Internal Report to
Office of Drinking Water
October 1986

ESTIMATION OF AROMATIC SOLUTE SOLUBILITY IN MISCIBLE SOLVENT/WATER SYSTEMS

Jaw-Kwei Fu
Charles L. Brooks, III
Richard G. Luthy

Department of Civil Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213-3890

No. CR-812147

Project Officer

A. Lynn Wood
Processes and Systems Research Division
Robert S. Kerr Environmental Research Laboratory
Ada, Oklahoma 74820

ROBERT S. KERR
ENVIRONMENT RESEARCH LABORATORY
OFFICE OF RESEARCH AND DEVELOPMENT
U.S. ENVIRONMENT PROTECTION AGENCY
ADA, OKLAHOMA 74820

DISCLAIMER

This document is intended for internal Agency use only. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

ABSTRACT

This report describes the development of computational methodologies and computer programs that may be employed to estimate aromatic organic solute solubility in miscible polar solvent/water mixtures. This information is used to predict the sorption partition coefficient for sorption of aromatic solutes onto soils or sediments in aqueous systems containing miscible polar solvent. These procedures assist in the prediction of facilitated, near-source, solute transport in soil or sediment in the event of spill or discharge of organic waste containing water-soluble solvents.

The chemical thermodynamic basis for estimating organic solute solubility in water and in solvent/water mixtures is reviewed. This information is synthesized and employed in the design of a computer program, named AROSOL, to aid prediction of aromatic solubility in water and in miscible organic solvent/water mixtures. The program AROSOL is formulated to accommodate various levels of input data and physical constants. The program utilizes four techniques to predict solute solubility in solvent/water mixtures:

(i) Log linear,

(ii) UNIFAC,

(iii) Excess free energy, and

(iv) Molecular surface area.

The user may select any or all of these techniques to evaluate solubility depending on the availability of data, physical constants, and other specific information required for each approach.

The solubility prediction is then used in conjunction with a chemical thermodynamic sorption model to estimate solute sorption partition coefficient, K_p, in water and in solvent/water mixtures.

This report also describes the development of a general purpose program, named

MOLACCS, to compute molecular surface area. The program MOLACCS is used to estimate molecular surface area for use in solute solubility predictions. Three types of surface areas are computed:

(i) The solvent accessible area,

(ii) The contact surface area, and

(iii) The van der Waals area

The program allows for specification of solvent, or probe radius, and individual atomic parameters including degree of hydrophobicity. The contribution of each atom to the surface area is displayed, as well as the net value of the individually estimated atomic group contributions to hydrophobic and polar surface area. The program is designed for use by the non-expert, in which molecules are constructed from existing atomic groups and molecular fragments. The user may construct new molecules and molecular fragments through the program operations comprising: building, perturbing, replacing, adding, and combining.

The report presents various sample calculations for both AROSOL and MOLACCS. Program listings are also included.

TABLE OF CONTENTS

Chapter One

INTRODUCTION

The purpose of this investigation is to develop computational procedures which can be used to estimate the solubility of aromatic solutes in miscible solvent/water mixtures. This information can then be employed to predict the partition coefficient for sorption of aromatic solutes onto soils or sediments.

The effect of miscible solvents, e.g. low molecular weight polar sovents, on solubility of aromatic solutes would be evident wherever water, miscible solvents, and solutes are comingled, such as in concentrated wastewaters or in waste liquids from chemical manufacturing. The effect of miscible solvents on sorption of aromatic solutes onto soils would be manifested when concentrated waste liquids contact soil or sediment material. Understanding the combined effects of solvents on solubility and sorption will aid the assessment of the tendency for aromatic solutes to undergo facilitated transport in soil/sediment systems in the presence of miscible polar solvents. This will allow for prediction of near-source contaminant transport in soils in the event of spillage or discharge of organic waste containing water-soluble solvents.

The following chapter describes the general methodologies employed in the computational procedures for estimation of solute solubility in water and in miscible solvent/water systems. Also described are the effects of organic solvents on sorption of aromatic solutes onto soil, and the calculation of molecular surface area and its use in predicting solubility. This is followed by a discussion of the use of the computer programs and example calculations. An appendix contains a listing of the computer programs. A summary of the organization and content of this report is presented below.

The chemical thermodynamic basis for estimating organic solute solubility in water and in solvent/water mixtures is reviewed. This information is synthesized and employed in the design of a computer program, named AROSOL, to aid prediction of aromatic solute solubility in water and in miscible organic solvent/water mixtures. The program has been

specifically designed to aid the prediction of aromatic solute solubility; however, the general methodologies and procedures are also applicable for other organic compounds as well. The program AROSOL is formulated to accommodate various levels of input data and physical constants. The program utilizes four techniques to predict solute solubility in solvent/water mixtures:

(i) Log linear,

(ii) UNIFAC,

(iii) Excess free energy, and

(iv) Molecular surface area

The user may select any or all of these techniques to evaluate solubility depending on the availability of data, physical constants, and other specific information required for each approach.

The solubility prediction is then used in conjunction with a chemical thermodynamic sorption model to estimate solute sorption partition coefficient, K_p, in water and in solvent/water mixtures.

The report then describes the development of a general purpose program, named MOLACCS, to compute molecular surface area. The program MOLACCS is used to estimate molecular surface area for use in solute solubility predictions. Three types of surface areas are computed:

(i) The solvent accessible area,

(ii) The contact surface area, and

(iii) The van der Waals area

This program allows for specification of solvent, or probe radius, and individual atomic parameters including degree of hydrophobicity. The contribution of each atom to the surface area is displayed, as well as the net value of the individually estimated atomic groups contributions to hydrophobic and polar surface area. The program is designed for

use by the non-expert, in which molecules are constructed from existing atomic groups and molecular fragments. The user may construct new molecules and molecular fragments through operations comprising: building, perturbing, replacing, adding, and combining.

The report presents various sample calculations for both AROSOL and MOLACCS. Program listings are also included.

Chapter Two

SOLUTE SOLUBILITY AND SORPTION ONTO SOIL IN WATER AND MISCIBLE SOLVENT-WATER SYSTEMS

The following is a synthesis of current chemical thermodynamic techniques that may be used to estimate solute solubility in miscible solvent/water systems. This information is used in conjunction with a chemical thermodynamic sorption model to describe a methodology by which the sorption of aromatic solutes onto soils may be predicted for liquid phases comprised of miscible solvent–water systems. This discussion is adapted in part from the methodological procedures presented in Fu and and Luthy (1986 a and b) with additional information being provided on the subject of molecular surface area calculations. The techniques described below have been incorporated in the computer program named AROSOL for estimation of aromatic compound solubility in miscible solvent/water mixtures, and for prediction of aromatic solute sorption onto soils and sediments in solvent/water mixtures. This chapter also explains techniques for molecular surface area calculations and their use in predicting solubility.

Phase Equilibria and Activity

Expressions that relate solute activity coefficient and mole fraction are employed in several techniques to estimate solute solubility in solvent/water systems. The relationship between activity and solubility are described below.

The solubility of a liquid or solid non–electrolyte solute in aqueous solution can be described by the thermodynamics of phase equlibria. The solute chemical potential can be expressed in terms of fugacity, and the aqueous solubility of a hydrophobic organic compound in water or water/miscible solvent mixture can be expressed in terms of fugacity and activity using the Raoult's law convention. For liquid components the relationship between mole fraction solubility, X, and activity coefficient, γ, is

$$X = \frac{1}{\gamma} \qquad (1)$$

Both terms in this equation are dimensionless. For solid solutes (e.g., naphthalene) the relationship between mole fraction and activity is

$$X = \frac{f^s}{f^R \gamma} \tag{2}$$

in which f^s = the pure solid fugacity, or vapor pressure; and f^R = the reference or standard state fugacity, which is usually defined as an extrapolated pure component liquid vapor pressure below the triple point. As explained below, the ratio (f^s/f^R) can be estimated by several standard procedures.

The conventional definition of the reference state for a solvent, identified as component 1, is $\gamma_1 \rightarrow 1$ as $X_1 \rightarrow 1$. For a solute, identified as component 2, the reference state is the infinitely dilute solution. In general, $\gamma_2 > 1$ for a dilute solution of a given component in a given solvent. As the solution becomes increasingly dilute, the value of γ_2 approaches a limiting value, γ_2^{∞}, known as the infinite dilution activity coefficient (Prausnitz, 1969). Knowledge of the infinite dilution activity coefficient can be used to estimate solute solubility in water and in solvent/water mixtures, as well as other physico-chemical properties important to the environmental scientist, such as solvent/water partition coefficients and Henry's law constants (Campbell and Luthy, 1985; Grain, 1982).

A simplified expression for (f^s/f^R) employs heat of fusion of the solid, $\Delta H_{f,m}$, cal/mole, at the melting point (Prausnitz, 1969)

$$\ln (\gamma_2 X_2) = \frac{\Delta H_{f,m}}{RT}\left[\frac{T}{T_m} - 1\right] = \ln \frac{f^s}{f^R} \tag{3}$$

in which T = system temperature, °K; T_m = melting temperature of pure solid, °K; and R = the gas constant, 1.987 cal/mole-°K. Eq. 2 neglects certain correction terms including those depending on the difference of specific heat between solid and liquid, ΔC_p, cal/mole-°K. Hildebrand and Scott (1950) proposed that the heat of fusion of a solid at a temperature, T, can be calculated from the heat of fusion at the melting point:

$$\Delta H_f = \Delta H_{f,m} - \Delta C_p (T_m - T) \tag{4}$$

in which ΔH_f = heat of fusion at the system temperature. This expression can be used to compensate in part for some of the error introduced by omission of the ΔC_p term in the simplified fugacity ratio expression. Heat capacity data are available for relatively few solid

solutes, and thus it is fortunate that the correction is usually small compared to the $\Delta H_{f,m}$ term, as well as compared to other uncertainties in estimating activity coefficient (Gmehling et al., 1978; Prausnitz, 1969).

In the case of solid solutes for which heat of fusion data are not available, heat of fusion may be calculated from entropy of fusion, ΔS_f (cal/mole-°K)

$$\Delta H_f = T_m \Delta S_f \tag{5}$$

For many moderately-sized organic molecules, including substituted aromatic hydrocarbons, the entropy of fusion is reported to be nearly constant at about 13 - 13.5 cal/mole-°K (Tsonopoulus and Prausnitz, 1971; Yalkowsky, 1979). Then, Eqs. 3 and 5 reduce to the following relation, assuming ΔS_f is constant at 13 cal/mole-°K

$$\ln (\gamma_2 X_2) = 6.56 \left[\frac{T - T_m}{T} \right] \tag{6}$$

The AROSOL computer program which was developed for the project allows for the determination of the fugacity ratio expression as follows. $\Delta H_{f,m}$ is employed according to Eq. 3 if heat of fusion data are available. If $\Delta H_{f,m}$ is not available, then Eq. 5 is used in conjunction with Eq. 3 as the fugacity ratio expression, which reduces to Eq. 6.

Estimation of Infinite Dilution Activity Coefficient

The solute infinite dilution activity coefficient, and the solvent and water activity coefficients, are estimated in the computer program by a group contribution method using the Universal Quasi-Chemical Functional Group Activity Coefficient (UNIFAC) approach. This approach computes the activity coefficients from knowledge of the molecular structure of the solute and the solvents through equations that employ a data base comprising functional group size and interaction parameters. This represents an especially utilitarian technique for prediction of chemical properties of mixtures, as no specific experimental data or correlation coefficients are required.

The UNIFAC approach is a group contribution method for predicting activity coefficients of nonelectrolytes in liquid mixtures (Fredenslund et al., 1975). The model assumes that the logarithm of the activity coefficient is comprised of two parts

$$\ln \gamma = \ln \gamma^{c} + \ln \gamma^{R} \quad (7)$$

Herein, γ^{c} is a combinatorial part due to the difference in molecular size and shape of the molecules in a mixture, and γ^{R} is a residual part due to molecular interactions. The computational procedures employed in the UNIFAC model are described in Fredenslund et al. (1975) and Gmehling et al. (1978). The most current tabulation of group size and interaction parameters is given by Gmehling et al. (1982).

The UNIFAC approach has been applied to various problems for estimation of solution properties including estimating activity coefficients in organic solvent mixtures (Fredenslund et al., 1979), estimating the solubility of a solid in a solvent (Martin et al., 1981), estimation of octanol/water partition coefficient (Arbuckle, 1983), predicting the solubility of organic compounds in water (Banerjee, 1984, 1985), estimating aromatic solute distribution coefficients for both polar and nonpolar organic compounds (Campbell and Luthy, 1985), and estimating solute solubility in solvent/water mixtures (Fu and Luthy, 1985). The calculation of solvent and water activity coefficients, and solute infinite dilution activity coefficient, was facilitated by the adaptation of a computer program developed by Fredenslund et al. (1975) and updated by Anderson (1983). Examples of the computational methodology for predicting activity coefficients are provided by Grain (1982) and Fu and Luthy (1985).

The UNIFAC approach was adapted in this investigation in order to estimate solute infinite dilution activity coefficient (γ^{∞}) in pure solvent and pure water, and in solvent/water mixtures. The methodology entailed treating the solvent/water system as one component and the solute as a second component. The solvent/water system was treated as a single component to facilitate estimation of solute solubility from γ^{∞}. In this approach the solvent/water system may be envisioned as comprised of a "molecule" of water and solvent in proportion to the mole fraction solvent/water composition of interest. It does not matter if the "molecule" is comprised of water and solvent in noninteger ratios when using a solution-of-groups approach to estimate activity coefficients.

For the case of liquid solutes, mole fraction solubility can be estimated from γ^{∞} using Eq. 1, provided γ^{∞} is sufficiently large, i.e., $\gamma^{\infty} > 1{,}000$. If γ^{∞} is $< 1{,}000$, then the mole fraction solubility calculation must account for the fact that at infinite dilution there is appreciable solubility of solute, and that the solvent system mole fraction does not approach unity. In these cases, mole fraction solubility is estimated according to procedures described by Lyman (1982), which includes an evaluation to determine if the component is completely miscible. For solid solutes, mole fraction solubility can be estimated from γ^{∞} with Eq. 3 if γ^{∞} is > 100. If γ^{∞} is < 100 then the mole fraction solubility is estimated according to Lyman (1982) to account for the solvent system mole fraction being less than unity.

It is useful for purposes of practical environmental engineering calculations to be able to express solute concentration in terms of molar concentration (moles per liter) or mass concentration (mg or g per liter). This introduces a small difficulty because in computing molar or mass composition from mole fractions, it is necessary to incorporate a value for the volume of the solute/solvent/water mixture. In using the activity coefficient data to compute solute molar or mass concentration, it was assumed that the separate volumetric contributions of the solute, solvent, and water are conserved. This assumption is often made in theories pertaining to thermodynamic properties of mixtures of nonelectrolytes (Hildebrand et al., 1970). This assumption was addressed in experiments and discussion by Fu and Luthy (1985) for the case of methanol/water and acetone/water mixtures. In these cases the conservation of volume assumption generally resulted in errors less than about 5%, and often less than 1 or 2%.

Solubility Prediction by an Excess Free Energy Approach

Williams and Amidon (1984a) derived relationships between solute activity coefficient, solute Henry's law constant in pure solvent, and solute-free solvent and water volume fractions. These relationships were then used with an expression for the excess Gibbs free energy of mixing to estimate the solubility of a solute in a binary solvent system. The resulting expression was

$$\ln X_2 = z_1 \ln X_s + z_3 \ln X_w - A_{1\text{-}3} z_1 z_3 (2z_1 - 1) \frac{q_2}{q_1} + 2A_{3\text{-}1} z_1^2 z_3 \frac{q_2}{q_3} + C_2 z_1 z_3 \quad (8)$$

in which the subscripts 1, 2, and 3 refer to solvent, solute, and water, respectively; z_i = solute-free volume fraction of solvent or water; q_i = molar volume for component i; $A_{3\text{-}1}$ and $A_{1\text{-}3}$ = solvent/water interaction constants (dimensionless). The first two terms in Eq. 8 represent proportional solubility of solute in pure solvent and water. The next two terms represent the contribution of solvent/water interactions, and the last term accounts for the interactions between solute and solvent/water. This last term, C_2, is essentially a ternary correction parameter. The solvent-water interaction constants may be estimated from the molar excess free energy of mixing for solute-free systems (Williams and Amidon, 1984b)

$$X_1 \ln \gamma_1 + X_3 \ln \gamma_3 = \frac{A_{1\text{-}3}}{q_1}[z_1 z_3^2 (X_1 q_1 + X_3 q_3)] + \frac{A_{3\text{-}1}}{q_3}[z_1^2 z_3 (X_1 q_1 + X_3 q_3)] \quad (9)$$

X_1 and X_3 = solvent and water mole fraction, respectively; and γ_1 and γ_3 = solvent and water activity coefficients, respectively. Williams and Amidon (1984b) determined $A_{1\text{-}3}$ and $A_{3\text{-}1}$ for a solvent/water mixture from estimation of γ_1 and γ_3 through use of experimental partial pressure data. In the present AROSOL program, γ_1 and γ_3 are estimated for different solvent/water compositions by the UNIFAC method. The constants $A_{1\text{-}3}$ and $A_{3\text{-}1}$ are then obtained by a two-parameter statistical regression of Eq. 9. The statistical regression procedure is performed according to the techniques described by Ryan et al. (1981).

Williams and Amidon (1984b) employed Eqs. 8 and 9 to describe the solubility in ethanol/water mixtures for ten compounds of interest in pharmaceutical science, where C_2 was estimated by linear regression of the difference between experimental solubility and calculated solubility without the C_2 term using Eq. 8. It was noted that the C_2 term was correlated with the solute octanol-water partition coefficient, K_{ow}. This suggested that it may be possible to estimate C_2 from octanol-water partition coefficient data, and then use an estimated C_2 term in conjunction with the solvent-water terms and pure solvent solubilities to predict the solute solubility-solvent/water composition profile. In additional work, Williams and Amidon (1984c) concluded that Eq. 8 predicted a semi-logarithmic

increase in solute solubility with volume fraction solvent when the solvent/water interactions were small compared to the interaction between the solute and the mixture.

In summary, three parameters must be determined in order to use the excess free energy approach: the solvent interaction parameters A_{1-3} and A_{3-1}, and the solute–solvent interaction parameter, C_2. The solvent interaction parameters are determined in the AROSOL program from estimation of activity coefficients for the solvent and water using the UNIFAC procedure with Eq. 9 and a two-parameter statistical regression technique. The results of this technique for four solvent–water systems have shown that the parameters, A_{1-3} and A_{3-1}, are constants for the binary solvent systems being considered (Fu and Luthy, 1986a). The computer program developed for this investigation uses Eq. 9 to determine A_{1-3} and A_{3-1}, and considered these parameters as constants, as found previously Fu and Luthy (1986a) and Williams and Amidon (1984b, 1984c).

The solute–solvent interaction parameter, C_2, was estimated from the experimental data of Fu and Luthy (1985). Statistical analysis of the data for eighteen aromatic solute/solvent/water systems investigated by Fu and Luthy (1985) showed that C_2 may be correlated with aromatic solute octanol–water partition coefficient, K_{ow}, by

$$C_2 = -2.69 - 1.22 \log K_{ow} \qquad r^2 = 0.86 \tag{10}$$

For comparison, Williams and Amidon (1984b) found that the correlation for eight solutes and ethanol–water was

$$C_2 = 3.96 - 2.66 \log K_{ow} \qquad r^2 = 0.90 \tag{11}$$

The AROSOL computer program employs Eq. 10 for prediction of solubility by the excess free energy approach. This is because it is desired to employ an expression for C_2 that is more appropriate for aromatic solutes, rather than a more general, but less precise regression equation. Nonetheless, the user of the program has the flexibility to modify the expression for C_2 as conditions may warrant. Fu and Luthy (1986a) have shown that statistical analysis of the combined eighteen aromatic solute/solvent/water systems investigated by Fu and Luthy (1985), plus the eight solute/ethanol/water systems reported by Williams and Amidon (1984b), results in a correlation for C_2 with $r^2 = 0.69$. The poor

correlation for the larger data set suggests that the expression for C_2 is either system-specific, or that the expression for C_2 must be expanded to include additional terms.

Solubility Prediction by Log-Linear Relationships

Yalkowsky et al. (1972) reported on the solubility of alkyl p-aminobenzoates in water-propylene glycol mixtures, and found that the solubility could be described by a semi-logarithmic relationship,

$$\log S_2 = \log S_w + \sigma z_1 \qquad (12)$$

in which S_2 = the solute solubility in the mixture, moles/L; S_w = the solute solubility in water, moles/L; z_1 = the volume fraction of solvent; and σ = a parameter that was characteristic of the system under study. Later it was reported by Yalkowsky and Flynn (1974) that the solubility of certain compounds in solvent/water mixtures required Eq. 12 to be expanded to a fifth-degree polynomial in z_1 in order to account for non-linearity of solubility with increasing fraction of solvent. Eq. 12 has been examined by Martin et al. (1982) where it was noted that it was applicable to systems where the polarity of the compound was significantly less than either of the solvents in the binary mixtures. It was shown (Martin et al., 1982) that the linear dependence of logarithmic solubility on volume fraction of the solvent was applicable when the Hildebrand solubility parameter of solvent was larger than the solubility parameter of the solute. The Hildebrand solubility parameter, δ, in units of $(cal/cm^3)^{1/2}$, is defined as the square root of the pure liquid component cohesive energy density. Weast (1983) and Barton (1983) provide tabulations of δ for various compounds. These values typically range from less than ten (e.g., δ = 7.3 for hexane) to over 20, as that for water (δ = 23.45). Eq. 12 is related to the Hildebrand solubility theory (Martin et al., 1982), where it can be shown that the mole fraction solubility of solute may be given as a power series in terms of solvent volume fraction, z_1, and constants

$$\log X_2 = \log X_w + \log \gamma_w - K_o + K_1 z_1 - K_2 z_1^2 + \ldots \qquad (13)$$

in which the subscript w refers to water; and K_o, K_1, and K_2, etc., are the polynomial regression constants. Martin et al. (1982) showed that the solubility of semi-polar drugs in solvent/water systems could be described by a simplified expression

$$\log X_2 = \log X_w + \log \gamma_w - K_o + K_1 z_1 \quad (14)$$

If regression analysis of Eq. 14 was performed with perfect accuracy then the log γ_w and the K_o terms would cancel in order that $X_2 = X_w$ as $z \rightarrow 0$. Thus, Eq. 14 would reduce to

$$\log X_2 = \log X_w + K_1 z_1 = \log X_w + \sigma z_1 \quad (15)$$

This is a log-linear solubility relationship with σ being related to mole fraction solute solubility and volume fraction solvent. Martin et al. (1982) concluded that the log-linear solubility relationship describes systems of semipolar solutes with solvent/water mixtures when the solubility parameters of the solute is about 3 solubility parameter units lower than that of the solvent in the solvent/water mixture. When the solvent is a strong solvating agent, the solute solubility may be described by the log-linear relationship up to 100% solvent even though the solubility parameters of solute and solvent are similar.

The AROSOL computer program estimates solute solubility by the log-linear approach in units of mole fraction (Eq. 15), and these results are converted to customary units of mg per liter of solution volume, assuming conservation of volume of the separate components.

Solubility Prediction by a Molecular Surface Area Approach

Yalkowsky et al. (1976) developed a molecular surface area approach to describe the log-linear solubility relationship. This approach relates the mole fraction solubility of a liquid solute in a solvent/water mixture to the solute hydrophobic surface area (HSA) and polar surface area (PSA). The total surface area (TSA) of the solute is equal to the surface area of the polar moiety (PSA) plus the hydrocarbonaceous, or hydrophobic, surface area (HSA)

$$TSA = PSA + HSA \quad (16)$$

The solute solubility in a solvent/water mixture can be expressed as

$$\ln X_2 = \ln X_w + z_1 \left[\frac{\Delta\epsilon_H(HSA) + \Delta\epsilon_p(PSA)}{kT} \right] \quad (17)$$

in which X_2 = the mole fraction solute solubility in the solvent/water mixture; X_w = the

mole fraction solute solubility in water; z_1 = the solute-free volume fraction of solvent; k = the Boltzman constant; and T = the system temperature. $\Delta\epsilon_H$ describes the microscopic interfacial free energy between hydrocarbonaceous surface area and the solvent, and $\Delta\epsilon_p$ is an analogous interfacial free energy term, which is dependent upon the interaction between the solvent and the polar portion of the solute. For the case of relatively nonpolar compounds, the hydrophobic interactions are dominant relative to polar interactions. This is equivalent to assuming that the term ($\Delta\epsilon_H$ HSA) is much greater than the term ($\Delta\epsilon_p$ PSA). Under these conditions Eq. 17 can be reduced to

$$\ln X_2 = \ln X_w + z_1 \left[\frac{\Delta\epsilon_H(\text{HSA})}{kT}\right] \tag{18}$$

The AROSOL computer program employs Eq. (17) to estimate solute solubility by the molecular surface area approach. The user must input the values of $\Delta\epsilon_H$ and $\Delta\epsilon_p$, and HSA and PSA. A separate molecular surface area program, MOLACCS, permits the user to estimate HSA and PSA.

Presently, the molecular surface approach is limited in application by lack of information for values of microscopic interfacial energies, i.e., $\Delta\epsilon_H$ and $\Delta\epsilon_p$. These data are experimentally determined and are available for only a few solvents. This approach is also limited in that the proportion of total surface area attributable to HSA and PSA is generally not known, nor can these parameters be predicted in a straightforward manner. However, the later restriction has been reduced to a large extent by the development of the MOLACCS program in this investigation for computation of molecular surface area. As described later, the program permits the user to ascribe relative polarity values to various atomic subunits from which the proportion of polar versus nonpolar surface area may be calculated. The following section elaborates on the methods commonly employed for estimation of molecular surface areas.

Calculation of Molecular Surface Area and Its Use in Predicting Solubility

Presented below is a synopsis of the definitions of molecular surface area and methods by which molecular surface area may be calculated. This is followed by a discussion of

the use of molecular surface area for correlation with aqueous solubility and partition coefficient.

The purpose of this review is to provide the user with appropriate background information on the execution and use of molecular surface area calculations. This will help the user in making judicious selection of atomic and molecular input parameters, as well as providing a reference base from which to make rational comparisons.

Molecular Surface Area

Richards (1977) has reviewed the procedures for calculation of molecular surface area, and he has discussed some of the applications of this parameter in the field of protein chemistry. Richards notes that there is an intuitive appeal with being able to correlate thermodynamic properties of condensed phases with the packing of groups of atoms in a molecule and the area of the molecule. This appeal derives in part from geometrical concepts being generally easy to grasp, as well as from the success with which correlation with molecular surface area and molecular volume may describe phase partitioning and solubility, as well as other molecular properties that may relate to exposed surface and the nature of the exposed groups.

The calculation of molecular surface area is usually made by assigning to each atom in a molecule a bond length and bond angle, and a van der Waals radius. It is well understood that the surface of a molecule must relate to the radial distribution of electrons surrounding the molecule, and for atoms in a molecule the distribution of electrons is not spherically symmetric nor isotropic. Nonetheless, as summarized by Richards (1977), the hard sphere model of chemically bonded atoms has a long and successful record for explanation of molecular properties. Richards's view is that more realistic and complex models have improved the explanation of certain details not provided by the hard sphere model, but these approaches have not altered the principal characteristics ascribed by hard sphere models. This view has remained essentially unchanged, as summarized in the review by Pearlman (1986) which concludes that even though the electron cloud surrounding the nucleus of an atom has no well-defined surface, the intuitive appeal and empirical success of the hard sphere or van der Waals radius concept is widely recognized.

The van der Waals surface, A_w, of a molecule may be envisioned as shown in Figure 1-a. Each atom in a molecule is represented as a sphere centered at the nucleus and having a radius equal to the van der Waals radius of the atom, r_w. The van der Waals surface is defined as the exterior surface of the union of all the van der Waals spheres in the molecule (Pearlman, 1986). The van der Waals surface area of a molecule represents the boundary surface of the molecular electronic distribution, and hence the van der Waals surface is one type of descriptor of molecular surface area that may be used in estimating solute-solvent interactions.

Hermann (1972) and Richards (1977) recognized that not all of the van der Waals surface is accessible to the solvent, depending upon the size of a solvent molecule. This is illustrated in Figure 1-b and Figure 2. These figures show a trace of the van der Waals surface of some atoms in which a spherical solvent molecule with radius r_{solv}, or probe with radius R_1, is allowed to trace the van der Waals surface by rolling on the outside of the van der Waals surface. Figure 2 illustrates that atoms 3, 4 and 11 are never contacted by the probe, and as such these atoms may be considered as interior atoms which are not part of the surface of the molecule. For this reason alternate definitions of molecular surface have been proposed which attempt to account for surface in actual contact with solvent.

One procedure for defining the surface of a molecule which is in contact with solvent is to use the continuous sheet described by the locus of the center of the probe as the probe rolls over the van der Waals surface. This is termed the accessible surface by Richards (1977) and Pearlman (1986). Another procedure is to consider those parts of the molecular van der Waals surface that can actually be in contact with the surface of the probe. This is termed the contact surface by Richards (1977), and it is illustrated by the heavy line in Figure 2 for a probe of radius R_1. The definition of contact surface results in a series of disconnected patches. The patches in contact with the probe are separated by a segment given by the interior-facing part of the probe when it is simultaneously in contact with more than one atom. These interior-facing segments are termed the reentrant surface by Richards (1977). Taken together, the contact surface and the reentrant surface

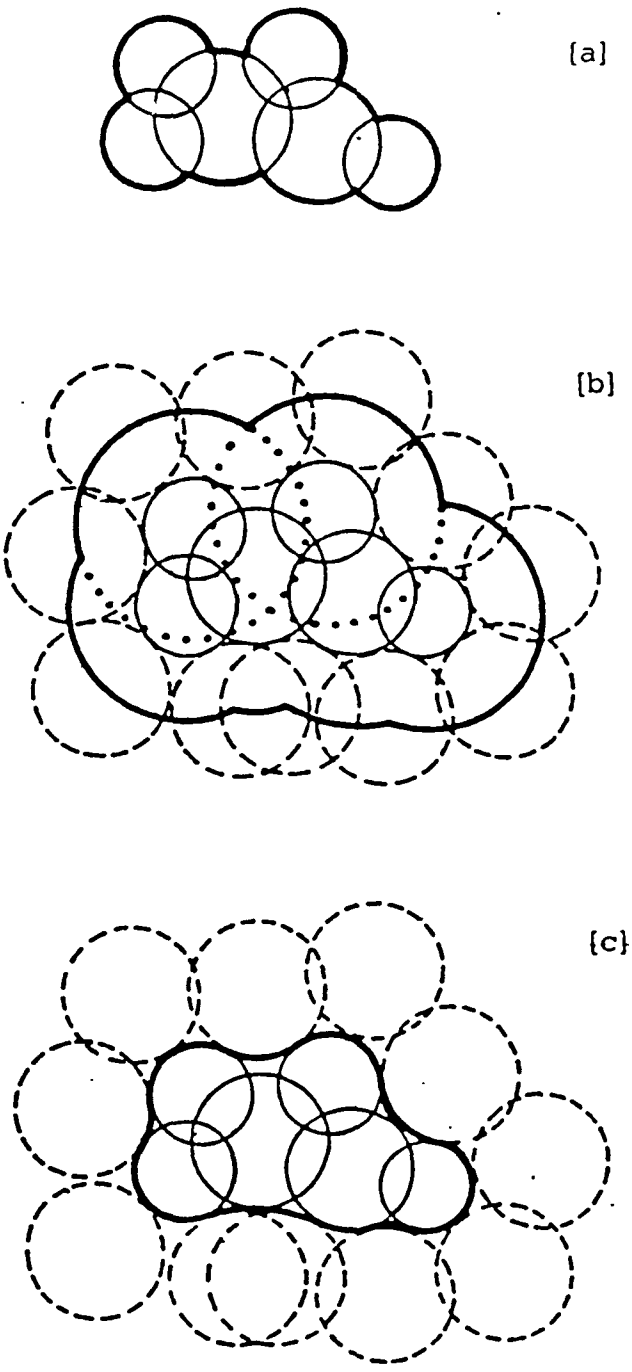
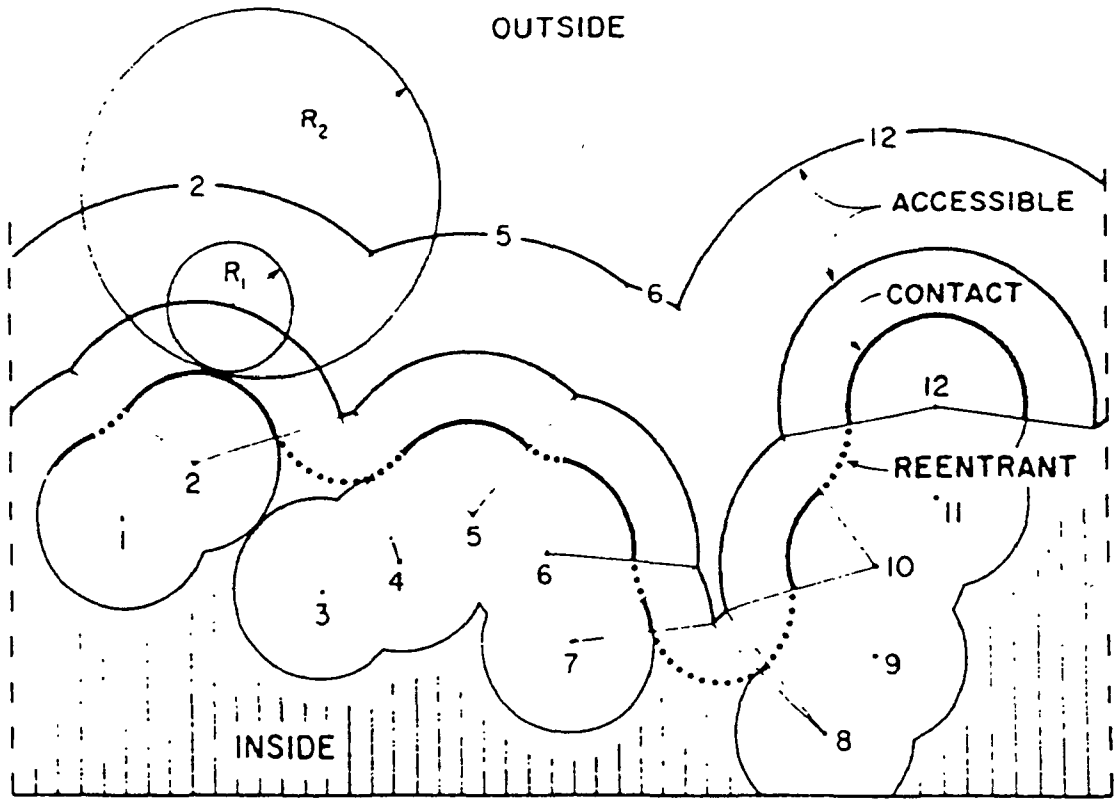


Figure 1. Molecular surface area definitions, after Pearlman (1986)



Schematic representation of possible molecular surface definitions. A section through part of the van der Waals envelope of a hypothetical protein is shown with the atom centers numbered.

Figure 2. Molecular surface area definitions and features, after Richards (1977)

represent a continuous sheet that is termed molecular surface by Richards (1977); this continuous sheet is termed contact surface by Pearlman (1986). Hence, there is a conceptual difference in definition of contact surface among these authors. Pearlman's definition of contact surface includes the reentrant surface as defined by Richards. Richards (1977) explains, as in Figure 2 by the nature of the geometrical construction, that the accessible surface has no reentrant sections.

Note that as the size of the probe, or solvent radius, approaches zero the accessible surface area approaches the van der Waals surface area. Also, small changes in the choice of solvent radius can have a relatively large effect on the accessible surface area. Figure 2 shows for a change in probe size from R_1 to R_2, that the accessible surface becomes much smoother and the number of interior atoms increases. It is generally agreed that the smallest reasonable probe is a water molecule, which is considered as a sphere with a radius of 1.4 or 1.5 Å

Calculation of Surface Area

For covalently bonded atoms the van der Waal hard shell spheres are normally truncated by a plane perpendicular to the interatomic bond, with the plane chosen to divide the bond into two segments proportional to the radii of the bonded atoms. Another normal approximation is to specifically include the contribution of hydrogen atoms to the van der Waals surface. This technique incorporates the radius of the hydrogen atoms into that of the heavy atoms to which they are bonded. This is because the bond length and van der Waal radii for atoms of carbon and higher atomic number result in the hydrogen being "buried" to a great extent within the radii of the larger atom. Hence, a common procedure is to expand the heavy atoms, and C,N,O and S, into a series of groups with zero, one, two or three hydrogen atoms attached, with each one of these groups considered to be spherically symmetrical (Richards, 1977). This procedure is termed an extended atom approach, in which the radius for each atomic group attempts to account for the contribution of hydrogen to the surface area. Figure 3 illustrates this concept as applied to a terminal methyl group, $-CH_3$ with three tetrahedral hydrogens (Valvani et al., 1976). The van der Waal radii for aliphatic carbon was taken as 1.6, and that for hydrogen as 1.2,

with the C–H interatomic bond length of 1.09 Å. The solid curve shows the planar view of the terminal methyl group without solvent radius with three tetrahedral hydrogens, while the dotted curve shows the methyl group as single sphere with radius of 2.0 Å.

Valvani et al. (1976) evaluated procedures for calculation of surface area for various aliphatic alcohols and hydrocarbons. Three methods for computation of surface area were compared:

Method A: Hermann's procedure (1972) for van der Waals surface area, which has been adopted by Pearlman (1986), where the calculation considers the molecule as comprised of intersecting spheres with hydrogen assigned a specific van der Waals radius. The accessible surface also was computed with a solvent radius of 1.5 °A (water).

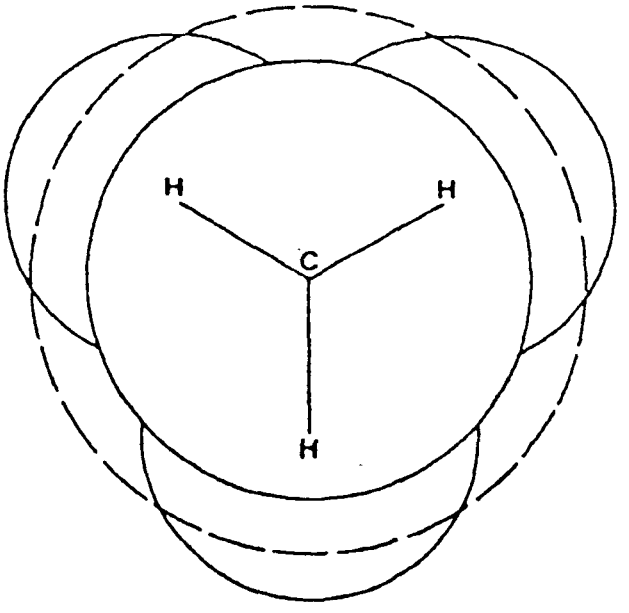
Method B: An extended atom approach was used in the calculation with methyl, $-CH_3$, methylene, $-CH_2-$, and the hydroxyl groups in alcohols, $-OH$, considered as a spherical group rather than as individual atoms. The accessible surface was computed with a solvent radius of 1.5 °A.

Method C: An extended atom procedure was used to compute the van der Waals surface area, i.e. the computation was similar to Method B with the solvent radius excluded.

Some of the conclusions from comparison of these procedures for computation of surface area were:

1. Accessible surface areas calculated by the extended atom approach, Method B, were generally very comparable to Method A.

2. Surface area was correlated with mole fraction aqueous solubility for 51 alcohols that are liquid at 25°C, plus four alcohols that are solids at 25°C for which the solubility of the pure subcooled liquid was used. The correlation showed for the 55 compounds that an expression of the form:



A planar view of a terminal methyl group. The solid curves show a carbon atom in the center with van der Waals radius of 1.6 Å and, three hydrogen atoms with van der Waals radius 1.2 Å. The broken curve shows the whole methyl group treated as a single sphere with radius of 2.0 Å.

Figure 3. Definition of extended atom approach for a terminal methyl group, after Valvani et al. (1976)

$$\log (X) = \beta \text{ Surface Area} + \alpha$$

correlated the data equally well by Method A, B or C with a correlation coefficient in the range 0.986–0.989. A similar conclusion was made for correlation of surface area with aqueous solubility for seventeen hydrocarbons ($r = 0.980$).

3. Another correlation with aqueous solubility was performed in which the total surface area for the alcoholic molecules was partitioned between hydrocarbonaceous surface area (HSA), and the polar surface area (PSA) associated with the exposed portion of the hydroxyl group. The correlating equation had the form:

$$\log (X) = \beta \text{ HSA} + \gamma \text{ PSA} + \alpha$$

where α, β and γ were correlating coefficients. This correlating equation was used to evaluate the three methods for computation of surface area. This evaluation gave essentially similar results ($r \sim 0.99$). Although the correlation coefficient was similar to that found in No. 2 above, the authors judged from these results that the extended atom approach was able to account for the contribution of the hydroxyl group to the solubility of the alcohol.

4. The extended atom approach (Method B) afforded several computational advantages over the single atom approach: (a) the extended atom technique allowed one to eliminate the arbitrary selection of a specific arrangement of hydrogen atoms in the molecule, (b) the simplest standard geometrical representation of a molecule compared to within about 2% of that calculated by Hermann's procedure which accounted for exact conformation, or weighted average when several conformations were possible, and (c) the extended atom approach offered considerable advantages in terms of computation time and costs. In summary, Valvani et al. (1976) concluded that the extended atom

approach can give at least as good, or slightly better, correlation with solubility than the single atom approach, and that the extended atom approach can be consistently utilized in solubility–surface area calculations. It was also discussed for the case of alcohols that the extended atom approach for calculation of surface area tended to eliminate from the surface area calculations the inaccessible portions of the molecule. Although somewhat inconsistent, the authors claimed that the use of extended atoms without solvent radius had much the same effect as inclusion of a solvent radius term. For this reason, method C gave comparable correlation coefficient as Method B. The authors preferred use of Method C (i.e., zero solvent radius or van der Waals surface area) as a method for estimating surface area, since this eliminated the need to arbitrarily select a solvent radius, which may vary somewhat from solvent to solvent.

Problem of Determining Hydrogen-Atom van der Waals Radius

Another fundamental reason for use of the extended atom approach relates to the experimental difficulty in determining the van der Waals radius of the hydrogen atom. Unlike interatomic bond lengths and bond angles, van der Waals radii are less well defined. Further, the parameterization with explicit hydrogen atomic parameters seems to unduely complicate the process of building and manipulation of molecules from fragments for purposes of surface area calculation. In addition, extracting explicit hydrogen van der Waals parameters is, at best, speculative. Most of the current molecular force fields models, which are used describe the interactions between molecules, use an extended atom representation (Brooks et al. 1983; Jorgensen and Swenson, 1985 and references therein), and we have adapted this procedure in the current program. The difficulty associated with extracting van der Waals parameters from standard transport data and viscosity measurements arises because the hydrogen electron density is often buried within the heavy atom to which it is attached, and consequently it is not "seen" by the experiments. Therefore, these parameters have often been either empirically adjusted to fit some set of experimental measurements or inferred from data on H_2, in which case the heavy atom parameters require empirical adjustment.

Richards (1977) concludes that "within limits the choice of van der Waals radii is arbitrary, with each author having his favorite list for the different atoms. The most appropriate values for successful predictions may vary with the problem."

Correlation of Hydrocarbon Solubility with Solute Surface Area

Various investigators have proposed relationships between molecular surface area, or molecular volume, and aqueous solubility or partition coefficient, such as octanol/water partition coefficient. These relationships derive from the earlier work of Langmuir who proposed in 1925 that the logarithm of organic compound aqueous solubility should be linearly proportional to molecular surface area. The relationship between solubility and molecular volume was proposed in Scatchard's regular solution theory in 1931 in which the logarithm of aqueous solubility is linearly proportional to molecular volume (Pearlman, 1986; Richards, 1977).

Herman (1972) explained that the number of water molecules that can be packed around a given hydrocarbon solute molecule is an important quantity in predicting solute solubility in aqueous solution. This is because there is a decrease in entropy due to the tendency of the water dipoles in the layer of water adjacent to the hydrocarbon to orient with respect to the water molecules in the next water layer. This does not happen in the bulk liquid away from the surface of a hydrocarbon molecule. The number of water molecules that can be packed around a hydrocarbon is related to the surface area of the solvent cavity if the surface area is defined as that which passes through the centers of the water molecules adjacent to the solute. This is analogous to the definition of accessible surface area.

The relationship between molar solubility, S, and accessible surface area, A_{acc}, was given by Hermann (1972) as

$$b\ A_{acc} = -kT \ln (S) - c \qquad (19)$$

where b and c are temperature dependent constants, and k is the Boltzmann constant. This type of relationship may consider different molecular confirmations by defining A_{acc} as a weighted average of the various confirmations, although this was shown by Valvani et al.

(1976) not to be necessary for alphatic alcohols and various hydrocarbons. Hermann found for single conformation hydrocarbon compounds that b = 0.033 kcal mole^{-1} Å^2 at 25°C, while b = 0.030 kcal mol^{-1} Å^2 for alkylbenzenes.

Amidon et al. (1975) correlated the aqueous solubilities and molecular surface areas for the following classes of monofunctional organic nonelectrolytes: hydrocarbons, alcohols, ethers, ketones/aldehydes, esters, carboxylic acids, and olefins. Molecular surface areas were calculated by Hermann's procedure with a solvent radius of 1.5 Å. Amidon et al. (1975) summarized that the use of the surface area approach to explain solubility results from consideration of the following steps in transfer of a solute from pure liquid to aqueous solution: removal of solute from its pure liquid, creation of a cavity in water, and placement of the solute into the cavity. These sequence of steps give the result at equilibrium that the logarithm of the mole fraction solubility is proportional to the total surface area in the organic solute. The total surface area may be divided into hydrocarbonaceous (HYSA) and functional group surface area (FGSA), and the effects of these contributions to total surface area may be determined by regression analysis assuming that the hydrocarbonaceous and functional group portions of surface area contribute independently to solubility. Amidon et al. (1975) showed for the different classes of aliphatic hydrocarbons that the logarithm of organic compound aqueous molal solubility could be correlated almost equally well with HYSA and FGSA, as with total surface area. The effects of the functional group (except for olefins) seemed to make the same contribution to solubility for the case of pure liquid being chosen as the standard state. The sign on the regression equations which employed an FGSA term indicated that among any class of monofunctional aliphatic compounds, decreasing the functional group surface area increased compound aqueous solubility.

Amidon et al. (1975) showed that the relationship between the logarithm of the aqueous solubility and molecular surface area for all 227 solutes being considered (alkanes, alcohols, ethers, ketones, aldehydes, esters and carboxylic acids) was 0.988, which was almost as high as the average of the correlation coefficients obtained from separate regression equations for each class of solutes. This suggested to Amidon et al. (1975) and Pearlman

(1986) that the aqueous solubility of the organic solutes depends almost entirely on molecular surface area and was seemingly independent of the chemical nature of the solute. However, it must be recognized that this conclusion was based on monofunctional substituted compounds. The conclusions from the discussions of Amidon et al. (1975) and Pearlman (1986) are not necessarily broadly applicable to a variety of solute types; also, these discussions are somewhat inconsistent with solubility theories which attempt to account for interactions between polar and nonpolar entities.

In subsequent work, Yalkowsky and Valvani (1979) showed relationships between molecular surface area and aqueous solubility and octanol/water partition coefficient for rigid aromatic hydrocarbons. The surface areas were computed with zero solvent radius and using an extended atom approach for methyl and methylene groups. For thirty-two components having melting points equal to or greater than 25°C, the logarithm of the molar solubility was related to TSA and melting point as

$$\log S_w = \alpha(\text{TSA}) + \beta(\text{mp}) + \delta \qquad (20)$$

The relationship was derived from the recognition that the molar solubility for poorly soluble solutes is proportional to the mole fraction solubility, which is inversely proportional to the activity coefficient with a temperature dependent crystal energy term. The agreement between calculated surface areas and solubilities was judged to reflect the concept that the molecular interactions were determined by the molecular area of contact. It was not necessary to specifically correct for structural features such as branching and proximity effects because it was judged that these effects were reflected in the surface area calculation and manifested as the amount of contact between the hydrocarbon molecule and the aqueous solution. Yalkowsky and Amidon (1979) also showed that TSA was linearly correlated with the logarithm of calculated values of octanol-water partition coefficient for rigid aromatic solutes.

Pearlman (1980, 1986) observed for typical hydrocarbons, including normal and branched alkanes and alkyl substituted aromatics, that molecular surface area and total molecular volume are linearly related. The linear relationship would not be expected for a series of essentially spherical molecules nor for globular entities such as proteins in which case

molecular surface area and volume would vary as molecular radius to the two-thirds power.

Pearlman (1986) explains for partitioning that Amidon's et al. (1975) observation that, the correlation of molecular surface area with aqueous solubility for all solutes combined was almost as high as the average of correlation coefficients obtained from separate regressions for each type of solute, may lead to the supposition that the free energy of solute-solvent interaction is essentially the same for all solvents. However, Pearlman suggests that as solvents (and presumably solutes) become increasingly different, differences in solute-solvent interaction energy are expected. Nonetheless, a weak solute-solvent interaction in the enthalpic sense (i.e. weak "bonds"), is more favored entropically (i.e. solvent molecules near the solute cavity surface are more asymmetric and less structured than when solute-solvent interactions are stronger). Hence, while the enthalpy of solute-solvent interaction may differ between solutes and solvents, the free energy of solute-solvent interaction will differ to a lesser extent. This type of argument is employed by Pearlman (1986) to explain why a single parameter equation involving either the total surface area or the total volume of the solute provides an adequate correlation for partitioning for a variety of solutes. An analogus argument also may hold for aqueous solubility, whereas although some functional groups can "interact more strongly with water than *others, they also interact with themselves more strongly, with the net difference being* nearly the same (Amidon et al., 1975; Pearlman, 1986). As a result of these explanations, and the success of single-parameter surface area regression equations, it was concluded by Pearlman (1986) for the case of 64 alkyl- and halo-substituted aromatics that correlation of aqueous solubility with TSA was satisfactory, while an equation which explicitly accounted for group-dependent differences in solute-solvent interactions was not particularly advantageous.

