knowledge of instantaneous positions is not necessary and can be replaced by an average configurational description in terms of collective density variables. Such formulation bypasses the need for explicit statistical sampling associated with MD or MC simulations with the focus shifted on building the most appropriate statistical mechanics model of the system. The approach is appealing not only as a way of gaining fundamental understanding, but also as practical tool for calculation of quantities such as solvent density distributions and free energies at efficiencies far exceeding conventional MD applications.

Much of CDFT development and application efforts have been centered around primitive simple fluid representations of molecular liquids. The introduction of molecular structure detail has been a challenging affair and has stimulated a number of interesting research directions, such as the Fundamental Measure Theory (FMT) [5], the Interfacial Statistical Associating Fluid Theory (iSAFT) [6], and Molecular Density Functional Theory [7].

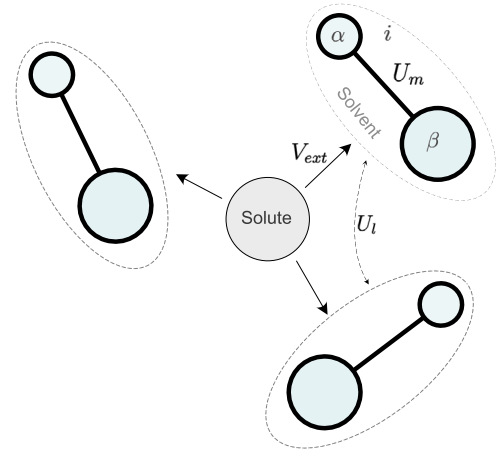
The approach that has been the focus of our research is based on the site density representation of the molecular liquid, introduced by Chandler in the form of the Reference Interaction Site Model (RISM) [8]. The basis of description, in this case, consists of average densities of atoms/sites in the solvent molecules:

$$\rho_\alpha(\mathbf{r}) = \left\langle \sum_{i=1}^N \delta(\mathbf{r} - \mathbf{x}_{i\alpha}) \right\rangle \quad (1)$$

where brackets denote statistical averaging and  $\mathbf{x}_{i\alpha}$  refer to the position of atom  $\alpha$  in solvent molecule  $i$ . The main appeal of such representation is a direct mapping to molecular mechanics models used in classical MD simulation and ability to analyze contributions of individual atoms or chemical groups. Bearing a close resemblance to simple fluid theory, RISM and its later extension 3D-RISM, [9,10] are able utilize many of the existing numerical techniques and have been used extensively in a wide range of problems [11–39].

One of the main shortcomings of RISM methods is related to the inadequate treatment of intra-molecular (i.e. chemical bond effects). [40] The issue arises from the fact that no provision is made to accommodate the stiff nature of these interactions. Instead, they are treated on the same footing as more diffused inter-molecular forces using hypernetted chain (HNC) approximation from simple liquid theory. This strategy, however, is not ideal, leading in some cases to unphysical results. To address this issue we have recently developed an alternative approach that separates the treatment of intra- and inter-molecular scales. Physically, this formulation reduces the molecular liquid system to the atomic mixture with renormalized inter-molecular interactions leading to what we refer to as Renormalized Site Density Functional Theory (RSDFT). The approach retains numerical efficiency of RISM methods but leads to significant improvements in accuracy.

The theoretical aspects of RSDFT approach has been discussed previously. [41,42] In this work we describe and share our current computational implementation of the approach for diatomic molecular liquids. Alongside RSDFT, we also provide an implementation of the RISM methodology. The latter, to our knowledge, is publicly available only as part of the AMBER [43,15] or NWChem [44,45] distributions, which complicates modifications or standalone applications. The code is built in the form of a Python package, which can be easily embedded in third-party applications or run on its own without specialized knowledge of CDFT. At this stage, it is mainly focused on the calculation of ion solvation -



**Fig. 1.** Solute-solvent system. Solvent molecules are indicated by dumbbells, while spherical particle corresponds to solute.

a standard benchmark reference problem that has provided valuable insight into the nature of solvation processes. In the following sections, we outline briefly the theoretical background including the RSDFT and RISM models, describe our computational approach, and provide several application examples.

## 2. Theoretical background

### 2.1. General considerations

Let us consider a molecular liquid system consisting of  $N$  solvent molecules of the same type enclosed in volume  $V$ . Each molecule is comprised of  $M$  atomic sites interacting with each other through general many-body potential  $\mathbf{U}$ , which is assumed to be decomposable into inter-molecular ( $\mathbf{U}_m$ ) and intra-molecular ( $\mathbf{U}_l$ ) components (see Fig. 1)

$$\mathbf{U} = \mathbf{U}_m + \mathbf{U}_l \quad (2)$$

The system is subject to an external site-dependent potential

$$V_{ext} = \sum_{i\alpha} v_\alpha(\mathbf{x}_{i\alpha}) \quad (3)$$

which, in the context of this work, will be associated with charged or neutral Lennard-Jones (LJ) particle. As such the interaction with solvent site  $\alpha$  located at distance  $r$  away from solute is given by:

$$v_\alpha(\mathbf{r}) = 4\varepsilon_{\alpha s} \left[ \left( \frac{\sigma_{\alpha s}}{r} \right)^{12} - \left( \frac{\sigma_{\alpha s}}{r} \right)^6 \right] + \frac{Qq_\alpha}{r} \quad (4)$$

where  $\sigma_{\alpha s}$  and  $\varepsilon_{\alpha s}$  denote solute-solvent LJ parameters and  $Q, q_\alpha$  solute and solvent site charges correspondingly. The standard combination rules,  $\sigma_{\alpha s} = \sigma_\alpha/2 + \sigma_s/2$  and  $\varepsilon_{\alpha s} = \sqrt{\varepsilon_\alpha \varepsilon_s}$  are assumed.

