Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 221 (2007) 303–329

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Zonal methods for the parallel execution of range-limited *N*-body simulations

Kevin J. Bowers [a], Ron O. Dror [a], David E. Shaw [a,b,*]

[a] *D. E. Shaw Research, LLC, New York, NY 10036, United States*
[b] *Center for Computational Biology and Bioinformatics, Columbia University, New York, NY 10032, United States*

Received 14 January 2006; received in revised form 2 May 2006; accepted 6 June 2006
Available online 2 August 2006

Abstract

Particle simulations in fields ranging from biochemistry to astrophysics require the evaluation of interactions between all pairs of particles separated by less than some fixed interaction radius. The applicability of such simulations is often limited by the time required for calculation, but the use of massive parallelism to accelerate these computations is typically limited by inter-processor communication requirements. Recently, Snir [M. Snir, A note on N-body computations with cutoffs, Theor. Comput. Syst. 37 (2004) 295–318] and Shaw [D.E. Shaw, A fast, scalable method for the parallel evaluation of distance-limited pairwise particle interactions, J. Comput. Chem. 26 (2005) 1318–1328] independently introduced two distinct methods that offer asymptotic reductions in the amount of data transferred between processors. In the present paper, we show that these schemes represent special cases of a more general class of methods, and introduce several new algorithms in this class that offer practical advantages over all previously described methods for a wide range of problem parameters. We also show that several of these algorithms approach an approximate lower bound on inter-processor data transfer.

Keywords: Molecular simulation; Molecular dynamics; Parallel computing; *N*-body problem; Pairwise particle interactions

1. Introduction

Simulations in many fields require the explicit evaluation of interactions between all pairs of particles separated by less than some interaction radius R (*near interactions*). Examples of such *range-limited N-body problems* include molecular dynamics simulations in biochemistry and materials science, gravitational simulations in astrophysics, particle simulations in plasma physics, and smooth particle hydrodynamic simulations in fluid dynamics [3–11]. The remaining pairwise interactions are either neglected or approximated using one of several less expensive methods [3,12–17]. The computation of near interactions is often the dominant cost in such

[*] Corresponding author.
E-mail address: david@deshaw.com (D. E. Shaw).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.06.014

simulations, so efficient methods for parallelizing this computation are critical to accelerating these simulations. Although near interactions can be evaluated in parallel, the communication required to bring particle pairs together on the same processor can severely limit scalability.

Traditional methods for parallelizing range-limited N-body problems are described in several review papers [18,19]. They include atom, force, and spatial decomposition methods. Spatial decomposition methods offer an advantage over atom and force decomposition methods in that the amount of data to be transferred into and out of each processor (the method's *communication bandwidth*) decreases as the interaction radius decreases. Force decomposition methods, on the other hand, offer an advantage over spatial and atom decomposition methods in that the communication bandwidth decreases as the number of processors increases.

Two independently developed and recently published methods – Snir's hybrid method (referred to here as the *SH Method*) [1] and Shaw's *NT* (for "Neutral Territory") *method* [2] – combine the advantages of traditional spatial and force decomposition methods. These newer algorithms have the property that the communication bandwidth decreases both as the interaction radius decreases and as the number of processors increases. Table 1 compares the asymptotic scaling properties of the SH and NT methods with those of traditional methods. The SH and NT methods have the same scaling properties, which are superior to those of traditional methods. The asymptotic scaling properties of various parallelization methods will be discussed in more detail in Sections 5 and 6.

It has been shown [2] that both the SH and NT methods offer significant advantages over traditional methods for practical machine and simulation sizes. Furthermore, the NT method always requires less communication than the SH method, but the SH method can be optimized using ideas from the NT method such that the difference is small. The NT-optimized version of the SH method is referred to as the *SNT Method* [2].

A given processor will compute a significantly different set of near interactions under the NT method than it would under the SH method. The two methods, however, do share several essential elements. In both the NT method and the SH method, each processor assumes primary responsibility for all particles falling within some rectangular box. In both methods, the processor that computes the interaction between a pair of particles is typically not the one on which either particle resides, but a third processor that imports both of these particles. In both methods, each processor interacts particles residing in one region of space with particles residing in another region, after importing some of the particles in each region from other processors. This suggests that the two methods may be related and raises the question of whether other new methods exhibiting similar properties might have advantages over both methods for various choices of problem parameters.

In this paper, we describe a new class of parallelization methods, which we refer to as *zonal methods*. This class includes as special cases the NT method, the SH method, and the SNT method, along with traditional spatial decomposition methods and a number of novel methods introduced in this paper. We refer to a parallelization method for range-limited N-body problems as a *generalized spatial decomposition method* if each processor assumes responsibility for updating the positions of particles in a distinct region of space, regardless of where the particle-particle interactions are computed. Generalized spatial decomposition methods can be divided into *home territory methods*, in which a pair of particles always interacts on the processor on which one of them resides, and *neutral territory methods*, in which a pair of particles may interact on a processor in which neither of them resides. All zonal methods are generalized spatial decomposition methods, but some (for example, traditional spatial decomposition methods) are home territory methods, while others (including the NT, SH, and SNT methods, as well as the new methods introduced in this paper) are neutral territory methods.

Table 1
Asymptotic scaling properties of communication bandwidth required by various parallelization methods

	Exploitable range limitation	Scaling with number of processors
Atom decomposition methods	None	No scaling
Force decomposition methods	None	O($p^{-1/2}$) scaling
Spatial decomposition methods	O(R^3) neighbors	No scaling
NT and SH methods	O($R^{3/2}$) neighbors	O($p^{-1/2}$) scaling

R is the interaction radius and p is the number of processors.

Which parallelization method minimizes inter-processor communication bandwidth depends on the properties of both the computation to be performed (e.g., the interaction radius and the dimensions of the physical system) and the hardware available for the simulation (e.g., the number of nodes and the network topology). In this paper, we analyze the communication requirements of a variety of zonal methods. We describe new methods that require less communication bandwidth than any previously published methods for a wide range of such parameter values, including those associated with many practical systems. In some cases, these new methods represent minor variations on the NT and SNT methods; in others, they constitute entirely novel methods.

Section 2 reviews the NT and SH methods, as well as a traditional spatial decomposition method that we call the *HS Method*. Sections 3 and 4 describe the general class of zonal methods, and introduce certain tools that prove useful for their analysis. In Section 3, we assume that each processor will interact particles from a single spatial region with particles from another single spatial region. We describe the pairs of regions one can choose to ensure that all near interactions are computed. We also present a systematic process to take advantage of the fact that each particle interacts with other particles that reside within a surrounding sphere, rather than a cube; we refer to this process as *rounding*. The HS, NT, and SNT methods already exploit a form of rounding, but some methods support a stronger form that we refer to as *three-dimensional rounding*.

Section 4 generalizes the ideas of Section 3 by considering *k-zone methods* that specify a set of spatial regions along with a schedule that determines the order in which each processor must compute interactions between particles from various pairs of regions. Such methods can further reduce communication requirements and can increase the extent to which communication and computation are performed simultaneously.

In Section 5, we derive lower bounds on the amount of data to be communicated by any zonal method. These bounds, which depend particularly on the interaction radius and on the number of processors available, guide the design of novel parallelization methods. In Section 6, we specify several methods, each of which has the lowest communication bandwidth requirement of any published method of which we are aware for some set of practical parameter settings. While we do not prove these methods to be optimal, we show that they come close to an approximate lower bound on communication bandwidth requirements.

2. Existing parallelization methods and terminology

We refer to the region containing the system to be simulated as the *global cell*. Zonal methods require that the global cell be divided into a lattice of identically shaped regions, with each processor assuming primary responsibility for updating the coordinates of all particles in that subregion. To simplify our exposition, we will assume in this paper that the global cell is a rectangular parallelepiped of dimensions $G_x \times G_y \times G_z$ and that it is divided into a regular, three-dimensional grid of smaller rectangular parallelepipeds that we call *boxes*. Each processor updates the coordinates of particles in one box, referred to as the *home box* of that processor, of those particles, and of any point in the box. In the interest of simplicity, we will refer interchangeably to a processor and its home box. The dimensions of each box are $b_x \times b_y \times b_z$. We refer to the quantities b_x/b_y and b_x/b_z as the *box aspect ratios*. The *base coordinates* of a given box (and of any particle located within that box) are defined as the coordinates of the low-coordinate corner of that box. We will assume that the low-coordinate corner of the global cell has coordinates (0, 0 ,0). If a box has base coordinates (c_x, c_y, c_z), we define the coordinates $(c_x/b_x, c_y/b_y, c_z/b_z)$ as the *logical base coordinates* of that box and of any particle located within that box.

We will assume that the global cell tiles an infinite space by repeating in each dimension with a period equal to the side length of the global cell in that dimension. The periodic boundary conditions imposed by this assumption simplify our exposition, but the methods discussed in this paper are also applicable to systems with other boundary conditions. We will also assume for simplicity that $G_x \geqslant b_x + 2R$, $G_y \geqslant b_y + 2R$, and $G_z \geqslant b_z + 2R$, so that at most one image of a given particle interacts with any particle in a given box.[1]

[1] While the base coordinates (c_x, c_y, c_z) of a given box will satisfy $c_x \in [0, G_x]$, $c_y \in [0, G_y]$, and $c_z \in [0, G_z]$, the coordinates of an imported particle may fall outside those ranges, indicating that the particle was imported from a different image of the global cell. When specifying the box on which a pair of particles will interact, we treat corresponding particles in different images of the global cell as separate particles with different coordinates.

This section describes the NT and SH methods as well as an example of a traditional spatial decomposition method, the HS (for "Half-Shell") method. These methods can each be described using two spatial regions called *zones*. Each method interacts all particles in one zone with all particles in the other zone (subject to a local filtering process to be described later). The location of each zone is specified relative to the box in which these interactions take place (the *interaction box*), and each box has the same spatial relationship with its zones as each other box. For each of these three methods, both zones include the interaction box itself, but are otherwise non-intersecting.

We refer to particles that reside within a particular box as *local* to that box, and to other particles as *remote* from that box. Similarly, we refer to interactions between particles that have the same home box as *local interactions*, and to interactions between particles with different home boxes as *remote interactions*.

We refer to the region of space from which a given processor "imports" data that ordinarily resides within other processors as its *import region*, and to the volume of its import region as its *import volume* (V_i). If particle density is uniform, the amount of particle data that must be transferred into each processor is proportional to its import volume. Particle density is approximately uniform in many particle simulations – e.g., in explicit solvent molecular dynamics and other kinds of condensed matter or fluid simulations – so we will use import volume as a measure of communication bandwidth requirements.

For the parallelization methods discussed in this paper, the shape of the import region depends only on the ratios of the box side lengths to the interaction radius R. In three dimensions, the ratio of the import volume V_i to the box volume V_b for a particular method can be determined uniquely given R/b_x, R/b_y, and R/b_z. Alternatively, V_i/V_b can be expressed as a function of the box aspect ratios b_x/b_y and b_x/b_z and the *parallelization parameter* α_R, where we define α_R as the geometric mean of R/b_x, R/b_y, and R/b_z:

$$\alpha_R = \left(\frac{R}{b_x}\frac{R}{b_y}\frac{R}{b_z}\right)^{1/3} = \left(\frac{R^3}{V_b}\right)^{1/3} = \frac{R}{V_b^{1/3}}.$$

For cubic boxes of side length b, α_R is simply R/b. The parallelization parameter α_R might be viewed as a measure of the extent to which a particular simulation has been parallelized. For convenience, we also define the *normalized box side lengths* $\alpha_x = b_x/V_b^{1/3}$, $\alpha_y = b_y/V_b^{1/3}$, and $\alpha_z = b_z/V_b^{1/3}$, which can be uniquely determined from the box aspect ratios.

When interacting a pair of particles a and b, we may wish to compute either a single quantity summarizing the interaction (e.g., an interaction energy) or one quantity for each of the two particles (e.g., the force exerted on each). The former scenario may be viewed as a special case of the latter, as we can assign a single quantity, or a scaled version of that quantity, to both particles. For the sake of generality, we will assume throughout this paper that we wish to compute one quantity for each of the two particles, and we will refer to the quantity associated with particle a as the *influence* of b on a and to the quantity associated with particle b as the influence of a on b.[2] In the HS, NT, and SH methods, a pair of particles always interacts in a single processor that computes the influence of each particle on the other.

