A reference implementation of the adaptive resolution scheme in ESPResSo

Christoph Junghans *, Simón Poblete

Max Planck Institute for Polymer Research, Ackermannweg 10, 55128 Mainz, Germany

ARTICLE INFO

ABSTRACT

We review the adaptive resolution scheme (AdResS) from a technical perspective and collect arguments from several years of research, which culminates into the implementation of AdResS into the open-source package ESPResSo. This flexible implementation allows us to repeat all previous AdResS simulations with one program. We test this reference implementation and resimulate some results of the well-studied tetrahedral fluid from various previous studies to show the functionality of the package.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Even with constantly growing computer power, atomic scale molecular dynamics simulations can barely reach the time and length scales that are important for "real world" applications. Hence, coarse-graining, which reduces the number of the degrees of freedom, has been a hot topic for years [1,2]. One advantage of coarse-graining is of course to save a lot of computational resources, which usually go into the simulation of unimportant (and unwanted) degrees of freedom, mostly the surrounding solvent. Another aspect is to understand nature on a simplified level, which is reduced to the essential properties. There are several adaptive scale bridging methods [3–5], which enable one to couple an atomistic system to a coarse-grained one. An adaptive method provides the possibility of free particle exchange between the subsystems. This allows one to overcome the problem of sampling unimportant information by keeping the high resolution in one part of the simulation box and using a lower resolution in the rest.

The AdResS method [3] is one of these scale-bridging methods used in molecular simulations and soft matter studies. So far it has been applied to dense model liquids [6] and liquid TIP3P water [7, 8] and has been used to couple atomistic to continuum simulations via a coarse-grained level, thus combining three scales [9]. The fundamental theoretical basis is built using fractional calculus [10, 11]. The aim of this paper is to explain the connection between the fundamental theory and the technical details of this reference implementation into ESPResSo [12], and to provide the reader with knowledgeable details when setting up a simulation. Some useful commands in the scripting language of ESPResSo (TCL) will be ex-
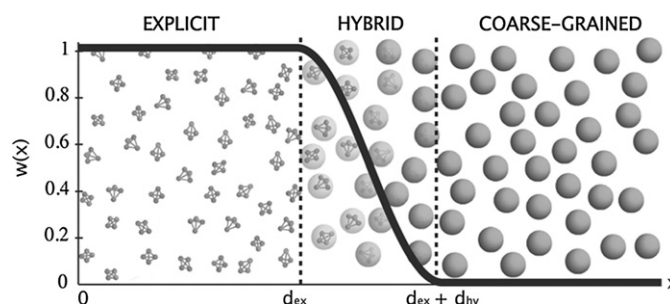

Fig. 1. A schematic illustration of the AdResS method for a tetrahedral liquid. In this figure the $\cos^2$ weighting function is shown.

plained briefly and will allow the reader to directly use AdResS for their own purposes due to the fact that ESPResSo is available under public license [13].

2. Method

This section gives a brief overview of the theory behind the AdResS method to allow consistent naming in the equations. We will review the necessary parts from various research papers. However, for details we refer the reader to the original papers. We also discuss the basics of additional algorithms needed to run AdResS simulations.

2.1. AdResS

The adaptive resolution scheme couples two systems with different resolutions by a force interpolation scheme. Below we discuss this for a coarse-grained and an atomistic representation of the same system, see Fig. 1 for illustration.

* Corresponding author.
E-mail address: junghans@mpip-mainz.mpg.de (C. Junghans).

Every molecule needs a well-defined mapping point in both representations, usually the center of mass, but any other linear combination of particle coordinates is also sufficient.

The force between two molecules is given by [3][1]:

$$\boldsymbol{F}_{\alpha\beta} = w_\alpha w_\beta \boldsymbol{F}^{\mathrm{ex,mol}}_{\alpha\beta} + [1 - w_\alpha w_\beta]\boldsymbol{F}^{\mathrm{cg,mol}}_{\alpha\beta}, \tag{1}$$

where α and β label the two molecules and w_α, w_β are the adaptive weights of the two molecules, which are a function of the position of their mapping points. In the next section we will show how to distribute these forces between molecules to their atoms using virtual sites. The first part, which results from the explicit interaction of the molecules, can be written as:

$$\boldsymbol{F}^{\mathrm{ex,mol}}_{\alpha\beta} = \sum_{\substack{i\ \mathrm{in} \\ \mathrm{mol.}\ \alpha}} \sum_{\substack{j\ \mathrm{in} \\ \mathrm{mol.}\ \beta}} \boldsymbol{F}^{\mathrm{ex}}_{ij}, \tag{2}$$

where $\boldsymbol{F}^{\mathrm{ex}}_{ij}$ is the force between the ith atom in αth molecule and the jth atom in the βth molecule, which is given by an explicit force field. The second part of Eq. (1) comes from the coarse-grained interaction of the molecules. The coarse-grained force field is usually derived from the atomistic system by structure-based and state point coarse-graining [15]. Of course, different coarse-graining methods, such as force-matching [16], are possible as well.

