2000

Determination of nizatidine-guest complexation constants and computer aided molecular design

Carolyn J. Mordas
Colby College

Follow this and additional works at: https://digitalcommons.colby.edu/honorstheses

Part of the Chemistry Commons

Colby College theses are protected by copyright. They may be viewed or downloaded from this site for the purposes of research and scholarship. Reproduction or distribution for commercial purposes is prohibited without written permission of the author.

Recommended Citation
https://digitalcommons.colby.edu/honorstheses/171

This Honors Thesis (Open Access) is brought to you for free and open access by the Student Research at Digital Commons @ Colby. It has been accepted for inclusion in Honors Theses by an authorized administrator of Digital Commons @ Colby. For more information, please contact mfkelly@colby.edu.
Determination of Nizatidine-Guest Complexation Constants and Computer Aided Molecular Design

Carolyn J. Mordas

Department of Chemistry
Colby College
Waterville, Maine
2000
Determination of Nizatidine-Guest Complexation Constants and Computer Aided Molecular Design

Carolyn J. Mordas

Department of Chemistry
Colby College
Waterville, Maine
2000
Determination of Nizatidine-Guest Complexation Constants and Computer Aided Molecular Design

Carolyn J. Mordas

In fulfillment of Honors Research in Chemistry

Thomas Shattuck, Mentor

Stephen Dunham, Reader

Dasan Thamattoor, Reader

May 21, 2000
I held them in every light. I turned them in every attitude.
I surveyed their characteristics. I dwelt upon their peculiarities.
I pondered their conformation.

- Edgar Allen Poe
# Table of Contents

- Acknowledgements 2
- Vita 3
- Abstract 4
- Introduction 5
- Theory 14
- Experimental 25
- Results 36
- Discussion 47
- Summary 50
- Future Work 51
- Appendix A - NMR Spectra 52
- Appendix B - Distance Geometry Program 76
- References 142
Acknowledgments

I would like to thank Professor Shattuck for his guidance throughout the two years that we worked on this project. Thank you for your enthusiasm and energy and for being an understanding advisor. I am confident that I have learned an amazing amount of chemistry from you and am delighted at how you have taught me to approach problems. Thank you for all of your time and advice and for preparing me well for graduate school.

I would also like to thank the rest of the Colby College chemistry department for an incredible four years of education that I received here. Thank you for creating a warm learning atmosphere and for the help all of you offered throughout the years.

I would like to thank the National Science Foundation, Merck, and the Howard Hughes Medical Research Institute for funding. Your generous donations are greatly appreciated and have given Colby the opportunity to show the rest of the chemistry community how far we have progressed in recent years.

I would like to thank Paul J. Schupf for his donation to purchase our computational center. We have incredible facilities for our use and because of this, Colby students will continue to be accepted to and excel at exception graduate schools.

I would like to thank all of the chemistry majors in the class of 2000 for putting in long hours in Keyes and helping each other out when we need it. I would especially like to thank Jason St. Clair for sharing twenty-one classes with me in four years and keeping me sane throughout them all.

I would like to thank all of my friends and roommates for their support in my years at Colby and for putting up with all of my chemistry talk.

I would like to thank my father for encouraging me to pursue science and for raising me to be the person I am today. I would like to thank my sister, Jessica, for her tremendous support and encouragement throughout my life and especially in the last four years.
Vita

Carolyn J. Mordas was born on July 11, 1978 in Livonia, Michigan to Donald and Vita Mordas. She lived in Michigan until 1988, at which point she moved with her family to Long Island, New York where they resided in the town of South Huntington.

In June of 1996, Carolyn received her high school diploma from Walt Whitman High School and in September of the same year, she began coursework at Colby College in Waterville, Maine. After completing a research project at the University of Vermont under the guidance of Dr. William Geiger in the summer of 1998, she began a two-year research project at Colby with Dr. Thomas Shattuck as her advisor. On May 21, 2000, Carolyn graduated from Colby College with a B.A. with honors in Chemistry – ACS and a B.A. in Mathematics-Mathematical Sciences.

In September of 2000, Carolyn enrolled at Princeton University to pursue a Ph.D. in physical chemistry.
Abstract

Nizatidine, N-[2-[[2-[[dimethylamino)methyl]-4-thiazolyl]methyl]thio]ethyl]-N'-methyl-2-nitro-1,1, ethenediamine, has been investigated for its potential to act as a host to small molecules through hydrogen bonding and pi-pi interactions. In previous work, pyridoxine (vitamin B6) was hypothesized to have a high guest-host binding constant with nizatidine. This was supported through experimental methods involving one and two dimensional NMR spectroscopy, UV-visible spectroscopy, and high performance liquid chromatography. Computer aided molecular design and three dimensional conformationally flexible database searching led to the discovery of other possible guests, including albuterol, which has exhibited even higher binding constants from HPLC.
Introduction

Computer aided molecular design is a process involving the use of computational techniques and database manipulation to develop new molecules for some purpose in mind. When this technique is combined with experiment, results from computer design can be tested to determine their success. Our experiments involve the formation of guest-host complexes, which can demonstrate activity based on binding interactions. By studying these interactions as molecules form complexes, it is possible to develop better binding pairs by maximizing favorable interactions and minimizing those which are unfavorable.

Binding between guests and hosts is an integral part of many processes. Biological processes depend greatly on intermolecular interactions. For example, binding between enzymes and substrates is the initial stage of catalysis, which can lower the activation energy of reactions. Guest-host complexes are also extremely useful in incorporating a molecule into a phase in which it is not soluble. This characteristic has been manipulated by manufacturers who are able to market compounds in specific phases.

Guests and hosts can interact primarily by three different means. The first of these is hydrogen bonding. Hydrogen bonding occurs when a hydrogen that is bonded to a highly electronegative atom (good examples are nitrogen, oxygen, fluorine) interacts with a similar atom. The hydrogen bond acceptor can either be in the same or a different molecule, meaning that the bonding is capable of being intermolecular or intramolecular. A visual description of hydrogen bonding is as follows:

![Hydrogen Bonding](image)

Figure 1
Hydrogen bonding
Hydrogen bonds can enhance the stability of a guest host complex by lowering the energy of the complex. However, it is also possible that intramolecular interactions in a host can be so strong that it is highly unfavorable for a guest to be incorporated into a complex if the intramolecular hydrogen bond must be broken.\(^2\)

Hydrophobic interactions also play a large role in guest-host complexations. Different regions of a molecule can possess both hydrophobic and hydrophilic regions. Cyclodextrins are a good example of molecules which have hydrophobic inner cavities and a hydrophilic outer rim. The structure of B-cyclodextrin is:

![Figure 2: β-Cyclodextrin](image)

The hydroxyl groups which are oriented outward enable the molecule to be soluble in polar solvents, such as water. However, the inner cavity, which is about 6.5 Å in diameter, is capable of incorporating very non-polar molecules into its cavity, thus forming a guest-host complex.\(^3\) This use of a guest-host complex can be practically applied when it is necessary for a molecule to be incorporated into some solvent or phase in which it is not soluble. An example of this might be a hydrophobic drug, which can be incorporated into a biological system through a cyclodextrin-guest complex.

Pi-pi interactions are also highly stabilizing and arise when electron density of a pi system interacts with a more positive electron density around the protons of another. This can happen in two different ways:
In both the edge-face and the face-face case, there are favorable interactions between the slightly offset electron cloud and the positively charged protons, which leads to energy stabilization. Although cyclodextrin has been reported in literature to form tight binding guests with hundreds of compounds, it is not a particularly interesting host to study for the purpose of molecular design because it is not selective in choosing guests. The main reason for cyclodextrin-guest complexation is the hydrophobic cavity, which will incorporate just about any small molecule with non-polar groups. In our research, we chose to study a host which would be more selective. We also wished to emphasize aqueous systems.

Aqueous guest-host systems are particularly interesting for several reasons. First, biological systems are primarily aqueous so the emphasis of drug design is aqueous systems. Water is a very good hydrogen bond donor and can compete for hydrogen bonding sites, making it somewhat hard to predict how molecules may interact with each other. In choosing a molecule on which to perform guest-host studies, we decided to pick something that could be obtained without too much difficulty and that was also water soluble so that we could study aqueous systems and observe hydrogen bonding between the guest and host.

Since many biological processes result because of intermolecular binding interactions, it was decided that it would be interesting to choose a host that is already

Figure 3
Pi-pi Interactions
known to bind in some way. For this reason, we chose H2 antagonists as our hosts for our experiments. The H2 receptor in biological systems is responsible for the secretion of gastric acids. Scientists who created H2-antagonists did so with the intent of binding to this site and preventing the secretion of gastric acids in persons with frequent heartburn, ulcers, or other gastric problems.\(^6\) Two commonly used H2 antagonists are:

![Nizatidine and Cimetidine](image)

Figure 4
H2 Antagonists

As can be seen in the above figure, H2 antagonists are capable of two types of guest-host interactions: hydrogen bonding and pi-pi interactions. However, these molecules each have a somewhat floppy sidechain (unlike cyclodextrin) and unfavorable changes of entropy should be taken into consideration when determining the favorability of complexation. It can be hypothesized that interaction with a small molecule through hydrogen bonding and pi-pi interactions would be sufficient enough to stabilize a guest-host complex. However, based on Gibb's Free Energy, \(\Delta G = \Delta H - T\Delta S\), the decrease in entropy of the complex might overcompensate for the favorable change in energy, meaning that complexation would not be spontaneous. Cram's principles in guest-host chemistry state that there should be little conformational change in the host when a complex forms because the change in entropy will be unfavorable.\(^7\) Our choice of hosts is interesting because a complex formed between one of these H2 antagonists and a guest will result in a much more rigid conformation for the host, thus requiring an unfavorable change in entropy for the host.
The procedure for computer aided molecular design involves some repetition in order to maximize binding between a guest and a host. First, the structure of either a guest or a host must be determined. In our research, we chose nizatidine as our host. Next, a model should be built to determine possible binding sites. This is done through the creation of a pharmacophore, which is a three dimensional schematic describing the arrangement in space of functional groups responsible for binding. An example of a pharmacophore used in past work is:

![Pharmacophore Example](image)

32.8 - 52.8°
6.52 - 7.52 Å

Once a pharmacophore has been established, a database search can be done to give “hits” for molecules that fit the pharmacophore. This database involves the use of three dimensional conformationally flexible parameters, such as bond angle and distances between atoms, because many of guest and host molecules used in our studies can change conformation easily. The “hit” molecules can then be docked with the host to determine the favorability of the complex. The complex is modeled and its energy can be determined. At this point, the guest can be synthesized or ordered and the experimental binding constant can be determined. We employ methods including fluorescence, UV-Visible and nuclear magnetic resonance spectroscopy along with titration calorimetry and high performance liquid chromatography to determine binding constants. The end result
is the design of a new molecule that is able to bind according to our model and can be used for its originally designated purpose.

Previous work on this project determined the favored conformation of nizatidine to be U-shaped as is depicted in the following computer-generated model:

![Figure 6](image)

*Figure 6*

Lowest Energy Conformation of Nizatidine

In addition, it was determined that the following Z,E conformation for the guanidine fragments in nizatidine to be the lowest in energy:

![Figure 7](image)

*Figure 7*

Z,E Conformation of Guanidine Fragments
Using *ab initio* methods, calculations were also performed on the cyanoguanidine, nitroguanidine and methylimidazole type regions of nizatidine to determine the potential for pi-pi interactions with guests such as benzene or pyridine. The calculations for the cyanoguanidine fragments were not successful as they did not converge. For the nitroguanidine fragments, it was found that pyridine was better able to stabilize the system than benzene, and the lowest energy orientation had the nitrogen in pyridine oriented away from the nitro group in nizatidine. For the methylimidazole type fragment, it was determined that the lowest energy conformation had benzene interacting with the aromatic ring in an edge-face manner.  

Based on the potential for the determined hydrogen bond and pi-pi guest-host interactions, a pharmacophore was used to search a database resulting in 527 hits. In choosing the best potential guests, solubility and size were the major considerations. The following molecules were selected for studies: caffeine, pyridoxine, sorbitol, phenylalanine, luminol, tryptophan, nicotinamide, and pyrogallol.

Additional work involved the use of fluorescence, UV-Visible, and nuclear magnetic resonance spectroscopy to determine binding constants for interactions between these molecules and nizatidine. Fluorescence experiments yielded the following complexation constants:

<table>
<thead>
<tr>
<th>Guest</th>
<th>$K_v$ (L/mol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>caffeine</td>
<td>$4.49 \times 10^4$</td>
</tr>
<tr>
<td>luminol</td>
<td>$2.07 \times 10^4$</td>
</tr>
<tr>
<td>pyridoxine</td>
<td>$3.21 \times 10^{13}$</td>
</tr>
<tr>
<td>tryptophan</td>
<td>$9.76 \times 10^5$</td>
</tr>
<tr>
<td>tryptophan @ 25° C initial</td>
<td>$4.17 \times 10^4$</td>
</tr>
<tr>
<td>tryptophan @ 25° C after 2 min.</td>
<td>$3.44 \times 10^3$</td>
</tr>
</tbody>
</table>

Table 1

**Binding Constants from Fluorescence Spectroscopy**

Although these constants are suspiciously high for guest-host binding, pyridoxine seems to be the best guest with nizatidine. The next experiments performed involved the use of UV-Visible spectroscopy to determine binding constants resulting in the following:
Guest | $K_a$ (L/mol)  
--- | ---  
caffeine | $6 \times 10^{11}$  
nicotinamide | $9 \times 10^{5}$  
phenylalanine | $2 \times 10^{12}$  
pyrogallol | $4 \times 10^{6}$  

Table 2
Binding Constants from UV-Visible Spectroscopy

At this point, nuclear magnetic resonance experiments were performed on systems with tryptophan and caffeine as guests. No peak shifts were observed, indicating that complexation was not occurring. In continuing this project, we planned to refine previous experiments to determine if nizatidine is a selective host for small molecules through pi-pi interactions and hydrogen bonding. Del Sesto’s extensive computational work was a good foundation for predicting how nizatidine might form guest-host complexes. Corbeil’s fluorescence and UV-visible experiments were very promising since her results indicated complexation. However, her NMR experiments were contradictory. We planned on repeating some of the UV-Visible and NMR experiments to determine reproducibility. We also planned on pursuing additional methods which might be better techniques of determining the binding constants between the hits obtained from Del Sesto’s database search and nizatidine.

In addition to the refinement of Corbeil’s UV-Visible absorption and proton NMR work, we performed experiments using titration calorimetry and two-dimensional (specifically NOESY) NMR spectroscopy. The use of reverse phase high performance liquid chromatography was also employed extensively in determining guest-host binding constants related to this project. Before developing experiments for the nizatidine-guest systems, we validated our HPLC methods using cyclodextrin as a host for small molecules which have been extensively reported in literature determined from numerous experimental techniques. One reference in particular, which used HPLC, chose the use of an ion exchange column instead of a reverse phase column. Since the nature of the separations was so different, we felt that we could not trust our binding constants determined by reverse phase HPLC until we had validated our experiments using cyclodextrin-guest systems and had compared our constants with literature values.
determined from different methods. Upon conformation of our technique, Nizatidine-guest systems were also studied using reverse phase HPLC. We also conducted computer simulated reverse phase experiments to better understand the accuracy of the method and the effect of the partition coefficient for the complex.