Effect of Organic Solvent on Sorption of Aromatic Solutes onto Soil

The following describes the mathematical formulations that are employed in AROSOL to describe the effect of miscible organic solvent in water on sorption of aromatic solutes onto soil. The theoretical development for these formulations have been presented in Fu and Luthy (1986b).

The role of soil organic matter in the sorption of uncharged organic solutes has been studied extensively, and it has been found that the organic matter in soil/sediments is primarily responsible for sorption (Karickhoff, 1981). In studies of hydrophobic organic solutes at low loadings, a linear correlation is often observed between solute partition coefficient, K_p (L/kg), and soil/sediment organic carbon content. The dependence of the linear partition coefficient on organic carbon content can be expressed as

$$K_p = K_{oc}\ OC \tag{21}$$

where OC is the fraction organic carbon content; and K_{oc} is the normalized organic carbon partition coefficient. Hamaker and Thompson (1972) suggested that K_{oc} is highly soil or sediment independent and is constant for a particular organic solute. A similar conclusion is made by Karickhoff (1984) in his review of sorption of uncharged organic solutes of limited aqueous solubility ($< 10^{-3}$ m/L) that are not susceptible to special interactions with soil organic carbon.

Karickhoff (1984), following Mackay (1979), explains that sorption equilibrium may be defined as the state in which sorbate fugacities are the same in the aqueous and sorbed phases. For systems in which sorption to organic matter dominates over sorption to mineral matter, the organic carbon normalized-partition coefficient may be envisioned as being proportional to the ratio of the compound's activity coefficient in the aqueous phase, γ_w, and in the organic phase, γ_{oc}, i.e.

$$K_{oc} \propto \frac{\gamma_w}{\gamma_{oc}} \tag{22}$$

In Eq. 22 the proportionality constant contains the reference state fugacities and appropriate unit conversion factors. The activity coefficients "contrast" interactions of the solute in a given phase with the cohesive interactions in the reference state (i.e., pure liquid or subcooled liquid). Thus it may be expected for relatively hydrophobic organic solutes that γ_w would be highly variable, as are variations in aqueous solubility, while γ_{oc}, reflecting cohesive interactions in the organic carbon phase, should be similar to that for the reference state and therefore much less variable. Hence, K_{oc} should be dominated by variations in γ_w, and as a first approximation

$$K_{oc} \propto \gamma_w \tag{23}$$

Partition Coefficients

The octanol/water partition coefficient, K_{ow}, like K_{oc}, describes the partitioning of a solute between an aqueous phase and a relatively immiscible hydrophobic phase. For a solute in equilibrium with octanol and water, the fugacity is the same in each phase, and K_{ow} is given as the ratio of the mass concentration in each phase. Thus,

$$K_{ow} = \frac{C_{oct}}{C_w} = \beta \frac{\gamma_w}{\gamma_{oct}} \tag{24}$$

in which C_{oct} = solute concentration in octanol; C_w = solute concentration in water; and γ_{oct} = solute activity coefficient in octanol. In Eq. 24 the proportionality constant, β, is a conversion factor which entails the ratio of the molar volumes of water and octanol. The same standard state is chosen for the solute in each phase (i.e., pure liquid or subcooled liquid).

Since K_{oc} and K_{ow} both describe organic solute partitioning between water and a hydrophobic organic phase, it may be expected that these parameters would be related

$$K_{ow} \propto K_{oc} \tag{25}$$

The concept embodied in Eq. 25 entails correlation of a partition coefficient for a given system with that for a reference solvent/water system. This has been termed a linear free energy relationship (Leo et al., 1971). Linear free energy correlations with K_{ow} have been used to describe aqueous solubility (Chiou et al., 1982), bioaccumulation (Chiou et al., 1977; Neely and MacKay, 1982), and sorption of organics onto soils (Dzombak and Luthy, 1984; Lambert, 1967).

Various investigations have developed empirical expressions to describe the relationship between K_p and K_{ow}. These investigations have reported excellent correlation between K_{oc} and K_{ow} for hydrophobic solute sorption, with a linear regression equation usually given in the form

$$\log K_{oc} = a \log K_{ow} + b \quad (26)$$

where a and b are regression coefficients. Karickhoff (1984) concluded from these results that the correlation between K_{oc} and K_{ow} was "a somewhat divergent group of relationships." This was attributed to various factors including hydrophilic contribution to sorption, as well as kinetic or steric effects.

K_{oc} and Solute Solubility

Organic solute solubility in water can be related to the solute's activity coefficient, γ, as explained previously. For hydrophobic solutes that are liquid at ambient temperature and which have sufficiently large values of γ, mole fraction solute solubility, X, can be expressed as the reciprocal of the activity coefficient. For hydrophobic solutes that are solid at ambient temperature, a crystal energy term must be taken into account, and if γ is sufficiently large, the mole fraction solute solubility can be expressed as in Eq. 3. Entropy of fusion, ΔS_f, may be incorporated into Eq. 3 for heat of fusion, then the relationship between solid solute activity and mole fraction solubility can be expressed as

$$\log \gamma = -\log X - \frac{\Delta S_f (T_m - T)}{2.303\ RT} \quad (27)$$

For hydrophobic liquid solutes, the system temperature is greater than the melting temperature, and T is set equal to T_m and the crystal term vanishes. Yalkowsky and Valvani (1980) have reviewed ΔS_f data for "rigid" organic solutes which are solids at 25°C, and found that ΔS_f is not highly variable and is in the range of 12–15 cal/mol–°K. "Rigid" solutes included cyclic compounds (aliphatic or aromatic) and molecules with less than five atoms in a flexible chain (Yalkowsky and Valvani, 1980). It is recognized by chemical thermodynamicists that the value of ΔS_f is in the range of 13 cal/mole–°K for solid organic compounds (Prausnitz, 1969), and this average value of ΔS_f was employed in this investigation.

Eq. 23 suggests that K_{oc} should be proportional to the solute activity coefficient, γ. Hence by combining Eqs. 23 and 27, an empirical equation of the following type may be expected to fit observed sorption data

$$\log K_{oc} = -\alpha \log X - \frac{\Delta S_f}{2.303\ RT}(T_m - T) + \beta \qquad (28)$$

in which α and β are regression-fitted parameters. Karickhoff (1981) performed an evaluation of Eq. 28 for condensed ring aromatic compounds using K_{oc} data of Hassett et al. (1980) for benzene and polycyclic aromatic hydrocarbon (PAH) compounds, with ΔS_f assumed to be 13.0 cal/mole-°K and system temperature T at 298 °K (25 °C). The empirical equation was given as

$$\log K_{oc} = 0.921 \log X - 0.00953 (T_m - 298) - 1.405 \qquad (29)$$

This equation was evaluated for other families of hydrophobic organic solutes (triazines, carbamates, organophosphates, and chlorinated hydrocarbons), and was found to estimate K_{oc} usually within a factor of 2 to 3 of measured values (Karickhoff, 1984). It was found that Eq. 29 worked well for low molecular weight compounds but tended to overestimate sorption of highly chlorinated, high molecular weight compounds. It was concluded that α values for these type of compounds may be in the range of 0.7-0.8 which is considerably less than that for polycyclic aromatic hydrocarbons. The sorption literature values for 47 organic compounds gave an α value of 0.83 and a β value of -0.93 (Karickhoff, 1984).

Solvent Effect on Solute Sorption

The theory behind Eqs. 21-25 and Eq. 28 is summarized by Fu and Luthy (1986b) as follows. Linear partition coefficients are often observed for hydrophobic organic solute sorption onto soil or sediments. The sorption partition coefficient may be normalized for soils and sediments on the basis of fraction organic carbon, and normalized for various solutes on the basis of octanol/water partition coefficient or aqueous solubility. Hence for a given soil or sediment, organic solute sorption is inversely proportional to aqueous solubility. The following explains a theoretical approach for predicting the observed effect of a miscible organic solvent in the aqueous phase on organic solute sorption onto soil or sediment. This approach is based on the linkage between K_{oc} and aqueous solubility, and the effect of solvent on solubility.

It has been demonstrated in Fu and Luthy (1985, 1986a) that aromatic solute solubility in a

solvent/water mixture generally increases semi-logarithmically with increase of volume fraction of solvent. Using a simple log-linear solubility model (1986a), the mole fraction solubility of the solute in the solvent/water mixture can be expressed as

$$\log X_m = \log X_w + \sigma z \tag{30}$$

where X is the mole fraction solubility of the solute, and subscripts m and w represent solvent/water mixture and water, respectively; z is the volume fraction of solvent; and σ is the regression constant obtained from experimental mole fraction solubility data.

By combining Eqs. 28 and 30, the solute sorption coefficient K_{oc} can be expressed in terms of solute solubility

$$\log K_{oc} = -a \log X_w - a\sigma z - \frac{\Delta S_f}{2.303\ RT}(T_m - T) + \beta \tag{31}$$

Rearranging Eq. 31 gives

$$\log K_{oc} = \left[-a \log X_w - \frac{\Delta S_f}{2.303\ RT}(T_m - T) + \beta \right] - a\sigma z \tag{32}$$

The terms within the brackets on the right-hand side of Eq. 32 represent the solute K_{oc} in pure water, denoted as $K_{oc,w}$, therefore

$$\log K_{oc} = \log K_{oc,w} - a\sigma z \tag{33}$$

Substituting into Eq. 21 gives

$$\log \frac{K_{oc}}{K_{oc,w}} = \log \frac{K_p}{K_{p,w}} = -a\sigma z \tag{34}$$

in which $K_{p,w}$ = the partition coefficient for solvent-free water.

It is important to recognize that the basis of derivation of Eq. 34 depended on the relationship between mole fraction solubility and K_{oc}. The units in K_{oc} must therefore be consistent with the expression of solubility in terms of mole fraction. This is not strictly necessary for the case of comparison of different values of log $K_{oc,w}$, as in Eqs. 28 and 29, where the partition coefficient may be employed with customary units of L/kg. In this case the number of moles per liter is constant for dilute aqueous systems, i.e., 55.34 moles/L, and this value becomes incorporated in the regression constant β in Eq. 29.

However, for the case of solvent/water mixtures, the total number of moles per liter is not constant and the partition coefficient must be expressed in units of mole/kg. Thus, in order to use Eq. 34, the ratio of the experimentally determined sorption coefficients must be expressed on the basis of total moles per unit mass of soil

$$\log \left[K_p \left(\frac{V_{water}}{q_{water}} + \frac{V_{solvent}}{q_{solvent}} \right) / K_{p,w} (55.34) \right] = - \alpha \sigma z \qquad (35)$$

where V refers to the solute-free volume of water or solvent in the mixture, and q represents the molar volume of water or solvent.

Eq. 35 indicates that K_p for a solvent/water mixture decreases semi-logarithmically with the increase of solvent volume fraction. The semi-logarithmic relationship predicted by Eq. 35 can be shown on a semi-logarithmic plot with volume fraction solvent on the abscissa and K_p (mole/kg) on the ordinate. The slope of this plot represents the combined effect of both α and σ. The σ term represents the effect of solvent on increase of solute solubility, while the α term relates to the dominance of γ_w in K_{oc} among various solutes. The α term should approach unity if the fugacity coefficient for solute in soil/sediment organic carbon is relatively independent of solute (Karickhoff, 1984), and if the soil organic carbon properties are independent of change in solution phase composition.

Soil sorption partition coefficient data were presented by Fu and Luthy (1986b) for various lower molecular weight, aromatic solutes in solvent/water systems. From these data it was possible to determine experimental values of the solvent volume fraction-coefficient in Eq. 35, $(\alpha\sigma)_{exp}$. The values of σ were known from data presented in Fu and Luthy (1985, 1986a), thus it was possible to determine an explicit observed value for α, i.e. α_{obs}. These results indicate that if α_{obs} is significantly less than unity, then the effect of solvent on sorption partition coefficient is not as significant as the effect of solvent on solute solubility. If α_{obs} approaches unity, then the solvent effect on sorption partition coefficient is inversely proportional to the solvent effect on solute solubility.

Partition Coefficient in Solvent Water Mixtures

The effect of solvent on organic solute sorption partitioning was examined by Fu and Luthy (1985, 1986a, 1986b) using Eq. 35 and experimental solubility and sorption data. Eq. 35 was examined using experimental sorption data for seven systems in conjunction with the respective σ values. The observed α values showed that the α_{obs} values were in the range of 0.41–0.63. The average α_{obs} value was 0.51 with a standard deviation of 0.060.

The range of observed α values was significantly less than unity, as well as uniformly less than a value of 0.92 as found by Karickhoff (1984) for condensed ring aromatic solute sorption onto soil/sediment for solvent–free systems. This implies, for a given solvent/water mixture, that the decrease of the sorption partition coefficient cannot be attributed soley to the increase in solute solubility. Further, the range of α_{obs} values suggests that the logarithmic decrease in K_p is approximately half the logarithmic increase of solute solubility in a solvent/water mixture. This is believed to be a result of the solvent increasing the accessibility of the solute to the organic carbon (Fu and Luthy, 1986b). These observations hold for lower molecular weight aromatic solutes. Additional sorption data are required in order to learn if the value of α_{obs} may increase to unity, or vary in some other regular manner, for solute systems ranging from moderately insoluble to very insoluble.

The AROSOL computer program estimates the effect of organic solvent on the soil partition coefficient by use of Eq. 35 with an assumed average value of α = 0.51. This computation gives the ratio, $K_p/K_{p,w}$. $K_{oc,w}$ is then determined by use of Eq. 29, and this value is then multiplied by a specific value of OC to give $K_{p,w}$. The estimate of $K_{p,w}$ is then used with the ratio $K_p/K_{p,w}$ to determine K_p for the miscible solvent–water system. Note in the estimation of K_p for solvent–water systems that the parameter σ is estimated by the log–linear approach.

In the event that the log–linear approach is not selected for estimation of the solute solubility relationship, the AROSOL program employs the UNIFAC approach for estimation of the parameter σ. This estimated parameter is then used with Eq. 35 and 29 to predict K_p for the solvent–water system.

Chapter Three

DESCRIPTION OF THE COMPUTATIONAL PROGRAMS AND EXAMPLE CALCULATIONS

This chapter discusses the general features of the computer programs, named AROSOL and MOLACCS. The program AROSOL is designed to aid prediction of aromatic solute solubility in water, and in miscible organic solvent-water mixtures, and to estimate the effect of miscible organic solvent in water on sorption of aromatic solutes on to soils/sediments. The program MOLACCS computes MOLecular ACCessible Surface area. A molecular surface area approach is one technique by which organic solute aqueous solubility and solubility in miscible solvent/water mixtures may be predicted, and the MOLACCS program provides surface area parameters for this approach. The theoretical approaches and the computational methodologies employed in AROSOL and MOLACCS have been described in the previous chapter. The use of the programs is described in turn below.

AROSOL Program Organization

Figure 4 is an outline of the computational methodology employed in the AROSOL program. The program utilizes four techniques to predict solubility: (i) log linear, (ii) UNIFAC, *(iii) excess free energy, and (iv) molecular surface area.* The user may select any or all of these techniques to evaluate solubility depending on the availability of data, physical constants, and other specific information required for each approach. The program was developed to accommodate a variety of input parameters to predict solubility. The program consists of the following subroutines:

(a) INPUT: Input data are read from this subroutine

(b) SETUP: Reads input data from the console and stores input into a data file

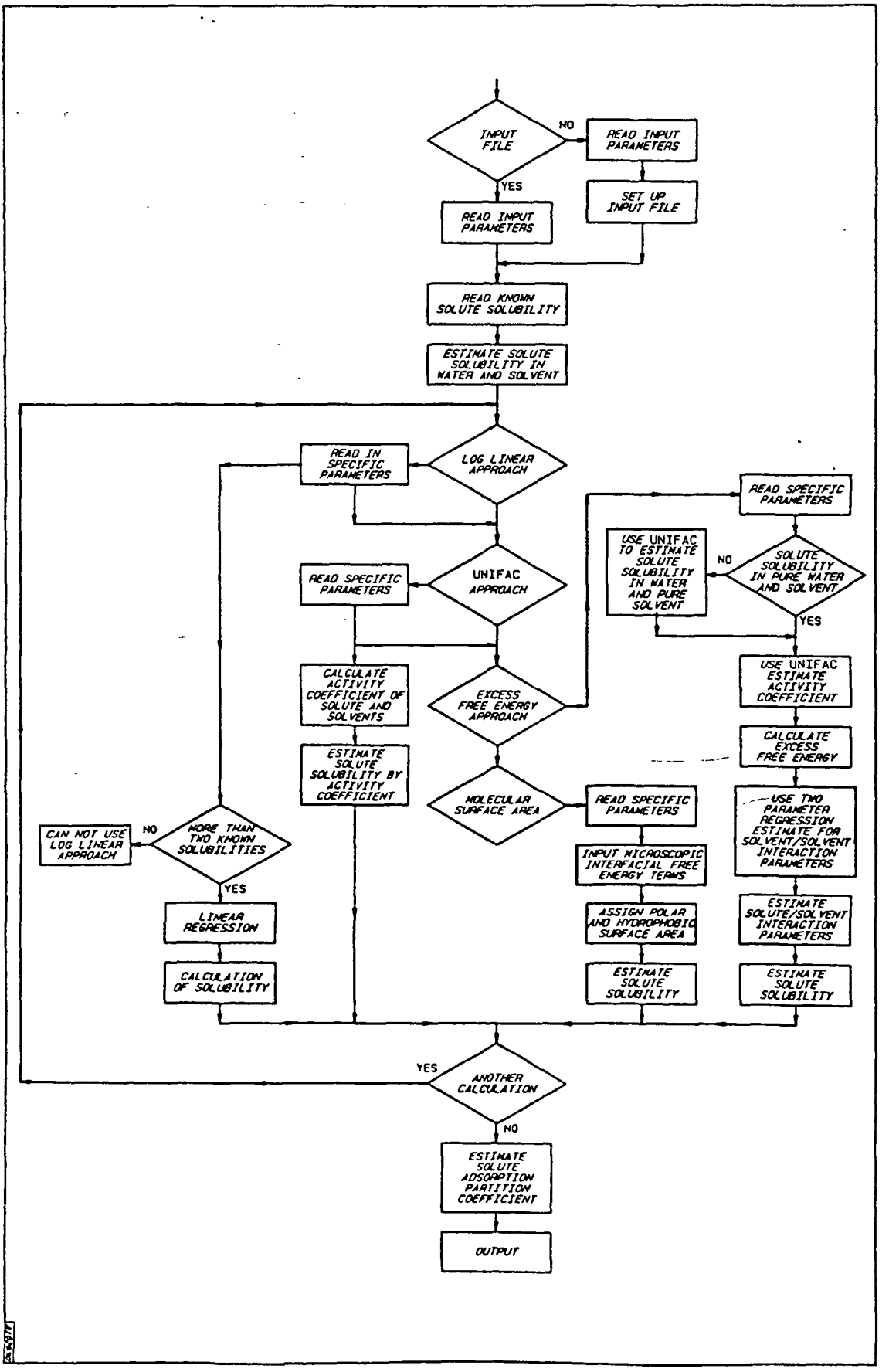
(c) UNIFAC: Calculates activity coefficients for each component

(d) REG1: Linear least-square regression for the log-linear procedure

(e) REG2: Two parameter least-square regression for the excess free energy procedure

(f) SOLCAL: Calculates solute solubility

Figure 4. AROSOL Program Structure



(g) LOGNR: Log-linear approach subroutine calculations

(h) UNIEST: Numerical estimation technique for estimating solute solubility mole fraction from activity coefficient

(i) MSA: Molecular surface area approach calculations

(j) EXFREN: Excess free energy approach calculations

(k) ADS: Calculates solute sorption partition coefficient

The AROSOL computer program is written in FORTRAN-77, and it can be run on any IBM or IBM-compatible personal computer with at least 256 K internal memory. A listing of the program is presented in Appendix A.

The following examples illustrate the use of the program. These examples illustrate the calculation approach employed in the log-linear, UNIFAC, excess free energy, and molecular surface area approaches.

Estimation of Solubility and Sorption Partition Coefficient

Example Calculation I: No Input File
Quinoline in Methanol-Water

Figure 5 shows an example of calculation of quinoline solubility in methanol-water mixtures by the log-linear, UNIFAC, and excess free energy approaches. In this example the user inputs data from the terminal.

The program begins with a RETURN key stroke. The user is asked if there is an input file, for which in this example the response is "no". The user enters a response that an input file is to be established, and in this example the input file is named Q.I (for quinoline input). The user is then asked to type in the name of the solvent (methanol, component 1) and solute (quinoline, component 2), using twenty characters or less for each name. The user is then asked to input the molecular weight of solvent, solute, and water. The format for input is free format, in which a space or comma between the data entry identifies the

appropriate molecular weight in the order solvent, solute, and water. The user is then asked to input the densities of the components in the order solvent, solute, and water using the same free input format.

The user is then asked to input known solubility for the solute in the solvent. The data are input in the order per cent by volume solvent, followed by solubility in mg/L in the mixture, using the free format with a separate line being used for each data pair. The last data entry is followed by the input of −1 −1 on a separate line to indicate completion of the data entry file.

The user now specifies which of the four calculation approaches are to be employed. Each of the four approaches are employed in this example.

The user is then asked if K_p for the mixed solvent system is to be calculated. In this example the response is "yes;" and the user will be asked later to input the fraction organic carbon content of the soil or sediment. As explained later, if the organic carbon content of the soil or sediment is unknown, the program computes K_{oc} by inputting OC = 100%.

The user now inputs solute heat of fusion in cal/mole, solute melting temperature (K), and system temperature (K) using the free format. If solute heat of fusion is not known, then the user is to enter zero (0) as the value, in which case the fugacity ratio expression will be computed by Eq. 6.

The user is now asked if it is desired to see the secondary group listing for the UNIFAC calculations. The listing is requested in this example, and the 89 secondary groups are displayed. The user is asked how many secondary groups appear in the solvent component, in this example there is only one solvent secondary group. Similarly the user is asked how many secondary groups appear in the solute, in this example the solute may be constructed from two subgroups. The user now enters the number of times each secondary group appears in the solvent, and the identification number of the secondary group. In this example the solvent is comprised of one secondary group, number 16

Figure 5. Example Calculation I: No Input File-Quinoline Solubility in Methanol/Water

```
C:\JKF >
C:\JKF >arosol
          ****************************************

          *                AROSOL                *
          *     Aromatic Solute Solubility       *
          *     in Solvent/Water Mixtures        *
          *                                      *
          *              by                      *
          *          Jaw-Kwei Fu                 *
          *          Charles Brooks              *
          *          Richard G. Luthy            *
          *     Carnegie-Mellon University       *
          *     Pittsburgh, Pennsylvania         *
          *                                      *
          *          August, 1986                *

          ****************************************

Hit RETURN key to continue

 This program estimates aromatic solute solubility in
 solvent/water mixtures.  This program is designed to
 utilize different levels of input parameters to estimate
 aromatic solute solubility. This program employes four
 approaches: LOG-LINEAR, UNIFAC, EXCESS FREE ENERGY, and
 MOLECULAR SURFACE AREA. The input parameters can be read
 from either an existing input file or from the terminal.
 The program estimates solute solubility via approaches
 specified by the user, and stores the results into an
 output file.


 Hit RETURN key to continue

DO YOU HAVE INPUT FILE? (Y OR N)
n
DO YOU WANT TO SET UP AN INPUT FILE? (Y OR N)
y
GIVE THE FILE NAME IN WHICH INPUT DATA ARE TO BE STORED
q.i
INPUT THE NAME OF COMPONENT  1 IN 20 CHARACTERS 1-SOLVENT, 2-SOLUTE
METHANOL
INPUT THE NAME OF COMPONENT  2 IN 20 CHARACTERS 1-SOLVENT, 2-SOLUTE
QUINOLINE
INPUT MOLECULAR WEIGHT OF SOLVENT, SOLUTE, WATER
32.04 129.16 18.02
INPUT DENSITIES OF SOLVENT, SOLUTE AND WATER
.7914 1.0929 .9971
INPUT KNOWN SOLUTE SOLUBILITY IN % SOLVENT, AND SOLUTE SOLUBILITY IN MG/L,
FINISHED AS -1 -1,  DATA INPUT IN PAIRS WITH ONE PAIR PER LINE
0 6832
10 14603
20 34048
30 75358
40 125493
50 251189
-1 -1
DO YOU WANT TO ESTIMATE SOLUTE SOLUBILITY BY  THE LOG-LINEAR APPROACH?
 (Y OR N)
y
DO YOU WANT TO ESTIMATE SOLUTE SOLUBILITY BY THE UNIFAC APPROACH? (Y OR N)
y
DO YOU WANT TO ESTIMATE SOLUTE SOLUBILITY BY THE EXCESS FREE ENERGY APPROACH?
(Y OR N)
y
DO YOU WANT TO ESTIMATE SOLUTE SOLUBILITY BY
THE MOLECULAR SURFACE AREA APPROACH? (Y OR N)
y
DO YOU WANT TO ESTIMATE ADSORPTION PARTITION COEFFICIENT? (Y OR N)
y
ENTER SOLUTE HEAT OF FUSION (CAL/MOLE), SOLUTE MELTING TEMPERATURE (K),  AND
SYSTEM TEMPERATURE (K), IF HEAT OF FUSION IS NOT AVAILABLE USE 0 AS THE VALUE
3751.8 288.6 298
DO YOU WANT TO SEE THE UNIFAC SECONDARY GROUP LISTING?(Y OR N)
y
```

Figure 5. Example Calculation I (continued)

```
 1 CH3            2 CH2            3 CH             4 C              5 CH2=CH
 6 CH=CH          7 CH2=C          8 CH=C           9 C=C           10 ACH.
11 AC            12 ACCH3         13 ACCH2         14 ACCH          15 OH
16 CH3OH         17 H2O           18 ACOH          19 CH3CO         20 CH2CO
21 CHO           22 CH3COO        23 CH2COO        24 HCOO          25 CH3O
26 CH2O          27 CH-O          28 FCH2O         29 CH3NH2        30 CH2NH2
31 CHNH2         32 CH3NH         33 CH2NH         34 CHNH          35 CH3N
36 CH2N          37 ACNH2         38 C5H5N         39 C5H4N         40 C5H3N
41 CH3CN         42 CH2CN         43 COOH          44 HCOOH         45 CH2CL
46 CHCL          47 CCL           48 CH2CL2        49 CHCL2         50 CCL2
51 CHCL3         52 CCL3          53 CCL4          54 ACCL          55 CH3NO2
56 CH2NO2        57 CHNO2         58 ACNO2         59 CS2           60 CH3SH
61 CH2SH         62 FURFURAL      63 (CH2OH)2      64 I             65 BR
66 CH-TRIP-C     67 C-TRIP-C      68 ME2SO         69 ACRY          70 CL(C=C)
71 ACF           72 DMF-1         73 DMF-2         74 CF3           75 CF2
76 CF            77 COO           78 SIH3          79 SIH2          80 SIH
81 SI            82 SIH2O         83 SIHO          84 SIO           85 TERT-N
86 AMIDE         87 CON(ME)2      88 CONMECH2      89 CON(CH2)2
 INPUT THE NUMBER OF SECONDARY GROUPS IN COMPONENT METHANOL
1
 INPUT THE NUMBER OF SECONDARY GROUPS IN COMPONENT QUINOLINE
2
 INPUT NUMBER OF TIMES SECONDARY GROUP i APPEARS IN COMPONENT  METHANOL
 REPEAT n TIMES UNTIL n = NUMBER OF SECONDARY GROUPS
1 16
 INPUT NUMBER OF TIMES SECONDARY GROUP i APPEARS IN COMPONENT  QUINOLINE
 REPEAT n TIMES UNTIL n = NUMBER OF SECONDARY GROUPS
4 10 1 40
 INPUT LOG OCTANOL/WATER PARTITION COEFFICIENT
 OF SOLUTE FOR THE EXCESS FREE ENERGY CALCULATION
2.04
INPUT SOLUTE HYDROPHOBIC SURFACE AREA AND POLAR SURFACE AREA.
INPUT IN UNITS A**2
142.877 9.078
INPUT SOLVENT MICROSCOPIC INTERFACIAL FREE ENERGY BETWEEN HSA AND PSA,
INPUT IN UNITS OF DYNE/CM**2
24.6 47.7
INPUT ORGANIC CARBON CONTENT OF ADSORBENT, IN %
2
GIVE THE OUTPUT FILE NAME IN WHICH THE SOLUBILITY
CALCULATION DATA WILL BE STORED
q.o
INPUT PERCENT VOLUME SOLVENT IN THE MIXTURE TO BE EVALUATED
0
LOG-LINEAR REGRESSION INTERCEPT    -3.0002 SLOPE      .03573
   LOG-LINEAR ESTIMATION METHOD

        SOLVENT FRACTION [% VOL]          .00
        LOG LINEAR ESTIMATION SOLUBILITY [MOLE FRACTION]    .100E-02
        LOG LINEAR ESTIMATION SOLUBILITY [MG/L]    .710E+04

  UNIFAC ACTIVITY COEFFICIENT ESTIMATION
    COMPONENT          MOLE FRAC  LN ACTCF        ACTCF

   QUINOLINE                .0000     7.51707      1839.1690
   METHANOL                 .0000      .00000         1.0000


   UNIFAC ESTIMATION
        METHANOL            FRACTION [% VOL]          .00
        QUINOLINE           SOLUBILITY [MOLE FRACTION]    .544E-03
        QUINOLINE           SOLUBILITY [MG/L]    .387E+04
 THE SOLVENT-SOLVENT INTERACTION PARAMETERS ARE  .7644  AND .4566

   EXCESS FREE ENERGY ESTIMATION METHOD
        SOLVENT FRACTION [% VOL]          .00
        EXCESS FREE ENERGY ESTIMATION SOLUBILITY [MOLE FRACTION]    .961E-03
        EXCESS FREE ENERGY ESTIMATION SOLUBILITY [MG/L]       .683E+04
   MOLECULAR SURFACE AREA APPROACH

        SOLVENT FRACTION [% VOL]          .00
        MOLECULAR SURFACE AREA APPROACH SOLUBILITY [MOLE FRACTION]    .961E-03
        MOLECULAR SURFACE AREA APPROACH SOLUBILITY [MG/L]    .683E+04
```

Figure 5. Example Calculation I (continued)

```
 ADSORPTION COEFFICIENT OF QUINOLINE            IN WATER/METHANOL
MIXTURES IS    .515E+02
ANOTHER CALCULATION? (Y=1,N=2)
1
INPUT PERCENT VOLUME SOLVENT IN THE MIXTURE TO BE EVALUATED
20
   LOG-LINEAR ESTIMATION METHOD

        SOLVENT FRACTION [% VOL]       20.00
        LOG LINEAR ESTIMATION SOLUBILITY [MOLE FRACTION]    .518E-02
        LOG LINEAR ESTIMATION SOLUBILITY [MG/L]    .321E+05

  UNIFAC ACTIVITY COEFFICIENT ESTIMATION
    COMPONENT          MOLE FRAC  LN ACTCF        ACTCF

   QUINOLINE                  .0000     5.84863       346.7577
   METHANOL                   .1004      .00000         1.0000


   UNIFAC ESTIMATION
        METHANOL            FRACTION [% VOL]       20.00
        QUINOLINE           SOLUBILITY [MOLE FRACTION]    .309E-02
        QUINOLINE           SOLUBILITY [MG/L]    .194E+05

   EXCESS FREE ENERGY ESTIMATION METHOD
        SOLVENT FRACTION [% VOL]       20.00
        EXCESS FREE ENERGY ESTIMATION SOLUBILITY [MOLE FRACTION]    .253E-02
        EXCESS FREE ENERGY ESTIMATION SOLUBILITY [MG/L]       .159E+05
   MOLECULAR SURFACE AREA APPROACH

        SOLVENT FRACTION [% VOL]       20.00
        MOLECULAR SURFACE AREA APPROACH SOLUBILITY [MOLE FRACTION]    .655E-02
        MOLECULAR SURFACE AREA APPROACH SOLUBILITY [MG/L]    .404E+05


 ADSORPTION COEFFICIENT OF QUINOLINE            IN WATER/METHANOL
MIXTURES IS    .223E+02
ANOTHER CALCULATION? (Y=1,N=2)
2
Stop - Program terminated.

C:\JKF >
```

Figure 5. Example Calculation I (continued)

```
C:\JKF >
C:\JKF >type q.i
METHANOL
QUINOLINE
          32.0400000        129.1600000         18.0200000
   7.914000E-001          1.0929000  9.971000E-001
           .0000000       6832.0000000
         10.0000000      14603.0000000
         20.0000000      34048.0000000
         30.0000000      75358.0000000
         40.0000000     125493.0000000
         50.0000000     251189.0000000
         -1.0000000         -1.0000000
y
y
y
y
y
       3751.8000000        288.6000000         298.0000000
             1
             2
          1.0000000            16
          4.0000000            10          1.0000000            40
          2.0400000
        142.8770000          9.0780000
         24.6000000         47.7000000
          2.0000000

C:\JKF >


C:\JKF >
C:\JKF >type q.o

 ********************************************************************************

 SOLUTE USED IN THE CALCULATION IS     QUINOLINE
 SOLVENT USED IN THE CALCULATION IS    METHANOL

 PARAMETER           METHANOL          QUINOLINE                 WATER
 MOLECULAR WEIGHT       32.04              129.16                 18.02
 DENSITY                .7914              1.0929                 .9971
 MOLAR VOLUME           40.49              118.18                 18.07

 OCTANOL/WATER PARTITION COEFFICIENT  .110E+03

 ********************************************************************************

 SOLUTE SOLUBILITY PREDICTIONS IN MIXED SOLVENT SYSTEM

 VOLUME   LOG-LINEAR        UNIFAC        EXCESS FREE     MOLECULAR SURFACE
 SOLVENT  APPROACH          APPROACH      ENERGY APPROACH AREA APPROACH
        MOLE     SOL.     MOLE     SOL.     MOLE     SOL.     MOLE     SOL.
    %     [-]  [MG/L]      [-]  [MG/L]      [-]  [MG/L]      [-]  [MG/L]

    .00 .10E-02 .71E+04 .54E-03 .39E+04 .96E-03 .68E+04 .96E-03 .68E+04
  20.00 .52E-02 .32E+05 .31E-02 .19E+05 .25E-02 .16E+05 .66E-02 .40E+05


 ********************************************************************************

  SOLUTE SORPTION PARTITION COEFFICIENT ESTIMATION

 METHANOL              OC OF ADSORBENT          KP OF SOLUTE
    [%]                     [%]                   [MOLE/KG]
        .00                 2.00                      51.53
      20.00                 2.00                      22.26

C:\JKF >
```

(methanol). The user now inputs the number of times each secondary group appears in the solute, and the identification number of each secondary group. This is repeated using free floating format until all the secondary groups have been specified. In this example, quinoline is constructed from a pyridine-type derivative and aromatic -CH groups (ACH). Thus four (4) secondary group ACH, number (10), and one (1) secondary group C5H3N, number (40), comprise the molecule quinoline.

Next the user inputs the logarithm of the solute octanol/water partition coefficient; this value is used in the excess free energy calculation.

The user is then asked to input the solute hydrophobic surface area and the polar surface area in units of Å. The data are input in their respective order using the free format. These data are obtained from the MOLACCS program, as demonstrated in a latter example calculation. The user is then asked to input the solvent microscopic interfacial free energies for the hydrophobic surface area and the polar surface area in units of dynes per cm^2.

The user is then asked to input the percent organic carbon content of the soil or sediment. In this example the calculation is performed for a soil having 2% organic carbon content. If the organic carbon content is unknown, the user should specify a value of 100%, in which case K_p becomes equal to K_{oc}.

The user is then asked to name the output file in which the results of the calculations will be stored. In this example the output file is named Q.O, for quinoline output. The user is then asked to specify a solvent/water volume composition to be evaluated. In this example the user has requested calculation for 0% by volume methanol (i.e. 100% water).

The output for the various calculations are now presented, for which the program computes and displays firstly the slope and the intercept values for the log-linear method; the solvent volume fraction specified is also shown. The log-linear computation method employs Eq. 15, in which the log-linear solute solubility relationship is computed in terms of mole fraction solute in conjunction with volume fraction solvent. The log-linear

solubility output is shown as mole fraction, for which quinoline solubility in pure water is estimated from the input data as 1.0×10^{-3}. The predicted mole fraction solute solubility is then converted by procedures described in Chapter 2 to customary solubility units of mg/L of solution, or 7100 mg/L in this example.

Next the results for the UNIFAC calculation procedure are presented. First, the solute and solvent activity coefficients are shown. The specified volume fraction solvent is shown, followed by the presentation of the solute mole fraction solubility. The solute activity coefficient is related to mole fraction by procedures discussed in Chapter 2, which is then converted and displayed in units of mg/L, i.e. 3870 mg/L in this example.

Next, the results for the excess free energy calculation procedure are presented in units of both mole fraction and mg/L. The regression coefficients for determination of the solvent-water interaction parameters, A_{1-3} and A_{3-1}, are also shown.

The results from the molecular surface area calculation are presented next.

The solute sorption partition coefficient is then calculated by procedures described in Chapter 2.

The user is then asked if another calculation is to be performed. In this example the user inputs a "1" for yes, and specifies 20% by volume methanol for the next calculation.

The results for the new solvent volume fraction calculation are presented in the same order as in the previous discussion. The regression coefficients for the log-linear and excess free energy calculations are not repeated, however. At the conclusion of this calculation the user terminates the calculation by inputing the number "2".

The manner in which data in the input and output files may be listed is shown at the end of the example.

Example calculation II: No Solubility Data
Solubility of Monochlorobenzene in Acetone-Water

In this example the user has no solubility information, hence the only calculation approach which may be employed is the UNIFAC technique. The user is asked to input the solvent, solute, and water properties. This is done according to the procedures described in the previous example. The user does not request that the calculations be performed by the the log-linear, or the excess free energy approach, or the molecular surface area approach because of lack of data. Normally one would desire some data from which to establish a correlation for the log-linear approach, and in the case of the excess free energy approach one would normally desire at least pure solvent solute solubility as an input parameter.

For the case of the UNIFAC approach, the solvent, acetone, is recognized as being comprised of two secondary groups: one $-CH_3$ (secondary group 1) and one CH_3CO- (secondary group 19). The solute is comprised of two secondary groups: Five - aromatic CH (secondary group 10), and one - aromatic C-Cl (secondary group 54). The user requests that the solubility prediction calculation be performed for the case of 0% and 20% by volume solvent. The results are displayed in the same fashion as in the previous example, with the input and output files shown also at the end of the calculation.

Example Calculation III: Read from an Input File,
Naphthalene Solubility in Methanol-Water

In this example the user has an input data file and requests the program to estimate solubility and sorption partition coefficient in 0, 10, and 50% by volume solvent. The results from the example calculation are shown in Figure 7. In this example the input file is shown at the end of the calculation. The necessary solvent, solute, and water parameters are entered, as well as the solubility data. The necessary data are entered according to the format explained in Example I. Note for purposes of the UNIFAC calculation that naphthalene is comprised of eight aromatic-CH secondary groups, and two aromatic-C secondary groups. The calculation proceeds as described previously, and the results are displayed in an output file.

Figure 6. Example Calculation II: No Solubility Data - Monochlorobenze Solubility in Acetone/Water

```
C:\JKF >
C:\JKF >arosol
          ****************************************

          *                AROSOL                *
          *     Aromatic Solute Solubility       *
          *     in Solvent/Water Mixtures        *
          *                                      *
          *                by                    *
          *          Jaw-Kwei Fu                 *
          *          Charles Brooks              *
          *          Richard G. Luthy            *
          *     Carnegie-Mellon University       *
          *     Pittsburgh, Pennsylvania         *
          *                                      *
          *          August, 1986                *

          ****************************************

Hit RETURN key to continue

 This program estimates aromatic solute solubility in
 solvent/water mixtures.  This program is designed to
 utilize different levels of input parameters to estimate
 aromatic solute solubility. This program employes four
 approaches: LOG-LINEAR, UNIFAC, EXCESS FREE ENERGY, and
 MOLECULAR SURFACE AREA. The input parameters can be read
 from either an existing input file or from the terminal.
 The program estimates solute solubility via approaches
 specified by the user, and stores the results into an
 output file.


 Hit RETURN key to continue

DO YOU HAVE INPUT FILE? (Y OR N)
n
DO YOU WANT TO SET UP AN INPUT FILE? (Y OR N)
y
GIVE THE FILE NAME IN WHICH INPUT DATA ARE TO BE STORED
cbace.i
INPUT THE NAME OF COMPONENT  1 IN 20 CHARACTERS 1-SOLVENT, 2-SOLUTE
ACETONE
INPUT THE NAME OF COMPONENT  2 IN 20 CHARACTERS 1-SOLVENT, 2-SOLUTE
1CHLOROBENZENE
INPUT MOLECULAR WEIGHT OF SOLVENT, SOLUTE, WATER
58.08 112.56 18.02
INPUT DENSITIES OF SOLVENT, SOLUTE AND WATER
.7899 .9630 .9971
INPUT KNOWN SOLUTE SOLUBILITY IN % SOLVENT, AND SOLUTE SOLUBILITY IN MG/L,
FINISHED AS -1 -1,  DATA INPUT IN PAIRS WITH ONE PAIR PER LINE
-1 -1
DO YOU WANT TO ESTIMATE SOLUTE SOLUBILITY BY  THE LOG-LINEAR APPROACH?
 (Y OR N)
n
DO YOU WANT TO ESTIMATE SOLUTE SOLUBILITY BY THE UNIFAC APPROACH? (Y OR N)
y
DO YOU WANT TO ESTIMATE SOLUTE SOLUBILITY BY THE EXCESS FREE ENERGY APPROACH?
(Y OR N)
n
DO YOU WANT TO ESTIMATE SOLUTE SOLUBILITY BY
THE MOLECULAR SURFACE AREA APPROACH? (Y OR N)
n
DO YOU WANT TO ESTIMATE ADSORPTION PARTITION COEFFICIENT? (Y OR N)
n
ENTER SOLUTE HEAT OF FUSION (CAL/MOLE), SOLUTE MELTING TEMPERATURE (K),  AND
SYSTEM TEMPERATURE (K), IF HEAT OF FUSION IS NOT AVAILABLE USE 0 AS THE VALUE
0 295.74 298
DO YOU WANT TO SEE THE UNIFAC SECONDARY GROUP LISTING?(Y OR N)
y
```

Figure 6. Example Calculation II (continued)

```
 1 CH3           2 CH2           3 CH            4 C             5 CH2=CH
 6 CH=CH         7 CH2=C         8 CH=C          9 C=C          10 ACH
11 AC           12 ACCH3        13 ACCH2        14 ACCH         15 OH
16 CH3OH        17 H2O          18 ACOH         19 CH3CO        20 CH2CO
21 CHO          22 CH3COO       23 CH2COO       24 HCOO         25 CH3O
26 CH2O         27 CH-O         28 FCH2O        29 CH3NH2       30 CH2NH2
31 CHNH2        32 CH3NH        33 CH2NH        34 CHNH         35 CH3N
36 CH2N         37 ACNH2        38 C5H5N        39 C5H4N        40 C5H3N
41 CH3CN        42 CH2CN        43 COOH         44 HCOOH        45 CH2CL
46 CHCL         47 CCL          48 CH2CL2       49 CHCL2        50 CCL2
51 CHCL3        52 CCL3         53 CCL4         54 ACCL         55 CH3NO2
56 CH2NO2       57 CHNO2        58 ACNO2        59 CS2          60 CH3SH
61 CH2SH        62 FURFURAL     63 (CH2OH)2     64 I            65 BR
66 CH-TRIP-C    67 C-TRIP-C     68 ME2SO        69 ACRY         70 CL(C=C)
71 ACF          72 DMF-1        73 DMF-2        74 CF3          75 CF2
76 CF           77 COO          78 SIH3         79 SIH2         80 SIH
81 SI           82 SIH2O        83 SIHO         84 SIO          85 TERT-N
86 AMIDE        87 CON(ME)2     88 CONMECH2     89 CON(CH2)2
 INPUT THE NUMBER OF SECONDARY GROUPS IN COMPONENT ACETONE
2
 INPUT THE NUMBER OF SECONDARY GROUPS IN COMPONENT 1CHLOROBENZENE
2
 INPUT NUMBER OF TIMES SECONDARY GROUP i APPEARS IN COMPONENT  ACETONE
 REPEAT n TIMES UNTIL n = NUMBER OF SECONDARY GROUPS
1 1 1 19
 INPUT NUMBER OF TIMES SECONDARY GROUP i APPEARS IN COMPONENT  1CHLOROBENZENE
 REPEAT n TIMES UNTIL n = NUMBER OF SECONDARY GROUPS
5 10 1 54
GIVE THE OUTPUT FILE NAME IN WHICH THE SOLUBILITY
CALCULATION DATA WILL BE STORED
cbace.o
INPUT PERCENT VOLUME SOLVENT IN THE MIXTURE TO BE EVALUATED
0

  UNIFAC ACTIVITY COEFFICIENT ESTIMATION
    COMPONENT          MOLE FRAC  LN ACTCF       ACTCF

   1CHLOROBENZENE          .0000    9.86048    19158.0400
   ACETONE                 .0000     .00000        1.0000


   UNIFAC ESTIMATION
        ACETONE            FRACTION [% VOL]        .00
        1CHLOROBENZENE     SOLUBILITY [MOLE FRACTION]   .522E-04
        1CHLOROBENZENE     SOLUBILITY [MG/L]   .325E+03
ANOTHER CALCULATION? (Y=1,N=2)
1
INPUT PERCENT VOLUME SOLVENT IN THE MIXTURE TO BE EVALUATED
20

  UNIFAC ACTIVITY COEFFICIENT ESTIMATION
    COMPONENT          MOLE FRAC  LN ACTCF       ACTCF

   1CHLOROBENZENE          .0000    8.46858     4762.7260
   ACETONE                 .0579     .00000        1.0000


   UNIFAC ESTIMATION
        ACETONE            FRACTION [% VOL]      20.00
        1CHLOROBENZENE     SOLUBILITY [MOLE FRACTION]   .210E-03
        1CHLOROBENZENE     SOLUBILITY [MG/L]   .111E+04
ANOTHER CALCULATION? (Y=1,N=2)
2
Stop - Program terminated.

C:\JKF >
```

Figure 6. Example Calculation II (continued)

```
C:\JKF >
C:\JKF >type cbace.i
ACETONE
1CHLOROBENZENE
        58.0800000        112.5600000        18.0200000
   7.899000E-001  9.630000E-001  9.971000E-001
        -1.0000000         -1.0000000
n
Y
n
n
n
       3844.6200000        295.7400000        298.0000000
             2
             2
          1.0000000             1          1.0000000            19
          5.0000000            10          1.0000000            54

C:\JKF >
C:\JKF >
C:\JKF >type cbace.o

 ********************************************************************************

 SOLUTE USED IN THE CALCULATION IS     1CHLOROBENZENE
 SOLVENT USED IN THE CALCULATION IS    ACETONE

 PARAMETER           ACETONE           1CHLOROBENZENE           WATER
 MOLECULAR WEIGHT      58.08               112.56                18.02
 DENSITY               .7899                .9630                .9971
 MOLAR VOLUME          73.53               116.88                18.07


 ********************************************************************************

 SOLUTE SOLUBILITY PREDICTIONS IN MIXED SOLVENT SYSTEM

 VOLUME   LOG-LINEAR        UNIFAC        EXCESS FREE --- MOLECULAR SURFACE
 SOLVENT  APPROACH          APPROACH      ENERGY APPROACH AREA APPROACH
        MOLE     SOL.     MOLE     SOL.     MOLE     SOL.     MOLE     SOL.
    %     [-]  [MG/L]      [-]  [MG/L]      [-]  [MG/L]      [-]  [MG/L]

    .00 .00E+00 .00E+00 .52E-04 .33E+03 .00E+00 .00E+00 .00E+00 .00E+00
  20.00 .00E+00 .00E+00 .21E-03 .11E+04 .00E+00 .00E+00 .00E+00 .00E+00


C:\JKF >
```

