Multi-grid Solver for the Reference Interaction Site Model of Molecular Liquids Theory

Volodymyr P. Sergiievskyi, Wolfgang Hackbusch, Maxim V. Fedorov*
Max Planck Institute of Mathematics in the Sciences,
Inselstrasse 22, 04103 Leipzig, Germany

May 15, 2011

Abstract

In this paper we propose a new multi-grid based algorithm for solving integral equations of the Reference Interactions Site Model (RISM). We also investigate the relationship between the parameters of the algorithm and the numerical accuracy of the hydration free energy calculations by RISM. For this purpose we analyzed the performance of the method for several numerical tests with polar and non-polar compounds. The results of this analysis provide some guidelines for choosing an optimal set of parameters to minimize computational expenses. We compared the performance of the proposed multi-grid-based method with the one-grid Picard iteration and nested Picard iteration methods. We show that the proposed method is over 30 times faster than the one-grid iteration method, and in the high accuracy regime it is almost 7 times faster than the nested Picard iteration method.

1 Introduction

Integral equations theory of liquids (IETL) is a powerful method for the description of structural and thermodynamical parameters of liquids [1, 2]. In

*e-mail:fedorov@mis.mpg.de, phone:+49 (0) 341 9959 804, fax:+49 (0) 341 9959 999
terms of computational expenses, it offers a compromise between computationally expensive fully-atomistic simulations [3, 4, 5] and rather approximate continuum electrostatic models [6, 7, 8].

IETL describes the structure of a liquid in terms of the correlation functions. The most important relation in the theory is the Ornstein-Zernike (OZ) equation [9, 1]. For a simple liquid of spherical particles this equation reduces to an one-dimensional integral equation (1D-OZ). However, for molecular liquids the correlation functions depend on the mutual displacements and mutual orientations of the molecules resulting in a six dimensional integral equation - the Molecular Ornstein-Zernike (MOZ) equation [1].

To solve an OZ equation one should complete this by a closure relation which gives additional non-linear relations for the correlation functions, and, therefore, makes the system OZ+closure solvable. However, the closure relation incorporates a so-called bridge function, which is practically incomputable due to the infinite number of terms in the exact representation of this functional[1, 10]. Therefore, in practice one uses approximate closure relations [11, 2].

Nowadays, only few methods exist for solving the 6D MOZ equation. One class of methods is based on the decomposition of the correlation functions into the basis of rotational invariants [12, 13]. Recently an implementation of straightforward method which explicitly treats angular degrees of freedom was reported [14]. Nevertheless, till now the mentioned above methods were applied to the small and simple molecules only. This can be explained by the high computational cost of solution of six-dimensional problem with reasonable accuracy. Therefore, in practice one usually uses simplified models and the Reference Interaction Sites Model (RISM) is the most popular among them. The model was proposed by Chandler and Andersen [15]. The main assumption in the model is that the molecular correlation functions can be represented as a sum of spherically symmetric functions corresponding to the selected parts of the molecule (so-called sites). As a result, the operations with the spherically symmetric functions can be reduced to the operations with only their radial parts and that makes the RISM integral equations effectively one-dimensional. The six-dimensional MOZ equation is replaced by a set of one-dimensional non-linear integral equations. Later on, there has been proposed more sophisticated theories which allow to obtain three-dimensional correlation functions, e.g. 3D-RISM theory [16, 2], or methods based on the solid harmonics decomposition [17, 18]. These methods better describe the structural solute-solvent correlations [19, 20, 21, 22]. For the
sake of simplicity we illustrate our method for the 1D RISM equations. As we describe our method in a general way, an extension of the method to the 3D RISM theory is straightforward. Despite of its relative simplicity 1D RISM theory have several important applications. 1D RISM equations can be self-consistently used to introduce an implicit solvent model in Quantum mechanical calculations (RISM-SCF method) [23, 24, 25, 26]. This method gives much better description of the solvent structure than continuum electrostatics models. Another application of 1D RISM theory is the prediction of thermodynamical properties of solutions, such as the Hydration Free Energy (HFE). The approach is based on end-point relations for HFE [11, 27] and semi-empirical parametrization [28, 29, 30]. It has been recently shown that after proper parametrization of the 1D RISM-based method, this can provide good correspondence with experimental data for HFEs [28, 29, 30].

The goal of the present paper is to introduce a fast solver for RISM equations with emphasis on the efficiency of HFE calculations by RISM.

The simplest scheme for solving equations of the integral equation theory of liquids uses the Picard iteration [31], which, however, may converge rather slowly. One may use faster convergent schemes, such as the Newton-Raphson (NR) iteration [32], NR-GMRES algorithm [33], modified method of direct inversion in iterative subspace (DIIS) [34], combination of modified NR and DIIS iteration [35] or vector extrapolation technique [36]. However, for the grids with a large number of points and/or molecular systems with a large number of interacting sites these methods are computationally expensive. An alternative way to increase efficiency is to use a multi-scale approach. A Commonly used approach is the combined two-level NR-Picard scheme, so-called Gillan method [37]. Similar two-level NR-Picard schemes are used also in the Labík-Malijevský-Voňka method [38, 39] or wavelet-based methods [40, 41, 42, 43]. Although two-level methods give an essential improvement with respect to Picard iteration, the two level approach has limitations, because fine-grid and coarse-grid resolutions cannot differ too much. Recently, an effective multilevel NR-GMRES algorithm has been applied to the problem [44]. This algorithm does not have such restrictions, as the two-level methods; NR-GMRES algorithm is a fast convergent method, and it needs much less iteration steps to converge, than the Picard iteration. However, faster convergence in terms of number of iterations does not necessarily mean better performance in terms of computer time. The Newton-Raphson method requires computation and inversion of Jacobian matrix of size $N \times N$, where $N$ is number of discretization points on the coarse grid.
Although the NR implementation in the work [44] does not need inversion of Jacobian matrix, it requires operations with matrices of size $N \times N$, which demand additional storage and computational time and make each iteration step much more computationally expensive, than a Picard iteration step. For the 1D RISM, where number of coarse-grid points $N$ is not very large these additional computational expanses for each iteration step can be compensated with much less number of iteration steps. However, it is not so for the three dimensional problem, where the number of grid points grows cubically with respect to the one-dimensional problem. Because operations with matrices of size $N^3 \times N^3$ are unfeasible, NR-based methods have only limited applicability.

Despite the fact that many methods use sophisticated algorithms to enhance computational performance, these methods do not fully exploit the advantages of the multi-scale approach. The multi-grid method is a multi-scale technique in the sense that the iteration makes use of different grids (different discretization levels) and is more efficient than simple nested iteration schemes [45]. The multi-grid scheme is not restricted to a specific type of iterations and can be applied to any kind of iteration process. Multi-grid methods are actively used in different applications in computational chemistry [46, 47, 48, 49]. Recently it has been shown that a multi-grid technique incorporating the Picard iteration of 1D OZ equation for simple liquids is able to improve the computational efficiency up to several dozen times [50].

In the current paper we propose an algorithm using the multi-grid scheme for the solution of the RISM equations. In our work we are focused on the practical application of the solver to the Hydration Free Energy calculation. We perform additional investigations to determine some guidelines for choosing algorithm parameters which are optimal for the Hydration Free Energy calculations. We compare the numerical performance of the proposed algorithm with the one-grid Picard iteration and with the nested Picard iteration.

2 Theory

2.1 RISM equations

In the RISM theory, the molecular interaction potential is a superposition of spherically symmetric site-site potentials. The interacting sites are described by their displacement with regards to the centre of the molecule.
Spherically symmetric site-site correlation functions are found by averaging six-dimensional molecular correlation functions over the rotational degrees of freedom. The following functions are involved in the RISM equation for the infinitely diluted solution: (i) total and direct site-site correlation functions \{h_{s\alpha}(r)\} and \{c_{s\alpha}(r)\} describing correlations between the site \(s\) of the solute molecule and sites \(\alpha\) of the solvent molecules, (ii) intramolecular correlation functions \{w_{ss'}(r)\} describing the structure of the solute molecule, and (iii) bulk solvent susceptibility functions \{\chi_{\alpha\alpha'}(r)\} describing the structure of the pure solvent. Assume that the solute molecule has \(M\) sites and the solvent molecule has \(K\) sites. The RISM convolution equation is easier to formulate in the Fourier space:

\[
\hat{H} = \hat{W} \cdot \hat{C} \cdot \hat{X},
\]

where the matrices \(\hat{H}, \hat{C}, \hat{W}, \hat{X}\) are defined as follows: \(\hat{H} = [\hat{h}_{s\alpha}(k)]_{M \times K}\), \(\hat{C} = [\hat{c}_{s\alpha}(k)]_{M \times K}\), \(\hat{W} = [\hat{w}_{ss'}(k)]_{M \times M}\), \(\hat{X} = [\hat{\chi}_{\alpha\alpha'}(k)]_{K \times K}\). Here the hat symbol (\(\hat{\cdot}\)) denotes the Fourier transformed function. The transformation of a spherically symmetric function \(f(r)\) is defined by the Bessel-Fourier transform \(F[\cdot]\):

\[
F[f(r)](k) = \hat{f}(k) = \frac{4\pi}{k} \int_0^\infty f(r) r \sin kr \, dr.
\]

