3DRISM Multi-grid Algorithm for Fast Solvation Free Energy Calculations

Volodymyr P. Sergiievskyi[†] and Maxim V. Fedorov[*,‡]

Max Planck Institute for Mathematics in the Sciences, Inselstrasse 22, 04103 Leipzig, Germany, and Nanoscience Division Department of Physics Scottish Universities Physics Alliance (SUPA) Strathclyde University Room JA 6.10 John Anderson Building 107 Rottenrow East Glasgow, U.K. G4 0NG

E-mail: maxim.fedorov@strath.ac.uk

Phone: +44 (0)141-5484012. Fax: +44 (0)141-5522891

Abstract

In the paper we present a fast and accurate method for modeling solvation properties of organic molecules in water with main focus on predicting solvation (hydration) free energies of small organic compounds. The method is based on a combination of (i) a molecular theory, three-dimensional Reference Interaction Sites Model (3DRISM); (ii) fast multi-grid algorithm for solving the high-dimensional 3DRISM integral equations; (iii) recently introduced universal correction (UC) for the 3DRISM solvation free energies by properly scaled molecular partial volume (3DRISM-UC, Palmer et al., J. Phys.: Condens. Matter 22, 492101, (2010)). A fast multi-grid algorithm is the core of the method because it helps to reduce the high computational costs associated with solving the 3DRISM equations. To facilitate future applications of the method we performed benchmarking of the algorithm on a set of several model solutes

[*]To whom correspondence should be addressed
[†]Max Planck Institute for Mathematics in the Sciences
[‡]Strathclyde University

in order to find optimal grid parameters and to test the performance and accuracy of the algorithm. We have shown that the proposed new multi-grid algorithm is in average 24 times faster than the simple Picard method and at least 3.5 times faster than the MDIIS method which is currently actively used by the 3DRISM community (e.g. the MDIIS method has been recently implemented in a new 3DRISM implicit solvent routine in the recent release of the AmberTools 1.4 molecular modeling package (Luchko et al. J. Chem. Theor. Comput., 6, 607-624 (2010)). Then we have benchmarked the multi-grid algorithm with chosen optimal parameters on a set of 99 organic compounds. We show that average computational time required for one 3DRISM calculation is 3.5 minutes per a small organic molecule (10-20 atoms) on a standard personal computer. We also benchmarked predicted solvation free energy values for all of the compounds in the set against the corresponding experimental data. We show that by using the proposed multi-grid algorithm and the 3DRISM-UC model it is possible to obtain good correlation between calculated and experimental results for solvation free energies of aqueous solutions of small organic compounds (correlation coefficient 0.97, root mean square deviation <1 kcal/mol).

Introduction

Integral equation theory of liquids (IETL) is a useful method for theoretical studies of structural and thermodynamical properties of liquids. IETL describes the liquid structure in terms of correlation functions. The central equation in IETL is the Ornstein-Zernike (OZ) equation.[1] In its general molecular form this equation operates with six-dimensional correlation functions even in the case of isotropic molecular systems.[2] Because of the high computational complexity, an efficient numerical solution of the high-dimensional molecular OZ equation is still an open problem. Therefore, there have been developed some approximate models that help to reduce dimensionality of integral equations. Most popular model in this field is the Reference Interaction Site Model (RISM).[3] One of the man approximations behind the original RISM model is that the high-dimensional molecular correlation functions are represented by a set of spherically symmetric

site-site functions. That approximation reduces the original high-dimensional problem to a set of (technically) one-dimensional equations. Due to this fact, the RISM theory is also referenced as 1DRISM.

From a computational point of view it is relatively inexpensive to solve the 1DRISM equations numerically for small molecular solutes ($<10^2$ atoms) with modern computers; and, typically, solutions of the 1DRISM equations give a qualitatively correct description of the solvent structure around solute. To compare, it was shown that RISM solvent representation is more accurate than continuum solvent representation in continuum electrostatics models.[4–6] In addition, RISM theory gives end-point expressions for solvation free energy (SFE) that avoid thermodynamical integration.[7,8] We note though that the original formulae for SFE calculations[7,8] provide only qualitative predictions of trends in the differences of SFEs for different compounds.[9] Recently there were proposed several methods for parameterizing RISM solvation free energy (SFE) calculations that predict SFEs with an accuracy around 1 kcal/mol.[9–13] However, decomposition of molecular functions to site-site spherically symmetric functions leads to inaccurate representation of molecular structure. Therefore, a considerable number of empirical corrections is necessary to achieve good accuracy of predictions.

Another approximation of the Ornstein-Zernike equation is the so-called three-dimensional RISM (3DRISM)[14,15] where a solute molecule is represented as a three dimensional object. The 3DRISM operates with a set of three-dimensional equations and that model provides better spatial description of solute-solvent correlations than the 1DRISM. The 3DRISM method is currently widely used in biochemical applications for the description of solvation properties of biomolecules.[16–19] Another promising application of the 3DRISM theory is computational screening of large databases of drug candidates. As it was recently shown, a 3DRISM-based method accurately predicts thermodynamic parameters of hydrated organic molecules including drug-like molecules.[20,21] However, for small molecules, numerical solution of the multidimensional 3DRISM equations requires significantly more computational time than solution of the 1DRISM equations.[21] High computational expenses of 3DRISM calculations is a real bottleneck of this method that inhibits wider

applications of this technique. In the current work we show that this problem can be overcomed by using of highly efficient multi-grid algorithms.

Coming back to the history of the IETL, the first algorithm used for solving OZ-like integral equations was presumably the Picard iteration method.[22] This method is easy to implement. However, it has comparably low convergence rate. Therefore, there were proposed several alternative iteration schemes in order to improve the convergence rate such as the Newton-Raphson method (NR),[23] the NR-GMRES (Generalized minimal residual method) algorithm,[24] the combined NR-DIIS(direct inversion in iterative subspace) iteration,[25] the Modified DIIS (MDIIS) method[26] and the vector extrapolation method.[27] Recently an efficient 3DRISM equations solver which uses the MDIIS algorithm was implemented in the Amber molecular modeling software.[18]

Another way to increase the speed of calculations is to use two-scale and multi-scale methods.[28–34] However, from a mathematical point of view it is necessary to use all advantages of the multi-scale approach. From the applied mathematics perspective, there is a general 'multi-grid' (MG) technique that is well investigated theoretically and it is rigorously proven to be effective.[35] These days the multi-grid technique is widely used in several areas of computational chemistry (particularly in quantum chemistry and material sciences).[36–39] In spite of that, only recently the multi-grid methods attracted attention of the RISM community.[40–42] In our recent work we have shown, that the multi-grid technique allows one to increase the performance of the numerical 1DRISM solver up to dozen times.[42] One of the main goals of the current work is to develop a fast algorithm for solving the 3DRISM equations, because the latter have been proven to be more advanced from a theoretical point of view.[21,43–47]

We note that general theoretical framework of the multi-grid method for solving RISM equations proposed in Ref. 42 allows one to combine this method with other different numerical solvers. In our work we investigate the numerical performance of two modifications of the multi-grid 3DRISM algorithm where the multi-grid is combined with (i) the Picard iteration method (MG-Picard); and (ii) with the MDIIS method (MG-MDIIS) respectively. By benchmarking of these methods on a set of model compounds we determine the optimal grid parameters for solvation

(hydration) free energy calculations. We test the numerical performance of the proposed methods and compare it to the performance of the standard Picard iteration method and the MDIIS method.

Additionally, we benchmark the speed and accuracy of the algorithm on an extended set of 99 organic compounds. Firstly, we test computational performance of the algorithm. Then we test the accuracy of the SFE calculations with the Universal Correction model (UC) as proposed in Ref. 10. To check the accuracy of the free energy results we calculate the correlation coefficient and root mean square deviation between calculated and experimental data.

