Approximate Atomic Surfaces from Linear Combinations of Pairwise Overlaps (LCPO)

JÖRG WEISER, PETER S. SHENKIN, W. CLARK STILL

Department of Chemistry, Columbia University, New York, New York 10027

Received 19 May 1998; accepted 16 September 1998

ABSTRACT: A fast analytical formula was derived for the calculation of approximate atomic and molecular van der Waals (vdWSA), and solvent-accessible surface areas (SASAs), as well as the first and second derivatives of these quantities with respect to atomic coordinates. This method makes use of linear combinations of terms composed from pairwise overlaps of hard spheres; therefore, we term this the LCPO method for linear combination of pairwise overlaps. For higher performance, neighbor-list reduction (NLR) was applied as a preprocessing step. Eighteen compounds of different sizes (8–2366 atoms) and classes (organic, proteins, DNA, and various complexes) were chosen as representative test cases. LCPO/NLR computed the SASA and first derivatives of penicillopepsin, a protein with 2366 atoms, in 0.87 s (0.22 s for the creation of the neighbor list, 0.35 s for NLR, and 0.30 s for SASA and first derivatives) on an SGI R10000/194 Mhz processor. This appears comparable to or better than timings reported previously for other algorithms. The vdWSAs were in good agreement with the numerical results: relative errors for total molecular surface areas ranged from 0.1 to 2.0% and average absolute atomic surface area deviations from 0.3 to 0.7 Å^2. For SASAs without NLR, the LCPO method exhibited relative errors in the range of 0.4–9.2% for total molecular surface areas and average absolute atomic surface area deviations of 2.0–2.7 Å^2; with NLR the relative molecular errors ranged from 0.1 to 7.8% and the average absolute atomic surface area deviation from 1.6 to 3.0 Å^2. © 1999 John Wiley & Sons, Inc. J Comput Chem 20: 217–230, 1999

Keywords: van der Waals surface area (vdWSA); solvent accessible surface area (SASA); analytical surface areas; derivatives; neighbor-list reduction (NLR)

Correspondence to: P. S. Shenkin; e-mail: shenkin@columbia.edu

Contract/grant sponsor: Deutsche Forschungsgemeinschaft

Contract/grant sponsor: NSF; contract/grant number: CHE97-07870

CCC 0192-8651 / 99 / 020217-14

Introduction

The solvent-accessible surface area (SASA) of a molecule is widely used in describing the solvation of solutes and macromolecules. The definition of the SASA was given by Lee and Richards,[1] who presented the image of rolling a sphere, representing a solvent molecule, over the van der Waals surface of a protein. The SASA is described by the locus of points swept out by the center of the solvent sphere. Much of the current interest in the SASA is due to the observation that, at least for nonpolar molecules, the free energy of aqueous solvation is roughly proportional to this quantity,[2] which in turn is roughly proportional to the number of solvent molecules that can contact the solute molecule.

A molecular mechanics program may require surface areas, as well as their first and second derivatives with respect to nuclear coordinates, in order to compute solvation energies in the context of a implicit solvation model. In a molecular simulation utilizing such a program, the exposed surface area of each atom may have to be computed at every energy or gradient evaluation and millions of such evaluations may be carried out during a single simulation. Thus, if a certain degree of inaccuracy may be tolerated in the surface calculation, a fast algorithm can be highly useful even if it is not completely accurate. In the Generalized Born/Surface Area (GB/SA) solvation model[3, 4] the surface contribution accommodates only the cavity and nonpolar attractive solvation terms, whereas the usually much larger electrostatic solvation energies are handled by means of a generalization of the Born equation. Thus, any inaccuracy in the computation of surface areas can be tolerated to a greater extent than would be the case if, for example, the entire solvation energy were computed based on the exposed surface.[5, 6] These considerations have led us to derive faster, approximate, analytical methods to calculate atomic SASAs and their derivatives.[3, 6–21]

In this article we present a fast approximate method for computing the exposed areas of atoms in molecules. We term this the LCPO method, which stands for linear combination of pairwise overlaps, because the method uses linear combinations of terms composed from pairwise overlaps of hard spheres. This approach is very much in the spirit of Wodak and Janin[7] and Hasel et al.[3] The functional form of the LCPO method allows first and second derivatives to be readily computed, and the expressions for these are given in Appendix A.

Eighteen compounds of different sizes (8–2366 atoms) and classes (organic, proteins, DNA, and various complexes) were chosen as representative test cases. Because the intended use for an LCPO is the computation of surfaces for use with the GB/SA solvation model, and because this method utilizes united atoms for the computation of surface areas, all compounds studied were represented in the united-atom approximation. These compounds are listed in Table I.

When one calculates a SASA, rather than a van der Waals surface area (vdWSA), the number of overlapping neighboring atoms grows by a factor of 5–7.[22] This increases the computation time; however, application of neighbor-list reduction (NLR)[22] can make the computation more efficient than it otherwise would be. We present our SASA results with and without an NLR preprocessing step.

Method

When two hard spheres, i and j, only, overlap, then A_i, the accessible surface area of sphere i, is given by

$$A_i = S_i - A_{ij}. \tag{1}$$

S_i is the surface area of the isolated sphere,

$$S_i = 4\pi r_i^2, \tag{2}$$

where r_i is the radius of the sphere. The radii used in this work are shown in Table II and are those used in ref. 22.

A_{ij} is the area of sphere i buried inside sphere j or what we term the overlap of i with j, which can be expressed as a simple function of the internuclear distance d_{ij} and the sphere's radii r_i and r_j (ref. 7):

$$A_{ij} = 2\pi r_i \left(r_i - \frac{d_{ij}}{2} - \frac{r_i^2 - r_j^2}{2d_{ij}} \right). \tag{3}$$

The LCPO method extends eq. (1) to multiple spheres by adding several additional terms, all involving pairwise overlaps of spheres.

TABLE I.
Compounds.

chb: 3-chloro-4-hydroxybenzoic acid ($C_7H_5O_3Cl$); complexed with protocatechuate 3, 4-dioxygenase (Brookhaven entry 3pch); inhibitor observed in two binding sites; first, which we used, is in active site of each protomer (residue 550).
ctc: 7-chlorotetracycline ($C_{22}H_{23}N_2O_8Cl$); complexed with tetracycline repressor from Escherichia coli (Brookhaven entry 2tct)
sip: sipholenol-a monoacetate ($C_{32}H_{54}O_5$); global MM3(92) energy minimum[25]
nmx: nitromethyldethia coenzyme A ($C_{22}H_{37}N_8O_{18}P_3$); complexed with chicken citrate synthase complex (Brookhaven entry 1amz); two conformers of side chain given in pdb file; took conformer A
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: insuline from bovine (Bos taurus)
163d: rev responsive element (RBE, 30 ribonucleotide fragment) complexed with HIV rev protein (residues 34–50)
1lz1: human lysozone
2stw: human ETS1 / DNA complex
2tra: transfer ribonucleic acid (yeast, Asp) with spermine ($C_{10}H_{26}N_4$); took conformer A
1sbg: HIV-1 protease complexed with the inhibitor SB203386
5tra: transfer ribonucleic acid (yeast); we did not take into account the metal atom in pdb file (M7, atom 255)
1inc: porcine pancreatic elastase complex with benzoxazinone inhibitor ($C_{17}H_{22}N_2O_4Cl$)
1kvd: killer toxin from halotolerant yeast
3app: fungus acid proteinase (penicillopepsin)

All acronyms containing four characters name Brookhaven entries.[24]

$$A_i = P_1 S_1 + P_2 \sum_{j \varepsilon N(i)} A_{ij} + P_3 \sum_{\substack{j,\, k \varepsilon N(i) \\ k \varepsilon N(j) \\ k \neq j}} A_{jk} + P_4 \sum_{j \varepsilon N(i)} A_{ij} \left(\sum_{\substack{j,\, k \varepsilon N(i) \\ k \varepsilon N(j) \\ k \neq j}} A_{jk} \right). \quad (4)$$

Here $N(i)$ stands for the neighbor list (NL) of i (the list of spheres that overlap with sphere i). Thus, the second term involves the sum of pairwise overlaps of sphere i with its neighbors as in eq. (1).