The inverse Bessel-Fourier transform is defined by

\[
F^{-1}[\hat{f}(k)](r) = f(r) = \frac{1}{2\pi r} \int_0^\infty k\hat{f}(k) \sin kr \, dk.
\]

Intramolecular correlation functions in the Fourier space \(\hat{w}_{ss'}(k)\) are found via the relation

\[
\hat{w}_{ss'}(k) = \delta_{ss'} + (1 - \delta_{ss'}) \frac{\sin kr_{ss'}}{Kr_{ss'}},
\]

where \(\delta_{ss'}\) is the Kronecker delta and \(r_{ss'}\) is the distance between the sites \(s\) and \(s'\) of the solute molecule. Susceptibility functions of bulk solvent functions are defined as

\[
\hat{\chi}_{\alpha\alpha'}(k) = \hat{w}_{\alpha\alpha'}^{\text{solv}}(k) + \rho \hat{h}_{\alpha\alpha'}^{\text{solv}}(k),
\]

where \(\rho\) is the density of the solvent and \(\{w^{\text{solv}}_{\alpha\alpha'}(k)\}\) and \(\{h^{\text{solv}}_{\alpha\alpha'}(k)\}\) are intramolecular and total correlation functions of the bulk solvent. In the current work we use previously calculated water susceptibility functions [40],
therefore we do not discuss these calculations here, and just assume them to be known functions in Eq.(1). There are two matrices of unknown functions: \( \hat{H} \) and \( \hat{C} \) in Eq.(1), but only one relation for them. Therefore, Eq.(1) must be completed by the closure relation:

\[
h_{sa}(r) + 1 = \exp (-\beta u_{sa}(r) + h_{sa}(r) - c_{sa}(r) + B_{sa}(r)),
\]

where \( \beta = 1/k_B T \), \( k_B \) is a Boltzmann constant, \( T \) is a temperature, \( u_{sa}(r) \) is the site-site potential and \( B_{sa}(r) \) is the so-called bridge function.

Rigorously speaking, the bridge function is expressed as a series of integrals in growing dimensions [10], and, therefore, practically incomputable. One should find a reasonable approximation for this functional, which is still a challenging problem in the field. The simplest bridge approximation is the Hyper-Netted-Chain (HNC) approximation: \( B_{sa}(r) \equiv 0 \). However, such an approximation can cause uncontrolled growth of the argument of the exponent in Eq.(6) during the numerical solution. To overcome this drawback, it has been proposed to use the so-called Partially-Linearized HNC (PL-HNC) closure which linearizes the HNC expression, when the argument of the exponent is positive [51]:

\[
h_{sa}(r) + 1 = \begin{cases} 
  e^{\Xi_{sa}(r)}, & \Xi_{sa}(r) < 0, \\
  \Xi_{sa}(r), & \Xi_{sa}(r) > 0,
\end{cases}
\]

where \( \Xi_{sa}(r) = -\beta u_{sa}(r) + h_{sa}(r) - c_{sa}(r) \) appears in the argument of the exponential function in Eq.(6).

For the numerical treatment of the problem the site-site potential, \( u_{sa}(r) \) is split into the long-term Coulomb part \( u_{sa}^L(r) \) and short-term part \( u_{sa}^S(r) \). Usually, the Gauss error function \( \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-z^2} dz \) is used for this splitting:

\[
u_{sa}^L(r) = \frac{q_s q_\alpha}{r} \text{erf}(tr), \quad u_{sa}^S(r) = u_{sa}(r) - u_{sa}^L(r),
\]

where \( \frac{q_s q_\alpha}{r} \) is the Coulomb potential, \( q_s, q_\alpha \) are partial charges of the sites \( s \) and \( \alpha \), and the parameter \( t \) controls the smoothness of transition between the long and short range parts (usually \( t = 1 \)). One should note that the Bessel-Fourier transform of the functions \( \{ u_{sa}^L(r) \} \) can be found analytically yielding a rapidly decaying function in the Fourier space [52]:

\[
\hat{u}_{sa}^L(k) = \frac{4\pi q_s q_\alpha}{k^2} e^{-k^2/4t^2}.
\]
The long-term parts of the direct correlation functions \( \{c^{L}_{sa}(r)\} \) are found using the Mean Spherical Approximation [53]:

\[
c^{L}_{sa}(r) = -\beta u^{L}_{sa}(r).
\] (10)

The short range functions \( \{c^{S}_{sa}(r)\} \) are defined as

\[
c^{S}_{sa}(r) = c_{sa}(r) - c^{L}_{sa}(r) = c_{sa}(r) + \beta u^{L}_{sa}(r).
\] (11)

As well, instead of the total correlation functions, it is helpful to use the indirect correlation functions

\[
\gamma_{sa}(r) = h_{sa}(r) - c^{S}_{sa}(r).
\] (12)

In such definitions, the RISM equation is reformulated as

\[
\hat{\Gamma} = \hat{W} \cdot (\hat{C}^{S} - \beta \hat{U}^{L}) \cdot \hat{X} - \hat{C}^{S},
\] (13)

where \( \hat{U}^{L} = [\hat{u}^{L}_{sa}(k)]_{M \times K} \) is a matrix of long-term site-site potentials, \( \hat{\Gamma} = [\hat{\gamma}_{sa}(k)]_{M \times K} \), and \( \hat{C}^{S} = [\hat{c}^{S}_{sa}(k)]_{M \times K} \).

Eq.(13) is completed by the closure relation in the real space:

\[
C^{S} = C[\Gamma] = \left[ e^{-\beta u^{S}_{sa}(r) + \gamma_{sa}(r) + B_{sa}(r)} - \gamma_{sa}(r) - 1 \right]_{M \times K}.
\] (14)

### 3 Method

#### 3.1 Discretization of the problem

In both, real and Fourier space, we discretize the problem on a uniform grid. The grid sizes in the real and Fourier spaces are connected by \( \Delta k = \frac{\pi}{\Delta R} \). We denote the real space grid with \( N \) points and step size \( \Delta R \) as \( \{N, \Delta R\} \). Each grid is also characterized by the cutoff distance, which is the upper limit of the support. For the grid \( \{N, \Delta R\} \), the cutoff distance is \( R_{\text{cutoff}} = N \Delta R \). The corresponding grid in the Fourier space is denoted by \( \{N, \Delta k\} \).

Functions are represented by vectors which contain the values of the functions on the grid points. We denote these grid functions by bold letters and indicate their grid in the superscript:

\[
\gamma^{\{N, \Delta R\}}_{sa} = (\gamma_{sa}(\Delta R), \ldots, \gamma_{sa}(N \Delta R)),
\] (15)
\[
\mathbf{c}_s^{(N \Delta R)} = (c_{s1}^{S}(\Delta R), \ldots, c_{sN}^{S}(N \Delta R)),
\]
\[
\mathbf{u}_s^{(N \Delta R)} = (u_{s1}^{S}(\Delta R), \ldots, u_{sN}^{S}(N \Delta R)).
\]

The grid functions in the Fourier space are indicated by the hat symbol (^): 
\[
\hat{\mathbf{c}}_s^{(N \Delta k)} = (\hat{c}_{s1}^{S}(\Delta k), \ldots, \hat{c}_{sN}^{S}(N \Delta k)),
\]
\[
\hat{\mathbf{w}}_{ss'}^{(N \Delta k)} = (\hat{w}_{ss'}(\Delta k), \ldots, \hat{w}_{ss'}(N \Delta k)),
\]
\[
\hat{\mathbf{\chi}}_{\alpha\beta}^{(N \Delta k)} = (\hat{\chi}_{\alpha\beta}(\Delta k), \ldots, \hat{\chi}_{\alpha\beta}(N \Delta k)),
\]
\[
\hat{\mathbf{u}}_s^{(N \Delta k)} = (\hat{u}_{s1}^L(\Delta k), \ldots, \hat{u}_{sN}^L(N \Delta k)).
\]

