Joint neighbors approximation of macromolecular solvent accessible surface area

ARTICLE *in* JOURNAL OF COMPUTATIONAL CHEMISTRY · SEPTEMBER 2007

Impact Factor: 3.59 · DOI: 10.1002/jcc.20550 · Source: PubMed

2 AUTHORS:

Georgy Rychkov
Petersburg Nuclear Physics Institute
15 PUBLICATIONS **39** CITATIONS
SEE PROFILE

Michael Petukhov
Petersburg Nuclear Physics Institute
39 PUBLICATIONS **438** CITATIONS
SEE PROFILE

Joint Neighbors Approximation of Macromolecular Solvent Accessible Surface Area

GEORGY RYCHKOV,[1,2] MICHAEL PETUKHOV[1]

[1]*Division of Molecular and Radiation Biophysics, St. Petersburg Nuclear Physics Institute, The Russian Academy of Sciences (PNPI RAS), Gatchina, St. Petersburg 188300, Russia*

[2]*"Biophysics," Research and Education Center of PNPI RAS and St. Petersburg State Polytechnic University, St. Petersburg, 194021 Russia*

Received 19 May 2006; Revised 27 July 2006; Accepted 10 September 2006
DOI 10.1002/jcc.20550
Published online 3 April 2007 in Wiley InterScience (www.interscience.wiley.com).

Abstract: A new method for approximate analytical calculations of solvent accessible surface area (SASA) for arbitrary molecules and their gradients with respect to their atomic coordinates was developed. This method is based on the recursive procedure of pairwise joining of neighboring atoms. Unlike other available methods of approximate SASA calculations, the method has no empirical parameters, and therefore can be used with comparable accuracy in calculations of SASA in folded and unfolded conformations of macromolecules of any chemical nature. As shown by tests with globular proteins in folded conformations, average errors in absolute atomic surface area is around 1 Å^2, while for unfolded protein conformations it varies from 1.65 to 1.87 Å^2. Computational times of the method are comparable with those by GETAREA, one of the fastest exact analytical methods available today.

 J Comput Chem 28: 1974–1989, 2007

Key words: solvent-accessible surface area; analytical surface area; analytical gradients; hydration; folded and unfolded proteins

Introduction

Water is a natural environment for numerous soluble proteins and other biological macromolecules, including DNA, RNA, etc. Basic functions and spatial structures of these molecules depend on their interactions with surrounding water molecules. The solute–water interactions have been in focus of several recent experimental and theoretical studies.[1–4] As shown, the molecular dynamics simulations in explicit water boxes are capable to adequately account for solute–water interactions.[5–7] Although the explicit water models are quite accurate, they are extremely computationally demanding and require long computation times to achieve thermodynamic equilibration of a water box and obtain the hydration energy, particularly when the solute undergoes large scale conformational transitions.

There have been many hydration models developed based on solvent accessible surface area (SASA) calculations.[8–10] The term "solvent accessible surface" (SAS) was first introduced by Lee and Richards in 1971.[11] Assuming approximate sphericity of water molecules, the authors defined the SAS of a solute molecule as an aggregate of all points occupied by the center of a water molecule probe sphere having a direct contact with the van der Waals surface of a solute molecule without a penetration. The van der Waals surface is defined as the exposed surface of a molecule represented by spheres with radii equal to their atomic van der Waals radii. Thus, SAS is a surface determined by solute atoms with radii equal to $r_i + r_p$, where r_i is van der Waals atomic radii and r_p is the radius of water probe sphere (normally 1.4 Å). The SAS of a given atom of a molecule is a part of a total SAS produced by the spherical probe rolling over this atom.

As often assumed in the implicit hydration models, the atomic free energy contributions into hydration energy are linearly proportional to their atomic SASA[8,10,12] and the total molecular hydration energy can be expressed as a sum of the atomic SASAs multiplied by empirical parameters called atomic

Correspondence to: G. Rychkov; e-mail: georgy-rychkov@yandex.ru

Contract/grant sponsor: Russian Foundation for Basic Research; contract/grant number: 02-04-49259-a

Contract/grant sponsor: Ministry of Education and Science of Russian Federation; contract/grant number: RNP.2.2.1.1.4663

Contract/grant sponsor: St. Petersburg Scientific Center, the Russian Academy of Sciences, 2004–2005

Contract/grant sponsor: Competitive Center of Fundamental Natural Sciences (Saint-Petersburg), 2005

Contract/grant sponsor: Fogarty International Research Collaboration Award; contract/grant number: 2 R03 TW00 1319-05 (to Prof. Cox)

solvation parameters (ASP), which are obtained by fitting to experimentally observed hydration energies of a series of small organic molecules:

$$\Delta G = \sum_i \sigma_i A_i,$$

where A_i is an atom SASA and σ_i is its ASP. The molecular dynamics simulations require calculations of the forces affecting all interacting atoms. In turn, it demands the calculation of energy gradients with respect to solute atomic coordinates.

There have been analytical and numeric methods proposed for calculations of the molecular SASA and its first derivative. The analytical methods can be classified into two types, the exact and the approximate ones. The exact analytical methods were based initially on the Gauss–Bonnet theorem.[13,14] Since that time main efforts were devoted to improve the computation efficiency of the exact methods.[15–17] Other approaches based on direct calculations of the sphere intersection surface areas become a next step in evolution of exact analytical methods calculating atomic SASA.[18–20] The α-shapes were also employed to simplify determination of the sphere intersection geometry.[21,22] Recently, a new original exact analytical method based on projection of three-dimensional molecular system onto a plane has been proposed.[23]

The development of numerical methods was initiated by Shrake and Rupley.[24] This class of methods for SASA calculations can be subdivided into two classes according to geometrical representation of the spherical surfaces. Namely, the subclass where the molecular surface is represented by dots distributed on the surface of atomic spheres,[24–26] and the one where the surface is represented by triangles.[27–29] Unfortunately, it was found that the problem of uniform dot distribution on spherical surface has no analytical solution.[30] This drawback decreased the accuracy of the numerical methods.

The pioneer work on the approximate analytical method for SASA calculations was carried out by Wodak and Janin.[31] Henceforth, the evolution of the approximate methods is developed to increase their accuracy. Generally, the approximate SASA methods are based on easy calculated geometric parameters such as the pairwise intersections of neighboring spheres[32–34] and the distances between these spheres.[35,36] Also, there are a few reports on the approximate SASA calculations based on the Gaussian density functions.[37,38]

The disadvantages of exact analytical methods are in their complexity and related difficulties in obtaining the high-order derivatives of SASA with respect to atomic coordinates often required in the efficient minimization algorithms. Since the approximate methods utilize simple analytical formula, they are more computationally effective and more suitable for obtaining the higher order analytical derivatives of the atomic SASA. However, the disadvantages of the available approximate methods are in lack of the accuracy and the great deal of empirical parameters depending not only on a chemical nature of a molecule under consideration but also on its conformation. Thus, a reasonable balance between the accuracy of exact analytical methods and the simplicity as well as of the computational efficiency of the approximate approaches still remains to be found.

In this work, we present a new analytical approximate method for SASA calculations of arbitrary molecules. This method is based on a novel recursive procedure of pairwise joining of neighboring atomic spheres (the joint neighbor approximation or JNA). This approximation does not require any empirical parameters and was found to provide a good accuracy, which is similar in both folded and unfolded conformations of different macromolecules. The analytical derivatives of SASA with respect to atomic coordinates are also provided by this approach.

Materials and Methods

All atomic coordinates for the small organic molecules, proteins, DNA, RNA, and their complexes were taken from PDB[39] and are listed in Table 2. To add hydrogen atoms to proteins listed in Table 2 and to model their unfolded states, the ICM program package for molecular modeling[40] was used. The unfolded conformations were obtained by assigning the values of 180° to the dihedral angles φ, ψ, and Ω and subsequent 200 local energy minimization steps by the conjugate gradient method.

To accelerate the computations of SASA, the neighbor-list reduction (NLR) algorithm was used as has been described by Weiser et al.[41] The NLR removes selected neighbors of an atom under consideration from the computation, resulting in a shorter neighbor list. The atoms to be removed by NLR are those having intersections with the atom under consideration, designated as S_i and falling entirely within unions of other atomic intersections with the atom S_i. In the SASA calculations, the NLR reduction rate can be as high as 74% in some cases.[41] In our work, we used the 3NLR version of the NLR algorithm, which considers three hard spheres simultaneously—a central atom, a candidate to be removed, and another neighboring atom.

The JNA Method

Similar to other methods for calculations of the molecular SASA, our method is based on the assumption that a solute molecule is composed of a set of N rigid interpenetrating atomic spheres with known Cartesian coordinates and van der Waals radii. Two sets of atomic radii used in the work were taken from works by Weiser et al.[33] and Guvench and Brooks[36] (data shown in Table 1). Here and further in the text, it is assumed that the radius r_i of each atomic sphere is equal to the sum of its van der Waals radius and

Table 1. Atomic van der Waals Radii.

	Radius, Å	
Atom	Weiser et al.[a]	Guvench and Brooks[b]
H	–	1.20
C	1.70	1.70
O	1.60	1.52
N	1.65	1.55
S	1.90	1.80
P	1.90	–
Cl	1.80	–

[a]Radii are taken from work by Weiser et al.[33]
[b]Radii are taken from work by Guvench and Brooks.[36]

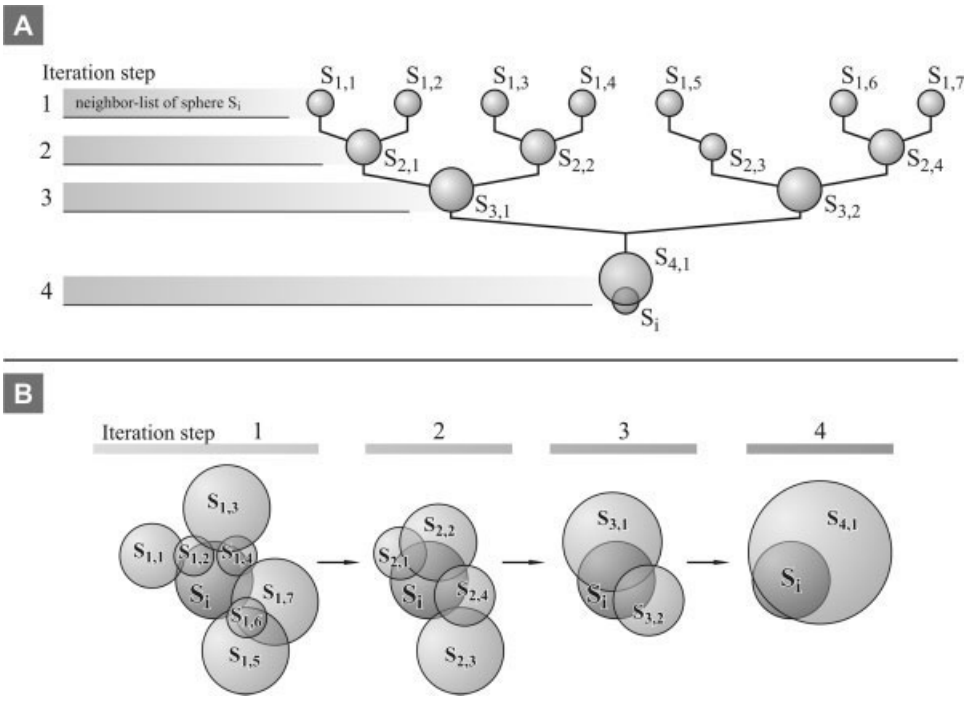


Figure 1. The recursive method of neighboring spheres joining. (A) The scheme illustrates calculations of SASA for a sphere S_i located at the root of the tree. $S_{l,m}$ are neighbors of the sphere S_i, where subscript l indicates number of iteration step and m indicates location of sphere $S_{l,m}$ in the neighbor-list of sphere S_i. (B) Representation of the scheme for distribution of seven neighboring spheres around central sphere S_i in two-dimensional case. At the first-iteration step, spheres $S_{1,1}$ and $S_{1,2}$ are joined producing sphere $S_{2,1}$ that goes to the second iteration step; spheres $S_{1,3}$ and $S_{1,4}$ produce sphere $S_{2,2}$; spheres $S_{1,6}$ and $S_{1,7}$ produce sphere $S_{2,3}$; and unpaired sphere $S_{1,5}$ goes to the second iteration step without changes and it is redesignated as $S_{2,4}$. The procedure continues until single neighboring sphere $S_{4,1}$ will be formed. In two-dimensional case, the length of arc of circle S_i buried inside circle $S_{4,1}$ is equal to that buried inside union of all neighboring circles from first-iteration step.

radius of the probe sphere (1.4 Å). The method calculates SASA of each individual atom. The SASA of whole molecule is calculated as the sum of SASA of all individual atoms. Every atom of any molecule has at least one covalently bonded neighboring atom that has nonzero overlap with its surface. In macromolecules with complicated structure and dense packing of atoms, the number of neighbors overlapping the surface of an individual atom S_i is obviously not restricted by the number of covalently bonded atoms.

