Improved Methods for Semiempirical Solvation Models

DANIEL A. LIOTARD*

*Laboratoire de Physicochimie Theorique, Université de Bordeaux 1, 351 cours de la Liberation,
33405 Talence CEDEX, France*

**GREGORY D. HAWKINS, GILLIAN C. LYNCH,
CHRISTOPHER J. CRAMER,* and DONALD G. TRUHLAR***

*Department of Chemistry and Supercomputer Institute, University of Minnesota,
Minneapolis, Minnesota 55455-0431*

Received 16 March 1994; accepted 17 June 1994

ABSTRACT

We present improved algorithms for the SMx ($x = 1, 1\text{a}, 2, 3$) solvation models presented previously [see the overview in C. J. Cramer and D. G. Truhlar, *J. Comp.-Aided Mol. Design*, **6**, 629 (1992)]. These models estimate the free energy of solvation by augmenting a semiempirical Hartree-Fock calculation on the solute with the generalized Born (GB) model for electric polarization of the solvent and a surface tension term based on solvent-accessible surface area. This article presents three improvements in the algorithms used to carry out such calculations, namely (1) an analytical accessible surface area algorithm, (2) a more efficient radial integration scheme for the dielectric screening computation in the GB model, and (3) a damping algorithm for updating the GB contribution to the Fock update during the iterations to achieve a self-consistent field. Improvements (1) and (2) decrease the computer time, and improvement (3) leads to more stable convergence. Improvement (2) removes a small systematic numerical error that was explicitly absorbed into the parameterization in the SMx models. Therefore, we have adjusted the parameters for one of the previous models to yield essentially identical performance as was obtained originally while simultaneously taking advantage of improvement (2). The resulting model is called SM2.1. The fact that we obtain similar results after removing the systematic quadrature bias attests to the robustness of the original parameterization. © 1995 by John Wiley & Sons, Inc.

*Authors to whom correspondence may be addressed.

Journal of Computational Chemistry, Vol. 16, No. 4, 422–440 (1995)

CCC 0192-8651 / 95 / 040422-19

Introduction

There has been considerable interest in recent years in including the effects of solvation in molecular orbital calculations by treating the solvent as a continuum. This work is summarized in a recently completed review.[1] The present work has been motivated by the success achieved with an approach which combines (1) the generalized Born (GB) model[2-5] of electric polarization throughout the volume of a bulk continuum dielectric surrounding the solute, and (2) a surface tension model[6-8] of additional free energy effects due to interactions and structural changes in the first solvation shell. For the latter, the number of solvent molecules in the first solvation shell of the continuum solvent is taken to be proportional to the solvent-accessible surface area (SASA)[9] with the proportionality constant being a surface tension (ST); thus, these models may be called GB/ST models.

The first GB/ST model was proposed in the context of molecular mechanics by Still et al.,[10] who also introduced a dielectric screening model for the GB term that takes account of the molecular shape of the solute modeled as a set of atomic spheres, which for typical geometries are overlapping. Both the dielectric screening algorithm and the SASA calculation require repeated calculations of the exposed surface area of a sphere overlapping various collections of other spheres, which is the same numerical algorithm required to calculate solvent-accessible surface area. Still and coworkers[11] proposed and employed an approximate but rapid algorithm for this computational step. Still et al. used a single surface tension for all parts of the solute surface in the SASA terms.

The GB/ST type of model was extended to molecular orbital theory by two of the present authors,[12-15] who also introduced a dependence of the atomic surface tensions on the type of atom. These extensions resulted in a series of parameterized solvation models called SM1,[12] SM1a,[12] SM2,[13] and SM3[14] or—when referring to the whole set—SM*x*.[1] The use of molecular orbital theory allows for solute electronic charge redistribution in the presence of solvent[13,16,17] and for the treatment of structures for which molecular mechanics is inappropriate (e.g., transition states of organic reactions[17] or stable states of species for which molecular mechanics parameters are unavailable or inadequate). The SM*x* models retain the dielectric screening model of Still et al.[10] with minor modifications.[12-15]

In implementing the SM*x* and other GB/ST models, a critical computational quantity is the solvent-accessible surface area (i.e., exposed surface area) of a single sphere in a set of overlapping spheres. This quantity is required both in the dielectric screening algorithm of the electrostatic part of the calculation (i.e., the GB part) and in the calculation of the first-solvation-shell effects. In AMSOL—versions 1 to 3.0.1—exposed surface areas were calculated only by a numerical algorithm, called DOTS, based on a quasi-uniform distribution of quadrature nodes (dots) on spherical surfaces.[12,15,18] A later version of the computer code[19] provided a more efficient implementations of the DOTS algorithm and also incorporated an algorithm,[20-22] called GEPOL, for this same computation based on equal subdivisions of pentakisdodecahedrons on the spherical surfaces.

This article presents three computational improvements in the algorithms used for SM*x* and other GB/ST calculations, at least one of which (number 1 in the following list) may also be of use in a much wider context. The three improvements advanced in this article are as follows:

1. An analytic exposed surface area algorithm, which we will call the ASA algorithm, is proposed for the calculation of exposed surface areas.
2. A more efficient numerical quadrature is proposed for the specialized form of radial integration that plays a pivotal role in the dielectric screening portion of the GB algorithm. Reducing the number of quadrature nodes in this integration reduces the number of required calculations of the exposed surface areas, thereby reducing the computational effort.

Enhancements (1) and (2) also cut down on the level of numerical noise during the self-consistent field (SCF) cycles, thereby leading to smoother convergence. Finally,

3. A damping scheme is proposed for updating the GB terms during the SCF cycles. This is particularly important because the GB solvation contribution is highly nonlinear in the electronic density matrix elements.

Enhancements (1) and (3) were implemented in version 4.0 of the AMSOL computer code,[23] and enhancements 1 through 3 have all been implemented in version 4.1 of the AMSOL code,[24] and they will be available in a new version of AMPAC[25] as well. In the process of implementing these changes in AMSOL version 4.1, we also eliminated some inefficiencies in the way the DOTS and GEPOL methods had been coded. This article will compare well-coded versions of all three algorithms.

The following sections present the algorithmic enhancements, evaluate their performance, and discuss the necessary reparameterization.

Theory

ANALYTIC EXPOSED SURFACE AREAS

Background

The exposed surface area, A_i, of a sphere i partially buried in a set of other spheres is the quantity which must be calculated efficiently in the SM*x* methods. Previously, in the DOTS and GEPOL algorithms mentioned earlier, we used numerical methods in which a set of points was placed reasonably uniformly on the surface of each sphere. Any point on a sphere that lay within the volume of another sphere was discarded, and the ratio of the exposed surface area to the total area of the sphere was set equal to the number of nondiscarded points divided by the original number of points. These methods can yield two to three significant figures of accuracy using several hundred to 1000 points per sphere. To improve the accuracy, the number of points on each sphere must be increased, and such increases rapidly become computationally expensive. When large molecules (e.g., those of pharmaceutical or biological interest) are examined, one is motivated to seek more efficient algorithms.

A properly implemented analytic method can sometimes deliver a higher degree of accuracy than a numerical method. Several years ago, Connolly developed the analytic expressions for exposed surface areas,[26] but a code that incorporates his approach has recently tested slower than a numerical calculation.[27] The problem is that Connolly's analytic approach requires an extensive topological analysis of each spherical intersection. Because typically many of these intersections are buried within the volume of the set of spheres, a large fraction of the computational time is spent calculating intermediate quantities that are never used. Additionally, floating-point round-off errors cause topological inconsistencies near junction points, where three or more spheres intersect or nearly intersect on the surface of a fourth sphere, and such junction points can cause serious errors in the calculation of the exposed area, A_i, of the fourth sphere.

Recently, a fast analytical approach for calculating the exposed surface area of a molecule has been proposed.[28] This yields a fast computation not only of the area but also of its derivatives with respect to atomic coordinates.[29] Unfortunately, this algorithm does not consider every topological possibility and is still sensitive to junction points. The resulting errors are not too severe when only the overall exposed area is required, but the method is unsuitable for calculating the exposed areas of individual atomic spheres, as required here.

Mathematical Notation

In this article we will denote spheres by i, j, k, l, and h. The points at the centers of these spheres will be denoted I, J, K, L, and H. Other points will be denoted P, Q, and S, sometimes with subscripts and/or superscripts.

A vector connecting two points, for example, I and P, will be written

$$\mathbf{r}_{IP} = x_{IP}\hat{\mathbf{e}}_x + y_{IP}\hat{\mathbf{e}}_y + z_{IP}\hat{\mathbf{e}}_z \tag{1}$$

We will also need a notation for unit vectors (e.g., $\hat{\mathbf{r}}_{IP}$) and magnitudes of vectors (e.g., r_{IP}) related by

$$\mathbf{r}_{IP} = r_{IP}\hat{\mathbf{r}}_{IP} \tag{2}$$

and we will need a notation for the components of unit vectors, namely

$$\hat{\mathbf{r}}_{IP} = d_x^{IP}\hat{\mathbf{e}}_x + d_y^{IP}\hat{\mathbf{e}}_y + d_z^{IP}\hat{\mathbf{e}}_z \tag{3}$$

Let $\Phi(P, Q, S)$ be the angle between $\hat{\mathbf{r}}_{QP}$ and $\hat{\mathbf{r}}_{QS}$ so that

$$\cos[\Phi(P, Q, S)] = \hat{\mathbf{r}}_{QP} \cdot \hat{\mathbf{r}}_{QS} \tag{4}$$

Mathematical Formulation and Algorithmic Implementation

The new algorithm presented here will be called the ASA algorithm. It calculates the exposed surface area, A_i, as R_i^2 times the exposed surface solid angle, Ω_i, of a sphere i of radius R_i within a set of spheres.

The present approach calculates Ω_i as the total solid angle of the surface of the sphere (4π) minus the solid angles covered by several pieces built from eventually overlapping buried spherical caps. The union of these spherical segments admits an exact partitioning involving only three categories of objects—namely, whole spherical caps, slices of spherical caps, and spherical polygons of p sides ($p > 3$).

These pieces have known mensuration formulas.[26] The algorithm computes one Ω_i at a time, for sphere i among all the other spheres. To proceed with the determination Ω_i, we must first construct a list, L_i, of the spheres within the set that intersect sphere i. For any sphere j, if

$$d_{IJ} < R_i + R_j \tag{5}$$

where d_{IJ} is the distance between I and J, then j intersects i. If d_{IJ} plus R_i is less than R_j, then sphere i is buried within sphere j, and 0 is returned for the value of Ω_i. If d_{IJ} plus R_j is less than R_i, then sphere j is buried within sphere i, and j is dropped from L_i. If no spheres intersect i, then Ω_i is given the value 4π and returned.