Similarly, matrix-valued grid functions carry a subscript denoting the grid. Matrices without a hat sign symbolize real-space functions: \( \mathbf{\Gamma}_{(N \Delta R)} = \begin{bmatrix} \gamma_{s_1}^{(N \Delta R)}(k) \end{bmatrix}_{M \times K} \). We use the hat symbol (^) to denote the matrices of functions in the Fourier space: 
\( \hat{\mathbf{\Gamma}}_{(N \Delta k)} = \begin{bmatrix} \hat{\gamma}_{s_1}^{(N \Delta k)}(k) \end{bmatrix}_{M \times K}, \hat{\mathbf{C}}^{(N \Delta k)} = \begin{bmatrix} \hat{c}_{s_1}^{S(N \Delta k)} \end{bmatrix}_{M \times K}, \hat{\mathbf{W}}_{(N \Delta k)} = \begin{bmatrix} \hat{w}_{ss'}^{(N \Delta k)} \end{bmatrix}_{M \times M}, \hat{\mathbf{U}}_s^{L(N \Delta k)} = \begin{bmatrix} \hat{u}_{s1}^L(N \Delta k) \end{bmatrix}_{M \times K}, \) and 
\( \hat{\mathbf{\chi}}_{(N \Delta k)} = \begin{bmatrix} \hat{\chi}_{\alpha\beta}^{(N \Delta k)} \end{bmatrix}_{K \times K}. \)

To map the grid functions from the real to the Fourier space and back, we use the discrete forward and inverse Bessel-Fourier transformations \( F_{(N \Delta R)}[\cdot] \), \( F^{-1}_{(N \Delta k)}[\cdot] \) respectively:

\[
\hat{\mathbf{f}}_{(N \Delta k)} = F_{(N \Delta R)}[\mathbf{f}_{(N \Delta R)}],
\]
\[
\mathbf{f}_{(N \Delta R)} = F^{-1}_{(N \Delta k)}[\hat{\mathbf{f}}_{(N \Delta k)}].
\]

Vectors \( \mathbf{f}_{(N \Delta R)} \), \( \hat{\mathbf{f}}_{(N \Delta k)} \) are defined as 
\[
\mathbf{f}_{(N \Delta R)} = (f(r_1), \ldots, f(r_N)), \quad r_n = n \Delta R,
\]
\[
\hat{\mathbf{f}}_{(N \Delta k)} = (\hat{f}(k_1), \ldots, \hat{f}(k_N)), \quad k_m = m \Delta k,
\]
and components of these vectors are connected via the relations
\[
\hat{f}(k_m) = \frac{4\pi}{k_m} \sum_{n=1}^{N} f(r_n) r_n \sin\left(\frac{\pi m n}{N}\right) \Delta R,
\]
\[ f(r_m) = \frac{1}{2\pi^2 r_n} \sum_{m=1}^{N} \hat{f}(k_m) k_m \sin\left(\frac{\pi mn}{N}\right) \Delta k. \]  

(28)

Discrete analogues of equation Eq.(13) and the closure relation Eq.(14) are formulated as

\[ \hat{\Gamma}_{\{N,\Delta R\}} = \hat{\mathbf{W}}_{\{N,\Delta R\}} \left( \hat{\mathbf{C}}_{\{N,\Delta R\}}^S - \beta \hat{\mathbf{U}}_{\{N,\Delta R\}}^L \right) \cdot \hat{\mathbf{X}}_{\{N,\Delta R\}} - \hat{\mathbf{C}}_{\{N,\Delta R\}}^S, \]  

(29)

\[ \mathbf{C}_{\{N,\Delta R\}}^S = \mathbf{C}[\Gamma_{\{N,\Delta R\}}] = \left[ e^{-\beta u^{S,sa} + \gamma_{sa} + B_{sa} - \gamma_{sa}} - 1 \right]_{M \times K}, \]  

(30)

where the mathematical operations between vectors are understood entry-wise.

### 3.2 Picard iteration

Combining Eq.(29) and Eq.(30), we can define the iterative operator \( \mathcal{K}_{\{N,\Delta R\}} \):

\[ \mathcal{K}_{\{N,\Delta R\}}[\Gamma_{\{N,\Delta R\}}] = \mathcal{F}_{\{N,\Delta R\}}^{-1} \left[ \hat{\mathbf{W}}_{\{N,\Delta R\}} \left( \mathcal{F}_{\{N,\Delta R\}} \left[ \mathbf{C}[\Gamma_{\{N,\Delta R\}}] \right] - \beta \hat{\mathbf{U}}_{\{N,\Delta R\}}^L \right) \hat{\mathbf{X}}_{\{N,\Delta k\}} - \mathcal{F}_{\{N,\Delta R\}} \left[ \mathbf{C}[\Gamma_{\{N,\Delta R\}}] \right] \right], \]  

(31)

We consider the generalized task

\[ \Gamma_{\{N,\Delta R\}} = \mathcal{K}_{\{N,\Delta R\}}[\Gamma_{\{N,\Delta R\}}] + \mathbf{f}_{\{N,\Delta R\}} \]  

(32)

for a given right-hand side vector \( \mathbf{f}_{\{N,\Delta R\}} \). Problem Eq.(29) - Eq.(30) corresponds to the case \( \mathbf{f}_{\{N,\Delta R\}} = \mathbf{0} \). Necessity of introducing the generalized problem will be described below during the description of the multi-grid method.

The \( n \)-th iterate of an iterative scheme is denoted by \( \Gamma_{\{N,\Delta R\}}^{(n)} \). The damped Picard iteration with the damping parameter \( \lambda \) is defined as

\[ \Gamma_{\{N,\Delta R\}}^{(n+1)} = (1 - \lambda)\Gamma_{\{N,\Delta R\}}^{(n)} + \lambda \Gamma'_{\{N,\Delta R\}}, \]  

(33)

where \( \Gamma'_{\{N,\Delta R\}} \) abbreviates

\[ \Gamma'_{\{N,\Delta R\}} = \mathcal{K}_{\{N,\Delta R\}}[\Gamma_{\{N,\Delta R\}}^{(n)}] + \mathbf{f}_{\{N,\Delta R\}}. \]  

(34)
We use a short notation for this operator:

$$
\Upsilon \{ N, \Delta R \} \Gamma^{(n)}_{\{ N, \Delta R \}}, f_{\{ N, \Delta R \}} \equiv (1 - \lambda) \Gamma^{(n)}_{\{ N, \Delta R \}} + \lambda \Gamma'_{\{ N, \Delta R \}}.
$$

One iteration step of the algorithm consists of the partial steps

$$
\Gamma^{(n)}_{\{ N, \Delta R \}} \xrightarrow{\text{Eq. (30)}} C_S \{ N, \Delta R \} \xrightarrow{\text{IBFT, Eq. (23)}} \Gamma'_{\{ N, \Delta R \}} \xrightarrow{\text{damping, Eq. (33)}} \Gamma^{(n+1)}_{\{ N, \Delta R \}}.
$$

As a measure of accuracy we use the $L_2$ norm between two successive iterates averaged over all site-site functions:

$$
\left\| \Gamma^{(n+1)}_{\{ N, \Delta R \}} - \Gamma^{(n)}_{\{ N, \Delta R \}} \right\| = \frac{1}{MK} \sum_{s \alpha} \sqrt{\sum_{m=1}^{N} \left( \gamma^{(n+1)}_{s \alpha}(m \Delta R) - \gamma^{(n)}_{s \alpha}(m \Delta R) \right)^2 \Delta R}.
$$

We stop the iteration when the iterates differ by less than a given threshold $\varepsilon$:

$$
\left\| \Gamma^{(n+1)}_{\{ N, \Delta R \}} - \Gamma^{(n)}_{\{ N, \Delta R \}} \right\| \leq \varepsilon.
$$

By $n( N, \Delta R, \varepsilon)$ we denote the minimal number $n$ such that Eq. (38) holds. So, using the operator power notation, the iterative process to obtain the solution $\Gamma^\varepsilon_{\{ N, \Delta R \}}$ with accuracy $\varepsilon$ can be written as

$$
\Gamma^\varepsilon_{\{ N, \Delta R \}} = ( \Upsilon_{\{ N, \Delta R \}} )^{n( N, \Delta R, \varepsilon)} [ \Gamma^{(0)}_{\{ N, \Delta R \}}, f_{\{ N, \Delta R \}} ].
$$

### 3.3 Mapping approximated solutions from one grid to another

In the multi-scale methods which we discuss below, several grids are used. Below we define operators, which map grid functions from one grid to another: restriction operator $r[\cdot]$, interpolation operator $p[\cdot]$ and extension operator $e[\cdot]$.