Figure 1 illustrates the work of recursive core of the algorithm proposed here where all neighboring spheres turn into a single sphere of a larger size. Let a given sphere (designated as S_i) have seven neighbors indicated from $S_{1,1}$ to $S_{1,7}$. The subscripts of neighboring spheres indicate the current iteration step number and the position of a neighbor in the neighbor-list of spherical atom S_i at this iteration step. At each iteration step, spheres from the neighbor-list of sphere S_i with nonzero intersections are pairwise joined and then replaced by a new sphere that goes to the next iteration step, in such a way that the surface area of S_i buried inside the new sphere is equal to that buried inside the union of the two preceding spheres. In case of an odd number of neighbors, an unpaired sphere is directly transferred to next step of the JNA algorithm. This recursive procedure continues until one or more nonoverlapping spheres remain in the neighbor-list of sphere S_i. Thus, the surface area of the sphere S_i included within the set of the remaining spheres (or a single sphere) is equivalent to the area included within all its neighbors from the initial step of the recursive procedure.

To maintain reasonable simplicity of formulas, the method considers only two and three sphere intersections. Such an approach was found to produce sufficiently accurate results.

Intersection of Two Spheres

Every sphere S_j from the neighbor-list of a given sphere S_i satisfies the following two inequalities: $r_i + r_j > d_{ij}$ and $|r_i - r_j| < d_{ij}$, where r_i and r_j are radii of spheres S_i and S_j, respectively, and d_{ij} is the distance between them. Thus, the surface area of sphere S_i buried inside all the spheres from its neighbor list is nonzero. The surface area of sphere S_i buried inside sphere S_j, is designated as A_{ij} and calculated by formula:[31]

$$A_{ij} = 2\pi r_i \left(r_i - \frac{d_{ij}}{2} - \frac{r_i^2 - r_j^2}{2d_{ij}} \right). \quad (1)$$

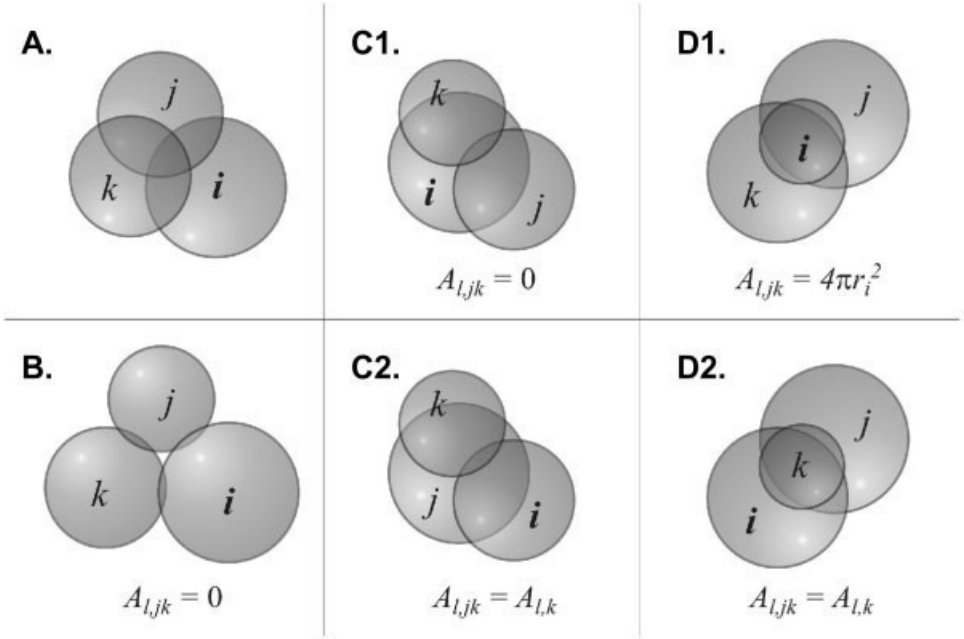


Figure 2. All possible types of triple intersections of three spheres. A sphere under consideration is marked by letter i and its neighboring spheres are marked by letters j and k. $A_{l,j}$ and $A_{l,k}$ are surface areas of the sphere S_i buried inside spheres S_j and S_k, respectively. $A_{l,jk}$ is the surface area of the sphere S_i buried inside intersection of three spheres.

In further discussion, we will use this relation for calculation of the surface area of sphere S_i included within sphere S_j at iteration step l, designated as $A_{l,j}$.

Intersection of Three Spheres

The spheres S_j and S_k from the neighbor list of the sphere S_i were considered to form triple intersection with sphere S_i, if $r_j + r_k > d_{jk}$ and $|r_j - r_k| < d_{jk}$. Figure 2 shows all possible types of the three-sphere intersections to be analyzed by the JNA algorithm. $A_{l,jk}$ designates the surface area of sphere S_i buried inside an intersection of neighboring spheres S_j and S_k, where subscript l indicates the iteration step and subscript jk indicates the two intersecting spheres from the neighbor list of S_i.

All the types of the triple intersections shown in Figure 2 can be separated into the two classes (possibly containing several subclasses) in accordance with the surface area of the sphere S_i buried inside the union of other two spheres:

1. The "true" triple intersection of the spheres when there are two points common to the surfaces of all three spheres as illustrated in panel A of Figure 2. This type of intersection satisfies the following set of requirements [eq. (2)]. In this case, $A_{l,jk}$ is calculated by Gibson and Scheraga method.[18]

$$\begin{cases} A_{l,jk} > 0 \\ A_{l,jk} \neq A_{l,j} \\ A_{l,jk} \neq A_{l,k} \\ A_{l,jk} \neq 4\pi r_i^2, \end{cases} \quad (2)$$

where $4\pi r_i^2$ is the surface of isolated sphere S_i.

2. All other variants of triple intersections are "false" because at least one of the requirements [eq. (2)] remains unsatisfied. However, they must be taken into account in the calculations since the surface of sphere S_i can be buried inside an intersection of two neighboring spheres.

These cases include (b) each of three spheres intersects each other but without a common triple intersection; (c1) sphere i contains the intersection of the other two spheres; (c2) spheres j or k contain the intersection of the other two spheres; (d1) sphere i is buried inside the union of the other two spheres; (d2) spheres j or k are buried inside the union of the other two spheres.

All these cases of triple intersections have been earlier described in details by Gibson and Scheraga[42] and are illustrated in Figure 2.

In case of the "true" triple intersection [panel A of Fig. 2], the area $A_{l,jk}$ can be calculated with the following formula:[18]

$$A_{l,jk} = 2r_i^2 \left\{ \tan^{-1}\left[\frac{d_{l,ik}\omega}{r_i q_2}(1-\varepsilon_2)\right] + \tan^{-1}\left[\frac{d_{l,ij}\omega}{r_i q_3}(1+\varepsilon_3)\right] \right\} - r_i \left\{ d_{l,ij}(1+\varepsilon_3)\tan^{-1}\left[\frac{2\omega}{q_3}\right] + d_{l,ik}(1-\varepsilon_2)\tan^{-1}\left[\frac{2\omega}{q_2}\right] \right\}, \quad (3)$$

where

$$\varepsilon_1 = \frac{r_{l,j}^2 - r_{l,k}^2}{d_{l,jk}^2}$$
$$\varepsilon_2 = \frac{r_{l,k}^2 - r_i^2}{d_{l,ik}^2} \quad (4)$$
$$\varepsilon_3 = \frac{r_i^2 - r_{l,j}^2}{d_{l,ij}^2},$$

$$q_1 = d_{l,jk}[d_{l,ik}^2 + d_{l,ij}^2 - d_{l,jk}^2 + r_{l,j}^2 + r_{l,k}^2 - 2r_i^2 + \varepsilon_1(d_{l,ik}^2 - d_{l,ij}^2)]$$
$$q_2 = d_{l,ik}[d_{l,ij}^2 + d_{l,jk}^2 - d_{l,ik}^2 + r_i^2 + r_{l,k}^2 - 2r_{l,j}^2 + \varepsilon_2(d_{l,ij}^2 - d_{l,jk}^2)]$$
$$q_3 = d_{l,ij}[d_{l,jk}^2 + d_{l,ik}^2 - d_{l,ij}^2 + r_i^2 + r_{l,j}^2 - 2r_{l,k}^2 + \varepsilon_3(d_{l,jk}^2 - d_{l,ik}^2)], \quad (5)$$

and

$$\Omega = \omega^2 = (r_i^2 d_{l,jk}^2 + r_{l,j}^2 d_{l,ik}^2 + r_{l,k}^2 d_{l,ij}^2)(d_{l,jk}^2 + d_{l,ik}^2 + d_{l,ij}^2) - 2(r_i^2 d_{l,jk}^4 + r_{l,j}^2 d_{l,ik}^4 + r_{l,k}^2 d_{l,ij}^4) + d_{l,jk}^2 d_{l,ik}^2 d_{l,ij}^2 \times (\varepsilon_1\varepsilon_2 + \varepsilon_2\varepsilon_3 + \varepsilon_1\varepsilon_3 - 1). \quad (6)$$

In case of the "false" triple intersections, the area cut $A_{l,jk}$ is calculated according to the following rules. In the cases b and c1, the area cut, $A_{l,jk} = 0$. In the cases c2 and d2, the area cut is equal to minimum of two areas cuts by sphere j and k separately, $A_{l,jk} = \min(A_{l,j}, A_{l,k})$. And in the case d1 area cut is equal to surface area of isolated sphere i, $A_{l,jk} = 4\pi r_i^2$.

Spheres Joining

In the JNA method, pairs of atoms from the neighbor list of the sphere S_i are joined based on the surface area of sphere S_i buried inside double and triple intersections of these neighbor spheres. A main point of joining is to obtain the surface area of the central sphere buried inside the resulting sphere, $A_{l+1,m}$, to be equal to that buried inside the union of the two preceding spheres. Here, the subscript $l+1$ indicates that the new sphere to be transferred to a next iteration step.

However, preliminary tests showed that in case of minor $A_{l,jk}$ values, the union of two joined spheres is poorly approximated by the resulting sphere because its shape is highly nonspherical. To avoid this problem, first we normalize $A_{l,jk}$ values by the minimal value of $A_{l,j}$ and $A_{l,k}$ areas for each pair of neighbors:

$$\min_A = \min(A_{l,j}, A_{l,k}),$$
$$\mathrm{nrm}A_{l,jk} = \frac{A_{l,jk}}{\min_A}; \quad (7.1)$$

and then pairs of neighbors are joined in the decreasing order of their $\mathrm{nrm}A_{l,jk}$ values. It was found that such rearrangement provides a better accuracy and minimal distortions of the shape of the area buried inside the union of the two joined spheres (Fig. 3).

