

MASKER: improved solvent-excluded molecular surface area estimations using Boolean masks

Christopher Bystroff

Department of Biology, Rensselaer Polytechnic Institute, Troy, NY 12180, USA
E-mail: bystrc@rpi.edu

A fast algorithm for computing the solvent-accessible molecular surface area (SAS) using Boolean masks [Le Grand, S.M. and Merz, K.M.J. (1993). *J. Comput. Chem.*, 14, 349–352] has been modified to estimate the solvent-excluded molecular surface area (SES), including contact, toroidal and re-entrant surface components. Numerical estimates of arc lengths of intersecting atomic SAS are used to estimate the toroidal surface and intersections between those arcs are used to estimate the re-entrant surface area. The new method is compared with an exact analytical method. Boolean molecular surface areas are continuous and pairwise differentiable and should be useful for molecular dynamics simulations, especially as the basis for an implicit solvent model.

Keywords: binary operations/hydrophobic effect/implicit solvent/molecular graphics/molecular simulations/molecular surface/water

Introduction

The solvent-excluded molecular surface (SES) is defined as the boundary of the solvent-excluded molecular volume. The earliest methods for calculating the SES used the ‘rolling ball’ numerical integration method, in which a spherical probe is rolled over the exposed contact surface of each atom (Richards and Richmond, 1977). Numerous algorithms now exist for SES and solvent-accessible surface (SAS) calculations (Shrake and Rupley, 1973; Connolly, 1983, 1993; Richmond, 1984; Rashin *et al.*, 1986; Le Grand and Merz, 1993), including a fast, analytical approach (Sanner *et al.*, 1995, 1996). A review of the history of molecular surface calculations can be found on-line (Connolly, 1996).

SES can be described as a surface with three types of curvature (Figure 1): (1) the solvent-accessible surface area (*contact*), (2) the saddle-shaped surface where a probe makes contact with two atoms (*toroidal*) and (3) the concave bowl-shaped surface where a probe makes contact with exactly three atoms (*re-entrant*). Although points may exist where a spherical probe simultaneously contacts four or more atoms, any such surfaces are exactly equal to the sum of the component three-atom surfaces. The term ‘solvent-accessible surface’ (SAS) is usually applied to the surface traced out by the probe center. This is a scalar multiple of the *contact* surface.

Fast surface area calculations hold the promise of replacing explicit waters or other solvent molecules, in molecular simulations, potentially greatly increasing the efficiency of these calculations (Arnold and Ornstein, 1994; Fraternali and Van Gunsteren, 1996; Lazaridis and Karplus, 1999; Ferrara *et al.*, 2002). Simple pairwise potentials for implicit solvation succeed in reproducing radial distribution functions for simple systems

at fixed concentrations (S. Garde, personal communication), but fail to model the non-additive effect of multibody interactions. Implicit solvent potentials based on SES calculations account for some of these multibody effects, including correctly assigning a solvation free energy of zero to completely buried atoms and reproducing the free energy barrier to hydrophobic association (Rank and Baker, 1997, 1998; Czaplewski *et al.*, 2000).

The incorporation of SES into molecular dynamics simulations has been hampered by the complexity of its computation and especially the computation of its pairwise partial derivatives. Without derivatives, molecular simulations are limited to less efficient sampling methods such as Monte Carlo. The pairwise partial derivatives of the complete SES are continuous; however, the component parts are non-differentiable and sometimes non-continuous. For example, a single *re-entrant* surface segment may disappear with an infinitesimal change in the atom positions. When this happens, another surface segment, bordered by a different set of atoms, appears in its place. Although not impossible, an efficient analytical solution to this problem has not yet been offered. The method described here offers a potential solution to this problem.

A Boolean masks approach for computing SAS was first proposed by LeGrand and Merz (Le Grand and Merz, 1993). In their approach, a set of points were evenly spaced on the surface of a sphere centered at the origin and having radius 1.0 and each was represented by a single bit in a multi-byte word. A set of ‘masks’ was created by placing a probe sphere of unit radius at all positions around the origin and all distances from zero to 2. Points within the probe radius had their bits set to zero, while the remaining bits were set to one. Therefore, one multi-byte word (*mask*) corresponded to one probe sphere location. A binary AND operation over any number of masks resulted in another mask whose non-zero bits represent the SAS for one atom. SAS is estimated quickly and accurately by summing the 1-bits and scaling. Binary masking operations can be done very efficiently, even for long multi-byte words, especially if they are optimized to take advantage of the machine architecture.

The work described here takes the method of LeGrand and Merz one step further by using masks to calculate SES. The problem is broken down to its three component parts. The *contact* surface is a scalar multiple of SAS. The *toroidal* surface can be estimated based on the length of exposed circular edge at the intersection of two atom surfaces. The *re-entrant* surfaces are calculated by placing a mask at each position where a probe is in contact with exactly three atoms. Intersections and self-burial of *toroidal* and *re-entrant* surfaces are accounted for by additional masking operations, as are the pairwise partial derivatives.

Methods

Here we describe the calculations required to estimate the SES given a set of atomic coordinates, \mathbf{v} and radii, r , including the probe radius r_w .

