

Version 7.1 Manual

by Gregory D. Hawkins*, David J. Giesen*, Gillian C. Lynch*,
Candee C. Chambers*, Ivan Rossi*, Joey W. Storer*, Jiabo Li*,
Tianhai Zhu*, Jason D. Thompson*, Paul Winget*, Benjamin J.
Lynch*, Daniel Rinaldi**, Daniel A. Liotard***, Christopher
J. Cramer*, and Donald G. Truhlar*.

* Department of Chemistry, University of Minnesota Minneapolis,
Minnesota 55455-0431, USA

** Laboratoire de Chimie Theorique, Universite de Nancy I
Vandoeuvre-Nancy 54506, France

*** Laboratoire de Physico-Chimie Theorique, Universite de Bordeaux
33405 Talence, France

AMSOL-version 7.1

by Gregory D. Hawkins, David J. Giesen, Gillian C. Lynch,
Candee C. Chambers, Ivan Rossi, Joey W. Storer, Jiabo Li,
Tianhai Zhu, Jason D. Thompson, Paul Winget, Benjamin J.
Lynch, Daniel Rinaldi, Daniel A. Liotard, Christopher J.
Cramer, and Donald G. Truhlar.

based in part on AMPAC, version 2.1

by D. A. Liotard, E. F. Healy, J. M. Ruiz, and M. J. S. Dewar
and on EF by Frank Jensen

Date of release of AMSOL-version 7.1: December 6, 2004

Date of most recent update to this manual: May 11, 2010

Copyright© 1992-2004 Regents of the University of Minnesota. All Rights
Reserved

Table of Contents

1. Licensing.....1
AMSOL - version 7.1 is licensed under the [Apache License, Version 2.0](#).
The manual of AMSOL - version 7.1 is licensed under [CC-BY-4.0](#).....1
Publications of results obtained with the AMSOL - version 7.1 software
should cite the program and/or the article describing the program.....1
No guarantee is made that this software is bug-free or suitable for
specific applications, and no liability is accepted for any limitations
in the mathematical methods and algorithms used within. No consulting
or maintenance services are guaranteed or implied.....1
The use of the AMSOL - version 7.1 implies acceptance of the terms of
the licenses.....1

2. INTRODUCTION TO AMSOL.....2
2.1. Introduction to AMSOL..... 2
2.2. General SMx Models..... 3
2.3. SRP Models..... 7
2.4. Accuracy Obtained in SMx Free Energies of Solvation..... 8
2.5. CM1 Models..... 10
2.6. CM2 Models..... 11
2.7. CM3 Models..... 11
2.7. SMx and CMx Notation..... 11
2.8. Availabilty of analytic gradients..... 16

3. BIBLIOGRAPHY.....17
3.1. Suggested References..... 17
3.2. Bibliography: Univ. of Minnesota Papers on Solvation Models.. 21

4. SMx AND CMx MODELS.....22
4.1. Analyzing the NDDO Density Matrix..... 22
4.1.1. Population Analysis and Bond Orders..... 22
4.1.2. Mulliken Analysis in the Zero-Overlap Approximation..... 22
4.1.3. Mulliken Analysis in a De-Orthogonalized Basis..... 23
4.1.4. Bond Order..... 23
4.1.5. Bibliography..... 24
4.1.6. References..... 24
4.2. Geometry Optimization..... 25
4.2.1. Overview..... 25
4.2.2. The Eigenvector Following Method for Optimization..... 27
4.2.3. The Trust Optimizer..... 31
4.2.4. Convergence of Geometry Optimizations..... 31
4.2.5. Suggested Strategy for Geometry Optimizations..... 35

5. PROGRAM OVERVIEW.....37
5.1. Description and Organization of Code..... 37
5.2. Files in Distribution..... 38
5.3. Execution..... 43
5.4. Installation..... 44
5.4.1. FORTRAN compiler comments..... 45

6. USAGE AND KEYWORD EXPLANATIONS.....47
6.1. General Usage Comments..... 47
6.2. Automatic Input Files, Time Limits, and Restarts..... 50
6.3. Keywords..... 51
6.3.1. Keyword ALPHA=x..... 51
6.3.2. Keyword AREAS..... 52
6.3.3. Keyword ASA..... 52
6.3.4. Keyword BETA=x..... 52
6.3.5. Keyword BFGS..... 52
6.3.6. Keyword CART..... 52

6.3.7.	Keyword CM1.....	53
6.3.8.	Keyword CM2.....	53
6.3.9.	Keyword CM3.....	53
6.3.10.	Keyword CS1.....	54
6.3.11.	Keyword CS2.....	54
6.3.12.	Keyword CS3.....	54
6.3.13.	Keyword DDMAX=y (EF keyword).....	54
6.3.14.	Keyword DDMIN=x (EF keyword).....	54
6.3.15.	Keyword DERINU.....	55
6.3.16.	Keyword DERISA.....	55
6.3.17.	Keyword DEV (Developer keyword).....	55
6.3.18.	Keyword DFP.....	55
6.3.19.	Keyword DIELEC (Developer keyword).....	56
6.3.20.	Keyword DMAX=x (EF keyword).....	56
6.3.21.	Keyword DOTS.....	56
6.3.22.	Keyword EFOLLOW (EF keyword).....	56
6.3.23.	Keyword EXTCM (Developer keyword).....	57
6.3.24.	Keyword EXTM.....	57
6.3.25.	Keyword EXTSM (Developer keyword).....	58
6.3.26.	Keyword FACARB=x.xx.....	59
6.3.27.	Keyword FEHALO=x.xx.....	59
6.3.28.	Keyword FOCK.....	59
6.3.29.	Keyword FORWRD.....	59
6.3.30.	Keyword GAMMA=x.....	59
6.3.31.	Keyword GCOMP.....	59
6.3.32.	Keyword GEPOL.....	59
6.3.33.	Keyword HESS=x (EF keyword).....	60
6.3.34.	Keyword INPUT.....	60
6.3.35.	Keyword IOFR=x.....	60
6.3.36.	Keyword IUPD=x (EF keyword).....	60
6.3.37.	Keyword KICK=x.....	60
6.3.38.	Keyword MODE=x (EF keyword).....	61
6.3.39.	Keyword MSURFT.....	61
6.3.40.	Keyword MULLIK.....	61
6.3.41.	Keyword NDIVCD=x.....	61
6.3.42.	Keyword NDIVEP=x.....	61
6.3.43.	Keyword NDOTCD=x.....	61
6.3.44.	Keyword NDOTEP=x.....	61
6.3.45.	Keyword NOFOC.....	61
6.3.46.	Keyword NOINP.....	62
6.3.47.	Keyword NONR (EF keyword).....	62
6.3.48.	Keyword NOPRINTGEOM.....	62
6.3.49.	Keyword NOPRINTCOUL.....	62
6.3.50.	Keyword NOPRINTRAD.....	62
6.3.51.	Keyword NOPRINTPOL.....	62
6.3.52.	Keyword NOTRUS.....	62
6.3.53.	Keyword NOVOL.....	62
6.3.54.	Keyword OMIN=x (EF keyword).....	62
6.3.55.	Keyword PRECISE.....	62
6.3.56.	Keyword PRINTGEOM.....	63
6.3.57.	Keyword PRINTCOUL.....	63
6.3.58.	Keyword PRINTRAD.....	63
6.3.59.	Keyword PRINTPOL.....	63
6.3.60.	Keyword RAD=x (Developer keyword).....	63
6.3.61.	Keyword RECALC=x.....	63
6.3.62.	Keyword RESTART.....	64
6.3.63.	Keyword RMAX=y (EF keyword).....	64

6.3.64.	Keyword RMIN=x (EF keyword)	64
6.3.65.	Keyword RSCAL (EF keyword)	64
6.3.66.	Keyword SCFCRT=x	64
6.3.67.	Keyword SM1	64
6.3.68.	Keyword SM1a	65
6.3.69.	Keyword SM2	65
6.3.70.	Keyword SM2.1	66
6.3.71.	Keyword SM2.2	66
6.3.72.	Keyword SM2.2PDA	66
6.3.73.	Keyword SM3	67
6.3.74.	Keyword SM3.1	67
6.3.75.	Keyword SM4	67
6.3.76.	Keyword SM5.0R	68
6.3.77.	Keyword SM5.05R	68
6.3.78.	Keyword SM5.2PDx	70
6.3.79.	Keyword SM5.2	70
6.3.80.	Keyword SM5.2R	70
6.3.81.	Keyword SM5.42	71
6.3.82.	Keyword SM5.42R	71
6.3.83.	Keyword SM5.4PDx	72
6.3.84.	Keyword SM5.4x	73
6.3.85.	Keyword SOLVNT=x	73
6.3.86.	Keyword STDM	75
6.3.87.	Keyword SVCDRD (Developer keyword)	75
6.3.88.	Keyword SVCSRD (Developer keyword)	75
6.3.89.	Keyword TDUMP=x	75
6.3.90.	Keyword TEXPN=x (Developer keyword)	75
6.3.91.	Keyword TLIMIT=x	77
6.3.92.	Keyword TONE=x (Developer keyword)	77
6.3.93.	Keyword TRUES	77
6.3.94.	Keyword TRUSTE	78
6.3.95.	Keyword TRUSTG	78
6.3.96.	Keyword TSTATE (EF keyword)	78
6.3.97.	Keyword VOLUME	78
6.3.98.	Keyword XKW (Developer keyword)	78
6.3.99.	Keyword XYZ	78
7.	COMPUTATIONAL DETAILS	83
7.1.	Numerical Precision Table	83
7.2.	Observed Discontinuities in the Gradient of Solvent-Accessible Surface Areas (SASAs) of Atoms	85
8.	OUTPUT	85
9.	FILE USAGE	86
10.	TEST RUN INPUT AND OUTPUT	88
11.	REVISION HISTORY	102
	VERSION 3.0.1 (August 1992)	102
	VERSION 3.0.2 (April 1993)	103
	VERSION 3.0.3 (May 1993)	105
	VERSION 3.5 (May 1993)	105
	VERSION 3.5.1 (October 1993)	107
	VERSION 3.5.2 (October 1993)	107
	VERSION 3.5.3 (November 1993)	107
	VERSION 4.0 (November 1993)	107
	VERSION 4.0.1 (March 1994)	109
	VERSION 4.0.2 (May 1994)	110
	VERSION 4.1 (May 1994)	110
	VERSION 4.1.1 (July 1994)	112
	VERSION 4.5 (August 1994)	112

VERSION 4.5.1 (August 1994).....	112
VERSION 4.5.2 (January 1995).....	112
VERSION 4.6 (January 1995).....	113
VERSION 4.6.1 (January 1995).....	114
VERSION 5.0 (April 1995).....	115
VERSION 5.0.1 (September 1995).....	117
VERSION 5.2 (September 1995).....	118
VERSION 5.2.1 (October 1995).....	118
VERSION 5.2.2 (November 1995).....	119
VERSION 5.3 (November 1995).....	119
VERSION 5.3.1 (December 1995).....	119
VERSION 5.4 (December 1995).....	119
VERSION 5.4.1 (April 1996).....	119
VERSION 5.4.2 (April 1997).....	120
VERSION 5.9 (April 1997).....	120
VERSION 6.0 (May 1997).....	120
VERSION 6.1 (May 1997).....	121
VERSION 6.1.1 (August 1997).....	121
VERSION 6.1.2 (February 1998).....	121
VERSION 6.5 (February 1998).....	121
VERSION 6.5.1 (February 1998).....	121
VERSION 6.5.2 (June 1998).....	121
VERSION 6.5.3 (September 1998).....	121
VERSION 6.5.4 (November 1999).....	122
VERSION 6.6 (November 1999).....	122
VERSION 6.6.1 (August 2001).....	123
VERSION 6.7 (September 2001).....	123
VERSION 6.7.1 (October 2001).....	123
VERSION 6.7.2 (January 2002).....	124
VERSION 6.8 (September 2002).....	124
VERSION 6.8.1 (March 2003).....	124
VERSION 6.9 (March 2003).....	124
VERSION 7.0 (October 2003).....	125
VERSION 7.1 (December 2004).....	125
12. INPUT INCOMPATIBILITIES WITH PREVIOUS VERSIONS.....	126
VERSION 5.3.....	126
VERSION 5.9.....	126
VERSION 6.0.....	126
VERSION 6.6.....	126
13. DISTRIBUTED VERSION SUMMARY.....	127
14. RECENT NON-DISTRIBUTED VERSIONS.....	131
15. COMPUTERS AND OPERATING SYSTEMS ON WHICH AMSOL HAS BEEN DEVELOPED.....	132
16. ACKNOWLEDGMENTS.....	140

1. Licensing

AMSOL - version 7.1 is licensed under the [Apache License, Version 2.0](#).
The manual of AMSOL - version 7.1 is licensed under [CC-BY-4.0](#).

Publications of results obtained with the AMSOL - version 7.1 software should cite the program and/or the article describing the program.

No guarantee is made that this software is bug-free or suitable for specific applications, and no liability is accepted for any limitations in the mathematical methods and algorithms used within. No consulting or maintenance services are guaranteed or implied.

The use of the AMSOL - version 7.1 implies acceptance of the terms of the licenses.

2. INTRODUCTION TO AMSOL

2.1. Introduction to AMSOL

AMSOL is an SCF program with unique capabilities for calculating free energies of solvation of molecules and ions in solution and for calculating class IV partial atomic charges. The solvation models and charge models are based on NDDO semiempirical molecular orbital theory in which the terms required to calculate the free energy of solvation are included in the solute Hamiltonian. AMSOL employs either the AM1, PM3, or MNDO model for the solute electronic Hamiltonian. The free energy of solvation is based on two terms: The first is a generalized Born approximation (GBA) contribution that accounts for electric polarization of the continuum-dielectric solvent, i.e., for the electronic and orientational polarization of the solvent molecules and for the resulting feedback of this effect on the solute charge distribution. The partial charges used in this step are obtained by either an NDDO Mulliken population analysis or a CMx mapping of the NDDO Mulliken population analysis of the AM1 or PM3 solute wave function optimized in solution. The second term in the free energy of solvation is a solvent-accessible-surface-area (SASA) term that accounts for the free energy of cavity formation, dispersion interactions, and first-solvation-shell effects such as solvent structure changing and the non-electrostatic part of hydrogen bonding. This surface area term is sometimes called the surface tension (ST) term or the cavity-dispersion-solvent-structure (CDS) term, and it includes both solvophilic and solvophobic contributions. This term, being semiempirical, allows as well for errors in AM1, PM3, or MNDO (especially in the partial charges for those solvation models that use the NDDO Mulliken population analysis), for the deviation of the dielectric constant in the first hydration shell from its bulk value, for the intrinsic uncertainty in the definition of atomic radii, for the error due to replacing the continuous charge distribution of the solute by a discrete set of nuclear-centered partial charges, and so forth. AMSOL is capable of incorporating class IV charges, in particular Charge Model 1 (CM1), Charge Model 2 (CM2), or Charge Model 3 into gas-phase calculations. For CM1 and CM2, solvation calculations are also available. All models whose name includes .4 are based on CM1 charges, as is the SM4 model. Models whose name includes .42 are based on CM2 charges. CM1, CM2, and CM3 mappings begin with the set of partial charges from an NDDO Mulliken population analysis and results in a new set of partial charges that more accurately reproduce experimental dipole moments. The resultant CM1, CM2, and CM3 partial charges also agree quite closely with partial charges obtained from correlated large-basis-set ab initio calculations using the CHELPG fitting method. CM1, CM2, and CM3 charges, however, are typically 2 to 4 orders of magnitude faster to compute. The semiempirical mapping of the CM1 charge models is accomplished through a set of scale factors and offsets that depend on atomic number. Some parameters are also dependent on bond orders. Using the scale factors and offsets, charge is shifted locally in the molecule so that individual bond dipoles are altered, but the overall charge on the molecule remains constant. CM1 charges mapped from AM1 are called CM1A, and CM1 charges mapped from PM3 are called CM1P. CM2 and CM3 charges can be mapped from AM1 or PM3 using either the CM2 keyword or the CM3 keyword and the appropriate semiempirical Hamiltonian keyword. For gas-phase calculations, AMSOL contains all the semiempirical methods and parameters available in MOPAC-version 6, i.e.,

MINDO/3, MNDO, AM1, and PM3. PM3 parameters are also available for gas-phase calculations on molecules incorporating Li.

2.2. General SMx Models

Several general parameterizations are available for calculating the solvation terms. They are (followed by the version number of the code in which they were introduced): SM1 and SM1a for water (1.0), SM2 and SM3 for water (3.0), SM2.1 and SM2.2 for water (4.1 and 5.2 respectively), SM3.1 for water (4.6), SM4 for alkanes (5.0), SM5.4/U, SM5.4PD/AM1, SM5.4PD/PM3, SM5.2PD/AM1, SM5.2PD/PM3, and SM2.2PD/AM1 for water (6.0), SM5.0R for water(6.0), SM5.4/AM1 and SM5.4/PM3 for all solvents (6.0), SM5.2R for water and organic solvents using the AM1, PM3, or MNDO Hamiltonians (6.5), SM5.0R for organic solvents (6.5), and SM5.42R for water and organic solvents based on the AM1 or PM3 Hamiltonians (6.5).

For all models with a lower model number than SM4, there is a unique underlying gas-phase Hamiltonian which is AM1 for SM1, SM1a, SM2, SM2.1, and SM2.2, and is PM3 for SM3 and SM3.1. Where a solvation model has multiple parameter sets corresponding to choices of the underlying gas-phase model, then the solvation model has a suffix following a /. The SM4 alkane model and all parameterizations designated /U function with both AM1 and PM3. All parameterizations designated /A or /AM1 function with the AM1 Hamiltonian, all models designated /P or /PM3 function with the PM3 Hamiltonian, and all parameterizations designated /M function with the MNDO Hamiltonian.

The solvation models SM2.1, SM2.2, and SM3.1, and all models SM4 and higher use an improved integration scheme for the radial quadrature of the screening calculation as compared to the earlier solvation models. The SM2.1 and SM3.1 methods' parameters were derived by fitting to SM2 (or SM3) results and thus the name SM2.1 (SM3.1) indicates a method which should yield results similar to those obtained with SM2 (SM3), but utilizing the improved integration scheme. The majority of parameters for all other models were determined by fitting to experimental results.

The original SM1 and SM1a models and the determination of their parameters are described in "General Parameterized SCF Model for Free Energies of Solvation in Aqueous Solution" by C. J. Cramer and D. G. Truhlar, *Journal of the American Chemical Society* 113, 8305-8311, 9901(E) (1991). The SM1 model can be used for ionic or neutral systems made up of H, C, N, O, F, P, S, Cl, Br, and I atoms. (Parameters for all except P were introduced in the reference above; parameters for P were introduced in the 1992 J. Computer-Aided Molecular Design reference below.) The SM1a model is a more specialized parameter set which is applicable to neutral molecules that are composed of the same atoms as in the SM1 model but that do not have hypervalent centers, three-center bonds, or unusual hybridization at N or O. The SM1a model requires that H, C, and N be assigned a "type", e.g., sp³ or sp² oxygen.

The SM2 model was introduced in "An SCF Solvation Model for the Hydrophobic Effect and Absolute Free Energies of Solvation" by C. J. Cramer and D. G. Truhlar, *Science* (Washington, D. C.) 256, 213-217 (1992). This parameter set is an improved solvation model for the same atoms listed above for the SM1 model, including P.

The SM3 model was introduced in "PM3-SM3: A New General Parameterization for Including Aqueous Solvation Effects in the PM3 Molecular Orbital Model" by C. J. Cramer and D. G. Truhlar, *Journal of Computational Chemistry* 13, 1089-1097 (1992). The parameterization is like the SM2 model but it is based on the PM3 Hamiltonian.

A full comparison of the first four methods, together with a detailed description of both the models and the computational methodologies employed by AMSOL through version 3.0, may be found in "AM1-SM2 and PM3-SM3 Parameterized SCF Solvation Models for Free Energies in Aqueous Solution" by C. J. Cramer and D. G. Truhlar, *Journal of Computer-Aided Molecular Design* 6, 629-666 (1992).

The derivation of the SM2.1 model and the new integration scheme it utilizes may be found in "Improved Methods for Semiempirical Solvation Models" by D. A. Liotard, G. D. Hawkins, G. C. Lynch, C. J. Cramer, and D. G. Truhlar, *Journal of Computational Chemistry* 16, 422-440 (1995). The new radial integration scheme, called the force trapezoid algorithm, provides several benefits; from a user's point of view the most important benefit may be that it allows a tradeoff between precision and cost in the radial quadratures so that less precise calculations may be performed at lower cost in exploratory work if desired. The SM3.1 model is described in a note added in proof to this paper. A model called SM2.2 is the same as the SM2.1 model except that it involves the pairwise descreening approximation for the electrostatics and a new set of surface tensions designed to be consistent with these. The details of the pairwise descreening approximation and the SM2.2 model can be found in "Pairwise Solute Descreening of Solute Charges from a Dielectric Medium" by G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, *Chemical Physics Letters* 246, 122-129 (1995). The pairwise dielectric descreening approximation in SM2.2 was parameterized to reproduce the electrostatics predicted by the full generalized Born formula for H, C, O, and N containing compounds. A more fully developed version of this method is available in the SM2.2PD/A model, but the pairwise descreening parameters were not restricted to reproduce the electrostatics of a full generalized Born calculation.

Models SM1 through SM3.1 are based on partial charges computed from AM1 and PM3 wave functions by Mulliken population analysis with neglect of differential overlap. This is called either NDDO Mulliken analysis or zero-overlap Mulliken analysis (see Section 4.1 for discussion). In SM4 and all models with .4 in the name, the partial charges are obtained from class IV charge models. The SM2, SM2.1, SM2.2, SM3, and SM3.1 models involve empirical cut-off gaussians (COGs) in the electrostatic terms for molecules involving geminal oxygens or vicinal N and H.

The SM4 model was introduced in "A General Semiempirical Quantum Mechanical Solvation Model for Nonpolar Solvation Free Energies. n-Hexadecane" by D. J. Giesen, J. W. Storer, C. J. Cramer, and D. G. Truhlar, *Journal of the American Chemical Society* 117, 1057-1068 (1995), which provided parameters for n-hexadecane solvent. The SM4 model was extended to all alkanes, including cycloalkanes, in "A Semiempirical Quantum Mechanical Model for Solvation Free Energies in All Alkane Solvents" by D. J. Giesen, C. J. Cramer, and D. G. Truhlar, *Journal of Physical Chemistry* 99, 7137-7146 (1995). The SM4 model uses the CM1 models (see Section 2.5), an improved definition of hydrogen surface tension that depends on the bond order to heavy atoms, and two solvent radii to account for short- and intermediate-range interactions. Surface tensions associated with these ranges are labeled CD and CS, respectively.

The SM5.4/U, SM5.4/AM1, (also called SM5.4/A) and SM5.4/PM3 (also called SM5.4/P) aqueous parameterizations are presented in "A Model for Aqueous Solvation Based on Class IV Atomic Charges and First-Solvation-Shell Effects" by C. C. Chambers, C. J. Cramer, and D. G. Truhlar,

Journal of Physical Chemistry 100, 16385-16398 (1996). The SM5.4 model is based on a new set of geometry-based functional forms for parameterizing effective coulomb radii and atomic surface tensions for solutes encompassing a wide variety of functional groups in water. The functional forms of microscopic surface tensions depend on solute atomic numbers and geometries, and any model employing the same or similar geometry dependencies is called an SM5 model. (There is a whole suite of SM5 models, some complete and some under development corresponding to a variety of treatments of the electrostatics, and some of those are parameterized for more than one solute Hamiltonian.) All models employing these functional forms (sometimes with a few modifications) along with solvent-accessible surface areas (in certain models the van der Waals surface is used rather than the classical solvent-accessible surface area. Note that the van der Waals surface can be considered a special case of the solvent-accessible surface area where the solvent radius is taken to be zero.) and atom-centered point charges are denoted SM5.x where x specifies the class of the charges. In the SM5.4 model, the atomic charges are obtained using the CM1 class IV charge model. The only differences between the SM5.4/AM1, SM5.4/PM3, and SM5.4/U parameterizations are small changes in the microscopic surface tensions. SM5.4/AM1 and SM5.4/PM3 have parameters optimized for use with the AM1 and PM3 solute Hamiltonians, respectively, whereas SM5.4/U is considered an "unspecific" parameterization that can be either AM1 or PM3 or (in principle) with any solute treatment giving similar electrostatics to the GBA method with CM1 partial charges. These parameterizations are based on 215 neutral solutes and 34 ionic solutes. The SM5.4/U parameterization is available only for water, but the SM5.4/AM1 and SM5.4/PM3 parameterizations are also available for organic solvents.

The parameterization of SM5.4/AM1 and SM5.4/PM3 for all organic solvents was accomplished in three stages. First we obtained a general set of parameters for all organic solvents. These parameters are accepted as the final SM5.4/AM1 and SM5.4/PM3 parameters for all organic solvents except chloroform, benzene and toluene. For chloroform the parameters were extensively readjusted; however this was done by building on the general parameterization rather than by starting from scratch, so one obtains a consistent model. The reparameterization for aromatic hydrocarbon solvents is also based on the general organic parameterization, and for these two solvents only one parameter is changed from parameter sets.

One could conceivably use the SM5.4 general organic parameters for chloroform and aromatic hydrocarbons, but the average errors would be larger if this were done; therefore the notation SM5.4/AM1 and SM5.4/PM3 is defined to imply the use of the special parameters for these solvents. In general, the notation SM5.4/AM1 and SM5.4/PM3 without designation of a parameter set implies the use of the special parameters for water, chloroform, benzene, and toluene and the general organic parameters for any other solvent. When it is required to explicitly distinguish the various parameter sets, the general organic parameters are sometimes called the organic parameters, and their use may be denoted by SM5.4/AM1-organic or SM5.4/PM3-organic; these names are sometimes in turn abbreviated as OSM5.4/AM1 and OSM5.4/PM3, which may be read Organic Solvation Model 5.4/AM1 and Organic Solvation Model 5.4/PM3.

The SM5.4/AM1-organic, and SM5.4/PM3-organic parameterizations are presented in "A Universal Organic Solvation Model" by D. J. Giesen, M. Z. Gu, C. J. Cramer, and D. G. Truhlar, Journal of Organic Chemistry 61, 8720-8721 (1996), which is a communication, and in a follow-up full

paper, "A Universal model for the Quantum Mechanical Calculation of Free Energies of Solvation in Non-aqueous Solvents," D. J. Giesen, C. J. Cramer, D. G. Truhlar, *Theoretical Chemistry Accounts*, 98, 85-109 (1997). The communication considers only the /AM1 parameterization and it does not present stages 2 and 3 in which the final parameters were obtained for chloroform, benzene, and toluene solvents. The full paper considers both the /AM1 and /PM3 parameterizations, and it presents the general parameters as well as the special case of benzene and toluene. The SM5.4/AM1 and SM5.4/PM3 parameterizations for chloroform are presented in "Solvation Model for Chloroform Based on Class IV Atomic Charges" by C. C. Chambers, D. J. Giesen, M. Z. Gu, C. J. Cramer, and D. G. Truhlar, *J. Phys. Chem.* 101, 2061-2069 (1997). All SM5.4 parameterizations are based on the same solute-geometric dependencies for surface tensions as the aqueous models, and they also involve the same forms for the electrostatic terms, in particular they have no electrostatic COGs. Because the COGs are eliminated and the partial charges are more accurate (class IV charges), these models are considered to be our most realistic models in terms of the faithfulness of individual terms for representing the underlying physics of solvation. In addition, the general organic, chloroform, benzene, and toluene parameterizations include a CS surface tension term (similar to the SM4 models), which is not present as a separate term for water. Furthermore, in the SM5.4-organic parameterizations, the microscopic surface tensions depend on solvent properties as well as on solute-atom atomic number and solute geometry, thereby allowing accurate solvation free energies to be predicted for any solvent for which all necessary solvent properties are known. Since the solvent properties we use were specifically selected for their wide availability, the SM5.4 general organic parameterizations are effectively universally applicable to nonaqueous solvents.

The SM5.4PD/U, SM5.4PD/AM1, (also called SM5.4PD/A) and SM5.4PD/PM3 (also called SM5.4PD/P) models are introduced in "Parameterized Models of Aqueous Free Energies of Solvation Based on Pairwise Descreening of Solute Atomic Charges from a Dielectric Medium" by G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, *Journal of Physical Chemistry* 100, 19824-19839 (1996). These parameterizations use the SM5 functional forms for surface tensions and have no electrostatic COGs. They are based on the CM1 class IV charges and are parameterized for water only for use with the pairwise descreening algorithm. This algorithm replaces a volume integral over the free energy density in the solvent by an analytic approximation to the descreening effect. This speeds up the calculation considerably, although one might expect less robustness in applications to systems that differ qualitatively from those in the training set.

The SM5.2PD/AM1 (also called SM5.2PD/A) is introduced in the same paper as the SM5.4PD/AM1 and SM5.4PD/PM3 model. The SM5.2PD model uses the SM5 functional forms for surface tensions and has no electrostatic COGs. It is based on partial atomic charges determined by NDDO Mulliken population analysis of AM1 solution-phase wave functions (as used in the SM2 and SM2.1 models), and it is parameterized for water only for use with the pairwise descreening algorithm. The motivation for using these simpler charges is that it cuts the cost of the calculation, and so this method is recommended for use only when cost is a paramount consideration.

For parameterizations which may be used with more than one solute Hamiltonian, in particular the SM4 or the /U parameterizations, the

solute Hamiltonian should (when completeness is desired) be specified as a prefix, e.g., AM1-SM4 or PM3-SM5.4/U.

The SM5.0R and the SM5.05R models do not require a solvent Hamiltonian. The SM5.0R model treats all electrostatic effects implicitly, and the entire estimation of the free energy of solvation is based on surface tension coefficients. The SM5.05R model uses a very simplified representation for the charges for certain typed groups in ionic and zwitterionic molecules. See reference 27 in Section 3.1.

The SM5.2R model is presented in "A Universal Quantum Mechanical Model for Solvation Free Energies Based on Gas-Phase Geometries" by G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, *Journal of Physical Chemistry B* 102, 3257-3271 (1998). This model uses SM5 functional forms (slightly different from those used for SM5.4/AM1 and SM5.4/PM3) for surface tensions and has no electrostatic COGS. It is parameterized for water and organic solvents and designed to be used with accurate gas-phase geometries.

The SM5.42R model is presented in "Extension of the SM5.42R Universal Solvation Model to Additional Fock and Kohn-Sham Operators and to Hybrid Hartree-Fock-Density-Functional Theory", by J. Li, T. Zhu, G. D. Hawkins, D. A. Liotard, C. J. Cramer, and D. G. Truhlar, *Theor. Chem. Acc.*, in press. It uses the same functional forms as the SM5.2R model, but is based on Charge Model 2 (CM2) class IV charges. It was designed to be used with accurate gas-phase geometries and either the AM1 or PM3 semiempirical Hamiltonian.

2.3. SRP Models

In addition to using built-in general parameterizations, AMSOL can also perform calculations using user-supplied solvation parameters. From the point of view of solutes, all standard SMx models are general parameterizations, i.e., the parameters are optimized for as broad as possible a range of solute functionalities. AMSOL can also accept user-supplied specific reaction parameters (also called specific range parameters or system-specific parameters). These are parameters that have been optimized for a single solute system or a single reaction or a small range of solutes or reactions, rather than for all solutes for which data are available. Two examples of this usage are provided and are discussed in Section 2.3.

We have developed two sets of SM4-SRP parameters for aqueous solvation of specific ranges of compounds containing H, C, and O. The first, the AM1-SM4-SRP:Claisen model, was developed to study the Claisen Rearrangement and was presented in reference 15 of Section 3.1. The second, the AM1-SM4-SRP:sugar model, was developed for the study of glucose or other sugars and was presented in reference 20 of Section 3.1. Because the methods have been parameterized over limited sets of compounds, they are called Specific Range Parameter (SRP) models, and they are not recommended for use on functionalities beyond those for which they were parameterized. The compounds used in the development of the Claisen model include hydrocarbons, ethers, and aldehydes. The sugar model was developed for hydrocarbons, ethers, aldehydes, and alcohols.

The AM1-SM4-SRP models are not treated the same as other solvation models in this version of AMSOL because they are not general models. However, it is possible to run either type of calculation using the SOLVNT=H2OSRP keyword (see Section 6.3). To do this, the following keywords must be specified:

```
SOLVNT=H2OSRP
DIELEC=78.3
```

```
MSURFT=0.0
SVCDRD=1.4
EXTSM
DEV
```

The SM4-SRP models use the same functional forms as the SM4 model with one exception; in particular SM4-SRP models do not include the O-O electrostatic COG that is included in the SM4 alkane models. This COG is turned off by the use of the SOLVNT=H2OSRP keyword.

In addition, for the AM1-SM4-SRP: Claisen model, the EXTSM file must contain the following lines:

```
H 1.20D0    8.886D0    0.000D0    0.590D0    1.283D0    -0.300D0    0.100D0
C 1.70D0   -6.910D0    0.000D0    1.798D0    0.000D0    0.000D0    0.100D0
O 2.00D0   -6.424D0    0.000D0    1.350D0    0.000D0    0.000D0    0.100D0
CC  0.0D0
CS  0.0D0
```

For the AM1-SM4-SRP:sugar parameters, the EXTSM file must contain the following lines:

```
H 1.20D0    0.000D0    0.000D0    0.590D0    1.283D0    -0.300D0    0.100D0
C 1.70D0   -6.910D0    9.090D0    1.798D0    0.000D0    0.000D0    0.100D0
O 2.00D0   -6.420D0  -188.000D0  1.350D0    0.000D0    0.000D0    0.100D0
CC  0.0D0
CS  0.0D0
```

Note that setting the CS surface tension in the EXTSM file overrides any surface tension that may be calculated from the MSURFT keyword. However, use of the SOLVNT=H2OSRP keyword requires that MSURFT be used to input a macroscopic surface tension.

Trial runs tr6a2.dat and tr7a2.dat illustrate the use of these models, and the files tr6a2.xsm and tr7a2.xsm are the appropriate .xsm files for the Claisen and sugar models, respectively.

2.4. Accuracy Obtained in SM_x Free Energies of Solvation

The following table is provided to show the user how well the free energies of solvation for various SM_x models compare to experiment. All values in the tables are mean unsigned errors (MUEs), i.e., the average of the absolute values of the errors. The numbers in this table are taken from a variety of sources, as summarized next.> The MUE provided in the table for the SM1, SM1a, SM2, SM2.1, SM3, and SM3.1 models are over the 147 neutrals in table 5 in reference 8 of Section 3.1 or the 28 ions of Table 6 in the same reference. No error is listed for the SM1a model under ions because that model is only valid for molecules in which atoms take on their standard hybridizations; this is a situation which is not realized for many of the ions in table 6 of reference 8.

- The MUE provided in the table for the SM2.2 model are for the 139 molecules listed in reference 16 of Section 3.1
- The MUE provided in the table for the SM4 Alkane models are for the 506 solvent/solute pairs in references 12 and 13 of Section 3.1.
- The MUE provided in the table for the SM5.0R model is over 260 solute compounds with 248 data points in water and 1836 data points in organic solvents.

- The MUE for the AM1-SM5.4/U, PM3-SM5.4/U, SM5.4/AM1, and SM5.4/PM3 aqueous models are based on the 215 neutrals and 34 ions in the SM5.4 aqueous model neutral and ions training sets. An additional 4 neutral phosphorus molecules have been factored into the reported MUE for SM2.2PD/A, SM5.2PD/A, and SM5.2PD/P.
- The MUE for the SM5.4/AM1-organic and SM5.4/PM3-organic parameterizations are based on a total 1449 measurements of neutral molecules in a total of 78 different solvents. This excludes chloroform, benzene, and toluene.
- The MUE for the SM5.4/AM1-chloroform and SM5.4/PM3-chloroform parameterizations are based on free energies of solvation derived from measurements on 82 neutrals.
- The MUE for the SM5.4-benzene parameterizations are based on free energies of solvation derived from measurements on 60 neutrals.
- The MUE for the SM5.4-toluene parameterizations are based on free energies of solvation derived from measurements on 45 neutrals.
- The MUE provided in the table for the AM1-SM4-SRP:Claisen model are for 39 alkanes, cycloalkanes, alkenes, conjugated polyenes, ethers, and aldehydes listed in table 2 of reference 14 of Section 3.1. The MUE provided in the table for the AM1-SM4-SRP:sugar model are for 47 compounds consisting of these 39 plus 8 alcohols as specified in reference 15 of Section 3.1.
- The MUE provided in the table for the SM5.2R model is over 260 non-silicon solute compounds with 248 data points in water and 1836 data points in organic solvents. The error associated with the ions is over 43 ions.
- The MUE provided in the table for the SM5.42R model is over 260 non-silicon solute compounds with 248 data points in water and 1836 data points in organic solvents. The error associated with the ions is over 43 ions. For silicon-containing compounds, see Ref. 33 in Section 3.1.

The user should note that since these MUE are not all based on the same set of solutes; thus even when they do refer to the same solvent, they are not strictly comparable. Nevertheless they are provided as a rough guide as to the reliability of the various methods.

Algoritithm	Solvent(s)	MUE	MUE
		Neutrals	Ions
AM1-SM1	water	1.1	2.9
AM1-SM1a	water	0.6	-
AM1-SM2	Water	0.7	4.1
AM1-SM2.1	Water	0.7	3.2
AM1-SM2.2	Water	1.2	-
AM1-SM2.2PD/AM1	Water	1.2	-
PM3-SM3	Water	1.0	3.5
PM3-SM3.1	Water	0.9	3.4
AM1-SM4	Alkanes	0.3	-
PM3-SM4	Alkane	0.3	-
AM1-SM4-SRP:Claisen model	Water	1.0	-
AM1-SM4-SRP:sugar model	Water	0.9	-
SM5.0R//HF/MIDI!	Water and organic solvents	0.4	-
SM5.0R//AM1	Water and organic solvents	0.4	-
AM1-SM5.2PD/AM1	Water	0.5	4.2
PM3-SM5.2PD/PM3	Water	0.5	3.6
AM1-SM5.4PD/AM1	Water	0.5	3.6
PM3-SM5.4PD/PM3	Water	0.5	3.7
AM1-SM5.4/U	Water	0.6	-
PM3-SM5.4/U	Water	0.5	-
SM5.4/AM1	Water	0.5	4.3
SM5.4/PM3	Water	0.4	4.4
SM5.4/AM1	Organic, excl. chloroform, benzene, toluene	0.5	-
SM5.4/PM3	Organic, excl. chloroform, benzene, toluene	0.4	-
SM5.4/AM1	Chloroform	0.4	-
SM5.4/PM3	Chloroform	0.4	-
SM5.4/AM1	Benzene	0.5	-
SM5.4/PM3	Benzene	0.4	-
SM5.4/AM1	Toluene	0.3	-
SM5.4/PM3	Toluene	0.3	-
SM5.2R/MNDO//HF/MIDI!	Water and organic solvents	0.4	3.8
SM5.2R/AM1//HF/MIDI!	Water and organic solvents	0.4	3.7
SM5.2R/PM3//HF/MIDI!	Water and organic solvents	0.4	3.6
SM5.42R/AM1//HF/MIDI!	Water and organic solvents	0.4	4.1
SM5.42R/PM3//HF/MIDI!	Water and organic solvents	0.4	4.2

2.5. CM1 Models

The Charge Model 1 methods CM1A and CM1P for the AM1 and PM3 Hamiltonians, respectively, begin with partial charges calculated by an NDDO Mulliken population analysis, and they map these charges to new sets of partial charges that more accurately reproduce experimental dipole moments. The resultant partial charges also agree quite closely with partial charges obtained from correlated large-basis-set ab initio

calculations using the CHELPG fitting method; CM1 charges, however, are typically 2 to 4 orders of magnitude faster to compute. AMSOL uses the class IV partial charge method, CM1, in the calculation of polarization free energies in SM4, and all models designated with a .4. The CM1x methods were introduced in "Class IV Charge Models: A New Semiempirical Approach in Quantum Chemistry" by J. W. Storer, D. J. Giesen, C. J. Cramer, and D. G. Truhlar, *Journal of Computer-Aided Molecular Design* 9, 87-110 (1995).

The semiempirical mappings that lead to the CM1A and CM1P charge models are accomplished through a set of scale factors and offsets that depend on atomic number. Some parameters are also dependent on bond orders. Using the scale factors and offsets, charge is shifted locally in the molecule so that individual bond dipoles are altered, but the overall charge on the molecule remains constant.

Note that CM1A is also called CM1/AM1 and CM1P is also called CM1/PM3.

2.6. CM2 Models

Charge Model 2 is our second Class IV charge model. See reference 31 in Section 3.1.

2.7. CM3 Models

Charge Model 3 is our third Class IV charge model. See reference 32 in Section 3.1.

2.7. SMx and CMx Notation

Abbreviations like SM2 or SM3 should be used only to refer to the precise models specified by the authors in the original papers unless the method is labeled as modified, and the changes are clearly and precisely explained by persons making modifications. In such cases we recommend affixing the SRP suffix followed by a colon and a designation of the class of systems for which the modification is designed to be used. For example SM3.1-SRP:PCB would designate a reparameterization of the SM3.1 model for polychlorinated biphenyls. There are several different SMx models, and the rest of this section provides a brief discussion of the naming of SMx models and the differences between them. The section closes with a few comments on CM1 notation.

The SMx models may be specified in three possible ways:

- 1 - by the pair of keywords that specifies a given model. One keyword specifies the solute Hamiltonian; the other specifies the solvation method.
- 2 - by the standard name of the model. This is a short name that uniquely distinguishes a model from other currently available or planned models.
- 3 - by specifying the choices made for the features that distinguish the models.

Specification of type 1 is of course useful for using the AMSOL code, but the second type is the preferred method for publications. The third type is necessary for understanding and background.

Note that OSM5.4 is an alternative name for SM5.4-organic, and this notation is not recommended for general use.

In the authors' own work, we try to be consistent in our notation, but readers should be aware that other people sometimes use the notation incorrectly.

The following solvation models are available in this version of AMSOL.