It is noteworthy that use of the condition expressed by formula (7.1) requires long computational times, because of vast majority of the precalculated triple intersections of two spheres to be joined are invalidated by the spheres joining. The computational efficiency of the JNA method can be significantly improved by replacing expression (7.1) with a simple expression shown by formula (7.2) where only one triple intersection area $A_{l,jk}$ is calculated for each pair of joining spheres.

$$\max_A = \min(A_{l,j}, A_{l,k})$$
$$\rho = \frac{\max}{\min_A}\frac{1}{d_{l,jk}^2}. \quad (7.2)$$

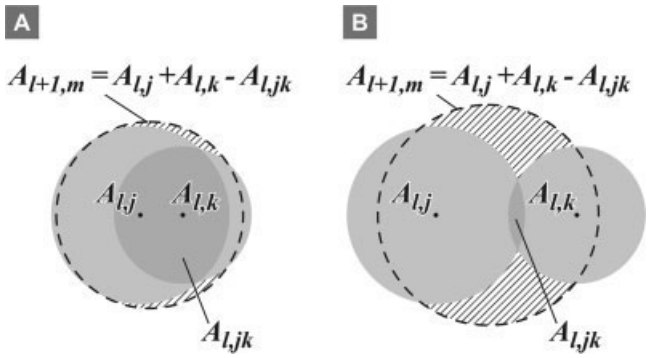


Figure 3. Conditions of two neighboring spheres joining. Surface areas of the sphere S_i buried both inside spheres S_j and S_k ($A_{l,j}$ and $A_{l,k}$, respectively) and inside their intersection, $A_{l,\,jk}$, are illustrated as a projection to a plane. Panel A shows the case where the area of intersection $A_{l,jk}$ takes the most part of intersection $A_{l,k}$. In this case, the system of two circles can be accurately approximated by a circle, $A_{l+1,m}$, indicated by dotted line. The area of circle $A_{l+1,m}$ is equal to area of the union of circles $A_{l,j}$ and $A_{l,k}$. Panel B shows the case where large part of the union of $A_{l,j}$ and $A_{l,k}$ circles is outside of the circle $A_{l+1,m}$. Additionally in this case the circle $A_{l+1,m}$ contains large parts of surface of the sphere S_i, that do not belong to the union of $A_{l,j}$ and $A_{l,k}$ circles (marked by shading). Therefore, in the case B the approximation of two circles by one circle is not sufficiently accurate.

However, the elevated speed of the algorithm comes at expense of somewhat decreased accuracy of results produced by the JNA method (see Results and Discussion section). In the further discussions, the results of the JNA algorithm are obtained using more accurate formula (7.1).

The surface areas of the sphere S_i buried inside a new sphere, $A_{l+1,m}$, are calculated in accordance with the following rules:

In the case of "true" triple intersections (panel A of the Fig. 2),

$$A_{l+1,m} = A_{l,j} + A_{l,k} - A_{l,jk}. \quad (8.1)$$

In the case of "false" triple intersections (panels C2 and D2 of Fig. 2),

$$A_{l+1,m} = \max(A_{l,j}, A_{l,k}). \quad (8.2)$$

In the case of "false" triple intersections (panels B and C1 of Fig. 2), two spheres from the neighbor list of the sphere S_i are not joined. If such two spheres have no triple intersection with other spheres from neighbor list of S_i, they are transferred to the next iteration step, $l+1$, without any transformations, otherwise they are joined with neighboring spheres having nonzero triple intersection.

In the case d1, where the sphere S_i is completely buried inside a union of two neighboring spheres, zero value is assigned to the SASA of the sphere S_i and the algorithm proceeds to the next sphere S_{i+1}.

After joining of two neighbor spheres, Cartesian coordinates of a new sphere are calculated in a way that its center is located on the surface of the central sphere S_i in a point of average weighted distances from the centers of preceding spheres. The

coordinates of the resulting joint sphere $S_{l+1,m}$ are calculated by the two consequent coordinate transformations:

$$(x_{l+1,m}, y_{l+1,m}, z_{l+1,m})^* = (x_{l,j}, y_{l,j}, z_{l,j})\ \mathrm{nrm}A_{l,jk} + (x_{l,k}, y_{l,k}, z_{l,k}) \times (1 - \mathrm{nrm}A_{l,jk}) \quad \text{if} \quad A_{l,j} > A_{l,k}$$

or

$$(x_{l+1,m}, y_{l+1,m}, z_{l+1,m})^* = (x_{l,k}, y_{l,k}, z_{l,k})\ \mathrm{nrm}A_{l,jk} + (x_{l,j}, y_{l,j}, z_{l,j}) \times (1 - \mathrm{nrm}A_{l,jk}) \quad \text{if} \quad A_{l,k} > A_{l,j}. \qquad (9)$$

After this transformation, the center of the sphere $S_{l+1,m}$ will be placed between the centers of the two joined spheres closer to the one that makes bigger contribution to the area of triple intersection. Designating the distance between the centers of new sphere and sphere S_i as $d_{l+1,m}$, one can express the transformation that locates the center of the sphere $S_{l+1,m}$ on the surface of sphere S_i by:

$$(x_{l+1,m}, y_{l+1,m}, z_{l+1,m}) = (x_{l+1,m}, y_{l+1,m}, z_{l+1,m})^* \frac{r_i}{d_{l+1,m}}. \qquad (10)$$

This way of the coordinates shifting for the sphere $A_{l+1,m}$ was found to significantly decrease the number of "false" triple intersections at the next iteration step. Since the distance between centers of the sphere S_i and new neighboring sphere, $d_{l+1,m}$, is equal to radius of the S_i sphere (r_i) using formula (1) the radius of the sphere $S_{l+1,m}$ can be expressed as:

$$r_{l+1,m} = \sqrt{\frac{A_{l+1,m}}{\pi}}. \qquad (11)$$

Stop Condition of Spheres Joining

The process of joining of neighbor spheres continues until at least one nonzero triple intersection exists. If no nonzero triple intersections remain, the list of neighbors is transferred to the next iteration step. Only two criteria are used to stop the sphere joining cycle followed by calculations of SASA for the S_i sphere, designated as A_i: (a) the sphere S_i is completely buried inside union of two neighboring spheres (panel D1 of Fig. 2); then $A_i = 0$; and (b) there are no nonzero triple intersections and none of the spheres from the neighbor list was joint at current iteration step l, in which case

$$A_i = 4\pi r_i^2 = \sum_m A_{l,m}, \qquad (12)$$

where sum is over all the number-neighboring spheres at the last iteration step l. Having calculated A_i, algorithm proceeds to the next central sphere S_{i+1}.

Data Filtering for Accuracy Improvement

To improve the accuracy of our method, we utilize a filter preventing joining of two spheres if the normalized value of $\mathrm{nrm}A_{l,jk}$ formed by the spheres is below a certain limit. Three threshold values were employed: 0.5, 0.12, and 0.01. Initially, the value of 0.5 is used. In this case, two neighbors of the sphere S_i are allowed to joint if $\mathrm{nrm}A_{l,jk} > 0.5$. If none of neighboring pairs satisfies this condition, the threshold value is decreased. Finally, two neighbors are considered to have no triple intersection if their $\mathrm{nrm}A_{l,jk} < 0.01$. The filtering prevents significant perturbations of the sphere S_i neighborhood because, as has been discussed earlier, two spheres with a greater normalized intersection are better approximated by one sphere than two spheres with a smaller normalized intersection (Fig. 3).

Partial contributions to ASA derivatives with respect to atom coordinates are calculated at every iteration step. The analytical formulas are described in the Appendix section.

Table 2. The Compounds of Different Chemical Nature and Structure Used in Our Calculations.

chb[a]	3-Chloro-4-hydroxybenzoate ($C_7H_5O_3Cl$) in complex with protocatechuate 3,4-dioxygenase (Brookhaven entry 3pch)
ctc	7-Chlorotetracycline ($C_{22}H_{23}N_2O_8Cl$) in complex with tetracycline repressor from *E. coli* (Brookhave nentry 2tct)
nmx	Nitromethyldethia coenzyme A ($C_{22}H_{37}N_8O_{18}P_3$) in complex with chicken citrate synthase complex (Brookhaven entry 1amz)
1van	Vancomycin complex with L-Lys-D-Ala-D-Ala (theoretical model)
1crn	Crambin
103d	DNA (5′-(D*GP*TP*GP*GP*AP*AP*TP*GP*GP* AP*AP*C)-3′) (antiparallel DNA duplex; human centromere repeat)
2ins	Bovine insuline (Bos taurus)
163d	Rev responsive element (RBE, 30 ribonucleotide fragment) in complex with HIV rev protein (residues 34–50)
1lz1	Human lysozyme
2stw	Human ETS/DNA complex
2tra	Transfer ribo-nucleic acid (yeast,Asp) with spermine ($C_{10}H_{26}N_4$)
1sbg	HIV-1 protease in complex with the inhibitor sb203386
5tra	Transfer ribonucleic acid (yeast); metal atom in pdb file (M7, atom 255) was not taken into account
1inc	porcine pancreatic elastase in complex with benzoxazinone inhibitor ($C_{17}H_{22}N_2O_4Cl$)
1kvd	Killer toxin from halotolerant yeast
3app	Fungus acid proteinase (penicillopepsin)
1pgb[b]	B1 Ig-binding domain of streptococcal protein G
1nyf	SH3 domain from human Fyn proto-oncogene tyrosine kinase
1csp	Major cold-shock protein from *Bacillus subtilis*
1coa	Chymotrypsin inhibitor (I76V mutant)
1lmb	Lambda repressor/operator complex
2abd	Acyl-coenzyme A binding protein from bovine liver
1aps	Acylphosphatase from horse muscle
1brn	Barnase from *Bacillus amyloliquefaciens*

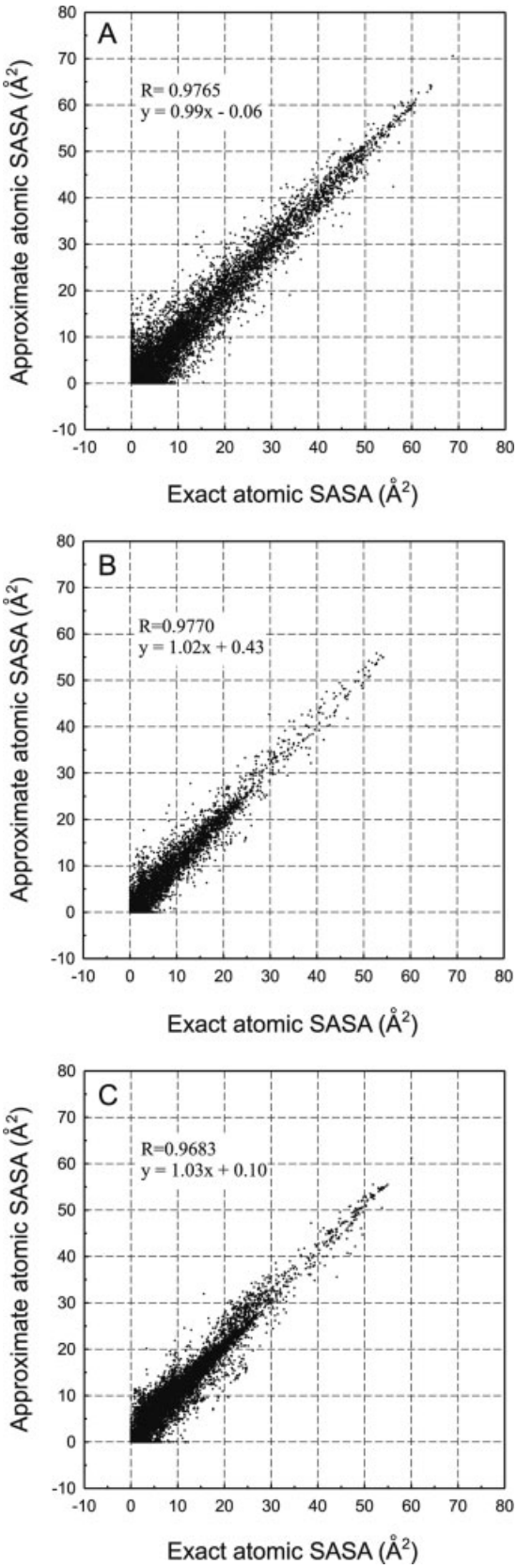
All acronyms correspond to Brookhaven protein data bank (PDB) entries.[39] For PDB structure files containing more than one set of atomic coordinates, the first set was taken. Only the protein atom coordinates are taken.

[a]The series of compounds taken from work by Weiser et al.[33]

[b]The proteins with varying content of secondary structure (taken from work by Guvench and Brook[36]). Only the protein atom coordinates are taken.

Results and Discussion

To evaluate accuracy of the proposed method, we have compared the results of SASA calculations by exact[14] and approxi-



mate methods for molecules of different nature including: the series of small organic molecules, RNA, DNA, and globular proteins in both folded and unfolded conformations. The series of compounds were taken from two studies, where alternative methods for approximate SASA calculation have been proposed and tested.[33,36] The list of the compounds used in these calculations are shown in Table 2. Note that the theoretical model of sipholenol-A monoacetate (sip) used by Weiser et al.[33] was excluded from the list because the structure of this compound is unavailable in public access. Since some recent reports[36] as well as our preliminary calculations showed that the accuracy of approximate SASA calculations differs for proteins in folded versus unfolded states we tested JNA method on both these two cases of protein conformations. Proteins in unfolded conformations were obtained as described in Materials and Methods section. The results of calculations are summarized in Tables 3–5 and illustrated by scatter plots in Figures 4 and 5.