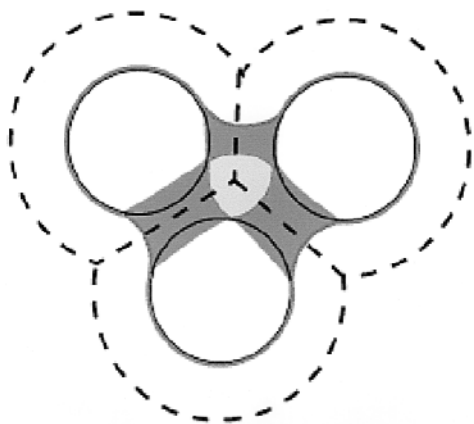


Fig. 1. $SES = contact + toroidal + re-entrant$. The solvent-accessible surfaces, SAS, are the accessible limits of probe centers (dotted lines). The molecular surface (gray outline) is SAS projected inward by one water radius and is composed of three types of curvature: the spherical *contact* surface, saddle-shaped *toroidal* surfaces (dark shading) and three-body, concave, *re-entrant* surfaces (light shading).

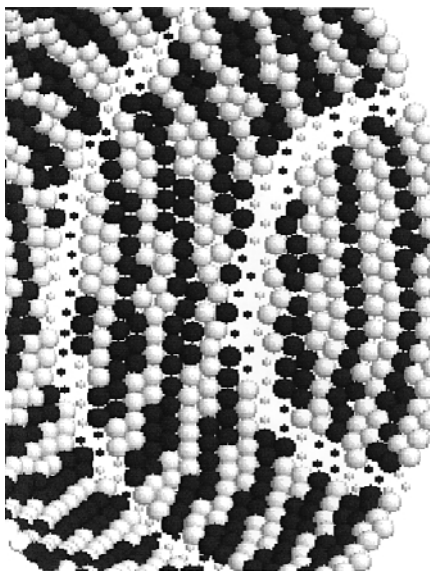


Fig. 2. Close-up view of part of the template mask. Each dot is represented by one bit in a multi-byte word. Shading shows the *edge_masks* for one set of polar coordinates $\psi\phi$. They are continuous and unbroken. Small dots show two intersecting *edge_masks*, having different $\psi\phi$. Pairs of intersecting *edge_masks* always share at least one point.

Generation of the template

In a Boolean mask of the type used here, each bit in a multi-byte word is associated with one point in a template, a sphere of evenly spaced points. The problem of distributing N points evenly on a sphere does not have an exact analytical solution (Coxeter, 1961). Instead, numerical approximations have been used. Here is an approach that works quickly and consistently for any number of points.

4096 points were placed randomly on a unit sphere. A Monte Carlo search was carried out while applying short arc shifts to each point. The function being minimized was the sum of the inverse cube of the pairwise arc distances. The cubic dependence of the energy function on the arc length assured that the nearest neighbor points dominated the gradient.

Figure 2 shows a close-up view of a the mask template. The points are evenly distributed throughout. Coloring and size in Figure 2 show sets of *edge_masks*, which are defined below.

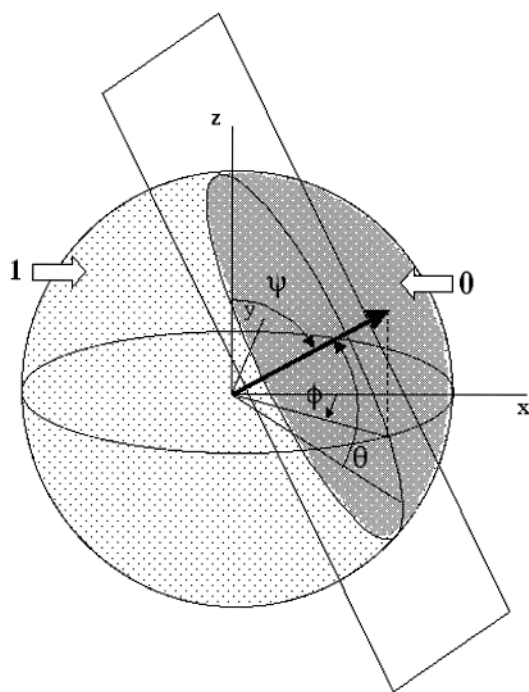


Fig. 3. Polar angles $\psi\phi$ and the spherical angle θ made by the bisecting plane are the indices for each mask, defined as a template sphere with points on one side set to 1 and points on the other 'buried' side set to 0.

Indexing of masks

A *mask* is defined as a set of points on a sphere, bisected by a plane. Points on one side of the bisecting plane are set to zero and the rest are set to 1. Each mask is represented as a 512-byte word (4096 bits); 1-bits are 'exposed' and 0-bits are 'buried' (Figure 3).

Each mask has three indices: two spherical polar coordinates (ψ, ϕ) defining the normal to the bisecting plane and a spherical angle (θ) defined by the distance from the center of the sphere to the bisecting plane (Figure 3). A mask was generated for each value of ψ, ϕ and θ , through their respective ranges. θ ($0 \leq \theta \leq 90^\circ$) and ψ ($0 \leq \psi \leq 180^\circ$) were each incremented by $\epsilon = 4.5^\circ$. ϕ ($0 \leq \phi \leq 360^\circ$) was incremented by $d\phi$ as follows:

$$d\phi = \epsilon / \cos\psi \quad (1)$$

A total of 40 820 masks were generated, occupying a total of 20.9 Mbytes of RAM. Note that the range $0 < \theta < 90^\circ$ represents only half of all possible bisecting planes. For all masks with $\theta > 90^\circ$, we use the inverse of the mask with indices ($180^\circ - \psi, \phi + 180^\circ, 180^\circ - \theta$).