Method

3D RISM

In our work we use the Kovalenko-Hirata formulation of the 3D RISM theory[44,48] in order to describe infinitely diluted solutions of small organic solute molecules. Solvent (water) molecules are described by the 1DRISM approximation, while a solute molecule is a three-dimensional object. Structure of the solvent is described by the total and direct correlation functions $h_\alpha(\mathbf{r})$, $c_\alpha(\mathbf{r})$ where α indicates a solvent site. The 3DRISM equations are written in the following way:

$$h_\alpha(\mathbf{r}) = \sum_{\xi=1}^{N_{solvent}} \int_{\mathbb{R}^3} c_\xi(\mathbf{r}) \chi_{\xi\alpha}(\mathbf{r}-\mathbf{r}) d\mathbf{r} \tag{1}$$

where N_{solvent} is the number of solvent sites, $\chi_{\xi\alpha}(\mathbf{r})$ is the solvent susceptibility function for sites ξ and α. Solvent susceptibility functions $\chi_{\xi\alpha}(\mathbf{r})$ are defined as following:

$$\chi_{\xi\alpha}(\mathbf{r}) = \omega_{\xi\alpha}(r) + \rho h^{\text{solv}}_{\xi\alpha}(r), \tag{2}$$

where $r = |\mathbf{r}|$, $\omega_{\xi\alpha}(r) = \delta_{\xi\alpha} + (1-\delta_{\xi\alpha})\delta(r-r_{\xi\alpha})/(4\pi r^2_{\xi\alpha})$, $r_{\xi\alpha}$ is the distance between the sites ξ and α of a solvent molecule, $h^{\text{solv}}_{\xi\alpha}(r)$ is the total site-site correlation function of the solvent sites ξ and α, $\delta_{\xi\alpha}$ is the Kronecker delta and $\delta(r)$ is the Dirac delta function. In this paper we used the functions $h^{\text{solv}}_{\xi\alpha}(r)$ calculated in Ref. 49.

Eq. (1) is completed by closure relations:

$$h_\alpha(\mathbf{r}) = e^{-\beta U_\alpha((r)) + h_\alpha(\mathbf{r}) - c_\alpha(\mathbf{r}) + B_\alpha(\mathbf{r})} - 1, \tag{3}$$

where $\beta = 1/k_B T$, k_B is the Boltzmann constant, T is the temperature, $U_\alpha(\mathbf{r})$ is the interaction potential corresponding to a solute site α, $B_\alpha(\mathbf{r})$ is the bridge functional.

To use iterative solvers we rewrite Eq. (1) in the following form:[50]

$$\gamma_\alpha(\mathbf{r}) = \sum_{\xi=1}^{N_{\text{solvent}}} \int_{\mathbb{R}^3} \mathscr{C}[\gamma_\xi(\mathbf{r}' - \mathbf{r})] \cdot \chi_{\xi\alpha}(\mathbf{r}') d\mathbf{r}' + \theta_\alpha(\mathbf{r}) - \mathscr{C}[\gamma_\alpha(\mathbf{r})] \tag{4}$$

where $\gamma_\alpha(\mathbf{r}) = h_\alpha(\mathbf{r}) - c^S_\alpha(\mathbf{r})$, $c^S_\alpha(\mathbf{r}) = c_\alpha(\mathbf{r}) + \beta U^L_\alpha(\mathbf{r})$, $U_\alpha(\mathbf{r}) = U^S_\alpha(\mathbf{r}) + U^L_\alpha(\mathbf{r})$, $U^S_\alpha(\mathbf{r})$ is a short range potential, $U^L_\alpha(\mathbf{r})$ is a long range potential, $\theta_\alpha(\mathbf{r}) = -\beta \sum_\xi \int_{\mathbb{R}^3} U^L_\xi(\mathbf{r} - \mathbf{r}') \chi_{\xi\alpha}(\mathbf{r}') d\mathbf{r}'$, $\mathscr{C}[\cdot]$ is a closure (bridge) functional.

We use interaction potentials which are superpositions of the site-site interaction potentials:

$$U^S_\alpha(\mathbf{r}) = \sum_{s=1}^{N_{\text{solute}}} u^S_{s\alpha}(|\mathbf{r} - \mathbf{r_s}|); \tag{5}$$

$$U^L_\alpha(\mathbf{r}) = \sum_{s=1}^{N_{\text{solvent}}} u^L_{s\alpha}(|\mathbf{r} - \mathbf{r_s}|); \tag{6}$$

where $\mathbf{r}_s$ is the position of a solute site s with respect to the center of a molecule, N_{solute} is the number of solute sites. In our work the site-site potentials contain Lennard-Jones and Coulomb part. Pair Lennard-Jones parameters are obtained from the atomic LJ parameters by using the Lorentz-Berthelot mixing rules:

$$\sigma_{s\alpha} = \frac{1}{2}(\sigma_s + \sigma_\alpha) \qquad \varepsilon_{s\alpha} = \sqrt{\varepsilon_s \varepsilon_\alpha} \tag{7}$$

To avoid divergence of the algorithm due to the long range behavior of the interaction potentials we separate the short range and the long range of the potentials that we then treat separately by using

the Ng procedure.[51] We use the atomic units for distance and energy Bohr=$0.52918 \cdot 10^{-10}$m and Hartree=$4.35974394 \cdot 10^{18}$ J. This allows us to avoid scaling coefficients in the representation of the Coulomb potential. Thus expressions for the short-range and long-range potentials are written as following:

$$u^{S}_{s\alpha}(r) = u^{LJ(short)}_{s\alpha}(r) + u^{C}_{s\alpha}(r)(1 - \mathrm{erf}(\tau r)) \tag{8}$$

$$u^{L}_{s\alpha}(r) = u^{LJ(long)}_{s\alpha}(r) + u^{C}_{s\alpha}(r)\mathrm{erf}(\tau r) \tag{9}$$

where $u^{C}_{s\alpha}(r)$ is the Coulomb component of the site-site potential, $\mathrm{erf}(r) = \int_{-\infty}^{r} e^{-t^2} dt$, τ=0.5 Bohr^{-1}, $u^{LJ(short)}_{s\alpha}(r)$, $u^{LJ(long)}_{s\alpha}(r)$ are short-range and long-range components of the Lennard-Jones potential respectively. The latter are defined by the following relations:

$$u^{LJ(short)}_{s\alpha}(r) = \begin{cases} u^{LJ}_{s\alpha}(r) - u^{LJ}_{s\alpha}(R_{cut}) & \text{when } r < R_{cut} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$u^{LJ(long)}_{s\alpha}(r) = u^{LJ}_{s\alpha}(r) - u^{LJ(short)}_{s\alpha}(r) \tag{11}$$

where $u^{LJ}_{s\alpha}(r)$ is a Lennard-Jones component of a site-site potential, R_{cut}=8Å .

In the article we use the Kovalenko-Hirata (KH) closure, which is defined as following:[52]

$$\mathscr{C}[\gamma_\alpha(\mathbf{r})] = \begin{cases} e^{-\beta U^{S}_{\alpha}(\mathbf{r}) + \gamma_\alpha(\mathbf{r})} - \gamma_\alpha(\mathbf{r}) - 1 & \text{when } -\beta U^{S}_{\alpha}(\mathbf{r}) + \gamma_\alpha(\mathbf{r}) > 0 \\ -\beta U^{S}_{\alpha}(\mathbf{r}) & \text{otherwise} \end{cases} \tag{12}$$

In the numerical representation of Eq. (4) the functions $\gamma_\alpha(\mathbf{r})$, $\chi_{\xi\alpha}(\mathbf{r})$, $\theta_\alpha(\mathbf{r})$ are defined by their values in the grid points of an uniform Cartesian grid. A grid is defined by two parameters: *spacing* and *buffer*. *Spacing* is the smallest distance between the grid points and *buffer* is the minimal distance from the solute atoms to the boundaries of the grid (see Figure 1 for explanations). At first glance, such parameterization may seem to be inconvenient from a theoretical point of view because the same buffer and spacing parameters may give different grids for different solutes. However, our work is mostly oriented towards future practical applications of the method

and in practical applications we are interested in the accuracy of calculations for different cutoff distances of the correlation functions; and these cutoff distances for a Cartesian grid are defined by the buffer parameter. Using the same buffer parameter we can adjust the size and the shape of the grid preserving a constant cutoff of the solvent correlation functions for different solutes. That provides us a straightforward way to control the accuracy of calculations.