The restriction operator $r[\cdot]$ maps the matrix of grid functions to the coarser grid with doubled grid size and, therefore, half the number of grid points. For example, the restriction maps from the grid $\{ 2N, \Delta R \}$ to the grid $\{ N, 2\Delta R \}$:

$$
r[\Gamma_{\{ 2N, \Delta R \}}] = \Gamma_{\{ N, 2\Delta R \}}.
$$
In the current work we use the trivial injection as a restriction operator. Let 
\( \Gamma_{(2N,\Delta R)} = [\gamma_{sa}^{(2N,\Delta R)}]_{M \times K} \), where

\[
\gamma_{sa}^{(2N,\Delta R)} = (\gamma_{sa}(\Delta R), \gamma_{sa}(2\Delta R), \ldots, \gamma_{sa}(2N\Delta R))
\]  
(41)

and \( \Gamma_{(N,2\Delta R)} = [\gamma_{sa}^{(N,2\Delta R)}]_{M \times K} = r[\Gamma_{(2N,\Delta R)}] \). Then the vectors \( \gamma_{sa}^{(N,2\Delta R)} \) are defined by

\[
\gamma_{sa}^{(N,2\Delta R)} = (\gamma_{sa}(2\Delta R), \gamma_{sa}(4\Delta R), \ldots, \gamma_{sa}(2N\Delta R)).
\]  
(42)

We should mention that the restriction operator \( r[\cdot] \) is linear:

\[
r[a\Gamma'_{(2N,\Delta R)} + b\Gamma''_{(2N,\Delta R)}] = a \cdot r[\Gamma'_{(2N,\Delta R)}] + b \cdot r[\Gamma''_{(2N,\Delta R)}],
\]  
(43)

The interpolation operator \( p[\cdot] \) maps the matrix of grid functions to the grid with half the grid size and, thereby, the double number of grid points. For example, the interpolation maps from the grid \( \{N,\Delta R\} \) to the grid \( \{2N,\Delta R\} \):

\[
p[\Gamma_{(N,\Delta R)}] = \tilde{\Gamma}_{(2N,\Delta R)} = [\tilde{\gamma}_{sa}^{(2N,\Delta R)}].
\]  
(44)

In the current work we use cubic spline interpolation as interpolation operator. Let \( \Gamma_{(N,2\Delta R)} = [\gamma_{sa}^{(N,2\Delta R)}]_{M \times K} \), where vectors \( \{\gamma_{sa}^{(N,2\Delta R)}\} \) are defined by Eq.(42). Then \( \tilde{\gamma}_{sa}^{(2N,\Delta R)} \) in Eq.(44) are defined by

\[
\tilde{\gamma}_{sa}^{(2N,\Delta R)} = (\tilde{\gamma}_{sa}(\Delta R), \gamma_{sa}(2\Delta R), \tilde{\gamma}_{sa}(3\Delta R), \ldots, \tilde{\gamma}_{sa}((2N - 1)\Delta R), \gamma_{sa}(2N\Delta R)),
\]  
(45)

where the values \( \tilde{\gamma}_{sa}((2k - 1)\Delta R) \) are obtained from the values \( \gamma_{sa}(2k\Delta R) \) using the cubic spline interpolation.

We note that \( r[\cdot] \) is the left-inverse of \( p[\cdot] \), i.e., \( r[p[\Gamma_{(N,\Delta R)}]] = \Gamma_{(N,\Delta R)} \), but not the right-inverse:

\[
\tilde{\Gamma}_{(2N,\Delta R)} = p[r[\Gamma_{(2N,\Delta R)}]] \neq \Gamma_{(2N,\Delta R)}.
\]  
(46)

However, sufficiently smooth functions satisfy

\[
\tilde{\Gamma}_{(2N,\Delta R)} = p[r[\Gamma_{(2N,\Delta R)}]] = \Gamma_{(2N,\Delta R)} + O(\Delta R).
\]  
(47)

The extension operator \( e[\cdot] \) does not change the grid size, but doubles the number of points. For example, the extension maps from the grid \( \{N,\Delta R\} \) to the grid \( \{2N,\Delta R\} \):

\[
e[\Gamma_{(N,\Delta R)}] = \Gamma_{(2N,\Delta R)} = [\gamma_{sa}^{(2N,\Delta R)}]_{M \times K}
\]  
(48)
Indirect correlation functions decay fast to zero as the distance increases. Thus, it is natural to extend them by zeros yielding the zero extension operator which we use in the current work. Let $\Gamma_{\{N,\Delta R\}} = [\gamma_{so}^{\{N,\Delta R\}}]_{M \times K}$ with vectors $\{\gamma_{so}^{\{N,\Delta R\}}\}$ defined by Eq.(15). Then the functions $\{\gamma_{so}^{\{2N,\Delta R\}}\}$ in Eq.(48) are defined by

$$\gamma_{so}^{\{2N,\Delta R\}} = (\gamma_{so}(\Delta R), \ldots, \gamma_{so}(N\Delta R), 0, \ldots, 0).$$  \hspace{1cm} (49)

### 3.4 Nested Picard iteration

Independently of the convergence rate of an iterative scheme, the error of the $n$-th iterate is the smaller the better the initial guess is. Having at hand different grids, the idea of the nested Picard iteration [45] is straightforward: use as initial guess the (approximate) solution from the coarse grid with a smaller number of grid points. Here we exploit that computations in the coarse grid are cheaper. Below we describe the scheme of the nested Picard iteration. Consider two grids: the “coarse” grid $\{N, 2\Delta R\}$ and the “fine” grid $\{2N, \Delta R\}$. We start from the coarse-grid solution $\Gamma_{\{0\}}^{\{N,2\Delta R\}}$. We perform an iteration process of type Eq.(39) to obtain a solution with accuracy $\varepsilon$ on the coarse grid, interpolate it to the fine grid and use it as the initial approximation for the fine-grid iteration.

The scheme for performing the two-grid nested Picard iteration is written as follows:

$$\Gamma_{\{0\}}^{\{N,2\Delta R\}} \xrightarrow{\Theta_{\{N,2\Delta R\}}} \Gamma_{\{1\}}^{\{N,2\Delta R\}} \xrightarrow{p} \Gamma_{\{2\}}^{\{N,2\Delta R\}} \xrightarrow{\Theta_{\{2N,\Delta R\}}} \Gamma_{\{3\}}^{\{2N,\Delta R\}}.$$

The nested Picard iteration scheme for more than two grids $\{N, 2^L\Delta R\}$, $\{2N, 2^{L-1}\Delta R\}$, $\ldots$, $\{2^LN, \Delta R\}$ with the same cutoff distance $R_{cutoff} = 2^LN\Delta R$ is

$$\Gamma_{\{0\}}^{\{N,2^L\Delta R\}} \xrightarrow{\Theta_{\{N,2^L\Delta R\}}} \Gamma_{\{1\}}^{\{N,2^L\Delta R\}} \xrightarrow{p} \Gamma_{\{2\}}^{\{N,2^{L-1}\Delta R\}} \xrightarrow{\Theta_{\{2N,2^{L-1}\Delta R\}}} \ldots$$

To obtain the solution on a grid with larger cutoff distance, we may continue
the process in a similar way using the extension operator $e[\cdot]$:

$$
\Gamma^{(0)}_{(2^{L+1}N, \Delta R)} = e[\Gamma^e_{(2^LN, \Delta R)}], \\
\Gamma^e_{(2^{L+1}N, \Delta R)} = \left(\Gamma_{(2^{L+1}N, \Delta R)}^e\right)_{n(2^{L+1}N, \Delta R)} [\Gamma^{(0)}_{(2^LNN, \Delta R)}].
$$

(52)

The same process can be defined for multiple grids $\{2^LN, \Delta R\}, \{2^{L+1}N, \Delta R\}, \ldots, \{2^{L+P}N, \Delta R\}$:

$$
\Gamma^e_{(2^LN, \Delta R)} \xrightarrow{e} \Gamma^{(0)}_{(2^{L+1}N, \Delta R)} \xrightarrow{\Gamma^e_{(2^{L+1}N, \Delta R)}} \Gamma^{e}_{(2^{L+1}N, \Delta R)} \xrightarrow{e} \cdots \xrightarrow{e} \Gamma^{(0)}_{(2^{L+P}N, \Delta R)} \xrightarrow{\Gamma^e_{(2^{L+P}N, \Delta R)}} \Gamma^{e}_{(2^{L+P}N, \Delta R)}.
$$

(53)

### 3.5 Two- and multi-grid iteration

Although the nested Picard iteration scheme is able to essentially enhance the performance of numerical iteration, there is a drawback which limits its efficiency. Consider two grids $\{N, 2\Delta R\}$ and $\{2N, \Delta R\}$, to which we refer below as coarse and fine grid, respectively. Denote the exact solutions $\Gamma^*_{\text{coarse}}$, $\Gamma^*_{\text{fine}}$ on the respective coarse and fine grids by

$$
\Gamma^*_{\text{coarse}} = K_{\text{coarse}}[\Gamma^*_{\text{coarse}}] + r_f[\Gamma^*_{\text{fine}}], \\
\Gamma^*_{\text{fine}} = K_{\text{fine}}[\Gamma^*_{\text{fine}}] + f_{\text{fine}}.
$$

(54)

We note, that the restricted fine grid solution is **not** the coarse grid solution:

$$
\Gamma^*_{\text{coarse}} \neq r_f[\Gamma^*_{\text{fine}}].
$$

(55)

Due to this fact, the coarse-grid iteration is not able to give a very good approximation of the fine-grid solution, which limits the performance of the nested iterative schemes. The multi-grid scheme is able to overcome this limitation. For the complete description of the different multi-grid schemes we refer to the book [45]. Below we briefly describe the multi-grid-based algorithm for solving the RISM equation which is used in the current work.