A common feature of both RSDFT and RISM methods is the use of the corresponding correlation functions in lieu of the explicit solvent-solvent interaction potentials. In particular, for molecular liquids the molecular gas structure factor  $\mathbf{S}_m(\mathbf{r})$  serves as a proxy for intra-molecular interactions ( $\mathbf{U}_m$ ) corresponding to the homogeneous solvent system where inter-molecular interactions are absent. For rigid molecular liquids molecular gas structure factor  $\mathbf{S}_m(\mathbf{r})$  can be written down explicitly as

$$S_{m,\alpha\beta}(\mathbf{r}) = \delta_{\alpha\beta} \delta(\mathbf{r}) + D_{\alpha\beta}^{(2)}(\mathbf{r}) \quad (5)$$

Here intra-molecular pair-correlation function  $\mathbf{D}^{(2)}(\mathbf{r})$  is given by

$$D_{\alpha\alpha'}^{(2)}(\mathbf{r}) = (1 - \delta_{\alpha\alpha'}) \frac{\delta(r - l_{\alpha\alpha'})}{4\pi r^2} \quad (6)$$

where  $l_{\alpha\alpha'}$  is the distance between  $\alpha$  and  $\alpha'$  atomic sites. The inter-molecular interactions are accounted for through the structure factor  $\mathbf{S}(\mathbf{r})$  of the full solvent system, reflecting both intra- and inter-molecular interactions. It differs from its molecular gas counterpart by extra contribution in the form of inter-molecular pair-correlation function  $H_{\alpha\beta}(\mathbf{r})$ :

$$S_{\alpha\beta}(\mathbf{r}) = S_{m,\alpha\beta}(\mathbf{r}) + \rho_M H_{\alpha\beta}(\mathbf{r}) \quad (7)$$

where  $\rho_M = N/V$  is the molecular density. We note that the inter-molecular pair-correlation function  $H_{\alpha\beta}(\mathbf{r})$  is directly connected to the familiar radial distribution function (RDF),  $H_{\alpha\beta}(\mathbf{r}) = g_{\alpha\beta}(\mathbf{r}) - 1$ , which can be obtained either from experiment or simulations.

## 2.2. RSDFT approach

The main system of equations in the RSDFT [41] approach is given by

$$\begin{cases} \mathbf{h}(\mathbf{r}) = [1 + \xi(\phi, \mathbf{r})] e^{-\beta(\mathbf{v}(\mathbf{r}) + \phi(\mathbf{r}))} - 1 \\ \beta\phi(\mathbf{r}) = [\mathbf{S}^{-1} - \mathbf{S}_m^{-1}] * \mathbf{h}(\mathbf{r}) \end{cases} \quad (8)$$

where  $\mathbf{h}(\mathbf{r})$  denotes density fluctuation function

$$\mathbf{h}(\mathbf{r}) = \frac{\rho(\mathbf{r}) - \rho_0}{\rho_0} \quad (9)$$

with  $\rho_0$  being the asymptotic value of density at large distances ( $\rho_0 \equiv \lim_{r \rightarrow \infty} \rho(\mathbf{r})$ ). Correlation hole,  $\xi(\phi, \mathbf{r})$ , contains effects of intra-molecular interactions and for the diatomic molecules can be expressed as

$$\xi(\phi, \mathbf{r}) = [\mathbf{D}^{(2)} * \mathbf{f}(\phi)](\mathbf{r}) \quad (10)$$

where  $\mathbf{f}(\phi, \mathbf{r})$  is the Mayer function:

$$\mathbf{f}(\phi, \mathbf{r}) = e^{-\beta(\mathbf{v}(\mathbf{r}) + \phi(\mathbf{r}))} - 1 \quad (11)$$

Evaluation of the correlation potential  $\phi(\mathbf{r})$  is facilitated by introducing, an additional variable - direct correlation function

$$\bar{\mathbf{c}}(\mathbf{r}) = [(\mathbf{S}_m \mathbf{S}^{-1}) * \mathbf{h}](\mathbf{r}) \quad (12)$$

This way, the inversion of the full structure factor  $\mathbf{S}$  can be avoided in lieu of pair of coupled equations

$$\begin{cases} \bar{\mathbf{c}}(\mathbf{r}) = \mathbf{h}(\mathbf{r}) + \beta[\mathbf{S}_m * \phi](\mathbf{r}) \\ \beta\phi(\mathbf{r}) = -\rho_0[\tilde{\mathbf{H}} * \bar{\mathbf{c}}](\mathbf{r}) \end{cases} \quad (13)$$

where

$$\rho_0 \tilde{\mathbf{H}} = \mathbf{S}_m^{-1} \cdot (\mathbf{S} - \mathbf{S}_m) \cdot \mathbf{S}_m^{-1} \quad (14)$$

The convolution operation is performed in reciprocal space as

$$[\tilde{\mathbf{H}} * \bar{\mathbf{c}}](\mathbf{r}) \xrightarrow{FFT} \tilde{\mathbf{H}}(\mathbf{k}) \cdot \bar{\mathbf{c}}(\mathbf{k}) \quad (15)$$

Given the low- $k$  behavior of direct correlation function

$$\lim_{k \rightarrow 0} \bar{\mathbf{c}}(\mathbf{k}) = -\beta \mathbf{S}_m(\mathbf{k}) \cdot \mathbf{v}(\mathbf{k}) \quad (16)$$

the evaluation of (15) may be problematic for slow decaying Coulomb component of the solute potential. Thus, we separate its treatment by decomposing solute potential into short and long range components as

$$\mathbf{v}(\mathbf{r}) = \mathbf{v}_s(\mathbf{r}) + \mathbf{v}_l(\mathbf{r}) \quad (17)$$

choosing long range component to be

$$v_{l\alpha}(\mathbf{r}) = \frac{Q q_\alpha}{r} \operatorname{erf}\left(\frac{r}{r_0}\right) \quad (18)$$

Correspondingly the short range component is given by

$$v_\alpha(\mathbf{r}) = 4\varepsilon_{\alpha s} \left[ \left( \frac{\sigma_{\alpha s}}{r} \right)^{12} - \left( \frac{\sigma_{\alpha s}}{r} \right)^6 \right] + \frac{Q q_\alpha}{r} \left( 1 - \operatorname{erf}\left(\frac{r}{r_0}\right) \right) \quad (19)$$