To look up a mask from this library, the indices are calculated as follows, using the interatomic vector, $v_{ij} = v_j - v_i$, and the atom radii:

$$\theta = \cos^{-1}\{[(r_2 + r_w)^2 - (r_1 + r_w)^2 - |v_{ij}|^2] / [-2|v_{ij}|(r_1 + r_w)]\} \quad (2)$$

$$\psi = \cos^{-1}(v_{ij} \cdot \hat{u}_z) \quad (3)$$

$$\phi = \tan^{-1}(v_{ij} \cdot \hat{u}_y / v_{ij} \cdot \hat{u}_x) \quad (4)$$

where \hat{u}_x is a unit vector along x , and so on.

edge_masks

An *edge_mask* is defined as the difference mask between two masks of adjacent θ :

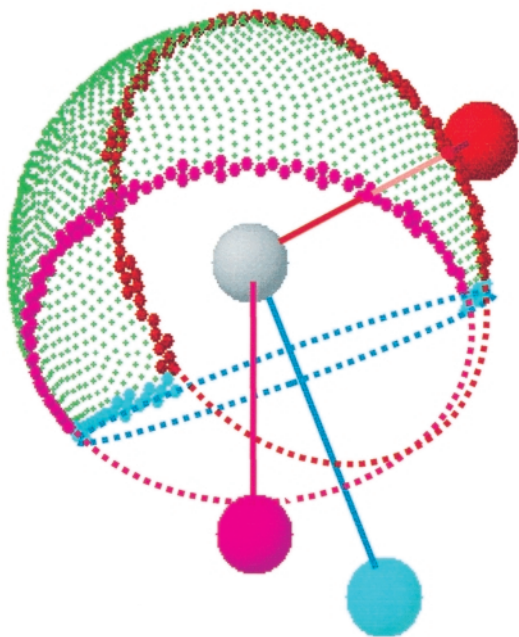


Fig. 4. The green dotted surface represents the contact surface for one atom (gray) with three (colored) neighbors. The large dots represent the exposed circular arcs, whose lengths are proportional to segments of the *toroidal* surface area.

$$\text{edge_mask}(\theta, \psi, \phi) = \text{mask}(\theta, \psi, \phi) \& \text{!mask}(\theta + \varepsilon, \psi, \phi) \quad (5)$$

Here ‘&’ is the binary AND operator and ‘!’ is the binary NOT operator. The arc increment $\varepsilon = 4.5^\circ$ was chosen as the smallest difference for which the intersection of adjacent masks produced a single, unbroken, circular row of closely spaced points (see sizes and shading in Figure 2). This characteristic was necessary for the accurate estimation of arc distances and for the reliable localization of points of intersection for calculating the *re-entrant* surfaces. In other words, if two *edge_masks* intersect, they must share at least one point.

atom_masks

The *atom_mask* is defined as the product of and (binary ‘and’) operations over masks defined by all atoms (j) neighboring the current atom (i). This may be written as a bit-wise vector product:

$$\text{atom_mask}(i) = \prod_j \text{mask}(\theta_{ij}, \psi_{ij}, \phi_{ij}) \quad (6)$$

Here the subscripts denote the atoms used to calculate the indices. Atoms i and j are neighbors if their distance is less than the sum of their radii plus two probe radii ($d_{ij} < r_i + r_j + 2r_w$).

atom_edge_masks

The *atom_edge_mask* for atom i and each neighbor atom j is defined as the intersection of the *atom_mask* and an *edge_mask*:

$$\text{atom_edge_mask}(i,j) = \text{atom_mask}(i) \& \text{edge_mask}(i,j) \quad (7)$$

Figure 4 shows three *atom_edge_masks* for a four-atom case, each edge a different color. The number of 1-bits in an *atom_edge_mask* is proportional to a *toroidal* surface area.

Distance resolution and linear interpolation

The resolution of the numerically estimated surfaces with atom-atom distance is a function of the increment value ε , the

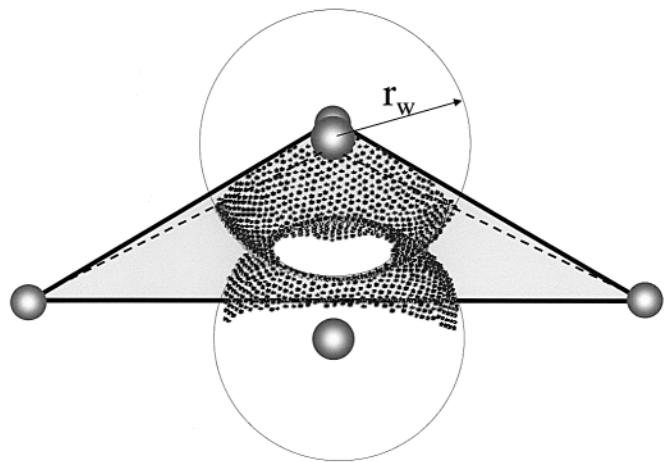


Fig. 5. Self-intersecting *re-entrant* surface. The probe sphere mask is clipped on all four sides of a tetrahedron defined by the three base atoms and the probe.

radius of the atoms and the radius of the probe. For example, two atoms of radius 2.0 Å, probe radius 1.4 Å and $\varepsilon = 4.5^\circ$ gives a pairwise distance resolution of about 0.3 Å at close distances and 0.1 Å at longer distances. The resolution is proportional to

$$\cos(\theta) - \cos(\theta + \varepsilon) \quad (8)$$

However, we may smooth the function by interpolating between masks of two adjacent θ indices, using *edge_masks* to estimate the exposed arc segments. There was no attempt to interpolate in ψ and ϕ .