The third term is the sum of overlaps of the neighbors of i with each other. The rationale for including such a term is that if j and k are neighbors of i, then, if j and k do not have any mutual overlap, eq. (1) would properly describe the exposed area of i if an A_{jk} term were subtracted, together with the A_{ij} term. To the extent that j and k overlap and this overlap includes part of i, such an equation would underestimate A_i. Adding terms proportional to A_{jk}, the j–k overlap is an attempt to correct for this effect.

TABLE II.
Atomic van der Waals Radii.

Element	Radius (Å)
C	1.70
N	1.65
O	1.60
P	1.90
S	1.90
Cl	1.80

The fourth and last term is a further correction for multiple overlaps. If spheres j and k, neighbors of i, overlap with each other, we would expect the effect of this mutual overlap on A_i to be greater, the greater the overlap of either with i. For example, it is easy to draw pictures of three spheres in which j and k overlap but in which the j–k intersection does not overlap with i. Such situations occur only when the overlaps of j and k with i are both small. Term 4 "counts" A_{jk} more if A_{ij} is high.

P_1–P_4 are parameters obtained by multiple linear regression. For each atom in a set of test molecules, the actual accessible surface area is obtained by numerical means and the sums in eq. (4) are computed and tabulated. The values of P_1–P_4 that cause eq. (4) to best reproduce the numerical surface areas of the atoms in the least-squares sense are then obtained by least-squares regression.[23] We verified that deleting either term

3 or term 4 from this equation significantly worsens the quality of the fit.

Our initial trials of the LCPO method involved computation of P_1–P_4 for the SASA of the atoms in several protein molecules of varying size, parameterizing each molecule separately. We were encouraged by the observation that the values of the parameters did not change much when the molecule used to obtain them was changed. The method appeared to work better for more exposed atoms; there was a strong tendency to overestimate the exposed surface areas of some nearly buried atoms. In particular, nearly buried sp^2 carbons with three neighbors, such as the carboxyl carbons of Asp, Asn, Gln, and Glu residues, consistently exhibited the greatest overestimations of accessible surface area. Efforts to use purely geometric criteria to identify and treat these planar, trigonal systems in a special manner were attempted without great success; however, this observation led us to experiment with parameterization based on atom type and number of bonded neighbors, which proved to be more successful.

In the implementation discussed here, LCPO atom types are defined based on atomic number, hybridization, and number of bonded neighbors. Each such type is given its own set of parameters, which are shown in Appendix B. To obtain the A_{ij} values, atomic radii are needed for all atom types; however, the subsequent determination of P_1–P_4 for any LCPO atom type does not depend on the parameter values for any other atom type. The parameters are determined by means of a linear fit to the actual accessible surface areas for all atoms of a given type alone. Thus, if it becomes necessary to later add a new atom type to the data set, its parameters can be determined independently of the existing parameters for the other atom types. Because four parameters are required for each atom type, each type must occur at least 4 times in the data set used for parametrization.

Some molecular mechanics procedures require the first or even the second derivatives of the energy with respect to the Cartesian coordinates of the atoms. When surface area is involved, the corresponding atomic energetic contribution is generally computed as an energy factor multiplied by the SASA; thus, the determination of the corresponding energetic derivatives requires the first and second derivatives of the SASA with respect to the coordinates of the atoms. Because the only factors in eq. (4) that depend on these coordinates are the A_{ij}, which are pairwise and central, these derivatives can be easily constructed from the derivatives of the i–j interatomic distances with respect to the coordinates of atoms i and j, as shown in Appendix A.

The LCPO method occasionally produces negative accessible atomic areas for atoms that are buried or nearly so. It would be possible to set these (and the corresponding derivatives) to zero. This can only improve the accuracy of the results for surfaces, simply because the absolute average atomic error would be smaller. However, we have not done so in the work described here, although this might be done in molecular mechanics applications.

We showed[22] that when atomic overlaps are large, as when SASA is being computed, $N(i)$, the neighbor list of atom i, can be considerably shortened by removal of atoms whose elimination cannot affect the exposed surface of i. Carrying out this elimination or NLR considerably decreases the computation time. To this end, we experimented with applying NLR to the $N(i)$ before applying the LCPO method to the calculation of SASAs. Because the A_{ij}, in contrast to true surface areas, do change when the neighbor lists are reduced, new LCPO parameters must be fitted for use with NLR. When closely examining how the surface areas of individual atoms varied with small, continuous changes in conformation, however, the NLR results exhibited a jagged appearance not found with the unreduced LCPO method. This was because at each stage of the conformational change different neighbors were eliminated by the NLR algorithm. This problem was circumvented by defining a threshold distance, d^*_{ij}, such that if atoms i and j are closer together than d^*_{ij}, atom j will never be removed from $N(i)$. For the LCPO/NLR results reported here we used the definition

$$d^*_{ij} = 0.62(r_i + r_j). \tag{5}$$

Results and Conclusions

We calculated the vdWSAs (Table III) and SASAs (Tables IV and V) of the compounds listed in Table I, as well as 18 conformers of ethylcyclohexane described below. The data set consisted of 16,448 atoms. We used a solvent-probe radius of 1.4 Å as in ref. 22. All coordinates were published previously[24,25] except the 18 conformations of axial ethylcyclohexane, which were constructed with standard geometries using the MacroModel program[26] with the axial ethyl side chain rotated in

TABLE III.
LCPO vdWSA Calculations.

					Atomic Error[b] (Å^2)	
Compound	No. Atoms	Numerical vdWSA (Å^2)	LCPO vdWSA (Å^2)	Total Diff.[a] (%)	Ave. Abs.	Max. Abs.
chb	11	159.3	156.5	−1.8	0.3	0.8
ctc	33	394.7	391.2	−0.9	0.4	1.4
sip	37	488.7	483.1	−1.1	0.4	1.3
nmx	51	672.6	678.5	0.9	0.5	1.6
1van	121	1505.2	1535.7	2.0	0.7	3.1
1crn	327	4252.1	4223.4	−0.7	0.3	1.2
103d	500	5762.0	5722.4	−0.7	0.4	1.9
2ins	770	9840.8	9883.6	0.4	0.4	3.6
163d	813	9519.1	9461.3	−0.6	0.4	2.0
1lz1	1029	13270.1	13245.1	−0.2	0.3	2.2
2stw	1488	18121.9	18039.4	−0.5	0.4	2.2
2tra	1544	17545.4	17754.5	1.2	0.4	3.8
2sbg	1559	20483.7	20503.1	0.1	0.4	3.6
5tra	1821	20075.0	20315.6	1.2	0.4	4.0
1inc	1846	23909.9	23838.7	−0.3	0.4	1.8
1kvd	1988	25843.6	25740.8	−0.4	0.3	2.5
3app	2366	30404.2	30214.1	−0.6	0.3	1.6

[a] Calculated as follows: 100% × (analytical-numerical) / numerical.
[b] Atomic surfaces range from 0.4 to 35.2 Å^2.