Figure 7. Example Calculation III: Read from an Input File - Naphthalene Solubility in Methanol/Water

```
****************************************

*              AROSOL                  *
*     Aromatic Solute Solubility       *
*     in Solvent/Water Mixtures        *
*                                      *
*              by                      *
*          Jaw-Kwei Fu                 *
*          Charles Brooks              *
*          Richard G. Luthy            *
*     Carnegie-Mellon University       *
*     Pittsburgh, Pennsylvania         *
*                                      *
*          August, 1986                *

****************************************

Hit RETURN key to continue

 This program estimates aromatic solute solubility in
 solvent/water mixtures.  This program is designed to
 utilize different levels of input parameters to estimate
 aromatic solute solubility. This program employes four
 approaches: LOG-LINEAR, UNIFAC, EXCESS FREE ENERGY, and
 MOLECULAR SURFACE AREA. The input parameters can be read
 from either an existing input file or from the terminal.
 The program estimates solute solubility via approaches
 specified by the user, and stores the results into an
 output file.


 Hit RETURN key to continue

DO YOU HAVE INPUT FILE? (Y OR N)
y
INPUT FILE NAME=
name.i
GIVE THE OUTPUT FILE NAME IN WHICH THE SOLUBILITY
CALCULATION DATA WILL BE STORED
name.o
INPUT PERCENT VOLUME SOLVENT IN THE MIXTURE TO BE EVALUATED
0
LOG-LINEAR REGRESSION INTERCEPT    -5.4287 SLOPE     .03724
   LOG-LINEAR ESTIMATION METHOD

        SOLVENT FRACTION [% VOL]         .00
        LOG LINEAR ESTIMATION SOLUBILITY [MOLE FRACTION]    .373E-05
        LOG LINEAR ESTIMATION SOLUBILITY [MG/L]    .264E+02

  UNIFAC ACTIVITY COEFFICIENT ESTIMATION
    COMPONENT          MOLE FRAC  LN ACTCF        ACTCF

   NAPHTHALENE                .0000   11.84144   138890.8000
   METHANOL                   .0000     .00000        1.0000


   UNIFAC ESTIMATION
        METHANOL             FRACTION [% VOL]         .00
        NAPHTHALENE          SOLUBILITY [MOLE FRACTION]    .216E-05
        NAPHTHALENE          SOLUBILITY [MG/L]    .153E+02
 THE SOLVENT-SOLVENT INTERACTION PARAMETERS ARE  .7644  AND .4566

   EXCESS FREE ENERGY ESTIMATION METHOD
        SOLVENT FRACTION [% VOL]         .00
        EXCESS FREE ENERGY ESTIMATION SOLUBILITY [MOLE FRACTION]    .437E-05
        EXCESS FREE ENERGY ESTIMATION SOLUBILITY [MG/L]       .310E+02
   MOLECULAR SURFACE AREA APPROACH

        SOLVENT FRACTION [% VOL]         .00
        MOLECULAR SURFACE AREA APPROACH SOLUBILITY [MOLE FRACTION]    .437E-05
        MOLECULAR SURFACE AREA APPROACH SOLUBILITY [MG/L]    .310E+02


 ADSORPTION COEFFICIENT OF NAPHTHALENE          IN WATER/METHANOL
MIXTURES IS    .102E+04
```

Figure 7. Example Calculation III (continued)

```
ANOTHER CALCULATION? (Y=1,N=2)
1
INPUT PERCENT VOLUME SOLVENT IN THE MIXTURE TO BE EVALUATED
10
   LOG-LINEAR ESTIMATION METHOD

      SOLVENT FRACTION [% VOL]      10.00
      LOG LINEAR ESTIMATION SOLUBILITY [MOLE FRACTION]    .878E-05
      LOG LINEAR ESTIMATION SOLUBILITY [MG/L]    .589E+02

UNIFAC ACTIVITY COEFFICIENT ESTIMATION
  COMPONENT          MOLE FRAC  LN ACTCF      ACTCF

NAPHTHALENE           .0000    10.95405    57185.4500
METHANOL              .0473      .00000        1.0000


UNIFAC ESTIMATION     FRACTION [% VOL]      10.00
     METHANOL         SOLUBILITY [MOLE FRACTION]    .525E-05
     NAPHTHALENE      SOLUBILITY [MG/L]    .352E+02
     NAPHTHALENE

EXCESS FREE ENERGY ESTIMATION METHOD
      SOLVENT FRACTION [% VOL]      10.00
      EXCESS FREE ENERGY ESTIMATION SOLUBILITY [MOLE FRACTION]    .715E-05
      EXCESS FREE ENERGY ESTIMATION SOLUBILITY [MG/L]    .479E+02
MOLECULAR SURFACE AREA APPROACH

      SOLVENT FRACTION [% VOL]      10.00
      MOLECULAR SURFACE AREA APPROACH SOLUBILITY [MOLE FRACTION]    .111E-04
      MOLECULAR SURFACE AREA APPROACH SOLUBILITY [MG/L]    .744E+02


ADSORPTION COEFFICIENT OF NAPHTHALENE          IN WATER/METHANOL
MIXTURES IS    .656E+03
ANOTHER CALCULATION? (Y=1,N=2)
1
INPUT PERCENT VOLUME SOLVENT IN THE MIXTURE TO BE EVALUATED
50
   LOG-LINEAR ESTIMATION METHOD

      SOLVENT FRACTION [% VOL]      50.00
      LOG LINEAR ESTIMATION SOLUBILITY [MOLE FRACTION]    .271E-03
      LOG LINEAR ESTIMATION SOLUBILITY [MG/L]    .139E+04

UNIFAC ACTIVITY COEFFICIENT ESTIMATION
  COMPONENT          MOLE FRAC  LN ACTCF      ACTCF

NAPHTHALENE           .0000     7.33769     1537.1500
METHANOL              .3086      .00000        1.0000


UNIFAC ESTIMATION     FRACTION [% VOL]      50.00
     METHANOL         SOLUBILITY [MOLE FRACTION]    .195E-03
     NAPHTHALENE      SOLUBILITY [MG/L]    .100E+04
     NAPHTHALENE

EXCESS FREE ENERGY ESTIMATION METHOD
      SOLVENT FRACTION [% VOL]      50.00
      EXCESS FREE ENERGY ESTIMATION SOLUBILITY [MOLE FRACTION]    .137E-03
      EXCESS FREE ENERGY ESTIMATION SOLUBILITY [MG/L]    .705E+03
MOLECULAR SURFACE AREA APPROACH

      SOLVENT FRACTION [% VOL]      50.00
      MOLECULAR SURFACE AREA APPROACH SOLUBILITY [MOLE FRACTION]    .462E-
      MOLECULAR SURFACE AREA APPROACH SOLUBILITY [MG/L]    .236E+04


ADSORPTION COEFFICIENT OF NAPHTHALENE          IN WATER/METHANOL
MIXTURES IS    .114E+03
ANOTHER CALCULATION? (Y=1,N=2)
2
Stop - Program terminated.

C:\JKF >
```

Figure 7. Example Calculation III (continued)

```
C:\JKF >
C:\JKF >type name.i
METHANOL
NAPHTHALENE
          32.0400000        128.1900000         18.0200000
   7.914000E-001  9.625000E-001  9.971000E-001
             .0000000          31.0000000
            1.0000000          39.1000000
            5.0000000          46.5000000
           10.0000000          58.3000000
           20.0000000         104.0000000
           30.0000000         243.0000000
           40.0000000         468.0000000
           50.0000000        1230.0000000
           62.0000000        2956.0000000
           71.0000000        6362.0000000
           75.0000000        9961.0000000
           84.0000000       19831.0000000
           92.0000000       36591.0000000
          100.0000000       66200.0000000
          100.0000000       71093.0000000
           -1.0000000          -1.0000000
Y
Y
Y
         4540.0000000         353.5000000         298.0000000
                1
                2
            1.0000000             16
            8.0000000             10          2.0000000             11
            3.3400000
Y
          155.8000000             .0000000
           24.6000000          47.7000000
Y
            2.0000000
C:\JKF >
C:\JKF >type name.o
```

****

SOLUTE USED IN THE CALCULATION IS NAPHTHALENE
SOLVENT USED IN THE CALCULATION IS METHANOL

PARAMETER	METHANOL	NAPHTHALENE	WATER
MOLECULAR WEIGHT	32.04	128.19	18.02
DENSITY	.7914	.9625	.9971
MOLAR VOLUME	40.49	133.18	18.07

OCTANOL/WATER PARTITION COEFFICIENT .219E+04

****

SOLUTE SOLUBILITY PREDICTIONS IN MIXED SOLVENT SYSTEM

VOLUME SOLVENT	LOG-LINEAR APPROACH		UNIFAC APPROACH		EXCESS FREE ENERGY APPROACH		MOLECULAR SURFACE AREA APPROACH	
%	MOLE [-]	SOL. [MG/L]	MOLE [-]	SOL. [MG/L]	MOLE [-]	SOL. [MG/L]	MOLE [-]	SOL. [MG/L]
.00	.37E-05	.26E+02	.22E-05	.15E+02	.44E-05	.31E+02	.44E-05	.31E+02
10.00	.88E-05	.59E+02	.52E-05	.35E+02	.72E-05	.48E+02	.11E-04	.74E+02
50.00	.27E-03	.14E+04	.20E-03	.10E+04	.14E-03	.70E+03	.46E-03	.24E+04

****

SOLUTE SORPTION PARTITION COEFFICIENT ESTIMATION

METHANOL [%]	OC OF ADSORBENT [%]	KP OF SOLUTE [MOLE/KG]
.00	2.00	1016.51
10.00	2.00	656.44
50.00	2.00	114.16

```
C:\JKF >
```

Molecular Surface Area Calculations

The program MOLACCS calculates MOLecular ACCessible Surface. The program MOLACCS is presently written in standard FORTRAN-77 and requires minor modifications for use on a personal computer. The following explains the general features of the program. This is followed by the presentation of several example calculations.

Program Description

The FORTRAN-77 program MOLACCS is a general purpose program to compute molecular surfaces of molecules using the Richards algorithm (1977). Three types of surface areas are computed: (1) The solvent accessible area as defined by addition of a solvent radius to each solute atom, which is the surface accessibility algorithm referred by Richards as the accessible area; (2) The contact surface area, which is the accessible area without the solvent radius as computed using the Richards algorithm and does not include the re-entrant surface area; and (3) The van der Waals surface area is that corresponding to the accessible area for a solvent probe of zero radius. Note that the areas computed by procedures (1) and (2) are dependent on solvent radius, as explained previously in Chapter 2, while the area computed by procedure (3), the van der Waals area, is not dependent on probe, or solvent, radius.

The program contains several features which permit ease of preparation, manipulation or generation of molecular structures for surface area calculations. These features include the ability to read in existing Cartesian coordinates for a molecule and then calculate the surface areas; the ability to build linear and simply connected molecules; and the ability to add atoms onto existing molecules or to replace atoms in an existing molecule. The ability to add and replace atoms in a molecule is collectively known in MOLACCS as perturbing the existing structure. Finally the program has the ability to combine two existing molecules to create a new molecule.

In order to permit this flexibility, MOLACCS maintains two internal files. These are a

parameter file and a molecular fragment library file. These files are read and written as binary files in MOLACCS, and thus to maintain portability, a utility program CONVERT is provided to change files from binary to formatted, and vice versa.

The molecular fragment library may be manipulated as described above. Additional features are also included which permit the modification of the atomic parameter file. These include (1) addition of new atom types, and (2) changing existing types.

Program Structure

The overall structure of MOLACCS is diagrammed in Figure 8. The program exists as a collection of three modules and a task router. The task router is the main program MOLACCS. In MOLACCS the user is prompted with regard to the specific task to be performed. The options are:

1. PARAMeters – this option routes to the parameter manipulation module, NEWPARAM.

2. FRAGment – this option routes to the molecular fragment manipulation module (task router NEWFRAG).

3. SURFACE – this option calculates surface areas for particular fragments.

4. QUIT – allows for finishing the program and exiting.

Parameter and Fragment Library File Structure

Parameter File:

The parameter file contains all the information required to compute atomic surface areas or to construct molecular fragments. The information associated with each atom type is:

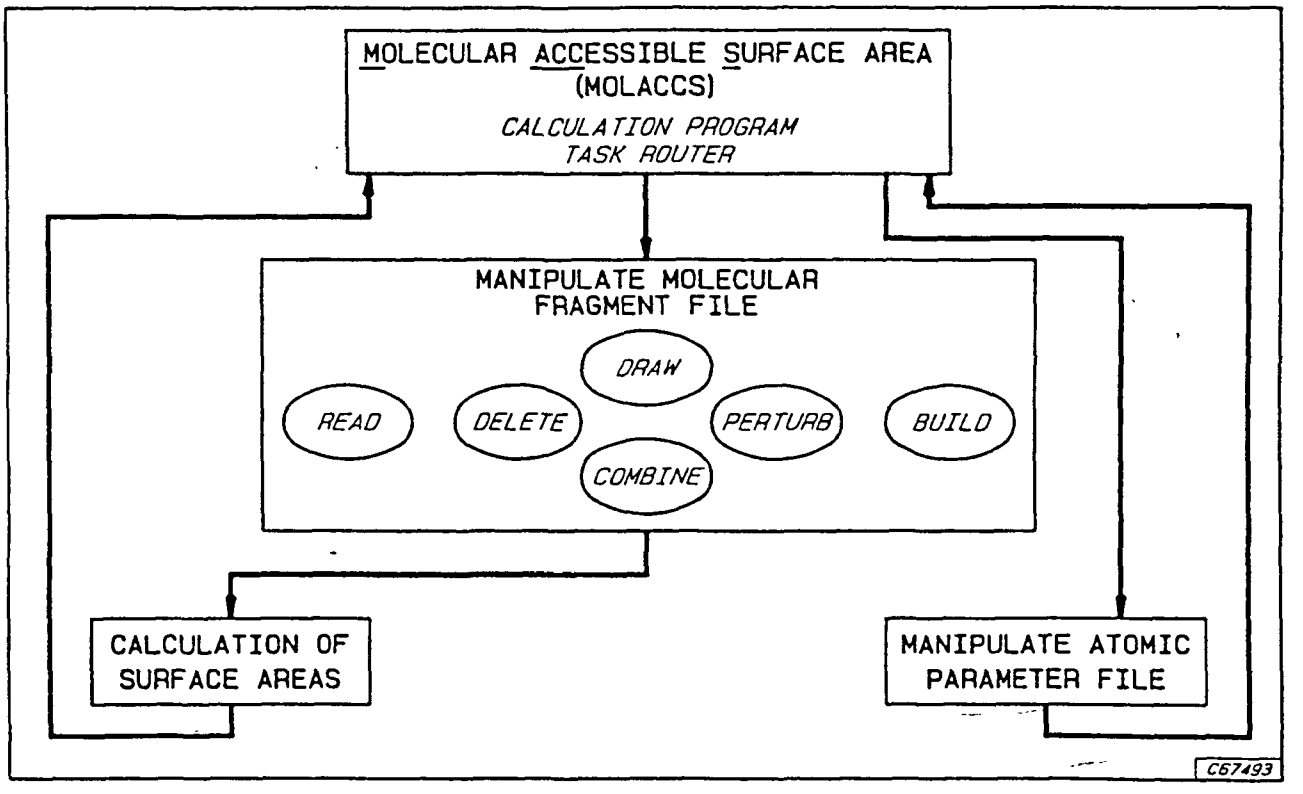
NAME – this is a 4 character identifier.

TYPE – the number of the atom as it appears in the file.

BOND – the bonding radius for an atom.

Figure 8.

MOLACCS Program Structure



ANGLE – the standard bond angle associated with a valence bond angle in which the atom is central.

HYDROPHOBICITY – an integer value between 0 (polar) and 100 (nonpolar) indicating the degree of hydrophobic character. This is used in partitioning total surface areas between polar and hydrophobic surface areas.

van der Waals RADIUS – the van der Waals radius associated with an atom.

A list of the parameters currently resident in the file parameter.bin (for binary version) or parameter.fmt (for formatted version) is given in Table I. Atom type numbers 3 and 4 are to be used if the user desires to develop polycyclic aromatic hydrocarbon fragments in a non-extended atom format using an aromatic-type carbon and an aromatic-type hydrogen atom.

All of the H, C, N, O and S bond lengths, bond angles, and van der Waals radii reported in Table I are taken from the data base employed in CHARMM (Brooks et al., 1983). These parameters are considered to represent an internally consistent and reliable set of atomic parameters. This data base was developed from the analysis of spectroscopic data, for bond and valence angle parameters, and ab-initio quantum chemical calculations, for van der Waals parameters, for use in molecular mechanics calculations on macromolecules (Brooks et al., 1983).

The values for the bond lengths, bond angles, and van der Waals radii reported in Table I for the halogens, i.e. F, Cl, Br, and I, were obtained from: (1) stanadard bond length information, (2) standard hybridization angles, and (3) van der Waal radii based on nearest nobel gas (i.e. $\sigma_F = \sigma_{Ne}$, $\sigma_{Cl} = \sigma_{Ar}$, etc). The source of these data were: Castallan, Physical Chemistry (1971), and Laidler and Meiser, Physical Chemistry (1982).

It is important to recognize that the hydrophocity values assigned to H, C, N and O are estimates of probable reasonable values. The hydrophobicity values asigned to S, F, Cl, Br and I are only initial, arbitrary values. These parameters should be modified accordingly in conjunction with intepretation of results from experimental observations.

The parameters listed in Table I may be compared with values reported in the literature.

Table I

Atom Parameters for Surface Area Calculations

	Atom Type (Name & Number)	Bonding Radius	Angle	VDW Radius	Hydrophobicity
1.	Csp3 (aliphatic, tetrahedral, etc.)	1.53	111.0	2.0	100
2.	Csp2 (aromatic, hybrid, etc.)	1.35	120.0	2.0	100
3.	CARO	1.39	120.0	1.7	100
4.	HARO	0.6	120.0	1.2	100
5.	Nsp3	1.4	120.0	1.7	0
6.	Osp3	1.35	110.0	1.7	0
7.	S	2.00	110.0	1.9	80
8.	F	1.3	111.0	1.65	90
9.	Cl	2.0	111.0	1.9	90
10.	Br	2.3	111.0	2.0	90
11.	I	2.7	111.0	2.1	100

Note: Atom type numbers 3 and 4 are to be used if the user desires to build aromatic molecular fragments without the use of the extended atom approach.

The van der Waals radii values for the non-extended atom format for the aromatic carbon (No. 3) and the hydrogen atom (No. 4) agree with average values reported by Bondi (1968), 1.7 Å and 1.2 Å respectively. These tabulated values are also reasonably close to those chosen by Hermann, 1.77 Å and 1.0 Å respectively. The bonding radii and van der Waals radii reported in Table I are consistent with Valvani et al. (1976) as employed in their "Method B" calculation procedure which used an extended atom format for methyl, methylene, and the hydroxyl group in alcohols; the respective bond lengths and van der Waals radii were: 1.54/2.0, 1.54/2.0, and 1.43/1.7 Å These values are the same as shown in Table I for atom type Nos. 1 and 6, except for a slightly smaller bond distance for the oxygen bond.

Yalkowsky and Valvani (1979) computed the van der Waals surface areas of rigid aromatic hydrocarbons using an extended atom format for methyl and methylene groups. The following respective interatomic distances and van der Waals radii were used: aromatic C-C, 1.40/1.70 Å and aromatic C-H, 1.08/1.20 Å The bonding radius of the aliphatic-aromatic C-C was taken as 1.54 Å and the methyl or methylene group was assigned a van der Waals radius of 2.0 Å These values are consistent with the parameters in Table I, except for the choice of aromatic C-H bonding radius.

Molecular Fragment Library File:

This file contains the cartesian coordinates of molecular fragments plus all the identifiers necessary for surface area calculation or structure manipulation. Specifically, the file contains:

NAME - an 8 character name identifying the fragment.

NUMBER of ATOMS - an integer specifying the number of atoms in a fragment.

ATOM TYPES - an array containing a list of the atom types for all atoms in a fragment.

COORDINATES - the cartesian coordinates for each atom in the fragment.

One fragment library file is included, it is called fragment.bin (for binary version) or fragment.fmt (for formatted version).

Manipulation and Generation of Molecular Fragments

The fragment module permits the manipulation of existing fragments, and the creation or deletion, of fragments in the library file. A brief description of the various options is given below.

READ – the read option reads a new fragment into the library file. The user is prompted for the name of the file from which to read the single fragment. This file is in a specific format and contains the following:

```
* FRAGMENT_NAME      2X, A8 (fragment name)
*                  A2
  NATOM              I5   (# atoms in fragment)
   X1   Y1  Z1  Type1 (20X, 3F10.5, 10X, F10.5)
   .    .   .     .
   .    .   .     .
   .    .   .     .
   X_NATOM  .     .          (Cartesian coordinates plus
                             atom type for all atoms)
```

Appendix 2 shows the existing fragment file; this provides an example of the input format for creation of a new fragment.

BUILD – This routine builds a simple unbranched chain given a user-specified sequence of atom types and dihedral angles. Judicious choice of dihedral angles allows for the building of aromatic (planer) ring structures.

PERTURB – This routine executes a simple modification of existing fragment with two options ADD and REPLACE.

ADD – This attaches single atom substituents to an existing molecular fragment. This routine allows fragments like BENZENE to be perturbed to toluene, or phenol, or chlorobenzene, or to 1,2-dichlorobenzene, etc. The user is asked to specify the three atom sequence indicating where the addition is to take place. The atom is added trans to the first atom specified, e.g., for a final atomic sequence i – j – k – l, it is taken that the atomic sequence i, j, and k are existing atoms with the new atom, l, being added trans to atom i.

REPLACE – This affects single atom replacements in existing fragment by substituting atom type identifiers with no adjustment of geometries. For example, a fragment like BENZENE may be perturbed to PYRIDINE by perturbing a carbon to a nitrogen. If the perturbation is viewed as too dramatic, the replacement is ignored. For example, the replacement of an sp2 carbon by an sp3 carbon is not permitted.

COMBINE – This combines two existing fragments to form a new one. This subroutine combines a parent and a secondary fragment (order of specification) into a third. The user is prompted for a three atom sequence on the parent fragment which indicates where the bond is to be formed and a two atom sequence on the secondary fragment to indicate point of attachment. For example, to combine benzene and butane one would specify 6,1,2 on the parent fragment, benzene, in order to add butane at the 2 position. The sequence 1, 2 is specified for the secondary fragment, butane, to indicate that atom 1 on butane is to join benzene at the specified location.

DELETE – This deletes a fragment from the library.

CHOOSE – This chooses a fragment for a surface calculation.

Additional Notes and Options

Other options are available in this program, including LOG which sets up a log file to which all surface area calculations are saved. There is also included a facility which crudely draws the molecular fragments. This is a crude 40 x 20 bit map to be displayed on the terminal screen and is intended only to guide the user in building or combining molecular fragments.

The aromatic molecular compounds, and related molecules, currently resident in the fragment file are shown in Figure 9. Functional group substituents in the fragment file are shown in Figure 10.

Figure 9. Aromatic and Related Molecules
in the MOLACCS Fragment File

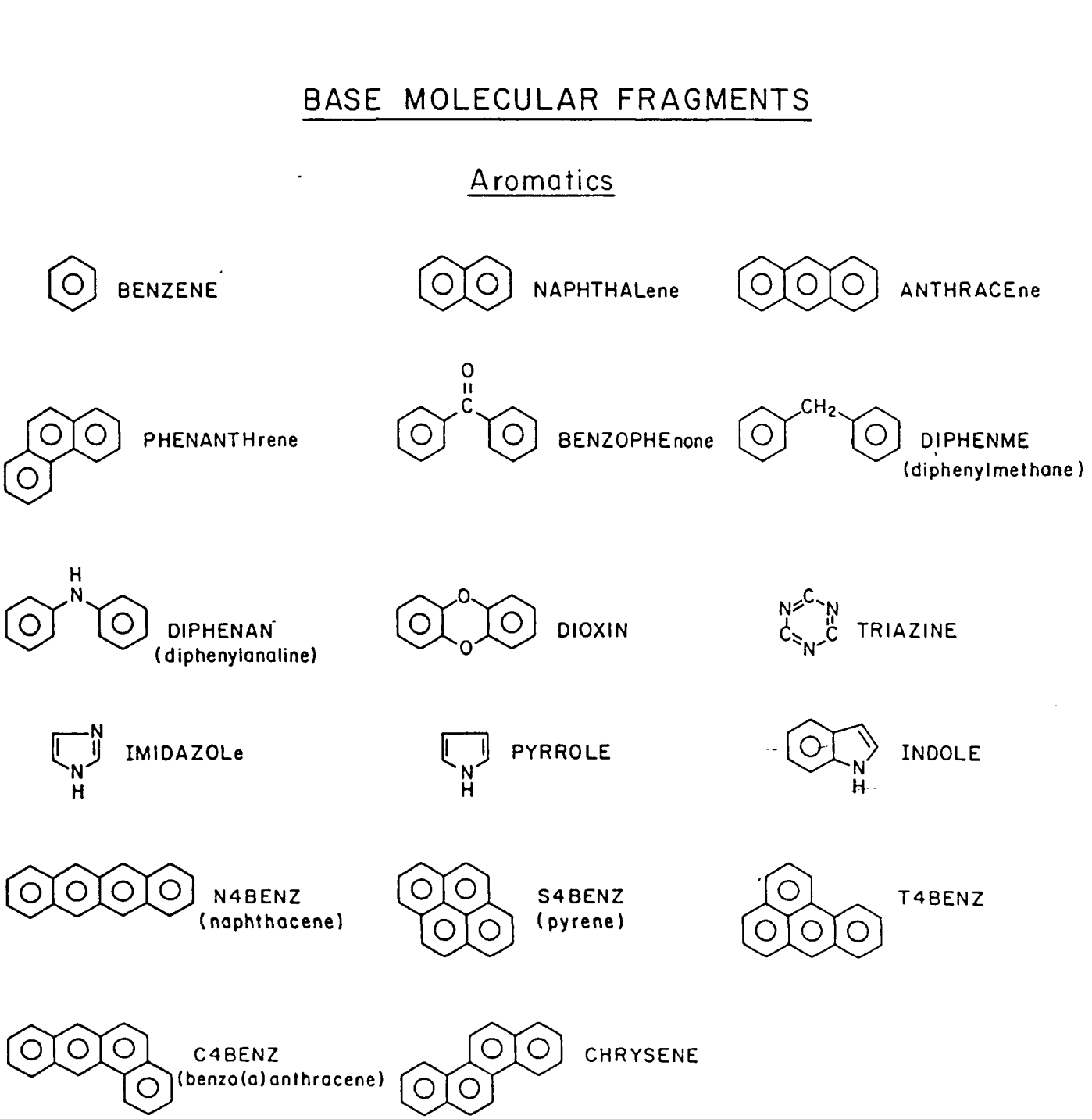
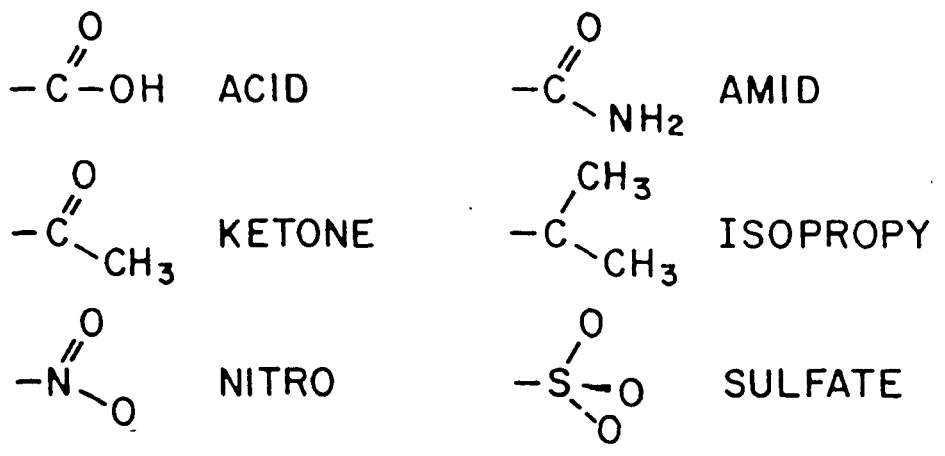


Figure 10. Functional Group Substituents
in the MOLACCS Fragment File



Testing and Initial Results

A number of compounds have been used for evaluative purposes in initial surface area calculations in order to make comparisons with existing literature data. Using the current atomic parameters listed in Table I, calculations of the van der Waals surface areas have been performed. Some of the results are tabulated in Table II along with comparison of results from Pearlman (1986) and Yalkowsky and Valvani (1979) for benzene and several polycyclic aromatic hydrocarbons, and Pearlman (1986) and Valvani et al. (1976) for normal alkanes and alcohols.

The comparisons with the results of Valvani et al. (1976) on straight chain hydrocarbons and alcohols, as computed by "Method C" in their paper, are all very good. A slightly smaller total area is found with the MOLACCS extended atom approach and this is most probably due to a small difference in assumed bond angles. Valvani et al. (1976) did not specify what bond angles were used. Nonetheless, the values computed by the extended atom approach and those of Valvani et al. (1976) agree very well, as the two methods agree within about 1%.

It is noted that calculated values of surface areas for the n-alkanes reported by Pearlman (1986) are not in general agreement with either the current extended atom approach, or the calculations of Valvani et al. (1976). The values reported by Pearlman are about 15% greater than those obtained by either the MOLACCS extended atom approach or by Valvani et al. (1976). Pearlman (1986) did not report assumed values of bond length, or bond angle, or van der Waals radii. Thus the explicit reason for the descrepancy cannot be stated. Nonetheless, this points to the difficulty with making comparisons among different sets of surface area calculations for which the atomic parameters employed for the surface area calculation are unknown.

Table II also shows comparison of van der Waals surface area calculation for benzene and various polycyclic aromatic hydrocarbons and aliphatic-substituted polycyclic aromatic hydrocarbons. The results show comparison of the MOLACCS extended atom approach

Table II

Comparison of van der Waals Molecular Surface Area Calculations

Aromatic Compounds	MOLACCS	Pearlman (1986)	Yalkowsky and Valvani (1979)
Benzene	111.7	110.0	–
Naphthalene	155.0	156.8	155.8
1-Methylnaphthalene	169.4	–	172.5
1-Ethylnaphthalene	184.4	–	187.4
Anthracene	198.2	203.5	202.2
Biphenyl	183.2	–	182.0
Phenanthrene	196.1	199.4	198.0
2-Methylanthracene	214.8	–	226.6
Pyrene	208.04	–	213.0
Chrysene	236.0	–	241.0
Naphthacene	240.1	–	248.0

n-Aliphatic Compounds	MOLACCS	Pearlman (1986)	Valvani et al. (1976)
Butane	104.6	116.1	105.9
Butanol	114.6	–	115.8
Pentane	122.5	138.8	124.0
Pentanol	132.3	–	134.0
hexane	140.3	161.5	142.1
Hexanol	150.2	–	152.1
Heptane	158.2	184.2	160.3
heptanol	168.1	–	170.3

Substituted Benzene	MOLACCS	Pearlman (pers. comm.)
Toluene	128.5	131.2
Benzoic Acid	140.1	140.0
Nitrobenzene	140.5	133.0
Benzamide	139.0	141.9
Aniline	120.5	125.0
Fluorobenzene	119.5	114.4
Chlorobenzene	129.0	127.1
Bromobenzene	133.5	134.5

with data reported by Pearlman (1986) for four compounds, and data reported by Yalkowsky and Valvani (1979) for ten compounds. Pearlman's (1986) data are slightly larger than those of Yalkowsky and Valvani (1979). Yalkowsky and Valvani (1979) calculated molecular van der Waals surface areas according to "Method C" of Valvani et al. (1976), as described earlier. The current extended atcm approach agrees within about 1–2% of the van der Waals surface areas reported by Yalkowsky and Valvani (1979) for most compounds.

Additional comparison of results from MOLACCS is given in Table II for surface area computation data provided by Pearlman (personal communication) for purposes of this investigation. This comparison is made for eight substituted benzene compounds. This comparison shows good agreement of van der Waals area between the method of Pearlman (1986) and MOLACCS for the eight substituted benzene derivatives. Although there appears to be a small difference in selection of atomic parameters for F and N. Also, the comparison depends upon an assumed value of the dihedral angle between the plane of the parent fragment, benzene, and the plane of the branched fragment, such as for the carboxylic or nitro substituent.

Example Calculations

Figure 11 shows example surface area calculations. The presentation in Figure 11 shows how to initiate the program and how to use the various features. The example calculations show listings of accessible and contact surface area with respect to a probe radius of 1.5 Å and the van der Waals surface area. The listings also show the contribution of each atom to the surface area. Also shown is the net value of the individually estimated atomic group contributions to the hydrophobic and polar surface areas.

Reading the Parameter and Fragment Files

Figure 11 shows the initiation of the program in which the user is first presented with a question regarding the use of the prompt, which is indicated by the arrow. The first task the user must perform is either input an atomic parameter file, or read from the existing

atomic parameter file. The user has indicated "p" for input atomic parameters, and "o" to read the "old" file of existing atomic parameters. The atomic parameters are in the file "param.bin", and after typing the file name the user is asked if the atomic parameters should be displayed. After entering "y", for yes, the atomic parameters are shown; these data are the same as shown previously in Table II. The user is asked if any of these parameters should be changed or if any new parameters are to be added to this file.

Next the user asks to have the program read the existing molecular fragment file by entering "f", and "o". The molecular fragment file name is entered, i.e. "fragment.bin." The fragment file is opened, and now the user is presented with a list of options.

List

The first option the user chooses is to list the existing fragments by entering "l." The fragments are shown, and the user is then presented with the several program options.

Choose - Example of Surface Area of Molecular Fragment

The first option the user selects is "choose", which performs a surface area calculation on one of the molecular fragments. The user sets up a file, "surface.o", in which to store the calculation, and again requests a listing of the fragments, and chooses fragment No. 1, "BENZENE". The user is then asked to input a probe radius, and 1.5 Å is selected. The results of the calculation are given for the accessible, contact, and van der Waals surface area, as well as the individual atomic contributions to these surface area values.

Build - Example of Construction of a New Molecular Fragment from Atomic Parameters

After this calculation, the user is presented again with the various options, and the user selects to build a new, unbranched molecular fragment from the existing atomic parameters. The user enters "build", and decides to build "HEXANE". The current atom types and their numbers are presented, and the user enters the sequence of atomic members (six atomic fragments No. 1) for the molecular fragment hexane. The fragment is constructed in the standard trans configuration, and a schematic drawing of the fragment is given.

Figure 11.

Examples of Molecular Surface Area Calculations

```
$ run molaccs
                    MOLACCS
          MOLecular ACCessible Surface
                Version May-86

          Author Charles L. Brooks III
             Department of Chemistry
           Carnegie-Mellon University
             Pittsburgh, PA  15213

 This program will compute molecular surface areas using the
 Richards algorithm with varying solvent probe radii.
 Prompts are given by the symbol ==>, and default
 answer is given in parenthesis, e.g.,
 Do you want to input parameters? (n) ==>
 Do you understand? (n) ==>
y
 Congratulations, you got it!
 Now onto more interesting things.
 Do you want to input atomic parameters <param>,
 set-up a molecular fragment <frag>, calculate
 a surface area <surface> or quit <quit>? (surface) ==>
p

 Current parameter file is empty or non-existent
 Do you want to create a new file <new>,
 or read an old one <old>?  (new) ==>
o
 What is file name, <filename.ext>? (for090.dat) ==>
param.bin
File param.bin status=old will be opened and read
 Do you want to list current atom parameters? (n) ==>
y
Atom #   Name   Bond    Angle    vdW_radius   Hydrophobicity

    1  CSP3     1.530   111.000     2.000    100
    2  CSP2     1.350   120.000     2.000    100
    3  CARO     1.390   120.000     1.700    100
    4  HARO     0.600   120.000     1.200    100
    5  NSP3     1.400   120.000     1.700      0
    6  OSP3     1.350   110.000     1.700      0
    7  S        2.000   110.000     1.900     80
    8  F        1.300   111.000     1.650     90
    9  CL       2.000   111.000     1.900     90
   10  BR       2.300   111.000     2.000     90
   11  I        2.700   111.000     2.100    100
 Do you want to input parameters? (n) ==>

 Do you want to change existing parameters? (n) ==>

 Do you want to input atomic parameters <param>,
 set-up a molecular fragment <frag>, calculate
 a surface area <surface> or quit <quit>? (surface) ==>
f

 Current fragment file is empty or non-existent
 Do you want to create a new file <new>,
 or read an old one <old>?  (new) ==>
o
 What is file name, <filename.ext>? (for091.dat) ==>
fragment.bin
File fragment.bin status=old will be opened and read
         24 fragments read from fragment file
 List existing fragments <list>,
 build new fragment <build>,
 read in new fragment <read>,
 perturb an existing fragment <perturb>,
 combine two existing fragments <combine>,
 delete an existing fragment <delete>
 or choose fragment for surface calculation <choose>? (choose) ==>
l
```

Reading the Parame

and Fragment Files

Figure 11. Example of Molecular Surface Area Calculations (continued)

```
   1 BENZENE    2 NAPHTHAL   3 ANTHRAC    4 PHENANTH
   5 BENZOPHE   6 DIPHENME   7 DIPHENAN   8 DIOXIN
   9 TRIAZINE  10 N4BENZ    11 S4BENZ    12 T4BENZ
  13 C4BENZ    14 CHRYSENE  15 PYRIDINE  16 IMIDAZOL
  17 PYRROLE   18 INDOLE    19 ACID      20 AMIDE
  21 KETONE    22 ISOPROPY  23 NITRO     24 SULFATE
 build new fragment <build>,
 read in new fragment <read>,
 perturb an existing fragment <perturb>,
 combine two existing fragments <combine>,
 delete an existing fragment <delete>
 or choose fragment for surface calculation <choose>? (choose) ==>

 Do you want to set up a LOG file to store
 the surface area calculations?  (n) ==>
y
 What is file name, <filename.ext>? (for093.dat) ==>
surface.o
File surface.o status=new will be opened
 Which molecular fragment do you want?
 list identifiers <list> or input fragment # (input) ==>
l
    1 BENZENE    2 NAPHTHAL   3 ANTHRAC    4 PHENANTH
    5 BENZOPHE   6 DIPHENME   7 DIPHENAN   8 DIOXIN
    9 TRIAZINE  10 N4BENZ    11 S4BENZ    12 T4BENZ
   13 C4BENZ    14 CHRYSENE  15 PYRIDINE  16 IMIDAZOL
   17 PYRROLE   18 INDOLE    19 ACID      20 AMIDE
   21 KETONE    22 ISOPROPY  23 NITRO     24 SULFATE
 Input fragment # ==>
1
 Input solvent probe radius. ==>
1.5
          Surface areas for fragment BENZENE
       computed with respect to probe of radius   1.500

      ACCESSIBLE              CONTACT            van der Waals

Hydrophobic 246.697           80.554          111.714
Polar         0.000            0.000            0.000

                          Atomic Breakdown

  1 CSP2     41.116              13.426             18.619
  2 CSP2     41.116              13.426             18.619
  3 CSP2     41.116              13.426             18.619
  4 CSP2     41.116              13.426             18.619
  5 CSP2     41.116              13.426             18.619
  6 CSP2     41.116              13.426             18.619
 Do you want to input atomic parameters <param>,
 set-up a molecular fragment <frag>, calculate
 a surface area <surface> or quit <quit>? (surface) ==>
f

 List existing fragments <list>,
 build new fragment <build>,
 read in new fragment <read>,
 perturb an existing fragment <perturb>,
 combine two existing fragments <combine>,
 delete an existing fragment <delete>
 or choose fragment for surface calculation <choose>? (choose) ==>
b
 A single unbranched molecular fragment will be built.
 Is that what you want? (yes) ==>

 What is new fragment name, only 8 characters? ==>
HEXANE
New molecular fragment HEXANE will be built
 Number of atoms in linear chain? ==>
6
 Now specify atom types, do you want them listed? (n) ==>
y
```

List

Choose

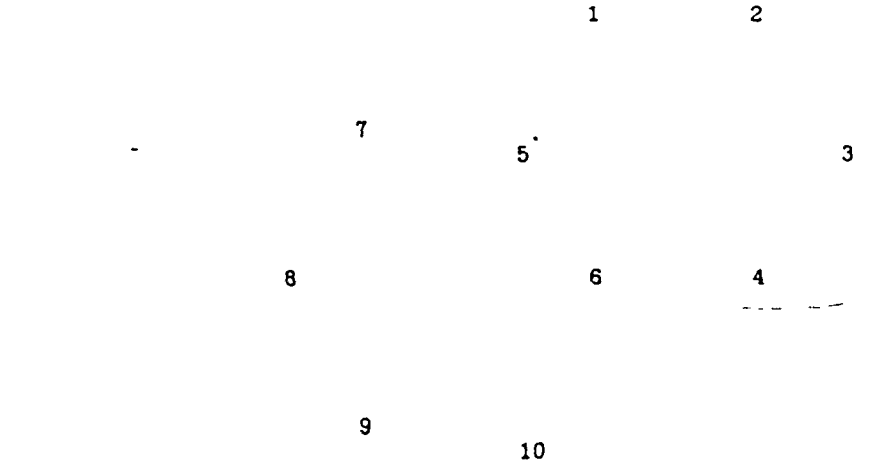
Build

Figure 11. Example of Molecular Surface Area Calculations (continued)

```
Current atom type names and their numbers
 CSP3    1     CSP2    2
 CARO    3     HARO    4
 NSP3    5     OSP3    6
 S       7     F       8
 CL      9     BR     10
 I      11
Input sequence of atom type numbers in assending order
1 1 1 1 1 1
The chain will be built in an all trans configuration
unless otherwise specified.
Do you want other dihedral angles? (n) ==>

Molecular fragment HEXANE   will be drawn
                     1          2

                                     3          4

                                                   5
                                                             6
Return to continue

Save fragment in the library file? (n) ==>

Fragment            25 to be used in surface calculation
Is that what you want? (y) ==>

Input solvent probe radius. ==>
1.5
          Surface areas for fragment HEXANE
      computed with respect to probe of radius    1.500

      ACCESSIBLE              CONTACT              van der Waals

Hydrophobic 298.781           97.561         140.323
Polar         0.000            0.000           0.000

                         Atomic Breakdown

   1 CSP3      83.258             27.186              33.003
   2 CSP3      37.597             12.276              19.319
   3 CSP3      28.536              9.318              17.840
   4 CSP3      28.536              9.318              17.840
   5 CSP3      37.597             12.276              19.319
   6 CSP3      83.258             27.186              33.003
Do you want to input atomic parameters <param>,
set-up a molecular fragment <frag>, calculate
a surface area <surface> or quit <quit>? (surface) ==>
f
```

Figure 11. Example of Molecular Surface Area Calculations (continued)

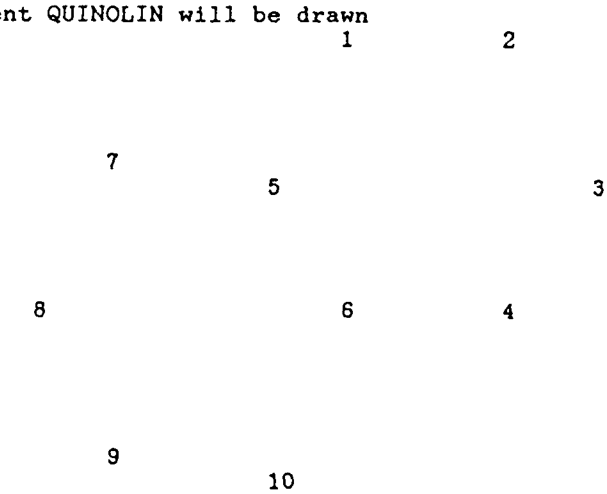
```
 List existing fragments <list>,
 build new fragment <build>,
 read in new fragment <read>,                                    Perturb and Replace
 perturb an existing fragment <perturb>,
 combine two existing fragments <combine>,
 delete an existing fragment <delete>
 or choose fragment for surface calculation <choose>? (choose) ==>
p
 Do you want to replace atoms in existing fragment <replace>
 or add single atom substituents <add>? (add) ==>
r
 Which molecular fragment do you want to perturb?
 list identifiers <list> or input fragment # (input) ==>
l
    1 BENZENE     2 NAPHTHAL    3 ANTHRAC     4 PHENANTH
    5 BENZOPHE    6 DIPHENME    7 DIPHENAN    8 DIOXIN
    9 TRIAZINE   10 N4BENZ     11 S4BENZ     12 T4BENZ
   13 C4BENZ     14 CHRYSENE   15 PYRIDINE   16 IMIDAZOL
   17 PYRROLE    18 INDOLE     19 ACID       20 AMIDE
   21 KETONE     22 ISOPROPY   23 NITRO      24 SULFATE
 Input fragment # ==>
2
Atoms will be replaced in fragment NAPHTHAL
 What is new fragment name, only 8 characters? ==>
QUINOLINE
New molecular fragment QUINOLINE will be built from perturbation of NAPHTHAL
 How many atoms to be replaced in fragment
1
 Molecular fragment NAPHTHAL will be drawn
```




```
 Return to continue

 Atoms will be replaced one at a time
 List atom to be replaced on fragment NAPHTHAL
1
 Now specify new atom type, do you want them listed? (n) ==>
y
 Current atom type names and their numbers
  CSP3    1     CSP2    2
  CARO    3     HARO    4
  NSP3    5     OSP3    6
  S       7     F       8
  CL      9     BR     10
  I      11
 Number for new atom type?
5
Atom type CSP2 at site  1
replaced by atom type NSP3
```

Figure 11. Example of Molecular Surface Area Calculations (continued)

Molecular fragment QUINOLIN will be drawn



Return to continue

Save fragment in the library file? (n) ==>

Fragment 25 to be used in surface calculation
Is that what you want? (y) ==>

Input solvent probe radius. ==>
1.5

Surface areas for fragment QUINOLIN
computed with respect to probe of radius 1.500

	ACCESSIBLE	CONTACT	van der Waals
Hydrophobic	295.527	96.499	142.877
Polar	16.920	4.775	9.078

Atomic Breakdown

1 NSP3	16.920	4.775	9.078
2 CSP2	50.698	16.555	21.855
3 CSP2	41.859	13.668	18.531
4 CSP2	34.574	11.289	17.470
5 CSP2	7.480	2.442	7.146
6 CSP2	4.419	1.443	5.011
7 CSP2	38.203	12.474	18.332
8 CSP2	41.859	13.668	18.531
9 CSP2	41.859	13.668	18.531
10 CSP2	34.574	11.290	17.470

Do you want to input atomic parameters <param>,
set-up a molecular fragment <frag>, calculate
a surface area <surface> or quit <quit>? (surface) ==>
f

Figure 11. Example of Molecular Surface Area Calculations (continued)

Perturb and Add

```
 List existing fragments <list>,
 build new fragment <build>,
 read in new fragment <read>,
 perturb an existing fragment <perturb>,
 combine two existing fragments <combine>,
 delete an existing fragment <delete>
 or choose fragment for surface calculation <choose>? (choose) ==>
p
 Do you want to replace atoms in existing fragment <replace>
 or add single atom substituents <add>? (add) ==>
a
 Which molecular fragment do you want to perturb?
 list identifiers <list> or input fragment # (input) ==>

 Input fragment # ==>
1
Atoms will be added to fragment BENZENE
 What is new fragment name, only 8 characters? ==>
13DICLBENZ
 Name greater than 8 characters, truncated.
New molecular fragment 13DICLBEN will be built by addition to BENZENE
 How many atoms to add to fragment
2
 Molecular fragment BENZENE  will be drawn
                          1                 2



          4                                      3



                          5                 6
 Return to continue

 Atoms will be added one at a time
 Atom added where on fragment BENZENE
 List i,j and k, where k is the bond to add
 atom to and j and i are the k-1, k-2 atoms
 bonded to k
5 4 1
 Now specify atom type to be added
 Do you want them listed? (n) ==>
y
 Current atom type names and their numbers
  CSP3    1    CSP2    2
  CARO    3    HARO    4
  NSP3    5    OSP3    6
  S       7    F       8
  CL      9    BR     10
  I      11
 Number of atom type to be added?
9
 Atom added where on fragment BENZENE
 List i,j and k, where k is the bond to add
 atom to and j and i are the k-1, k-2 atoms
 bonded to k
1 2 3
 Now specify atom type to be added
 Do you want them listed? (n) ==>
```

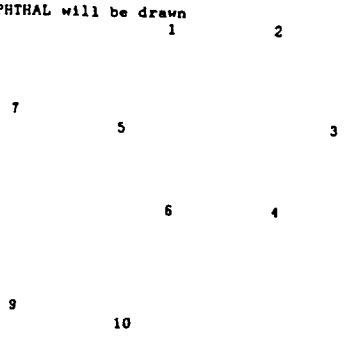
Figure 11. Example of Molecular Surface Area Calculations (continued)

```
 Number of atom type to be added?
9
 Molecular fragment 13DICLBE will be drawn
                  7


                          1           2



                     4                      3             8



                          5           6

 Return to continue

 Save fragment in the library file? (n) ==>

 Fragment            25 to be used in surface calculation
 Is that what you want? (y) ==>

 Input solvent probe radius. ==>
1.5
          Surface areas for fragment 13DICLBE
      computed with respect to probe of radius   1.500

      ACCESSIBLE              CONTACT           van der Waals

Hydrophobic 289.659           92.637       140.400
Polar        15.178            4.740         6.074

                         Atomic Breakdown

  1 CSP2       5.905                 1.928              6.719
  2 CSP2      29.192                 9.532             17.557
  3 CSP2       5.905                 1.928              6.719
  4 CSP2      35.312                11.531             18.075
  5 CSP2      41.432                13.529             18.593
  6 CSP2      35.312                11.531             18.075
  7 CL        75.889                23.699             30.368
  8 CL        75.889                23.699             30.368
 Do you want to input atomic parameters <param>,
 set-up a molecular fragment <frag>, calculate
 a surface area <surface> or quit <quit>? (surface) ==>
f

 List existing fragments <list>,
 build new fragment <build>,
 read in new fragment <read>,
 perturb an existing fragment <perturb>,
 combine two existing fragments <combine>,
 delete an existing fragment <delete>
 or choose fragment for surface calculation <choose>? (choose) ==>
p
 Do you want to replace atoms in existing fragment <replace>
 or add single atom substituents <add>? (add) ==>

 Which molecular fragment do you want to perturb?
 list identifiers <list> or input fragment # (input) ==>
```