Standard Keywords	Name of model	Descreening charge algorithm	class
AM1 SM1	SM1	RD	2
AM1 SM1a	SM1a	RD	2
AM1 SM2	SM2	RD	2
AM1 SM2.1	SM2.1	DD	2
AM1 SM2.2	SM2.2	[preliminary version of SM2.2PD/A]	
AM1 SM2.2PDA	SM2.2PD/AM1	PD	2
PM3 SM3	SM3	RD	2
PM3 SM3.1	SM3.1	DD	2
AM1 SM4	AM1-SM4	DD	4 (also requires SOLVNT keyword)
PM3 SM4	PM3-SM4	DD	4 (also requires SOLVNT keyword)
AM1 SM5.4U	AM1-SM5.4/U	DD	4 (also requires SOLVNT keyword)
PM3 SM5.4U	PM3-SM5.4/U	DD	4 (also requires SOLVNT keyword)
AM1 SM5.4A	SM5.4/AM1	DD	4 (also requires SOLVNT keyword)
PM3 SM5.4P	SM5.4/PM3	DD	4 (also requires SOLVNT keyword)
AM1 SM5.4PDU	AM1-SM5.4PD/U	PD	4 (also requires SOLVNT keyword)
PM3 SM5.4PDU	PM3-SM5.4PD/U	PD	4 (also requires SOLVNT keyword)
AM1 SM5.4PDA	SM5.4PD/AM1	PD	4 (also requires SOLVNT keyword)
PM3 SM5.4PDP	SM5.4PD/PM3	PD	4 (also requires SOLVNT keyword)
AM1 SM5.2PDA	SM5.2PD/AM1	PD	2 (also requires SOLVNT keyword)
PM3 SM5.2PDA	SM5.2PD/PM3	PD	2 (also requires SOLVNT keyword)
SM5.0R	SM5.0R	N.A.	N.A. (also requires SOLVNT keyword)
SM5.05R	SM5.05R	DD	0.5 (also requires SOLVNT keyword)
AM1 SM5.2R	SM5.2R	DD	2 (also requires SOLVNT keyword)
PM3 SM5.2R	SM5.2R	DD	2 (also requires SOLVNT keyword)
MNDO SM5.2R	SM5.2R	DD	2 (also requires SOLVNT keyword)
AM1 SM5.2	SM5.2	DD	2 (also requires SOLVNT keyword)
PM3 SM5.2	SM5.2	DD	2 (also requires SOLVNT keyword)
MNDO SM5.2	SM5.2	DD	2 (also requires SOLVNT and DERINU keywords)
AM1 SM5.42R	SM5.42R	DD	4 (also requires SOLVNT keyword)
PM3 SM5.42R	SM5.42R	DD	4 (also requires SOLVNT keyword)
AM1 SM5.42	SM5.42	DD	4 (also requires SOLVNT keyword)
PM3 SM5.42	SM5.42	DD	4 (also requires SOLVNT and DERINU keywords)

Notice that the slashes are omitted in the keywords, but the decimals are included. The SOLVNT keyword is required if more than one parameterization set is available (e.g., SM5.4/AM1-aqueous, SM5.4/AM1-organic, and SM5.4/AM1-chloroform). The differences among the models are explained next.

The general nomenclature for a method is one of the following:

SMx
SMx.#
SMx.#/Y
SM5.4/Y-parset
SMx.#Z/Y

The meanings of the various parts of the names are as follows:

SM is pronounced "Solvation Model."

The x primarily denotes the functional form of surface tensions and in practice it denotes the functional form of the generalized Born term as well.

in SMx.# (namely, SM2.1, SM2.2, SM3.1) denotes a further development of the SMx model, whereas # in a method with a / in the name (e.g., SM5.4/AM1 or SM2.2PD/AM1) denotes the type of atomic charges used plus, possibly, another digit that is a version number. In the latter context, # may be 0, 2, 4, or 42.

Y or Y-parset denotes the parameter set (surface tension coefficients, radii, etc.). Typically Y denotes the solute Hamiltonian for which parameters were developed, or Y may be U, denoting nspecific. In AMSOL, unspecific parameterizations are those that were developed simultaneously for AM1 and PM3. If parset is present in the name, it denotes the solvent or set of solvents for which the parameters were developed.

Z denotes nonstandard algorithm choices. In AMSOL there are two possible nonstandard options. If Z has the value PD it indicates use of the pairwise descreening algorithm. R in the Z position signifies a "rigid" model, where parameters were determined such that the solvation free energy may be predicted from an accurate gas phase structure which is not allowed to relax in solution. (Note that in "rigid" models the electronic structure is still allowed to relax in the field created by the solvent.)

Further details about the issues involved in these name components are summarized in the rest of this section.

One may use parts of names to refer to groups of models. For example, SM5 refers to all models with SM5 anywhere in the name, and SM5.4 refers to all models with SM5.4 anywhere in the name.

The following is a complete list of solvents supported by various models in this version of AMSOL:

Water	SM1 through SM3.1, all SM5.x
Any alkane	SM4, SM5.4, SM5.0R, SM5.2R, SM5.42R
Any other organic solvent	SM5.4, SM5.0R, SM5.2R, SM5.42R

The x in SMx denotes the functional form of surface tensions. This x also specifies the functional form of the generalized Born term, in particular whether electrostatic COGs are used, since that goes with the choice of surface tension functional forms.

SM1 surface tensions based only on atomic number, i.e., nuclear charge; O-O and N-H electrostatic COGs are used SM1a surface tensions depend on atomic number and hybridization, i.e., sp³, sp², ...; O-O and N-H electrostatic COGS are used.

- SM2 surface tensions depend on atomic number and bond orders; O-O and N-H electrostatic COGs are used
- SM3 same as SM2
- SM4 surface tensions depend on atomic number and bond orders of the solute atom and on the macroscopic surface tension of the solvent; O-O electrostatic COGs are used.
- SM5 surface tensions depend on interatomic distances; electrostatic COGs are not used.

The decimal notation has a different meaning in models named with or without a slash.

- In models without a slash, the # after the point is simply a chronological label for versions that are variations on earlier methods, e.g., SM2.1 is a variation of SM2.
- In models with a slash in the name or in which the numeral after M is 5, the # after the slash denotes the class of charges employed. For example, an SM5.2 model uses class II charges (such as charges based on Mulliken analysis), an SM5.4 model uses CM1 class IV charges, an SM5.42R model uses CM2 class IV charges, and an SM5.0 model is one in which all charges are set equal to zero.

The # after the decimal point takes the form of a digit:

- 0 denotes no explicit electrostatics; neither the generalized Born term nor any other electrostatic treatment (e.g., COSMO) is used.
- 1 denotes electrostatics based on Class I charge models, i.e., charge models not based on wave functions.
- 2 denotes electrostatics based on Class II zero-overlap Mulliken charges from a solution-phase wave function.
- 4 denotes electrostatics based on CM1 Class IV charges from a solution-phase wave function.
- 42 denotes our second model using class IV vcharges for electrostatics; in particular, this model is based on CM2 Class IV charges from a solution-phase wave function.

Note that this version of AMSOL only includes # = 0, 2, 4, or 42. Models corresponding to # = 1 are being considered for later development. The availability of various levels (a "level" refers to the choices for # and Z) will allow the user to choose an appropriate compromise of speed and reliability, depending on the size of the solute, the computational speed requirements, and other aspects of the application.

Three different descreening algorithms have been used in the generalized Born approximation:

- DD density descreening (without further approximations, in particular with our well converged trapezoidal rule or --in principle-- any other radial quadrature scheme since if one converges the integral it doesn't really matter how it was done)
- DD(GL) DD with Gauss-Legendre quadrature
- RD rectangular-rule descreening, an approximation to DD
- PD pairwise descreening

In SM1, SM1a, SM2, and SM3, the RD algorithm is used. In all other models the DD algorithm is used, unless PD is specified in the model name. The SM4 model and the SM5 model whose parameter set is designated U (U denotes unspecific) may be used with more than one solute Hamiltonian--in such cases one may indicate the solute Hamiltonian as a prefix, as in AM1-SM4 or AM1-SM5.4/U. In fact this is required for complete specification of the method. In other cases, e.g., SM2, one may also denote the solute Hamiltonian this way, e.g., AM1-SM2, but this is optional since correct usage of SM2 requires that the solute Hamiltonian be AM1. In column 2 of the table above, we include AM1- or PM3- only where it is needed for completeness, not where it is optional.

The parameter set is a label for a specific set of surface tension coefficients. The parameter set label is usually an abbreviation for the solute Hamiltonian that was used to develop the parameters, and unless this part of the label is U, the parameter set should be used only with the Hamiltonian for which it was developed. Solute Hamiltonian labels that have been defined so far are:

A	surface tension coefficients were developed specifically for use with AM1.
AM1	same as A (in other words, SM5.4/A and SM5.4/AM1 have the same meaning)
P	surface tension coefficients were developed specifically for use with PM3.
PM3	same as P (in other words, SM5.4/P and SM5.4/PM3 have the same meaning)
MNDO	surface tension coefficients developed specifically for use with MNDO.
U	surface tension coefficients are unspecific as to using either AM1 or PM3 as solute Hamiltonian.

The current preferred usage for AM1 and PM3 parameterizations is /AM1 and /PM3, but we used /A and /P in the earliest papers, so either notation is valid.

The U models in this version of AMSOL were developed with both CM1A and CM1P geometries and charges, and by reflecting a compromise between these two choices they are in principle less likely to be making up for errors that are specific to the use of a particular solute Hamiltonian.

In various sections of this manual, we often say only part of a name when that is all that is required for the point under discussion. For example, SM5.4 is a generic way to refer to all models having SM5.4 anywhere in the name. The meaning is usually clear from the context. Furthermore, SMx refers to all solvation models with SM anywhere in the name.

The SM5.4, SM5.0R, SM5.2R, and SM5.42R models compute the free energy of solvation in the solvent specified by the user. This specification is accomplished by the use of the IOFR, ALPHA, BETA, GAMMA and DIELEC keywords (Plus possibly the FEHALO and FACARB keywords) which input the index of refraction, alpha, beta, macroscopic surface tension, and dielectric constant of the desired solvent. (FEHALO is the fraction of non-hydrogenic atoms in the solvent which are F, Cl, or Br and FACARB is the fraction of non-hydrogenic atoms in the solvent which are aromatic carbons.) For more information on these solvent properties, see the respective keyword explanation. These models use the SM5 formalism with the additional feature that the macroscopic surface

tensions and solvent dielectric are functions of measured solvent properties, allowing the model to perform calculations over a wide range of solvents.

SM5.4 or SM5.4/AM1 or SM5.4/PM3 without specifying the solvent parameter set is defined to mean SM5.4-aqueous for water solvent, SM5.4-chloroform for chloroform solvent, SM5.4-benzene for benzene, SM5.4-toluene for toluene, and SM5.4-organic for all other solvents. (The use of SM5.4-organic parameters for chloroform, benzene, or toluene is to be considered incorrect.)

The SM5.0R model uses geometry-dependent atomic surface tensions (SM5 formalism) with implicit electrostatics (the generalized Born formula is not used) for calculating aqueous or organic free energies of solvation. As mentioned earlier, the R indicates a "rigid" model where the solvation free energy is predicted from an accurate gas-phase geometry and the structure is not allowed to relax within the solvent. The SM5.05R model is a simple extension of the SM5.0R model which allows the treatment of large molecules which just happen to contain charged groups, but is not intended for use when the site of interest within a solute molecule is charged.

Note that AMSOL does not include d orbitals; SMx models including d orbitals are available in other programs, namely, AMPAC, DGAUSS, GAMESS, and GAUSSIAN. SM5.2R parameterizations available in AMSOL are based on AM1, PM3, and MNDO. A version based on MNDO/d has been created for AMPAC, distributed by SemiChem Inc. Note that SM5.42R parameterizations in AMSOL are based on the NDDO Hamiltonians, AM1 and PM3. In addition to this pair of SM5.42R/NDDO parameterizations, SM5.42R/HF parameterizations based on ab initio Hartree-Fock models and SM5.42R/DFT parameterization based on density functional theory have also been developed. Such calculations are carried out with other computer codes, in particular GAMESS, DGAUSS, and GAUSSIAN.

The notation SRP is a generic name for any model that (as opposed to being a general parameterization for all compounds or a very wide group of compounds containing some subset of the elements in the periodic table) is proposed for a specific reaction or a specific range of solutes (hence the name, which means "specific reaction parameters" or "specific range parameters"). Thus, e.g., SM2-SRP may be used to denote any SRP model based on SM2 and it does not denote a unique set of parameters. SM4-SRP:x though denotes a specific set of parameters specified by x. In summary, SRP denotes a general type of approach, not a set of parameters and functional forms, whereas SRP:x denotes a specific set of parameters.

The CM1x notation is unique in the same way as the SMx notation for early SMx models. Thus CM1A is also called AM1-CM1A, but the prefix is unnecessary since, if the charge model is changed, the x in CM1x will be changed--not just the prefix. Furthermore CM1A may also be written CM1/AM1 and CM1P may also be written CM1/PM3. The CM2 parameterizations which utilize the AM1 or PM3 Hamiltonian are designated AM1-CM2 and PM3-CM2 respectively.

2.8. Availability of analytic gradients

In this version of AMSOL, the only model that has analytic gradients is SM5.4. Any other model, including SM5.42, that does not have a R in its name may be used to optimize structures in the liquid phase, but only with numerical gradients

3. BIBLIOGRAPHY

3.1. Suggested References

1. AMSOL code:

General reference:

AMSOL-version 7.0 by G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, J. Li, J. D. Thompson, P. Winget, B. J. Lynch, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar, University of Minnesota, Minneapolis, 2003, based in part on AMPAC-version 2.1 by D. A. Liotard, E. F. Healy, J. M. Ruiz, and M. J. S. Dewar.

If the EF optimizer (now the default for calculations that include geometry optimization) is used, add:

and on the EF routines by Frank Jensen.

2. AMPAC-Version 2.1 code:

AMPAC-version 2.1 by D. A. Liotard, E. F. Healy, J. M. Ruiz, and M. J. S. Dewar, Quantum Chemistry Program Exchange program no. 506, QCPE Bulletin 9, 123 (1989).

3. Van der Waals radii:

"Van der Waals Volumes and Radii" by A. Bondi, Journal of Physical Chemistry 68, 441-451 (1964).

4. EF method:

"Search for Stationary Points on Surfaces" by A. Banerjee, N. Adams, J. Simons, and R. Shepard, Journal of Physical Chemistry 89, 52-57 (1985).

"An Algorithm for the Location of Transition States," by J. Baker, Journal of Computational Chemistry 7, 385-395 (1986).

"A Quasi-Newton Algorithm for First-Order Saddle-Point Location," by P. Culot, G. Dive, V. J. Nguyen, and J. M. Ghuyssen, Theoret. Chim. Acta 82, 189-205 (1992).

"Transition State Optimizations by Trust-Region Image Minimization" T. Helgaker, Chem. Phys. Lett. 182, 503 (1991).

5. Original AM1 reference:

"AM1: A New General Purpose Quantum Mechanical Molecular Model" by M. J. S. Dewar, E. G. Zoebisch, E. F. Healy, and J. J. P. Stewart, Journal of the American Chemical Society 107, 3902-3909 (1985).

6. AM1 reference for halogens:

"Extension of AM1 to the Halogens" by M. J. S. Dewar and E. G. Zoebisch, Journal of Molecular Structure (Theochem) 180, 1-21 (1988).

7. AM1 reference for phosphorus:

"Extension of AM1 to the Halogens" by M. J. S. Dewar and C. Jie, *Journal of Molecular Structure (Theochem)* 187, 1-13 (1989).

8. Original PM3 reference:

"Optimization of Parameters for Semiempirical Methods II. Applications" by J. J. P. Stewart, *Journal of Computational Chemistry* 10 221-264 (1989).

9. AM1 reference for sulfur:

"AM1 Parameters for Sulfur" M. J. S. Dewar and Y-C Yuan, *Inorganic Chemistry* 29, 3881-3890 (1990).

10. SM1 and SM1a models for aqueous solvation:

"General Parameterized SCF Model for Free Energies of Solvation in Aqueous Solution" by C. J. Cramer and D. G. Truhlar, *Journal of the American Chemical Society* 113, 8305-8311, 9901(E) (1991).

11. SM2 model for aqueous solvation:

"An SCF Solvation Model for the Hydrophobic Effect and Absolute Free Energies of Solvation" by C. J. Cramer and D. G. Truhlar, *Science* 256, 213-217 (1992).

12. SM3 model for aqueous solvation:

"PM3-SM3: A General Parameterization for Including Aqueous Solvation Effects in the PM3 Molecular Orbital Model" by C. J. Cramer and D. G. Truhlar, *Journal of Computational Chemistry* 13, 1089-1097 (1992).

13. Long paper on SM1, SM1a, SM2, and SM3 models:

"AM1-SM2 and PM3-SM3 Parameterized SCF Solvation Models for Free Energies in Aqueous Solution" by C. J. Cramer and D. G. Truhlar, *Journal of Computer-Aided Molecular Design* 6, 629-666 (1992).

14. PM3 reference for lithium:

"Optimization and Application of Lithium Parameters for PM3" by E. Anders, R. Koch, and P. Freunscht, *Journal of Computational Chemistry* 14, 1301-1312 (1993). (Not all the PM3 lithium parameters are in this paper; when PM3 lithium parameters were added to AMSOL, the parameters missing in the published paper were taken from the MOPAC93 computer program by J. J. P. Stewart.)

15. AM1-SM4-SRP:Claisen model for aqueous solvation:

"Solvation Modeling in Aqueous and Nonaqueous Solvents: New Techniques and a Reexamination of the Claisen Rearrangement" by J. W. Storer, D. J. Giesen, G. D. Hawkins, G. C. Lynch, C. J. Cramer, D. G. Truhlar, and D. A. Liotard, *In Structure and Reactivity in Aqueous*

Solution: ACS Symposium Series 568, C. J. Cramer, D. G. Truhlar, Eds., American Chemical Society: Washington, DC, 24-49 (1994).

16. Details of the ASA algorithm, the force trapezoidal quadrature algorithm, and the SM2.1 and SM3.1 methods for water:

"Improved Methods for Semiempirical Solvation Models" by D. A. Liotard, G. D. Hawkins, G. C. Lynch, D. G. Truhlar, and C. J. Cramer, Journal of Computational Chemistry 16, 422-440 (1995).

17. Charge Models CM1A and CM1P:

"Class IV Charge Models: A New Semiempirical Approach in Quantum Chemistry" by J. W. Storer, D. J. Giesen, C. J. Cramer, and D. G. Truhlar, Journal of Computer-Aided Molecular Design 9, 87-110 (1995).

18. AM1-SM4 and PM3-SM4 models for solvation in n-hexadecane:

"A General Semiempirical Quantum Mechanical Solvation Model for Nonpolar Solvation Free Energies. n-Hexadecane" by D. J. Giesen, J. W. Storer, C. J. Cramer, and D. G. Truhlar, Journal of the American Chemical Society 117, 1057-1068 (1995).

19. AM1-SM4 and PM3-SM4 models for solvation in all other alkanes:

"A semiempirical Quantum Mechanical Model for Solvation Free Energies in All Alkane Solvents" by D. J. Giesen, C. J. Cramer, and D. G. Truhlar, Journal of Physical Chemistry 99, 7137-7146 (1995).

20. AM1-SM4-SRP:sugar model for aqueous solvation:

"Relative Stability of Alternative Chair Forms and Hydroxymethyl Conformations of D-glucopyranose" by S. E. Barrows, F. J. Dulles, C. J. Cramer, A. D. French, and D. G. Truhlar, Carbohydrate Research 276, 219-251 (1995).

21. Pairwise descreening and SM2.2 model for aqueous solvation:

"Pairwise Solute Descreening of Solute Charges from a Dielectric Medium" by G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, Chemical Physics Letters 246, 122-129 (1995).

22. Reference for SM5.4/U-aqueous, SM5.4/AM1-aqueous, and SM5.4/PM3-aqueous:

"A Model for Aqueous Solvation Based on Class IV Atomic Charges and First-Solvation-Shell Effects" by C. C. Chambers, G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, Journal of Physical Chemistry 100, 16385-16398, (1996).

23. Original reference (communication) for SM5.4/AM1-organic:

"A Universal Computational Model for Solvation Free Energies" by D. J. Giesen, M. Z. Gu, C. J. Cramer, and D. G. Truhlar, J. Org. Chem 61, 8720-8721 (1996) (Further details of this model are given in reference 24.)

24. Reference for further details of SM5.4/AM1-organic as well as for SM5.4/PM3-organic, SM5.4/AM1-benzene, SM5.4/PM3-benzene, SM5.4/AM1-toluene, and SM5.4/PM3-toluene:

"A Universal Model for the Quantum Mechanical Calculation of Free Energies of Solvation in Non-Aqueous Solvents" by D. J. Giesen, G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, *Theoretical Chemistry Accounts* 98, 85-109 (1997).

25. Reference for SM5.4/AM1-chloroform and SM5.4/PM3-chloroform:

"A Solvation Model for Chloroform Based on Class IV Atomic Charges" by C. C. Chambers, D. J. Giesen, C. J. Cramer, and D. G. Truhlar, *J. Phys. Chem.* 101, 2061-2069 (1997).

26. Reference for pairwise descreening and the SM2.2PD/AM1, SM5.4PD/AM1, SM5.4PD/PM3, SM5.4PD/U, SM5.2PD/AM1, and SM5.2PD/PM3 aqueous models:

"Parameterized Models of Aqueous Free Energies of Solvation Based on Pairwise Descreening of Solute Atomic Charges from a Dielectric Medium" by G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, *J. Phys. Chem.* 100, 19824-19839 (1996).

27. SM5.0R and SM5.05R aqueous models:

"Parameterized Model for Aqueous Free Energies of Solvation using Geometry-Dependent Atomic Surface Tensions with Implicit Electrostatics." by G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, *J. Phys. Chem. B* 101, 7147-7157 (1997).

28. SM5.0R universal solvation model:

G. D. Hawkins, D. A. Liotard, C. J. Cramer, and D. G. Truhlar, *J. Org. Chem.* 63, 4305-4313 (1998).

29. Reference for the analytical derivatives for the ASA algorithm:

"Computation of Hydration Free Energies using a Parameterized Continuum Model. Study of Equilibrium Geometries and Reactive Processes in Water Solution." I. Tunon, J. Bertran, M.F. Ruiz-Lopez and D. Rinaldi, *J. Comp. Chem.* 17, 148-155 (1996).

30. SM5.2R solvation model for water and organic solvents:

"A Universal Quantum Mechanical Model for Solvation Free Energies Based on Gas-Phase Geometries" by G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, *J. Phys. Chem. B.* 102, 3257-3271 (1998).

31. Reference for Charge Model 2:

"A New Class IV Charge Model for Extracting Accurate Partial Charges from Wave Functions" by J. Li, T. Zhu, C. J. Cramer, and D. G. Truhlar, *J. Phys. Chem. A.* 102, 1820-1831 (1998).

32. Reference for Charge Model 3:

"Parameterization of Charge Model 3 For AM1, PM3, BLYP, and B3LYP" by J. D. Thompson, C. J. Cramer, and D. G. Truhlar, J. Comput. Chem. submitted.

33. SM5.42R/AM1 and SM5.42R/PM3 for water and organic solvents:

"Extension of the Platform of Applicability of the SM5.42R Universal Solvation Model" by J. Li, T. Zhu, G. D. Hawkins, P. Winget, D. A. Liotard, C. J. Cramer, and D. G. Truhlar, Theor. Chem. Acc., 103, 9-63 (1999).

34. SM5.42R/AM1 and SM5.42R/PM3 parameterization for silicon in water and organic solvents:

"Parameterization of A Universal Solvation Model for Molecules Containing Silicon" by P. Winget, C. J. Cramer, and D. G. Truhlar, J. Phys. Chem. B, submitted.

3.2. Bibliography: Univ. of Minnesota Papers on Solvation Models

A bibliography of papers from the University of Minnesota that present or employ SMx solvation models is available. It also contains all errata (known to us) in our papers. It is located at the AMSOL home page on the World Wide Web. The URL is

<http://comp.chem.umn.edu/amsol/>

4. SMx AND CMx MODELS

4.1. Analyzing the NDDO Density Matrix

4.1.1. Population Analysis and Bond Orders

Call the molecular orbitals PSI and assume that the I-th molecular orbital is given by

$$\text{PSI}(I) = \text{Sum-over-P } \text{PHI}(P) * C(P, I) \quad (1)$$

where PHI is an atomic orbital; I, J, ... denote molecular orbitals; P, Q, denote atomic orbitals; and C(P, I) is a coefficient (here assumed real). Then the density matrix is

$$\text{RHO}(P, Q) = \text{Sum-over-I } N(I) * C(P, I) * C(Q, I) \quad (2)$$

where N(I) is the number of electrons in orbital I. In valence-electron-only neglect-of-differential-overlap (NDO) theories, like the AM1 and PM3 models included in AMSOL, the overlap matrix S(P, Q) is assumed to be the unit matrix. Thus

$$\text{Sum-over-P } \text{RHO}(P, P) = N \quad (3)$$

where N is the total number of electrons and the partial charge on atom A is

$$Q(A) = Z(A) - \text{Sum-over-(P on A)} \text{RHO}(P, P) \quad (4)$$

where Z(A) is the atomic number of A minus the number of core electrons.

We also note that although RHO(P, Q) just defined is usually called the density matrix, the true idempotent density matrix has elements

$$D(P, Q) = \text{Sum-over-Q } \text{RHO}(P, Q) * S(P, Q) \quad (5)$$

McIver and coworkers suggested calling RHO(P, Q) the population coefficient matrix instead of the density matrix to avoid the confusing situation.

4.1.2. Mulliken Analysis in the Zero-Overlap Approximation

Define the following symbols:

$$\text{NAP}(A) = \text{net atomic population on A} \quad (6)$$

$$\text{OP}(A, B) = \text{overlap population between A and B} \quad (7)$$

$$\text{GAP}(A) = \text{gross atomic population on A} \quad (8)$$

If we continue to interpret the overlap matrix S(P, Q) as the identity matrix, Mulliken population analysis is rather trivial and yields the following results:

$$\text{NAP}(A) = \text{Sum-over-(P on A)} \text{RHO}(P, P) \quad (9)$$

$$\text{OP}(A, B) = 0 \quad (10)$$

$$\text{GAP}(A) = \text{NAP}(A) \quad (11)$$

Thus,

$$Q(A) = Z(A) - GAP(A) \quad (12)$$

Equation (12) is the formula used for partial atomic charge in SM1, SM1a, SM2, SM2.1, SM2.2, SM3, and SM3.1, and it is labeled the Mulliken partial charge.

Some workers call equation (12) with equations (9) and (11) for GAP(A) the Coulson atomic charge because of its use in pi electron NDO theories by Coulson. Since it is a special case of the population analysis advanced by Mulliken in 1935 (and stated in more general language by him in 1955), we prefer to call it the Mulliken zero-overlap partial charge or -- for brevity -- the Mulliken partial charge. In any event, equations (2), (9), (11), and (12) completely define what we mean when we say NDDO Mulliken population analysis or zero-overlap population analysis. Since we always neglect overlap in the population analysis in SMx and CMx methods, no confusion should result from the fact that some of our papers just call this Mulliken analysis without emphasizing the neglect of overlap.

4.1.3. Mulliken Analysis in a De-Orthogonalized Basis

When overlap is not zero the Mulliken analysis is

$$NAP(A) = \text{Sum-over-(P on A)} \text{ Sum-over-(Q on A)} RHO(P,Q) S(P,Q) \quad (13)$$

$$OP(A,B) = \text{Sum-over-(P on A)} \text{ Sum-over-(Q on B)} 2 * S(P,Q) * RHO(P,Q) \quad (14)$$

$$GAP(A) = NAP(A) + 0.5 * \text{Sum-over-(B.NE.A)} OP(A,B) \quad (15)$$

We can equivalently write

$$GAP(A) = \text{Sum-over-(P on A)} D(P,P) \quad (16)$$

Equation (4) still gives the partial charges.

Pople and Segal and Giessner-Prettre and Pullman suggested that the MO coefficients C(P,I) resulting from an NDO calculation should be thought of as the coefficients of a symmetrically orthogonal set of atomic orbitals, and these coefficients can then be replaced by a new set of coefficients C'(P,I), which may be thought of as the coefficients of the original non-orthogonal atomic orbitals. One finds that

$$C'(P,I) = \text{Sum-over-Q} A(P,Q) C(Q,I) \quad (17)$$

where

$$A(P,Q) = S(P,Q) ** (-1/2) \quad (18)$$

Then one may carry out the Mulliken population analysis using the non-orthogonal formulation of equations (13)-(16). The MULLIK keyword in AMSOL invokes this method with S(P,Q) determined from the NDDO slater AO's.

Although we do not use the de-orthogonalized version of Mulliken analysis in any version of SMx or CMx, the AMSOL code will carry out this analysis. Note that this was not debugged in AMPAC-version 2.1, but it has been corrected in AMSOL starting with version 4.6.

4.1.4. Bond Order

Coulson originally defined bond order for the NDO case as

$$BO(A,B) = \text{Sum-over-(P on A)} \text{ Sum-over-(Q on B)} RHO(P,Q) \quad (19)$$

For the case of nonzero overlap, Chirgwin and Coulson introduced a definition equivalent to

$$BO(A,B) = \text{Sum-over-(P on A)} \text{ Sum-over-(Q on B)} D(P,Q) \quad (20)$$

The latter definition was also used by Ehrenson and Seltzer. These definitions are clearly closely related to population analysis.

A second way to measure bond order has been introduced based on squares of density matrix elements. The result yielded by such a scheme will be called the covalent bond index BI(A,B). For the case of NDO wave functions, the bond index was given by Wiberg as

$$BI(A,B) = \text{Sum-over-(P on A)} \text{ Sum-over-(Q on B)} \text{ RHO}(P,Q)**2 \quad (21)$$

This expression was elaborated by Armstrong et al. for NDO wave functions and extended to the nonzero overlap case by Mayer.

Covalent bond indices in the SMx models for NDO methods are based on eq. (21) without de-orthogonalization.

4.1.5. Bibliography

Population analysis:	1, 7-9, 16
Partial Charges:	2, 4, 5, 7-9
Bond order based on population analysis:	3, 5, 7, 15
Symmetric orthogonalization:	6, 10, 14
De-orthogonalization:	11, 12
Population analysis based on de-orthogonalization:	16
Covalent bond index:	13, 17, 18
Comparison of bond order and bond index definitions:	16

4.1.6. References

1. R.S. Mulliken, *J. Chem. Phys.* 3 (1935) 564.
2. C. A. Coulson, *Proc. Roy. Soc. (London)* A164 (1938) 383.
3. C. A. Coulson, *Proc. Roy. Soc. (London)* A169 (1939) 413.
4. C. A. Coulson and H. C. Louquet-Higgins, *Proc. Roy. Soc. (London)* A191 (1947) 39.
5. B. H. Chirgwin and C. A. Coulson, *Proc. Roy. Soc. (London)* A201 (1950) 196.
6. P.-O. Lowdin, *J. Chem. Phys.* 18 (1950) 365.
7. R.S. Mulliken, *J. Chem. Phys.* 23 (1955) 1833.
8. R.S. Mulliken, *J. Chem. Phys.* 23 (1955) 1841.
9. R.S. Mulliken, *J. Chem. Phys.* 36 (1962) 3428.
10. Fischer-Hjalmars, *Adv. Quantum Chem.* 2 (1965) 25.
11. J. A. Pople and G. A. Segal, *J. Chem. Phys.* 43 (1965) S136.
12. C. Giessner-Prettre and A. Pullman, *Theor. Chim. Acta* 11 (1968) 159.
13. K. B. Wiberg, *Tetrahedron* 24 (1968) 1083.
14. P.-O. Lowdin, *Adv. Quantum Chem.* 5 (1970) 185.
15. S. Ehrenson and S. Seltzer, *Theor. Chim. Acta* 20 (1971) 17.
16. J. W. McIver, Jr., P. Coppens, and D. Nowak, *Chem. Phys. Lett.* 11 (1971) 82.
17. D. R. Armstrong, P. G. Perkins, and J. J. P. Stewart, *J. Chem. Soc. Dalton Trans.* 1973 (1973) 838.
18. I. Mayer, *Chem. Phys. Lett.* 97 (1983) 270.

4.2. Geometry Optimization

4.2.1. Overview

The geometry optimizers included in AMSOL are of two general types, optimizers that are used to find a minimum energy structure and transition state optimizers that may be used to find the nearest stationary point, which may be either an energy minimum or a saddle point.

The following are the optimizers that can be used to find minimum-energy structures and the keywords that specify them:

Optimizer	Keyword
Eigenvector Following (default)	(none) or EFOLLOW
Trust	TRUSTE
Broyden-Fletcher-Goldfarb-Shanno	BFGS
Davidon-Fletcher-Powell	DFP
Newton algorithm on energy	NEWTON

The following are the optimizers that can be used to find transition state structures and the keywords that specify each optimizer:

Optimizer	Keyword
Eigenvector Following (default)	TSTATE
Trust	TRUSTG
Powell	POWELL
Bartels' Method (nonlinear least squares)	NLLSQ
Newton algorithm on gradient norm	LTRD
Chain	CHAIN
Saddle	SADDLE
McIver-Komornicki	SIGMA

4.2.1.1. Newton and quasi-Newton methods

All Newton (NEWTON and LTRD) and quasi-Newton (BFGS, DFP and EF) methods are based on a quadratic approximation of the energy based on a Taylor expansion around the geometry of the stationary point to the current iteration geometry. As the iterations proceed, one must in principle calculate the function, its gradient, and its Hessian at each iteration, i.e., at each step in geometry. In all methods in AMSOL, the energy and gradient are in fact calculated exactly at each step. In the original Newton method, the quadratic term in the expansion is also recalculated exactly at every step, and the formula for the step that minimizes the function (the Newton-Raphson step) is $X = -HG$, where H is the inverse Hessian matrix and G is the gradient vector. None of the methods in AMSOL is a pure Newton method; even when the NEWTON or LTRD keywords are used, the step is not always in the direction of the Newton-Raphson step, and when it is in that direction, in order to account for the fact that the quadratic region of the surface is limited, the Newton-Raphson step size is not set equal to $-HG$, but instead a line search along the direction defined by X is performed, in order to minimize the energy or the gradient respectively. Thus the various Newton and quasi-Newton methods in AMSOL differ both in the way the Hessian is calculated or approximated and in the algorithm for the step.

First consider the Hessian. When the NEWTON or LTRD keywords are used, the Hessian is recalculated at every iteration. These methods are reliable, but they are also very CPU-time expensive.

BFGS and DFP are called quasi-Newton methods because the Hessian matrix is approximated rather than calculated exactly. In particular, in DFP and BFGS, at the first step, the Hessian is assumed to be the unit matrix, and at later steps it is updated by a rank-one update on the basis of the value of the gradient. The Broyden-Fletcher-Goldfarb-Shanno [1-4] (BFGS) and Davidon-Fletcher-Powell (DFP) algorithms are two methods for the update, as described below. The BFGS method is described in the following references:

- [1] C. G. Broyden, *J. Inst. Math. Appl.* 6 (1970) 76;
- [2] R. Fletcher, *Comput. J.* 13 (1970) 317;
- [3] D. Goldfarb, *Math. Comp.* 24 (1970) 23;
- [4] D. F. Shanno, *Math. Comp.* 24 (1970) 647.

It is possible to force the system to replace the rank-one update with an alternative update every few steps by using the RECALC keyword. For BFGS and DFP it consists in setting the Hessian to the unit matrix, which can often be helpful if the effect of successive updates leads to a qualitatively unreasonable Hessian, as sometimes happens.

The difference between the BFGS method and the DFP methods is just in the formula used to estimate the Hessian matrix. The use of the DFP formula is considered to be less efficient than BFGS (see, e. g., L. E. Scales, *Introduction to Non-linear Optimization*; Springer-Verlag, New York, 1985).

Notice that the default method used in AMPAC-version 2.1 and in AMSOL-version 1.0 through 4.6 is DFP, and not BFGS as erroneously stated in some sections of the AMPAC-version 2.1 manual. In fact, AMPAC-version 2.1 does not even include the BFGS algorithm, but it has been added to AMSOL, beginning with version 5.0. Starting in AMSOL-version 5.0, the default was changed from DFP to the Eigenvector Following method (keyword: EFOLLOW), which is discussed in more detail in Section 4.2.2. Both the EFOLLOW and the DFP methods are fast, robust, and usually yield the same answer (± 0.005 kcal/mol in 95% of our trial comparisons) in similar computing times. The BFGS method also yields similar performance, and it should be preferred to DFP on the basis of the literature mentioned above. However, as explained in Section 4.2.2, the EFOLLOW algorithm has many options which can be tuned to yield convergence in hard-to-converge cases, and so it has now been made the default optimizer in AMSOL. In some cases, it is possible that one optimizer may converge when the other does not. For difficult-to-converge cases, it is recommended that both EFOLLOW and BFGS be tried before resorting to more time-consuming options. In preliminary trials, the TRUSTE optimization method by Liotard outperformed both the EF and BFGS methods.

Next consider the step. Choosing the step involves choosing its direction and choosing its magnitude. When the NEWTON or LTRD keyword is used, in order to avoid instabilities near inflection points, the step direction is adaptively selected from among three possible directions: the Newton-Raphson direction, the steepest descent direction, or the eigenvector associated with the lowest eigenvalue of the Hessian matrix. In all cases the step magnitude is determined by a third-degree polynomial extrapolation along the chosen direction. When the NEWTON keyword is used, the step size is selected in order to minimize the energy, and when the LTRD keyword is used, the step size is

selected in order to minimize the gradient norm, allowing the algorithm to locate transition states as well as minima.

The BFGS and DFP methods choose the step size by a line search to minimize the energy along the Newton-Raphson direction.

4.2.1.2. Other Methods

Next we consider the non-Newton/non-quasi-Newton methods.

Trust (TRUSTE and TRUSTG) is discussed in Section 4.2.3

EF is discussed in Section 4.2.2.

Powell (POWELL) and the McIver-Komornicki (SIGMA) and Bartels' (NLLSQ) algorithms are non-linear least square methods that here are used to minimize the gradient norm as a function of the geometrical parameters.

The SADDLE and CHAIN methods try to locate a transition state for a reaction by extrapolating its geometry based on the geometry of the reactant and product (SADDLE) or of reactants, products, and a trial transition state structure (CHAIN).

4.2.2. The Eigenvector Following Method for Optimization

The Eigenvector Following method is now the default geometry optimizer in AMSOL. The Eigenvector Following code included in AMSOL is based on code developed by:

Frank Jensen
Department of Chemistry
Odense University
5230 Odense, Denmark

The Eigenvector Following (EF) algorithm may be invoked by using either the EFOLLOW or TSTATE keyword, the former invoking EF to search for the nearest minimum and the latter invoking EF to search for the nearest transition state. If none of the optimizer keywords mentioned at the beginning of Section 4.2.1 is invoked, the code proceeds in the same way as if EFOLLOW were invoked.

4.2.2.1. Description of the EF Algorithm

The EF optimization routine used here is a combination of the original EF algorithm of Simons et al. [5] as implemented by Baker [6] and the QA algorithm of Culot et al. [7], with some added features (see RMIN, RMAX, and OMIN described below) for improving stability.

[5] A. Banerjee, N. Adams, J. Simons and R. Shepard, J. Phys. Chem. 89 (1985) 52.

[6] J. Baker, J. Comp. Chem. 7 (1985) 385.

[7] P. Culot, G. Dive, V. H. Nguyen and J. M. Ghuysen, Theo. Chim. Acta 82 (1992) 189.

[8] T Helgaker, Chem. Phys. Lett. 182 (1991) 503.

The version of EF used here is a modified port of the October 1992 version by Frank Jensen as included in the public domain MOPAC-7 project by J. J. P. Stewart. It has been updated January 1997.

The procedures followed in the Eigenvector Following methods (EFOLLOW and TSTATE keywords) are closely related to quasi-Newton methods, the main difference being that the Hessian is modified by

adding suitable shift parameters before calculation of the geometry step. Like the quasi-Newton methods, at each optimization iteration, the energy, the gradient, and some estimate of the Hessian associated with the current geometry are necessary in order to calculate the next optimization step. The default for EFOLLOW is to approximate the Hessian by the a diagonal matrix in internal coordinates at the first step. When the TSTATE keyword is invoked, the starting Hessian is instead calculated accurately by forward finite differences. For either EFOLLOW or TSTATE, the Hessian is then updated at subsequent steps by the formula specified by the IUPD keyword (see the keyword section). Like the quasi-Newton methods, one can force the system to replace the rank-one update with an alternative update every few steps by using the RECALC keyword. For EFOLLOW and TSTATE, the alternative update is a full Hessian calculation.

There are three fundamental operations in determining the next geometry:

- Find the "best" step within or on the hypersphere with the current trust radius.
- Possibly reject this step based on various criteria.
- Update the trust radius.

For a minimum search (keyword: EFOLLOW) the correct Hessian has only $3N-6$ positive eigenvalues, where N is the number of atoms. For a transition state search (KEYWORD: TSTATE) the correct Hessian should have exactly one negative eigenvalue, and the corresponding eigenvector should be in the direction of the desired reaction coordinate.

The geometry step is calculated as $g/(s-H)$, where s is a shift factor which ensures that the modified Hessian has the correct number of negative eigenvalues. If the unmodified Hessian has the correct structure, a pure Newton-Raphson step is first attempted. This corresponds to setting the shift factor to zero. If this step is longer than the trust radius, s is calculated using the RFO algorithm. If this second trial step is also too long, then the best step on the trust radius hypersphere is made via the QA formula. The pure NR step can be avoided by giving keyword NONR, in which case the RFO step is taken. The QA step can be avoided by giving the keyword RSCAL, in which case a too long RFO step is simply scaled down by a multiplicative factor. Specifying both NONR and RSCAL gives the original EF algorithm. The default, and recommended options, is not to specify NONR and RSCAL. For minimum searches there is only one shift parameter in the RFO and QA methods. For a TS search, the RFO method has two shift parameters, while QA only has one. The QA method for a TS search is identical to the TRIM method of Helgaker, thus the combined acronym QA/TRIM is normally used.