Estimating contact surface

The estimated contact surface for atom i is the number of 1-bits in *atom_mask*, times a scale factor:

$$\text{contact}(i) = 4\pi r_i^2 (\#\text{atom_mask}(i)/4096) \quad (9)$$

where ‘#’ denotes the number of 1-bits, out of 4096 possible. Smoothing by interpolation in θ is done by summing over all *atom_edge_masks*, using precalculated linear functions.

Estimating toroidal surface

The bits in the *atom_edge_mask* are summed and divided by the (pre-summed) number of 1-bits in *edge_mask*(i,j) to obtain the exposed arc length, α :

$$\alpha = 2\pi \#\text{atom_edge_mask}(i,j)/\#\text{edge_mask}(i,j) \quad (10)$$

If the value of α is non-zero, then the atoms i and j share an exposed *toroidal* surface. The *atom_edge_mask* may contain more than one separate, contiguous arc, separated by buried *toroidal* surface (for example, cyan points in Figure 4). In this case, the individual arc segments are not calculated, nor are they needed.

The *toroidal* surface intersects itself if the outer radius of the torus, $(r_i + r_w)\sin\theta$, is less than the radius of the probe (r_w). The relevant (and differentiable) surface is the part that is not buried by this self-intersection. This is taken into account by setting the integration limits, τ_{\min} and τ_{\max} , properly. The *toroidal* surface in Å² units is then calculated by integrating the surface of a torus of outer radius $(r_i + r_w)\sin\theta$, and inner radius r_w , over τ and scaling by the arc length, αr_w :

$$\text{toroidal}(i) = \alpha r_w \int_{\tau_{\min}}^{\tau_{\max}} [(r_i + r_w)\sin\theta - r_w \cos \tau] d\tau \quad (11)$$

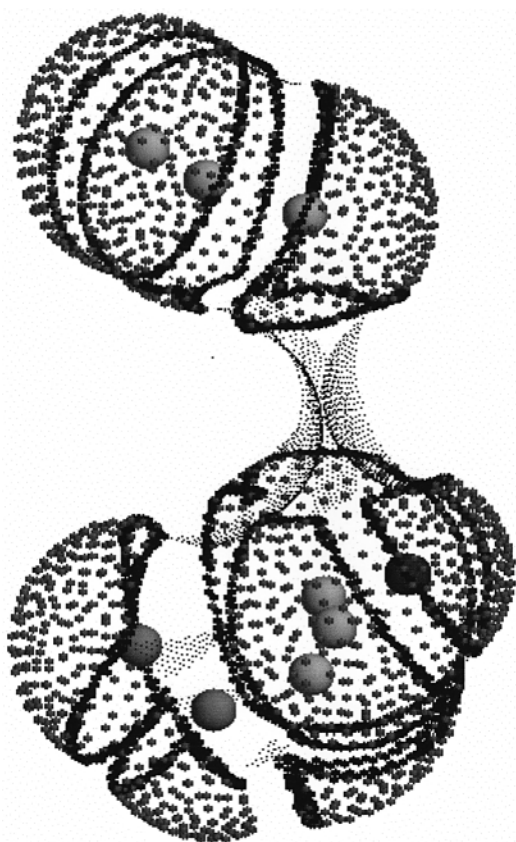


Fig. 6. A RasMol dot surface generated using MASKER. Contact (gray) and re-entrant (dot) surfaces are shown. Large black dots mark the *atom_edge_masks*, used to estimate the toroidal surfaces (not shown).

The upper integration limit $\tau_{\max} = (\pi/2) - \theta$. The lower integration limit $\tau_{\min} = 0$ if $(r_i + r_w)\sin\theta > r_w$, otherwise $\tau_{\min} = \arccos[(r_i + r_w)\sin\theta/r_w]$. Note that *toroidal*(*i*) is the half of the *toroidal* surface that is closest to atom *i*. The part closest to atom *j* is calculated similarly. The two halves of the *toroidal* surface are equal only if the atom radii are equal.

Special case: embedded atoms

If $\theta > \pi/2$, then *toroidal*(*i*) = 0. This happens when the distance r_{ij} is less than $r_j + r_w$. In this case atom *i* may be viewed as being partially ‘embedded’ within the solvent-excluded region of atom *j*. If $r_{ij} < r_i - r_j$, then the atom is completely embedded and it has no exposed surface at all. Generally, these configurations are energetically impossible with normal radii and are of minor relevance. Nonetheless, MASKER handles them properly.

If atom *i* partially embeds atom *j*, then a *toroidal* surface exists for *i*, but the lower integration limit is now $\tau_{\min} = \arccos[(r_i + r_w)\sin\theta/(r_j + r_w)]$. The calculation proceeds as in (11).