2.1. HS method

In traditional spatial decompositions, two particles always interact within a box in which at least one of them resides. In the HS method, the particles interact within the home box of the particle with the smaller x base coordinate. If the two particles have the same x base coordinate, the particles interact within the home box with the smaller y base coordinate. If they also have the same y base coordinate, they interact within the home box with the smaller z base coordinate. If the two particles reside within the same box, they interact within that box.

As a result, each interaction box has an import region that consists of half the points external to the interaction box that lie within a distance R of the interaction box. Fig. 1 shows this import region in red and the interaction box itself in purple. Every particle in the interaction box interacts with every imported particle, and

[2] Variants of the methods described in this paper are also applicable to cases where one wishes to compute the influence of one set of particles on another set of particles, but not vice versa – for example, if one wishes to map charge from atoms to mesh points.

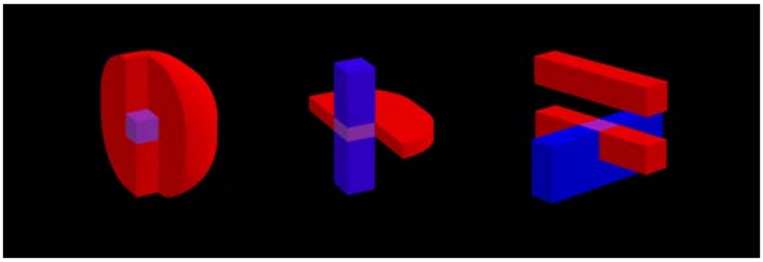


Fig. 1. Zones associated with each of several previously described parallelization methods. The interaction box and import region are shown for the HS method (left), the NT method (middle), and the SH method (right). In each case, one zone consists of the union of the interaction box (purple) and the red region, while the other consists of the union of the interaction box and the blue region. The import region consists of the union of the red and blue regions.

with every other particle in the interaction box. Equivalently, the two zones are (a) the interaction box itself, and (b) the union of the interaction box and the import region.

This method is guaranteed to interact all pairs of particles separated by less than a distance R. Because the interaction box has a finite volume, however, this method might, in the absence of any modifications, also interact some pairs separated by a distance greater than R. By way of example, if the x base coordinate of the interaction box is c_x, a particle within the interaction box whose x coordinate is $c_x + \delta$ (for some sufficiently small positive value of δ) could be interacted with a particle within the import region whose x coordinate is $c_x + b_x + R - \delta$, despite the fact that the two particles are separated by a distance greater than R. Such excess interactions can be eliminated by explicitly testing the distance between each pair to be interacted. Other tests can be used to ensure that a given particle residing within the interaction box does not interact with itself, and that each local interaction is computed once rather than twice. The application of such tests to eliminate inappropriate interactions will be referred to as *filtering*. Throughout this paper, when considering whether a particular method guarantees that all required particle interactions are computed, we will ignore filtering, as it will be used only to eliminate unnecessary interactions.

2.2. NT method

In the NT method, two particles interact within a box that has the x and y base coordinates associated with one particle and the z base coordinate associated with the other particle. Specifically, the x and y base coordinates of the interaction box are those of the particle with the smaller x base coordinate, with ties broken first in favor of the particle with the smaller y base coordinate and then in favor of the particle with the larger z base coordinate. The z base coordinate of the interaction box is the z base coordinate of the other particle.

The resulting import region of each interaction box is the union of its *outer tower* and *outer plate*. The outer tower is shown in blue in Fig. 1, and comprises all points that lie directly above or below the interaction box within a distance R of the boundary. The outer plate is shown in red, and includes half of the region external to the interaction box that lies in its xy-plane within a distance R of the boundary. The zones are the *tower*, which consists of the union of the outer tower and the interaction box, and the *plate*, which consists of the union of the outer plate and the interaction box.

Interacting each particle in the tower with each particle in the plate ensures that all near interactions will be computed at least once [2]. To ensure that no near interaction is computed more than once, we can filter pairwise interactions such that plate particles in the interaction box will not interact with particles in the lower half of the outer tower. The filtering criteria of the HS method are also necessary. One can minimize the import requirements of the NT method by choosing the box aspect ratios appropriately [2].

2.3. SH method

The SH method employs a cubical box of side length b. An interaction box with logical base coordinates (i,j,k) imports particles from two zones. The first is a contiguous region that we refer to as the *base*. This zone is shown in blue and purple in Fig. 1, and consists of the set of boxes with logical base coordinates $(i + u,j,k - w_1)$, where $u \in [-r,r]$, $w_1 \in [0,\tilde{r} - 1]$, $r = \lceil R/b \rceil$ and $\tilde{r} = \lceil \sqrt{r+1} \rceil$. The second zone is a set of non-contiguous regions that we refer to as the *comb*. This zone is shown in red and purple in Fig. 1, and consists of the set of boxes with logical base coordinates $(i, j + v, k + w_2\tilde{r})$, where $v \in [-r,r]$ and $w_2 \in [0, \lfloor r/\tilde{r} \rfloor]$. Note that both the comb and the base include the interaction box. Snir [1] showed that interacting the comb and base ensures that all near interactions will be computed at least once. Filtering is required to ensure that each near interaction is computed only once, as for the NT method.

3. Two-zone methods

The HS, NT, and SH methods differ in their choice of zones, which in turn determine their import regions. In this section, we consider the question of which zone choices will guarantee that all required interactions are computed. We consider methods with two zones, and we assume that interactions are computed between each particle in one zone and each particle in the other zone, subject to filtering. In all cases, both zones include the interaction box itself but are otherwise non-intersecting.

We focus initially on describing a broad class of parallelization methods for range-limited N-body problems. The merits of various methods are considered later.

3.1. Simplified model

For clarity of exposition, we initially consider a simplified problem where:

1. The particles are restricted to the two-dimensional xy-plane.
2. A pair of particles are required to interact if they are separated by a distance less than R in the x dimension and by a distance less than R in the y dimension.
3. The only allowable parallelization schemes are those whose zones consist only of a set of complete boxes. We refer to such a parallelization scheme as a *voxelized method.*
4. We will ignore the fact that when computing the influence of two particles on one another, one typically uses the same data, which may (depending on problem-specific considerations) make it advantageous to perform the computation on the same processor. Instead, we assume that each processor computes the influence of particles in one zone (the *red zone*) on particles in the other zone (the *blue zone*), *but not vice versa.*

These simplifying assumptions will be progressively relaxed.

We define the *influence region* of a box as the smallest region of space that is guaranteed to contain all particles with which any particle in the box could interact. In this simplified problem, the influence region is a rectangle extending a distance R beyond the box in the positive and negative x and y directions.

An analog to the HS method for the simplified problem is shown in Fig. 2(a). Each box computes the influence on its own particles of all particles in its influence region. Each box imports all the particles in its influence region that do not already reside in the box.

An analog to the NT method is shown in Fig. 2(b). In this method, each box computes the influence of all particles in a row of neighboring boxes spanning the influence region horizontally (red) on all particles in a column of neighboring boxes spanning the influence region vertically (blue). Each box imports both the remote particles in its red zone and the remote particles in its blue zone. The import region of this method is a strict subset of the import region of the HS analog, even though the two methods compute exactly the same interactions when the union of all calculations by all processors is considered.

Other choices for the two zones can also ensure that all required interactions are computed. Fig. 2(c) shows a method reminiscent of the SH method. In this method, the red zone comprises a set of equally spaced horizontal slabs, while the blue zone is a vertically oriented block whose height is equal to the spatial period of the slabs. Again, each processor imports the particles in its two zones and computes the influence of particles in its red zone on particles in its blue zone. This method is guaranteed to compute the influence of all particles in a

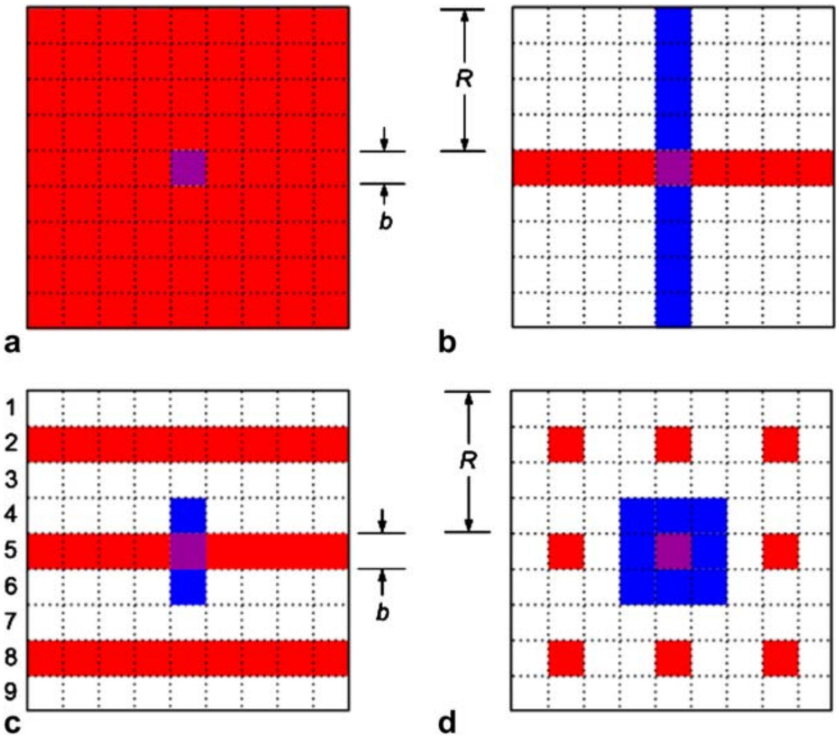


Fig. 2. Four different two-zone methods for the simplified two-dimensional problem. The interaction box and import region are shown for analogs of: (a) the HS method, (b) the NT method, (c) the SH method, and (d) the foam method. In each case, as in Fig. 1, one zone consists of the union of the interaction box (purple) and the red region, while the other consists of the union of the interaction box and the blue region. The import region again consists of the union of the red and blue regions. All figures illustrating two-dimensional methods (Figs. 2–6, and 8) use square boxes with side length b. In this figure, the side length of each box is equal to 1/4 of the interaction radius ($R = 4b$). The solid line encloses the influence region of the interaction box. In (c), we have numbered the rows of this influence region.

box's influence region on all particles in the box. In Fig. 2(c), we have numbered the rows of the influence region of the interaction box B from 1 to 9. Box B will compute the influence of particles in rows 2, 5, and 8 on particles in B. The box directly above B will compute the influence of particles in rows 1, 4, and 7 on particles in B, and the box directly below B will compute the influence of particles in rows 3, 6, and 9 on particles in B.

Fig. 2(d) shows a novel method that is also guaranteed to compute all required interactions. The blue zone comprises a rectangular area centered on the interaction box. The red zone is a foam-like structure that consists of boxes positioned such that their period in each dimension is equal to the width of the blue zone. This method is an analog of the three-dimensional *foam method* discussed in Section 3.6.

In all the methods we have described, the intersection of the two zones is simply the interaction box. We will assume throughout Section 3 that the zones do not overlap outside the interaction box, even though our analysis generally applies even if they do. If the zones were to overlap outside the interaction box, then some pairs of particles that reside in the same box would be interacted in a different box.

3.2. The convolution criterion

Given a pair of zones associated with each interaction box, how can one check whether all required interactions are computed? We first address this question under the assumptions of the simplified two-dimensional model. For any particular method, we define the *coverage region* of a box as the largest region of space such that the method will compute the influence of any particle in the coverage region on any particle in the box. If the coverage region of a box includes the influence region of that box, then all required interactions will be computed (any additional, redundant interactions can be eliminated by filtering). Note that the coverage region is a property of the parallelization method employed, while the influence region is not.

Under the assumptions of the simplified model, we can easily determine the coverage region of a box B given the zones associated with a particular method. Under the assumptions of our simplified model, each box computes the influence of each particle in its red zone on each particle in its blue zone. We first find the set of boxes whose blue zone includes B. We then take the union of the red zones for all boxes in this set to obtain the coverage region of B.