Then introducing short range direct correlation function (cf. (16))

$$\bar{\mathbf{c}}_s(\mathbf{r}) = \bar{\mathbf{c}}(\mathbf{r}) + \beta[\mathbf{S}_m * \mathbf{v}_l](\mathbf{r}) \quad (20)$$

and defining indirect correlation function

$$\boldsymbol{\gamma}(\mathbf{r}) = -\beta(\mathbf{v}_l(\mathbf{r}) + \phi(\mathbf{r})) \quad (21)$$

the RSDFT system equations can be rewritten as

$$\begin{cases} \mathbf{h}(\mathbf{r}) = [1 + \xi(\boldsymbol{\gamma}, \mathbf{r})] e^{-\beta \mathbf{v}_s(\mathbf{r}) + \boldsymbol{\gamma}(\mathbf{r})} - 1 \\ \bar{\mathbf{c}}_s(\mathbf{r}) = \mathbf{h}(\mathbf{r}) - [\mathbf{S}_m * \boldsymbol{\gamma}](\mathbf{r}) \\ \boldsymbol{\gamma}(\mathbf{r}) = \rho_0[\tilde{\mathbf{H}} * \bar{\mathbf{c}}_s](\mathbf{r}) + \boldsymbol{\gamma}_l(\mathbf{r}) \end{cases} \quad (22)$$

where

$$\boldsymbol{\gamma}_l(\mathbf{r}) = -\beta[(\mathbf{I} + \rho_0 \tilde{\mathbf{H}} \mathbf{S}_m) * \mathbf{v}_l](\mathbf{r}) \quad (23)$$

Once converged solution has been obtained, the free energy of solvation can be calculated as

$$\Delta G = -\frac{\rho_0 \cdot \bar{\mathbf{c}}_s}{\beta M} - \frac{\phi \cdot \Delta \rho}{2} \quad (24)$$

## 2.3. RISM approach

RISM represents an approximate version of RSDFT approach described in the previous section. Namely, the correlation hole  $\xi(\mathbf{r})$  is no longer treated exactly (see (10)), but approximated as

$$\xi(\mathbf{r}) \approx e^{[\mathbf{D}^{(2)} * \rho](\mathbf{r})} - 1 \quad (25)$$

In the context of RSDFT equations (22), such approximation is equivalent to setting  $\mathbf{S}_m$  to identity  $\mathbf{I}$  leading to the following result:

$$\begin{cases} \mathbf{h}(\mathbf{r}) = e^{-\beta \mathbf{v}_s(\mathbf{r}) + \boldsymbol{\gamma}(\mathbf{r})} - 1 \\ \mathbf{c}_s(\mathbf{r}) = \mathbf{h}(\mathbf{r}) - \boldsymbol{\gamma}(\mathbf{r}) \\ \boldsymbol{\gamma}(\mathbf{r}) = \rho_0[\tilde{\mathbf{H}} * \mathbf{c}_s](\mathbf{r}) + \boldsymbol{\gamma}_l(\mathbf{r}) \end{cases} \quad (26)$$

where

$$\rho_0 \tilde{\mathbf{H}} = \mathbf{S} - \mathbf{I} \quad (27)$$

and

$$\boldsymbol{\gamma}_l(\mathbf{r}) = -\beta[\mathbf{S} * \mathbf{v}_l](\mathbf{r}) \quad (28)$$

The expression for solvation free energy expression remains the same

$$\Delta G = -\frac{\rho_0 \cdot \mathbf{c}_s}{\beta} - \frac{\phi \cdot \Delta \rho}{2} \quad (29)$$

except that now it is calculated with respect to  $\mathbf{c}_s$  and not  $\bar{\mathbf{c}}_s$ .