Figure 4 shows the correlation between results of exact atomic SASA calculations[14] and that obtained by our method for the representative set of molecules used by Weiser et al.[33] and the proteins in both folded and unfolded conformations used by Guvench and Brooks.[36] The results show a very high correlation between the exact and the approximate SASA calculations at the level of 0.97–0.98 with slope close to 1.0. The correlation coefficients obtained for folded proteins and other compounds shown in Figure 4 (panels A and B) are almost identical and for unfolded proteins shown in Figure 4 (panel C) are only slightly lower, indicating similar accuracy of the JNA method for folded and unfolded molecules.

Table 3 summarizes the performance of the JNA method when compared with exact[14] and LCPO[33] SASA calculations. The results shows that total molecular SASA differs from the exact one by 0.05% (1inc) to 7.30% (ctc). Figure 5 (panel A) shows a very high correlation between the exact and the approximate total molecular SASA. One can see from these data that although approximate values have minor positive and negative deviations from exact values, the correlation coefficient is very close to 1.0. The average absolute atomic SASA errors range from 1.20 Å^2 (3app) to 3.47 Å^2 (chb) and the maximum absolute atomic errors range from 5.09 Å^2 (nmx) to 18.91 Å^2 (1kvd). Among the tested compounds, the maximum standard deviation between approximate atomic SASA and exact SASA is 3.57 Å^2 (chb) (the only one greater than 3.0 Å^2), while the smallest one is 2.13 Å^2 (103d) (Table 3). An interesting peculiarity is that the accuracy of JNA is rather better for large molecules than for small molecules. As expected, atoms of small molecules possess a relatively small number of neighboring atoms having common intersections with them. Therefore, in such molecular systems, the joining of two neighbors perturbs the environment of a

Figure 4. The correlation plot of JNA approximate versus exact atomic SASAs for all atoms in two data sets: (A) the series of compounds of different chemical nature and structure (taken from work by Weiser et al.[33]); (B) series of the folded proteins; and (C) series of the unfolded proteins from the representative set used by Guvench and Brooks.[36] The unfolded conformations of the proteins were obtained as described in Materials and Methods section.

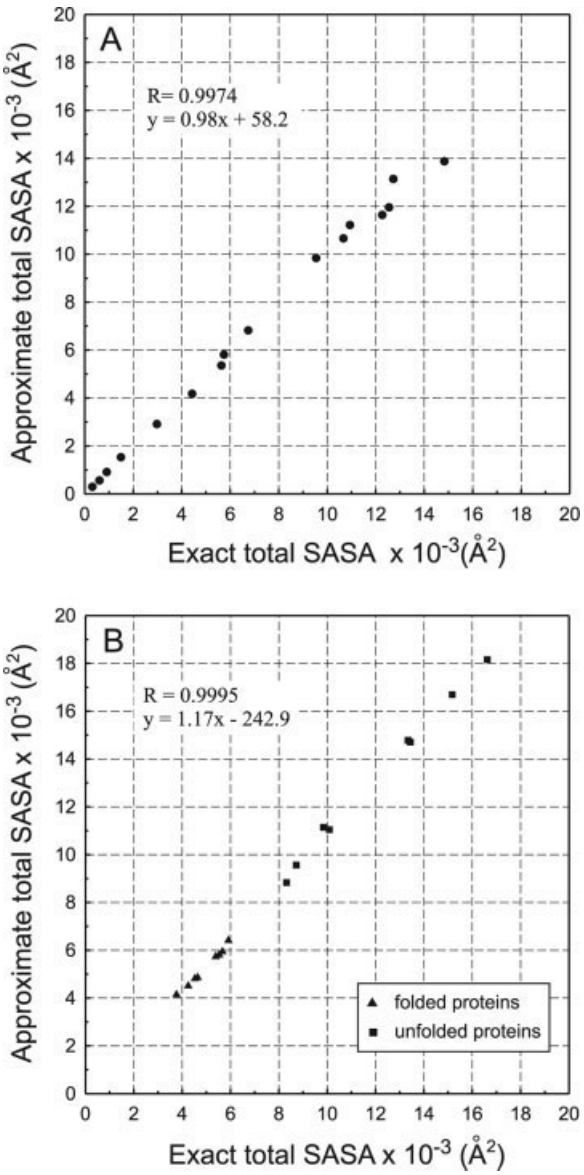


Figure 5. The correlation plot of JNA approximate versus exact total SASAs for two data sets: (A) series of compounds of different chemical nature and structure (taken from work by Weiser et al.[33]); (B) series of the folded and unfolded proteins from representative set used by Guvench and Brooks.[36] The regression line equations and the correlation coefficients are shown in the figure.

sphere under consideration to a greater extent, compared to the large molecules having more populated neighborhood. It is noteworthy that the JNA method generally produces more accurate average absolute atomic deviations than LCPO method[33] (Table 3). For small organic molecules, the accuracy of JNA method and that by LCPO are approximately at the same level. However, for macromolecules, the average absolute atomic errors are always lower than those produced by LCPO, showing a distinct trend decreasing an average absolute atomic error as the total number of atoms in a molecular system under consideration increases. The differences in maximal absolute atomic deviations between these two methods have similar values: the maximal difference is 18.9 Å^2 *vs.* 19.6 Å^2 and the minimal one is 5.1 Å^2 *vs.* 6.4 Å^2 for JNA *vs.* LCPO, respectively. Additionally, the differences in total molecular SASA obtained by these two methods are comparable too: the maximal difference is 7.3% *vs.* 7.8% and the minimal one is 0.05% *vs.* 0.1% for JNA *vs.* LCPO, respectively.

Our preliminary SASA calculations using LCPO for the series of proteins in unfolded conformations showed a significant decrease of accuracy when compared with the proteins in folded conformations. This is not surprised since the parameters of LCPO were optimized for series of folded macromolecules. Recently, Guvench and Brooks[36] (designated in further discussions as G&B) have reported a method for approximate SASA calculations and presented data on its accuracy for both folded and unfolded proteins. Moreover, all these proteins used in both conformations contained hydrogen atoms. Tables 4 and 5 show the correlation between results obtained by JNA and G&B methods.

For proteins in folded conformations, JNA show total SASA errors in the range of 3.0% (1aps) to 8.52% (1pgb). The average absolute atomic SASA errors range from 0.78 Å^2 (1brn) to 0.97 Å^2 (1csp) and the maximum absolute atomic errors vary from 10.61 Å^2 (1coa) to 14.86 Å^2 (1lmb). The maximum standard deviations between the approximate atomic SASA and exact SASA vary from 1.58 Å^2 (1brn) to 1.85 Å^2 (1nyf). As seen from Table 4, the results of atomic SASA calculations produced by JNA for proteins in native conformations are more accurate, when compared with those obtained by G&B method. The average absolute atomic errors produced by JNA are below 1.0 Å^2. The differences in maximal absolute atomic deviations obtained by JNA are also somewhat better: the maximal value is 14.9 Å^2 *vs.* 21.0 Å^2 and the minimal one is 10.6 Å^2 *vs.* 11.9 Å^2 for JNA *vs.* G&B method, respectively. At the same time, the differences in total molecular SASAs obtained by these two methods are quite comparable: the maximal difference is 8.5% *vs.* 9.5% and the minimal one is 3.0% *vs.* 1.3% for JNA *vs.* G&B, respectively.

From the data shown in the Tables 4 and 5, one can conclude that similarly to G&B and LCPO, our method also produced somewhat elevated errors for unfolded proteins. This effect is probably because of the property of the JNA method to produce more accurate results for atoms having a larger number of neighbors that result in more efficient canceling out of the SASA approximation errors. The JNA results and their comparison with that by G&B for a set of unfolded proteins are shown in Table 5. The total approximate SASAs deviate from exact ones in the range of 2.37% (1pgb) to 8.63% (1coa). The average absolute atomic errors range from 1.65 Å^2 (1pgb) to 1.87 Å^2 (1brn, 1aps) and the maximum absolute atomic errors vary from 9.47 Å^2 (1coa) to 16.78 Å^2 (1brn). The maximum standard deviation varies between 2.31 Å^2 (1coa) and 2.63 Å^2 (1aps). Comparing the results obtained by JNA and those obtained by G&B, one can see that JNA is more accurate in terms of both the absolute average atomic errors (the minimal error produced by G&B is 2.10 Å^2 while the maximal error by JNA is 1.87 Å^2) and the absolute maximum atomic errors (the maximal value is 16.78 Å^2 *vs.* 23.35 Å^2 and the minimal one is 9.47 Å^2 *vs.* 12.88 Å^2 for

Table 3. Comparison of SASAs Calculated by the Exact and the Approximate Methods for Representative Set of Molecules[a].

Compound	No. of atoms	Exact SASA[b] (Å^2)	JNA SASA (Å^2)	JNA total diff.[c] (%)	Atomic SASA errors			
					JNA ave. abs. (Å^2)	JNA max. abs. (Å^2)	JNA sigma (Å^2)	LCPO ave. abs.[d] (Å^2)
chb	11	316.63	295.55	−6.66	3.47	7.54	3.57	2.3
ctc	33	608.44	564.00	−7.30	2.45	5.68	2.49	2.7
nmx	51	906.83	911.30	0.49	1.97	5.09	2.46	2.4
1van	121	1495.37	1531.42	2.41	1.82	9.01	2.62	2.6
1crn	327	2976.46	2915.22	−2.06	1.64	13.49	2.70	2.0
103d	500	4427.05	4175.21	−5.69	1.52	7.51	2.13	1.6
2ins	770	5741.12	5807.95	1.16	1.60	14.35	2.85	2.2
163d	813	5635.74	5355.93	−4.96	1.32	11.98	2.16	1.9
1lz1	1029	6741.04	6821.73	1.20	1.35	16.98	2.47	2.0
2stw	1488	12266.77	11637.03	−5.13	1.90	15.07	2.94	2.3
2tra	1574	12549.34	11953.42	−4.75	1.54	16.70	2.54	2.0
1sbg	1559	9539.02	9839.52	3.15	1.48	18.00	2.74	2.1
5tra	1821	14830.03	13875.02	−6.44	1.55	12.67	2.43	2.0
1inc	1846	10665.76	10660.86	−0.05	1.23	18.25	2.38	1.9
1kvd	1988	10934.76	11218.10	2.59	1.26	18.91	2.40	2.0
3app	2366	12722.02	13136.18	3.26	1.20	17.66	2.38	1.9

[a]Taken from work by Weiser et al.[33]
[b]Calculated with Richmond's algorithm.[14]
[c]Calculated as 100% × (approximate – exact)/exact.
[d]Results obtained by Weiser et al.[33] with LCPO/NLR algorithm.

JNA *vs.* G&B, respectively), while G&B produces more accurate results in calculation of total molecular SASA (the minimal difference is 2.37% *vs.* 0.30% and the maximal one is 8.63% *vs.* 6.66% for JNA *vs.* G&B, respectively). Thus, JNA produces higher average absolute atomic errors for proteins in unfolded state than those for proteins in native states, but the deviations from the total SASA and the maximum absolute atomic errors stay in the same range for both folded and unfolded proteins.