To determine the coverage region graphically, we first reflect the blue zone of B through the center of B to show the set of boxes whose blue zones include B. In two dimensions, reflection through the center of B is equivalent to a 180° rotation about B. If the center of B is at (x, y), then the reflected blue zone will include the point $(x - a, y - b)$ if and only if the blue zone of B includes the point $(x + a, y + b)$. The coverage region is the union of the red zones corresponding to each box in the reflected blue zone, as each box in the reflected blue zone will compute the influence of all particles in its own red zone on all particles in B. The red zones of different boxes differ from one another only by translation, so we can also compute the coverage region by translating the red zone of B such that it has the same spatial relationship to each box in the reflected blue zone of B that it originally had to B, and then taking the union of these translated red zones.

This process can also be described as a convolution operation. The coverage region is the support of the convolution of an indicator function over the blue zone with an indicator function over the red zone. For that reason, we term the criterion that the coverage region include the entire influence region the *convolution criterion*. In a voxelized method without filtering, all particles in a box are interacted with the same set of particles outside that box. If some part of the influence region of a box is not in the box's coverage region, then no interactions will be computed between particles in the box and particles in that part of the influence region. For a voxelized method, the convolution criterion is thus not only sufficient but also necessary to guarantee that all required particle interactions are computed. We consider non-voxelized methods in Section 3.4.

For the four methods illustrated in Fig. 2, the coverage region of the interaction box is identical to its influence region. In the absence of filtering, all particles in a particular box will therefore be interacted with all other particles whose distance from the box is no more than R in both the x and y dimensions. Fig. 3 illustrates several other pairs of zones that also satisfy the convolution criterion. The method shown in Fig. 3(a) is similar to that of Fig. 2(d), but in Fig. 3(a), the two zones associated with a particular box are not centered on that box. This method also differs from those of Fig. 2 in that the coverage region of the box is larger than its influence region, with parts of the red zone lying outside the influence region. If we eliminated these parts of the red zone, however, the coverage region would no longer include the entire influence region, so the method would no longer ensure that all required interactions were computed. Figs. 3(b) and (c) show two exotic methods that also satisfy the convolution criterion.

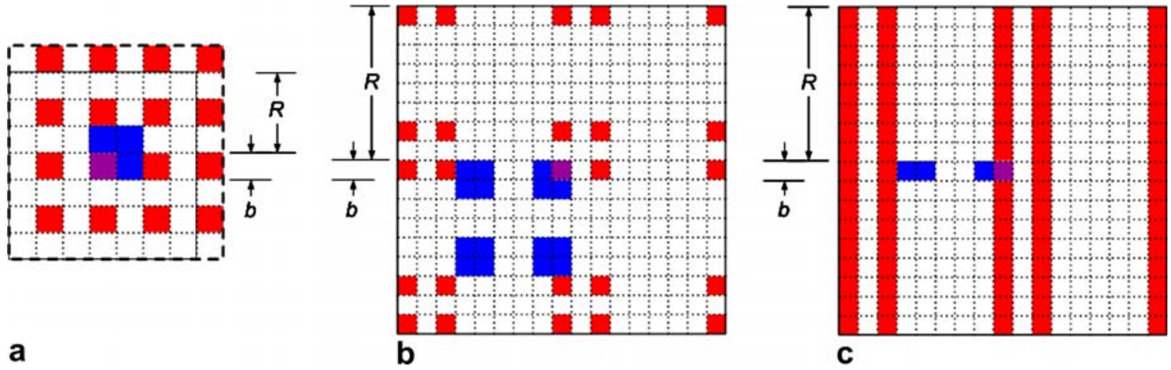


Fig. 3. Three additional methods that satisfy the convolution criterion. In (a) $R = 3b$, while in (b) and (c), $R = 8b$. Colors are assigned as in Fig. 2, and the solid line again encloses the influence region of the interaction box. The dashed line in (a) encompasses the coverage region associated with this method.

3.3. Symmetric interactions

So far, we have assumed that each box computes the influence of particles in its red zone on particles in its blue zone, but not vice versa. Because the influence of particle a on particle b typically depends on some or all of the same data as the influence of particle b on particle a, we can reduce communication requirements by allowing each box to also compute the influence of its blue zone on its red zone.

While the convolution criterion still holds in this case, we must modify the procedure used to derive the coverage region from the zones. We assume for now that assumptions 1–3 of our simplified model are still in effect (i.e., particles in a two-dimensional space interact if they are separated by a distance less than R in each dimension, and interactions are computed using some voxelized method). We now compute the coverage region as the union of a red-on-blue coverage region, determined using the procedure described in Section 3.2, and a blue-on-red coverage region, computed using the same procedure but with the roles of the red and blue zones reversed.

For a voxelized method, the blue-on-red coverage region of a box B is simply the red-on-blue coverage region reflected through the center of B. The coverage region will therefore include the entire influence region of B if and only if, for any point in the influence region that does not fall within the red-on-blue coverage region, the point obtained by reflecting that point through the center of B falls within the red-on-blue coverage region.

Fig. 4 shows modifications to the import regions of Fig. 2, assuming that each box will compute the influence of both its blue zone on its red zone and its red zone on its blue zone. In each case, we have cut one of the two zones roughly in half. The red-on-blue coverage region for each scheme is enclosed within a dashed line. In each case, the red-on-blue coverage region includes at least half the influence region, and the union of the red-on-blue coverage region and its reflection covers the entire influence region. Again, filtering can be used to ensure that no particle pair will interact twice and that only required interactions are computed.

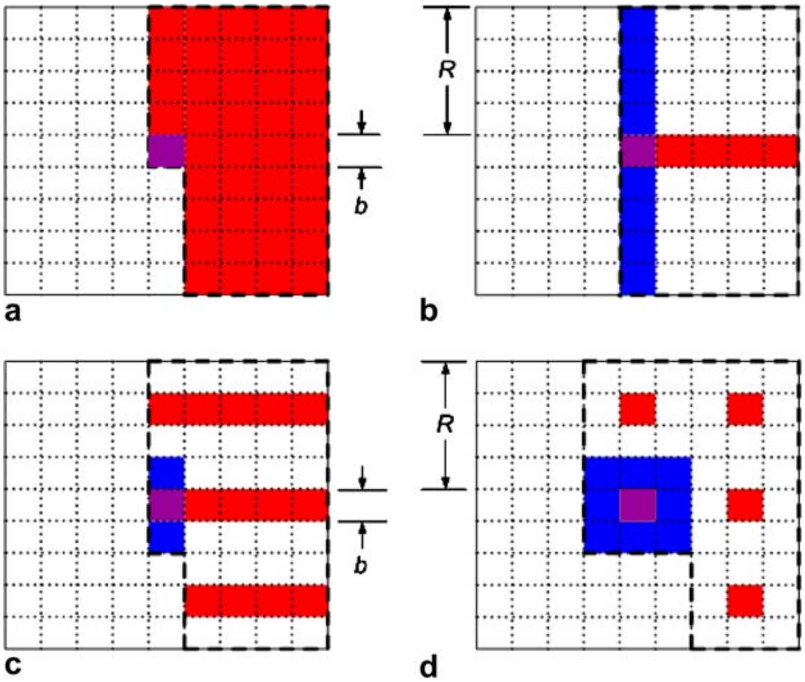


Fig. 4. Methods that take advantage of particle–particle interaction symmetry. The methods shown are two-dimensional analogs of: (a) the HS method, (b) the NT method, (c) the SH method, and (d) the foam method, with $R = 4b$. In each case, the dashed line encloses the red-on-blue coverage region of the interaction box, and the solid line encloses its influence region.

3.4. Non-voxelized methods

Thus far, we have assumed that each zone contains only whole boxes. We can often reduce the volume of the import region, however, by using zones that contain one or more *partial* boxes.

To handle such *non-voxelized* methods, we must generalize the convolution criterion. In particular, we will compute a coverage region for each *point* in a box, rather than for the box as a whole. Given a particular method, we define the coverage region of a point as the region whose influence on a hypothetical particle at that point will be computed by that method. The coverage region of a point may again be constructed as the union of a red-on-blue coverage region and a blue-on-red coverage region. We construct the red-on-blue coverage region for a point q using a procedure similar to that used for its home box B, except that we take the union of red zones of the set of boxes whose blue zones include the point q rather than the full box B. We compute the blue-on-red coverage region of a point using a similar procedure, but with the roles of the red zone and the blue zone reversed.

We define the influence region of a point as the minimal region guaranteed to contain all particles required to interact with a particle at that point. In the simplified model, the influence region is a square of side length $2R$ centered on the point. In the more typical scenario where particles are required to interact if separated by a Euclidean distance less than R in three dimensions, the influence region is a sphere of radius R surrounding the point.

The generalized version of the convolution criterion states that for each point in a box, the coverage region of that point must include the entire influence region of that point. This criterion is both necessary and sufficient to ensure that each particle will interact with all other particles in its influence region.

Fig. 5 shows a modified version of the NT-like method of Fig. 4(b), where the zones contain partial boxes. In this illustration, we have assumed that assumptions 1 and 2 of the simplified model still apply (i.e., two-dimensional problem with a square influence region), but we have chosen R to be a non-integer multiple of b. The method shown ensures that each point in the box interacts with all points within a square of side length $2R$ centered on that point, even though no point in the box is guaranteed to interact with all points within the influence region of the box.

3.5. Rounding and the rounding criterion

Under assumption 2 of the simplified model, two particles were required to interact if they were separated by a distance no greater than R in each dimension. We now adopt the more typical assumption that two particles are required to interact if and only if they are separated by a Euclidean distance less than R. In two dimensions, the influence region of each point is therefore enclosed by a circle of radius R rather than a square of side length $2R$. Likewise, the influence region of each box is a subset of the rectangular region considered in the previous sections; its corners are rounded. We can take advantage of this fact to reduce the size of the import region, a technique that we will refer to as *rounding*.

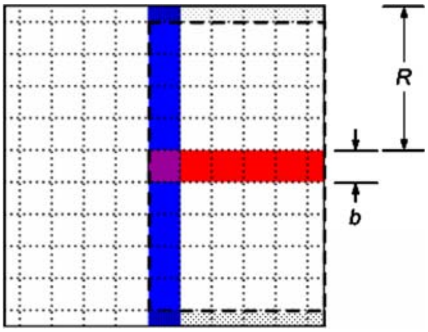


Fig. 5. A non-voxelized version of the two-dimensional NT analog of Fig. 4, for a problem where $R = 4.5b$. All points in the interaction box have a red-on-blue coverage region that includes the area enclosed by the dashed line, while some points have a red-on-blue coverage region that includes part of the dithered areas. The solid line again encloses the influence region of the interaction box.

A point may be discarded from zone v if the point is further than R away from the closest point in the set of zones with which v interacts. We refer to this condition as the *rounding criterion*. For the two-zone methods considered in this section, the rounding criterion implies that points in the red zone may be discarded if they are further than R away from all points in the blue zone, and points in the blue zone may be discarded if they are further than R away from all points in the red zone. Whereas the rounding criterion is a sufficient condition to guarantee that a point can be safely discarded from a zone, it is not a necessary condition, because in some cases one can also safely eliminate points that do not meet this condition; we describe such a method in a separate paper [20]. One can, however, always use the generalized form of the convolution criterion to test whether a particular rounded method is guaranteed to compute all required interactions.

Fig. 6 shows rounded versions of the HS and SH analogs. While rounding never increases import volume, it does not always reduce it. The NT analog in Fig. 4(b) and the foam analog in Fig. 4(d), for example, are not affected by rounding.

3.6. Two-zone methods in higher dimensions

The analytical tools discussed in the previous sections generalize in a straightforward manner to the case where the particles lie in a three-dimensional space (or, for that matter, a higher-dimensional space). Given a method where each box interacts particles in one zone with particles in another zone, we can use the convolution criterion to determine whether all particle pairs within a distance R will be interacted. The influence region is three-dimensional, and the "convolution-like" process of determining the coverage region takes place in three dimensions (or in d dimensions, if the particles lie in a d-dimensional space). We can also apply the rounding criterion to reduce the volumes of the zones. Again, the convolution criterion is a necessary and sufficient condition for ensuring that all particles within a distance R will be interacted. The rounding criterion, on the other hand, is not guaranteed to produce the smallest permissible zones.