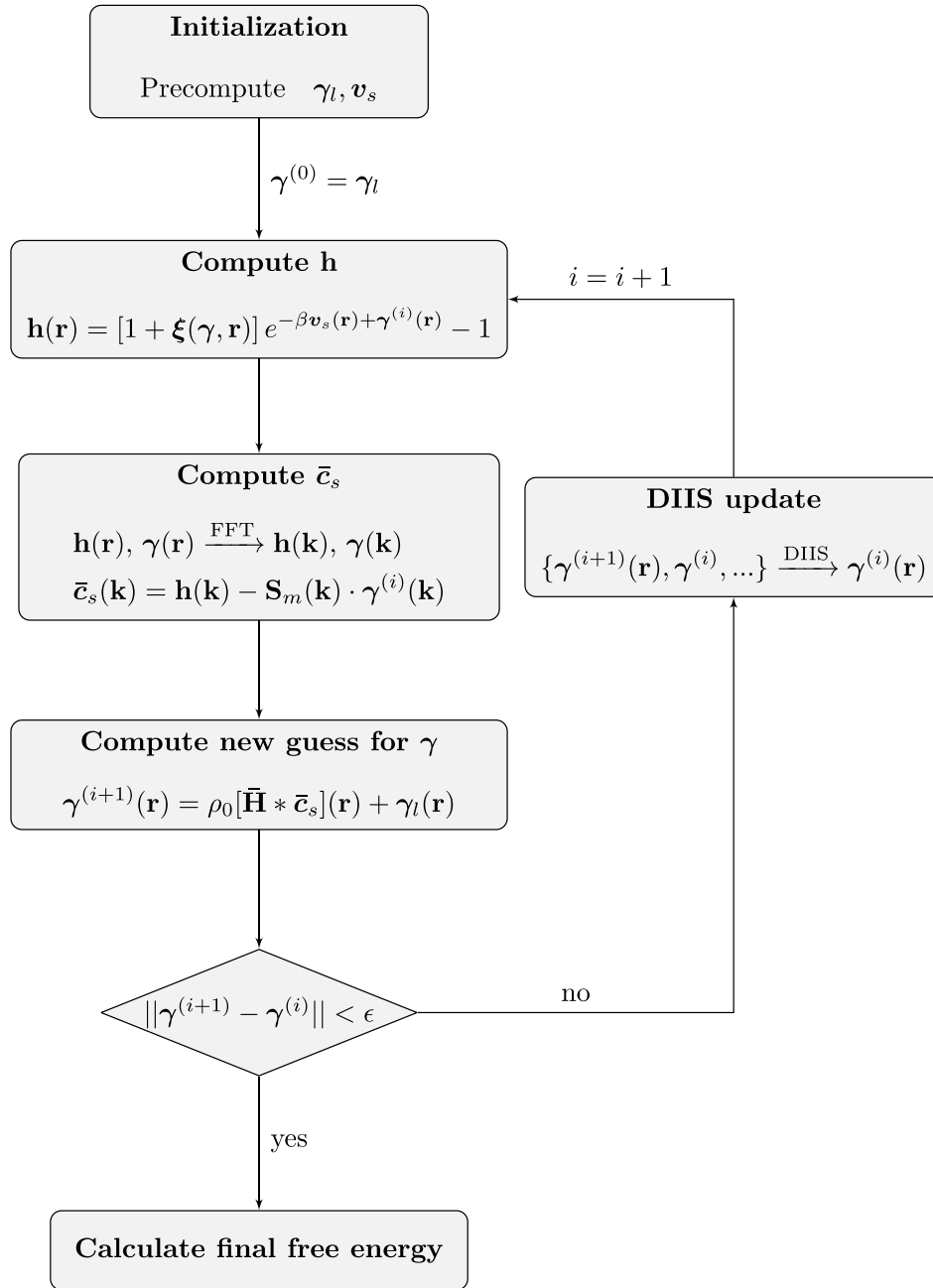


Fig. 2. RSDFT self-consistent cycle.

### 3. Implementation

#### 3.1. Algorithm

The schematics of the algorithm for the solution RSDFT equations (22) is shown on Fig. 2. The solution for RISM equations is approached in a similar way. The procedure involves a self-consistent loop consisting of iterative updates to indirect correlation function  $\gamma(r)$ , with calculations involving both real and reciprocal spaces.

Prior to self-consistent loop *initialization step* is performed, involving processing of the input file and pre-computing the short-range potential  $v_s$  and long range component of indirect correlation function  $\gamma_l$ . The initial guess for  $\gamma$  is set to  $\gamma_l$ . The first step in the iterative cycle involves calculation of density fluctuation

function  $h(r)$  using current value of the indirect correlation function  $\gamma(r)$ . In the second step, after the transformation of both  $h(r)$  and  $\gamma(r)$  into the reciprocal space, we calculate short range direct correlation function  $\bar{c}_s(k)$ . In the third step we calculate new guess for indirect correlation function  $\gamma(r)$ . If the difference between old and new guess for  $\gamma$  falls below specified convergence criteria, the self-consistent loop is exited. Otherwise, the new value for  $\gamma$  is generated following DIIS procedure [46] and the entire process is repeated. Upon convergence, the final value of the solvation free energy is calculated.

Given the spherically symmetric nature of the single ion solute, the real space representation can be defined solely on the basis of the radial distance  $r$  from the solute ion, which serves as the origin of the coordinate system. Correspondingly the three-dimensional

Fourier transforms can be reduced to one-dimensional Fourier Sine transform (FST):

$$h(k) = \iiint h(r) e^{-i\mathbf{k}\cdot\mathbf{r}} d\mathbf{r} = \frac{4\pi}{k} \int_0^\infty rh(r) \sin(kr) dr \quad (30)$$

$$h(r) = \iiint h(k) e^{i\mathbf{k}\cdot\mathbf{r}} d\mathbf{k} = \frac{1}{2\pi^2 r} \int_0^\infty kh(k) \sin(kr) dk \quad (31)$$

In our case the above expressions are evaluated using Type IV Discrete Sine Transform (DST). [47] As such, the real space representation involves uniform midpoint-based grid covering the finite range from 0 up to cutoff distance  $R_{max}$ :

$$r[i] = \left(i - \frac{1}{2}\right) \frac{R_{max}}{N}, \quad i = 1, \dots, N \quad (32)$$

The corresponding grid in reciprocal space is given by

$$k[i] = \left(i - \frac{1}{2}\right) \frac{\pi}{R_{max}}, \quad i = 1, \dots, N \quad (33)$$

The solvent structure **S** is currently pre-calculated using classical molecular dynamics simulation, as described in [48,49]. The available solvents in the current version of the code are water under normal conditions (SPCE [50] and S2 [51] models), molecular nitrogen N2, polar and apolar HCl solutions (HCl and HCl\_neutral, respectively). However the SPCE usage is currently restricted to RISM method only.

### 3.2. Code structure

The `cdftpy` package provides python (version 3.9) implementation of RSDFT and RISM solution procedures described in the previous section. The external dependencies are kept at minimum, and include `numpy`, `scipy`, `click`, `prettytable`, `panel`, and `holoviews` python packages. The `numpy` is utilized to perform matrix/array operations and `scipy` for FFT calculations. Command line interface is facilitated by `click`. Dashboard capabilities are implemented using `panel` and `holoviews`, with `prettytable` used for tabular output.

The `cdftpy` package is comprised of several modules, located in `cdftld` subdirectory. In particular, the `simulation` module defines `IonSolvation` class that encapsulates all the information required for calculations with RSDFT/RISM calculations triggered by `cdft` method. The information about the solvent and related operations is handled by `Solvent` class defined in `solvation` module. The operations pertaining to electrostatic and short-range potentials are facilitated by `coulomb` and `potential` modules correspondingly. Module `rad_fft` provides wrappers to external FFT calls to `scipy`. The rest of the modules are described in `readme.txt` file, which can be found in the code distribution. The solvent data files in the form of `smdl` files (see Appendix A for details) are located in the `data` directory

### 3.3. Usage

The easiest way to utilize `cdftpy` package in the external applications is through the use of `IonSolvation` class. Simplest example of how this can be accomplished is shown below.

```
from cdftpy import IonSolvation
sim = IonSolvation(charge=-1.0, sigma=4.83, eps=0.0535)
free_energy = sim.cdft()
```

```
<solute>
#name  sigma (A)      eps (kj/mol)    charge (e)
Cl      4.83         0.05349244     -1.0
<simulation>
solvent s2
method rsdft
tol 1.0E-7
max_iter 200
output_rate 10
rcoul 1.25
rmax 100
```

Fig. 3. Example input file for  $\text{Cl}^-$  ion solvation.