One can see from Figure 5 (panel B) that despite a very high correlation coefficient ($R = 0.999$), JNA tends to minor overestimation of the total molecular SASA both for proteins in folded and unfolded states in all-atom representation. However, because of so high correlation between exact and approximate total SASA, more accurate results can be obtained by a simple linear transformation (SASA = (SASA + 243)/1.17). The minor overestimation of the total SASA for proteins with explicitly considered hydrogen atoms can be explained by a peculiarity in the

Table 4. Comparison of SASAs Calculated by the Exact and the Approximate Methods for Folded Proteins.[a]

Compound	No. of atoms	Exact SASA[b] (Å^2)	JNA SASA (Å^2)	JNA total diff.[c] (%)	Atomic SASA errors			
					JNA ave. abs (Å^2)	JNA max. abs. (Å^2)	JNA sigma (Å^2)	G & B ave. abs.[d] (Å^2)
1pgb	853	3772.04	4093.32	8.52	0.92	14.77	1.81	1.31
1nyf	903	4255.29	4429.48	4.09	0.96	14.71	1.85	1.29
1csp	1011	4637.78	4907.42	5.81	0.97	10.77	1.82	1.32
1coa	1055	4514.55	4838.85	7.18	0.91	10.61	1.72	1.38
1lmb	1350	5388.89	5700.53	5.78	0.86	14.86	1.74	1.27
2abd	1389	5541.97	5776.20	4.23	0.82	12.92	1.61	1.28
1aps	1547	5697.37	5868.33	3.00	0.80	13.52	1.62	1.26
1brn	1697	5921.91	6312.66	6.60	0.78	10.73	1.58	1.20

[a]Taken from work by Guvench and Brooks.[36]
[b]Calculated with Richmond's algorithm.[14]
[c]Calculated as 100% × (approximate – exact)/exact.
[d]Results obtained by Guvench and Brooks[36] at 298 K.

Table 5. Comparison of SASAs Calculated by the Exact and the Approximate Methods for Unfolded Proteins.[a]

Compound	No. of atoms	Exact SASA[b] (Å^2)	JNA SASA (Å^2)	JNA total diff.[c] (%)	Atomic SASA Errors			
					JNA ave. abs. (Å^2)	JNA max. abs. (Å^2)	JNA sigma (Å^2)	G&B ave. abs.[d] (Å^2)
1pgb	853	8014.61	8204.77	2.37	1.65	11.68	2.37	2.21
1nyf	903	8458.98	9036.07	6.82	1.71	11.18	2.34	2.17
1csp	1011	9799.06	10472.18	6.87	1.83	11.26	2.57	2.1
1coa	1055	9563.60	10389.38	8.63	1.72	9.47	2.31	2.16
1lmb	1350	12955.12	13977.51	7.89	1.79	13.14	2.48	2.32
2abd	1389	13039.96	13965.75	7.10	1.69	12.51	2.34	2.22
1aps	1547	14723.88	15766.81	7.08	1.87	16.35	2.63	2.22
1brn	1697	16108.51	17186.80	6.69	1.87	16.78	2.60	2.18

[a]Taken from work by Guvench and Brooks.[36]
[b]Calculated with Richmond's algorithm.[14]
[c]Calculated as: 100% × (approximate – exact)/exact.
[d]Results obtained by Guvench and Brooks[36] at 800 K.

proposed mechanism of neighbors joining. When a pair of spheres is joined, center of a new sphere is located on the surface of S_i between centers of preceding spheres (closer to that with a greater double intersection value). As has been discussed in the Intersection of Three Spheres section, in the case of "false" triple intersections (shown at panels C2 and D2 of Fig. 2), new sphere will be identical to one of preceding spheres, but its position will slightly change. In presence of hydrogen atoms, there are many of such cases and therefore this peculiarity can be a source of minor overestimations in SASA. Unfortunately, in the present version of JNA algorithm, a simple elimination of sphere with smaller value of double intersection is impossible, because it collects contributions to partial derivatives with respect to coordinates of atoms.

Comparing results presented in Tables 3–5, one can see that presence of hydrogen atoms in molecules significantly increases the accuracy of the JNA method. It can be explained by at least two reasons. First, introduction of additional covalently bonded atoms significantly changes the neighborhood density of every atom in a molecule and thus provides more efficient canceling out SASA calculation errors. Also, the hydrogen atoms have smaller van der Waals radii when compared with that of heavy atoms with relatively similar radii, and shorter distances of covalent bonds. Therefore, hydrogen atoms generate double and triple intersections with relatively small weighted values (formula 7.1). Addition of triple intersections with small weighted values decreases the extent of perturbation in neighborhood distribution after joining of two neighbors, which result in a significant increase of JNA accuracy.

Presence of hydrogen atoms in macromolecules is essential for many physical interactions, such as hydrogen bonding, hydration, and intraprotein cavity filling. Therefore, molecular dynamics simulations are normally carried out using all atom representation models. As has been recently shown, the free energy of water-protein H-bonding can be approximated based on a solvent accessibility of protein's polar or charged groups.[43] That is why, in all atom representation, the hydrogen atoms are essential for any molecular modeling including SASA calculations.

Indirect quantitative comparison of computational efficiencies of our and other methods for approximate SASA calculations is complicated by uncertainties in hardware and software environment. Nevertheless such a comparison can be done using GETAREA,[17] a publicly available exact method for molecular surface calculations, which computational efficiency has been compared against several approximate SASA algorithms.[33,35–37] The benchmark test was performed on a PC with AMD AthlonXP 3200+ processor with FORTRAN-90 implementation of the algorithm compiled at the/optimize:3 level with Compaq Visual Fortran v6.6 compiler. GETAREA code has been kindly provided by Dr. W. Braun and was compiled using the same compiler options. Similarly to other works[33,35–37] the protein penicillopepsin (PDB code: 3app) in all-atom representation with 4549 atoms was used for the CPU time tests. Atomic radii were assigned in accordance with Guvench and Brooks[36] work. The CPU times of GETAREA and JNA algorithms were averaged over five calls.

The results of tests show that GETAREA calculations of SASA for 3app take ~0.790 s. While the JNA with condition of sphere joining given by formula (7.1) takes 1.638 s (including 0.094 s for calculation of the matrix of interatomic distances using cubic lattice algorithm,[17] 0.222 s for the 3NLR[41] algorithm, and 1.322 s for the JNA algorithm itself). The calculations of derivatives of SASA with respect to atomic coordinates by the JNA algorithm take additional 0.532 s. Use of spheres joining condition given by formula (7.2) shows slightly less accurate results (for 3app in all-atom representation, deviation of the total approximate SASA from exact SASA is 5.60% *vs.* 6.74%, the average absolute atomic error is 1.11 Å^2 *vs.* 0.64 Å^2, and the maximum absolute atomic error is 22.8 Å^2 *vs.* 11.71 Å^2), but CPU time is shorter by factor of two (0.575 s for JNA excluding times required for formation of the matrix of interatomic distances and the 3NLR algorithm). In this case, the calculations of SASA derivatives with respect to atomic coordinates

take additional 0.113 s. Thus, the computational efficiency of this version of JNA algorithm is similar to that of GETAREA algorithm.

It is noteworthy that in indirect comparison of the competitive approaches for approximate SASA calculations done with the same protein shows 4 times (LCPO[33]), 15 times (TDND[35]), and 45 times (Guvench and Brooks[36]) shorter CPU times than that by GETAREA (all excluding times required for calculations of the matrix of interatomic distances).

This is the first version of JNA algorithm, which was mainly developed to check an accuracy of the method, and hence, complete attention was not paid to its computational efficiency. We believe that the efficiency of JNA could be significantly improved. As was shown, flexibility in terms of choosing different conditions of sphere joining is an additional reserve for JNA computational efficiency. The tests show that computation efficiency of the JNA method is comparable with that by the fastest exact analytical GETAREA method implicating restrictions on JNA use in conventional energy minimization and molecular dynamics simulations.

However, it is noteworthy that trade accuracy for speed was not a primary goal in the development of this method. Unlike several other approximate SASA algorithms, we attempted to develop a sufficiently accurate nonparametric method suitable for a global optimization using the highly efficient interval analysis algorithms.[44–46] In this approach, instead of exact atomic spatial positions the molecular conformations are represented by ranges of allowed atomic Cartesian coordinates and therefore the complicated analysis of buried and solvent exposed vertices of intersecting atomic spheres used in the exact methods,[14,17] unfortunately is not applicable. On the other hand, in this case, the available approximate SASA algorithms are not directly applicable as well since their parameterization was verified for the series of protein conformations with limited patterns of neighboring atoms distributions. Instead the recursive summing of analytical expressions for double and triple sphere intersections, which are depended only on radiuses and distances of intersecting atoms may provide required rigorous nonparametric upper estimates for a range of SASA variations if atomic coordinates are allowed to vary in a certain range.

Conclusions

In this work, we present a new analytical approximate method for computing the SASA and its first-order derivatives with respect to atomic coordinates for molecules of different chemical nature including proteins, DNA, and small organic and inorganic molecules. Unlike the majority of earlier published methods for approximate SASA calculations, the JNA method is based on purely geometrical approach and therefore has no empirical parameters. Therefore, the method is applicable to any molecule regardless of its chemical nature or conformation. The method was tested on a large representative set of molecules. It was shown that for the globular proteins in folded conformations, the average error in absolute atomic surface area is around 1 Å^2, while for unfolded protein conformations, it varies from 1.65 to 1.87 Å^2 SASA per atom. The CPU times of the JNA method are comparable with that of the fastest exact analytical GETAREA[17] method. This implicates certain restrictions on JNA use in conventional energy minimization and molecular dynamics simulations. However, relative simplicity of mathematical formulations, its independence on chemical nature of molecules under consideration, absence of any empirical parameters and high accuracy for both folded and unfolded conformations makes it applicable in the novel and very promising field of efficient global optimization using interval analysis algorithms.[44–46]

Acknowledgments

We thank Prof. M. Cox, Prof. V. Lanzov and Dr. D. Lebedev for helpful discussions. We thank Prof. W. Braun for giving the source code of GETAREA algorithm. We also thank Dr. M. Petitjean for supplying a reprint of ref. 19.

Appendix

The following symbols are used:

j,k	neighbors of sphere S_i;
m	a neighboring sphere to sphere S_i resulted from joining of two other spheres;
l	iteration step number;
c_α	atomic Cartesian coordinate ($c = x$, y, or z);
$C_{l,j}$	Cartesian coordinate ($C = x$, y, or z) of sphere S_j at an iteration step l;
$C^*_{l,j}$	intermediate value of Cartesian coordinate;
r_i	radius of sphere S_i;
$r_{l,j}$	radius of sphere S_j at an iteration step l;
d_{ij}	distance between spheres S_i and S_j;
$d_{l,jk}$	distance between spheres S_j and S_k at an iteration step l;
$A_{l,j}$	surface area of sphere S_i buried inside sphere S_j at iteration step l;
$A_{l,jk}$	surface area of sphere S_i buried inside intersection of spheres S_j and S_k at an iteration step l;
$F()$	any function of a given list of arguments;
ε_1, ε_2, ε_3, q_1, q_2, q_3, Ω, ω, P_1–P_4, M_1, M_2, U_1, and U_2	auxiliary quantities.

As has been obtained in the JNA Method section, SASA of individual atom is calculated by formula (12):

$$A_i = 4\pi r_i - \sum_m A_{l,m},$$

where $A_{l,m}$ is a combined function of variables belonging to all the precursor spheres as can be seen from formulas (8.1) and (8.2). More specifically,

$$A_{l+1,m} = F(A_{l,j}, A_{l,k}, A_{l,jk}(d_{l,ik}, d_{l,ij}, d_{l,jk}, r_i, r_{i,j}, r_{l,k})),$$
$$A_{l,m} = F(A_{l-1,j}, A_{l-1,k}, A_{l-1,jk}(d_{l-1,ik}, d_{l-1,ij}, d_{l-1,jk}, r_i, r_{l-1,j}, r_{l-1,k})), \quad \text{(A1)}$$

and so on. Expanding the formula in reverse order of sphere, joining results in an individual spherical atom SASA, A_i, as a function of atomic coordinates and interatomic distances. Since the obtaining of A_i is a recursive procedure (see the JNA Method section), it is natural to calculate the A_i derivatives with respect to atomic coordinates by a similar recursive procedure. At first-iteration step, derivatives from the surface area of the sphere S_i buried inside sphere S_j are calculated by formula:

$$\frac{\partial A_{1,j}}{\partial c_\alpha} = \frac{dA_{1,j}}{dd_{ij}}\frac{\partial d_{ij}}{\partial c_\alpha} = \pi r_i\left(\frac{r_i^2 - r_j^2}{d_{ij}^2} - 1\right)\frac{\partial d_{ij}}{\partial c_\alpha} \quad \alpha = i \text{ or } j$$
$$\frac{\partial d_{ij}}{\partial c_i} = \frac{c_i - c_j}{d_{ij}};\ \frac{\partial d_{ij}}{\partial c_j} = -\frac{c_i - c_j}{d_{ij}} \tag{A2}$$