Vectors from I to the centers of all spheres in L_i are constructed, and the cosines of the angles between these unit vectors and vectors from I to the circle of intersection of the spheres are calculated by

$$\cos \gamma_j^{(i)} = \frac{(d_{IJ})^2 + (r_i)^2 - (r_j)^2}{2 d_{IJ} r_i} \tag{6}$$

where j is a sphere in L_i, and $2\pi(1 - \cos \gamma_j^{(i)})$ is the solid angle of sphere i buried by sphere j. The surface of a sphere buried by another sphere is called a spherical cap (see Fig. 1). We will use the notation $SC_j^{(i)}$ to denote the spherical cap of i buried by sphere j.

Next a Boolean connectivity matrix between the spherical caps on the surface of sphere i is constructed. Initially, a connectivity matrix element $C_{jk}^{(i)}$ of the connectivity matrix $\mathbf{C}^{(i)}$ is defined to be true if $SC_j^{(i)}$ intersects with $SC_k^{(i)}$. It is defined to be false otherwise. The spherical caps of any two spheres j and k on the list L_i intersect on the surface of sphere i if

$$\left|\cos \Phi(J, I, K) - \cos \gamma_j^{(i)} \cos \gamma_k^{(i)}\right| < \sin \gamma_j^{(i)} \sin \gamma_k^{(i)} \tag{7}$$

For any two spheres j and k on the list L_i that do not satisfy the inequality in eq. (7) but do satisfy

$$\cos \Phi(J, I, K) - \cos \gamma_j^{(i)} \cos \gamma_k^{(i)} > \sin \gamma_j^{(i)} \sin \gamma_k^{(i)} \tag{8}$$

then we know one spherical cap is contained in another. In particular, if $\cos \gamma_j^{(i)} > \cos \gamma_k^{(i)}$, then $SC_k^{(i)}$ is contained within $SC_j^{(i)}$, and k is dropped from L_i. If $\cos \gamma_j^{(i)} = \cos \gamma_k^{(i)}$, then $SC_k^{(i)}$ lies directly on top of $SC_j^{(i)}$, and one of the two spheres is dropped from list L_i.

The code also detects whether sphere i is completely buried by two spherical segments, in which

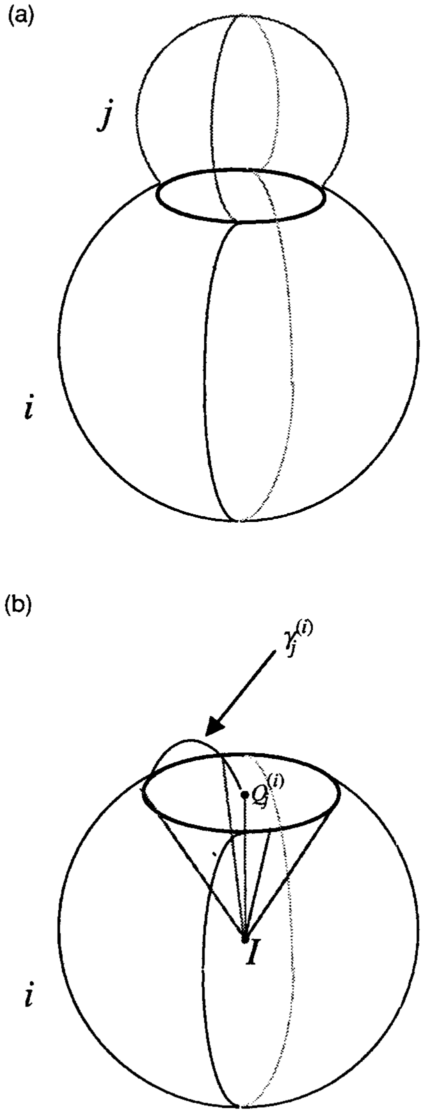

FIGURE 1. (a) The intersection of two spheres i and j. (b) Sphere i and the spherical cap created by its intersection with sphere j. The angle $\gamma_j^{(i)}$ is defined below eq. (6). The point $Q_j^{(i)}$ is the point that lies on the surface of i and on the line that connects I and J.

case the value 0 is returned for Ω_i. For each sphere j on the list L_i, whose spherical cap does not intersect with any of the other spherical caps, the angle $\gamma_j^{(i)}$ is extracted from the cosine. Then $\omega_{SC}^{(i)}$, the total solid angle of i eclipsed by spherical caps that do not intersect any other spherical caps, is defined as

$$\omega_{SC}^{(i)} = 2\pi \sum_j \left(1 - \cos\gamma_j^{(i)}\right) \qquad (9)$$

When $\gamma_j^{(i)}$ for a given sphere is added to the sum in eq. (9), that sphere is dropped from L_i. If there are no more spheres left in L_i, then the value of Ω_i is set equal to $4\pi - \omega_{SC}^{(i)}$ and returned.

If L_i is not a null set at this stage of the calculation, then there must be intersecting spherical caps. For a pair of intersecting spherical caps, $SC_k^{(i)}$ and $SC_j^{(i)}$, there are exactly two points where their circular boundaries intersect. Let these points be $P_{jk}^{(i)}$ and $P_{kj}^{(i)}$ (see Fig. 2). A list, M_i, of all such points is created.

To avoid errors which would cause, later in the calculation, the same kind of numerical instabilities as observed in other methods,[28,29] we check to see if any point, $P_{jk}^{(i)}$, lies within a predetermined angle range of 1×10^{-11} radians of any spherical cap, $SC_l^{(i)}$, due to a sphere other than j or k. If this occurs, the point $P_{jk}^{(i)}$ is called a junction point. If the point $P_{jk}^{(i)}$ is not within an angle range of 1×10^{-11} radians of any third spherical cap, it is called a free point. To check for this occurrence, the unit vectors $\hat{\mathbf{r}}_{IP_{jk}^{(i)}}$, pointing from I to point $P_{jk}^{(i)}$, and $\hat{\mathbf{r}}_{IP_{kj}^{(i)}}$, pointing from I to point $P_{kj}^{(i)}$, must be defined. We will derive expressions for both unit vectors simultaneously, and thus we will use the notation $\hat{\mathbf{r}}_{IP_{[j,k]}^{(i)}}$ to indicate that the equations are true for both intersections of $SC_k^{(i)}$ and $SC_j^{(i)}$. The unit vector $\hat{\mathbf{r}}_{IP_{[j,k]}^{(i)}}$ can be defined as follows:

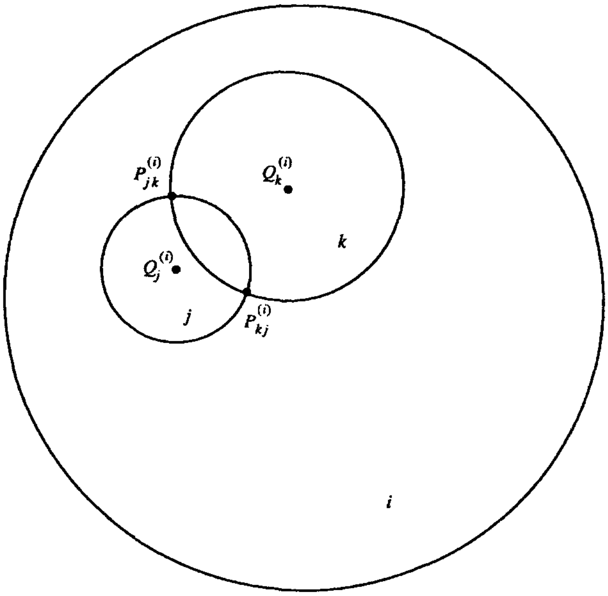

FIGURE 2. The spherical caps, $SC_k^{(i)}$ and $SC_j^{(i)}$ on the surface of sphere i. The points $P_{jk}^{(i)}$ and $P_{kj}^{(i)}$ are the two points where the circular boundary $SC_k^{(i)}$ intersects with the circular boundary of $SC_j^{(i)}$.

$$\hat{\mathbf{r}}_{IP_{[j,k]}^{(i)}} = A\hat{\mathbf{r}}_{IJ} + B\hat{\mathbf{r}}_{IK} + C(\hat{\mathbf{r}}_{IJ} \times \hat{\mathbf{r}}_{IK}) \qquad (10)$$

where A, B, and C have yet to be determined. To solve for the three unknown values, we note three conditions which $\hat{\mathbf{r}}_{IP_{[j,k]}^{(i)}}$ must satisfy. The relation

$$\hat{\mathbf{r}}_{IP_{[j,k]}^{(i)}} \cdot \hat{\mathbf{r}}_{IJ} = \cos\gamma_j^{(i)} \qquad (11)$$

must hold because $P_{[j,k]}^{(i)}$ lies on the circular border of $SC_j^{(i)}$. Similarly, the relation

$$\hat{\mathbf{r}}_{IP_{[j,k]}^{(i)}} \cdot \hat{\mathbf{r}}_{IK} = \cos\gamma_k^{(i)} \qquad (12)$$

must hold because $P_{[j,k]}^{(i)}$ lies on the circular border of $SC_k^{(i)}$. Finally, the relation

$$\hat{\mathbf{r}}_{IP_{[j,k]}^{(i)}} \cdot \hat{\mathbf{r}}_{IP_{[j,k]}^{(i)}} = 1 \qquad (13)$$

because the vector $\hat{\mathbf{r}}_{IP_{[j,k]}^{(i)}}$ is a unit vector. Substituting eq. (10) into eq. (11) yields

$$\cos\gamma_j^{(i)} = A(\hat{\mathbf{r}}_{IJ} \cdot \hat{\mathbf{r}}_{IJ}) + B(\hat{\mathbf{r}}_{IK} \cdot \hat{\mathbf{r}}_{IJ}) + C((\hat{\mathbf{r}}_{IJ} \times \hat{\mathbf{r}}_{IK}) \cdot \hat{\mathbf{r}}_{IJ}) \qquad (14)$$

Substituting eq. (10) into eq. (12) yields

$$\cos\gamma_k^{(i)} = A(\hat{\mathbf{r}}_{IJ} \cdot \hat{\mathbf{r}}_{IK}) + B(\hat{\mathbf{r}}_{IK} \cdot \hat{\mathbf{r}}_{IK}) + C((\hat{\mathbf{r}}_{IJ} \times \hat{\mathbf{r}}_{IK}) \cdot \hat{\mathbf{r}}_{IK}) \qquad (15)$$