Fig. 7 shows the SNT method as well as three novel three-dimensional parallelization methods, to which we refer as the *clouds*, *city*, and *foam* methods. All of these might be viewed as rounded, three-dimensional generalizations of the two-dimensional methods of Figs. 4(c) and (d). The foam method is discussed further in Section 6.3.

These three-dimensional methods differ from one another in the extent to which they can take advantage of rounding. The NT and SNT methods can only use *two-dimensional rounding* – in the rounded versions of these methods, the coverage region of each point includes a cylinder of radius R and height $2R$. In the high parallelization limit, when boxes are very small, two-dimensional rounding reduces the size of one of the zones and of the coverage region by a factor of $\pi/4$ compared to the unrounded case. Other methods, including the HS, clouds, city, and foam methods, allow *three-dimensional rounding* – they reduce the coverage region of each point to a sphere of radius R in the high parallelization limit. Asymptotically, such rounding reduces the size of one of the zones and of the coverage region by a factor of $\pi/6$ compared to the unrounded case and by a

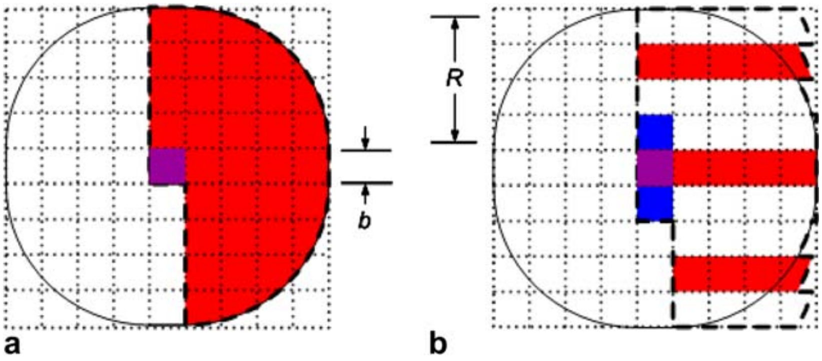


Fig. 6. Rounded two-dimensional analogs of (a) the HS method and (b) the SH method, with $R = 4b$. In each case, the dashed line encloses the red-on-blue coverage region of the interaction box, and the solid line encloses its influence region. As illustrated in (b), one cannot perform valid rounding by simply discarding all parts of the import region that lie outside the influence region.

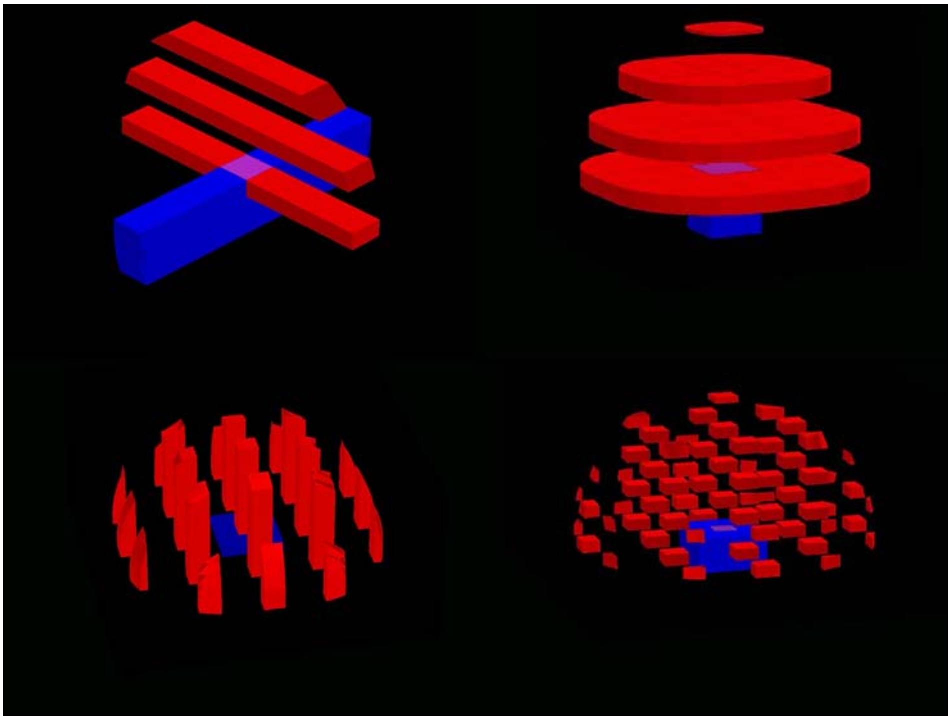


Fig. 7. Zones of several three-dimensional, two-zone parallelization methods. This figure illustrates the import regions and the zones of the SNT (upper left), clouds (upper right), city (lower left) and foam (lower right) methods, with colors assigned as in Fig. 1.

factor of 2/3 compared to the two-dimensional-rounded case. Likewise, in spaces of more than three dimensions, rounding reduces the coverage region of each point to a multidimensional sphere in the high parallelization limit for certain methods but not for others.

In a space with an odd number of dimensions, the two zones of a method guaranteed to compute all near interactions will typically have different shapes and different volumes. In order to reduce total import volume, one can adjust the relative volumes of the zones in two ways:

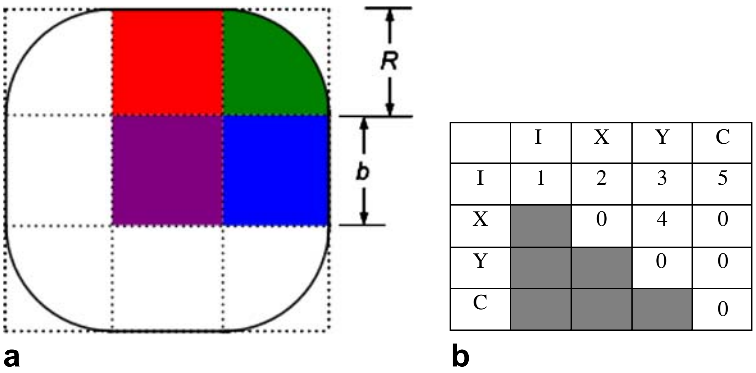
- Use non-cubical boxes, making one of the zones larger than it would have been otherwise and the other smaller. The NT method employs this technique.
- In methods where one or both zones involve multiple non-contiguous pieces (e.g., the SH method), choose the spacing between the pieces appropriately.

Methods in which the two zones have approximately equal volumes typically have a smaller total import volume than methods where one zone is much larger than the other. We will return to the topic of minimizing import volume in Sections 5 and 6.

4. *k*-Zone methods

The methods considered thus far involve the interaction of two overlapping zones. These methods interact all particles in one zone with all particles in the other zone, subject to filtering criteria. In this section, we consider a class of methods that make use of three or more zones, interacting any number of pairs of these zones. Such *k*-zone methods interact all particles in a given zone with all particles in a subset of the other zones, subject to filtering criteria. Two-zone methods and *k*-zone methods together constitute the class of zonal methods.

The four zones of a particular two-dimensional *k*-zone method are shown in Fig. 8(a). Each processor imports a rectangle extending a distance R beyond the interaction box in the $+x$ direction (zone X), a rectangle



	I	X	Y	C
I	1	2	3	5
X	▨	0	4	0
Y	▨	▨	0	0
C	▨	▨	▨	0

Fig. 8. (a) Zones of a two-dimensional *k*-zone method, with $R = b$. Zone I appears in purple, X in blue, Y in red, and C in green. The solid line encloses the influence region of the interaction box. (b) Corresponding interaction schedule.

extending a distance R beyond the interaction box in the $+y$ direction (zone Y), and a quarter circle of radius R (zone C) adjacent to both zone X and zone Y. The interaction box is treated as a distinct zone (zone I). Zone X interacts with zone Y, and zone I interacts with all the zones, itself included. Filtering eliminates particle pairs separated by a distance greater than R, as well as duplicated local interactions. In this method, two particles interact within a box whose x base coordinate is that of the particle with the smaller x coordinate and whose y base coordinate is that of the particle with the smaller y coordinate. The import volume of this method is always less than that of the rounded HS analog shown in Fig. 6(a). When R is small relative to the box side lengths, the import volume is also less than that of the NT analog shown in Fig. 4(b).

The convolution criterion introduced in the context of two-zone methods also applies to *k*-zone methods: all required particle pair interactions will be computed if and only if the coverage region for each point in a box covers the entire influence region of that point. The procedure for computing the coverage region of a point, however, is somewhat more complicated for a *k*-zone method than for a two-zone method. To determine the coverage region of a point for a *k*-zone method, one takes the union of the *zone-pair coverage regions* of that point for all pairs of zones interacted by the method, where we define the zone-pair coverage region of a point for a given pair of zones as the region whose influence on a hypothetical particle at that point will be computed by the interaction of those two zones. Each zone-pair coverage region can be computed using the procedure described in Section 3.4 for computation of the coverage region for a two-zone method.

Alternatively, one can compute the coverage region for the full box by taking the union of zone-pair coverage regions of the box for all pairs of zones to be interacted, where we define the zone-pair coverage region of a box for a given pair of zones as the region whose influence on any particle in the box will be computed by the interaction of those two zones. Each zone-pair coverage region can be determined using the procedure of Section 3.3. All required particle–pair interactions are guaranteed to take place if the coverage region of the box includes the entire influence region of the box. For a voxelized method, the converse is also true.

In the *k*-zone method illustrated in Fig. 8, the union of the zone-pair coverage regions for the interaction of the four zones with zone I includes all but two quarter-circle portions of the interaction box influence region. The zone-pair coverage region for the interaction of zone X and zone Y covers these two remaining quarter circles.

In our discussion of *k*-zone methods, we will assume that the zones do not overlap, and that a zone cannot interact with itself. The only exception is the interaction box, which we define as a separate zone that will interact with itself. In the absence of these assumptions, we could convert any *k*-zone method into a method with just one zone by defining the single zone to include the entire import region and the interaction box. However, this would lead to a situation where some pairs of particles that reside in the same processor would interact on a different processor.

The *k*-zone methods have several practical benefits. First, they can reduce communication bandwidth requirements relative to two-zone methods, both by allowing more near interactions to be computed for a given import volume (as in the method of Fig. 8) and by exposing additional rounding opportunities (as will be illustrated shortly). Second, k-zone methods can be used to simplify the filtering criteria required to avoid redundant computations. For example, the two-zone formulation of the NT method requires filtering to prevent pairs of particles with the same x and y base coordinates from being interacted twice, while a k-zone formulation of the NT method, presented below, eliminates the need for such filtering in all zone-zone interactions except for the interaction of the interaction box with itself. Third, the use of additional zones allows more opportunities to overlap communication and computation, because interactions between some pairs of zones can be computed while other zones are being imported, as discussed below.

The extent to which communication between processors can be overlapped with computation depends on the order in which pairs of zones are interacted. We use an *interaction schedule* (or simply *schedule*) to specify which pairs of zones should be interacted and in what order those interactions should take place. The schedule may be represented as an upper triangular matrix in which the entry in row i, column j (where $i \leqslant j$) corresponds to the interaction, if any, between zone i and zone j. A zero entry indicates that the corresponding pair of zones is not interacted at all. The remaining entries are unique positive integers indicating the order in which the zone–zone interactions are computed: the entry containing a 1 represents the first interaction to be computed, a 2 represents the second interaction to be computed, and so on. Part (b) of Fig. 8 shows the schedule corresponding to the method illustrated part (a).