TABLE IV.
LCPO SASA Calculations.

					Atomic Error[b] (Å^2)	
Compound	No. Atoms	Numerical SASA (Å^2)	LCPO SASA (Å^2)	Total Diff.[a] (%)	Ave. Abs.	Max. Abs.
chb	11	315.9	289.8	−8.3	2.7	6.6
ctc	33	608.4	606.3	−0.4	2.6	6.4
sip	37	730.3	714.3	−2.2	2.7	11.0
nmx	51	906.9	846.0	−6.7	2.7	10.0
1van	121	1495.5	1552.4	3.8	2.6	9.8
1crn	327	2976.3	3064.9	3.0	2.3	12.5
103d	500	4426.8	4460.4	0.8	2.0	13.0
2ins	770	5740.4	6267.5	9.2	2.4	15.8
163d	813	5636.6	5297.0	−6.0	2.0	12.5
1lz1	1029	6739.9	6681.9	−0.9	2.2	16.9
2stw	1488	12266.9	12582.4	2.6	2.5	17.8
2tra	1544	12507.3	12586.4	0.6	2.0	14.3
1sbg	1559	9540.2	9593.6	0.6	2.2	15.8
5tra	1821	14830.5	15109.4	1.9	2.2	19.6
1inc	1846	10666.1	10471.2	−1.8	2.0	15.3
1kvd	1988	10934.0	10804.1	−1.2	2.1	14.9
3app	2366	12722.8	12209.6	−4.0	2.1	15.5

[a] Calculated as follows: 100% × (analytical-numerical) / numerical.
[b] Atomic surfaces range from 0.0 to 68.9 Å^2.

TABLE V.
LCPO / NLR SASA Calculations.

Compound	No. Atoms	Numerical SASA (Å^2)	LCPO / NLR SASA (Å^2)	Total Diff.[a] (%)	Atomic Error[b] (Å^2)	
					Ave. Abs.	Max. Abs.
chb	11	315.9	291.3	−7.8	2.3	4.7
ctc	33	608.4	690.2	0.1	2.7	12.9
sip	37	730.3	733.8	0.5	3.0	10.9
nmx	51	906.9	890.6	−1.8	2.4	6.8
1van	121	1495.5	1567.6	4.8	2.6	14.3
1crn	327	2976.3	2915.6	−2.0	2.0	11.8
103d	500	4426.8	4182.1	−5.5	1.6	8.8
2ins	770	5740.4	6031.0	5.1	2.2	22.4
163d	813	5636.6	5349.7	−5.1	1.9	11.1
1lz1	1029	6739.9	6578.2	−2.4	2.0	11.9
2stw	1488	12266.9	12701.8	3.5	2.3	17.2
2tra	1544	12507.3	12472.9	−0.3	2.0	19.4
1sbg	1559	9540.2	10045.9	5.3	2.1	16.5
5tra	1821	14830.5	14765.1	−0.4	2.0	25.0
1inc	1846	10666.1	10720.4	0.5	1.9	13.5
1kvd	1988	10934.0	10920.5	−0.1	2.0	19.7
3app	2366	12722.8	12582.6	−1.1	1.9	14.3

[a] Calculated as follows: 100% × (analytical-numerical) / numerical.
[b] Atomic surfaces range from 0.0 to 68.9 Å^2.

20° increments. LCPO parameters for the atom types that appeared in the data set are given in Appendix B. Numerical surface areas required for the parameterization were computed using MacroModel.

In the following paragraphs we compare the numerical and analytical results (Tables III−V) and discuss the CPU times for the LCPO computations (Tables VI, VII). In the tables of CPU times the results shown for first derivatives include the times

TABLE VI.
CPU Times for LCPO vdWSA Calculations.

Compound	No. Atoms	CPU LCPO (s)		
		Surface	First Deriv.[a]	Second Deriv.[b]
chb	11	0.0001	0.0002	0.0007
ctc	33	0.0005	0.0009	0.0038
sip	37	0.0006	0.0010	0.0043
nmx	51	0.0005	0.0009	0.0040
1van	121	0.002	0.002	0.011
1crn	327	0.004	0.006	0.029
103d	500	0.008	0.011	0.052
2ins	770	0.011	0.015	0.070
163d	813	0.013	0.018	0.085
1lz1	1029	0.014	0.019	0.092
2stw	1488	0.021	0.029	0.136
2tra	1544	0.025	0.035	0.162
1sbg	1559	0.021	0.029	0.131
5tra	1821	0.031	0.045	0.199
1inc	1846	0.025	0.036	0.159
1kvd	1988	0.027	0.038	0.172
3app	2366	0.033	0.048	0.208

[a] Calculation of first derivatives includes calculation of the surface.
[b] Calculation of second derivatives includes calculation of the surface and first derivatives.

TABLE VII.
CPU Times for LCPO and LCPO_NLR-SASA Calculations.

		CPU LCPO (s)			CPU NLR	CPU LCPO_NLR (s)			LCPO Surface Vs.	LCPO First Deriv. Vs.	LCPO Second Deriv. Vs.
Compound	No. Atoms	Surface	First Deriv.[a]	Second Deriv.[b]	Overhead (s)	Surface	First Deriv.[a, c]	Second Deriv.[b, c]	LCPO_NLR Surface[d]	LCPO_NLR First Deriv.[d]	LCPO_NLR Second Deriv.[d]
chb	11	0.0002	0.0004	0.0017	0.0001	0.0001	0.0003	0.0010	1.00	1.00	1.55
ctc	33	0.003	0.006	0.027	0.001	0.001	0.002	0.009	1.50	2.00	2.70
sip	37	0.003	0.005	0.025	0.001	0.001	0.002	0.009	1.50	1.67	2.50
nmx	51	0.005	0.008	0.041	0.002	0.002	0.003	0.013	1.25	1.60	2.73
1van	121	0.028	0.044	0.237	0.010	0.006	0.011	0.055	1.75	2.10	3.65
1crn	327	0.094	0.142	0.762	0.034	0.019	0.030	0.161	1.77	2.22	3.91
103d	500	0.173	0.261	1.441	0.063	0.038	0.060	0.319	1.71	2.12	3.77
2ins	770	0.237	0.348	1.863	0.096	0.056	0.083	0.427	1.56	1.94	3.56
163d	813	0.303	0.446	2.397	0.112	0.068	0.103	0.526	1.68	2.07	3.76
1lz1	1029	0.351	0.512	2.722	0.139	0.079	0.117	0.595	1.61	2.00	3.71
2stw	1488	0.416	0.619	3.223	0.177	0.106	0.161	0.789	1.47	1.83	3.34
2tra	1544	0.508	0.763	4.096	0.200	0.122	0.189	0.952	1.58	1.96	3.56
1sbg	1559	0.506	0.735	3.838	0.212	0.123	0.180	0.901	1.51	1.87	3.45
5tra	1821	0.595	0.900	4.827	0.230	0.144	0.225	1.122	1.59	1.98	3.57
1inc	1846	0.662	0.960	5.033	0.273	0.157	0.230	1.150	1.54	1.91	3.54
1kvd	1988	0.728	1.057	5.564	0.296	0.168	0.249	1.239	1.57	1.94	3.62
3app	2366	0.857	1.242	6.542	0.354	0.203	0.300	1.505	1.54	1.90	3.52

[a] Calculation of first derivatives includes calculation of the surface.