During an optimization, the maximum allowed stepsize (the "trust radius") is changed dynamically based on the ratio between the predicted and actual change in energy. For TS searches, the overlap between successive TS modes is also used. The upper and lower bounds for the trust radius are given by the DDMAX and DDMIN keywords. The default values are 0.5 (0.3 for TS search) and 0.01, respectively. Setting DDMAX=DDMIN effectively turns off the trust radius updating. The starting value for the trust radius is given by DMAX, =0.2 for EFOLLOW and =0.1 for TSTATE.

In order to further improve the stability of optimization, the ratio of the actual to the predicted energy change between two steps is required to stay between the RMIN and RMAX limits. If the second order

expansion of the energy is a correct approximation to the real surface, the ratio should ideally be 1. If the actual ratio deviates substantially from this value, the second order expansion is no longer accurate and neither is the optimization step. If the ratio is outside the interval defined by the RMIN and RMAX limits, then the step is rejected, the trust radius reduced by a factor of two and a new step is determined. The second criteria is that the eigenvector along which the energy is being maximized should not change substantially between iterations. The overlap of the transition state eigenvector with that of the previous iteration should be larger than OMIN, otherwise the step is rejected, the trust radius reduced by a factor of two, and a new step determined.

For a minimum search the default RMIN and RMAX are 0.0 and 1000, respectively. The RMIN value effectively prevents the energy from increasing, while very large drops in energy are permitted by the large RMAX value. For TS searches, the limits are RMIN=0.25 and RMAX=4.0 since TS searches are much harder to converge. Since the Hessian normally is approximate (updated) it is not guaranteed that there exist a step (however small) which satisfy the RMIN/RMAX and possibly OMIN criteria. Once the trust radius drops below the DDMIN value, the step is taken, regardless of the consequences. The energy may thus increase for a minimum search, even when RMIN=0.0.

After the geometry optimization step is determined, the energy, gradient, and Hessian at the new geometry are evaluated and this process is iterated until the convergence criterion is met.

4.2.2.2. How to Fine Tune EF Optimizations

4.2.2.2.1. RMIN and RMAX

The acceptance criterion for the optimization step is that the ratio of the calculated energy to the predicted energy should be larger than RMIN and lower than RMAX. If this ratio is outside this interval, the step is rejected, the trust radius reduced by a factor of two and a new step is predicted.

Setting RMIN and RMAX close to one will give increase the stability of the optimization, but also forcing very small steps to be taken, which slows down the optimization. Note that DDMIN may require adjustment also if tight criteria for RMIN/RMAX are used. Wide limits on RMIN and RMAX may in some cases give a faster convergence, but there is always the risk that very poor steps are accepted, causing the optimization to diverge. The default values for TSTATE are probably on the conservative side. If transition state searches are found to cause problems, we recommend changing the limits of RMIN and RMAX, for example, narrowing the interval to 0.5 and 2. Tighter limits like 0.8 and 1.2, or even 0.9 and 1.1, will almost always slow the optimization down significantly, but they may be necessary in some cases.

4.2.2.2.2. OMIN

OMIN has been designed for ensuring that the nature of the transition state mode only changes gradually, specifically the overlap between two successive geometrical displacement should be higher than OMIN. While this technique at first appears very desirable it may cause problems when the Hessian is updated. As the updated Hessian in each step is only approximately correct, there is an upper limit on how large the transition state mode overlap between steps can be. To understand this, consider a series of steps made from the same geometry (e.g. at

some point in the optimization), but with steadily smaller step sizes. The update adds corrections to the Hessian to make it a better approximation to the exact Hessian. As the step size becomes small, the updated Hessian converges toward the exact Hessian, at least in the direction of the step. The overlap between transition state modes does not converge toward 1, but rather to a constant value which indicates how good a guess the first approximate Hessian was to the exact Hessian. It appears that an updated Hessian in general is not of sufficient accuracy for reliably rejecting steps with transition state overlaps much greater than 0.80. The default OMIN of 0.80 reflects the typical use of an updated Hessian and allows fairly large changes to occur, and should be suitable for most uncomplicated systems. If the Hessian is recalculated in each step (HESS=1 or 3), the overlap will go towards 1 as the stepsize decreases, and OMIN may be increased in such cases to follow a given mode more closely.

If problems are encountered with many steps being rejected due to small transition state mode overlaps, try reducing OMIN, maybe all the way down to 0. This most likely will work if the transition state mode is the lowest Hessian eigenvector, but it is doubtful that it will produce any useful results if a high lying mode is followed.

Note that the only way to turn off the step rejection criteria is to give suitable values to RMIN, RMAX, and OMIN, e.g. the choice of RMIN=-100 and RMAX=100 effectively inhibits step rejection. Similarly setting OMIN=0 disables step rejection based on large changes in the structure of the transition state mode.

4.2.2.2.3. MODE

The algorithm has the capability of following a Hessian eigenvector other than the one with the lowest eigenvalue toward a transition state, using the keyword MODE=x. It is always more difficult to attempt such higher-mode following, and will in general only be possible in a stable fashion if the mode being followed is only weakly coupled to all lower modes (i.e. of different symmetry or spatially distant). As the optimization progresses, the TS mode must at some point become the lowest eigenvector. In order to do this, it will at some point become degenerate with the lower lying mode, at which point the direction of the mode being followed essentially is determined by the coupling between the modes (i.e. higher (>2) order derivatives).

Note that during transition state optimizations, the default value MODE=1 means that mode following is active. This means that the transition state MODE=1 will be followed, and in some cases this may eventually change to some higher mode, causing the optimization to fail. To turn off mode following, and thus following at every step the mode with lowest eigenvalue, set MODE=0.

Remember that following modes other than the one with the lowest eigenvalue toward a transition state indicates that the starting geometry is not a good guess of the transition state one. In most cases it is better to further refine the starting geometry, than to try following high-lying modes. There are cases, however, where it is very difficult to locate a starting geometry which has the desired Hessian, and higher-mode following may be useful.

4.2.2.2.4. HESS and RECALC

In certain very rigid systems or in some transition state optimizations, the initial default Hessian may be too approximate. In

such it may be useful to increase the precision of the Hessian by actually calculating it at the initial geometry using the HESS keyword.

In some very difficult cases, especially TS searches, it may be necessary to recalculate the Hessian every few iterations, using the RECALC keyword. Unfortunately, setting RECALC to low values is very expensive in terms of computer time. A RECALC=1 in combination with tighter values of RMIN and RMAX, and possibly also slightly lower DDMIN, will force almost all minimizations to converge (with the exceptions of discontinuous surfaces arising from numerical integration). This also represent the best Newton based method for locating transition states, but there is no guaranty that it will work! The basic problem is that a TS search require a direction along which to maximize the energy. In NR based methods, this is given as one of the Hessian eigenvectors, but this is only strictly valid near the stationary point.

4.2.3. The Trust Optimizer

TRUSTE and TRUSTG are minimization algorithms based on trust region methods with variable radius and an original estimate of the Hessian matrix. TRUSTE and TRUSTG use different criteria for minimization. TRUSTE minimizes the energy, whereas TRUSTG minimizes the gradient norm. Thus TRUSTE is suitable for optimizing stable species while TRUSTG is suitable for searching for any kind of stationary point. Both methods are robust in that the quantity being minimized is not allowed to increase during the iteration process, and the algorithm is almost insensitive to ill-behaved coordinates (e.g., redundancies in internal coordinates or the use of a full set of Cartesian coordinates). Convergence is reached when the largest component of the gradient falls below some threshold in absolute value.

The basic trust region strategy with variable radius is similar to the QA algorithm (Culot et al., Theoret. Chim. Acta, 82, 189 (1992)). However the rules governing the trust radius are not the strict rules of Fletcher, but rather they mimic a linear search without performing it.

The initial guess for the hessian is built from a 2-body expansion of the potential energy based on the 2-body components delivering the analytic molecular SCF gradient and accounting for the anisotropy of the electronic cloud shared by each atomic pair. The cost (in the context of the rest of the algorithm) is only one extra evaluation of the gradient. Additional refinements are achieved (in TRUSTG only) by finite differences at a cost of about one third of a normal hessian evaluation.

Hessian updates are based on the BFGS formula in the TRUSTE case and on Powell's formula in the TRUSTG case. However, the hessian is automatically reset (based on the 2-body guess) when the trust radius becomes too small or the quadratic prediction becomes too poor. The latter feature contributes to speed and robustness significantly.

Standard keywords (e.g., PRINT= , GNORM= , CYCLES= , TLIMIT=) are supported, as is the restart capability (RESTART keyword). There are no auxiliary keywords for the TRUSTE and TRUSTG algorithms.

4.2.4. Convergence of Geometry Optimizations

4.2.4.1. Convergence Criterion Employed in AMSOL

For geometry optimizations, the same convergence criterion is used for the EFOLLOW, TSTATE, TRUSTE, TRUSTG, BFGS, and DFP geometry optimization methods. All convergence tests (specifically, the Herberts, Peters, XPARAM, Heat-of-Formation, and Gradient-Norm convergence tests) used in versions prior to AMSOL version-5.2 for the four previously mentioned optimization methods have been removed. The

previous convergence tests have been replaced starting in AMSOL 5.3 by a two-step test that depends on the magnitude of the largest component of the gradient vector and the energy change for the most recent change of geometry. This two-step test ensures that all forces on the molecule are small at the point of convergence. Some previous convergence tests were based solely on the change in geometry (XPARAM) or solely on the change in total energy ("Heat of Formation"), and others (Herberts and Gradient Norm) consisted of what we consider to be less satisfactory tests on the gradient norm; these tests often resulted in "convergence" while the forces were still large on portions of the molecule. The test which is now used for all geometry optimizations requires that the largest component of the gradient is less than a user-controllable parameter called GCOMP, and that the energy calculated at the final geometry changed by less than 0.1 kcal/mol from the previous geometry.

The default value of GCOMP has been set to 0.45 kcal/RU where 1 RU = one Reduced Unit = 1 Angstrom or 1 radian, depending on whether the internal coordinate in question is a distance or an angle. The value of 0.45 kcal/RU was chosen because, when used in conjunction with the check that the energy changed less than 0.1 kcal/mol in the most recent geometry change, it gives nearly identical results (energies converged to within 0.005 kcal and very small deviations in geometry) to fully converged calculations (i.e., calculations in which the convergence criterion is set very small for test purposes), yet it yields significant computer time savings over more conservative criteria.

Convergence criteria for all other geometry optimization methods (NEWTON, POWELL, NLLSQ, LTRD, CHAIN, SADDLE, and SIGMA) are unchanged from AMPAC-version 2.1.

4.2.4.2. User Control of the Geometry Convergence Process

When using the EF, TRUSTE, BFGS, or DFP methods, the user has control over several aspects of the geometry convergence process through the keywords GCOMP=x, RECALC=x, CYCLES=x, and those keywords listed above in the Eigenvector Following method section (Section 4.2.2). The first convergence keyword has the form GCOMP=x. It can be used to raise or lower the convergence criterion discussed in Section 4.2.4.1. The second, RECALC=x, controls how often (every x cycles) the Hessian is recalculated in the EF, BFGS and DFP methods. In the Eigenvector Following method, the Hessian can further be controlled using the keyword HESS=n to specify the method for calculating the Hessian. In the BFGS and DFP methods, the Hessian is simply set to the unit matrix every x cycles. Finally, the user can control the total number of geometry optimization cycles (i.e., geometry steps, not to be confused with SCF cycles) before termination by the CYCLES=x keyword. The default limit on CYCLES is 100.

4.2.4.3. The Hessian Matrix

Eigenvector Following: There are four methods available for Hessian calculation that can be specified using the HESS=x keyword. The Hessian is calculated at the beginning of the optimization process using one of these methods:

- x=0 The Hessian is approximated as diagonal (see below)
- x=1 The Hessian is calculated by forward finite differences
- x=2 The Hessian is read from disk
- x=3 The Hessian is calculated by central finite differences

The diagonal Hessian used by the Eigenvector Following method uses values that are taken from the Hessian for an optimized cyclohexane molecule. In addition, the Hessian can be recalculated by the method specified by the HESS=x keyword every n cycles using the RECALC=n keyword. Between recalculations, the Hessian is updated every cycle using the method specified by the IUPD keyword (see the keyword section).

BFGS and DFP: There is only one method for calculating the Hessian in these methods. This method sets the Hessian equal to the unit matrix. The Hessian is set equal to the unit matrix at the beginning of the calculation and then updated at each cycle. The Hessian is reset to the unit matrix if the cosine of the angle between the vector defined by the search direction and the vector defined by the geometry is less than 0.05. The Hessian can also be reset using the RECALC=n keyword. In this case, the Hessian will be reset if n iterations have passed since the Hessian was last reset. The Hessian will also be reset after performing a "KICK" or before ending with a line minimization error.

For the EF, BFGS and DFP methods, restart files (file.inp, file.res, and file.den) are written before every recalculation of the Hessian except for the case when the BFGS or DFP optimization is ending with an error, in which case the restart files are written after the Hessian is reset.

4.2.4.4. Restarting Troublesome Jobs

Because the new convergence criterion, which examines only the components of the gradient, is tighter than those used by DFP in previous versions of AMSOL, it is possible that some jobs may be difficult to converge. In cases such as this, the user must decide what determines satisfactory convergence. To this end, the EF, TRUST, BFGS, and DFP geometry optimization methods print out a one line summary after each geometry optimization cycle. Included in this summary line is the total energy, largest component of the gradient, and a value describing the change in geometry during that step. For more information on this geometry value, see Section 4.2.4.5. By examining these lines, the user can determine whether the molecule has reached convergence. If it is decided that convergence has been reached before the gradient component test criterion has been met, the job can be terminated and then restarted from the restart files with new convergence limits (see Section 4.2.4.2) that will allow proper termination of the job. This should be done with caution, and only after both the heat of formation and geometry have remained stable for a number of steps. To help avoid wasting computer cycles, a message is printed every time the restart files are updated. The most efficient use of computer time occurs when the termination/restart is executed shortly after the restart files are updated. The time between restart file updates can be controlled with the TDUMP=x keyword, where x is the number of seconds between restart updates. Section 4.2.4.3 contains more information on restart file updates. If a job terminates with a line minimization error (BFGS or DFP), the Hessian is reset before writing restart files. Continuation of a job with the old Hessian would only lead to the same error occurring in the restarted job. In the case of an EF optimization, a new Hessian will be calculated at the beginning of the restart run, according to the method specified by the HESS=x keyword, if it is specified; if it is not, the default values for HESS will apply. If the restart files are updated after a user-defined "convergence" has been reached, the job may be restarted with the keyword 1SCF added to the

input file. This will result in single SCF cycle at the converged geometry and yield the appropriate .out and .arc files. Should a job be terminated by the user rather than the program, the restart files will be named as follows: file.inp = fort.18, file.res = fort.9, and file.den = fort.10. The user should rename these accordingly.

4.2.4.5. Geometry Change during Optimization

One of the values printed out each cycle by the EF, TRUST, BFGS and DFP geometry optimization methods is a quantity that describes the relative magnitude of the change in geometry in the previous cycle. This quantity is calculated in the following manner. Consider the coordinates of the molecule to define a vector called XPARAM, whose components are in the Reduced Units defined above. The geometry changes during a cycle by adding a new vector, called D, to the vector XPARAM. The relative change in geometry is then calculated as the norm of D divided by the norm of XPARAM. Typical values of this relative change in geometry at convergence are less than 0.0005.

4.2.4.6. Additional Notes on Geometry Optimization in AMSOL

The main reason for the EF method failing to converge is usually that the updating scheme for the Hessian produces a poor Hessian matrix. For a minimization, one can try the Powell update (IUPD=1) as an alternative to the default BFGS (IUPD=2). For a TS search, this is not an option, as only the Powell update allows negative eigenvalues. Alternatively, a more accurate Hessian can be used in a restart, by giving the keyword "HESS=x" and a proper x value.

The algorithm that calculates effective Born radii is numerical rather than analytical, and thus it has discontinuities. This has almost no effect on the final geometry and energy; however it occasionally causes the final stage of the optimization to fail when using the BFGS or DFP minimization methods. This occurs primarily when the faster convergence strategy CS1 is employed. If this happens the warning "LINE MINIMIZATION FAILED TWICE IN A ROW. TAKE CARE" will be printed in the output files. This error does not appear to be a cause for worry in uncharged systems unless the gradient norm is unusually high. For neutrals, if the gradient norm is less than about 5 kcal/Angstrom (or radian), the calculation is probably acceptable. Aromatics seem to give higher gradient norms (even in the gas phase) when the calculation is acceptable. Use of the default KICK=1 option should eliminate most of these errors. If the KICK option is used and a LINE MINIMIZATION error still occurs, it is possible that the geometry is still in the perturbed state, thus great caution should be used while interpreting the results. Fortunately, LINE MINIMIZATION errors appear to be extremely rare when using the KICK option. Care should be taken when checking the gradient norms that a geometry change has actually occurred when the LINE MINIMIZATION error is obtained. If there is no change, try perturbing the initial starting geometry slightly to avoid the LINE MINIMIZATION error.

Ions pose particular problems since discontinuities in the solvation free energy can sometimes fool the optimizer into thinking the geometry has converged when it has not. While the deviation from the optimized geometry tends to be fairly small, the solvation free energy for an ion is much larger than for a neutral compound, and that can cause small deviations in geometry to significantly affect the solvation free energy. One way to assess whether this might be a problem is to start from two slightly (or appreciably) different geometries, and

verify that each gives essentially the same result. In those rare instances when they disagree, additional calculations from slightly different geometries or using other convergence methods are usually sufficient to identify which is the correct result.

The initial starting geometry can be perturbed by changing one of the bond lengths of one of the heavier atoms by about 0.2 Angstrom or several coordinates by 0.1 Angstrom. In particular, if the aqueous geometry optimization does not converge well when the gas-phase geometry is used as the initial guess, it may be helpful to start from a significantly perturbed geometry. Additionally, use of the Powell routine (keyword POWELL) to optimize geometries often avoids this problem. Moreover, the Powell method is quite useful for locating transition states since it typically optimizes to the nearest stationary point on the hypersurface. Note though that since the Hessian matrix is obtained by numerical differentiation, care should be exercised in verifying the nature of the stationary points that the Powell algorithm finds. In addition, there are occasions where the Powell method is very slow to converge, so it is usually better employed as an alternative than as the default.

Numerical derivatives used in both the DERISA and DERINU options for gas-phase geometry optimizations are calculated by central finite differences if the FORWRD keyword is not specified. If the keyword FORWRD is used, one-sided numerical derivatives are employed.

Whenever it is important for the results to be as independent as possible of the starting geometry of an optimization or to be as continuous as possible functions of reaction-path coordinates, the user can use the EXTM option (which is available in conjunction with all models based on the TR algorithm - see Section 2.6) to force the M values of the radial quadrature to be fixed for a run or a sequence of runs. If the user fixes the values of M to be 1 or more units larger than the values obtained by default for a trial run at a similar geometry, no loss of accuracy should occur, but the numerical continuity of geometries along a reaction path will be much better guaranteed.

4.2.5. Suggested Strategy for Geometry Optimizations

The following is a suggested strategy for obtaining optimized geometries even in difficult to converge cases. The suggestions represent the experiences of individual users who have used the AMSOL package extensively. It is possible that with additional experience a user may find a scheme which works better on the particular set of molecules under consideration. This scheme was developed using a large set of molecules with a balanced variety of functional groups. The recommended strategy is to try the optimizers in the order listed below until one of them leads to good convergence.

4.2.5.1. Strategy for Minima

- 1) Begin with the default EF optimizer, using its default settings and the best available guess for the geometry.
- 2) Try the TRUSTE optimizer.
- 3) Use the EF optimizer, but calculate the Hessian matrix by central finite differences (keyword HESS=3) and manually perturb the starting geometry from the one used in the previous attempt. Generally, shortening one of the bond lengths involving two nonhydrogenic atoms by approximately 0.05 Angstroms has been the most successful choice.

- 4) Try using the BFGS optimizer (keyword BFGS) on the original best guess geometry.
- 5) Use the BFGS optimizer and manually perturb a nonhydrogenic bond length in the starting geometry by about 0.1-0.2 Angstrom.
- 6) Try the EF optimizer using HESS=3 and manually perturb more than one of the bond lengths or perturb different bond lengths than that changed in step 2.
- 7) Use the EF optimizer, but have the Hessian calculated by forward finite differences (HESS=1). This can be tried with either the original best guess geometry or with a perturbed geometry.
- 8) Change the default values of RMIN and RMAX.
- 9) Try the DFP optimizer.
- 10) Use the GRADIENTS keyword to determine the portion of the molecule which is still experiencing large forces. Try to re-write the Z matrix to allow for more efficient optimization. This strategy can be used with any of the optimizers.
- 11) Set the keyword RECALC= to a number between 1/2 and 3/4 of the number of cycles taken before an error occurred and attempt to optimize the geometry. This method is most useful with the EF optimizer, but it can be used with any optimizer.

Experienced users have found that options 1-4 nearly always lead to a successful optimization, however if problems are encountered there are a few other tricks which have proved successful on occasion.

- If the molecule of interest will optimize when using a solvation model other than the method desired, try that final geometry as a starting point in the desired solvation calculation.
- Try perturbing a bond angle by five degrees rather than changing a bond length.

4.2.5.2. Strategy for Transition States

- 1) Begin with the EF optimizer, using the TSTATE keyword rather than the EF keyword (see Sections 4.2.2.1 and 6.3.65).
- 2) Try the TRUSTG optimizer. If problems are encountered and you believe the geometry may be very close to that for the transition state, try perturbing one of the heavy atom bond lengths by 0.05 angstroms and re-optimize.
- 3) In conjunction with the TSTATE keyword, try adjusting the value of the OMIN keyword (see Section 6.3.40) or recalculating the Hessian with suitable intervals.
- 4) Try using the keyword POWELL.
- 5) Try using CHAIN.

5. PROGRAM OVERVIEW

5.1. Description and Organization of Code

AMSOL is adapted from AMPAC-version 2.1. The major enhancements in this version of AMSOL as compared to AMPAC-version 2.1 are

- (1) The PM3 parameter set has been added, both for the original elements and for Li (references in Section 3 of this manual).
- (2) Several SMx solvation models (references in Section 3 of this manual) have been added.
- (3) The CM1, CM2, and CM3 charge models (references in Section 3 of this manual) have been added for gas-phase calculations.
- (4) The code has been made portable for a variety of Unix environments.
- (5) The SCF convergence strategy has been improved.
- (6) The TRUST, EF, and BFGS geometry optimization algorithms have been added, a convergence-promoting option has been added to the DFP optimizer as well as the new ones, and an improved convergence criterion for geometry optimization has been implemented for all optimizers.
- (7) The MULLIK keyword has been debugged.

AMSOL with either the AM1 or PM3 keyword calculates gas-phase electronic energies and, optionally, optimized geometries by the SCF method with the neglect of diatomic differential overlap and a parameterized Fock matrix with specific reaction parameters.

AMSOL with a Solvation Model keyword (see below) performs calculations in which free energy of solvation terms have been added to the solute Hamiltonian, thereby delivering electronic structures and, optionally, optimized geometries which incorporate solvation effects. AMSOL with the CM1A, CM1P, or CM2 keywords calculates class IV charges from Mulliken population analyses.

We especially note that in this version of AMSOL (in fact any version from 3.0 onward) differs from AMPAC-version 2.1 and AMSOL-version 1.0 in that it is written in very portable FORTRAN 77. (The only extension to FORTRAN77 that is employed is the INCLUDE extension, and the only nonstandard usage is lower case file names.) Thus the program is fully in DOUBLE PRECISION, all occurrences of the same common block have the same length, a more portable version of the block diagonalization routine is included, the timing calls have been reduced to a form that allows machine-dependent timing routines to be used conveniently, and many additional non-standard-usage and problems in AMPAC-version 2.1 which do not deal with portability issues have been corrected or improved.

To make AMSOL more portable, the INCLUDE statement has been changed as compared to AMPAC-version 2.1. In the present version, the line (which occurs in many subprograms)

```
INCLUDE "SIZES"
```

is replaced by

```
INCLUDE 'SIZES.i'
```

wherever it occurs. Since these changes were made globally throughout the program, subprograms which only have these changes are not treated as modified for portability. All the other modifications (those made for portability or those made to include the solvation effects) are indicated by initials and a date in columns 73-80.

Two variables that are set in the SIZES.i file may be used to control the maximum number of heavy and light atoms. In particular MAXHEV is the maximum number of heavy atoms (non-hydrogen atoms) that will be allowed in an AMSOL run, and MAXLIT is the maximum number of hydrogen atoms that will be used. These two variables are used to set dimensions for arrays involved in the calculations. If these numbers are set too low (lower than the number of atoms input in a .dat file), the code will stop and a message explaining how to reset the values for these variables will be written in the .out file. Once the values have been changed in the SIZES.i include file, the AMSOL code must be recompiled in order to update the executable. In the distributed package of this version of AMSOL, MAXHEV has a value of 40 and MAXLIT has a value of 60.

5.2. Files in Distribution

This version of AMSOL constitutes a single package that consists of the following directories and files:

doc/

This directory contains the following documentation files:

File	Description
amsol.doc	This is the present "on-line manual" (ASCII documentation file).
amsol.pdf	This is the AMSOL manual in pdf form.
search.man	This file is a C shell script which will search the amsol.doc file and pull out every paragraph which contains the given search string. Note: This script is designed to use the grep command with the -p option.
ampacdoc.pdf	This is a pdf manual for AMPAC-version 2.1.

include/

This directory contains the following files that are used in conjunction with the FORTRAN INCLUDE statement

File	Description
SIZES.i and PARAM.i	These are files from AMPAC that are required in compilation because they are referenced by multiple INCLUDE statements. The file "SIZES.i" has two additional lines compared to AMPAC-version 2.1 (assigning the parameters NPACK and TDEF); "PARAM.i" has been extended to incorporate the PM3 Hamiltonian parameters. While the INCLUDE statement is not standard FORTRAN 77, it is so useful that we have elected to retain it for the

convenience of users who may wish to change dimension limits or atom parameters. If the local host environment does not support INCLUDE, then SIZES.i, SIZES2.i, PARAM.i, PARAMS.i, and possibly SIZESc.i will simply (laboriously) have to be inserted into the source code at each occurrence of an INCLUDE statement. The variable TDEF in SIZES.i is the default value for the TLIMIT=x keyword; thus it is the maximum CPU time allowed for a job in which TLIMIT=x is not mentioned. TDEF is set equal to 65000 in the distributed version. This is large enough to run all 75 test runs on all machines listed under this version of AMSOL in table 1 of Section 13. The user may easily change TDEF by using the TLIMIT=x keyword. See page 6-38.

- PARAMS.i This INCLUDE file contains parameters that are used in the SMx models, those used for the CM1 charge models, and those used for solvent properties.
- SIZES2.i This INCLUDE file includes parameters that should be used to make dimensions of arrays greater than or equal to the maximum number of points that will be allowed on any sphere in the surface area calculations. It also includes parameters that determine the extent to which information should be saved for each surface area calculation where the coordinates of each atom have not changed. Finally, it includes a common block that contains the value of PI and several commonly used multiples of PI. The variable MXPT in SIZES2.i is the maximum number of points that will be distributed on the surface of each sphere for either the surface tension or the coulombic portion of the solvation energy calculation. The parameters MXPTSV and IATMSV are the maximum number of points that will be distributed on the surface of any sphere in the surface area calculation or the maximum number of atoms in a molecule, respectively, for which all reusable information in the distance calculation between a point on the surface of any sphere and all connected spheres is stored in an array. If the number of points on each sphere is less than MXPTSV and the number of atoms in the molecule are less than IATMSV, then certain pieces of information needed for the distance calculation are stored in a three-dimensional array that is MXPTSV by IATMSV by IATMSV. If this method is not used, the calculation will generally take 1-2% longer to complete.

mach/

This directory contains the following machine-dependent files

File	Description
util.f	This file contains the utility and header subprograms needed for AMSOL.
datesv.cray	This is the date/time subprogram for the Cray computers.
datesv.dec	This is the date/time subprogram for the DEC 3000 AXP workstation.
datesv.ibm	This is the date/time subprogram for the IBM RS/6000 workstation.
dateclock.c	This is a C subprogram which is used by datesv.ibm to obtain the date and time.
dateclock1.c	This is a C subprogram used when the AMSOL code is compiled for Linux to obtain the date and time.
datesv.iris	This is the date/time subprogram for the IRIS workstations.
datesv.sun	This is the date/time subprogram for the SUN workstations.
porcpu.cray	This subprogram determines the CPU time for the Cray computers.
porcpu.dec	This subprogram determines the CPU time for the DEC 3000 AXP workstation.
porcpu.ibm	This subprogram determines the CPU time for the IBM RS/6000 workstation.
porcpu.iris	This subprogram determines the CPU time for the IRIS workstations.
porcpu.sun	This subprogram determines the CPU time for the SUN workstations.
second1.c	This is a C subprogram used when compiling the code for Linux.

new/

This directory contains a series of FORTRAN77 files; each one is an individual subroutine. The files in this directory are either new code introduced as AMSOL subroutines, or subroutines taken from other sources and modified for reasons other than portability.

obj/

This directory is used during the compilation process to store object files.

port/

This directory contains a series of FORTRAN77 files, each one is an individual subroutine. The files in this directory were originally taken from other sources and modified for portability purposes.

PC/

This directory contains the files required to compile AMSOL on Windows. It includes the date and time routines, Include files, Makefile, and developer project file.

test/

This directory contains scripts to run AMSOL, test input files, EXTSM files that can be modified for the input of external parameters, and a file of solvent properties for use with SM5.4-organic calculations. Files included are:

File	Description
SMx.i	These files contain the preset parameters for the appropriate SMx model (where x denotes a specific model) which can be altered using the EXTSM keyword.
amsoli.c	This file is a C shell script for interactive execution of the AMSOL program on a computer where Unix does not support the ASSIGN statement.
envaq.c	This file is a C shell script that writes the code numbers for SM1a atom types to standard output (your window if run interactively). It is useful to be able to remind oneself quickly of these values when preparing .dat files for input for SM1a runs see page 6-15.
solv.prp	This file contains preformatted keyword lines which provide the solvent properties needed to run SM5.4-organic calculations for many common organic solvents. To use a line from the file, include the keywords for the desired solvent with the other keywords in the .dat or .xkw file.
solvR.prp	This file contains preformatted keyword lines which provide the solvent properties needed to run any of the SM5.*R models for predicting free energies of solvation in organic solvents. To use a line from the file, include the keywords for the desired solvent with the other keywords in the .dat or .xkw file. (Note this differs from the values in the solv.prp file by the inclusion of the FACARB=x.xx and FEHALO=x.xx keywords.)
solv.txt	This file contains our most comprehensive list of solvent descriptors. The values in this list can easily be converted to the keywords necessary for any of the models capable of predicting free energies of solvation in an organic solvent.
trxx.dat	This is the data file for test suite run xx. An additional 'a' after the number indicates an aqueous solvation run, and an addition 'n' indicates a non-aqueous solvation run. A total of 106 runs constitute the test suite.
trxCM3y.dat	This is the data file for test suite run x, a gas-phase CM3 calculation using the AM1 (y = A) or PM3 (y = P) method
trxx.out	This is the output file for test suite run xx resulting from a run on a IBM SP Power3 supercomputer.
trxCM3y.out	This is the output file for test suite run x resulting from a run on a IBM SP Power3 supercomputer.
trxx.xsm	This is the eXternal Solvation Model file for test suite

run xx. It is used in conjunction with the EXTSM option.

trxx.xkw This is the eXternal KeyWord file for the test suite run
xx. It is used in conjunction with the XKW option.

unmod/

This directory contains FORTRAN77 files that were taken from other sources and not modified. It contains the following files:

ampac_unmod.f This file contains all the subprograms from AMPAC-version 2.1 which have not been modified.
 ef_lib.f This file contains public domain math routines (from LINPACK and Level-3 BLAS) that are used in the EF algorithm.
 trust_unmod.f This file contains all the subprograms required to implement the TRUST algorithm that have not been modified.

5.3. Execution

There is a C shell execution script included in the test directory of the AMSOL package.

amsoli.c - This script is used for interactive execution of the AMSOL program if the UNIX system utilized does not support the ASSIGN statement.

This run script must be customized to work within your directory structure. The following statements must be changed to configure the script.

```
set username = "username"
```

"username" should be changed to your user name on the computer on which AMSOL will be run.

```
set amsolexe = /fullpathnamefortheextcutablefile
```

The path name for the executable needs to be placed on the right of the equal sign. NOTE: only RELATIVE PATHNAMES are allowed. In other words, if the AMSOL package is used as distributed, where the amsol.exe is compiled in the src directory and the script is in the test directory, then the set statement should be changed as follows:

```
set amsolexe = ../src/amsol.exe
```

The pathname of the EXTSM file needs to be placed on the right side of the equal sign.

```
set extsmfile = /fullpathnamefortheEXTSMfile
```

For more information of the EXTSM option, see the keyword section of the manual. If you do not use the EXTSM option the line can be changed as follows:

```
set extsmfile = EXTSM.file
```

As long as the EXTSM keyword is not used, or the EXTSM.file is empty, the code will use the internal solvation model parameters requested in the keyword line.

```
set xkwfile = /setfullpathnamefortheXKWfile
```

The pathname of the XKW file needs to be placed on the right side of the equal sign. For more information of the XKW option, see the keyword section of the manual. If you do not use the XKW keyword the line can be set as follows:

```
set xkwfile = XKW.file
```

As long as the XKW keyword is not used, or the XKW.file is empty, the only keywords used will be those specified on the keyword line(s) inside the .dat file. Relative pathnames must be used with the amsoli.c script and absolute pathnames must be used with the amsolb.c script.

```
set workd = `pwd`
```

This set statement does not need to be changed unless the .dat files are not located in the same directory as that from which the script command is executed. If the .dat files are located elsewhere, the pathname of the directory containing the .dat files needs to be placed on the right side of the equal sign.

When using the run-script, only the root name of the data file is used. For example to start a run with the tr1.dat data file, type:

```
amsoli.c tr1
```

not

```
amsoli.c tr1.dat
```

If the .dat extension is accidentally added, the script will prompt with a message indicating that the .dat extension should not be added.

Note that the run-scripts supplied with this version of AMSOL refers to device fort.9, not fort.09. The former is recognizable as a legitimate FORTRAN device, while the latter, used in the scripts supplied with version 1.0 of AMSOL, is not, and is treated as a normal file name.

5.4. Installation

The C shell script amsol.compile has been included to install AMSOL on any of the Unix or Cygwin computers for which the code has been tested. The script assumes that the directory and file structure is set up as explained in Section 5.2, and it can be used to compile AMSOL using several different methods. When amsol.compile is invoked, the user is prompted to choose 'auto'matic or 'man'ual installation. If the user chooses 'auto' or simply types enter, the script will attempt to determine the proper compiler options for the system and compile the source code.

If the user types 'man', the manual installation script begins and the user is prompted for the appropriate machine type for compilation, and also for the executable name. The script will also initiate manual installation if the automatic setup fails.

In the next step of manual installation, the user is given options to control the type of compilation that occurs. The user can choose either a 's'tandard optimization (all subroutines are compiled and

linked into the executable), or the generation of a 'm'akefile. The standard optimization is recommended for nearly all users; the makefile is only recommended for those users familiar with makefiles who also need to compile AMSOL on a regular basis. Additionally, when the standard option is chosen, the user may specify whether object files are 'h'eld after compilation (default) or 'n'eatly removed. IBM users are given an additional choice - whether or not their machine uses a FORTRAN preprocessor. (Note that this option will not work if the Fortran preprocessor, fppv, is not located in the directory '/usr/lpp/xlf/bin/'.) The script also contains a choice to compile the script for Linux.

5.4.1. FORTRAN compiler comments

When compiling AMSOL on some of the machines for which it has been tested, the compiler issues warnings and other observations about the optimizations being completed. What follows is a listing of the warnings and compiler comments we obtained when testing AMSOL on various computers, along with the approximate number of occurrences of each warning. The warnings listed below have not been found to cause problems with any AMSOL calculation.

Machine	Type of comment or warning	Approx. # of occur.
Compaq ES40 Alpha 500	No warnings	
Cray C90	Exponentiation replaced with multiplication	2
	Array of length 1 equivalenced to a common block	16
IRIS Power Challenge (R8000, R10000)	Overlapping library definitions	20
	Library not used	1
IRIS Indigo R4000	READS and WRITES subroutines are not optimized*	1
	Lines of code cannot be reached	2
	Length of common block /SCRCHR/ doesn't match**	1
SGI Origin 2000 (R10000, R12000, R14000)	Library not used	1
IBM RS6000 model 550 & 590	No warnings	
IBM SP Power3	No warnings	
IBM Regatta Power 4	No warnings	
Sun	Unused local variables	5
SPARCStation	Lines of code cannot be reached	2
IPX	Length of common block /SCRCHR/ doesn't match**	1
Sun Blade 2000	make: warning: Clock skew detected. Your build may be incomplete	1
DEC 3000/500X AXP	No path to this line	3
Linux	Missing comma in FORMAT statement	2

* The memory configuration on the IRIS Indigo R4000 used was not sufficient to fully optimize the subroutines.

** The code is always compiled using static memory options. The common block /SCRCHR/ is used as scratch space, and values contained within The block are only used locally and not passed between subroutines.

6. USAGE AND KEYWORD EXPLANATIONS

6.1. General Usage Comments

To run AMSOL in a UNIX or LINUX environment, the user must type a command having the following form:

```
~/amsol7.1/amsol.exe < input-file.dat > output-file.out
```

where "~/amsol7.1/amsol.exe" is replaced by the directory path to the amsol executable file and the name of the executable file, "input-file.dat" is replaced by the name of the input file, and "output-file.out" is replaced by the name of the output file. (Note that including the "<" and ">" symbols is essential.)

The user is assumed to be familiar with the usage of AMPAC-version 2.1, and a Postscript copy of the AMPAC-version 2.1 manual is distributed as part of the AMSOL package for this purpose.

AMPAC calculates total electronic energies (i.e., electronic energy plus nuclear repulsion) by semiempirical molecular orbital theory. These are output in two forms--as energies and as "heats of formation". However, it should be noted that in AMPAC, AMSOL, and MOPAC, the only difference between the "heat of formation" and the sum of the electronic energy plus nuclear repulsion is the choice of the zero of energy. In other words, the standard semiempirical packages compute the heat of formation by adding a constant to the sum of the electronic energy and the nuclear repulsion; this constant depends only on the number of each kind of atom. For example, the constant is the same for ethanol and dimethyl ether.

As the above explanation makes clear, the term "heat of formation" in the standard semiempirical packages is really a misnomer (i.e., differences in zero-point energies, thermal vibrational excitation energies, etc. are ignored). The historical origin of this confusion is that, in the original parameterization of the semiempirical molecular orbital parameters, differences of the sum of the electronic energy and nuclear repulsion were fit using differences of experimentally determined heats of formation because these were the most readily available experimental data. In this manual and in the output files of AMSOL, when we say "heat of formation" we are following this standard convention, i.e., heats of formation are the same as the sum of the electronic energy and nuclear repulsion with the conventional semiempirical choice of the zero of energy.

AMSOL is a modified version of AMPAC-version 2.1 that, in addition to carrying out electronic energy calculations, also has some new features. Primary among these is that it can also compute free energies of solvation and class IV charges. The latter may be computed for either the gas-phase or condensed-phase wavefunctions (CM1 and CM2, but not CM3). Modifications to the AMPAC-version 2.1 program are invoked by any of the new keywords listed in Section 6.3, including the keywords that indicate a specific solvation model, an improved convergence strategy for SCF cycles, and/or the new partial charge method (Charge Model 1), or they are invoked by performing any calculation involving geometry optimization.

The rest of this section and Section 6.2 provide an overview of the options and keywords. More details on the options are available under the keywords themselves, which are listed alphabetically in Section 6.3.

The keyword CM1 specifies that the Charge Model 1 mapping of the Mulliken population analysis should be used. The CM1 partial atomic charges and the dipole moment resulting from these point charges are

printed out in addition to the regular NDDO Mulliken population analysis results. The CM1 keyword is valid for either AM1- or PM3- based calculations and is implicit for SM4 solution-phase calculations.

For the solvation models specified by the SM1, SM1a, SM2, and SM3 keywords, the default radial integration scheme for the dielectric screening computation (i.e., the computation of the atomic effective Coulomb radii) is the force rectangle algorithm. These models were all parameterized using this algorithm, and the small systematic quadrature bias present was absorbed into the parameterization. The default radial integration scheme for all other non-pairwise descreening method is the force trapezoidal algorithm. The SM2.1 (SM3.1) parameter set was obtained by using this algorithm and fitting to results obtained with the SM2 (SM3) parameter set and the original radial quadrature. When using the pairwise descreening approximation of SM2.2 and the PD models, the radial integration does not need to be calculated numerically and is therefore calculated analytically. It is possible to run any parameter set with any of the radial quadratures using the "RAD=x" keyword. "RAD=0" causes the force rectangle quadrature to be utilized, and "RAD=1" causes the force trapezoidal algorithm to be used, and "RAD=2" causes the pairwise descreening approximation to be used and the integration is done analytically. Since combinations other than the default combinations are of dubious value (SMx parameter sets should be used with the radial quadrature for which they were optimized), they should be used only by developers, and the results should be interpreted with care. Three keywords, TEXPN, TONE, and EXTM, are available to control details of the numerical radial integrations.

When using the SM4 keyword or any SM5 keyword, the solvent must be specified using the SOLVNT keyword. When performing a General Alkane calculation ("SOLVNT=GENALK"), the keyword MSURFT can be used to input a macroscopic surface tension to be used for calculating the CS surface tension. Several keywords are available to change solvent characteristics when performing any solvation calculation. See the descriptions of keywords DIELEC, SVCDRD, and SVCSR. D.