The order of interactions that allows maximal overlap of computation and communication depends on details of the available computational hardware. The schedules presented in this paper were formulated using the following procedure:

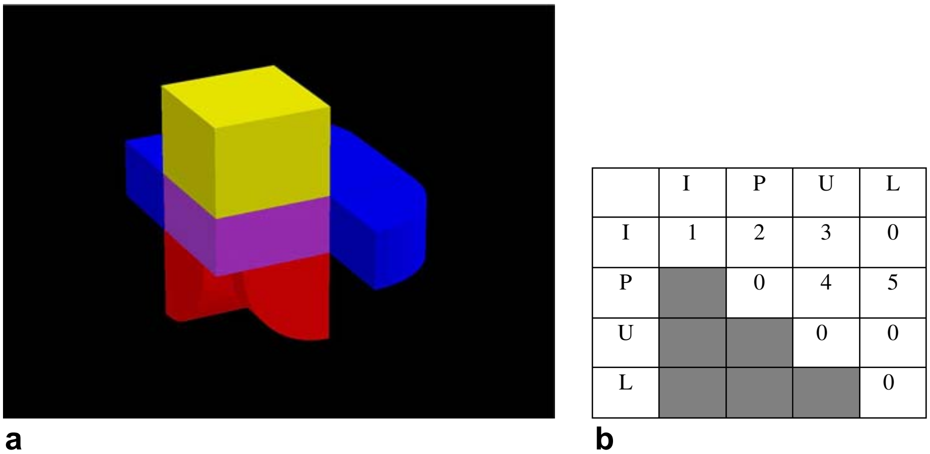
- First, we determine the order in which the zones will be imported, requiring that zone u be imported before zone v if zone u participates in more zone–zone interactions than zone v.
- We then number the zones according to the order in which they are imported, with zone 1 representing the interaction box, zone 2 representing the first zone imported, zone 3 representing the second zone imported, and so on.
- We set schedule entries corresponding to pairs of zones that should not be interacted to zero. We then number the remaining entries in column-major order; that is, the interaction in row i_1 and column j_1 will be computed before the interaction in row i_2 and column j_2 if $j_1 < j_2$, or if $j_1 = j_2$ and $i_1 < i_2$. This ensures that computation of the interactions between a pair of zones that have already been imported never has to wait for the computation of an interaction involving a zone that has not yet been imported.

To illustrate the advantages of k-zone methods, we describe the kZ-NT method, a reformulation of the NT method using four zones. The zones correspond to the interaction box (I), the lower half of the outer tower (L), the upper half of the outer tower (U), and the outer plate (P), as shown in Fig. 9(a). Zone I interacts with zones I, U, and P, and zone P interacts with zones U and L; the interaction schedule is shown in Fig. 9(b) corresponding interaction schedule. We can start computing interactions between pairs of particles in the interaction box while importing particles from the other zones. We can also compute interactions between P and I while importing U and L, and we can complete the import of L while interacting U with I and P. By eliminating the interaction between I and L, we prevent pairs of particles lying in different boxes with the same x and y base coordinates from being interacted twice. The k-zone formulation of the NT method exposes an additional rounding opportunity not exploited by the original NT method. Because the lower tower L interacts only with P, the portions of L that are more than a distance R from any point in P can be eliminated, as shown in Fig. 9.

5. Lower bounds on the import volume

5.1. A general bound

In the Appendix, we establish a lower bound for the import volume V_i of a zonal method. We assume that the zones are non-overlapping, that one of the zones is the interaction box, and that only the interaction box



	I	P	U	L
I	1	2	3	0
P	▨	0	4	5
U	▨	▨	0	0
L	▨	▨	▨	0

Fig. 9. (a) Zones of the kZ-NT method. Zone I appears in purple, L in red, U in yellow, and P in blue. (b) Corresponding interaction schedule.

zone interacts with itself. We refer to zones other than the interaction box as *remote zones*. We denote the total number of zones by N_z and the number of remote zones by $N_{zr} = N_z - 1$. We find that

$$\begin{aligned} V_i &\geqslant \frac{V_b}{\left(1 - \frac{1}{N_{zr}}\right)} \left[\sqrt{\left(1 - \frac{1}{N_{zr}}\right) \frac{V_{IR,remote}}{V_b} + 1} - 1 \right] \quad \text{when } N_{zr} > 1, \\ V_i &\geqslant \frac{1}{2} V_{IR,remote} \quad \text{when } N_{zr} = 1, \end{aligned} \tag{1}$$

where V_b is the volume of a box and where $V_{IR,remote}$ is the average across all points in a box B of the volume of the portion of the point's influence region that lies outside B. $V_{IR,remote}$ is given by

$$V_{IR,remote} = \frac{4}{3}\pi R^3 - \frac{1}{V_b} \int_{r' \in B} \mathrm{d}^3 r' \int_{r \in B} \mathrm{d}^3 r \, \Theta(|r - r'| < R),$$

where Θ is the indicator function

$$\Theta(a) = \begin{cases} 1 \text{ if } a \text{ is true}, \\ 0 \text{ if } a \text{ is false}. \end{cases}$$

Eq. (1) is a strict bound that holds for all zonal methods.

5.2. A useful approximate lower bound

As discussed in the Appendix, one can obtain a larger, though approximate, lower bound by replacing $V_{\mathrm{IR,remote}}$ in the above bound by the volume of the *interaction neighborhood* V_{IN}, defined as the portion of a box's influence region that lies outside the box. Using the parallelization parameter α_R and the normalized box side lengths α_x, α_y, and α_z, defined in Section 2, we can express the volume of the interaction neighborhood V_{IN} as

$$V_{IN} = V_b \left[\frac{4\pi\alpha_R^3}{3} + \pi\alpha_R^2(\alpha_x + \alpha_y + \alpha_z) + 2\alpha_R(\alpha_x^{-1} + \alpha_y^{-1} + \alpha_z^{-1}) \right].$$

In the high parallelism limit, where the box volume V_b approaches zero and the parallelization parameter α_R approaches infinity, V_{IN} and $V_{IR,remote}$ converge to $4\pi R^3/3$, so the approximate lower bound approaches the strict one.

The appendix also shows that if the box volume and interaction radius are fixed, then one minimizes V_{IN} by using cubic boxes (this does not imply that cubic boxes minimize import volume for any particular parallelization method). Thus, for any box aspect ratios,

$$V_{IN} \geqslant V_b\left(\frac{4\pi\alpha_R^3}{3} + 3\pi\alpha_R^2 + 6\alpha_R\right).$$

Substituting this minimum value of V_{IN} for $V_{IR,remote}$ in Eq. (1) gives an approximate lower bound that is independent of box aspect ratios:

$$\begin{aligned} V_i &\geqslant V_b\left(1-\frac{1}{N_{zr}}\right)^{-1}\left\{\sqrt{\left(1-\frac{1}{N_{zr}}\right)\left(\frac{4\pi\alpha_R^3}{3} + 3\pi\alpha_R^2 + 6\alpha_R\right)+1}-1\right\} \quad \text{when } N_{zr} > 1, \\ V_i &\geqslant \frac{1}{2}V_b\left(\frac{4\pi\alpha_R^3}{3} + 3\pi\alpha_R^2 + 6\alpha_R\right) \quad \text{when } N_{zr} = 1. \end{aligned} \tag{2}$$

In the high parallelism limit, as α_R goes to infinity, the α_R^3 term in Eq. (2) dominates, so the approximate bound of Eq. (2) becomes

$$\begin{aligned} V_i &\geqslant \left[\frac{4\pi\alpha_R^3 V_b^2}{3\left(1-\frac{1}{N_{zr}}\right)}\right]^{1/2} \quad \text{when } N_{zr} > 1, \\ V_i &\geqslant \frac{2}{3}\pi\alpha_R^3 V_b \quad \text{when } N_{zr} = 1. \end{aligned} \tag{3}$$

Because V_{IN} and $V_{IR,remote}$ converge in the high parallelism limit, the bound of Eq. (3) becomes strict in that limit. The import volume of the foam method discussed in Section 6.3 approaches this lower bound asymptotically for $N_{zr} = 2$.

By substituting $\alpha_R = R/V_b^{1/3}$ in Eq. (3) and noting that $V_b = V_{gc}/p$, where V_{gc} is the volume of the global cell, one can show that the minimal import volume for a zonal method scales as $R^{3/2}p^{-1/2}$ for $N_{zr} > 1$ and as R^3 for $N_{zr} = 1$. Snir also proved that scaling of $R^{3/2}p^{-1/2}$ is asymptotically optimal, albeit under slightly different assumptions [1].

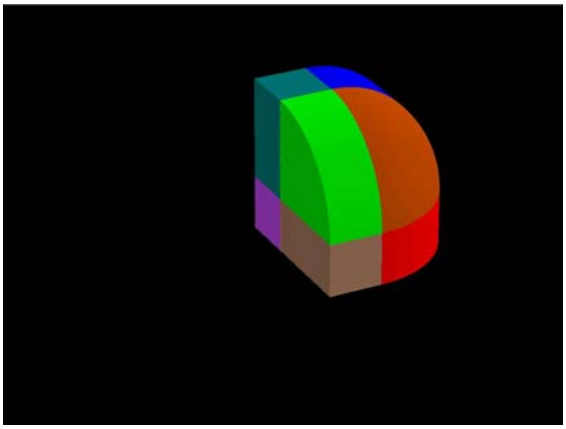
The factor $(1-\frac{1}{N_{zr}})^{-\frac{1}{2}}$ in Eq. (3) decreases as N_{zr} grows, starting at $\sqrt{2}$ for $N_{zr} = 2$ and approaching 1 as N_{zr} approaches infinity. In practice, the use of multiple zones typically gives a smaller improvement in import volume, because it is generally not possible to devise a method where every pair of remote zones interacts and where every particle pair considered before filtering corresponds to a unique near interaction.

6. Specific methods

This section details several specific zonal methods that require a lower import volume than any previously described method of which we are aware for some range of parallelization parameter values, where the parallelization parameter $\alpha_R = R/V_b^{1/3}$ determines the effective degree of parallelism, as discussed in Section 2. These methods are guaranteed to interact any pair of particles separated by a distance less than R. They will interact a pair of particles twice only if both particles lie in the same box – a form of redundancy that can be eliminated by filtering.

6.1. ES method

Intuitively, the HS method might seem to be optimal in terms of import volume in the low-parallelism limit, but this is not the case. Fig. 10(a) shows the zones of the *ES* (for "Eighth-Shell") *method*, which is the three-dimensional generalization of the two-dimensional method illustrated in Fig. 8. In the ES method, two particles interact within a box whose x base coordinate is that of the particle with the smaller x coordinate, whose y base coordinate is that of the particle with the smaller y coordinate, and whose z base coordinate is that of the particle with the smaller z coordinate. This method utilizes eight non-overlapping zones: the interaction box



a

	I	FX	FY	FZ	EX	EY	EZ	C
I	1	2	3	5	8	10	12	14
FX		0	4	6	9	0	0	0
FY			0	7	0	11	0	0
FZ				0	0	0	13	0
EX					0	0	0	0
EY						0	0	0
EZ							0	0
C								0

b

Fig. 10. (a) Zones of the ES method. Zone I appears in purple, EX in blue, EY in green, EZ in red, C in orange, FX in beige, and FZ in cyan. Zone FY is hidden from view. (b) Corresponding interaction schedule.

(I); three *face subregions* (FX, FY and FZ) that abut the $+x$, $+y$, and $+z$ faces, respectively, of the interaction box; three *edge subregions* (EX, EY and EZ) that abut the $+y+z$, $+x+z$, and $+x+y$ edges, respectively, of the interaction box; and one *corner subregion* (C) that touches the $+x+y+z$ corner of the interaction box. The import region of the ES method is smaller than that of the HS method at any degree of parallelism. The import region of the ES method, which consists of one corner subregion, three edge subregions, and three face subregions, is a strict subset of the import region of the HS method, which consists of four corner subregions, six edge subregions, and three face subregions. The total import volume of the ES method is

$$V_i = V_b\left[\frac{\pi\alpha_R^3}{6} + \frac{\pi\alpha_R^2}{4}(\alpha_x+\alpha_y+\alpha_z) + \alpha_R(\alpha_x^{-1}+\alpha_y^{-1}+\alpha_z^{-1})\right].$$

If the home box volume V_b and the parallelization parameter α_R are fixed, the import volume of the ES method is minimized by using cubic home boxes, for which $\alpha_x = \alpha_y = \alpha_z = 1$.

The ES method is so named because in the high parallelism limit, its import region becomes one-eighth of the interaction neighborhood, an "eighth-shell." In that limit, its import volume will be one-quarter that of the half-shell method. On the other hand, the import volume of the ES method has the same asymptotic scaling properties as the HS method, so at high levels of parallelism, it proves inferior to the NT method.

Fig. 10(b) shows an interaction schedule for the ES method. Zone I interacts with all of the zones (including itself). In addition, each face zone interacts with the other two face zones and with one edge zone.