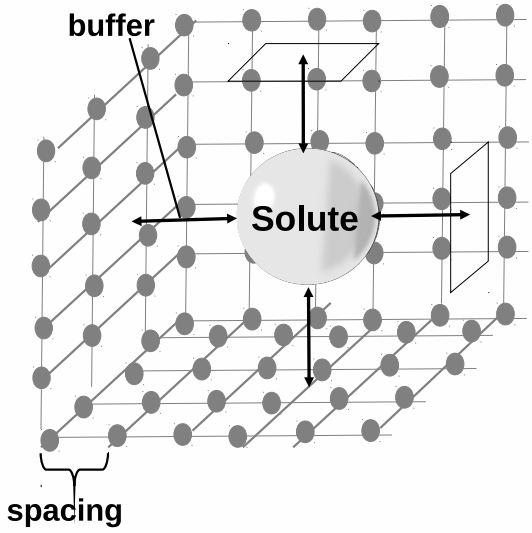


Figure 1: *Spacing* is the minimal distance between the grid points, *buffer* is the minimal distance from the solute atoms to the boundaries of the grid

We denote the forward and the inverse Fourier transforms on the grid $\mathscr{G}$ as $\mathscr{F}_{\mathscr{G}}[\cdot]$, $\mathscr{F}_{\mathscr{G}}^{-1}[\cdot]$ correspondingly. Then a discrete analogue of Eq. (4) reads as:

$$\boldsymbol{\Gamma}^{\mathscr{G}} = \mathscr{F}_{\mathscr{G}}^{-1}\left[\hat{\mathbf{X}} \cdot \mathscr{F}_{\mathscr{G}}\left[\mathscr{C}\left[\boldsymbol{\Gamma}^{\mathscr{G}}\right]\right]\right] + \boldsymbol{\Theta}^{\mathscr{G}} - \mathscr{C}\left[\boldsymbol{\Gamma}^{\mathscr{G}}\right] \tag{13}$$

where $\boldsymbol{\Gamma}^{\mathscr{G}} = \left(\boldsymbol{\gamma}_1^{\mathscr{G}}, \ldots, \boldsymbol{\gamma}_{N_{\text{solvent}}}^{\mathscr{G}}\right)^T$, $\boldsymbol{\Theta}^{\mathscr{G}} = \left(\boldsymbol{\theta}_1^{\mathscr{G}}, \ldots, \boldsymbol{\theta}_{N_{\text{solvent}}}^{\mathscr{G}}\right)^T$, $\hat{\mathbf{X}}^{\mathscr{G}} = [\hat{\boldsymbol{\chi}}_{\xi\alpha}^{\mathscr{G}}]_{N_{\text{solvent}} \times N_{\text{solvent}}}$, $\hat{\boldsymbol{\chi}}_{\xi\alpha}^{\mathscr{G}} = \mathscr{F}_{\mathscr{G}}[\boldsymbol{\chi}_{\xi\alpha}]$, upper index $\mathscr{G}$ means that functions are given by their values in the grid points of the grid $\mathscr{G}$.

Eq. (13) can be written in a more compact way:

$$\mathbf{\Gamma}^{\mathscr{G}} = F[\mathbf{\Gamma}^{\mathscr{G}}] \tag{14}$$

where $F[\mathbf{\Gamma}^{\mathscr{G}}] = \mathscr{F}_{\mathscr{G}}^{-1}\left[\hat{\mathbf{X}} \cdot \mathscr{F}_{\mathscr{G}}\left[\mathscr{C}\left[\mathbf{\Gamma}^{\mathscr{G}}\right]\right]\right] + \mathbf{\Theta}^{\mathscr{G}} - \mathscr{C}\left[\mathbf{\Gamma}^{\mathscr{G}}\right]$.

The Picard iteration method is defined by the following recurrent formula:

$$\mathbf{\Gamma}^{\mathscr{G}}_{n+1} = (1-\lambda)\mathbf{\Gamma}^{\mathscr{G}}_{n} + \lambda F[\mathbf{\Gamma}^{\mathscr{G}}_{n}] \tag{15}$$

where $\mathbf{\Gamma}^{\mathscr{G}}_{n}$ is the n-th step approximation, λ is the coupling parameter.

DIIS and MDIIS iteration

Direct inverse in the iterative subspace (DIIS) method is an iteration method initially introduced to improve convergence of Schrödinger equation solvers.[53] Later modified DIIS (MDIIS) method was applied to the 3DRISM equations.[26] In the DIIS method on the n-th iteration step one finds an approximate solution $\mathbf{\Gamma}^{\mathscr{G}}_{*}$ which is a linear combination of the approximations on the k previous iteration steps:

$$\mathbf{\Gamma}^{\mathscr{G}}_{*} = \sum_{i=1}^{k} C_i \mathbf{\Gamma}^{\mathscr{G}}_{n-k+i} \tag{16}$$

Below we describe the DIIS and MDIIS algorithms which solve the 3DRISM equations in the form (Eq. (14)). We also plan to use the MDIIS algorithm in our multi-grid scheme. This will require to consider a generalized task in the following form:

$$\mathbf{\Gamma}^{\mathscr{G}} = F[\mathbf{\Gamma}^{\mathscr{G}}] + \mathbf{D}^{\mathscr{G}} \tag{17}$$

where $\mathbf{D}^{\mathscr{G}} = \left(\mathbf{d}^{\mathscr{G}}_{1}, \ldots, \mathbf{d}^{\mathscr{G}}_{N_{\text{solvent}}}\right)^{T}$ is an arbitrary vector of corrections. The vector of corrections will be calculated during the multi-grid algorithm when we move from one grid to another one. This procedure is described in the next section of the paper. In the current section we describe one-grid

solvers where vector $\mathbf{D}^{\mathscr{G}}$ is given. Below we describe the DIIS and MDIIS algorithms for a general case of an arbitrary vector $\mathbf{D}$ having in mind that the 3DRISM equations (Eq. (14)) correspond to the case $\mathbf{D}^{\mathscr{G}} \equiv 0$.

In the DIIS method the coefficients C_i in Eq. (16) are chosen to minimize the norm of the residue $\Delta_*^{\mathscr{G}} = \mathbf{\Gamma}_*^{\mathscr{G}} - F[\mathbf{\Gamma}_*^{\mathscr{G}}] - \mathbf{D}^{\mathscr{G}}$. If one assumes linearity of the operator F (which for smooth operators is locally true) then the task reduces to the following system of linear equations:[53]

$$\begin{pmatrix} a_{11} & \dots & a_{1k} & -1 \\ \vdots & \ddots & \vdots & -1 \\ a_{k1} & \dots & a_{kk} & -1 \\ 1 & \dots & 1 & 0 \end{pmatrix} \begin{pmatrix} C_1 \\ \vdots \\ C_k \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \tag{18}$$

where $a_{ij} = \int_{\mathbb{R}^3} \Delta_i(\mathbf{r})\Delta_j(\mathbf{r})d\mathbf{r}$, $\Delta_i(\mathbf{r}) = \mathbf{\Gamma}_{n-k+i}^{\mathscr{G}} - F[\mathbf{\Gamma}_{n-k+i}^{\mathscr{G}}] - \mathbf{D}^{\mathscr{G}}$. In the DIIS method $\mathbf{\Gamma}_*^{\mathscr{G}}$ is used as a solution approximation on the (n+1)-st iteration step. However, such a procedure can lead to a linearly dependent system of equations. The MDIIS iteration method avoids this problem by adding a weighted residue to the (n+1)-st step approximation:[26]

$$\mathbf{\Gamma}'^{\mathscr{G}} = \mathbf{\Gamma}_*^{\mathscr{G}} + \eta \left(F[\mathbf{\Gamma}_*^{\mathscr{G}}] + \mathbf{D}^{\mathscr{G}} - \mathbf{\Gamma}_*^{\mathscr{G}} \right) \tag{19}$$

where η is a weight for the residue. In combination with the standard damping technique the solution approximation on the (n+1)-st step $\mathbf{\Gamma}_{n+1}^{\mathscr{G}}$ in the MDIIS method can be found by using the following formula:

$$\mathbf{\Gamma}_{n+1}^{\mathscr{G}} = (1-\lambda)\Gamma_n^{\mathscr{G}} + \lambda\mathbf{\Gamma}_*^{\mathscr{G}} + \lambda\eta \left(F[\mathbf{\Gamma}_*^{\mathscr{G}}] + \mathbf{D}^{\mathscr{G}} - \mathbf{\Gamma}_*^{\mathscr{G}} \right) \tag{20}$$

In our work we use $\lambda = 0.5$, $\eta = 0.3$. These values are sub-optimal and allow to ensure stability of the algorithm and in the same time retain reasonable performance. Detailed description of the dependence of the computation time on λ and η parameters is given in the supporting information

to the paper.