Estimating re-entrant surface

The set of points where a water makes contact with exactly three atoms correspond to a set of concave re-entrant surfaces that are, in the simplest cases, spherical triangles of radius r_w (Figure 5). Bordering these spherical triangles are three *toroidal* surfaces, one for each side of the triangle. In the MASKER algorithm, the intersection of any two *atom_edge_masks* implies the existence of a *re-entrant* surface. Using all neighbor atoms, a test is made for pairs of intersecting *atom_edge_masks*. If the condition

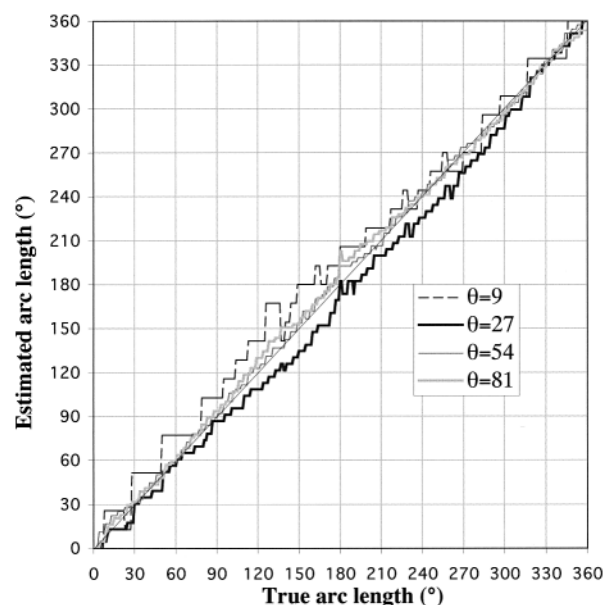


Fig. 7. Arc lengths estimated using *atom_edge_masks* are compared with exact arc lengths. Discreteness in the three angle dimensions and the point density in the mask template are the sources of the errors. *Atom_edge_masks* with small θ have the smallest radius and therefore the largest arc length errors, but also the smallest toroidal areas. Thus, the absolute errors in surface area are evenly distributed in θ .

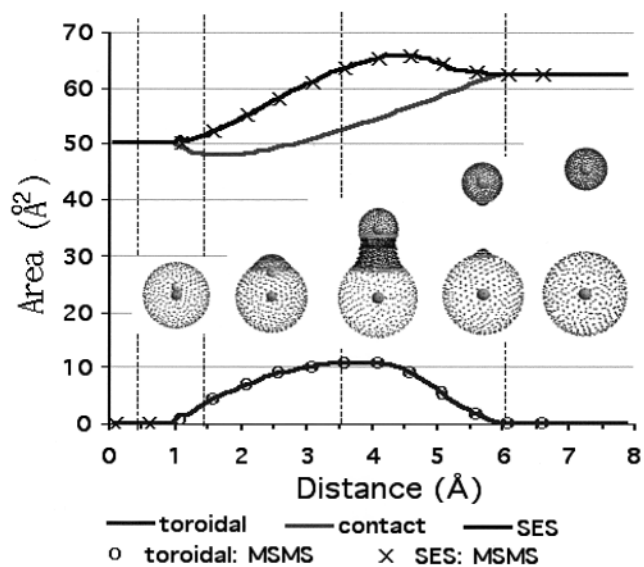


Fig. 8. SES for two atoms, with radii 2.0 and 1.0 Å, were calculated using MASKER and MSMS. Dotted lines indicate distances used for the inserted images. From left to right: completely embedded, partially embedded, peak *toroidal* surface, self-intersecting *toroidal* and completely separated. See discussion in text.

$$\#[\text{atom_edge_mask}(i,j) \ \& \ \text{atom_edge_mask}(i,k)] > 0 \quad (12)$$

is true, then atoms *i*, *j* and *k* share a *re-entrant* surface. An additional masking step determines whether the surface is right-handed, left-handed or both. In Figure 4, the red *atom_edge_mask* intersects the cyan *atom_edge_mask* twice, implying two *re-entrant* surfaces, one on either side of the same tree atoms. Figure 5 shows an example where both the right and left-handed surfaces are present and in this case they are self-intersecting.