6.2. kZ-NT and kZ-SNT methods

The kZ-NT method was introduced in Section 4. The import volume of this method is smaller than that of the original NT method because the kZ-NT method exploits an additional rounding opportunity. More specifically, the import volume of the original NT method is

$$V_i = 2Rb_{xy}^2 + 2Rb_{xy}b_z + \frac{\pi R^2 b_z}{2}.$$

One can show that the import volume of the kZ-NT method is smaller than that of the NT method by

$$Rb_{xy}^2 - \frac{\pi b_{xy}R^2}{2} + \frac{2}{3}R^3 \quad \text{when } R \leqslant b_{xy},$$

$$Rb_{xy}^2 - b_{xy}R^2\arcsin\left(\frac{b_{xy}}{R}\right) + \frac{2}{3}R^3 + R^2 z_o - \frac{1}{3}z_o^3 \quad \text{when } R > b_{xy},$$

where $z_o = \sqrt{1-(\frac{b_{xy}}{R})^2}$ and where we have assumed that $b_x = b_y = b_{xy}$, because the import volumes of both the NT and kZ-NT methods are minimized for fixed V_b, b_z, and R when $b_x = b_y$. This reduction in import volume is small relative to the total import volume for high parallelization parameter values, when R is typically much larger than b_{xy}, but is significant for lower parallelization parameter values.

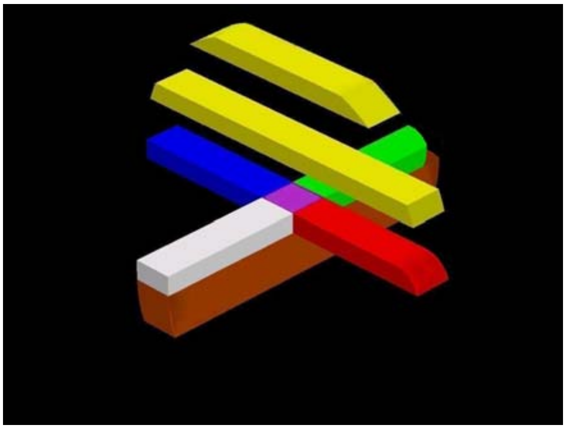
The SNT method can also be reformulated using additional zones. Fig. 11 shows the zones of the *kZ-SNT method*, and gives the interaction schedule. The import volume of the kZ-SNT method is smaller than that of the original SNT method because the kZ-SNT method exploits two additional rounding opportunities, illustrated in Fig. 11.

If there is no space between the bars of the comb in the kZ-SNT method, then the zone BA disappears and import of zone E becomes unnecessary, as it only interacts with BA. In that case, the remaining import region exactly corresponds to that of the NT method. It can be shown that if one has the freedom to optimize the box aspect ratios in addition to the spacing between the bars in the kZ-SNT method, then one minimizes the import volume of the kZ-SNT method by reducing it to the kZ-NT method.

If the box aspect ratios are fixed, on the other hand, the SNT method has an advantage over the NT method in that the SNT method can approximately balance the volume of the base and the comb by adjusting the spacing between the bars of the comb, substantially reducing the total import volume at higher levels of parallelism. The SNT method therefore has a lower import volume than the NT method for high parallelization parameter values at fixed box aspect ratios. Likewise, the kZ-SNT method has a lower import volume than the kZ-NT method under these conditions.

6.3. Foam method

The foam method is illustrated in Fig. 7. The interaction box imports a *brick* consisting of s^3 boxes, which are configured as an $s \times s \times s$ cube centered on the interaction box in the x and y dimensions and extending below, but not above, the interaction box in the z dimension. The interaction box also imports a *foam* of individual boxes spaced every s boxes in each dimension. The foam extends above the interaction box, and is rounded to form an approximately hemispherical structure. The foam method dominates the kZ-NT and kZ-SNT methods at very high levels of parallelism because it exploits three-dimensional rounding.



	I	BA	CO	W	S	N	E
I	1	2	3	5	7	0	0
BA		0	4	6	0	0	12
CO			0	0	8	10	0
W				0	9	11	0
S					0	0	0
N						0	0
E							0

b

Fig. 11. (a) Zones of the kZ-SNT method. Zone I appears in purple, BA in orange, CO in yellow, W in blue, S in white, N in green, and E in red. This k-zone formulation exposes two new rounding opportunities: (1) a portion of E near one of its edges exceeds a distance R from all points in BA and can be eliminated, because E interacts only with BA; and (2) a portion of N near one of its corners exceeds a distance R from all points in CO and W and can be eliminated, because N interacts only with CO and W. (b) Corresponding interaction schedule.

The asymptotic import volume of the foam method for large numbers of processors is

$$V_{i\infty} = s^3 V_b + \frac{2\pi R^3}{3s^3}.$$

Optimizing s to minimize the asymptotic import volume yields

$$s^6 = \frac{2\pi R^3}{3V_b}$$

and

$$V_{i\infty} = \sqrt{\frac{8\pi R^3 V_b}{3}} = \sqrt{\frac{8\pi\alpha_R^3 V_b^2}{3}}.$$

Asymptotically, the foam method achieves the lower bound of Eq. (3) for a method with two remote zones. Shaw [2] showed that the NT and SNT methods achieve an asymptotic import volume of $V_{i\infty} = \sqrt{4\pi R^3 V_b}$ (with optimized box aspect ratios). In the high parallelism limit, the foam method has $\sqrt{2/3} \approx 0.82$ times the import volume of the NT and SNT methods in either their original or k-zone formulations.

Because the foam method only becomes practical at very high degrees of parallelism, efficient implementations would likely need to use a large number of zones to hide import latencies. A practical interaction schedule is omitted in the interest of brevity. Conceptually, the key elements of the schedule are that the

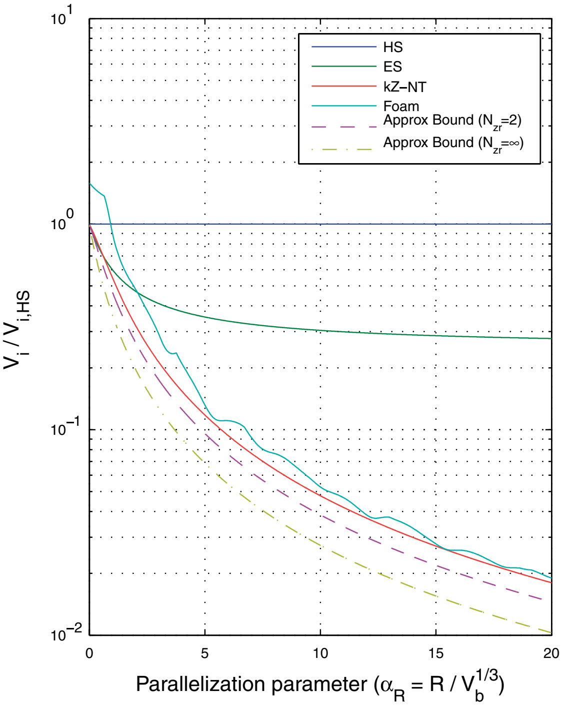


Fig. 12. Comparison of the import volume for several methods, with optimized box aspect ratios. The parallelization parameter α_R increases with increasing number of processors, increasing interaction radius, and decreasing simulated system size. Import volumes for all methods are represented relative to that of the HS method ($V_{\mathrm{i,HS}}$) at each parallelization parameter setting and plotted on a log axis. The box aspect ratios for each method, as well as the spacing parameter s for the foam method, were optimized at each setting of the parallelization parameter to minimize import volume. The optimization procedure for the foam method involved an analytic approximation, so a fully optimized foam method may have a lower import volume than that shown here. The kZ-SNT method is not included in the figure because it always requires more bandwidth than the kZ-NT method when box aspect ratios are tunable. The approximate lower bounds are those of Eq. (2).

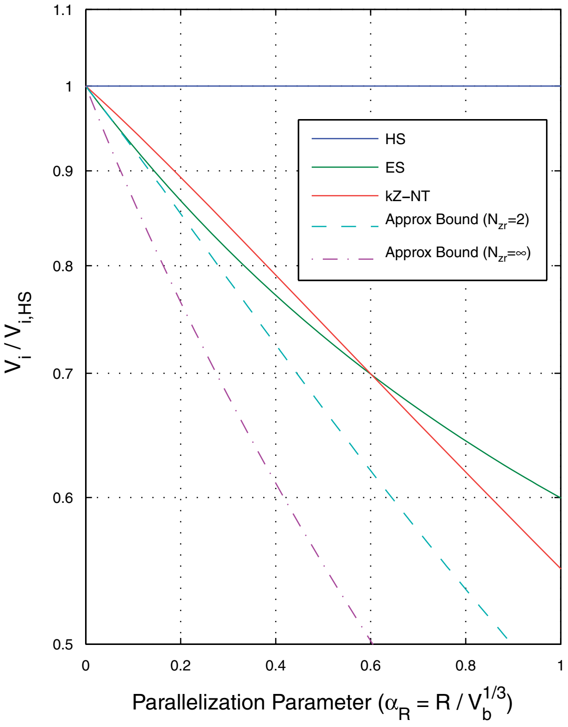


Fig. 13. Expanded view of the low-parallelization portion of Fig. 12. This figure provides a more detailed view of that portion of the data presented in Fig. 12 that corresponds to a relatively small number of processors, and thus a relatively large box size, relative to the interaction radius R. The foam method has been omitted, since it fares poorly throughout this range of parameter values.

interaction box interacts with the foam, the brick, and itself, while the foam and the brick interact with one another.

6.4. Comparison of communication bandwidth requirements

Figs. 12–15 graphically depict the import volumes of several zonal methods over a range of parallelization parameter values. These figures include the methods that have the lowest import volume among the methods discussed in this paper for some range of parallelization parameter values. Figs. 12 and 13 assume that the box aspect ratios of each method are tuned to minimize import volume at each setting of the parallelization parameter, while Figs. 14 and 15 assume that boxes are constrained to be cubical.

When box aspect ratios are tunable, the ES method has the lowest import volume of the methods we have discussed for α_R below 0.60. The kZ-NT method has the lowest import volume for values of α_R ranging from 0.60 to approximately 15. The foam method has the lowest import volume in the high parallelism limit, but becomes competitive with the kZ-NT method only when α_R is greater than about 15.

As an example, consider a *reference system* having a cubic global cell measuring 80 Å on a side and an interaction radius of 12 Å. These parameters are within the range that might be typical for a molecular dynamics simulation of a biomolecular system; at a typical density of 0.1 atoms/Å^3, such a system would contain about 51,000 atoms. For this system, the relationship between α_R and the number of processors p is $p \approx 296\alpha_R^3$, so the ES method minimizes import volume for fewer than about 64 processors[3] (small computer clusters). The kZ-NT method minimizes import volume between about 64 and a million processors (large clusters, or even

[3] These figures are approximate because we have ignored the constraints on box aspect ratios due to the finite number of processors. For example, in order for the boxes to be exactly cubical when the global cell is cubical, the number of processors must be the cube of some integer. In practice, one might choose not to use a few of the available processors in order to obtain more convenient aspect ratios.

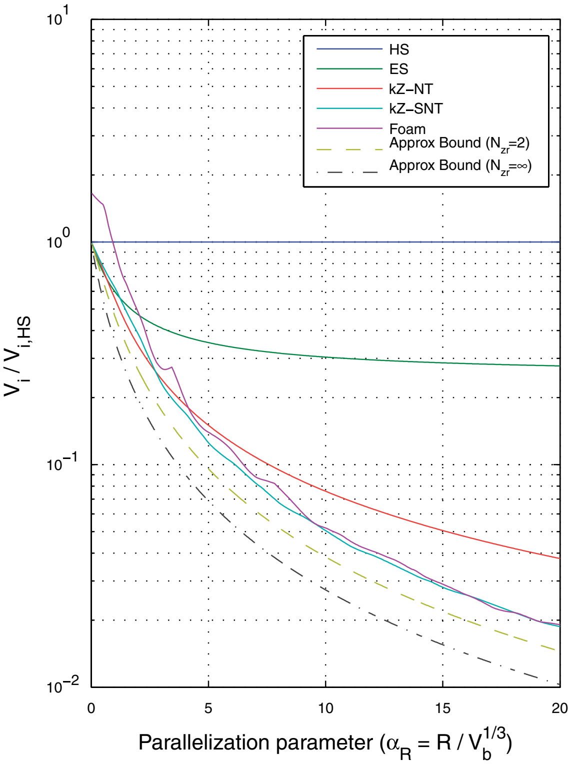


Fig. 14. Comparison of the import volume for several methods, with boxes constrained to be cubical. The parallelization parameter α_R increases with increasing number of processors, increasing interaction radius, and decreasing simulated system size. Import volumes for all methods are represented relative to that of the HS method ($V_{\text{i,HS}}$) at each parallelization parameter setting and plotted on a log axis. The spacing parameters for the foam and kZ-SNT methods were optimized at each setting of the parallelization parameter to minimize import volume. The approximate lower bounds are those of Eq. (2).

ultraparallel hardware such as QCDOC [21] or Blue Gene/L [22]). The foam method dominates only at extreme levels of parallelism involving more than a million processors.