The following grid difference operator $G[\cdot]$ applies to fine-grid functions and indicates the difference of $K_{\text{fine}}$ and $K_{\text{coarse}}$:

$$
G[\Gamma_{\text{fine}}] = r [K_{\text{fine}}[\Gamma_{\text{fine}}]] - K_{\text{coarse}}[r_f[\Gamma_{\text{fine}}]].
$$

(56)

One may say, that the fact that $G[\Gamma_{\text{fine}}] \neq 0$ means that the restriction and iteration operators do not commute. However, this is not a strict statement,
because formally there is not only one iterative operator, but there exist several ones (coarse-grid and fine-grid operators are in principle different). Thus, although the conception of commutativity can help to understand the multi-grid method, we will avoid this term below.

Let us consider the task

\[ \Gamma_{\text{coarse}} = \mathcal{K}_{\text{coarse}}[\Gamma_{\text{coarse}}] + r[\Gamma^*_\text{fine}] + G[\Gamma_{\ast}\text{fine}] \]  

(57)

Substituting here the restricted exact fine grid solution \( \Gamma_{\text{coarse}} = r[\Gamma^*_\text{fine}] \) and using the definition Eq.(56), we have:

\[ r[\Gamma^*_\text{fine}] = r[\mathcal{K}_{\text{fine}}[\Gamma^*_\text{fine}]] + r[\Gamma^*_\text{fine}] \]  

(58)

This equality holds due to the linearity of the restriction operator Eq.(43) and second equality in Eq.(54). One can see, that the task Eq.(57) is of the form of Eq.(32) where \( \mathbf{f}_{\text{coarse}} = r[\Gamma^*_\text{fine}] + G[\Gamma^*_\text{fine}] \). This shows the role of the vector \( \mathbf{f}_{\text{coarse}} \): it accumulates the grid differences between the finer and coarser grids during the multi-grid iteration.

The task Eq.(57) can be used to find the solution \( r[\Gamma^*_\text{fine}] \). We do not know the exact difference \( G[\Gamma^*_\text{fine}] \). However, even if the approximate solution is far from the exact solution, the value \( G[\Gamma_{\ast}\text{fine}] \) can be accurate enough:

\[ \left\| G[\Gamma_{\ast}\text{fine}] - G[\Gamma^*_\text{fine}] \right\| \ll \left\| \Gamma^*_\text{fine} - \Gamma_{\ast}\text{fine} \right\| \]  

(59)

In the two-grid scheme one performs a small number \( \nu_1 \) of fine-grid iteration steps before solving the coarse grid task and uses the coarse-grid solution to eliminate the low-frequency errors of fine-grid iterate. However, for some operators the interpolation of the coarse-grid solution may be not smooth enough. That is why one may need to perform some additional number \( \nu_2 \) of so-called smoothing fine-grid iteration steps. Let \( \Gamma^{(n)}_{\text{fine}} \) be the fine-grid approximation on the \( n \)-th step of two-grid iteration. The two-grid iteration process can be written in a following way:

\[ \Gamma^{(n+1)}_{\text{fine}} = \mathcal{T}[\Gamma^{(n)}_{\text{fine}}, \mathbf{f}_{\text{fine}}] \]  

(60)

where the two-grid operator \( \mathcal{T}[\cdot, \cdot] \) is defined by the following algorithm:

**Input:** \( \Gamma^{(n)}_{\text{fine}}, \mathbf{f}_{\text{fine}} \)  
**Output:** \( \Gamma^{(n+1)}_{\text{fine}} \)
1. Perform $\nu_1$ fine-grid iteration steps:

$$\Gamma'_{\text{fine}} = (\Upsilon_{\text{fine}})^{\nu_1} [\Gamma^{(n)}_{\text{fine}}, f_{\text{fine}}].$$

2. Define a coarse-grid analogue of $\Gamma'_{\text{fine}}$:

$$\Gamma^{(0)}_{\text{coarse}} = r[\Gamma'_{\text{fine}}].$$

3. Calculate the grid correction $G[\Gamma'_{\text{fine}}]$:

$$G[\Gamma'_{\text{fine}}] = r[K_{\text{fine}}[\Gamma'_{\text{fine}}]] - K_{\text{coarse}}[\Gamma^{(0)}_{\text{coarse}}].$$

4. Determine the solution $\Gamma^*_{\text{coarse}}$ the coarse grid problem

$$\Gamma_{\text{coarse}} = K_{\text{coarse}}[\Gamma_{\text{coarse}}] + G[\Gamma'_{\text{fine}}] + r[f_{\text{fine}}]. \quad (61)$$

5. Add the coarse-grid correction:

$$\Gamma''_{\text{fine}} = \Gamma'_{\text{fine}} + p[\Gamma^*_{\text{coarse}} - \Gamma^{(0)}_{\text{coarse}}].$$

6. Perform $\nu_2$ smoothing fine-grid iteration steps:

$$\Gamma^{(n+1)}_{\text{fine}} = (\Upsilon_{\text{fine}})^{\nu_2} [\Gamma''_{\text{fine}}, f_{\text{fine}}].$$

In the two-grid algorithm it is not specified how the coarse-grid equation on the step 4 is solved. If the same algorithm is used recursively for solving the coarse-grid problem, we obtain the multi-grid iterative scheme [45]. Assume that we have the grids $\{N, 2^L \Delta R\}, \{2N, 2^{L-1} \Delta R\}, \ldots, \{2^L N, \Delta R\}$. We will use the subscript $grid$ to refer to any of these grids. On each grid we define the multi-grid iteration with iterative operator $M_{\text{grid}}^{\text{level}}[\Gamma^{(n)}_{\text{grid}}, f_{\text{grid}}]$:

$$\Gamma^{(n+1)}_{\text{grid}} = M_{\text{grid}}^{\text{level}}[\Gamma^{(n)}_{\text{grid}}, f_{\text{grid}}], \quad (62)$$

where $\Gamma^{(n)}_{\text{grid}}$ is the $n$-th multi-grid iterate, $f_{\text{grid}}$ is given, and the superscript $\text{level}$ indicates the number of recursions which are done while calculating the operator. The multi-grid iteration converges to the solution of the task

$$\Gamma_{\text{grid}} = K_{\text{grid}}[\Gamma_{\text{grid}}] + f_{\text{grid}}. \quad (63)$$
The multi-grid operator at level zero is a single-grid solver of Eq. (63) on the coarsest grid. In our work we use $n$ steps of the damped Picard iteration:

$$
\mathcal{M}^0_{\{N,2^\ell \Delta R\}} \left[ \Gamma_{\{N,2^\ell \Delta R\}}^{(0)}, f_{\{N,2^\ell \Delta R\}} \right] = \left( \Upsilon_{\{N,2^\ell \Delta R\}} \right)^n \left[ \Gamma_{\{N,2^\ell \Delta R\}}^{(0)}, f_{\{N,2^\ell \Delta R\}} \right]
$$

(64)

The proper choice of the number $n$ of iteration steps on the coarsest grid is discussed in Subsection 3.7. For the sake of brevity, below we use the subscript “fine” to refer to the grid $\{2^\ell N, 2^{L-\ell} \Delta R\}$ and the subscript “coarse” to refer to the grid $\{2^{\ell-1} N, 2^{L-\ell+1} \Delta R\}$. The multi-grid operator $\mathcal{M}_{\text{fine}}^{(n)}[\Gamma_{\text{fine}}, f_{\text{fine}}]$ of level $\ell > 0$ is defined by the following algorithm:

**Input:** $\Gamma_{\text{fine}}^{(n)}, f_{\text{fine}}, \ell$

**Output:** $\Gamma_{\text{fine}}^{(n+1)}$

1. Perform $\nu_1$ steps of the fine-grid Picard iteration:

$$
\Gamma'_{\text{fine}} = \left( \Upsilon_{\text{fine}} \right)^{\nu_1} \left[ \Gamma_{\text{fine}}^{(n)}, f_{\text{fine}} \right].
$$

2. Define a coarse-grid analogue of $\Gamma'_{\text{fine}}$ and use it as starting guess of the iteration in Step 4:

$$
\Gamma_{\text{coarse}}^{(0)} = r[\Gamma'_{\text{fine}}].
$$

3. Calculate grid correction $G[\Gamma'_{\text{fine}}]$:

$$
G[\Gamma'_{\text{fine}}] = r[\mathcal{K}_{\text{fine}}[\Gamma'_{\text{fine}}]] - \mathcal{K}_{\text{coarse}}[\Gamma_{\text{coarse}}^{(0)}].
$$