[b] Calculation of second derivatives includes calculation of the surface and first derivatives.

[c] Not including NLR overhead.

[d] Calculated as follows: LCPO / (NLR + LCPO_NLR).

for the computation of the surface and times shown for second derivatives include those for surfaces and first derivatives. The times shown for SASA calculations with NLR list the NLR overhead separately; however, the overall times plotted in Figure 1 include the NLR overhead. None of the times include the construction of the pairwise interatomic distance matrix that is used to construct the neighbor list, because this is common to all the calculations. Times for construction of the distance matrix can be found in ref. 22.

All calculations were performed on an SGI R10000/194 MHz processor (Power Onyx), using Fortran code optimized at the -n32 -mips3 -O3 level.

VAN DER WAALS SURFACES

Table III shows the results of vdWSA calculations. NLR was not used, because NLR does not significantly reduce the NL in this situation.[22] The molecular numerical and analytical surfaces were in good agreement with each other and the agreement appeared to be better than that obtained by Grant and Pickup; they used a Gaussian method to calculate vdWSA[27] (LCPO vs. Gaussian: 1 crn, 0.7 vs. 1.5%; 2ins, 0.4 vs. 0.9%; 1lz1, 0.2 vs 0.7%; 3app, 0.6 vs. 1.1%). The individual atomic vdWSAs also appear to be computed accurately: the average absolute error is approximately 0.4 Å^2 and the maximum absolute error is about 4 Å^2. These results are also better than those obtained by Hasel et al.[3]: for crambin (1 crn) they reported a 12.2% error in total surface and an average absolute atomic surface error of 1.5 Å^2. Figure 2a shows a scatter plot of LCPO versus numerical vdWSA values for all the atoms in the data set.

The CPU times for vdWSA surface calculations and first and second derivatives increased linearly with the number of atoms in the molecule (Table VI, Fig. 1a). The inclusion of first derivatives in a vdWSA calculation increases the CPU time by about 45%, whereas inclusion of first and second derivatives slows the calculation by a factor of about 6.5 over the time for the surface only. The peaks in Figure 1a at about 1600 and 1800 atoms are due to the nucleic acids in the data set. The large number of phosphorus atoms in these compounds increases the CPU time, because these atoms have larger radii and therefore more neighbors than the typical atom types in the other compounds.

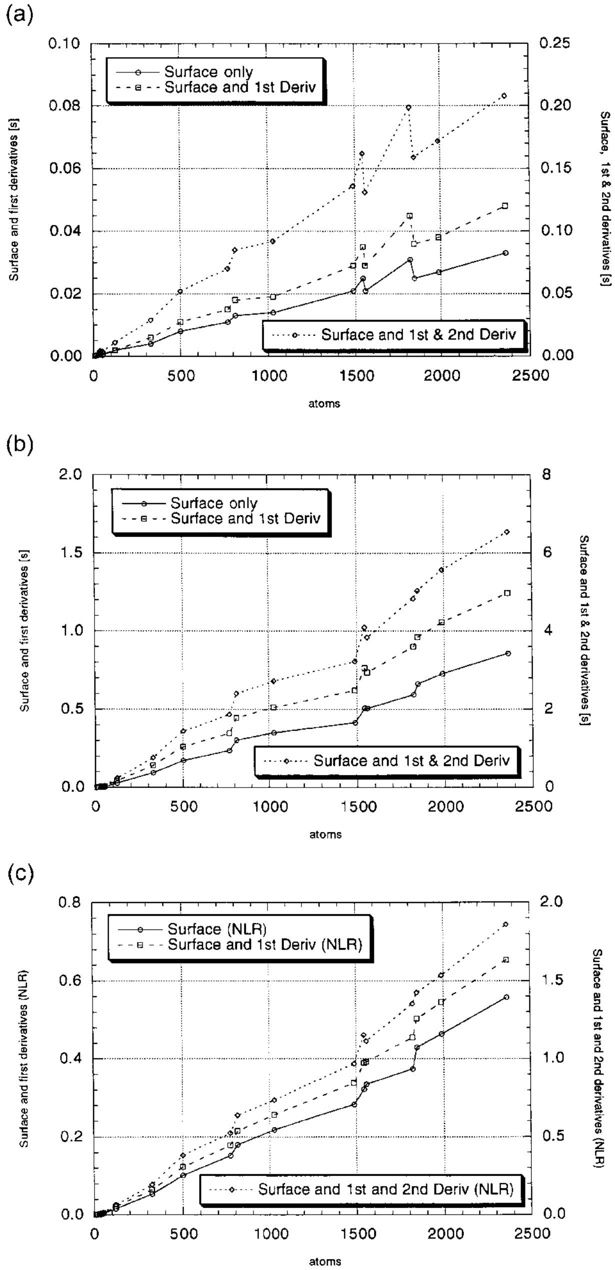

FIGURE 1. (a) CPU times, vdW surfaces. CPU times for vdWSA surface calculations, first and second derivatives (from Table VI). (b) CPU times, SASA surfaces. CPU times for SASA surface calculations, first and second derivatives (from Table VII). (c) CPU times, SASA / NLR surfaces. CPU times for SASA / NLR surface calculations, first and second derivatives (from Table VII).

SOLVENT-ACCESSIBLE SURFACES

Recall that the SASA can be thought of as a surface of an assembly of hard-sphere atoms with augmented van der Waals radii. Each sphere has many more overlapping neighbors in a SASA computation than in a vdWSA computation, and the individual overlaps are greater as well. As can be seen by comparing the total surface areas in Tables III and IV, molecular SASAs for large molecules are smaller than the corresponding vdWSAs.* Thus, typical atomic SASAs for large molecules are smaller than the corresponding vdWSAs. But in the LCPO method they are obtained from sums containing more terms that are, moreover, larger than the terms in the vdWSA sums. Also, atomic SASAs exhibit a wider range (0.0–68.9 Å^2) than atomic vdWSAs (0.4–35.2 Å^2). The addition and subtraction of many large numbers to give smaller average results with a wider range of values makes the LCPO method less accurate for SASA than for vdWSA, as can be seen in Tables III and IV and Figure 2.

Without NLR the LCPO method exhibits relative errors in the range of 0.4–9.2% for total molecular surface areas; with NLR the range is 0.1–7.8% (Table V). Hasel et al.[3] report for crambin (1 crn) a 7.0% deviation in SASA from the numerical results

* All atoms contribute to the vdWSA of a molecule, whereas more than half of the atoms in compact molecules are inaccessible to a water probe. Going from a vdWSA to a SASA, the reduction in the surface areas of the buried atoms more than compensates for the increase in the surface areas of the exposed atoms.