Two-dimensional NOESY NMR results are presented to determine information about the binding conformation of pyridoxine and nizatidine. The new pharmacophore was created from both molecular mechanics and dynamics calculations, which were performed using nOe based constraints, and distance geometry calculations. Once we had found a tight binding guest, we conducted another database search with an even more specific pharmacophore in order to test other guests experimentally. From all of these methods, we hoped to achieve the ultimate goal of a better understanding of molecular recognition and molecular design.
Theory

UV/Visible

The absorption of ultraviolet or visible radiation results from the excitation of electrons. The absorption of ultraviolet or visible radiation occurs when a molecule, \( A \), is excited by a given wavelength of light

\[
A + h\nu \rightarrow A^*
\]

where \( A^* \) exists for only nanoseconds and then relaxes. The most common type of relaxation results in the excitation energy turning to heat

\[
A^* \rightarrow M + \text{heat}
\]

Since the lifetime of \( A^* \) is so short, its concentration at a given moment is negligible. The intensity of absorbance can be measured as host is added to the sample cell containing guest and an equilibrium constant can be determined with the following derivation.\[1\]

From the Beer-Lambert law:

\[
A = \varepsilon b [C]
\]

where \( \varepsilon \) is the molar extinction coefficient of the compound of interest. In a guest-host system, the total absorbance, \( A \), is defined as:

\[
A = \varepsilon_G b [G] + \varepsilon_{GH} b [G-H] + \varepsilon_H b [H]
\]

\([G]\) = concentration of free guest
\([G-H]\) = concentration of guest-host complex
\([H]\) = concentration of free host

For simplicity, we will group constants and define three additional terms:

\[
\begin{align*}
\Omega_G &= \varepsilon_G b \\
\Omega_{GH} &= \varepsilon_{GH} b \\
\Omega_H &= \varepsilon_H b
\end{align*}
\]
So the absorbance is now equal to:

\[ A = \Omega_g[G] + \Omega_{GH}[G-H] + \Omega_h[H] \]  \hspace{1cm} (6)

Accounting for all concentrations:

\[ [G]_0 = [G] + [G-H] \]  \hspace{1cm} (7)
\[ [H]_0 = [H] + [G-H] \]  \hspace{1cm} (8)

Combining equations 6, 7 and 8:

\[ A = \Omega_g([G]_0 - [G-H]) + \Omega_{GH}[G-H] + \Omega_h([H]_0 - [G-H]) \]  \hspace{1cm} (9)

The equilibrium constant for the formation of guest-host complex can be written:

\[ K = \frac{[G-H]}{([G]_0 - [G-H])([H]_0 - [G-H])} \]  \hspace{1cm} (10)

Rearranging equation 9 to form a quadratic equation equal to zero:

\[ [G-H]^2 - ([G] + [H]_0 + K^{-1})[G-H] + [G]_0[H]_0 = 0 \]  \hspace{1cm} (11)

We can solve for the roots of equation 11, disregarding the solution which has the concentration of the complex greater than the concentration of guest.

\[ [G-H] = 0.5\left(\left([G]_0 + [H]_0 + K^{-1}\right) - \sqrt{\left([G]_0 + [H]_0 + K^{-1}\right)^2 - 4\left([G]_0[H]_0\right)^{\frac{1}{2}}}\right) \]  \hspace{1cm} (12)

Substituting equation 12 into equation 9, we can rewrite the absorbance:

\[ A = \Omega_g[G]_0 + \Omega_{GH}[H]_0 + 0.5(\Omega_{GH} - \Omega_g - \Omega_h) \times \left(\left([G]_0 + [H]_0 + K^{-1}\right) - \sqrt{\left([G]_0 + [H]_0 + K^{-1}\right)^2 - 4\left([G]_0[H]_0\right)^{\frac{1}{2}}}\right) \]  \hspace{1cm} (13)

This equation can be used to calculate a theoretical UV-Visible absorption at given [G], [H], K, \( \Omega_g \), \( \Omega_h \), and \( \Omega_{GH} \). Inputting guesses for K, \( \Omega_g \), \( \Omega_h \), and \( \Omega_{GH} \) a curve is fit as best as possible to the actual data on a plot of absorbance signal versus concentration added host. The best fit is found by minimizing the square of the residual (actual y values – fit y values). A plot might look as follows:
Figure 8
UV-Visible Curve Fitting

The approximation for $K$ can be found when the square of the differences between the fit values and the experiment data is at a minimum.

**Thermometric Titration Calorimetry**

The use of a solution calorimeter enables the enthalpy of a reaction to be measured quantitatively. Calorimetry can be applied to a guest-host complex if a temperature of the reaction vessel containing host is monitored as guest is titrated into the solution. In most titrations, sensitivity in determining the endpoint is logarithmically related to concentration of the reactants. However, in titration calorimetry, the sensitivity of the method is linearly proportional to concentration which makes it a good endpoint determination method for dilute solutions.

In addition to being able to measure an enthalpy of reaction with titration calorimetry, we are also able to determine an equilibrium constant if the magnitudes of $K$ and $\Delta H$ are within a certain limit. The shape of a titration curve is a function of the $K$ value, where higher equilibrium constants result in sharper curves. An example is given below:
Figure 9  
Titration Curve Fit Example

It is ideal that log K be less than 4 for quantitative measurements because there are only slight differences in curvature for reactions with Ks in this range. It is also desirable that log K be greater than 1 since reactions with low K generally have low reaction enthalpies and there is difficulty in measurement. A temperature change of at least 0.01° is recommended for any quantitative measurements.

Since there is no direct way to measure an equilibrium constant using titration calorimetry, numerical curve fitting is used to extract the binding constant. A relationship for the heat of reaction at a point p is defined as:

\[ Q_{e.p} = \Delta H(\Delta n_p) \tag{14} \]

where \( \Delta n_p \) is the number of moles of complex formed from the start of the reaction to p. \( \Delta n_p \) cannot be determined without knowing K. However, it is possible to input guesses for K and observe the result in \( \Delta H \), which should be relatively constant. Using multiple points for p along the entire titration curve, K can be approximated when a value is found which yields the same \( \Delta H \) for various points in the run.\(^{12}\)

**1-D Nuclear Magnetic Resonance**

Peak positions in an \(^1\)H NMR spectrum are sensitive to the environment of the hydrogen. When complexation occurs, the environment of the immediate hydrogens is changed. In the case of the interactions we see in a guest-host complex, the resulting
peak in the spectrum will be an average of the peaks of the complexed molecule and free molecule. As in the case of the thermometric titration calorimetry, the binding constant for the complexation cannot be measured directly, but is also found numerically. This can be done by using the equation:

\[
K = \frac{\Delta \delta_{\text{obs}}}{(\Delta \delta_{\text{GH}} - \Delta \delta_{\text{obs}})([G] - ([H] \Delta \delta_{\text{obs}}) / \Delta \delta_{\text{GH}})}
\]  

(15)

\( \Delta \delta_{\text{obs}} \) = observed chemical shift in the spectrum
\( \Delta \delta_{\text{GH}} \) = chemical shift of the guest-host complex
\( [G] \) = concentration of free guest
\( [H] \) = concentration of free host

Both \( K \) and \( \Delta \delta_{\text{GH}} \) are unknown. A plot can be made of trial \( K \) versus \( \Delta \delta_{\text{GH}} \). If this is done for several concentrations, the point of intersection for the different concentration lines will occur at the actual binding constant. An example might look something like this:

![Figure 10](image)

**Figure 10**

*K Determination from NMR*

**2-D Nuclear Magnetic Resonance**

One particular two dimensional NMR technique is based on the nuclear Overhauser effect, which results in the generation of a NOESY spectrum. Cross peaks,
which are peaks not on the diagonal of the spectrum, result from protons which interact through space, instead of through bonds as is in the case with other types of 2-D NMR experiments. Usually, this distance is less than 4 Å.\textsuperscript{13} When a two-dimensional spectrum is generated, the proton spectrum will be plotted on both the x and y axes. Each peak in the proton spectrum can be assigned to a particular hydrogen environment in the molecule being studied. By tracing the cross peaks in the 2-D NOESY spectrum to the proton spectrum on each axis, it is possible to determine which hydrogen environments are interacting through space. Some examples of crosspeaks are circled in the following spectrum:

![Sample NOESY spectrum](image)

Figure 11
Sample NOESY spectrum

NOESYs are very useful in helping us determine how nizatidine may conform in solution because we can observe what hydrogens are interacting. We can also use this technique to determine what particular hydrogens are close to each other in a nizatidine-guest complex.
Determining a binding constant from high performance liquid chromatography is based on the principle that as the concentration of a host is increased in the mobile phase, the retention time of the guest on the column will decrease. This is due to the fact that the guest is likely to form a complex with the host in the eluent and come off the column sooner.

The reaction equation defining the complexation is:

$$K_{G-H} = \frac{[G-H]_m}{([G]_m - [H]_m)}$$  \hspace{1cm} (16)

Where $[G-H]_m$, $[G]_m$ and $[H]_m$ are the concentrations of the guest-host complex, the free guest in the mobile phase, and the free host in the mobile phase, respectively.

We can define a distribution ratio between guest in the mobile phase and guest in the stationary phase.

$$D_0 = \frac{[G]_m}{[G]_m}$$  \hspace{1cm} (17)

Similarly, the ratio between complex in each phase is:

$$D_{G-H} = \frac{[G-H]_m}{[G-H]_m}$$  \hspace{1cm} (18)

A distribution ratio for the entire system can be defined as:

$$D_{obs} = \frac{(G_0 + [G-H]_t)/(G_m + [G-H]_m)}{(G_0 + [G-H]_t)/(G_m + [G-H]_m)}$$  \hspace{1cm} (19)

By substituting equations 15, 16, and 17 into 18 we can redefine the total distribution ratio:

$$D_{obs} = (D_0 + D_{G-H} \cdot K_{G-H} \cdot [H]_m)/(1 + K_{G-H} \cdot [H]_m)$$  \hspace{1cm} (20)

Assuming that the addition of host in the mobile phase does not bring about significant changes in the void volume of the column, the relationships between concentration distribution ratio and retention times are:

$$D_0 = \frac{(T_0' - T_0)}{T_0} \cdot \frac{V_m}{V_s}$$  \hspace{1cm} (21)

$$D_c = \frac{(T_c - T_0)}{T_0} \cdot \frac{V_m}{V_s}$$  \hspace{1cm} (22)

$$D_{obs} = \frac{(T_{obs} - T_0)}{T_0} \cdot \frac{V_m}{V_s}$$  \hspace{1cm} (23)
$T_0 = \text{retention time of nonretained band}$

$T_0' = \text{retention time of the guest}$

$T_c = \text{retention time of the complex}$

$T_{obs} = \text{retention time of the guest at a particular concentration of host}$

$V_s = \text{volume of the stationary phase}$

$V_m = \text{void volume within the column}$

Substituting equations 20, 21, and 22 into equation 19 yields:

$$H_m/(T_0' - T_{obs}) = H_m/(T_0' - T_c) + 1/(K_c (T_0' - T_c))$$

(24)

A plot of the left hand term versus the concentration of host gives the binding constant value as the slope divided by the intercept.

**HPLC Simulation**

The HPLC Simulator uses a purely equilibrium model for chromatography to simulate the separation of compounds. The simulation assumes complete thermodynamic equilibrium to be established for each plate in the column during the separation. Equations 16-20 are solved numerically in each plate. At each time step in the simulation the species in the mobile phase are passed to the next plate and the equilibrium distributions is recalculated.

**Molecular Modeling**

The idea behind molecular modeling of guest-host complexes is relatively simple. First, we can model the host alone to see how it conforms when it is completely uncomplexed. This can be done both solvated and unsolvated, keeping in mind that solvated molecules require much more time for calculations since there are many more atoms. Once this conformation is determined, it is possible to build a model of the binding site. This is done by studying pi-pi and hydrogen bond interactions between a host and a potential guest. Using these interactions, one can build a pharmacophore, as described in the introduction, which gives ranges of motion of functional groups of a guest. This pharmacophore is then used to search a three dimensional conformationally
flexible database of compounds resulting in hits which are potential guests. These guests
can then be modeled with the host and a model for the complex can be determined.

**Database Searching**

Since the interaction between a guest and a host occurs in three dimensional space
and our host is rather flexible, it is important to use conformationally flexible parameters
to search a database to find potential guests. We can do this by searching a two
dimensional database and then applying these conformationally flexible parameters,
mainly distance and dihedral constraints, to a pharmacophore to limit the number of hits
even further. A pharmacophore can contain exclusion spheres, which are helpful for
defining a space into which a guest could fit (see Figure 5). It can also contain molecular
fragments which maximize favorable pi-pi and hydrogen bond interactions.8

**Molecular Dynamics**

Molecular dynamics can be used to determine low energy conformations of
systems similar to energy minimization. However, it is often possible that molecular
dynamics finds additional low energy conformations because simple energy minimization
can determine a local minimum energy value which falls within some set tolerance, but
there may exist a different global minimum. Molecular dynamics takes advantage of
molecular motion to try to find additional low energy conformations through a series of
three steps: heating, equilibration, and simulation. The result of the simulation is the
generation of low energy conformations, some of which may be lower in energy than
those generated by simple energy minimization.14

**Hydrogen Bond Strength Calculations**

We can determine the strength of a hydrogen bond between a guest and a host
computationally. This is accomplished by building and minimizing the host alone. In
our studies, we perform semi-empirical PM3 level calculations to determine the energy of
the geometry optimized host. We also decided to use water as both an hydrogen bond acceptor and donor to determine the strength of these bonds. The strength is determined from:

\[ \text{H-bond strength} = [E_{o,n} - (E_o + E_n)] \]  

(25)

where the guest in these calculations is water and the host is nizatidine.

### Distance Geometry

The term “distance geometry” refers to programs which are designed to build molecules based on specific, pre-defined atom distances. These programs are useful when a few distance constraints are already known, such as those obtained from NOESY spectra. Each program has four main components: input preparations, bounds generation and smoothing, embedding, and optimization. Input preparation results in a rough layout of what atoms are connected to each other. Bounds generation uses information about bonds and already known distances to create constraints. Embedding is the operation that converts this information into coordinates. A center of the molecule is usually defined as the reference point from which all other coordinates are related. Distances of each atom to this center are entered in as diagonal elements of a distance matrix. The law of cosines is applied to this matrix to obtain the metric matrix, which is central to the embedding process. The square roots of the eigenvalues of the metric matrix are the principle moments of the molecule while the eigenvectors give the distribution of the atoms along the axes. The origin of these axes is the reference point (the center) which was already defined in the distance matrix. The coordinates of the molecule can be calculated from the eigenvalues and eigenvectors by:

\[ c_m = \lambda_m^{1/2} W_m \]  

(26)

Where \( c, \lambda, \) and \( W \) are the coordinates, eigenvalues, and eigenvectors for each dimension. Generally, the metric matrix is found to have more than three eigenvalues, meaning that the corresponding molecule will fit into more than three dimensions. In most cases, the
largest eigenvalues and corresponding eigenvectors are used and further optimization ensures that molecule meets all boundary conditions. This is done by measuring the error between the distances and the desired boundary conditions and then altering the coordinates to minimize this error.$^{15}$
Experimental

Host Extraction

Nizatidine was extracted from two types of Axid AR pills. Axid tablets containing 150 mg nizatidine were ground up using a mortar and pestle. Axid caplets were emptied by pulling apart the plastic coating. The nizatidine from both types of pill was extracted using chloroform (approximately 1 ml per pill). The remaining solid was filtered off using a Bücher funnel and the resulting solution was filtered through a 0.22-µm Millipore filter to remove any remaining solids. The chloroform was allowed to evaporate. Solid nizatidine was recrystallized from ethanol after concentration of the solution under vacuum. Water was added dropwise to the solution to promote crystal formation.

Guest Information

Pyridoxine monohydrogenchloride, CAS 58-56-0, Lot 61C-2050, Sigma Chemical
Caffeine, CAS 58-08-2, Nutritional Biochemical
L-phenylalanine, CAS 63-91-2, Acros Chemical
Salbutamol (Albuterol), CAS 18559-94-9, Lot 15F03831, Sigma Chemical

Buffer solutions

Sodium Phosphate/Potassium Phosphate buffer (pH=7.4)
1.179 grams KH$_2$PO$_4$ (0.0087 M) and 4.30 grams Na$_2$HPO$_4$ (0.0302 M) were diluted to 1 liter with deionized water.

Potassium Hydrogen Phthalate buffer (pH=4.0)
10.12 grams of KHP (Potassium Hydrogen Phthalate) were diluted to 1 liter with deionized water.