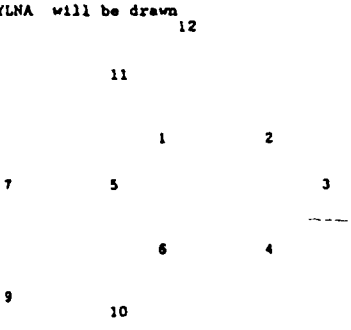
Perturb and Add

Figure 11. Example of Molecular Surface Area Calculations (continued)

```
 Input fragment # ==>
2
Atoms will be added to fragment NAPHTHAL
 What is new fragment name, only 8 characters? ==>
ethylna
New molecular fragment ETHYLNA will be built by addition to NAPHTHAL
 How many atoms to add to fragment
2
 Molecular fragment NAPHTHAL will be drawn
```




```
 Return to continue

 Atoms will be added one at a time
 Atom added where on fragment NAPHTHAL
 List i,j and k, where k is the bond to add
 atom to and j and i are the k-1, k-2 atoms
 bonded to k
6 5 1
 Now specify atom type to be added
 Do you want them listed? (n) ==>

 Number of atom type to be added?
1
 Atom added where on fragment NAPHTHAL
 List i,j and k, where k is the bond to add
 atom to and j and i are the k-1, k-2 atoms
 bonded to k
5 1 11
 Now specify atom type to be added
 Do you want them listed? (n) ==>

 Number of atom type to be added?
1
 Molecular fragment ETHYLNA  will be drawn
```

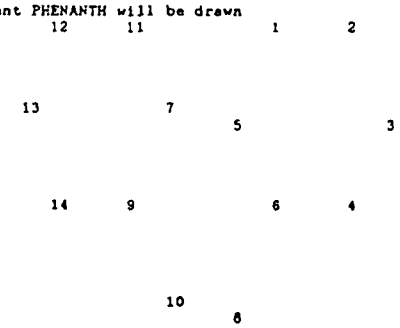



```
 Return to continue

 Save fragment in the library file? (n) ==>

 Do you want to set up a LOG file to store
 the surface area calculations?  (n) ==>
y
 What is file name, <filename.ext>? (for093.dat) ==>
ethyna.sur
File ethyna.sur status=new will be opened
 Fragment           25 to be used in surface calculation
 Is that what you want? (y) ==>

 Input solvent probe radius. ==>
1.5
          Surface areas for fragment ETHYLNA
       computed with respect to probe of radius   1.500

     ACCESSIBLE              CONTACT            van der Waals

Hydrophobic 349.233          114.035          184.428
Polar         0.000            0.000            0.000

                          Atomic Breakdown

  1 CSP2      3.933              1.284             5.258
  2 CSP2     21.494              7.019            14.724
  3 CSP2     41.116             13.426            18.619
  4 CSP2     33.834             11.048            17.558
  5 CSP2      3.717              1.214             5.130
  6 CSP2      3.717              1.214             5.130
  7 CSP2     23.685              7.727            15.451
  8 CSP2     41.116             13.428            18.619
  9 CSP2     41.116             13.426            18.619
 10 CSP2     33.835             11.048            17 558
 11 CSP3     29.273              9.559            17.355
 12 CSP3     72.417             23.646            30 407
 Do you want to input atomic parameters <param>,
 set-up a molecular fragment <frag>, calculate
 a surface area <surface> or quit <quit>? (surface) ==>
f
```

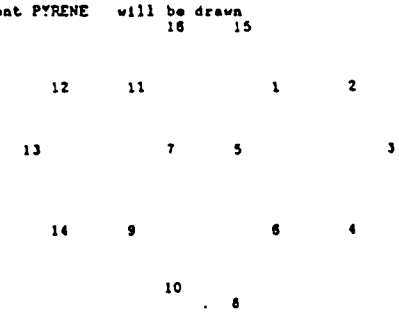
Figure 11. Example of Molecular Surface Area Calculations (continued)

Perturb and Add

```
 List existing fragments <list>,
 build new fragment <build>,
 read in new fragment <read>,
 perturb an existing fragment <perturb>,
 combine two existing fragments <combine>,
 delete an existing fragment <delete>
 or choose fragment for surface calculation <choose>? (choose) ==>
p
 Do you want to replace atoms in existing fragment <replace>
 or add single atom substituents <add>? (add) ==>

 Which molecular fragment do you want to perturb?
 list identifiers <list> or input fragment # (input) ==>
1
    1 BENZENE     2 NAPHTHAL    3 ANTHRAC     4 PHENANTH
    5 BENZOPHE    6 DIPHENME    7 DIPHENAN    8 DIOXIN
    9 TRIAZINE   10 N4BENZ     11 S4BENZ     12 T4BENZ
   13 C4BENZ     14 CHRYSENE   15 PYRIDINE   16 IMIDAZOL
   17 PYRROLE    18 INDOLE     19 ACID       20 AMIDE
   21 KETONE     22 ISOPROPY   23 NITRO      24 SULFATE
 Input fragment # ==>
4
Atoms will be added to fragment PHENANTH
 What is new fragment name, only 8 characters? ==>
PYRENE
New molecular fragment PYRENE will be built by addition to PHENANTH
 How many atoms to add to fragment
2
 Molecular fragment PHENANTH will be drawn
```




```
 Return to continue

 Atoms will be added one at a time
 Atom added where on fragment PHENANTH
 List i,j and k, where k is the bond to add
 atom to and j and i are the k-1, k-2 atoms
 bonded to k
3 2 1
 Now specify atom type to be added
 Do you want them listed? (n) ==>
y
 Current atom type names and their numbers
  CSP3   1    CSP2   2
  CARO   3    HARO   4
  NSP3   5    OSP3   6
  S      7    F      8
  CL     9    BR    10
  I     11
 Number of atom type to be added?
2
 Atom added where on fragment PHENANTH
 List i,j and k, where k is the bond to add
 atom to and j and i are the k-1, k-2 atoms
 bonded to k
13 12 11
 Now specify atom type to be added
 Do you want them listed? (n) ==>
2
 Number of atom type to be added?
2
 Molecular fragment PYRENE   will be drawn
```




```
 Return to continue
```

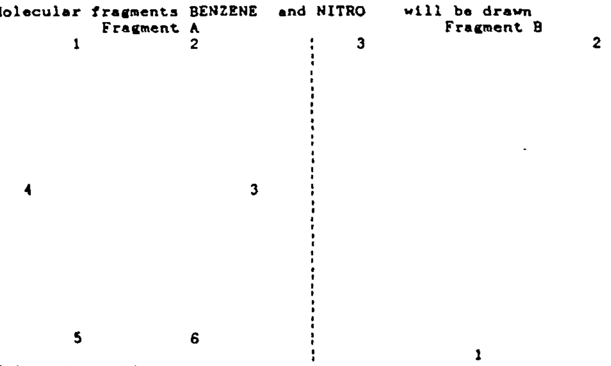
Perturb and Ad

```
 Save fragment in the library file? (n) ==>

 Fragment          25 to be used in surface calculation
 Is that what you want? (y) ==>

 Input solvent probe radius. ==>
1.5
        Surface areas for fragment PYRENE
     computed with respect to probe of radius   1.500

      ACCESSIBLE          CONTACT          van der Waa
Hydrophobic 374.701        122.351        208.039
Polar         0.000          0.000          0.000

                    Atomic Breakdown
  1 CSP2      3.564          1.164          5.019
  2 CSP2     33.854         11.054         17.536
  3 CSP2     41.116         13.426         18.619
  4 CSP2     33.834         11.048         17.558
  5 CSP2      3.714          1.213          5.130
  6 CSP2      3.717          1.214          5.130
  7 CSP2      3.714          1.213          5.130
  8 CSP2     33.835         11.048         17.558
  9 CSP2      3.717          1.214          5.130
 10 CSP2     33.834         11.048         17.558
 11 CSP2      3.564          1.164          5.019
 12 CSP2     33.854         11.054         17.536
 13 CSP2     41.116         13.426         18.819
 14 CSP2     33.834         11.048         17.558
 15 CSP2     33.717         11.010         17.470
 16 CSP2     33.717         11.010         17.470
```

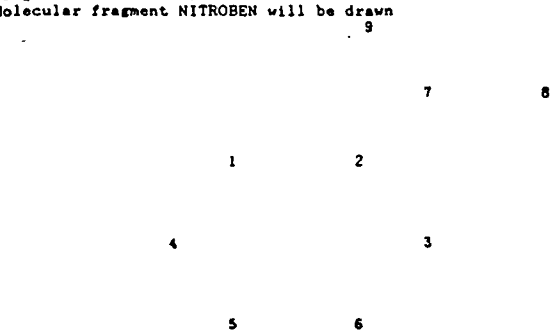
Figure 11. Example of Molecular Surface Area Calculations (continued)

Combine

```
 List existing fragments <list>,
 build new fragment <build>,
 read in new fragment <read>,
 perturb an existing fragment <perturb>,
 combine two existing fragments <combine>,
 delete an existing fragment <delete>
 or choose fragment for surface calculation <choose>? (choose) ==>
com
 Which molecular fragments do you want to combine?
 list identifiers <list> r input fragment # (input) ==>
l
     1 BENZENE     2 NAPHTHAL   3 ANTHRAC     4 PHENANTH
     5 BENZOPHE    6 DIPHENME   7 DIPHENAN    8 DIOXIN
     9 TRIAZINE   10 N4BENZ    11 S4BENZ     12 T4BENZ
    13 C4BENZ     14 CHRYSENE  15 PYRIDINE   16 IMIDAZOL
    17 PYRROLE    18 INDOLE    19 ACID       20 AMIDE
    21 KETONE     22 ISOPROPY  23 NITRO      24 SULFATE
 Input fragment # ==>
1 23
Molecular fragment NITRO    will be added to fragment BENZENE
 What is new fragment name, only 8 characters? ==>
NITROBENZ
New molecular fragment NITROBENZ will be built by combining BENZENE  and NITRO

 Molecular fragments BENZENE  and NITRO    will be drawn
```




```
 Return to continue

 Fragment B added where on fragment A
 List i,j and k, where k is the bond to add
 atom to and j and i are the k-1, k-2 atoms
 bonded to k
6 3 2
 Now specify atom on fragment B where bond is formed
 and the atom its bonded to
1 3
 Current dihedral value around bond connecting
 fragments is    180.0000
 What dihedral do you want?  ==>
180
 Current dihedral value around bond connecting
 fragments is    180.0000
 Molecular fragment NITROBEN will be drawn
```




```
 Return to continue

 Is this the configuration you want? (yes) ==>
y
 Save fragment in the library file? (n) ==>
n
 Do you want to set up a LOG file to store
 the surface area calculations?  (n) ==>
y
 What is file name, <filename.ext>? (for093.dat) ==>
com log
File com.log status=new will be opened
 Fragment            25 to be used in surface calculation
 Is that what you want? (y) ==>

 Input solvent probe radius. ==>
1.5
```

Surface areas for fragment NITROBEN
computed with respect to probe of radius 1.500

	ACCESSIBLE	CONTACT	van der Waals
Hydrophobic	191 354	62.483	95 772
Polar	105.289	29.715	44.770

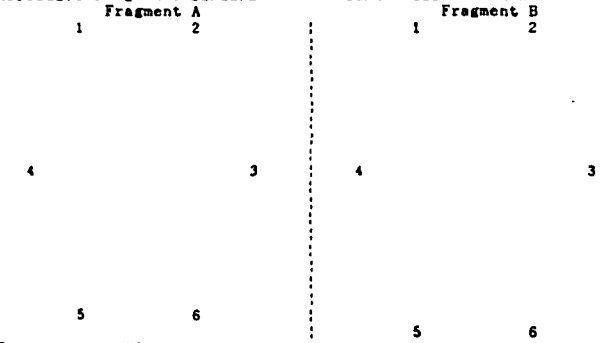
Atomic Breakdown

1	CSP2	29.286	9.563	16.649
2	CSP2	7.937	2.592	7.207
3	CSP2	28.553	9.323	16.322
4	CSP2	41.859	13.668	18.531
5	CSP2	41.859	13.668	18.531
6	CSP2	41.859	13.668	18.531
7	NSP3	1.670	0.471	2.772
8	OSP3	50.787	14 333	20.671
9	OSP3	52.832	14.911	21.327

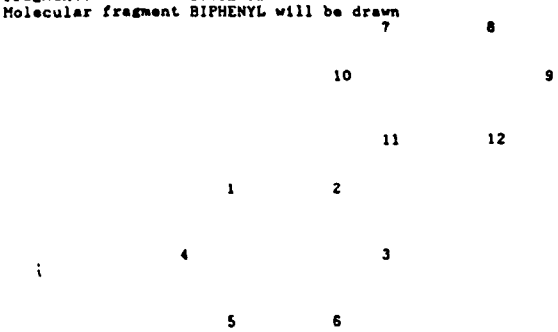
Figure 11. Example of Molecular Surface Area Calculations (continued)

Combine

```
 List existing fragments <list>,
 build new fragment <build>,
 read in new fragment <read>,
 perturb an existing fragment <perturb>,
 combine two existing fragments <combine>,
 delete an existing fragment <delete>
 or choose fragment for surface calculation <choose>? (choose) ==>
com
 Which molecular fragments do you want to combine?
 list identifiers <list> or input fragment # (input) ==>
l
    1 BENZENE     2 NAPHTHAL    3 ANTHRAC     4 PHENANTH
    5 BENZOPHE    6 DIPHENME    7 DIPHENAN    8 DIOXIN
    9 TRIAZINE   10 N4BENZ     11 S4BENZ     12 T4BENZ
   13 C4BENZ     14 CHRYSENE   15 PYRIDINE   16 IMIDAZOL
   17 PYRROLE    18 INDOLE     19 ACID       20 AMIDE
   21 KETONE     22 ISOPROPY   23 NITRO      24 SULFATE
 Input fragment # ==>
1 1
Molecular fragment BENZENE  will be added to fragment BENZENE
 What is new fragment name, only 8 characters? ==>
BIPHENYL
New molecular fragment BIPHENYL will be built by combining BENZENE  and BENZENE
 Molecular fragments BENZENE  and BENZENE  will be drawn
```




```
 Return to continue

 Fragment B added where on fragment A
 List i,j and k, where k is the bond to add
 atom to and j and i are the k-1, k-2 atoms
 bonded to k
6 3 2
 Now specify atom on fragment B where bond is formed
 and the atom its bonded to
5 6
 Current dihedral value around bond connecting
 fragments is -1.9782342E-02
 What dihedral do you want?  ==>
0.0
 Current dihedral value around bond connecting
 fragments is -1.9782342E-02
 Molecular fragment BIPHENYL will be drawn
```




```
 Return to continue

 Is this the configuration you want? (yes) ==>

 Save fragment in the library file? (n) ==>

 Fragment          25 to be used in surface calculation
 Is that what you want? (y) ==>

 Input solvent probe radius. ==>
1.5
```

Surface areas for fragment BIPHENYL
computed with respect to probe of radius 1.500

	ACCESSIBLE	CONTACT	van der Waals
Hydrophobic	358.596	116.439	183.268
Polar	0.000	0.000	0.000

Atomic Breakdown

1 CSP2	24.510	8.003	15.369
2 CSP2	4.346	1.419	5.024
3 CSP2	24.510	8.003	15.369
4 CSP2	41.898	13.681	18.624
5 CSP2	41.898	13.681	18.624
6 CSP2	41.898	13.681	18.624
7 CSP2	41.916	13.687	18.629
8 CSP2	41.899	13.681	18.624
9 CSP2	41.117	13.426	18.619
10 CSP2	24.528	8.009	15.374
11 CSP2	4.347	1.420	5.024
12 CSP2	23.728	7.748	15.363

```
 Do you want to input atomic parameters <param>,
 set-up a molecular fragment <frag>, calculate
 a surface area <surface> or quit <quit>? (surface) ==>
q

MOLACCS normal termination
$
```

This fragment is assigned a number in the event that the user may wish to save it, then a probe radius of 1.5 Å is selected, and the accessible, contact, and van der Waals surface areas are calculated.

Perturb and Replace - Example of Replacement of an Atom in a Fragment

The next example shows use of "perturb" and "replace", in which the molecular fragment naphthalene is used to construct quinoline. The user selects "p" for perturb and then "r" for replace. The molecular fragments are listed; fragment No. 2 is selected; the new fragment is named; and the molecular fragment "NAPHTHAL" is drawn. Atom number 1 in NAPHTHAL is selected for replacement; the atom types are listed; atom number 5 is selected from the list; and the new molecular fragment is drawn. The surface area calculations are then performed for a probe radius of 1.5 Å

The weighted proportion of the total surface area comprised of hydrophobic and polar atomic entities is shown, as in the other examples. These values for hydrophobic and polar surface area for quinoline were used in the AROSOL program Example Calculation I as shown previously in Figure 5.

Perturb and Add - Example of Addition of an Atom to a Fragment

The next calculation shows an example of "perturb" and "add" in which two chlorine atoms are added to benzene to form 1,3-dichlorobenzene. The user enters "p" and "a" for perturb and add; and molecular fragment No. 1, "BENZENE", is selected for perturbation. The user then inputs the number "2" to indicate the addition of two atoms to benzene. The molecular fragment is drawn. The sequence of atoms 5,4,1 on benzene are specified, the atom types are listed, and atom type No. 9 (Cl) is selected. Chlorine is added at the one position, and the steps are repeated to add chlorine at the 3 position by specifying the atomic sequence 1,2,3 on the aromatic ring. The molecule 1,3-dichlorobenzene is drawn, and the surface area calculations are performed for a probe radius of 1.5 Å

Perturb and Add - Example of Addition of an Atom to a Fragment

The next example illustrates the construction and surface area calculation for 1-ethylnaphthalene. The calculation proceeds as in the above example with the atomic sequence 6,5,1 indicated to identify the location of the atomic addition. The atoms are added one at a time, and an aliphatic carbon is first added at the 1-position and the second carbon is added to the new carbon atom by specifying the sequence 5,1,11. The map shows that first additional carbon, labled number 11 was added at the 1-position, and the example continues with the additional carbon being added at the 11-position. The second carbon atom is labled No. 12.

Perturb and Add - Example of Addition for an Atom to a Fragment

The next example illustrates the construction of pyrene from phenanthrene. As before, the user inputs "p" and "a", and the molecular fragments are shown. Molecular fragment number 4, "PHENANTH," is specified, the new fragment is named. The new fragment will be constructed by addition of two aromatic carbon atoms. The location of the first atom to be added is given as 3, 2, 1 for addition at the 1-position. The atom type is selected, i.e. atom type 2 for aromatic-CH. This procedure is repeated with the next atom being added at position 11 by specifying 13, 12, 11. The calculation then proceeds as in the previous examples.

Combine - Example of Combining a Functional Group Substituent to a Fragment

The next example illustrates the construction of nitrobenzene from the molecular fragment benzene and the nitro functional group substituent. The user enters "c" for combine, and then enters the fragment numbers 1 and 23 for "BENZENE" and "NITRO", respectively. The fragment numbers are entered with the parent fragment identification first, followed by the secondary fragment identification. The new fragment is named, and then the parent and secondary fragments are displayed. Atoms "6, 3, 2" are specified on the parent fragment to indicate the point of attachment at atom number 2 on benzene. The two atom sequence "1, 3" is specified for the secondary fragment to indicate the orientation and attachment of the nitrogen atom, which is atom number 1 in the NITRO functional group. The program then indicates that the dihedral angle between the plane containing benzene and the plane

containing the nitro group is 180°. This value of the dihedral angle is accepted for calulation. The configuration is checked, and the surface area calculation is performed.

NOTE: The fragment atomic numbering system is shown in the appendix in the fragment file. This numbering system must be used to ensure that the fragments are combined in the manner desired.

The combine operation may have to executed at one or two different locations around a symmetric aromatic ring until the desired atomic configuration is achieved. This requirement results from the manner in which the combine operation is performed. The combining of two molecular fragments to form a daughter fragment is accomplished by a series of three operations. The first step entails the atattachment of the principal atom (i.e. atom number 1) of the secondary fragment to the parent fragment with the correct bond angle. The computer program then translocates the secondary fragment over the attached atom to achieve superposition of the principal atoms. This translocation is executed by moving the secondary fragment to the attached atom on the parent fragment without rotation. This translocation is performed by superimposing the fragments in the orientation shown in the orginal presentation of fragment A and B. The computer program then rotates the branched atoms on the secondary fragment to achieve the correct bond angle between the branched atoms and the principal atom on the secondary fragment. This rotation is restricted to about plus or minus 45°. Therefore, the correct strategy for combining fragments is to select a location on the parent fragment for which a direct lateral translocation of the secondary group results in approximately the correct orientation of the branched atoms.

The third step of the combine operation is the rotation of attacheded secondary fragment to the specified dihedral angle between the parent fragment and secondary fragment. The dihedral angle is the angle formed by the plane containing the primary fragment and the plane containing the secondary fragment. A dihedral angle of 0° or 180° means that both fragments lie in the same plane, while a dihedral angle of 90° means that the two fragments lie in planes perpendicular to each other.

Combine - Example of Combining Two Fragments

The last example shows the combination of benzene and benzene to form biphenyl. Again, the user selects combine, and the fragment benzene is taken for both parent and secondary fragment. The user specifies the sequence "6, 3, 2" for attachment at location number 2 on the parent fragment, and "5, 6" for attachment of the secondary fragment at location 5. The molecular fragment is drawn, and the calculation is performed for a dihedral angle of 0° (actually 0.02° owing to runoff error).

The "combine" operation permits only one bond to be formed between fragments. As indicated above, the operation may have to be executed several times at different locations around the parent ring in order to achieve the correct symmetry and orientation for the new molecule.

REFERENCES

Anderson, F.E., "Supplement to Program UNIFAC," 3rd revision, Dept. of Chemical Engineering, Univ. of California, Berkeley, CA, 1983.

Arbuckle, W.B., "Estimating Activity Coefficients for Use in Calculating Environmental Parameters," *Environmental Science and Technology*, Vol. 17, No.9, 1983, pp. 537-542.

Barton, A.F. M., *Handbook of Solubility Parameters and Other Cohesion Parameters* CRC Press Inc., Boca Raton, FL, 1983.

Benerjee, S., "Solubility of Organic Mixtures in Water," *Environmental Science and Technology*, Vol. 18, 1984, pp. 587-591.

Benerjee, S., "Calculation of Water Solubility of Organic Compounds with UNIFAC-Derived Parameters," *Environmental Science and Technology*, Vol. 19, No. 4, 1985, pp. 364-369.

Brooks, B.R., Bruccoleri, R., Olafson, B., States, D., Swaminathan, S., and Karplus, M., "Chemistry at Harvard - CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations," *Journal of Computational Chemistry*, Vol. 4, 1983, pp. 183.

Campbell, J.R., Luthy, R.G., and Carrondo, M.J.T., "Measurement and Prediction of Distribution Coefficients for Wastewater Aromatic Solutes," *Environmental Science and Technology*, Vol. 17, No. 10, 1983, pp. 582-590.

Campbell, J.R., and Luthy, R.G., "Prediction of Aromatic Solute Partition Coefficients Using the UNIFAC Group Contribution Model," *Environmental Science and Technology*, Vol. 19, No. 10, 1985, pp. 980-985.

Castallan, G.W., *Physical Chemistry*, 2nd Ed., Addison Wesley, Reading, MA, 1971.

Chiou, C.T., Freed, V.H., Schmedding, D.W. and Kohnert, R.G., "Partition Coefficient and

Bioaccumulation of Selected Organic Chemicals," *Environmental Science and Technology*, Vol. 11, 1977, pp. 475-478.

Chiou, C.T., Schmedding, D.W., and Manes, M., "Partitioning of Organic Compounds in Octanol-Water Systems," *Environmental Science and Technology*, Vol. 16, No. 1, 1982, pp. 4-10.

Dzombak, D.A., and Luthy, R.G., "Estimating Adsorption of Polycyclic Aromatic Hydrocarbons on Soils," *Soil Science*, Vol. 137, No. 5, 1984, pp. 292-308.

Fredenslund, A., Jones, R.L., and Prausnitz, J.M., "Group Contribution Estimation of Activity Coefficients in Nonideal Liquid Mixtures," *AIChE Journal*, Vol. 21, No. 6, 1975, pp. 1086-1099.

Fu, J.K., and Luthy, R.G., "Aromatic Compound Solubility in Solvent/Water Mixtures," *Journal of Environmental Engineering*, ASCE, Vol. 112, No. 2, Apr., 1986a, pp. 328-345.

Fu, J.K., and Luthy, R.G., "Effect of Organic Solvent on Sorption of Aromatic Solutes onto Soils," ASCE, *Journal of Environmental Engineering*, Vol. 112, No. 2, Apr., 1986b, pp. 346-366.

Fu, J.K., and Luthy, R.G., "Pollutant Sorption onto Soil/Sediments in Organic/Aqueous Solvent Systems," Report to US Environmental Protection Agency, Environmental Research Laboratory, Athens, GA, under Contract No. R810878-01, May, 1985.

Gmehling, J.G., Anderson, T.F., and Prausnitz, J.M., "Solid-Liquid Equilibria Uisng UNIFAC," *Industrial & Engineering Chemistry; Fundamentals*, Vol. 17, No. 4, 1978, pp. 269-273.

Gmehling, J.G., Rasmussen, P., and Fredenslund, A., "Vapor-Liquid Equilibria by UNIFAC Group Contribution, Revision and Extension 2," *Industrial & Engineering Chemistry; Process Design & Development*, Vol. 21, 1982, pp. 118-127.

Grain, C.F., "Activity Coefficient," *Handbook of Chemical Property Estimation Methods*, W.J.

Lyman, W.F. Reehl, and D.H. Rosenblatt, Eds., McGraw-Hill Book Company, New York, NY, 1982.

Hamaker, J.W., and Thompson, J.M., "Adsorption," *Organic Chemicals in the Soil Environment*, Vol. 1, C.A.I. Goring and J.W. Hamaker, Eds., Marcel-Dekker, New York, NY, 1972.

Hassett, J.J., Means, J.C., Banwart, W.L., and Wood, S.G., "Sorption Properties of Sediments and Energy-Related Pollutants," US Environmental Protection Agency, No. EPA-600/3-80-041, 1980.

Hermann, R.B., "Theory of Hydrophobic Bonding II. The Correlation of Hydrocarbon Solubility in Water with Solvent Cavity Surface Area," *Journal of Physical Chemistry*, Vol. 76, No. 19, 1972, pp. 2754-2759.

Hildebrand, J.H., and Scott, R.L., *The Solubility of Nonelectrolytes*, 3rd ed., Reinhold Publishing Company, New York, NY, 1950.

Hildebrand, J.H., Prausnitz, J.M., and Scott, R.L., *Regular and Related Solutions*, Von Nostrand Reinhold Company, New York, NY, 1970.

Jorgenson, W. and Swerison, C., "Optimized Intermolecular Potential Functions for Amides and Peptides. Hydration of Amides.", *Journal American Chemical Society*, Vol. 107, 1985, pp. 569 - 578.

Jorgenson, W., and Swerison, C., "Optimized Intermolecular Potential Functions for Amides and Peptides. Structures and Properties of Liquid Amides," *Journal American Chemical Society*, Vol. 107, 1985, pp. 1489 - 1496.

Karickhoff, S.W., "Semi-Empirical Estimation of Sorption of Hydrophobic Pollutants on Natural Sediments and Soils," *Chemosphere*, Vol. 10, No. 8, 1981, pp. 833-846.

Karickhoff, S.W., "Organic Pollutant Sorption in Aquatic Systems," *Journal of Hydraulic Engineering*, ASCE, Vol. 110, No. 6, 1984, pp. 707-725.

Laidler and Meiser, *Physical Chemistry*, 1982.

Lambert, S.M., "Functional Relationship Between Sorption in Soil and Chemical Structure," *Journal of Agricultural and Food Chemistry*, Vol. 15, No. 4, 1967, pp. 572-576.

Leo, A.J., Hansch, C., and Elkins, D., "Partition Coefficients and Their Use," *Chemical Reviews*, Vol. 71, No. 6, 1971, pp. 525-616.

Lyman, W.J., "Solubility in Various Solvents," *Handbook of Chemical Property Estimation Methods*, W.J. Lyman, W.F. Reehl, and D.H. Rosenblatt, Eds., McGraw-Hill Book Company, New York, NY, 1982.

Mackay, D., "Finding Fugacity Feasible," *Environmental Science and Technology*, Vol. 13, No. 10, 1979, pp. 1218-1223.

Mackay, D., "Aqueous Solubility of Polynuclear Aromatic Hydrocarbons," *Journal of Chemical and Engineering Data*, Vol. 22, No. 4, 1977, pp. 399-402.

Martin, A., Wu, P.L., Adjei, A., Beerbower, A., and Prausnitz, J.M., "Extended Hasen Solubility Approach: Naphthalene in Individual Solvents," *Journal of Pharmaceutical Sciences*, Vol. 70, No. 11, 1981, pp. 1260-1264.

Martin, A., Wu, P.L., Adjei, A., Lindstrom, R.E., and Elworthy, P.H., "Extended Hildebrand Solubility Approach and the Log-Linear Solubility Equation," *Journal of Pharmaceutical Sciences*, Vol. 71, No. 8, 1982, pp. 849-856.

Miller, M.M., Wasik, S.P., Wang, G.-L., Shiu, W.-Y., and Mackay, D., "Relationships Between Octanol -Water Partition Coefficient and Aqueous Solubility, *Environmental Science and Technology*, Vol. 19, No. 6, 1985, pp. 522-529.

Neely, W.B., and MacKay, D., "Evaluative Model for Estimating Environmental Fate," *Modelling the Fate of Chemicals in the Aquatic Environment*, K.L. Dickson, A.W. Maki, and J. Cairns, Jr., Eds., Ann Arbor Science, Ann Arbor, MI, 1982.

Pearlman, R.S., "Molecular Surface Area and Volume: Their Calculation and Use in Predicting Solubilities and Free Energies of Desolvation," in *The Partitioning Coefficient: Theory and Measurement*, W. Dunn, J. Black and R.S. Pearlman, Eds., Pergammon Press, New York, NY 1986.

Pearlman, R.S., "Molecular Surface Areas and Volumes and Their Use in Structure/Activity Relationships," in *Physical Chemical Properties of Drugs*, Marcel Dekker, New York, NY, 1980 pp. 321-347.

Prausnitz, J.M., *Molecular Thermodynamics of Fluid-Phase Equilibria*, Prentice Hall, Inc., Englewood Cliffs, NJ, 1969.

Richards, F.M., "Areas, Volumes, Packing, and Protein Structure", *Annual Reviews of Biophysics and jBioengineering*, Vol. 6, 1977, pp. 151-176.

Ryan, T.A., Jr., Joiner, B.L., and Ryan, B.F., *Minitab Student Handbook*, Duxbury Press, Boston, MA, 1981.

Tsonopoulos, C., and Prausnitz, J.M., "Activity Coefficient of Armotic Solutes in Dilute Aqueous Solutions," *Industrial & Engineering Chemistry; Fundamentals*, Vol. 10, No. 4, 1971, pp. 593-600.

Valvani, S.C., Yalkowsky, S.H. and Amidon, G.L., "Solubility of Nonelectrolytes in Polar Solvents VI. Refinements in Molecular Surface Area Computations," *Journal of Physical Chemistry*, Vol. 80, No. 8, 1976, pp. 829-835.

Weast, R.C., Ed., *CRC Handbook of Chemistry and Physics*, 63rd ed., Chemical Rubber Publishing company, Boca Raton, FL, 1983.

Williams, N.A., and Amidon, G.L., "Excess Free Energy Approach to the Estimation of Solubility in Mixed Solvent Systems I: Theory," *Journal of Pharmaceutical Sciences*, Vol. 73, No. 1, 1984a, pp. 9-13.

Williams, N.A., and Amidon, G.L., "Excess Free Energy Approach to the Estimation of

Solubility in Mixed Solvent Systems II: Ethanol-Water Mixtures," *Journal of Pharmaceutical Sciences*, Vol. 73, No. 1, 1984b, pp. 14-18.

Williams, N.A., and Amidon, G.L., "Excess Free Energy Approach to the Estimation of Solubility in Mixed Solvent Systems III: Ethanol-Propylene Glycol-Water Mixtures," *Journal of Pharmaceutical Sciences*, Vol. 73, No. 1, 1984c, pp. 18-23.

Yalkowsky, S.H., "Estimation of Entropies of Fusion of Organic Compounds," *Industrial & Engineering Chemistry; Fundamentals*, Vol. 18, No. 2, 1979, pp. 108-111.

Yalkowsky, S.H., Flynn, G.L., and Amidon, G.L., "Solubiltiy of Nonelectrolytes in Polar Solvents," *Journal of Pharmaceutical Sciences*, Vol. 61, No. 6, 1972, pp. 983-984.

Yalkowsky, S.H., and Flynn, G.L., "Correlation and Prediction of Mass Transport Across Membranes II: Influence of Vehicle Polarity on Flux from Solutions and Suspensions," *Journal of Pharmaceutical Sciences*, Vol., 63, No. 8, 1974, pp. 1276-1280.

Yalkowsky, S.H., Valvani, S.C., and Amidon, G.L., "Solubility of Nonelectrolytes in Polar Solvents IV: Nonpolar Drugs in Mixed Solvents," *Journal of Pharmaceutical Sciences*, Vol. 65, No. 10, 1976, pp. 1488-1494.

Yalkowsky, S.H. and Valvani, S.C., "Partition Coefficient and Surface Areas of Some Alkylbenzenes," *Journal of Medicinal Chemistry* Vol. 19, No. 5, 1976, pp. 727-726.

Yalkowsky, S.K., and Valvani, S.C., "Solubility and Partitioning I: Solubility of Nonelectrolytes in Water," *Journal Pharmaceutical Sciences*, Vol., 69, No. 8, 1980, pp. 912-922.

Yalkowsky, S.H., and Valvani, S.C., "Solubilities and Partitioning 2. Relationships Between Aqueous Solubilities, Partition Coefficients, and Molecular Surface Areas of Rigid Aromatic Hydrocarbons," *Journal of Chemical and Engineering Data*, Vol. 24, No. 2, 1979, pp. 127-129.