Keeping in mind that $(\hat{r}_{IJ} \times \hat{\mathbf{r}}_{IK}) \cdot \hat{\mathbf{r}}_{IJ}$ and $(\hat{\mathbf{r}}_{IJ} \times \hat{\mathbf{r}}_{IK}) \cdot \hat{\mathbf{r}}_{IK}$ must equal zero and using eq. (4), the system of eqs. (14) and (15) can be solved to give

$$A = \frac{\cos\gamma_j^{(i)} - \cos\gamma_k^{(i)} \cos[\Phi(J, I, K)]}{\sin^2[\Phi(J, I, K)]} \qquad (16)$$

and

$$B = \frac{\cos\gamma_k^{(i)} - \cos\gamma_j^{(i)} \cos[\Phi(J, I, K)]}{\sin^2[\Phi(J, I, K)]} \qquad (17)$$

Note that $\sin[\Phi(J, I, K)] = 0$ only in cases where one spherical cap is completely buried by another spherical cap. Such instances are dealt with earlier in the algorithm. Using eqs. (13) and (10), we find

$$\hat{\mathbf{r}}_{IP^{(i)}_{[j,k]}} \cdot (A\hat{\mathbf{r}}_{JI} + B\hat{\mathbf{r}}_{IK} + C(\hat{\mathbf{r}}_{IJ} \times \hat{\mathbf{r}}_{IK})) = 1 \quad (18)$$

Finally, using eqs. (11) and (12), we find that

$$C = \pm\sqrt{\frac{1 - A\cos\gamma_j^{(i)} - B\cos\gamma_k^{(i)}}{\sin^2[\Phi(J, I, K)]}} \quad (19)$$

The check for a junction point can then be accomplished using the following equation:

$$\left|\hat{\mathbf{r}}_{IP^{(i)}_{jk}} \cdot \hat{\mathbf{r}}_{IL} - \cos\gamma_k^{(i)}\right| < (1 \times 10^{-11})\sin\gamma_l^{(i)} \quad (20)$$

If the inequality is satisfied, then the radius of sphere i is increased by a small increment, 4 × 10^{-11} in the current code, and the calculation is restarted.

If a point $P^{(i)}_{jk}$ of M_i does not lie within the volume of a sphere other than sphere j or k, it is defined as a free point. Thus, for every sphere l in L_i not equal to sphere j or k, if

$$\hat{\mathbf{r}}_{IL} \cdot \hat{\mathbf{r}}_{IP^{(i)}_{jk}} \leq \cos\gamma_l^{(i)} \quad (21)$$

then $P^{(i)}_{jk}$ is a free point. Otherwise we drop $P^{(i)}_{jk}$ from M_i and modify the connectivity matrix so that an element, $C^{(i)}_{jk}$, is true only if $P^{(i)}_{jk}$ is an element of M_i. In other words, $C^{(i)}_{jk}$ can be false even if $SC^{(i)}_j$ intersects $SC^{(i)}_k$. Note that if $C^{(i)}_{jk}$ is true, we say that j is connected to k. The list L_i is also revised so that if $SC^{(i)}_j$ is completely buried by other spherical caps, it is dropped. Note that this is equivalent to saying that there are no free points on the circular boundary of $SC^{(i)}_j$.

By using the connectivity matrix, the rest of the buried solid angle of i can be decomposed into contributions from clusters of intersecting spherical caps. Define the point $Q^{(i)}_j$, where j is some sphere, to be the point that lies on the surface of sphere i and on a line connecting J and I (see Fig. 3). The solid angle buried by each cluster can be divided into two portions. The first portion is comprised of spherical polygons which can be visualized by noting that each free point is a vertex of a spherical polygon and that each point $Q^{(i)}_j$ may be the vertex of one or more spherical polygons. Then the surface(s) inscribed in lines drawn from $Q^{(i)}_j$ to $P^{(i)}_{jk}$ and from $P^{(i)}_{jk}$ to $Q^{(i)}_k$ is (are) the spherical polygon(s). Note that each spherical polygon has an even number of sides (≥ 4) (see Fig. 4). The portion of each spherical cap that is not buried by one of the spherical polygons is defined

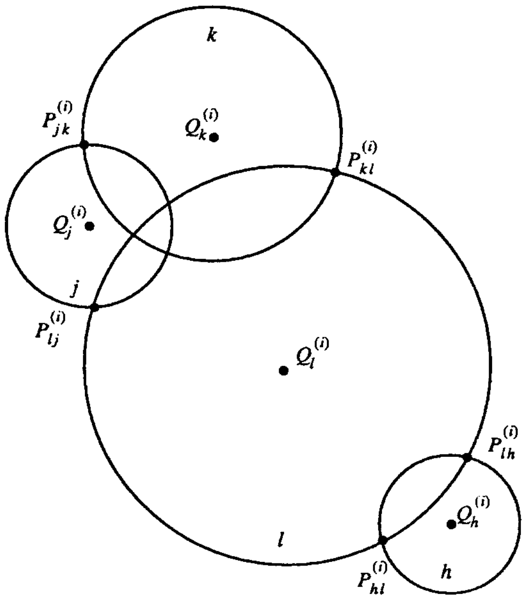

FIGURE 3. A planar projection of a cluster of four spherical caps on the surface of sphere i. All the points $P^{(i)}$ that are pictured are free points.

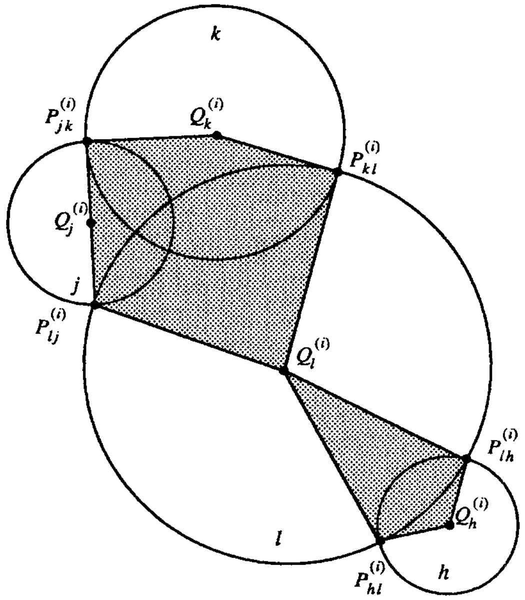

FIGURE 4. A planar projection of a cluster of spherical caps on the surface of sphere i. The unshaded areas are spherical cap sections, whereas the shaded areas comprise the spherical polygons of eclipsed solid angle.

as a slice of a spherical cap and comprises our second portion of buried solid angle.

The solid angle buried by all spherical cap slices, $\omega_{SS}^{(i)}$, is defined as follows:

$$\omega_{SS}^{(i)} = \sum_{j}\left(2\pi - \sum \theta_{Q_j^{(i)}}^{(i)}\right)\left(1 - \cos\gamma_j^{(i)}\right) \quad (22)$$

where $\theta_{Q_j^{(i)}}^{(i)}$ is the interior angle of one of the vertices of a spherical polygon centered at $Q_j^{(i)}$. To calculate the angles of the spherical polygon vertices centered on a certain Q point, $\theta_{Q_j^{(i)}}^{(i)}$, for each sphere j, a free point of the form $P_{jk}^{(i)}$ or $P_{lj}^{(i)}$ must be chosen to be the first point. Next the angles from the first free point on sphere j to all other free points on sphere j are determined following trigonometric conventions. These angles may be either very small or near 2π, and therefore we need information from both the sine and the cosine. We combine this information into one expression which will minimize floating-point round-off effects, which could cause abrupt errors of exactly 2π if the ASIN and ACOS functions were used. If the first free point is $P_{jk}^{(i)}$, the angles $\Phi(P_{jk}^{(i)}, Q_j^{(i)}, P_{lj}^{(i)})$ can be calculated by the following formula, where k and l are any spheres:

$$\Phi\left(P_{jk}^{(i)}, Q_j^{(i)}, P_{lj}^{(i)}\right) = \mathrm{ATAN2}\frac{\hat{\mathbf{r}}_{P_{jk}^{(i)}i} \cdot \hat{\mathbf{r}}_{ik} \times \hat{\mathbf{r}}_{P_{lj}^{(i)}i}}{\hat{\mathbf{r}}_{P_{jk}^{(i)}i} \cdot \hat{\mathbf{r}}_{P_{lj}^{(i)}i} - \cos^2\gamma_j^{(i)}} \quad (23)$$

Because $\Phi(P_{jk}^{(i)}, Q_j^{(i)}, P_{lj}^{(i)})$ must be an angle between 0 and 2π and the ATAN2 function returns a value between $-\pi$ and π, if $\Phi(P_{jk}^{(i)}, Q_j^{(i)}, P_{lj}^{(i)})$ is negative for any combination k and l, we add 2π. The angles $\Phi(P_{jk}^{(i)}, Q_j^{(i)}, P_{jl}^{(i)})$ are calculated similarly. We then sort the set of angles $\Phi(P_{jk}^{(i)}, Q_j^{(i)}, P_{lj}^{(i)})$ and $\Phi(P_{jk}^{(i)}, Q_j^{(i)}, P_{jl}^{(i)})$ and rename them in ascending order so that λ_1 is the smallest angle and, if there are n such angles, λ_n is the largest. Note that n will always be even because of the elimination of junction points, as described earlier, and that λ_n will always be equal to 2π. We must then determine whether λ_1 is the interior angle at a vertex of a spherical polygon or whether it defines a spherical cap slice. At this point, it is important to note that for the purposes of this algorithm, the interior and the exterior of each spherical polygon were defined as local properties (which they are not). The interior of each polygon was defined so that angles of the type $\Phi(Q_j^{(i)}, P_{jk}^{(i)}, Q_k^{(i)})$ always lie between 0 and π. Although this trick avoids a global topological analysis of the entire sphere i, it also can cause orientation errors, which will be dealt with shortly.

Suppose the angle $\Phi(P_{jk}^{(i)}, Q_j^{(i)}, P_{lj}^{(i)})$ equals λ_1. Then if

$$\hat{\mathbf{r}}_{JI} \cdot \hat{\mathbf{r}}_{KI} \times \hat{\mathbf{r}}_{P_{lj}^{(i)}I} > 0 \quad (24)$$

λ_1 is the interior angle at a vertex of a spherical polygon, and

$$\sum \theta_{Q_j^{(i)}}^{(i)} = \lambda_1 + \sum_{f=1}^{\frac{n}{2}-1} (\lambda_{2f+1} - \lambda_{2f}) \quad (25)$$

If eq. (24) is not true, then

$$\sum \theta_{Q_j^{(i)}}^{(i)} = \sum_{f=1}^{\frac{n}{2}} (\lambda_{2f} - \lambda_{2f-1}) \quad (26)$$

See Figure 5.