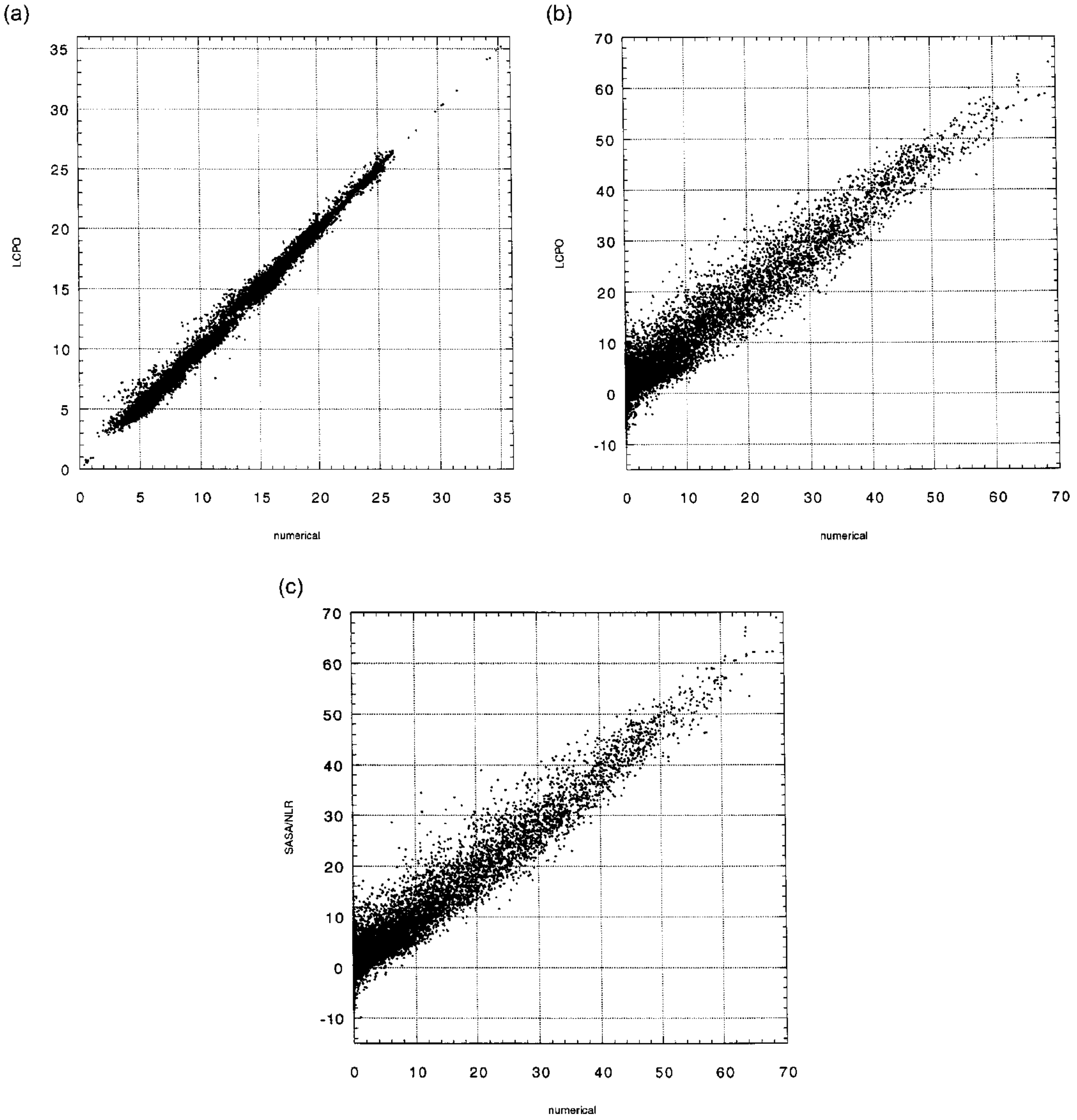

FIGURE 2. (a) vdW atomic surfaces. Scatter plot of LCPO vs. numerical vdW atomic surfaces for all atoms in the data set. (b) SAS atomic surfaces. Scatter plot of LCPO vs. numerical SASA atomic surfaces for all atoms in the data set. (c) SAS / NLR atomic surfaces. Scatter plot of LCPO / NLR vs. numerical SASA atomic surfaces for all atoms in the data set.

and an average absolute atomic surface area error of 3.8 Å^2. LCPO gives 3.0% and 2.3 Å^2 without NLR and 2.0% and 2.0 Å^2 with NLR.

Comparison of the LCPO results with and without NLR (Tables IV, V, Fig. 2b, c, respectively) indicate more accurate molecular surfaces and average absolute atomic errors but less accurate maximum atomic errors when NLR is used. Figure 2c (scatter plot with NLR) appears denser about the line $x = y$ than Figure 2b (scatter plot without NLR), but the worst outliers lie further from the line in Figure 2b.

These results are consistent with the explanation given for the fact that LCPO works better for vdWSA than for SASA. When NLR is incorporated, the sums in eq. (4) include fewer terms (the NL is reduced by a factor of about 3, ref. 22) and thus these sums are smaller, leading to better overall results.

We carried out two more SASA/NLR calculations on larger compounds than those used in our test set (Table I): microbial ribonuclease (2547 atoms, 1bni) and bovine chymotrypsin complexed to the inhibitor domain of Alzheimer's amyloid (4346 atoms, 1ca0). The total SASA of both compounds was in good agreement with the numerical results (1bni, −0.2%; 1ca0, +1.7%), and the average absolute atomic SASA error (1bni, 2.0 Å^2; 1ca0, 1.9 Å^2) and the max. absolute atomic SASA error (1bni, 19.2 Å^2; 1ca0, 20.9 Å^2) were in the same range as compounds of the test set. Thus, the parameters derived from the test set appear to generalize well to other compounds.

As for vdWSA, the CPU times for computing SASA surfaces and first and second derivatives increased linearly with the number of atoms in the molecule (Fig 1b, c). For SASA the computation of first derivatives raises the CPU time by about 50%, and computation of first and second derivatives raises it by a factor of about 8 over the computation of the surface alone.

Comparison with published CPU times is difficult: different hardware, different compilers and compilation options and different software implementations of the same algorithm can all lead to different CPU times. The vdW radii used also affect the CPU times through the effect of radius on the size of the NL. In addition, published work rarely states whether the CPU times presented include the system time attributable to the process (ours do not). Nevertheless, we attempt at least a crude comparison. A recently published analytical method (GETAREA)[20] computes atomic surfaces and first derivatives of a 2325-atom protein in 2.10 s; the same article presents results for the approximate methods MSEED[10] (1.30 s) and SASAD[16] [1.13 s as SASAD (4, 12) and 1.47 s as SASAD (4, 24), a higher level of accuracy], all on a platform very similar to ours. Most vdW radii from ref. 20 are smaller than ours, which would be expected to give smaller NLs and faster computations. For penicillopepsin (3app) with 2366 atoms we obtained timings of 1.24 s without NLR and 0.65 with NLR, including the NLR overhead. The construction of the required interatomic distance matrix takes an additional 0.22 s.[22]

Although the NLR procedure itself is more time consuming than the subsequent surface computation, the sum of both when NLR is used is still less than the CPU time for the surface calculation without NLR (Table VII). The partial NLR procedure used here, which was implemented with eq. (5), is less time consuming than the full procedure (0.35 s for 3app instead of 0.43 s) because only remote atoms are considered for removal. Surprisingly, however, the NLR reduction achieved is still quite high: the combined NLs of 3app include 50,529 overlapping pairs without NLR; with full NLR this is reduced to 18,457 pairs[22]; with the use of eq. (5), as in the results reported here, 20,253 pairs are included. Thus, the use of eq. (5) provides 94% of the reduction provided by the full NLR algorithm. Use of NLR speeds the SASA computation (including the NLR overhead) by factors of about 1.5 when the only surface is computed, 1.9 when first derivatives are included and 3.5 when second derivatives are also included.