APPENDIX 1: AROSOL PROGRAM LISTING

```
C**********************************************************************
C
C                              AROSOL
C          (Aromatic Solute Solubility in Solvent/Water Mixtures)
C
C       This program was developed to estimate aromatic solute
C       solubility in solvent/water mixtures.  This software is comprised
C       of four different approaches: UNIFAC, LOG-LINEAR, EXCESS FREE
C       ENERGY, and MOLECULAR SURFACE AREA.  The program was developed
C       to accept different degree of input parameters in order to
C       estimate solubility in solvent/water mixtures.
C
C     This program consists of the following subroutines:
C       INPUT: INPUT DATA TO BE READ FROM THIS SUBROUTINE
C       SETUP: READ INPUT DATA FROM TERMINAL AND STORE INTO A FILE
C       UNIFAC: CALCULATE ACTIVITY COEFFICIENTS FOR EACH COMPONENT
C       REG1: LINEAR LEAST-SQUARE REGRESSION
C       REG2: TWO PARAMETER LEAST SQUARE REGRESSION
C       SOLCAL: CALCULATE SOLUTE SOLUBILITY
C       LOGLNR: LOG-LINEAR APPROACH
C       UNIEST: NUMERICAL ESTIMATION FOR SOLUTE SOLUBILITY USING
C               ACTIVITY COEFFICIENT CALCULATED BY UNIFAC
C       EXFREN: EXCESS FREE ENERGY APPROACH
C       MSA: MOLECULAR SURFACE AREA APPROACH
C       ADS: ESTIMATE ADSORPTION PARTITION COEFFICIENT
C
C     THE VARIABLES USED IN EACH SUBROUTINE ARE LISTED AS FOLLOWS
C     SUBROUTINE SETUP AND INPUT
C         MOL: COMPONENT NAME
C         TEMP: SYSTEM TEMPERATURE
C         TM: MELTING TEMPERATURE OF SOLUTE
C         PC: SOLUTE OCTANOL/WATER PARTITION COEFFICIENT
C         OC: ORGANIC CARBON CONTENT OF ADSORBENT
C         HF: HEAT OF FUSION SOLUTE
C         MW1: MOLECULAR WEIGHT OF SOLVENT
C         MW2: MOLECULAR WEIGHT OF SOLUTE
C         MW3: MOLECULAR WEIGHT OF WATER
C         VM1: MOLAR VOLUME OF SOLVENT
C         VM2: MOLAR VOLUME OF SOLUTE
C         VM3: MOLAR VOLUME OF WATER
C         D1: DENSITY OF SOLVENT
C         D2: DENSITY OF SOLUTE
C         D3: DENSITY OF WATER
C         V1: SOLVENT VOLUME FRACTION OF KNOWN SOLUTE SOLUBILITY
C         S1: KNOWN SOLUTE SOLUBILITY IN MG/L
C         XLL: KNOWN SOLUTE SOLUBILITY IN MOLE FRACTION
C         X0: SOLUTE SOLUBILITY IN 0% SOLVENT
C         X100: SOLUTE SOLUBILITY IN PURE SOLVENT
C         LSTA: SOLUTE STATE IN SYSTEM TEMPERATURE
C         NMG: NUMBER OF MAIN GROUP FOR UNIFAC
C         NSG: NUMBER OF SECONDARY GROUP FOR UNIFAC
C         RR: VAN der WAALS VOLUME FOR SENCONARY GROUP
C         QQ: VAN der WAALS SURFACE OF FUNCTIONAL GROUP
C         NKTAB: MAIN GROUP NUMBER
C         GNAM: FUNCTIONAL GROUP STRUCTURE
C         KODE: INTERACTION PARAMETER IDENTIFICATION CODE
C         PARAM: FUNCTIONAL GROUP INTERACTION PARAMETER
C         NGMOL: NUMBER OF SECONDARY GROUP IN COMPONENT
C         NOGP: NUMBER OF SECONDARY GROUP
C         IDGP: SECONDARY GROUP INDEX
C         TSA: TOTAL SURFACE AREA
C         PSA: POLAR SURFACE AREA
C         HSA: HYDROCARBONEOUS SURFACE AREA
C         DEH: HYDROCARBONEOUS INTERFACIAL FREE ENERGY
C         DEP: POLAR INTERFACIAL FREE ENERGY
C
C     SUBROUTINE SOLCAL
C         NCNLL: NUMBER OF CALCULATIONS FOR LOG-LINEAR APPROACH
C         NCNUNI: NUMBER OF CALCULATIONS FOR UNIFAC APPROACH
C         NCNEXF: NUMBER OF CALCULATIONS FOR EXCESS FREE ENERGY APPROACH
C         NCNMSA: NUMBER OF CALCULATIONS FOR MOLECULAR SURFACE AREA APPROACH
C         VSOV: VOLUME FRACTION OF SOLVENT
C         XSOV: MOLE FRACTION OF SOLVENT
C         NCAL: NUMBER OF CALCULATION
C
C
C     SUBROUTINE LOGLNR
C         XL: LOG OF KNOWN SOLUTE MOLE FRACTION SOLUBILITY
C         K1: INTERCEPT OF LINEAR REGRESSION ANALYSIS
C         SLP1: SOLPE OF LINEAR REGRESSION
C         XLOG: LOG-LINEAR APPROACH MOLE FRACTION SOLUTE SOLUBILITY
C         SLOG: LOG-LINEAR APPROACH SOLUTE SOLUBILITY [MG/L]
C
C     SUBROUTINE EXFREN
C         C2: SOLUTE SOLVENT INTERACTION PARAMETER
C         NMIX: NUMBER OF MIXTURES FOR UNIFAC ACTIVITY COEFFICIENT ESTIMATION
C         Z1,Z3: SOLVENT MOLE FRACTION (SOLUTE FREE)
C         ACT: ACTIVITY COEFFICIENT
C         ALGAC: LOG OF ACTIVITY COEFFICIENT
C         FRENG: EXCESS FREE ENERGY
C         RPA1,RPA2: TEMPERORY PARAMETER FOR REGRESSION ANALYSIS
C         A13,A31: SOLVENT INTERACTION PARAMETER
C         XFE: MOLE FRACTION SOLUTE SOLUBILITY ESTIMATED BY EXCESS FREE
C              ENERGY APPROACH
C         SFE: SOLUTE SOLUBILITY ESTIMATED BY EXCESS FREE ENERGY APPROACH
C
C
C     SUBROUTINE MSA
C         K: BOLTZMANN CONSTANT
C         XMSA: MOLE FRACTION SOLUTE SOLUBILITY ESTIMATED BY MOLECULAR
C               SURFACE AREA APPROACH
C         SMSA: SOLUTE SOLUBILITY ESTIMATED BY MOLECULAR SURFACE AREA METHOD
C
C
C     SUBROUTINE UNIEST
C         NGMTEM: TEMPERORY PARAMETER FOR NGMOL
C         NOGTEM: TEMPERORY PARAMETER FOR NOGP
C         IDGTEM: TEMPERORY PARAMETER FOR IDGP
C         NSP: NUMEBR OF COMPONENT IN THE MIXTURE
C         NGM,NGP,IGP: EQUIVALENT PARAMETER OF NGMOL, NOGP AND IDGP
C         R: GAS CONSTANT
C         AC: ACTIVITY COEFFICIENT
C         ACLG: LOG OF ACTIVITY COEFFICIENT
C         A: LOWER BOUNDARY FOR NUMERICAL ESTIMATION
C         B: UPPER BOUNDARY FOR NUMERICAL ESTIMATION
C         EPS: TOLERENCE FOR NUMERICAL ESTIMATION
C         FA,FB,FX: CALCULATED FUNCTION VALUE FOR LOWER BOUNDARY, UPPER
C                   BOUNDARY AND ESTIMATED VLAUE
C         XUNI: MOLE FRACTION SOLUTE SOLUBILITY ESTIMATED BY UNIFAC METHOD
C         SUNI: SOLUTE SOLUBILITY ESTIMATED BY UNIFAC METHOD
C
C
C     SUBROUTINE UNIFAC
C         NSPS: NUMBER OF COMPONENTS IN THE MIXTURE
C         NGPT: NUMBER OF GROUPS IN THE MIXTURE
C         NSG: NUMBER OF SECONDARY GROUPS IN THE CALCULATION
C         COMLN: COMBINATORIAL PART CONTRIBUTION TO ACTIVITY COEFFICIENT
C         RESLN: RESIDUAL PART CONTRIBUTION TO ACTIVITY COEFFICIENT
```

```
C
C
C      SUBROUTINE REG1
C          X: PARAMETER FOR LINEAR REGRESSION
C          Y: FUNCTION VALUES FOR LINEAR REGRESSION
C          N: NUMBER OF VLAUES FOR LINEAR REGRESSION
C          SUMX,SUMY: SUMMATION OF X AND Y VALUES
C          XM,YM: MEAN VALUE OF X AND Y
C          SXX,SYY,SXY: LEAST SQUARE ESTIMATION VALUE OF X-X, Y-Y AND X-Y
C          SLP: SLOPE OF THE LINEAR LEAST SQUARE REGRESSION
C          B: INTERCEPT OF THE LINEAR SQUARE REGRESSION
C
C
C      SUBROUTINE REG2
C          X,Y: PARAMETERS FOR THE LEAST SQUARE REGRESSION
C          Z: FUNCTIONAL VALUE FOR LEAST SQUARE REGRESSION
C          XM,YM,ZM: MEAN VALUES OF X, Y AND Z
C          C1,C2: REGRESSION COEFFICIENTS
C
C
C      SUBROUTINE ADS
C          NCADS: NUMBER COUNT OF ADSORPTION PARTITION COEFFICIENT ESTIMATIONS
C          X: MOLE FRACTION OF SOLVENT
C          V: VOLUME FRACTION OF SOLVENT
C          XU: MOLE FRACTION SOLUTE SOLUBILITY
C          PKOCLG: LOG OF PARTITION COEFFICIENT BASED ON ORGANIC CARBON
C          PKOC: SORPTION PARTITION COEFFICIENT BASED ON ORGANIC CARBON
C          KPW: SORPTION PARTITION COEFFICIENT IN PURE WATER
C          PKOC: SORPTION PARTITION COEFFICIENT IN SOLVENT/WATER SYSTEMS
C          SIGMA: SOLUTE SOLUBILITY LINEAR REGRESSION SLOPE
C
C**********************************************************************

C     MAIN PROGRAM
C     THIS PROGRAM ESTIMATES SOLUTE SOLUBILITY
C      IN SOLVENT/WATER MIXTURES AND ESTIMATES THE ADSORPTION
C      PARTITION COEFFICIENT OF THE SOLUTE IN SOILS/SEDIMENTS.
C      DIMENSION XMOL(3,50),COMLN(11),RESLN(11),
C     1AL(11),RI(11),QI(11),QGPI(6,11),QGP(11),PARM(11,11),ID(11),
C     3GSUM(11),G(11),TNAM(11)
      DIMENSION IDGP(50,3),NOGP(50,3),ALGAC(3,50),ACT(3,50)
C      DIMENSION NGM(3),IGP(50,3),NGP(50,3),ACLG(3,50),AC(3,50)
      DIMENSION NGMOL(3)
      COMMON /A/MOL(2,3),MW1,MW2,MW3,D1,D2,D3,VM1,VM2,VM3,VSOV,XSOV
      COMMON /B/NCNLL,FLAG3,V1(50),XLL(50),S2(50),XL(50),J2
      COMMON /C/NCNUNI,FLAG4,FLAG7,LSTA,HF,TM,TEMP
      COMMON /D/NMG,NSG,RR(100),QQ(100),NKTAB(100),GNAM(100)
      COMMON /DD/KODE(50,50),PARAM(50,50)
      COMMON /E/NCNEXF,FLAG5,PC,X0,X100
      COMMON /F/NCNMSA,FLAG6,TSA,PSA,HSA,DEH,DEP
      COMMON /G/FILIN,FILSET,FILOUT
      COMMON /I/NCNADS,OC,KPW,PKOC,KP,FLAG8
      REAL MW1,MW2,MW3,NOGP,NGP,OC,KPW,KP
      CHARACTER*10 FILSET,FILIN,FLAG10
      CHARACTER*10 MOL,FLAG1,FLAG2,FLAG3,FLAG4,FLAG5,FLAG6,FLAG8
      CHARACTER*10 LNK,FILOUT,GNAM,TNAM,ANS,POTER,BLK,FLAG7

      BLK=' '
C     START THE PROGRAM
      WRITE(*,1011)
1011  FORMAT(1X,T15,41('*'),//,1X,T15,
     *'*',15X,'AROSOL',T55,'*',/,1X,T15,
     *'*',5X,'Aromatic Solute Solubility',T55,'*',/,1X,T15,
     *'*',5X,'in Solvent/Water Mixtures',T55,'*',/,1X,T15,
     *'*',T55,'*',/,1X,T15,
     *'*',15X,'by',T55,'*',/,1X,T15,
     *'*',10X,'Jaw-Kwei Fu',T55,'*',/,1X,T15,
     *'*',10X,'Charles Brooks',T55,'*',/,1X,T15,
     *'*',10X,'Richard G. Luthy',T55,'*',/,1X,T15,
     *'*',5X,'Carnegie-Mellon University',T55,'*',/,1X,T15,
     *'*',5X,'Pittsburgh, Pennsylvania',T55,'*',/,1X,T15,
     *'*',T55,'*',/,1X,T15,
     *'*',10X,'August, 1986',T55,'*',//,1X,T15,41('*'),///,1X
     * 'Hit RETURN key to continue')

      READ (*,112)POTER
112   FORMAT(A10)
      IF(POTER.EQ.BLK)GO TO 2011
2011  CONTINUE

      WRITE(*,1012)
1012  FORMAT(1X,
     *' This program estimates aromatic solute solubility in',/,1x,
     *' solvent/water mixtures.  This program is designed to',/,1x,
     *' utilize different levels of input parameters to estimate',/,1x,
     *' aromatic solute solubility. This program employes four',/,1x,
     *' approaches: LOG-LINEAR, UNIFAC, EXCESS FREE ENERGY, and',/,1x,
     *' MOLECULAR SURFACE AREA. The input parameters can be read',/,1x,
     *' from either an existing input file or from the terminal.',/,1x,
     *' The program estimates solute solubility via approaches',/,1x,
     *' specified by the user, and stores the results into an',/,1X,
     *' output file.'///,1x,' Hit RETURN key to continue')

      READ (*,112)POTER
      IF(POTER.EQ.BLK)GO TO 111
111   CONTINUE


C     DETERMINE AN EXISTING INPUT DATA FILE
      WRITE(*,'(A)')' DO YOU HAVE INPUT FILE? (Y OR N)'
      READ(*,1001)FLAG1
1001  FORMAT(A)
      IF(FLAG1.EQ.'y'.or.FLAG1.EQ.'Y')GO TO 3
      WRITE(*,'(A)')' DO YOU WANT TO SET UP AN INPUT FILE? (Y OR N)'
      READ(*,1001)FLAG2
      IF(FLAG2.EQ.'n'.or.FLAG2.EQ.'N')GO TO 2

C     INPUT FILE NAME WHICH ALL THE INPUT DATA TO BE STORED
      WRITE(*,'(A)')
     1' GIVE THE FILE NAME IN WHICH INPUT DATA ARE TO BE STORED'
      READ(*,1002)FILSET
1002  FORMAT(A10)
      GO TO 20

C     READ INPUT DATA FROM EXISTING DATA FILE
3     WRITE(*,'(A)')' INPUT FILE NAME='
      READ(*,1002)FILIN
      CALL INPUT(NSPS,NGMOL,NOGP,IDGP)
      IF(NSPS.GT.1)GO TO 5
      WRITE(*,1003)
      GO TO 10
2     FILSET='FILE.TEM'

C     READ INPUT DATA FROM TERMINAL AND STORE INTO THE SPECIFIED FILE
20    CALL SETUP(NSPS,NGMOL,NOGP,IDGP)
      IF(NSPS.GT.1)GO TO 5
      WRITE(*,1003)
1003  FORMAT(' ONLY ONE COMPONENT  CAN NOT ESTIMATE SOLUBILITY')
```

```
      GO TO 10
5     WRITE(*,1004)
1004  FORMAT(' GIVE THE OUTPUT FILE NAME IN WHICH THE SOLUBILITY',/,
     *' CALCULATION DATA WILL BE STORED')
      READ(*,1002)FILOUT
      OPEN(31,FILE=FILOUT,STATUS='NEW')

      WRITE(31,2110)
2110  FORMAT(/,1X,75('*'),/)

      WRITE (31,5001)MOL(1,2),MOL(2,2)
5001  FORMAT(' SOLUTE USED IN THE CALCULATION IS ',T40,2A10)
      WRITE (31,5002)MOL(1,1),MOL(2,1)
5002  FORMAT(' SOLVENT USED IN THE CALCULATION IS ',T40,2A10,/)
      WRITE (31,5003)MOL(1,1),MOL(2,1),MOL(1,2),MOL(2,2)
5003  FORMAT(1X,'PARAMETER',T22,2A10,T40,2A10,T65,'WATER')
      WRITE(31,5004)MW1,MW2,MW3
5004  FORMAT(1X,'MOLECULAR WEIGHT',T20,F10.2,T40,F10.2,T60,F10.2)
      WRITE(31,5005)D1,D2,D3
5005  FORMAT(1X,'DENSITY',T20,F10.4,T40,F10.4,T60,F10.4)
      WRITE(31,5006)VM1,VM2,VM3
5006  FORMAT(1X,'MOLAR VOLUME',T20,F10.2,T40,F10.2,T60,F10.2,/)

      IF(FLAG4.NE.'Y'.OR.FLAG4.NE.'y')GO TO 5007
      IF(LSTA.EQ.1)GO TO 5008
      WRITE(31,5009)LSTA
5009  FORMAT(' SOLUTE IS LIQUID AT SYSTEM TEMPERATURE')
      GO TO 5011
5008  WRITE(31,5010)LSTA
5010  FORMAT(' SOLUTE IS SOLID AT SYSTEM TEMPERATURE')
5011  WRITE(31,5012)HF,TM,TEMP
5012  FORMAT(' HEAT OF FUSION [CAL/MOLE]',T40,F10.2,/,
     *   ' SOLUTE MELTING TEMPERATURE [K]',T40,F10.2,/,
     *   ' SYSTEM TEMPERATURE [K]',T40,F10.2)
5007  IF(FLAG5.EQ.'Y'.or.FLAG5.EQ.'y')THEN
      XPC=10**PC
      WRITE(31,5013)XPC
5013  FORMAT(' OCTANOL/WATER PARTITION COEFFICIENT',E10.3)
      ENDIF

C     CALCULATE SOLUTE SOLUBILITY
      CALL SOLCAL(NSPS,NGMOL,NOGP,IDGP)

10    STOP
      END


      SUBROUTINE SETUP(NSPS,NGMOL,NOGP,IDGP)
C      DIMENSION XMOL(3,50),COMLN(11),RESLN(11),
C     1AL(11),RI(11),QI(11),QGPI(6,11),QGP(11),PARM(11,11),ID(11),
C     3GSUM(11),G(11),TNAM(11)
      DIMENSION IDGP(50,3),NOGP(50,3),ALGAC(3,50),ACT(3,50)
C      DIMENSION NGM(3),IGP(50,3),NGP(50,3),ACLG(3,50),AC(3,50)
      DIMENSION NGMOL(3)
      COMMON /A/MOL(2,3),MW1,MW2,MW3,D1,D2,D3,VM1,VM2,VM3,VSOV,XSOV
      COMMON /B/NCNLL,FLAG3,V1(50),XLL(50),S2(50),XL(50),J2
      COMMON /C/NCNUNI,FLAG4,FLAG7,LSTA,HF,TM,TEMP
      COMMON /D/NMG,NSG,RR(100),QQ(100),NKTAB(100),GNAM(100)
      COMMON /DD/KODE(50,50),PARAM(50,50)
      COMMON /E/NCNEXF,FLAG5,PC,X0,X100
      COMMON /F/NCNMSA,FLAG6,TSA,PSA,HSA,DEH,DEP
      COMMON /G/FILIN,FILSET,FILOUT
      COMMON /I/NCNADS,OC,KPW,PKOC,KP,FLAG8
      REAL MW1,MW2,MW3,NOGP,NGP,OC,KPW,KP
      CHARACTER*10 FILIN,FILSET,FLAG7
      CHARACTER*10 MOL,FLAG1,FLAG2,FLAG3,FLAG4,FLAG5,FLAG6,FLAG8
      CHARACTER*10 LNK,FILOUT,GNAM,TNAM,ANS,ITEM
      OPEN(28,FILE=FILSET,STATUS='NEW')
      OPEN(25,FILE='UNIFAC.DAT',STATUS='OLD')


      LNK = ' '
      NSPS=3
      J1 = 1
      DO 35 J1=1,2
      WRITE(*,1003)J1
1003  format(' INPUT THE NAME OF COMPONENT ',I2,' IN 20 CHARACTERS',
     1' 1-SOLVENT, 2-SOLUTE')
      READ (*,1004)MOL(1,J1),MOL(2,J1)
1004  FORMAT(2A10)
      WRITE (28,1004)MOL(1,J1),MOL(2,J1)
35    CONTINUE
      MOL(1,3)='WATER'
      MOL(2,3)=LNK

      WRITE(*,'(A)')' INPUT MOLECULAR WEIGHT OF SOLVENT, SOLUTE, WATER'
      READ(*,*)MW1,MW2,MW3
      WRITE(28,*)MW1,MW2,MW3
      WRITE(*,'(A)')' INPUT DENSITIES OF SOLVENT, SOLUTE AND WATER'
      READ(*,*)D1,D2,D3
      WRITE(28,*)D1,D2,D3
      VM1=MW1/D1
      VM2=MW2/D2
      VM3=MW3/D3

      J2=1
      WRITE(*,1011)
1011  FORMAT(' INPUT KNOWN SOLUTE SOLUBILITY IN % SOLVENT,'
     *' AND SOLUTE SOLUBILITY IN MG/L,',/,' FINISHED AS -1 -1, ',
     *' DATA INPUT IN PAIRS WITH ONE PAIR PER LINE')
4     READ(*,*)V1(J2),S2(J2)
      WRITE(28,*)V1(J2),S2(J2)
      TEM=(((1000-(S2(J2)/(1000*D2)))*(V1(J2)/100)*D1)/MW1)
      TEM=TEM+S2(J2)/(MW2*1000)
      TEM=TEM+(((1000-(S2(J2)/(D2*1000)))*(1-(V1(J2)/100))*D3)/MW3)
      XLL(J2)=(S2(J2)/(MW2*1000))/TEM
      IF(V1(J2).LT.0.0)GO TO 5
      J2=J2+1
      GO TO 4
5     J2=J2-1
      DO 2 I=1,J2
      IF(V1(I).EQ.0)X0=XLL(I)
      IF(V1(I).EQ.100.)X100=XLL(I)
2     CONTINUE
      WRITE(*,1012)
1012  FORMAT(' DO YOU WANT TO ESTIMATE SOLUTE SOLUBILITY BY '
     1' THE LOG-LINEAR APPROACH?',/,'  (Y OR N)')
      READ(*,1)FLAG3
      WRITE(28,1)FLAG3
      IF(FLAG3.NE.'Y'.OR.FLAG3.NE.'y')GO TO 8
      IF(J2.GT.1) GO TO 8
      WRITE(*,1013)
1013  FORMAT(' ONLY ONE KNOWN SOLUTE SOLUBILITY, CAN NOT'
     1' USE LOG-LINEAR APPROACH')
      FLAG3='N'

8     WRITE(*,1014)
1014  FORMAT(' DO YOU WANT TO ESTIMATE SOLUTE SOLUBILITY BY '
```

```
     1'THE UNIFAC APPROACH? (Y OR N)')
      READ(*,1)FLAG4
      WRITE(28,1)FLAG4

      WRITE(*,1015)
1015  FORMAT(' DO YOU WANT TO ESTIMATE SOLUTE SOLUBILITY BY '
     1'THE EXCESS FREE ENERGY APPROACH?',/,' (Y OR N)')
      READ(*,1)FLAG5
      WRITE(28,1)FLAG5

      WRITE(*,1017)
1017  FORMAT(' DO YOU WANT TO ESTIMATE SOLUTE SOLUBILITY BY',/,
     1' THE MOLECULAR SURFACE AREA APPROACH? (Y OR N)')
      READ(*,1)FLAG6
      WRITE(28,1)FLAG6
1     FORMAT(A10)

      WRITE(*,1020)
1020  FORMAT(' DO YOU WANT TO ESTIMATE ADSORPTION PARTITION '
     1'COEFFICIENT? (Y OR N)')
      READ(*,1)FLAG8
      WRITE(28,1)FLAG8

      FLAG7='N'
      IF(FLAG4.EQ.'y'.OR.FLAG4.EQ.'Y')FLAG7='Y'
      IF(FLAG4.EQ.'N'.AND.FLAG5.EQ.'N')GO TO 7
      IF(FLAG4.EQ.'n'.AND.FLAG5.EQ.'n')GO TO 7
      IF(X0.NE.0)GO TO 21
      FLAG4='Y'
21    IF(X100.NE.0)GO TO 22
      FLAG4='Y'
22    IF(FLAG4.EQ.'n'.AND.FLAG5.EQ.'y')GO TO 11
      IF(FLAG4.EQ.'N'.AND.FLAG5.EQ.'Y')GO TO 11
      IF(FLAG4.EQ.'n'.AND.FLAG5.EQ.'y')GO TO 11
      IF(FLAG4.EQ.'N'.AND.FLAG5.EQ.'Y')GO TO 11

C      WRITE(*,'(A)')'  IS SOLUTE A SOLID(1) OR LIQUID(2)'
C      READ(*,*)LSTA
C      WRITE(28,*)LSTA
      WRITE(*,1016)
1016  FORMAT(' ENTER SOLUTE HEAT OF FUSION (CAL/MOLE), SOLUTE MELTING'
     *' TEMPERATURE (K),',1X,' AND',/,' SYSTEM TEMPERATURE (K),',
     *' IF HEAT OF FUSION IS NOT AVAILABLE USE 0 AS THE VALUE')
      READ(*,*)HF,TM,TEMP
      IF(HF.EQ.0)HF=13*TM
      WRITE(28,*)HF,TM,TEMP

      IF(TM.GE.TEMP)THEN
         LSTA=1
        ELSE
         LSTA=2
      END IF

C*****************************************************************C
C     READ DATA BANK.                                             C
C*****************************************************************C
11    READ (25,4001) NMG,NSG
      DO 13 I=1,NSG
      READ(25,4002)RR(I),QQ(I),NKTAB(I),GNAM(I)
13    CONTINUE
C      WRITE(*,'(A)')'  FIRST READ ENCOUNTERED'
      DO 12 I=1,NMG
      READ(25,4003)(KODE(I,J),PARAM(I,J),J=1,NMG)
12     CONTINUE
C      WRITE(*,'(A)')'  SECOND READ ENCOUNTERED'
      CLOSE(25,STATUS='KEEP')
 4001 FORMAT (2I2)
 4002 FORMAT(F6.4,1X,F5.3,1X,I2,A10)
 4003 FORMAT(7(I1,F9.3))
      WRITE(*,'(A)')
     1' DO YOU WANT TO SEE THE UNIFAC SECONDARY GROUP LISTING?(Y OR N)'
      READ(*,1)ITEM
      IF(ITEM.EQ.'N'.OR.ITEM.EQ.'n') GO TO 45
      WRITE(*,4004)(J,GNAM(J),J=1,NSG)
4004  FORMAT(5(1X,I2,1X,A10,2X))
45    DO 40 J = 1,2
      WRITE(*,1006)MOL(1,J),MOL(2,J)
1006  FORMAT('  INPUT THE NUMBER OF SECONDARY GROUPS IN COMPONENT ',
     12A10)
      READ (*,*) NGMOL(J)
      WRITE(28,*) NGMOL(J)
40    CONTINUE
      NGMOL(3)=1
      NOGP(1,3)=1
      IDGP(1,3)=17
      DO 50 J=1,2
      NN = NGMOL(J)
      WRITE(*,1005) MOL(1,J),MOL(2,J)
1005  FORMAT('  INPUT NUMBER OF TIMES SECONDARY GROUP i APPEARS IN'
     *' COMPONENT',2X,2A10/,
     *'  REPEAT n TIMES UNTIL n = NUMBER OF SECONDARY GROUPS')
      READ (*,*) (NOGP(I,J),IDGP(I,J),I=1,NN)
50    WRITE(28,*) (NOGP(I,J),IDGP(I,J),I=1,NN)

6     IF(FLAG5.EQ.'n'.or.FLAG5.EQ.'N')GO TO 7
      WRITE(*,1007)
1007  FORMAT('  INPUT LOG OCTANOL/WATER PARTITION COEFFICIENT',/,
     *'  OF SOLUTE FOR THE EXCESS FREE ENERGY CALCULATION')
      READ(*,*)PC
      WRITE(28,*)PC

7     IF(FLAG6.EQ.'n'.or.FLAG6.EQ.'N')GO TO 80
      WRITE(*,1018)
1018  FORMAT(' INPUT SOLUTE HYDROPHOBIC SURFACE AREA AND POLAR SURFACE'
     *' AREA.',/,' INPUT IN UNITS A**2')
      READ(*,*)HSA,PSA
      WRITE(28,*)HSA,PSA
      WRITE(*,1019)
1019  FORMAT(' INPUT SOLVENT MICROSCOPIC INTERFACIAL FREE ENERGY'
     *' BETWEEN HSA AND PSA,',/,' INPUT IN UNITS OF DYNE/CM**2')
      READ(*,*)DEH,DEP
      WRITE(28,*)DEH,DEP

80    IF(FLAG8.EQ.'n'.or.FLAG8.EQ.'N')GO TO 90
      WRITE(*,'(A)')' INPUT ORGANIC CARBON CONTENT OF ADSORBENT, IN %'
      READ(*,*)OC
      WRITE(28,*)OC

90    IF(FILSET.EQ.'FILE.TEM')CLOSE(28,STATUS='DELETE')
      CLOSE(28,STATUS='KEEP')

      RETURN
      END



      SUBROUTINE INPUT(NSPS,NGMOL,NOGP,IDGP)
```

```
C      DIMENSION XMOL(3,50),COMLN(11),RESLN(11),
C     1AL(11),RI(11),QI(11),QGPI(6,11),QGP(11),PARM(11,11),ID(11),
C     3GSUM(11),G(11),TNAM(11)
      DIMENSION IDGP(50,3),NOGP(50,3),ALGAC(3,50),ACT(3,50)
C      DIMENSION NGM(3),IGP(50,3),NGP(50,3),ACLG(3,50),AC(3,50)
      DIMENSION NGMOL(3)
      COMMON /A/MOL(2,3),MW1,MW2,MW3,D1,D2,D3,VM1,VM2,VM3,VSOV,XSOV
      COMMON /B/NCNLL,FLAG3,V1(50),XLL(50),S2(50),XL(50),J2
      COMMON /C/NCNUNI,FLAG4,FLAG7,LSTA,HF,TM,TEMP
      COMMON /D/NMG,NSG,RR(100),QQ(100),NKTAB(100),GNAM(100)
      COMMON /DD/KODE(50,50),PARAM(50,50)
      COMMON /E/NCNEXF,FLAG5,PC,X0,X100
      COMMON /F/NCNMSA,FLAG6,TSA,PSA,HSA,DEH,DEP
      COMMON /G/FILIN,FILSET,FILOUT
      COMMON /I/NCNADS,OC,KPW,PKOC,KP,FLAG8
      REAL MW1,MW2,MW3,NOGP,NGP,OC,KPW,KP
      CHARACTER*10 MOL,FLAG1,FLAG2,FLAG3,FLAG4,FLAG5,FLAG6,FLAG8
      CHARACTER*10 LNK,FILOUT,FILIN,FILSET,GNAM,FLAG7

      OPEN(21,FILE=FILIN,STATUS='OLD')
      OPEN(25,FILE='UNIFAC.DAT',STATUS='OLD')

      LNK = ' '
      NSPS=3
      DO 35 J1=1,2
      READ(21,1004)MOL(1,J1),MOL(2,J1)
1004  FORMAT(2A10)
35    CONTINUE
      MOL(1,3)='WATER'
      MOL(2,3)=LNK

      READ(21,*)MW1,MW2,MW3
      READ(21,*)D1,D2,D3
      VM1=MW1/D1
      VM2=MW2/D2
      VM3=MW3/D3

      J2=1
 4    READ(21,*)V1(J2),S2(J2)
      TEM=(((1000-(S2(J2)/(1000*D2)))*V1(J2)*D1/100)/MW1)
      TEM=TEM+S2(J2)/(MW2*1000)
      TEM=TEM+(((1000-(S2(J2)/(D2*1000)))*(1-V1(J2)/100)*D3)/MW3)
      XLL(J2)=(S2(J2)/(MW2*1000))/TEM
      IF(V1(J2).LT.0.0)GO TO 5
      J2=J2+1
      GO TO 4
5     J2=J2-1
      DO 2 I=1,J2
      IF(V1(I).EQ.0)X0=XLL(I)
      IF(V1(I).EQ.100.)X100=XLL(I)
2     CONTINUE
      READ(21,1)FLAG3
      IF(FLAG3.NE.'Y'.OR.FLAG3.NE.'y')GO TO 8
      IF(J2.GT.1) GO TO 8
      WRITE(*,'(A)')' ONLY ONE KNOWN SOLUTE SOLUBILITY, CAN NOT
     1 USE LOG-LINEAR APPROACH'
      FLAG3='N'

8     READ(21,1)FLAG4
      READ(21,1)FLAG5
      READ(21,1)FLAG6
      READ(21,1)FLAG8
1     FORMAT(A10)
      FLAG7='N'
      IF(FLAG4.EQ.'y'.OR.FLAG4.EQ.'Y')FLAG7='Y'
      IF(FLAG4.EQ.'N'.AND.FLAG5.EQ.'N')GO TO 7
      IF(FLAG4.EQ.'n'.AND.FLAG5.EQ.'n')GO TO 7
      IF(X0.NE.0)GO TO 21
      FLAG4='Y'
21    IF(X100.NE.0)GO TO 22
      FLAG4='Y'
22    IF(FLAG4.EQ.'n'.AND.FLAG5.EQ.'y')GO TO 11
      IF(FLAG4.EQ.'N'.AND.FLAG5.EQ.'Y')GO TO 11

C      READ(21,*)LSTA
      READ(21,*)HF,TM,TEMP
      IF(HF.EQ.0.)HF=13.0*TM
      IF(TM.GE.TEMP)THEN
         LSTA=1
        ELSE
         LSTA=2
      END IF

C*********************************************************************C
C     READ DATA BANK.                                                 C
C*********************************************************************C
11    READ (25,4001) NMG,NSG
      DO 13 I=1,NSG
      READ(25,4002)RR(I),QQ(I),NKTAB(I),GNAM(I)
13    CONTINUE
C      WRITE(*,'(A)')'  FIRST READ ENCOUNTERED'
      DO 12 I=1,NMG
      READ(25,4003)(KODE(I,J),PARAM(I,J),J=1,NMG)
12    CONTINUE
C      WRITE(*,'(A)')'  SECOND READ ENCOUNTERED'
      CLOSE(25,STATUS='KEEP')
 4001 FORMAT (2I2)
 4002 FORMAT(F6.4,1X,F5.3,1X,I2,A10)
 4003 FORMAT(7(I1,F9.3))
      DO 40 J = 1,2
      READ (21,*) NGMOL(J)
40    CONTINUE
      NGMOL(3)=1
      NOGP(1,3)=1
      IDGP(1,3)=17
      DO 50 J=1,2
      NN = NGMOL(J)
50    READ (21,*)  (NOGP(I,J),IDGP(I,J),I=1,NN)

6     IF(FLAG5.EQ.'n'.or.FLAG5.EQ.'N')GO TO 7
      READ(21,*)PC

7     IF(FLAG6.EQ.'n'.or.FLAG6.EQ.'N')GO TO 80
      READ(21,*)HSA,PSA
      READ(21,*)DEH,DEP


80    IF(FLAG8.EQ.'n'.or.FLAG8.EQ.'N')GO TO 90
      READ(21,*)OC

90    CLOSE(21,STATUS='KEEP')
      RETURN
      END


      SUBROUTINE SOLCAL(NSPS,NGMOL,NOGP,IDGP)
      DIMENSION XMOL(3,50),VADS(100),ADSCOE(100)
      DIMENSION IDGP(50,3),NOGP(50,3),ALGAC(3,50),ACT(3,50)
```

```
      DIMENSION NGM(3),IGP(50,3),NGP(50,3),ACLG(3,50),AC(3,50)
      DIMENSION NGMOL(3),NGMTEM(3),NOGTEM(50,3),IDGTEM(50,3)
      COMMON /A/MOL(2,3),MW1,MW2,MW3,D1,D2,D3,VM1,VM2,VM3,VSOV,XSOV
      COMMON /B/NCNLL,FLAG3,V1(50),XLL(50),S2(50),XL(50),J2
      COMMON /C/NCNUNI,FLAG4,FLAG7,LSTA,HF,TM,TEMP
      COMMON /D/NMG,NSG,RR(100),QQ(100),NKTAB(100),GNAM(100)
      COMMON /DD/KODE(50,50),PARAM(50,50)
      COMMON /E/NCNEXF,FLAG5,PC,X0,X100
      COMMON /F/NCNMSA,FLAG6,TSA,PSA,HSA,DEH,DEP
      COMMON /H/XLOG,SLOG,XFE,SFE,XMSA,SMSA
      COMMON /I/NCNADS,OC,KPW,PKOC,KP,FLAG8
C     COMMON /J/X10,X20,X30
      REAL MW1,MW2,MW3,NOGP,NGP,OC,KPW,KP
      CHARACTER*10 MOL,FLAG1,FLAG2,FLAG3,FLAG4,FLAG5,FLAG6,FLAG8
      CHARACTER*10 LNK,FILOUT,GNAM,TNAM,ANS,FLAG7

      WRITE(31,2111)
2111  FORMAT(/,1X,75('*'),//,
     *' SOLUTE SOLUBILITY PREDICTIONS IN MIXED SOLVENT SYSTEM',/)
      WRITE(31,5001)
5001  FORMAT(1X,'VOLUME',T11,'LOG-LINEAR',T28,'UNIFAC',T41,
     *'EXCESS FREE',T57,'MOLECULAR SURFACE',/,
     *1X,'SOLVENT',T11,'APPROACH',T28,'APPROACH',T41,
     *'ENERGY APPROACH',T57,'AREA APPROACH',/,
     *T9,'MOLE',T17,'SOL.',T25,'MOLE',T33,'SOL.',T41,'MOLE'
     *,T49,'SOL.',T57,'MOLE',T65,'SOL.',/,
     *3X,'%',T12,'[-]',T17,'[MG/L]',T28,'[-]',T33,'[MG/L]',
     *T44,'[-]',T49,'[MG/L]',T60,'[-]',T65,'[MG/L]',/)

      VSOV=0
      XLOG=0
      SLOG=0
      XUNI=0
      SUNI=0
      XFE=0
      SFE=0
      XMSA=0
      SMSA=0

      NCNLL=1
      NCNUNI=1
      NCNEXF=1
      NCNMSA=1
      NCNADS=1
1     WRITE(*,1011)
1011  FORMAT(' INPUT PERCENT VOLUME SOLVENT IN THE MIXTURE'
     1' TO BE EVALUATED')
      READ(*,*)VSOV
      XSOV=(D1*VSOV/MW1)/((D1*VSOV/MW1)+(D3*(100-VSOV)/MW3))
      IF(FLAG3.EQ.'n'.OR.FLAG3.EQ.'N')GO TO 5
      CALL LOGLNR(K1,SLP1)
      NCNLL=NCNLL+1

5     IF(FLAG4.EQ.'n'.AND.FLAG7.EQ.'n')GO TO 10
      IF(FLAG4.EQ.'N'.AND.FLAG7.EQ.'N')GO TO 10
      IF(FLAG4.EQ.'n'.AND.FLAG7.EQ.'y')GO TO 29
      IF(FLAG4.EQ.'N'.AND.FLAG7.EQ.'Y')GO TO 29
      VTEM=VSOV
      XTEM=XSOV
      IF(FLAG5.EQ.'N'.OR.FLAG5.EQ.'n')GO TO 27
      IF(X0.NE.0)GO TO 26
      VSOV=0.0
      XSOV=0.0
      CALL UNIEST(ACLG,AC,XUNI,SUNI,NSPS,NGMOL,NOGP,IDGP)
      X0=XUNI
      NCNUNI=NCNUNI+1
26    IF(X100.NE.0)GO TO 27
      VSOV=100.0
      XSOV=1.0
      CALL UNIEST(ACLG,AC,XUNI,SUNI,NSPS,NGMOL,NOGP,IDGP)
      X100=XUNI
      NCNUNI=NCNUNI+1
C27   VSOV=10.0
C     XSOV=(D1*VSOV/MW1)/((D1*VSOV/MW1)+(D3*(100-VSOV)/MW3))
C     CALL UNIEST(ACLG,AC,XUNI,SUNI,NSPS,NGMOL,NOGP,IDGP)
C     XUNI10=XUNI
C     NCNUNI=NCNUNI+1
C     VSOV=20.0
C     XSOV=(D1*VSOV/MW1)/((D1*VSOV/MW1)+(D3*(100-VSOV)/MW3))
C     CALL UNIEST(ACLG,AC,XUNI,SUNI,NSPS,NGMOL,NOGP,IDGP)
C     XUNI20=XUNI
C     NCNUNI=NCNUNI+1
C     VSOV=30.0
C     XSOV=(D1*VSOV/MW1)/((D1*VSOV/MW1)+(D3*(100-VSOV)/MW3))
C     CALL UNIEST(ACLG,AC,XUNI,SUNI,NSPS,NGMOL,NOGP,IDGP)
C     XUNI30=XUNI
C     NCNUNI=NCNUNI+1
27    IF(FLAG7.EQ.'N'.OR.FLAG7.EQ.'n')GO TO 28
      VSOV=VTEM
      XSOV=XTEM

29    CALL UNIEST(ACLG,AC,XUNI,SUNI,NSPS,NGMOL,NOGP,IDGP)
      NCNUNI=NCNUNI+1

28    IF(FLAG7.EQ.'N'.OR.FLAG7.EQ.'n')GO TO 10

      WRITE(*,2007)
C     WRITE(31,2007)
      WRITE(*,2008)MOL(1,2),MOL(2,2),XMOL(1,1),ACLG(1,1),AC(1,1)
C     WRITE(31,2008)MOL(1,2),MOL(2,2),XMOL(1,1),ACLG(1,1),AC(1,1)
      WRITE(*,2008)MOL(1,1),MOL(2,1),XSOV,ACLG(2,1),AC(2,1)
C     WRITE(31,2008)MOL(1,1),MOL(2,1),XSOV,ACLG(2,1),AC(2,1)
 2007 FORMAT (/,'   UNIFAC ACTIVITY COEFFICIENT ESTIMATION',/,
     15X,'COMPONENT',9X,'MOLE FRAC',2X,'LN ACTCF',7X,
     1      'ACTCF'/)
 2008 FORMAT(4X,2A10,1X,F10.4,1X,F10.5,1X,F13.4)

      WRITE(*,1001)MOL(1,1),MOL(2,1),VSOV,MOL(1,2),MOL(2,2),XUNI,
     1MOL(1,2),MOL(2,2),SUNI
C     WRITE(31,1001)MOL(1,1),MOL(2,1),VSOV,MOL(1,2),MOL(2,2),XUNI,
C     1MOL(1,2),MOL(2,2),SUNI
1001  FORMAT(//1X,T5,'UNIFAC ESTIMATION',/,1X,T10,2(A10),
     1'FRACTION [% VOL] ',F10.2,/,1X,T10,2(A10),
     2'SOLUBILITY [MOLE FRACTION] ',E10.3,/,1X,T10,2(A10),
     3'SOLUBILITY [MG/L] ',E10.3)

10    IF(FLAG5.EQ.'n'.OR.FLAG5.EQ.'N')GO TO 15
      CALL EXFREN(NSPS,NGMOL,NOGP,IDGP)
      NCNEXF=NCNEXF+1

15    IF(FLAG6.EQ.'n'.OR.FLAG6.EQ.'N')GO TO 20
      CALL MSA
      NCNMSA=NCNMSA+1

20    IF(FLAG8.EQ.'n'.OR.FLAG8.EQ.'N')GO TO 30
      CALL ADS(NSPS,NGMOL,NOGP,IDGP,SLP1)
      VADS(NCNADS)=VSOV
      ADSCOF(NCNADS)=KP
```

```
      NCNADS=NCNADS+1


30    WRITE(31,5002)VSOV,XLOG,SLOG,XUNI,SUNI,XFE,SFE,XMSA,SMSA
5002  FORMAT(1X,F6.2,T9,E7.2,1X,E7.2,1X,E7.2,1X,E7.2,1X,E7.2,1X,
     *E7.2,1X,E7.2,1X,E7.2)

      WRITE(*,'(A)')' ANOTHER CALCULATION? (Y=1,N=2)'
      READ(*,*)NCAL
      IF(NCAL.EQ.1) THEN
         GO TO 1
      ELSE
         IF(FLAG8.EQ.'N'.OR.FLAG8.EQ.'n')GO TO 41
         WRITE(31,5003)MOL(1,1),MOL(2,1)
5003     FORMAT(///,1X,75('*'),//, '  SOLUTE SORPTION PARTITION '
     *'COEFFICIENT ESTIMATION',//,1X,2A10,T25,'OC OF ADSORBENT',
     *T50,'KP OF SOLUTE',/,1X,T5,'[%]',T30,'[%]',T53,'(MOLE/KG)')
         NCNADS=NCNADS-1
         DO 40 I=1,NCNADS
         WRITE(31,5004)VADS(I),OC,ADSCOE(I)
5004     FORMAT(1X,T5,F7.2,T28,F6.2,T52,F10.2)
40       CONTINUE
         RETURN
41     END IF

      END



      SUBROUTINE LOGLNR(K1,SLP1)
      DIMENSION XMOL(3,50)
C      DIMENSION XMOL(3,50),COMLN(11),RESLN(11),
C     1AL(11),RI(11),QI(11),QGPI(6,11),QGP(11),PARM(11,11),ID(11),
C     3GSUM(11),G(11),TNAM(11)
C      DIMENSION IDGP(50,3),NOGP(50,3),ALGAC(3,50),ACT(3,50)
C      DIMENSION NGM(3),IGP(50,3),NGP(50,3),ACLG(3,50),AC(3,50)
C      DIMENSION NGMOL(3)
      COMMON /A/MOL(2,3),MW1,MW2,MW3,D1,D2,D3,VM1,VM2,VM3,VSOV,XSOV
      COMMON /B/NCNLL,FLAG3,V1(50),XLL(50),S2(50),XL(50),J2
C      COMMON /C/NCNUNI,FLAG4,FLAG7,LSTA,HF,TM,TEMP
C      COMMON /D/NMG,NSG,RR(100),QQ(100),NKTAB(100),GNAM(100)
C      COMMON /DD/KODE(50,50),PARAM(50,50)
C      COMMON /E/NCNEXF,FLAG5,PC,X0,X100
C      COMMON /F/NCNMSA,FLAG6,TSA,PSA,HSA,DEH,DEP
      COMMON /H/XLOG,SLOG,XFE,SFE,XMSA,SMSA
      REAL MW1,MW2,MW3,K1,OC,KPW,KP
      CHARACTER*10 MOL,FLAG1,FLAG2,FLAG3,FLAG4,FLAG5,FLAG6
      CHARACTER*10 LNK,FILOUT,GNAM,TNAM,ANS

C  This is the log-linear approach
      IF(NCNLL.GT.1) GO TO 20
      DO 1 I=1,J2
      XL(I)=LOG10(XLL(I))
1     CONTINUE
      IF(J2.GT.2)THEN
         CALL REG1(V1,XL,J2,K1,SLP1)
         WRITE(*,1011)K1,SLP1
1011     FORMAT(' LOG-LINEAR REGRESSION INTERCEPT ',F10.4,
     *' SLOPE ',F10.5)
         GO TO 20
         ELSE
         SLP1=(XL(1)-XL(2))/(V1(1)-V1(2))
         K1=XL(1)-SLP1*V1(1)
         GO TO 20
      END IF
20    XLOG=10**(K1+SLP1*VSOV)
      TE6=(XSOV*(1-XLOG)*VM1)+(XLOG*VM2)+(1-XSOV)*(1-XLOG)*VM3
      SLOG=(XLOG*MW2*1000000.0)/TE6
C      WRITE(*,1001)VSOV,XLOG,SLOG
      WRITE(*,1001)VSOV,XLOG,SLOG
C      WRITE(31,1001)VSOV,XLOG,SLOG

1001  FORMAT(1X,T5,'LOG-LINEAR ESTIMATION METHOD',//,1X,T10,
     *'SOLVENT FRACTION [% VOL] ',F10.2,/,1X,T10,
     1'LOG LINEAR ESTIMATION SOLUBILITY [MOLE FRACTION] ',E10.3,/,1X
     2,T10,'LOG LINEAR ESTIMATION SOLUBILITY [MG/L] ',E10.3)
      RETURN
      END


      SUBROUTINE EXFREN(NSPS,NGMOL,NOGP,IDGP)
      DIMENSION XMOL(3,50)
C      DIMENSION XMOL(3,50),COMLN(11),RESLN(11),
C     1AL(11),RI(11),QI(11),QGPI(6,11),QGP(11),PARM(11,11),ID(11),
C     3GSUM(11),G(11),TNAM(11)
      DIMENSION IDGP(50,3),NOGP(50,3),ALGAC(3,50),ACT(3,50)
C      DIMENSION NGM(3),IGP(50,3),NGP(50,3),ACLG(3,50),AC(3,50)
      DIMENSION FRENG(100),RPA1(100),RPA2(100)
      DIMENSION NGMOL(3)
      COMMON /A/MOL(2,3),MW1,MW2,MW3,D1,D2,D3,VM1,VM2,VM3,VSOV,XSOV
C      COMMON /B/NCNLL,FLAG3,V1(50),XLL(50),S2(50),XL(50),J2
      COMMON /C/NCNUNI,FLAG4,FLAG7,LSTA,HF,TM,TEMP
      COMMON /D/NMG,NSG,RR(100),QQ(100),NKTAB(100),GNAM(100)
      COMMON /DD/KODE(50,50),PARAM(50,50)
      COMMON /E/NCNEXF,FLAG5,PC,X0,X100
C      COMMON /F/NCNMSA,FLAG6,TSA,PSA,HSA,DEH,DEP
      COMMON /H/XLOG,SLOG,XFE,SFE,XMSA,SMSA
      REAL MW1,MW2,MW3,NOGP,NGP,NUM
      CHARACTER*10 MOL,FLAG1,FLAG2,FLAG3,FLAG4,FLAG5,FLAG6,FLAG7
      CHARACTER*10 LNK,FILOUT,GNAM,TNAM,ANS

C  This is the excess free energy approach

      IF(NCNEXF.GT.1) GO TO 20
      C2=-2.69-1.22*PC
      NMIX=21
      XMOL(1,1)=1.00
      XMOL(2,1)=0.00
      XMOL(3,1)=0.00
      DO 2 I=2,NMIX
      XMOL(1,I)=XMOL(1,I-1)-0.05
      XMOL(2,I)=0.00
      XMOL(3,I)=1-XMOL(1,I)
2     CONTINUE
      CALL UNIFAC(NSPS,NOGP,IDGP,NGMOL,NMIX,XMOL,ALGAC,ACT)
      DO 10 I=1,NMIX
      Z1=(XMOL(1,I)*VM1)/((XMOL(1,I)*VM1)+(XMOL(3,I)*VM3))
      Z3=(XMOL(3,I)*VM3)/((XMOL(1,I)*VM1)+(XMOL(3,I)*VM3))
      FRENG(I)=(XMOL(1,I)*ALGAC(1,I))+XMOL(3,I)*ALGAC(3,I)
      RPA1(I)=Z1*Z3*Z3*(XMOL(1,I)*VM1+XMOL(3,I)*VM3)/VM1
10    RPA2(I)=Z1*Z1*Z3*(XMOL(1,I)*VM1+XMOL(3,I)*VM3)/VM3
      CALL REG2(RPA1,RPA2,FRENG,A13,A31,NMIX)

20    Z1=(XSOV*VM1)/((XSOV*VM1)+(1-XSOV)*VM3)
      Z3=((1-XSOV)*VM3)/((XSOV*VM1)+(1-XSOV)*VM3)
      TEM1=Z1*ALOG(X100)+Z3*ALOG(X0)
      TEM11=TEM1-A13*Z1*Z3*(2*Z1-1)*VM2/VM1
```

```
      XFE=EXP(TEM2)

      TEM3=(XSOV*(1-XFE)*VM1)+(XFE*VM2)+((1-XSOV)*(1-XFE)*VM3)
      SFE=(XFE*MW2*1000000.0)/TEM3
C      WRITE(*,1001)VSOV,XFE,SFE
      WRITE(*,1001)VSOV,XFE,SFE
C      WRITE(31,1001)VSOV,XFE,SFE
1001  FORMAT(1X,/1X,T5,'EXCESS FREE ENERGY ESTIMATION METHOD',/,
     11X,T10,'SOLVENT FRACTION [% VOL] ',F10.2,/,1X,T10,
     1'EXCESS FREE ENERGY ESTIMATION SOLUBILITY [MOLE FRACTION] ',
     2E10.3,/,1X,T10,'EXCESS FREE ENERGY ESTIMATION SOLUBILITY [MG/L]
     3 ',E10.3)
      RETURN
      END




      SUBROUTINE MSA
C      DIMENSION XMOL(3,50),COMLN(11),RESLN(11),
C     1AL(11),RI(11),QI(11),QGPI(6,11),QGP(11),PARM(11,11),ID(11),
C     3GSUM(11),G(11),TNAM(11)
C      DIMENSION IDGP(50,3),NOGP(50,3),ALGAC(3,50),ACT(3,50)
C      DIMENSION NGM(3),IGP(50,3),NGP(50,3),ACLG(3,50),AC(3,50)
C      DIMENSION FRENG(100),RPA1(100),RPA2(100)
C      DIMENSION NGMOL(3)
      COMMON /A/MOL(2,3),MW1,MW2,MW3,D1,D2,D3,VM1,VM2,VM3,VSOV,XSOV
      COMMON /B/NCNLL,FLAG3,V1(50),XLL(50),S2(50),XL(50),J2
      COMMON /C/NCNUNI,FLAG4,FLAG7,LSTA,HF,TM,TEMP
C      COMMON /D/NMG,NSG,RR(100),QQ(100),NKTAB(100),GNAM(100)
C      COMMON /DD/KODE(50,50),PARAM(50,50)
      COMMON /E/NCNEXF,FLAG5,PC,X0,X100
      COMMON /F/NCNMSA,FLAG6,TSA,PSA,HSA,DEH,DEP
      COMMON /H/XLOG,SLOG,XFE,SFE,XMSA,SMSA
      REAL MW1,MW2,MW3,K
      CHARACTER*10 MOL,FLAG1,FLAG2,FLAG3,FLAG4,FLAG5,FLAG6,FLAG7
      CHARACTER*10 LNK,FILOUT,GNAM,TNAM,ANS

C     This is Molecular surface area approach

      K=1.38E-16
      TEM=(VSOV/100)*(((DEH*HSA+DEP*PSA)*1.0E-16)/(K*TEMP))
      TEM=ALOG(X0)+TEM
      XMSA=EXP(TEM)
      TEM3=(XSOV*(1-XMSA)*VM1)+(XMSA*VM2)+((1-XSOV)*(1-XMSA)*VM3)
      SMSA=(XMSA*MW2*1000000.0)/TEM3
      WRITE(*,1001)VSOV,XMSA,SMSA
1001  FORMAT(1X,T5,'MOLECULAR SURFACE AREA APPROACH',//,1X,T10,
     *'SOLVENT FRACTION [% VOL] ',F10.2,/,1X,T10,
     1'MOLECULAR SURFACE AREA APPROACH SOLUBILITY [MOLE FRACTION] ',
     2E10.3,/,1X,T10,'MOLECULAR SURFACE AREA APPROACH SOLUBILITY '
     3'[MG/L] ',E10.3)
      RETURN
      END
```

```
      SUBROUTINE UNIEST(ACLG,AC,XUNI,SUNI,NSPS,NGMOL,NOGP,IDGP)

      DIMENSION XMOL(3,50)
C      DIMENSION XMOL(3,50),COMLN(11),RESLN(11),
C     1AL(11),RI(11),QI(11),QGPI(6,11),QGP(11),PARM(11,11),ID(11),
C     3GSUM(11),G(11),TNAM(11)
      DIMENSION IDGP(50,3),NOGP(50,3),ALGAC(3,50),ACT(3,50)
      DIMENSION NGM(3),IGP(50,3),NGP(50,3),ACLG(3,50),AC(3,50)
      DIMENSION NGMTEM(3),NOGTEM(50,3),IDGTEM(50,3)
      DIMENSION NGMOL(3)
      COMMON /A/MOL(2,3),MW1,MW2,MW3,D1,D2,D3,VM1,VM2,VM3,VSOV,XSOV
C      COMMON /B/NCNLL,FLAG3,V1(50),XLL(50),S2(50),XL(50),J2
      COMMON /C/NCNUNI,FLAG4,FLAG7,LSTA,HF,TM,TEMP
      COMMON /D/NMG,NSG,RR(100),QQ(100),NKTAB(100),GNAM(100)
      COMMON /DD/KODE(50,50),PARAM(50,50)
      COMMON /E/NCNEXF,FLAG5,PC,X0,X100
C      COMMON /F/NCNMSA,FLAG6,TSA,PSA,HSA,DEH,DEP
      COMMON /H/XLOG,SLOG,XFE,SFE,XMSA,SMSA
      COMMON /I/NCNADS,OC,KPW,PKOC,KP,FLAG8
      REAL MW1,MW2,MW3,NOGP,NGP,OC,KPW,KP
      CHARACTER*10 MOL,FLAG1,FLAG2,FLAG3,FLAG4,FLAG5,FLAG6,FLAG8
      CHARACTER*10 LNK,FILOUT,GNAM,TNAM,ANS,FLAG7


      IF(NCNUNI.EQ.1)GO TO 40
      DO 32 I=1,NSPS
      NN=NGMOL(I)
      NGMOL(I)=NGMTEM(I)
      DO 31 J=1,NN
      NOGP(J,I)=NOGTEM(J,I)
      IDGP(J,I)=IDGTEM(J,I)
31    CONTINUE
32    CONTINUE
      GO TO 41
40    DO 51 I=1,NSPS
      NN=NGMOL(I)
      NGMTEM(I)=NGMOL(I)
      DO 52 J=1,NN
      NOGTEM(J,I)=NOGP(J,I)
      IDGTEM(J,I)=IDGP(J,I)
52    CONTINUE
51    CONTINUE

41    NSP=2
      NGM(1)=NGMOL(2)
      NN=NGM(1)
      DO 42 I=1,NN
      NGP(I,1)=NOGP(I,2)
      IGP(I,1)=IDGP(I,2)
42    CONTINUE
      NGM(2)=NGMOL(1)
      NN=NGM(2)
      DO 43 I=1,NN
      NGP(I,2)=XSOV*NOGP(I,1)
      IGP(I,2)=IDGP(I,1)
43    CONTINUE
      NGM(2)=NGM(2)+1
      NH2O=NGM(2)
      IGP(NH2O,2)=IDGP(1,3)
      NGP(NH2O,2)=(1-XSOV)*NOGP(1,3)
      NMIX=1
```

```
      XMOL(1,1)=0.0
      XMOL(2,1)=1.0

      CALL UNIFAC(NSP,NGP,IGP,NGM,NMIX,XMOL,ACLG,AC)

      R=1.987
      IF(LSTA.EQ.2)GO TO 50
      IF(AC(1,1).LT.100.)GO TO 3
      TEM=(HF/(R*TEMP))*((TEMP/TM)-1)-ACLG(1,1)
      XUNI=EXP(TEM)
      GO TO 8
3     IF(AC(1,1).LT.10.)GO TO 10
      TEM=(HF/(R*TEMP))*((TEMP/TM)-1)-ACLG(1,1)
      A=EXP(TEM)
      B=1.
      EPS=0.001
      I=0
      FA=ALOG(A)-(HF/(R*TEMP))*((TEMP/TM)-1)+((1-A)**2)*ACLG(1,1)
      IF(ABS(FA).GT.EPS)GO TO 4
      XUNI=A
      GO TO 8
4     FB=ALOG(B)-(HF/(R*TEMP))*((TEMP/TM)-1)+((1-B)**2)*ACLG(1,1)
      IF(ABS(FB).GT.EPS)GO TO 7
      XUNI=B
      GO TO 8
7     I=I+1
      XUNI=A+(FA*(B-A)/(FA-FB))
C      XUNI=(A+B)/2
      FX=ALOG(XUNI)-(((HF/(R*TEMP))*((TEMP/TM)-1))-ACLG(1,1))
      IF(ABS(FX).LE.EPS)GO TO 8
      IF(FX*FA)5,8,6
5     B=XUNI
      FB=FX
      GO TO 7
6     A=XUNI
      FA=FX
      GO TO 7
10    A=EXP(((HF/(R*TEMP))*((TEMP/TM)-1))-ACLG(1,1))
      B=1.
      EPS=A/100.
      FA=ALOG(A*AC(1,1))-(HF/(R*TEMP))*((TEMP/TM)-1)
      IF(ABS(FA).GT.EPS)GO TO 21
      XUNI=A
      GO TO 8
21    FB=ALOG(B*AC(1,1))-(HF/(R*TEMP))*((TEMP/TM)-1)
      IF(ABS(FB).GT.EPS)GO TO 22
      XUNI=B
      GO TO 8
22    I=I+1
      XUNI=A+(FA*(B-A)/(FA-FB))
C      XUNI=(A+B)/2.
      NMIX=1
      XMOL(1,1)=XUNI
      XMOL(2,1)=1-XUNI
      CALL UNIFAC(NSP,NGP,IGP,NGM,NMIX,XMOL,ACLG,AC)
      TEM=XUNI*AC(1,1)
      FX=ALOG(TEM)-((HF/(R*TEMP))*((TEMP/TM)-1))
      IF(ABS(FX).LE.EPS)GO TO 8
      TEM=XUNI-1.
      IF(ABS(TEM).GE.EPS)GO TO 23
      TEM=A-B
      IF(ABS(TEM).GT.EPS)GO TO 23
      XUNI=A+(FA*(B-A)/(FA-FB))
C      XUNI=(A+B)/2
      GO TO 8
23    IF(FX*FA)15,8,16
15    B=XUNI
      FB=FX
      GO TO 22
16    A=XUNI
      FA=FX
      GO TO 22

8     IF(XUNI.GT.1.0)XUNI=1.0
      TE6=(XSOV*(1-XUNI)*VM1)+(XUNI*VM2)+((1-XSOV)*(1-XUNI)*VM3)
      SUNI=(XUNI*MW2*1000000.0)/TE6
      RETURN

50    IF(AC(1,1).LT.1000)GO TO 55
C50    IF(AC(1,1).LT.50.)GO TO 55
      XUNI=1/AC(1,1)
      GO TO 130
55    IF(AC(1,1).LE.7.4)GO TO 61
      IF(AC(1,1).LT.50)GO TO 62

      I=0
      A=0.001
      B=0.999
      EPS=0.001
      FA=(1-4*A+3*(A**2))*ACLG(1,1)+(2*A-3*(A**2))*ACLG(2,1)
     1+ALOG(A)-ALOG(1-A)
      IF(ABS(FA).GT.EPS)GO TO 14
      XUNI=A
      GO TO 130
14    FB=(1-4*B+3*(B**2))*ACLG(1,1)+(2*B-3*(B**2))*ACLG(2,1)
     1+ALOG(B)-ALOG(1-B)
      IF(ABS(FB).GT.EPS)GO TO 17
      XUNI=B
      GO TO 130
17    I=I+1
      XUNI=A+(FA*(B-A)/(FA-FB))
      FX=(1-4*XUNI+3*(XUNI**2))*ACLG(1,1)+(2*XUNI-3*(XUNI**2))
     1*ACLG(2,1)+ALOG(XUNI)-ALOG(1-XUNI)
C       WRITE(*,1001)A,B,XUNI,FA,FB,FX
C1001  FORMAT(1X,3(F6.4,2X),3(E10.3,1X))
      IF(ABS(FX).LE.EPS)GO TO 130
      IF(FX*FA)25,130,26
25    B=XUNI
      FB=FX
      GO TO 17
26    A=XUNI
      FA=FX
      GO TO 17
62    XUNI=0.01
      EPS=0.001
C63    WRITE(*,1002)XUNI,SUNI
63    FX=(1-4*XUNI+3*(XUNI**2))*ACLG(1,1)+(2*XUNI-3*(XUNI**2))
     1*ACLG(2,1)+ALOG(XUNI)-ALOG(1-XUNI)
C1002  FORMAT(1X,F6.4,2X,E10.3)
      IF(ABS(FX).LE.EPS)GO TO 130
      XUNI=XUNI-((1-4*XUNI+3*(XUNI**2))*ACLG(1,1)+(2*XUNI-3*(XUNI**2))
     1*ACLG(2,1)+ALOG(XUNI)-ALOG(1-XUNI))/((-4+6*XUNI)*ACLG(1,1)
     2+(2-6*XUNI)*ACLG(2,1)+(1/XUNI)+(XUNI/(1-XUNI)))
      GO TO 63
61    XUNI=1
130   IF(XUNI.GT.1.0)XUNI=1.0
      TE6=(XSOV*(1-XUNI)*VM1)+(XUNI*VM2)+((1-XSOV)*(1-XUNI)*VM3)
      SUNI=(XUNI*MW2*1000000.0)/TE6
```

```
      RETURN
      END



      SUBROUTINE UNIFAC(NSPS,NOGP,IDGP,NGMOL,NMIX,XMOL,ALGAC,ACTCF)
C*****************************************************************C
C                                                                 C
C        CALCULATE ESTIMATES OF ACTIVITY COEFFICIENTS IN NON-     C
C        ELECTROLYTE LIQUID MIXTURES.                             C
C                                                                 C
C     METHOD                                                      C
C           COMBINES THE SOLUTION-OF-FUNCTIONAL-GROUPS CONCEPT WITH C
C           A MODEL FOR ACTIVITY COEFFICIENTS BASED ON AN EXTENTION C
C           OF THE QUASI-CHEMICAL THEORY OF LIQUID MIXTURES (UNIQUAC). C
C                                                                 C
C     NOTE                                                        C
C         ADAPTED FROM ANDERSON (1983), AND FREDENSLUND ET AL. (1975) C
C*****************************************************************C

      DIMENSION XMOL(3,50),COMLN(11),RESLN(11),
     1AL(11),RI(11),QI(11),QGPI(6,11),QGP(11),PARM(11,11),ID(11),
     3GSUM(11),G(11),TNAM(11)
      DIMENSION IDGP(50,3),NOGP(50,3),ALGAC(3,50),ACTCF(3,50)
C      DIMENSION NGM(3),IGP(50,3),NGP(50,3),ACLG(3,50),AC(3,50)
      DIMENSION NGMOL(3)
      COMMON /A/MOL(2,3),MW1,MW2,MW3,D1,D2,D3,VM1,VM2,VM3,VSOV,XSOV
C      COMMON /B/NCNLL,FLAG3,V1(50),XLL(50),S2(50),XL(50),J2
      COMMON /C/NCNUNI,FLAG4,FLAG7,LSTA,HF,TM,TEMP
      COMMON /D/NMG,NSG,RR(100),QQ(100),NKTAB(100),GNAM(100)
      COMMON /DD/KODE(50,50),PARAM(50,50)
      COMMON /E/NCNEXF,FLAG5,PC,X0,X100
C      COMMON /F/NCNMSA,FLAG6,TSA,PSA,HSA,DEH,DEP
      REAL MW1,MW2,MW3,NOGP,NUM
      CHARACTER*10 MOL,FLAG1,FLAG2,FLAG3,FLAG4,FLAG5,FLAG6,FLAG7
      CHARACTER*10 LNK,FILOUT,GNAM,TNAM,ANS

C*****************************************************************C
C     READ COMPONENTS AND NUMBER OF GROUPS PER COMPONENT.         C
C*****************************************************************C
C      WRITE(*,'(A)')'  PLEASE GIVE OUTPUT FILE NAME FOR UNIFAC ='
C      READ(*,1100)FILOUT
C1100  FORMAT(A)
C      OPEN(26,FILE=FILOUT,ACCESS='SEQUENTIAL',STATUS='NEW')

      NGPT = 0
      DO 60 J=1,NSPS
      NN = NGMOL(J)
      DO 55 I=1,NN
      IF (NGPT.EQ.0) GO TO 50
      DO 45 K=1,NGPT
      IF (IDGP(I,J).NE.ID(K)) GO TO 45
      IDGP(I,J) = K
      GO TO 55
   45 CONTINUE
   50 NGPT = NGPT + 1
      ID(NGPT) = IDGP(I,J)
      IDGP(I,J) = NGPT
   55 CONTINUE
   60 CONTINUE
C*****************************************************************C
C     PRINT SUMMARY OF INPUT DATA.                                C
C*****************************************************************C
C      WRITE (26,2001)
      DO 62 J=1,NSPS
C      WRITE (26,2002) MOL(1,J),MOL(2,J)
      K = NGMOL(J)
      DO 61 I=1,K
      L = IDGP(I,J)
      L = ID(L)
      M = NKTAB(L)
      NUM = NOGP(I,J)
C   61 WRITE (26,2003)GNAM(L),M,L,NUM,QQ(L),RR(L)
61    CONTINUE
   62 CONTINUE
C*****************************************************************C
C     READ TEMPERATURE AND NUMBER OF MIXTURES TO BE CALCULATED.   C
C*****************************************************************C
C      WRITE(*,1007)
C1007  FORMAT(' INPUT TEMPERATURE OF MIXTURE IN Kelvin, AND # OF
C     1DIFFERENT COMPOSITIONS'/'  FOR WHICH ACTIVITY COEFFICIENTS ARE TO
C     2BE CALAULATED')
C   63 READ (*,*) TEMP,NMIX
      IF (NMIX.EQ.0) NMIX = 1
C      IF (TEMP.LT.200.) GO TO 25
C      WRITE(26,2012)
      DO 65 I=1,NGPT
      K = ID(I)
      K = NKTAB(K)
      DO 65 J=1,NGPT
      L = ID(J)
      L = NKTAB(L)
      IF (L.EQ.K) GO TO 65
C*****************************************************************C
C     KODE IS A POINTER FOR PARAMETER BACKGROUND NOTES.           C
C     IF  KODE = 0   PARAMETERS ARE FROM REFERENCE 5              C
C     (    (   = 1       (       (   (       (     4              C
C     (    (   = 2       (       (   (       (     6              C
C     (    (   = 3       (       (   (       (     7              C
C     NO NOTES ARE PRINTED IF PARAMETERS ARE FROM REFERENCE 5     C
C*****************************************************************C
      IF(KODE(K,L).EQ.0) GO TO 64
      IF((L.GT.K).OR.(KODE(K,L).EQ.0)) GO TO 64
      GO TO (66,67,68) KODE(K,L)
   66 CONTINUE
C      WRITE(26,2009) K,L,ID(I),ID(J)
      WRITE(31,2009) K,L,ID(I),ID(J)
      GO TO 64
   67 CONTINUE
C      WRITE(26,2010) K,L,ID(I),ID(J)
      WRITE(31,2010) K,L,ID(I),ID(J)
      GO TO 64
   68 CONTINUE
C      WRITE(26,2011) K,L,ID(I),ID(JH)
      WRITE(31,2011) K,L,ID(I),ID(JH)
   64 CONTINUE
      IF (ABS(PARAM(K,L)).GT.1.E-3) GO TO 65
C      WRITE (26,3001) K,L
      WRITE (31,3001) K,L
 3001 FORMAT (1X,'INTERACTION PARAMETER FOR THE ',I2,'-',I2,
     1       ' NOT AVAILABLE')
      STOP
   65 PARM(I,J) = EXP(-PARAM(K,L)/TEMP)
C*****************************************************************C
C     PRINT GROUP INTERACTION PARAMETER DATA                      C
```

```
C*****************************************************************C
      DO 70 I=1,NGPT
      K = ID(I)
   70 TNAM(I)=GNAM(K)
C      WRITE (26,2004) NMIX,TEMP
C      WRITE (26,2005) (TNAM(I),I=1,NGPT)
C      DO 75 I=1,NGPT
C   75 WRITE (26,2006)TNAM(I), (PARM(I,J),J=1,NGPT)
C*****************************************************************C
C     READ SET OF COMPOSITIONS.                                     C
C*****************************************************************C
C      WRITE(*,1008)
C1008  FORMAT(' INPUT MOLE FRACTION FOR ALL COMPONENTS IN THE MIXTURE,
C     1'/'  FOR AS MANY TIMES AS SPECIFIED ABOVE')

C      DO 170 II=1,NMIX
C      READ (*,*) (XMOL(J,II),J=1,NSPS)
C*****************************************************************C
C     COMPUTE COMBINATORIAL PORTION AND RELATED SUMS.               C
C*****************************************************************C
      DO 170 II=1,NMIX
      RSUM = 0.
      QSUM = 0.
      DO 80 I=1,NGPT
   80 QGP(I) = 0.
      DO 95 J=1,NSPS
      RI(J) = 0.
      QI(J) = 0.
      K = NGMOL(J)
      DO 85 I=1,K
      A = NOGP(I,J)
      L = IDGP(I,J)
      M = ID(L)
      RI(J) = RI(J) + A*RR(M)
      QI(J) = QI(J) + A*QQ(M)
      QGPI(I,J) = A*QQ(M)
   85 QGP(L) = QGP(L) + QGPI(I,J)*XMOL(J,II)
      RSUM = RSUM + RI(J)*XMOL(J,II)
      QSUM = QSUM + QI(J)*XMOL(J,II)
      DO 90 I=1,K
   90 QGPI(I,J) = QGPI(I,J)/QI(J)
   95 CONTINUE
      DO 100 I=1,NGPT
  100 QGP(I) = QGP(I)/QSUM
      SUML = 0.
      DO 105 J=1,NSPS
      AL(J) = 5.*(RI(J) - QI(J)) - RI(J) + 1.
  105 SUML = SUML + AL(J)*XMOL(J,II)
      DO 120 J=1,NSPS
      CON1 = RI(J)/RSUM
      CON2 = QI(J)/QSUM
  120 COMLN(J) = ALOG(CON1) + 5.*QI(J)*ALOG(CON2/CON1)
     1           + AL(J) - CON1*SUML
C*****************************************************************C
C     COMPUTE RESIDUAL PORTION.                                     C
C*****************************************************************C
      DO 125 I=1,NGPT
      GSUM(I) = 0.
      DO 125 K=1,NGPT
  125 GSUM(I) = GSUM(I) + PARM(K,I)*QGP(K)
      DO 135 I=1,NGPT
      SUM = 0.
      DO 130 K=1,NGPT
  130 SUM = SUM + QGP(K)*PARM(I,K)/GSUM(K)
  135 G(I) = 1. - ALOG(GSUM(I)) - SUM
      DO 155 J=1,NSPS
      RESLN(J) = 0.
      K = NGMOL(J)
      DO 140 I=1,K
      M = IDGP(I,J)
      GSUM(I) = 0.
      M = IDGP(I,J)
      GSUM(I) = 0.
      DO 140 N=1,K
      L = IDGP(N,J)
  140 GSUM(I) = GSUM(I) + PARM(L,M)*QGPI(N,J)
      DO 150 I=1,K
      SUM = 0.
      M = IDGP(I,J)
      DO 145 N=1,K
      L = IDGP(N,J)
  145 SUM = SUM + QGPI(N,J)*PARM(M,L)/GSUM(N)
      JJ = ID(M)
      CONST =NOGP(I,J)*QQ(JJ)*(G(M) - (1.-ALOG(GSUM(I))-SUM))
  150 RESLN(J) = RESLN(J) + CONST
      ALGAC(J,II) = COMLN(J) + RESLN(J)
  155 ACTCF(J,II) = EXP(ALGAC(J,II))
C*****************************************************************C
C     PRINT RESULTS OF MIXTURE CALCULATIONS.                        C
C*****************************************************************C
C      WRITE (26,2007)
C      WRITE (*,2007)
C      WRITE (31,2007)
C      DO 160 J=1,NSPS
C  160 WRITE(26,2008)MOL(1,J),MOL(2,J),XMOL(J,II),ALGAC(J,II),ACTCF(J,II)
C      WRITE(*,2008)MOL(1,J),MOL(2,J),XMOL(J,II),ALGAC(J,II),ACTCF(J,II)
C      WRITE(31,2008)MOL(1,J),MOL(2,J),XMOL(J,II),ALGAC(J,II),ACTCF(J,II)
C160   CONTINUE
  170 CONTINUE
C      GO TO 63
C      CLOSE(26,STATUS='KEEP')
C 1001 FORMAT (I2,2A10)
C 1002 FORMAT (16I5)
C 1003 FORMAT (F10.0,I5)
C 1004 FORMAT (8F10.0)
C 2001 FORMAT (1H1,'UNIFAC CALCULATION OF ACTIVITY COEFFICIENTS')
C 2002 FORMAT (1H0,2A10//8X,'GROUP',4X,'PRI',2X,'SEC',3X,'NUM',7X,
C     1       'Q',9X,'R'/)
C 2003 FORMAT (6X,A10,' (',I2,')',1X,' (',I2,')',1X,F5.3,2X,F6.4,4X,F6.4)
C 2004 FORMAT (1H1,'ACTIVITY COEFFICIENTS FOR ',I2,' COMPOSITIONS AT ',
C     1       F6.2,' DEGREES KELVIN'//1X,'PARAMETERS'/)
C 2005 FORMAT (14X,11A10)
C 2006 FORMAT (/1X,A10,11F10.5)
C 2007 FORMAT (1H1,5X,'COMPONENT',9X,'MOLE FRAC',2X,'LOG ACTCF',7X,
C     1       'ACTCF'/)
C 2008 FORMAT(1X,2A10,1X,F10.4,1X,F10.5,1X,F13.4)
 2009 FORMAT(1X,'PARAMETERS FOR THE ',I2,'-',I2,' GROUP BINARY ARE FROM
     1REFERENCE 4',/,1X,'THESE PARAMETERS ARE USED FOR THE ',I2,'-',I2,
     2' SECONDARY GROUP INTERACTION',/)
 2010 FORMAT(1X,'PARAMETERS FOR THE ',I2,'-',I2,' GROUP BINARY ARE FROM
     1REFERENCE 6',/,1X,'THESE PARAMETERS ARE USED FOR THE ',I2,'-',I2,
     2' SECONDARY GROUP INTERACTION',/)
 2011 FORMAT(1X,'PARAMETERS FOR THE ',I2,'-',I2,' GROUP BINARY ARE FROM
     1REFERENCE 7',/,1X,'THESE PARAMETERS ARE USED FOR THE ',I2,'-',I2,
     2' SECONDARY GROUP INTERACTION',/)
C 2012 FORMAT(1X,/)
      RETURN
      END
```

```
      SUBROUTINE REG1(X,Y,N,B,SLP)
      DIMENSION  X(100),Y(100),XI(100)
      CHARACTER*10 ANS
      SUMX=X(1)
      SUMY=Y(1)
      DO 1 I=2,N
      SUMX=SUMX+X(I)
      SUMY=SUMY+Y(I)
1     CONTINUE
      XM=SUMX/N
      YM=SUMY/N
      DO 11 I=1,N
      XI(I)=X(I)-XM
11    CONTINUE
      SXX=XI(1)**2
      SYY=Y(1)**2
      SXY=XI(1)*Y(1)
      DO 2 I=2,N
      SXX=SXX+(XI(I)**2)
      SYY=SYY+(Y(I)**2)
      SXY=SXY+(XI(I)*Y(I))
2     CONTINUE
C     SLP: SLOPE
4      SLP=SXY/SXX
C     B:INTERCEPT
       B=YM-SLP*XM
C     VAR: VARIANCE
C       VAR=((SXX*SYY)-(SXY**2))/(N*(N-2)*SXX)
C     STDEV: STANDARD DEVIATION
C       STDEV=SQRT(ABS(VAR))
C     CORCOE:CORRELATION OF COEFFICIENT
C       CORCOE=SXY/(SQRT(ABS(SXX*SXY)))
      RETURN
      END
```

```
      SUBROUTINE REG2(X,Z,Y,C1,C2,N)
      DIMENSION X(100),Y(100),Z(100),XI(100),YI(100),ZI(100)

      SUMX=X(1)
      SUMY=Y(1)
      SUMZ=Z(1)
      DO 1 I=2,N
      SUMX=SUMX+X(I)
      SUMY=SUMY+Y(I)
      SUMZ=SUMZ+Z(I)
1     CONTINUE
      XM=SUMX/N
      YM=SUMY/N
      ZM=SUMZ/N
      DO 2 I=1,N
      XI(I)=X(I)-XM
      YI(I)=Y(I)-YM
      ZI(I)=Z(I)-ZM
2     CONTINUE
      SUMXX=XI(1)**2
      SUMZZ=ZI(1)**2
      SUMXY=XI(1)*YI(1)
      SUMXZ=XI(1)*ZI(1)
      SUMYZ=YI(1)*ZI(1)
      DO 3 I=2,N
      SUMXX=SUMXX+(XI(I)**2)
      SUMZZ=SUMZZ+(ZI(I)**2)
      SUMXY=SUMXY+(XI(I)*YI(I))
      SUMXZ=SUMXZ+(XI(I)*ZI(I))
      SUMYZ=SUMYZ+(YI(I)*ZI(I))
3     CONTINUE
C      WRITE(*,*)SUMXY,SUMXX,SUMXZ
C      WRITE(*,*)SUMYZ,SUMXZ,SUMZZ
      TEM=SUMXZ*SUMZZ
      IF(TEM.GE.0)GO TO 11
      C1=(SUMXY*SUMZZ+SUMYZ*SUMXZ)/(SUMXX*SUMZZ+SUMXZ*SUMXZ)
      GO TO 12
11    C1=(SUMXY*SUMZZ-SUMYZ*SUMXZ)/(SUMXX*SUMZZ-SUMXZ*SUMXZ)
12    TEM=SUMXX*SUMXZ
      IF(TEM.GE.0)GO TO 13
      C2=(SUMXY*SUMXZ+SUMYZ*SUMXX)/(SUMXZ*SUMXZ-SUMZZ*SUMXX)
      GO TO 14
13    C2=(SUMXY*SUMXZ-SUMYZ*SUMXX)/(SUMXZ*SUMXZ-SUMZZ*SUMXX)
14    WRITE(*,1001)C1,C2
1001  FORMAT(1X,' THE SOLVENT-SOLVENT INTERACTION PARAMETERS ARE '
     *,F6.4,2X,'AND',F6.4)
      RETURN
      END
```

```
      SUBROUTINE ADS(NSPS,NGMOL,NOGP,IDGP,SLP1)
      DIMENSION X(100),V(10),XU(10),SU(10)
      DIMENSION IDGP(50,3),NOGP(50,3),ALGAC(3,50),ACT(3,50)
      DIMENSION NGM(3),IGP(50,3),NGP(50,3),ACLG(3,50),AC(3,50)
      DIMENSION NGMOL(3),NGMTEM(3),NOGTEM(50,3),IDGTEM(50,3)
      COMMON /A/MOL(2,3),MW1,MW2,MW3,D1,D2,D3,VM1,VM2,VM3,VSOV,XSOV
      COMMON /C/NCNUNI,FLAG4,FLAG7,LSTA,HF,TM,TEMP
C      COMMON /D/NMG,NSG,RR(100),QQ(100),NKTAB(100),GNAM(100)
C      COMMON /DD/KODE(50,50),PARAM(50,50)
      COMMON /E/NCNEXF,FLAG5,PC,X0,X100
      COMMON /I/NCNADS,OC,KPW,PKOC,KP,FLAG8
C      COMMON /J/X10,X20,X30
      REAL MW1,MW2,MW3,NOGP,NGP,KPW,KP,OC,INT
      CHARACTER*10 MOL,FLAG4,FLAG5,FLAG6,FLAG7,FLAG8,GNAM
C     The partition KOC is calculated based on Karickhoff(1981)
      VTEM=VSOV
      IF(NCNADS.GT.1)GO TO 50
      IF(X0.EQ.0)THEN
        VSOV=0.0
        XSOV=0.
        CALL UNIEST(ACLG,AC,XUNI,SUNI,NSPS,NGMOL,NOGP,IDGP)
        X0=XUNI
      END IF

      X(1)=0.
      V(1)=0.
      XU(1)=X0
      PKOCLG=-0.83*LOG10(X0)-0.01*(TM-298)-0.93
      PKOC=10**PKOCLG
      KPW=PKOC*OC/100

      IF(SLP1.EQ.0.0)THEN
      VSOV=10.
      XSOV=(D1*VSOV/MW1)/((D1*VSOV/MW1)+(D3*(100-VSOV)/MW3))
```

```
      CALL UNIEST(ACLG,AC,XUNI,SUNI,NSPS,NGMOL,NOGP,IDGP)
      X(2)=XSOV
      V(2)=VSOV
      XU(2)=XUNI
      SU(2)=SUNI
      VSOV=20.
      XSOV=(D1*VSOV/MW1)/((D1*VSOV/MW1)+(D3*(100-VSOV)/MW3))
      CALL UNIEST(ACLG,AC,XUNI,SUNI,NSPS,NGMOL,NOGP,IDGP)
      X(3)=XSOV
      V(3)=VSOV
      XU(3)=XUNI
      SU(3)=SUNI
      VSOV=30.
      XSOV=(D1*VSOV/MW1)/((D1*VSOV/MW1)+(D3*(100-VSOV)/MW3))
      CALL UNIEST(ACLG,AC,XUNI,SUNI,NSPS,NGMOL,NOGP,IDGP)
      X(4)=XSOV
      V(4)=VSOV
      XU(4)=XUNI
      SU(4)=SUNI
         CALL REG1(V,XU,4,INT,SIGMA)
      SLP1=SIGMA
      END IF
50    VSOV=VTEM
      ADSTEM=SLP1*.51*VSOV
      TEM=LOG10(55.3*KPW)-ADSTEM
      KP=10**TEM

      WRITE(*,5014)MOL(1,2),MOL(2,2),MOL(1,1),MOL(2,1),KP
5014  FORMAT(//,1X,' ADSORPTION COEFFICIENT OF ',A10,A10,' IN WATER/'
     *,2A10,/,' MIXTURES IS ',E10.3)

      RETURN
      END
```

```
4689
0.9011 0.848  1CH3
0.6744  .540  1CH2
0.4469  .228  1CH
0.2195 0.000  1C
1.3454 1.176  2CH2=CH
1.1167  .867  2CH=CH
1.1173 0.988  2CH2=C
0.8886  .676  2CH=C
0.6605  .485  2C=C
0.5313 0.400  3ACH
0.3652  .120  3AC
1.2663  .968  4ACCH3
1.0396 0.660  4ACCH2
0.8121  .348  4ACCH
1.0000 1.200  5OH
1.4311 1.432  6CH3OH
0.9200 1.400  7H2O
0.8952  .680  8ACOH
1.6724 1.488  9CH3CO
1.4457 1.180  9CH2CO
0.9980  .948 10CHO
1.9031 1.728 11CH3COO
1.6764 1.420 11CH2COO
1.2420 1.188 12HCOO
1.1450 1.088 13CH3O
0.9183  .780 13CH2O
0.6908  .468 13CH-O
0.9183 1.100 13FCH2O
1.5959 1.544 14CH3NH2
1.3692 1.236 14CH2NH2
1.1417 0.924 14CHNH2
1.4337 1.244 15CH3NH
1.2070  .936 15CH2NH
0.9795 0.624 15CHNH
1.1865  .940 16CH3N
0.9597  .632 16CH2N
1.0600 0.816 17ACNH2
2.9993 2.113 18C5H5N
2.8332 1.833 18C5H4N
2.6670 1.553 18C5H3N
1.8701 1.724 19CH3CN
1.6434 1.416 19CH2CN
1.3013 1.224 20COOH
1.5280 1.532 20HCOOH
1.4654 1.264 21CH2CL
1.2380 0.952 21CHCL
1.0060  .724 21CCL
2.2564 1.988 22CH2CL2
2.0606 1.684 22CHCL2
1.8016 1.448 22CCL2
2.8700 2.410 23CHCL3
2.6401 2.184 23CCL3
3.3900 2.910 24CCL4
1.1562  .844 25ACCL
2.0086 1.868 26CH3NO2
1.7818 1.560 26CH2NO2
1.5544 1.248 26CHNO2
1.4199 1.104 27ACNO2
2.0570 1.650 28CS2
1.8770 1.676 29CH3SH
1.6510 1.368 29CH2SH
3.1680 2.481 30FURFURAL
2.4088 2.248 31(CH2OH)2
```

```
1.2640 0.992 32I
0.9492  .832 33BR
1.2920 1.088 34CH-TRIP-C
1.0613 0.784 34C-TRIP-C
2.8266 2.472 35ME2SO
2.3144 2.052 36ACRY
0.7910 0.724 37CL(C=C)
0.6948  .524 38ACF
3.0856 2.736 39DMF-1
2.6322 2.120 39DMF-2
1.4060 1.380 40CF3
1.0105  .920 40CF2
0.6150 0.460 40CF
1.3800 1.200 41COO
1.6035 1.263 42SIH3
1.4443 1.006 42SIH2
1.2851  .749 42SIH
1.0470  .410 42SI
1.4338 1.062 43SIH2O
1.3030  .764 43SIHO
1.1044  .466 43SIO
0.2854 0.092 44TERT-N
1.466  1.336 45AMIDE
2.859  2.428 46CON(ME)2
2.632  2.120 46CONMECH2
2.405  1.812 46CON(CH2)2
0    0.   2   86.0200   61.1300   76.5000  986.5000  697.2000 1318.000
0 1333.0000  476.4000  677.0000  232.1000  741.4000  251.5000  391.500
0  255.7000  206.6000 1245.0000  287.7000  597.0000  663.5000   35.930
0   53.7600   24.9000  104.3000  321.5000  661.5000  543.0000  153.600
0  184.4000  354.5000 3025.0000  335.8000  479.5000  298.9000  526.500
0  689.0002   -4.1890  125.8000  485.3000   -2.8592  387.1003  407.200
3  327.0001  383.0001-1380.0001  729.0000    0.   0    0.   0    0.
2  -35.3600    0.   2   38.8102   74.1502  524.1002  787.6002  270.600
2  526.1002  182.6001  -35.1002   37.8502  449.1002  214.5002  240.900
2  163.9002   61.1102    0.   2    0.   2  336.9002  318.9002  204.600
2    5.8922  -13.9902 -109.7002  393.1002  357.5002    0.   2   76.300
2    0.   2    0.   2    0.   2    0.   2    0.   2   31.1402 -137.400
2    0.   2  -66.4602    0.   2  -70.4502    0.   2   48.3303    0.
3    0.   0    0.   1-2340.0000    0.   0    0.   0    0.   0    0.
0  -11.1202    3.4460    0.   0  167.0000  636.1000  637.3000  903.800
0 1329.0000   25.7700    0.   0    5.9940    0.   0   32.1400 -161.700
0  122.8000   90.4900  668.2000   -4.4490  212.5000  537.4000  -18.810
0 -144.4000 -231.9000    3.0000  538.2000  168.0000  194.9000   52.070
0  -10.4300  -64.6900  210.4000  113.3000  -13.5900    0.   0  169.900
0    0.   2 -259.1000  389.3000  245.6000    0.   2  103.5003  551.900
3  254.3001  109.0001   75.9001  784.0000    0.   0    0.   0    0.
0  -69.7002 -113.6000 -146.8000    0.   0  803.2000  603.2000 5695.000
0  884.9000  -52.1000    0.   0 5688.0000    0.   0  213.1000    0.
0  -49.2900   23.5000  764.7000   52.8000 6096.0000  603.8000 -114.100
0    0.   0  -12.1400 -141.3000 -126.9000 3629.0000 4448.0000   -9.451
0    0.   0  -20.3600 4975.0000    0.   0 -171.3000    0.   0 4284.000
0    0.   2    0.   0  101.4000 5629.0000    0.   2   69.2603  683.300
3  355.5001 1320.0001  482.0001  386.0000    0.   0    0.   0    0.
0  156.4002  457.0000   89.6000   25.8200    0.   0 -137.1000  353.500
0 -259.7000   84.0000  441.8000  101.1000  193.1000   28.0600   83.020
0   42.7000 -323.0000 -348.2000  170.0000    6.7120  199.0000   75.620
0 -112.1000  -98.1200  143.1000  287.8000   61.1100  157.1000  477.000
0  147.5000 -120.5000 -318.9000  313.5000  133.4000    0.   0 -202.100
0    0.   2  225.8000   44.7800 -143.9000    0.   2  190.3003  269.100
3  202.7000    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0   16.5102  -12.5200  -50.0000  -44.5000  249.1000    0.   0 -181.000
0 -101.7000   23.3900  306.4000  -10.7200  193.4000 -180.6000  359.300
0  266.0000   53.9000  335.5000  580.5000   36.2300 -289.5000  -38.320
0 -102.5000 -139.4000  -67.8000   17.1200   75.1400    0.   0  -31.090
0   37.8400    0.   0    0.   0    0.   0    0.   0    0.   0 -399.300
```

```
0    0.   2   33.4700  -48.2500 -172.4000    0.   2  165.7003    0.
3    0.   1  214.0000    0.   0    0.   0    0.   0    0.   0    0.
0  300.0002  496.1000  362.3000  377.6000 -229.1000  289.6000    0.
0  324.5000 -195.4000 -257.3000   14.4200    0.   0  540.5000   48.890
0  168.0000  304.0000  213.0000  459.0000  112.6000  -14.0900  325.400
0  370.4000  353.7000  497.5000  678.2000  220.6000  399.5000  887.100
0    0.   0  188.0000    0.   0    0.   0    0.   0    0.   0 -139.000
0  160.8002    0.   0    0.   0  319.0000    0.   2 -197.5003    0.
3    0.   1  365.0000    0.   0    0.   0    0.   0    0.   0    0.
0  275.8002  217.5000   25.3400  244.2000 -451.6000 -265.2000 -601.800
0    0.   0 -356.1000    0.   0 -449.4000    0.   0    0.   0    0.
0    0.   0    0.   0    0.   0 -305.5000    0.   0    0.   0    0.
0    0.   0    0.   0 1827.0000    0.   0    0.   0    0.   0    0.
0    0.   0    0.   0 -687.1000    0.   0    0.   0    0.   0    0.
0    0.   2    0.   0    0.   0    0.   0    0.   2 -494.2003    0.
3    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0   26.7602   42.9200  140.1000  365.8000  164.5000  108.7000  472.500
0 -133.1000    0.   0  -37.3600 -213.7000    0.   0    5.2020    0.
0    0.   0    0.   0  937.9000  165.1000  481.7000  669.4000 -191.700
0 -284.0000 -354.6000  -39.2000  174.5000  137.5000    0.   0  216.100
0  -46.2800 -163.7000    0.   0   53.5900  245.2000 -246.6000  -44.580
0    0.   2  -34.5700    0.   0  -61.7000    0.   2  -18.8003    0.
3    0.   1  135.0001-1680.0001  -58.0000    0.   0    0.   0    0.
0  505.7001  133.0000    0.   0    0.   0 -404.8000 -340.2000  232.700
0    0.   0  128.0000    0.   1 -448.0000    0.   0  304.1000    0.
0    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0  751.900
0    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0    0.   2    0.   0    0.   0    0.   0    0.   2    0.   3    0.
3    0.   1   -7.1801  333.0001 6810.0000    0.   0    0.   0    0.
0  114.8002  132.1000   85.8400 -170.0000  245.4000  249.600010000.000
0  -36.7200  372.2001 2390.0000    0.   0  372.9000 -235.7000    0.
0  -73.5000    0.   0    0.   0    0.   0  494.6000  660.2000    0.
0  108.9000 -209.7000   54.4700  629.0000    0.   0    0.   0  183.000
0    0.   0  202.3000 -101.7000  148.3000    0.   0    0.   0   52.080
0    0.   2  -83.3000    0.   0    0.   0    0.   2  560.2003    0.
3    0.   1  -54.6000    0.   1 6960.0000    0.   0    0.   0    0.
0   90.4902  -62.5500    0.   0    0.   0  191.2000  155.7000    0.
0    0.   0    0.   0    0.   0 -261.1000    0.   0    0.   0    0.
0    0.   0    0.   0    0.   0    0.   0    0.   0 -356.3000    0.
0    0.   0 -287.2000    0.   0    0.   0    0.   0    0.   0    0.
0    4.3390    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0    0.   2    0.   0    0.   0    0.   0    0.   2  -70.2403    0.
3    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0   83.3602   26.5100   52.1300   65.6900  237.7000  339.7000 -314.700
0    0.   0   52.3800   -7.8380  461.3000    0.   0    0.   0    0.
0  141.7000    0.   0    0.   0    0.   0    0.   0  664.6000  140.900
0  137.8000 -154.3000   47.6700    0.   0   95.1800    0.   0  140.900
0   -8.5380    0.   0  -20.1100 -149.5000 -202.3000    0.   0  172.100
0    0.   2  240.2000    0.   0  254.8000    0.   2  417.0003    0.
3    0.   1 5780.0001  131.0000    0.   0    0.   0    0.   0    0.
0  -30.4802    1.1630  -44.8500    0.   0 -164.0000 -481.7000 -330.400
0    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0   63.7200  -41.1100    0.   0    0.   0    0.   0    0.   0    0.
0    0.   0    0.   0  -99.8100   68.8100    0.   0    0.   0    0.
0  -70.1400    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0    0.   2    0.   0    0.   0    0.   0    0.   2    0.   3    0.
3    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0   65.3302  -28.7000  -22.3100  223.0000 -150.0000 -500.4000 -448.200
0    0.   0    0.   0    0.   0  136.0000    0.   0  -49.3000  108.800
0    0.   0 -189.2000    0.   0    0.   0    0.   0    0.   0    0.
0    0.   0    0.   0   71.2300 4350.0000    0.   0    0.   0    0.
0    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0    0.   2    0.   0    0.   0    0.   0    0.   2  -38.7703    0.
3    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0  -83.9802  -25.3800 -223.9000  109.9000   28.6000 -406.8000 -598.800
```

```
0    0.    0    0.    0    0.    0    0.    0    0.    0    0.    0   38.890
0  865.9000    0.    0    0.    0    0.    0    0.    0    0.    0    0.
0  -73.8500 -352.9000   -8.2830  -86.3600    0.    0    0.    0    0.
0    0.    0    0.    0    0.    0    0.    0    0.    0    0.    0    0.
0    0.    2    0.    0    0.    0    0.    0    0.    2    0.    3    0.
3    0.    0    0.    0    0.    0    0.    0    0.    0    0.    0    0.
0 5339.0002    0.    0  650.4000  979.8000  529.0000    5.1820 -339.500
0    0.    0 -399.1000    0.    0    0.    0    0.    0    0.    0    0.
0    0.    0    0.    0    0.    0    0.    0 -216.8000    0.    0    0.
0    0.    0    0.    0 8455.0000  699.1000    0.    0  -62.7300    0.
0    0.    0    0.    0  125.3000    0.    0    0.    0    0.    0    0.
0    0.    2    0.    0    0.    0 -293.1000    0.    2    0.    3    0.
3    0.    0    0.    0    0.    0    0.    0    0.    0    0.    0    0.
0 -101.6002    0.    0   31.8700   49.8000 -132.3000 -378.2000 -332.900
0 -341.6000  -51.5400    0.    0    0.    0    0.    0    0.    0    0.
0    0.    0    0.    0    0.    0    0.    0 -169.7000 -153.7000    0.
0 -351.6000 -114.7000 -165.1000    0.    0    0.    0    0.    0    0.
0    0.    0    0.    0    0.    0    0.    0    0.    0    0.    0    0.
0    0.    2    0.    0    0.    0    0.    0    0.    2    0.    3    0.
3    0.    0    0.    0    0.    0    0.    0    0.    0    0.    0    0.
0   24.8202  -40.6200  -22.9700 -138.4000 -185.4000  157.8000  242.800
0    0.    0 -287.5000    0.    0 -266.6000    0.    0    0.    0    0.
0    0.    0    0.    0  617.1000  134.3000    0.    0    0.    0    0.
0    0.    0  -15.6200  -54.8600   52.3100    0.    0    0.    0  230.900
0   21.3700    0.    0    0.    0    0.    0    0.    0 -203.0000    0.
0   81.5702    3.5090    0.    0    0.    0    0.    2  120.3003    0.
3    0.    0    0.    0    0.    0    0.    0    0.    0    0.    0    0.
0  315.3002 1264.0000   62.3200  268.2000 -151.0000 1020.0000  -66.170
0    0.    0 -297.8000    0.    0 -256.3000  312.5000 -338.5000    0.
0    0.    0    0.    0    0.    0 -313.5000    0.    0    0.    0   44.420
0 -183.4000   76.7500  212.7000    0.    0    0.    0    0.    0    0.    0
0    0.    0    0.    0    0.    0    0.    0    0.    0    0.    0    0.
0    0.    2  -11.1600    0.    0    0.    0    0.    2 -337.0003  169.300
3  127.2000    0.    0    0.    0    0.    0    0.    0    0.    0    0.
0   91.4602   97.5100    4.6800  122.9000  562.2000  529.0000  698.200
0    0.    0  286.3000  -47.5100    0.    0    0.    0  225.4000    0.
0    0.    0    0.    0    0.    0    0.    0    0.    0  326.4000    0.
0  108.3000  249.2000   62.4200  464.4000    0.    0    0.    0  450.100
0   59.0200    0.    0    0.    0    0.    0 -125.9000    0.    0    0.
0    0.    2 -245.4000    0.    0    0.    0    0.    2    0.    3    0.
3    0.    0    0.    0    0.    0    0.    0    0.    0    0.    0    0.
0   34.0102   18.2500  121.3000    0.    0  747.7000  669.9000 -708.700
0    0.    0  423.2000    0.    0 -132.9000    0.    0 -197.7000    0.
0    0.    0 -141.4000    0.    0  587.3000    0.    0 1821.0000  -84.530
0    0.    0    0.    0   56.3300    0.    0    0.    0    0.    0    0.
0    0.    0    0.    0    0.    0  177.6000    0.    0    0.    0  215.000
0    0.    2    0.    0    0.    0    0.    0    0.    2  -96.8703    0.
3    0.    0    0.    0    0.    0    0.    0    0.    0    0.    0    0.
0   36.7002   51.0600  288.5000   33.6100  742.1000  649.1000  826.700
0    0.    0  552.1000    0.    0  176.5000  488.9000  -20.9300    0.
0    0.    0 -293.7000    0.    0   18.9800   74.0400 1346.0000 -157.100
0    0.    0    0.    0  -30.1000    0.    0    0.    0    0.    0  116.600
0    0.    0  -64.3800    0.    0   86.4000    0.    0    0.    0  363.700
0    0.    2  111.2000    0.    0    0.    0    0.    2  255.8003    0.
3    0.    0    0.    0    0.    0    0.    0    0.    0    0.    0    0.
0  -78.4502  160.9000   -4.7000  134.7000  856.3000  860.1000 1201.000
010000.0000  372.0000    0.    0  129.5000    0.    0  113.9000  261.100
0   91.1300 -126.0000 1301.0000  309.2000  492.0000  689.0000   11.800
0   17.9700   51.9000    0.    0  475.8000  490.9000  534.7000  132.200
0    0.    0  546.7000    0.    0  247.8000   41.9400    0.    0  337.700
0    0.    2  187.1000  215.2000  498.6000    0.    2  256.5003  639.300
3    0.    0    0.    0    0.    0    0.    0    0.    0    0.    0    0.
0 -141.3002 -158.8000 -237.7000  375.5000  246.9000  661.6000  920.400
0    0.    0  128.1000    0.    0 -246.3000    0.    0    0.    0  203.500
0 -108.4000 1088.0000  323.3000    0.    0  356.9000    0.    0 -314.900
0    0.    0    0.    0 -255.4000    0.    0 -154.5000    0.    0    0.
```

```
0     0.    0     0.    0     0.    0     0.    0  -60.7000    0.    0     0.
0     0.    2     0.    0     0.    0     0.    0     0.    2 -145.1003    0.
3     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0  -32.6902   -1.9960   10.3800  -97.0500  341.7000  252.6000  417.900
0     0.    0 -142.6000    0.    0     0.    0     0.    0     0.    0  -94.4900    0.
0     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0     0.    0     0.    0   -34.6800  794.4000    0.    0   533.2000    0.
0     0.    0     0.    0   139.8000  304.3000   10.1700  -27.7000    0.
0     0.    2   10.7600    0.    0     0.    0     0.    2     0.    3     0.
3     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0  5541.0002    0.    0  1824.0000 -127.8000  561.6000    0.    0   360.700
0     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0     0.    0     0.    0  5250.0000    0.    0     0.    0     0.    0     0.
0     0.    0     0.    0   514.6000    0.    0   -85.1200    0.    0     0.
0     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0     0.    2     0.    0     0.    0     0.    0     0.    2     0.    3     0.
3     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0  -52.6502   16.6200   21.5000   40.6800  823.5000  914.2000 1081.000
0     0.    0   303.7000    0.    0   243.8000    0.    0   112.4000    0.
0     0.    0     0.    0     0.    0     0.    0   335.7000    0.    0   -73.090
0     0.    0   -26.0600  -60.7100    0.    0     0.    0     0.    0     0.
0     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0     0.    2   -47.3700    0.    0     0.    0     0.    2   469.8003    0.
3     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0   -7.4812    0.    0   28.4100    0.    0   461.6000  382.8000    0.
0     0.    0   160.6000    0.    0     0.    0   239.8000   63.7100  106.700
0     0.    0     0.    0     0.    0     0.    0   125.7000    0.    0   -27.940
0     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0   31.660
0     0.    2     0.    0     0.    0   78.9200    0.    2     0.    3     0.
3     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0  -25.3102    0.    0   157.3000  404.3000  521.6000    0.    0   23.480
0     0.    0   317.5000    0.    0  -146.3000    0.    0     0.    0     0.
0     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0     0.    0   48.4800 -133.1000    0.    0     0.    0     0.    0     0.
0     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0     0.    2     0.    0     0.    0     0.    0     0.    2     0.    3     0.
3     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0  140.0002    0.    0   221.4000  150.6000  267.6000    0.    0     0.
0  838.4000    0.    0     0.    0   152.0000    0.    0   9.2070    0.
0     0.    0     0.    0   164.4000    0.    0     0.    0     0.    0     0.
0     0.    0     0.    0     0.    0     0.    0   481.3000    0.    0     0.
0     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0  -417.200
0     0.    2     0.    0     0.    0   302.2000    0.    2     0.    3     0.
3     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0  128.0002    0.    0   58.6800    0.    0   501.3000    0.    0     0.
0     0.    0   138.0000    0.    0   21.9200    0.    0   476.6000    0.
0     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0  -40.8200   21.7600   48.4900    0.    0   64.2800    0.    0     0.
0     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0     0.    2     0.    0     0.    0     0.    0     0.    2   68.5503    0.
3     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0  -31.5202    0.    0   155.6000  291.1000  721.9000    0.    0     0.
0     0.    0 -142.6000    0.    0     0.    0     0.    0   736.4000    0.
0     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0  1169.000
0     0.    0     0.    0   225.8000  224.0000  125.3000    0.    0     0.
0     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0     0.    2     0.    0     0.    0     0.    0     0.    2     0.    3     0.
3     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.
0  -72.8802   41.3800    0.    0     0.    0     0.    0     0.    0     0.    0.
0     0.    0   443.6000    0.    0     0.    0     0.    0     0.    0     0.    0.
0     0.    0     0.    0     0.    0     0.    0   329.1000    0.    0     0.    0.
0     0.    0     0.    0     0.    0     0.    0   174.4000    0.    0     0.    0.
0     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0.
0     0.    2     0.    0     0.    0  -119.8000    0.    2     0.    3     0.    0.
3     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0     0.    0.
```

```
0   50.4902  422.4000   -2.5040 -143.2000  -25.8700  695.0000 -240.000
0    0.   0  110.4000    0.   0   41.5700    0.   0 -122.1000    0.
0    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0 -215.0000 -343.6000  -58.4300    0.   0    0.   0    0.   0    0.
0   85.7000    0.   0  535.8000    0.   0    0.   0    0.   0    0.
0    0.   2    0.   0    0.   0  -97.7100    0.   2  153.7003    0.
3    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0 -165.9002    0.   0    0.   0    0.   0    0.   0    0.   0  386.600
0    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0    0.   0    0.   0    0.   0    0.   0  -42.3100    0.   0    0.
0    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0    0.   2    0.   0    0.   0    0.   0    0.   2  423.4003    0.
3    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
2   47.4102  124.2002  395.8002    0.   2  738.9002  528.0002    0.
2    0.   2  -40.9002    0.   2   16.9902    0.   2 -217.9002    0.
2    0.   2    0.   2    0.   2    0.   2  304.0002  898.2002  428.500
2    0.   2 -149.8002  134.2002    0.   2  379.4002    0.   2  167.900
2    0.   2    0.   2    0.   2    0.   2    0.   2    0.   2    0.
2    0.   0    0.   2    0.   2    0.   2    0.   2  730.8003    0.
3    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0   -5.1322    0.   0 -237.2000 -157.3000  649.7000  645.9000    0.
0    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0    0.   0    0.   0 -124.6000    0.   0    0.   0    0.   0    0.
0    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0    0.   2    0.   0    0.   0    0.   0    0.   2    0.   3    0.
3    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0  -31.9502  249.0000 -133.9000 -240.2000   64.1600  172.2000 -287.100
0    0.   0   97.0400    0.   0    0.   0    0.   0 -158.2000    0.
0    0.   0    0.   0  335.6000    0.   0    0.   0    0.   0    0.
0    0.   0    0.   0 -186.7000    0.   0    0.   0    0.   0    0.
0  -71.0000    0.   0 -191.7000    0.   0    0.   0    6.6990  136.600
0    0.   2    0.   0    0.   0    0.   0    0.   2    0.   3    0.
3    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0  147.3002    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
0    0.   2    0.   0    0.   0    0.   0    0.   2    0.   3    0.
3    0.   0    0.   0    0.   0    0.   0    0.   0    0.   0    0.
2  529.0002 1397.0002  317.6002  615.8002   88.6302  171.0002  284.400
2 -167.3002  123.4002    0.   2 -234.9002   65.3702 -247.8002    0.
2  284.5002    0.   2    0.   2    0.   2  -61.6002 1179.0002    0.
2  305.4002 -193.0002  335.7002 1107.0002    0.   2    0.   2  885.500
2    0.   2    0.   2    0.   2  288.1002    0.   2    0.   2  -29.340
2  -53.9102 -198.0002    0.   2    0.   2    0.   2    0.   3    0.
3    0.   2    0.   2    0.   2    0.   2    0.   2    0.   2    0.
3   97.9003    0.   3 -184.3003  191.6003   85.1903    0.   3    0.
3    0.   3    0.   3    0.   3    0.   3    0.   3    0.   3    0.
3    0.   3    0.   3    0.   3    0.   3    0.   3 2450.   3    0.
3    0.   3    0.   3    0.   3    0.   3    0.   3    0.   3    0.
3    0.   3    0.   3    0.   3    0.   3    0.   3    0.   3    0.
3    0.   3    0.   3    0.   3    0.   3    0.   3    0.   3    0.
3  498.8  3    0.   3    0.   3    0.   3    0.   3    0.   3    0.
3  109.2003    0.   3  293.8003  221.8003   84.8503    0.   3    0.
3    0.   3    0.   3    0.   3    0.   3    0.   3    0.   3    0.
3    0.   3    0.   3    0.   3    0.   3    0.   3 2496.0003    0.
3    0.   3    0.   3  102.0003    0.   3    0.   3    0.   3    0.
3    0.   3    0.   3    0.   3    0.   3    0.   3    0.   3    0.
3    0.   3    0.   3    0.   3    0.   3    0.   3    0.   3  639.300
3    0.   3    0.   3    0.   3    0.   3    0.   3    0.   3    0.
1  272.0001    0.   1 -288.0001-1020.0001    0.   1 -668.0001-1080.000
1    0.   1 -435.0001 -686.0001 -463.0001    0.   1 2880.0000    0.
1    0.   1    0.   1    0.   1    0.   1    0.   1    0.   1    0.
```