The important element of this interpolation (see Eq. (1)) is the adaptive weight function (for illustration see Fig. 1):

$$w(x) = \begin{cases} 1: & \text{atomistic/explicit region,} \\ 0 < w < 1: & \text{hybrid region,} \\ 0: & \text{coarse-grained region} \end{cases} \tag{3}$$

has a value between 0 and 1. Obviously this *definition of w* gives a purely explicit force in the explicit region and a purely coarse-grained force in the coarse-grained region, so essentially Eq. (1) describes something new for the hybrid region only. Depending on the physical interest of the research, several functions can be implemented as long as the necessary boundary conditions are fulfilled [3]. These conditions are that the function is 1) continuous, 2) monotonic and 3) having zero derivatives at the boundaries.

In most applications a $\cos^2$-like function is used as a weighting function:

$$w(x) = \begin{cases} 0: & x > d_{\mathrm{ex}} + d_{\mathrm{hy}}, \\ \cos^2(\frac{\pi}{2d_{\mathrm{hy}}}(x - d_{\mathrm{ex}})): & d_{\mathrm{ex}} + d_{\mathrm{hy}} > x > d_{\mathrm{ex}}, \\ 1: & d_{\mathrm{ex}} > x, \end{cases} \tag{4}$$

where d_{ex} and d_{hy} are the sizes of the explicit and the hybrid region, respectively.

In the literature both spherical [17] and one-dimensional [6] splitting of the simulation box has been reported and depending on this, the distance x to the center of the explicit region is calculated as follows:

$$x = \begin{cases} |(\boldsymbol{R}_\alpha - \boldsymbol{R}_{\mathrm{ct}}) \cdot \hat{e}|: & \text{splitting in } \hat{e} \text{ direction,} \\ |\boldsymbol{R}_\alpha - \boldsymbol{R}_{\mathrm{ct}}|: & \text{spherical splitting,} \end{cases} \tag{5}$$

where $\boldsymbol{R}_{\mathrm{ct}}$ is the center of the explicit zone, which can be fixed or moved in time (e.g. with the center of a molecule). $\boldsymbol{R}_\alpha$ is the mapping point of the αth molecule. For the center of mass mapping, it is given by:

$$R_\alpha = \frac{\sum_{i\in\alpha} m_i r_i}{\sum_{i\in\alpha} m_i}. \tag{6}$$

Note that the value of the weighting function depends exclusively on the mapping of the molecule.

2.2. Virtual sites

Virtual sites are needed in AdResS due to the fact that there are forces acting on the coarse-grained representation and on the explicit representation simultaneously, which need be to redistributed in the correct way to the explicit representation.

These sites are well-known constructions [18] that allow for interactions between arbitrary points in a molecule. One can imagine these interaction points as virtual atoms, whose movement is determined by a geometrical rule rather than by Newton's equation of motion. In formulas this means the potential between two real atoms is given by:

$$V_{ij} = V^{\mathrm{ex}}(\boldsymbol{r}_i, \boldsymbol{r}_j, \ldots) + V^{\mathrm{vs}}(\boldsymbol{r}^{\mathrm{vs}}(\boldsymbol{r}_i, \boldsymbol{r}_j, \ldots)), \tag{7}$$

where the first part depends on the positions of the real atoms and the second part depends on the position of the virtual site $\boldsymbol{r}^{\mathrm{vs}}$, which is also a function of the positions of the real atoms. The force on the ith atom deriving from this interaction is simply given by:

$$\boldsymbol{F}_i = -\frac{\partial(V^{\mathrm{ex}} + V^{\mathrm{vs}})}{\partial \boldsymbol{r}_i} = \boldsymbol{F}^{\mathrm{ex}}_i + \boldsymbol{F}^{\mathrm{vs}}\frac{\partial \boldsymbol{r}^{\mathrm{vs}}}{\partial \boldsymbol{r}_i}, \tag{8}$$

where the last product is a matrix multiplication. In the case where a virtual site is located in the center of mass, which is the most important case in standard AdResS, Eq. (8) simplifies to:

$$\boldsymbol{F}_i = \boldsymbol{F}^{\mathrm{ex}}_i + \frac{m_i}{\sum_{i\in\alpha} m_i}\boldsymbol{F}^{\mathrm{vs}}. \tag{9}$$

2.3. Interface pressure correction

The interpolation of forces (see Eq. (1)) can produce inhomogeneities in the density and affect the structure and total pressure of the system in comparison to a purely explicit simulation.