If there is a *re-entrant* surface based on atoms *i*, *j* and *k*,

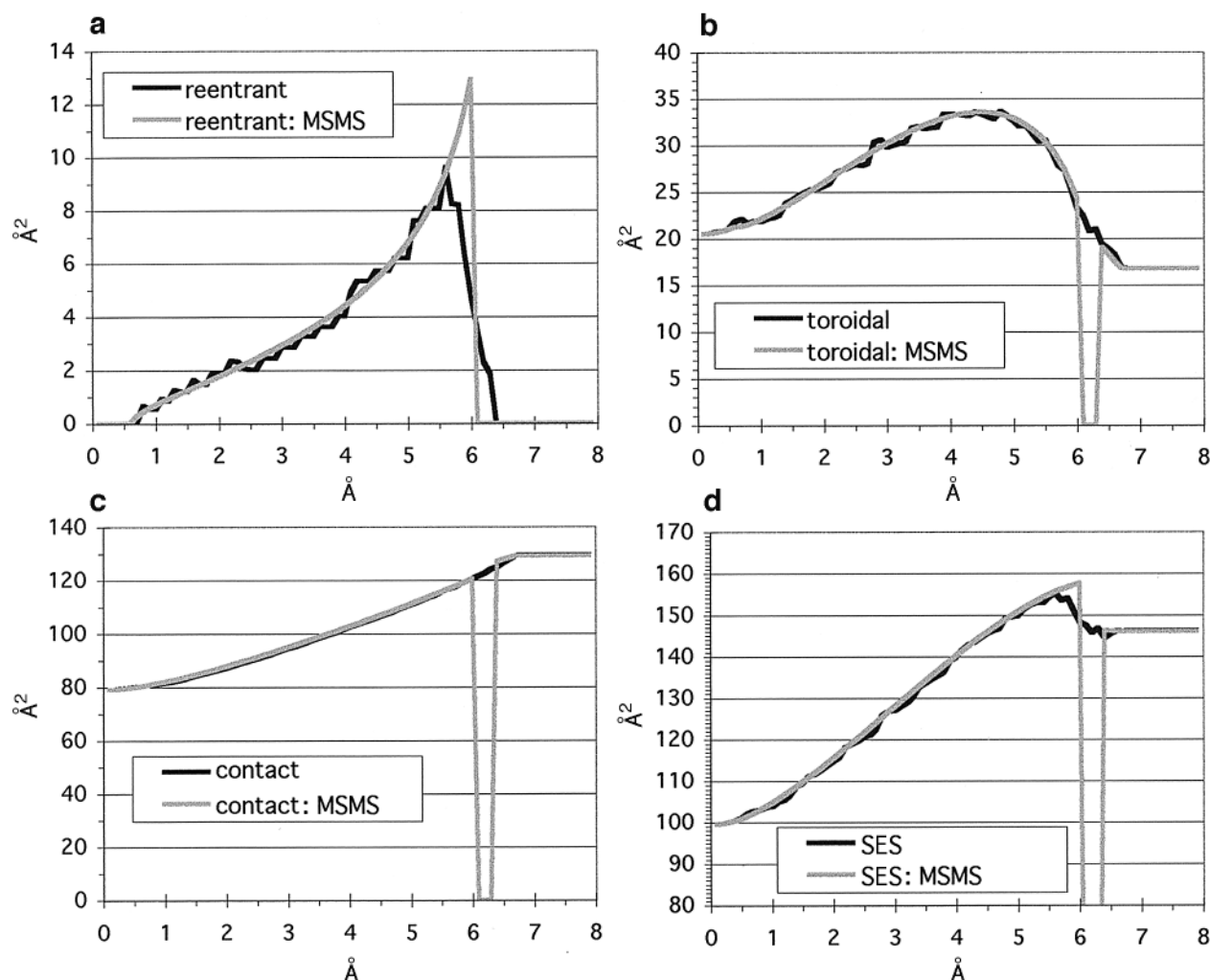


Fig. 9. SES for a three-atom trajectory using MASKER or MSMS. All atoms have radius 2.0 Å. The three atoms make an isosceles triangle with base 4Å. The abscissa is the height of the triangle in Å. (a) *re-entrant*; (b) *toroidal*; (c) *contact*; (d) total SES.

then its area is calculated by placing a probe at v_w , the vertex of a tetrahedron with sides $(d_{ij}, d_{ik}, d_{kj}, r_i + r_w, r_j + r_w, r_k + r_w)$. A mask (*probe_mask*) is placed on that position. The re-entrant surface area is $4\pi r_w^2(\#probe_mask(ijk)/4096)$, where

$$\begin{aligned}
 probe_mask(ijk) = & mask(\theta_{ijkw}, \psi_{ijk}, \phi_{ijk}) \\
 & \& mask(\pi/2, \psi_{ijw}, \phi_{ijw}) \\
 & \& mask(\pi/2, \psi_{jkw}, \phi_{jkw}) \\
 & \& mask(\pi/2, \psi_{kiw}, \phi_{kiw})
 \end{aligned} \quad (13)$$

Here, the mask indices are subscripted by the atoms used to compute them. For example, ψ_{ijw} is the ψ angle for the vector $(v_{ij} \otimes v_{jw})$ and θ_{ijkw} is the spherical angle for the distance between v_w and the *ijk* plane. If this distance is greater than r_w , then $\theta = 0$ and the mask contains only 1s (in which case it is ignored).

Like the *toroidal* surface, the *re-entrant* surface can intersect its mirror image or another *re-entrant* surface. Any points on the *probe_mask* that fall below the base plane of the tetrahedron (the *ijk* plane) intersect another *re-entrant* surface and are therefore not part of the SES (Sanner, 1992). Including these self-intersecting surfaces would make SES non-differentiable. The first mask in Equation 13 performs this function.

Bit-counting

Summing the exposed surface area or arc length requires counting the number of 1-bits in a mask of 4096 points. Bit

counting was made more efficient by the use of a lookup table. Entries in this table, indexed from $-32\,768$ to $+32\,767$, are set to the number of 1-bits required to represent that same number in a two-byte word. A mask consists of 256 two-byte words, hence 256 additions are required to obtain the bit count. Using this shortcut, bit counting accounted for only 0.1% of the CPU time.

Implementation and availability

The MS calculation has been encoded as a Fortran90 module called MASKER, linkable to Fortran programs in addition to C. Currently it runs only in a Unix environment, since it calls some basic Unix system tools, but could be easily modified to work in a Mac or Windows environment. The Fortran90 module MASKER and its library of binary masks may be freely downloaded for non-profit use from <http://isites.bio.rpi.edu/bystrc/pub/masker>. Auxiliary programs, such as those used to create the mask library, are available upon request.