Here the charge specifies charge of the ion in the elementary (e) charge units. The LJ parameters  $\sigma$  and  $\varepsilon$  (see Eq. (4)) are provided as `sigma` and `eps` in Å and kJ/mol units correspondingly. Initialized this way, the invocation of the `cdft` method runs default RSDFT calculation with `s2` water model.

Full set of initialization parameters for `IonSolvation` class is shown below.

```
IonSolvation(charge=None, sigma=None, eps=None,
             name="ion",
             method="rsdft",
             solvent="s2",
             rmax = None,
             ndiis=2,
             tol=1.0e-9,
             output_rate=10,
             max_iter=200,
             rcoul=1.25
            )
```

Parameters `charge`, `sigma`, and `eps` are required, and the rest are optional with default values as indicated.

- `name` assigns name to an ion
- `method` defines calculation method, either "rsdft" or "rism"
- `solvent` specifies solvent, such as "s2", "hcl", "hcl\_neutral", "n2", or "spce" (see Appendix A for more information)
- `rmax` allows to redefine size of the simulation, which would otherwise be based on the internal simulation size of the solvent
- `max_iter` allows to define maximum number of iterations after which calculation will stop regardless of convergence status.
- `tol` denotes accepted convergence tolerance for indirect correlation function  $\gamma(\mathbf{r})$  (see Fig. 2)
- `output_rate` defines frequency of the output.
- `rcoul` specifies Coulomb cutoff radius ( $r_0$  in (18))
- `ndiis` specifies number of diis vectors

For command line applications `cdftpy` package provides executable script `cdftld`. The latter requires an input file, an example of which is shown on Fig. 3. The input file is free format with comments lines denoted by hash symbol at the beginning of the line. It is comprised of several sections, with the beginning of each section marked by its name enclosed in angular brackets:

- `<solute>` is a required section, containing specification of the solute ion comprised of its name, LJ parameters ( $\sigma$  in Å and  $\varepsilon$  in kJ/mol), and charge.
- `<simulation>` is an optional section that contains simulation parameters, which, with the exception of the name,

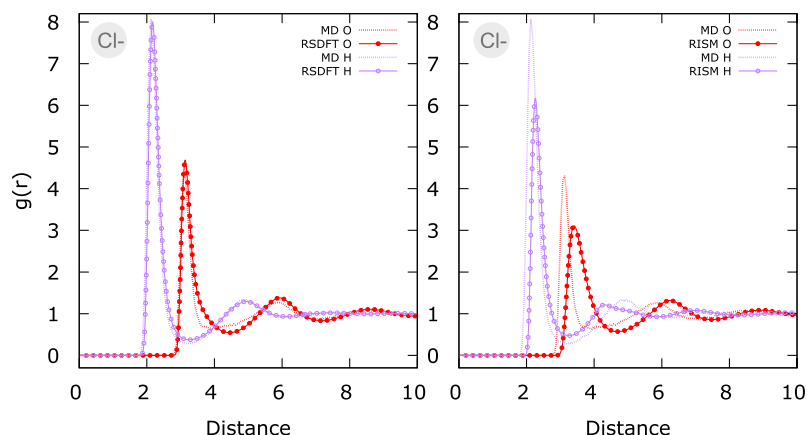


Fig. 4. Normalized solvent site densities  $g(r) = \rho(r)/\rho_0$  around  $\text{Cl}^-$  ion obtained with RSDFT (left) and RISM (right) calculations. Reference classical MD data is also shown.

mirrors optional initialization parameters for IonSolvation class

With the input file at hand CDFT calculation can be ran as simply as

```
cdftld INPUT_FILE
```

Additionally, number of options can be provided to modify the calculation parameters. The complete list of those is shown below

Usage: cdftld [OPTIONS] INPUT\_FILE

Options:

```
-m, --method [rism|rsdft]
    Calculation method (default:rsdft)
-s, --solvent <solvent model>
    solvent model [default:s2]
-a, --adjust [charge|sigma|eps] <value>
    adjust solute parameters
-r, --range [charge|sigma|eps] <values>
    Run calculation over the range of solute
    "charge","sigma","eps" values. Values could
    specified as comma separated sequence (e.g.
    0,0.5,...) or in triplets notation
    [start]:stop:nsteps. To avoid issues with
    blank spaces, it is recommended that values
    are enclosed in double quotes.
-o, --output
    generate data output (currently only site density)
-d, --dashboard [filename]
    Generate dashboard for analysis.
    The dashboard will be saved as html file
    under the name provided by optional argument.
    In the absence of the latter dashboard will be
    open in browser
--version
    display version and exit
--help
    Show this message and exit.
```

The first three options (-m, -s, -a) allow to override method of calculation, solvent, and solute parameters specified in the input file or by default. The -r option activates what could be called a range calculation, where series of calculations are performed over range of different solute parameters. The -o option enables output of normalized site densities  $\Delta\rho(r)/\rho_0$  in the run directory under the name density.dat. For series of calculations (as triggered by -r option) numbered density file will be generated for each point (e.g. density-00.dat, density-01.dat,...). The -d option generates interactive html dashboard for analysis of calculation results, which could be displayed in the browser window or saved as html file. The latter is self-contained including both the