If no solvation model is specified, i.e., when a gas-phase calculation is performed, one of two options, OPT or 1SCF, is used to control SCF cycling and geometry optimizations. If a solvation model is specified, one of the three options, OPT, NOPOL, or 1SCF, is used to control SCF cycling and geometry optimization. If none of these keywords is explicitly stated, the default keyword OPT is used.

Geometry optimizations involve gradient calculations at every step. The default gradient algorithm for gas-phase calculations in AMSOL, which is also the default in AMPAC-version 2.1 and can be explicitly requested for gas-phase calculations in AMSOL with the keyword DERISA, calculates each gradient as the sum of all of the contributions from all possible pairwise combinations of atoms; the matrix operations are considerably simplified by restricting them to two-atom combinations. However, for solvation calculations in AMSOL the individual gradients CANNOT be calculated in this manner. Instead, energy calculations must be made for the entire molecule with each movement of an atom, and the gradient then calculated as a numerical derivative. The keyword for using numerical derivatives, DERINU, accomplishes this. (Note: As in AMPAC-version 2.1, DERINU can also be used for gas-phase calculations.)

If no keyword is used to specify a geometry optimization method, the EF method will be used with the default settings. For more information on the EF method and how to change to default settings to

achieve optimization for troublesome structures, see Section 4.2 and the various EF keywords in Section 6.3.

For troublesome geometry optimizations, the keywords RECALC and KICK may be used with the EF, BFGS and DFP geometry optimizers to increase the likelihood of convergence.

The keyword NOPOL indicates that the AMSOL program is to proceed with a normal gas-phase calculation, including geometry optimization in the gas-phase, but upon completion it prints out the free energy of solvation which would be obtained using the gas-phase geometry and wave function (in particular the solvation free energy depends on the atomic charges and--in some models--on the bond orders calculated from the solute wave function). Often considerably smaller polarization free energies are obtained with the geometry and wave function "frozen" this way than when they are re-optimized in the presence of solvent. Note: NOPOL does not work with any of the SMxR (R for use of "rigid" gas-phase geometries that are not allowed to relax) models. The same type of result can be achieved for a "rigid" model by using the EXTCM option with gas-phase point charges.

The keyword 1SCF performs a single SCF calculation at the input geometry. This keyword can be used for either gas-phase or solution calculations.

A calculation of the true solvation energy requires subtracting the gas-phase energy at the optimized gas-phase geometry from the solution-phase free energy at the optimized solution-phase geometry. The keyword TRUES is available for the convenient calculation of true solvation energies. Combining the TRUES keyword with the HFCALC option (described under the TRUES keyword) even allows this to be done in a single run.

We note that solution-phase calculations require more computer time than gas-phase calculations, and for larger molecules it often saves computer time to run solution-phase calculations using the TRUES keyword with the CALC option. This performs a gas-phase optimization first, and then a solution-phase optimization beginning at the gas-phase minimum energy structure. This is, of course, unnecessary if the initial geometry already corresponds to the gas-phase minimum energy structure.

The keywords ASA and DOTS determine which of the two solvent-accessible surface area algorithms is used for the determination of the surface area in both the coulombic and surface tension portions of the solvation calculation. If none of these keywords is explicitly specified, the default in version 5.4 is ASA. ASA is an analytical method for calculating the areas, while DOTS places quadrature points on the surfaces of various spheres and calculate surface areas numerically.

The keywords VOLUME and NOVOL are used to choose whether or not to calculate the volume of the optimized molecule (or--if optimization is not carried out--for the input geometry). The default is NOVOL, for which the volume is not calculated.

The keyword EXTSM is available to advanced users to adjust most of the solvation model parameters used in the various SMx models.

The keyword XKW is available to advanced users to allow the use of additional keyword lines or the easy incorporation of keywords into series of trials which use similar keywords.

The old keywords AQUO and ENVAQ used to specify SM1 and SM1a calculations respectively in version 1.0 of AMSOL are no longer supported. They are replaced by SM1 and SM1a, respectively.

The keyword PRECISE is no longer supported for calculations using the EF, BFGS, or DFP geometry optimization methods. The five functions of PRECISE (to decrease SCFCRT, to tighten the geometry optimization

convergence condition, to specify central finite differences (with a larger step size) instead of forward finite differences, to turn off the "frozen SCF" method when doing geometry optimizations, and to request an extra SCF check) have been taken over for such cases by individual keywords or have been made the default.

6.2. Automatic Input Files, Time Limits, and Restarts

The keywords INPUT and NOINP are used to choose whether or not to create a new input file with the optimized geometry. The default option is NOINP, meaning that a new input file is not created. If INPUT is specified, the keywords and optimized geometry are placed into a new file with the run name and the ".inp" suffix. Note that the heat of formation provided in the .inp file is the current value obtained for a given gas-phase run. For a solution-phase run, the heat of formation is listed in the .inp file only if it was listed in the .dat file used to create the .inp file; in that case, the heat of formation provided is a copy of the data provided in the .dat file.

There are two possible time limits which can be exceeded, thereby forcing a calculation to terminate prior to being completed. The first is an internal time limit which is set to a default value TDEF (see description of SIZES.i file in Section 5.2). The second, which may be present in some computing environments, is an external time limit which may be set by queue or interactive time limits on the machine you are using. The internal time limit can be controlled by the keyword T = x, where x is in seconds. It can be used to increase the maximum allowed time limit above the default value. The external time limit cannot be changed from within the code.

If a job called jobA is stopped because of the internal time limit, the jobA.out file will indicate that the job has been stopped due to insufficient time, and it can be restarted. The restart information has been stored in three files. These files are jobA.res, jobA.den, and jobA.inp. The jobA.inp file contains the last calculated geometry of the run and can be used as a new .dat file to restart the run. In this case, the keyword RESTART will also be added to the keyword line. To restart the run, the user should change the extension on the .inp file to .dat and resubmit using the script amsol[x].c where x = i, b, ai, or ab. The calculation will continue from where it was interrupted. (To continue the calculation, the keyword T = x must be used to increase the time limit beyond the point of last termination.)

If a job, call it jobB, is stopped because of an external time limit, the user might be able to restart the calculation. In the working directory, if the script amsoli.c or amsolb.c was used to run the job or create a command file, there should be a directory called jobB.dir. (Note: If your machine stops the executable, amsol.exe, but allows the script to run to completion, the directory jobB.dir will have been removed and the job can be restarted by following the instructions provided for restarting a trial run with the script amsolai.c or amsolab.c.) Inside that directory, all the fort files for the run are stored. If the job took longer than 1800 seconds and TDUMP was not specified or if the keyword "TDUMP=x", where x is seconds, was used and the job took longer than x seconds to complete, then three restart files were created. If this is the case, the files fort.9, fort.10, and fort.18 should be included in the jobB.dir directory. To restart the run, move fort.9 to the working directory and rename it jobB.res, move fort.10 to the working directory and rename it jobB.den, and move fort.18 to the working directory and rename it jobB.dat (NOTE: This file is DIFFERENT from the file jobB.dat already in the working directory and

care should be taken not to overwrite the original .dat file if it needs to be kept.) The trial can then be restarted using the amsoli.c or amsolb.c script.

If the script amsolai.c or amsolab.c is used to run a job or create a command file, then the jobB.dir is not created. Since use of these two scripts implies that the Unix system supports the ASSIGN command, the fort files are simply assigned to the corresponding jobB.[extension] files automatically. In order to restart the run, the file jobB.inp must be renamed jobB.dat and the job resubmitted using the amsolai.c or amsolab.c script. In either case, the restart files ([filename].res, [filename].den, and [filename].inp or fort.9, fort.10, and fort.18) are created or refreshed at least every 1800 seconds (or x seconds if keyword "TDUMP=x" is used). These files may be refreshed more often as described in the optimization section. Thus, if a calculation is terminated prior to its completion, it may be restarted from the point last saved in the restart files.

The default time limit for particular calculations in the code, e.g., the SCF iterations, which was hard-wired in AMPAC-version 2.1 to be 3600 seconds has been converted to a variable. As mentioned above, this variable is initialized to the value TDEF; TDEF is a parameter which is set in the INCLUDE file SIZES.i. TDEF can be changed by the TLIMIT=x keyword. This change was necessary because the limit of 3600 seconds was too low to get all the test runs in the test suite to run to completion on the workstations without having to use the RESTART option. Two versions of SIZES.i are distributed. The one in file SIZES.i has TDEF = 65000, and the one in file SIZESc.i has TDEF = 3600.

6.3. Keywords

Any AMSOL keyword that is an extension beyond the keywords provided in AMPAC version 2.1 is listed alphabetically in this section.

--Standard keywords-- are intended for use by both regular and advanced users. Unless otherwise specified, the below listed keywords are standard.

--EF keywords-- are standard keywords that are applicable only when using the Eigenvector Following (EF) method. Note that the EF method may be invoked by using the EFOLLOW or TSTATE keywords or--since it is the default geometry optimizer--by not invoking any of the other optimizers. EF keywords are specifically identified as such in this section.

--Developer keywords--, as opposed to standard keywords, are intended for use only by developers or advanced users, and they require the use of the DEV keyword. Use of developer keywords may yield results that require careful analysis and interpretation. In this section, developer keywords are all identified as such in the subheading. Note that certain keyword combinations can be developer combinations even if the individual keywords used in the combination are standard keywords. This is to prevent the user from unknowingly using a combination that has not been tested or combinations of keywords that were not designed to be used together. If one of these combinations is entered without the DEV keyword, the code will stop and an explanation will be written in the .out file.

All keywords are case insensitive.

6.3.1. Keyword ALPHA=x

The ALPHA keyword specifies alpha of the desired solvent for an organic solvation calculation which uses the keyword SOLVNT=GENORG. ALPHA is defined to be the value computed by Abraham as Sigma-Alpha, and

denotes a solvent molecule's ability to donate a hydrogen bond. The file solv.prp in the test directory of the distributed code contains values of ALPHA for many common organic solvents. Some of the many references for values not contained in that file are:

- [1] Abraham, M. H., Chem. Soc. Rev., (1993) 73.
- [2] Abraham, et. al, J. Pharm. Sci., 83 (1994) 1085.

6.3.2. Keyword AREAS

The use of the keyword AREAS for a solvation calculation prints out a very detailed listing of the solvent-accessible surface areas in the molecule and related information about the components of the CDS term. The specific areas printed out by this keyword depend on the solvent model chosen. All areas printed by this keyword, however, have already been modified by non-surface tension terms. For SM4, the areas have been multiplied by bond orders where appropriate, for SM5, the areas have been multiplied by geometry-dependent COTs where appropriate. To convert each area to a CDS energy, the value supplied was simply be multiplied by the appropriate surface tension. For a non-aqueous calculation the surface tension listed in the table (Sigma k or Sigma k,k') is a sum over the solvent descriptors multiplied by the appropriate solvent-dependent surface tension. The dependence of the overall CDS term with respect to the solvent parameters is also listed. This keyword is only compatible with SM4- and SM5-type calculations.

6.3.3. Keyword ASA

The ASA keyword, which is now the default, utilizes an analytical algorithm to calculate the exposed surface solid angle of a sphere in both the coulombic and surface tension parts of the calculation. At most one of the two choices, ASA or DOTs, should be specified.

Further details can be found in reference 16 of Section 3.1 of this manual.

6.3.4. Keyword BETA=x

The BETA keyword specifies beta of the desired solvent for an organic solvation calculation that used the keyword SOLVNT=GENORG. BETA is defined to be the value computed by Abraham as Sigma-Beta, and denotes a solvent molecule's ability to accept a hydrogen bond. The file solv.prp in the test directory of the distributed code contains values of BETA for many common organic solvents. Some of the many references for values not contained in that file are:

- [1] Abraham, M. H., Chem. Soc. Rev., (1993) 73.
- [2] Abraham, et. al, J. Pharm. Sci., 83 (1994) 1085.

6.3.5. Keyword BFGS

The keyword BFGS specifies that geometry minimization should be done by a quasi-Newton method using the BFGS method of updating the Hessian. See Section 4.2 for more information on this method.

6.3.6. Keyword CART

The keyword CART specifies that Cartesian coordinate input will be used. The code no longer recognizes Cartesian coordinate input from the lack of connectivities as it did in versions previous to 6.0. This change was necessary to allow the use of the EXTCM and EXTM keywords with a Cartesian input deck. Note: Calculations are no longer performed in Cartesian coordinates, so geometries input using this keyword will be transformed by the program into internal coordinates for the

calculation. Future versions of AMSOL may allow geometry optimizations using Cartesian coordinates.

6.3.7. Keyword CM1

The keyword CM1 specifies that Charge Model 1 (CM1) should be used. The choice of parameterization, either CM1A or CM1P, is made by choosing either the AM1 or PM3 Hamiltonian with a separate keyword. CM1A (alsocalled CM1/AM1) is used with the AM1 Hamiltonian, and CM1P (also called CM1/PM3) is used with the PM3 Hamiltonian. CM1 causes the Charge Model 1 charges and dipole moment to be printed directly above the Mulliken population analysis results in in the .out file. In addition, CM1 results will replace the Mulliken charges and dipole moments in the .arc and .inp files.

The CM1 mapping occurs in two stages. In stage 1, Mulliken partial atomic charges are scaled and/or adjusted by an offset, where the scale factors and offsets are parameters of the method. In extensive testing, Mulliken partial atomic charges for carbon were found to be adequate, and thus carbon has no CM1 parameters, i.e., partial charges on C do not change in stage 1. However, partial charges on H, N, O, F, Si, S, Cl, Br, and I do change. In stage 2 of a CM1 calculation, however, integral charge (0 for neutrals, plus or minus integer for ions) is restored for the molecule by local adjustments that are dependent for each atom on its bond order(s) to atoms that were adjusted in stage 1. So a carbon attached to an oxygen, for instance, will typically become more positive to provide some of the charge required to make oxygen more negative.

CM1 is parameterized for H, C, N, O, F, Si, S, Cl, Br, and I. Other atoms for which AM1 and/or PM3 parameters exist will behave like carbon with respect to how their charges may be affected by a CM1 calculation, i.e., their partial charge may change in stage 2 but not in stage 1. Note, however, that their lack of CM1 parameters is indicative of their not having been tested. They are thus distinct from carbon, which has a "optimized" value of zero for the two change parameters.

Further details are in reference 11 of Section 3.1 of this manual.

6.3.8. Keyword CM2

The keyword CM2 specifies that Charge Model 2 (CM2) should be used. CM2 parameterizations are available in AMSOL for either the AM1 or PM3 Hamiltonian. Note that unlike CM1, the user does not need to specify the choice of Hamiltonian explicitly in the CM2 keyword. When the CM2 keyword is chosen in addition to the AM1 or PM3 Hamiltonian, the Charge Model 2 charges and dipole moment will be printed directly above the Mulliken population analysis results in the .out file. In addition, CM2 results will replace the Mulliken charges and dipole moments in the .arc and .inp files.

CM2 is parameterized for H, C, N, O, F, Si, P, S, Cl, Br, and I.

Note that CM2 charges are implicitly used when the user employs the SM5.42R solvation model. Further details of this model can be found under the SM5.42R keyword description.

Details of the theory of the CM2 model are given in reference 31 of Section 3.1 in this manual.

6.3.9. Keyword CM3

The keyword CM3 specifies that Charge Model 3 (CM3) should be used. CM3 parameterizations are available in AMSOL for either the AM1 or PM3 Hamiltonian. Note that unlike CM1, the user does not need to specify the choice of Hamiltonian explicitly in the CM3 keyword. When the CM3 keyword is chosen in addition to the AM1 or PM3 Hamiltonian, the Charge

Model 3 charges and dipole moment will be printed directly above the Mulliken population analysis results in the .out file. In addition, CM3 results will replace the Mulliken charges and dipole moments in the .arc and.inp files.

CM2 is parameterized for H, C, N, O, F, Si, P, S, Cl, and Br. For the Pm3 method, it is also parameterized for Li.

Details of the theory of the CM3 model are given in reference 32 of Section 3.1 in this manual.

6.3.10. Keyword CS1

In versions of AMSOL prior to version 6.0, the code offered more than one choice for the SCF convergence strategy. (CS1, CS2, and CS3) Since CS2 and CS3 have proven less useful, we have removed the options. The SCF convergence strategy always used by the code corresponds to the CS1 option in AMSOL-version 3.6-AMSOL-version 5.4.2. and the default scheme used in AMPAC-version 2.1 and in AMSOL-version 3.5 and all earlier versions of AMSOL.

For a 1SCF calculation the Fock matrix is updated with Born information derived from the latest atomic charges every four cycles. These updates are necessary because the SCF calculations make implicit use of the linearity of the Fock operator with respect to the density matrix. This property does not hold for SMx schemes because of the sigmoidal dependence of the diagonal of the Fock matrix on the density matrix. The solvation updates may cause severe oscillations to occur, preventing convergence. In order to efficiently damp these oscillations, 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. For geometry optimizations updating occurs at iterations 1, 4, 9, 16, Should the SCF fail to converge, the program will output a message to the effect that the NSTAR index has been increased by one, and updating will now occur twice in a row, but only half as often, e.g., for a geometry optimization at iterations 1, 2, 9, 10, 25, 26, Should oscillation still be observed the updating will occur three times in a row, but only a third as often. If oscillations still prevent the SCF from converging, the NSTAR index will again be increased to 4 and the updating will take place four times in a row, but only a fourth as often. Finally, if the SCF is still unable to converge, the pattern will be repeated with an NSTAR index of 5. If convergence is still not achieved, an error will be printed in the output file and the program will terminate. With this formalism, in general only extremely high energy (unusual) arrangements of atoms, which fail to converge in the gas phase as well, will fail to give a converged SCF.

6.3.11. Keyword CS2

The CS2 keyword is no longer supported.

6.3.12. Keyword CS3

The CS3 keyword is no longer supported.

6.3.13. Keyword DDMAX=y (EF keyword)

See explanation of DDMIN.

6.3.14. Keyword DDMIN=x (EF keyword)

The specifications "DDMIN=x" and "DDMAX=y" set the limits for trust radius (in Angstroms) in EF optimizations. The defaults

are DDMIN equals 0.01 and DDMAX equals 0.3 (TSTATE keyword) or DDMAX equals 0.5 (EFOLLOW keyword).

6.3.15. Keyword DERINU

The DERINU keyword refers to using purely numerical methods to evaluate the gradients and derivatives within a geometry optimization or when the keyword GRADIENTS is used. By default, the derivatives in this method are calculated using a 2-point central finite differencing method. This can be changed to a single-point finite differencing scheme using the keyword FORWARD. The DERINU keyword is the default method (and only allowed method) for solvation models SM1 through SM4.

6.3.16. Keyword DERISA

The DERISA keyword refers to using semi-analytical methods to evaluate the gradients and derivatives within a geometry optimization, or when using the keyword GRADIENTS. The DERISA keyword is the default method for gas-phase calculations, for the SM5.4 model, the SM5.2 model, and the SM5.42 model. The keyword DERISA is not compatible with any of the R models. When DERISA is used with SM5.4, SM5.2, or SM5.42, the derivatives of the surface areas (in the surface tension and electrostatic algorithms) with respect to geometry are calculated analytically.

6.3.17. Keyword DEV (Developer keyword)

Beginning with version 4.1 of AMSOL, the keyword set has been divided into two categories. The first subset includes keywords and keyword combinations that are fundamental to the use of the AMSOL package and are intended to be used for standard applications. Such combinations include AM1 with SM1, SM1A, SM2, SM2.1, and SM2.2 or PM3 with SM3 or SM3.1. These combinations have been designed to work together and the results are fairly predictable. This class of keywords also includes innocuous keywords which while not necessarily mandatory for the proper use of AMSOL, do not pose a significant threat of leading to wrong results if not used with extreme care. Such keywords include INPUT and VOLUME which provide additional features and information but do not compromise results obtained by the core of the code. The second subset of keywords and keyword combinations involve options which may affect even the most basic results obtained with the program. In general, the value and validity of such options have not been fully explored, and the results must be carefully interpreted before they can be used. For this reason, the user must supply the DEV (short for DEVeloper) keyword in order to complete a run using such keywords or combinations of keywords. If the DEV keyword is not entered, the program will stop after reading the keyword line, and an explanation of the problem will be dumped into the .out file. Such combinations include the use of AM1 with SM3 or SM3.1 or PM3 with SM1, SM1A, SM2, SM2.1, or SM2.2. Also included are any combinations of the MNDO Hamiltonian with a solvation model.

6.3.18. Keyword DFP

The keyword DFP specifies that geometry minimization should be done by a quasi-Newton method using the DFP method of updating the Hessian. The DFP method was replaced by the EF method as the default optimizer, beginning with version 5.0 of AMSOL. See Section 4.2.1.1 for more information on this method.

6.3.19. Keyword DIELEC (Developer keyword)

The keyword DIELEC can be used to change the solvent dielectric constant that is used in computing the polarization energy and Fock matrix contributions during an SMx calculation. The form of the keyword is "DIELEC=x.x" where x.x is the dielectric constant of the solvent in free format. There should be no space in this keyword. Note that the use of this keyword for any calculation other than an organic solvation free energy calculation is non-standard, and the final results beyond the polarization energy are of dubious value. Because of this, the keyword DEV is required in conjunction with this when used in a non-standard way.

6.3.20. Keyword DMAX=x (EF keyword)

"DMAX=x" changes the value of the starting trust radius (in Angstroms) in EF optimizations. The default is DMAX equals 0.2 for EFOLLOW and 0.1 forTSTATE.

6.3.21. Keyword DOTS

The keyword DOTS utilizes AMSOL's original algorithm to position points on the surface of all spheres for the determination of the surface area of a sphere in the coulombic and surface tension portions of the calculation. At most one of the two choices, ASA or DOTS, should be specified; ASA is the default. The DOTS algorithm, which is explained clearly in reference 7 of Section 3.1, places n evenly spaced points around a great circle of the sphere. The sphere is then divided by n-1 longitudinal lines that are evenly spaced from -PI/2 to PI/2 radians. DOTS determines how many points should lie on any of these lines by multiplying the number of points on the great circle by the cosine of the angle between the great circle and the line. The number of points on the sphere is equal to the sum of the points on all of the longitudinal lines. The keyword "NDOTEP=x" specifies the number of points on each coulomb sphere in the electric polarization part of the calculation, and "NDOTCD=x" specifies the number of points on each sphere for the surface tension portion.

Due to the structure of the DOTS algorithm, spheres cannot be constructed with an arbitrary number of points. Thus when a number of points is entered for the keyword "NDOTEP=x" or "NDOTCD=x", a subprogram, ARCCH, chooses the highest allowable number of points that can be placed on a sphere that is less than or equal to the number input in the keyword.

If the keyword DOTS is used without specifying the number of points for each sphere in the coulombic and surface tension sections, using the keywords "NDOTEP=x" or "NDOTCD=x" respectively, the default values are those used in versions 3.0.1 and earlier of AMSOL, namely 816 points on each sphere for the coulombic section and 2610 points on each sphere for the surface tension section.

The defaults for the DOTS algorithm are NDOTCD = 2610 and NDIVEP = 816, corresponding to the number of points on the surface of each sphere. Note: The value input for "NDOTEP=x" and "NDOTCD=x" must be greater than or equal to 2, less than or equal to 28761, and less than or equal to MXPT which is a parameter set in SIZES2.i.

6.3.22. Keyword EFOLLOW (EF keyword)

The Eigenvector Following (EF) method is now the default geometry optimization algorithm. This keyword explicitly invokes the Eigenvector Following routine to optimize to a minimum-energy structure; thus it is just an explicit way to invoke the default setting.

6.3.23. Keyword EXTCM (Developer keyword)

This keyword allows the user to fix use nonstandard charges throughout an entire run. When using the EXTCM keyword, the code reads in the charge for each non-dummy atom from the .dat file. These values should be entered (one real value for each atom) on the same line as the Z-matrix data for the atom and in a field separated from the M value or final connectivity information for that atom by at least one blank space. (If neither the keyword STDM nor EXTM is used then the charge should follow the final connectivity information. If the keyword STDM or EXTM is used then the charge information should follow the M value on the line for a particular atom.) This option is designed for advanced users, and thus the keyword DEV must also be included on the keyword line to make use of it.

When using the INPUT keyword, the charge values (whether calculated when not using EXTCM or input when using EXTCM fixed by the EXTCM keyword) will automatically be placed in the .inp file. No additional keywords will be placed in the keyword line.

Users should carefully verify that their atomic charges sum to the desired molecular charge. Note that if EXTCM is specified, then charges must be input for all non-dummy atoms. For dummy atoms one may put zero or blank.

Note that one may use EXTCM with OPT, 1SCF, or NOPOL. In the case of OPT or 1SCF the charges will be fixed through all SCF cycles and will not be self-consistent.

EXTCM is used primarily to see the effect of different charges on the electrostatics. Since surface tension coefficients are parameterized for a specific charge prescription, CDS terms and total free energies are not necessarily meaningful when EXTCM is used.

6.3.24. Keyword EXTM

The EXTM and STDM keywords allow the user to specify the number of expansion shells about each atom, M, used in the radial quadrature of a trial using the force trapezoid algorithm. If the keyword EXTM is used, the code reads in the M value for each atom from the .dat file. These values should be entered (1 integer for each non-dummy atom) on the same line as the Z-matrix data for that atom and in a column that is at least one blank space after the last connectivity information. Section 6.5 gives a list of trial runs that use this option. Note that each line of the Z-matrix (including the line for the first atom) uses "0"'s as place holders even for unnecessary bond lengths, optimization statuses, bond angles, dihedral angles, and connectivities. When using the EXTM keyword, the M values for every non-dummy atom must be specified. To further control the size and number of the expansion shells, the keyword "TEXPN=x" can also be used in conjunction with the EXTM keyword, but the "TONE=x" keyword can not be used. If "TEXPN=x" is not used, there will be M expansion shells and each shell will have a radius which is 1.2 times larger than the previous shell. (1.2 is the default value of "TEXPN=x" for the force trapezoid algorithm.) The M value for each atom will be placed in the .out file. As previously mentioned, the keyword DEV must be used with the keyword EXTM, since it is an option intended for advanced users.

When using the INPUT keyword, the M values (whether calculated or fixed by the EXTM keyword) will automatically be placed in the .inp file. In addition, if the EXTM keyword was not used when the .inp file was created, the keyword STDM (use STAnDard M values) will be placed on the keyword line. This means that although the M values are included in

the file they should be ignored and the M values should be calculated as normal. Although the program will not stop if STDM is not used and the M values are present, this keyword is necessary when using the EXTCM option since the charge values must be placed after the M values.

Further discussion of EXTM is provided under the TEXPN keyword.

6.3.25. Keyword EXTSM (Developer keyword)

The keyword EXTSM is available to advanced users to adjust most of the parameters used in the various SMx models. To use this option, a file must be designated in the amsol[y].c script (where y = i, b, ai, or ab). For SMx models where x = 1, 2, 3, 2.1 or 3.1, the EXTSM file contains the R-k, Sigma(0), Sigma(1), Rho(0), Rho(1), q(0), and q(1) parameters for a certain parameterized atom as defined in reference 8. Note that this option is not available for use with the SM1A parameter set. For the SM4 models, the EXTSM file contains the R(k), Sigma(k), Sigma(Hk'), Rho(0), Rho(1), q(0), and q(1) parameters for a certain element as defined in references 12 and 13 of Section 3.1. The value for Sigma(Hk') is entered on the line for element k'. In addition, the EXTSM file for the SM4 models may contain two additional lines that define the CS surface tension (CS) and the carbon-carbon bond-order-dependent surface tension (CC) used in SM4 alkane models. The EXTSM file for the SM5 models contains R(k), Sigma(k), Sigma(Hk'), Rho(0), Rho(1) and is terminated by 2 columns of zeros. The value for Sigma(Hk') is entered on the line for element k'. In addition, the SM5 EXTSM file may contain up to 5 lines which contains a two-letter abbreviation (OO, ON, OC, NC, SS) followed by a free-format real number, and one line which contains a two-letter abbreviation (CC) and two free-format real numbers. These additional lines give the values for the geometry-dependent surface tensions. The SM5.4-organic and SM5.4-chloroform EXTSM files use the same format as the SM5.4-aqueous EXTSM file and also may contain a line for the CS surface tension which uses a two character tag (CS) and a single free-format real number. Since EXTSM is an advanced option, the keyword DEV must also be included in the keyword line.

The templates for this external file are included in the test directory as SM1.i, SM2.i, SM2.1, SM3.i, SM3.1, SM4, SM5.4A, SM5.4P and SM5.4U. These files correspond to the SM1, SM2, SM2.1, SM3, SM3.1, SM4 and SM5.4/A, SM5.4/P and SM5.4U aqueous parameter sets respectively. It is only necessary to include entries for the atom types the user wishes to change, but all parameters for any atom for which a change will be made must be included. The format for this file is that the first two columns of each line must have the atom's atomic symbol, then the rest of the line must contain the parameters for that atom (real numbers for each) in the order they are listed. The form for the two special SM4 lines is a two letter key in place of the element (either CS or CC) followed by a single real number. The suggested approach to using this keyword is to copy the desired template to another file, specify this new file in the script used to run the trials (amsoli.c or amsolb.c), then simply change the desired parameter in the file while leaving the other parameters untouched. If this keyword is used, all such parameters will be dumped into the .out file. This option is provided only for developers and advanced users and thus the keyword DEV must also be used. One or more lines of comment may be inserted in the EXTSM file by placing a number sign (#) at the beginning of the line. For further examples of EXTSM files, see Section 2.3.

6.3.26. Keyword FACARB=x.xx

The FACARB keyword specifies the fraction of non-hydrogenic atoms in the solvent that are aromatic carbons. For example, when prediction solvation free energies in benzene solvent, FACARB=1.00. For toluene, FACARB=0.857. Note that by definition the value of FACARB should always be between 0 and 1. This keyword is required for all organic solvation calculations that use the keyword SOLVNT=GENORG except for the SM5.4/A and SM5.4/P models.

6.3.27. Keyword FEHALO=x.xx

The FEHALO keyword specifies the fraction of non-hydrogenic atoms in the solvent that are electronegative halogen atoms (F, Cl, or Br). For example, when prediction solvation free energies in chloroform, FEHALO=0.75. For carbon tetrachloride, FEHALO=0.80. Note that by definition the value of FEHALO should always be between 0 and 1. This keyword is required for all organic solvation calculations that use the keyword SOLVNT=GENORG except for the SM5.4/A and SM5.4/P models.

6.3.28. Keyword FOCK

If the keyword FOCK is included in the data file, the final Fock matrix is printed (as with AMPAC-version 2.1), and additionally the contributions to the Fock matrix diagonal elements from the generalized Born treatment are printed by atom in eV. The default is NOFOC, which does not print the final Fock matrix.

6.3.29. Keyword FORWRD

The keyword FORWRD specifies that forward finite differences should be used for numerical derivatives in geometry optimizations. The default is to use central finite differences when numerical derivatives are calculated. The use of forward finite differences is faster than central finite differences, but may cause optimization difficulties.

6.3.30. Keyword GAMMA=x

The GAMMA keyword specifies the macroscopic surface tension of the desired solvent for an organic calculation which uses the keyword SOLVNT=GENORG. The value of GAMMA should be in units of cal mol⁻¹ Angstrom⁻². The file solv.prp in the test directory of the distributed code contains values of GAMMA for many common organic solvents. A convenient reference for values not contained in that file is the "CRC Handbook of Chemistry and Physics", published by CRC Press. A much more complete reference is Jasper, J. J., J. Phys. Chem. Ref. Data, 1 (1972) 841-1009.

6.3.31. Keyword GCOMP

The use of the GCOMP keyword allows the user to change the geometry convergence criterion used in the EF, TRUST, BFGS, and DFP geometry optimization methods. The default value of GCOMP is 0.45 kcal/RU, where RU denotes a Reduced Unit explained in Section 4.2.4.1. To meet the convergence criterion, the largest component of the gradient must be less than GCOMP. The default value of GCOMP (0.45 kcal/RU) was found in test runs to be low enough to yield results that differ in energy from fully converged results by less than 0.005 kcal/mol.

6.3.32. Keyword GEPOL

The keyword GEPOL is no longer supported.

6.3.33. Keyword HESS=x (EF keyword)

"HESS=x" specifies how the Hessian matrix will be calculated. "HESS=0" specifies that the initial Hessian will be approximated as diagonal "HESS=1" Calculate the Hessian using forward finite differences "HESS=2" Read the Hessian from disk, from a previous run "HESS=3" Calculate the Hessian using central finite differences. The default is HESS equals 0 for minimum optimization (EFOLLOW keyword) and HESS equals 1 for transition state optimization (TSTATE keyword). See further discussion in Section 4.2.2.2.6.3.33. Keyword HFCALC=x Where x is 1SCF or OPT. See discussion under the TRUES keyword.

6.3.34. Keyword INPUT

The keywords INPUT and NOINP are used to choose whether or not to create a new input file with the optimized geometry. The default option is NOINP, meaning that a new input file is not created. If INPUT is specified, the keywords and optimized geometry are placed into a new file with the trial name and the ".inp" suffix. In addition, the partial charges and heat of formation (for gas-phase runs) are placed in the file. The charges are placed at the end of each line in the Z-matrix and the heat of formation is placed on the fourth line (or eighth line if the XKW keyword is used) of the .inp file. In order to start a run using a .inp file, the suffix of the .inp file must be changed to .dat, then it can be used as an input data file with one of the provided scripts. See further discussion in Section 6.2. The keyword NOINP is incompatible with the HFCALC option of the TRUES keyword.

6.3.35. Keyword IOFR=x

The IOFR keyword specifies the index of refraction of the desired solvent for an organic solvation calculation that used the keyword SOLVNT=GENORG. The file solv.prp in the test directory of the distributed code contains values of IOFR for many common organic solvents. A convenient reference for values not contained in that file is the "CRC Handbook of Chemistry and Physics", published by CRC Press.

6.3.36. Keyword IUPD=x (EF keyword)

IUPD=x selects the Hessian updating scheme in EF optimizations.
IUPD=0 No updating IUPD=1 Powell updating scheme[1]
IUPD=2 BFGS updating scheme[2]

The defaults are IUPD=1 for transition state search (TSTATE keyword) and IUPD=2 for minimum (EFOLLOW keyword).

[1] M. J. D. Powell, Math. Prog., 1 (1971) 26.

[2] R. Fletcher, Practical Methods of Optimization: Unconstrained Optimization, Vol. 1, Wiley, New York (1980).

6.3.37. Keyword KICK=x

The keyword "KICK=x" is used to avoid some errors that can occur during geometry optimization in AMSOL. These errors can be avoided by perturbing the geometry of the molecule and continuing the optimization. If KICK is specified and a line minimization error (BFGS and DFP), AMSOL will kick the geometry by perturbing a degree of freedom. The degree of freedom to be kicked is chosen as follows. If any bond lengths are being optimized, then the first unique bond length will be increased 0.15 Angstroms by the first kick, the second unique bond length will be increased 0.15 Angstroms by the second kick, and so on. If more kicks are performed than the total number of unique bond lengths, AMSOL will begin again at the first unique bond length. If no bond lengths are being optimized, AMSOL will increase bond angles 5 degrees using the

same algorithm as described for the bond lengths. If no bond lengths or angles are being optimized, AMSOL will increase dihedrals 10 degrees using the same algorithm as described for the bond lengths. Usage of this keyword is "KICK=x" where x is zero or any positive number and indicates the number of kicks to try before ending with a line minimization error. The default value of KICK is 1 for the DFP and BFGS optimization methods, and this has been sufficient to avoid line minimization errors in all neutral molecules used in the development of this keyword when using the DFP or BFGS methods. The default value of KICK is 0 for the EF method.

6.3.38. Keyword **MODE=x** (EF keyword)

"MODE=x" specifies that the n-th Hessian eigenvector will be followed in the first step of a TSTATE optimization. MODE equals 1 means the eigenvector with the lowest eigenvalue, MODE equals 2 the second lower, and so on. Note that the eigenvectors corresponding to translational and rotational motion, which have zero eigenvalues, are projected out of the Hessian and automatically renumbered as the last six eigenvectors. The next steps will be selected on the basis of the overlap between two consecutive steps (see further discussion under keyword OMIN). If MODE equals 0 the eigenvector with the lowest eigenvalue will be followed, regardless of the overlap with the previous optimization step. The defaults are MODE equals 1 for TSTATE. It is not used for EFOLLOW. See further discussion in Section 4.2.2.2.

6.3.39. Keyword **MSURFT**

MSURFT is used in conjunction with the SM4 general alkane model to input the solvent macroscopic surface tension in calories per mol per square Angstrom. This is used to calculate the solvent CS surface tension using the equation: CS Surface Tension = MSURFT*0.03332 + 15.95 cal mol⁻¹ Angstrom⁻² where both the CS Surface Tension and MSURFT are in units of cal mol⁻¹ Angstrom⁻². The form of the MSURFT keyword is MSURFT=x.x with no spaces. MSURFT is only compatible with the SM4 general alkane model. The CS surface tension can also be set directly using the EXTSM keyword, and doing this overrides any value set with MSURFT. The default value for MSURFT is 38.9 cal mol⁻¹ Angstrom⁻², the value for n-hexadecane.

6.3.40. Keyword **MULLIK**

The MULLIK keyword enables the calculation of Mulliken population analysis in a de-orthogonalized basis. This option is described more completely in Section 4.1 of this manual.

6.3.41. Keyword **NDIVCD=x**

The keyword NDIVCD is no longer supported.

6.3.42. Keyword **NDIVEP=x**

The keyword NDIVEP is no longer supported.

6.3.43. Keyword **NDOTCD=x**

See discussion under the keyword DOTS.

6.3.44. Keyword **NDOTEP=x**

See discussion under the keyword DOTS.

6.3.45. Keyword **NOFOC**

See discussion under the keyword FOCK.

6.3.46. Keyword NOINP

See discussion in Section 6.2 and under keyword INPUT.

6.3.47. Keyword NONR (EF keyword)

Specification of this keyword causes the program to skip the Newton-Raphson step in EF optimizations. See also Section 4.2.2.

6.3.48. Keyword NOPRINTGEOM

See discussion in Section 6.2 and under keyword PRINTGEOM.

6.3.49. Keyword NOPRINTCOUL

See discussion in Section 6.2 and under keyword PRINTCOUL,

6.3.50. Keyword NOPRINTRAD

See discussion in Section 6.2 and under keyword PRINTRAD.

6.3.51. Keyword NOPRINTPOL

See discussion in Section 6.2 and under keyword PRINTPOL.

6.3.52. Keyword NOTRUS

See discussion under the keyword TRUES.

6.3.53. Keyword NOVOL

The keyword NOVOL is no longer supported.

6.3.54. Keyword OMIN=x (EF keyword)

During transition state optimizations, the algorithm calculates the dot product between the previously followed direction and the Hessian eigenvectors. The new step will be along the direction defined by the eigenvector for which this dot product is maximum if this value is greater than OMIN. The default is OMIN equal to 0.8 when the TSTATE keyword is employed. It is not used with the EFOLLOW keyword. See further discussion under keyword MODE and in Section 4.2.2.2. The specific OSM5 keyword used chooses which functions of the solvent properties are used. OSM5.4/A chooses functions optimized for the AM1 Hamiltonian, OSM5.4/P chooses functions optimized for the PM3 Hamiltonian.

6.3.55. Keyword PRECISE

Because of the multiple functionality of the PRECISE keyword, it can be very confusing to understand its action. Therefore it is no longer supported for calculations using the EF (which is the default geometry optimizer), TRUST, BFGS, or DFP geometry optimizers. Instead, when those optimizers are used, the many options that were controlled by PRECISE are now individually controlled as follows. Central finite differences will be used for calculating numerical derivatives of the energy with respect to the geometrical coordinates, and the step size for the finite differences is increased by a factor of 10 over the value calculated in the routine DERIV, functions formerly indicated by the PRECISE keyword. In versions prior to AMSOL-version 5.0, central finite differences were the default when performing solution-phase runs. The old gas-phase default of forward finite differences can be specified using the keyword FORWRD with either gas-phase or solution-phase calculations. The SCF convergence criterion can be controlled using the keyword "SCFCRT=x". In particular, one can use "SCFCRT=1E-7" to obtain the same precision as was previously used when PRECISE was specified. The geometry convergence criterion can be controlled using the keyword

"GCOMP=x". A fourth issue controlled by PRECISE in AMPAC-version 2.1 is to turn off the "frozen SCF" method. When doing a numerical derivative during geometry optimizations, the frozen SCF method does not relax the density matrix each time an atom is moved to calculate a gradient component. Thus, turning off the frozen SCF method causes additional density matrix calculations to be performed during geometry optimization. We found in test calculations that the frozen SCF method does not save very much computer time, if any, and so it has been removed from AMSOL beginning with version 5.0. Finally, an additional SCF check that was turned on for solution-phase calculations or by the keyword PRECISE will now be performed during all calculations. In this version of AMSOL, the function of the keyword PRECISE remains the same as in previous versions for all other geometry optimizers besides EF, BFGS, and DFP, and is described in the AMPAC-version 2.1 manual. Thus, when PRECISE is specified with any of the geometry optimizer keywords other than EFOLLOW, TSTATE, TRUSTE, TRUSTG, DFP, or BFGS, it has the same effect as in AMPAC-version 2.1.

6.3.56. Keyword PRINTGEOM

When not optimizing the geometries will no longer be printed. Using the PRINTGEOM keyword will print the geometries to allow use with previously written scripts. Default is off.

6.3.57. Keyword PRINTCOUL

Prints the coulomb integrals in block matrix form.

6.3.58. Keyword PRINTRAD

Prints the effective Born radii and effective interatomic distances in block matrix form. Default is off.

6.3.59. Keyword PRINTPOL

Prints the Generalized Born polarization energy decomposition in blockmatrix form. Default is off.

6.3.60. Keyword RAD=x (Developer keyword)

As mentioned in Section 6.1, it is possible to run any parameter set with any of the radial quadrature schemes using the "RAD=x" keyword. If x equals 0 then the force rectangle quadrature is utilized, if x equals 1 then the force trapezoidal algorithm is used, and if x equals 2 is used then the pairwise descreening approximation is used and the quadrature is replaced by the analytic pairwise descreening approximation. Since combinations other than the default combinations are of dubious value (SMx parameter sets should be used with the radial quadrature for which they were optimized), they should not be used without serious consideration being given to the purpose of making such combinations. To that end, these combinations are considered "developer" options and thus the keyword DEV must appear on the keyword line of any such trial. If DEV does not appear and a combination other than a default option is attempted, the program will stop after reading the keyword line and write the reason for halting in the .out file.