To calculate the solid angle buried by the spherical polygons, the rest of the vertex angles, of the form $\Phi(Q_j^{(i)}, P_{jk}^{(i)}, Q_k^{(i)})$ where $P_{jk}^{(i)}$ is a free point and j and k are any spheres, must be calculated. As previously mentioned, the angle $\Phi(Q_j^{(i)}, P_{jk}^{(i)},$

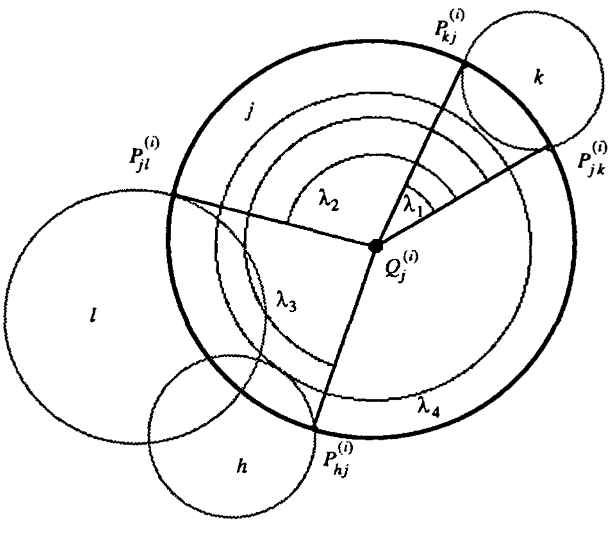

FIGURE 5. A planar mapping of $SC_j^{(i)}$ in a cluster of spherical caps on the surface of sphere i. $P_{jk}^{(i)}$ has been chosen as the first free point, and the λ angles are as listed. Note that angles λ_1 and $\lambda_3 - \lambda_2$ are the interior angles of spherical polygons, and $\lambda_2 - \lambda_1$ and $\lambda_4 - \lambda_3$ are angles of defining spherical cap sections.

$Q_k^{(i)}$) lies between 0 and π and therefore can be calculated with the following formula:

$$\Phi\left(Q_j^{(i)}, P_{jk}^{(i)}, Q_k^{(i)}\right) = \arccos\frac{\Phi(J, I, K) - \cos\gamma_j^{(i)}\cos\gamma_k^{(i)}}{\sin\gamma_j^{(i)}\sin\gamma_k^{(i)}} \quad (27)$$

In general, the formula for the solid angle of sphere i buried by a spherical polygon with n sides, $\omega_{\text{SP}}^{(i)}$, can be written as

$$\omega_{\text{SP}}^{(i)} = \sum_{j=1}^{n} \alpha_j + (2 - n)\pi \quad (28)$$

where α_j is one of the interior angles of the spherical polygon.

However, due to the local way in which the interior and exterior of the spherical polygons were defined and the possibility of having one spherical polygon imbedded in a second spherical polygon, it is possible that the orientation of a spherical polygon might need to be reversed to yield the correct result. Noting that the solid angle buried by the inside and outside of a single spherical polygon must always sum to 4π, we can correct for wrong orientations and imbedded spherical polygons by summing all interior angles modulo 4π. Thus, the solid angle buried by the spherical polygons, $\omega_{\text{SP}}^{(i)}$, can be determined by

$$\omega_{\text{SP}}^{(i)} = \left(\sum \Phi\left(Q_j^{(i)}, P_{jk}^{(i)}, Q_k^{(i)}\right) + \sum \theta_{Q_j^{(i)}}^{(i)} + (2n_{\text{SP}} - n_{\text{A}})\pi\right)(\text{mod}\, 4\pi) \quad (29)$$

where all Φ angles are the interior angles at vertices of the spherical polygon(s), n_{SP} is the number of separate spherical polygons, and n_{A} is the total number of sides for all the spherical polygons. Thus the contributions of all types of buried surface angle of sphere i have been calculated, and the final value of Ω_i is calculated as follows:

$$\Omega_i = 4\pi - \omega_{\text{SC}}^{(i)} - \omega_{\text{SS}}^{(i)} - \omega_{\text{SP}}^{(i)} \quad (30)$$

RADIAL QUADRATURES IN THE DIELECTRIC SCREENING ALGORITHM

The most computationally expensive step in the SM*x* algorithm is the calculation of the electrostatic free energy of a sphere of radius ρ_k at the origin embedded in an infinite continuum dielectric medium of relative permittivity ε but shielded from this medium by a set of $(N - 1)$ nearby spheres, centered at points $\mathbf{x}_{k'}$ and having radii $\rho_{k'}$ and relative permittivities of 1, where N is the number of atoms in the solute. Some of the nearby spheres may partially overlap the sphere at the origin or one another. The critical step in this calculation is the evaluation of the integral

$$\frac{1}{\alpha_k} = \int_{\rho_k}^{\infty} \frac{dr}{r^2} \frac{A(r, \{\mathbf{x}_{k'}, \rho_{k'}\})}{4\pi r^2} \quad (31)$$

where $A(r, \{\mathbf{x}_{k'}, \rho_{k'}\})$ is the area of a sphere at the origin of radius r in the presence of $(N - 1)$ spheres characterized by the $(6N - 2)$ parameters in the set $\{\mathbf{x}_{k'}, \rho_{k'}\}$. The exposed area is the area not contained in any of the other spheres. Let $a(r)$ be the fractional exposed area of a sphere of radius r, that is

$$a(r) \equiv A(r, \{\mathbf{x}_{k'}, \rho_{k'}\})/4\pi r^2 \quad (32)$$

Because the centers of the N spheres are at the locations of the nuclei of the solute, when r gets large enough, all $(N - 1)$ other spheres become engulfed by the sphere at the origin, and $a(r)$ becomes unity. If the distance at which this occurs is R, the integral may be written

$$\frac{1}{\alpha_k} = \int_{\rho_k}^{R'} dr \frac{a(r)}{r^2} + \frac{1}{R'} \quad (33)$$

where $R' \geq R$. The computational expense is due to the repetitive evaluation of exposed surface areas at the quadrature nodes of the radial integration.

In the original algorithm,[10,15] to be called the force rectangle algorithm, an interval $[\rho_k, R']$, where $R' \geq R$, is divided into segments with centers at r_i corresponding to spherical shells with thickness T_i, where

$$\rho_k = r_1 - 0.5T_1 \quad (34)$$

and

$$R' = r_M + 0.5T_M \quad (35)$$

The thicknesses form a geometric progression defined by

$$T_i = T_1(1 + F)^{i-1} \quad (36)$$

where T_1 and F are parameters of the method. The upper limit R' was taken as the first value of $r_i + 0.5T_i$ that is greater than or equal to R, and thus it depends on molecular geometry, as does R. The contribution of each segment to the integral was approximated by treating $a(r)$ as approximately constant over the interval, which yields

$$\int_{r_i-0.5T_i}^{r_i+0.5T_i} dr\frac{a(r)}{r^2} = a(r_i)\left(\frac{1}{r_i-0.5T_i} - \frac{1}{r_i+0.5T_i}\right) \quad (37)$$

The values finally adopted for T_1 were 0.1 Å for SM1 and SM1a and 0.05 Å for SM2 and SM3, and F was fixed at 0.5 in all methods. This leads to small but significant systematic errors in the integrals, and these were explicitly[15,17,18] absorbed in the parameterization.

In the new scheme proposed here, to be called the force trapezoid algorithm, $a(r)$ is approximated by $c_0^{(i)} + c_1^{(i)}r$ over the domain of interval i. As before, M intervals are determined by preselected values of T_1 and F, but we would like to insure the equality $R' = R$ rather than an inequality, as before. To accomplish this, we must first define the precise number of intervals necessary to engulf the molecule. M_0. An initial estimate M_0 of M is calculated as follows:

$$M_0 = \frac{\ln\left(\frac{FR}{T_1^{(\text{in})}} + 1\right)}{\ln(F+1)} \quad (38)$$

where $T_1^{(\text{in})}$ represents the preselected value of the first interval width. Because M_0 is most often a noninteger, we need to adjust T_1 so that the final interval ends exactly when the entire molecule is engulfed. The final value of T_1 obtained this way is called $T_1^{(\text{adj})}$, and it is given by

$$T_1^{(\text{adj})} = \frac{M_0}{M} T_1^{(\text{in})} \quad (39)$$

where

$$M = \text{INT}[M_0 + 1] \quad (40)$$

and INT[x] denotes the integer part of x. It is convenient to write the new algorithm in terms of the boundaries $[b_{i-1}, b_i]$ of segment i rather than its center r_i. Thus

$$b_0 = \rho_k \quad (41)$$

and

$$b_i = \rho_k + T_1^{(\text{adj})}(1+F)^{i-1}, \quad i = 1, \ldots, M \quad (42)$$

A little algebra yields

$$c_0^{(i)} = \frac{a_{i-1}b_i - a_ib_{i-1}}{b_i - b_{i-1}} \quad (43)$$

and

$$c_1^{(i)} = \frac{a_i - a_{i-1}}{b_i - b_{i-1}} \quad (44)$$

where

$$a_i \equiv a(b_i) \quad (45)$$

This yields the following approximation to eq. (33):

$$\frac{1}{\alpha_k} = \frac{a_0}{\rho_k} + \sum_{i=1}^{M-1} \frac{a_i - a_{i-1}}{b_i - b_{i-1}}(\ln b_i - \ln b_{i-1}) \quad (46)$$

Notice that a_M equals 1 and does not need to be evaluated numerically. Also, as a consequence of this fact, one of the terms from shell M has canceled the final term of eq. (33) because $R' = b_M = R$. This exact cancellation of the bulk term makes the algorithm less susceptible to numerical noise.

It is important to note that during a geometry optimization new values of R, and hence of $T_1^{(\text{adj})}$, are calculated each time the geometry changes. As a consequence, the value of M could vary during the course of a geometry optimization. In certain cases, the M values would oscillate, causing problems with the SCF convergers because the number of intervals varies in the radial quadrature. To avoid this problem, for every atom in a given molecule, the initially calculated M value is stored. If subsequent geometry changes cause the value of M yielded by eq. (40) to change, this change is accepted only if the new M value is larger than the largest previous M value for a particular atom. This allows the number of intervals in the radial quadrature to increase, but never decrease, during the course of a run, thus preventing the oscillatory behavior and avoiding certain SCF convergence problems. Notice that in the new algorithm, as in the previous one, M denotes the number of $a(r)$ values to be numerically evaluated, including a_0.