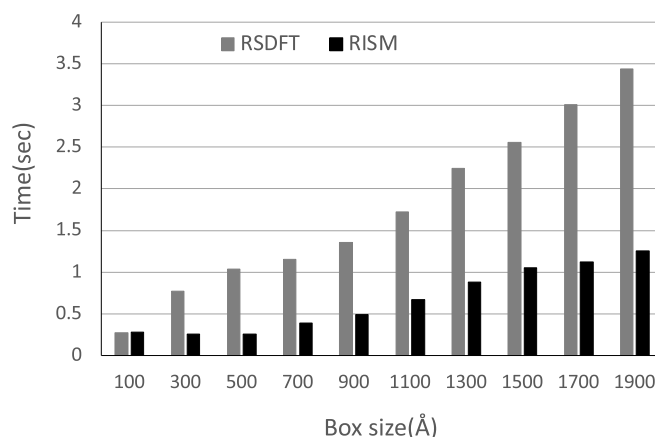


Fig. 5. CPU time (seconds) as a function box size (Å).

data and its rendering. This is an [example of dashboard](#) for single ion solvation calculation. For range calculation (-r option) it will look like [this](#).

Example of such command line calculation for  $\text{Cl}^-$  solvation (as defined by the input file from Fig. 3) can be found on Fig. 4. Both, RSDFT and RISM, calculations utilized real space grid consisting of 8192 points with the grid step equal to 0.01 Å (approximate box size of 82 Å). Molecular dynamics calculations were based on the AMBER package [43] using 60 Å cubic simulation box containing 7227 solvent molecules. All calculations were ran at temperature set to 300 K. The calculated solvent density profiles, shown on Fig. 4, demonstrate that RSDFT approach provides much better accuracy compared to RISM, yielding nearly the same results as classical MD simulations. At the same time RSDFT calculations are orders of magnitude more efficient, taking fractions of a second to complete as opposed to hours of classical MD runs. This point is further illustrated on Fig. 5 which shows total CPU times of RSDFT/RISM calculations for different box sizes using commodity (Macbook Pro) laptop. The main numerical cost are the Fourier transforms characterized by  $\approx N \log N$  scaling. As expected RSDFT calculations take longer to complete due to increased number of operations. Nevertheless, even for box sizes as large 1900 Å, the total CPU time remains on the order of several seconds - a feat which would be near impossible to achieve by conventional MD calculations even on the most advanced super-computing architectures. Such efficiency, provides near real time capability to probe ion solvation processes across different physical parameters. On example of such application is shown on Figs. 6 and 7, where RSDFT



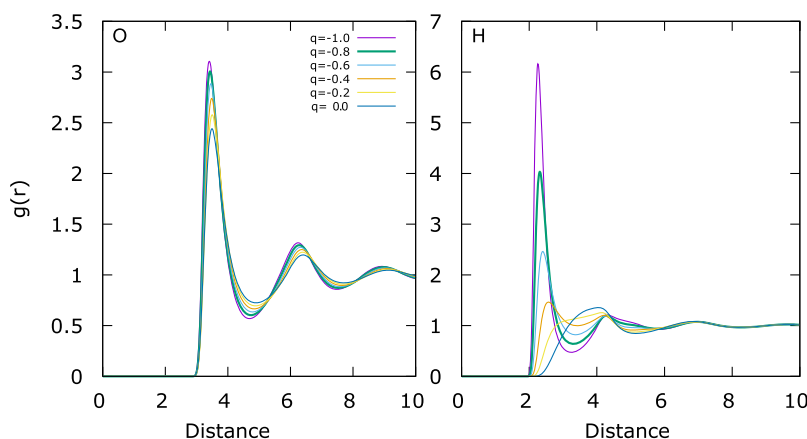


Fig. 6. Normalized solvent oxygen (left) and hydrogen (right) site densities  $g(r) = \rho(r)/\rho_0$  around  $\text{Cl}^-$  ion obtained with RSDFT over the range of charge states.

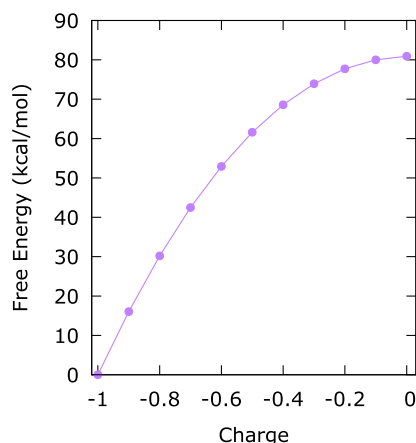


Fig. 7. Relative Free energy of solvated  $\text{Cl}^-$  over the range of charge states.

calculations were run across different the range of different charge parameters.

#### 4. Conclusion and outlook

The `cdftpy` python package calculates structural and thermodynamical parameters of ion solvation in a molecular liquid using RSDFT and RISM implementations of CDFT. Other than recent Python version (3.9 or higher) and a few common python packages (e.g. `numpy`, `scipy`, etc.), the code is self-contained and can be used either from command line or as a library. The code was built with intention of being accessible to a wide range users and, in addition to standard static data files, can generate an interactive dashboard for calculation analysis. The hardware requirements are modest, with single ion calculations taking fractions of a second. With such efficiency, it becomes easy to investigate changes in solvation behavior over the interaction parameters, which is one of the options offered by the program.

The modular nature of the code, the use of Python programming language meant to ease the task of development new functionalities. Among planned developments are different functional forms for solute-solvent interactions (e.g. hard sphere potentials), additional solvent types, and poly-atomic solutes. The `cdftpy` package is released under the open source license, with source code hosted on <https://github.com/opencdft>. We hope that it will serve as a useful resource for general DFT community.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgements

M.V. was supported by the U.S. Department of Energy, Office of Basic Energy Sciences, Division of Chemical Sciences, Geosciences, and Biosciences. PNNL is operated by Battelle Memorial Institute for the United States Department of Energy under DOE Contract Number DE-AC05-76RL1830.

The work of G.N.C. and M.V.F. in part of methodology of RSDFT and RISM approaches (subsections 2.2 and 2.3) was supported by the Russian Science Foundation (project No. 22-23-00184, <https://rscf.ru/en/project/22-23-00184/>).