One way to reduce such effects in AdResS is to replace the coarse-grained potential $\boldsymbol{F}^{\mathrm{cg,mol}}_{\alpha\beta}$ of Eq. (1) in the hybrid region by a corrected one of the form [17][2]:

$$\boldsymbol{F}^{\mathrm{cg,mol}}_{\alpha\beta} \rightarrow \boldsymbol{F}^{\mathrm{cg,corrected,mol}}_{\alpha\beta} = s(w_\alpha w_\beta)\boldsymbol{F}^{\mathrm{cg,mol}}_{\alpha\beta} \\ + [1 - s(w_\alpha w_\beta)]\boldsymbol{F}^{\mathrm{hy,mol}}_{\alpha\beta}, \tag{10}$$

where $\boldsymbol{F}^{\mathrm{hy,mol}}$ is obtained from the structure and state point-based coarse-graining of a constant-w system with $w = 1/2$ (see Eq. (1)). A common choice for s is

$$s(x) = 4(\sqrt{x} - 1/2)^2, \tag{11}$$

which ensures that the coarse-grained particles ($w = 0$) interact with the usual coarse-grained potential, while hybrid particles with $w = 1/2$ will feel the full, corrected coarse-grained force field $\boldsymbol{F}^{\mathrm{hy,mol}}$. The rest will interact with an interpolated force that varies smoothly with $s(x)$ (see Fig. 2 for illustration). Depending on the system, other smoothing functions and more interpolation points could possibly lead to better results.

2.4. Thermodynamic force

Another way of reducing the inhomogeneities due to the force interpolation is the so-called thermodynamic force [19]. It consists of a space-dependent external field applied in the region of interpolation in order to produce a flat density profile, based on thermodynamic consistency considerations. The force provides the

---

[1] Note that the equation obeys Newton's third law, which is not the case for other interpolation schemes [14].

[2] Again, it is important to note that this interpolation still obeys Newton's third law.
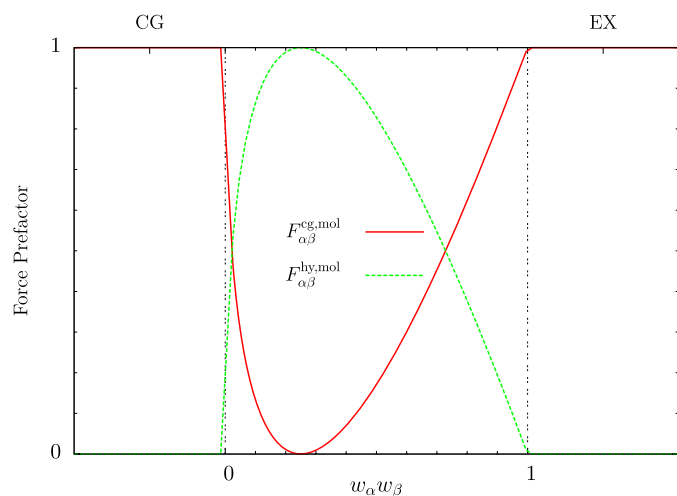
Fig. 2. The force prefactors of the interface pressure correction (see Eq. (10)). Please note that zone in the middle of the plot, $0 < w_\alpha w_\beta < 1$, is not necessarily the hybrid zone, but contains the hybrid zone.

amount of work necessary to compensate the difference in excess chemical potential responsible for the density artifacts. Its effect is exerted on the coarse-grained site of the molecules and can be used to correct local density inhomogeneities in the hybrid region. The force can also be used to couple explicit and coarse-grained regimes of different state points.

3. Implementation

In this section we explain how the method is integrated in ESPResSo [12]. It is important to note that we are not aiming primarily for computational speed but rather for flexibility and modularity, which was the major aim of ESPResSo since the beginning. A fast implementation with a new adaptive integrator will be released in the successor of ESPResSo: ESPResSo++ [20].

3.1. General overview

We implemented the AdResS simulation as an explicit simulation with additional constraints, namely a virtual site in the center of mass of each molecule that represents the coarse-grained particle. Therefore, we have hybrid molecules throughout the simulation box and we integrate only the explicit *representations* of the particles using a velocity Verlet scheme, after we have distributed the forces from their coarse-grained *representations*. The idea of keeping the number of "virtual" particles constant over the whole box is well-known from grandcanonical simulations [21]. On one hand, this makes our code slightly slower, but, on the other hand, one can omit the calculation of the explicit interaction in the coarse-grained region and the calculation of the coarse-grained interaction in the explicit region, which would lead to an increase in the total speed of an adaptive simulation compared to a pure explicit simulation. This *double representation* trick for every molecule saves having to destroy and create particles on the fly, which is a very expensive operation in the sense of memory access and bookkeeping in the case of a parallel simulation. It is important to note that the force calculation is the most time consuming operation in most molecular dynamics simulations, which essentially leads us to this approach. Of course we do not calculate the forces on the particles where it is not needed. By not destroying the explicit particle this implementation favors molecules with only a few particles, so it must be rethought for molecules with more than 25 particles. However, this trick also allows us to perform constant-w simulations easily which are sometimes needed to determine the interface pressure correction potential $F^{hy,mol}$ (see Section 2.3).