The images and graphs in this paper were made using a simple program that uses the MASKER module. The input files are a library of masks, a table of mask index data, an atom library and the atomic coordinates in PDB format. Optionally, a PDB-format file of template mask points is input and used to draw the dot surfaces. The program returns the SES, including the atom-by-atom breakdown by surface type. The program optionally outputs a dot surface in PDB format

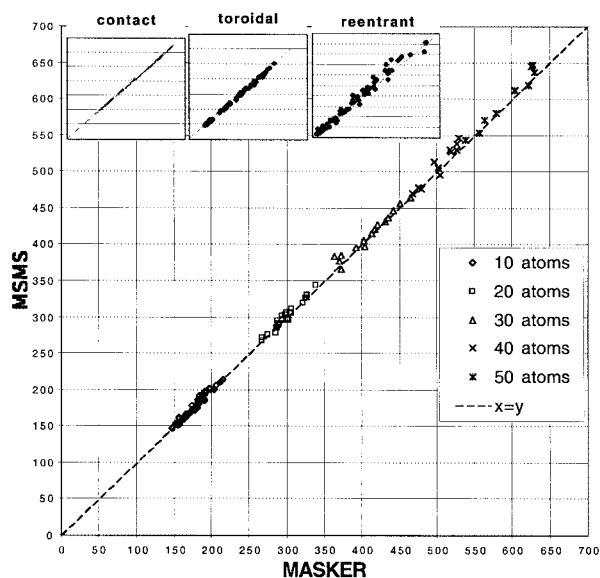


Fig. 10. MASKER SES estimates compared with analytical MSMS calculations for 10, 20, 30, 40 or 50 atoms of radii ranging from 1.5 to 2.2 Å (there are no fully embedded atoms). The probe size is 1.5 Å. The correlation between the two methods is $r = 0.9994$. Small systematic deviations may result from MSMS errors for certain three-body configurations.

(Figure 6) for visualization using RasMol (Sayle and Milner-White, 1995; Bernstein, 2000) or other molecular display programs.

Results and discussion

Sources of error

The accuracy of the *toroidal* surface estimates and of the SES derivatives is proportional to the accuracy in estimating the exposed arc lengths. Figure 7 shows the true arc length plotted against the arc length estimated by masking. The accuracy is worst when θ is small, because the radius of the arc is $r\sin\theta$. However, the absolute errors in surface are roughly constant over θ .

Comparison with MSMS

MASKER estimated SES estimations were compared with analytically calculated values using MSMS (Sanner *et al.*, 1995, 1996). Figure 8 shows the MSMS values plotted against the MASKER estimates for a two-atom trajectory, with the two atoms having radii 2.0 and 1.0 Å. Except for the embedded configuration (distance < 1.0 Å), both methods agree with each other and with exact calculations. MSMS does not properly handle the completely embedded configuration, but this configuration is almost always irrelevant in practice.

Figure 9 shows the results of a three-atom trajectory. The three atoms, each with a radius of 2 Å, form an isosceles triangle with a base of 4 Å. The trajectory axis is the height of the triangle in ångströms. MASKER and MSMS were applied to each configuration and the results are broken down by components.

Several points may be made here. First, the version of MSMS used for this study has a known bug that gives the wrong values at certain distances in the three-atom case (M.Sanner, personal communication). Because of this, there is a difference in the detailed shape of the *re-entrant* surface profile. MSMS incorrectly shows an over-estimate and then a discontinuity at the distance where the *re-entrant* surface goes

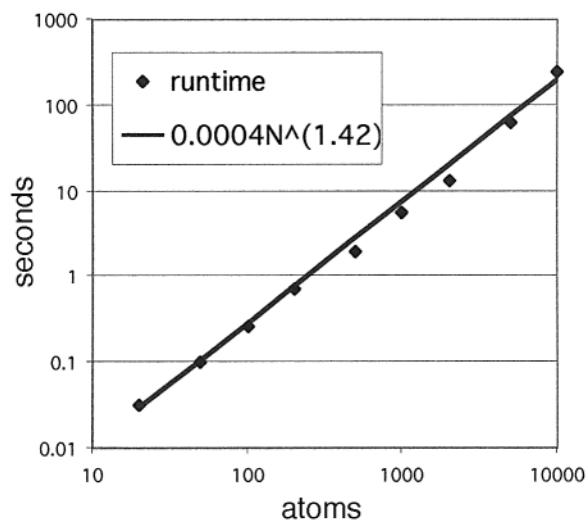


Fig. 11. MASKER run times for various numbers of atoms, plotted as logs. The dependence on the number of atoms is low-order polynomial, $O(N^{1.42})$.

to zero. The correct distance profile and the MASKER profile have the *re-entrant* surface disappearing smoothly with increasing distance, as the self-intersecting part dominates. The true total SES is continuous in the first derivative. Third, the *toroidal* and *re-entrant* surfaces match the MSMS calculations over most of the trajectory but MASKER surfaces have small estimation errors stemming from the errors in the exposed arc distance, discussed above. The *contact* surfaces are more precise, having been interpolated.

Figure 10 shows the comparison of the two methods for larger sets of atoms of mixed radii. Most of the SES calculations agreed within a few square ångströms. A small systematic difference in the re-entrant surface is detected for large atom sets, where MASKER gives lower numbers than MSMS (see insets in Figure 10) and is not due to numerical errors in MASKER.

Runtime dependence on number of atoms

It should be noted that MSMS is much faster than the current version of MASKER. MASKER has greater computational overhead, since it must read a library of masks and mask data before beginning. The memory requirements of MASKER are also greater, again because of the library of masks. However, the new algorithm is robust and because of its more brute-force approach, it is more likely to be immune to algorithmic bugs that are associated with certain surface singularities (Sanner, 1992).

Figure 11 shows the complexity of the computation with the number of atoms. These numbers reflect the runtime after initialization of the masks, using a Pentium III, 750 MHz, running RedHat Linux v6.2. The runtime is polynomial with an exponent of ~ 1.42 . The slow step in the process is currently the masking operations which consumes about 73% of the CPU time. These binary operations may in the future be encoded in special hardware. Additional speed-up is possible by initially computing the 'reduced surface' (Sanner *et al.*, 1996) and by using a neighbor-tree data structure for the atoms (Sanner, 1992; Varshney *et al.*, 1994).