The algorithm just described is the default. There are cases, though, for which it is desirable to use a slightly different strategy, in particular to fix F and M instead of F and T_1. The cases for which this is useful are calculations wherein it is important for the calculated energy to be a function of geometry alone, not of initial guess. This is important (1) when using the CHAIN optimizer, for which—to discover a crossing of states as a function of a reaction coordinate—the same value must be found when a geometry is calculated twice with different starting guesses; (2) when it is desired that an energy profile along a reaction coordinate be identical when computed in either direction; and (3) when computing a force constant matrix, so that M does not change for two calculations at nearby geometries that will be finite differenced. There is an option in AMSOL version 4.1 by which the user can select fixed M in these or any other situations.

UPDATING THE FOCK MATRIX

Algorithms for converging self-consistent-field calculations generally make implicit use of the linearity of the Fock operator with respect to the density matrix. This property does not hold for SM*x* schemes because of the nature of the dependence of the diagonal of the Fock matrix on the density matrix. The solvation updates may lead to convergence difficulties in the SCF step.

To damp these oscillations efficiently, the solvation terms in the Fock matrix are frozen at some SCF steps and only allowed to relax gradually with the addition of scheduled updates to the Fock matrix. This methodology is implemented in the dynamic level shift technique,[33,34] Pulay's method,[35] and to some extent in the three-point interpolation of the density matrix method; all three SCF convergers are available in AMSOL.[23]

The important issue to resolve is how the freezing and relaxing of solvation terms in the Fock matrix should be coupled to changes in energy during the course of the SCF cycles. In the original implementation in AMSOL, during a geometry optimization the Fock matrix is updated with Born information derived from the latest atomic charges at iterations 1, 4, 9, 16, etc. This rigid implementation occasionally causes the oscillations to be overly damped and can by its abrupt nature prevent SCF convergence.

For the solvation energy to better approximate a continuous function of the density matrix, the solvation terms in the Fock matrix should be updated more often. The updates should take place at almost every SCF cycle, but using damped charges that weakly reflect trends occurring in the density matrix, rather than the full charges. A new SCF convergence strategy was implemented in AMSOL starting in version 4.0 to incorporate this scheme. To determine the extent to which the charges should be damped, we created a damping factor that ranges from 1, corresponding to no update, to 0, meaning that the charges do not need to be damped when updating the solvation terms. The damping factor is determined as a function of the mean oscillations in the energy and the density matrix. When the oscillations are extreme, the solvation terms are frozen and the dynamic level shift method is used in the damped mode. When the SCF is nearly converged, the solvation terms are updated from undamped charges at every cycle, and by using Pulay's method we can expect nearly quadratic convergence. This procedure is usually efficient, and it can often shorten the number of SCF cycles by a factor of 2.

Performance

ASA ALGORITHM

The code for implementing the ASA algorithm was originally developed and implemented on an IBM 3090 computer with 8-byte floating-point numbers. The tolerance factors used for recognizing sphere intersections and the occurrence of junction points were then calibrated so that the algorithm provided an accuracy of 12 significant figures. To test for flaws in the algorithm, the results obtained by implementing the method described in the section titled "Analytic Exposed Surface Areas" were compared with results obtained from a numerical approach using 10,000 points on the surface of each sphere. There was no significant discrepancy for more than 1 million random tests.

The code was then compared to the numerical approach, DOTS, used in AMSOL—versions 1 through 3.0.1—and the improved DOTS algorithms used in AMSOL—versions 3.5 to 4.0—and AMSOL version 4.1. The DOTS algorithm places n evenly spaced points around an equatorial great circle of the sphere for which the exposed surface area is to be calculated. The sphere is then divided by $n - 1$ latitudinal circles that are evenly spaced in angle from $-\pi/2$ to $\pi/2$ radians. DOTS determines how many points should lie on each of these latitudinal circles by multiplying the number, n, of points on the great circle by the cosine of the angle between the great circle and the latitudinal circle. The number of points on each sphere is then equal to the sum of the points on all the circles. To calculate A_i, we count the number of points that are exposed on the surface of the group of spheres. For each point $D_m^{(i)}$, where m is the index of one of the points on sphere i, we determine whether $D_m^{(i)}$ lies within the volume of a sphere j. If

$$\left(J_x - (D_m^{(i)})_x\right)^2 + \left(J_y - (D_m^{(i)})_y\right)^2 + \left(J_z - (D_m^{(i)})_z\right)^2 < R_j^2 \quad (47)$$

then $D_m^{(i)}$ is buried in the volume of sphere j and is not exposed on the surface of the group of spheres. If N is the number of points originally placed on the surface of sphere i and N_{exp} is the number of points of i that are exposed on the surface of the group of spheres, then A_i can be calculated using

$$A_i = 4\pi R_i^2\left(\frac{N_{\text{exp}}}{N}\right) \quad (48)$$

The improved DOTS algorithm of AMSOL version 3.5 is more efficient. Rather than placing points on the surface of each sphere and calculating their coordinates in a Cartesian coordinate system, a unit sphere and vectors to the chosen number of evenly spaced points on its surface are created. The unit sphere is created in a Cartesian coordinate system with its center at the origin. The sphere is then divided into hemispheres by the xy, yz, and xz planes. Four other planes are also used to divide the unit sphere into hemispheres; they are the planes that pass through $[(1, 1, 1), (-1, 0, 1), (-1, -1, -1)]$, $[(-1, 1, -1), (-1, 0, 1), (1, -1, 1)]$, $[(1, 1, -1), (1, 0, 1), (-1, -1, 1)]$, and $[(-1, 1, 1), (1, 0, 1), (1, -1, -1)]$. For each of the 14 hemispheres, a list of the points on the half sphere are stored. In addition, the unit vectors that originate at the center of the unit sphere, point to the surface of a specific hemisphere, and are perpendicular to the plane creating the same hemisphere are stored. Henceforth, these vectors will be referred to as the hemisphere vectors. For each sphere, i, for which the exposed surface area is to be calculated, the points on the surface of i are created by scaling the radius of the unit sphere to the radius of i and moving the center of the unit sphere to I. Then the vectors which point from the center of i to the center of every sphere that intersects i are created. Using these vectors, the hemisphere vectors, and eq. (6), we then determine whether the intersection between i and some sphere j lies completely within one of the half spheres or completely buries one of the half spheres. If a hemisphere of i is buried by j, none of the points on that hemisphere are exposed on the surface of the set of spheres for which the calculation is made, and thus the distance calculations between those points and all other spheres in the set need never be made. If the intersection between sphere i and j lies completely within a half sphere, then no points on the disjoint hemisphere of i can be buried within j. For any point $D_m^{(i)}$ on a sphere i that has not been characterized as being buried or exposed, if

$$d_{IJ}^2 + R_i^2 - 2d_{IJ}R_i\left(\hat{\mathbf{r}}_{ID_m^{(i)}} \cdot \hat{\mathbf{r}}_{IJ}\right) < R_j^2 \quad (49)$$

then $D_m^{(i)}$ is buried in the volume of sphere j and not exposed on the surface of set of spheres for which the calculation is made. After characterizing each point on the surface of i as being buried or exposed, the exposed surface area of i is then calculated with eq. 48).

The final improvement to the version 3.5 DOTS algorithm anticipates the way the algorithm is used within the AMSOL package in versions 3.5 and later. In the radial integration for the Coulombic portion of the solvation energy calculation, a given sphere, i, expands until it engulfs the entire set of spheres. Because the number of points on a sphere remains constant throughout the entire radial integration, once the solid angle buried by a sphere j begins to decrease, the points on the surface of i that could possibly be buried by sphere j are a subset of the previously eclipsed points. Thus all other points on i are not buried by sphere j and need not be checked. These changes taken together reduce the number of times eq. (49) must be calculated by over a factor of 2.

The additional improvement in the DOTS algorithm in version 4.1 is simply to remove unnecessary calculations of the exposed surface areas at r values where they are not used. Distributed versions of AMSOL prior to version 4.1 involved twice the number of A_i calculations for the DOTS algorithm than are actually required because the exposed area was unnecessarily calculated at intermediate points. This is corrected in version 4.1. All DOTS and ASA timings and average M values in this article are explicitly referenced to a particular version, 3.0.1, 4.0, or 4.1, of AMSOL to make clear which of these improvements are involved.

In the first round of timing tests, on the IBM 3090, we calculated A_i values for a test set of 17 molecules (including planar, nonplanar, globular, and elongated shapes) with 7 to 51 atoms, and the radii of the spheres were varied from small to large with respect to typical van der Waals values. In contrast to the previously reported comparison,[28] the analytical approach was invariably faster than the DOTS algorithm of AMSOL version 3.0.1. The mean speedup was a factor of 17, with a significant dispersion (from 4 to 91).

Further tests were run on a superscalar IBM RS/6000 model 550 workstation and a vector Cray Y-MP C916/8256 supercomputer using the 15 neutral molecules in the AMSOL test suite.[23] In these tests, we made full SM*x* calculations, including both SCF and geometry optimizations.

In addition, the accuracy was again tested using the GEPOL algorithm's method[20–22] of symmetrically placing points on the surface of each sphere. The GEPOL code was used only to position the points on the surface of the unit sphere, and then portions of the extended DOTS algorithm were used to determine which points were exposed on the surface of the group of spheres. These tests were conducted using 15,360 points on each sphere.

The present analytical algorithm was always faster than the original (version 3.0.1) numerical DOTS algorithm using the default number of points on each sphere, 816 points for the Coulombic calculation and 2610 points for the surface tension calculation. The geometric mean speedup factor was 5.1 on the vector Cray machine and 22 on the superscalar IBM machine. On the superscalar machine, the ASA algorithm was faster than the improved DOTS algorithm, and on the vector machine the ASA algorithm and the improved DOTS algorithm ran at comparable speeds, although the ASA algorithm provided a more accurate result. The difference in speedup factors between the vector and superscalar machine is due to the fact that the ASA algorithm is not efficiently vectorizable. (See Table I for further speed data.)

There are several factors which make the implementation of the ASA algorithm more efficient. Whenever possible, the inverse trigonometric functions were avoided. Rather than immediately calculating the γ and Φ angles of eqs. (6), (7), and (9) through (11), the cosines of these angles were used

TABLE I.
Performance of the ASA Algorithm versus the Numerical Surface Area Methods in AMSOL.