To make notations shorter we introduce the MDIIS operator $\Xi[\cdot,\cdot]$:

$$\Xi[\mathbf{\Gamma}_n^{\mathscr{G}},\mathbf{D}^{\mathscr{G}}] = (1-\lambda)\mathbf{\Gamma}_n^{\mathscr{G}} + \lambda\mathbf{\Gamma}_*^{\mathscr{G}} + \lambda\eta\left(F[\mathbf{\Gamma}_*^{\mathscr{G}}] + \mathbf{D}^{\mathscr{G}} - \mathbf{\Gamma}_*^{\mathscr{G}}\right) \tag{21}$$

Multi-grid

We use the multi-grid technique in order to decrease the computation time spent on solving the 3DRISM equations. General description of the multi-grid theory can be found in the book.[35] Here we give only short description of the multi-grid method applied to the 3DRISM equations. More information on the theoretical background of the method can be found in our recent paper where a similar computational framework for an efficient algorithm for solving the 1D-RISM equations is described.[42]

In the multi-grid method the numerical task is discretized on several grids with the same buffer but different spacings. Grids with smaller numbers of points and larger spacings are called *coarse* grids, grids with larger number of the points and smaller spacings are called *fine* grids. In our work we consider grids where number of points differ by the factor of 2^n, where $n = 0,1,2,....$

We introduce operators $p[\cdot]$, $r[\cdot]$, which convert a coarse grid to a finer one and vice versa. We introduce an operator $R[\cdot]$ which map a fine-grid function to a coarse grid.

$$R[\mathbf{\Gamma}^{\mathscr{G}}] = \mathbf{\Gamma}^{r[\mathscr{G}]} \tag{22}$$

Also we introduce an operator $P[\cdot]$ which interpolates a coarse-grid function to a fine grid:

$$P[\mathbf{\Gamma}^{r[\mathscr{G}]}] = \mathbf{\Gamma}_{\mathbf{1}}^{\mathscr{G}} \tag{23}$$

In the paper we use the linear interpolation operator.

To make notations simpler we introduce an operator $\Lambda[\cdot;\cdot]$:

$$\Lambda[\mathbf{\Gamma}_{\mathscr{G}};\mathbf{D}^{\mathscr{G}}] = (1-\lambda)\mathbf{\Gamma}^{\mathscr{G}} + \lambda\left(F_{\mathscr{G}}[\mathbf{\Gamma}^{\mathscr{G}}] + \mathbf{D}^{\mathscr{G}}\right). \tag{24}$$

A multi-grid iterative algorithm which solves the task Eq. (17) can be written in the following form:

$$\mathbf{\Gamma}^{\mathscr{G}}_{n+1} = \mathscr{M}^{l}_{\mathscr{G}}\left[\mathbf{\Gamma}^{\mathscr{G}}_{n};\mathbf{D}^{\mathscr{G}}\right], \tag{25}$$

where $\mathbf{\Gamma}^{\mathscr{G}}_{n}$ is the n-th step approximation, $\mathscr{M}^{l}_{\mathscr{G}}[\cdot;\cdot]$ is a multi-grid operator which performs one multi-grid iteration step of the depth l on the grid $\mathscr{G}$. To calculate the multi-grid operator of the depth $l=0$ one performs m_0 one-grid iteration steps on the grid $\mathscr{G}$. The multi-grid technique can be applied to both: the Picard and the MDIIS iteration methods. We define a generalized operator $\mathbf{\Phi}[\cdot;\cdot]$ in the following way:

$$\mathbf{\Phi}[\mathbf{\Gamma}^{\mathscr{G}}_{n};\mathbf{D}^{\mathscr{G}}] = \begin{cases} \Lambda[\mathbf{\Gamma}^{\mathscr{G}}_{n};\mathbf{D}^{\mathscr{G}}] & \text{for MG-Picard method} \\ \Xi[\mathbf{\Gamma}^{\mathscr{G}}_{n};\mathbf{D}^{\mathscr{G}}] & \text{for MG-MDIIS method} \end{cases} \tag{26}$$

Then the multi-grid operator of the depth $l=0$ is defined as:

$$\mathscr{M}^{0}_{\mathscr{G}}\left[\mathbf{\Gamma}^{\mathscr{G}};\mathbf{D}^{\mathscr{G}}\right] = \mathbf{\Phi}^{m_0}\left[\mathbf{\Gamma}^{\mathscr{G}};\mathbf{D}^{\mathscr{G}}\right] \tag{27}$$

For $l>0$, given the n-th step approximation $\mathbf{\Gamma}^{\mathscr{G}}_{n}$ and the correction vector $\mathbf{D}^{\mathscr{G}}$, the multi-grid operator $\mathscr{M}^{l}_{\mathscr{G}}[\cdot;\cdot]$ is calculated by the following algorithm:

Input parameters: $\mathbf{\Gamma}^{\mathscr{G}}_{n}$, $\mathbf{D}^{\mathscr{G}}$, l

Result: $\mathbf{\Gamma}^{\mathscr{G}}_{n+1} = \mathscr{M}^{l}_{\mathscr{G}}[\mathbf{\Gamma}^{\mathscr{G}}_{n};\mathbf{D}^{\mathscr{G}}]$

1. Perform ν_1 Picard iteration steps on the fine grid (in our work $\nu_1 = 5$):

$$\mathbf{\Gamma}'^{\mathscr{G}} = \Lambda^{\nu_1}\left[\mathbf{\Gamma}^{\mathscr{G}}_{n};\mathbf{D}^{\mathscr{G}}\right]$$

2. Move to the coarse grid $r[\mathscr{G}]$:

$$\mathbf{\Gamma}^{r[\mathscr{G}]}_{(0)} = R[\mathbf{\Gamma}'^{\mathscr{G}}];$$

3. Calculate the coarse-grid correction:

$$\mathbf{E}^{r[\mathscr{G}]} = R\left[F[\mathbf{\Gamma}'^{\mathscr{G}}]\right] - F[\mathbf{\Gamma}^{r[\mathscr{G}]}_{(0)}]$$

4. Perform recursively μ multi-grid iteration steps of depth $l-1$ on the coarse-grid (in our work μ=1):

$$\mathbf{\Gamma}^{r[\mathscr{G}]}_{(\mu)} = \left(\mathscr{M}^{l-1}_{r[\mathscr{G}]}\right)^{\mu}\left[\mathbf{\Gamma}^{r[\mathscr{G}]}_{(0)}; R[\mathbf{D}^{\mathscr{G}}] + \mathbf{E}^{r[\mathscr{G}]}\right]$$

5. Correct the fine-grid solution using the coarse-grid results:

$$\mathbf{\Gamma}''^{\mathscr{G}} = \mathbf{\Gamma}'^{\mathscr{G}} + P\left[\mathbf{\Gamma}^{r[\mathscr{G}]}_{(\mu)} - \mathbf{\Gamma}^{r[\mathscr{G}]}_{(0)}\right]$$

6. Perform ν_2 Picard iteration steps on the fine grid (in our work $\nu_2 = 0$):

$$\mathbf{\Gamma}^{\mathscr{G}}_{n+1} = \Lambda^{\nu_2}\left[\mathbf{\Gamma}''^{\mathscr{G}}; \mathbf{D}^{\mathscr{G}}\right]$$

In the paper, the number of the iteration steps m_0 in the multi-grid operator of the depth $l = 0$ depends on the number of the multi-grid iteration step n: $m_0 = m_0(n)$. We define $m_0(n)$ in such a way that after $m_0(n)$ iteration steps, a residue decays by the factor K_n:

$$K_n||\Phi^{m_0(n)}[\mathbf{\Gamma}^{\mathscr{G}}_n; \mathbf{D}^{\mathscr{G}}] - \Phi^{m_0(n)+1}[\mathbf{\Gamma}^{\mathscr{G}}_n; \mathbf{D}^{\mathscr{G}}]|| < ||\mathbf{\Gamma}^{\mathscr{G}}_n - \Phi[\mathbf{\Gamma}^{\mathscr{G}}_n; \mathbf{D}^{\mathscr{G}}]|| \tag{28}$$

We call the value K_n *the decay factor*.

Constant decay factor may lead to a non-smooth decay of residue from one multi-grid iteration step to another which in turn leads to increasing of the number of the idle coarse-grid iteration steps (see Figure 2, solid line). To achieve a smoother decay of the error, in our paper we change

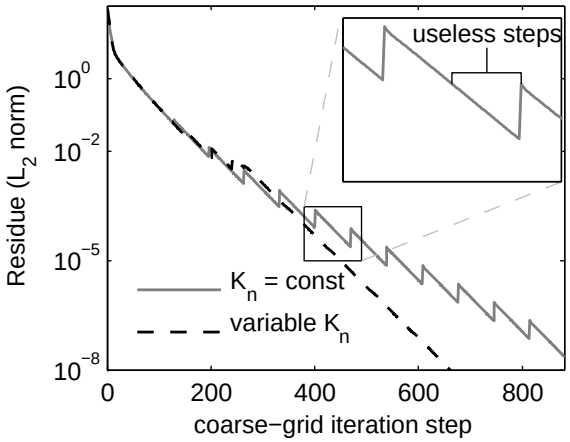


Figure 2: Coarse-grid residue decays with the number of the iteration steps in the multi-grid method. Two cases are shown: constant decay factor $K_n = 10$ (solid line), and variable decay factor K_n (dashed line). System: argon aqueous solution, spacing 0.1Å , buffer 6.4Å Peaks on the saw-shaped line (K_n=const) correspond to the boundaries of multi-grid iteration steps. The coarse-grid correction is re-calculated when iteration returns from the coarse grid to the fine grid. Saw-shaped line means that iteration steps on a coarse grid are performed even after the desired accuracy of the coarse-grid correction calculation has been achieved. Thus, a significant number of coarse-grid iteration steps are actually idle because they do not improve the final result. Introducing a variable decay factor allows one to adjust the accuracy of the coarse-grid calculations and to avoid the idle iteration steps.