6.3.61. Keyword RECALC=x

This keyword requests that the program recalculate the Hessian if it has not been recalculated in the last x iterations. For the EF method, the Hessian will be recalculated using the method specified by the HESS keyword and may be quite CPU-expensive. For the BFGS and DFP methods, this "recalculation" amounts to setting the Hessian to the unit

matrix. For a discussion of RECALC with the EF method see Section 4.2.2.2, and for a discussion of RECALC with the BFGS and DFP methods, see Section 4.2.1.1. RECALC has no effect if used with other optimizers.

6.3.62. Keyword RESTART

If this keyword appears in the file fort.18 or [job].inp file, then the code has been forced to end prior to completion. For detailed instructions on how to restart a job once it has been halted, see pages 6-6 and 6-7. The keyword can also be placed on the keyword line by the user. Note that in order to use this function, the files [job].res and [job].den MUST be present in the same directory as the [job].dat file.

6.3.63. Keyword RMAX=y (EF keyword)

See explanation of keyword RMIN.

6.3.64. Keyword RMIN=x (EF keyword)

For an EF step to be accepted, the value of the ratio of the calculated energy to the predicted energy must be bracketed by the values of RMIN and RMAX. Default values are RMIN and RMAX are 0.0 and 1000.0 for EFOLLOW, and 0.25 and 4.0 for TSTATE. See also further discussion in Section 4.2.2.2.

6.3.65. Keyword RSCAL (EF keyword)

RSCAL scales the EF step to trust radius instead of using QA formula. The default is to use QA formula for scaling. See Section 4.2.2.

6.3.66. Keyword SCFCRT=x

The default value of the SCF convergence criterion (SCFCRT) in this version of AMSOL is 1.0E-6, and in test cases we found that this gives well converged heats of formation and energies. This value may be overridden by use of the keyword "SCFCRT=x", where x is a floating point value expressed in either decimal or exponential notation (e.g., 0.000001 is the same as 1.0E-06). Using still tighter tolerances occasionally causes the SCF calculation to converge very slowly due to the numerical discontinuities in the generalized Born terms (this problem is most prevalent when using CS1 or CS2) but is fully allowed. Solution-phase calculations on ions are the most sensitive to SCFCRT, and 1.0E-7 might be tried for very high accuracy in such cases. The default SCF convergence criterion for gas-phase calculations in AMPAC-version 2.1 is 1.0E-05. For runs that do not use the EF, BFGS or DFP optimizers, the keyword PRECISE sets the SCF convergence criterion to 1.0E-07 (see the description for the keyword PRECISE for more information). If both the PRECISE and "SCFCRT=x" keywords are specified, the value provided for the SCFCRT=x keyword will be taken for the SCF convergence criterion.

6.3.67. Keyword SM1

By using both SM1 and AM1, the user requests a calculation in aqueous solution by the AM1-SM1 method. The actual quantities output are, inter alia, the gaseous heat of formation relative to elemental standard states plus the aqueous free energy of solvation as well as the electronic energy plus the aqueous free energy of solvation. One additional item worthy of note is that the ionization potential and the HOMO energy are no longer adequately related by Koopmans' theorem. Hence, AMSOL reports the HOMO energy labeled as such.

In SM1 the accessible-surface-area parameters are independent of chemical environment; thus there is a unique value of each surface tension for each atomic type (i.e., atomic number). Other than including the SM1 keyword, no modifications need be made to the standard AMPAC input file.

6.3.68. Keyword SM1a

The keyword SM1a together with AM1 requests a calculation in aqueous solution by the AM1-SM1a method. The keyword SM1A functions analogously to SM1 with respect to energetics and output, with the exception that the accessible-surface-area terms are dependent not only on atomic number but also on chemical environment.

Note that this keyword, like all others, is case insensitive, e.g., SM1a, SM1A, and smla are all allowed. When SM1A is specified the atom types must be provided to the program in the input file. They follow the blank line which concludes the symmetry and/or reaction path information and should be entered line-by-line in I3 format in the same order as the atoms appear in the Z-matrix. No entry need be made for dummy atoms as the program will not try to read an atom type for them. The allowed atom types for the SM1A model in version 5.4 are as follows:

1. carbon atom or hydrogen atom attached to a carbon atom
2. hydrogen atom attached to a nitrogen atom
3. hydrogen atom attached to an oxygen atom
4. hydrogen atom attached to a sulfur atom
5. sp³ or amide nitrogen atom
6. sp, sp², or aromatic nitrogen atom
7. sp³ oxygen atom
8. sp² oxygen atom (e.g., ketone, aldehyde, sulfoxide, nitro, etc.)
9. fluorine atom
10. sulfur atom
11. chlorine atom
12. bromine atom
13. iodine atom
14. phosphorus atom or hydrogen atom attached to a phosphorus atom

The output file will echo back the atom types it has read along with the geometric information, etc.

Execution of the script envaq.c displays the SM1a atom types on standard output as a reminder that may be useful when preparing data files.

While AM1-SM1a is more successful than AM1-SM2 in reproducing experimental free energies of solvation for neutral solutes, it suffers from the necessity of assigning explicit chemical environments. For cases where that is ambiguous, the user is left to his own intuition or else the AM1-SM2, AM1-SM2.1, AM1-SM2.2, PM3-SM3, or PM3-SM3.1 solvation models should be used. These models are very successful in approaching nearly the accuracy of AM1-SM1a, and they are applicable to ions and unusual bonding situations, whereas AM1-SM1a is not.

6.3.69. Keyword SM2

By using both the SM2 and AM1 keywords, the user requests a calculation in aqueous solution by the AM1-SM2 method. While analogous to the SM1 method in its generality, the SM2 model recognizes the importance of classifying hydrogen atoms based on the heavy atom to which they are attached and derives that information from the bond order matrix. As a result, it is considerably more accurate than the SM1

approach. The output from such a run will include the final bond order matrix together with the contributions to surface-area-dependent terms from both heavy-atom surface tensions and attached hydrogen atoms (which do not block the solvent-accessible-surface-area of the heavy atoms to which they are attached). A complete description of the formalism may be found in references 11 and 13 in Section 3.1.

In SM2 all accessible-surface-area parameters are independent of chemical environment; thus there is a unique value for all surface tension parameters for each atomic type (i.e., atomic number). Other than including the SM2 keyword, no modifications need be made to the standard AMPAC input file.

6.3.70. Keyword SM2.1

By using both the SM2.1 and AM1 keywords, the user requests a calculation in aqueous solution by the AM1-SM2.1 method, which utilizes the force trapezoid algorithm for the radial quadrature rather than the force rectangular algorithm with which the original AM1-SM2 method was parameterized. The SM2.1 model was parameterized by a fitting to the results obtained with the original AM1-SM2 method. Further details of this process can be found in reference 16 of Section 3.1 of this manual.

The algorithm used for SM2.1 calculations provides faster timings and higher precision than the algorithm used for SM2. The results tend to be very similar.

6.3.71. Keyword SM2.2

By using both the SM2.2 and AM1 keywords, the user requests a calculation in aqueous solution by the AM1-SM2.2 method, which utilizes the pairwise descreening approximation to calculate the electrostatic portion of the polarization free energy of solvation. The surface tensions used in this model were determined by fitting the surface tensions coupled with the newly calculated polarization free energy to experimental results. Using this approximation, the radial quadrature calculated numerically in the SM2 model can be calculated analytically, substantially simplifying the complexity and reducing the cost of the calculation.

The SM2.2 method is only available for molecules containing H, C, O, and/or N. The method was tested on a set of 139 molecules, and when compared to the previous SM2.1 model, we found that the RMS error relative to experiment increased by only 0.2 kcal/mol.

Further details are in reference 16 of Section 3.1 of this manual.

6.3.72. Keyword SM2.2PDA

The SM2.2PD/AM1 (also called SM2.2PD/A) model utilizes the pairwise descreening approximation for the calculation of the polarization free energy for a molecule. This model uses a single intrinsic coulomb radius for each atom type. The surface tension functionals are those developed for the SM2 family of solvation models, and zero overlap Mulliken charges are employed. This model was parameterized for use with the AM1 Hamiltonian. The use of the pairwise descreening approximation allows the radial quadrature calculated numerically in the SM2 model to be calculated analytically, substantially simplifying the complexity and reducing the cost. In addition, the simple functional forms used in calculating the polarization free energy should allow the incorporation of analytical gradients in a future version which will again significantly reduce the cost of the calculation. When 10 of the largest molecules from our training set were tested on a SGI Power Challenge L R8000, the

SM5.2PD/AM1 model obtained a speedup factor of 2.7 when compared to the SM2.1 model with version 5.4.1 of AMSOL.

The SM2.2PD/AM1 model was parameterized by fitting to the experimental free energies of solvation for 219 neutral solute molecules and 34 ionic molecules containing a wide range of organic functional groups and the atom types H, C, N, O, F, P, S, Cl, Br, and I. For the neutral molecules in the N, O, F, P, S, Cl, Br, and I. For the neutral molecules in the parameterization set, the SM2.2PD/AM1 model achieves a mean unsigned error of 0.56 kcal/mol, a mean signed error of 0.06 kcal/mol, and a root-mean-squared-error of 0.80 kcal/mol.

Further details are in reference 21 of Section 3.1 of this manual.

6.3.73. Keyword SM3

By using both SM3 and PM3, the user requests a calculation in aqueous solution by the PM3-SM3 method. This is completely analogous to the SM2 method except that the underlying gas-phase Hamiltonian is the PM3 model of Stewart the solvation model parameters are based upon it. The most desirable feature of the PM3 model is that it appears to predict more reasonable hydrogen bonds in terms of linearity, distance, and energy than does AM1. A significant drawback in comparison to AM1 is that the PM3 nitrogen charges are usually much too positive and the SM3 model is thus reduced in its effectiveness when employed for amines, nitriles, and nitro compounds. In SM3 all accessible-surface-area parameters are independent of chemical environment, thus there is a unique value for all surface tension parameters for each atomic type (i.e., atomic number). Other than including the SM3 keyword, no modifications need be made to the standard AMPAC input file. PM3-SM3 and PM3-SM3.1 are less accurate than AM1-SM2 and AM1-SM2.1 when nitrogen atoms are involved. Further details of the SM3 model are given in references 12 and 13 of Section 3.1.

6.3.74. Keyword SM3.1

By using both the SM3.1 and PM3 keywords, the user requests a calculation in aqueous solution by the PM3-SM3.1 method, which utilizes the force trapezoid algorithm for the radial quadrature rather than the force rectangular algorithm with which the original PM3-SM3 method was parameterized. The SM3.1 model was parameterized by a fitting solvation energies to the results obtained with the original PM3-SM3 method. Further details of this process (for SM2.1) can be found in reference 16 of Section 3.1 of this manual, and SM3.1 is related to SM3 in the same way that SM2.1 is related to SM2. See also Section 2 of this manual.

The algorithm used for SM3.1 calculations provides faster timings and higher precision than the algorithm used for SM3. The results tend to be very similar.

6.3.75. Keyword SM4

The SM4 keyword requests a solvation calculation using the SM4 method. The solvents available for SM4 calculations are the alkane solvents listed under the keyword SOLVNT. Like the SM2, SM2.1, SM3, and SM3.1 models, SM4 treats hydrogen atoms according to their environment. As in these methods and SM1, it is not required for the user to specify the hydrogen environment. The dependence is achieved by placing a surface tension on the hydrogen atom that depends on that atom's bond order to other atoms. A major improvement in the SM4 models compared to previous models is that SM4 calculations use the CM1 models to obtain atomic partial charges. The CM1 models provide partial charges that are more accurate than the Mulliken population analysis used in earlier

models. This eliminates some difficulties such as the problematic nitrogen charges in PM3, but--more significantly--it yields uniformly more quantitative accuracy even in non-problematic cases. The CM1 models yield class IV charges; see reference 12 of Section 3.1 for a discussion of the advantages of class IV charges.

Because the SM4 solvation model can be used with either AM1 or PM3, the Hamiltonian must be specified by using either the AM1 or PM3 keyword. Further information on the SM4 model can be found in references 12 and 13 in Section 3.1. To use the SM4 keyword, no change must be made to the standard AMPAC input file beyond the inclusion of the SM4 and SOLVNT keywords. See the discussion under the SOLVNT keyword for more information about the SM4 solvent model.

6.3.76. Keyword SM5.0R

The keyword SM5.0R is used to run a calculation with the SM5.0R model. This model predicts aqueous or organic solvation energies based entirely on geometry-dependent atomic surface tensions. The R in the name represents the "rigid"ness of the model where the calculation is made from a gas-phase geometry which is not allowed to relax in the presence of the solvent. (The method was parameterized using reasonably accurate gas-phase geometries.) The SM5.0R model is especially suited for problems requiring high efficiency because of the size of the system, e. g., protein folding, or problems requiring rapid estimations because of the large number of calculations required, e. g., scoring of combinatorial libraries.

The method is designed to be employed with accurate gas-phase geometries from any source and will predict solvation free energies for organic molecules containing H, C, N, O, F, S, Cl, Br, and I. The model was parameterized for aqueous solvation free energies using a training set containing 243 neutral solutes with a variety of functional groups, and it achieves a mean unsigned error of 0.53 kcal/mol when the model is applied using gas-phase geometries calculated at the Hartree-Fock level with a heteroatom-polarized valence-double-zeta basis set (HF/MIDI!) and a mean unsigned error of 0.57 kcal/mol when it is applied using gas-phase geometries from Austin Model 1 (AM1). For organic solvents, the training set included 90 solvents and 1836 organic solvation free energy data points. Using HF/MIDI! or AM1 geometries, the SM5.0R model achieves a mean unsigned error of 0.38 kcal/mol in predicting the organic solvation free energies of the molecules in the organic training set.

The SM5.0R keyword must be used with the keyword SOLVNT=xxx where xxx is either WATER or GENORG. If using the GENORG portion of the model, the keywords DIELEC=x.xx, ALPHA=x.xx, BETA=x.xx, GAMMA=x.xx, IOFR=x.xx, FACARB=x.xx, and FEHALO=x.xx must also be used to describe the solvent. Full descriptions of these keywords are listed alphabetically in section 6.3.

Note: Since geometries are not allowed to relax and there are no explicit charges within the SM5.0R model, keywords such as OPT, 1SCF, and EXTCM have no meaning and the code will stop if they are used. Additionally, since the model was parameterized to always yield true solvation energies, the keyword TRUES does not need to be used. Further information about the SM5.0R model can be obtained from the references for the SM5.0R model in Section 3.1 of this manual.

6.3.77. Keyword SM5.05R

The keyword SM5.05R is used to run the SM5.05R model for ions. This model represents a simple extension of the SM5.0R model which allows calculations for certain kinds of charged groups. The model is

"rigid" in the sense that the geometry is not allowed to relax within the field created by the solvent. (The method was parameterized using reasonably accurate gas-phase geometries.) This model is not intended for use in systems where the primary focus of interest is the charged systems themselves.

For a description of the charged groups for which this model was parameterized see the reference for the SM5.05R model in Section 3.1. In general, for a positively charged nitrogen center this model assumes that the entire charge is divided evenly among the hydrogen atoms attached to the charged nitrogen center. For oxygen or sulfur anions, the model approximates the entire charge as being on the sulfur atom or divided equally among the oxygen atoms within the charged group.

Consider the positively charged methyl ammonium molecule. The SM5.05R model assumes that the charge is distributed evenly between the 3 hydrogen atoms attached to the nitrogen. All other atomic centers have zero charge. The generalized Born formula is used to account for the interactions between our simplified charge distribution within the solute and the field created by the bulk solvent. This contribution is then added to the result obtained by treating the entire molecule as a neutral with the SM5.0R model.

To indicate to the AMSOL code which atoms within a molecule are assigned a charge consider the following example .dat file:

```
SM5.05R SOLVNT=WATER CHARGE=1
ATOM 3 5 6
TYPE 3 3 3
Methyl ammonium

N 0.000000 0 0.000000 0 0.000000 0 0 0 0
C 1.525395 1 0.000000 0 0.000000 0 1 0 0
H 1.026698 1 111.400000 1 0.000000 0 1 2 0
H 1.080446 1 108.027000 1 180.000000 1 2 1 3
H 1.026698 1 111.400000 1 60.000000 1 1 2 4
H 1.026698 1 111.400000 1 -60.000000 1 1 2 4
H 1.080446 1 108.027000 1 60.000000 1 2 1 3
H 1.080446 1 108.027000 1 -60.000000 1 2 1 3
```

The word ATOM is placed in the line that directly follows the last keyword line, and the word TYPE is placed in the line immediately following the ATOM line. (Note that the words ATOM and TYPE must each start in the first column of their respective lines.) According to the connectivities of this Z-matrix, hydrogens 3, 5, and 6 are attached to the nitrogen atom and comprise the charged center for this ion. Since the simplified charge distribution built into the SM5.05R model requires that the charge for this molecule be divided evenly among the three hydrogens that are attached to the nitrogen, we must specify that atoms 3, 5, and 6 are the charged atoms by placing their numbers on the ATOM line. The type of a given charge is x where 1/x is the amount of charge to be given to each specified atom. In other words, we have given hydrogens 3, 5, and 6 each a charge of 1/3. (Note that when determining atom number within a Z-matrix, no dummy atoms should be counted.)

Note: Since geometries are not allowed to relax and the typed charges are not allowed to relax within the field created by the solvent within the SM5.05R model, keywords such as OPT and 1SCF have no meaning and the code will stop if they are used. The keyword EXTCM is also not currently supported. Additionally, since the model was parameterized to always yield true solvation energies, the keyword TRUES does not need to

be used. For further details see the reference for the SM5.05R model in Section 3.1 of this manual.

6.3.78. Keyword SM5.2PDx

The keywords SM5.2PDA and SM5.2PDP are used to run aqueous solvation calculations with an SM5.2PD-type model. Both of these models uses a pairwise descreening approximation to simplify the electrostatics portion of a solvation free energy calculation. These models use the SM5-type surface tension functionals, Mulliken charges, and a single intrinsic coulomb radius for each atom type. In addition, the radial quadrature is calculated analytically. These models were parameterized by fitting to experimental free energies of solvation for 219 neutral molecules and 34 ionic solute molecules containing a wide range of organic functional groups and the atom types H, C, N, O, F, P, S, Cl, Br, and I. When 10 of the largest molecules from our training set were tested on a SGI Power Challenge L R8000, the SM5.2PD models has a speedup factor of 5.8 when compared to the SM5.4 models, and a speedup factor of 2.5 when compared to SM2. In addition, the simple functional forms used in calculating the polarization free energy should allow the incorporation of analytical gradients in a future version which will again significantly reduce the cost of the calculation.

The SM5.2PDA keyword is used to request a calculation with the SM5.2PD/AM1 model. Since this model was parameterized for use with the AM1 Hamiltonian, the AM1 keyword must also be used. For the neutral molecules in the parameterization set, the SM5.2PD/AM1 model achieves a mean unsigned error of 0.49 kcal/mol, a mean signed error of 0.02 kcal/mol, and a root-mean-squared-error of 0.70 kcal/mol.

The SM5.2PDP keyword is used to request a calculation with the SM5.2PD/PM3 model. Since this model was parameterized for use with the PM3 Hamiltonian, the PM3 keyword must also be used. For the neutral molecules in the parameterization set, the SM5.4PD/PM3 model achieves a mean unsigned error of 0.52 kcal/mol, a mean signed error of 0.06 kcal/mol, and a root-mean-squared-error of 0.71 kcal/mol.

Further details are in the reference for the SM5.2PD model in Section 3.1 of this manual.

6.3.79. Keyword SM5.2

The keyword SM5.2 is used to run a calculation with the SM5.2 solvation model. This model is the same as the SM5.2R solvation model (see below) with the exception that the geometry is optimized in solution (the R designation for a rigid model is missing).

6.3.80. Keyword SM5.2R

The keyword SM5.2R is used to run a calculation with the SM5.2R solvation model. This model predicts aqueous or organic solvation energies based on geometry-dependent atomic surface tensions and electrostatic polarization energies calculated with class II zero-overlap Mulliken charges calculated from the wavefunction produced by either the MNDO, AM1, or PM3 semiempirical Hamiltonians. The R in the name represents the "rigid"ness of the model where the calculation is made from a gas-phase geometry which is not allowed to relax in the presence of the solvent. (The internal charge distribution within a solute is still allowed to equilibrate (relax) in response to the field produced by the aqueous solvent.)

The SM5.2R method is designed to be employed with accurate gas-phase geometries from any source and will predict solvation free energies for organic molecules containing H, C, N, O, F, P, S, Cl, Br,

and I. The model was parameterized for aqueous solvation free energies using a training set containing 248 neutral solutes with a variety of functional groups, and it achieves a mean unsigned error of about 0.5 kcal/mol when the model is applied using gas-phase geometries calculated at the Hartree-Fock level with a heteroatom-polarized valence-double-zeta basis set (HF/MIDI!) and either the MNDO, AM1 or PM3 semiempirical Hamiltonians. For organic solvents, the training set included 90 solvents and 1836 organic solvation free energy data points. Using HF/MIDI! geometries, the SM5.2R model achieves a mean unsigned error of about 0.4 kcal/mol in predicting the organic solvation free energies of the molecules in the organic training set with MNDO, AM1, or PM3.

The SM5.2R keyword must be used with the keyword SOLVNT=xxx where xxx is either WATER or GENORG. If using the GENORG portion of the model, the keywords DIELEC=x.xx, ALPHA=x.xx, BETA=x.xx, GAMMA=x.xx, IOFR=x.xx, FACARB=x.xx, and FEHALO=x.xx must also be used to describe the solvent. Full descriptions of these keywords are listed alphabetically in section 6.3.

Note: Since the SM5.2R model incorporates any geometrical changes due to the solvation of the solute into the parameterization, the keyword OPT is not allowed and the keyword 1SCF is required to indicate that no geometry optimization is required in solution.

6.3.81. Keyword SM5.42

The keyword SM5.42 is used to run a calculation with the SM5.42 solvation model. This model is the same as the SM5.42R solvation model (see below) with the exception that the geometry is optimized in solution (the R designation for a rigid model is missing).

6.3.82. Keyword SM5.42R

The keyword SM5.42R is used to run a calculation with the SM5.42R solvation model. This model predicts aqueous or organic solvation energies based on geometry-dependent atomic surface tensions and electrostatic polarization energies calculated with class IV CM2 point charges which are mapped from zero-overlap Mulliken charges calculated from the wavefunction produced by either the AM1 or PM3 semiempirical Hamiltonian. The R in the name represents the "rigid"ness of the model, in particular, it represents the fact that the parameters were obtained using a gas-phase geometry which is not allowed to relax in the presence of the solvent. (The internal charge distribution within a solute is still allowed to equilibrate(relax) in response to the field produced by the aqueous solvent.)

The SM5.42R method is designed to be employed with accurate gas-phase geometries from any source and will predict solvation free energies for organic molecules containing H, C, N, O, F, P, S, Cl, Br, and I. The model was parameterized for aqueous solvation free energies using a training set containing 248 neutral solutes with a variety of functional groups, and it achieves a mean unsigned error of about 0.5 kcal/mol when the model is applied using gas-phase geometries calculated at the Hartree-Fock level with a heteroatom-polarized valence-double-zeta basis set (HF/MIDI!) and either the AM1 or PM3 semiempirical Hamiltonian. For organic solvents, the training set included 90 solvents and 1836 organic solvation free energy data points. Using HF/MIDI! geometries, the SM5.42R model achieves a mean unsigned error of about 0.4 kcal/mol in predicting the organic solvation free energies of the molecules in the organic training set with either AM1 or PM3.

The SM5.42R keyword must be used with the keyword SOLVNT=xxx where xxx is either WATER or GENORG. If using the GENORG portion of the model,

the keywords DIELEC=x.xx, ALPHA=x.xx, BETA=x.xx, GAMMA=x.xx, IOFR=x.xx, FACARB=x.xx, and FEHALO=x.xx must also be used to describe the solvent. Full descriptions of these keywords are listed alphabetically in section 6.3.

Note: Since the SM5.42R model incorporates any geometrical changes due to the solvation of the solute into the parameterization, the keyword OPT is not allowed and the keyword 1SCF is required to indicate that no geometry optimization is required in solution.

6.3.83. Keyword SM5.4PDx

The keywords SM5.4PDA, SM5.4PDP, or SM5.4PDU are used to run aqueous solvation calculations with an SM5.4PD-type model. Each of these models uses a pairwise descreening approximation to simplify the electrostatics portion of a solvation free energy calculation. These models use the SM5-type surface tension functionals, class IV charges, and a single intrinsic coulomb radius for each atom type, except hydrogen. The radial quadrature, which is calculated numerically in the SM5.4 models which do not incorporate the pairwise descreening approximation, is calculated analytically. These models were parameterized by fitting to experimental free energies of solvation for 215 neutral molecules and 34 ionic solute molecules containing a wide range of organic functional groups and the atom types H, C, N, O, F, S, Cl, Br, and I. When 10 of the largest molecules from our training set were tested on a SGI Power Challenge L R8000, the SM5.4PD models has a speedup factor of 1.3 when compared to the SM5.4 models with version 5.4.1 of AMSOL. In addition, the simple functional forms used in calculating the polarization free energy should allow the incorporation of analytical gradients in a future version which will again significantly reduce the cost of the calculation.

The SM5.4PD/U model was created as an intermediate step in parameterizing the SM5.4PD/AM1 and SM5.4PD/PM3 models. In SM5.4PD/U, the appropriate pairwise descreening and intrinsic coulomb radius parameters were found by fitting to experimental solvation free energies using results obtained with both the AM1 and PM3 Hamiltonians and a single set of surface tension coefficients. The SM5.4PD/AM1 and SM5.4PD/PM3 models were then obtained by using the pairwise descreening and intrinsic coulomb radii parameters obtained for SM5.4PD/U but refitting the surface tension coefficients for a best fit in the solvation free energies for only AM1 or PM3, respectively. The performance of this model is similar to the SM5.4/U model. Since the SM5.4PD/U model is just an intermediate step to creating the SM5.4PD/AM1 and SM5.4PD/PM3 models, the DEV keyword must be used with the SM5.4PDU keyword.

The SM5.4PDA keyword is used to request a calculation with the SM5.4PD/AM1 model. Since this model was parameterized for use with the AM1 Hamiltonian, the AM1 keyword must also be used. For the neutral molecules in the parameterization set, the SM5.4PD/AM1 model achieves a mean unsigned error of 0.46 kcal/mol, a mean signed error of 0.01 kcal/mol, and a root-mean-squared-error of 0.66 kcal/mol.

The SM5.4PDP keyword is used to request a calculation with the SM5.4PD/PM3 model. Since this model was parameterized for use with the PM3 Hamiltonian, the PM3 keyword must also be used. For the neutral molecules in the parameterization set, the SM5.4PD/PM3 model achieves a mean unsigned error of 0.45 kcal/mol, a mean signed error of 0.02 kcal/mol, and a root-mean-squared-error of 0.62 kcal/mol.

Further details are in reference 9 of Section 3.1 of this manual.

6.3.84. Keyword SM5.4x

The SM5.4AM1, SM5.4PM3, and SM5.4U keywords can be used to yield a solvation calculation using the SM5.4 formalism. The functional form of this model is geometry-based, rather than having bond-order and charge-dependencies in the first-solvation-shell terms. The effective coulomb radii and atomic surface tensions depend in some cases on distances to nearby atoms. Atomic charges are obtained by either the AM1-CM1A and PM3-CM1P class IV charge models. The SM5.4AM1 and SM5.4PM3 models can be used for aqueous or organic solvents; the SM5.4U keyword can be used only for aqueous solvent.

The specific SM5.4 keyword used chooses which first-solvation-shell terms are used. SM5.4A chooses terms optimized for the AM1 Hamiltonian, SM5.4P chooses terms optimized for the PM3 Hamiltonian, and SM5.4U chooses terms optimized for both the AM1 and PM3 Hamiltonians. The SM5.4U keyword also may be useful when performing calculations that are not 'true' AM1 or PM3 calculations, i.e. the geometry or charges used are not calculated by the chosen Hamiltonian.

6.3.85. Keyword SOLVNT=x

Please note up front the following very important points about the SOLVNT keyword:

- (1) This keyword is required (there is no default) if one uses the SM4 or any of the SM5.x keywords. However, if SOLVNT is used with any other SMx keyword, it generates a fatal error.
- (2) The SOLVNT keyword should not be interpreted as simply specifying the solvent. What it actually specifies in SM5.4A and SM5.4P calculations is the parameterization set (the set of surface tension coefficients and other parameters that define a particular parameterization of an SM5.4 model). In SM4 calculations, SOLVNT can be used to specify certain solvents, or it can be used to indicate that general alkane parameters will be provided, or it can be used to specify that an EXTSM file will be provided.

We have provided the SOLVNT keyword because more than one solvent is available for an SM4, SM5.4/AM1, SM5.4/PM3, SM5.42R/AM1, or SM5.42R/PM3 calculation. (The other SM5.x solvation models are parameterized only for water or a general organic solvent whose characteristics are specified by other keywords.) The form of the SOLVNT keyword is "SOLVNT=x" (note that there are no spaces in this keyword) where x is a solvent key according to the following table:

AM1-SM4 or PM3-SM4

Solvent or other choice	Keyword
n-pentane	N5ANE
n-hexane	N6ANE
n-heptane	N7ANE
n-octane	N8ANE
n-nonane	N9ANE
n-decane	N10ANE
n-undecane	N11ANE
n-dodecane	N12ANE
n-tetradecane	N14ANE
n-pentadecane	N15ANE

n-hexadecane	N16ANE
2-methylbutane	2M4ANE
4-methylpentane	3M5ANE
2,2,4-trimethylpentane	ISOOCT
cyclohexane	CYC6
decalin (cis and trans)	DECLIN
general alkane	GENALK (requires DIELEC and MSURFT)
water SRP model	H2OSRP (requires EXTSM)

Explanation for SM4 models

The first 17 elements in the list specify solvents. If one of these is chosen, the code automatically uses the values of DIELEC and MSURFT for that solvent along with the general alkane parameter set. Note that for N16ANE (hexadecane), this automatically reduces to the earlier hexadecane parameter set. The next choice, GENALK, is a general alkane keyword that can be used to carry out a calculation for an alkane solvent that is not in the group of 17 named solvents.

The use of the solvent key GENALK requires additional input from the user in the form of the keywords DIELEC and MSURFT. The default values of DIELEC and MSURFT for a GENALK calculation are 2.06 and 38.9 respectively, which are the values for an n-hexadecane calculation. See these keyword descriptions for more information.

The H2OSRP keyword enables the user to use the SM4 formalism with specific reaction parameters or specific range parameters. If H2OSRP is used, then the EXTSM keyword is required. For information on the use of the H2OSRP keyword, see Section 2.3. See also the description of the EXTSM keyword in the present alphabetical list of keywords.

SM5.4/AM1 or SM5.4/PM3

Parameter set	Keyword
Water	WATER
Chloroform	CHLOROFORM
Benzene	BENZENE
Toluene	TOLUENE
Organic	GENORG (requires IOFR, ALPHA, BETA, and GAMMA)

Explanation for SM5.4/AM1 or SM5.4/PM3 methods

The keyword WATER specifies the parameterization for aqueous solutions. The keyword CHCL3 specifies the parameterization for solutions in which the solvent is chloroform. The BENZENE and TOLUENE keywords specify the parameterization for benzene and toluene solvents. The keyword GENORG specifies the parameterization for general organic solvents; this keyword should be used for all solvents except water, chloroform, benzene and toluene.

The GENORG solvents also require the five keywords IOFR, ALPHA, BETA, GAMMA, and DIELEC which are described in their own sections in the present alphabetical list of keywords. (Note that these five keywords should not be used with the WATER, CHCL3, BENZENE and TOLUENE keywords.) In order to make it easier for users to supply the five necessary keywords, we supply the file solv.prp (in the test directory of the distribution) which contains the values of these solvent parameters for the 91 solvents used in the parameterization of the organic solvation model plus a few additional solvents.

6.3.86. Keyword **STDM**

See discussion under the keyword **EXTM**.

6.3.87. Keyword **SVCDRD (Developer keyword)**

The keyword **SVCDRD** can be used to alter the solvent radius. In solvation models with only a single solvent radius, such as the water models, the **SVCDRD** keyword controls the size of that radius. In solvation models with 2 solvent radii, this keyword controls the small, CD (cavity-dispersion) solvent radius, and **SVCSR**D controls the size of the large, CS solvent radius. The **SVCDRD** keyword is compatible with any solvation model; however, because this keyword allows the use of a non-standard solvent radius, the keyword **DEV** must be used with this keyword. If **DEV** is not specified along with **SVCDRD**, the program will halt and an error message will be displayed in the .out file. Usage: "**SVCDRD=x.xx**" where x.xx is the desired radius in Angstroms.

6.3.88. Keyword **SVCSR**D (Developer keyword)

The keyword **SVCSR**D can be used to alter a solvent radius. In solvation models with two solvent radii, this keyword controls the large, CS (cavity-solvent structure) solvent radius. In solvation models with only a single solvent radius such as the SM1 - SM3.1 models, the **SVCSR**D keyword has no meaning, and the keyword **SVCDRD** should be used if it is desired to change the solvent radius. Because the **SVCSR**D keyword allows the use of a non-standard solvent radius, the keyword **DEV** must be used with this keyword. If **DEV** is not specified along with **SVCSR**D, the program will halt and an error message will be displayed in the .out file. Usage: "**SVCSR**D=x.xx" where x.xx is the desired radius in Angstroms.

6.3.89. Keyword **TDUMP=x**

The keyword **TDUMP** specifies that the .res, .den, and .inp files should be refreshed every x seconds. These files can be used to restart a calculation if a calculation fails. See the discussion of time limits in Section 6.2 for more information on this keyword.

6.3.90. Keyword **TEXPN=x (Developer keyword)**

Two keywords, **TEXPN** and **TONE** are available for diagnostics of the numerical integrations used to determine the atomic effective Coulomb radii. **TEXPN** and **TONE** have default values of 1.5 and 0.02 (for SM2 and SM3) or 0.10 (for SM1 and SM1a) respectively (Each of these use the force rectangle quadrature). Note that in the notation of reference 8 in Section 3.1, $TEXPN = 1+F$, and $TONE = T-1$. For SM2.1, SM3.1, SM4, and SM5.x methods, which use the force trapezoid algorithm, the defaults are 1.2 for "**TEXPN=x**" and 0.15 for "**TONE=x**". These two keywords are not compatible with the SM2.2 model.

"**TEXPN=x**" controls the expansion factor used for concentric shells expanded about each atom in the numerical integration sequence. For the force rectangle quadrature, the default value is 1.5, which means that each subsequent shell has a radius 50% larger than its immediate precursor. By decreasing this value to its limit of 1.0, the integration will deliver values more nearly converged to the analytical effective radius; however large increases in time are required for the computation. Typically $TEXPN = 1.2$ doubles the run-time by comparison to the default.

"**TONE=x**" is the thickness for the first shell about the atom and has marginal effect unless very large (unreasonable) values are chosen.

The program default values MUST be used for "TEXPN=x" and "TONE=x" for true SMx runs (x = 1, 1a, 2, and 3) because the parameters for these solvation models have been developed from them. However, situations where unusual molecular clefts or cavities are present might conceivably deliver radii uncharacteristically far from convergence. Use of these two keywords may then be illuminating for understanding the problem. For this reason, these keywords are considered "developers" keywords and the DEV keyword must appear in the keyword line in order to use these options otherwise the code will stop after reading only the keyword line and the reason for halting will be dumped into the .out file. For a complete description of the numerical methods, see the Journal of Computer-Aided Molecular Design article (reference 7 in Section 3).

The force trapezoid quadrature improves the integration step with respect to both speed and stability. In the force rectangle algorithm, the concentric shells expanded about each atom in the numerical integration sequence do not necessarily end at the precise point where the rest of the molecule was engulfed by the shell. Rather, the first shell has a thickness of TONE and each succeeding shell is TEXPN times the thickness of its immediate precursor thick. This progression continues until a given shell completely engulfs the rest of the molecule. (Note that this is satisfied as long as the distance from the center of atom about which the shells are being expanded to the shell is greater than or equal to the shortest distance sufficient to for the shell to engulf the rest of the molecule.) In the force trapezoid quadrature the value of TONE is adjusted so that the last shell has precisely the smallest radius sufficient to engulf the rest of the molecule. Thus, it will take an exact number of such shells to completely engulf the rest of the molecule. This improves stability.

Using this strategy, it is possible that the number of expansion shells about a particular atom necessary to exactly engulf the rest of the molecule, call it M, could change during the course of a geometry optimization. If the value of M oscillates during the course of a run, it can interfere destructively with the SCF convergers causing a need for an increased number of cycles for convergence or even divergence of the convergers. To combat this problem, the M value for each atom is stored during the course of a given run. After each geometry optimization, the appropriate M value is calculated but the employed M value is changed only if the new M value is larger than any prior calculated M value for a given atom. Thus, the value is allowed to increase, but not to decrease, and hence oscillations are prevented. The final M values for each atom are then printed in the .out file.

In addition, the concerned user is able to control the M value for each atom through the use of the EXTM keyword (EXTernal M values). By placing the desired values in the .dat file in the columns following the Z matrix, the user can set M values which will be used throughout the entire run and will not change. This might be a useful option to try if difficulties with convergence are encountered in solution-phase geometry optimizations.

The expansion factor between shells (keyword "TEXPN=x") can also be used with the EXTM option. Note that when using EXTM, the thickness of the first shell (keyword "TONE=x") cannot be specified because by fixing the M value and the expansion factor, the thickness of the first shell is already defined. The EXTM option is only available for the force trapezoid algorithm. Since TEXPN is provided only for advanced users, the DEV keyword must accompany these options on the keyword line otherwise the code will stop after reading the keyword line and write an explanation of the problem in the .out file.

The keyword `STDM` is also related to the `EXTM` keyword, and it is explained in the separate keyword section. The keyword `EXTCM` controls an option which allows the advanced user to set the charges for each atom for an entire run. To use this option, the desired charges are placed in the `.dat` file in the columns following the Z-matrix. Since this is an option provided only for advanced users, the keyword `DEV` must accompany the `EXTCM` keyword.

6.3.91. Keyword `TLIMIT=x`

The Keyword `TLIMIT` specifies the maximum CPU time (in seconds) allowed for a calculation.

6.3.92. Keyword `TONE=x` (Developer keyword)

See discussion under keyword `TEXPN`. Note that `TONE` replaces the old keywords `TINIT` and `TSTEP`.

6.3.93. Keyword `TRUES`

The keyword `TRUES` can be used to calculate the true solvation free energy, which is the final heat of formation plus solvation free energy for the geometry in solution minus the final heat of formation for the gas-phase geometry. The keyword `TRUES`, used in a run with one of the solvation model keywords causes the program to calculate this quantity. To use a known or previously calculated heat of formation for the gas-phase geometry, the `TRUES` keyword must be used in conjunction with the `"HF1SCF=x"` or `"HFOPT=x"` keywords.

The `"HF1SCF=x"` keyword specifies that `x` is the heat of formation calculated in the gas-phase by a `1SCF` calculation from the geometry present in the `.dat` or `.inp` file in which the `HF1SCF=x` keyword appears. Similarly, the `"HFOPT=x"` keyword specifies that `x` is the heat of formation calculated in the gas-phase from the optimized gas-phase geometry. (NOTE: When using `HFOPT` or `HF1SCF`, the Hamiltonian used to obtain the heat of formation and optimized geometry should be consistent with the parameterized model to be used for the solvation calculation. In other words, a heat of formation calculated with the AM1 Hamiltonian should be used solvation calculations involving the AM1 Hamiltonian, and a heat of formation calculated with the PM3 Hamiltonian should be used in solvation calculations using the PM3 Hamiltonian. Failure to supply the correct heat of formation can lead to disastrous results!) The `"HF1SCF=x"` and `"HFOPT=x"` keywords have no effect if used without the `TRUES` keyword.

When the keywords `TRUES` and `"HF1SCF=x"` or `"HFOPT=x"` are used together, where `x` is a free formatted number, `x` is assumed to be the heat of formation in the gas-phase in kcal/mol and it is used to calculate the true solvation free energy at the end of the calculation.

A second option is available where the keyword `TRUES` is used with the keyword `"HFCALC=1SCF"` or `"HFCALC=OPT"`. If one of these two combinations is chosen then the calculation will proceed in two steps. First, AMSOL will run a gas-phase calculation. If the keyword `"HFCALC=1SCF"` was used then the heat of formation will be calculated from a `1SCF` calculation and stored. If the keyword `"HFCALC=OPT"` was used then the geometry will be optimized in the gas-phase and the heat of formation at the optimized geometry will be stored. In both cases the `.arc` and `.out` files will be produced by the gas-phase calculations. The heat of formation is stored by being placed in the keyword line of the `.inp` file with the keyword `"HF1SCF=x"` or `"HFOPT=x"` respectively where `x` is the heat of formation calculated in kcal/mol. Then a solvation calculation will be performed in accordance with the solvation keywords

which the user has present on the keyword line and starting at the final gas-phase geometry, and then using the results of both calculations to calculate the true solvation free energy. The solution-phase results are appended to the .arc and .out files. (NOTE: Using the TRUES keyword without the "HF1SCF=x", "HFOPT=x", "HFCALC=1SCF", or "HFCALC=OPT" keywords is the same as using the TRUES keyword with "HFCALC=OPT" keyword. If "HFCALC=1SCF" or "HFCALC=OPT" is used in an XKW file, then after the gas phase calculation the "HF1SCF=x" or the "HFOPT=x" keyword will be added to the .inp file to be read in the solvated calculation, but the HFCALC= keyword will remain in the XKW file. Since the .dat file is read first, the calculation will be performed correctly, but we warn the user that it could be very confusing. The XKW file is never altered from the form submitted by the user.)