Sodium Borate buffer (pH=9.2)
3.8 grams Na$_3$B$_4$O$_7$ were diluted to 1 liter with deionized water.
Ammonium acetate buffer (pH=7)

Equimolar amounts of concentrated ammonium hydroxide and acetic acid were added to deionized water and the pH was adjusted accordingly using NaOH or HCl.

Phosphate buffer (pH=7, pH=3)

0.1 M H₃PO₄ was prepared in deionized water. The pH was adjusted using a combination of NaOH pellets and 3 M NaOH. The ionic strength was adjusted to 0.5 M using NaCl.

**METHODS**

*Synthesis of N-oxide*

The following synthesis was performed¹⁶:

![N-Oxide Synthesis](image)

Using a graduated cylinder, 7.5 ml of glacial acetic acid was added to 0.5 grams pyridoxine HCl. Approximately 2 ml of 30% hydrogen peroxide was added to the solution and heated in water bath at 70-80°C. After three hours, another 1 ml of hydrogen peroxide was added. The solution was maintained 9 hours at 70-80°C. After cooling, it was concentrated to 3 ml with rotary evaporation and diluted with 3 ml of H₂O. The concentrated solution was made strongly alkaline with anhydrous sodium
carbonate and then shaken with 2 ml of dichloromethane to extract the N-oxide into the organic layer. Sodium salt crystals were collected with paper filter. The remaining solution was dried with sodium sulfate and the solvent was removed by rotary evaporation. Due to the small amount of synthesized N-oxide, the solid was not recrystallized before use. The identity of the solid was confirmed with NMR.

UV-Visible Spectroscopy

Instrument

Hewlett Packard 8452A Diode Array Spectrophotometer

Solutions

100 ml stock solutions of 1 mM host and 1 mM guest were prepared in pH 7.4 phosphate buffer. From these solutions, additional solutions were prepared to make one solution of 0.01 mM host and 0.05 mM guest/0.01 mM host.

Spectra

A background spectrum was obtained from pH 7.4 sodium/potassium phosphate buffer in a quartz cuvette. 2.00 mL of the 0.01 mM host solution was pipetted into the cuvette and a spectrum was obtained in the region from 200 to 800 nm, although the area of interest was expanded (from 200 to 400 nm). 0.100 mL of host/guest solution was added and the cuvette solution was mixed thoroughly using a disposable pipette. Absorbance readings were taken at the peak of interest and also at 360 or 380 nm to subtract off the baseline. Measurements were taken over a short period of time (< 1 hour) so temperature control was not employed.
Titration Calorimetry

Instrument

Tronac Isoperibol Titration Calorimeter – Model 450

Solutions

0.01 M nizatidine and 0.1 M pyridoxine were prepared in pH 7.4 phosphate buffer. 1.5 ml of the nizatidine solution were added to the dewar and the buret was filled with the pyridoxine solution. The dewar and buret were lowered into the 25.000 C temperature bath and were allowed to equilibrate before each run.

Data

A chart rate of 5 cm/min was used with a full scale of 2 mV. At the beginning of each run, the buret was reversed for 20 seconds to isolate the titrant in the delivery tube with air. After complete titration of the nizatidine solution with pyridoxine, the heater was turned on until the original temperature was obtained, at which point the heater was turned off. The buret delivery rate was calibrated with deionized water.

Nuclear Magnetic Resonance

Instrument

JEOL GSX400 Superconducting Magnet; JEOL FT NMR Spectrometer;
GSX Data System

Sample Preparation

Initial samples were prepared by weighing out approximately enough host to produce a 50 mM solution and varying concentrations of guest. Solid material was diluted with 0.80 ml (micropipette) sodium/potassium phosphate buffer and 0.20 ml (micropipette) D$_2$O to obtain a lock signal. Sodium 2,2-Dimethyl-2-
Silapentane-5-Sulfonate (DSS) was added to the sample tubes to be used as a reference.

Acquiring of Spectra

All proton spectra were obtained on the 400 MHz NMR in automated mode. Homogated decoupling was used for solvent suppression. An initial spectrum was taken using a minimal amount of scans (8) and points (8024) from which the water peak was identified and irradiated to produce an additional spectrum (using 16 scans and 16384 points). From this spectrum, the frequency range was decreased to just contain the peaks (about 8 ppm or 1600 Hz) and increase resolution. The most upfield peak corresponding to DSS was set to 0.0 ppm as a reference and all host and guest peaks were measured in both ppm and Hz. Except for variable temperature experiments, the temperature was fixed at 27°C.

2-D NOESY spectra were obtained in manual mode using the samples prepared from the 1-D experiments. A mixing time of 400 ms was used. The temperature was fixed at 30.0°C for the entire nOe experiments. The spin speed for all experiments was 15 Hz. The samples were not degassed.

High Performance Liquid Chromatography

Instrument

Gilson 231 XL Sampling Injector; LDC Membrane Degasser; LDC 3200 Analytical UV Detector; LDC 4100 Analytical Quaternary Solvent Delivery System

Columns

Alltech Econosphere C18 column, 150 mm, Lot 1039, Part No.288144
Alltech Platinum C18 rocket column, 53 mm, Lot 26/348/3 Part No. 50523
Nizatidine/Guest Solutions

The pH 7.0 phosphate buffer was prepared as described above and filtered through a 0.22- micron filter and added to an eluent bottle. Host solutions (approximately 1 mM) were prepared in buffer and similarly filtered. Guest samples were prepared by adding approximately 10 mg of solid to dissolve in 10 ml of buffer and then filtered.

Instrument Startup

Before running samples, the flow rate was slowly increased from 0.0 mL/min to either 0.5 or 1.5 mL/min for either the traditional C18 (150 mm) or the rocket column (53 mm) respectively. The solvent delivery system was purged of air by attaching a large syringe to the system and drawing solvent/gas out. The injection needle was primed periodically to eliminate air bubbles. Eluent was run through the column at these flow rates for at least twenty minutes prior to sample ejection to eliminate drift and to allow equilibration.

Sample Runs

A sample injection volume of 50 μL was used for all experiments. The initial run was eluted with 100% buffer. Additional runs included 20%, 40%, 60%, 80%, and 100% host as the eluent with the remaining percent buffer. These parameters were changed using the controls on the LDC solvent delivery system. After each eluent concentration change, the system was allowed to equilibrate for about ten minutes.

The detection wavelength for the nizatidine experiments was set to the maximum UV absorbance wavelength for the guest in buffer, which was determined using the HP Diode Array UV-Visible Spectrometer.

Cyclodextrin Experiments

All cyclodextrin experiments were intended to mirror literature experiments, and as a result, all conditions were duplicated to the best of our abilities.10,17,5
Guests: Sulfapyridine, sulfathiazole, benzoic acid, naphthalene sulfonate

Column: Alltech Econosphere C18 column

Flow rate: 0.5 mL/min

Buffers: Sulfapyridine and sulfathiazole: pH 3.0 phosphate buffer
Benzoic acid and naphthalene sulfonate: pH 7.0 ammonium acetate

Sample preparation:
- Sulfapyridine, sulfathiazole: 10 mg guest added to 10 mL ethanol
- Benzoic acid, naphthalene sulfonate: 10 mg guest added to 10 mL buffer

Ionic strength: 0.2 M

Chart Recorder Speed: 1.0 mL/min

After some variation in results, column temperature was stabilized through the addition of a foam layer surrounding the column.

Since there was some concern regarding cyclodextrin degradation, host solution concentrations were calibrated using polarimetry by comparing optical rotation at the start and finish of each experiment. A specific rotation of $162.5^\circ \pm 0.5^\circ$ was obtained from literature to use for calculation of concentration.

Between experiments, the solvent delivery system was set to change eluent concentration from 100% buffer to 100% acetonitrile over a period of five minutes. This was done to remove any remaining less polar residue from the column.
Computational Work

Molecular Modeling

All molecular modeling was done using Quanta/CHARMM version 98.1111. Nizatidine was built using edited parameters taken from the Quanta Parameter Handbook. An edited atom type (NC2) was entered for the guanidine type group nitrogens, which corresponds to neutral guanidinium group – Arginine sidechain nitrogens. The preferred conformation for nizatidine was found through dihedral adjustments and energy minimizations. Based on previous work, the guanidine fragments were arranged in the lowest energy (Z,E) conformation. All minimizations were done first through the steepest descents method with 1000 steps and a tolerance of .001 and then with the conjugate gradient method with an unlimited amount of steps and an energy gradient tolerance of 0.0001. Pyridoxine was built and complexed with nizatidine to form the maximum amount of hydrogen bonds and favorable interactions. Information taken from the nOe spectrum was used to create distance constraints (d < 4 Å) between nizatidine and pyridoxine protons. The complex was minimized accordingly.

Molecular Dynamics

Molecular dynamics simulations were also performed with Quanta to determine if there were any additional low energy conformations for the complex. The following parameters were used for the simulation:

<table>
<thead>
<tr>
<th>SHAKE OPTIONS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Shake On</td>
<td>BonH</td>
</tr>
<tr>
<td>Shake Tolerance</td>
<td>1E-09</td>
</tr>
<tr>
<td>Max. # iterations</td>
<td>500</td>
</tr>
<tr>
<td>parameter specified geometry</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DYNAMICS OPTIONS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HEATING SETUP</td>
<td></td>
</tr>
<tr>
<td>dynamics steps</td>
<td>3000</td>
</tr>
<tr>
<td>restart read file</td>
<td></td>
</tr>
<tr>
<td>restart write file</td>
<td>heat</td>
</tr>
<tr>
<td>coordinate trajectory file</td>
<td>heat</td>
</tr>
</tbody>
</table>
EQUILIBRATION SETUP

dynamics steps 3000
restart read file heat
restart write file equil
coordinate trajectory file equil
energy values file equil
output frequency 10
equilibration frequency 200
time step 0.001
temperature 300.0

SIMULATION SETUP

dynamics steps 3000
restart read file equil
restart write file simul
coordinate trajectory file simul
energy values file simul
output frequency 10
time step 0.001
temperature 300.0

restart equilibration from the restart file

ANIMATION SETUP

Use Charm header
Dataset range from 6040 to 9000
step size 40
clock speed 1
number steps to average over 0
regenerate hydrogen bonds
display geometry monitors
display trials
do not display dipole

Table 3
Dynamics Parameters

The dynamics simulation was done with a solvation sphere of 15 Å of TIP3 water molecules.
**Hydrogen Bond Energy**

Energy calculations were performed in order to find out the strength of a hydrogen bond between different atoms in nizatidine and water. Calculations were done using the Spartan program (version 5.1.1) at the semi-empirical (PM3) level. As outlined in the theory section, the strength of the hydrogen bonds was measured by subtracting the sum of the energies of nizatidine and water from the total energy for the complex. Hydrogen bond strength calculations were also performed on the electronegative atoms in the ring portion of nizatidine using density functional theory and were determined to yield similar results to those obtained using PM3.

**Distance Geometry**

Please see appendix for distance geometry program.

**HPLC Simulations**

The program used to perform HPLC simulations was written by Tom Shattuck. The column diameter was set to 4.5 mm and the length to 100 mm with a packing fraction of 80% and 100 plates. A D₅₀ of 0.10 was used (see equation 17). Concentration of mobile phase additive (host) was varied from 0% to 100%, similar to actual HPLC experiments. The flow rate was set to 1 mL/min and the injection volume to 10 µL. Although there are parameters for several injections, only one sample injection was simulated. A binding constant of 1000 was entered since it seemed to be a good approximation of the constants we had been getting experimentally. The value for the partition coefficient for the complex was changed from experiment to experiment to determine the effect on the equilibrium constant.

**Database Searching**

Database searching was performed using Isis Host and Isis Base, version 98.1. The database which was searched was the Compendium of Medicinal Chemistry (CMC 3D), which contains 7497 commercially used pharmaceutical compounds as of 5/15/00.
Search parameters were set to the default, with the following exceptions: Bump checking enabled, 5 rotatable bonds, RMS = 0.7, change in Van der Waal's energy = 5.0.
Results

**UV-Visible**

We determined the UV-visible spectrum of approximately 0.01 mM nizatidine in pH 7.4 phosphate buffer to be the following:

![UV-Visible Absorbance of Nizatidine](image)

Figure 13
UV-Visible Absorbance of Nizatidine

From this spectrum, we could obtain approximations for the molar extinction coefficient for this compound.

\[
\varepsilon_{284} = 28500 \text{ cm}^{-1} \text{ M}^{-1}
\]

\[
\varepsilon_{314} = 16300 \text{ cm}^{-1} \text{ M}^{-1}
\]

The first experiment, which was performed by adding 0.05 mM nizatidine to 0.01 mM caffeine solution, yielded significant overlap in the spectrum, so the concentrations were switched to see if results improved. This experiment yielded a binding constant of 8000 M\(^{-1}\). The procedure was repeated with pyridoxine as the guest, which led to a determination of a binding constant of 30,000 M\(^{-1}\). This evidence was supportive that binding was occurring between nizatidine and the two chosen guests, but the peaks in the UV-Visible spectrum seemed to have some overlap, making the results dubious. Also, in determining a binding constant, we used numerical methods which seemed to find a wide range of good fits for several of the guessed parameters, mainly K and \(\Omega_{GH}\) since values for \(\Omega_G\) and \(\Omega_H\) were approximately known. Because of this, the binding constant was
determined where the error between the fit and the experimental data was the least. This was done by plotting the sum of the squared differences between the experimental and fit values versus the binding constant as follows:

![Graph showing Ka vs. Sum R^2](image)

Figure 14
Determination of Nizatidine-Pyridoxine K

Although these results were optimistic, it became clear that we would need to find another method to determine binding constants more accurately.

**Purity Determination of Nizatidine**

Since the UV-Visible experiments were yielding suspicious data, it was decided to test the purity of the nizatidine extraction using HPLC. A solution of 0.1 mM nizatidine was prepared in 60:40 methanol/water. The HPLC trace of the compound revealed only one peak, reasonably narrow in width, indicating that our compound was pure.

**Determination of the pKa of Nizatidine**

An aqueous solution of 0.02 M nizatidine has a pH of approximately 8.2. Since the solution is slightly basic, the pKa of nizatidine could be determined through pH
titration with hydrochloric acid. The equivalence point was reasonably sharp and the pH was found by measuring the pH to be 6.9 at the midpoint of the titration.

**Thermometric Titration Calorimetry**

Several experiments were performed by titrating 0.010 M nizatidine with 0.1 M pyridoxine. Corrections for the heat of dilution were made. There was an observable equivalence point for the titration although the break was rather small. A curve fit to the data was attempted but was unsuccessful and no binding constant for the complexation could be determined.

**Nuclear Magnetic Resonance**

The first NMR experiment involved varying the concentration of caffeine while keeping the concentration of nizatidine constant and observing the peak shifts. This was done for a range of concentrations from 10 to 80 mM for caffeine with a nizatidine concentration of 50 mM and no shift in peaks was observed.

A similar experiment was performed with pyridoxine as the guest and this time varying the range of concentrations from 10 to 50 mM. Since the sample was prepared in 20% D$_2$O, it was difficult for the instrument to find an internal reference. Because of the complications with peak shifting, a standard NMR reference (Sodium 2,2-Dimethyl-2-Silapentane-5-Sulfonate) was added to each NMR tube before conducting a repeat experiment. There was considerable shifting of three peaks in the next referenced experiment: a pyridoxine peak at 8.03 ppm, a nizatidine peak at 3.97 ppm, and a nizatidine peak at 2.37 ppm. Spectra were also taken at a single concentration but varying temperature to see if temperature was a factor, but there were no noticeable peak shifts over a small temperature range (25-40° C).

At this point, a NOESY experiment was performed using a 1:1 solution of pyridoxine and nizatidine (50 mM) in 20% deuterated water to see if cross peaks could be found, indicating that there are intermolecular interactions between the two compounds. Our spectrum revealed two strong cross peaks between the two compounds.
An additional NOESY experiment was conducted in DMSO revealing similar cross peaks. The peaks from the aqueous spectrum are depicted below. Peak shifts from proton spectrum are in ppm.