K_n by the following recursive formula:

$$K_{n+1} = \begin{cases} \max(\frac{1}{\alpha}K_n, K_{\min}) & \text{if } ||\mathbf{\Gamma}^{\mathscr{G}}_{n,m_0} - \Phi[\mathbf{\Gamma}^{\mathscr{G}}_{n,m_0}; \mathbf{D}^{\mathscr{G}}]|| < ||\mathbf{\Gamma}^{\mathscr{G}}_{n+1} - \Phi[\mathbf{\Gamma}^{\mathscr{G}}_{n+1}; \mathbf{D}^{\mathscr{G}}]|| \\ \min(\beta K_n, K_{\max}) & \text{otherwise} \end{cases} \tag{29}$$

where $\mathbf{\Gamma}^{\mathscr{G}}_{n,m_0} = (\Phi_{\mathscr{G}})^{m_0(n)}[\mathbf{\Gamma}^{\mathscr{G}}_n; \mathbf{D}^{\mathscr{G}}]$, $\alpha = 2$, $\beta = 1.2$. For the MG-Picard method we use $K_0 = 10$, $K_{\min} = 5$, $K_{\max} = 100$, for the MG-MDIIS method we use $K_0 = 100$, $K_{\min} = 10$, $K_{\max} = 100$. This allows us to smooth the decay of error and to reduce the total number of the iteration steps (see Figure 2, dashed line).

Usually iterative algorithms stop when the norm of the residue is less than some threshold. However, this method has its own disadvantages. The first one is that a small residue between two iteration steps does not necessarily imply a small distance from the current approximation to the exact solution. The second one is that a threshold is typically given in dimensionless values which have no physical meaning and thus one has no guidelines to chose an appropriate threshold. In the current work we use another criteria to stop iteration steps. Multi-grid iteration stops on the n-th iteration step if the following condition is satisfied:

$$||\mathbf{\Gamma}_n - \mathbf{\Gamma}_{n+m}|| < \varepsilon_{\text{tres}} \tag{30}$$

where m is such, that

$$||\mathbf{\Gamma}^{\mathscr{G}}_{n+m} - \mathbf{\Gamma}^{\mathscr{G}}_{n+m+1}|| < 0.01||\mathbf{\Gamma}^{\mathscr{G}}_n - \mathbf{\Gamma}^{\mathscr{G}}_{n+1}|| \tag{31}$$

We use such a condition because usually $\mathbf{\Gamma}^{\mathscr{G}}_{n+m}$ is a good approximation of the exact solution. In the paper we use a norm based on the Solvation Free Energy calculations:

$$||\mathbf{\Gamma}^{\mathscr{G}}_1 - \mathbf{\Gamma}^{\mathscr{G}}_2|| = |\Delta G_{KH}(\mathbf{\Gamma}_1) - \Delta G_{KH}(\mathbf{\Gamma}_2)| \tag{32}$$

The solvation free energy is calculated in the 3DRISM-KH approximation:[54]

$$\Delta G_{KH}(\mathbf{\Gamma}^{\mathscr{G}}) = \rho k_B T \sum_{\alpha}^{N_{\text{solvent}}} \int_{\mathbb{R}^3} \theta(-h_\alpha(\mathbf{r}))h_\alpha(\mathbf{r}) - \frac{1}{2}c_\alpha(\mathbf{r})h_\alpha(\mathbf{r}) - c_\alpha(\mathbf{r})d\mathbf{r} \tag{33}$$

where $\theta(\cdot)$ is the Heaviside step function. Because of such definition our threshold has well-defined physical meaning and is measured in energy units. In our work we use $\varepsilon_{\text{tres}}$=0.001 kcal/mol.

To make the calculations faster, in addition to the multi-grid technique we use several grids with the same spacing but different buffers. We introduce a grid-enlargement operator $e[\cdot]$ which enlarges the buffer of a grid. We introduce an operator $E[\cdot]$ which extrapolates a solution $\mathbf{\Gamma}^{\mathscr{G}}$ to a grid $e[\mathscr{G}]$.

$$E[\mathbf{\Gamma}^{\mathscr{G}}] = \mathbf{\Gamma}^{e[\mathscr{G}]} \tag{34}$$

Because functions $\gamma_\alpha(\mathbf{r})$ tend to zero when $|\mathbf{r}| \to \infty$, operator $E[\cdot]$ extrapolates functions by adding zeros at those parts of the grid $e[\mathscr{G}]$ which do not belong to the grid $\mathscr{G}$. The scheme of the iteration can be written in the following way:

$$\mathbf{\Gamma}_0^{\mathscr{G}} \xrightarrow{\text{solve 3DRISM eqs.}} \mathbf{\Gamma}_*^{\mathscr{G}} \xrightarrow{E[\cdot]} \mathbf{\Gamma}_0^{e[\mathscr{G}]} \xrightarrow{\text{solve 3DRISM eqs.}} \mathbf{\Gamma}_*^{e[\mathscr{G}]} \xrightarrow{E[\cdot]} \dots \tag{35}$$

We start from a zero approximation $\mathbf{\Gamma}_0^{\mathscr{G}}$ on the grid $\mathscr{G}$ with a small buffer and using the scheme Eq. (35) after several steps we obtain a solution on a grid with a large buffer.

Computation details

We performed 3DRISM calculations for infinitely diluted aqueous solutions of argon, methane, methanol and dimethyl ether (DME). For the partial charges and Lennard-Jones (LJ) parameters of the solute molecules we used the OPLS-AA force-field parameters.[55] We used the MSPC-E water model[49] to describe solvent. In the 3DRISM calculations we used total site-site correlation functions of water which were initially calculated by the dielectrically consistent RISM technique.[49] Pairwise σ Lennard-Jones parameters were calculated as an arithmetic mean of atomic parameters,

pairwise ε Lennard-Jones parameters were calculated as a geometric mean of atomic parameters:

$$\sigma_{12} = \frac{\sigma_1 + \sigma_2}{2}; \qquad \varepsilon_{12} = \sqrt{\varepsilon_1 \cdot \varepsilon_2} \tag{36}$$

Calculations were performed on the Intel(R) Xeon(R) CPU X5650 with the clocking 2.67GHz. For calculation of the direct and inverse fast Fourier transforms the FFTW3 library was used.[56]

Results

Finding the optimal parameters for the multi-grid solver

We performed 3DRISM calculations for infinitely diluted aqueous solutions of four compounds: argon, methane, methanol and DME. To determine the optimal grid parameters we performed solvation free energy (SFE) calculations on grids with different spacing parameters and different buffers. In Figure 3 the dependence of calculation errors on the spacing parameter is shown. For the calculations we used several different grids with the fixed buffer of 8Å and different spacings which vary from 0.1Å to 2Å . Errors were calculated as absolute values of the differences between SFEs calculated on a current grid and SFEs calculated on the very fine grid with the spacing of 0.05Å and the buffer of 8Å . The results show that the the grid with the spacing of 0.2Å provides an error that is less than 0.1 kcal/mol for all solutes which is acceptable for the most of chemical applications. Thus in our work we use the grid with spacing of 0.2Å .

In Figure 4 we show the dependency of calculation errors on the grid buffer. The calculations were performed on grids with fixed spacing of 0.2Å and different buffers varying from 8Å to 20Å . Errors were calculated as differences between the SFEs calculated on the current grid and the SFEs calculated on the very fine grid with spacing of 0.2Å and buffer of 30Å . The figure shows that the grid with the buffer of 15Å is enough to provide the accuracy of SFE calculations $\leqslant$ 0.1kcal/mol.