The usage of the TRUES keyword, as well as that of the other AMSOL keywords, is illustrated in various test runs. The default of TRUES is NOTRUS, which does not carry out a true solvation energy calculation. The keyword TRUES is compatible with either the OPT or 1SCF keywords but not with the NOPOL keyword.

6.3.94. Keyword TRUSTE

The TRUSTE keyword chooses the TRUST optimizer discussed in Section 4.2.3. Optimization is performed to minimize the energy of the input geometry. In preliminary trials, the TRUSTE optimizer proved to be faster and more robust than the EF optimizer; however it has been tested to a lesser extent than has the default EF optimizer.

6.3.95. Keyword TRUSTG

The TRUSTG keyword chooses the TRUST optimizer discussed in Section 4.2.3. Optimization is performed to minimize the gradient of the input geometry and can be useful when searching for transition states.

6.3.96. Keyword TSTATE (EF keyword)

With the TSTATE keyword, the Eigenvector Following routine is invoked to optimize to a transition state. The TSTATE method is much faster and more reliable than either SADDLE or NLLSQ. TSTATE appears to work well even with Cartesian coordinates.

6.3.97. Keyword VOLUME

The keyword VOLUME is no longer supported.

6.3.98. Keyword XKW (Developer keyword)

The keyword XKW is available to advanced users to allow the use of additional keyword lines or the easy incorporation of keywords into series of trials which use similar keywords. To use this option, a file which must be designated in the amsol[x].c script (where x = i, b, ai, or ab) contains up to four additional lines of keywords. Since every keyword in the file will essentially be dumped into trials which contain the XKW keyword, it can be used to place similar keywords into a whole series of trials. As with all advanced-option keywords, the keyword DEV must also be used. It is not recommended to put the HFCALC=1SCF or HFCALC=OPT keywords in the .xkw file for reasons explained in the TRUES keyword section.

6.3.99. Keyword XYZ

The keyword XYZ is not supported in this version of AMSOL. The XYZ keyword is described inconsistently in the AMPAC-version 2.1 manual.

6.4. Default Settings for Keywords in AMSOL

Note: na denotes not applicable.

Keyword	Default setting
ALPHA	OFF
AREAS	OFF
ASA	ON (Solvation runs only)
BETA	OFF
BFGS	OFF
CART	OFF
CM1	OFF
CM2	OFF
CM3	OFF
DDMAX	EFOLLOW : 0.5 TSTATE : 0.3
DDMIN	EFOLLOW : 0.01
DEV	OFF
DFP	OFF
DIELEC	SOLVNT=x : Dielectric constant of x x=GENALK: 2.06
DMAX	0.2
DOTS	OFF
NDOTCD=x	2610
NDOTEP=x	816
EFOLLOW	ON
EXTCM	OFF
EXTM	OFF
STDN	OFF
EXTSM	OFF
FOCK	OFF
FORWARD	OFF
GAMMA	OFF
GCOMP=x	0.45
HESS=x	EFOLLOW : 0 TSTATE : 1
HFCALC=x	OPT (If TRUES is used.)
IOFR	OFF
INPUT	OFF
IUPD=x	EFOLLOW : 2 TSTATE : 1
KICK=x	DFP and BFGS : KICK=1 EF : KICK=0
MODE=x	EFOLLOW : 0 TSTATE : 1
MSURFT	SOLVNT=x: Surface tension of x x=GENALK:38.9
MULLIK	OFF
NOFOC	ON

Keyword	Default setting
NOINP	ON
NOTRUS	ON
OMIN	Transition State : 0.8 Minimization : na
PRINTGEOM	OFF
PRINTCOUL	OFF
PRINTRAD	OFF
PRINTPOL	OFF
RAD	SM1, SM1A, SM2, SM3 : 0 SM2.1, SM3.1, SM4 : 1 SM2.2 : 2
RECALC=x	OFF
RESTART	OFF
RMAX=x	EFOLLOW : 1000 TSTATE : 4.0
RMIN=x	EFOLLOW : 0 TSTATE : 0.25
RSCAL	OFF
SCFCRT=x	1.0E-6
SM1	OFF
SM1A	OFF
SM2	OFF
SM2.1	OFF
SM2.2	OFF
SM3	OFF
SM3.1	OFF
SM4	OFF
SOLVNT=x	No default
MSURFT=x	38.9
SM5*	OFF
SOLVNT	Solvent and/or parset. No default. Use only with SM4, SM5.4A, or SM5.4P.
SVCDRD=x	CD radius of chosen solvent water=1.4, alkanes=2.0 Angstroms
SVCSRDX	CS radius of chosen solvent water=na, alkanes=4.9 Angstroms
TLIMIT=x	65000 [3600 if compiled with amsol*ccl.cray]
TDUMP	1800
TEXPN=x	SM1 & SM1A : 0.10 SM2 & SM3 : 1.5 SM2.1, SM3.1, & SM4 : 1.2
TONE=x	SM1 & SM1A : 0.10 SM2 & SM3 : 0.02 SM2.1, SM3.1, & SM4 : 0.15
TRUES	OFF
TSTATE	OFF
XKW	OFF

6.5 Keyword Usage in Trial Runs

The follow table shows which trial runs use non-default keywords.

Keyword	Trial Runs Using Keyword
ALPHA=x	01n, 05n, 08n, 09n, 10n, 12n, 13n, 14n, 16n, 28n, 30n, 31n, 34n, 35f, 35n, 36n, 38n, 39n, 40, 42n
AREAS	01n, 02n, 02n, 03n, 07a2, 08n, 14a, 24a, 24a2, 29a, 35f, 40
ASA	01a, 01n, 02a2, 03a, 03n, 04a, 04n, 05a, 08a, 08n, 09a2, 09n, 10a, 10n, 11a, 11n, 12a, 12n, 13a, 13n, 14a, 14n, 15a, 15n, 17a, 19a2, 20a, 21a, 22a, 23a, 24a, 24a2
BETA=x	01n, 05n, 08n, 09n, 10n, 12n, 13n, 14n, 16n, 28n, 30n, 31n, 34n, 35f, 35n, 36n, 38n, 39n, 40, 42n
BFGS	09, 15, 15a
CART	07a, 41
CM1	06, 09, 14
CM2	41z
CM3	1CM3A - 11CM3A, 1CM3P - 12CM3P
DERINU	01a, 02a, 02a2, 02n, 03a, 03n, 04a, 05a, 08a, 09, 09a, 09a2, 10a, 11a, 11n, 13a, 14, 15, 15a, 15n, 16a, 17a, 19a, 19a2, 22a, 22a3, 22a4, 22a5, 22a7, 22a8, 23a, 25a3, 41, 41z
DERISA	01, 01n, 02, 03, 04, 05, 05n, 07a, 08, 08n, 09n, 10, 10n, 11, 12, 12a, 12n, 13, 13n, 14a, 16, 17, 18, 18a, 19, 20, 20a, 21, 21a, 22, 22a2, 23, 24, 24a
DEV	02n, 06a2, 07a2, 12a, 22a2, 22a3, 22a4, 22a6, 26a2
DFP	15n
DIELEC	01n, 05n, 06a2, 07a2, 07n, 08n, 09n, 10n, 12n, 13n, 14n, 16n, 28n, 30n, 31n, 34n, 35f, 35n, 36n, 38n, 39n, 40, 42n
DOTS	19a
EFOLLOW	07
EXTCM	22a4, 26a2
EXTM	22a5
EXTSM	06a2, 07a2, 22a3
FACARB=x	28n, 30n, 31n, 34n, 35f, 35n, 36n, 38n, 39n, 40, 42n
FEHALO=x	28n, 30n, 31n, 34n, 35f, 35n, 36n, 38n, 39n, 40, 42n
FOCK	21, 21a
FORWRD	07
GAMMA=x	01n, 05n, 08n, 09n, 10n, 12n, 13n, 14n, 16n, 28n, 30n, 31n, 34n, 35f, 35n, 36n, 38n, 39n, 40, 42n
GCOMP	13, 22a, 22a5
HESS=x	14, 24a
HFCALC=x	01a, 02n, 10n, 27a, 28a, 31a, 31n, 32a, 33a, 34a, 34n, 35a, 35f, 35n, 36a, 36n, 37a, 38n, 39a, 39n, 40, 42a, 42n
INPUT	03a, 03n, 06, 07, 09a2, 09n, 16, 19, 22a2, 26a, 26a2
IOFR=x	01n, 05n, 08n, 09n, 10n, 12n, 13n, 14n, 16n, 28n, 30n, 31n, 34n, 35f, 35n, 36n, 38n, 39n, 40, 42n
KICK=x	07, 13a
MSURFT	06a2, 07a2, 07n
MULLIK	02
NDOTCD=x	19a
NDOTEP=x	19a
NOINP	01, 01n, 02, 02a2, 03, 04, 04a, 04n, 05, 05a, 05n, 08, 08a, 08n, 09, 09a, 10, 10a, 10n, 11, 11a, 11n, 12, 12a, 12n, 13, 13a, 13n, 14a, 14n, 15a, 15n, 17, 17a, 18, 18a, 19a, 20, 20a, 21, 21a, 22, 23, 23a, 24, 24a, 24a2

Keyword	Trial Runs Using Keyword
NDOTCD=x	19a
NDOTEP=x	19a
NOINP	01, 01n, 02, 02a2, 03, 04, 04a, 04n, 05, 05a, 05n, 08, 08a, 08n, 09, 09a, 10, 10a, 10n, 11, 11a, 11n, 12, 12a, 12n, 13, 13a, 13n, 14a, 14n, 15a, 15n, 17, 17a, 18, 18a, 19a, 20, 20a, 21, 21a, 22, 23, 23a, 24, 24a, 24a20
MIN=x	24a
PRINTGEOM	14a
PRINTCOUL	01a, 02a, 03a, 04a, 05n, 06a, 06a2, 08a, 09a, 10a, 14a
PRINTRAD	01a, 02a, 03a, 04a, 05n, 06a, 06a2, 08a, 09a, 10a, 14a
PRINTPOL	01a, 02a, 03a, 04a, 05n, 06a, 06a2, 08a, 09a, 10a, 14a
RAD=x	1a, 02a, 03a, 04a, 05n, 06a, 06a2, 08a, 09a, 10a, 14a, 22a3
RECALC=x	14, 22a
SCFCRT=x	03, 08a, 08n, 09a2, 09n, 11a, 11n, 17a, 18a
SM1	04a, 10a
SM1A	05a, 08a, 13a
SM2	01a, 03a, 17a, 19a, 22a3
SM2.1	22a4, 22a5, 25a
SM2.2	06a, 25a2
SM2.2PDA	09a, 23a
SM3	02a, 11a, 16a, 22a
SM3.1	02a2, 19a2, 22a7
SM4	03n, 06a2, 06n, 07a2, 07n, 11n, 15n
SM5.0R	25a3, 28n, 29a, 38a
SM5.05R	22a8
SM5.2	18d, 43
SM5.2R	27a, 28a, 35a, 35f, 35n, 37a, 38n, 39a, 39n
SM5.42	41, 41d, 41z
SM5.42R	30a, 30n, 31a, 31n, 32a, 33a, 34a, 34n, 36a, 36n, 40, 42a, 42n
SM5.4A	01n, 04n, 07a, 08n, 09n, 10n, 12n, 20a, 21a, 24a, 26a, 26a2
SM5.4P	02n, 05n, 13n, 14n, 15a, 16n, 24a2
SOLVNT=x	01n, 02n, 03n, 04n, 05n, 06a2, 06n, 07a, 07a2, 07n, 08n, 09a2, 09n, 10n, 11n, 12a, 12n, 13n, 14a, 14n, 15a, 15n, 16n, 18a, 20a, 21a, 22a2, 22a8, 24a, 24a2, 25a3, 26a, 26a2, 27a, 28a, 28n, 29a, 30a, 30n, 31a, 31n, 32a, 33a, 34a, 34n, 35a, 35f, 35n, 36a, 36n, 37a, 38a, 38n, 39a, 39n, 40, 41, 41z, 42a, 42n
STDM	22a4
SVCDRD	06a2, 07a2
SVCSR	07a2
TDUMP=x	07n
TLIMIT=x	05a, 05n, 08a, 08n, 11a, 11n
TRUES	01a, 01n, 02a, 02a2, 02n, 06a, 06a2, 06n, 07a, 07a2, 07n, 10n, 19a, 19a2, 22a, 25a, 25a2, 27a, 28a, 30a, 30n, 31a, 31n, 32a, 33a, 34a, 34n, 35a, 35f, 35n, 36a, 36n, 37a, 38n, 39a, 39n, 40, 41, 41z, 42a, 42n
TRUSTE	04n, 06n, 18a
TRUSTG	16n, 23a, 24a2
TSTATE	16, 24, 24a
XKW	22z

7. COMPUTATIONAL DETAILS

Most algorithmic details are published in journal articles, but this section contains some additional notes that may be useful.

7.1. Numerical Precision Table

The tables in this subsection are provided to help the user to choose which surface area algorithm and which settings within that algorithm should be used to obtain the desired compromise of precision and speed for a given application. The values were obtained using AMSOL 4.6 and will not change appreciably in AMSOL-version 7.0.

When cost is not a problem, the reader should just use ASA and ignore this section.

All numerical precision values in the tables are referenced to the ASA algorithm running on the same machine. The numerical precision values are given in kcal/mol and represent the average, for the applicable portion of the test suite, 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 choices and version specified in columns 1 and 4.

The testing set contains fifteen molecules from the original AMSOL test suite. There are 12 neutrals in the test set and 3 ions. These runs used the BFGS optimization scheme.

There are two tables. The first table gives information on the algorithms when run on the Cray C90 (a vector machine). The second table gives information on the algorithms when run on the IBM RS/6000 Model 550 (a superscalar machine).

The default for AMSOL-version 7.0 is ASA.

Table 1. Vector Machine Results (Cray C90)

Algorithm	# of Points for the Coulombic Calculation	#of Points for the Surface Tension Calculation	Version	Numerical Precision for the Ions (kcal/mol)	Numerical Precision for the Neutrals (kcal/mol)
DOTS	816	2610	4.6	0.14	0.06
DOTS	258	258	4.6	0.4	0.18
DOTS	66	66	4.6	0.6	0.27
ASA	---	---	4.6	0.0	0.0

Table 2. Scalar Machine Results (IBM RS/6000 Model 550)

Algorithm	# of Points for the Coulombic Calculation	#of Points for the Surface Tension Calculation	Version	Numerical Precision for the Ions (kcal/mol)	Numerical Precision for the Neutrals (kcal/mol)
DOTS	816	2610	4.6	0.22	0.06
DOTS	258	258	4.6	0.45	0.15

ASA

4.6

0.0

0.0

7.2. Observed Discontinuities in the Gradient of Solvent-Accessible Surface Areas (SASAs) of Atoms

AMSOL calculates the SASA of each atom in a molecule based on the geometry of the molecule and on the set of atom-centered spheres of known radii. The SASAs, which are used to compute the effective Born radii and the G-CDS contribution to the free energy of solvation, are calculated in AMSOL with the ASA algorithm; see "Improved Methods for Semiempirical Solvation Models" by D. A. Liotard, G. D. Hawkins, G. C. Lynch, C. J. Cramer, and D. G. Truhlar, *Journal of Computational Chemistry* 16, 422-440 (1995). In general, the SASA of an atom is a continuous function of the geometry of the molecule. However, a discontinuity in the gradient of the SASA can occur in a few special cases. In particular, this discontinuity has been observed when the geometry of the molecule yields two atom-centered spheres that simultaneously come into contact with a third atom-centered sphere. This does not affect the calculation of the gradients of the Born radii, but it does affect the calculation of the gradient of the G-CDS contribution to the free energy of solvation. As a consequence, geometry optimizations that employ the originally published algorithm and that lead to geometries that produce this type of discontinuity always fail, regardless of the optimizer used (EF, TRUSTE, etc.).

These discontinuities, or cusps, cannot be avoided as long as the atom-centered spheres are true spheres. The cusps can be minimized, however, if small, sharp projections, or nibs, are raised out from the surface of the spheres. To obtain continuous derivatives of higher order, the "sharpness" of the nibs must be increased.

Starting with AMSOL version 6.9, a smoothing procedure has been added to the original ASA algorithm that allows the SASA to be a continuous function of the geometry up to the second derivative. The improved algorithm introduces these nibs to atom-centered spheres that are less than 0.001 angstroms from one another.

8. OUTPUT

The use of a solvation model keyword will automatically ensure that the accessible-surface-area terms are printed out in kcal/mol, by atom as well as summed over atoms. The effective Born radii and effective interatomic distances are printed out in block matrix form. The generalized Born polarization energy decomposition is also printed in block matrix form. Note that these energies will be significantly more negative than the actual electronic contribution to the free energy of solvation since there is an energy loss associated with the internal solute energy due to electronic reorganization. That is, the formalism optimizes to the minimum of the sum of the internal solute energy and the generalized Born polarization energy.

The output file also contains a date/time stamp, as explained in Section 5. This date/time stamp is the date and time at the beginning of execution. It is printed two places on the output file (.out file) and one place on the archive file (.arc file); this should be helpful in making a unique correspondence between these files if they should ever get separated or confused.

9. FILE USAGE

AMSOL reads input from 1 to 6 files, depending on the options chosen. The possible input files are as follows:

FORTRAN Unit Number	Extension	Type	Description
5	.dat	ASCII	This file contains standard data for a run
9	.res	Binary	This file contains the restart data for a run. It is read when the keyword RESTART is used.
10	.den	Binary	This file contains the density matrix in binary format and is read if either the keyword RESTART or the keyword OLDENS is used.
15	.dmt	ASCII	This file contains the density matrix in ASCII text format and is read if the keyword OLDMAT is used.
19	.xsm	ASCII	This file contains the solvation parameters for a run.
20	.xkw	ASCII	This file contains additional keywords for a run.

AMSOL runs produce from 3 to 8 output files, depending on the options chosen. The possible output files are as follows:

FORTTRAN Unit Number	Extension	Type	Description
*	.com	Commands	Contains echos of commands submitted by the script and communications from the operating system, This file is created whenever one of the two job submission scripts is used.
6	.out	ASCII	This file contains the long output file and is always created.
9	.res	Binary	Restart file that is created if the execution time exceeds the value of TDEF which is set in SIZES.i or SIZESc.i or if the number of SCF cycles for a geometry exceeds the default value for a particular set of options or the value set by the keyword CYCLES.
10	.den	Binary	A file containing the density matrix that is created if the restart file is created.
12	.arc	ASCII	Archive file (summary of .out file).
13	.gpt	Binary	A file containing data for the program DENSITY. This file is created when the keyword GRAPH is used.
18	.inp	ASCII	A new data file containing the optimized geometry that is created when the keyword INPUT is used.
21	-	ASCII	A scratch file used during the calculation, removed when calculation completes.

10. TEST RUN INPUT AND OUTPUT

The keyword and comment lines from the .dat files of the test runs are provided here as a reference. Notice that in the present version of AMSOL, a .dat file always begins with one or two lines of keywords followed by two comment lines. If the second line of a .dat file begins with an ampersand (&) then this line is assumed to be a second keyword line. Otherwise it is assumed to be the first comment line. In the test suite, the second comment line is used for several test runs, and it is blank for all the others. This blank line is, nevertheless, required.

For every test run the .dat file and--in each case--either the associated .out file or the .arc file provided as part of the distributed version as enumerated in Section 8. The output in the /testo directory is generated using the machine, operating system, and compiler listed in Section 15. A list of the keywords used in the test runs is provided in Section 6.5. Note that many of the keywords specified in the test runs are defaults, but we specify them for clarity and to illustrate their use. More information on the use of keywords in the trial runs is supplied in Section 6.3.

The data files provided in the distribution package are the files that were run successfully on all the machines on which the code has been tested. Using these data files gave energies that were always within 0.2 kcal/mol (and usually within 0.1 kcal/mol) across all the machines in the test runs we made (see Section 13). In a few cases on a few machines, the largest component of the gradient remained above the 0.45 threshold, but the energy was within 0.2 kcal/mol of converged runs on other machines. The gradient component values can usually be reduced on any given machine by perturbing the input geometry, but different perturbations are required on different machines.

As discussed in Section 4.2, the difficulty of geometry optimization often makes it necessary to perturb the starting geometry for solvation calculations to obtain convergence. Because of this problem, certain test runs within the distributed test suite may exhibit this behavior when running on machines other than an IBM SP Power 3. If the optimization does not terminate normally, but the gradient norm is low, then the effect on the overall free energy is probably still within the stated difference from machine to machine. Otherwise, the starting geometry should be perturbed until the geometry optimization is able to converge. The data files included in the distribution package are those used to run the AMSOL test suite on an IBM SP Power 3, and the .arc and the .out files produced in those runs are included as part of the distribution package.

In order to prevent distributed output files from being overwritten by user test runs, care should be taken to store the provided output files in a different subdirectory from the one in which user calculations are being performed. By preserving the distributed files, they can be compared with user results.

The keyword and comment lines from each test run in the test suite are as follows:

tr01.dat:
AM1 SYMMETRY NOINP OPT DERISA
ammonia (gas phase)

tr01a.dat:
AM1 HFCALC=OPT SM2 DERINU SYMMETRY TRUES
& ASA OPT PRINTPOL PRINTRAD PRINTCOUL
ammonia (aqueous)

tr01n.dat:
AM1 SM5.4A SYMMETRY TRUES NOINP HFOPT=-7.283460 DERISA ASA OPT AREAS
& IOFR=1.3575 ALPHA=0.00 BETA=0.00 GAMMA=22.3 DIELEC=1.84 SOLVNT=GENORG
ammonia (pentane)

tr02.dat:
PM3 SYMMETRY NOINP OPT DERISA
& MULLIK
water (gas phase)

tr02a.dat:
PM3 SM3 DERINU SYMMETRY TRUES 1SCF HF1SCF=-53.426475
& PRINTPOL PRINTRAD PRINTCOUL
water (aqueous)

tr02a2.dat:
PM3 SM3.1 DERINU SYMMETRY TRUES OPT HFOPT=-53.426475
& ASA NOINP
water (aqueous)

tr02n.dat:
1SCF PM3 SM5.4P DEV DERINU SYMMETRY AREAS HFCALC=OPT TRUES
& SOLVNT=BENZENE AREAS
water (solvent : xylene)

tr03.dat:
SCFCRT=0.000001 AM1 UHF
& NOINP DERISA OPT
nitrogen oxide (gas-phase)

tr03a.dat:
DERINU SM2 AM1 UHF OPT
& ASA INPUT PRINTPOL PRINTRAD PRINTCOUL
nitrogen oxide (aqueous)

tr03n.dat:
DERINU SM4 AM1 UHF OPT SOLVNT=N7ANE AREAS
& ASA INPUT
nitrogen oxide (heptane)

tr04.dat:
AM1 SYMMETRY NOINP OPT DERISA
diethyl ether (gas phase)

tr04a.dat:
AM1 SM1 DERINU SYMMETRY OPT
& ASA NOINP PRINTPOL PRINTRAD PRINTCOUL
diethyl ether (aqueous)

tr04n.dat:
AM1 SYMMETRY SM5.4A SOLVNT=TOLUENE
& ASA NOINP TRUSTE
diethyl ether (solvent : toluene)

tr05.dat:
AM1 NOINP DERISA OPT
Rotation coordinate (30 deg inc.) for hydroxyl of ethanoic acid (gas
phase)

tr05a.dat:
DERINU SM1A TLIMIT=200000 AM1
& ASA NOINP OPT
Rotation coordinate (30 deg increment) for hydroxyl of ethanoic acid
(aqueous)

tr05n.dat:
DERISA SM5.4P TLIMIT=200000 PM3 NOINP OPT PRINTPOL PRINTRAD PRINTCOUL
& IOFR=1.4054 ALPHA=0.00 BETA=0.00 GAMMA=32.2 DIELEC=1.96 SOLVNT=GENORG
Rotation coordinate (30 deg increment) for hydroxyl of ethanoic acid
(nonane)

tr06.dat:
AM1 SYMMETRY INPUT CM1
1,2-ethanediol (all trans) (gas phase)

tr06a.dat:
AM1 SYMMETRY TRUES SM2.2 HFOPT=-107.495881 PRINTPOL PRINTRAD PRINTCOUL
1,2-ethanediol (all trans) (aqueous phase)

tr06a2.dat:
AM1 SYMMETRY TRUES SM4 SOLVNT=H2OSRP DEV DIELEC=78.3
& MSURFT=0.0 SVCDRD=1.4 EXTSM HFOPT=-107.495881 PRINTPOL PRINTRAD
PRINTCOUL
1,2-ethanediol (all trans) (aqueous phase AM1-SM4-[HCO]SRP-(sugar)
model)

tr06n.dat:
AM1 SYMMETRY TRUES SM4 SOLVNT=3M5ANE HFOPT=-107.495881 TRUSTE
1,2-ethanediol (all trans) (3-methylpentane)

tr07.dat:
EFOLLOW AM1 INPUT KICK=0 FORWRD
allyl vinyl ether (gas phase)

tr07a.dat:
DERISA AM1 SM5.4A SOLVNT=WATER TRUES HFOPT=-5.396552 CART
allyl vinyl ether (aqueous phase)

tr07a2.dat:
AM1 SYMMETRY TRUES SM4 SOLVNT=H2OSRP DEV DIELEC=78.3
& MSURFT=0.0 SVCDRD=1.4 EXTSM SVCSRD=0.0 AREAS HFOPT=-5.396552
allyl vinyl ether (aqueous AM1-SM4-[HCO]SRP-(ether/aldehyde))

tr07n.dat:
AM1 SYMMETRY TRUES SM4 SOLVNT=GENALK DIELEC=2.00
& MSURFT=38.0 TDUMP=50 HFOPT=-5.396552
allyl vinyl ether (General Alkane)

tr08.dat:
AM1 SYMMETRY NOINP OPT DERISA
morpholine (Cs symmetry) (gas phase)

tr08a.dat:
TLIMIT=200000 DERINU SM1A AM1 SYMMETRY SCFCRT=1.0D-6
& ASA NOINP OPT PRINTPOL PRINTRAD PRINTCOUL
morpholine (Cs symmetry) (aqueous)

tr08n.dat:
TLIMIT=200000 DERISA AM1 SYMMETRY SCFCRT=1.0D-6 SM5.4A ASA NOINP OPT
AREAS
& IOFR=1.4102 ALPHA=0.00 BETA=0.00 GAMMA=33.6 DIELEC=1.99 SOLVNT=GENORG
morpholine (Cs symmetry) (decane)

tr09.dat:
BFGS AM1 SYMMETRY DERINU
& NOINP CM1
4-pyridone (gas phase)

tr09a.dat:
SM2.2PDA AM1 SYMMETRY DERINU
& NOINP NOPOL PRINTPOL PRINTRAD PRINTCOUL
4-pyridone (aqueous phase)

tr09a2.dat:
SM5.2PDA DERINU CYCLES=50 SCFCRT=1.0D-6 AM1 SYMMETRY
& ASA INPUT OPT SOLVNT=WATER
4-pyridone (aqueous)

tr09n.dat:
DERISA CYCLES=50 SCFCRT=1.0D-6 AM1 SYMMETRY SM5.4A GAMMA=39.0
SOLVNT=GENORG
& ASA INPUT OPT IOFR=1.3720 ALPHA=0.61 BETA=0.44 DIELEC=6.25
4-pyridone (solvent : acetic acid)

tr10.dat:
AM1 SYMMETRY NOINP DERISA OPT
Z-1,2-dichloroethylene (C2v symmetry) (gas phase)

tr10a.dat:
DERINU SM1 AM1 SYMMETRY
& ASA NOINP OPT PRINTPOL PRINTRAD PRINTCOUL
Z-1,2-dichloroethylene (C2v symmetry) (aqueous)

tr10n.dat:
DERISA AM1 SYMMETRY SM5.4A IOFR=1.3611 ALPHA=0.37 BETA=0.48
GAMMA=31.6
& ASA NOINP OPT DIELEC=24.85 HFCALC=1SCF TRUES SOLVNT=GENORG
Z-1,2-dichloroethylene (C2v symmetry) (solvent : ethanol)

tr11.dat:
PM3 SYMMETRY NOINP DERISA OPT
methylcyclohexane (gas phase)

tr11a.dat:
TLIMIT=200000 SM3 DERINU SCFCRT=1D-5 PM3 SYMMETRY OPT
& ASA NOINP
methylcyclohexane (aqueous)

tr11n.dat:
TLIMIT=200000 SM4 DERINU SCFCRT=1D-5 PM3 SYMMETRY OPT
& ASA NOINP SOLVNT=N14ANE
methylcyclohexane (tetradecane)

tr12.dat:
AM1 SYMMETRY NOINP DERISA OPT
methyl butanoate (Cs symmetry) (gas phase)

tr12a.dat:
DERISA AM1 SYMMETRY OPT SM5.4PDU DEV SOLVNT=WATER
& ASA NOINP
methyl butanoate (Cs symmetry) (aqueous)

tr12n.dat:

DERISA SM5.4A AM1 SYMMETRY OPT ASA NOINP
& IOFR=1.4315 ALPHA=0.00 BETA=0.00 GAMMA=38.3 DIELEC=2.03 SOLVNT=GENORG
methyl butanoate (Cs symmetry) (pentadecane)

tr13.dat:

AM1 NOINP OPT DERISA GCOMP=0.40
1,1,1-trifluoropropan-2-ol (gas phase)

tr13a.dat:

DERINU SM1A AM1
& ASA NOINP OPT KICK=2
1,1,1-trifluoropropan-2-ol (aqueous)

tr13n.dat:

DERISA SM5.4P PM3 ASA NOINP OPT
& IOFR=1.4345 ALPHA=0.00 BETA=0.00 GAMMA=38.9 DIELEC=2.06 SOLVNT=GENORG
1,1,1-trifluoropropan-2-ol (hexadecane)

tr14.dat:

PM3 DERINU CM1 GEO-OK HESS=1 RECALC=1
3-ethyl-2-methoxypyrazine (gas phase)

tr14a.dat:

PM3 GEO-OK SM5.4PDP SOLVNT=WATER
& ASA NOINP DERISA AREAS PRINTPOL PRINTRAD PRINTCOUL PRINTGEOM
3-ethyl-2-methoxypyrazine (aqueous)

tr14n.dat:

PM3 GEO-OK IOFR=1.4295 ALPHA=0.37 BETA=0.48 GAMMA=39.0 DIELEC=9.87
& ASA NOINP SM5.4P SOLVNT=GENORG
3-ethyl-2-methoxypyrazine (solvent : octanol)

tr15.dat:

BFGS PM3 DERINU
3-ethyl-2-methoxypyrazine (gas phase)

tr15a.dat:

BFGS SM5.4P 1SCF PM3 SOLVNT=WATER
& ASA NOINP DERINU
3-ethyl-2-methoxypyrazine (aqueous)

tr15n.dat:

DFP SM4 1SCF PM3 SOLVNT=MCYC6
& ASA NOINP DERINU
3-ethyl-2-methoxypyrazine (methylcyclohexane)

tr16.dat:
TSTATE PM3 SYMMETRY OPT DERISA
& INPUT ITER
TS of Diels-Alder reaction cyclopentadiene and methylvinylketone (gas phase)

tr16a.dat:
POWELL PM3 SYMMETRY SM3 DERINU OPT
TS of Diels-Alder reaction cyclopentadiene and methylvinylketone (aqueous)

tr16n.dat:
PM3 SM5.4P SYMMETRY OPT TRUSTG
&IOFR=1.4513 ALPHA=0.00 BETA=0.00 GAMMA=26.4 DIELEC=1.94 SOLVNT=GENORG
TS of Diels-Alder reaction cyclopentadiene and methylvinylketone (isooctane)

tr17.dat:
AM1 SYMMETRY NOINP OPT DERISA
phosphine (gas phase)

tr17a.dat:
SM2 DERINU SCFCRT=1D-6 AM1 SYMMETRY OPT ASA NOINP
phosphine (aqueous)

tr18.dat:
AM1 SYMMETRY NOINP OPT DERISA
phosphinous acid, rotation coordinate for P-O bond (gas phase)

tr18a.dat:
SM5.2PDA DERISA SCFCRT=1D-6 AM1 SYMMETRY OPT TRUSTE
& NOINP SOLVNT=WATER
phosphinous acid, rotation coordinate for P-O bond (aqueous)

tr18d.dat:
SM5.2 DERISA SCFCRT=1D-6 AM1 SYMMETRY OPT TRUSTE
& NOINP SOLVNT=WATER
phosphinous acid, rotation coordinate for P-O bond (aqueous)

tr19.dat:
CHARGE=-1 AM1 UHF
& INPUT DERISA OPT
superoxide anion (gas-phase)

tr19a.dat:
DERINU SM2 CHARGE=-1 AM1 UHF TRUES HFOPT=-22.819247

& DOTS NDOTCD=2610 NDOTEP=816 NOINP OPT
superoxide anion (aqueous)

tr19a2.dat:
DERINU SM3.1 CHARGE=-1 PM3 UHF TRUES
& ASA OPT
superoxide anion (aqueous)

tr20.dat:
AM1 CHARGE=-1 NOINP OPT DERISA
chloride anion (gas phase)

tr20a.dat:
SM5.4A AM1 CHARGE=-1 NOINP OPT DERISA
& ASA SOLVNT=WATER
chloride anion (aqueous)

tr21.dat:
CHARGE=1 AM1 SYMMETRY FOCK DERISA OPT
& NOINP
dimethylammonium cation (C2v symmetry) (gas phase)

tr21a.dat:
DERISA SM5.4A SOLVNT=WATER CHARGE=1 AM1 SYMMETRY FOCK
& CYCLES=60 ASA NOINP OPT POWELL
dimethylammonium cation (C2v symmetry) (aqueous)

tr22.dat:
CHARGE=-1 PM3 SYMMETRY NOINP OPT DERISA
acetate anion (Cs symmetry) (gas phase)

tr22a.dat:
DERINU SM3 CHARGE=-1 PM3 SYMMETRY OPT TRUES
& ASA RECALC=7 GCOMP=1.0
acetate anion (Cs symmetry) (aqueous)

tr22a2.dat:
DERISA SM5.4U DEV CHARGE=-1 AM1 INPUT SOLVNT=WATER
acetate anion (Cs symmetry) (aqueous)

tr22a3.dat:
DERINU SM2 CHARGE=-1 AM1 SYMMETRY RAD=1 EXTSM DEV
acetate anion (Cs symmetry) (aqueous)

tr22a4.dat:
DERINU SM2.1 CHARGE=-1 AM1 SYMMETRY DEV EXTCM STD
acetate anion (Cs symmetry) (aqueous)

tr22a5.dat:
DERINU SM2.1 CHARGE=-1 AM1 SYMMETRY EXTM GCOMP=0.750
acetate anion (Cs symmetry) (aqueous)

tr22a6.dat:
DEV XKW
acetate anion (Cs symmetry) (aqueous)

tr22a7.dat:
DERINU SM3.1 CHARGE=-1 PM3 SYMMETRY
acetate anion (Cs symmetry) (aqueous)

tr22a8.dat:
DERINU SOLVNT=WATER SM5.05R CHARGE=-1
ATOM 4 7

tr23.dat:
CYCLES=30 POWELL AM1 CHARGE=-1 SYMMETRY OPT
& NOINP DERISA
bromide methyl iodide (SN2 reaction transition state) (gas phase)

tr23a.dat:
SM2.2PDA DERINU CYCLES=30 TRUSTG AM1 CHARGE=-1 SYMMETRY
& ASA NOINP OPT
bromide methyl iodide (SN2 reaction transition state) (aqueous)

tr24.dat:
CYCLES=30 TSTATE AM1 CHARGE=-1 SYMMETRY OPT
& NOINP DERISA
bromide methyl iodide (SN2 reaction transition state) (gas phase)

tr24a.dat:
SM5.4A SOLVNT=WATER DERISA TSTATE AM1 CHARGE=-1 SYMMETRY
& ASA NOINP OPT HESS=3 OMIN=.9 AREAS
bromide methyl iodide (SN2 reaction transition state) (aqueous)

tr24a2.dat:
SM5.4P PM3 SOLVNT=WATER CHARGE=-1 SYMMETRY
& ASA NOINP OPT TRUSTG AREAS
bromide methyl iodide (SN2 reaction transition state) (aqueous)

tr25a.dat:
AM1 SM2.1 TRUES HFOPT=-53.710625
p-hydroxybenzaldehyde

tr25a2.dat:

AM1 SM2.2 TRUES HFOPT=-53.710625
p-hydroxybenzaldehyde

tr25a3.dat:

DERINU SM5.0R SOLVNT=WATER
p-hydroxybenzaldehyde

tr26a.dat:

AM1 SM5.4A SOLVNT=WATER NOPOL CHARGE=-1 INPUT
NH2-

tr26a2.dat:

AM1 SM5.4A SOLVNT=WATER DEV EXTCM NOPOL CHARGE=-1 INPUT
NH2-

tr27a.dat:

SOLVNT=WATER TRUES AM1 SM5.2R 1SCF HFCALC=1SCF
dimethyl 5-(4-chloro)bicyclo[3.2.0]heptyl phosphate in water

tr28a.dat:

SOLVNT=WATER TRUES PM3 SM5.2R 1SCF HFCALC=1SCF
bromotoluene in water

tr28n.dat:

SM5.0R
& IOFR=1.5562 ALPHA=0 BETA=0.28 GAMMA=62.54 DIELEC=34.81
& FACARB=0.67 FEHALO=0.00 SOLVNT=GENORG
bromotoluene in nitrobenzene

tr29a.dat:

SOLVNT=WATER SM5.0R AREAS
trichloroethene in water

tr30a.dat:

SOLVNT=WATER TRUES AM1 SM5.42R 1SCF HF1SCF=CALC
dipropyl sulfide in water

tr30n.dat:

TRUES AM1 SM5.42R 1SCF HF1SCF=CALC
& IOFR=1.4684 ALPHA=0 BETA=0.1 GAMMA=38.37 DIELEC=5.47
& FACARB=0.86 FEHALO=0.14 SOLVNT=GENORG
dipropyl sulfide in fluorobenzene

tr31a.dat:

SOLVNT=WATER TRUES PM3 SM5.42R 1SCF HFCALC=1SCF
ethanamide in water

tr31n.dat:

TRUES PM3 SM5.42R 1SCF HFCALC=1SCF
& IOFR=1.4994 ALPHA=0 BETA=0.19 GAMMA=39.65 DIELEC=2.27
& FACARB=0.67 FEHALO=0.00 SOLVNT=GENORG
ethanamide in mesitylene

tr32a.dat:

TRUES SM5.42R AM1 1SCF HFCALC=1SCF SOLVNT=WATER
butanonitrile in water

tr33a.dat:

TRUES SOLVNT=WATER SM5.42R AM1 1SCF HFCALC=1SCF
methyl hexanoate in water

tr34a.dat:

SOLVNT=WATER TRUES PM3 SM5.42R 1SCF HFCALC=1SCF
butanoic acid in water

tr34n.dat:

TRUES PM3 SM5.42R 1SCF HFCALC=1SCF
& IOFR=1.4295 ALPHA=0.37 BETA=0.48 GAMMA=39.01 DIELEC=9.87
& FACARB=0.00 FEHALO=0.00 SOLVNT=GENORG
butanoic acid in water

tr35a.dat:

SOLVNT=WATER TRUES PM3 SM5.2R 1SCF HFCALC=1SCF
5-nonanone in water

tr35f.dat:

SOLVNT=GENORG TRUES PM3 SM5.2R 1SCF HFCALC=1SCF AREAS
& DIELEC=9.86 IOFR=1.4295 ALPHA=0.37
& BETA=0.48 GAMMA=39.01 FEHALO=0.00 FACARB=0.00
5-nonanone in water

tr35n.dat:

TRUES PM3 SM5.2R 1SCF HFCALC=1SCF
& IOFR=1.4345 ALPHA=0 BETA=0 GAMMA=38.93 DIELEC=2.06
& FACARB=0.00 FEHALO=0.00 SOLVNT=GENORG
5-nonanone in hexadecane

tr36a.dat:

SOLVNT=WATER TRUES AM1 SM5.42R 1SCF HFCALC=1SCF
tetrahydropyran in water

tr36n.dat:

TRUES AM1 SM5.42R 1SCF HFCALC=1SCF
& IOFR=1.4459 ALPHA=0.15 BETA=0.02 GAMMA=38.39 DIELEC=4.71
& FACARB=0.00 FEHALO=0.75 SOLVNT=GENORG
tetrahydropyran in chloroform

tr37a.dat:

SOLVNT=WATER TRUES MNDO SM5.2R 1SCF HFCALC=1SCF
cyclopentanol in water

tr38a.dat:

SOLVNT=WATER SM5.0R
anthracene in water

tr38n.dat:

TRUES AM1 SM5.2R 1SCF HFCALC=1SCF
& IOFR=1.3878 ALPHA=0 BETA=0 GAMMA=28.28 DIELEC=1.91
& FACARB=0.00 FEHALO=0.00 SOLVNT=GENORG
anthracene in heptane

tr39a.dat:

SOLVNT=WATER TRUES SM5.2R AM1 1SCF HFCALC=1SCF
1-pentyne in water

tr39n.dat:

TRUES SM5.2R MNDO 1SCF HFCALC=1SCF
& IOFR=1.5095 ALPHA=0 BETA=0.52 GAMMA=52.62 DIELEC=12.98
& FACARB=0.83 FEHALO=0.00 SOLVNT=GENORG
1-pentyne in pyridine

tr40.dat:

1SCF AM1 SOLVNT=GENORG TRUES SM5.42R HFCALC=1SCF AREAS
& DIELEC=9.86 IOFR=1.4295 ALPHA=0.37
& BETA=0.48 GAMMA=39.01 FEHALO=0.00 FACARB=0.00
piperazine in 1-octanol

tr41.dat:

AM1 SM5.42 CART SOLVNT=WATER TRUES DERINU
methanol in water

tr41d.dat:

AM1 SM5.42 CART SOLVNT=WATER TRUES DERISA
methanol in water

tr41z.dat:

AM1 SM5.42 SOLVNT=WATER TRUES DERINU
methanol in water

tr42a.dat:

SOLVNT=WATER TRUES SM5.42R AM1 1SCF HFCALC=1SCF
orthosilicic acid in water

tr42n.dat:

PM3 IOFR=1.4295 ALPHA=0.37 BETA=0.48 GAMMA=39.0 DIELEC=9.87
& FEHALO=0.0 FACARB=0.0 1SCF HFCALC=1SCF SM5.42R SOLVNT=GENORG TRUES
orthosilicic acid in 1-octanol

trxCM3A.dat (x = 1 - 11):

AM1 CM3 BONDS CART 1SCF
Charge Model 3 calculation

trxCM3P.dat (x = 1 - 12):

PM3 CM3 BONDS CART 1SCF
Charge Model 3 calculation

11. REVISION HISTORY

First and second level revisions, e.g., 4.0, 4.1, and 4.5 are enhancements. Third level revisions, e.g., 4.5.1, are bug fixes. Revisions subsequent to version 3.0 are summarized below.