Figure 15  
NOe Cross Peaks

Since the pyridoxine that was being used in the experiments was acidic, there was consideration that the shifts that were being observed in the proton spectrum were due to pH shifts and not complexation. Spectra of nizatidine were taken in pH 4, 7, and 10 buffers and peak shifts similar to those previously observed for the pyridoxine-nizatidine complex resulted in the acid solution. Experiments were repeated for the complex with adjusting the pH after pyridoxine addition. Peak shifts were still observed in the spectrum, but on a much smaller order of magnitude than the previous experiment (approximately 10-12 Hz, or about $10^{-6}$ ppm, instead of 0.3-0.4 ppm). Using the numerical methods described in the theory section, a binding constant could not be determined because the linear plots of $K_{\text{guess}}$ versus $\delta_{\text{OH}}$ did not intersect at any point for any guesses entered. It was decided that the chemical shifts could not be determined with sufficient accuracy to determine a binding constant with this method, even after the resolution was increased by narrowing the observed frequency as much as possible.

Additional experiments were performed with sorbitol, pyrogallol, and albuterol and no peak shifts were observed.
Molecular Mechanics

After docking the minimized energy nizatidine and pyridoxine conformations, a low energy model was developed for the complexation. The following distance constraints were used for modeling. Each dashed line represents a distance constraint of less than 4 Å.

Figure 16
Minimization Distance Constraints

These constraints are similar to the cross peaks that resulted from the NOESY spectrum and are designed to promote favorable hydrogen bonding interactions. The resulting model has three stabilizing hydrogen bonds but no pi-pi interactions. The complex conformation is shown in figure 17.
Molecular dynamics simulations confirmed that this conformation for the complex was a low energy conformation. Using this model for the binding of pyridoxine with nizatidine, the following pharmacophore was created to show the binding interactions:
This pharmacophore was used to search the CMC 3D database, using conformationally flexible constraint matching and resulted in the following 48 hits:

- 2,5-dihydroxybenzoic acid
- 2-hydroxy-3-methoxybenzyl alcohol
- 2-hydroxy-5-nitrobenzoic acid
- 2-hydroxybenzyl alcohol
- 3-methoxysalicylic acid
- 4-chlorosalicylic acid
- 5-chlorosalicylic acid
- 5-methoxysalicylic acid
- abunidazole
- albuterol
- amino salicylic acid
- balsalazide
- benzoylpas
- clorindanic acid
- diflunisal
- fendasal
- fepentolic acid
- forfenimex
- gentisic acid
- hydroxytoluic acid
- imidizole salicylate
- ipsalazide
- lasalocid
- ledazerol
- leucocianidol
- meclocycline sulfosalicylate
- mesalamine
- olsalazine
- oxycinchophen
- pararosaniline embonate
- pasiniazid
- pirbuterol
- piridoxilate
- pirsudanol
- pyridofylline
- pyridoxilate
- pyritinol
- salazodine
- salazosulfadimidine
- salazosulfamide
- salazosulfathiazole
- salicyl alcohol
- salicylate meglumine
- salicylic acid
- salmefamol
- salmeterol
- sulfasalazine
- susalimod

Table 4
CMC 3D Hits

It was decided that the best choices would be reasonably small in size and have the ability to be somewhat flexible in conformation. Albuterol and salicyl alcohol were ordered for experimentation.

Figure 19
Albuterol and Salicyl Alcohol
Dimerization Studies

Since we had been unable to determine binding constants for the pyridoxine-
nizatidine complex even though we had evidence for complexation (2-D NMR), we
decided to perform concentration studies to determine if nizatidine was dimerizing in
solution. Starting with approximately 0.05 M nizatidine in water, NMR spectra were
taken as the sample was diluted five times to give a concentration of about 0.01 M. The
spectra did not change with varying concentration, indicating that dimerization was not
occurring at this range of concentrations.

N-Oxide: Synthesis and NMR Experiment

A NMR spectrum (see appendix) was taken of the N-oxide which was synthesized from
pyridoxine and agrees with the proposed structure. The N-Oxide was added to a solution
of nizatidine and no peak shifts were observed.

High Performance Liquid Chromatography

The initial HPLC experiments worked nicely resulting in a binding constant of
2500 for nizatidine and pyridoxine. In order to determine the accuracy of our methods,
we repeated literature experiments using cyclodextrin and benzoic acid systems and
determined binding constants within the range of the literature values. The summary of
the HPLC experiments is as follows:

<table>
<thead>
<tr>
<th>Host</th>
<th>Guest</th>
<th>Column</th>
<th>pH</th>
<th>flow rate</th>
<th>K</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nizatidine</td>
<td>Pyridoxine</td>
<td>53 mm</td>
<td>7</td>
<td>1.5</td>
<td>2600</td>
<td>400</td>
</tr>
<tr>
<td>Nizatidine</td>
<td>Albuterol</td>
<td>53 mm</td>
<td>7</td>
<td>1.5</td>
<td>6900</td>
<td>300</td>
</tr>
<tr>
<td>Nizatidine</td>
<td>Phenylalanine</td>
<td>150 mm</td>
<td>7</td>
<td>0.5</td>
<td>130</td>
<td>70</td>
</tr>
<tr>
<td>Nizatidine</td>
<td>Pyrogallol</td>
<td>150 mm</td>
<td>7</td>
<td>0.5</td>
<td>140</td>
<td>10</td>
</tr>
<tr>
<td>Nizatidine</td>
<td>Phenylalanine</td>
<td>53 mm</td>
<td>7</td>
<td>1</td>
<td>90</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 5
Binding Constants for Nizatidine-Guest Complexes
The flow rate was varied depending on the retention time of the guest and the column used in the experiment, where guests with extremely long retention times (> 20 min) on a typical 150 mm C18 column were run on a 53 mm rocket column instead. Cyclodextrin experiments repeated on different columns showed good agreement in the binding constant, implying that the column used would not affect the variation in binding constant much. The literature values for the binding constants of β-cyclodextrin-benzoic acid are given in Table 6.

<table>
<thead>
<tr>
<th>Method</th>
<th>log K</th>
<th>log error</th>
</tr>
</thead>
<tbody>
<tr>
<td>calorimetry</td>
<td>2.1</td>
<td>0.4</td>
</tr>
<tr>
<td>calorimetry</td>
<td>3.26</td>
<td>not reported</td>
</tr>
<tr>
<td>potentiometry</td>
<td>2.737</td>
<td>0.002</td>
</tr>
<tr>
<td>calorimetry</td>
<td>2.57</td>
<td>not reported</td>
</tr>
<tr>
<td>calorimetry</td>
<td>2.56</td>
<td>not reported</td>
</tr>
<tr>
<td>liquid chromatography</td>
<td>2.4</td>
<td>not reported</td>
</tr>
</tbody>
</table>

Table 6
Literature Binding Constants for β-cyclodextrin/Benzoic Acid

Our experimental binding constants for the same systems are as given in Table 7.

<table>
<thead>
<tr>
<th>Host</th>
<th>Guest</th>
<th>Column</th>
<th>pH</th>
<th>log K</th>
<th>log error</th>
</tr>
</thead>
<tbody>
<tr>
<td>β-Cyclodextrin</td>
<td>Benzoic Acid</td>
<td>150 mm</td>
<td>7.0</td>
<td>3.35</td>
<td>2.30</td>
</tr>
<tr>
<td>β-Cyclodextrin</td>
<td>Benzoic Acid</td>
<td>150 mm</td>
<td>7.0</td>
<td>2.10</td>
<td>1.48</td>
</tr>
<tr>
<td>β-Cyclodextrin</td>
<td>Benzoic Acid</td>
<td>53 mm</td>
<td>3.0</td>
<td>3.65</td>
<td>2.60</td>
</tr>
<tr>
<td>β-Cyclodextrin</td>
<td>Benzoic Acid</td>
<td>53 mm</td>
<td>3.0</td>
<td>3.34</td>
<td>2.36</td>
</tr>
<tr>
<td>β-Cyclodextrin</td>
<td>Benzoic Acid</td>
<td>130 mm</td>
<td>3.0</td>
<td>2.95</td>
<td>1.85</td>
</tr>
<tr>
<td>β-Cyclodextrin</td>
<td>Benzoic Acid</td>
<td>130 mm</td>
<td>3.0</td>
<td>2.78</td>
<td>1.78</td>
</tr>
<tr>
<td>β-Cyclodextrin</td>
<td>Benzoic Acid</td>
<td>130 mm</td>
<td>3.0</td>
<td>3.04</td>
<td>2.04</td>
</tr>
</tbody>
</table>

Table 7
Experimental Binding Constants for β-cyclodextrin/Benzoic Acid
Hydrogen Bond Strength Calculations

Using Spartan at the semi-empirical PM3 level, the strength of hydrogen bonds to water were calculated and the following energies were obtained:

![Chemical structure](image)

\[ \text{DG-H} = 0.01 \text{ D.O.} \]

<table>
<thead>
<tr>
<th>[G] (M)</th>
<th>( D_{G-H} = 0.01 )</th>
<th>( D_{G-H}=0.10 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.43</td>
<td>0.43</td>
</tr>
<tr>
<td>0.1</td>
<td>0.42</td>
<td>0.43</td>
</tr>
<tr>
<td>0.2</td>
<td>0.41</td>
<td>0.43</td>
</tr>
<tr>
<td>0.5</td>
<td>0.40</td>
<td>0.43</td>
</tr>
<tr>
<td>1</td>
<td>0.38</td>
<td>0.43</td>
</tr>
</tbody>
</table>

Table 8
Effect of \( D_{G-H} \) on Retention Time

Using the same methods that we used for experimental HPLC, a binding constant of 1300 ± 200 was obtained for the first experiment where the distribution coefficient for the complex was set to 0.01. The expected equilibrium constant which was input into the
simulation was 1000. The uncertainty results primarily from round-off error in the calculated retention times. A $D_{G-H}$ of 0.01 corresponds to a retention time for "pure" guest-host complex of 0.335 minutes and for $D_{G-H}$ equal to 0.10, the retention time of the complex is 0.43 minutes, which is the same as the retention time of the guest. The corresponding ideal intercept then changes from $1/1000 \times (0.43 - 0.335)$ for $D_{G-H} = 0.01$ to zero for $D_{G-H} = 0.10$ since the guest and the guest-host complex have identical retention times.

*Distance Geometry*

One conformation obtained for the nizatidine-guest complex using distance geometry is:

![Figure 21](image)

**Nizatidine-Pyridoxine Model from Distance Geometry**

The distance geometry model is similar to the model obtained from molecular modeling and molecular dynamics. Further minimization would result in a conformation even more like those models.
Discussion

UV-Visible Spectroscopy

The results from the UV-Visible experiments seem to suggest binding, but it is important to realize that this type of spectroscopy is more qualitative than quantitative for the nizatidine-guest system. From the data, we were able to predict that pyridoxine would be a good binding guest since the constant was so high, but other guests with reasonable constants from UV-Visible spectroscopy did not show similar performance in other methods, such as NMR and HPLC. We were able to obtain a good curve fit for equation 13 for a wide range of parameters, implying that these parameters were highly correlated and that we would not be able to determine a binding constant with any degree of certainty.

Titration Calorimetry

For nizatidine, it appears that titration calorimetry is not much help in determining information about the complex. We were able to see a titration endpoint, but were not able to conclude a binding constant from the experiment. It is possible that better data may have resulted from higher concentration solutions, but we were working with several constraints. First, it is a somewhat time consuming process to extract copious amounts of pure nizatidine from the Axid AR tablets. Second, the solubility limit of nizatidine in water is not incredibly high (approximately 80 mM) and it would not have been possible to get much more concentrated solutions on which to perform experiments. Although this method has shown good results in the literature, it was not effective in determining a binding constant for our system.
Nuclear Magnetic Resonance

The proton nuclear magnetic resonance experiments seem to be reasonable indicators of complexation, as was demonstrated in the pH controlled nizatidine-pyridoxine experiment. However, this method is also more qualitative than quantitative for our system and the peak shifts cannot be resolved accurately enough to determine a binding constant.

Two dimensional nuclear magnetic resonance was proven to be a key component in the entire design process. Based on our NOESY data, we were able to determine what hydrogens in nizatidine and pyridoxine were interacting through space to produce nOe peaks in the spectrum. These peaks were an integral part in modeling the complexation because distance constraints were created for the model complex. Based on this model, we were able to develop a pharmacophore for the complex. This pharmacophore was searched in a 3D database and resulted in approximately fifty hits for potential guests. We chose a reasonable guest from the list and decided to try out high performance liquid chromatography techniques to determine a binding constant for both that complex and the pyridoxine complex.

High Performance Liquid Chromatography

Based on our results, it seems that HPLC is the most reliable method for determining a binding constant in our systems. Constants we determined for the cyclodextrin experiments agreed very well with values found in literature. Also from the HPLC data, we determined that albuterol has an even higher binding constant than pyridoxine, which shows that our database search was successful in finding a good binding guest. This implies that our model of the pyridoxine-nizatidine complex may, in fact, be a correct representation of the intermolecular interactions between the guest and the host.
HPLC Simulation

The HPLC Simulator does a good job of simulating HPLC experiment using the equilibrium model defined in the theory section. Using the simulator, we were able to observe the effect of the distribution ratio of the complex on retention time, which was used in determining binding constants. We can conclude that if the distribution ratio for the complex is high (0.10), the retention time for the complex is the same as that of pure guest and a binding constant cannot be determined. This implies that HPLC determination of binding constants may not be applied to all guest-host systems.

Computational Work

We have developed three excellent methods for modeling guest-host complexes computationally. The first of these, molecular modeling, was a relatively quick and easy way to create a model for the nizatidine-pyridoxine complex and based on this model, we were able to successfully find an additional tight binding guest by creating a pharmacophore. We were also able to use molecular dynamics as a check in determining other potential low energy conformations. For the nizatidine-pyridoxine system, the low energy conformations found by this method were similar to those found through molecular modeling, giving us reassurance that the model is a good one. Distance geometry reaffirmed this conclusion further since we developed a similar model using the geometry program.
Summary

Based on our success, we can summarize that we are able to study and predict molecular recognition. We have tested numerous experimental methods in determining binding constants and have decided that high performance liquid chromatography is our best technique because we have been able to obtain reasonable binding constants for nizatidine-guest complexes. We have also tested HPLC using cyclodextrin systems found in literature and found that the method is accurate in replicating literature constants.

Our computational methods for predicting binding interactions have also been successful and we can conclude that our techniques for predicting tight-binding pairs are excellent. We were able to draw similar conclusions based on the variety of our computational work, implying that each technique is an excellent predictor of binding.

We can also conclude that in choosing nizatidine for a host, the change in entropy was not unfavorable enough to prevent complexes from forming. This opens the door to a large variety of types of molecules to be used as hosts in studying molecular recognition.
Future Work

As with most research projects, additional work can always be done, even if conclusions have already been drawn. Extensive experimental work was performed on the nizatidine-pyridoxine complex, but the high cost of albuterol has deterred our research from conducting many more experiments with that particular guest. A next step would be to perform two dimensional NMR experiments on albuterol-nizatidine and look for cross peaks. Using the same procedure as was used for pyridoxine, these cross peaks can be used as distance constraints to model the complex computationally. The distance geometry program has not been used too extensively and it would be a good idea to test the program with other complexes.

The database search, which resulted from the pharmacophore developed from the pyridoxine-nizatidine model, yielded 48 hits, of which we only tested albuterol and hydroxybenzyl alcohol. Although we can rule out many of these hits because of their large size, it would be interesting to test other guests from the list. Our pharmacophore was relatively simple and it could be refined even further.