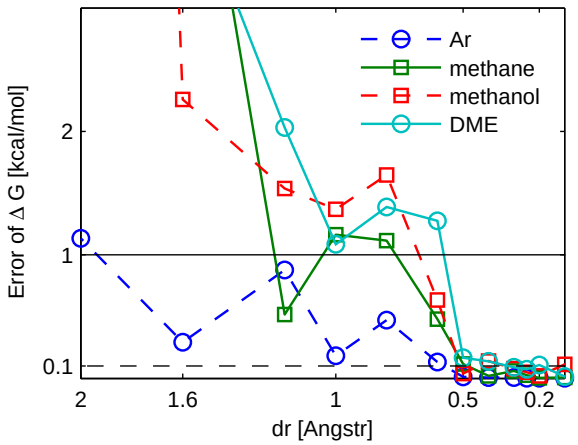


Figure 3: Dependency of the calculation errors on grid spacing at constant buffer (8Å)

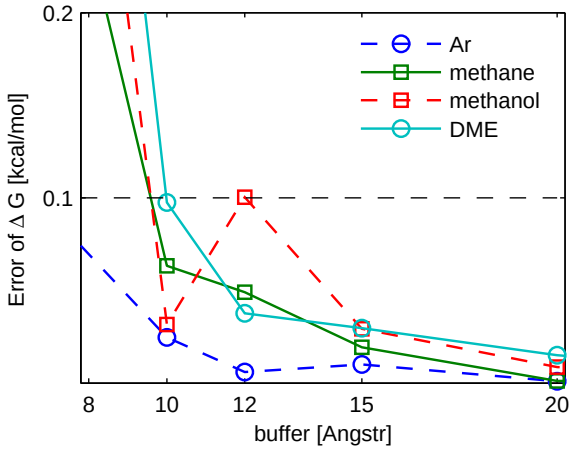


Figure 4: Dependency of calculation errors on the grid buffer with constant spacing of the grid (0.2Å).

Computational benchmarks of different 3DRISM solvers

To check the numerical performance of the proposed multi-grid algorithm we performed 3DRISM calculations for infinitely diluted aqueous solutions of argon, methane, methanol and DME using the Picard iteration, the MDIIS, the MG-Picard and the MG-MDIIS methods. For the Picard and the MDIIS methods the grid with spacing of 0.2Å and buffer of 15Å was used. For the multi-grid methods (MG-Picard, MG-DIIS) we used the scheme Eq. (35) with two enlargements: we started from the grid with the buffer of 7.65Å , then moved to the grid with the buffer of 10.71Å and finished iteration on the grid with the buffer of 15Å . Solutions on the girds with smaller buffers

were used as initial guesses for the grids with larger buffers. For each buffer we used the multi-grid algorithm which uses 3 different grids (depth $l = 2$). All calculations were performed on the same personal computer, Intel(R) Xeon(R) CPU X5650 with clocking 2.67GHz.

Table 1: Computation expenses of 3DRISM calculations with the Picard iteration, MDIIS, MG-Picard and MG-MDIIS methods.

Compound	**Picard iteration**	**MDIIS**	**MG-Picard**	**MG-DIIS**
argon	1148 sec	167 sec	46 sec	50 sec
methane	1484 sec	154 sec	149 sec	82 sec
methanol	1857 sec	416 sec	165 sec	83 sec
dimethyl ether	4462 sec	509 sec	241 sec	133 sec

Computational expenses on solving 3DRISM equations for each of the investigated four methods are presented in Table 1. These results show that the Picard iteration is the least efficient method, while the most efficient is the MG-MDIIS method. We note that the multi-grid methods in all investigated cases are more efficient than the one-grid methods.

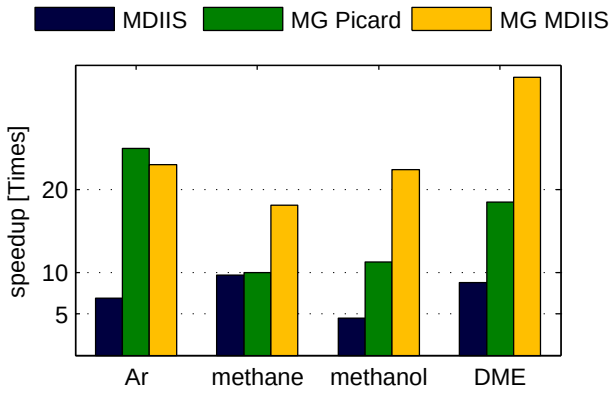


Figure 5: Speed up of the calculations by using the MDIIS, the MG-Picard and the MG-MDIIS methods as compared to the Picard iteration method.

Figure 5 compares computational performance of the MDIIS, the MG-Picard and the MG-MDIIS with the Picard iteration method. The figure shows that for all four compounds multi-grid methods give more than 10 times speedup while for three of these four compounds the MG-MDIIS method is more than 20 times faster than the Picard iteration method. Average speedup factors with respect to the plain Picard method for the MDIIS, the MG-Picard and the MG-MDIIS methods are

correspondingly 7.4, 16.2 and 24.2. The most effective is the MG-MDIIS method that is in average about 3.5 faster than the MDIIS method. Difference between the multi-grid methods is not very large: the MG-MDIIS method is in average only 1.5 times faster than the MG-Picard method. The results show that the multi-grid scheme can be effectively used in combination with different types of coarse-grid iteration methods for solving the 3DRISM equations for aqueous solutions of small non-charged molecules.

Computational benchmarks on a large set of organic molecules

The main goal of this part of our study was to investigate the overall efficiency of the new method in a view of large-scale practical applications like, e.g. physical-chemical profiling of large sets of organic compounds. We performed an additional benchmark and tested the efficiency of the new algorithm on a set of organic molecules as well as the accuracy of the SFE prediction. We estimate average computational expanses for the 3DRISM calculations also check whether numerical accuracy of the calculations is enough for accurate estimation of SFE.

We have chosen a set of 99 organic molecules. This set of molecules is a part of the set used in Ref. 21. The set includes alkanes, ketones, alkyl-benzenes, alcohols, alkyl-phenols, ethers and other (polyfunctional) molecules. Quantity of atoms in molecules of the set varies from 5 to 31. Average number of atoms in a molecule is 16. The full list of molecules in the set is provided in the supporting information. The Antechamber tool[57] from the Amber Tools 1.4 Package[58] was used for molecular structure optimization and assigning Force-Field parameters. Structures of the molecules were optimized by using the AM1 method.[59] Atomic partial charges were calculated by using the bond charge correction(BCC) method.[60,61] LJ parameters from the General Amber Force Field (GAFF)[62] were assigned to the solutes. The benchmark calculations for simple molecules reported above show that the most effective is a combination of the multi-grid and MDIIS(MG-MDIIS) methods. Therefore, we use the MG-MDIIS algorithm in our benchmarking of the overall efficiency of the method.

Figure 6 shows dependency of the computational time spent on MG-MDIIS 3D-RISM calcula-

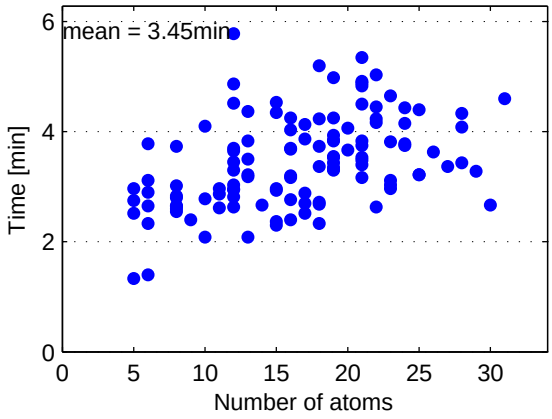


Figure 6: Dependency of the computational time spent on MG-MDIIS calculations on the molecule number of atoms for 99 organic molecules from the chosen molecule set.

tions on the number of atoms in a molecule. The plot shows that the computational time can essentially vary for different molecules even if the molecules have the same number of atoms. However, this somehow counterintuitive result has a straightforward explanation. Indeed, convergence of the algorithm depends not only on the number of atoms but also on the chemical composition of a molecule and its structure, particularly on the distribution of atomic partial charges and the molecule surface accessible area. This is illustrated by the results shown in Figure 3 and Figure 4 that show different error dependencies for polar and non-polar molecules. Also, even if two different molecules have the same number of atoms, they may still have rather different shapes. This can result in different grid sizes for them. More compact molecules need smaller grids than the less compact molecules, even if they have the same buffer and the same number of atoms. Therefore, combination of these two factors causes this significant deviation of computational time for molecules of the same number of atoms. However, the computational time for any molecule in the set is still less than 6 minutes. Average computational time is some 3.5 minutes (3 min 27 sec).