Application to solvent-excluded surface area gradient calculations

Derivatives of molecular surfaces have been explored previously for the solvent-accessible surface (Perrot *et al.*, 1992;

Richmond, 1984; Gogonea and Osawa, 1994). Unfortunately, the SAS derivative with respect to pairwise distance has a discontinuity at the separation distance (one probe diameter, $d_{ij} = r_i + r_j + 2r_w$). Also, the SAS is not the best model for the hydrophobic effect, lacking the barrier to hydrophobic association. No published work can be found for derivatives of the SES.

SES is continuous in the first derivative with respect to pairwise distance. Proof of this may be offered in two parts. (1) The sum of the *toroidal* and *contact* surfaces approaches a constant in the limit of low θ angle, since the tiny exposed disk of the toroid is tangential to the atom sphere. (2) The change in *re-entrant* surface with distance is proportional to the length of the edge between the *re-entrant* and *toroidal* surfaces (this is the side of the truncated spherical triangle in Figure 5). Since this length is continuous with distance, so is the derivative of the *re-entrant* surface.

Pairwise numerical derivatives of the SES can be computed using masks. The *atom_edge_mask* (Equation 7) is proportional to the derivatives of the *contact* surface and the *toroidal* surface. The derivatives of the *re-entrant* surfaces may be calculated using *probe_edge_masks*, defined as follows:

$$\begin{aligned} \text{probe_edge_mask}(i,j,k) = & \text{probe_mask}(ijk) \\ & \& \text{edge_mask}(\pi/2, \psi_{jkw}, \phi_{jkw}) \end{aligned} \quad (14)$$

There are three *probe_edge_masks* for each *re-entrant* surface. Their bit-counts are proportional to the gradient for one atom along a vector normal to the opposite three-atom plane (one of the atoms is the probe, w). The pairwise ij partial derivative can be solved from these three vectors. If the *re-entrant* surface is partially buried by self-intersection (as in Figure 5), then an additional term is added to the derivative to account for the changing size of the 'hole':

$$\#[\text{probe_edge_mask}(ijk) \& \text{edge_mask}(\theta_{wik}, \psi_{jkw}, \phi_{jkw})] \quad (15)$$

is proportional to the change in self-intersecting *re-entrant* surface. Testing of these gradient calculations is work in progress.

References

- Arnold, G.E. and Ornstein, R.L. (1994) *Proteins*, **18**, 19–33.
 Bernstein, H.J. (2000) *Trends Biochem. Sci.*, **25**, 453–455.
 Connolly, M.L. (1983) *Science*, **221**, 709–713.
 Connolly, M.L. (1993) *J. Mol. Graphics*, **11**, 139–141.
 Connolly, M.L. (1996) <http://www.netsci.org/Science/Compchem/feature14.html>.
 Coxeter, H.S.M. (1961) *Introduction to Geometry*. Wiley, New York.
 Czaplowski, C., Rodziewicz-Motowidlo, S., Liwo, A., Ripoll, D.R., Wawak, R.J. and Scheraga, H.A. (2000) *Protein Sci.*, **9**, 1235–1245.
 Ferrara, P., Apostolakis, J. and Caffisch, A. (2002) *Proteins*, **46**, 24–33.
 Fraternali, F. and Van Gunsteren, W.F. (1996) *J. Mol. Biol.*, **256**, 939–948.
 Gogonea, V. and Osawa, E. (1994) *J. Mol. Struct.*, **311**, 305–324.
 Lazaridis, T. and Karplus, M. (1999) *J. Mol. Biol.*, **288**, 477–487.
 Le Grand, S.M. and Merz, K.M.J. (1993) *J. Comput. Chem.*, **14**, 349–352.
 Perrot, G., Cheng, B., Gibson, K.D., Vila, J., Palmer, K.A., Nayeem, A., Maigret, B. and Scheraga, H.A. (1992) *J. Comput. Chem.*, **13**, 1–11.
 Rank, J.A. and Baker, D. (1997) *Protein Sci.*, **6**, 347–354.
 Rank, J.A. and Baker, D. (1998) *Biophys. Chem.*, **71**, 199–204.
 Rashin, A.A., Iofin, M. and Honig, B. (1986) *Biochemistry*, **25**, 3619–3625.
 Richards, F.M. and Richmond, T. (1977) *Ciba Found. Symp.*, 23–45.
 Richmond, T.J. (1984) *J. Mol. Biol.*, **178**, 63–89.
 Sanner, M.F. (1992) *Modelling and Applications of Molecular Surfaces*. University of Haute-Alsace, Mulhouse, France.
 Sanner, M.F., Olson, A.J. and Spehner, J.C. (1995) Presented at the Eleventh Annual Symposium on Computational Geometry, Vancouver, BC, Canada.
 Sanner, M.F., Olson, A.J. and Spehner, J.C. (1996). *Biopolymers*, **38**, 305–320.
 Sayle, R.A. and Milner-White, E.J. (1995) *Trends Biochem. Sci.*, **20**, 374.

Shrake, A. and Rupley, J.A. (1973) *J. Mol. Biol.*, **79**, 351–371.

Varshney, A., Brooks, F.P. and Wright, W.V. (1994) *IEEE Comput. Graphics*, **14**, 19–25.

Received February 8, 2002; revised June 12, 2002; accepted October 4, 2002