Since $A_{l,m}$ at next iteration steps are obtained based on previously calculated variables, it should be noted that the formula (A2) is used only at the first-iteration step. In the next steps of the JNA algorithm, the surface area of sphere S_i buried inside the sphere (obtained by joining spheres S_j and S_k) and its derivatives are calculated by formula:

$$\frac{\partial A_{l+1,m}}{\partial c_\alpha} = \frac{\partial A_{l,j}}{\partial c_\alpha} + \frac{\partial A_{l,k}}{\partial c_\alpha} - \frac{\partial A_{l,jk}}{\partial c_\alpha}, \tag{A3}$$

where $\frac{\partial A_{l,j}}{\partial c_\alpha}$ and $\frac{\partial A_{l,k}}{\partial c_\alpha}$ have been calculated in the previous step of the procedure. The remaining contribution to the derivative is calculated by formula:

$$\frac{\partial A_{l,jk}}{\partial c_\alpha} = \frac{dA_{l,jk}}{dd_{l,jk}}\frac{dd_{l,jk}}{dC_\beta}\frac{\partial C_\beta}{\partial c_\alpha} + \frac{dA_{l,jk}}{dd_{l,ik}}\frac{dd_{l,ik}}{dC_\beta}\frac{\partial C_\beta}{\partial c_\alpha} + \frac{dA_{l,jk}}{dd_{l,ij}}\frac{dd_{l,ij}}{dC_\beta}\frac{\partial C_\beta}{\partial c_\alpha}$$
$$+ \frac{dA_{l,jk}}{dr_{l,j}}\frac{dr_{l,j}}{dC_\beta}\frac{\partial C_\beta}{\partial c_\alpha} + \frac{dA_{l,jk}}{dr_{l,i}}\frac{dr_{l,i}}{dC_\beta}\frac{\partial C_\beta}{\partial c_\alpha}, \tag{A4}$$

where c_α are atomic Cartesian coordinates ($\alpha = x$, y, or z); C_β are Cartesian coordinates of spheres to be joined ($\beta = x$, y, or z), and j and k are neighbor-list indexes at an iteration step l.

The above-mentioned expression is based on the fact that the C_β are coordinates of spheres resulting from joining of precursor spheres and therefore they are variables depending on initial atomic coordinates. One can obtain the following relations for the terms participating in eq. (A4). Since the radius of the sphere S_i is constant,

$$r_i = \text{Const} \ \Rightarrow \frac{\partial r_i}{\partial c_\alpha} = 0,$$

then:

$$d_{l,jk} = \sqrt{(C_{l,j} - C_{l,k})^2},$$
$$\frac{\partial d_{l,jk}}{\partial C_{l,j}} = \frac{C_{l,j} - C_{l,k}}{d_{l,jk}},$$
$$\frac{\partial d_{l,jk}}{\partial C_{l,k}} = -\frac{C_{l,j} - C_{l,k}}{d_{l,jk}},$$
$$\frac{\partial d_{l,jk}}{\partial c_\alpha} = \frac{dd_{l,jk}}{dC_\beta}\frac{\partial C_\beta}{\partial c_\alpha}; \tag{A5}$$

while the terms $\frac{dd_{l,ij}}{dc_\beta}\frac{\partial c_\beta}{\partial c_\alpha}$ and $\frac{dd_{l,ik}}{dc_\beta}\frac{\partial c_\beta}{\partial c_\alpha}$ have been calculated at the previous step $l-1$. The derivatives of $A_{l,jk}$ with respect to $d_{l,ij}$, $d_{l,ik}$, $d_{l,jk}$, r_i, $r_{l,j}$, and $r_{l,k}$ at iteration step l are calculated by following formulas. Assuming that all the terms are taken at the iteration step l and omitting index l for simplicity, one can rewrite the formula (3) as:

$$\begin{aligned} A_{l,jk} &= F(d_{jk}, d_{ik}, d_{ij}, r_i, r_j, r_k) \\ &= 2r_i^2\left\{\tan^{-1}\left[\frac{d_{ik}\omega}{r_i q_2}(1-\varepsilon_2)\right] + \tan^{-1}\left[\frac{d_{ij}\omega}{r_i q_3}(1+\varepsilon_3)\right]\right\} \\ &\quad - r_i\left\{d_{ij}(1+\varepsilon_3)\tan^{-1}\left(\frac{2\omega}{q_3}\right) + d_{ik}(1-\varepsilon_2)\tan^{-1}\left(\frac{2\omega}{q_2}\right)\right\} \\ &= 2r_i^2\{\tan^{-1}(P_1) + \tan^{-1}(P_2)\} - r_i\{M_1\tan^{-1}(P_3) \\ &\quad + M_2\tan^{-1}(P_4)\} = U_1 - U_2. \end{aligned} \tag{A6}$$

Then

$$\frac{dF}{dd_{jk}} = 2r_i^2\left\{-\frac{1}{1-(P_1)^2}\frac{dP_1}{dd_{jk}} - \frac{1}{1-(P_2)^2}\frac{dP_2}{dd_{jk}}\right\} - r_i\left\{\frac{dM_1}{dd_{jk}}\tan^{-1}(P_3) - M_1\frac{1}{1-(P_3)^2}\frac{dP_3}{dd_{jk}} + \frac{dM_2}{dd_{jk}}\tan^{-1}(P_4) - M_2\frac{1}{1-(P_4)^2}\frac{dP_4}{dd_{jk}}\right\},$$
$$\frac{dF}{dd_{ik}} = 2r_i^2\left\{-\frac{1}{1-(P_1)^2}\frac{dP_1}{dd_{ik}} - \frac{1}{1-(P_2)^2}\frac{dP_2}{dd_{ik}}\right\} - r_i\left\{\frac{dM_1}{dd_{ik}}\tan^{-1}(P_3) - M_1\frac{1}{1-(P_3)^2}\frac{dP_3}{dd_{ik}} + \frac{dM_2}{dd_{ik}}\tan^{-1}(P_4) - M_2\frac{1}{1-(P_4)^2}\frac{dP_4}{dd_{ik}}\right\},$$
$$\frac{dF}{dd_{ij}} = 2r_i^2\left\{-\frac{1}{1-(P_1)^2}\frac{dP_1}{dd_{ij}} - \frac{1}{1-(P_2)^2}\frac{dP_2}{dd_{ij}}\right\} - r_i\left\{\frac{dM_1}{dd_{ij}}\tan^{-1}(P_3) - M_1\frac{1}{1-(P_3)^2}\frac{dP_3}{dd_{ij}} + \frac{dM_2}{dd_{ij}}\tan^{-1}(P_4) - M_2\frac{1}{1-(P_4)^2}\frac{dP_4}{dd_{ij}}\right\},$$
$$\frac{dF}{dr_i} = \frac{2U_1}{r_i} - \frac{U_2}{r_i} + 2r_i^2\left\{-\frac{1}{1-(P_1)^2}\frac{dP_1}{dr_i} - \frac{1}{1-(P_2)^2}\frac{dP_2}{dr_i}\right\} - r_i\left\{\frac{dM_1}{dr_i}\tan^{-1}(P_3) - M_1\frac{1}{1-(P_3)^2}\frac{dP_3}{dr_i} + \frac{dM_1}{dr_i}\tan^{-1}(P_4) - M_2\frac{1}{1-(P_4)^2}\frac{dP_4}{dr_i}\right\}$$
$$\frac{dF}{dr_j} = 2r_i^2\left\{-\frac{1}{1-(P_1)^2}\frac{dP_1}{dr_j} - \frac{1}{1-(P_2)^2}\frac{dP_2}{dr_j}\right\} - r_i\left\{\frac{dM_1}{dr_j}\tan^{-1}(P_3) - M_1\frac{1}{1-(P_3)^2}\frac{dP_3}{dr_j} + \frac{dM_2}{dr_j}\tan^{-1}(P_4) - M_2\frac{1}{1-(P_4)^2}\frac{dP_4}{dr_j}\right\}$$
$$\frac{dF}{dr_k} = 2r_i^2\left\{-\frac{1}{1-(P_1)^2}\frac{dP_1}{dr_k} - \frac{1}{1-(P_2)^2}\frac{dP_2}{dr_k}\right\} - r_i\left\{\frac{dM_1}{dr_k}\tan^{-1}(P_3) - M_1\frac{1}{1-(P_3)^2}\frac{dP_3}{dr_k} + \frac{dM_2}{dr_k}\tan^{-1}(P_4) - M_2\frac{1}{1-(P_4)^2}\frac{dP_4}{dr_k}\right\}, \tag{A7}$$

where

$$
\begin{aligned}
P_1 &= \frac{d_{ik}\omega}{r_i q_2}(1-\varepsilon_2)\\
\frac{dP_1}{dd_{jk}} &= \frac{r_i d_{ik}\left[q_2\frac{d\omega}{dd_{jk}} - \omega\frac{dq_2}{dd_{jk}}\right]}{(r_i q_2)^2}(1-\varepsilon_2)\\
\frac{dP_1}{dd_{ik}} &= \frac{\left(\omega + d_{ik}\frac{d\omega}{dd_{ik}}\right)}{r_i q_2}(1-\varepsilon_2) - \frac{\left[(1-\varepsilon_2)\frac{dq_2}{dd_{ik}} + \frac{d\varepsilon_2}{dd_{ik}}q_2\right]}{(r_i q_2)^2}d_{ik}\omega r_i\\
\frac{dP_1}{dd_{ij}} &= \frac{r_i d_{ik}\left[q_2\frac{d\omega}{dd_{ij}} - \omega\frac{dq_2}{dd_{ij}}\right]}{(r_i q_2)^2}(1-\varepsilon_2)\\
\frac{dP_1}{dr_i} &= \frac{\left(\frac{d\omega}{dr_i}(1-\varepsilon_2) - \omega\frac{d\varepsilon_2}{dr_i}\right)}{r_i q_2}d_{ik} - \frac{d_{ik}\omega\left[q_2 + r_i\frac{dq_2}{dr_i}\right]}{(r_i q_2)^2}(1-\varepsilon_2)\\
\frac{dP_1}{dr_j} &= \frac{r_i d_{ik}\left[q_2\frac{d\omega}{dr_j} - \omega\frac{dq_2}{dr_j}\right]}{(r_i q_2)^2}(1-\varepsilon_2)\\
\frac{dP_1}{dr_k} &= \frac{\left(\frac{d\omega}{dr_k}(1-\varepsilon_2) - \omega\frac{d\varepsilon_2}{dr_k}\right)}{r_i q_2}d_{ik} - \frac{d_{ik}\omega(1-\varepsilon_2)r_i\frac{dq_2}{dr_k}}{(r_i q_2)^2}
\end{aligned}
\tag{A8}
$$

$$
\begin{aligned}
P_2 &= \frac{d_{ij}\omega}{r_i q_3}(1+\varepsilon_3)\\
\frac{dP_2}{dd_{jk}} &= \frac{d_{ij}r_i\left(\frac{d\omega}{dd_{jk}}q_3 - \omega\frac{dq_3}{dd_{jk}}\right)}{(r_i q_3)^2}(1+\varepsilon_3)\\
\frac{dP_2}{dd_{ik}} &= \frac{d_{ij}r_i\left(\frac{d\omega}{dd_{ik}}q_3 - \omega\frac{dq_3}{dd_{ik}}\right)}{(r_i q_3)^2}(1+\varepsilon_3)\\
\frac{dP_2}{dd_{ij}} &= \frac{\left(\omega + d_{ij}\frac{d\omega}{dd_{ij}}\right)(1+\varepsilon_3) + d_{ij}\omega\frac{d\varepsilon_3}{dd_{ij}}}{r_i q_3} - \frac{d_{ij}\omega(1+\varepsilon_3)r_i\frac{dq_3}{dd_{ij}}}{(r_i q_3)^2}\\
\frac{dP_2}{dr_i} &= \frac{d_{ij}\left(\frac{d\omega}{dr_i}(1+\varepsilon_3) + \omega\frac{d\varepsilon_3}{dr_i}\right)}{r_i q_3} - \frac{d_{ij}\omega\left(q_3 + r_i\frac{dq_3}{dr_i}\right)}{(r_i q_3)^2}(1+\varepsilon_3)\\
\frac{dP_2}{dr_j} &= \frac{d_{ij}\left(\frac{d\omega}{dr_j}(1+\varepsilon_3) + \omega\frac{d\varepsilon_3}{dr_j}\right)}{r_i q_3} - \frac{d_{ij}\omega(1+\varepsilon_3)r_i\frac{dq_3}{dr_j}}{(r_i q_3)^2}\\
\frac{dP_2}{dr_k} &= \frac{d_{ij}r_i\left(\frac{d\omega}{dr_k}q_3 - \omega\frac{dq_3}{dr_k}\right)}{(r_i q_3)^2}(1+\varepsilon_3)
\end{aligned}
\tag{A9}
$$