VERSION 3.0.1 (August 1992)

1. In the subprogram SRFCTY:

A bug has been corrected. This bug only affects SM2 and SM3 calculations in which a hydrogen atom is separated from the nearest heavy atom by approximately 1.7 Angstroms or more. This manifests itself in allowing the hydrogen atom to have a non-zero volume for the solvent accessible surface area computations which may cause it to shield some of the surface area of the nearby heavy atoms. This bug has never arisen in any of the parameterization calculations or in any of the supplied test runs, however, it is possible for this situation to arise in the modeling of a proton transfer or (de)protonation.

2. In the subprogram JINCAR:

The common block name SCRACH has been changed to SCRCHR to match similar changes made in creating version 3.0. This bug fix does not affect any of the calculations because this common block is used for scratch space in the subprograms where it occurs.

3. Common block OPTMCL:

The size of this common block has been increased by 1 in all the subprograms where this common block occurs. This change was made to prevent one of the arrays in this common block from being dimensioned zero in the event that a SIZES include file is created in which the integer NCHAIN is set equal to 1. This involved changes in main and the subprogram `ampac_nosol`.

4. Common blocks SCRAH2 and SCRCHR:

A compiler error can occur in these common blocks due to the computed dimensions of arrays coming out zero for certain values of the other dimensions which are set in PARAMETER statements. The arrays involved are ones whose sole purpose is to make the common blocks have the same length at all occurrences. The bug was fixed by increasing the size of these common blocks by 1. This involved changes in main and in the subprograms `holder`, `ampac_nosol`, and `ampac_sol`.

5. A bug in the job control file `amsol3ccl.cray` has been corrected. This bug caused the wrong SIZES include file to be used when the MAIN subprogram was compiled.

6. The `clear` command in the `amsoli` C-Shell script has been removed. This was done because command files created by `amsoli` with the `clear` command did not execute correctly on the IBM RS/6000 when the command files were run in the background with `nohup`.

7. The following files have been renamed and the appropriate changes made to all the `amsolcl.machine`, `amsol4cl.cray`, and `amsol4ccl.cray` job control files: `ampac_nosol.f` has been renamed `ampac_port.f`, `ampac_sol.f` has been renamed `ampac_s.f`, and `amsol_util.f` has been renamed `util.f`.

8. A spurious message about the shortage of scratch space for analytic CI derivatives during single-point CI calculations (i.e. no derivatives required) has been suppressed in `ampac_s.f`.

VERSION 3.0.2 (April 1993)

1. In the subprogram FLEPO:
A bug has been corrected in this subprogram. In the IF statement used to initialize the logical variable FULSCF and O should have been a zero.
2. In the subprogram POLAR:
The call to the subprogram VECPRN had one extra variable in the argument list. This bug has been corrected.
3. In the subprogram SUPDOT:
The call to the subprogram SDOT had one extra variable in the argument list. This bug has been corrected.
4. The variables LATOM and LPARAM were used before they were initialized in the subprograms DRC and SVOPTS. This bug has been corrected by including the common block REPATH in these two subprograms.
5. In the subprogram OSINV:
The variable IPRT was used before it was initialized. This bug has been corrected by including the common block OPTIMI in this subprogram.
6. In the subprogram REACT1:
The variable NORBS was used before it was initialized. This bug has been fixed by including the common block MLKSTI in this subprogram.
7. In the subprogram STAT:
The logical variable FAIL was not declared as a logical variable. This bug has been corrected.

8. In the subprogram BONDS:
The variable NATMS2, which is used to dimension an array in this subprogram, was defined as
$$\text{NATMS2} = \text{MAXPAR} * \text{MAXPAR} - \text{MAXORB} * \text{MAXORB}.$$
This variable had the potential to be negative; this variable has been redefined as
$$\text{NATMS2} = \text{MAXPAR} * \text{MAXPAR}.$$
This change does not affect the memory of the code because the array dimensioned by this variable is in the common block SCRCHR which is used for scratch space throughout the code.
9. The common block SCRCHR:
In some subprograms the arrays dimensioned in this common block have the potential to occupy more space than the maximum allowable space allocated for this common block in the main subprogram. The size of these arrays are dependent on the values chosen for MAXHEV and MAXLIT. To avoid array bounds problems for this common block, the size of this common block has been increased. This involved changes in the following files: `ampac_port.f`, `ampac_s.f`, `amsol.cray`, `main.f`, and `util.f`
10. In the subprogram DCART:
The variable CUC is used in the argument list of the calling statement for the subprogram DHC. This variable is a logical variable which was not declared as a logical in the subprogram DCART; this bug has been corrected.
11. Syntax errors in write statements in the following subprograms have been corrected: DRC, FORCE, and GETGEO.
12. In the subprograms DERIV, VECRED, and REACT1, local variables which were never used have been removed.
13. Variables which were initialized and or used as character-type variables but which were not explicitly declared as character type have

been declared as character-type variables in the following subprograms: COL, ECRIT, FINDS, LDATA, SELECT, and STAT1.

14. In the subprogram SVOPTS:

The common block GEOVAR was not byte aligned. This common block has been byte aligned.

15. The common block REPATH has been byte aligned. This involved changes in the following subprograms: DFPSAV, DRC, GEOUT, PATHS, FLEPO, ITER, READS, SVOPTS, and WRITES.

16. In the subprograms GETGEO and STAT3, EXIT statements have been converted to STOP statements.

17. In the subprogram LTRD:

The DEC Alpha workstation gave a compiler warning for the following statement

```
ELSEIF (MAXPAR .LT. 20) THEN
```

The compiler warning was that there was no path to this statement. This compiler problem was fixed by adding the following lines to this subprogram:

```
ITEST = MAXPAR
```

```
ELSEIF (ITEST .LT. 20) THEN
```

18. All SAVE statements have been moved to positions that are before the DATA statements. The previous location of the SAVE statement was flagged as non-ANSI; this change was made in all subprograms in which this reverse positioning occurred.

19. Hollerith constants used in the initialization of character variables in data statements were flagged as non-ANSI. These data statements have been modified so as not to use Hollerith constants. This involved changes in the following subprograms: COL, FINDS, LDATA, SELECT, and STAT1.

20. Non-standard intrinsic function calls such as DABS, DMAX0, etc. have been changed to the generic ANSI-standard FORTRAN 77 calls in all of the source code.

21. In the subprogram SRFCTY:

The non-standard variable names T_COS and T_SIN have been changed to T1COS and T1SIN, respectively. Also, mixed-mode arithmetic in this subprogram has been corrected.

22. The subprogram PATH has been renamed PATHAM and all calls to this subprogram have been modified accordingly. This change was made to allow AMSOL to be interfaced with the dynamics program POLYRATE.

23. The include files SIZES, SIZES3c, and PARAM have been renamed SIZES.i, SIZESc.i, and PARAM.i. The FORTRAN include statements in all the subprograms have been updated to reflect this change. This change was made so as to accommodate the magnetic tape writing utility ANSITAPE which requires all files to have extensions.

24. In the subprogram ITER:

The call to the function MECI has been changed from

```
MECI(EIGS,C,CBETA,EIGB,NORB,NMOS,NCIS,.FALSE.,.TRUE.)
```

to

```
MECI(EIGS,C,CBETA,EIGB,NORB,NMOS,NCIS,.FALSE.,LGRAD)
```

This change was made in order to avoid calculating the 2-electron integral transformations unnecessarily; these transformations are only needed if a gradient calculation is to be performed.

25. The subprogram DCOPY has been renamed SCOPY and all calls to DCOPY have been changed accordingly. This change was made because the subprograms DCOPY and SCOPY are the same, except that one is a double precision subprogram and the other is a single precision subprogram. Therefore, on the Crays the Cray library version of SCOPY can be used as the code is run in single precision (note that the Crays use 64-bit

words in single precision), and on all the other machines on which the code was tested the double precision subprogram SCOPY in the distribution package can be used. Similar changes were also made for the following pairs of subprograms:

SDOT and DDOT, SAXPY and DAXPY, and ISMAX and IDAMAX. In all cases the single precision subprogram name is maintained to allow use of the Cray libraries. These changes resulted in the elimination of the file holder.f.

VERSION 3.0.3 (May 1993)

1. The subprogram ISMAX was erroneously replaced by the subprogram IDAMAX in version 3.0.2. This error affected the subprograms DERI2 and DIAG and was introduced in version 3.0.2. This causes the program to optimize to slightly different geometries from version 3.0.1. This bug has been corrected.
2. A bug in the script for compiling the program on the DEC workstations has been corrected. This bug affected the DEC intrinsic CPU function ETIME.

VERSION 3.5 (May 1993)

1. Five subprograms, GEPMK, TES, DIVIDE, CALVER, and CALCEN, were added as an alternate method for distributing the points on the spheres used in calculating the surface areas for the coulombic and surface tension parts of the solvation free energy calculation. These subprograms are adapted from the GEPOL program (see reference on page 3-1), and the method employed is based on using a generic pentakis dodecahedron to divide each spherical surface in triangular tesseræ in order to guarantee a symmetrical distribution of points on each sphere. This method is used by AMSOL when the user gives the keyword GEPOL. Additional optional keywords associated with this method are NDIVEP=x and NDIVCD=x which are described in detail in the usage section. The point distribution algorithm in earlier versions of AMSOL is henceforth referred to as DOTS.

2. The DOTS algorithm has been modified to increase flexibility. The user now has control over the number of points distributed on the spheres used for calculating the surface areas for the coulombic and surface tension parts of the solvation free energy calculation. The additional optional keywords associated with this increased flexibility are NDOTEP=x and NDOTCD=x.

3. The subprogram SRFCTY:

This subroutine has been revamped for performance.

- a. In earlier versions, the DOTS algorithm was used to place points on the surface of each sphere each time SRFCTY was called. In version 3.5, the DOTS algorithm or the corresponding GEPOL algorithm is called only once per run and a geometry independent unit sphere is stored with the relative positions of each point on the surface.

- b. Rather than calculating the exact position in space and the distance to every attached sphere of each point on a given sphere's surface, the distance from each point to the center of attached spheres is calculated using only the unit sphere with unit vectors from the center of the sphere to the points on its surface, the radius of a given sphere and a vector representing the distance between it and all attached spheres, the angle between these two vectors, and the cosine law.

c. Using the unit vectors as defined above, the angle between the distance vector, pointing from a given sphere to an attached sphere, and the vector from the center of the given sphere to the edge of the intersection between the two spheres was calculated; this is henceforth called the angle of intersection. This angle along with those from the distance vector to each of the three Cartesian axes was used to determine if the intersecting sphere lies only in or completely engulfs any of seven defined half spheres. By eliminating these half spheres, we saved the distance calculations between the points in that half sphere and the attached sphere.

d. In the radial integration in the coulombic section of the solvation energy calculation, a given sphere expands until it engulfs the entire molecule. Since the number of points on a sphere remains constant throughout the entire radial integration, once the angle of intersection from the given sphere to any attached sphere begins to decrease, the points on the surface of the given atom that could possibly be eclipsed by the attached atom must be a subset of the previously eclipsed points. By storing this information, unnecessary distance calculations were eliminated.

4. In the subprogram BORNPL:

The section of the code that calculates the change in the hydrogen bond orders in the surface tension calculation has been moved from SRFCTY into BORNPL. This allows the calculation of the exposed surface area to be completed only when the coordinates of the spheres have actually changed rather than when only their respective charges are changed. The subroutine COORCK has been added to check whether the coordinates of the atoms have changed from the previous area calculation.

5. In the subprogram READS:

The input format has been changed so that two lines of keywords can be accepted. In order to utilize the second keyword line, an ampersand must appear in the first column of the second line. Otherwise, the data file is treated as if it contains only one keyword line. In addition, the subprogram READIN was altered so that if a restart file is created for a data file that has only one keyword line, a second blank keyword line will appear with an ampersand in the first column. The size of the keyword line was changed from 80 to 160 characters in all the subprograms.

6. The common block GEPCOM has been added to control both the GEPOL and DOTS algorithms. GEPCOM holds variables that allow one to avoid needless recalculation of the unit sphere for each atom after each optimization and SCF iteration. In addition, it holds the variables that control the number of points that appear on each sphere in the surface area determination for both the coulombic and surface tension portions of the calculation. This common block appears in the subprograms READS, BORNPL, and SRFCTY.

7. The INCLUDE file SIZES2.i has been added to the source code. This file contains the parameters MXPT, MXPTSV, and IATMSV which control the size of the arrays in the solvation portion of the code. MXPT is the maximum number of points that the user will allow to be placed on a sphere by the keywords NDOTCD=x and NDOTEP=x or NDIVCD=x and NDIVEP=x when calculating the exposed surface area of a molecule. MXPTSV and IATMSV refer to the maximum number of points on a sphere and the maximum

number of atoms in the molecule, respectively, for which the user is willing to store extra information in order to save unnecessary partial distance calculations. There is also a new common block, PICOM, which holds commonly used multiples of PI in the MAIN subprogram and stores them for later calculations.

VERSION 3.5.1 (October 1993)

1. In the subprogram SRFCTY:

The code was altered so that if a sphere X is completely buried in sphere Y, and the code is trying to determine the exposed surface area of X, the area will immediately be set equal to zero rather than trying to calculate it from the angle of intersection via the ACOS function. This prevents using the ACOS function with an argument that is greater than 1 or less than -1.

VERSION 3.5.2 (October 1993)

1. As compared to AMPAC-version 2.1, changes have been made to the atomic BLOCK DATA in order to deliver physically meaningful frequencies and centers of mass. In particular, the weighted average natural abundance atomic masses used by AMPAC-version 2.1 have been replaced with the exact masses of the most abundant isotopes. When there is not a naturally occurring isotope, the code uses the atomic mass of the longestlived isotope. The center of mass issue arises in the computation of dipole moments for charged species. AMSOL does this calculation automatically, another change from AMPAC-version 2.1. For charged species, the dipole moment depends on the origin, and AMSOL puts this origin at the center of mass.

VERSION 3.5.3 (November 1993)

1. In the subprogram SRFCTY:

In previous versions of AMSOL, the array RAL, which is used to dampen changes in the coulombic radii, was never initialized. This problem has been corrected so that all values of RAL are initially set to zero, which causes the code to begin iterating with the intrinsic coulomb radii.

VERSION 4.0 (November 1993)

1. In the subprogram SRFCTY:

The routine has been modified so that it only controls the radial integration of the electrostatic calculation and sets the radius for the surface tension section of the calculation. The portions that calculate the area using the DOTS and GEPOL algorithms have been moved to a new subroutine, AREAH. This allows for the easy incorporation of new algorithms to calculate the surface area, such as the analytical method, ASA, which has been added to this version of AMSOL.

2. The subprogram AREAL was added:

This subroutine uses an analytical method, called ASA (analytical surface area), for calculating the surface area of a molecule. As compared to previous analytical methods, this method simplifies the topological analysis, thereby cutting computation time, yet it still deals with all topological cases encountered, and it is not sensitive to how junction points are approached. The common block AREACM was added to control the user's choice for surface area and SCF convergence routines.

The keyword associated with this option is ASA. This is the default method to calculate the surface areas in version 4.5.

3. New options have been added to the SCF convergence routines. The subprograms CNVG2, DIAG2, DIAGIV2, HQRII2, ITER2, MAMULT2, and PULAY2 have been added to implement the changes in the corresponding original subprograms. Two new subprograms, UNCANO and DROT, have also been added. Together these subroutines invoke only one converger at a time where the order used is first the level shift method, then the Pulay method, and finally the CNVG method. A dynamic level shift method is now employed and the CNVG method has been revised.

The keyword associated with this option is CS2, which stands for convergence strategy 2. The default is convergence strategy 1 (CS1), which is the same method used in AMPAC-version 2.1.

4. The subroutine BORNPL2 has been added to implement a new dynamic schedule for updating the Fock matrix in an SCF calculation during a geometry optimization. The keyword CS3 invokes this new updating schedule along with the new SCF convergence routines mentioned in item 3. Once again the default convergence strategy is CS1, which was the method used in AMPAC-version 2.1.

5. A new option has been added to calculate the total volume of a molecule. It uses GEPOL to calculate the volume of the molecule in the optimized configuration. The precision of this calculation can be controlled with the keyword NDIVCD. In order to implement this option the common block VOLMCM was added, and the subroutine DIV2 was added to GEPOL portion of the code. DIV2 is a specialized version of the original DIV subprogram.

The keywords associated with this new option are VOLUME and NOVOL. The default is NOVOL, in which the volume for the molecule is not calculated.

6. The common block INPCOM was added to provide the option to automatically create a new input file with the optimized geometry from a run. The new input file will have the run's name with a .inp suffix.

The keywords associated with this option are INPUT and NOINP. (For example, if the user is running tr2.dat from the test suite with the keyword INPUT, the file tr2.inp will be created as an input file with the optimized geometry.) The default is NOINP, where no new input file is created.

7. The common block CYCLCM was added to count the number of times that the geometry is changed, the number of times that the atomic Born radii (alpha solvation parameters) are recalculated, and the number of SCF cycles completed during the geometry optimization, including the final geometry. These values are always written in the long output file.

8. The subroutine GEPINF was added to initialize arrays in GEPOL which were initialized in a block data statement.

9. The common block ARACOM was added as additional storage to reduce the amount of calculation associated with the DOTS and GEPOL surface area calculations.

10. The AMPAC-version 2.1 routine READA has been replaced. The new routine (called READIF) allows the user the added flexibility of entering numbers on the keyword line in scientific notation as well as in floating point notation (the latter being the only option available for AMPAC-version 2.1 and for earlier versions of AMSOL). The conversion from keyword character string to the numbers is completed by reading from an internal file. This method eliminates the need to "recalculate" the number one digit at a time as was required in READA. The files that had to be modified to make use of this new subroutine, along with READIF itself, have been moved into a new FORTRAN file called `ampac_if.f`.

11. All the `dattim.machine` files, which are machine-specific as indicated by the extensions `.dec`, `.ibm`, `.iris`, and `.sun`, have been changed to `datesv.machine` files and their use has changed slightly. Instead of returning the time and date at which they are called, a time and date stamp will be created at the beginning of a run, and this job-specific stamp will be inserted into the output files whenever `DATESV` is called. The common block `DATECM` was added to accommodate this change.

12. The status of the keyword `DERINU` has been changed. In previous versions of AMSOL, for a solution-phase calculation one of the following three keywords had to be explicitly stated: `DERINU`, `1SCF`, or `NOPOL`. In this version of the code, one of the following three keywords is used to control SCF cycling and geometry optimization: `OPT`, `1SCF`, or `NOPOL`. The default `OPT` is used if neither of the other two keywords are explicitly stated.

The keyword `DERINU` indicates that the gradient should be calculated as a numerical derivative. For gas-phase calculations, there is a semi-analytic method to calculate the derivatives which can be specified by the keyword `DERISA`. `DERISA` is the default for gas-phase calculations, although `DERINU` may be specified as an option. Since `DERISA` is not available for solution-phase calculations, `DERINU` is the default and in fact the only option for solution-phase calculations. For a more complete discussion of these keywords see Section 6 of this manual.

VERSION 4.0.1 (March 1994)

1. In the subprogram `SRFCTY`:

The routine has been modified to eliminate some unnecessary calculations of the exposed surface area of any atom, `i`, within a test molecule. In implementing this change, it was necessary to alter the criterion for recognizing when the entire molecule was engulfed by the expansion of a certain atom, `i`, within the molecule. In versions 4.0 and earlier, the molecule was considered completely engulfed if the exposed surface area `i` was at least $0.999 \cdot 4 \cdot \pi \cdot (r^2)$, where `r` is the radius of sphere `i`. In version 4.0.1, the rest of the molecule is considered engulfed by sphere `i` if the radius of sphere `i` is larger than the `MAX`(distance from `i` to `j` + the radius of `j`) for any sphere `j`. This new check is more accurate than the original check and thus to complete the radial quadrature it may occasionally be necessary for the new method to make one additional expansion shell (when compared to the original method) before the rest of the molecule is considered fully engulfed. This could possibly alter the results obtained for the radial quadrature slightly, but these differences are within the error of the method.

2. A bug was fixed in the subprogram SRFCTY which caused the keyword "TDIFF=" not to function properly. If this keyword were chosen in AMSOL versions 3-4.0 there would be no effect taken and the initial half shell thickness was .05 for the SM1 or SM1a solvation models and .01 for the SM2 and SM3 solvation models. In the current version of the code, the TDIFF= keyword has been dropped in favor of TONE=. By utilizing this keyword, the whole first shell thickness (not the half thickness) can be set. A more complete description of this keyword and its function can be found in the keyword section.

VERSION 4.0.2 (May 1994)

1. A bug was fixed in the subprogram GETGEO which could cause spurious results if the number of heavy (non-hydrogen) atoms used in a given run was larger than the parameter MAXHEV set in the SIZES.i include file. This parameter is used to set the dimension for arrays which store orbital information for each atom. In previous versions, if the total number of atoms in a given run was less than the sum of the MAXHEV and MAXLIT parameters, the run was completed even if the number of heavy atoms was larger than MAXHEV (or the number of light atoms was larger than MAXLIT). This led to under-dimensioned arrays and spurious results. In this and future versions of the code, the number of heavy and light atoms are counted separately and compared to the MAXHEV and MAXLIT parameters, respectively. If either of the parameters is exceeded, the code will stop and print an explanation of how to increase the parameter values in the .out file.

VERSION 4.1 (May 1994)

1. A more efficient radial integration scheme for the dielectric screening computation in the GB model has been incorporated into this version of AMSOL. In earlier versions of the AMSOL code, the force rectangle algorithm was used for the integration. This algorithm led to small, but systematic errors in the integrals which were explicitly absorbed in the parameterization. The new algorithm, which is called the force trapezoid algorithm, was incorporated into the subroutine SMX1. When the force trapezoid algorithm is chosen, this subroutine is used instead of the SRFCTY subroutine which contains the force rectangle algorithm.

For a more complete description of these two algorithms see Reference10 of Section 3 of this manual.

The keywords associated with this new method are RAD=, EXTM, and STDm. Each of these are explained in more detail in the keyword section of this manual.

2. Since the new integration algorithm does not produce the small systematic errors which were explicitly absorbed into the parameterization of the solvation models, the solvation model SM2 was reparameterized so that it produces final results similar to the original SM2 parameterization but using the new integration scheme. This new solvation parameter set is called SM2.1. By using the keyword SM2.1 the new integration scheme and new parameters are called for. This changes involved modifying the BORNPL subroutine.

3. The set of allowable keywords has been divided into two sections in AMSOL version 4.1. The first set of keywords works in the same way as in prior versions of the code. They need only to be entered on the keyword line(s). If there is an unallowed or untested combination of keywords input on the keyword line, the program will stop after reading

the line and write an explanation of the conflicting keywords to the .out file. The second set of keywords includes specialized keywords which control advanced aspects of the AMSOL code (or untested combinations of options which may be of interest to an advanced user or code developer.)

In order to use these advanced keywords or combinations of keywords, the keyword "DEV" (meaning developer) must be included on the keyword line. This prevents the basic user from using options that are designed for developers or that are inconsistent or that have not been fully tested. For a full description of these keywords and combinations see the keyword section of this manual (Section 6). To implement this change, modifications were made to the READS subroutine.

4. A new keyword, XKW, is now available. This keyword allows for the use of up to 4 additional lines of keywords (beyond the two lines already accepted in AMSOL-versions 3.5-4.0.1) By adding this keyword onto one of the two keyword lines in the standard input file (.dat) the code will look into a file specified in the amsoli.c or amsolb.c script and add the contents of the file into the keyword array.

Note: This option does not support the use of the RESTART file and keyword and its associated restart file. When restart files are created, only the original one or two lines of keywords will be placed in the restart file. This causes an error if the XKW file (specified in the aforementioned scripts) has been altered. In addition when using the

INPUT keyword, the .inp file will also only echo only the keyword lines which appeared in the original .dat file. To use this file (which contains the optimized geometry) with the same keywords as were used with the original .dat file, the XKW file which was specified for use with the .dat file must still be present when running the .inp file.

For a more complete description of this keyword and its uses, see the keyword section. In order to implement this keyword, modifications were made to the READS and READIF subroutines and the size of the KEYWRD array.

5. A new option was incorporated into version 4.1 of the AMSOL code which allows the user to set the charges for each atom to preassigned values rather than values computed from the electronic wave function. This option requires using the EXTSM and DEV keywords. These charges are then used throughout the entire calculation. A full description of this new option can be found in the keyword section. Incorporating these new keywords involved modifying the READS and WRITES subroutines as well as all portions of the code which reassign charge values immediately following a call to the CHRGE subroutine.

6. A new option was installed which allows the user to set certain solvation parameters when the EXTSM and DEV keywords are used. The user is able to control the Beta-k, Sigma(0), Sigma(1), Rho(0), Rho(1), q(0), and q(1) parameters as defined by Reference 8 of Section 3 of this manual. This new option can be used with both the force rectangle and force trapezoid integration schemes, and incorporating it involved modifying the SRFCTY and READS subroutines. A full description of this option can be found in the keyword section.

7. The scripts used to run trials on the various machines have been modified so that they explicitly request the csh environment for which they were designed.

VERSION 4.1.1 (July 1994)

1. A bug was fixed in the AREAL analytical area routine which returned the incorrect value for the buried solid angle of a sphere if the sphere was completely buried by exactly two spherical caps. This is a rare occurrence and did not effect any of the trials in this package.
2. A bug was corrected in the WRITES subroutine which printed incorrect information about the number of geometry optimizations for a given gas phase run. In addition, the partial second geometry table which was inadvertently printed was removed.

VERSION 4.5 (August 1994)

1. The keyword CM1 was added. CM1 is used in a gas-phase calculation to indicate that Charge Model 1 charges should be calculated at the end of the run and printed in the output file. The use of the CM1 keyword does not affect the calculation in any way except that the CM1 charges and dipole moment are printed in the output file along with the usual AMSOL gas-phase output. In addition, charges and dipole moments printed in the .arc and .inp files will also be CM1 charges and dipole moments. The parameter set used (either CM1A or CM1P) is determined by the use of the AM1 or PM3 Hamiltonian.

For a more complete description of this keyword and its uses, see the keyword section. In order to implement this keyword, modifications were made to the READS and WRITES subroutines.

2. The subprogram CHGMP1 was added to calculate Charge Model 1 partial charges. During the final printout, CHGMP1 is called if the keyword CM1 has been specified. The subroutines CHRGE and CHGMP1 accept and return the same parameters. However, CHRGE returns the Atom Electron Density as determined by Mulliken population analysis, and CHGMP1 returns the Atom Electron Density that has been altered by the CM1 mapping scheme.

VERSION 4.5.1 (August 1994)

1. A bug was corrected in the WRITES subprogram. This bug caused the original heat of formation to be printed in the .inp file even if a new gas-phase optimization had been run since the original heat of formation was obtained. In the current version, if the keyword HFL is used and the run creating the .inp file is an aqueous run, then the original heat of formation (obtained by a gas-phase calculation) is put on the fourth line of the .inp file. If a gas-phase calculation is run then the newly calculated heat of formation is put on the fourth line of the inp file.

VERSION 4.5.2 (January 1995)

1. A bug in the original AMPAC code caused some open shell ROHF runs to end prematurely due to a floating point error. This was fixed by ensuring that the array PNERT in subroutine DERI21 was not zero during the first iteration. This bug did not affect closed shell or UHF calculations. In addition, ROHF calculations that ran to completion were not affected.
2. A bug was corrected in the subroutine WRITES. This bug gave erroneous results only for charged molecules, and only if hyperfine coupling constants or localized molecular orbitals were calculated, or if the MULLIK option was used. This involved changing the name of the variable that holds one of the moments of inertia from C to CMOM.

VERSION 4.6 (January 1995)

1. A new solvation model parameter set SM3.1 has been developed to make use of the new integration algorithm introduced in AMSOL-version 4.1 in conjunction with the PM3 Hamiltonian. The solvation model SM3 was reparameterized so that it produces final results similar to the original SM3 parameterization but using the improved integration scheme. By using the keyword SM3.1 the new integration scheme and new parameters are called for. This change involved modifying the BORNPL subroutine.
2. A method was implemented for obtaining restart files when an AMSOL run crashed, e.g., due to an external time limit. Now, if an AMSOL run is initiated with the file [filename].dat, the directory [filename].dir is created. All fort files are temporarily stored in this directory and, by default, restart fort files are created or overwritten every 1800 seconds. (The amount of time spent prior to creating the restart files can be controlled with the keyword TDUMP.) If the run terminates normally or is stopped by the internal clock of the code, the fort files are moved to the correct [filename].[extension] name (described in Section 6) and the directory [filename].dir is removed. Further discussion of this option can be found in Section 6 under the RESTART keyword. This modification caused changes to the FLEPO and DFPSAV routines.
3. A new option has been created which allows the user to combine a gas-phase calculation with a solvation calculation to obtain a free energies of solvation in one run where the gas-phase heat of formation is unknown beforehand. This is accomplished by using the keyword TRUES in a solvated run and placing the word CALC on the special TRUES line. (See the keyword TRUES in Section 6 for further details.)
4. In this version of AMSOL, we have corrected the MULLIK and GRAPH options, which involved several changes in the subroutine MULLIK. We note that the error in MULLIK was not an AMSOL bug, but rather a feature that did not work correctly in any version of AMPAC or MOPAC; those packages have very similar incorrect code. This option is described in Section 4.
5. The default values of MAXHEV and MAXLIT have been altered. The new values for AMSOL 4.6 are 40 for MAXHEV and 60 for MAXLIT. In the Cray-specific version (AMSOL 4.6c) the values for MAXHEV and MAXLIT are 60 and 90 respectively.
6. The code was altered so that the user is unable to mix a Hamiltonian with an incompatible solvation model. If a combination such as AM1 and SM3 are used together, the code will stop and print out a warning that these options are incompatible.
7. A new INCLUDE file, PARAMS.i, was created to hold all the SMx parameters along with the parameters necessary for the CM1 charge models.
8. The amsolcl.ibm script was changed to remove the preprocessor flag from the compiling options. The FORTRAN compiler packaged with some IBM RS6000 machines does not include a preprocessor. For users who have the preprocessor and wish to use it, instructions are included within the

amsolcl.ibm script on how the script can be modified to make use of the option.

VERSION 4.6.1 (January 1995)

1. In the subroutine AABABC, the variable KEYWRD was dimensioned improperly as an array of length 160. This has been fixed to the proper size of 480.
2. When the EXTSM option is used, the user-supplied EXTSM file is echoed to the .out file. An incorrect printout has been fixed. The first column of the EXTSM file should contain the value of R(k), the atomic radius, not the value of Beta-k, the sum of the atomic radius and solvent radius, as indicated in previous versions. This does not affect the actual performance of the code.
3. In the subroutine TES, the common block POLI was incomplete. This has been fixed so that it has the same length in all occurrences.
4. Subroutine DFPSAV was changed so that the keyword RESTART was only added once to the keyword line during a calculation.
5. AMSOL now checks, for atom numbers 4 and greater in the Z matrix, that there are no connectivity numbers that are zero if the input geometry is in internal coordinates. If any zeroes are found, the calculation ends with a warning message.
6. Several non-FORTRAN77-standard conventions were removed from the code.
 - a. All double quotes not used as text inside of a write or print statement were removed.
 - b. All tab characters were removed from the code.
 - c. Two "END DO" statements were removed from gepol_mod.f and replaced with FORTRAN77-standard "DO x" statements.
 - d. The FORTRAN code in util.f was converted from non-standard lowercase to uppercase.
 - e. Variables or subroutine names longer than 6 characters were shortened to 6 characters. The variable HRPCAL2 was changed to HRPCL2. Subroutine names changed were BORNPL2 to BRNPL2, DIAGIV2 to DIAGV2, MAMULT2 to MAMLT2, and READCOOR to RDCOOR.
7. The keyword NDIVCD was echoed to the .out file with an incorrect format. This has been fixed. The fix involved a change to the WRITES subroutine.
8. The subroutine DFPSAV was altered so that the keyword RESTART is echoed to the keyword line only once during long runs
9. Previously, if either of the keywords DENMAT or OLDMAT were used, it was not echoed properly to the .out file. This has been fixed and involved a change to subroutine READS.
10. The subroutine WRITES was altered so that the correct molecular formula was written to the .arc file. Previously -beginning in version 4.0- use of the keyword INPUT would cause an incorrect formula to be printed in some instances.
11. Because using the CALC option of the TRUES keyword necessarily creates a .inp file, the NOINP keyword has been made incompatible with this option.
12. Subroutine WRITES was altered so that the gas phase heat of formation echoed to a .inp file for a calculation using the TRUES keyword was printed on the correct line. Previously, calculations that ended due to an internal or external time limit had incorrectly placed gas phase heats of formation.
13. An error in the "set" commands in the scripts amsolab.c and amsolai.c which caused them not to run on some machines was corrected.

14. The datesv.sun file was corrected so that the date and time is properly echoed to the job.out and job.arc files.

VERSION 5.0 (April 1995)

1. SM4 solvation models for alkane solvents were added, and external parameter files for two SM4-SRP models for water are provided. The SM4 alkane models can be specified using the SM4 keyword and either AM1 or PM3. In addition, the particular alkane solvent is indicated using the SOLVNT keyword. The addition of SM4 models involved the following changes:

a. A new subprogram, CM1DRV was added to amsol_new.f. This calculates the derivative of the polarization energy with respect to the density matrix when using the CM1 models to compute atomic partial charges.

b. Modifications were made to subprograms ITER, ITER2, BORNPL, BORNPL2, SRFCTY, SMX1, WRITES, and READS to accommodate the new models.

c. The format of the EXTSM file is changed when using the SM4 models because of the new form of the parameters involved. See the description of the EXTSM keyword for more information. Changes were required in SMX1 to accomplish this.

d. The common blocks FOKMAT, TRADCM, SURF and BORN were made longer and the common blocks HBONDA, HEXTSM, and CM1SUM were created to accommodate variables required for SM4.

e. A new subprogram, READCH, was added to amsol_new.f. This reads and returns a character string from the keyword line. It is used to read in the solvent name specified using the SOLVNT keyword.

f. The keyword MSURFT was added for the input of the solvent macroscopic surface tension.

g. The keyword AREAS was added to print out the solvent-accessible surface areas in a way that is convenient for SM4 calculations.

2. The eigenvector following (EF) and Broyden-Fletcher-Goldfarb-Shanno (BFGS) methods for geometry optimization have been added, and the EF method has been made the default. Three new keywords were added to invoke these options, EFOLLOW and TSTATE to invoke EF and BFGS to invoke BFGS.

Inclusion of the EF method also involved a number of other alterations of the code:

a. The files ef_mod.f, ef_port.f, and ef_lib.f have been added. They contain the new subroutines required for the eigenvector following algorithm that has been added in version 5.0 of AMSOL. The compilation scripts have been changed accordingly.

b. Several specialized keywords applicable only to EF have been added. For more information on these keywords, see Section 6.3.

c. Subprogram WRTKEY was modified.

d. The print out lines indicating the success or failure of the geometry optimization were updated in subroutine WRITES.

e. The subprogram RESINP was created to write the .inp file that can be used to restart a stopped job.

f. The EF optimizer is now the default optimizer for the PATHS option from AMPAC-2.1. The BFGS or DFP optimizers can also be specified for this option.

g. The EF optimizer is now the default optimizer for the FORCE option from AMPAC-2.1. The BFGS or DFP optimizer can also be specified for this option.

Inclusion of BFGS involved alterations only in subprogram FLEPO.

3. The convergence criteria used for the DFP geometry optimizer in versions of AMSOL prior to 5.0 (in which DFP was the default) have been

removed. The EF, BFGS, and DFP optimizers now use a single, new convergence criterion. This involved changes to the FLEPO, EF, and EFSTR subroutines. In addition, a one-line summary of the previous optimization cycle is now included for all three optimizers, and messages are printed out when significant events, such as Hessian recalculations and restart file updates, occur.

4. The keyword PRECISE is no longer supported for calculations using the EF, BFGS, and DFP optimizers. Three characteristics controlled by PRECISE are now controlled by individual keywords. More information is given in the keyword section.

5. In previous versions of AMSOL, calculations frequently ended prematurely due to LINE MINIMIZATION errors. These could be avoided by restarting the calculation using a perturbed geometry. The KICK keyword has been added to automatically automate this process when using the BFGS and DFP optimizers so that the calculation will terminate normally more often. The KICK option can also help avoid Trust Radius errors in the EF method implemented in this version.

6. The restart files are now printed out more often. In addition to updating the restart files every x seconds (where x is user controllable), the files are updated each time the Hessian is recalculated and before ending the optimization with an error.

7. Lithium parameters were added for PM3 gas-phase calculations. This involved an additional parameter block in the file PARAM.i and a change to the routine READS.

8. This version of AMSOL has been made more internally consistent so it is easier to use as a subprogram of a larger program such as a genetic algorithm search or a dynamics program. The changes required for this are:

a. In previous versions of AMSOL, there were three different methods to determine if a call to a subroutine was the first call to it. This was done so that some sections of code would only be executed on the first call to the subroutine. This has now been standardized so that only one method is used. To accomplish this, many subroutines were modified, but their behavior is not altered.

The common block ONESCM was added to the code.

b. The common blocks AM1PCM and IDAACM were added to store the AM1 parameters.

c. Places in the code that should be modified if AMSOL is being used as a subroutine but not as a stand alone program are commented with the phrase 'GA-FRIENDLY'.

9. The dielectric constant, the small and large solvent radii, and the CS surface tension are now included in DATA statements in the file PARAMS.i. These solvent properties were formerly hard-wired in the code but are now variables. The change was made because water is no longer the only solvent treated by AMSOL and to allow these properties to be changed by the user using keywords and the EXTSM file. This involved the addition of the SOLVCM common block.

10. The keyword TRUES has been made compatible with the keyword 1SCF. This involved a change to the subroutine READS.

11. The keyword NOCOGS was added to allow the user to run a non-standard calculation by turning off the cutoff gaussian (COG) functions in the generalized Born terms for O-O and N-H electrostatic interactions.

12. Lines from the AMPAC-version 2.1 code that checked for the string 'NOIN' to test for the presence of the keyword NOINTER have been

modified to search for the entire keyword. This was done to prevent conflict with the keyword NOINPUT.

13. The subprogram SELECT was renamed AMSEL and the subprogram STAT was renamed AMSTAT to avoid conflicts with the libc C math libraries of the same names.

14. The capability to place comment lines in the EXTSM file was added.

15. The messages printed to the .out file when the THERMO keyword is specified have been improved.

16. The trial runs have been rearranged. For each trial, there is a tr#.dat which is a gas-phase run, a tr#a.dat which is an aqueous-phase run, and a tr#n.dat which is a non-aqueous-phase solvation run. For some trials, there is more than one aqueous run to demonstrate additional features. These runs are named tr#a.dat, tr#a2.dat, tr#a3.dat, etc. Also, the order of the runs has been changed so that neutral molecules come first and ions come last. There are no tr#n.dat runs for ions or for molecules containing phosphorous because these were not included in the parameterization for the nonaqueous solvents.

VERSION 5.0.1 (September 1995)

1. A bug which caused the SM4 models and the CS3 option to be incompatible has been fixed. In version 5.0, derivatives of the CM1 charges used in the SM4 models were not always recalculated when needed. This resulted in an unoptimizable run or an error in the printed results. Changes necessary to correct the problem were made in the BRNPL2 subroutine.

2. A bug which caused run-on optimizations when very abrupt changes in the gradient were experienced has been fixed. This situation is very rarely experienced when one is using the predefined solvation models provided with the AMSOL package, but it occurs more readily when using the EXTSM option. The changes necessary to correct the problem were made in the FLEPO subroutine.

3. A bug which caused the GCOMP keyword to be ineffective when coupled with either the BFGS or DFP optimizer has been fixed. This prevented the user from controlling the convergence criteria. The default criteria (GCOMP=0.45) was always used. The changes necessary to correct this problem were made in the FLEPO subroutine.

4. The following changes were made to the output:

a. AMSOL now prints out that the solvent is water when using the H2OSRP models.

b. Two spelling errors in the output were corrected. EXTM was changed to EXTCM in the partial charge/dipole printout, and consistent was changed to consistent in one location.

c. The warning message printed when using the EXTCM model was changed to reflect the inclusion of the SM4 model and future solvation models.

d. A specific solvent name is no longer specified if the keyword DIELEC is used for calculations other than an H2OSRP calculation.

e. If EXTCM charges are chosen, they are printed in the final solvation summary in both the .arc and .out files. Previously, CM1 or Mulliken charges were printed in this spot. The dipole moment calculated from EXTCM partial charges is printed in the .arc file if EXTCM was used. Note that the Mulliken charges (and CM1 charges if used) are still printed elsewhere in the .out file when EXTCM charges are specified.

f. The order in which the dipole moments are printed has been changed so that the dipole moment from EXTCM charges prints near the

EXTCM charges. In addition, the Hybrid and Sum dipole moments that were erroneously printed with the EXTCM dipole moment have been removed. When EXTCM charges are specified, the Hybrid and Sum dipole moments as calculated with the Mulliken charges are still printed for informational purposes, but are not used.

g. Extraneous calls to the subroutine AXIS have been removed from WRITES.

h. If AMSOL terminates prematurely due to an error in reading EXTCM charges, a warning message is printed to standard output.

i. Erroneous print statements indicating that CMI or Mulliken charges would be used when EXTCM charges were specified were removed from subprogram WRITES.