4. Perform, recursively, $\mu$ steps of the coarse-grid multi-grid iteration of level $(\ell - 1)$:

$$
\Gamma_{\text{coarse}}^{(\mu)} = \left( \mathcal{M}_{\text{coarse}}^{\ell-1} \right)^\mu \left[ \Gamma_{\text{coarse}}^{(0)}, r[f_{\text{fine}}] + G[\Gamma'_{\text{fine}}] \right].
$$

5. Add the coarse-grid correction:

$$
\Gamma_{\text{fine}}'' = \Gamma'_{\text{fine}} + p[\Gamma_{\text{coarse}}^{(\mu)} - \Gamma_{\text{coarse}}^{(0)}].
$$

6. Perform $\nu_2$ steps of the fine-grid Picard iteration:

$$
\Gamma_{\text{fine}}^{(n+1)} = \left( \Upsilon_{\text{fine}} \right)^{\nu_2} \left[ \Gamma_{\text{fine}}'', f_{\text{fine}} \right].
$$
If in Step 4 the number of the multi-grid iteration steps is $\mu = 1$, the multi-grid iteration is called $V$-cycle. If $\mu = 2$, the iteration is called $W$-cycle \cite{45}. In the current work we use $\mu = 1$. In our case the iterative operator $K[\cdot]$ is smooth enough, thus for the multi-grid iteration we use $\nu_1 = 1$ and $\nu_2 = 0$ on the steps 1, 6 of the algorithm, which is standard for the multi-grid of the second kind \cite{45}.

Now we assume that grids with different cutoff distances are given: $\{2^L N, \Delta R\}$, $\{2^{L+1} N, \Delta R\}$, $\ldots$, $\{2^{L+P} N, \Delta R\}$. We can use a scheme similar to Eq.(53) for the multi-grid iteration: having solved the task on the grid $\{2^L N, \Delta R\}$ by the multi-grid iteration up to accuracy $\varepsilon$, extend the solution to the grid $\{2^{L+1} N, \Delta R\}$ and use it as initial guess for a next multi-grid iteration and so on. We denote by $\Gamma^{(0)}_{grid}$ the initial approximation on the grid $grid$ and by $\Gamma^{(\varepsilon)}_{grid}$ the solution with the $L_2$-norm accuracy $\varepsilon$, obtained via the iterative process Eq.(62). The multi-grid iteration with the grid extension can be written schematically as follows:

$$
\begin{align*}
\Gamma^{(0)}_{\{2^L N, \Delta R\}} & \xrightarrow{\mathcal{M}^I_{\{2^L N, \Delta R\}}} \Gamma^{(\varepsilon)}_{\{2^L N, \Delta R\}} \xrightarrow{\varepsilon} \\
\Gamma^{(0)}_{\{2^{L+1} N, \Delta R\}} & \xrightarrow{\mathcal{M}^I_{\{2^{L+1} N, \Delta R\}}} \ldots \\
\varepsilon & \xrightarrow{\Gamma^{(0)}_{\{2^{L+P} N, \Delta R\}}} \\
\mathcal{M}^I_{\{2^{L+P} N, \Delta R\}} & \xrightarrow{\varepsilon} \Gamma^{(\varepsilon)}_{\{2^{L+P} N, \Delta R\}}.
\end{align*}
$$

\subsection{3.6 Nested multi-grid}

In any iteration the error of the $n$-th iterate is the smaller, the better the initial guess is. The nested iteration technique suggests to use an approximate coarse-grid solution as initial guess for the fine-grid iteration. In the multi-grid iteration the fine-grid initial guess can be found via the multi-grid iteration on the coarser grid. Typically, it is enough to perform a single multi-grid iteration on the coarser grid to obtain a good initial guess for the fine-grid iteration. Assume grids $\{N, 2^L \Delta R\}$, $\{2N, 2^{L-1} \Delta R\}$, $\ldots$, $\{2^L N, \Delta R\}$ and an initial guess $\Gamma^{(0)}_{\{N, 2^L \Delta R\}}$ on the coarsest grid are given. We find an initial guess $\Gamma^{(0)}_{\{2^L N, \Delta R\}}$ on the fine grid $\{2^L N, \Delta R\}$ by the following algorithm:

\begin{itemize}
\item Input: $\Gamma^{(0)}_{\{N, 2^L \Delta R\}}$
\item Output: $\Gamma^{(0)}_{\{2^L N, \Delta R\}}$
\item for $\ell=0\ldots L-1$:
\end{itemize}
1. Perform one multi-grid iteration on the coarse grid:

\[ \Gamma^{(4)}_{\{2^\ell N,2^\ell-1 \Delta R\}} = M^\ell_{\{2^\ell N,2^\ell-1 \Delta R\}}(\Gamma^{(0)}_{\{2^\ell N,2^\ell-1 \Delta R\}}) \]

2. Interpolate the result to the finer grid:

\[ \Gamma^{(0)}_{\{2^{\ell+1} N,2^{\ell-1} \Delta R\}} = p[\Gamma^{(1)}_{\{2^\ell N,2^\ell-1 \Delta R\}}] \]

The same can be written schematically:

\[ \Gamma^{(0)}_{\{N,2\Delta R\}} \xrightarrow{M^0_{\{N,2\Delta R\}}} \Gamma^{(1)}_{\{N,2\Delta R\}} \xrightarrow{p} \Gamma^{(0)}_{\{2N,2\Delta R\}} \xrightarrow{M^1_{\{2N,2\Delta-1\Delta R\}}} \cdots \xrightarrow{M^{L-1}_{\{2^{L-1} N,2\Delta R\}}} \Gamma^{(1)}_{\{2^{L-1} N,2\Delta R\}} \xrightarrow{p} \Gamma^{(0)}_{\{2^L N,\Delta R\}} \]

Having the initial guess \( \Gamma^{(0)}_{\{2^L N,\Delta R\}} \) one can use the multi-grid iteration process Eq.(62) to obtain the approximate solution on the grid \( \{2^L N, \Delta R\} \) with a given accuracy \( \varepsilon \). After that, using scheme Eq.(65), one may extend the solution to the grid \( \{2^{L+P} N, \Delta R\} \). In the current paper we call the process Eq.(66) - Eq.(62) - Eq.(65) the nested multi-grid iteration, and compare its performance to the multi-grid iteration Eq.(62) - Eq.(65), to the nested Picard iteration Eq.(51) - Eq.(53), and to the one-level Picard iteration Eq.(36).

### 3.7 Determining the optimal number of coarse-grid iteration steps

In the multi-grid algorithm on the coarsest grid we solve the task of type Eq.(61) with correction \( G[\Gamma^{\prime}_{\text{fine}}] \). Because \( G[\Gamma^{\prime}_{\text{fine}}] \) is only an approximation of the \( G[\Gamma^{\prime}_{\text{fine}}] \) there is no need to solve this task with accuracy better than the accuracy \( \varepsilon_{G[\Gamma^{\prime}_{\text{fine}}]} \) of calculation of \( G[\Gamma^{\prime}_{\text{fine}}] \), which is defined as follows:

\[ \varepsilon_{G[\Gamma^{\prime}_{\text{fine}}]} = \| G[\Gamma^{\prime}_{\text{fine}}] - G[\Gamma^{*}_{\text{fine}}] \|. \quad (67) \]

The value of \( \varepsilon_{G[\Gamma^{\prime}_{\text{fine}}]} \) can be estimated using the expression

\[ \varepsilon_{G[\Gamma^{\prime}_{\text{fine}}]} \approx \| G[\Gamma^{(n+1)}_{\text{fine}}] - G[\Gamma^{(n+1)}_{\text{fine}}] \|. \quad (68) \]
Let us assume that the error \( \varepsilon(n) \) of the solution decays exponentially with the number \( n \) of the coarse-grid iteration steps:

\[
\varepsilon(n) = \| \Gamma^{(n)}_{\text{coarse}} - \Gamma^*_{\text{coarse}} \| = \varepsilon(0) \cdot \delta^n
\]  

(69)

The value of \( \delta \) can be found to be

\[
\delta = \frac{\varepsilon(1)}{\varepsilon(0)}
\]

(70)

We may estimate \( \varepsilon(1) \) and \( \varepsilon(0) \) by the expressions

\[
\varepsilon(1) \approx \| \Gamma^{(1)}_{\text{coarse}} - \Gamma^{(n)}_{\text{coarse}} \|, \\
\varepsilon(0) \approx \| \Gamma^{(0)}_{\text{coarse}} - \Gamma^{(n)}_{\text{coarse}} \|.
\]

(71)

For the optimal number \( n_{\text{opt}} \) of iteration steps we obtain

\[
\varepsilon(n_{\text{opt}}) = \varepsilon(0) \cdot \delta^{n_{\text{opt}}} = \varepsilon_G[\Gamma_{\text{fine}}]
\]

(72)

Let the actual number of the iteration steps be \( n \).