We used the 3D RISM correlation functions calculated by MG-MDIIS method as an input for SFE calculations for all molecules from the above mentioned set of 99 organic compounds. We used 25 molecules as a training set and the rest of the set (74 molecules) as a test set. The lists of all compounds in the training and test sets are given in the supporting information. For accurate SFE calculations we used the Universal Correction (UC) method that was introduced in recent papers

Ref. 20,21. We tested two modifications of the UC method. The first one (UC-KH method) is based on the Kovalenko-Hirata (KH) Free Energy functional Eq. (33). The second one (UC-GF method) is based on the Free Energy calculations using the Gaussian Fluctuations (GF) formula:[63]

$$\Delta G_{GF} = \rho k_B T \sum_{\alpha=1}^{N_{\text{site}}} \int_{\mathbb{R}^3} (-\frac{1}{2} h_\alpha(\mathbf{r}) c_\alpha(\mathbf{r}) - c_\alpha(\mathbf{r})) d\mathbf{r} \tag{37}$$

where k_B is the Boltzmann constant, T is the temperature, ρ is the number density of a bulk solvent. In the UC-GF method (ΔG_{UC}^{GF}) and in the UC-KH method (ΔG_{UC}^{KH}) SFE is calculated by using the following relations:

$$\Delta G_{UC}^{GF} = \Delta G_{GF} + a_{GF} \rho V + b_{GF} \tag{38}$$

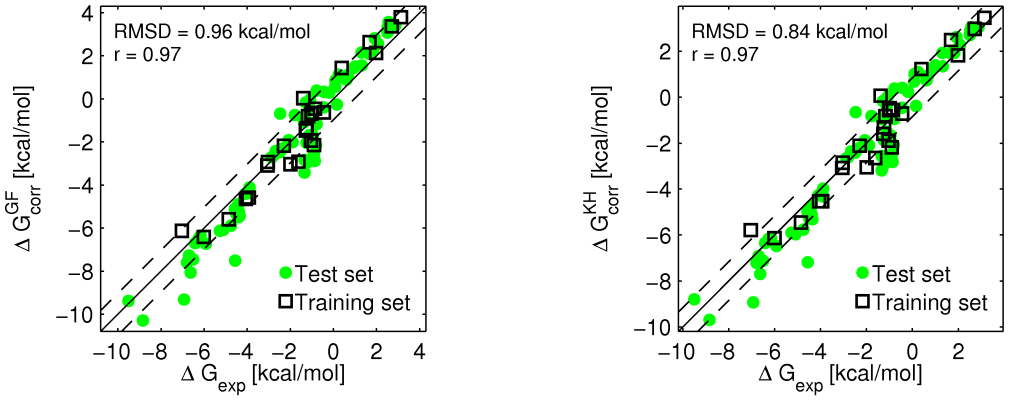
$$\Delta G_{UC}^{KH} = \Delta G_{KH} + a_{KH} \rho V + b_{KH} \tag{39}$$

where V is the partial molar volume of the molecule, a_{GF}, b_{GF}, a_{KH}, b_{KH} are calculated by using the linear regression method to fit experimental data. Partial molar volume of a molecule was calculated by the following formula:[64,65]

$$V = \left(\frac{1}{\rho} + 4\pi \int_0^\infty (g_{oo}(r) - 1) r^2 dr \right) \left(1 - \rho \sum_{\alpha=1}^{N_{\text{site}}} \int_{\mathbb{R}^3} c_\alpha(\mathbf{r}) d\mathbf{r} \right) \tag{40}$$

where $g_{oo}(r)$ is the oxygen-oxygen RDF of bulk water.

Using the training set of compounds, the following values of coefficients were obtained by the linear regression fitting procedure[11,21] : a_{GF}=-2.23 kcal/mol b_{GF}=1.28 kcal/mol for the UC-GF method and a_{KH} = -3.51 kcal/mol, b_{KH} = 0.81 kcal/mol for the UC-KH. Figure 7(a) and Figure 7(b) shows the correlation between the experimental values ΔG_{exp} and the calculated values $\Delta G_{\text{UC}}^{GF}$ and ΔG_{UC}^{KH}. The correlation coefficient is 0.97 for the both methods. Root mean square deviation (RMSD) on a test set is 0.96 kcal/mol for the UC-GF method and 0.84 kcal/mol for the UC-KH method. Accuracy of predictions is comparable with accuracies of experimental methods[12,66] and corresponds to accuracies of current state-of-the-art methods for SFE calculations by molecular dynamics[67–70] and other advanced molecular theories (e.g. energy representation



(a) Universal Correction based on the GF expression: $\Delta G^{GF}_{UC} = \Delta G_{GF} + a_{GF}\rho V + b_{GF}$, where $a_{GF} = -2.23$ kcal/mol, $b_{GF} = 1.28$ kcal/mol

(b) Universal Correction based on the KH expression: $\Delta G^{KH}_{UC} = \Delta G_{KH} + a_{KH}\rho V + b_{KH}$, where $a_{KH} = -3.51$ kcal/mol, $b_{KH} = 0.81$ kcal/mol

Figure 7: Correlation of experimentally measured SFEs with the SFEs values calculated by the Universal Correction method for the investigated set of organic molecules.

method by Matubayasi and Nakahara[71–73]).[9,74] Thus we show that the numerical accuracy of the algorithm is enough for SFE calculations and parameterization of calculation results.

Conclusions

In the paper we proposed a new multi-grid based method which solves the 3DRISM equations. To determine the optimal grid parameters we performed 3DRISM calculations for infinitely diluted aqueous solutions of argon, methane, methanol and dimethyl ether. We showed that on the grid with the spacing of 0.2Å and the buffer of 15Å the maximal error is less than 0.1 kcal/mol. We tested two modifications of the multi-grid algorithm: MG-Picard and MG-MDIIS methods. We compared the numerical efficiency of the multi-grid algorithms with the the numerical efficiency of the standard Picard iteration method and the MDIIS method. We showed that the MG-MDIIS algorithm is more than 24 times faster than the Picard iteration method and more than 3.5 times faster than the MDIIS method.

In turn, efficiencies of the MG-DIIS and MG-Picard methods do not differ very much. The MG-DIIS method is about 1.5 times faster than the MG-Picard method. We suggest that the most

effective MG-MDIIS method can be used in the future as a fast tool for calculations of Solvation Free Energy for organic molecules. To support this statement we performed 3DRISM calculations for aqueous solutions of 99 organic compounds. For all compounds in the set the computational time does not exceed 6 minutes per one molecule while the average computational time is only 3.5 minutes per one molecule on a standard personal computer. We calculated solvation free energies by using GF and KH expressions with the universal partial molar volume corrections (UC-GF and UC-KH methods). We showed that calculated and experimental values of solvation free energy are strongly correlated to each other (correlation coefficient is 0.97). RMSD error for the test set of compounds is less than 1 kcal/mol for both UC-GF and UC-KH methods. The performed tests show that the proposed algorithm can be used for fast and accurate predictions of aqueous solvation free energies of neutral molecules.

Acknowledgement

We are grateful to Wolfgang Hackbusch for insightful discussions on different aspects of multi-grid mathematical background. We thank David S. Palmer for critical reading of the manuscript. This work was partially supported by Deutsche Forschungsgemeinschaft (DFG) - German Research Foundation, Research Grant FE 1156/2-1 and REA Research Executive Agency, Grant No. 247500 "BioSol", Programme: FP7-PEOPLE-2009-IRSES.

Supporting Information Available

The list of the organic molecules used in the paper for benchmarking of the algorithm and for the dependencies of the computational time on the λ and η parameters. This material is available free of charge via the Internet at `http://pubs.acs.org/`.

References

(1) Ornstein, L. S.; Zernike, F. *Proceedings of the Koninklijke Akademie Van Wetenschappen Te Amsterdam* **1914**, *17*, 793–806.

(2) Hansen, J.-P.; McDonald, I. R. *Theory of Simple Liquids, 4th ed*; Elsevier Academic Press, Amsterdam, The Netherlands, 2000.

(3) Chandler, D.; Andersen, H. C. *J. Chem. Phys.* **1972**, *57*, 1930–1937.

(4) Ten-no, S.; Hirata, F.; Kato, S. *Chem. Phys. Lett.* **1993**, *214*, 391–396.

(5) Sato, H.; Hirata, F.; Kato, S. *J. Chem. Phys.* **1996**, *105*, 1546–1551.

(6) Yokogawa, D.; Sato, H.; Sakaki, S. *J. Chem. Phys.* **2007**, *126*, 244504.

(7) Singer, S. J.; Chandler, D. *Mol. Phys.* **1985**, *55*, 621–625.

(8) Ten-no, S. *J. Chem. Phys.* **2001**, *115*, 3724–3731.

(9) Karino, Y.; Fedorov, M. V.; Matubayasi, N. *Chem. Phys. Lett.* **2010**, *496*, 351–355.

(10) Palmer, D. S.; Sergiievskyi, V. P.; Jensen, F.; Fedorov, M. V. *J. Chem. Phys.* **2010**, *133*, 044104.

(11) Ratkova, E. L.; Chuev, G. N.; Sergiievskyi, V. P.; Fedorov, M. V. *J. Phys. Chem. B* **2010**, *114*, 12068–12079.

(12) Ratkova, E. L.; Fedorov, M. V. *J. Chem. Theory Comput.* **2011**, *7*, 1450–1457.