Figure 3 shows the variation in exposed surface area for the methyl carbon of axial ethylcyclohex-

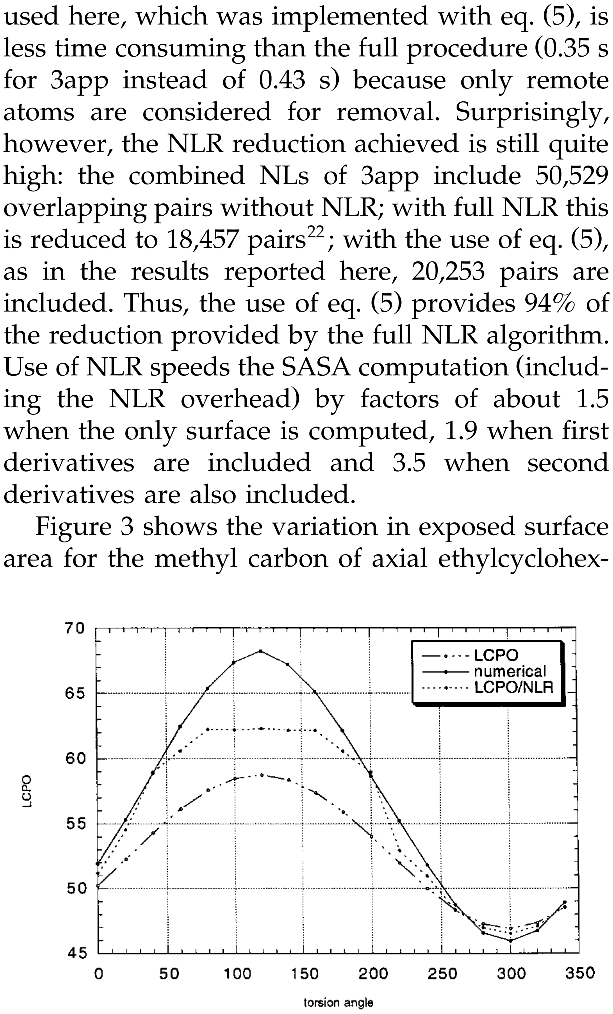

FIGURE 3. Atomic SASA during a conformational change. Variation of exposed SASA (numerical, LCPO, and LCPO / NLR) for the methyl carbon of axial ethylcyclohexane when the ring to side chain torsion is varied through 360° in 20° steps.

ane when the ring to side chain torsion is varied through 360°. LCPO underestimates the amplitude of the variation, but the use of NLR improves the fit.

Summary

LCPO appears to be the fastest of the approximate surface area methods available, particularly when NLR[22] is incorporated. It is simple in principle and affords a good approximation to the hard-sphere results. Self-consistent first and second derivative expressions were derived and are given in Appendix A. To the best of our knowledge, second derivatives for analytical SASA calculations were reported only for the method of Hasel et al.[3] The vdWSA are in excellent agreement with the numerical results; the calculated SASA results, while not as accurate as the vdWSA results, are still good.

The NLR method,[22] which was previously shown to be useful in the context of a numerical surface area algorithm, is shown here to be useful in the context of a much faster analytical computation. The NLR overhead is the same in either case; the fact that NLR reduces the overall time, even when the method to which it is applied is itself fast, demonstrates the utility of the method. Moreover, the evidence presented seems to indicate that the incorporation of NLR gives more realistic results, particularly when comparing conformations of a single compound.

The LCPO method appears to be more accurate than the method described by Hasel et al.,[3] which has been utilized for some years in the MacroModel program.[26] By default, the current version of MacroModel (Version 6.5) uses the LCPO method in combination with NLR for the computation of surface areas.

Appendix A

The following symbols are used:

j, k	neighbors of atom i
l, m	any atoms in the molecule
A_i	accessible surface area of atom i
A_{ij}	surface area of atom i included within atom j
c_i, c'_i	Cartesian coordinate ($c = x,\ y$, or z) of atom i
r_i	radius of atom i
d_{ij}	distance between atoms i and j
P_1, P_2, P_3, P_4	LCPO parameters

FIRST DERIVATIVES

The first derivative of A_{ij} with respect to a Cartesian coordinate of an atom is given by

$$\frac{\partial A_i}{\partial c_l} = P_2 \sum_j \frac{dA_{ij}}{dd_{ij}} \cdot \frac{\partial d_{ij}}{\partial c_l} + P_3 \sum_{k \neq j} \frac{dA_{jk}}{dd_{jk}} \cdot \frac{\partial d_{jk}}{\partial c_l} + P_4 \sum_j \left(\frac{dA_{ij}}{dd_{ij}} \cdot \frac{\partial d_{ij}}{\partial c_l} \cdot \sum_{k \neq j} A_{jk} + A_{ij} \cdot \sum_{k \neq j} \frac{dA_{jk}}{dd_{jk}} \cdot \frac{\partial d_{jk}}{\partial c_l} \right). \tag{A.1}$$

A_{ij} is a function of interatomic distance; its derivative is given by

$$\frac{dA_{ij}}{dd_{ij}} = \pi r_i \left(\frac{r_i^2 - r_j^2}{d_{ij}^2} - 1 \right). \tag{A.2}$$

The interatomic distance d_{ij} is given by

$$d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}. \tag{A.3}$$

Its partial derivatives with respect to Cartesian coordinates are given by

$$\frac{\partial d_{ij}}{\partial c_i} = -\frac{c_j - c_i}{d_{ij}}, \tag{A.4}$$

$$\frac{\partial d_{ij}}{\partial c_j} = \frac{c_j - c_i}{d_{ij}}. \tag{A.5}$$

SECOND DERIVATIVES

The second derivatives of A_{ij} with respect to Cartesian coordinates are given by

$$\frac{\partial^2 A_i}{\partial c_l \cdot \partial c'_m} = P_2 \sum_j \frac{\partial^2 A_{ij}}{\partial c_l \cdot \partial c'_m} + P_3 \sum_{k \neq j} \frac{\partial^2 A_{jk}}{\partial c_l \cdot \partial c'_m} + P_4 \sum_j \frac{\partial^2 T_{ijk}}{\partial c_l \cdot \partial c'_m}. \tag{A.6}$$

Here

$$T_{ijk} = A_{ij} \cdot \sum_{k \neq j} A_{jk}.$$

For a single A_{ij} term we have

$$\frac{\partial^2 A_{ij}}{\partial c_l \cdot \partial c'_m} = \frac{dA_{ij}}{dd_{ij}} \cdot \frac{\partial^2 d_{ij}}{\partial c_l \cdot \partial c'_m} + \frac{\partial d_{ij}}{\partial c_l} \cdot \frac{\partial d_{ij}}{\partial c'_m} \cdot \frac{d^2 A_{ij}}{dd_{ij}^2}, \tag{A.7}$$

$$\frac{d^2 A_{ij}}{dd_{ij}^2} = -2\pi r_i \left(\frac{r_i^2 - r_j^2}{d_{ij}^3} \right). \tag{A.8}$$