Dividing Eq.(69) by Eq.(72) we get

\[
\frac{\varepsilon(n)}{\varepsilon_G[\Gamma]} = \frac{\varepsilon(0)\delta^n}{\varepsilon_G[\Gamma]} = \delta^{n-n_{\text{opt}}}
\]

(73)

and find the optimal number of iteration steps as

\[
n_{\text{opt}} = \log_\delta \frac{\varepsilon_G[\Gamma]}{\varepsilon(0)}
\]

(74)

where \( \delta, \varepsilon_G[\Gamma], \varepsilon(0) \) are estimated through Eq.(70), Eq.(68), Eq.(71) , respectively. So after each multi-grid iteration step we may estimate the optimal number of iteration steps on the coarsest grid and use this number in the next multi-grid iteration step. To avoid fast change of \( n \) from one multi-grid iteration step to another, we start from some number \( n^{(0)} \) of coarse-grid iteration steps and use a damped iteration process for the number \( n^{(k)} \) of the coarsest-grid iteration steps on the \( k \)-th multi-grid iteration step:

\[
n^{(k+1)} = \alpha n^{(k)} + (1 - \alpha)n_{\text{opt}}
\]

(75)

where \( 0 < \alpha < 1 \) is the damping parameter.
3.8 Choice of optimal grid parameters for Hydration Free Energy calculations

In the sections above we explained how to solve the RISM equations Eq.(29) - Eq.(30) numerically. During the numerical solution we perform iteration steps on several grids with different grid sizes and cutoff distances. In principle, we are free to choose the parameters of the iterations. However, we plan to apply the RISM for calculations of the Hydration Free Energy (HFE). Thus, we would like to choose parameters which yield HFE values with given numerical accuracy at minimal computational cost. HFE can be calculated using the Kirkwood’s thermodynamic integration formula [54]. In the RISM approximation, the HFE of a molecule is found as the sum of the partial HFEs of the sites. In the scope of the HNC approximation, the thermodynamic integration can be performed analytically and HFE ($\Delta G$) may be found explicitly from the solutions of the RISM equation [11]:

$$\Delta G_{HNC} = \frac{2}{\pi \rho k T} \sum_{s \alpha} \int_{0}^{\infty} \left[ -2 c_{sa}(r) + \gamma_{sa}(r) (c_{sa}(r) + \gamma_{sa}(r)) \right] r^2 dr$$

where $\rho$ is the bulk number density, $k$ is the Boltzmann constant and $T$ is the temperature.

Usually, HFE is measured in kilo-calories per mole (kcal/mol). The accuracy of experimental HFE measurements for bioactive compounds is $\gtrsim 0.1$ kcal/mol [55]. To make some theoretical and statistical investigations, we typically need to obtain a computational accuracy of about 100 times higher than the experimental one. That means that we should choose the grid parameters which allow us to calculate the expression Eq.(76) with accuracy of at least 0.001 kcal/mol. We use a uniform grid which can be described by the grid size $\Delta R$ and cutoff distance $R_{cutoff}$. First, we try to determine an appropriate grid size $\Delta R$. We perform series of RISM calculations with same cutoff distance and different grid sizes, and for each grid size we calculate the free energy of solvation using Eq.(76). We assume that the solution on the finest grid yields an almost exact value of $\Delta G_{HNC}$. Let us denote by $\Delta G_{HNC}^{\Delta R}$ the HFE-value calculated on the grid with step $\Delta R$, and by $\Delta G_{HNC}^{best}$ the value of the HFE calculated on the finest grid. We can estimate the error of the HFE calculations as difference of $\Delta G_{HNC}^{\Delta R}$ and the best value $\Delta G_{HNC}^{best}$:

$$\text{Error}(\Delta G_{HNC}^{\Delta R}) = |\Delta G_{HNC}^{\Delta R} - \Delta G_{HNC}^{best}|$$

(77)
In our calculations, we measure distances in atomic units (Bohr, 1 Bohr \( \approx 0.52 \, \text{Å} \)) as they are the most natural for the atomic scale. As the base grid size we choose \( \Delta R_0 = 0.1 \, \text{Bohr} \), and then make the solvation free energy calculations for the different grid sizes \( \Delta R = \frac{\Delta R_0}{2^k}, k = -3 \ldots 6 \) (grid sizes from 0.8 Bohr to \( \frac{1}{640} \) Bohr). The value of \( \frac{1}{640} \) Bohr is taken to be an approximation \( \Delta G_{HNC}^{\text{best}} \) of the exact HFE-value. The cutoff distance is 204.8 Bohr. Calculations on all grids are performed up to \( L_2 \)-norm accuracy \( \varepsilon = 10^{-10} \).

To find the optimal cutoff distance \( R_{\text{cutoff}} \), we perform calculations with the fixed \( \Delta R \) but different \( R_{\text{cutoff}} \). We estimate the error of HFE calculations by taking the integral in Eq.(76) over the interval \( (R_{\text{cutoff}}, \infty) \), because this is the part of the axis which we omit while using the function with finite support:

\[
\text{Error}(\Delta G_{HNC}) = 2\pi\rho kT \sum_{R_{\text{cutoff}}} \int_{R_{\text{cutoff}}}^{\infty} \left[ -2c_{sa}(r) + \gamma_{sa}(r) (c_{sa}(r) + \gamma_{sa}(r)) \right] r^2 dr \quad (78)
\]

To evaluate the infinite integral Eq.(78), we can calculate functions \( \gamma_{sa}(r) \) and \( c_{sa}(r) \) on the grid with a very large cutoff distance \( R_{\text{cutoff}}^{\infty} \) and calculate the integral over the interval \( (R_{\text{cutoff}}, R_{\text{cutoff}}^{\infty}) \). As large cutoff distance we use \( R_{\text{cutoff}}^{\infty} = 409.6 \, \text{Bohr} \).

The most of computational time in the multi-grid iteration is spent on the coarsest grid. The coarsest-grid solution should give a good approximation to the low-frequent part of the exact solution. That means that using the coarsest-grid solution, we as well should be able to roughly approximate chemical properties of the system, in particular the free energy of hydration. HFE-values for a wide class of compounds lie in the range from -5 kcal/mol to +5 kcal/mol. That means that to obtain some qualitative information about the value of HFE we need to have an accuracy in the calculations of at least 1-2 kcal/mol. In the current work grid size and cutoff distance of the coarsest grid are chosen to give the numerical error in HFE calculations \( \leq 1 \) kcal/mol.

It is known that the solution of the RISM equations behave differently for neutral and charged systems. That is why for determining the optimal grid parameters we have chosen five different systems: single uncharged atom (Argon), simple charged ion pair (Sodium chloride), uncharged molecule (methane), polar molecule with high partial charges of atoms (methanol)
Figure 1: Dependencies of the error in hydration free energy calculations on the grid step $\Delta R$ (logarithmic scale). Calculations are done for the cutoff distance $R_{\text{cutoff}}=204.8$ Bohr.

and water.

We can find how much faster are nested Picard iteration, multi-grid and nested multi-grid than the one-grid Picard iteration. To do the comparison we use the speed-up factors:

\begin{align}
S_{\text{NMG}}(\varepsilon) &= \frac{t_{\text{one-grid}}(\varepsilon)}{t_{\text{NMG}}(\varepsilon)} \\
S_{\text{MG}}(\varepsilon) &= \frac{t_{\text{one-grid}}(\varepsilon)}{t_{\text{MG}}(\varepsilon)} \\
S_{\text{Nest}}(\varepsilon) &= \frac{t_{\text{one-grid}}(\varepsilon)}{t_{\text{NP}}(\varepsilon)}
\end{align}

where $t_{\text{NMG}}(\varepsilon)$, $t_{\text{MG}}(\varepsilon)$, $t_{\text{NP}}(\varepsilon)$, and $t_{\text{one-grid}}(\varepsilon)$ are the computer times to solve the RISM equations by the nested multi-grid method, multi-grid method, nested Picard iteration method and one-level Picard iteration method up to the $L_2$-norm accuracy $\varepsilon$ respectively. The values $t_{\text{NP}}(\varepsilon)/t_{\text{MG}}(\varepsilon)$ and $t_{\text{NP}}(\varepsilon)/t_{\text{NMG}}(\varepsilon)$ show how many times faster are the multi-grid and nested multi-grid methods than the nested Picard iteration respectively.
4 Results

4.1 Optimal grid parameters

To determine the appropriate grid parameters, the RISM Hydration Free Energy calculations were performed for grids with different fine-grid sizes and different cutoff distances. One can find details of calculation in the supporting information of the paper. In Figure 1 one can see how the error of HFE calculations depends on the grid size. For all investigated systems starting from the grid size $\Delta R = 0.05$ Bohr, the error is smaller than the desired threshold 0.001 kcal/mol. We take the grid with $\Delta R = 0.05$ Bohr as the fine-grid for the numerical solution of the RISM equations. To determine the optimal cutoff distance, we performed the RISM calculations for the systems with very large cutoff distance $R_{\text{cutoff}} = 409.6$ Bohr and calculated the numerical error of HFE calculations with Eq.(78) for argon, sodium chloride, methane, methanol and water. We use grids with $2^p$ ($p = 8\ldots13$) points which gives cutoff distances from 12.8 Bohr to 409.6 Bohr. The results of the calculations are presented in Figure 2. We can see that to achieve the desired accuracy of HFE calculations one should use a cutoff distance 204.8 Bohr, which corresponds to 4096 grid points with the grid size 0.05 Bohr.