The AdResS feature of ESPResSo can be enabled by adding:

```
#define ADRESSO
```

to the ESPResSo configuration file, e.g. `myconfig.h` and the following line in the header of the steering `TCL` script:

```
adress set topo $kind \
        width $width $hybrid_width \
        center x $R_x wf $wf
```

where `kind` determines the type of AdResS simulation:
- 0   disabled
- 1   constant weight function
- 2   one-dimensional splitting
- 3   spherical splitting

`wf` the type of weighting function:
- 0   see Eq. (4)
- 1   user defined

`width` and `hybrid_width` are the widths of the explicit and hybrid regions (see Eq. (4)), respectively and `R_x` is the x position of the center of the explicit zone.

3.2. Integrator

The integration in ESPResSo is done by a standard velocity Verlet integrator with a simple addition to allow *NVT* ensemble simulations by a Langevin [22] or a DPD [23,24] thermostat.

1. $v(t + \Delta t/2) = v(t) + \Delta t/2 \cdot f(t)/m$
2. $p(t + \Delta t) = p(t) + \Delta t\, v(t + \Delta t/2)$
3. Calculate $f(t + \Delta t)$ from $p(t + \Delta t)$, $v(t + \Delta t/2)$
4. $v(t + \Delta t) = v(t + \Delta t/2) + \Delta t/2 \cdot f(t + \Delta t)/m$

The scheme must be extended in the case of an AdResS simulation:

1. $v(t + \Delta t/2) = v(t) + \Delta t/2\, f(t)/m$
2. $p(t + \Delta t) = p(t) + \Delta t\, v(t + \Delta t/2)$
2b. Update the positions, velocities and weighting functions $w(\mathbf{R})$ of the virtual sites (see Eqs. (3) and (6))
3. Calculate $f(t + \Delta t)$ from $p(t + \Delta t)$, $v(t + \Delta t/2)$
3b. Distribute the force of the virtual sites to the real particles (see Eq. (9))
4. $v(t + \Delta t) = v(t + \Delta t/2) + \Delta t/2\, f(t + \Delta t)$

The *double representation* of every molecule allows us to apply the actual integration (steps 1, 2, 3, 4) on the explicit particles only, regardless of their real *representation* given by the weighting function. This is obviously not the most effective method of implementation, but avoids the need to develop a new integrator that can switch the actual calculation between explicit and coarse-grained particles on the fly.

3.3. Parallelization scheme

The ESPResSo package basically divides the simulation box into sub-boxes, which are distributed among different processors. To allow the calculation of interactions across the boundaries of these sub-boxes, every sub-box is surrounded by a so-called ghost layer. This layer contains copies of the particles, which are in a different sub-box (and so on a different processor), and are called ghost particles. These ghosts are updated after every propagation step of their real counterparts by communication between the neighboring processors. A detailed description can be found in the ESPResSo publication [12].

Only the communication part of the propagator needs to be modified in the presence of virtual sites. Essentially, the amount of

communication *doubles* due to the fact that the positions of all the particles (including the ghost particles) have to be updated to calculate the positions of the virtual sites (first communication) and the position of the virtual sites has to be distributed again (second communication).

The same effect comes into play when the forces are collected from all the copies of a particle (ghosts and real particles). To distribute the force from a virtual site one needs to sum up its forces first (first communication) and then distribute them, but as the forces are possibly distributed to ghost particles one will have to collect the forces again (second communication). It is clear that in the absence of virtual sites the second communication is not necessary.

3.4. Thermostat

ESPResSo provides several stochastic thermostats, which are of a local nature and can be extended to the AdResS scheme. We will discuss all the details for the Langevin thermostat [22], but they also apply for the DPD thermostat [23,24]. We want to obtain a canonical distribution of the velocities of the atoms in the hybrid and explicit regions and of the centers of mass in the whole system. This requires the proper application of the thermostat on each region with a consistent initialization of the atoms' velocities whenever a molecule crosses from the coarse-grained to the hybrid regime. While we have chosen to apply the Langevin thermostat over the complete box on the explicit particles, independent of their nature, there are two tested ways of initializing the explicit velocities when the additional degrees of freedom are introduced:

- Copy the atoms' velocities relative to the center of mass of a molecule from a random explicit molecule when a molecule crosses the border to the hybrid region. This method was used in the original description of AdResS [3].
- The thermostat acts on the explicit *representation* of the particle everywhere, regardless if they are in the coarse-grained region or not. In this case the initialization does not matter, since the atoms are already thermalized. It is clear that the coarse-grained particles (technically virtual sites at the centers of mass) will automatically have the correct velocity distribution, if all the explicit particles have the correct distribution.[3]

The main difference between these two approaches is that the second case is much easier to parallelize, since less communication is involved. Under both these schemes the temperature and density profiles are unchanged and the structural properties remain intact. Thus, the correct distribution of velocities is guaranteed.