#### Appendix A. Solvent models

The solvent models in the `cdftpy` package are provided in the form of `smdl` files:

```
<solvent_name>.smdl
```

The format of `smdl` file is similar to input files utilized in `cdftpy` (see Fig. 3). They are comprised of several sections, with the beginning of each section marked by its name enclosed in angular brackets. The description of the sections is provided below

- `<geometry>` section defines solvent geometry comprised of the name, type and  $x, y, z$  coordinates in Å units for each atom in the solvent molecule
- `<interaction>` section defines van der Waals parameters ( $\sigma$  in Å and  $\epsilon$  in kJ/mol), and charge for each atom type used in `geometry` section
- `<rism_patch>` options section that allows to override parameters defined in `<interaction>` section for RISM calculation.
- `<state>` section defines solvent density in Å<sup>-3</sup> units, temperature in Kelvin and dielectric constant
- `<s_k>` section defines structure factor in k-space. The first line in the section specifies length of the `<s_k>` along k-space dimension and number of solvent sites  $n$ . Following that there are  $1 + n(n+1)/2$  columns. The first column contains k-space grid and the rest are structure factors for all of  $n(n+1)/2$  pairs of sites in the system.

## Parameters of s2 model

```

<geometry>
# site   type   x       y       z
  O      OW     1.0     2.0     1.0
  H      HW     1.0     1.0     1.0
<interactions>
# type      sigma      eps (kJ/mol)      charge (e)
  OW        3.1600E+00    0.648954      -0.38
  HW        0.0000E+00    0.19260       0.38
<rism_patch>
# type      sigma      eps (kJ/mol)
  HW        1.0000E+00    0.19260
<state>
density 0.0334583
temp 300
dielectric 70.0

```

## Parameters of n2 model

```

<geometry>
# site   type   x       y       z
N1       N      0       0       0
N2       N      1.1345  0       0
<interactions>
#type      sigma      eps      charge
  N         3.2513     0.7113    0.0
  N         3.2513     0.7113    0.0
<state>
density 0.022016
temp 300
dielectric 1.0

```

## Parameters of hcl model

```

<geometry>
# site   type   x       y       z
Cl       Cl     0       0       0
H        H      1.3     0       0
<interactions>
#type      sigma      eps      charge
  Cl        3.353      2.15     -0.2
  H         0.000178   0.166     0.2
<rism_patch>
# type      sigma      eps (kJ/mol)
  HW        1.0       0.166
<state>
density 0.02337
temp 300
dielectric 1.0

```

## Parameters of hcl\_neutral model

```

<geometry>
# site   type   x       y       z
Cl       Cl     0       0       0
H        H      1.3     0       0
<interactions>
#type      sigma      eps      charge
  Cl        3.353      2.15     0.0
  H         0.000178   0.166     0.0
<rism_patch>
# type      sigma      eps (kJ/mol)
  HW        1.0       0.166
<state>
density 0.02337
temp 300
dielectric 1.0

```

## Parameters of spce model

```

<geometry>
# site   type   x       y       z
  O      O      0.0     0.0     0.0
H1      H      1.0     0.0     0.0
H2      H      -0.33331325194  -0.94281614118  0.0
<interactions>
#type      sigma      eps      charge
  O         3.16      0.65     -0.8476
  H         1.0       0.1926    0.4238
  H         1.0       0.1926    0.4238
<state>
density 0.03332746082101245
temp 298
dielectric 70.0