Algorithm	Number of points for the Coulombic calculation	Number of points for the surface tension calculation	Machine[a]	Numerical precision[b] (kcal / mol)	Speed[c]
DOTS 3.0.1	816	2610	Cray	0.051	1.0
DOTS 4.0	816	2610	Cray	0.053	4.9
GEPOL 4.0	960	960	Cray	0.055	4.3
GEPOL 4.0	15360	15360	Cray	0.027	0.78
ASA 4.0	—	—	Cray	0.0	4.2
DOTS 4.1	816	2610	Cray	0.051	4.8
GEPOL 4.1	960	960	Cray	0.045	4.8
ASA 4.1	—	—	Cray	0.0	5.1
DOTS 3.0.1	816	2610	IBM	0.085	1.0
DOTS 4.0	816	2610	IBM	0.046	4.6
GEPOL 4.0	960	960	IBM	0.050	4.1
GEPOL 4.0	15360	15360	IBM	0.027	0.32
ASA 4.0	—	—	IBM	0.0	20
ASA 4.1	—	—	IBM	0.0	22

[a] In the machine column, the word Cray represents trials that were run on the Cray Y-MP C916 / 8256 (a vector machine in the C90 series), and IBM represents trials run on the IBM RS / 6000 Model 550 (a superscalar machine).

[b] All numerical precision values in the tables are referenced to the ASA algorithm running on the same machine. For rows corresponding to versions 3.0.1 -4.0 of AMSOL, the reference is the ASA calculation run with version 4.0. For the DOTS 4.1 and GEPOL 4.1 rows, the reference is the ASA calculation with version 4.1. The numerical precision values are given in kcal / mol and represent the average of the absolute value of the difference between the reference result for the total free energy of solvation and the result obtained with the algorithm specified in column 1 over the 15 neutral molecules of the AMSOL test suite. This test set includes a variety of SM1, SM1a, SM2, and SM3 calculations.

[c] All speeds are referenced to the DOTS algorithm in AMSOL—version 3.0.1 running on the same machine. They represent the geometric mean of the speeds obtained for each of the 15 neutral molecules of the AMSOL test suite. The speed obtained for a certain trial is the time the trial took when run with the default settings of the DOTS algorithm in AMSOL—version 3.0.1 divided by the time the trial took when running with the algorithm specified in column 1.

exclusively until after the list of free points, M_i, was constructed. Typically, many of the original spheres on list L_i and the points on M_i have been eliminated by this point, and many of these angles never need to be calculated. In addition, we make intensive use of previously calculated quantities. The determination of whether numerical instabilities may occur in the calculation due to the existence of junction points is made as early as possible. By finding any junction points and consequently restarting the algorithm prior to finding the free points, fruitless calculations can be limited.

As mentioned earlier, our algorithm is able to avoid the numerical instability that has been observed in other analytical approaches[27,28] from a point $P_{jk}^{(i)}$ being a junction point. Our algorithm for detecting junction points and, if one is encountered, adding a small increment to the radius of the principal sphere and restarting the calculation is effective and efficient. Even when the occurrence of junction points is relatively common, such as in highly symmetrical molecules like benzene, the code is still faster than numerical methods. The error introduced by this trick costs no more than one significant figure in the accuracy of the exposed surface area.

EFFICIENT QUADRATURE

To determine reasonable values for F and $T_1^{(\text{in})}$ in the effective Coulombic radial integration, we made several sets of trials. In these trials we monitored the average M value, $\langle M \rangle$, where the average is over the various atoms in a given solute. A set of 46 molecules, 24 ions and 22 neutrals, was chosen to model the functional groups of interest. These solutes are listed in Tables II and III. For selected trials, the old and new quadratures were converged. This was accomplished by reducing the

TABLE II.
Results for Ions.

		ΔG (kcal / mol)			
		SM2.1			
Molecule	⟨M⟩	ENP	CDS	Total	Exp.
OH^-	5.0	−102.8	−8.2	−111.0	−106
CN^-	6.0	−80.6	−2.6	−83.1	−77
O_2^-	6.0	−83.9	−3.8	−87.7	−87
HS^-	6.0	−74.2	−1.9	−76.0	−76
PH_2^-	6.7	−67.3	0.6	−66.7	−67
HC^-	7.3	−79.0	0.8	−78.3	−73
HO_2^-	7.0	−90.2	−7.0	−97.2	−101
N_3^-	7.3	−68.6	−5.6	−74.2	−74
NO_2^-	7.3	−72.3	−4.7	−76.9	−72
CH_3O^-	7.0	−81.7	−2.2	−83.9	−95
NO_3^-	7.5	−53.2	−5.0	−58.1	−65
CH_2CN^-	9.2	−67.9	−2.3	−70.2	−75
$CH_3CO_2^-$	9.1	−71.2	−3.0	−74.2	−77
ICH_3Br^- (TS)[a]	13.2	−65.1	−2.0	−67.1	—
H_3O^+	6.5	−101.4	−3.8	−105.2	−104
NH_4^+	6.6	−79.9	−1.8	−81.7	−79
$MeOH_2^+$	8.9	−78.5	−5.2	−83.7	−83
$MeSH_2^+$	9.4	−76.4	0.1	−76.4	−74
$MeNH_3^+$	9.1	−70.0	−3.2	−73.2	−70
Me_2OH^+	10.0	−61.5	−1.8	−63.2	−70
$Me_2NH_2^+$	10.0	−61.5	−1.9	−63.4	−63
Me_3NH^+	10.3	−54.3	0.3	−54.0	−59
Me_3PH^+	10.6	−58.0	1.5	−56.4	−53
$CH_3C(OH)NH_2^+$	10.2	−70.9	−5.8	−75.6	−66
Mean unsigned error				3.8	
rms error				4.8	

[a] TS denotes transition state.

TABLE III.
Results for Neutrals.

		ΔG (kcal / mol)			
		SM2.1			
Molecule	M	ENP	CDS	Total	Exp.
Methylcyclohexane	11.6	0.1	1.9	2.0	1.7
cis-1,2-Dimethylcyclohexane	11.7	0.2	2.0	2.2	1.6
m-Xylene	11.9	−3.0	0.6	−2.4	−0.8
Anthracene	13.8	−4.7	2.2	−2.4	−4.2
N-Methylpiperazine	11.5	−5.0	−1.6	−6.5	−7.8
Diethyl ether	11.6	−1.0	0.7	−0.3	−1.6
Methyl butanoate	12.4	−1.5	−0.9	−2.4	−2.8
Acetophenone	12.4	−4.2	−0.1	−4.3	−4.6
Phosphine	7.5	0.0	0.6	0.6	0.6
Ethanethiol	10.3	−0.3	−0.3	−0.6	−1.3
1,1,2-Trichloroethane	10.0	−1.2	−0.2	−1.4	−2.0
Z-Dichloroethylene	9.7	−1.2	0.1	−1.1	−1.2
2-Methoxyethanamine	11.5	−2.3	−4.7	−7.0	−6.6
2-Methyl-1-nitrobenzene	11.8	−3.1	−1.6	−4.7	−3.6
1,1,1-Trifluoropropan-2-ol	10.3	−2.9	−0.4	−3.3	−4.2
3-Ethyl-2-methoxypyrazine	12.6	−4.5	−1.4	−5.9	−4.4
Morpholine	10.9	−3.1	−2.6	−5.7	−7.2
Water	6.7	−1.3	−5.1	−6.5	−6.3
Ammonia	6.5	−0.4	−3.9	−4.3	−4.3
4-Pyridone	11.1	−13.1	−3.4	−16.5	—
Nitrogen oxide	5.5	−0.1	−4.3	−4.5	—
Diels-Alder TS[a]	14.4	−4.4	−2.5	−6.9	—
Mean unsigned error				0.8	
rms error				0.9	

[a] A transition state of the Diels-Alder reaction between cyclopentadiene and methyl vinyl ketone.

values of F and $T_1^{(\mathrm{in})}$, hence increasing the average M values, for each quadrature until any further such reductions did not affect the calculated polarization free energy. This took on the order of 1000 intervals for the new quadrature and 10,000 intervals for the original scheme. In each of the selected trials, the converged integral for the new quadrature matched the converged integral for the original quadrature. We then constructed a plot of various choices of F and $T_1^{(\mathrm{in})}$ with the new quadrature (see Figs. 6–13). The best compromise of accuracy and efficiency was chosen to be $T_1^{(\mathrm{in})} = 0.15$ Å and $F = 0.20$.

Reparameterization

Because the new Coulombic radial integration scheme affords a more accurate calculation of the effective Coulombic radius, we refit the SM2 parameters so that using the force trapezoid integration scheme provides results that coincide reasonably closely with the results from the same models parameterized with the force rectangle integration scheme. The new parameterization will henceforth be called AM1-SM2.1.

The free energy of solvation of a given solute in the SM*x* schemes is given by

$$\Delta G_S^0 = \Delta G_{\mathrm{ENP}} + G_{\mathrm{CDS}}^0 \qquad (50)$$

where

$$\Delta G_{\mathrm{ENP}} = \Delta E_{\mathrm{EN}} + G_P \qquad (51)$$

ΔE_{EN} is the solute electronic-nuclear internal reorganization energy. G_P is the free energy associated with electric polarization of the solvent, and G_{CDS}^0 is the cavity-dispersion-structural term. G_P is associated with GB terms and the dielectric screening algorithm, and G_{CDS}^0 is associated with the atomic surface tensions. Because the errors caused by the unconverged rectangular algorithm are systematic[15] (i.e., they cause the calculated polariza-

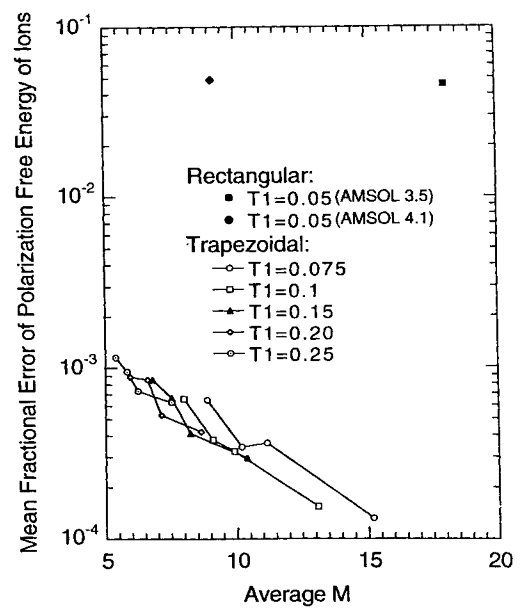

FIGURE 6. The mean fractional error of the calculated polarization free energy for ions versus average *M* for several values of T_1 and *F*. The points on each of the T_1 lines are, left to right, $F = 0.33$, 0.25, 0.20, and 0.10.