The thermostat can be set up as follows:

```
thermostat langevin $temperature $friction
```

or

```
thermostat dpd $temperature $friction $cutoff
```

3.5. Intramolecular initialization

For the initialization of the molecule's orientation and internal bond-lengths when crossing from the coarse-grained to the hybrid regime, we follow the same procedure as for the velocities in the

previous subsection, that is, we assign relative positions by selecting a random explicit particle or, alternatively keep the explicit *representations* that come from the coarse-grained region, where the atoms feel no intermolecular interaction. In both cases, the behavior is similar, but in the second approach the results are much simpler to implement and faster during computation.

3.6. Bonded interactions

The intra-molecular interactions are not described by the general interpolation equation (see Eq. (1)), but are fully included along the whole simulation box. For rigid molecules such as water there is no explanation needed, however, for flexible bonds, the bonded interactions are still calculated across the whole simulation box without any interpolation due to stability reasons. It is clear that this only works for coarse-grained models with one coarse-grained site per molecule. However, for more complicated structures where bonded interactions are also present in the low resolution regime, the force interpolation has to be rethought.

3.7. Cut-offs

As the representation (explicit, hybrid or coarse-grained) of the molecule is determined on the basis of the position of the center of mass of the molecule it is natural to use a cut-off based on the distance between the centers of mass. This strategy is well known [25] in water simulations with Reaction-Field to avoid artificial dipole moments at the cut-off. In other words, a molecular cut-off ensures that either all particles or no particles of a molecule interact.

We have implemented the molecular cut-off as a virtual interaction:

```
inter $type1 $type2 molcut 1 $cut-off
```

where `type1` and `type2` are the particle types. Internally the distance between the centers of mass of the molecules to which these particle types belong is used. All other individual interaction cut-offs stay active and have to be increased to a value higher than the molecular cut-off in the case that one wants to simulate a system with molecular cut-offs only.

3.8. Molecules and mappings

The basic support for molecules comes with the standard version of ESPResSo. The user is allowed to combine a set of particles into a molecule. We use this feature to equip every molecule with one virtual site to define the set of atoms from which the center of mass is calculated. The extension to center of geometry or multiple virtual sites in one molecule is in preparation.

A single tetrahedral molecule can be set up with:

```
set molecule_topology $molecule_type
part 1 pos $pos1x $pos1y $pos1z virtual 0
lappend molecule_topology 1
part 2 pos $pos2x $pos2y $pos2z virtual 0
lappend molecule_topology 2
part 3 pos $pos3x $pos3y $pos3z virtual 0
lappend molecule_topology 3
part 4 pos $pos4x $pos4y $pos4z virtual 0
lappend molecule_topology 4
part 5 pos $pos5x $pos5y $pos5z virtual 1
lappend molecule_topology 5
eval analyze set $molecule_topology
analyze set topo_part_sync
```

---

[3] For the center of mass of a molecule, one just has to add up the individual Langevin thermostats of the atoms of the molecule and it follows that this "summed thermostat" obeys the dissipation–fluctuation theorem itself, if the friction of the explicit particles is scaled with their masses.

and the position of the virtual site (in this case particle number 5) can be set to the right position by calling

```
integrate 0
```

From now on

```
integrate $steps
```

will integrate the whole system for steps steps and move the virtual sites accordingly.

3.9. Interface pressure correction

The interface correction is introduced as a new kind of non-bonded tabulated interaction.

```
inter $type1 $type2 adress_tab_ic $filename
```

where type1 and type2 are the types of particles subjected to the coarse-grained interaction that will be modified. They must be virtual particles. filename contains the tabulated fields corresponding to the interactions $F_{\alpha\beta}^{cg,mol}$ and $F_{\alpha\beta}^{hy,mol}$ according to Eq. (10). The syntax is consistent with the usual tabulated interactions, the first four lines being the special character #, followed by the number of points N and the minimum and maximum separation distances r_{min} and r_{max}. After this information, the two potentials are introduced in five columns, as r, $F_{\alpha\beta}^{cg,mol}/r$, $U_{\alpha\beta}^{cg,mol}$, $F_{\alpha\beta}^{hy,mol}/r$, $U_{\alpha\beta}^{hy,mol}$. The number of points and the cut-off radius are assumed to be the same for both potentials, while the values of r are equally distributed between r_{min} and r_{max} with a fixed distance $(r_{max} - r_{min})/(N - 1)$. The position and potential columns are ignored in the calculations, but are included for readability only.