It is convenient to distinguish four cases [eqs. (A.9)–(A.12)] in order to calculate the second derivatives of the interatomic distance with respect to Cartesian coordinates; in eqs. (A.9)–(A.12) c and c' are assumed to be different coordinates (e.g., they cannot both be x).

$$\frac{\partial^2 d_{ij}}{\partial c_i \cdot \partial c_i} = \frac{\partial^2 d_{ij}}{\partial c_j \cdot \partial c_j} = -\frac{(c_j - c_i)^2}{d_{ij}^3} + \frac{1}{d_{ij}}, \tag{A.9}$$

$$\frac{\partial^2 d_{ij}}{\partial c_i \cdot \partial c_j} = \frac{\partial^2 d_{ij}}{\partial c_j \cdot \partial c_i} = \frac{(c_j - c_i)^2}{d_{ij}^3} - \frac{1}{d_{ij}}, \tag{A.10}$$

$$\frac{\partial^2 d_{ij}}{\partial c_i \cdot \partial c'_i} = \frac{\partial^2 d_{ij}}{\partial c_j \cdot \partial c'_j} = -\frac{(c_j - c_i) \cdot (c'_j - c'_i)}{d_{ij}^3}, \tag{A.11}$$

$$\frac{\partial^2 d_{ij}}{\partial c_i \cdot \partial c'_j} = \frac{\partial^2 d_{ij}}{\partial c_j \cdot \partial c'_i} = \frac{(c_j - c_i) \cdot (c'_j - c'_i)}{d_{ij}^3}. \tag{A.12}$$

For $(\partial^2 T_{ijk})/(\partial c_l \cdot \partial c'_m)$ six cases may be distinguished; in eqs. (A.13)–(A.18) c and c' may be the same coordinate (e.g., they might both be x).

$$\frac{\partial^2 T_{ijk}}{\partial c_i \cdot \partial c'_i} = \frac{\partial^2 A_{ij}}{\partial c_i \cdot \partial c'_i} \cdot \sum_{k \neq j} A_{jk}, \tag{A.13}$$

$$\frac{\partial^2 T_{ijk}}{\partial c_k \cdot \partial c'_k} = A_{ij} \cdot \sum_{k \neq j} \frac{\partial^2 A_{jk}}{\partial c_k \cdot \partial c'_k}, \tag{A.14}$$

$$\frac{\partial^2 T_{ijk}}{\partial c_i \cdot \partial c'_k} = \frac{\partial A_{ij}}{\partial c_i} \cdot \sum_{k \neq j} \frac{\partial A_{jk}}{\partial c'_k}, \tag{A.15}$$

$$\frac{\partial^2 T_{ijk}}{\partial c_i \cdot \partial c'_j} = \frac{\partial A_{ij}}{\partial c_i} \cdot \sum_{k \neq j} \frac{\partial A_{jk}}{\partial c'_j} + \frac{\partial^2 A_{ij}}{\partial c_i \cdot \partial c'_j} \cdot \sum_{k \neq j} A_{jk}, \tag{A.16}$$

$$\frac{\partial^2 T_{ijk}}{\partial c_j \cdot \partial c'_k} = \frac{\partial A_{ij}}{\partial c_j} \cdot \sum_{k \neq j} \frac{\partial A_{jk}}{\partial c'_k} + A_{ij} \cdot \sum_{k \neq j} \frac{\partial^2 A_{jk}}{\partial c_j \cdot \partial c'_k}, \tag{A.17}$$

$$\frac{\partial^2 T_{ijk}}{\partial c_j \cdot \partial c'_j} = A_{ij} \cdot \sum_{k \neq j} \frac{\partial^2 A_{jk}}{\partial c_j \cdot \partial c'_j} + \frac{\partial A_{ij}}{\partial c_j} \cdot \sum_{k \neq j} \frac{\partial A_{jk}}{\partial c'_j} + \frac{\partial A_{ij}}{\partial c'_j} \cdot \sum_{k \neq j} \frac{\partial A_{jk}}{\partial c_j} + \frac{\partial^2 A_{ij}}{\partial c_j \cdot \partial c'_j} \cdot \sum_{k \neq j} A_{jk}. \tag{A.18}$$

Appendix B: Parametrization of Atom Types

Atom Type	No. Bonded Neighbors	No. in All Test Compounds	Surface Type	P_1	P_2	P_3	P_4
C *sp*3	1	876	vdW	9.2275e-01	−6.3754e-01	−3.1877e-02	1.5537e-02
			SASA	7.7887e-01	−2.8063e-01	−1.2968e-03	3.9328e-04
			SASA / NLR	8.6840e-01	−4.1776e-01	−8.5757e-04	7.8104e-04
C *sp*3	2	2177	vdW	7.1564e-01	−4.5311e-01	−2.8826e-02	1.1670e-02
			SASA	5.6482e-01	−1.9608e-01	−1.0219e-03	2.6580e-04
			SASA / NLR	6.2286e-01	−2.8190e-01	−2.4698e-03	5.3606e-04
C *sp*3	3	2655	vdW	4.5576e-01	−3.0530e-01	−2.7581e-02	8.9519e-03
			SASA	2.3348e-01	−7.2627e-02	−2.0079e-04	7.9670e-05
			SASA / NLR	2.8368e-01	−1.2982e-01	−1.5757e-03	2.4514e-04
C *sp*3	4	11	vdW	5.1728e-02	−2.7948e-02	−4.6281e-03	9.7076e-04
			SASA[a]	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
			SASA / NLR[a]	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
C *sp*2	2	1009	vdW	7.3022e-01	−5.7396e-01	−3.1689e-02	1.4689e-02
			SASA	5.1245e-01	−1.5966e-01	−1.9781e-04	1.6392e-04
			SASA / NLR	6.1006e-01	−2.4859e-01	1.9453e-03	2.7405e-04

Appendix B.
(Continued)