5. Certain machines have difficulty resolving subroutine calls in which one of the passed variables itself involves a call to an intrinsic subroutine. This was first noted with one of the Silicon Graphics machines. To avoid this problem, all such embedded function calls were removed. This involved altering lines in many of the subroutines.

6. A bug which caused output errors if the keywords EXTSM and EXTCM were used together has been fixed. The keywords may be used together but caution should be exercised because any surface tensions placed in the EXTSM file will be used rather than setting all surface tensions to zero as is done in a standard EXTCM run.

7. A bug which caused erroneous output when following a reaction path was fixed. The changes necessary to correct this problem were made in the PATHAM subroutine.

8. If the EXTCM keyword is used, surface tensions consistent with the chosen solvation model are now used. In previous versions, the surface tensions were always set to zero because the surface tensions consistent with the solvation model chosen were not derived for the charges used in the EXTCM file. Caution should be used to avoid misinterpretation.

VERSION 5.2 (September 1995)

1. An algorithm for incorporating a pairwise descreening approximation into the calculation of the electrostatic component of the polarization free energy of solvation within the generalized Born approximation was implemented in the AMSOL code. This algorithm replaces the need for a numerical radial integration about each of the atoms in the solute, hence the use of this algorithm greatly simplifies the complexity of the calculation and reduces the amount of computing resources used for a given calculation, especially for larger solutes. The current implementation of the pairwise descreening approximation has been parameterized only for SM2.1-type calculations where the solute molecule is composed only of H, C, O, and N atoms. The keyword associated with this option is SM2.2.

VERSION 5.2.1 (October 1995)

1. A bug which occurred in the calculation of the CDS terms of the free energy of solvation when the keyword EXTCM is used was corrected. The changes necessary to fix this problem were made in the BORNPL and BRNPL2 subroutines. This bug originated in AMSOL-version 5.2

2. A bug in the printing of the total solvation energy in the atom-by-atom summary table was corrected. The change necessary to fix this problem was made in the WRITES subroutine. This bug originated in AMSOL-version 5.2.

VERSION 5.2.2 (November 1995)

1. In previous versions of the code, if the cycle at which the code required restart information to be stored coincided with the maximum number of cycles set by the CYCLES=x keyword (or after 100 cycles if the keyword CYCLES=x was not used) the code would not stop until another termination point was reached. This bug was corrected in subroutine FLEPO.
2. In subroutine FLEPO, the time printed in the cycle summary section was changed from the time since the restart information was last stored to the total time of the calculation.

VERSION 5.3 (November 1995)

1. The convergence criterion for geometry optimizations was enhanced. The test which is now used for all geometry optimizations requires that the largest component of the gradient is less than a user-controllable parameter called GCOMP, and that the energy calculated at the final geometry dropped by less than 0.1 kcal/mol from the previous geometry.

These changes were incorporated into the new subroutine CONVCK. For more information see the explanations of convergence strategies in Section 4.2.4.1.

2. The use of the TRUES keyword has been enhanced and the new keyword HF=x has been added to replace the use of a extra line above the Z matrix to store the gas-phase heat of formation. The primary changes necessary for this change were incorporated into the new subroutine HFADD. For more information about this option and the current usage see the description of the keyword TRUES in the keyword section.

VERSION 5.3.1 (December 1995)

1. A bug was fixed which caused some of the geometry optimizers to revert to the convergence criterion of versions 5 to 5.2.2 of AMSOL if every gradient component for the initial geometry were initially below the parameter GCOMP.

VERSION 5.4 (December 1995)

1. The XYZ keyword was clarified in the manual and appropriate changes were made to the keyword explanation output in the code. See new description of the keyword XYZ in the keyword section.
2. The use of the keywords TRUES and HF=CALC with 1SCF is no longer allowed. This prevents the user from unknowingly using the energy calculated at an unoptimized geometry in the gas phase to calculate the true solvation free energy. The capability to calculate a constrained-geometry true solvation energy still exists, but such a calculation must be done in two runs. We believe this will avoid a likely source of confusion in .dat files.

VERSION 5.4.1 (April 1996)

1. A bug was fixed which prevented the use of the EXTCM or EXTM options when using Cartesian coordinates.
2. A bug was fixed in the optimizer tolerances which would have suggested that a geometry was optimized if the absolute value of the largest gradient component is less than 0.45 but the calculated energy increased from the previous step. The convergence strategy has been modified so that if the largest component of the gradient is less than 0.45 and the calculated energy has changed by less than 0.1 kcal/mol from the previously calculated energy then the geometry is considered converged.

3. A bug was fixed in subroutine DERIV which caused the gradient to be reported as 1/2 its actual magnitude. This bug affected only gas-phase calculations.
4. A bug in the subroutine FMAT was fixed which caused FORCE calculations to fail.
5. A bug in CS3 was fixed which caused the CDS term to be updated too infrequently during the SCF cycle.

VERSION 5.4.2 (April 1997)

1. A bug was fixed where the SCF energy in each iteration was calculated by adding together two numbers with different units in subprogram ITER. This potentially slowed the SCF convergence, although all converged results and all printed results were correct.
2. The calls to subroutine INVRT2 were corrected to contain the correct number of calling parameters. This bug affected only CS2 and CS3 (non-default) runs.
3. One call to subroutine DSCAL was corrected for containing an incorrect number of calling parameters. This bug affected only trials using the TRUSTE optimized that failed in the first iteration.

VERSION 5.9 (April 1997)

1. The SM5.4/AM1, SM5.4/PM3, SM5.4/U, SM5.4PD/AM1, SM5.4PD/PM3, SM5.4PD/U, SM5.2PD/AM1, SM5.2PD/PM3, and SM5.2PD/U models were added. The SM5.4/AM1 and SM5.4/PM3 models are parameterized for all organic solvents and water, while the others are parameterized only for water at this time.
2. The TRUSTE and TRUSTG optimizers were added.
3. The keyword verification system in AMSOL was revised so that any unrecognized keyword now causes the code to stop with an appropriate error message in the .out file. Note that these checks include completely unrecognizable keywords in addition to standard keywords with unrecognizable prefixes or suffixes.
4. In contrast to previous versions, the current version of AMSOL will now stop if the user attempts to use a solvation model with an atom type for which the model was not parameterized. (Unless the EXTSM keyword is used.)
5. The use of the HF=CALC keyword has been changed, and the current convention is incompatible with all previous versions of the code. For a description of the new form of the keywords for calculating a heat of formation in the gas-phase, then continuing on to a calculation in solution, see the TRUES keyword section.
6. In order to input Cartesian coordinates, the user must now use the keyword CART. Note that the use of the keyword CART is independent of the choice to have the calculations completed in internal coordinates or in Cartesians. (See keyword XYZ.)
7. The structure of the code was changed. Except for the unmod directory, individual subroutines are contained in their own files. The amsol_cl.x scripts have been replaced by a single amsol.compile script.
8. Changes made by Frank Jensen to the EF optimizers and manual sections have been added.

VERSION 6.0 (May 1997)

1. The SM5.0R and SM5.05R models were added to AMSOL.
2. The CS2 and CS3 SCF convergence strategies were removed from the code.

VERSION 6.1 (May 1997)

1. Analytic gradients and derivatives were added for all SM5.x models. The keyword to use the analytical derivatives is DERISA. (Note: This is now the default for all SM5.x solvation models.)

VERSION 6.1.1 (August 1997)

1. A bug was fixed in subroutine bornpl.f This bug caused the keyword EXTCM to have no effect except that the input charges were echoed in some of the output.
2. A bug was fixed in the subroutine keyflg.f. This bug had the potential to cause SM5.05R runs to fail on some platforms with an input format error message.

VERSION 6.1.2 (February 1998)

1. A bug was fixed which caused runs which used the TRUES keyword along with the CART keyword to fail.

VERSION 6.5 (February 1998)

1. The SM5.0R model for organic solvents was added to the code.
2. The SM5.2R model for use with the MNDO, AM1, and PM3 Hamiltonians for water and organic solvents were added to the code.
3. The CM2 charge model was added to the code.
4. The SM5.42R model was added for use with the AM1 and PM3 Hamiltonians for water and organic solvents.
5. The amsol.compile script was modified to include options for compiling the code for Linux. The order of some statements (DATA statements and declarations of arrays) was changed to prevent compiler errors under Linux.

VERSION 6.5.1 (February 1998)

1. In file ebrglq.f, the declaration of arrays was placed before the DATA statement.

VERSION 6.5.2 (June 1998)

1. The 1SCF keyword was removed from test input files tr28n.dat, tr29a.dat, and tr38a.dat. In files tr31a.dat and tr31n.dat, SM5.42 was replaced by the correct keyword SM5.42R. The keyword AM1 was added to files tr32a.dat and tr33a.dat.
2. The solv.txt file has been modified to give correct FC and FH values for carbon tetrachloride, dibromomethane, and iodobenzene.

VERSION 6.5.3 (September 1998)

1. The manual was updated to reflect the fact that the keyword T=x has been replaced by TLIMIT=x.
2. In file wrtkey.f, two write statements were added to echo keywords FACARB and FEHALO when used as input.
3. Subroutine XYZINT was modified to fix a problem of the conversion from Cartesian coordinates to internal coordinates when three or more atoms are colinear. The modified routine was moved from ampac_unmod.f to a new file xyzint.f under the directory "new".
4. On line 228 in file sm5rhh.f, "RTKK=" was changed to "RTKK=RTKK+". This change is necessary to make it agree with the published surface tension functional form for N triple bonded to C and avoid potential problems caused by the old code. Fortunately, the change does not affect AMSOL results for solutes in the parameterization sets. Those results were generated correctly by the old code.

VERSION 6.5.4 (November 1999)

1. In the file sm5rhb.f, two lines were changed to correct the output from the AREAS keyword for the H N(2) and H O(2) surface tensions of solvation models 5.2R and 5.42R. Lines 101 and 118 were changed from "RTKK" to "RTKK*RTKK2". This change does not affect the output of G-CDS or DeltaG-S(sol) in the breakdown of the solvation free energy.
2. The subroutine ECHOWD was modified to remove CART from the keyword line in the restart file. Several lines in the file efstr.f were added to correctly assign elements in the vector XPARAM, which allows optimization in solution when the input is in Cartesian coordinates.

VERSION 6.6 (November 1999)

1. The output for the AREAS keyword has been reformatted and now also includes the dependence on the solvent descriptors.
2. The by-atom breakdown in the output has been changed to include the atomic surface tension.
3. The code has been changed so that it is now required to use the keyword lSCF must be used in conjunction with the SM5.2R and SM5.42R models. To do a numerical optimization in solution with the SM5.2R and SM5.42R parameters, the SM5.2 or SM5.42 keywords should be used along with the keyword DERINU. The SM5.2 and SM5.42 keywords were added to allow users to do these optimizations.
4. The geometry is not printed out if the geometry is not changed. The keyword PRINTGEOM was added to print all of the geometries to allow use with previously written scripts, programs, etc.
5. The Coulomb integrals, effective Born radii and distances, and Born polarization free energy are no longer included in the output by default. The keywords PRINTCOUL, PRINTRAD, and PRINTPOL can be used to print these results.
6. Test runs: tr40.dat (piperazine in 1-octanol) and tr41 and tr41z (methanol in water) were added.

VERSION 6.6.1 (August 2001)

1. Commas were added in diprint.f and satbat.f to correct non-standard FORTRAN 77 FORMAT statements.
2. The amsol.compile script was modified to reduce the optimization on three routines; dareal.f, fock.f and fock2d.f. In particular we reduced it from 3 to 0 when using the sgi option in the compile script. The higher level of optimization caused errors when using MIPSpro Compiler version 7.3 and greater. Version numbers of the MIPSpro compiler can be determined by using the f77 -version command.
3. The am1.f file was changed to allow external parameters to be used instead of the standard parameters.
4. The sulfur reference for AM1 was updated.
5. The amsoli script was modified to correctly use .extsm files.

VERSION 6.7 (September 2001)

1. The Compaq option was added to the amsol.compile script. Details on the operating system, compiler version, and compiler options used are given in section 15.
2. The file sm5rhb.f and prareas.f were changed to include the area COT functions and the parameters for Si and O-Si.
3. Silicon parameters were added for the SM5.42R and SM5.42 models with the AM1 and PM3 parameterizations.
4. Test runs tr42a (orthosilicic acid in water) and tr42n (orthosilicic acid in 1-octanol) were added.

VERSION 6.7.1 (October 2001)

1. The parameter list for the call to SUBROUTINE VECPRN in file polar.f was modified. In versions previous to 6.7.1, this parameter list contained a constant, which is potentially modified in SUBROUTINE VECPRN. When it is modified, a segmentation fault occurs on the Compaq version of AMSOL. In addition, changing constants that are passed from one routine to another is not a FORTRAN standard. Therefore a dummy variable is now assigned to the constant and passed into VECPRN. The parameter list for the call to SUBROUTINE MATOUT in file polar.f also contained three constants which are potentially modified in MATOUT. This has also been changed so that dummy variables are assigned to these constants and passed into the routine.
2. The files, porcpu.compaq and datesv.compaq were added to the /mach directory. These files contain machine date and time routines specific to Compaq computers. (These routines and their syntax are identical to the routines used by the DEC ALPHA version of AMSOL). These file are necessary in order for the amsol.compile script to work properly.
3. Several additions were made to the amsol.compile script so that this script would support the Compaq option that was added in version 6.7. Also, a compile flag used to compile AMSOL on the Origin 2000

(R12000 processor) and Origin 38000 (R14000 processor) has been removed because it caused several unknown errors in at least three test suite calculations, namely, tr03a, tr03n, and tr14a. The compile flag is the -OPT flag, and it performs several optimizations to the code.

VERSION 6.7.2 (January 2002)

1. The correct values of the SM5.42 surface tension parameters were added. For more details, refer to "Parameterization of a Universal Solvation Model for Molecules Containing Silicon", Winget, P.; Thompson, J. D.; Cramer, C. J.; Truhlar, D. G. J. Phys. Chem. A 2002, 106, 5160.

VERSION 6.8 (September 2002)

1. The CM3 method for AM1 and PM3 was implemented in this version of AMSOL. The corresponding keyword, CM3 was also added. The include files and source code files that were modified were KEYS.i, kfinite.f, reads.f, findky.f, wrtkey.f, diprnt.f, linsum.f, and moldat.f. In addition, a new subroutine, called CHGMP3, which is located in chgmp3.f was created. This routine maps Mulliken charges to CM3 charges.

2. The linux option in the amsol.compile script was updated in order for the linux version of AMSOL to run correctly. In particular, the '-finit-local-zero' and '-fno-automatic' flags were added to the compile command. These options initialize all local variables in all subprograms of the main program. If these options are not used, then some of the test calculations get caught in an infinite loop (in routine EBRREC) or do not give correct results.

VERSION 6.8.1 (March 2003)

1. This version of AMSOL corrects a bug associated with the SM5.2 keyword. Versions of AMSOL previous to 6.8.1 did not recognize the SM5.2 keyword as a valid keyword. This has been corrected with a few minor modifications in the file reads.f. A new test calculation, which tests the SM5.2 keyword, was added to the test suite. The new test calculation is called tr43.dat.

2. The mechanism by which AMSOL determines whether or not the input options are mutually compatible has been found to fail in certain instances, so this mechanism has been simplified in this version of AMSOL. A new routine, called STPJOB, located in the file stpjob.f, was created.

3. The Gauss-Legendre quadrature, which is used to calculate the effective Born radii, was not enabled for calculations using the SM5.2R, SM5.2, SM5.4, SM5.42R, and SM5.42 models. This has been corrected in this version of AMSOL.

4. Minor modifications were made to the amsol.compile script in order to make the 'sun' option (the option used to compile AMSOL on Sun workstations) work properly.

VERSION 6.9 (March 2003)

1. The algorithm that calculates the solvent-accessible surface area (SASA) has been modified to minimize discontinuities in the gradient of

the G-CDS contribution to the free energy of solvation that have been observed in several cases. See Section 7.2 of this manual for more details. Appropriate modifications have been made in subroutine DAREAL.

2. Analytical gradients for the SM5.2 and SM5.42 methods were implemented. Subroutine SM5RHB, located in the file sm5rhb.f was modified accordingly. In addition, subroutine kwnono.f was modified so that keywords SM5.x (where x=2 or x=42) and DERISA (the keyword to request analytical gradients) are allowed in the same input. Starting with this version of AMSOL, analytical gradients are used by default for calculations using the SM5.2 and SM5.42 models. A test calculation that tests the compatibility between the SM5.2 and DERISA keywords has been added (tr18d.dat) to the test suite. A test calculation that tests the compatibility between the SM5.42 and DERISA keywords has also been added (tr41d.dat) to the test suite.

3. The '-O' optimization flag was added to the compile command for the linux option in the amsol.compile script.

VERSION 7.0 (October 2003)

1. The SM5.0R surface tension parameters now include phosphorus for non-aqueous solvents; previously only parameters for aqueous solution were included. A new test run (tr44.dat) has been included which tests this option.
2. Several changes were made to the routines that read input and write output to make them more portable.
3. The new /PC directory contains files to compile and test the code on Windows operating systems.
4. The AMSOL executables for Mac OSX and Windows98/NT/2000/XP are now distributed.
5. The amsol.compile script now has the option to automatically configure and compile the source code.
6. A script (run_all.pl) that runs all the test runs has been put in the /test directory.

VERSION 7.1 (December 2004)

1. The TRUSTE optimizer has been updated with methods proposed by R. Lindh, Chem. Phys. Lett. 241, (1995) 423.
2. The default optimizer has been changed from EF with a diagonal Hessian to EF with LINDH Hessian.
3. SCFCRT can now be set to values smaller than 1.0D-11 in the input file.
4. The SCF convergence tolerance has been adjusted in test runs 16n and 43 to give more consistent results across platforms.

12. INPUT INCOMPATIBILITIES WITH PREVIOUS VERSIONS

In general, in modifying AMSOL, it has been attempted to maintain backward input compatibility with previous versions of AMSOL and with AMPAC-version 2.1. Sometimes, however, other issues override this consideration. What follows below is a summary of known incompatibilities, arranged by the version in which they were introduced.

VERSION 5.3

1. The Heat of Formation Line is no longer supported. This line, placed previously as the line above the geometry, has been replaced by the keywords HF=OPT, HF=1SCF, and HF=CALC.

VERSION 5.9

1. The keywords HF=OPT, HF=1SCF, and HF=CALC have been replaced. See information under the keywords HF1SCF, HFOPT, and HFCALC.

VERSION 6.0

1. The code no longer recognizes Cartesian input from the lack of connectivity data. The CART keyword is required to use Cartesian geometry specification.

2. The keywords GEPOL, NDOTEP, NDOTCD, and VOLUME which used the GEPOL code for calculating solvent accessible surface areas or approximate volumes for solute molecules have been removed.

3. The keywords CS2 and CS3 for non-default SCF optimization strategies have been removed.

VERSION 6.6

1. Printing of the Coulomb integrals, effective Born radii, Born polarization energies, and unchanged geometries (when liquid-phase calculations are performed without re-optimizing the geometry) are no longer default options. The keywords PRINTCOUL, PRINTRAD, PRINTPOL, and PRINTGEOM must be used to accomplish these actions.

13. DISTRIBUTED VERSION SUMMARY

This section lists all versions that were distributed outside the AMSOL development group. Versions A-L were distributed from the University of Minnesota, and versions B-L were also distributed by QCPE. Version M is distributed by Oxford Molecular Ltd.

- (A) "AMSOL (version 1.0)," C. J. Cramer and D. G. Truhlar, QCPE program 606-original version. This program was submitted to QCPE in July 1991 and was announced in QCPE Bulletin 11, 57-58 (1991).
- (B) "AMSOL-versions 3.0 and 3.0c," C. J. Cramer, G. C. Lynch, and D. G. Truhlar, QCPE program 606-version 3.0. This program was submitted to QCPE in June 1992 and was announced in QCPE Bulletin 12, 62-63 (1992).
- (C) "AMSOL-versions 3.0.1 and 3.0.1c," C. J. Cramer, G. C. Lynch, and D. G. Truhlar, QCPE program 606-version 3.0.1. This program was submitted to QCPE in September 1992 and was announced in QCPE Bulletin 12, 77 (1992), but the announcement had an important printer's error. A corrected version of the announcement was published in QCPE Bulletin 13, 9-10 (1993).
- (D) "AMSOL-versions 3.0.2 and 3.0.2c," C. J. Cramer, G. C. Lynch, and D. G. Truhlar, QCPE program 606-version 3.0.2. This program was submitted to QCPE in April 1993.
- (E) "AMSOL-version 3.5," C. J. Cramer, G. C. Lynch, G. D. Hawkins, D. G. Truhlar, and D. A. Liotard, QCPE program 606-version 3.5. This version was submitted to QCPE in June 1993 and was announced in QCPE Bulletin 13, 55 (1993).
- (F) "AMSOL-version 4.0," C. J. Cramer, G. C. Lynch, G. D. Hawkins, D. G. Truhlar, and D. A. Liotard, QCPE program 606-version 4.0. This version was submitted to QCPE in November 1993 and was announced in QCPE Bulletin 13, 78 (1993).
- (G) "AMSOL-version 4.1," C. J. Cramer, G. D. Hawkins, G. C. Lynch, D. G. Truhlar, and D. A. Liotard, QCPE program 606-version 4.1. This version was submitted to QCPE in June 1994.
- (H) "AMSOL-version 4.5," C. J. Cramer, G. D. Hawkins, G. C. Lynch, D. J. Giesen, D. G. Truhlar, and D. A. Liotard, QCPE program 606-version 4.5. This version was submitted to QCPE in August 1994 and was announced in QCPE Bulletin 14, 55-57(1994).
- (I) "AMSOL-version 4.6," C. J. Cramer, G. D. Hawkins, G. C. Lynch, D. J. Giesen, D. G. Truhlar, and D. A. Liotard, QCPE program 606-version 4.6. This version was submitted to QCPE in August 1994.
- (J) "AMSOL-version 5.0," C. J. Cramer, G. D. Hawkins, G. C. Lynch, D. J. Giesen, I. Rossi, J. W. Storer, D. G. Truhlar, and D. A. Liotard, AMSOL-version 5.0. This version was submitted to QCPE in May 1995 and was announced in QCPE Bulletin 15,41-43 (1995).

(K) "AMSOL-version 5.2.1," G. D. Hawkins, G. C. Lynch, D. J. Giesen, I. Rossi, J. W. Storer, D. A. Liotard, C. J. Cramer, and D. G. Truhlar. This version was submitted to QCPE in October 1995 and was announced in QCPE Bulletin 15, 77-79 (1995).

Version 5.3.1 was submitted to QCPE in December, 1995 to replace version 5.2.1.

(L) "AMSOL-version 5.4," G. D. Hawkins, G. C. Lynch, D. J. Giesen, I. Rossi, J. W. Storer, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.

This version was announced in QCPE Bulletin 16, 11-13(1996). In fall 1996, we provided QCPE with an update, version 5.4.1.

(M) "AMSOL-version 6.1" G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.

This was the first version distributed by Oxford Molecular Group. This version was sent to OMG in May 1997.

(N) "AMSOL-version 6.1.1" G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.

This bug fix was sent to OMG in August 1997.
This version was sent to Makolab in October 1997.

(O) "AMSOL-version 6.5" G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, J. Li, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.

This version was sent to OMG in February 1998.

(P) "AMSOL-version 6.5.1" G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, J. Li, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.

This version was sent to OMG in February 1998.
This version was sent to Makolab in April 1998.
This version was made available on April 28, 1998 via the AMSOL homepage on the World Wide Web. The URL is:
<http://comp.chem.umn.edu/amsol>

(Q) "AMSOL-version 6.5.2" G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, J. Li, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.

This version was sent to OMG in August 1998.
This version was made available on July 10, 1998 via the AMSOL homepage on the World Wide Web. The URL is:
<http://comp.chem.umn.edu/amsol>

(R) "AMSOL-version 6.5.3" G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, J. Li, T. Zhu, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.

This version was sent to OMG in October 1998.

This version was sent to Makolab in November, 1998 and to Semichem in March, 1999.

This version was made available on November 10, 1998 via the AMSOL homepage on the World Wide Web. The URL is:

<http://comp.chem.umn.edu/amsol>

(S) "AMSOL-version 6.6" G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, J. Li, T. Zhu, P. Winget, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.

This version was made available in December, 1999 via the AMSOL homepage on the World Wide Web. The URL is:

<http://comp.chem.umn.edu/amsol>

(T) "AMSOL-version 6.7" G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, J. Li, T. Zhu, P. Winget, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.

This version was made available in September, 2001 via the AMSOL homepage on the World Wide Web. The URL is:

<http://comp.chem.umn.edu/amsol>

(U) "AMSOL-version 6.7.1" G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, J. Li, T. Zhu, P. Winget, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.

This version was made available in October, 2001 via the AMSOL homepage on the World Wide Web. The URL is:

<http://comp.chem.umn.edu/amsol>

(V) "AMSOL-version 6.7.2" G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, J. Li, T. Zhu, P. Winget, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.

This version was made available in January, 2002 via the AMSOL homepage on the World Wide Web. The URL is:

<http://comp.chem.umn.edu/amsol>

(T) "AMSOL-version 6.8" G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, J. Li, T. Zhu, J. D. Thompson, P. Winget, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.

This version was made available in September, 2002 via the AMSOL homepage on the World Wide Web. The URL is:

<http://comp.chem.umn.edu/amsol>

(U) "AMSOL-version 6.9" G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, J. Li, T. Zhu, J. D. Thompson, P. Winget, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.

This version was made available in March, 2003 via the AMSOL homepage on the World Wide Web. The URL is:

<http://comp.chem.umn.edu/amsol>

(V) "AMSOL-version 7.0" G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, J. Li, T. Zhu, J. D. Thompson, P. Winget, B. J. Lynch, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.

This version was made available October 31, 2003 via the AMSOL homepage on the World Wide Web. The URL is:

<http://comp.chem.umn.edu/amsol>

14. RECENT NON-DISTRIBUTED VERSIONS

- (A) "AMSOL-version 5.9" D. J. Giesen, G. D. Hawkins, C. C. Chambers, G. C. Lynch, I. Rossi, J. W. Storer, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.
- (B) "AMSOL-version 6.0" D. J. Giesen, G. D. Hawkins, C. C. Chambers, G. C. Lynch, I. Rossi, J. W. Storer, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.
- (C) "AMSOL-version 6.5.4" G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, J. Li, T. Zhu, P. Winget, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.
- (D) "AMSOL-version 6.6.1" G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, J. Li, T. Zhu, P. Winget, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.
- (E) "AMSOL-version 6.8.1" G. D. Hawkins, D. J. Giesen, G. C. Lynch, C. C. Chambers, I. Rossi, J. W. Storer, J. Li, T. Zhu, J. D. Thompson, P. Winget, D. Rinaldi, D. A. Liotard, C. J. Cramer, and D. G. Truhlar.

15. COMPUTERS AND OPERATING SYSTEMS ON WHICH AMSOL HAS BEEN DEVELOPED

AMSOL-version 1.0 was developed for the UNICOS (Unix) operating system on the Cray-2 and Cray X-MP series of supercomputers. AMSOL-versions 3.0 and later are portable programs tested on these supercomputers, on newer supercomputers, and on Unix workstations. The computers and operating systems on which AMSOL-versions 3.0 and later have been tested are listed in Table 1.

The AMSOL program is in FORTRAN 77 with the INCLUDE extension and using lower case letters. The code is written in double precision, and on the Compaq, IBM, Macintosh, Silicon Graphics, and Sun workstations the code is executed in double precision. Since these machines have 32-bit words, this yields REAL*8 floating point variables. Note that Cray computers use 64-bit words in single precision. Thus the REAL floating point variables in the program are interpreted as REAL*8 on Cray computers. When compiling AMSOL within the Cray environment, double precision should be disabled. (This is the -dp compiler option.)

In Tables 2 and 3 below the compiler and loader commands used for testing the code are listed.

Table 1. Operating systems on the various machines on which versions 3.0 and later of the code have been tested. The combination of machine, operating system, and compiler used for the test output for each version is marked with an asterisk.

Version	Machine	Operating system	Compiler Version
7.1	Altix - Itanium 2	Linux 2.4.21	g77, GNU 3.2.3
	* Apple Dual G5	OS10.3/Darwin7.5	g77, GNU 3.5.0
	Compaq ES40 Alpha 500	Tru64 4.0f	Compaq F77 5.4-1283
	IBM SP Power3	AIX 4.3	XL Fortran 7.1
		AIX 4.3	g77, GNU 3.3.2
	IBM Power4 Regatta	AIX 5.2	g77, GNU 3.4.1
		AIX 5.2	XL Fortran 7.1
	Pentium-III	RedHat 9.0	g77, GNU 3.2.2
	Pentium 4 2.8C	Windows XP	MS Fortran PS
		Windows XP	g77, GNU 2.9.5
		WinXP/Cygwin	g77, GNU 3.3.1
	SGI Octane R14000	IRIX 6.5	MIPSPro 7.3.1.3m
		IRIX 6.5	GNU F77 3.0.2(mips)
	Sunblade 2000	Solaris 8	FD7 F95 7.0
		Solaris 8	G77 3.0.3 (sparc)
7.0	Altix - Itanium 2	RedHat 7.3	g77, GNU 3.2.2
	* Apple G4	OS10.2.8/Darwin6.8	g77, GNU 3.1
	Compaq ES40 Alpha 500	Tru64 4.0f	Compaq F77 5.4-1283
	IBM SP Power3	AIX 4.3	XL Fortran 7.1
		AIX 4.3	g77, GNU 3.3.1
	IBM Power4 Regatta	AIX 5.1	XL Fortran 7.1
	Pentium-III	RedHat 9.0	g77, GNU 3.2.2
	Pentium 4 2.8C	Windows XP	MS Fortran PS
		Windows XP	g77, GNU 2.9.5
		WinXP/Cygwin	g77, GNU 3.3.1
	SGI Octane R14000	IRIX 6.5	MIPSPro 7.3.1.3m
		IRIX 6.5	GNU F77 3.0.2(mips)
	Sunblade 2000	Solaris 8	FD7 F95 7.0
		Solaris 8	G77 3.0.3 (sparc)
	6.9	* IBM SP, Winterhawk+, Power 3 processors	AIX 4.3.4.0
IBM SP, Nighthawk, Power 3 processors		AIX 5.1	XL Fortran 7.1.1.2
IBM Regatta, Power 4		AIX 5.1	XL Fortran 7.1.1.2
SGI Origin 3800 R14000		IRIX 6.5.12f	MIPSpro 7.3.1.2m
Compaq ES40 Alpha 500		Tru64 4.0f	Compaq FORTRAN
Netfinity Linux Cluster, Pentium III processors		RedHat 7.2, kernel 2.4.9	g77, gnu v3.2
Linux workstation, Pentium III processors		RedHat 7.3 kernel 2.4.18-3smp	g77, gnu v2.96
Sun Blade 2000		Solaris 8	Forte Developer
UltraSparc III processors			7 Fortran, v7.0

6.8	* IBM SP Power3 SGI Origin 2000 R12000 SGI Origin 3800 R14000 Compaq ES40 Alpha 500 Netfinity Linux Cluster	AIX 4.3.4.0 IRIX 6.5.12f IRIX 6.5.12f Tru64 4.0f RedHat 7.2, kernel 2.4.9	XL Fortran 6.1.0.3 MIPSpro 7.3.1.2m MIPSpro 7.3.1.2m Compaq FORTRAN g77, by gnu v3.0.2
6.7.2	* IBM SP Power3 SGI Origin 2000 R12000 SGI Origin 3800 R14000 SGI Indigo2 R10000 Compaq ES40 Alpha 500	AIX 4.3.4.0 IRIX 6.5.12f IRIX 6.5.12f IRIX 6.2 Tru64 4.0f	XL Fortran 6.1.0.3 MIPSpro 7.3.1.2m MIPSpro 7.3.1.2m MIPSpro 7.2.1 Compaq FORTRAN
6.7.1	* IBM SP Power3 SGI Origin 2000 R12000 SGI Origin 3800 R14000 Compaq ES40 Alpha 500	AIX 4.3.4.0 IRIX 6.5.12f IRIX 6.5.12f Tru64 4.0f	XL Fortran 6.1.0.3 MIPSpro 7.3.1.2m MIPSpro 7.3.1.2m Compaq FORTRAN
6.7	* IBM SP Power3 SGI Origin 2000 R12000 Compaq ES40 Alpha 500	AIX 4.3.4.0 IRIX 6.5.12f Tru64 4.0f	XL Fortran 6.1.0.3 MIPSpro 7.3.1.2m Compaq FORTRAN V5.4-1283-46ABA
6.6	* IBM SP Power3 SGI Origin 2000 R10000 Compaq ES40 Alpha 500 Enterprise 4500 Server with	AIX 4.3.3.0 IRIX 6.5 Tru64 4.0f Solaris 2.7	
6.5.3	IBM RS/6000 model 550 SGI Indigo2 R10000	AIX 4.2 IRIX 6.2	
6.5.2	IBM RS/6000 model 550 SGI Indigo2 R10000	AIX 4.2 IRIX 6.2	
6.0 -	Cray C90 IBM RS/6000 model 550 SGI Indigo2 R10000	UNICOS 8.0.3 AIX 4.1 IRIX 6.2	
5.9	DEC 3000/500X AXP	OSF/1 2.1	
5.4 and	SGI Power Challenge L (R8000)	IRIX 6.1	
5.3	Cray C90 IBM RS/6000 model 590 SGI Indigo R4000 SGI Power Challenge L (R8000)	UNICOS 8.0.3 AIX 3.2.5 IRIX 5.2 IRIX 6.1	
5.2.1	SGI Power Challenge L (R8000) Sun SPARCStation IPX	IRIX 6.0.1 SunOS 4.1.2	
5.2	Cray C90 IBM RS/6000 model 550 SGI Power Challenge L (R8000)	UNICOS 8.0.3 AIX 4.1 IRIX 6.0.1	

5.0.1	Cray C90	UNICOS 8.0.3
	SGI Indigo R4000	IRIX 5.2
	SGI Power Challenge L (R8000)	IRIX 6.0.1
	Power Macintosh 7100/66	MacOS 7.1.2
5	Cray C90	UNICOS 8.0.3
	Cray-2	UNICOS 8.0.3
	Cray X-MP-EA	UNICOS 8.0.3
	DEC 3000/500X AXP	OSF/1 2.1
	IBM RS/6000 model 550	AIX 3.2.5
	SGI Indigo R3000	IRIX 5.2
	Sun SPARCStation IPX	SunOS 4.1.4

Table 1. Operating systems on the various machines on which version 3.0 and later of the code have been tested. (Continued)

Versio	Machine	Operating systems	Compiler Version
4.6	Cray C90	UNICOS 7.C.3	
	Cray X-MP-EA	UNICOS 7.C.3	
	IBM RS/6000 model 550	AIX 3.2.5	
	Sun SPARCStation SLC	SunOS 4.1.2	
4.5	Cray C90	UNICOS 7.C.3	
	Cray-2	UNICOS 7.0.5	
	Cray X-MP-EA	UNICOS 7.C.3	
	DEC 3000/500X AXP	OSF/1 1.3	
	IBM RS/6000 model 550	AIX 3.2.3	
	SGI-4D/310GTXB	IRIX V.3	
	Sun SPARCStation SLC	SunOS 4.1.2	
4.1	Cray C90	UNICOS 7.C.3	
	Cray-2	UNICOS 6.1	
	Cray X-MP-EA	UNICOS 7.C.3	
	DEC 3000/500X AXP	OSF/1 1.3	
	IBM RS/6000 model 550	AIX 3.2.3	
	SGI-4D/310GTXB	IRIX 4.0.1	
	SGI Indigo R4000	IRIX 4.0.5F	
	Sun SPARCStation IPX	SunOS 4.1.2	
Sun SPARCStation LC	SunOS 5.1		
4	Cray C90	UNICOS 7.C.2	
	Cray-2	UNICOS 6.1	
	Cray X-MP-EA	UNICOS 7.C.2	
	DEC 3000/500X AXP	OSF 1.3	
	IBM RS/6000 model 550	AIX 3.2.3	
	SGI-4D/310GTXB	IRIX 4.0.1	
	SGI Indigo R4000	IRIX 4.0.5F	
	Sun SPARCStation IPX	SunOS 4.1.2	
Sun SPARCStation LC	SunOS 5.1		
3.5	Cray C90	UNICOS 7.C.2	
	Cray-2	UNICOS 6.1.1	
	Cray X-MP-EA	UNICOS 7.C.2	
	DEC 3000/500 AXP	OSF 1	
	IBM RS/6000 model 550	AIX 3.2.3	
	SGI-4D/310GTXB	IRIX 4.0.1	
	SGI Indigo R4000	IRIX 4.0.5F	
	Sun SPARCStation IPX	SunOS 4.1.2	
Sun SPARCStation LC	SunOS 5.1		

Table 1. Operating systems on the various machines on which version 3.0 and later of the code have been tested. (Continued)

Versio	Machine	Operating systems	Compiler Version
3.0.3	Cray C90	UNICOS 7.C.2	
	Cray-2	UNICOS 6.1	
	Cray X-MP-EA	UNICOS 7.C.2	
	DEC 3000/500 AXP	OSF 1	
	IBM RS/6000 model 550	AIX 3.2	
	SGI-4D/310GTXB	IRIX 4.0.1	
	SGI Indigo R4000	IRIX 4.0.5F	
	Sun SPARCStation IPX	SunOS 4.1.2	
	Sun SPARCStation LC	SunOS 5.1	
3.0.2	Cray M92	UNICOS 6.1	
	Cray-2	UNICOS 6.1	
	Cray X-MP-EA	UNICOS 6.1	
	DEC 3000/500 AXP	OSF 1	
	IBM RS/6000 model 550	AIX 3.2	
	SGI-4D/310GTXB	IRIX 4.0.1	
	SGI Indigo R4000	IRIX 4.0.5F	
	Sun SPARCStation IPX	SunOS 4.1.2	
	Sun SPARCStation LC	SunOS 5.1	
3.0.1	Cray Y-MP	UNICOS 6.1.5a	
	Cray-2	UNICOS 6.1	
	Cray X-MP-EA	UNICOS 6.1 and UNICOS 7.C.2	
	IBM RS/6000 model 550	AIX 3.2	
	SGI-4D/310GTXB	IRIX 4.0.1	
	Sun SPARCStation 4/330	SunOS 4.0.3	
3	Cray Y-MP	UNICOS 6.1.5a	
	Cray-2	UNICOS 6.1	
	Cray X-MP-EA	UNICOS 6.1	
	IBM RS/6000 model 550	AIX 3.2	
	SGI-4D/310GTXB	IRIX 4.0.1	
	Sun SPARCStation 4/330	SunOs 4.0.3	

Table 2. Recommended compiler commands for various machines.

Machine	Compiler commands
Compaq ES40	f77 -c -static -i4 -real_size 32 -O4
Cray C90	cft77 -dp -i64 -a static
Cray X-MP-EA	cft77 -dp -i64 -a static
DEC 3000/500X AXP	f77 -c -static -i4 -real_size 32 -O4
IBM RS/6000	xlf -c -qdpq -qmaf -O2 or xlf -c -qdpq -qmaf -O2 -P
IBM SP Power3	xlf -c -qdpq -qmaf -O2
SGI Indigo R4000	f77 -c -O2 -Olimit 1300 -static -i4
SGI Indigo R10000	f77 -c -O3 -d8 -i4 -static -mips4 -OPT:roundoff=3:IEEE_arithmetic=3: fast_sqrt=off:const_copy_limit=13000 -backslash
SGI Origin 2000 R10000	f77 -c -O3 -d8 -i4 -static -mips4 -OPT:roundoff=3:IEEE_arithmetic=3: fast_sqrt=off:const_copy_limit=13000 -backslash
SGI Origin 2000 R12000	f77 -c -O3 -d8 -i4 -static -mips4 -OPT:roundoff=3:IEEE_arithmetic=3: fast_sqrt=off:const_copy_limit=13000 -backslash
SGI Origin 2000 R14000	f77 -c -O3 -d8 -i4 -static -mips4 -OPT:roundoff=3:IEEE_arithmetic=3: fast_sqrt=off:const_copy_limit=13000 -backslash
SGI Power Challenge L	f77 -c -O3 -d8 -i4 -static -mips4 -OPT:roundoff=3:IEEE_arithmetic=3: fast_sqrt=off:const_copy_limit=13000 -backslash
Machines running RedHat Linux, version 7.2 and kernel 2.4.9 or higher	g77 -finit-local-zero -fno-automatic -c -o
Power Macintosh 7100/66	FORTRAN.PPC -opt=0
Sun with SunOS	f77 -c -O3 -temp

Table 3. Recommended loader commands.

Machine	Loader commands
Compaq ES40	f77 -o
Cray C90	segldr -o
Cray-2	segldr -
Cray X-MP-EA	segldr -o
DEC 3000/500X AXP	f77 -o
IBM RS/6000	xlfr -o
IBM SP Power 3	xlfr -o
SGI Indigo R4000	f77 -tU -o
SGI Indigo2 R10000	f77 -tU -o
SGI Indigo2 R12000	f77 -tU -o
SGI Origin 2000 R10000	f77 -64 -lfastm -mips4 -o
SGI Origin 2000 R12000	f77 -64 -lfastm -mips4 -o
SGI Origin 2000 R14000	f77 -64 -lfastm -mips4 -o
SGI Power Challenge L	f77 -lfastm -o
Machines running RedHat Linux, version 7.2 and kernel 2.4.9	g77 -o
Power Macintosh 7100/66	PPCLink -w
Sun with SunOS	f77 -o -O3

16. ACKNOWLEDGMENTS

We thank Yao-Yuan Chuang for the Linux implementation, Patton Fast for various forms of assistance, and Alek Marenich and Hannah Leverentz for improvements in the manual.