The process of changing guest molecules one piece at a time (as was demonstrated in the N-oxide synthesis) is one that should be explored further. Binding constants determined between nizatidine and these synthesized molecules will increase our understanding of molecular recognition much further because we will be able to pinpoint regions of the molecule most responsible for binding.
Appendix A
NMR Spectra

Sodium 2,2-Dimethyl-2-Silapentane-5-Sulfonate (DSS) in pH 7.4 buffer 53
50 mM nizatidine in pH 7.4 buffer 54
50 mM pyridoxine in pH 7.4 buffer 55
20 mM pyridoxine / 50 mM nizatidine in pH 7.4 buffer 56
30 mM pyridoxine / 50 mM nizatidine in pH 7.4 buffer 57
40 mM pyridoxine / 50 mM nizatidine in pH 7.4 buffer 58
50 mM pyridoxine / 50 mM nizatidine in pH 7.4 buffer 59
80 mM pyridoxine / 50 mM nizatidine in pH 7.4 buffer 60
50 mM nizatidine in pH 4.0 buffer 61
50 mM pyridoxine / 50 mM nizatidine in pH 4.0 buffer 62
50 mM nizatidine in pH 10.0 buffer 63
Pyridoxine N-oxide in pH 7.4 buffer 64
Nizatidine and N-oxide in pH 7.4 buffer 65
NOESY of nizatidine-pyridoxine (same solution as page 59) 66
50 mM nizatidine / 50 mM pyridoxine in DMSO 67
NOESY of nizatidine-pyridoxine (same solution as page 67) 68
33 mM pyridoxine / 37 mM nizatidine in pH 7.4 buffer (adjusted to 7.4 with NaOH) 69
66 mM pyridoxine / 37 mM nizatidine in pH 7.4 buffer (adjusted to 7.4 with NaOH) 70
50 mM albuterol in pH 7.4 buffer 71
7 mM albuterol / 50 mM nizatidine in pH 7.4 buffer 72
15 mM albuterol / 50 mM nizatidine in pH 7.4 buffer 73
24 mM albuterol / 50 mM nizatidine in pH 7.4 buffer 74
32 mM albuterol / 50 mM nizatidine in pH 7.4 buffer 75
Sodium 2,2-Dimethyl-2-Silapentane-S-Sulfonate (DSS) in pH 7.4 buffer
50 mM mizatidine in pH 7.4 buffer
50 mM pyridoxine in pH 7.4 buffer
20 mM pyridoxine / 50 mM nizatidine in pH 7.4 buffer
30 mM pyridoxine / 50 mM mizalidine in pH 7.4 buffer
40 mM pyridoxine / 50 mM nizatidine in pH 7.4 buffer
50 mM pyridoxine / 50 mM nizatidine in pH 7.4 buffer
80 mM pyridoxine / 50 mM nizatidine in pH 7.4 buffer
50 mM nizatidine in pH 4.0 buffer
50 mM pyridoxine / 50 mM nizatidine in pH 4.0 buffer
Pyridoxine N-oxide in pH 7.4 buffer
Nizatidine and N-oxide in pH 7.4 buffer
NOESY of nizatidine-pyridoxine
50 mM nizatidine / 50 mM pyridoxine in DMSO
CJM_NIZATADINE_PYRIDOXINE_PHADJ_CONC_4/27/99

JEOL
GSX-400
Colby College

27-APR-99 11:32:45
EXHQQ SGHMG
DBNOC 1H
DBFRQ 399.65 kHz
DBSET 124.00 kHz
DBFIN 10550.0 Hz
POINT 32768
FREQU 7002.8 Hz
FILTR 5000 Hz
SCANS 16
ACQTM 2.340 sec
PD 3.000 sec
PW1 6.8 usec
ADBIT 12
TEMP 27.0 c
SPEED 15 Hz
SLVNT D20
EXREF 0.00 ppm
BF 0.20 Hz

66 mM pyridoxine / 37 mM nizatidine in pH 7.4 buffer (adjusted to 7.4 with NaOH)
15 mM albuterol / 50 mM nizatidine in pH 7.4 buffer
Appendix B
Distance Geometry Program
DOUBLE PRECISION FUNCTION SAASB (SETB)
integer*8 function SAASB (SETB)
COMPUTE THE SET OF ATOMS ADJACENT TO A SET OF BONDS
DOUBLE PRECISION SAAB, OR, SETB, SETA
integer*8 SAAB, OR, SETB, SETA
SETA = 0
NB = 0
10 NB = NXM(SETB,NB)
   IF (NB.LE.0) GO TO 20
   SETA = OR(SETA,SAAB(NB))
   GO TO 10
20 SAASB = SETA
END

DOUBLE PRECISION FUNCTION SAASA (SETA)
integer*8 function SAASA (SETA)
COMPUTE THE SET OF ATOMS ADJACENT TO A SET OF ATOMS
DOUBLE PRECISION SAAS, OR, SETA, ASET
integer*8 saas, OR, SETA, ASET
ASET = 0
NA = 0
10 NA = NXM(ASET,NA)
   IF (NA.LE.0) GO TO 20
   ASET = OR(ASET,SAASA(NA))
   GO TO 10
20 SAASA = ASET
END

DOUBLE PRECISION FUNCTION SAAB (BOND)
integer*8 function SAAB (BOND)
COMPUTE THE SET OF ATOMS ADJACENT TO BOND
COMMON /BONDS/ NBDS, BNDAT(2,72), BTYPE(72)
INTEGER BOND, BNDAT, BTYPE
DOUBLE PRECISION SETA
integer*8 seta
SETA = 0
CALL ON(SETA,BNDAT(1,BOND))
CALL ON(SETA,BNDAT(2,BOND))
SAAB = SETA
RETURN
END

DOUBLE PRECISION FUNCTION SAAA (ATOM)
integer*8 function SAAA (ATOM)
COMMON /ATOMS/ NAT,ICONN(4,72),NCONN(72),NFILL(3,72)
INTEGER ATOM
DOUBLE PRECISION SETA
integer*8 seta
SETA = 0
LIM = NCONN(ATOM)
DO 10 I = 1,LIM
   CALL ON(SETA,ICONN(I,ATOM))
10 CONTINUE
SAAA = SETA
RETURN
END

DOUBLE PRECISION FUNCTION SBAA (ATOM)
integer*8 function SBAA (ATOM)
COMPUTE THE SET OF BONDS ADJACENT TO ATOM

COMMON /BONDS/ NBNDS, BNDAT(2,72), BTYPE(72)
INTEGER ATOM, AT1, AT2, BNDAT, BTYPE

DOUBLE PRECISION SETB
  integer*8 setb
  SETB = 0
  DO 10 I = 1, NBNDS
    AT1 = BNDAT(1, I)
    AT2 = BNDAT(2, I)
    IF (AT1.EQ.ATOM) CALL ON(SETB, I)
    IF (AT2.EQ.ATOM) CALL ON(SETB, I)
  CONTINUE

10 SBM = SETB
  RETURN
END

DOUBLE PRECISION FUNCTION SBAB (BOND)
  integer*8 function sbab (bond)
  C COMPUTE THE SET OF BONDS ADJACENT TO BOND
  COMMON /BONDS/ NBNDS, BNDAT(2,72), BTYPE(72)
  INTEGER BOND, AT1, AT2, BNDAT, BTYPE
  C DOUBLE PRECISION SETB, OR, SB AA
  integer*8 setb, or, sbaa
  SETB = 0
  AT1 = BNDAT(1, BOND)
  AT2 = BNDAT(2, BOND)
  SETB = OR(SBAA(AT1), SBAA(AT2))
  SBAB = SETB
  RETURN
END

INTEGER FUNCTION ADJAT (ATOM, BOND)
  C RETURNS THE OTHER END OF BOND
  COMMON /BONDS/ NBNDS, BNDAT(2,72), BTYPE(72)
  INTEGER ATOM, BOND, BNDAT, BTYPE, AT1, AT2
  AT1 = BNDAT(1, BOND)
  AT2 = BNDAT(2, BOND)
  ADJAT = AT1
  IF (AT1.EQ.ATOM) ADJAT = AT2
  RETURN
END
SUBROUTINE CORNER (DUMP, NRS, ATRNG)

SPECIAL TREATMENT FOR SMALL CYCLES: COMPUTE 1,3 DISTANCES USING NON-STANDARD BOND ANGLES.

DIMENSION ATRNG(10)
LOGICAL DUMP
INTEGER ATRNG

COMMON /ATOMS/ NAT, ICONN(4,72), NCONN(72), NAMES(72), NCNT(72), ITYPE(72),
           CHARACTER*4 NAMES, NAMEJ
COMMON /BNDS/ NPNTS, BND(72,72)

DIMENSION LM(2), IK(2)
EQUIVALENCE (IK(1), II), (IK(2), KK), (LM(1), LL), (LM(2), MM)

3000 DO 3500 J = 1, NRS
    JJ = ATRNG(J)
    LIM = NCONN(JJ)
    IF (LIM.LE.2 .AND. NRS.EQ.3) GO TO 3500
    DO NOT COMPUTE 1,3 DISTANCES ON NON-SUBSTITUTED CY-PROP CORNERS
    ONLY, BUT COMPUTE THESE DISTANCES (I.E. DIAGONALS) FOR CY-BUT.
    I = J - 1
    IF (I.EQ.0) I = NRS
    II = ATRNG(I)
    K = J + 1
    IF (K.GT.NRS) K = 1
    KK = ATRNG(K)
    DIST1 = BND(II, JJ)
    DIST2 = BND(JJ, KK)
    IF (NRS.EQ.4) GO TO 3400
    DIST3 = BND(II, KK)
    COSINT = (DIST1**2+DIST2**2-DIST3**2)/(2.0*DIST1*DIST2)
    COSLNT = COSD(120.0)
    GO TO 3450

3400 COSINT = 0.0
    COSEX = COSD(113.0)
    DIST3 = SQRT(DIST1**2+DIST2**2)
    BND(II, KK) = DIST3
    BND(KK, II) = DIST3

3450 COSMID = -SQRT((COSINT+1.0)*COSEX+COSINT+1.0)/2.0
    IVAL = 4
    NAMEJ = NAMES(JJ)
    IF (NAMEJ.EQ. 'N'. OR. NAMEJ.EQ. 'P') IVAL = 3
    IF (ITYPE(JJ).EQ.IVAL) GO TO 3800
    COSMID = -SQRT((1.0+COSINT)/2.0)

3800 IF (DUMP) WRITE(20,13000) II, DIST1, JJ, DIST2, KK, DIST3,
     1   COSINT, COSMID, COSEX
    IF (LIM.LE.2) GO TO 3500

LOOK FOR NEIGHBORS OF JJ WHICH ARE NOT IN THE CYCLE

NLM = 0
IPREV = 0
DO 3100 I = 1, LIM
    L = ICONN(I, JJ)
    IF (L /= JJ) GO TO 3100
3100 CONTINUE
IF (L.EQ.IPREV) GO TO 3100
IF (L.EQ.II.OR.L.EQ.KK) GO TO 3100
NLM = NLM + 1
LM(NLM) = L
IPREV = L
3100 CONTINUE
  IF (NLM.EQ.0) GO TO 3500
  DO 3250 N1 = 1,2
  DO 3200 N2 = 1,NLM
    DIST1 = BND(IK(N1),JJ)
    DIST2 = BND(JJ,LM(N2))
    DIST3 = SQRT(DIST1**2+DIST2**2-2.0*DIST1*DIST2*COSMID)
    BND(IK(N1),LM(N2)) = DIST3
    BND(LM(N2),IK(N1)) = DIST3
    IF (DUMP) WRITE(20,13000) IK(N1),DIST1,JJ,DIST2,LM(N2),DIST3,
    1        COSINT,COSMID,COSEXT
3200 CONTINUE
3250 CONTINUE
  IF (NLM.LT.2) GO TO 3500
  DIST1 = BND(LL,JJ)
  DIST2 = BND(JJ,MM)
  DIST3 = SQRT(DIST1**2+DIST2**2-2.0*DIST1*DIST2*COSEXT)
  BND(LL,MM) = DIST3
  BND(MM,LL) = DIST3
  IF (DUMP) WRITE(20,13000) LL,DIST1,JJ,DIST2,MM,DIST3,
  1        COSINT,COSMID,COSEXT
3500 CONTINUE
RETURN
END
**SUBROUTINE EMBED (ISSUE)**

C*********************************************
C THIS VERSION OF EMBED (JULY 1980) WAS PIECED TOGETHER OUT OF ROUT-
C TINES WRITTEN BY GORDON CRIPPS, IRWIN KUNTZ, TODD NORDLAND AND
C TIM HAVEL, AND WAS IMPLEMENTED BY THE LAST AUTHOR ON THE LAWRENCE-
C BERKELEY LABORATORY'S CDC 7600 USING THEIR VERSION OF THE MINNESOTA
C FORTRAN COMPILER. SINCE THIS FORTRAN IS FAIRLY STANDARD, NO GREAT
C PROBLEMS SHOULD ARISE IN ATTEMPTING TO IMPLEMENT IT ON OTHER MACH-
C INES AND COMPILERS. POSSIBLE REQUIRED CHANGES ARE AS FOLLOWS: THE
C INTRINSIC RANDOM NUMBER GENERATOR 'RANF' USED IN DSTMAT AND EIGENV
C MUST BE CHANGED TO WHATEVER IS AVAILABLE LOCALLY; THE 'ENCODE' USED
C FOR VARIABLE FORMATING IN EIGENV IS NONSTANDARD AND SUPERFLUOUS IN
C THIS CONTEXT, AND MAY BE DISCARDED IF NECESSARY; SINCE THE CDC 7600
C PACKS 10 CHARACTERS PER WORD, WHICH IS MORE THAN MOST MACHINES, THE
C HANDELING OF CHARACTER STRINGS MAY HAVE TO BE CHANGED, ESPECIALLY
C FOR THE VARIABLE 'STATUS' IN SCCG. ALL OF THE REQUIRED PARAMETERS,
C WORKSPACE, INPUT DATA AND OUTPUT ARE PASSED TO & FROM EMBED IN COM-
C MON. ITS ONLY ARGUMENT BEING THE SYMBOLIC PROBLEM NAME. ONCE INSIDE
C EMBED, HOWEVER, THINGS ARE PASSED TO LOWER-LEVEL ROUTINES AS ARGU-
C MENTS VARIABLY DIMENSIONED AS 'NOIM'. THE ONLY EXCEPTION TO THIS
C RULE OCCURS IN THE OPTIMIZATION ROUTINE 'SCCG' AND THE FUNCTIONS IT
C ACTS UPON. HERE ONLY THE VARIABLES ARE PASSED AS ARGUMENTS, THE
C PARAMETERS AND WORKSPACE BEING IN COMMON. GOOD LUCK!
C
C*********************************************
C MODIFIED 9/80-10/80 BY D.H. SMITH FOR DEC-10 COMPATIBILITY
C*********************************************
C
C INPUT: A SET OF UPPER & LOWER BOUNDS ON THE INTERPOINT DISTANCES.
C TOGETHER WITH A SET OF ASSIGNED CHIRALITIES TO ANY CHIRAL
C CENTERS. NECESSARY PARAMETERS ARE DEFINED BELOW.
C
C OUTPUT: A SET OF 3D COORDINATES WHOSE DISTANCES LIE WITHIN THE
C PRESCRIBED BOUNDS & WHOSE CHIRAL CENTERS HAVE THE ASSIGNED
C CHIRALITIES IF SUCH A SET EXISTS, OR A SET OF 'BEST FIT'
C COORDINATES IF NOT.
C
C RETURNS: 1 IF THE 'EMBEDDING' WAS SUCCESSFUL, -1 IF EMBEDDING WAS
C NOT POSSIBLE, & ZERO IF A DECISION COULD NOT BE REACHED
C WITHIN THE LIMIT ON THE NO. OF OPTIMIZATION ITERATIONS.
C
C EXTERNAL ERRFCT,CHIRER
LOGICAL OK
INTEGER EMBED1,EMBED2
DIMENSION EVL(3),TVL(3)
COMMON/INFO/ TITLE(16),LTITLE,IOPN1

COMMON/XZSI/NATOM,XYZ(3,72)
THE NO. OF ATOMS OR RESIDUES, AND THE OUTPUT COORDINATES.