When box aspect ratios are fixed, the kZ-SNT method requires a smaller import volume than the kZ-NT method for sufficiently large parallelization parameters. For cubic boxes, the ES method minimizes import volume for α_R below 0.66, followed by the kZ-NT method for α_R between 0.66 and 2.7 and the kZ-SNT method for α_R between 2.7 and approximately 16. The foam method again has the smallest import volume in the high parallelism limit, but becomes competitive with kZ-SNT only for α_R above 16. For our reference system, the ES method has the lowest communication bandwidth of the methods we have discussed for small clusters (up to about 85 processors), the kZ-NT method has the lowest bandwidth for large clusters (about 85 to about 5800 processors), the kZ-SNT method has the lowest bandwidth for existing ultraparallel hardware (about 5800 to 1.2 million processors), and the foam method has the lowest bandwidth only beyond 1.2 million processors.

In practice, the ease with which box aspect ratios can be tuned depends on the architecture of the communication network. The dimensions of the global cell are usually determined by the physical system being simulated. In a network where communication between any two processors is equally expensive, one can easily change the dimensions of the box grid, so aspect ratios are relatively unconstrained. On a network with a mesh or toroidal topology (e.g., in Blue Gene/L or in Cray's T3D, T3E, and XT3 systems [23,24]), on the other hand, it is generally most convenient for the box grid to correspond to the network grid, determining fixed box aspect ratios.

Figs. 12–15 also show the approximate lower bounds of Eq. (2) for two remote zones ($N_{zr} = 2$) and for an infinite number of remote zones ($N_{zr} = \infty$). The parallelization methods discussed in Section 6 come

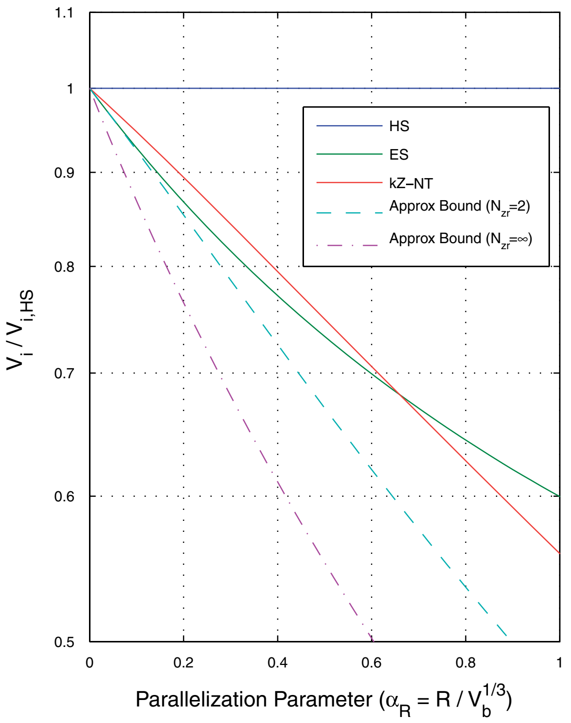


Fig. 15. Expanded view of the low-parallelization portion of Fig. 14. The foam and kZ-SNT methods have been omitted, since they fare poorly in this range of parameter values.

close to the $N_{zr} = 2$ bound at all levels of parallelism, with the foam method approaching the bound asymptotically, albeit slowly, at high levels of parallelism and the ES method actually beating the $N_{zr} = 2$ bound at low levels of parallelism by using more than two remote zones. The gap between actual import volumes and the $N_{zr} = \infty$ bound reflects the fact that this bound is tight only when all remote zones interact with one another without producing redundant interactions, which is rarely if ever the case for a method with a large number of zones.

We have not proven that the methods of Section 6 are optimal in terms of import volume. In fact, we have discovered novel neutral territory methods, to be described in a subsequent paper [25], whose import volumes are at least slightly smaller than that of any method described in this paper over a wide range of parallelization parameters. Another technique called the *midpoint method*, which we also describe in a separate paper [20], has an import volume identical to that of the ES method at all levels of parallelism and offers certain practical advantages over the ES method on some machines. IBM researchers recently published an independently developed parallelization technique related to the midpoint method that uses nonuniform boxes for load balancing [26,27].

Our assumption that $G_x \geqslant b_x + 2R$, $G_y \geqslant b_y + 2R$, and $G_z \geqslant b_z + 2R$ (Section 2) implies that the global cell must be partitioned at least once along each dimension (i.e., the mesh of boxes must be at least $2 \times 2 \times 2$). When only a small number of processors are available, methods that involve a partition of the space into boxes along only one or two of the three dimensions may minimize import volume. If we partition the global cell into a $1 \times 1 \times n$ mesh of boxes, implying that $b_y = G_y$ and $b_z = G_z$, then the only communication required will be in the positive and negative z directions. If we partition the global cell into an $1 \times m \times n$ mesh of boxes, implying that $b_z = G_z$, then the only communication required will be in the yz plane. In either case, one can still utilize the zonal methods we have introduced, but the formulae for their import volumes will be different from those presented previously. When parallelizing our reference system across 16 or fewer processors, application of the ES or kZ-NT methods with a partition along two dimensions will require a lower import volume than application of the ES method with a partition along three dimensions. If fewer than six processors are available, a traditional spatial decomposition method such as the HS method with a partition along one dimension will require a lower import volume than any of the methods we have described with a two- or three-dimensional partition.

7. Conclusions

This paper has introduced a broad class of techniques, which we refer to as zonal methods, for the efficient parallelization of range-limited N-body problems. This class includes both traditional spatial decomposition methods and the methods of Shaw and Snir as special cases; it also includes a wide variety of "neutral territory" methods that have not been described previously. We have described a test to determine whether a particular zonal method is guaranteed to interact all pairs of particles separated by a distance less than R. We have also demonstrated a systematic method for exploiting opportunities for "rounding", and have shown how the use of three or more "zones" can reduce communication bandwidth requirements, overlap communication time with computation time, and efficiently avoid the computation of redundant interactions. We have also derived strict and approximate lower bounds on the import volume requirements of zonal methods at various degrees of parallelism.

The current paper also introduces new methods that have lower communication bandwidth requirements than any previously published method of which we are aware at both very low and very high levels of parallelism. For the broad class of problem and system parameters that lie in between these two extremes, we have also described a new set of modifications to the previously described NT and SNT methods that reduce their communication bandwidth requirements below those of any published method of which we are aware.

While this paper has focused primarily on import volume as a metric for comparing different parallelization methods, the best choice of parallelization method may in practice depend on other factors as well. These include:

- *Load balancing.* The efficiency of a given computation may depend in part on the ability of the parallelization method used to balance computational load across the available processors. Unlike home territory methods, neutral territory methods can achieve a significant degree of load balancing even in the case where the number of simulated particles (as distinct from the number of particle *interactions*) is smaller than the number of available processors, since neutral territory methods can calculate pairwise interactions within processors that contain no particles. We discuss techniques for load balancing further in a separate paper [20].
- *Redundant interactions*. To maximize computational efficiency, it is desirable to interact only pairs of particles that lie within a distance R of one another, and to interact each such pair only once. One can eliminate redundant interactions through a process of "filtering," but the filtering process itself imposes an additional computational load. Parallelization methods that reduce the need for filtering will in general improve computational efficiency.
- *Communication latency*. The amount of time spent on communication depends not only on the amount of data to be communicated but also on the amount of time required to transmit even a minimum-sized packet of information between two processors (*communication latency*). In some parallel architectures (e.g., a mesh-connected machine), particles in nearby boxes can generally be imported more quickly than those in distant boxes, since the latter may require that a given information packet pass through multiple intermediate processors before arriving at its intended destination [23,24,28]. In such cases, methods like the midpoint method [20] that tend to import particles from nearby boxes may have a comparative advantage. Furthermore, some parallelization methods may do a better job than others in balancing the amount of data to be transmitted over different communication links.
- *Overlap of communication and computation*. The total time required to evaluate near interactions will in general depend on the extent to which communication and computation can be performed simultaneously. Certain k-zone methods lend themselves particularly well to hiding communication time by overlapping it with the required computation. More generally, the calculation of near interactions is often one of several computations to be performed in parallel for a particular simulation. This may present additional opportunities for overlapping computation and communication.

The best parallelization method for a particular range-limited N-body problem on a particular parallel machine will in general depend on a variety of parameters, including the size and shape of the system being simulated, the distribution of particles within the simulated system, the interaction radius necessary to achieve the level of accuracy required by the problem at hand, the number and speed of the available processors, the topology of the communication network through which they are connected, and the operational characteristics associated with each communication link. Given the number of and interactions among these considerations, it is not feasible to formulate a simple, single set of criteria for selecting the best choice of parallelization scheme for all possible applications. The analysis and methods presented in this paper, however, may provide a useful framework for choosing among an expanded range of highly efficient parallelization schemes, and may provide a theoretical foundation for future work in this area.

Acknowledgements

Thanks to Christine McLeavey for rendering many of the figures in this paper and for providing helpful comments on an earlier draft, and to Anne Weber for helping to prepare this manuscript. We benefited from fruitful discussions with members of the GROMACS [29] development team, particularly Berk Hess, Erik Lindahl, and David van der Spoel.

Appendix

Derivation of an exact lower bound on import volume

We derive a lower bound on the import volume of a zonal method under the following assumptions:

- The method uses boxes with a predetermined volume (V_b) and predetermined aspect ratios.
- The zones do not overlap. Each interaction box has a total of N_z zones, of which one is the box itself. We denote the volume of the import region by V_i and the number of remote zones by N_{zr}, where $N_{zr} = N_z - 1$. Note that a method with two zones whose intersection is the interaction box can be reformulated as a method with three non-overlapping zones.
- A pair of particles in the same box must interact in that box. This implies that only the interaction box zone can interact with itself.
- The method guarantees that any two particles within a distance R of one another are guaranteed to interact.
- As in the remainder of this paper, $G_x \geqslant b_x + 2R$, $G_y \geqslant b_y + 2R$, and $G_z \geqslant b_z + 2R$, implying that the global cell must be partitioned into at least a $2 \times 2 \times 2$ mesh of boxes.

In a zonal method, the positions of two particles uniquely determine the processor on which they will interact. We can therefore think of each processor as interacting points in space with one another, whether or not a particle is located at each point. By integrating over all pairs of points that must be interacted, we can express the total quantity of required point–point interactions in units of volume times volume. The influence region of any point q is a sphere of radius R, whose volume we denote by $V_{IR} = \frac{4}{3}\pi R^3$. The total quantity of required point–point interactions to be computed by the ensemble of p processors is therefore $\frac{1}{2} V_{gc} V_{IR}$, where V_{gc} is the volume of the global cell and the factor of 1/2 reflects the fact that each pair of points needs to be interacted only once. In a zonal method, each processor is responsible for the same quantity of point–point interactions, so the quantity required of each processor is

$$I_{required} = \frac{1}{2} V_{gc} V_{IR}/p = \frac{1}{2} V_b V_{IR}.$$

Some of these required interactions involve two points within the same box. We denote the quantity of local interactions to be computed in a box B by $I_{required,local}$ and express it as

$$I_{required,local} = \frac{1}{2} \int_{r' \in B} \mathrm{d}^3 r' \int_{r \in B} \mathrm{d}^3 r \Theta(|r - r'| < R),$$

where Θ is the indicator function

$$\Theta(a) = \begin{cases} 1 \text{ if } a \text{ is true,} \\ 0 \text{ if } a \text{ is false.} \end{cases}$$

The remainder of the required point–point interactions involve points with different home boxes, and are therefore remote interactions. We denote the quantity of remote interactions required of each processor by $I_{\text{required,remote}}$ and compute it as

$$I_{required,remote} = I_{required} - I_{required,local} = \frac{1}{2}V_b V_{IR} - \frac{1}{2}\int_{r'\in B} \mathrm{d}^3 r' \int_{r\in B} \mathrm{d}^3 r \Theta(|r - r'| < R).$$

We can write this expression as $I_{required,remote} = \frac{1}{2}V_b V_{IR,remote}$, where

$$V_{IR,remote} = V_{IR} - \frac{1}{V_b}\int_{r'\in B} \mathrm{d}^3 r' \int_{r\in B} \mathrm{d}^3 r \Theta(|r - r'| < R).$$

One can think of $V_{IR,remote}$ as the average volume of the portion of a point's influence region that lies outside its home box, with the average computed over all points in a box.