Atom Type	No. Bonded Neighbors	No. in All Test Compounds	Surface Type	P_1	P_2	P_3	P_4
C *sp*2	3	2750	vdW	4.7712e-01	−4.0301e-01	−1.9911e-02	9.3037e-03
			SASA	7.0344e-02	−1.9015e-02	−2.2009e-05	1.6875e-05
			SASA / NLR	8.9938e-02	−3.6938e-02	3.1896e-05	4.7525e-05
O *sp*3	1	523	vdW	9.8358e-01	−8.0584e-01	−3.8046e-02	1.8814e-02
			SASA	7.7914e-01	−2.5262e-01	−1.6056e-03	3.5071e-04
			SASA / NLR	9.2186e-01	−4.2574e-01	−1.1638e-03	7.6747e-04
O *sp*3	2	732	vdW	8.3864e-01	−6.5020e-01	−3.0516e-02	1.4112e-02
			SASA	4.9392e-01	−1.6038e-01	−1.5512e-04	1.6453e-04
			SASA / NLR	8.4608e-01	−3.7070e-01	4.5319e-03	4.0309e-04
O *sp*2	1	2187	vdW	1.0469e+00	−9.2023e-01	−1.9316e-02	1.3093e-02
			SASA	6.8563e-01	−1.8680e-01	−1.35573-03	2.3743e-04
			SASA / NLR	8.5289e-01	−3.5065e-01	1.4472e-03	4.7236e-04
O^- carboxylate	1	347	vdW	1.0154e+00	−7.0915e-01	−1.9396e-02	9.3825e-03
			SASA	8.8857e-01	−3.3421e-01	−1.8683e-03	4.9372e-04
			SASA / NLR	1.0454e+00	−5.2088e-01	1.9397e-03	8.7727e-04
N *sp*3	1	77	vdW	9.8589e-01	−8.6945e-01	−4.5999e-02	2.3146e-02
			SASA	7.8602e-02	−2.9198e-01	−6.5370e-04	3.6247e-04
			SASA / NLR	8.2167e-01	−3.8471e-01	−4.6166e-03	8.2382e-04
N *sp*3	2	20	vdW	8.2877e-01	−6.4026e-01	−1.5534e-02	1.1165e-02
			SASA	2.2599e-01	−3.6648e-02	−1.2297e-03	8.0038e-05
			SASA / NLR	2.4364e-01	−6.1578e-02	−6.0261e-03	2.6117e-04
N *sp*3	3	6	vdW	5.6332e-01	−5.0591e-01	−3.2932e-02	1.2708e-02
			SASA	5.1481e-02	−1.2603e-02	−3.2006e-04	2.4774e-05
			SASA / NLR	6.8601e-02	−3.5060e-02	1.8874e-03	−2.0272e-06
N *sp*2	1	459	vdW	1.0883e+00	−9.1239e-01	−1.4189e-02	1.3969e-02
			SASA	7.3511e-01	−2.2116e-01	−8.9148e-04	2.5230e-04
			SASA / NLR	8.7727e-01	−3.8862e-01	1.7690e-03	5.4838e-04
N *sp*2	2	2017	vdW	8.1368e-01	−5.8659e-01	−1.7450e-02	1.0426e-02
			SASA	4.1102e-01	−1.2254e-01	−7.5448e-05	1.1804e-04
			SASA / NLR	5.3237e-01	−2.0830e-01	2.5496e-03	1.9004e-04
N *sp*2	3	289	vdW	4.8967e-01	−3.7691e-01	−1.7523e-02	7.7321e-03
			SASA	6.2577e-02	−1.7874e-02	−8.3120e-05	1.9849e-05
			SASA / NLR	9.5163e-02	−3.7113e-02	2.5453e-04	3.9563e-05
S	1	5	vdW	9.9626e-01	−9.5899e-01	−3.6874e-02	2.5016e-02
			SASA	7.7220e-01	−2.6393e-01	1.0629e-03	2.1790e-04
			SASA / NLR	8.3709e-01	−3.1119e-01	1.3049e-02	−2.6910e-04
S	2	59	vdW	9.4768e-01	−8.3254e-01	−1.8536e-02	1.4674e-02
			SASA	5.4581e-01	−1.9477e-01	−1.2873e-03	2.9247e-04
			SASA / NLR	5.0886e-01	−2.3673e-01	−3.0065e-03	5.1519e-04
P	3	7	vdW	9.9646e-01	−9.2223e-01	7.6869e-02	−3.1453e-03
			SASA	3.8650e-01	−1.8249e-01	−3.6598e-03	4.2640e-04
			SASA / NLR	5.6715e-01	−4.6888e-01	8.0448e-03	9.1286e-04
P	4	237	vdW	5.0976e-01	−4.5064e-01	−2.0714e-02	9.7325e-03
			SASA	3.8730e-02	−8.9339e-03	8.3582e-06	3.0381e-06
			SASA / NLR	5.9645e-02	−2.8582e-02	3.3161e-04	2.7995e-05
Cl	1	5	vdW	9.7831e-01	−5.4685e-01	−1.6495e-02	5.6367e-03
			SASA	9.8318e-01	−4.0437e-01	1.1249e-04	4.9901e-04
			SASA / NLR	1.2902e+00	−9.6946e-01	−1.2609e-02	2.8306e-03

[a] All atoms are completely buried.

References

1. Lee, B.; Richards, F. M. J Mol Biol 1971, 55, 379–400.
2. Hermann, R. B. J Phys Chem 1972, 76, 2754–2759.
3. Hasel, W.; Hendrickson, T. F.; Still, W. C. Tetrahedron Comput Methodol 1988, 1, 103–116.
4. Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T. J Am Chem Soc 1990, 112, 6127–6129.
5. Ooi, T.; Némethy, G.; Scheraga, H. A. Proc Natl Acad Sci USA 1987, 84, 3086–3090.
6. Wesson, L.; Eisenberg, D. Protein Sci 1992, 1, 227–235.
7. Wodak, S. J.; Janin, J. Proc Natl Acad Sci USA 1980, 77, 1736–1740.
8. Richmond, T. J. J Mol Biol 1984, 178, 63–89.
9. Dodd, L. R.; Theodorou, D. N. Mol Phys 1991, 72, 1313–1345.
10. Perrot, G.; Cheng, B.; Gibson, K. D.; Vila, J.; Palmer, K. A.; Nayeem, A.; Maigret, B.; Scheraga, H. A. J Comput Chem 1992, 13, 1–11.
11. von Freyberg, B.; Braun, W. J Comput Chem 1993, 14, 510–521.
12. Eisenhaber, F.; Argos, P. J Comput Chem 1993, 14, 1272–1280.
13. Mumenthaler, C.; Braun, W. J Mol Model 1995, 1, 1–10.
14. Kurochkina, N.; Lee, B. Protein Eng 1995, 8, 437–442.
15. Liotard, D. A.; Hawkins, G. D.; Lynch, G. C.; Cramer, C. J.; Truhlar, D. G. J Comput Chem 1995, 16, 422–440.
16. Sridharan, S.; Nicholls, A.; Sharp, K. A. J Comput Chem 1995, 16, 1038–1044.
17. Sanner, M. F.; Olson, A. J.; Spehner, J.-C. Biopolymers 1996, 38, 305–320.
18. Gabdoulline, R. R.; Wade, R. C. J Mol Graphics 1996, 14, 341–353.
19. Cossi, M.; Mennucci, B.; Cammi, R. J Comput Chem 1996, 17, 57–73.
20. Fraczkiewicz, R.; Braun W. J Comput Chem 1998, 19, 319–333.
21. Cui, Y.; Chen, R. S.; Wong, W. H. Proteins 1998, 31, 247–257.
22. Weiser, J.; Weiser, A. A.; Shenkin, P. S.; Still, W. C. J Comput Chem 1988, 19, 797–808.
23. Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. Numerical Recipes in FORTRAN, Cambridge University Press: Cambridge, U.K. 1992.
24. Bernstein, F. C.; Koetzle, T. F.; Williams, G. J. B.; Meyer, E. F. Jr.; Brice, M. D.; Rodgers, J. R.; Kennard, O.; Shimanouchi, T.; Tasumi, M. J Mol Biol 1977, 112, 535–542. The Brookhaven Protein Data Bank is accessible via the internet at http://pdb.pdb.bnl.gov/.
25. Weiser, J.; Holthausen, M. C.; Fitjer, L. J Comput Chem 1997, 18, 1264–1281. Supplementary Material including data for Sipholenol-A monoacetate is available via the internet at http://journals.wiley.com/0192-8561/wilma/ wilma.cgi/ v18.1264.html/.
26. Mohamadi, F.; Richards, N. G. J.; Guida, W. C.; Liskamp, R.; Lipton, M.; Caufield, C.; Chang, G.; Hendrickson, T.; Still, W. C. J Comput Chem 1990, 11, 440–467. We used MacroModel Version 6.0.
27. Grant, J. A.; Pickup, B. T. J Phys Chem 1995, 99, 3503–3510.