COMMON/BNDS/ NPNTS,BND(72,72)
THE NO. OF GEOMETRIC POINTS = NATOM + NO. OF OUTRIGGERS, AND THE
UPPER (ABOVE DIAGONAL) & LOWER (BELOW DIAGONAL) DISTANCE BOUNDS.
NB: THESE BOUNDS ARE RETURNED Squared AFTER CALLING EMBED.

COMMON/CHIR/ NCHIR,LIG(4,120),VAL(120)
THE NO. OF CHIRAL CENTERS, THE ATOMS THEREOF, AND WORKSPACE.

COMMON/EMBD/ DUMP,NDIM,NTRY,SMALL,DWK(72),EWK(72,72)
LOGICAL DUMP
IF DUMP = .FALSE., NO OUTPUT EXCEPT ERROR MESSAGES IS PRODUCED.
NDIM = ARRAY DIMENSIONS = MAX. NO. OF POINTS TO BE USED.
NTRY = NO. OF ATTEMPTS TO PRODUCE VIEABLE TRIAL COORDINATES.
SMALL = MIN. RATIO FCT. VALUE TO GRAD. NORM FOR NONEMBEDDABILITY.
DW1 AND EWK ARE REQUIRED WORKSPACE.

COMMON /SCCGC/ WK1(3,72),WK2(3,72),WK3(3,72),GRD(3,72)
WORKSPACE FOR EMBED AND SCCG, AND GRADIENT OF ERROR FUNCTIONS.

COMMON /PARA/ GESFCT,STPMIN,STPMAX,CAPPA,DELTA,EPSLN,
& FAST,SLOW,ANGMAX,INTMAX,RESTAR,ISEED
INTEGER RESTAR
COMMON /CTRL/ SLFCOR,IFREQ,ITNLMIM,FCTMIN,GRDMIN
LOGICAL SLFCOR
PARAMETERS FOR THE OPTIMIZER SCCG; DEFINED IN THAT ROUTINE.

DIMENSION LABEL1(10),LABEL2(5)

DIMENSION IERROR FOR WORD LENGTH COMPATIBILITY

CHARACTER*5 IERROR(2)
DATA (LABEL1(I),I=1,10) /4HTRIA,4HNGLE,4HSMO,4HOTHE,4HD BO,
& 4HUNDS,4HTO,4HEMBE,4HD,4H /
DATA (LABEL2(I),I=1,5) /4HAFT,4HER,0,4HPTIM,4HIPZAT,4HION /
ISTRCT = 0
IF (DUMP) WRITE(20,10) (TITLE(NNNI,NNN=1,16)

INTITIALIZE THE CHIRALITY CONSTRAINTS; ENTRY EMBED1 SKIPS THIS.
IF (NCHIR.GT.0) CALL CHIRIN (DUMP,NDIM,NCHIR,LIG,BND,VAL)
ENTRY EMBED1 (ISSUE)
ISTRCT = ISTRCT + 1

USE THE TRIANGLE AND INVERSE TRIANGLE INEQUALITIES TO SMOOTH BOUNDS.

CALL TRNGL (DUMP,NDIM,NPNTS,10,BND)
CALL TRINV (DUMP,NDIM,NPNTS,10,BND)
IF (DUMP) CALL GRAFIC (NDIM,NPNTS,BND,LABEL1)
GO TO 2

IF WE USE ENTRY EMBED2, THE BOUNDS ARE ASSUMED SQUARED BUT OTHERWISE
READY TO USE. THUS EMBED2 MAY BE USED TO GENERATE ADDITIONAL STRUCT-
URES WITHOUT WASTING TIME ON MULTIPLE PASSES THROUGH TRNGL & TRINV.

ENTRY EMBED2 (ISSUE)
ISTRCT = ISTRCT + 1
NM1 = NPNTS-1
DO 1 I=1,NM1
IP1 = I+1
DO 1 J=IP1,NPNTS
BND(I,J) = SQRT(BND(I,J))
BND(J,I) = SQRT(BND(J,I))
1 CONTINUE

SET VARIABLES FOR USE IN TRIAL COORDINATE GENERATION; IF WE ARE
USING THE SELF-CORRECTION OPTION & OUTRIGGERS, WK2 MUST BE SET
TO THE COORDINATES OF THE OUTRIGGERS FOR USE IN SCCG/COEFCT.

2 RMS = 1.E35
RMS was 1.0E8 to represent a large number - we can only go to E38 on the PDP-10, so I made it slightly smaller...

DO 3 I=1,3
DO 3 J=1,NPNTS
WK1(I,J) = XYZ(I,J)
IF (SLFCOR .AND. J.GT.NATOM) WK2(I,J) = XYZ(I,J)
3 CONTINUE

GENERATE TRIAL COORDINATES FOR OPTIMIZATION USING THE PROCEDURE OF CRIPPEN AND HAVEL, ACTA CRYS (1978) A34 282-4. WE REPEAT THE RELATIVELY CHEAP METRIC MATRIX GENERATION UNTIL WE GET < 2 NEGATIVE SQUARED DISTANCES TO THE CENTER OF MASS, AND WE TRY UP TO NTRY TIMES TO GET THE EIGENVALUES TO BE POSITIVE AND IN ORDER.

DO 5 ITRY=1,NTRY
    added to prevent run away jobs 8/16/99 tws
    DO 4 ISTART=1,5
CALL DSTMAT (NDIM,NPNTS,EWK,BND,ISEED)
    IF (DUMP) WRITE (20,80)
80 FORMAT (/,' STARTING DISTANCE MATRIX')
    IF (DUMP) CALL MATPRT (EWK,NDIM,NPNTS)
    CALL METRIC (DUMP,NDIM,NPNTS,DWK,EWK,NNEG)
    IF (NNEG.LE.1) GO TO 800
4 CONTINUE
    IF (DUMP) WRITE(20,110)
110 FORMAT(/,' CANNOT FIND GOOD STARTING COORDINATES')
    ISSUE=-1
    RETURN
800 IF (DUMP) WRITE(20,90)
90 FORMAT(/,' METRIC MATRIX')
    IF (DUMP) CALL MATPRT (EWK,NDIM,NPNTS)
    CALL EIGENV (DUMP,NDIM,NPNTS,3,100,1.E-6,TVL,WK1,DWK,OK)
    CALL COORDS (DUMP,OK,NDIM,NATOM,NPNTS,(NTRY-ITRY),&, BND,TVL,EVL,WK1,XYZ,RMS)
    IF (OK) GO TO 6
5 CONTINUE
    WR ITE (20,20) (EVL(I),I=1,3),RMS

OPTIMIZE. 1ST ON THE CHIRALITY CONSTRAINTS ALONE AS GETTING THESE RIGHT TO BEGIN WITH GREATLY SPEEDS THINGS, AND THEN ON THE TOTAL ERROR FUNCTION WHICH GOES TO ZERO ONLY WHEN BOTH CHIRALITY AND DISTANCE CONSTRAINTS HAVE BEEN MET.

6 IF (DUMP) CALL XDSHOW (EWK,XYZ,NDIM,NPNTS)
    IF (NCHIR.LE.0) GO TO 7
    CALL SCCG (CHIRER,NATOM,XYZ,NFCT,NITN,FCTVAL,GRDNRM,IERROR,DUMP)
    IF (DUMP) CALL XDSHOW (EWK,XYZ,NDIM,NPNTS)
7 CALL SCCG (ERRFCT,NATOM,XYZ,NFCT,NITN,FCTVAL,GRDNRM,IERROR,DUMP)

I F (.NOT.DUMP) GO TO 8
    WR ITE (20,30) ISTRCT
    WR ITE (20,40) I NITN,NFCT,FCTVAL,GRDNRM
    IF ((IERROR(1).NE.' ' .AND. IERROR(2).NE.' ') .AND.
1(IERROR(1).NE.' OVER' .AND. IERROR(2).NE.' LIMIT'))
2 WR ITE (20,50) (IERROR(NNN), NNN=1,2)
    WR ITE (20,60)
7 WR ITE (20,70) (J,(XYZ(I,J),I=1,3),J=1,NPNTS)
    CALL RMSERS (DUMP,.TRUE.,NDIM,NPNTS,XYZ,BND,LABEL2, &, RMS,HI,IHI,JHI,HO,IHO,JHO)
CALL DMDUMP (NDIM, NPNTS, XYZ, BND, EWK)

C IF FCTVAL < FCTMIN, THE BOUNDS ARE ASSUMED TO BE EMBEDDABLE;
C ELSE IF THE GRADIENT IS NOT 'SMALL' COMPARED TO THE ERROR, THE
C SITUATION IS INDETERMINATE; ELSE THE STRUCTURE IS NONEMBEDDABLE.
C
8 ISSUE = -1
IF (FCTVAL/GRDNRM .LT. SMALL) ISSUE = 0
IF (FCTVAL .LT. FCTMIN) ISSUE = 1

RETURN

C

10 FORMAT(/,' BEGIN EMBED (' ,16A5,')'),/)
20 FORMAT(/,' *** CAUTION: POOR INITIAL COORDINATES ***',/)
& 5X, 'EIGENVALUES = ', 3E13.6, /,
& 5X, 'RMS DEVIATION FROM BOUNDS = ',E13.6, /)
30 FORMAT(/,' RESULTS OF EMBED: STRUCTURE NUMBER = ',I8)
40 FORMAT(/,' NUMBER OF ITERATIONS = ',I8,/, 
& 15X, 'NUMBER OF FUNCTION CALLS = ',I8,/, 
& 19X, 'FINAL ERROR ACHIEVED = ',E13.6,/, 
& 19X, 'FINAL GRADIENT NORM = ',E13.6,/) 
50 FORMAT(/,' *** CAUTION: ERROR ',2A5' IN SCCG ***',/)
60 FORMAT(/,' ' , ' EMBEDDED COORDINATES: ',/)
70 FORMAT(1X,I5.3F16.6)

END

SUBROUTINE CHIRIN (DUMP,NDIM,NCHIR,LIG,BND,VAL)

C SETS THE VALUES OF THE CHIRIAL TRIPLE PRODUCT FOR USE BY CHIRER.
C THIS IS EQUAL THE SQRT OF THE DETERMINANT OF THE MATRIX OF DOT
C PRODUCTS OF THE VECTORS FROM ATOM 4 TO ATOMS 1, 2 & 3. ASSUMES
C THAT IN EACH CHIRIAL QUARTET 1 -> 2 -> 3 COUNTERCLOCKWISE WHEN
C ATOM 4 IS POINTING AWAY FROM YOU. NB: ALL FOUR ATOMS MUST BE AT
C AT FIXED DISTANCES FROM EACHOTHER.
C
LOGICAL DUMP
DIMENSION LIG(4,NCHIR),VAL(NCHIR),BND(NDIM,NDIM)
DOT(A,B,C) = ( A**2 + B**2 - C**2 ) / 2.
IF (DUMP) WRITE(20,10)
DO 1 I=1,NCHIR
A1 = BND(LIG(1,I),LIG(4,I))**2
B2 = BND(LIG(2,I),LIG(4,I))**2
C3 = BND(LIG(3,I),LIG(4,I))**2
A2 = DOT(BND(LIG(1,I),LIG(4,I)),BND(LIG(2,I),LIG(4,I)),
& BND(LIG(1,I),LIG(2,I)))
A3 = DOT(BND(LIG(2,I),LIG(4,I)),BND(LIG(3,I),LIG(4,I)),
& BND(LIG(1,I),LIG(3,I)))
B3 = DOT(BND(LIG(2,I),LIG(4,I)),BND(LIG(3,I),LIG(4,I)),
& BND(LIG(2,I),LIG(3,I)))
B1 = A2
C1 = A3
C2 = B3
IF (DET.LT.1.0E-4) DET = 0.0
VAL(I) = SQRT(AMAX1(0.0,DET))
IF (DUMP) WRITE(20,11) DET
1 CONTINUE
RETURN
10 FORMAT(/,' RESULTS OF CHIRALITY INITIALIZATION: ',/)
11 FORMAT(' DETERMINANT = ',E13.6)
END
SUBROUTINE TRNGL (DUMP, ND, NA, LIM, B)

ITERATIVELY GO OVER ALL TRIPLES OF ATOMS, LOWERING THE UPPER
BOUNDS WHENEVER THESE EXCEED TRIANGLE INEQUALITY LIMIT.

DIMENSION B(ND,ND)
LOGICAL DUMP
N = NA
NM1 = N-1
DO 1 L=1,LIM
NCHG = 0
DO 2 I=1,NM1
IM1 = I-1
DO 2 J=IM1,N
   K = I, K < J
      IF (IM1 .EQ. 0) GO TO 4
      DO 3 K=IM1,1
         IF (B(K,I) + B(K,J) .GE. B(I,J)) GO TO 3
         B(I,J) = B(K,I) + B(K,J)
         IF (B(I,J) .LT. B(J,I)) WRITE(20,9) J,I,B(J,I),K,I,B(K,I),K,J.
            1B(K,J),L
            NCHG = NCHG+1
      CONTINUE
   3 CONTINUE
   CONTINUE
   K > I, K < J
      JM1 = J-1
      IF (JM1 .LT. IP1) GO TO 6
      DO 5 K=IP1,JM1
         IF (B(I,K) + B(K,J) .GE. B(I,J)) GO TO 5
         B(I,J) = B(I,K) + B(K,J)
         IF (B(I,J) .LT. B(J,I)) WRITE(20,9) J,I,B(J,I),I,K,B(I,K),K,J.
            1B(K,J),L
            NCHG = NCHG+1
      CONTINUE
   5 CONTINUE
   CONTINUE
   K > I, K > J
      JP1 = J+1
      IF (JP1 .GT. N) GO TO 2
      DO 7 K=JP1,N
         IF (B(I,K) + B(J,K) .GE. B(I,J)) GO TO 7
         B(I,J) = B(I,K) + B(J,K)
         IF (B(I,J) .LT. B(J,I)) WRITE(20,9) J,I,B(J,I),I,K,B(I,K),J,K.
            1B(J,K),L
            NCHG = NCHG+1
      CONTINUE
   7 CONTINUE
   CONTINUE
   IF (NCHG .EQ. 0) RETURN
   IF (DUMP) WRITE(20,8) L, NCHG
      8 FORMAT(1, ' ITERATION', IS, ' OF TRNGL', 15, ' CHANGES MADE')
   1 CONTINUE
      9 FORMAT(1, ' WARNING: LOWER BOUND', 214, ' > UPPER BOUND', 15, 15, 15, ' AT ITRN ', IS, ' OF TRNGL')
   2 RETURN
END

SUBROUTINE TRINV (DUMP, ND, NA, LIM, B)

ITERATIVELY GO OVER ALL ATOM TRIPLES, RAISING THE LOWER BOUNDS WHENEVER THESE ARE LESS THAN POSSIBLE BY THE INVERSE TRIANGLE EQUALITY.
NB: MUST BE DONE ONLY AFTER USING TRNGL ON THE UPPER BOUNDS.