$$
\begin{aligned}
M_1 &= d_{ij}(1+\varepsilon_3)\\
\frac{dM_1}{dd_{jk}} &= 0\\
\frac{dM_1}{dd_{ik}} &= 0\\
\frac{dM_1}{dd_{ij}} &= (1+\varepsilon_3) + d_{ij}\frac{d\varepsilon_3}{dd_{ij}}\\
\frac{dM_1}{dr_i} &= d_{ij}\frac{d\varepsilon_3}{dr_i}\\
\frac{dM_1}{dr_j} &= d_{ij}\frac{d\varepsilon_3}{dr_j}\\
\frac{dM_1}{dr_k} &= 0
\end{aligned}
\tag{A10}
$$

$$
\begin{aligned}
M_2 &= d_{ik}(1-\varepsilon_2)\\
\frac{dM_2}{dd_{jk}} &= 0\\
\frac{dM_2}{dd_{ik}} &= (1-\varepsilon_2) - d_{ik}\frac{d\varepsilon_2}{dd_{ik}}\\
\frac{dM_2}{dd_{ij}} &= 0\\
\frac{dM_2}{dr_i} &= -d_{ik}\frac{d\varepsilon_2}{dr_i}\\
\frac{dM_2}{dr_j} &= 0\\
\frac{dM_2}{dr_k} &= -d_{ik}\frac{d\varepsilon_2}{dr_k}
\end{aligned}
\tag{A11}
$$

$$
\begin{aligned}
P_3 &= \frac{2\omega}{q_3}\\
\frac{dP_3}{dd_{jk}} &= 2\frac{\frac{d\omega}{dd_{jk}}q_3 - \omega\frac{dq_3}{dd_{jk}}}{q_3^2}\\
\frac{dP_3}{dd_{ik}} &= 2\frac{\frac{d\omega}{dd_{ik}}q_3 - \omega\frac{dq_3}{dd_{ik}}}{q_3^2}\\
\frac{dP_3}{dd_{ij}} &= 2\frac{\frac{d\omega}{dd_{ij}}q_3 - \omega\frac{dq_3}{dd_{ij}}}{q_3^2}\\
\frac{dP_3}{dr_i} &= 2\frac{\frac{d\omega}{dr_i}q_3 - \omega\frac{dq_3}{dr_i}}{q_3^2}\\
\frac{dP_3}{dr_j} &= 2\frac{\frac{d\omega}{dr_j}q_3 - \omega\frac{dq_3}{dr_j}}{q_3^2}\\
\frac{dP_3}{dr_k} &= 2\frac{\frac{d\omega}{dr_k}q_3 - \omega\frac{dq_3}{dr_k}}{q_3^2}
\end{aligned}
\tag{A12}
$$

$$
\begin{aligned}
P_4 &= \frac{2\omega}{q_2}\\
\frac{dP_4}{dd_{jk}} &= 2\frac{\frac{d\omega}{dd_{jk}}q_2 - \omega\frac{dq_2}{dd_{jk}}}{q_2^2}\\
\frac{dP_4}{dd_{ik}} &= 2\frac{\frac{d\omega}{dd_{ik}}q_2 - \omega\frac{dq_2}{dd_{ik}}}{q_2^2}\\
\frac{dP_4}{dd_{ij}} &= 2\frac{\frac{d\omega}{dd_{ij}}q_2 - \omega\frac{dq_2}{dd_{ij}}}{q_2^2}\\
\frac{dP_4}{dr_i} &= 2\frac{\frac{d\omega}{dr_i}q_2 - \omega\frac{dq_2}{dr_i}}{q_2^2}\\
\frac{dP_4}{dr_j} &= 2\frac{\frac{d\omega}{dr_j}q_2 - \omega\frac{dq_2}{dr_j}}{q_2^2}\\
\frac{dP_4}{dr_k} &= 2\frac{\frac{d\omega}{dr_k}q_2 - \omega\frac{dq_2}{dr_k}}{q_2^2},
\end{aligned}
\tag{A13}
$$

where

$$
\begin{aligned}
\varepsilon_1 &= \frac{r_j^2 - r_k^2}{d_{jk}^2}\\
\frac{d\varepsilon_1}{dd_{jk}} &= -2\frac{r_j^2 - r_k^2}{d_{jk}^3}
\end{aligned}
$$

$$\begin{aligned}
&\frac{d\varepsilon_1}{dd_{ik}} = 0\\
&\frac{d\varepsilon_1}{dd_{ij}} = 0\\
&\frac{d\varepsilon_1}{dr_i} = 0\\
&\frac{d\varepsilon_1}{dr_j} = \frac{2r_j}{d_{jk}^2}\\
&\frac{d\varepsilon_1}{dr_k} = -\frac{2r_k}{d_{jk}^2}
\end{aligned} \tag{A14}$$

$$\begin{aligned}
&\varepsilon_2 = \frac{r_k^2 - r_i^2}{d_{ik}^2}\\
&\frac{d\varepsilon_2}{dd_{jk}} = 0\\
&\frac{d\varepsilon_2}{dd_{ik}} = -2\frac{r_k^2 - r_i^2}{d_{ik}^3}\\
&\frac{d\varepsilon_2}{dd_{ij}} = 0\\
&\frac{d\varepsilon_2}{dr_i} = -\frac{2r_i}{d_{ik}^2}\\
&\frac{d\varepsilon_2}{dr_j} = 0\\
&\frac{d\varepsilon_2}{dr_k} = \frac{2r_k}{d_{ik}^2}
\end{aligned} \tag{A15}$$

$$\begin{aligned}
&\varepsilon_3 = \frac{r_i^2 - r_j^2}{d_{ij}^2}\\
&\frac{d\varepsilon_3}{dd_{jk}} = 0\\
&\frac{d\varepsilon_3}{dd_{ik}} = 0\\
&\frac{d\varepsilon_3}{dd_{ij}} = -2\frac{r_i^2 - r_j^2}{d_{ij}^3}\\
&\frac{d\varepsilon_3}{dr_i} = \frac{2r_i}{d_{ij}^2}\\
&\frac{d\varepsilon_3}{dr_j} = -\frac{2r_j}{d_{ij}^2}\\
&\frac{d\varepsilon_3}{dr_k} = 0
\end{aligned} \tag{A16}$$

and

$$\begin{aligned}
&q_1 = d_{jk}[d_{ik}^2 + d_{ij}^2 - d_{jk}^2 + r_j^2 + r_k^2 - 2r_i^2 + \varepsilon_1(d_{ik}^2 - d_{ij}^2)]\\
&\frac{dq_1}{dd_{jk}} = \frac{q_1}{d_{jk}} + d_{jk}\left[-2d_{jk} + \frac{d\varepsilon_1}{dd_{jk}}(d_{ik}^2 - d_{ij}^2)\right]\\
&\frac{dq_1}{dd_{ik}} = d_{jk}\left[2d_{ik} + \frac{d\varepsilon_1}{dd_{ik}}(d_{ik}^2 - d_{ij}^2) + 2\varepsilon_1 d_{ik}\right] = 2d_{jk}d_{ik}(1 + \varepsilon_1)\\
&\frac{dq_1}{dd_{ij}} = 2d_{jk}d_{ij}(1 - \varepsilon_1)\\
&\frac{dq_1}{dr_i} = d_{jk}\left[-4r_i + \frac{d\varepsilon_1}{dr_i}(d_{ik}^2 - d_{ij}^2)\right] = -4d_{jk}r_i\\
&\frac{dq_1}{dr_j} = d_{jk}\left[2r_j + \frac{d\varepsilon_1}{dr_j}(d_{ik}^2 - d_{ij}^2)\right]\\
&\frac{dq_1}{dr_k} = d_{jk}\left[2r_k + \frac{d\varepsilon_1}{dr_k}(d_{ik}^2 - d_{ij}^2)\right]
\end{aligned} \tag{A17}$$

$$\begin{aligned}
&q_2 = d_{ik}[d_{ij}^2 + d_{ik}^2 - d_{jk}^2 + r_k^2 + r_i^2 - 2r_j^2 + \varepsilon_2(d_{ij}^2 - d_{jk}^2)]\\
&\frac{dq_2}{dd_{jk}} = 2d_{jk}d_{ik}(1 - \varepsilon_2)\\
&\frac{dq_2}{dd_{ik}} = \frac{q_2}{d_{ik}} + d_{ik}\left[-2d_{ik} + \frac{d\varepsilon_2}{dd_{ik}}(d_{ij}^2 - d_{jk}^2)\right]\\
&\frac{dq_2}{dd_{ij}} = 2d_{ik}d_{ij}(1 + \varepsilon_2)\\
&\frac{dq_2}{dr_i} = d_{ik}\left[2r_i + \frac{d\varepsilon_2}{dr_i}(d_{ij}^2 - d_{jk}^2)\right]\\
&\frac{dq_2}{dr_j} = -4d_{ik}r_j\\
&\frac{dq_2}{dr_k} = d_{ik}\left[2r_k + \frac{d\varepsilon_2}{dr_k}(d_{ij}^2 - d_{jk}^2)\right]
\end{aligned} \tag{A18}$$

$$\begin{aligned}
&q_3 = d_{ij}[d_{jk}^2 + d_{ik}^2 - d_{ij}^2 + r_i^2 + r_j^2 - 2r_k^2 + \varepsilon_3(d_{jk}^2 - d_{ik}^2)]\\
&\frac{dq_3}{dd_{jk}} = 2d_{ij}d_{jk}(1 + \varepsilon_3)\\
&\frac{dq_3}{dd_{ik}} = 2d_{ij}d_{ik}(1 - \varepsilon_3)\\
&\frac{dq_3}{dd_{ij}} = \frac{q_3}{d_{ij}} + d_{ij}\left[-2d_{ij} + \frac{d\varepsilon_3}{dd_{ij}}(d_{jk}^2 - d_{ik}^2)\right]\\
&\frac{dq_3}{dr_i} = d_{ij}\left[2r_i + \frac{d\varepsilon_3}{dr_i}(d_{jk}^2 - d_{ik}^2)\right]\\
&\frac{dq_3}{dr_j} = d_{ij}\left[2r_j + \frac{d\varepsilon_3}{dr_j}(d_{jk}^2 - d_{ik}^2)\right]\\
&\frac{dq_3}{dr_k} = -4d_{ij}r_k
\end{aligned} \tag{A19}$$

and

$$\begin{aligned}
\Omega = \omega^2 = {}& (r_i^2 d_{jk}^2 + r_j^2 d_{ik}^2 + r_k^2 d_{ij}^2)(d_{jk}^2 + d_{ik}^2 + d_{ij}^2)\\
&- 2(r_i^2 d_{jk}^4 + r_j^2 d_{ik}^4 + r_k^2 d_{ij}^4) + d_{jk}^2 d_{ik}^2 d_{ij}^2\\
&(\varepsilon_1\varepsilon_2 + \varepsilon_2\varepsilon_3 + \varepsilon_1\varepsilon_3 - 1)\\
\frac{d\Omega}{dd_{jk}} = {}& 2d_{jk}[r_i^2(d_{jk}^2 + d_{ik}^2 + d_{ij}^2) + (r_i^2 d_{jk}^2 + r_j^2 d_{ik}^2 + r_k^2 d_{ij}^2)\\
&- 4r_i^2 d_{jk}^2 + d_{ik}^2 d_{ij}^2(\varepsilon_1\varepsilon_2 + \varepsilon_2\varepsilon_3 + \varepsilon_1\varepsilon_3 - 1)]\\
&+ d_{jk}^2 d_{ik}^2 d_{ij}^2(\varepsilon_2 + \varepsilon_3)\frac{d\varepsilon_1}{dd_{jk}}
\end{aligned}$$