Next, we calculate the quantity of remote point–point interactions computed by each processor in a zonal method as

$$I_{actual,remote} = \sum_{j=1}^{N_z}\sum_{k=j+1}^{N_z} I_{jk} V_j V_k,$$

where V_j is the volume of the jth zone and I_{jk} equals 1 if zones j and k interact and 0 otherwise. (By omitting cases where $j = k$ from this sum, we have omitted the interaction between the interaction box and itself.) If we assume that zone 1 is the interaction box, then

$$I_{actual,remote} = V_b V_i + \sum_{j=2}^{N_z}\sum_{k=j+1}^{N_z} I_{jk} V_j V_k \leqslant V_b V_i + \sum_{j=2}^{N_z}\sum_{k=j+1}^{N_z} V_j V_k,$$

where equality holds if and only if all remote zones interact with all other remote zones. For a fixed number of remote zones and a fixed import volume V_i, one maximizes the quantity $\sum_{j=2}^{N_z}\sum_{k=j+1}^{N_z} V_j V_k$ by setting the volumes of the remote zones equal to one another, so

$$I_{actual,remote} \leqslant V_b V_i + \sum_{j=2}^{N_z}\sum_{k=j+1}^{N_z}\left(\frac{V_i}{N_{zr}}\right)\left(\frac{V_i}{N_{zr}}\right) = V_b V_i + \frac{N_{zr}(N_{zr}-1)}{2}\left(\frac{V_i}{N_{zr}}\right)^2 = V_b V_i + \frac{1}{2}\left(1 - \frac{1}{N_{zr}}\right)V_i^2.$$

The actual quantity of point–point interactions computed by each box must not be less than the required quantity, so $I_{actual,remote} \geqslant I_{required,remote}$. Substituting in the previously derived expressions for $I_{actual,remote}$ and $I_{required,remote}$ gives

$$V_b V_i + \frac{1}{2}\left(1 - \frac{1}{N_{zr}}\right)V_i^2 \geqslant \frac{1}{2}V_b V_{IR,remote}.$$

Solving for V_i, we find that

$$\begin{aligned} V_i &\geqslant \frac{V_b}{\left(1 - \frac{1}{N_{zr}}\right)}\left[\sqrt{\left(1 - \frac{1}{N_{zr}}\right)\frac{V_{IR,remote}}{V_b} + 1} - 1\right] \quad \text{when } N_{zr} > 1, \\ V_i &\geqslant \frac{1}{2}V_{IR,remote} \quad \text{when } N_{zr} = 1. \end{aligned} \tag{4}$$

A useful approximate lower bound on import volume

The lower bound of Eq. (4) involves $V_{IR,remote}$, the average volume of the portion of a point's influence region that lies outside its home box. The expression for $V_{IR,remote}$ involves a non-analytic integral, and the derivation of the bound provides only limited insight into how to design methods with an import volume near the lower bound. One can obtain a useful approximate lower bound by replacing $V_{IR,remote}$ in the bound with the volume of a box's *interaction neighborhood*, defined as the portion of a box's influence region that lies outside the box. (The volume of the interaction neighborhood is simply the volume of the influence region minus the volume of the box.) As the interaction radius becomes large relative to the side lengths of a box, this approximation for $V_{IR,remote}$ approaches the exact value of $V_{IR,remote}$. For smaller interaction radii, this replacement amounts to raising the lower bound to require that the quantity of remote point–point interactions computed by the method is greater than or equal to that computed by the HS method.

As described by Shaw [2], the interaction neighborhood (V_{IN}) of a box may be divided into six *face subregions*, rectangular parallelepipeds of width R abutting each face of the box; twelve *edge subregions*, quarter-cylinders of radius R abutting each edge of the box; and eight *corner subregions*, octants of a sphere of radius R abutting each corner of the box. The total volume of the corner subregions depends only on R, whereas the total volume of the edge subregions is proportional to the sum of the box side lengths, and the total volume of the face subregions is proportional to the surface area of the box. We can write the volume of the interaction neighborhood (V_{IN}) as

$$V_{IN} = \frac{4\pi R^3}{3} + \pi R^2(b_x + b_y + b_z) + 2R(b_y b_z + b_z b_x + b_x b_y),$$

where b_x, b_y, and b_z are the side lengths of the box. For a fixed box volume, one minimizes both the surface area and the sum of the side lengths by choosing the side lengths to be equal, as can be shown using calculus. Therefore V_{IN} is smaller for a cubic box than for any other box of the same volume.

Using the parallelization parameter $\alpha_R = R/V_b^{1/3}$ and the normalized box side lengths $\alpha_x = b_x/V_b^{1/3}$, $\alpha_y = b_y/V_b^{1/3}$, and $\alpha_z = b_z/V_b^{1/3}$, and noting that $\alpha_x\alpha_y\alpha_z = 1$, V_{IN} can be rewritten as

$$V_{IN} = V_b\left[\frac{4\pi\alpha_R^3}{3} + \pi\alpha_R^2(\alpha_x + \alpha_y + \alpha_z) + 2\alpha_R(\alpha_x^{-1} + \alpha_y^{-1} + \alpha_z^{-1})\right].$$

Substituting V_{IN} for $V_{IR,remote}$ in Eq. (4) gives the approximate lower bound:

$$V_i \geqslant V_b\left(1 - \frac{1}{N_{zr}}\right)^{-1}\left\{\sqrt{\left(1 - \frac{1}{N_{zr}}\right)\left[\frac{4\pi\alpha_R^3}{3} + \pi\alpha_R^2(\alpha_x + \alpha_y + \alpha_z) + 2\alpha_R(\alpha_x^{-1} + \alpha_y^{-1} + \alpha_z^{-1})\right] + 1} - 1\right\}$$

when $N_{zr} > 1$

$$V_i \geqslant \frac{1}{2}V_b\left[\frac{4\pi\alpha_R^3}{3} + \pi\alpha_R^2(\alpha_x + \alpha_y + \alpha_z) + 2\alpha_R(\alpha_x^{-1} + \alpha_y^{-1} + \alpha_z^{-1})\right] \quad \text{when } N_{zr} = 1.$$

References

[1] M. Snir, A note on N-body computations with cutoffs, Theor. Comput. Syst. 37 (2004) 295–318.

[2] D.E. Shaw, A fast, scalable method for the parallel evaluation of distance-limited pairwise particle interactions, J. Comput. Chem. 26 (2005) 1318–1328.

[3] R.W. Hockney, J.W. Eastwood, Computer Simulation Using Particles, Adam Hilger, Bristol, 1988.

[4] E. Bertshinger, J.M. Gelb, Cosmological N-body simulation, Comput. Phys. 5 (1991) 164–179.

[5] J.M. Thijssen, Computational Physics, Cambridge University Press, 1999.

[6] C.L. Brooks III, D.A. Case, Simulations of peptide conformational dynamics and thermodynamics, Chem. Rev. 93 (1993) 2487–2502.

[7] D. Frenkel, B. Smit, Understanding Molecular Simulation: From Algorithms to Applications, Academic Press, London, 2001.

[8] W. Wang, O. Donini, C.M. Reyes, P.A. Kollman, Biomolecular simulations: recent developments in force fields, simulations of enzyme catalysis, protein–ligand, protein–protein, and protein–nucleic acid noncovalent interactions, Annu. Rev. Biophys. Biomol. Struct. 30 (2001) 211–243.

[9] M. Karplus, J.A. McCammon, Molecular dynamics simulations of biomolecules, Nat. Struct. Biol. 9 (2002) 646–652.

[10] J.J. Monaghan, Smooth particle hydrodynamics, Annu. Rev. Astron. Astrophys. 30 (1992) 543–574.

[11] T. Schlick, R.D. Skeel, A.T. Brunger, L.V. Kalé, et al., Algorithmic challenges in computational molecular biophysics, J. Comput. Phys. 151 (1999) 9–48.

[12] T. Darden, D. York, L. Pedersen, Particle mesh Ewald: An $N\text{Log}(N)$ method for Ewald sums in large systems, J. Chem. Phys. 98 (1993) 10089–10092.

[13] H.Q. Ding, N. Karasawa, W.A. Goddard, The reduced cell multipole method for Coulomb interactions in periodic systems with million-atom unit cells, Chem. Phys. Lett. 196 (1992) 6–10.

[14] U. Essmann, L. Perera, M.L. Berkowitz, T. Darden, et al., A smooth particle mesh Ewald method, J. Chem. Phys. 103 (1995) 8577–8593.

[15] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (1987) 325–348.

[16] C. Sagui, T. Darden, Multigrid methods for classical molecular dynamics simulations of biomolecules, J. Chem. Phys. 114 (2001) 6578–6591.

[17] Y. Shan, J.L. Klepeis, M.P. Eastwood, R.O. Dror, D.E. Shaw, Gaussian split Ewald: a fast Ewald mesh method for molecular simulation, J. Chem. Phys. 122 (2005) 054101.

[18] G.S. Heffelfinger, Parallel atomistic simulations, Comput. Phys. Commun. 128 (2000) 219–237.

[19] S. Plimpton, B. Hendrickson, A new parallel method for molecular dynamics simulation of macromolecular systems, J. Comput. Chem. 17 (1996) 326–337.

[20] K.J. Bowers, R.O. Dror, D.E. Shaw, The midpoint method for parallelization of particle simulations, J. Chem. Phys. 124 (2006) 184109.

[21] P.A. Boyle, C. Jung, T. Wettig, The QCDOC supercomputer: hardware, software, and performance, in Proceedings of the International Conference on Computing in High Energy and Nuclear Physics (CHEP), 2003.

[22] F. Allen, G. Almasi, W. Andreoni, D. Beece, et al., Blue Gene: a vision for protein science using a Petaflop supercomputer, IBM Syst. J. 40 (2001) 310–327.

[23] R.E. Kessler, J.L. Schwarzmeier, Cray T3D: a new dimension for Cray Research, in Proceedings of the IEEE Computer Society International Conference (February 1993), pp. 176–182.

[24] S. Scott, G. Thomas, The Cray T3E network: adaptive routing in a high performance 3D torus, in Proceedings of the Hot Interconnects IV: A Symposium on High Performance Interconnects, Palo Alto, CA (August 1996), pp. 147–156.

[25] R.O. Dror, K.J. Bowers, D.E. Shaw (in press).

[26] R.S. Germain, B. Fitch, A. Rayshubskiy, M. Eleftheriou, et al., Blue Matter on Blue Gene/L: massively parallel computation for biomolecular simulation, in CODES+ISSS '05: Proceedings of the 3rd IEEE/ACM/IFIP International Conference on Hardware/software Codesign and System Synthesis, New York, NY (September 2005), pp. 207–212.

[27] B.G. Fitch, A. Rayshubskiy, M. Eleftheriou, T.J.C. Ward, et al., Blue Matter: Strong scaling of molecular dynamics on Blue Gene/L, IBM Research Report RC23688, IBM (2005).

[28] N.R. Adiga, G. Almasi, G.S. Almasi, Y. Aridor, et al., An overview of the Blue Gene/L supercomputer, in Proceedings of the 2002 ACM/IEEE Conference on Supercomputing, Baltimore, MD, 2002, pp. 1–22.

[29] D. van der Spoel, E. Lindahl, B. Hess, G. Groenhof, et al., GROMACS: fast, flexible, and free, J. Comput. Chem. 26 (2005) 1701–1718.