DIMENSION B(ND,ND)
LOGICAL DUMP
N = NA
NM1 = N-1
DO 1 L=1,LIM
NCHG = 0
DO 2 I=1,NM1
IP1 = I+1
IM1 = I-1
DO 2 J=IP1,N
   IF (IM1 .EQ. 0) GO TO 4
   DO 3 K=1,IM1
      IF (B(I,K) .GE. B(J,I) - B(K,J)) GO TO 33
      B(I,K) = B(J,I) - B(K,J)
      IF (B(K,I) .LT. B(I,K)) WRITE(20,9) K,I,B(K,I),J,I,B(J,I),K,J,
         1B(K,J),L
   GO TO 333
33 IF (B(J,K) .GE. B(J,I) - B(K,I)) GO TO 3
   B(J,K) = B(J,I) - B(K,I)
   IF (B(K,J) .LT. B(J,K)) WRITE(20,9) K,J,B(K,J),J,I,B(J,I),K,J,
      1B(K,I),L
333 NCHG = NCHG+1
3 CONTINUE
4 CONTINUE
C K < I, K < J
   JM1 = J-1
   IF (JM1 .LT. IP1) GO TO 6
   DO 5 K=IP1,JM1
      IF (B(K,I) .GE. B(J,I) - B(J,K)) GO TO 55
      B(K,I) = B(J,I) - B(J,K)
      IF (B(I,K) .LT. B(K,I)) WRITE(20,9) I,K,B(I,K),J,I,B(J,I),K,J,
         1B(K,I),L
   GO TO 555
55 IF (B(J,K) .GE. B(J,I) - B(I,K)) GO TO 5
   B(J,K) = B(J,I) - B(I,K)
   IF (B(K,J) .LT. B(K,J)) WRITE(20,9) K,J,B(K,J),J,I,B(J,I),I,K,
      1B(I,K),L
555 NCHG = NCHG+1
5 CONTINUE
6 CONTINUE
C K > I, K > J
   JP1 = J+1
   IF (JP1 .GT. N) GO TO 2
   DO 7 K=JP1,N
      IF (B(K,I) .GE. B(J,I) - B(J,K)) GO TO 77
      B(K,I) = B(J,I) - B(J,K)
      IF (B(I,K) .LT. B(K,I)) WRITE(20,9) I,K,B(I,K),J,I,B(J,I),J,K,
         1B(J,K),L
   GO TO 777
77 IF (B(K,J) .GE. B(J,I) - B(I,K)) GO TO 7
    B(K,J) = B(J,I) - B(I,K)
    IF (B(K,J) .LT. B(K,J)) WRITE(20,9) J,K,B(K,J),J,I,B(J,I),I,K,
       1B(I,K),L
777 NCHG = NCHG+1
7 CONTINUE
2 CONTINUE
IF (NCHG .EQ. 0) RETURN
IF (DUMP) WRITE (20, 8) L, NCHG
8 FORMAT (' ITERATION', IS, ' OF TRINV;', IS, ' CHANGES MADE')
1 CONTINUE
9 FORMAT (/,' WARNING: UPPER BOUND', 2X, ' = ', F8.3, '< LOWER BOUND',
& ' = ', F8.3, ' - UPPER BOUND', 2X, ' = ', F8.3,
& ' AT ITRN ', 12, ' OF TRINV'/)
RETURN
END

SUBROUTINE GRAFIC (ND, NA, BND, LABEL)
C
C OUTPUTS LARGE MATRICES
C
DIMENSION BND (ND, ND)
DIMENSION LABEL (10)
N = NA
WRITE (20, 11)
WRITE (20, 12) LABEL
DMIN = BND (1, 1)
DMAX = BND (1, 1)
AMIN = BND (1, 2)
AMAX = BND (1, 2)
BMIN = BND (2, 1)
BMAX = BND (2, 1)
DO 1 I = 2, N
IF (BND (I, I) .GT. DMAX) DMAX = BND (I, I)
IF (BND (I, I) .LT. DMIN) DMIN = BND (I, I)
IM1 = I - 1
DO 1 J = 1, IM1
IF (BND (J, I) .GT. AMAX) AMAX = BND (J, I)
IF (BND (J, I) .LT. AMIN) AMIN = BND (J, I)
IF (BND (I, J) .GT. BMAX) BMAX = BND (I, J)
IF (BND (I, J) .LT. BMIN) BMIN = BND (I, J)
1 CONTINUE
RCL = (BMAX - BMIN) / 5.
SCL = (AMAX - AMIN) / 5.
TCL = (DMAX - DMIN) / 10.
IF (RCL .EQ. 0) RCL = 1.
IF (SCL .EQ. 0) SCL = 1.
IF (TCL .EQ. 0) TCL = 1.
WRITE (20, 2) AMIN, AMAX
WRITE (20, 3) DMIN, DMAX
WRITE (20, 4) BMIN, BMAX
2 FORMAT (' ABOVE DIAG ELTS RANGE FROM', F9.3, ' TO', F9.3)
3 FORMAT (' DIAGONAL ELTS RANGE FROM', F9.3, ' TO', F9.3)
4 FORMAT (' BELOW DIAG ELTs RANGE FROM', F9.3, ' TO', F9.3, '/)
CALL MATPRT (BND, ND, NA)
WRITE (20, 11)
11 FORMAT ('----------------------------------------------------------'/)
12 FORMAT (1X, 10A4, ':'/)
RETURN
END

SUBROUTINE MATPRT (A, NDIM, N)

THIS ROUTINE PRINTS A REAL ARRAY IN A DECENT FORMAT

DIMENSION A(NDIM, NDIM)

LIM1 = 1
10 LIM2 = LIM1 + 11
IF (LIM2.GT.N) LIM2 = N
WRITE(20,1010) (K,K=LIM1,LIM2)
1010 FORMAT (/"/9X,11(I3,6X),13)
DO 20 I = 1,N
WRITE (20,1020) I, (A(I,J),J=LIM1,LIM2)
1020 FORMAT (1X,I3,2X,12F9.5)
20 CONTINUE
IF (LIM2.EQ.N) RETURN
LIM1 = LIM1 + 12
GO TO 10
END
SUBROUTINE XD SHOW (DIST,XYZ,ND,NA)
DIMENSION DIST(ND,ND), XYZ(3,72)
WRITE (20,10)
10 FORMAT (I, ' CURRENT COORDINATES ARE: ')
DO 100 I = LNA
WRITE (20,20) I, XYZ(1,I), XYZ(2,I), XYZ(3,I)
20 FORMAT (5X,I2,JF15.6)
100 CONTINUE
WRITE (20,30)
30 FORMAT (I, ' AND THE CURRENT DISTANCE MATRIX IS: ')
DO 200 I = 1,NA
DO 200 J = 1, I
DIST(I,J) = SQRT((XYZ(I,1)-XYZ(I,J)**2 + (XYZ(I,2)-XYZ(I,J)**2
1 + (XYZ(I,3)-XYZ(I,J)**2
DIST(J,I) = DIST(I,J)
200 CONTINUE
CALL MATPRT (DIST,ND,NA)
RETURN
SUBROUTINE DSTMAT (ND, NPNTS, DMX, BND, ISEED)
COMPUTES A MATRIX OF TRIANGLE-CORRELATED RANDOM FRACTIONS OF THE
DISTANCES BETWEEN THE UPPER AND LOWER BOUNDS (SEE CRIPPEN, IN PRESS)
DIMENSION DMX(ND,ND), BND(ND,ND)
N = NPNTS
NM1 = N-1
DO 20 I=1,NM1
JSTART=I+1
DMX(I,I) = 0.
DO 10 J=JSTART,N
DMX(J,I) = RAN(ISEED)
10 CONTINUE
DMX(N,N) = 0.
CALCULATE DISTANCE CORRELATIONS
DO 100 I=1,NM1
JSTART=I+1
DMX(I,I) = 0.
DO 90 J=JSTART,N
DENOM = 0.
DMX(I,J) = 0.
100 CONTINUE
DO 80 K=1,N
MIK = MAXO(I,K)
MJK = MAXO(J,K)
NIK = MINO(I,K)
NJK = MINO(J,K)
IF(K.EQ.I) GO TO 80
IF(K.NE.J) GO TO 40

C C SELF CORRELATION IS 1.
C
DMX(I,J)=DMX(I,J)+DMX(J,I)
DENOM=DENOM+1.
GO TO 80

C C POSITIVE CORRELATION CASES
C
40 IF(BND(NJK,MJK) .GT.0.2*BND(NIK,MIK)) GO TO 50
IF(I.GT.K) CORR=0.9*DMX(I,K)
IF(K.GT.I) CORR=0.9*DMX(K,I)
DMX(I,J)=DMX(I,J)+CORR
DENOM=DENOM+0.9
GO TO 80

50 IF(BND(NIK,MIK) .GT.0.2*BND(NJK,MJK)) GO TO 60
IF(J.GT.K) CORR=0.9*DMX(J,K)
IF(K.GT.J) CORR=0.9*DMX(K,J)
DMX(I,J)=DMX(I,J)+CORR
DENOM=DENOM+0.9
GO TO 80

C C NEGATIVELY CORRELATED DISTANCES
C
60 IF(BND(MIK,NIK) .LT.0.9*BND(NJK,MJK)) GO TO 70
IF(J.GT.K) CORR=0.5*(1.-DMX(J,K))
IF(K.GT.J) CORR=0.5*(1.-DMX(K,J))
DMX(I,J)=DMX(I,J)+CORR
DENOM=DENOM+0.5
GO TO 80

70 IF(BND(MJK,NJK) .LT.0.9*BND(NIK,MIK)) GO TO 80
IF(I.GT.K) CORR=0.5*(1.-DMX(I,K))
IF(K.GT.I) CORR=0.5*(1.-DMX(K,I))
DMX(I,J)=DMX(I,J)+CORR
DENOM=DENOM+0.5
GO TO 80

80 CONTINUE
90 DMX(I,J)=DMX(I,J)/DENOM
100 CONTINUE

C C SCALE WITHIN BOUNDS TO GET TRIAL DISTANCE MATRIX
C
DO 200 I=1,NM1
IP1 = I+1
DO 200 J=IP1,N
DMX(I,J) = BND(J,I) + (BND(I,J)-BND(J,I)) * DMX(I,J)
DMX(J,I) = DMX(I,J)
200 CONTINUE
RETURN
END

SUBROUTINE METRIC (DUMP,NDIM,NPNTS,DSQ,GMX,NNEG)

COMPUTES THE METRIC OR GRAM MATRIX OF ALL POSSIBLE DOT PRODUCTS
BETWEEN THE ATOMIC VECTORS TO THE CENTER OF MASS USING THE LAW
OF COSINES AND THE FOLLOWING FORMULA FOR THE DISTANCES TO THE CENTER OF MASS: 
\[ DCM(I) = \left( \frac{1}{N} \right) \sum_{J=1,N} (DST(I,J)^2) - \left( \frac{1}{N^2} \right) \sum_{J<K} (DST(J,K)^2) \]

LOGICAL DUMP

DIMENSION DSQ(NDIM),GMX(NDIM,NDIM)
N = NPNTS
NM1 = N-1

DSUMJK = 0.
DO 10 I=1,NM1
   IP1=I+1
   DO 10 J=IP1,N
      GMX(I,J) = GMX(I,J)^2
      GMX(J,I) = GMX(I,J)
      DSUMJK = DSUMJK + GMX(I,J)
   10 CONTINUE

DIVIDE BY N SQUARED FOR CENTER OF MASS CALCULATIONS

DSUMJK = DSUMJK / FLOAT(N*N)
NNEG = 0
DO 30 I=1,N

SUM SQUARED DISTANCES FROM ATOM I

DSUMJ = 0.
DO 20 J=1,N
   DSUMJ = DSUMJ + GMX(I,J)
20 CONTINUE

CENTER OF MASS DERIVED USING THE ABOVE FORMULA

DSQ(I) = DSUMJ / FLOAT(N) - DSUMJK
IF (DSQ(I) .LT. 0.) NNEG = NNEG+1
30 CONTINUE

CALCULATION OF METRIC MATRIX USING THE LAW OF COSINES

DO 40 I=1,NM1
   IP1 = I+1
   GMX(I,1) = DSQ(I)
   DO 40 J=IP1,N
      GMX(I,J) = (DSQ(I) + DSQ(J) - GMX(I,J)) / 2.
   40 CONTINUE

GMX(N,N) = DSQ(N)

IF (DUMP) WRITE (20,50) DSUMJK
IF (DUMP) WRITE 120,60) IOSQ(I) ,I=I,N)

RETURN

50 FORMAT(/, ' RESULTS OF METRIC: SQUARED RADIUS OF GYRATION = ', & F13.6, '/,' 'SQUARED DISTANCES TO THE CENTER OF MASS:/', ')
60 FORMAT(10F13.6)
END

SUBROUTINE EIGENV (DUMP,NDIM,NPNTS,LAM,LIM,RAT, EVL, EVC, DWK, EWK, OK)
COMPUTES THE FIRST LAM LARGEST MAGNITUDE EIGENVALUES/EIGENVECTORS OF
A REAL SYMMETRIC MATRIX BY THE POWER METHOD; OK = .TRUE. IMPLIES
THAT THE EIGENVALUES CAME OUT POSITIVE AND IN ORDER.

LOGICAL OK,DUMP
DIMENSION EVL(3),EVC(3,NDIM),DWK(NDIM),EWK(NDIM,NDIM)
INTEGER FMT(2)
N = NPNTS
OK = .TRUE.
ENCODE(10,1,FMT) LAM
1 FORMAT(1H1,I1,6HE1S.6)

DO 333 M=1,LAM
EVL(M) = 0.

1ST MATRIX MULTIPLICATION; INITIAL GUESS IS OLD COORDINATES

DOT1=0.
DO 101 I=1,N
DWK(I)=0.
DO 100 J=1,N
DWK(I) = DWK(I) + EWK(I,J)*EVC(M,J)
100 CONTINUE
DOT1 = DOT1 + DWK(I)**2
101 CONTINUE

IF IN OR NEAR NULL SPACE, USE A RANDOM GUESS AT FIRST VECTOR

IF (DOT1.GT.FLOAT(N)/10000.) GO TO 111
DO 110 I=1,N
DWK(I) = RAN(0)
110 CONTINUE

111 ITOT = 0
DO 300 ITER=1,LIM

MULTIPLY VECTOR BY MATRIX AND COMPUTE DOT PRODUCTS

DOT1=0.
DOT2=0.
DO 202 I=1,N
EVC(M,I)=0.
DO 200 J=1,N
EVC(M,I) = EVC(M,I) + EWK(I,J)*DWK(J)
200 CONTINUE
DOT1 = DOT1 + EVC(M,I)**2
DOT2 = DOT2 + EVC(M,I)*DWK(I)
202 CONTINUE

NORMALIZE NEW VECTOR AND SUBSTITUTE FOR OLD

RATIO = ABS((EVL(M)-DOT2)/DOT2)
EVL(M) = DOT2
DOT1 = SQRT(DOT1)
DO 220 I=1,N
EVC(M,I) = EVC(M,I) / DOT1
DWK(I) = EVC(M,I)
220 CONTINUE
WHEN DOT OF NEW WITH OLD HAS CONVERGED TO EIGENVALUE WITH ACCURACY
WITHIN RAT, GO TO COMPUTATION OF NEXT EIGENVALUE/EIGENVECTOR.

IF (RATIO.LT.RAT) GO TO 303

CONTINUE
OK = .FALSE.

ELIMINATE OLD EIGENVALUE FROM MATRIX

ITOT = ITOT + ITER
DO 330 I=1,N
    DO 330 J=1,N
    EWK(I,J) = EWK(I,J) - EVL(M)*EVC(M,I)*EVC(M,J)
330 CONTINUE

changed 0 to -0.2e-6 to catch accuracy problems
especially for planar molecules like ozone
IF (EVL(M).LT.-0.2E-6) OK = .FALSE.

CONTINUE

IF (ABS(EVL(1)).LT.ABS(EVL(2))) OK = .FALSE.
IF (ABS(EVL(2)).LT.ABS(EVL(3))) OK = .FALSE.

IF (DUMP) WRITE (20,1000) ITOT
IF (DUMP) WRITE (20,FMT) (EVL(I),!=1,LAM)
IF (DUMP) WRITE (20,1001)
RETURN

FORMAT(/,' RESULTS OF EIGENV: ',ITOT,' ITERATIONS TOTAL',/,,
      & ' EIGENVALUES: ')
1000 FORMAT(/,'
      ')
1001 FORMAT(/,' EIGENVECTORS: ')

END

SUBROUTINE COORDS (DUMP,OK,NDIM,NATOM,NPNTS,IPS,BND,
      & EVLA,EVLB,XYZA,XYZB,RMSB)

CONVERTS EIGENVALUES/VECTORS INTO COORDINATES AND CALLS RMSERS TO
COMPUTE THE RMS DEVIATION FROM THE BOUNDS; IF THIS VALUE IS THE
LOWEST SO FAR OR THE EIGENVALUES ARE POSITIVE & IN ORDER, IT SAVES
THESE COORDINATES FOR USE IN OPTIMIZATION. AND SQUARES THE BOUNDS
USE BY ERRFCT DURING OPTIMIZATION.