Also, from Figure 1 and Figure 2 one can see that a grid with $\Delta R_{\text{coarse}} = 0.8$ Bohr, $R_{\text{cutoff}} = 25.6$ Bohr is enough to give a numerical error in the HFE
Figure 3: Dependencies of the nested multi-grid speedup $S_{NMG}(\varepsilon) = \frac{t_{\text{one-grid}}(\varepsilon)}{t_{NMG}(\varepsilon)}$, the multi-grid speedup $S_{MG}(\varepsilon) = \frac{t_{\text{one-grid}}(\varepsilon)}{t_{MG}(\varepsilon)}$ and the nested Picard iteration speedup $S_{NP}(\varepsilon) = \frac{t_{\text{one-grid}}(\varepsilon)}{t_{NP}(\varepsilon)}$ on the $L_2$-norm accuracy of calculations.

calculations less than 1 kcal/mol. Thus, this grid size and cutoff distance are used in the current work as parameters of the coarsest-grid in the multi-grid algorithm.

4.2 Comparison of performance of the one-grid Picard iteration, nested Picard iteration, multi-grid and nested multi-grid

We compare the numerical performance of the one-grid iteration Eq.(36), the nested Picard iteration Eq.(51) - Eq.(53), the multi-grid iteration Eq.(62) - Eq.(65) and the nested multi-grid iteration Eq.(66) - Eq.(62) Eq.(65).

In the experiments, the coarsest grid has 32 grid points, grid size $\Delta R = 0.8$ Bohr and cutoff distance $R_{\text{cutoff}} = 25.6$ Bohr. The finest grid has 4096 grid points, grid size $\Delta R = 0.05$ Bohr and cutoff distance $R_{\text{cutoff}} = 204.8$ Bohr. The one-level iteration was performed on the finest grid only.

To compare the efficiency of the methods, the RISM equations were solved numerically for the same molecule (methane), with the same accuracy, with the one-level Picard iteration, nested Picard iteration and the proposed multi-grid-based algorithms.
Figure 4: Dependencies of the nested multi-grid and multi-grid speedup with regards to the nested Picard iterations \((t_{NP}(\epsilon)/t_{NMG}(\epsilon), t_{NP}(\epsilon)/t_{MG}(\epsilon))\) on the \(L_2\)-norm accuracy of calculations.

Table 1: Comparison between the multi-grid and nested Picard iteration. Number of iteration steps and percent of total computational time, spent on each grid.

<table>
<thead>
<tr>
<th>Grid Points</th>
<th>(\Delta R) (Bohr)</th>
<th>(R_{\text{cutoff}}) (Bohr)</th>
<th>Number of iteration steps</th>
<th>% of time spent on level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>multi-grid</td>
<td>Nested Picard</td>
</tr>
<tr>
<td>4096</td>
<td>0.05</td>
<td>204.8</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>2048</td>
<td>0.05</td>
<td>102.4</td>
<td>4</td>
<td>394</td>
</tr>
<tr>
<td>1024</td>
<td>0.05</td>
<td>51.2</td>
<td>18</td>
<td>3624</td>
</tr>
<tr>
<td>512</td>
<td>0.05</td>
<td>40.6</td>
<td>52</td>
<td>3868</td>
</tr>
<tr>
<td>406</td>
<td>0.1</td>
<td>40.6</td>
<td>52</td>
<td>4406</td>
</tr>
<tr>
<td>128</td>
<td>0.2</td>
<td>40.6</td>
<td>57</td>
<td>6399</td>
</tr>
<tr>
<td>64</td>
<td>0.4</td>
<td>40.6</td>
<td>4843</td>
<td>7895</td>
</tr>
<tr>
<td>32</td>
<td>0.8</td>
<td>40.6</td>
<td>11336</td>
<td>10422</td>
</tr>
</tbody>
</table>
The dependencies of the speed-up factors Eq.(80), Eq.(81) on the accuracy of the calculations are presented in Figure 3.

As one can see, the speed-up decreases when accuracy increases. This can be explained by the fact that for higher accuracies high frequencies of the solution are essential, so we need to perform more time-consuming Picard iteration steps on the fine grids. Nevertheless, even for an accuracy of $\varepsilon = 10^{-10}$ multi-grid methods are about 30 times faster than one-level iteration. One can see, that for low accuracies multi-grid and nested multi-grid have almost the same performance, while for the high accuracies nested multi-grid is slightly faster. The speedup of the nested Picard iteration is lower and decreases faster. For $\varepsilon = 10^{-10}$, the nested Picard iteration is only 4.5 times faster than the one-level iteration. Figure 4 presents the dependency of the multi-grid speedup $t_{NP}(\varepsilon)/t_{MG}(\varepsilon)$ and nested multi-grid speedup $t_{NP}(\varepsilon)/t_{NMG}(\varepsilon)$ with regard to the nested Picard iteration.

We see that for the low accuracy regime, the nested Picard iteration is less than 1.5 times slower than the multi-grid iteration, but when the accuracy increases, the efficiency of the nested Picard iteration becomes worse than the efficiency of the multi-grid methods. Indeed, for an accuracy of $\varepsilon = 10^{-10}$ the nested multi-grid iteration is almost 7 times faster than the nested iteration method. This happens because in the nested Picard iteration there is no correction for the difference between accurate solutions on different grids. While the accuracy of the solution is less than the difference between the accurate solutions on different grids, the multi-grid and nested Picard iteration have similar efficiency. But for the high accuracy regime, using the nested Picard iteration process it is not possible to produce a correct result on the coarse grid, thus the number of expensive fine-grid iteration steps increases and efficiency goes down. If we look at the Table 1, we see that for an accuracy of $\varepsilon = 10^{-10}$ multi-grid performs most of elementary iteration steps and spends most of the time on the coarse grids, while nested Picard iteration method performs a large number of iteration steps on the fine grids.

5 Conclusions

In the paper we described a new multi-grid-based algorithm for solving RISM equations. We adopted a general non-linear multi-grid scheme for solving the RISM equations. We also proposed an extension of the algorithm for the grids with different cutoff distances. Additionally we investigated the efficiency of
the coarse-grid solver and proposed an adaptive algorithm for calculating the optimal number of iteration steps on the coarsest grid. We performed numerical investigations to optimize the algorithm parameters to give the required numerical accuracy of the Hydration Free Energy calculations by RISM. The investigated parameters were: fine-grid cutoff distance, fine-grid discretization step, coarsest grid cutoff distance and coarsest grid size. By numerical tests on polar and non-polar simple and multi-atom molecules it was shown that RISM calculations with a fine grid \{4096, 0.05 \text{ Bohr}\} are able to give a numerical accuracy of the Hydration Free Energy value of 0.001 kcal/mol, which is satisfactory for most of the chemical applications. It was shown that the multi-grid calculations with coarsest grid \{32, 0.8 \text{ Bohr}\} are optimal.

The proposed multi-grid methods were compared to the one-grid Picard iteration scheme, which is the reference algorithm, and to the nested Picard iteration algorithm, which is the most straightforward implementation of the multi-scale scheme. It was shown that for high accuracies the proposed methods are about 30 times faster than the single-grid Picard iteration, and almost 7 times faster than the nested Picard iteration.

6 Acknowledgements

We would like to acknowledge the Deutsche Forschungsgemeinschaft (DFG) - German Research Foundation, and Research Grant FE 1156/2-1 for finance support of research. We would like to acknowledge David Palmer, Gennady Chuev, Andrey Frolov and Ekaterina Ratkova for discussions and critical reading of the manuscript.

7 Toc Entry

![Speedup of RISM solver](image-url)
Speedup factors of RISM solvers with regards to one-level Picard method for different L2-norm accuracies. Solid red line - nested multi-grid (proposed method), dash-dotted black line - multi-grid method, dashed green line - nested Picard iteration. One can see that the performance of the RISM equations solver can be improved more than 30 times with the Multi-Grid technique.

References


the Koninklijke Akademie Van Wetenschappen Te Amsterdam, 17:793–806, 1914.


[29] D. S. Palmer, V. P. Sergieievskyi, F. Jensen, and M. V. Fedorov. Accurate calculations of the hydration free energies of druglike molecules using the


32