3.10. Thermodynamic force

This new force is implemented as a tabulated external field that acts just in the switching region. Different kinds of particles can have different thermodynamic forces. The corresponding command is

```
thermodynamic_force $type $filename $prefactor
```

where type is the type of particle on which the tabulated force specified in filename will be exerted. prefactor is a coefficient that multiplies the force defined by the user.

The format of the tabulated external field follows the same definition as the tabulated pair potentials in standard ESPResSo. It starts with four special lines with the character #, the number of points and the minimum and maximum separation distances, r_{min} and r_{max}. Then, three columns should be entered, s, $-U/s$ and U. s corresponds to a dimensionless position x/d_{hy} in the hybrid region, going from 0 (coarse-grained representation) to 1 (explicit representation). The derivative of the potential has to be expressed as a function of this variable. The third column is the potential energy associated with the field; it has no effect on the equations of motion and is included only for readability.

4. Example

Here we limit ourselves to briefly resimulating a few previously published results that exhibit the whole set of features previously described. The tetrahedral molecule [3] is the most suitable case to demonstrate AdResS itself, the interface correction, and the action of thermodynamic forces.
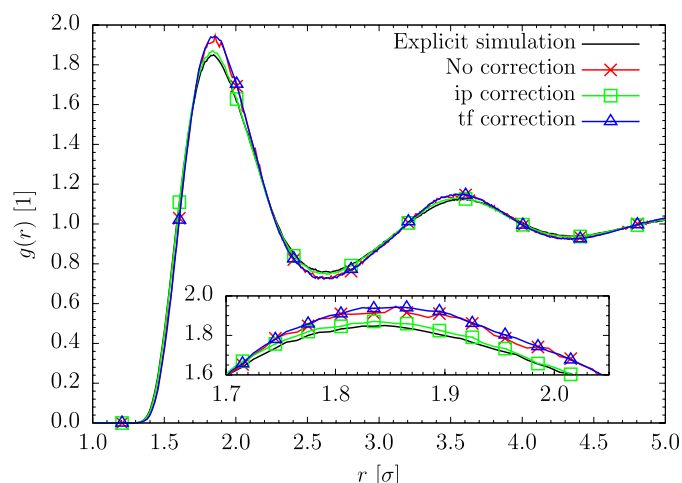
Fig. 3. Radial distribution functions between centers of mass for explicit and AdResS simulations with several corrections. *The inset shows a close-up of the first peak.*

The system is composed of 2520 molecules, each of them consisting of four atoms. Every atom interacts according to a purely-repulsive, shifted 12-6 Lennard-Jones potential

$$U_{LJ}^{ex}(r_{i\alpha j\beta})$$
$$= \begin{cases} 4\epsilon[(\sigma/r_{i\alpha j\beta})^{12} - (\sigma/r_{i\alpha j\beta})^6 + \frac{1}{4}], & r_{i\alpha j\beta} \leqslant 2^{1/6}\sigma, \\ 0, & r_{i\alpha j\beta} > 2^{1/6}\sigma, \end{cases} \quad (12)$$

while the atoms inside a molecule are bonded by an additional attractive FENE potential

$$U_{FENE}^{ex}(r_{i\alpha j\alpha}) = \begin{cases} -\frac{1}{2}kR_0^2 \ln[1 - (r_{i\alpha j\beta}/R_0)^2], & r_{i\alpha j\beta} \leqslant R_0, \\ \infty, & r_{i\alpha j\beta} > R_0, \end{cases} \quad (13)$$

where i and j are the atoms belonging to molecules α and β, respectively. For this case we chose $\sigma = \epsilon = 1$, $k = 30\epsilon/\sigma^2$ and $R_0 = 1.5\sigma$. The box dimensions, in units of the fundamental length σ, are $36 \times 20 \times 20$, splitted in the x direction with $d_{hy} = 12$. The explicit and coarse-grained regions have the same size of 6 reduced units. To show the influence of the force interpolation on the density the hybrid region was made twice the size of the explicit one.

The coarse-grained interaction, on the other hand, consists of a one-site tabulated potential able to reproduce satisfactorily the radial distribution function and the state point of the liquid [3].