1	0.	1	0.	1	0.	1	0.	1	0.	1	0.	1	0.
1	0.	1	0.	1	0.	1	0.	1	0.	1	0.	1	0.
1	0.	1	0.	1	0.	1	0.	1	0.	1	0.	1	0.
1	0.	1	0.	1	0.	1	0.	1	0.	1	0.	1	0.
1	8960.0001		-963.0001		-63.1001		-196.0001		0.	1	0.	1	0.
1	0.	1	-444.0001		-167.0001		0.	1	0.	1	-74.7001		0.
1	0.	1	0.	1	0.	1	0.	1	0.	1	0.	1	0.
1	0.	1	0.	1	0.	1	0.	1	0.	1	0.	1	0.
1	0.	1	0.	1	0.	1	0.	1	0.	1	0.	1	0.
1	0.	1	0.	1	0.	1	0.	1	0.	1	0.	1	0.
1	0.	1	0.	1	0.	1	0.	1	0.	1	0.	1	0.
1	-11.1001		0.	1	-11.8001		-36.6001		0.	1	0.	1	0.
1	0.	1	1530.0001		-60.8001		-466.0001		0.	1	0.	1	0.
1	0.	1	0.	1	0.	1	0.	1	0.	1	0.	1	0.
1	0.	1	0.	1	0.	1	0.	1	0.	1	0.	1	0.
1	0.	1	0.	1	0.	1	0.	1	0.	1	0.	1	0.
1	0.	1	0.	1	0.	1	0.	1	0.	1	0.	1	0.
1	0.	1	0.	1	0.	1	0.	1	0.	1	0.	1	0.