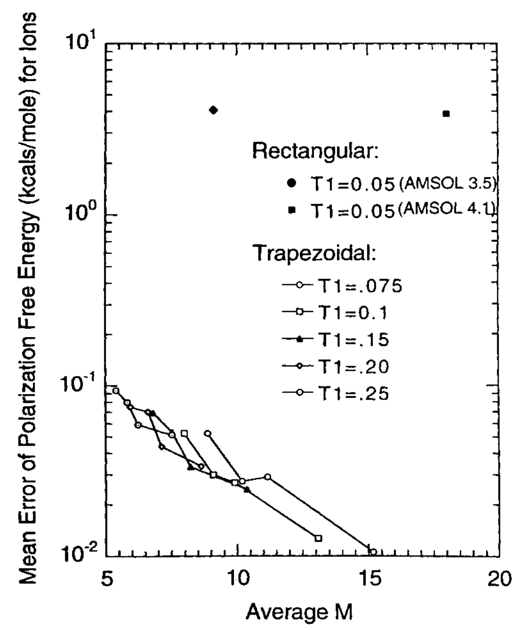

FIGURE 7. The mean error of the calculated polarization free energy in kcal / mol for 24 ions versus average *M* for several values of T_1 and *F*. The points on each of the T_1 lines are, left to right, $F = 0.33$, 0.25, 0.20, and 0.10.

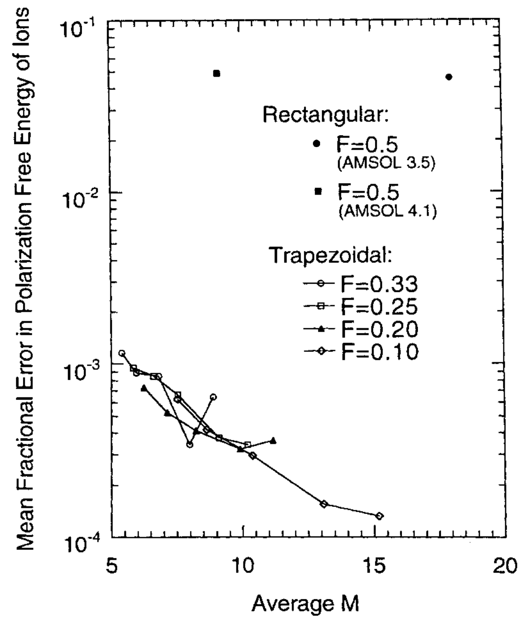

FIGURE 8. The mean fractional error of the calculated polarization free energy for 24 ions versus average *M* for several values of *F* and T_1. The points on each of the *F* lines are, left to right, $T_1 = 0.075$, 0.1, 0.15, 0.20, and 0.25.

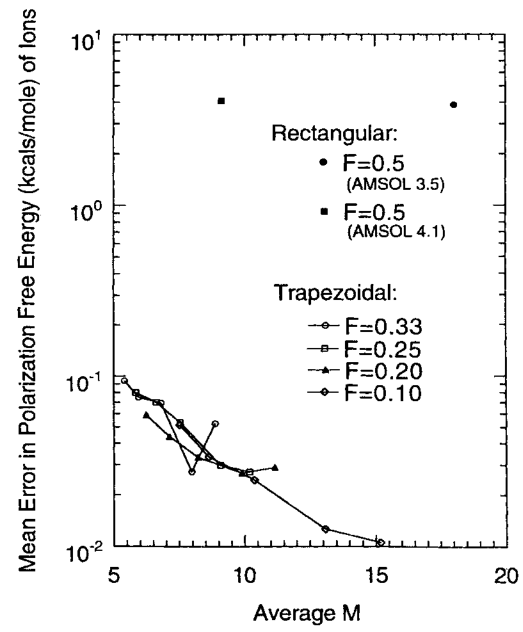

FIGURE 9. The mean error of the calculated polarization free energy in kcal / mol for 24 ions versus average *M* for several values of *F* and T_1. The points on each of the *F* lines are, left to right, $T_1 = 0.075$, 0.1, 0.15, 0.20, and 0.25.

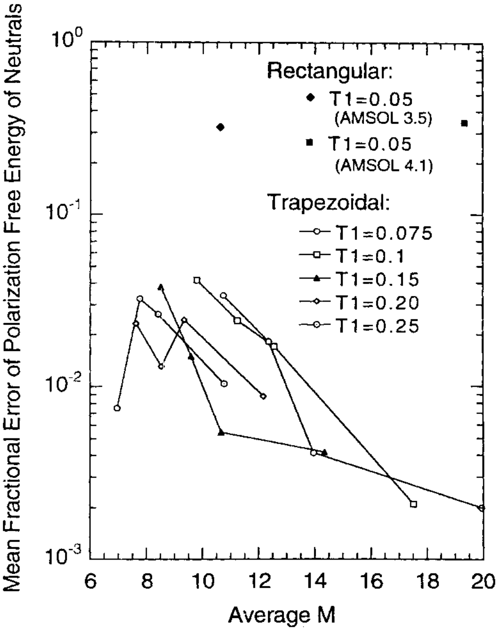

FIGURE 10. The mean fractional error of the calculated polarization free energy for 22 neutrals versus average M for several values of T_1 and F. The points on each of the T_1 lines are, left to right, $F = 0.33$, 0.25, 0.20, and 0.10.

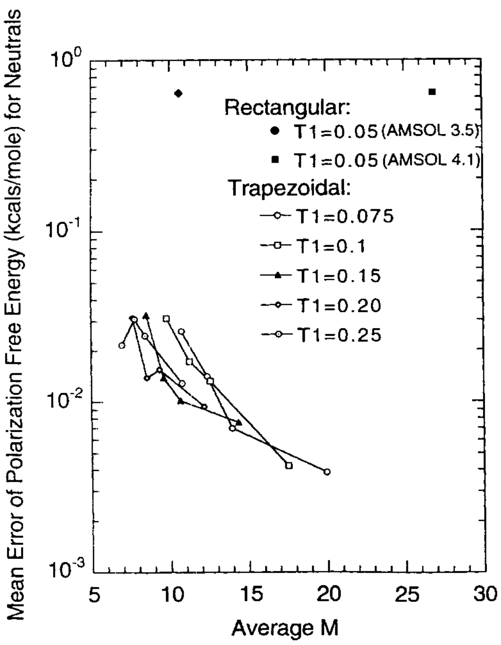

FIGURE 11. The mean error of the calculated polarization free energy in kcal / mol for 22 neutrals versus average M for several values of T_1 and F. The points on each of the T_1 lines are, left to right, $F = 0.33$, 0.25, 0.20, and 0.10.

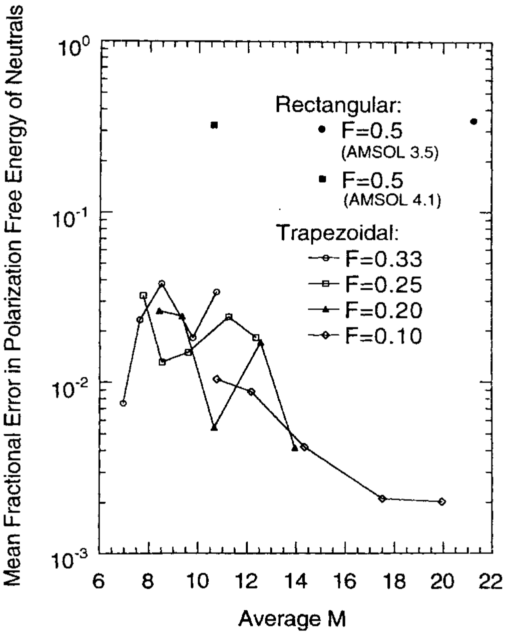

FIGURE 12. The mean fractional error of the calculated polarization free energy for 22 neutrals versus average M for several values of F and T_1. The points on each of the F lines are, left to right, $T_1 = 0.075$, 0.1, 0.15, 0.20, and 0.25.

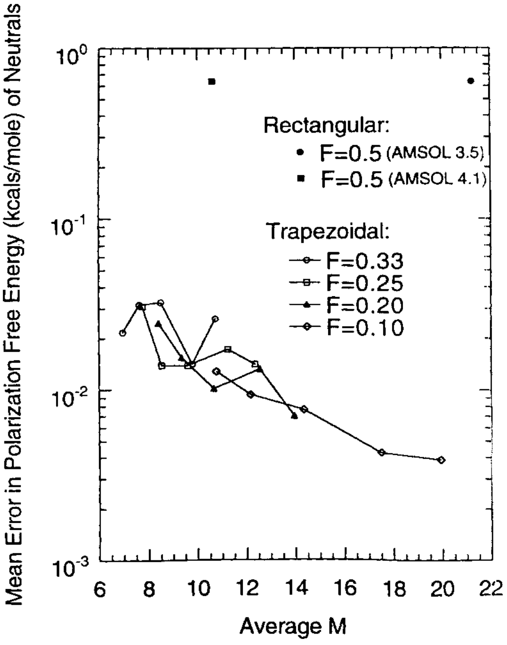

FIGURE 13. The mean error of the calculated polarization free energy in kcal / mol for 22 neutrals versus average M for several values of F and T_1. The points on each of the F lines are, left to right, $T_1 = 0.075$, 0.1, 0.15, 0.20, and 0.25.

tion free energy to err in the same direction almost all of the time), we found a single scale factor S for two of the empirical parameters, $\rho_k^{(0)}$ and $\rho_k^{(1)}$, that together determine the intrinsic Coulombic radius, ρ_k, by the prescription

$$\rho_k = \rho_k^{(0)} + \rho_k^{(1)}\left[-\frac{1}{\pi}\arctan\frac{q_k + q_k^{(0)}}{q_k^{(1)}} + \frac{1}{2}\right] \quad (52)$$

where $q_k^{(0)}$ and $q_k^{(1)}$ are parameters. The value of S was chosen to minimize the difference between polarization free energies, G_P, obtained with the AM1-SM2 model using the SM2 rectangular quadrature and using the trapezoidal converged quadrature for the 24 nonmonatomic ions listed in Table II. (We used geometries optimized in solution.) These parameters were not scaled for the halogens or for hydrogen because the sum of $\rho_k^{(0)}$ and $\rho_k^{(1)}$ for those atoms was originally determined using the monatomic anions and thus did not incorporate any systematic error of the original quadrature into their optimization. Further, for all atom types, we did not alter the ratio of $\rho_k^{(0)}$ to $\rho_k^{(1)}$, nor did we alter the parameters $q_k^{(0)}$ or $q_k^{(1)}$ of eq. (52). When the scaling factor was set to 1.000, there was a mean fractional deviation of 0.07 in the polarization free energies between the two quadrature schemes. The best fit of the polarization free energies calculated with the force trapezoid quadrature for the ions in Table II to the polarization free energies calculated with the force rectangle quadrature was obtained using a scaling factor of $S = 1.07$, which gave a mean fractional deviation of 0.03 for the set.