For a comparison of the basic structure of the simulations, Fig. 3 shows the radial distribution functions for the centers of mass of the molecules in the explicit and AdResS simulations (normal, interface corrected and under the effect of the thermodynamic force). In Fig. 4 we show the density profiles of the simulations for three cases: (a) simple AdResS interpolation, (b) AdResS with interface-pressure correction and (c) the same system under the effect of the thermodynamic force.[4] A detailed discussion about the density inhomogeneities and their elimination can be found in [19,17].

5. Conclusions and outlook

We have shown how to use AdResS in ESPResSo and have reproduced results from previous simulations. In addition, we ap-

---

[4] The thermodynamic force was determined by calculating the excess chemical potential for several systems at constant weighting function by means of the test particle insertion method [26] in the GROMACS package [18]. The force was obtained as the derivative of a continuous interpolation of this excess chemical potential profile.
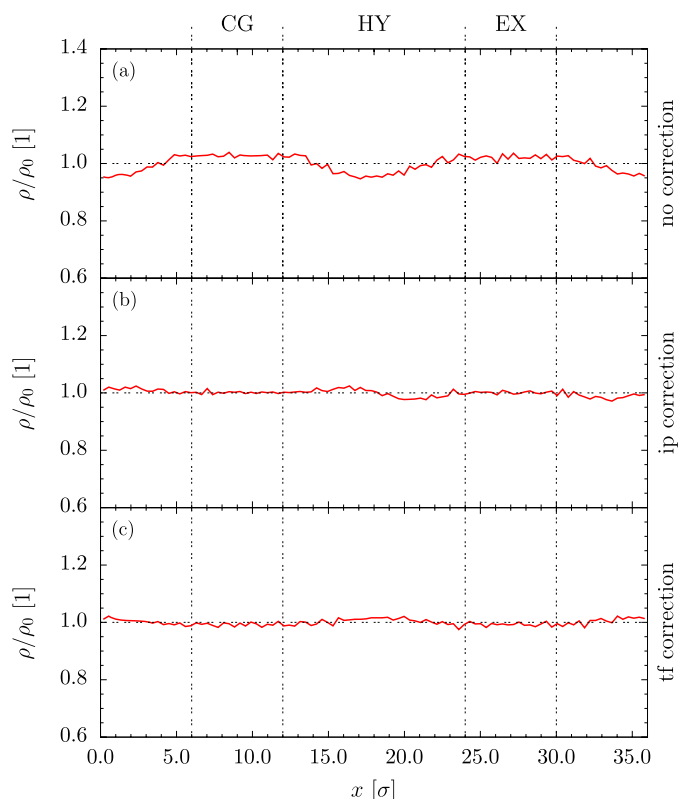
Fig. 4. AdResS simulation of tetrahedral molecule system with a hybrid region width of 12σ with (a) no correction in the hybrid region, (b) interface-pressure (ip) correction and (c) external thermodynamic force (tf) correction.

plied several correction algorithms to obtain a flat density distribution over the entire box. We also discussed several choices in the implementation, which were made based on performance issues, namely the thermostat and the reinitialization of the molecule. A sample AdResS simulation is available in the tutorial of ESPResSo.

The AdResS feature will be released in ESPResSo version 2.2, however ESPResSo++ [20] is also on the way and will replace ESPResSo in the near future. A similar implementation is also available for GROMACS [27].

Acknowledgements

References

[1] G.A. Voth, Coarse-Graining of Condensed Phase and Biomolecular Systems, CRC Press Inc., 2008.

[2] C. Peter, K. Kremer, Multiscale simulation of soft matter systems – from the atomistic to the coarse-grained level and back, Soft Matter 5 (22) (2009) 4357–4366.

[3] M. Praprotnik, L. Delle Site, K. Kremer, Adaptive resolution molecular-dynamics simulation: Changing the degrees of freedom on the fly, The Journal of Chemical Physics 123 (22) (2005) 224106–224114, doi:10.1063/1.2132286.

[4] M. Praprotnik, L. Delle Site, K. Kremer, Multiscale simulation of soft matter: From scale bridging to adaptive resolution, Annual Review of Physical Chemistry 59 (1) (2008) 545–571, doi:10.1146/annurev.physchem.59.032607.093707.

[5] B. Ensing, S.O. Nielsen, P.B. Moore, M.L. Klein, M. Parrinello, Energy conservation in adaptive hybrid atomistic/coarse-grain molecular dynamics, Journal of Chemical Theory and Computation 3 (3) (2007) 1100–1105, doi:10.1021/ct600323n.

[6] M. Praprotnik, L. Delle Site, K. Kremer, Adaptive resolution scheme for efficient hybrid atomistic-mesoscale molecular dynamics simulations of dense liquids, Physical Review E 73 (6) (2006), doi:10.1103/PhysRevE.73.066701.