(13) Sergiievskyi, V. P. *Russian Journal of Physical Chemistry B* **2011**, *5*, 326–331.

(14) Chandler, D.; Mccoy, J. D.; Singer, S. J. *J. Chem. Phys.* **1986**, *85*, 5971–5976.

(15) Beglov, D.; Roux, B. *J. Chem. Phys.* **1995**, *103*, 360–364.

(16) Imai, T. *Condens. Matter Phys.* **2007**, *10*, 343–361.

(17) Yokogawa, D.; Sato, H.; Imai, T.; Sakaki, S. *J. Chem. Phys.* **2009**, *130*, 064111.

(18) Luchko, T.; Gusarov, S.; Roe, D. R.; Simmerling, C.; Case, D. A.; Tuszynski, J.; Kovalenko, A. *J. Chem. Theory Comput.* **2010**, *6*, 607–624.

(19) Stumpe, M. C.; Blinov, N.; Wishart, D.; Kovalenko, A.; Pande, V. S. *J. Phys. Chem. B* **2011**, *115*, 319–328.

(20) Palmer, D. S.; Chuev, G. N.; Ratkova, E. L.; Fedorov, M. V. *Current Pharmaceutical Design* **2011**, *17*, 1695–1708.

(21) Frolov, A. I.; Ratkova, E. L.; Palmer, D. S.; Fedorov, M. V. *The Journal of Physical Chemistry B* **2011**, *115*, 6011–6022.

(22) Monson, P. A.; Morriss, G. P. *Adv. Chem. Phys.* **1990**, *77*, 451–550.

(23) Zerah, G. *J. Comput. Phys.* **1985**, *61*, 280–285.

(24) Booth, M. J.; Schlijper, A.; Scales, L.; Haymet, A. *Comput. Phys. Commun.* **1999**, *119*, 122–134.

(25) Kawata, M.; Cortis, C. M.; Friesner, R. A. *J. Chem. Phys.* **1998**, *108*, 4426–4438.

(26) Kovalenko, A.; Ten-No, S.; Hirata, F. *J. Comput. Chem.* **1999**, *20*, 928–936.

(27) Homeier, H.; Rast, S.; Krienke, H. *Comput. Phys. Commun.* **1995**, *92*, 188–202.

(28) Gillan, M. J. *Mol. Phys.* **1979**, *38*, 1781–1794.

(29) Labik, S.; Malijevsky, A.; Vonka, P. *Mol. Phys.* **1985**, *56*, 709–715.

(30) Woelki, S.; Kohler, H.; Krienke, H.; Schmeer, G. *Phys. Chem. Chem. Phys.* **2008**, *10*, 898–910.

(31) Chuev, G. N.; Fedorov, M. V. *J. Comput. Chem.* **2004**, *25*, 1369–1377.

(32) Chuev, G. N.; Fedorov, M. V. *J. Chem. Phys.* **2004**, *120*, 1191–1196.

(33) Fedorov, M. V.; Chuev, G. N. *J. Mol. Liq.* **2005**, *120*, 159–162.

(34) Fedorov, M. V.; Flad, H. J.; Chuev, G. N.; Grasedyck, L.; Khoromskij, B. N. *Computing* **2007**, *80*, 47–73.

(35) Hackbusch, W. *Multi-grid methods and Applications*; Springer-Verlag, Berlin, 1985.

(36) Wang, H. Y.; Jiang, W.; Wang, Y. N. *Plasma Sources Science & Technology* **2010**, *19*, 045023.

(37) Heiskanen, M.; Torsti, T.; Puska, M.; Nieminen, R. *Phys. Rev. B* **2001**, *63*, 245106.

(38) Janke, W.; Sauer, T. *Phys. Rev. E* **1994**, *49*, 3475–3479.

(39) Gygi, F.; Galli, G. *Phys. Rev. B* **1995**, *52*, R2229–R2232.

(40) Kelley, C. T.; Pettitt, B. M. *J. Comput. Phys.* **2004**, *197*, 491–501.

(41) Marucho, M.; Kelley, C. T.; Pettitt, B. M. *J. Chem. Theory Comput.* **2008**, *4*, 385–396.

(42) Sergiievskyi, V. P.; Hackbusch, W.; Fedorov, M. V. *J. Comput. Chem.* **2011**, *32*, 1982–1992.

(43) Kovalenko, A.; Hirata, F. *Chem. Phys. Lett.* **1998**, *290*, 237–244.

(44) Kovalenko, A.; Hirata, F. *J. Chem. Phys.* **2000**, *112*, 10403–10417.

(45) Kovalenko, A.; Truong, T. N. *J. Chem. Phys.* **2000**, *113*, 7458–7470.

(46) Kovalenko, A.; Hirata, F. *J. Chem. Phys.* **2000**, *113*, 2793–2805.

(47) Harano, Y.; Imai, T.; Kovalenko, A.; Kinoshita, M.; Hirata, F. *J. Chem. Phys.* **2001**, *114*, 9506–9511.

(48) Kovalenko, A.; Hirata, F. *J. Phys. Chem. B* **1999**, *103*, 7942–7957.

(49) Fedorov, M. V.; Kornyshev, A. A. *Mol. Phys.* **2007**, *105*, 1–16.

(50) Perkyns, J. S.; Lynch, G. C.; Howard, J. J.; Pettitt, B. M. *J. Chem. Phys.* **2010**, *132*, 064106.

(51) Ng, K. C. *J. Chem. Phys.* **1974**, *61*, 2680–2689.

(52) Kovalenko, A.; Hirata, F. *J. Chem. Phys.* **1999**, *110*, 10095–10112.

(53) Pulay, P. *Chem. Phys. Lett.* **1980**, *73*, 393–398.

(54) Hirata, F., Ed. *Molecular theory of solvation*; Kluwer Academic Publishers, Dordrecht, Netherlands, 2003.

(55) Jorgensen, W. L.; Maxwell, D. S.; Tirado-Rives, J. *J. Am. Chem. Soc.* **1996**, *118*, 11225–11236.

(56) Frigo, M. *Acm Sigplan Notices* **1999**, *34*, 169–180.

(57) Wang, J. M.; Wang, W.; Kollman, P. A.; Case, D. A. *Journal of Molecular Graphics & Modelling* **2006**, *25*, 247–260.

(58) Case, D. A.; Cheatham, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. *J. Comput. Chem.* **2005**, *26*, 1668–1688.

(59) Dewar, M. J. S.; Zoebisch, E. G.; Healy, E. F.; Stewart, J. J. P. *J. Am. Chem. Soc.* **1985**, *107*, 3902–3909.

(60) Jakalian, A.; Bush, B. L.; Jack, D. B.; Bayly, C. I. *J. Comput. Chem.* **2000**, *21*, 132–146.

(61) Jakalian, A.; Jack, D. B.; Bayly, C. I. *J. Comput. Chem.* **2002**, *23*, 1623–1641.

(62) Wang, J. M.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. *J. Comput. Chem.* **2004**, *25*, 1157–1174.

(63) Chandler, D.; Singh, Y.; Richardson, D. M. *J. Chem. Phys.* **1984**, *81*, 1975–1982.

(64) Imai, T.; Kinoshita, M.; Hirata, F. *Bull. Chem. Soc. Jpn.* **2000**, *73*, 1113–1122.

(65) Imai, T.; Harano, Y.; Kovalenko, A.; Hirata, F. *Biopolymers* **2001**, *59*, 512–519.

(66) Guthrie, J. P. *J. Phys. Chem. B* **2009**, *113*, 4501–4507.

(67) Udier-Blagovic, M.; De Tirado, P. M.; Pearlman, S. A.; Jorgensen, W. L. *J. Comput. Chem.* **2004**, *25*, 1322–1332.

(68) Shirts, M. R.; Pande, V. S. *J. Chem. Phys.* **2005**, *122*, 134508.

(69) Shivakumar, D.; Williams, J.; Wu, Y. J.; Damm, W.; Shelley, J.; Sherman, W. *J. Chem. Theory Comput.* **2010**, *6*, 1509–1519.

(70) Paluch, A. S.; Shah, J. K.; Maginn, E. J. *J. Chem. Theory Comput.* **2011**, *7*, 1394–1403.

(71) Matubayasi, N.; Nakahara, M. *J. Chem. Phys.* **2000**, *113*, 6070–6081.

(72) Matubayasi, N.; Nakahara, M. *J. Chem. Phys.* **2002**, *117*, 3605–3616.

(73) Matubayasi, N.; Nakahara, M. *J. Chem. Phys.* **2003**, *119*, 9686–9702.

(74) Matubayasi, N. *Frontiers in Bioscience* **2009**, *14*, 3536–3549.