We then adjusted the interfacial atomic surface tension parameters,[15] $\sigma_k^{(0)}$ and $\sigma_k^{(1)}$, to fit the experimental free energies of solvation for phosphine and the remaining set of 147 non-phosphorus-containing neutrals listed in Table V of our SM2-SM3 overview article.[15] In both fits, the squared error for octane was given a weight of 64 whereas all other compounds were given a weight of 1. The final parameters for SM2.1 were obtained by rounding to a reasonable number of significant figures, and they are listed in Table IV of this article. Table V illustrates the mean deviations from experiment for the SM2 and SM2.1 models.

TABLE IV.
Optimized Parameters for SM2.1 by Atom Type.

Atom	$\rho_k^{(0)}$ Å	$\rho_k^{(1)}$ Å	$\sigma_k^{(0)}$ cal mol^{-1} Å^{-2}	$\sigma_k^{(1)}$ cal mol^{-1} Å^{-2}
H	0.590	1.283	0	0
C	1.798	0	3.35	3.44
N	1.605	0.449	−30.14	−18.09
O	1.562	−0.161	−24.97	−37.17
F	1.370	0.145	26.54	0
P	1.391	1.070	3.97	0
S	1.391	0.856	−52.53	36.42
Cl	1.650	0.555	−1.72	0
Br	1.750	0.629	−7.26	0
I	1.880	0.885	−14.53	0

TABLE V.
Mean Deviations of Solvation Free Energies (kcal / mol) from Experiment.[a]

	Ions	Neutrals
Mean unsigned		
SM2	2.67	0.64
SM2.1	3.15	0.61
Root mean square		
SM2	3.96	0.86
SM2.1	4.36	0.82

[a] All based on AMSOL — version 4.1.

Note Added in Proof. A version of AMSOL containing the improved methods described here is available from the Quantum Chemistry Program Exchange. The reference for the latest version (4.5) of this computer code is C. J. Cramer, G. D. Hawkins, G. C. Lynch, D. J. Giesen, D. G. Truhlar, and D. A. Liotard, *QCPE Bull.*, **14**, 56 (1994). Additional features beyond those presented here are described in J. W. Storer, D. J. Giesen, C. J. Cramer, and D. G. Truhlar, *J. Comp.-Aided Mol. Design*, in press.

Second Note Added in Proof. The PM3-SM3 parameters have now been refit to utilize the force trapezoid integration scheme and provide results that coincide reasonably closely with the results from the same model parameterized with the force rectangle integration scheme. The new parameterization will henceforth be called PM3-SM3.1.

The method followed to arrive at these new parameters is identical to the method used to obtain the AM1-SM2.1 parameters discussed earlier. A final scaling factor of $S = 1.07$ was obtained for the PM3-SM3.1 method which yielded a mean fractional deviation in the polarization free energies between the PM3-SM3.1 and PM3-SM3 methods of 0.08 for the same 148 neutrals used to parameterize AM1-SM2.1. The final parameters for PM3-SM3.1 are listed in Table VI of this article, and a summary of the mean deviations from experiment is in Table VII. The PM3-SM3.1 method is available in AMSOL-version 4.6.

TABLE VI.
Optimized Parameters for SM3.1 by Atom Type.

Atom	$\rho_k^{(0)}$ Å	$\rho_k^{(1)}$ Å	$\sigma_k^{(0)}$ cal mol^{-1} $Å^{-2}$	$\sigma_k^{(1)}$ cal mol^{-1} $Å^{-2}$
H	0.590	1.289	0	0
C	1.691	0.214	4.45	2.70
N	1.712	0.342	−43.67	−25.36
O	1.776	−0.214	−36.43	−20.78
F	1.370	0.149	22.11	0
P	1.498	0.856	5.39	0
S	1.284	0.963	−53.57	35.42
Cl	1.650	0.559	−2.67	0
Br	1.750	0.610	−4.76	0
I	1.880	0.798	0.41	0

TABLE VII.
Mean Deviations of Solvation Free Energies (kcal / mol) from Experiment.[a]

	Ions	Neutrals
Mean unsigned deviation		
SM3	3.5	0.95
SM3.1	3.4	0.90
Root mean square deviation		
SM3	5.6	1.26
SM3.1	5.4	1.25

[a] All based on AMSOL—version 4.6.

Summary

We have presented and implemented a new analytical algorithm for the calculation of exposed surface areas of overlapping spheres. The algorithm is made efficient by a topological analysis of overlapping spherical caps that is more efficient than previous analyses and is combined with an accurate and efficient trick for avoiding floating-point round-off near a junction point. We also presented a more accurate radial integration algorithm for dielectric screening in generalized Born calculations. Finally, we presented new solvation parameters for use with the new radial integration scheme.

Acknowledgments

The authors are grateful to Joey W. Storer for assistance with the calculations. This work was supported in part by grants form the Computer Center for Intensive Calculations, CNUSC, Montpellier, France and the National Science Foundation of the United States and by a resource allocation from the Minnesota Supercomputer Institute.

References

1. C. J. Cramer and D. G. Truhlar, *Rev. Comp. Chem.*, **6**, in press.
2. G. J. Hoijtink, E. d. Boer, P. H. v. d. Meij, and W. P. Weijland, *Recl. Trav. Chim. Pays-Bas*, **75**, 487 (1956).
3. F. Peradejordi, *Cah. Phys.*, **17**, 393 (1963).
4. I. Jano, *Compt. Rend. Acad. Sci. (Paris)*, **261**, 103 (1965).
5. O. Tapia, In *Quantum Theory of Chemical Reactions*, vol. 2, R. Daudel, A. Pullman, L. Salem, and A. Veillard, Eds., Reidel, Dordrecht, 1980, p. 25.
6. R. B. Hermann, *J. Phys. Chem.*, **76**, 2754 (1972).
7. G. L. Amidon, S. H. Yalkowski, S. T. Anik, and S. C. Valvani, *J. Phys. Chem.*, **79**, 2239 (1975).
8. T. Ooi, M. Obatake, G. Nemethy, and H. A. Scheraga, *Proc. Natl. Acad. Sci. USA*, **84**, 3086 (1987).
9. B. Lee and F. M. Richards, *J. Mol. Biol.*, **55**, 379 (1971).
10. W. C. Still, A. Tempczyk, R. C. Hawley, and T. Hendrickson, *J. Am. Chem. Soc.* **112**, 6127 (1990).
11. W. Hasel, T. F. Hendrickson, and W. C. Still, *Tetrahedron Comp. Meth.*, **1**, 103 (1988).
12. C. J. Cramer and D. G. Truhlar, *J. Am. Chem. Soc.*, **113**, 8305, 9901(E) (1991).
13. C. J. Cramer and D. G. Truhlar, *Science*, **256**, 213 (1992).
14. C. J. Cramer and D. G. Truhlar, *J. Comp. Chem.*, **13**, 1089 (1992).
15. C. J. Cramer and D. G. Truhlar, *J. Comp.-Aid. Mol. Design*, **6**, 629 (1992).
16. C. J. Cramer and D. G. Truhlar, *Chem. Phys. Lett.*, **198**, 74 (1992); **202**, 567(E) (1993).
17. C. J. Cramer and D. G. Truhlar, *J. Am. Chem. Soc.*, **114**, 8794 (1992).
18. C. J. Cramer and D. G. Truhlar, AMSOL—version 1.0 (Quantum Chemistry Program Exchange Program no. 606—version 1.0), *QCPE Bull.*, **11**, 57 (1991).
19. C. J. Cramer, G. C. Lynch, G. D. Hawkins, and D. G. Truhlar, AMSOL—version 3.5 (Quantum Chemistry Program no. 606—version 3.5), *QCPE Bull.*, **13**, 55 (1993).
20. J. L. Pascual-Ahuir and E. Silla, *J. Comp. Chem.*, **11**, 1047 (1990).
21. E. Silla, I. Tuñon, and J. L. Pascual-Ahuir, *J. Comp. Chem.*, **12**, 1077 (1991).
22. J. L. Pascual-Ahuir, E. Silla, and I. Tuñon, GEPOL—version 12.0, Quantum Chemistry Program Exchange Program no. 554, Bloomington, IN, 1992.
23. C. J. Cramer, G. C. Lynch, G. D. Hawkins, D. G. Truhlar, and D. A. Liotard, AMSOL—version 4.0 (Quantum Chemistry Program Exchange Program no. 606—version 4.0),

QCPE Bull., **13**, 78 (1993).

24. C. J. Cramer, G. D. Hawkins, G. C. Lynch, D. G. Truhlar, and D. A. Liotard, AMSOL—version 4.1, (Quantum Chemistry Program Exchange Program no. 606—version 4.1).
25. AMPAC, distributed by SemiChem, Inc.
26. M. L. Connolly, *J. Appl. Cryst.*, **16**, 548 (1983).
27. J. Villa, R. L. Williams, M. Vasquez, and H. A. Scheraga, *Proteins: Structure, Function, Genetics*, **10**, 199 (1991).
28. G. Perrot and B. Maigret, *J. Mol. Graph.*, **8**, 141 (1990).
29. G. Perrot, B. Cheng, K. D. Gibson, J. Vila, K. A. Palmer, A. Nayeem, B. Maigret, and H. A. Scheraga, *J. Comp. Chem.*, **13**, 1 (1992).
30. S. M. Le Grand and K. M. Merz, Jr., *J. Comp. Chem.*, **14**, 349 (1993).
31. B. v. Freyberg and W. Braun, *J. Comp. Chem.*, **14**, 510 (1993).
32. F. Eisenmacher and P. Argos, *J. Comp. Chem.*, **14**, 1272 (1993).
33. R. Carbo, J. A. Hernandez, and F. Sarn, *Chem. Phys. Lett.*, **47**, 581 (1977).
34. A. V. Mitin, *J. Comp. Chem.*, **9**, 107 (1988).
35. P. Pulay, *Chem. Phys. Lett.*, **73**, 393 (1980).