[7] M. Praprotnik, S. Matysiak, L. Delle Site, K. Kremer, C. Clementi, Adaptive resolution simulation of liquid water, Journal of Physics: Condensed Matter 19 (29) (2007) 292201.

[8] S. Matysiak, C. Clementi, M. Praprotnik, K. Kremer, L. Delle Site, Modeling diffusive dynamics in adaptive resolution simulation of liquid water, The Journal of Chemical Physics 128 (2) (2008) 024503–024512, doi:10.1063/1.2819486.

[9] R. Delgado-Buscalioni, K. Kremer, M. Praprotnik, Concurrent triple-scale simulation of molecular liquids, The Journal of Chemical Physics 128 (11) (2008) 114110–114119, doi:10.1063/1.2890729.

[10] M. Praprotnik, K. Kremer, L. Delle Site, Adaptive molecular resolution via a continuous change of the phase space dimensionality, Physical Review E 75 (1) (2007), doi:10.1103/PhysRevE.75.017701.

[11] M. Praprotnik, K. Kremer, L. Delle Site, Fractional dimensions of phase space variables: A tool for varying the degrees of freedom of a system in a multiscale treatment, Journal of Physics A: Mathematical and Theoretical 40 (15) (2007) F281–F288.

[12] H. Limbach, A. Arnold, B. Mann, C. Holm, ESPResSo – an extensible simulation package for research on soft matter systems, Computer Physics Communications 174 (9) (2006) 704–727, doi:10.1016/j.cpc.2005.10.005.

[13] ESPResSo, http://www.espresso.mpg.de, 2010.

[14] L. Delle Site, Some fundamental problems for an energy-conserving adaptive-resolution molecular dynamics scheme, Physical Review E 76 (4) (2007), doi:10.1103/PhysRevE.76.047701.

[15] D. Reith, M. Pütz, F. Müller-Plathe, Deriving effective mesoscale potentials from atomistic simulations, Journal of Computational Chemistry 24 (13) (2003) 1624–1636, doi:10.1002/jcc.10307.

[16] S. Izvekov, G.A. Voth, Multiscale coarse graining of liquid-state systems, The Journal of Chemical Physics 123 (13) (2005) 134105, doi:10.1063/1.2038787.

[17] M. Praprotnik, L. Delle Site, K. Kremer, A macromolecule in a solvent: Adaptive resolution molecular dynamics simulation, The Journal of Chemical Physics 126 (13) (2007) 134902–134908, doi:10.1063/1.2714540.

[18] B. Hess, C. Kutzner, D. van der Spoel, E. Lindahl, GROMACS 4: Algorithms for highly efficient, Load-Balanced, and scalable molecular simulation, Journal of Chemical Theory and Computation 4 (3) (2008) 435–447, doi:10.1021/ct700301q.

[19] S. Poblete, M. Praprotnik, K. Kremer, L. Delle Site, Coupling different levels of resolution in molecular simulations, The Journal of Chemical Physics 123 (2010) 114101, arXiv:0907.5497.

[20] ESPResSo++, http://www.espresso-pp.de/, 2010.

[21] L.A. Rowley, D. Nicholson, N.G. Parsonage, Monte Carlo grand canonical ensemble calculation in a gas–liquid transition region for 12-6 argon, Journal of Computational Physics 17 (4) (1975) 401–414, doi:10.1016/0021-9991(75)90042-X.

[22] B. Dünweg, Langevin methods, in: B. Dünweg, A.I. Milchev, D.P. Landau (Eds.), Computer Simulations of Surfaces and Interfaces: Proceedings of the NATO Advanced Study Institute, Albena, Bulgaria, 9–20 September 2002, 1st ed., Springer, Netherlands, 2004, pp. 77–92.

[23] T. Soddemann, B. Dünweg, K. Kremer, Dissipative particle dynamics: A useful thermostat for equilibrium and nonequilibrium molecular dynamics simulations, Physical Review E 68 (4) (2003), doi:10.1103/PhysRevE.68.046702.

[24] C. Junghans, M. Praprotnik, K. Kremer, Transport properties controlled by a thermostat: An extended dissipative particle dynamics thermostat, Soft Matter 4 (1) (2008) 156–161.

[25] T.N. Heinz, W.F. van Gunsteren, P.H. Hunenberger, Comparison of four methods to compute the dielectric permittivity of liquids from molecular dynamics simulations, The Journal of Chemical Physics 115 (3) (2001) 1125–1136, doi:10.1063/1.1379764.

[26] D. Frenkel, B. Smit, Understanding Molecular Simulation: From Algorithms to Applications, 2nd ed., Academic Press, 2001.

[27] B. Lambeth, S. Fritsch, C. Junghans, AdressMacs, http://repo.or.cz/w/gromacs/adressmacs.git, 2010.