$$\frac{d\Omega}{dd_{ik}} = 2d_{ik}[r_j^2(d_{jk}^2 + d_{ik}^2 + d_{ij}^2) + (r_i^2 d_{jk}^2 + r_j^2 d_{ik}^2 + r_k^2 d_{ij}^2) - 4r_j^2 d_{ik}^2 + d_{jk}^2 d_{ij}^2(\varepsilon_1\varepsilon_2 + \varepsilon_2\varepsilon_3 + \varepsilon_1\varepsilon_3 - 1)] + d_{jk}^2 d_{ik}^2 d_{ij}^2(\varepsilon_1 + \varepsilon_3)\frac{d\varepsilon_2}{dd_{ik}}$$

$$\frac{d\Omega}{dd_{ij}} = 2d_{ij}[r_k^2(d_{jk}^2 + d_{ik}^2 + d_{ij}^2) + (r_i^2 d_{jk}^2 + r_j^2 d_{ik}^2 + r_k^2 d_{ij}^2) - 4r_k^2 d_{ij}^2 + d_{jk}^2 d_{ik}^2(\varepsilon_1\varepsilon_2 + \varepsilon_2\varepsilon_3 + \varepsilon_1\varepsilon_3 - 1)] + d_{jk}^2 d_{ik}^2 d_{ij}^2(\varepsilon_1 + \varepsilon_2)\frac{d\varepsilon_3}{dd_{ij}}$$

$$\frac{d\Omega}{dr_i} = 2r_i d_{jk}^2(d_{ik}^2 + d_{ij}^2 - d_{jk}^2) + d_{jk}^2 d_{ik}^2 d_{ij}^2 \left[\frac{d\varepsilon_2}{dr_i}(\varepsilon_1 + \varepsilon_3) + \frac{d\varepsilon_3}{dr_i}(\varepsilon_1 + \varepsilon_2)\right]$$

$$\frac{d\Omega}{dr_j} = 2r_j d_{ik}^2(d_{jk}^2 + d_{ij}^2 - d_{ik}^2) + d_{jk}^2 d_{ik}^2 d_{ij}^2 \left[\frac{d\varepsilon_1}{dr_j}(\varepsilon_2 + \varepsilon_3) + \frac{d\varepsilon_3}{dr_j}(\varepsilon_1 + \varepsilon_2)\right]$$

$$\frac{d\Omega}{dr_k} = 2r_k d_{ij}^2(d_{jk}^2 + d_{ik}^2 - d_{ij}^2) + d_{jk}^2 d_{ik}^2 d_{ij}^2 \left[\frac{d\varepsilon_1}{dr_k}(\varepsilon_2 + \varepsilon_3) + \frac{d\varepsilon_2}{dr_k}(\varepsilon_1 + \varepsilon_3)\right] \quad \text{(A20)}$$

When calculations of $A_{l,jk}$ and $A_{l+1,m}$ derivatives at an iteration step l are completed, the algorithm proceeds to the calculations of derivatives for coordinates $C_{l+1,m}$ and radius $r_{l+1,m}$ of a new sphere $S_{l+1,m}$. To obtain the derivatives of coordinates, one has to calculate derivatives of normalized area $\mathrm{nrm}A_{l,jk}$:

$$\frac{\partial(\mathrm{nrm}A_{l,jk})}{\partial c_\alpha} = \frac{\frac{\partial A_{l,jk}}{\partial c_\alpha}\min_A - A_{l,jk}\frac{\partial \min_A}{\partial c_\alpha}}{\min_A^2}, \quad \text{(A21)}$$

where

$$\min_A = \min(A_{l,j}; A_{l,k}),$$
$$\mathrm{nrm}A_{l,jk} = \frac{A_{l,jk}}{\min_A}.$$

Then, after the first transformation [formula (9)] derivatives of coordinates and distance between a new sphere and the sphere S_i are calculated as:

$$\frac{\partial C_{l+1,m}^*}{\partial c_\alpha} = \frac{\partial C_{l,j}}{\partial c_\alpha}\mathrm{nrm}A_{l,jk} + C_{l,j}\frac{\partial \mathrm{nrm}A_{l,jk}}{\partial c_\alpha} + \frac{\partial C_{l,k}}{\partial c_\alpha}(1 - \mathrm{nrm}A_{l,jk}) - C_{l,k}\frac{\partial \mathrm{nrm}A_{l,jk}}{\partial c_\alpha} \quad \text{if} \quad A_{l,j} > A_{l,k},$$

or

$$\frac{\partial C_{l+1,m}^*}{\partial c_\alpha} = \frac{\partial C_{l,k}}{\partial c_\alpha}\mathrm{nrm}A_{l,jk} + C_{l,k}\frac{\partial \mathrm{nrm}A_{l,jk}}{\partial c_\alpha} + \frac{\partial C_{l,j}}{\partial c_\alpha}(1 - \mathrm{nrm}A_{l,jk}) - C_{l,j}\frac{\partial \mathrm{nrm}A_{l,jk}}{\partial c_\alpha} \quad \text{if} \quad A_{l,k} > A_{l,j}, \quad \text{(A22)}$$

and

$$d_{l+1,im}^* = \sqrt{(C_i - C_{l+1,m}^*)^2}$$
$$\frac{dd_{l+1,im}^*}{dC_i} = \frac{C_i - C_{l+1,m}^*}{d_{l+1,m}^*}$$
$$\frac{dd_{l+1,im}^*}{dC_{l+1,m}^*} = -\frac{C_i - C_{l+1,m}^*}{d_{l+1,m}^*}$$
$$\frac{\partial d_{l+1,im}^*}{\partial c_\alpha} = \frac{dd_{l+1,m}^*}{dC_\beta}\frac{\partial C_\beta}{\partial c_\alpha}. \quad \text{(A23)}$$

After the second sphere $S_{l+1,m}$ center's coordinates transformation [formula (10)], these derivatives are calculated as:

$$C_{l+1,m} = C_{l+1,m}^*\frac{r_i}{d_{l+1,im}^*},$$
$$\frac{\partial C_{l+1,m}}{\partial c_a} = r_i\frac{\left(\frac{\partial C_{l+1,m}^*}{\partial c_a}d_{l+1,im}^* - C_{l+1,m}\frac{\partial d_{l+1,im}^*}{\partial c_a}\right)}{(d_{l+1,im}^*)^2}, \quad \text{(A24)}$$

and derivatives of $d_{l+1,im}$ are equivalent to those for $d^*_{l+1,im}$, [formula (A23)] with an exception that the $C^*_{l+1,m}$ coordinates after the first transformation are replaced by $C_{l+1,m}$ coordinates after the second transformation.

Finally, derivatives of new spheres radius $r_{l+1,m}$ are obtained:

$$\frac{\partial r_{l+1,m}}{\partial c_a} = \frac{dr_{l+1,m}}{dA_{l+1,m}}\frac{\partial A_{l+1,m}}{\partial c_a} = \frac{1}{2\pi r_{l+1,m}}\frac{\partial A_{l+1,m}}{\partial c_a}. \quad \text{(A25)}$$

All the above terms are stored in computer memory for derivative calculations at the next iteration step.

References

1. Raschke, T. M.; Levitt, M. Proc Natl Acad Sci USA 2005, 102, 6777.
2. Pal, S. K.; Zhao, L.; Zewail, A. H. Proc Natl Acad Sci USA 2003, 100, 8113.
3. Bagchi, B. Chem Rev 2005, 105, 3197.
4. Papoian, G. A.; Ulander, J.; Eastwood, M. P.; Luthey-Schulten, Z.; Wolynes, P. G. Proc Natl Acad Sci USA 2004, 101, 3352.
5. Kovacs, H.; Mark, A. E.; van Gunsteren, W. F. Proteins 1997, 27, 395.
6. Bonvin, A. M. J. J.; Sunnerhagen, M.; Otting, G.; van Gunsteren, W. F. J Mol Biol 1998, 282, 859.
7. Cheng, Y. K.; Rossky, P. J. Nature 1998, 392, 696.
8. Eisenberg, D.; McLachlan, A. D. Nature 1986, 319, 199.
9. Ooi, T.; Oobatake, M.; Nemethy, G.; Scheraga, H. A. Proc Natl Acad Sci USA 1987, 84, 3086.
10. Wesson, L.; Eisenberg, D. Protein Sci 1992, 1, 227.
11. Lee, B.; Richards, F. M. J Mol Biol 1971, 55, 379.
12. Juffer, A. H.; Eisenhaber, F.; Hubbard, S. J.; Walther, D.; Argos, P. Protein Sci 1995, 4, 2499.
13. Connolly, M. L. J Appl Crystallogr 1983, 16, 548.
14. Richmond, T. J. J Mol Biol 1984, 178, 63.
15. Eisenhaber, F.; Argos, P. J Comput Chem 1993, 14, 1272.
16. Perrot, G.; Cheng, B.; Gibson, K. D.; Vila, J.; Palmer, K. A.; Nayeem, A.; Maigret, B.; Scheraga, H. A. J Comput Chem 1992, 13, 1.

17. Fraczkiewicz, R.; Braun, W. J Comput Chem 1998, 19, 319.
18. Gibson, K. D.; Scheraga, H. A. Mol Phys 1988, 64, 641.
19. Petitjean, M. J Comput Chem 1994, 15, 507.
20. Augspurger, J. D.; Scheraga, H. A. J Comput Chem 1996, 17, 1549.
21. Edelsbrunner, H. Discrete Comput Geom 1995, 13, 415.
22. Liang, J.; Edelsbrunner, H.; Fu, P.; Sudhakar, P. V.; Subramaniam, S. Proteins 1998, 33, 1.
23. Hayryan, S.; Hu, C. K.; Skrivanek, J.; Hayryane, E.; Pokorny, I. J Comput Chem 2005, 26, 334.
24. Shrake, A.; Rupley, J. A. J Mol Biol 1973, 79, 351.
25. Bystroff, C. Protein Eng 2003, 15, 959.
26. LeGrand, S. M.; Merz, K. M. J Comput Chem 1993, 14, 349.
27. Connolly, M. L. J Appl Crystallogr 1985, 18, 499.
28. Eisenhaber, F.; Lijnzaad, P.; Argos, P.; Sander, C.; Scharf, M. J Comput Chem 1995, 16, 273.
29. Silla, E.; Villar, F.; Nilsson, O.; Pascual-Ahuir, J. L.; Tapia, O. J Mol Graph 1990, 8, 168, 151.
30. Coxeter, H. S. M. Introduction to Geometry; Wiley: New York, 1961.
31. Wodak, S.; Janin, J. Proc Natl Acad Sci USA 1980, 77, 1736.
32. Hasel, W.; Hendrickson, T. F.; Still, W. C. Tetrahedron Comput Methodol 1988, 1, 103.
33. Weiser, J.; Shenkin, P. S.; Still, W. C. J Comput Chem 1999, 20, 217.
34. Vasilyev, V.; Purisima, E. O. J Comput Chem 2002, 23, 737.
35. Weiser, J.; Shenkin, P. S.; Still, W. C. Biopolymers 1999, 50, 373.
36. Guvench, O.; Brooks, C. L., III. J Comput Chem 2004, 25, 1005.
37. Weiser, J.; Shenkin, P. S.; Still, W. C. J Comput Chem 1999, 20, 688.
38. Grant, J. A.; Pickup, B. T. J Phys Chem A 1995, 99, 3503.
39. Bernstein, F. C.; Koetzle, T. F.; Williams, G. J.; Meyer, E. E., Jr; Brice, M. D.; Rodgers, J. R.; Kennard, O.; Shimanouchi, T.; Tasumi, M. J Mol Biol 1977, 112, 535.
40. Abagyan, R.; Totrov, M. J Mol Biol 1994, 235, 983.
41. Weiser, J.; Weiser, A. A.; Shenkin, P. S.; Still, W. C. J Comput Chem 1998, 19, 797.
42. Gibson, K. D.; Scheraga, H. A. J Phys Chem 1987, 91, 4121.
43. Petukhov, M.; Rychkov, G.; Firsov, L.; Serrano, L. Protein Sci 2004, 13, 2120.
44. Lavor, C. Int J Quantum Chem 2003, 95, 336.
45. Lin, Y.; Stadtherr, M. A. J Chem Phys 2004, 121, 10159.
46. Lin, Y.; Stadtherr, M. A. J Comput Chem 2005, 26, 1413.