APPENDIX 2: MOLACCS PROGRAM LISTING

```
C *** MOLACCS ***
c\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
c\                                                              \
c\     COPYRIGHT (C)  1986,  CHARLES L. BROOKS III              \
c\     ALL RIGHTS RESERVED                                      \
c\                                                              \
c\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
c*******                                                 *******c
c*******                                                 *******c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      Program  MOLACCS
c***************************************************************c
c***************************************************************c
c                                                               c
c    Author         Charles L. Brooks III                       c
c                   Department of Chemistry                     c
c                   Carnegie-Mellon University                  c
c                   Pittsburgh, Pennsylvania  15213             c
c                                                               c
c***************************************************************c
c
c  Program written summer, 1986.  Purpose of the program is to;
c       1)  Build a molecular structure from internal coordinate
c           inputs.
c
c       2)  Calculate the surface accessible and contact surface
c           areas in accord to solvent radius.
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      Common/param/ atom_type(20), bond_length(20), bond_angle(20),
     *              atomic_radius(20), hydrophobicity(20), natm_type
      Real bond_length, bond_angle, atomic_radius
      Integer hydrophobicity, natm_type
      Character*4 atom_type
c
c  The varibles in Common/param/ list the dictionary of atomic parameters.
c
c       atom_type      -->  atomic type name and character, e.g., CSP3
c                           for a carbon with SP3 hybridization
c
c       bond_length    -->  atomic bonding radius for atom_type.  Here we
c                           use the approximation that the total bondlength
c                           between two atoms is the sum of their bonding
c                           radii.
c
c       bond_angle     -->  bond angle around central atom_type, e.g.,
c                           around SP3 hybridized atom have angle = 109.54
c
c       atomic_radius  -->  van der Waals radius for atom_type
c
c       hydrophobicity -->  number between 0 and 100 indicating
c                           hydrophobicity of atom_type, e.g., CSP3
c                           might have a value of 100 for fully hydrophobic
c
c
      Common/molfrag/ mol_name(100), natom_mol(100), coor_mol(100,40,3),
     *                atype_mol(100,40), nmol_frag
      Real coor_mol
      Integer natom_mol, atype_mol, nmol_frag
      Character*8 mol_name
c
c  The varibles in Common/molfrag/ list a dictionary of available molecular
c  fragments which may be used in building new molecules or ...
c  accessibility calculations.
c
c       mol_name       -->  character identifier for molecular fragment name
c
c       natom_mol      -->  number of atoms in fragment mol_name
c
c       coor_mol       -->  atomic Cartesian coordinates for fragment
c                           mol_name
c
c       atom_type      -->  atom type identifier (see Common/param/) for
c                           atoms in fragment mol_name
c
c       nmol_frag      -->  current total of molecular fragments in
c                           dictionary
c
c  declare local varibles
      Character*10 line
      Character*70 lineb
      Logical log
      Data log/.false./
c
c  print banner and some initial instructions
      Write(6,*)'                     MOLACCS'
      Write(6,*)'          MOLecular ACCessible Surface'
      Write(6,*)'                 Version May-86'
      Write(6,*)' '
      Write(6,*)'          Author Charles L. Brooks III'
      Write(6,*)'             Department of Chemistry'
      Write(6,*)'            Carnegie-Mellon University'
      Write(6,*)'              Pittsburgh, PA  15213'
      Write(6,*)' '
      Write(6,*)
     *' This program will compute molecular surface areas using the '
      Write(6,*)' Richards algorithm with varying solvent probe radii.'
100   Continue
      Write(6,*)' Prompts are given by the symbol -->, and default'
      Write(6,*)' answer is given in parenthesis, e.g., '
      Write(6,*)' Do you want to input parameters? (n) -->'
      Write(6,*)' Do you understand? (n) -->'
      Read(5,'(a10)') line
      Do 1 i=1,10
        if( (line(i:i).eq.'y') .or. (line(i:i).eq.'Y') ) go to 110
1     Continue
      go to 100
110   Continue
      Write(6,*)' Congratulations, you got it!'
      Write(6,*)' Now onto more interesting things.'
200   Continue
      nfrag=0
      Write(6,*)' Do you want to input atomic parameters <param>, '
      Write(6,*)' set-up a molecular fragment <frag>, calculate'
      Write(6,*)' a surface area <surface> or quit <quit>
     *? (surface) -->'
      Read(5,'(a10)') line
      Do 2 i=1,10
        write(6,'(a)') line(i:i)
        if( (line(i:i).eq.'p') .or. (line(i:i).eq.'P') ) go to 210
        if( (line(i:i).eq.'f') .or. (line(i:i).eq.'F') ) go to 220
        if( (line(i:i).eq.'s') .or. (line(i:i).eq.'S') ) go to 230
        if( (line(i:i).eq.'q') .or. (line(i:i).eq.'Q') ) go to 999
        write(6,'(a)') line(i:i)
2     Continue
210   Call Newparam
```

```
220   Call Newfrag(nfrag)
230   Continue
      If (.not. log) then
        Write(6,*)
     *  ' Do you want to set up a LOG file to store '
        Write(6,*)' the surface area calculations?  (n) ==>'
        Read (5,'(a10)') line
        Do 5 i=1,10
          if( (line(i:i).eq.'y') .or. (line(i:i).eq.'Y') ) go to 500
5       Continue
        go to 510
500     Continue
        Write(6,*)' What is file name, <filename.ext>? (for093.dat) ==>'
        Read(5,'(a)') lineb
        ifrst=0
        ilst=0
        do 6 i=1,70
          if (lineb(i:i).eq.' '.and.ifrst.eq.(i-1)) ifrst=i
          if((ifrst.ne.i.and.ilst.le.0).and.lineb(i:i).eq.' ') ilst=i
6       continue
        if(ifrst.eq.0) ifrst=1
        ilst=ilst-1
        if(ifrst.ge.ilst) then
          ifrst=1
          ilst=10
          lineb='for093.dat'
        endif
        Write(6,'(a)')
     *  ' File '//lineb(ifrst:ilst)//
     *  ' status=new will be opened'
C
          open(file=lineb(ifrst:ilst),form='formatted',status='new',
     *         access='sequential',err=999,unit=93,shared)
C
        log=.true.
      End if
510   If (nfrag.eq.0) then
        Write(6,*)' Which molecular fragment do you want?'
        Write(6,*)
     *  ' list identifiers <list> or input fragment # (input) ==>'
        Read(5,'(a10)') line
        Do 3 i=1,10
          if( (line(i:i).eq.'l') .or. (line(i:i).eq.'L') ) go to 300
3       Continue
        go to 310
300     Write(6,'(3x,i3,1x,a8,1x,i3,1x,a8,1x,i3,1x,a8,1x,i3,1x,a8)')
     *  (i, mol_name(i), i=1,nmol_frag)
310     Write(6,*)' Input fragment # ==>'
        Read(5,*)nfrag
      else
        Write(6,*)
     *  ' Fragment ',nfrag,' to be used in surface calculation'
        Write(6,*)' Is that what you want? (y) ==>'
        Read(5,'(a10)') line
        Do 4 i=1,10
          if( (line(i:i).eq.'n') .or. (line(i:i).eq.'N') ) go to 40
4       Continue
        go to 400
40      nfrag=0
        go to 230
      end if
400   Call Calsurf(nfrag,log,93)
      go to 200
999   Stop   'MOLACCS normal termination'
      End
C
C
      Subroutine Newparam
C
C  This subroutine reads and writes a binary parameter file as well as
C  reads in new parameters from the command stream
      Common/param/ atom_type(20), bond_length(20), bond_angle(20),
     *              atomic_radius(20), hydrophobicity(20), natm_type
      Real bond_length, bond_angle, atomic_radius
      Integer hydrophobicity, natm_type
      Character*4 atom_type
C
C  local varibles
      Character*10 line
      Character*70 lineb
      Logical old, change
C
C
      old=.false.
      change=.false.
      If (natm_type.eq.0) then
        Write(6,*)' Current parameter file is empty or non-existent'
        Write(6,*)' Do you want to create a new file <new>,'
        Write(6,*)' or read an old one <old>?  (new) ==>'
        Read(5,'(a10)') line
        do 1 i=1,10
          if ( (line(i:i).eq.'o') .or. (line(i:i).eq.'O') ) old=.true.
1       continue
        Write(6,*)' What is file name, <filename.ext>? (for090.dat) ==>'
        Read(5,'(a)') lineb
        ifrst=0
        ilst=0
        do 2 i=1,70
          if (lineb(i:i).eq.' '.and.ifrst.eq.(i-1)) ifrst=i
          if((ifrst.ne.i.and.ilst.le.0).and.lineb(i:i).eq.' ') ilst=i
2       continue
        if(ifrst.eq.0) ifrst=1
        ilst=ilst-1
        if(ifrst.ge.ilst) then
          ifrst=1
          ilst=10
          lineb='for090.dat'
        endif
        if (old) then
          Write(6,'(a)')
     *    ' File '//lineb(ifrst:ilst)//
     *    ' status=old will be opened and read'
C
          open(file=lineb(ifrst:ilst),form='unformatted',status='old',
     *         access='append',err=999,unit=90,shared)
          Rewind 90
          Read(90) natm_type, ( atom_type(i), bond_length(i),
     *         -    bond_angle(i), atomic_radius(i), hydrophobicity(i),
     *              i=1,natm_type )
        else
          Write(6,'(a)')
     *    ' File '//lineb(ifrst:ilst)//
     *    ' status=new will be opened and read'
C
          open(file=lineb(ifrst:ilst),form='unformatted',status='new',
     *         access='sequential',err=999,unit=90,shared)
        end if
      end if
```

```
C
C
      Write(6,*)
     *' Do you want to list current atom parameters? (n) ==>'
      Read(5,'(a10)') line
      do 6 i=1,10
        if ( (line(i:i).eq.'y') .or. (line(i:i).eq.'Y') ) go to 60
6     continue
      go to 61
60    continue
      Write(6,'(a,/)')
     *' Atom #   Name   Bond   Angle   vdW_radius   Hydrophobicity'
      Write(6,'(3x,i3,2x,a4,3x,f8.3,3x,f8.3,3x,f8.3,3x,i3)')
     *(iatm,Atom_type(iatm), bond_length(iatm),
     *bond_angle(iatm), atomic_radius(iatm),
     *hydrophobicity(iatm), iatm=1,natm_type)
61    continue
C
C  Now read new parameters
      Write(6,*)' Do you want to input parameters? (n) ==>'
      Read(5,'(a10)') line
      do 3 i=1,10
        if ( (line(i:i).eq.'y') .or. (line(i:i).eq.'Y') )
     *  go to 30
3     continue
      go to 71
30    continue
      change=.true.
      Write(6,*)' Number of new atom types to input? ==>'
      Read (5,*) iatm
      Write(6,*)'                Input new atom types, total of',iatm
      do 4 i=1,iatm
        natm_type=natm_type+1
        Write(6,'(a,1x,i3,a)')' Atom', i,' Atom_type ==>'
        Read(5,'(a)') line
        ifrst=0
        ilst=0
        do 5 j=1,10
          if (line(j:j).eq.' '.and.ifrst.eq.(j-1)) ifrst=j
          if((ifrst.ne.j.and.ilst.le.0).and.line(j:j).eq.' ') ilst=j
5       continue
        if(ifrst.eq.0) ifrst=1
        ilst=ilst-1
C  Convert string to upper case
        Call Cnvtuc(line(ifrst:ifrst+4),4)
C        Write(6,'(1x,a,2x,a)')' Testing',line(ifrst:ilst)
        Read(line(ifrst:ilst),'(a4)') Atom_type(natm_type)
C        Write(6,'(1x,a,2x,a)')' Testing', Atom_type(natm_type)
        Write(6,*)
     *  ' Bond   Angle   vdW_radius   Hydrophobicity'
        Read (5,*) bond_length(natm_type),
     *  bond_angle(natm_type), atomic_radius(natm_type),
     *  hydrophobicity(natm_type)
C        Write (5,*)' Testing', bond_length(natm_type),
C     *  bond_angle(natm_type), atomic_radius(natm_type),
C     *  hydrophobicity(natm_type)
4     continue
C
C  Now make any changes in parameters
71    continue
      Write(6,*)' Do you want to change existing parameters? (n) ==>'
      Read(5,'(a10)') line
      do 7 i=1,10
        if ( (line(i:i).eq.'y') .or. (line(i:i).eq.'Y') )
     *  go to 70
7     continue
      go to 31
70    continue
      change=.true.
      Write(6,*)' Number of atom to make changes for? ==>'
      Read (5,*) iatm
      Write(6,'(a)')' Input new parameters for atom '//atom_type(iatm)
      Write(6,*)
     *'          Bond   Angle   vdW_radius   Hydrophobicity'
      Write(6,*)' old ==>', bond_length(iatm),
     *bond_angle(iatm), atomic_radius(iatm),
     *hydrophobicity(iatm)
      Write(6,*)' new ==>'
      Read (5,*) bond_length(iatm),
     *bond_angle(iatm), atomic_radius(iatm),
     *hydrophobicity(iatm)
31    continue
      if ( change ) then
        Write(6,*)' New/Updated parameter file written'
        Rewind 90
        Write(90) natm_type, ( atom_type(i), bond_length(i),
     *            bond_angle(i), atomic_radius(i), hydrophobicity(i),
     *            i=1,natm_type )
      endif
      Return
999   Stop  'Newparam:  Problems with parameter file'
      End
C
C
      Subroutine Newfrag(nfrag)
      Common/molfrag/ mol_name(100), natom_mol(100), coor_mol(100,40,3),
     *                atype_mol(100,40), nmol_frag
      Real coor_mol
      Integer natom_mol, atype_mol, nmol_frag
      Character*8 mol_name
      Common/param/ atom_type(20), bond_length(20), bond_angle(20),
     *              atomic_radius(20), hydrophobicity(20), natm_type
      Real bond_length, bond_angle, atomic_radius
      Integer hydrophobicity, natm_type
      Character*4 atom_type
C
C  local varibles
      Character*10 line
      Character*70 lineb
      Logical old, add
C
C
      old=.false.
      If (nmol_frag.eq.0) then
        Write(6,*)' Current fragment file is empty or non-existent'
        Write(6,*)' Do you want to create a new file <new>,'
        Write(6,*)' or read an old one <old>?  (new) ==>'
        Read(5,'(a10)') line
        do 4 i=1,10
          if ( (line(i:i).eq.'o') .or. (line(i:i).eq.'O') ) old=.true.
4       continue
        Write(6,*)' What is file name, <filename.ext>? (for091.dat) ==>'
        Read(5,'(a)') lineb
        ifrst=0
        ilst=0
        do 5 i=1,70
          if (lineb(i:i).eq.' '.and.ifrst.eq.(i-1)) ifrst=i
          if((ifrst.ne.i.and.ilst.le.0).and.lineb(i:i).eq.' ') ilst=i
```

```
5       continue
        if(ifrst.eq.0) ifrst=1
        ilst=ilst-1
        if(ifrst.ge.ilst) then
          ifrst=1
          ilst=10
          lineb='for091.dat'
        endif
        if (old) then
          Write(6,'(a)')
     *    ' File '//lineb(ifrst:ilst)//
     *    ' status=old will be opened and read'
C
          open(file=lineb(ifrst:ilst),form='unformatted',status='old',
     *         access='append',err=999,unit=91,shared)
          Rewind 91
          Read(91) nmol_frag, ( mol_name(i), natom_mol(i), (
     *             atype_mol(i,j), ( coor_mol(i,j,k), k=1,3 ),
     *             j=1,natom_mol(i) ), i=1, nmol_frag )
          Write(6,*) nmol_frag,' fragments read from fragment file'
        else
          Write(6,'(a)')
     *    ' File '//lineb(ifrst:ilst)//
     *    ' status=new will be opened and read'
C
          open(file=lineb(ifrst:ilst),form='unformatted',status='new',
     *         access='sequential',err=999,unit=91,shared)
        end if
      end if
C
C
101   Write(6,*)' List existing fragments <list>,'
100   Write(6,*)' build new fragment <build>,'
      Write(6,*)' read in new fragment <read>,'
      Write(6,*)' perturb an existing fragment <perturb>,'
      Write(6,*)' combine two existing fragments <combine>,'
      Write(6,*)' delete an existing fragment <delete>'
      Write(6,*)
     *' or choose fragment for surface calculation <choose>?
     *(choose) ==>'
      Read(5,'(a10)') line
C  Convert line to upper case
      Call Cnvtuc(line,10)
      do 1 i=1,9
        if (line(i:i).eq.'L') go to 10
        if (line(i:i).eq.'B') go to 11
        if (line(i:i).eq.'R') go to 12
        if (line(i:i).eq.'P') go to 13
        if ( (line(i:i).eq.'C') .and. (line(i+1:i+1).eq.'O') ) go to 14
        if (line(i:i).eq.'D') go to 15
1     continue
      go to 16
C
C  List existing fragments
10    continue
      Write(6,'(3x,i3,1x,a8,1x,i3,1x,a8,1x,i3,1x,a8,1x,i3,1x,a8,1x,i3,1x,a8)')
     *(i, mol_name(i), i=1,nmol_frag)
      go to 100
C
C  Build new fragment
11    n=nmol_frag+1
      nfrag=n
      Call Build(nfrag)
C  Draw the new fragment to make sure its right
      Call Drawfrg(nfrag)
      Write(6,*)' Save fragment in the library file? (n) ==>'
      Read (5,'(a10)') line
      do 2 i=1,10
        if(line(i:i).eq.'y'.or.line(i:i).eq.'Y') go to 20
2     continue
      Return
20    continue
      nmol_frag=nmol_frag+1
      Write(6,*)
     *' Fragment ',mol_name(nfrag),' added to file and saved'
      Rewind 91
      Write(91) nmol_frag, ( mol_name(i), natom_mol(i), (
     *          atype_mol(i,j), ( coor_mol(i,j,k), k=1,3 ),
     *          j=1,natom_mol(i) ), i=1, nmol_frag )
      Return
C
C  Read in new fragment
12    n=nmol_frag+1
      nfrag=n
      Call Readfrag(nfrag)
C  Draw the new fragment to make sure its right
      Call Drawfrg(nfrag)
      Write(6,*)' Save fragment in the library file? (n) ==>'
      Read (5,'(a10)') line
      do 6 i=1,10
        if(line(i:i).eq.'y'.or.line(i:i).eq.'Y') go to 60
6     continue
      Return
60    continue
      nmol_frag=nmol_frag+1
      Write(6,*)
     *' Fragment ',mol_name(nfrag),' added to file and saved'
      Rewind 91
      Write(91) nmol_frag, ( mol_name(i), natom_mol(i), (
     *          atype_mol(i,j), ( coor_mol(i,j,k), k=1,3 ),
     *          j=1,natom_mol(i) ), i=1, nmol_frag )
      Return
13    Continue
      add=.false.
      Write(6,*)
     *' Do you want to replace atoms in existing fragment <replace>'
      Write(6,*)' or add single atom substituents <add>? (add) ==>'
      Read (5,'(a10)') line
      Do 7 i=1,10
        if(line(i:i).eq.'r'.or.line(i:i).eq.'R') go to 70
7     Continue
      add=.true.
70    Continue
      Write(6,*)' Which molecular fragment do you want to perturb?'
      Write(6,*)
     *' list identifiers <list> or input fragment # (input) ==>'
      Read(5,'(a10)') line
      Do 8 i=1,10
        if( (line(i:i).eq.'l') .or. (line(i:i).eq.'L') ) go to 80
8     Continue
      go to 81
80    Write(6,'(3x,i3,1x,a8,1x,i3,1x,a8,1x,i3,1x,a8,1x,i3,1x,a8,1x,i3,1x,a8)')
     *(i, mol_name(i), i=1,nmol_frag)
81    Write(6,*)' Input fragment # ==>'
      Read(5,*)nfrag
      Call Perturb(add,nfrag)
C  Draw the new fragment to make sure its right
      Call Drawfrg(nfrag)
```

```
      Write(6,*)' Save fragment in the library file? (n) -->'
      Read (5,'(a10)') line
      do 9 i=1,10
        if(line(i:i).eq.'y'.or.line(i:i).eq.'Y') go to 60
9     continue
      Return
14    Continue
C
C
      Write(6,*)' Which molecular fragments do you want to combine?'
      Write(6,*)
     *' list identifiers <list> or input fragment # (input) -->'
      Read(5,'(a10)') line
      Do 91 i=1,10
        if( (line(i:i).eq.'l') .or. (line(i:i).eq.'L') ) go to 990
91    Continue
      go to 991
990   Write(6,'(3x,i3,1x,a8,1x,i3,1x,a8,1x,i3,1x,a8,1x,i3,1x,a8)')
     *(i, mol_name(i), i=1,nmol_frag)
991   Write(6,*)' Input fragment # -->'
      Read(5,*)nfrag,nfrag1
      Call Combine(nfrag,nfrag1)
      Write(6,*)' Save fragment in the library file? (n) -->'
      Read (5,'(a10)') line
      do 92 i=1,10
        if(line(i:i).eq.'y'.or.line(i:i).eq.'Y') go to 60
92     continue
       Go to 16
C
C
15    Write(6,*)
     *' Which fragment do you want to delete, specify number? -->'
      Read (5,*) nfrag
      Call Deletfrg(nfrag)
      nfrag=0
      go to 100
C
16    Return
999   Stop 'Newfrag:  Problems with fragment file'
      End
C
C
      Subroutine Deletfrg(nfrag)
C  Delete a fragment and update file
C
      Common/molfrag/ mol_name(100), natom_mol(100), coor_mol(100,40,3),
     *                atype_mol(100,40), nmol_frag
      Real coor_mol
      Integer natom_mol, atype_mol, nmol_frag
      Character*8 mol_name
C
C  local varible
      Character*10 line
C
      Write(6,'(a,i3,a)')
     *' Fragment ',nfrag,' will be deleted,
     *is that what you want? (n) -->'
      Read (5,'(a10)') line
      do 1 i=1,10
        if(line(i:i).eq.'y'.or.line(i:i).eq.'Y') go to 10
1     continue
      Return
10    Continue
      Do 2 i=nfrag+1,nmol_frag
        mol_name(i-1)=mol_name(i)
        natom_mol(i-1)=natom_mol(i)
        Do 2 j=1,natom_mol(i-1)
          atype_mol(i-1,j)=atype_mol(i,j)
          Do 2 k=1,3
            coor_mol(i-1,j,k)=coor_mol(i,j,k)
2     Continue
C
C
      nmol_frag=nmol_frag-1
      Write(6,*)' Updated fragment library written with ',
     *nmol_frag,' molecular fragments'
      Rewind 91
      Write(91) nmol_frag, ( mol_name(i), natom_mol(i), (
     *          atype_mol(i,j), ( coor_mol(i,j,k), k=1,3 ),
     *          j=1,natom_mol(i) ), i=1, nmol_frag )
C
      Return
      End
C
C
      Subroutine Calsurf(nfrag,log,iunit)
C
C  Richards accessible surface program
C  Designed to compute both the accessible and the contact
C  and print the hydrophobic and polar contributions.
C
C
      Common/molfrag/ mol_name(100), natom_mol(100), coor_mol(100,40,3),
     *                atype_mol(100,40), nmol_frag
      Real coor_mol
      Integer natom_mol, atype_mol, nmol_frag
      Character*8 mol_name
      Common/param/ atom_type(20), bond_length(20), bond_angle(20),
     *              atomic_radius(20), hydrophobicity(20), natm_type
      Real bond_length, bond_angle, atomic_radius
      Integer hydrophobicity, natm_type
      Character*4 atom_type
C  Passed varibles
      Integer nfrag,iunit
      Logical log
C  local varibles
      Real x(40),y(40),z(40),rad(40),acces(40),acces0(40),contac(40)
      Real zlb(40),zub(40),xr1(40),yr1(40),zr1(40),rad1(40),
     *     rsec2(40),area(40),rsec(80),arci(15,40),arcf(15,40)
      Real Radius_probe,np
      Integer ict,iat,tag(40),tag1(40),inz(40),kn(40)
      Parameter(np=0.01)
      Parameter(ict=15)
C
C
      iat=0
C  Pick-up fragment information
      if( natm_type .eq. 0 ) then
        Write(6,*)' No parameters, calling Newparam'
        Call Newparam
      end if
      if( nmol_frag .eq. 0 ) then
        Write(6,*)' No fragments, calling Newfrag'
        Call Newfrag(nfrag)
      end if
      iat=natom_mol(nfrag)
      Do 1 i=1,iat
```

```
          Write(6,*)' Atom type does not exist in parameter file,'
          Write(6,*)' returning to Newparam to add it.'
          Call Newparam
        end if
        Rad(i)=atomic_radius(atype_mol(nfrag,i))
        x(i)=coor_mol(nfrag,i,1)
        y(i)=coor_mol(nfrag,i,2)
        z(i)=coor_mol(nfrag,i,3)
        acces(i)=0.0
        contac(i)=0.0
        acces0(i)=0.0
1     Continue
C
C  Gather information concerning probe radius
C
      Write(6,*)' Input solvent probe radius. ==>'
      Read (5,*) radius_probe
      If ( Radius_probe .le. 0.0 ) Radius_probe=1.4
C Call first with normal solvent radius
      Call Surfac(iat,ict,tag,tag1,inz,kn,zlb,zub,
     2            zr1,yr1,xr1,rad1,rsec2,area,rsec,
     3            arci,arcf,x,y,z,rad,np,radius_probe,
     4            acces,contac)
C  Next with radius=0.0
      Call Surfac(iat,ict,tag,tag1,inz,kn,zlb,zub,
     2            zr1,yr1,xr1,rad1,rsec2,area,rsec,
     3            arci,arcf,x,y,z,rad,np,0.0,
     4            acces0,acces0)
C
C  Calculate results
      totha=0.0
      totha0=0.0
      tothc=0.0
      totpa=0.0
      totpa0=0.0
      totpc=0.0
      Do 2 i=1,iat
        totha=totha +
     *  acces(i)*hydrophobicity(atype_mol(nfrag,i))
        totha0=totha0 +
     *  acces0(i)*hydrophobicity(atype_mol(nfrag,i))
        tothc=tothc +
     *  contac(i)*hydrophobicity(atype_mol(nfrag,i))
        totpa=totpa +
     *  acces(i)*(100-hydrophobicity(atype_mol(nfrag,i)))
        totpa0=totpa0 +
     *  acces0(i)*(100-hydrophobicity(atype_mol(nfrag,i)))
        totpc=totpc +
     *  contac(i)*(100-hydrophobicity(atype_mol(nfrag,i)))
2     Continue
      totha=totha/100.0
      totha0=totha0/100.0
      tothc=tothc/100.0
      totpa=totpa/100.0
      totpa0=totpa0/100.0
      totpc=totpc/100.0
C  Print out results
C
      Write(6,'(a)')
     *'            Surface areas for fragment '//mol_name(nfrag)
      Write(6,'(a,f8.3,/,/)')
     *'       computed with respect to probe of radius', Radius_probe
      Write(6,'(a,/)')
     *'       ACCESSIBLE                CONTACT                van der Waals'
      Write(6,'(a,f8.3,a,f8.3,a,f8.3)')
     *' Hydrophobic',totha,'          ',tothc,'      ',totha0
      Write(6,'(a,f8.3,a,f8.3,a,f8.3,/)')
     *' Polar      ',totpa,'          ',totpc,'      ',totpa0
      Write(6,'(a,/)')
     *'                              Atomic Breakdown'
      Write(6,'(2x,i2,1x,a4,3x,f8.3,13x,f8.3,10x,f8.3)')
     *(i,atom_type(atype_mol(nfrag,i)),acces(i),contac(i),
     * acces0(i),i=1,iat)
C
      If (Log) then
C
      Write(iunit,'(a)')
     *'            Surface areas for fragment '//mol_name(nfrag)
      Write(iunit,'(a,f8.3,/,/)')
     *'       computed with respect to probe of radius', Radius_probe
      Write(iunit,'(a,/)')
     *'       ACCESSIBLE                CONTACT                van der Waals'
      Write(iunit,'(a,f8.3,a,f8.3,a,f8.3)')
     *' Hydrophobic',totha,'          ',tothc,'      ',totha0
      Write(iunit,'(a,f8.3,a,f8.3,a,f8.3,/)')
     *' Polar      ',totpa,'          ',totpc,'      ',totpa0
      Write(iunit,'(a,/)')
     *'                              Atomic Breakdown'
      Write(iunit,'(2x,i2,1x,a4,3x,f8.3,13x,f8.3,10x,f8.3)')
     *(i,atom_type(atype_mol(nfrag,i)),acces(i),contac(i),
     * acces0(i),i=1,iat)
      End if
      Return
      End
C
C
      Subroutine Build(nfrag)
      Common/molfrag/ mol_name(100), natom_mol(100), coor_mol(100,40,3),
     *                atype_mol(100,40), nmol_frag
      Real coor_mol
      Integer natom_mol, atype_mol, nmol_frag
      Character*8 mol_name
C
      Common/param/ atom_type(20), bond_length(20), bond_angle(20),
     *              atomic_radius(20), hydrophobicity(20), natm_type
      Real bond_length, bond_angle, atomic_radius
      Integer hydrophobicity, natm_type
      Character*4 atom_type
C  passed varibles
      Integer nfrag
C  local varibles
      Real Phi(40)
      Character*10 line
      Character*70 lineb
C
C
      Write(6,*)
     *' A single unbranched molecular fragment will be built.'
      Write(6,*)' Is that what you want? (yes) ==>'
      Read (5,'(a10)') line
      Do 1 i=1,10
        if(line(i:i).eq.'n'.or.line(i:i).eq.'N') go to 10
1     Continue
      go to 11
10    Write(6,*)' Returning to Newfrag...'
      Write(6,*)' Use COMBINE to make branched chains.'
      Return
11    Write(6,*)' What is new fragment name, only 8 characters? ==>'
```

```
      Read(5,'(a)') lineb
      ifrst=0
      ilst=0
      do 2 i=1,70
        if (lineb(i:i).eq.' '.and.ifrst.eq.(i-1)) ifrst=i
        if((ifrst.ne.i.and.ilst.le.0).and.lineb(i:i).eq.' ') ilst=i
2     continue
      if(ifrst.eq.0) ifrst=1
      ilst=ilst-1
      if(ifrst.ge.ilst) then
        Write(6,*)' No name read, trying again'
        go to 11
      endif
      If ((ilst-ifrst) .gt. 8) then
        Write(6,*)' Name greater than 8 characters, truncated.'
        ilst=ifrst+8
      end if
C  Convert to upper case
      Call Cnvtuc(lineb(ifrst:ilst),ilst-ifrst)
      Write(6,'(a)')
     *' New molecular fragment '//lineb(ifrst:ilst)//
     *' will be built'
      read(lineb(ifrst:ilst),'(a8)') mol_name(nfrag)
      Write(6,*)' Number of atoms in linear chain? -->'
      Read(5,*) natom_mol(nfrag)
      Write(6,*)
     *' Now specify atom types, do you want them listed? (n) -->'
      Read (5,'(a10)') line
      Do 3 i=1,10
        if(line(i:i).eq.'y'.or.line(i:i).eq.'Y') go to 30
3     Continue
      go to 31
30    Write(6,*)' Current atom type names and their numbers'
      Write(6,'(3x,a4,2x,i2,4x,a4,2x,i2)')
     *     (atom_type(i),i,i=1,natm_type)
31    Continue
      Write(6,*)
     *' Input sequence of atom type numbers in assending order'
      Read (5,*,err=999) (atype_mol(nfrag,i),i=1,natom_mol(nfrag))
      Write(6,*)
     *' The chain will be built in an all trans configuration'
      Write(6,*)' unless otherwise specified.'
      Write(6,*)' Do you want other dihedral angles? (n) -->'
C  Set up dihedral angle array. Phi(i) is for rotation around
C  bond i,i+1 and effects placement of atom i+2 w/r i-1.
C  To build all trans in x-y plane Phi(i)=90.
      pi=acos(-1.0)
      pi2=pi/2.0
      Do 4 i=1,natom_mol(nfrag)
        Phi(i)=pi
4     Continue
      Read (5,'(a10)') line
      Do 5 i=1,10
        if(line(i:i).eq.'y'.or.line(i:i).eq.'Y') go to 50
5     Continue
      go to 51
50    Write(6,*)' Set-up dihedral angle array, angle is for'
      Write(6,*)' rotation about bond i,i+1 and affects placment'
      Write(6,*)' of atom i+2 w/r i-1.'
      Write(6,*)' How many dihedrals are different from trans?'
      Read (5,*)nntran
      Do 6 j=1,nntran
        Write(6,'(a,i2)')
     *  ' Input i and angle, in degrees, for non-trans dihedral ',j
        Read (5,*) i,angle
        phi(i)=angle*pi/180.0
6     Continue
C
C  Finally build structure
51    Continue
C  Place first three atoms
      coor_mol(nfrag,1,1)=0.0
      coor_mol(nfrag,1,2)=0.0
      coor_mol(nfrag,1,3)=0.0
      coor_mol(nfrag,2,1) = 0.5*( bond_length(atype_mol(nfrag,1))
     *                  + bond_length(atype_mol(nfrag,2)) )
      coor_mol(nfrag,2,2)=0.0
      coor_mol(nfrag,2,3)=0.0
      theta=bond_angle(atype_mol(nfrag,2))
      theta=theta*pi/180.0
      bond=0.5*( bond_length(atype_mol(nfrag,2))
     *                  + bond_length(atype_mol(nfrag,3)) )
      coor_mol(nfrag,3,1)=bond*cos(pi-theta)
     *                  + coor_mol(nfrag,2,1)
      coor_mol(nfrag,3,2)=bond*sin(pi-theta)
     *                  + coor_mol(nfrag,2,2)
      coor_mol(nfrag,3,3)=0.0
      Do 7 i=4,natom_mol(nfrag)
        ityp=atype_mol(nfrag,i)
        itypm1=atype_mol(nfrag,i-1)
        phii=phi(i-2)
        theta=bond_angle(itypm1)
        theta=theta*pi/180.0
        bond=0.5*( bond_length(itypm1) + bond_length(ityp) )
        Call Addatm1(nfrag,i-3,i-2,i-1,i,Bond,Theta,Phii)
7     Continue
C      Write(6,*)' Testing in Build, coordinates'
C      Write(6,'(3x,i2,2x,f8.4,2x,f8.4,2x,f8.4)')
C     *(i, (coor_mol(nfrag,i,j),j=1,3), i=1,natom_mol(nfrag))
      Return
999   Write(6,*)' Error reading sequence, try again'
      go to 31
      End
C
C
      Subroutine Readfrag(nfrag)
      Common/molfrag/ mol_name(100), natom_mol(100), coor_mol(100,40,3),
     *                atype_mol(100,40), nmol_frag
      Real coor_mol
      Integer natom_mol, atype_mol, nmol_frag
      Character*8 mol_name
C
      Common/param/ atom_type(20), bond_length(20), bond_angle(20),
     *              atomic_radius(20), hydrophobicity(20), natm_type
      Real bond_length, bond_angle, atomic_radius
      Integer hydrophobicity, natm_type
      Character*4 atom_type
C  passed variables
      Integer nfrag
C
C  This subroutine reads in a molecular fragment from a fixed format
C  file.  The format used is as follows:
C
C    * name                                              ('* ',a8)
C    *                                                   ('*  ')
C     natom                                              (i5)
C                  x        y        z        atype      (20X,3f10.5,10x,f10.5)
C
```

```
C  local varibles
      Character*10 line
      Character*70 lineb
C
C
      Write(6,*)' A single formatted fragment file will be read in.'
10    Write(6,*)' What is file name, <filename.ext>? ==>'
      Read(5,'(a)') lineb
      ifrst=0
      ilst=0
      do 1 i=1,70
        if (lineb(i:i).eq.' '.and.ifrst.eq.(i-1)) ifrst=i
        if((ifrst.ne.i.and.ilst.le.0).and.lineb(i:i).eq.' ') ilst=i
1     continue
      if(ifrst.eq.0) ifrst=1
      ilst=ilst-1
      if(ifrst.ge.ilst) then
        Write(6,*)' No file input, trying again'
        go to 10
      endif
      Write(6,'(a)')
     *' File '//lineb(ifrst:ilst)//
     *' status=old will be opened and read'
C
      open(file=lineb(ifrst:ilst),form='formatted',status='old',
     *    access='append',err=999,unit=92,shared)
      Rewind 92
      Read(92,'(2x,a8)') mol_name(nfrag)
C  Convert name to upper case
      Call Cnvtuc(mol_name(nfrag),8)
      Read(92,'(a)') line
      if(line.eq.'  ') then
        Read(92,'(i5)') natom_mol(nfrag)
        Do 2 i=1,natom_mol(nfrag)
          Read(92,'(20x,f10.5,f10.5,f10.5,10x,f10.5)')
     *          coor_mol(nfrag,i,1),coor_mol(nfrag,i,2),
     *          coor_mol(nfrag,i,3),temp
          atype_mol(nfrag,i)=temp
2       Continue
        Write(6,'(a,a,a,i5,a)')
     *  ' Fragment ',mol_name(nfrag),' with',natom_mol(nfrag),' atoms
     *read from fragment file'
      else
        go to 999
      end if
C
      Return
999   Stop   'Readfrag:  Problems with formatted fragment file'
      End
C
      Subroutine Perturb(add,nfrag)
C
C  This routine perturbs an existing molecular fragment by:
C  1) replacing single atoms in the structure by other atom types,
C  2) adding single atom substituents.
C
      Common/molfrag/ mol_name(100), natom_mol(100), coor_mol(100,40,3),
     *                atype_mol(100,40), nmol_frag
      Real coor_mol
      Integer natom_mol, atype_mol, nmol_frag
      Character*8 mol_name
C
      Common/param/ atom_type(20), bond_length(20), bond_angle(20),
     *              atomic_radius(20), hydrophobicity(20), natm_type
      Real bond_length, bond_angle, atomic_radius
      Integer hydrophobicity, natm_type
      Character*4 atom_type
C  passed varibles
      Character*70 lineb
      Character*10 line
      Integer nfrag
      Logical add
      If ( add ) then
        Write(6,'(a)')
     *  ' Atoms will be added to fragment '//mol_name(nfrag)
      Else
        Write(6,'(a)')
     *  ' Atoms will be replaced in fragment '//mol_name(nfrag)
      Endif
11    Write(6,*)' What is new fragment name, only 8 characters? ==>'
      Read(5,'(a)') lineb
      ifrst=0
      ilst=0
      do 1 i=1,70
        if (lineb(i:i).eq.' '.and.ifrst.eq.(i-1)) ifrst=i
        if((ifrst.ne.i.and.ilst.le.0).and.lineb(i:i).eq.' ') ilst=i
1     continue
      if(ifrst.eq.0) ifrst=1
      ilst=ilst-1
      if(ifrst.ge.ilst) then
        Write(6,*)' No name read, trying again'
        go to 11
      endif
      If ((ilst-ifrst) .gt. 8) then
        Write(6,*)' Name greater than 8 characters, truncated.'
        ilst=ifrst+8
      end if
C  Convert to upper case
      Call Cnvtuc(lineb(ifrst:ilst),ilst-ifrst)
      If ( add ) then
        Write(6,'(a)')
     *  ' New molecular fragment '//lineb(ifrst:ilst)//
     *  ' will be built by addition to '//mol_name(nfrag)
      Else
        Write(6,'(a)')
     *  ' New molecular fragment '//lineb(ifrst:ilst)//
     *  ' will be built from perturbation of '//mol_name(nfrag)
      End if
        nfrag1=nmol_frag+1
        read(lineb(ifrst:ilst),'(a8)') mol_name(nfrag1)
C  Transfer coordinates, name, etc to new fragment location
      If ( add ) then
        Write(6,*)' How many atoms to add to fragment'
        Read (5,*) natm
        natom_mol(nfrag1)=natom_mol(nfrag)+natm
        Do 2 i=1,natom_mol(nfrag1)-natm
          atype_mol(nfrag1,i)=atype_mol(nfrag,i)
          coor_mol(nfrag1,i,1)=coor_mol(nfrag,i,1)
          coor_mol(nfrag1,i,2)=coor_mol(nfrag,i,2)
          coor_mol(nfrag1,i,3)=coor_mol(nfrag,i,3)
2       Continue
        Call Drawfrg(nfrag)
        Write(6,*)' Atoms will be added one at a time'
        natm0=natom_mol(nfrag)
        Do 3 i=1,natm
          Write(6,*)' Atom added where on fragment '//mol_name(nfrag)
          Write(6,*)' List i,j and k, where k is the bond to add'
          Write(6,*)' atom to add i and i and j are the k-1 k-2 atoms'
```

```
        Write(6,*)' bonded to k'
        Read (5,*) isite, jsite, ksite
        itypek=atype_mol(nfragl,ksite)
        itypej=atype_mol(nfragl,jsite)
        itypei=atype_mol(nfragl,isite)
        Write(6,*)
     *  ' Now specify atom type to be added'
        Write(6,*)' Do you want them listed? (n) ==>'
        Read (5,'(a10)') line
        Do 4 j=1,10
          if(line(j:j).eq.'y'.or.line(j:j).eq.'Y') go to 40
4       Continue
        go to 41
40      Write(6,*)' Current atom type names and their numbers'
        Write(6,'(3x,a4,2x,i2,4x,a4,2x,i2)')
     *       (atom_type(j),j,j=1,natm_type)
41      Continue
        Write(6,*)' Number of atom type to be added?'
        Read (5,*) itypel
        lsite=natm0+i
        atype_mol(nfragl,lsite)=itypel
        Theta=bond_angle(itypek)*acos(-1.0)/180.0
        Phi=acos(-1.0)
        Bond=0.5*(bond_length(itypel)+bond_length(itypek))
        Call Addatml(nfragl,isite,jsite,ksite,lsite,Bond,Theta,Phi)
3     Continue
      Else
        Write(6,*)' How many atoms to be replaced in fragment'
        Read (5,*) natm
        natom_mol(nfragl)=natom_mol(nfrag)
        Do 5 i=1,natom_mol(nfragl)
          atype_mol(nfragl,i)=atype_mol(nfrag,i)
          coor_mol(nfragl,i,1)=coor_mol(nfrag,i,1)
          coor_mol(nfragl,i,2)=coor_mol(nfrag,i,2)
          coor_mol(nfragl,i,3)=coor_mol(nfrag,i,3)
5       Continue
        Call Drawfrg(nfrag)
        Write(6,*)' Atoms will be replaced one at a time'
        natm0=natom_mol(nfrag)
        Do 6 i=1,natm
          Write(6,*)
     *    ' List atom to be replaced on fragment '//mol_name(nfrag)
          Read (5,*) ksite
          itypek=atype_mol(nfragl,ksite)
          Write(6,*)
     *    ' Now specify new atom type, do you want them listed? (n) ==>'
          Read (5,'(a10)') line
          Do 7 j=1,10
            if(line(j:j).eq.'y'.or.line(j:j).eq.'Y') go to 70
7         Continue
          go to 71
70        Write(6,*)' Current atom type names and their numbers'
          Write(6,'(3x,a4,2x,i2,4x,a4,2x,i2)')
     *       (atom_type(j),j,j=1,natm_type)
71        Continue
          Write(6,*)' Number for new atom type?'
          Read (5,*) itypel
C  Check compatibility of substitution
          if
     *    ( (abs(bond_length(itypek)-bond_length(itypel)) .gt. 0.25 )
     *                              .or.
     *      (abs(bond_angle(itypek)-bond_angle(itypel)) .gt. 10.0 ) )
     *    then
            Write(6,'(a,i2)')
     *      ' Atom type '//atom_type(itypek)//
     *      ' cannot be placed at site ',ksite,
     *      ' -inconsistant replacment'
            go to 6
          endif
          atype_mol(nfragl,ksite)=itypel
          Write(6,'(a,i2)')
     *    ' Atom type '//atom_type(itypek)//' at site ',ksite
          Write(6,'(a)')' replaced by atom type '//atom_type(itypel)
6       Continue
      End if
      nfrag=nfragl
      Return
      End
C
      Subroutine Combine(nfrag,nfragl)
C
C  This routine combines two existing fragments
C
      Common/molfrag/ mol_name(100), natom_mol(100), coor_mol(100,40,3),
     *                atype_mol(100,40), nmol_frag
      Real coor_mol
      Integer natom_mol, atype_mol, nmol_frag
      Character*8 mol_name
C
      Common/param/ atom_type(20), bond_length(20), bond_angle(20),
     *              atomic_radius(20), hydrophobicity(20), natm_type
      Real bond_length, bond_angle, atomic_radius
      Integer hydrophobicity, natm_type
      Character*4 atom_type
C  passed varibles
      Character*70 lineb
      Character*10 line
      Real Rold(3),Rnew(3),U(3,3)
      Integer nfrag,nfragl
C
C
      Write(6,'(a)')
     *' Molecular fragment '//mol_name(nfragl)//
     *' will be added to fragment '//mol_name(nfrag)
110   Write(6,*)' What is new fragment name, only 8 characters? ==>'
      Read(5,'(a)') lineb
      ifrst=0
      ilst=0
      do 1 i=1,70
        if (lineb(i:i).eq.' '.and.ifrst.eq.(i-1)) ifrst=i
        if((ifrst.ne.1.and.ilst.le.0).and.lineb(i:i).eq.' ') ilst=i
1     continue
      if(ifrst.eq.0) ifrst=1
      ilst=ilst-1
      if(ifrst.ge.ilst) then
        Write(6,*)' No name read, trying again'
        go to 110
      endif
      If ((ilst-ifrst) .gt. 8) then
        Write(6,*)' Name greater than 8 characters, truncated.'
        ilst=ifrst+8
      end if
C  Convert to upper case
      Call Cnvtuc(lineb(ifrst:ilst),ilst-ifrst)
      Write(6,'(a)')
     *' New molecular fragment '//lineb(ifrst:ilst)//
     *' will be built by combining '//mol_name(nfrag)//
     *' and '//mol_name(nfragl)
```

```
        nfrag2=nmol_frag+1
        read(lineb(ifrst:ilst),'(a8)') mol_name(nfrag2)
C  Transfer coordinates, name, etc to new fragment location
      natom_mol(nfrag2)=natom_mol(nfrag)+natom_mol(nfrag1)
      Do 2 i=1,natom_mol(nfrag)
        atype_mol(nfrag2,i)=atype_mol(nfrag,i)
        coor_mol(nfrag2,i,1)=coor_mol(nfrag,i,1)
        coor_mol(nfrag2,i,2)=coor_mol(nfrag,i,2)
        coor_mol(nfrag2,i,3)=coor_mol(nfrag,i,3)
2     Continue
100   Call Drawfrg2(nfrag,nfrag1)
      natm0=natom_mol(nfrag)
      Write(6,*)' Fragment B added where on fragment A'
      Write(6,*)' List i,j and k, where k is the bond to add'
      Write(6,*)' atom to and j and i are the k-1, k-2 atoms'
      Write(6,*)' bonded to k'
      Read (5,*) isite, jsite, ksite
      itypek=atype_mol(nfrag2,ksite)
      itypej=atype_mol(nfrag2,jsite)
      itypei=atype_mol(nfrag2,isite)
      Write(6,*)
     *' Now specify atom on fragment B where bond is formed'
      Write(6,*)' and the atom its bonded to'
      Read (5,*) lsite,msite
      itypel=atype_mol(nfrag1,lsite)
      itypem=atype_mol(nfrag1,msite)
      lsite=lsite+natm0
      msite=msite+natm0
C  Add first atom in fragment B
      Theta=bond_angle(itypek)*acos(-1.0)/180.0
      Phi=acos(-1.0)
      Bond=0.5*(bond_length(itypel)+bond_length(itypek))
      Call Addatm1(nfrag2,isite,jsite,ksite,lsite,Bond,Theta,Phi)
C
C
C
C  Translate second fragment to overlap at added atom
      rnew(1)=coor_mol(nfrag2,lsite,1)-coor_mol(nfrag1,lsite-natm0,1)
      rnew(2)=coor_mol(nfrag2,lsite,2)-coor_mol(nfrag1,lsite-natm0,2)
      rnew(3)=coor_mol(nfrag2,lsite,3)-coor_mol(nfrag1,lsite-natm0,3)
      Do 6 i=1,natom_mol(nfrag1)
        atype_mol(nfrag2,natm0+i)=atype_mol(nfrag1,i)
        Do 7 j=1,3
          coor_mol(nfrag2,natm0+i,j)=coor_mol(nfrag1,i,j)+rnew(j)
7       Continue
6     Continue
C  Now find bond angle around atom msite and rotate to correct angle
        Rkl=0.0
        Rml=0.0
        Cst=0.0
      Do 8 i=1,3
        Rkl=Rkl + ( coor_mol(nfrag2,ksite,i)-coor_mol(nfrag2,lsite,i) )
     *          * ( coor_mol(nfrag2,ksite,i)-coor_mol(nfrag2,lsite,i) )
C
        Rml=Rml + ( coor_mol(nfrag2,msite,i)-coor_mol(nfrag2,lsite,i) )
     *          * ( coor_mol(nfrag2,msite,i)-coor_mol(nfrag2,lsite,i) )
C
        Cst=Cst + ( coor_mol(nfrag2,ksite,i)-coor_mol(nfrag2,lsite,i) )
     *          * ( coor_mol(nfrag2,msite,i)-coor_mol(nfrag2,lsite,i) )
8     Continue
C
      Theta= bond_angle(itypel)*acos(-1.0)/180.0
     *     - acos(Cst/Sqrt(Rkl*Rml))
      rnew(1)=coor_mol(nfrag2,ksite,1)-coor_mol(nfrag2,lsite,1)
      rnew(2)=coor_mol(nfrag2,ksite,2)-coor_mol(nfrag2,lsite,2)
      rnew(3)=coor_mol(nfrag2,ksite,3)-coor_mol(nfrag2,lsite,3)
C
      rold(1)=coor_mol(nfrag2,msite,1)-coor_mol(nfrag2,lsite,1)
      rold(2)=coor_mol(nfrag2,msite,2)-coor_mol(nfrag2,lsite,2)
      rold(3)=coor_mol(nfrag2,msite,3)-coor_mol(nfrag2,lsite,3)
      Call Rotate(rnew,rold,u,Theta,.false.,.false.)
C  Rotate atoms beyond lsite w/r to U
      rnew(1)=coor_mol(nfrag2,lsite,1)
      rnew(2)=coor_mol(nfrag2,lsite,2)
      rnew(3)=coor_mol(nfrag2,lsite,3)
C
      Do 9 i=1,natom_mol(nfrag1)
        rold(1)=coor_mol(nfrag2,natm0+i,1)
        rold(2)=coor_mol(nfrag2,natm0+i,2)
        rold(3)=coor_mol(nfrag2,natm0+i,3)
        Do 10 j=1,3
          Do 11 k=1,3
            coor_mol(nfrag2,natm0+i,j)=coor_mol(nfrag2,natm0+i,j) +
     *        u(j,k)*( rold(k) - rnew(k) )
11        Continue
          coor_mol(nfrag2,natm0+i,j)= coor_mol(nfrag2,natm0+i,j)
     *                              + rnew(j) - rold(j)
10      Continue
9     Continue
C  Compute dihedral angle around bond ksite,lsite in sequence j,k,l,m
      Call Getdih(nfrag2,jsite,ksite,lsite,msite,Phi)
      Write(*,*)' Current dihedral value around bond connecting'
      Write(*,*)' fragments is', Phi/acos(-1.0)*180.0
      Write(*,*)' What dihedral do you want?  ==>'
      Read (*,*) Phin
C  Now rotate around ksite,lsite bond to acheive Phin
      Phin=Phin/180.0*acos(-1.0)
      Phin=Phin-Phi
      rnew(1)=coor_mol(nfrag2,ksite,1)-coor_mol(nfrag2,lsite,1)
      rnew(2)=coor_mol(nfrag2,ksite,2)-coor_mol(nfrag2,lsite,2)
      rnew(3)=coor_mol(nfrag2,ksite,3)-coor_mol(nfrag2,lsite,3)
      Call Rotate(rnew,rold,u,Phin,.false.,.true.)
C  Rotate atoms beyond lsite w/r to U
      rnew(1)=coor_mol(nfrag2,lsite,1)
      rnew(2)=coor_mol(nfrag2,lsite,2)
      rnew(3)=coor_mol(nfrag2,lsite,3)
C
      Do 12 i=1,natom_mol(nfrag1)
        rold(1)=coor_mol(nfrag2,natm0+i,1)
        rold(2)=coor_mol(nfrag2,natm0+i,2)
        rold(3)=coor_mol(nfrag2,natm0+i,3)
        Do 13 j=1,3
          Do 14 k=1,3
            coor_mol(nfrag2,natm0+i,j)=coor_mol(nfrag2,natm0+i,j) +
     *        u(j,k)*( rold(k) - rnew(k) )
14        Continue
          coor_mol(nfrag2,natm0+i,j)= coor_mol(nfrag2,natm0+i,j)
     *                              + rnew(j) - rold(j)
13      Continue
12    Continue
C  Compute dihedral angle around bond ksite,lsite in sequence j,k,l,m
      Call Getdih(nfrag2,jsite,ksite,lsite,msite,Phi)
      Write(*,*)' Current dihedral value around bond connecting'
      Write(*,*)' fragments is', Phi/acos(-1.0)*180.0
      Call Drawfrg(nfrag2)
      Write(6,*)' Is this the configuration you want? (yes) ==>'
      Read (5,'(a10)') line
      Do 15 i=1,10
```

```
        if(line(i:i).eq.'n'.or.line(i:i).eq.'N') go to 100
15    Continue
      nfrag=nfrag2
      Return
      End
C
      Subroutine Drawfrg(nfrag)
      Common/molfrag/ mol_name(100), natom_mol(100), coor_mol(100,40,3),
     *                atype_mol(100,40), nmol_frag
      Real coor_mol
      Integer natom_mol, atype_mol, nmol_frag
      Character*8 mol_name
C  Local varibles
      Character*1 draw(40,20),one
      one='1'
      ione=ichar(one)
C
      Write(6,*)
     *' Molecular fragment '//mol_name(nfrag)//' will be drawn'
      xmax=0.0
      xmin=0.0
      ymax=0.0
      ymin=0.0
      Do 1 i=1,40
      Do 1 j=1,20
        Draw(i,j)=' '
1     Continue
C  Find maximum dimension in either x or y
      Do 2 i=1,natom_mol(nfrag)
        xmax=max( coor_mol(nfrag,i,1), xmax )
        xmin=min( coor_mol(nfrag,i,1), xmin )
        ymax=max( coor_mol(nfrag,i,2), ymax )
        ymin=min( coor_mol(nfrag,i,2), ymin )
2     Continue
      xrange=(xmax-xmin)/38.0
      yrange=(ymax-ymin)/18.0
      Do 3 i=1,natom_mol(nfrag)
        ix=int( (coor_mol(nfrag,i,1)-xmin)/xrange ) + 1
        iy=int( (coor_mol(nfrag,i,2)-ymin)/yrange ) + 1
        if ( i.ge.10 ) then
          ifact=i/10
          ichr1=ifact-1
          ichr2=i-10*ifact-1
          draw(ix,iy)=char(ione+ichr1)
          draw(ix+1,iy)=char(ione+ichr2)
        else
          ichr2=i-1
          draw(ix,iy)=char(ione+ichr2)
        end if
3     Continue
      Write(6,'(20x,40a1)') ( ( Draw(j,i),j=1,40), i=1,20)
      Write(6,*)' Return to continue'
      Read (5,'(i1)') ir
      Return
      End
      Subroutine Drawfrg2(nfrag,nfrag1)
      Common/molfrag/ mol_name(100), natom_mol(100), coor_mol(100,40,3),
     *                atype_mol(100,40), nmol_frag
      Real coor_mol
      Integer natom_mol, atype_mol, nmol_frag
      Character*8 mol_name
C  Local varibles
      Character*1 draw(60,20),one
      one='1'
      ione=ichar(one)
C
      Write(6,*)
     *' Molecular fragments '//mol_name(nfrag)//' and '
     *//mol_name(nfrag1)//' will be drawn'
      Do 1 i=1,60
      Do 1 j=1,20
        Draw(i,j)=' '
        if (i.eq.30) Draw(i,j)='|'
1     Continue
C
C  Process first fragment
      xmax=0.0
      xmin=0.0
      ymax=0.0
      ymin=0.0
C  Find maximum dimension in either x or y
      Do 2 i=1,natom_mol(nfrag)
        xmax=max( coor_mol(nfrag,i,1), xmax )
        xmin=min( coor_mol(nfrag,i,1), xmin )
        ymax=max( coor_mol(nfrag,i,2), ymax )
        ymin=min( coor_mol(nfrag,i,2), ymin )
2     Continue
      xrange=(xmax-xmin)/23.0
      yrange=(ymax-ymin)/18.0
C
C  Store things in bits 1-25 for x
      Do 3 i=1,natom_mol(nfrag)
        ix=int( (coor_mol(nfrag,i,1)-xmin)/xrange ) + 1
        iy=int( (coor_mol(nfrag,i,2)-ymin)/yrange ) + 1
        if ( i.ge.10 ) then
          ifact=i/10
          ichr1=ifact-1
          ichr2=i-10*ifact-1
          draw(ix,iy)=char(ione+ichr1)
          draw(ix+1,iy)=char(ione+ichr2)
        else
          ichr2=i-1
          draw(ix,iy)=char(ione+ichr2)
        end if
3     Continue
C
C  Process second fragment
      xmax=0.0
      xmin=0.0
      ymax=0.0
      ymin=0.0
C  Find maximum dimension in either x or y
      Do 4 i=1,natom_mol(nfrag1)
        xmax=max( coor_mol(nfrag1,i,1), xmax )
        xmin=min( coor_mol(nfrag1,i,1), xmin )
        ymax=max( coor_mol(nfrag1,i,2), ymax )
        ymin=min( coor_mol(nfrag1,i,2), ymin )
4     Continue
      xrange=(xmax-xmin)/24.0
      yrange=(ymax-ymin)/19.0
C
C  Store things in bits 35-60 for x
      ioff=34
      Do 5 i=1,natom_mol(nfrag1)
        ix=int( (coor_mol(nfrag1,i,1)-xmin)/xrange ) + 1
        iy=int( (coor_mol(nfrag1,i,2)-ymin)/yrange ) + 1
        if ( i.ge.10 ) then
          ifact=i/10
```

```
          ichr1=ifact-1
          ichr2=i-10*ifact-1
          draw(ix+ioff,iy)=char(ione+ichr1)
          draw(ix+ioff+1,iy)=char(ione+ichr2)
        else
          ichr2=i-1
          draw(ix+ioff,iy)=char(ione+ichr2)
        end if
5     Continue
C
C  Now draw things
      Write(6,'(13x,a,25x,a)')'Fragment A','Fragment B'
      Write(6,'(5x,60a1)') ( ( Draw(j,i),j=1,60), i=1,20)
      Write(6,*)' Return to continue'
      Read (5,'(i1)') ir
      Return
      End
      SUBROUTINE SURFAC(NATOM,ICT,TAG,TAG1,INZ,KN,ZLB,
     2                  ZUB,ZR1,YR1,XR1,RAD1,RSEC2,AREA,RSEC,
     3                  ARCI,ARCF,XR,YR,ZR,RAD,P,RH2O,ACCESS,CONTACT)
C
C     ACCESS: CALCULATE ACCESSIBLE CONTACT SURFACE AREA FOR A GROUP OF
C     ATOMS. THE ACCESSIBLE AREA FOR A GIVEN ATOM IS CALCULATED BY THE
C     FORMULA,
C
C          (ARCSUM) X (ATOM RADIUS+PROBE RADIUS) X (DELTAZ)
C
C     NUMERICAL INTEGRATION IS CARRIED OUT OVER Z. IN EACH Z-SECTION,
C     THE ARCSUM FOR A GIVEN ATOM IS THE ARCLENGTH OF THE CIRCLE
C     (INTERSECTION OF THE ATOM SPHERE WITH THE Z-SECTION) THAT IS NOT
C     INTERIOR TO ANY OTHER ATOM CIRCLES IN THE SAME Z-SECTION.
C
C input/ouput
      INTEGER NATOM, ICT, TAG(1), TAG1(1), INZ(1), KN(1)
      REAL    ZLB(1), ZUB(1), XR1(1), YR1(1), ZR1(1), RAD1(1)
      REAL    RSEC2(1), AREA(1), RSEC(1), ARCI(ICT,1), ARCF(ICT,1)
      REAL  XR(1), YR(1), ZR(1)
      REAL    RAD(1), P, RH2O, ACCESS(1), CONTACT(1)
C local
      INTEGER ANO, I, IDUM, N, K, J, J1, J2, J3, M, IEND, ITAB, ICT1
      INTEGER IANO, ICNT1, ICNT2, IFLAG
      REAL PIE, RMIN, PIEX2, ZUBMAX, RINC, ZOR, HZRES, ZRES, CUTOFF
      REAL ZNEXT, ZGRID, A, DX, DY, D, B, Q, D2
C begin
      PIE=ACOS(-1.0)
      PIEX2=2.0*PIE
      ICT1=ICT-1
      ANO=NATOM
      RMIN=10000.
C
      DO 7 I=1,ANO
C
C     CALCULATE LOWEST ZBOUND FOR EACH ATOM AND SORT ATOMS FROM LOW TO
C     HIGH ON ZLB
C
      ZLB(I)=ZR(I)-RAD(I)
    7 CONTINUE
      CALL SORTAG(ZLB,ANO,TAG)
      DO 6 I=1,ANO
      J=TAG(I)
      TAG1(J)=I
      XR1(I)=XR(J)
      YR1(I)=YR(J)
      ZR1(I)=ZR(J)
      RAD1(I)=RAD(J)
      IF(RAD1(I).LT.RMIN)RMIN=RAD1(I)
    6 CONTINUE
      ZUBMAX=0.0
C
C     THE RADIUS OF AN ATOM SPHERE = ATOM RADIUS + PROBE RADIUS
C
      RINC=RH2O
      ZOR=ZLB(1)-RINC
      DO 13 I=1,ANO
      RAD1(I)=RAD1(I)+RINC
      ZR1(I)=ZR1(I)-ZOR
      ZUB(I)=ZR1(I)+RAD1(I)
      ZLB(I)=ZR1(I)-RAD1(I)
      IF(ZUB(I).GT.ZUBMAX)ZUBMAX=ZUB(I)
      AREA(I)=0.0
   13 CONTINUE
C
C     Z RESOLUTION DETERMINED
C
      HZRES=(RMIN+RH2O)*P
      ZRES=2.0*HZRES
      CUTOFF=ZRES/100.
      IANO=1
      J1=1
      IFLAG=0
      ICNT1=0
      ICNT2=0
      ZNEXT=ZRES
      IEND=ZUBMAX/ZRES
C
C     SECTION ATOM SPHERES PERPENDICULAR TO THE Z AXIS
C
      DO 9 I=1,IEND
      ITAB=0
      ZGRID=ZNEXT
      ZNEXT=ZGRID+ZRES
      DO 10 N=IANO,ANO
C
C     THE UPPER AND LOWER Z BOUNDS OF AN ATOM ARE USED TO DETERMINE
C     WHETHER THIS Z-SECTION CUTS THE ATOM SPHERE
C
      IF(ZUB(N).LE.ZGRID) GO TO 21
      IF(ZLB(N).GE.ZGRID) GO TO 30
      KN(N)=0
      DO 34 K=1,ICT
      ARCI(K,N)=0.0
   34 CONTINUE
C
C     COUNT AND RECORD SPHERES CUT BY SECTION
C
      ITAB=ITAB+1
      INZ(ITAB)=N
C
C     FIND RADIUS OF CIRCLE LOCUS
C
      A=RAD1(N)**2-(ZGRID-ZR1(N))**2
      RSEC2(N)=A
      RSEC(N)=SQRT(A)
   21 IF(IFLAG.EQ.1)GO TO 10
C
C     FIND 1ST SPHERE CUT BY SECTION I,SKIP ATOMS PREDEEDING IT IN LIST
C     FOR NEXT SECTION
C
```

```
      IF(ZUB(N).GT.ZNEXT) GO TO 32
      J1=J1+1
      GO TO 10
   32 IFLAG=1
   10 CONTINUE
   30 IANO=J1
      IFLAG=0
C
C     ZERO, ONE, OR MORE CIRCLES ON SECTION REQUIRE DIFFERENT PROCESSING
C
      IF(ITAB.LT.1) GO TO 9
      IF(ITAB.EQ.1) GO TO 28
      J3=ITAB-1
C
C     FIND INTERSECTIONS OF CIRCLES IN SECTION CALARC CALLED TO FIND
C     INITIAL AND FINAL ANGLES OF INTERSECTION OF CIRCLES. IF ARCI AND
C     ARCF ARRAYS ARE FILLED, REDUCE IS CALLED. THE CURRENT 1ST INDEX OF
C     THESE ARRAYS FOR THE ATOM INDICATED BY 2ND INDEX IS STORED IN THE
C     ARRAY KN, IF KN FOR ANY ATOM IS 10000, THEN NO AREA REMAINS
C     ACCESSIBLE FOR THAT ATOM ON THIS SECTION OR THE ATOM IS NOT OF
C     INTEREST(KNSET=1).
C
      DO 11 K=1,J3
      N=INZ(K)
      J2=K+1
      DO 22 J=J2,ITAB
      M=INZ(J)
      A=RSEC(M)+RSEC(N)
      DX=XR1(M)-XR1(N)
      IF(ABS(DX).GE.A) GO TO 22
      DY=YR1(M)-YR1(N)
      IF(ABS(DY).GE.A) GO TO 22
      D2=DY**2+DX**2
      D=SQRT(D2)
      IF(D.GE.A)GO TO 22
      IF(KN(N).LT.ICT1)GO TO 4
      IF(KN(N).LE.ICT) THEN
        CALL REDUCE(N,0,ICT,ARCI,ARCF,TAG,TAG1,ZR1,RAD1,AREA,
     2              INZ,KN,ICNT1,ICNT2,
     3              PIE,PIEX2,ICT1,ZRES,RH2O,ZGRID,HZRES,CUTOFF)
      END IF
    4 KN(N)=KN(N)+1
      IF(KN(M).LT.ICT1)GO TO 5
      IF(KN(M).LE.ICT) THEN
        CALL REDUCE(M,0,ICT,ARCI,ARCF,TAG,TAG1,ZR1,RAD1,AREA,
     2              INZ,KN,ICNT1,ICNT2,
     3              PIE,PIEX2,ICT1,ZRES,RH2O,ZGRID,HZRES,CUTOFF)
      END IF
    5 KN(M)=KN(M)+1
C
C     DO THE CIRCLES INTERSECT, OR IS ONE COMPLETELY INSIDE THE OTHER?
C
      B=RSEC(M)-RSEC(N)
      IF(D.GT.ABS(B))GO TO 20
      IF(B.GT.0.0)GO TO 12
      KN(M)=10000
      KN(N)=KN(N)-1
      GO TO 22
   12 KN(N)=10000
      KN(M)=KN(M)-1
      GO TO 22
C
C     IF THE CIRCLES INTERSECT, FIND THE POINTS OF INTERSECTION
C
   20 Q=RSEC2(M)-RSEC2(N)
      D=2.0*D
      IF(KN(M).GT.ICT)GO TO 45
      CALL CALARC(M,1.0,ICT,D2,Q,D,RSEC,ARCI,ARCF,KN,DY,DX,PIE,PIEX2)
   45 IF(KN(N).GT.ICT)GO TO 22
      CALL CALARC(N,-1.0,ICT,D2,Q,D,RSEC,ARCI,ARCF,KN,DY,DX,PIE,PIEX2)
   22 CONTINUE
   11 CONTINUE
C
C     FIND THE ACCESSIBLE CONTACT SURFACE AREA FOR ALL THE SPHERES
C     INTERSECTING THIS SECTION
C
   28 CALL REDUCE(IDUM,ITAB,ICT,ARCI,ARCF,TAG,TAG1,ZR1,RAD1,
     2            AREA,INZ,KN,ICNT1,ICNT2,
     3            PIE,PIEX2,ICT1,ZRES,RH2O,ZGRID,HZRES,CUTOFF)
    9 CONTINUE
C
C
      DO 31 J=1,ANO
      I=TAG1(J)
C
C     SCALE AREA TO VDW SHELL IF NECESSARY
C
      CONTACT(J)=AREA(I)*((RAD1(I)-RH2O)/RAD1(I))**2
      ACCESS(J)=AREA(I)
   31 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE CALARC(M,SIGN,ICT,D2,Q,D,RSEC,ARCI,ARCF,KN,DY,DX,
     2                  PIE,PIEX2)
C
C     INITIAL AND FINAL ARC ENDPOINTS ARE FOUND FOR A REFERENCE CIRCLE
C     INTERSECTED BY ANOTHER CIRCLE CONTAINED IN THE SAME PLANE. THE
C     INITIAL ENDPOINT OF THE ENCLOSED ARC IS STORED IN ARCI, AND THE
C     FINAL ARC IN ARCF
C
C input/ouput
      INTEGER M
      REAL SIGN
      INTEGER ICT
      REAL D2, Q, D, RSEC(1), ARCI(ICT,1), ARCF(ICT,1)
      INTEGER KN(1)
      REAL DY, DX, PIE, PIEX2
C local
      INTEGER K1
      REAL ARG, ALPHA1, BETA1, TI, TF
C begin
      K1=KN(M)
C
C     LAW OF COSINES
C
      ARG=(D2+Q*SIGN)/(D*RSEC(M))
      ALPHA1=ACOS(ARG)
C
C     ALPHA1 IS THE ANGLE BETWEEN A LINE CONTAINING A POINT OF
C     INTERSECTION AND THE REFERENCE CIRCLE CENTER AND THE LINE
C     CONTAINING BOTH CIRCLE CENTERS
C
      BETA1=ATAN2(DY,DX)
C
C     BETA1 IS THE ANGLE BETWEEN THE LINE CONTAINING BOTH CIRCLE CENTERS
C     AND THE X-AXIS
```

```
C
      IF(SIGN.EQ.1.0)BETA1=BETA1+PIE
      TI=BETA1-ALPHA1
      TF=BETA1+ALPHA1
      IF(TI.LT.0.0)TI=TI+PIEX2
      IF(TF.GT.PIEX2)TF=TF-PIEX2
      IF(TF.LT.0.0)TF=TF+PIEX2
      IF(TF.GE.TI)GO TO 3
C
C     IF THE ARC CROSSES ZERO, THEN BREAK IT INTO TWO SEGMENTS. THE
C     FIRST ENDS AT 2XPI AND THE SECOND BEGINS AT ZERO
C
      ARCF(K1+1,M)=TF
      ARCF(K1,M)=PIEX2
      KN(M)=KN(M)+1
      GO TO 2
    3 ARCF(K1,M)=TF
    2 ARCI(K1,M)=TI
      RETURN
      END
```

```
      SUBROUTINE REDUCE(N,ITAB,ICT,ARCI,ARCF,TAG,TAG1,ZR1,RAD1,
     2                  AREA,INZ,KN,ICNT1,ICNT2,
     3                  PIE,PIEX2,ICT1,ZRES,RH2O,ZGRID,HZRES,CUTOFF)
C
C     1) ITAB=0, REMOVE DEGENERACIES IN ARCI AND ARCF FOR A SINGLE ATOM
C     SINCE THESE ARRAYS ARE FILLED, OR 2) ITAB>0, CALCULATE THE
C     ACCESSIBLE SURFACE AREA FOR THE N ATOMS IN THIS SECTION
C
C input/output
      INTEGER N, ITAB, ICT
      REAL    ARCI(ICT,1),ARCF(ICT,1)
      INTEGER TAG(1),TAG1(1)
      REAL    ZR1(1),RAD1(1),AREA(1)
      INTEGER INZ(1), KN(1), ICNT1, ICNT2
      REAL    PIE, PIEX2
      INTEGER ICT1
      REAL    ZRES, RH2O, ZGRID, HZRES, CUTOFF
C local
      INTEGER II, K1, JJ, K, M, I, I1, J
      REAL    ARCSUM, T, A, PAREA
C begin
      DO 23 II=1,MAX(1,ITAB)
C
C     N IS PASSED IN THE SUBROUTINE ARGUMENT LIST FOR 1), OR IN INZ FOR
C     2)
C
      IF(ITAB.NE.0)N=INZ(II)
      K1=KN(N)
      IF(K1.GT.ICT)GO TO 23
C
C     IF K1>ICT, THIS ATOM IS NOT OF INTEREST OR IT HAS NO AREA
C     REMAINING IS THIS CIRCLE INTERSECTED BY OTHERS?
C
      IF(K1.NE.0) GO TO 19
C
C     THERE IS ONLY ONE CIRCLE IN THIS SECTION
C
      ARCSUM=PIEX2
      GO TO 25
C
C     THE ARC ENDPOINTS ARE SORTED ON THE VALUE OF THE INITIAL ARC
C     ENDPOINT
C
   19 CALL SORTAG(ARCI(1,N),K1,TAG)
C
C     CALCULATE THE ACCESSIBLE AREA
C
      ARCSUM=ARCI(1,N)
      JJ=TAG(1)
      T=ARCF(JJ,N)
      IF(K1.EQ.1) GO TO 8
      DO 27 K=2,MAX(2,K1)
      IF(T.GE.ARCI(K,N))GO TO 39
      ARCSUM=ARCSUM+ARCI(K,N)-T
      JJ=TAG(K)
      T=ARCF(JJ,N)
      GO TO 27
   39 M=TAG(K)
      IF(ARCF(M,N).GT.T)T=ARCF(M,N)
   27 CONTINUE
    8 ARCSUM=ARCSUM+PIEX2-T
   25 A=ZR1(N)-ZGRID
      A=RAD1(N)-ABS(A)
```

```
C
C     THE AREA IS EQUAL TO THE ACCESSIBLE ARC LENGTH X THE SECTION
C     THICKNESS, CORRECTED IF IT IS THE FIRST OR LAST SECTION, X THE
C     RADIUS OF THE SPHERE
C
      PAREA=ARCSUM*(H2RES+AMIN1(A,H2RES))*RAD1(N)
C
C     ADD THE ACCESSIBLE AREA FOR THIS ATOM IN THIS SECTION TO THE AREA
C     FOR THIS ATOM FOR ALL THE SECTION ENCOUNTERED THUS FAR
C
      AREA(N)=AREA(N)+PAREA
   23 CONTINUE
C
C     IF THIS WAS THE FINAL ATOM FOR THIS 2 SECTION RETURN TO ACCESS
C
      IF(ITAB.GT.0)GO TO 1
C
C     THE ARCI AND ARCF ARRAYS WERE FILLED, DOES SIGNIFICANT AREA
C     REMAIN?
C
      IF(PAREA.GT.CUTOFF)GO TO 47
C
C     NO, THIS ATOM IS USED ONLY IN CALCS FOR OTHER ATOMS, UNTIL NEXT 2
C     SECTION
C
      KN(N)=10000
      ICNT1=ICNT1+1
      GO TO 1
C
C     YES, REMOVE DEGENERACIES FROM ARCI AND ARCF (COMBINE OVERLAPPING
C     ARCS)
C
   47 JJ=TAG(1)
      T=ARCF(JJ,N)
      I=1
      DO 2 K=2,MAX(2,K1)
      IF(T.GE.ARCI(K,N))GO TO 3
      I=I+1
      IF(I.EQ.ICT1) GO TO 4
      I1=I-1
      DO 7 J=K,MAX(K,K1)
      IF(TAG(J).EQ.I1)GO TO 6
    7 CONTINUE
      GO TO 9
    6 ARCF(JJ,N)=ARCF(I1,N)
      TAG(J)=JJ
    9 ARCF(I1,N)=T
      ARCI(I,N)=ARCI(K,N)
      JJ=TAG(K)
      T=ARCF(JJ,N)
      GO TO 2
    3 M=TAG(K)
      IF(ARCF(M,N).GT.T)T=ARCF(M,N)
    2 CONTINUE
      ARCF(I,N)=T
      KN(N)=I
      I=I+1
      DO 5 K=I,MAX(K1,I)
      ARCI(K,N)=0.0
    5 CONTINUE
C
C     REMOVE THE PARTIAL AREA ADDED, THIS CIRCLE MAY HAVE MORE
C     INTERSECTIONS
C
      AREA(N)=AREA(N)-PAREA
      ICNT2=ICNT2+1
      GO TO 1
C
C     NO DEGENERACIES WERE FOUND IN THE OVERLAPS, START OVER WITH LARGER
C     ICT
C
    4 WRITE(6,15) TAG1(N),ICT
   15 FORMAT('0ERROR IN SURFA1 - ATOM ',I5,
     2       ' HAS MORE THAN ',I3,' CONTACTS')
      Stop   ' REDUCE:  Problems with contacts'
    1 RETURN
      END
      SUBROUTINE SORTAG(A,N,TAG)
C
      INTEGER N,TAG(1)
      REAL A(N)
C
      REAL T, TT
      INTEGER I, L, M, J, K, IJ
      INTEGER TG, IU(16), IL(16)
C
      DO 1  I=1,N
      TAG(I)=I
    1 CONTINUE
      M=1
      I=1
      J=N
    5 IF(I.GE.J) GO TO 70
   10 K=I
      IJ=(J+I)/2
      T=A(IJ)
      IF(A(I).LE.T) GO TO 20
      A(IJ)= A(I)
      A(I)=T
      T=A(IJ)
      TG=TAG(IJ)
      TAG(IJ)=TAG(I)
      TAG(I)=TG
   20 L=J
      IF(A(J).GE.T) GO TO 40
      A(IJ)=A(J)
      A(J)=T
      T=A(IJ)
      TG=TAG(IJ)
      TAG(IJ)=TAG(J)
      TAG(J)=TG
      IF(A(I).LE.T) GO TO 40
      A(IJ)=A(I)
      A(I)=T
      T=A(IJ)
      TG=TAG(IJ)
      TAG(IJ)=TAG(I)
      TAG(I)=TG
      GO TO 40
   30 A(L)=A(K)
      A(K)=TT
      TG=TAG(L)
      TAG(L)=TAG(K)
      TAG(K)=TG
   40 L=L-1
      IF(A(L).GT.T) GO TO 40
      TT=A(L)
   50 K=K+1
```

```
      IF(A(K).LT.T) GO TO 50
      IF(K.LE.L) GO TO 30
      IF(L-I.LE.J-K) GO TO 60
      IL(M)=I
      IU(M)=L
      I=K
      M=M+1
      GO TO 80
 60   IL(M)=K
      IU(M)=J
      J=L
      M=M+1
      GO TO 80
 70   M=M-1
      IF(M.EQ.0) RETURN
      I=IL(M)
      J=IU(M)
 80   IF(J-I.GE.1) GO TO 10
      IF(I.EQ.1) GO TO 5
      I=I-1
 90   I=I+1
      IF(I.EQ.J) GO TO 70
      T=A(I+1)
      IF(A(I).LE.T) GO TO 90
      TG=TAG(I+1)
      K=I
 100  A(K+1)=A(K)
      TAG(K+1)=TAG(K)
      K=K-1
      IF(T.LT.A(K)) GO TO 100
      A(K+1)=T
      TAG(K+1)=TG
      GO TO 90
      END
      Subroutine Cnvtuc(String,Length)
C
C     This routine converts a string to upper case and removes
C     nonacceptable control characters. This routine is ASCII code
C     dependant.
C
C
      Integer Length
      Character*(*) String
      Character*1 Littla,Littlz,BigA,Blank
      Integer Itab
      Parameter (Littla='a',Littlz='z',BigA='A',Itab=9,Blank=' ')
      Integer ILittla,ILittlz,IbigA,Ichr,I
C
      ILittla=Ichar(Littla)
      ILittlz=Ichar(Littlz)
      IbigA=Ichar(BigA)-ILittla
C
      Do 1 I=1,Length
        Ichr=Ichar(String(I:I))
        If(Ichr.Ge.ILittla.And.Ichr.Le.ILittlz)
     *     String(I:I)=Char(Ichr+IbigA)
        If(Ichr .Eq. Itab)
     *     String(I:I)=Blank
1     Continue
      Return
      End
C
      Subroutine Addatml(nfrag,i,j,k,l,Bond,Theta,Phi)
C
C  This routine adds atom l to an existing chain of i,j,k.  The
C  atom is added with dihedral angle phi and valence angle
C  and bond distance given by the parameters.
C
      Common/molfrag/ mol_name(100), natom_mol(100), coor_mol(100,40,3),
     *                atype_mol(100,40), nmol_frag
      Real coor_mol
      Integer natom_mol, atype_mol, nmol_frag
      Character*8 mol_name
C
      Cst=cos(Theta)
      Snt=sin(Theta)
      Csp=cos(Phi)
      Snp=sin(Phi)
C
      Xa=coor_mol(nfrag,j,1)-coor_mol(nfrag,k,1)
      Ya=coor_mol(nfrag,j,2)-coor_mol(nfrag,k,2)
      Za=coor_mol(nfrag,j,3)-coor_mol(nfrag,k,3)
      Xb=coor_mol(nfrag,i,1)-coor_mol(nfrag,j,1)
      Yb=coor_mol(nfrag,i,2)-coor_mol(nfrag,j,2)
      Zb=coor_mol(nfrag,i,3)-coor_mol(nfrag,j,3)
C
      Ra=sqrt(Xa*Xa+Ya*Ya+Za*Za)
      Xa=Xa/Ra
      Ya=Ya/Ra
      Za=Za/Ra
C
      Xc=Yb*Za-Zb*Ya
      Yc=Zb*Xa-Xb*Za
      Zc=Xb*Ya-Yb*Xa
C
      Rc=sqrt(Xc*Xc+Yc*Yc+Zc*Zc)
C
      If (Rc.Lt.1.0e-09) Write(6,*)' Linear bond around atom',k
C
      Xc=Xc/Rc
      Yc=Yc/Rc
      Zc=Zc/Rc
C
      Xb=Ya*Zc-Za*Yc
      Yb=Za*Xc-Xa*Zc
      Zb=Xa*Yc-Ya*Xc
C
      Wa=bond*Cst
      Wb=bond*Snt*Csp
      Wc=bond*Snt*Snp
C
      coor_mol(nfrag,l,1)=coor_mol(nfrag,k,1)+Wa*Xa+Wb*Xb+Wc*Xc
      coor_mol(nfrag,l,2)=coor_mol(nfrag,k,2)+Wa*Ya+Wb*Yb+Wc*Yc
      coor_mol(nfrag,l,3)=coor_mol(nfrag,k,3)+Wa*Za+Wb*Zb+Wc*Zc
C
      Return
      End
C
C
      Subroutine Rotate(Rn1,Rn2,U,Phir,Parallel,Axis)
C
C  This routine takes two vectors, rn1 and rn2, and
C  ( Parallel=.True. ) finds the unitary
C  transformation which makes them parallel
C  ( Parallel=.False. ) Simply finds the transformation
C  which rotates by Phir w/r to an axis perpendicular
C  to both Rn1 and Rn2.
C  ( Axis= true  ) Rotate by Phir around Rn1
```

```
C
      Real Rn1(3),Rn2(3),Rot(3)
      Real U(3,3),A(3),B(3)
      Real Phir,Csp,Snp,Cst
      Integer I,J,Jpt,Kpt
      Logical Parallel,Axis
C
      If ( Parallel ) then
C  Find angle between two vectors
        dot12=0.0
        norm1=0.0
        norm2=0.0
        Do 1 i=1,3
          dot12=dot12+Rn1(i)*Rn2(i)
          norm1=norm1+Rn1(i)*Rn1(i)
          norm2=norm2+Rn2(i)*Rn2(i)
1       Continue
        Phir=Dot12/norm1/norm2
        If (abs(Phir).gt.1.0) Phir=Sign(1.0,Phir)
        Phir=Acos(Phir)
        Phir=Sign(Phir,Dot12)
C  Rotate in negative direction by Phir to overlap vectors
        Phir=-Phir
      End if
        Csp=cos(Phir)
        Snp=sin(Phir)
C
      If ( Axis ) then
C  Axis is equal to Rn1
        rotn=0.0
        Rot(1)=rn1(1)
        rotn=rot(1)*rot(1)
        Rot(2)=rn1(2)
        rotn=rotn+rot(2)*rot(2)
        Rot(3)=rn1(3)
        rotn=rotn+rot(3)*rot(3)
      Else
        rotn=0.0
        Rot(1)=rn1(2)*rn2(3)-rn1(3)*rn2(2)
        rotn=rot(1)*rot(1)
        Rot(2)=rn1(3)*rn2(1)-rn1(1)*rn2(3)
        rotn=rotn+rot(2)*rot(2)
        Rot(3)=rn1(1)*rn2(2)-rn1(2)*rn2(1)
        rotn=rotn+rot(3)*rot(3)
      End if
      Do 2 i=1,3
        Rot(i)=Rot(i)/Sqrt(Rotn)
2     Continue
C
C  Initialize the matrix
      Do 3 i=1,3
      Do 3 j=1,3
        U(j,i)=0.0
3     Continue
C
      Do 4 i=1,3
        Do 5 J=1,3
          A(j)=0.0
5       Continue
        A(i)=1.0
        Cst=Rot(i)
        Do 9 j=1,3
          A(j)=A(j)-Cst*Rot(j)
9       Continue
C
        Do 6 j=1,3
          Jpt=j+1
          Kpt=j+2
          If(Jpt.Gt.3) Jpt=Jpt-3
          If(Kpt.Gt.3) Kpt=Kpt-3
          B(j)=Rot(Jpt)*A(Kpt)-Rot(Kpt)*A(Jpt)
6       Continue
        Do 8 j=1,3
          U(i,j)=Cst*Rot(j)+Csp*A(j)-Snp*B(j)
8       Continue
4     Continue
C
      Return
      End
C
C
      Subroutine Getdih(nfrag,i,j,k,l,Ap)
      Common/molfrag/ mol_name(100), natom_mol(100), coor_mol(100,40,3),
     *                atype_mol(100,40), nmol_frag
      Real coor_mol
      Integer natom_mol, atype_mol, nmol_frag
      Character*8 mol_name
C
C
      Fx=coor_mol(nfrag,i,1)-coor_mol(nfrag,j,1)
      Fy=coor_mol(nfrag,i,2)-coor_mol(nfrag,j,2)
      Fz=coor_mol(nfrag,i,3)-coor_mol(nfrag,j,3)
      Gx=coor_mol(nfrag,j,1)-coor_mol(nfrag,k,1)
      Gy=coor_mol(nfrag,j,2)-coor_mol(nfrag,k,2)
      Gz=coor_mol(nfrag,j,3)-coor_mol(nfrag,k,3)
      Hx=coor_mol(nfrag,l,1)-coor_mol(nfrag,k,1)
      Hy=coor_mol(nfrag,l,2)-coor_mol(nfrag,k,2)
      Hz=coor_mol(nfrag,l,3)-coor_mol(nfrag,k,3)
C
C
C
      ax=fy*gz-fz*gy
      ay=fz*gx-fx*gz
      az=fx*gy-fy*gx
      bx=hy*gz-hz*gy
      by=hz*gx-hx*gz
      bz=hx*gy-hy*gx
C
      ra2=ax*ax+ay*ay+az*az
      rb2=bx*bx+by*by+bz*bz
      ra=sqrt(ra2)
      rb=sqrt(rb2)
      if(ra.le.0.1) ra=0.1
      rar=1.0/ra
      if(rb.le.0.1) rb=0.1
      rbr=1.0/rb
C
C
      axr=ax*rar
      ayr=ay*rar
      azr=az*rar
      bxr=bx*rbr
      byr=by*rbr
      bzr=bz*rbr
c
      ct=axr*bxr+ayr*byr+azr*bzr
C
C
```

```
      if (ct.gt.  1.00 ) ct= 1.00
      if (ct.lt.(-1.00)) ct=-1.00
      ap  =acos(ct)
c
      cx=ayr*bzr-azr*byr
      cy=azr*bxr-axr*bzr
      cz=axr*byr-ayr*bxr
c
      st=sqrt(cx*cx+cy*cy+cz*cz)
      s=gx*cx+gy*cy+gz*cz
      if(s.gt.0.0) then
      ap=-ap
      st=-st
      End if
      Return
      End
```

```
      if (ct.gt.  1.00 ) ct= 1.00
      if (ct.lt.(-1.00)) ct=-1.00
      ap  =acos(ct)
c
      cx=ayr*bzr-azr*byr
      cy=azr*bxr-axr*bzr
      cz=axr*byr-ayr*bxr
c
      st=sqrt(cx*cx+cy*cy+cz*cz)
      s=qx*cx+qy*cy+qz*cz
      if(s.gt.0.0) then
      ap=-ap
      st=-st
      End if
      Return
      End
```

```
type param.fmt
  11
    1  CSP3        1.530     111.000      2.000    100
    2  CSP2        1.350     120.000      2.000    100
    3  CARO        1.390     120.000      1.700    100
    4  HARO        0.600     120.000      1.200    100
    5  NSP3        1.400     120.000      1.700      0
    6  OSP3        1.350     110.000      1.700      0
    7  S           2.000     110.000      1.900     80
    8  F           1.300     111.000      1.650     90
    9  CL          2.000     111.000      1.900     90
   10  BR          2.300     111.000      2.000     90
   11  I           2.700     111.000      2.100    100
$
```

```
$
$ type fragment.fmt
  24
  BENZENE
   6
   1     2      0.00000      0.00000      0.00000
   2     2      1.38000      0.00000      0.00000
   3     2      2.07000      1.19512      0.00000
   4     2     -0.69000      1.19512      0.00000
   5     2      0.00000      2.39023      0.00000
   6     2      1.38000      2.39023      0.00000
  NAPHTHALEN
  10
   1     2      0.00000      0.00000      0.00000
   2     2      1.38000      0.00000      0.00000
   3     2      2.07000      1.19512      0.00000
   4     2      1.38000      2.39023      0.00000
   5     2     -0.69000      1.19512      0.00000
   6     2      0.00000      2.39023      0.00000
   7     2     -2.07000      1.19511      0.00000
   8     2     -2.76000      2.39023      0.00000
   9     2     -2.07000      3.58534      0.00000
  10     2     -0.69000      3.58535      0.00000
  ANTHRAC
  14
   1     2      0.00000      0.00000      0.00000
   2     2      1.38000      0.00000      0.00000
   3     2      2.07000      1.19512      0.00000
   4     2      1.38000      2.39023      0.00000
   5     2     -0.69000      1.19512      0.00000
   6     2      0.00000      2.39023      0.00000
   7     2     -2.07000      1.19511      0.00000
   8     2     -0.69000      3.58535      0.00000
   9     2     -2.76000      2.39023      0.00000
  10     2     -2.07000      3.58534      0.00000
  11     2     -4.14000      2.39023      0.00000
  12     2     -4.83000      3.58534      0.00000
  13     2     -4.14000      4.78046      0.00000
  14     2     -2.76000      4.78046      0.00000
  PHENANTH
  14
   1     2      0.00000      0.00000      0.00000
   2     2      1.38000      0.00000      0.00000
   3     2      2.07000      1.19512      0.00000
   4     2      1.38000      2.39023      0.00000
   5     2     -0.69000      1.19512      0.00000
   6     2      0.00000      2.39023      0.00000
   7     2     -2.07000      1.19511      0.00000
   8     2     -0.69000      3.58535      0.00000
   9     2     -2.76000      2.39023      0.00000
  10     2     -2.07000      3.58534      0.00000
  11     2     -2.76000      0.00000      0.00000
  12     2     -4.14000      0.00000      0.00000
  13     2     -4.83000      1.19511      0.00000
  14     2     -4.14000      2.39023      0.00000
  BENZOPHE
  14
   1     2      0.00000      0.00000      0.00000
   2     2      1.38000      0.00000      0.00000
   3     2      2.07000      1.19512      0.00000
   4     2     -0.69000      1.19512      0.00000
   5     2      0.00000      2.39023      0.00000
   6     2      1.38000      2.39023      0.00000
```

```
 7     2    -2.06979      1.21920      0.00000
 8     7    -2.67547      0.14866      0.00000
 9     2    -4.17063      2.41993      0.00000
10     2    -2.79084      2.39584      0.00000
11     2    -2.18589      3.63618      0.00000
12     2    -4.83967      3.62690      0.00000
13     2    -4.12891      4.80979      0.00000
14     2    -2.95757      4.78025      0.00000
DIPHENME
13
 1     2     0.00000      0.00000      0.00000
 2     2     1.38000      0.00000      0.00000
 3     2     2.07000      1.19512      0.00000
 4     2    -0.69000      1.19512      0.00000
 5     2     0.00000      2.39023      0.00000
 6     2     1.38000      2.39023      0.00000
 7     1    -2.06979      1.21920      0.00000
 8     2    -4.17063      2.41993      0.00000
 9     2    -2.79084      2.39584      0.00000
10     2    -2.18589      3.63618      0.00000
11     2    -4.83967      3.62690      0.00000
12     2    -4.12891      4.80979      0.00000
13     2    -2.95757      4.78025      0.00000
DIPHENAN
13
 1     2     0.00000      0.00000      0.00000
 2     2     1.38000      0.00000      0.00000
 3     2     2.07000      1.19512      0.00000
 4     2    -0.69000      1.19512      0.00000
 5     2     0.00000      2.39023      0.00000
 6     2     1.38000      2.39023      0.00000
 7     5    -2.06979      1.21920      0.00000
 8     2    -4.17063      2.41993      0.00000
 9     2    -2.79084      2.39584      0.00000
10     2    -2.18589      3.63618      0.00000
11     2    -4.83967      3.62690      0.00000
12     2    -4.12891      4.80979      0.00000
13     2    -2.95757      4.78025      0.00000
DIOXIN
14
 1     2     0.00000      0.00000      0.00000
 2     2     1.38000      0.00000      0.00000
 3     2     2.07000      1.19512      0.00000
 4     2    -0.69000      1.19512      0.00000
 5     2     0.00000      2.39023      0.00000
 6     2     1.38000      2.39023      0.00000
 7     6    -2.06979      1.21920      0.00000
 8     2    -4.17063      2.41993      0.00000
 9     2    -2.79084      2.39584      0.00000
10     2    -2.18589      3.63618      0.00000
11     2    -4.83967      3.62690      0.00000
12     2    -4.12891      4.80979      0.00000
13     2    -2.95757      4.78025      0.00000
14     6    -0.83918      3.73035      0.00000
TRIAZINE
 6
 1     2     0.00000      0.00000      0.00000
 2     5     1.37500      0.00000      0.00000
 3     2     2.06250      1.19079      0.00000
 4     5     1.37500      2.38157      0.00000
 5     2     0.00000      2.38157      0.00000
 6     5    -0.68750      1.19079      0.00000
N4BENZ
18
 1     2     0.00000      0.00000      0.00000
 2     2     1.38000      0.00000      0.00000
```