```

## References

- [1] R. Evans, Adv. Phys. 28 (2) (1979) 143–200.
- [2] A.P. Hughes, U. Thiele, A.J. Archer, Am. J. Phys. 82 (12) (2014) 1119–1129, <https://doi.org/10.1119/1.4890823>.
- [3] D. Henderson, Integral Equation Theories for Inhomogeneous Fluids, Fundamentals of Inhomogeneous Fluids, Marcel Dekker Inc., 1992.
- [4] J.-P. Hansen, I.R. McDonald, Theory of Simple Liquids: With Applications to Soft Matter, Elsevier Academic Press, Amsterdam, the Netherlands, 2013.
- [5] Y. Rosenfeld, Phys. Rev. Lett. 63 (1989) 980–983, <https://doi.org/10.1103/PhysRevLett.63.980>.
- [6] S. Tripathi, W.G. Chapman, J. Chem. Phys. 122 (9) (2005) 094506, <https://doi.org/10.1063/1.1853371>.
- [7] D. Borgis, S. Luukkonen, L. Belloni, G. Jeanmairet, J. Phys. Chem. B 124 (31) (2020) 6885–6893, <https://doi.org/10.1021/acs.jpcc.0c04496>.
- [8] D. Chandler, H.C. Andersen, J. Chem. Phys. 57 (5) (1972) 1930–1937.
- [9] D. Beglov, B. Roux, J. Phys. Chem. B 101 (1997) 7821–7826.
- [10] Q.H. Du, D. Beglov, B. Roux, J. Phys. Chem. B 104 (4) (2000) 796–805.
- [11] J.J. Howard, J.S. Perkyns, N. Choudhury, B.M. Pettitt, J. Chem. Theory Comput. 4 (11) (2008) 1928–1939.
- [12] S.M. Kast, Phys. Chem. Chem. Phys. 3 (2001) 5087–5092, <https://doi.org/10.1039/B106075A>.
- [13] G.N. Chuev, M.V. Fedorov, J. Chem. Phys. 131 (2009) 074503.
- [14] S. Genheden, T. Luchko, S. Gusarov, A. Kovalenko, U. Ryde, J. Phys. Chem. B 114 (25) (2010) 8505–8516.
- [15] T. Luchko, S. Gusarov, D.R. Roe, C. Simmerling, D.A. Case, J. Tuszynski, A. Kovalenko, J. Chem. Theory Comput. 6 (3) (2010) 607–624.
- [16] J.S. Perkyns, G.C. Lynch, J.J. Howard, B.M. Pettitt, J. Chem. Phys. 132 (2010) 064106.
- [17] D.S. Palmer, G.N. Chuev, E.L. Ratkova, M.V. Fedorov, Curr. Pharm. Des. 17 (17) (2011) 1695–1708.
- [18] D.J. Sindhikara, N. Yoshida, F. Hirata, J. Comput. Chem. 33 (18) (2012) 1536–1543.
- [19] S. Gusarov, B.S. Pujari, A. Kovalenko, J. Comput. Chem. 33 (17) (2012) 1478–1494.
- [20] M.V. Fedotova, S.E. Kruchinin, J. Mol. Liq. 169 (2012) 1–7.
- [21] M.V. Fedotova, S.E. Kruchinin, J. Mol. Liq. 186 (2013) 90–97.
- [22] M.V. Fedotova, S.E. Kruchinin, Biophys. Chem. 190 (2014) 25–31.
- [23] M.V. Fedotova, O. Dmitrieva, Amino Acids 47 (2015) 1015–1023.
- [24] M.V. Fedotova, O.A. Dmitrieva, New J. Chem. 39 (2015) 8594–8601.
- [25] A. Eiberweiser, A. Nazet, S.E. Kruchinin, M.V. Fedotova, R. Buchner, J. Phys. Chem. B 119 (2015) 15203–15211.
- [26] M.V. Fedotova, O. Dmitrieva, Amino Acids 48 (2016) 1685–1694.
- [27] M.V. Fedotova, S.E. Kruchinin, G.N. Chuev, J. Mol. Liq. 247 (2017) 100–108.
- [28] M.V. Fedotova, S.E. Kruchinin, G.N. Chuev, New J. Chem. 41 (2017) 1219–1228.
- [29] M.V. Fedotova, S.E. Kruchinin, J. Mol. Liq. 244 (2017) 489–498.
- [30] S. Güssregen, H. Matter, G. Hessler, E. Lionta, J. Heil, S.M. Kast, J. Chem. Inf. Model. 57 (7) (2017) 1652–1666.
- [31] O.A. Dmitrieva, M.V. Fedotova, R. Buchner, Phys. Chem. Chem. Phys. 19 (31) (2017) 20474–20483.
- [32] N. Yoshida, J. Chem. Inf. Model. 57 (11) (2017) 2646–2656.
- [33] M.V. Fedotova, J. Mol. Liq. 292 (2019) 111339.
- [34] M.V. Fedotova, S.E. Kruchinin, G.N. Chuev, J. Mol. Liq. 296 (2019) 112081.
- [35] M.V. Fedotova, S.E. Kruchinin, G.N. Chuev, J. Mol. Liq. 313 (2020) 113563.
- [36] M.V. Fedotova, S.E. Kruchinin, G.N. Chuev, J. Mol. Liq. 304 (2020) 112757.
- [37] S. Friesen, M.V. Fedotova, S.E. Kruchinin, R. Buchner, Phys. Chem. Chem. Phys. 23 (2021) 1590–1600.
- [38] S.E. Kruchinin, M.V. Fedotova, J. Phys. Chem. B 125 (40) (2021) 11219–11231.
- [39] B. Roux, J. Phys. Chem. B 124 (34) (2020) 7444–7451, <https://doi.org/10.1021/acs.jpcc.0c05519>.
- [40] G.N. Chuev, M.V. Fedotova, M. Valiev, J. Chem. Phys. 152 (4) (2020) 041101.
- [41] G.N. Chuev, M.V. Fedotova, M. Valiev, J. Stat. Mech. 2021 (3) (2021) 033205.



- [42] G.N. Chuev, M.V. Fedotova, M. Valiev, *J. Chem. Phys.* 155 (6) (2021) 064501.
- [43] D.A. Case, V. Babin, J.T. Berryman, R.M.Q. Betz, M. Cai, D. Cerutti, T.A. Darden, T.E. Cheatham III, R.E. Duke, R.E.H. Gohlke, A. Goetz, S. Gusarov, N. Homeyer, P. Janowski, J. Kaus, I. Kolossvary, A. Kovalenko, T.S. Lee, S. LeGrand, T. Luchko, R. Luo, B. Madej, K. Merz, F. Paesani, D. Roe, A. Roitberg, C.C. Sagui, R. Salomon-Ferrer, G.C. Seabra, C. Simmerling, L.W. Smith, J. Swails, R. Walker, J. Wang, R. Wolf, X. Wu, P. Kollman, 2014.
- [44] G.N. Chuev, M. Valiev, M.V. Fedotova, *J. Chem. Theory Comput.* 8 (4) (2012) 1246–1254, <https://doi.org/10.1021/ct2009297>.
- [45] M. Valiev, E.J. Bylaska, N. Govind, K. Kowalski, T.P. Straatsma, H.J.J. Van Dam, D. Wang, J. Nieplocha, E. Apra, T.L. Windus, W.A. de Jong, *Comput. Phys. Commun.* 181 (9) (2010) 1477–1489, <https://doi.org/10.1016/j.cpc.2010.04.018>.
- [46] A. Kovalenko, S. Ten-no, F. Hirata, *J. Comput. Chem.* 20 (9) (1999) 928–936.
- [47] V. Britanak, P.C. Yip, K.R. Rao, *Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations*, Elsevier, Berlin, 2010.
- [48] G.N. Chuev, I. Vyalov, N. Georgi, *Chem. Phys. Lett.* 561 (562) (2013) 175–178.
- [49] G.N. Chuev, I. Vyalov, N. Georgi, *J. Comput. Chem.* 35 (13) (2014) 1010–1023, <https://doi.org/10.1002/jcc.23586>.
- [50] H.J.C. Berendsen, J.R. Grigera, T.P. Straatsma, *J. Phys. Chem.* 91 (24) (1987) 6269–6271, <https://doi.org/10.1021/j100308a038>.
- [51] K.M. Dyer, J.S. Perkyns, G. Stell, B.M. Pettitt, *Mol. Phys.* 107 (4–6) (2009) 423–431, <https://doi.org/10.1080/00268970902845313>.