3	2	2.07000	1.19512	0.00000
4	2	1.38000	2.39023	0.00000
5	2	-0.69000	1.19512	0.00000
6	2	0.00000	2.39023	0.00000
7	2	-2.07000	1.19511	0.00000
8	2	-0.69000	3.58535	0.00000
9	2	-2.76000	2.39023	0.00000
10	2	-2.07000	3.58534	0.00000
11	2	-4.14000	2.39023	0.00000
12	2	-4.83000	3.58534	0.00000
13	2	-4.14000	4.78046	0.00000
14	2	-2.76000	4.78046	0.00000
15	2	2.05500	-1.16913	0.00000
16	2	3.42000	1.19512	0.00000
17	2	3.40500	-1.16913	0.00000
18	2	4.09500	0.02599	0.00000
S4BENZ				
16				
1	2	0.00000	0.00000	0.00000
2	2	1.38000	0.00000	0.00000
3	2	2.07000	1.19512	0.00000
4	2	1.38000	2.39023	0.00000
5	2	-0.69000	1.19512	0.00000
6	2	0.00000	2.39023	0.00000
7	2	-2.07000	1.19511	0.00000
8	2	-2.76000	2.39023	0.00000
9	2	-2.07000	3.58534	0.00000
10	2	-0.69000	3.58535	0.00000
11	2	-4.11000	2.39023	0.00000
12	2	-2.74500	0.02598	0.00000
13	2	-4.78500	1.22110	0.00000
14	2	-4.09500	0.02598	0.00000
15	2	-2.07000	-1.14316	0.00000
16	2	-0.67500	-1.16913	0.00000
T4BENZ				
17				
1	2	0.00000	0.00000	0.00000
2	2	1.38000	0.00000	0.00000
3	2	2.07000	1.19512	0.00000
4	2	1.38000	2.39023	0.00000
5	2	-0.69000	1.19512	0.00000
6	2	0.00000	2.39023	0.00000
7	2	-2.07000	1.19511	0.00000
8	2	-0.69000	3.58535	0.00000
9	2	-2.76000	2.39023	0.00000
10	2	-2.07000	3.58534	0.00000
11	2	-4.14000	2.39023	0.00000
12	2	-4.83000	3.58534	0.00000
13	2	-4.14000	4.78046	0.00000
14	2	-2.76000	4.78046	0.00000
15	2	-4.81500	1.22109	0.00000
16	2	-2.74500	0.02598	0.00000
17	2	-4.14000	0.05196	0.00000
C4BENZ				
18				
1	2	0.00000	0.00000	0.00000
2	2	1.38000	0.00000	0.00000
3	2	2.07000	1.19512	0.00000
4	2	1.38000	2.39023	0.00000
5	2	-0.69000	1.19512	0.00000
6	2	0.00000	2.39023	0.00000
7	2	-2.07000	1.19511	0.00000
8	2	-0.69000	3.58535	0.00000
9	2	-2.76000	2.39023	0.00000
10	2	-2.07000	3.58534	0.00000
11	2	-4.14000	2.39023	0.00000

```
12      2     -4.83000      3.58534      0.00000
13      2     -4.14000      4.78046      0.00000
14      2     -2.76000      4.78046      0.00000
15      2      3.42000      1.19512      0.00000
16      2      2.05500      3.55936      0.00000
17      2      4.09500      2.36426      0.00000
18      2      3.40500      3.55936      0.00000
CHRYSENE
18
 1      2      0.00000      0.00000      0.00000
 2      2      1.38000      0.00000      0.00000
 3      2      2.07000      1.19512      0.00000
 4      2      1.38000      2.39023      0.00000
 5      2     -0.69000      1.19512      0.00000
 6      2      0.00000      2.39023      0.00000
 7      2     -2.07000      1.19511      0.00000
 8      2     -0.69000      3.58535      0.00000
 9      2     -2.76000      2.39023      0.00000
10      2     -2.07000      3.58534      0.00000
11      2     -2.76000      0.00000      0.00000
12      2     -4.14000      0.00000      0.00000
13      2     -4.83000      1.19511      0.00000
14      2     -4.14000      2.39023      0.00000
15      2      3.42000      1.19512      0.00000
16      2      2.05500      3.55936      0.00000
17      2      4.09500      2.36426      0.00000
18      2      3.40500      3.55936      0.00000
PYRIDINE
 6
 1      5      0.00000      0.00000      0.00000
 2      2      1.37500      0.00000      0.00000
 3      2      2.05000      1.16913      0.00000
 4      2      1.37500      2.33827      0.00000
 5      2      0.02500      2.33827      0.00000
 6      2     -0.65000      1.16913      0.00000
IMIDAZOL
 5
 1      2      0.36674      1.06526      0.00000
 2      2      1.20085     -0.01815      0.00000
 3      5      0.33166     -1.05777      0.00000
 4      2     -0.95564      0.66545      0.00000
 5      5     -0.94361     -0.65479      0.00000
PYRROLE
 5
 1      2      0.36674      1.06526      0.00000
 2      2      1.20085     -0.01815      0.00000
 3      2      0.33166     -1.05777      0.00000
 4      2     -0.95564      0.66545      0.00000
 5      5     -0.94361     -0.65479      0.00000
INDOLE
 9
 1      2     -1.53762      1.09831      0.00000
 2      2     -0.21879      0.73748      0.00000
 3      2     -0.28479     -0.61601      0.00000
 4      2      0.97047      1.42322      0.00000
 5      2     -2.35104     -0.01833      0.00000
 6      5     -1.54495     -1.06399      0.00000
 7      2      0.81957     -1.42878      0.00000
 8      2      2.11023      0.63072      0.00000
 9      2      2.03692     -0.76262      0.00000
ACID
 3
 1      2      0.00000      0.37863      0.00000
 2      6      1.09102     -0.18932      0.00000
 3      6     -1.09102     -0.18932      0.00000
AMIDE
```

```
 3
 1    2    -0.04283     0.41902     0.00000
 2    6    -1.09284    -0.22159     0.00000
 3    5     1.13568    -0.19743     0.00000
KETONE
 3
 1    1     0.00000     0.60013     0.00000
 2    6     1.09102    -0.56795     0.00000
 3    1     1.22476    -0.90020     0.00000
ISOPROPY
 3
 1    1     0.00000     0.60013     0.00000
 2    1    -1.22476    -0.30007     0.00000
 3    1     1.22476    -0.30007     0.00000
NITRO
 3
 1    5     0.00000     0.37863     0.00000
 2    6     1.09102    -0.18932     0.00000
 3    6    -1.09102    -0.18932     0.00000
SULFATE
 4
 1    7     0.00000     0.00000     0.00000
 2    6    -0.96361     1.37618     0.00000
 3    6    -0.96361    -1.37618     0.00000
 4    6    -0.96361     0.00000     1.37618
$
```