LOGICAL OK,DUMP
DIMENSION EVLA(3),EVLB(3),XYZA(3,NDIM),XYZB(3,NDIM),BND(NDIM,NDIM)
DIMENSION LABEL(S)
DATA (LABEL(I),1=1,5) /4H BEF,4HORE,4HOPTI,4HMIZA,4HTION/
NA = NATOM
NP = NPNTS
NPM1 = NP-1

DO 1 I=1,3
    ROOT = SQRT(ABS(EVLA(I)))
    DO 1 J=1,NA
    XYZA(I,J) = ROOT * XYZA(I,J)
1 CONTINUE
CALL RMSERS (DUMP,.FALSE.,NDIM,NPNTS,XYZA,BND,LABEL,
& RMSA,HI,IHI,JHI,HO,IHO,JHO)

SAVE THE COORDINATES IF THEY'RE 'OK', OR HAVE THE LOWEST RMS SO FAR.

IF (.NOT.OK .AND. RMSA.GT.RMSB) GO TO 3
DO 2 I=1,3
EVLB(I) = EVLA(I)
DO 2 J=1,NA
XYZB(I,J) = XYZA(I,J)
2 CONTINUE
RMSB = RMSA

SQUARE THE BOUNDS FOR USE BY ERRFCT IF WE ARE READY TO GO ON.

3 IF (.NOT.OK .AND. IPS.GT.0) RETURN
DO 4 J=1,NPNTS-1
JPI = J+1
DO 4 K=JPI,NP
BND(J,K) = BND(J,K)**2
BND(K,J) = BND(K,J)**2
4 CONTINUE
RETURN
END

SUBROUTINE RMSERS (DUMP,SQUARE,NDIM,NPNTS,XYZ,BND,LABEL,
& RMSP,HIP,IHIP,JHIP,HOP,IHOP,JHOP)

COMPUTES THE RMS DEVIATION FROM THE BOUNDS AS WELL AS THE MAXIMUM
ABSOLUTE DEVIATIONS AND POSITIONS THEREOF FOR USE IN EVALUATING CONVERGENCE. IF SQUARE = .TRUE. THE BOUNDS ARE ASSUMED TO BE SQUARED.

LOGICAL DUMP,SQUARE
DIMENSION LABEL(5),XYZ(3,NDIM),BND(NDIM,NDIM)
REAL LOBD,UPBD
NM1 = NPNTS-1
RMS = 0.
HIMX = 0.
HOMX = 0.
DO 1 I=1,NM1
IPI = I+1
DO 1 J=IPI,NPNTS
UPBD = BND(I,J)
LOBD = BND(J,I)
IF (SQUARE) UPBD = SQRT(UPBD)
IF (SQUARE) LOBD = SQRT(LOBD)
DSTSQ = (XYZ(1,I)-XYZ(1,J))**2 + (XYZ(2,I)-XYZ(2,J))**2 + 
& (XYZ(3,I)-XYZ(3,J))**2
DST = SQRT(DSTSQ)
HI = DST-UPBD
HO = LOBD-DST
IF (HI.GT.0.) RMS = RMS + HI**2
IF (HO.GT.0.) RMS = RMS + HO**2
IF (.NOT.DUMP) GO TO 1
IF (HI.GT.HIMX) IHI = I
IF (HI.GT.HIMX) JHI = J
IF (HO.GT.HOMX) IHO = I
IF (HO.GT.HOMX) JHO = J
HIMX = AMAX1(HI,HIMX)
HOMX = AMAX1(HO,HOMX)
1 CONTINUE
NPAIR = NPNTS*(NPNTS-1)/2
RMS = SQRT(RMS/FLOAT(NPAIR))
IF (DUMP) WRITE (20,11) LABEL,HOMX,IHO,JHO,HIMX,IHI,JHI,RMS
RMSPI = RMS
HIP = HIMX
IHIP = IHI
JHIP = JHI
HOP = HOMX
IHOP = IHO
JHOP = JHO
RETURN
11 FORMAT(/,' FIT TO BOUNDS ',5A4,':',
& ' MAX LOWER BOUND VIOLATION = ',F12.6,' OCCURS AT ',215,
& '/36X,' MAX UPPER BOUND VIOLATION = ',F12.6,' OCCURS AT ',215,
& '/36X,' RMS DEVIATION FROM BOUNDS = ',F12.6,1)
END
SUBROUTINE DMDUMP (NDIM, NPNTS, XYZ, BND, DMD)
C
C GENERATES THE DISTANCE MATRIX OF THE FINAL STRUCTURE IN UPPER HALF
C OF DMD AND THE INDIVIDUAL TERMS OF THE FINAL ERROR IN THE LOWER.
C
DIMENSION LABEL(10), XYZ(3,NDIM), BND(NDIM,NDIM), DMD(NDIM,NDIM)
DATA (LABEL(I),I=1,10) /4HL FINA,4HL DI,4HL ST M,4HL HAT A,4HL BV;,
& 4HL DCM,4HL DIA,4HL GE,4HL ERROR,4HL BEL/ 
N = NPNTS
DO 1 I=1,N
DMD(I,I) = 0.
1 CONTINUE
NM1 = N-1
DSQSUM = 0.
DO 2 I=1,NM1
IP1 = I+1
DO 2 J=IP1,N
DSTSQ = (XYZ(1,I)-XYZ(1,J))**2 + (XYZ(2,I)-XYZ(2,J))**2 &
+ (XYZ(3,I)-XYZ(3,J))**2
DSQSUM = DSQSUM + DSTSQ
DMD(I,I) = DMD(I,I) + DSTSQ
DMD(J,J) = DMD(J,J) + DSTSQ
DST = SQRT(DSTSQ)
DMD(I,J) = DST
DMD(J,I) = 0.
IF (DSTSQ.GT.BND(I,J)) DMD(J,I) = ((DSTSQ-BND(I,J)) / BND(I,J))**2
IF (DSTSQ.LT.BND(J,I)) DMD(J,I) = ((BND(J,I)-DSTSQ) / BND(J,I))**2
2 CONTINUE
FN = FLOAT(N)
ROFGSQ = DSQSUM/FN**2
DO 3 I=1,N
DMD(I,I) = SQRT(DMD(I,I)/FN - ROFGSQ)
3 CONTINUE
CALL GRAFIC (NDIM,N,DMD,LABEL)
ROFGSI = SQRT(ROFGSQ)
WRITE(20,10) ROFGSI
RETURN
10 FORMAT(/,' RADIUS OF GYRATION = ',E13.6,1)
END
FUNCTION ERRFCT (NATOM,XYZ,GRD,NOGRAD)

C ERROR FUNCTION FOR EMBEDDING WHICH GOES TO ZERO IFF THE BOUNDS ARE SATISFIED, AND GRADIENT THEREOF WRT CARTESIAN COORDINATES.
C IN THIS VERSION WE DIVIDE EACH TERM BY THE BOUND VIOLATED SO THAT WE EXERT AN EQUAL 'PULL' ON SHORT AND LONG DISTANCES.
C NB: VARIABLY DIMENSIONING WITH NATOM WILL RESULT IN AN ARRAY SUBSCRIPT OUT OF BOUNDS IF WE HAVE OUTRIGGERS, BUT SINCE THE ARRAYS ARE CONTIGUOUS IN MEMORY THIS WILL BE ALL RIGHT.
C
COMMON /BND(72,72)/ NPNTS,BND(72, 72)
DIMENSION GRD(3,NATOM),XYZ(3,NATOM)
LOGICAL NOGRAD
NA = NATOM
NP = NPNTS
NPM1 = NP-1

COMPUTE CHIRALITY CONTRIBUTIONS TO ERROR AND GRADIENT.

ERROR = CHIRER (NA,XYZ,GRD,NOGRAD)

DO 100 I=1,NPM1
  IP1 = I+1
  DO 100 J=IP1,NP

CALCULATE DELTAS ALONG THE GIVEN AXES.

DX = XYZ(1,I) - XYZ(1,J)
DY = XYZ(2,I) - XYZ(2,J)
DZ = XYZ(3,I) - XYZ(3,J)

CALCULATE SQUARED DISTANCES AND BOUNDS.

DSTSQ = DX*DX + DY*DY + DZ*DZ
BUPSQ = BND(I,J)
BLOSQ = BND(J,I)

UPPER BOUND VIOLATION TEST.

IF (DSTSQ.LE.BUPSQ) GO TO 10
  RAWER = (DSTSQ-BUPSQ) / BUPSQ
  CHAIN = 4.0 * RAWER / BUPSQ
  GO TO 20

LOWER BOUND VIOLATION TEST.

10 IF (DSTSQ.GE.BLOSQ) GO TO 100
  RAWER = (DSTSQ-BLOSQ) / BLOSQ
  CHAIN = 4.0 * RAWER / BLOSQ

SUM UP ERRORS OVER ALL DISTINCT PAIRS.

20 ERROR = ERROR + RAWER**2

IF NOGRAD, WE DO NOT COMPUTE THE GRADIENT.

IF (NOGRAD) GO TO 100

CHAIN--COMPOUND FACTOR IS PARTIAL DERIVATIVE.
C
GX = DX * CHAIN
GY = DY * CHAIN
GZ = DZ * CHAIN
C
ADD TO GRADIENT OF NONOUTRIGGERS.
C
IF (I.GT.NA) GO TO 30
GRD(1,I) = GX + GRD(1,I)
GRD(2,I) = GY + GRD(2,I)
GRD(3,I) = GZ + GRD(3,I)
C
SUBTRACT FROM GRADIENT OF NONOUTRIGGERS.
C
30 IF (J.GT.NA) GO TO 100
GRD(1,J) = -GX + GRD(1,J)
GRD(2,J) = -GY + GRD(2,J)
GRD(3,J) = -GZ + GRD(3,J)
C
100 CONTINUE
C
ERRFCT = ERROR
RETURN
END
FUNCTION CHIRER (NATOM,XYZ,GRD,NOGRAD)
C
COMPUTES THE ORIENTED VOLUME OF THE PARALLELEPIPED SPANNED BY THE
VECTORS FROM THE ASYMMETRIC CARBON TO THREE OF ITS LIGANDS, AND
SQUARES THE DIFFERENCE BETWEEN THIS AND THE ORIENTED VOLUME WHEN
THE CHIRALITY AND BOND ANGLES ARE CORRECT. RETURNS THE SUM OF THIS
QUANTITY OVER ALL CHIRAL CENTERS ALONG WITH THE CORRESPONDING
GRADIENT UNLESS NOGRAD.
C
LOGICAL NOGRAD
COMMON /CHIR/ NCHIR,LIG(4,120),VAL(120)
DIMENSION XYZ(3,NATOM) ,GRD(3,NATOM)
DIMENSION ABC(3,3)
NA = NATOM
C
ZERO GRADIENT AND ERROR FOR CHIRER AND ERRFCT.

CHIRER=0.
IF (NOGRAD) GO TO 2
DO 1 I=1,3
DO 1 J=1,NA
GRD(I,J) = 0.
1 CONTINUE
2 IF(NCHIR.LE.0) RETURN

DO 100 M=1,NCHIR
C
COMPUTE MATRIX OF RELATIVE COORDINATES, TAKE ITS DETERMINANT AND
SUM SQUARES OF DIFFERENCES WITH CORRECT VALUES OVER ALL CHIRALS.

DO 10 I=1,3
DO 10 J=1,3
ABC(I,J) = XYZ(I,LIG(J,M)) - XYZ(I,LIG(4,M))
10 CONTINUE
COFT1 = ABC(2,2)*ABC(3,3)-ABC(2,3)*ABC(3,2)
COFT2 = ABC(2,3)*ABC(3,1)-ABC(2,1)*ABC(3,3)
C
DOFT3 = ABC(2,1)*ABC(3,2)-ABC(2,2)*ABC(3,1)
VOL = ABC(1,1)*DOFT1 + ABC(1,2)*DOFT2 + ABC(1,3)*DOFT3
CHIRER = CHIRER + (VOL-VALIM)**2
IF (NOGRAD) GO TO 100

DERIVATIVE OF CHIRER EQUALS 2 X DIFFERENCE X COFACTOR OF MATRIX
ABC WRT ELEMENT CONTAINING COORDINATE X SIGN OF COORDINATE SUMMED.

GP = 2.0 * (VOL-VALIM)
GQ = ABC(2,2)*ABC(3,3)-ABC(3,2)*ABC(2,3)
GRD(1,LIG(1,M)) = GRD(1,LIG(1,M)) + GP*GQ
GRD(1,LIG(4,M)) = GRD(1,LIG(4,M)) - GP*GQ
GQ = ABC(3,2)*ABC(1,3)-ABC(1,2)*ABC(3,3)
GRD(2,LIG(1,M)) = GRD(2,LIG(1,M)) + GP*GQ
GRD(2,LIG(4,M)) = GRD(2,LIG(4,M)) - GP*GQ
GQ = ABC(1,2)*ABC(2,3)-ABC(2,2)*ABC(1,3)
GRD(3,LIG(1,M)) = GRD(3,LIG(1,M)) + GP*GQ
GRD(3,LIG(4,M)) = GRD(3,LIG(4,M)) - GP*GQ
GQ = ABC(1,1)*ABC(3,3)-ABC(3,1)*ABC(1,3)
GRD(1,LIG(2,M)) = GRD(1,LIG(2,M)) + GP*GQ
GRD(1,LIG(4,M)) = GRD(1,LIG(4,M)) - GP*GQ
GQ = ABC(2,1)*ABC(1,3)-ABC(1,1)*ABC(2,3)
GRD(2,LIG(2,M)) = GRD(2,LIG(2,M)) + GP*GQ
GRD(2,LIG(4,M)) = GRD(2,LIG(4,M)) - GP*GQ
GQ = ABC(2,1)*ABC(3,2)-ABC(3,1)*ABC(2,2)
GRD(1,LIG(3,M)) = GRD(1,LIG(3,M)) + GP*GQ
GRD(1,LIG(4,M)) = GRD(1,LIG(4,M)) - GP*GQ
GQ = ABC(1,1)*ABC(2,2)-ABC(2,1)*ABC(1,2)
GRD(3,LIG(3,M)) = GRD(3,LIG(3,M)) + GP*GQ
GRD(3,LIG(4,M)) = GRD(3,LIG(4,M)) - GP*GQ

100 CONTINUE
RETURN
END
INTEGER FUNCTION LOPNUM (LINE, PTR, LN)
INTEGER*4 FUNCTION LOPNUM (LINE, PTR, LN)
IMPLICIT INTEGER (A-Z)
IMPLICIT INTEGER*4 (A-Z)
DIMENSION LINE(LN)
  character*1 line(ln)
  character*1 blank, plus, minus, zero, nine
DATA BLANK, PLUS, MINUS, ZERO, NINE /* ', '+', '-', '0', '9' */
  ichzero = ichar(ZERO)
NUM = 0
SIGN = 1
1 IF (ichar(LINE(PTR)) .NE. ichar(BLANK)) GO TO 2
  PTR = PTR + 1
  IF (PTR .GT. LN) GO TO 10
  GO TO 1
2 IF (ichar(LINE(PTR)) .EQ. ichar(BLANK) .OR. PTR .GT. LN) GO TO 10
  IF (ichar(LINE(PTR)) .NE. ichar(MINUS) .OR. ichar(LINE(PTR)) .NE. ichar(PLUS)) GO TO 3
  SIGN = -SIGN
  PTR = PTR + 1
  GO TO 2
3 IF (ichar(LINE(PTR)) .LT. ichzero .OR. ichar(LINE(PTR)) .GT. ichchar(NINE)) GO TO 4
  num = num * 10 + ichar(line(ptr)) - ichzero
  PTR = PTR + 1
  GO TO 2
4 LOPNUM = 0
RETURN
10 LOPNUM = SIGN * NUM
RETURN
END

FUNCTION XLOP (LINE, IPTR, LN)
RETURN A REAL NUMBER FROM THE INPUT LINE
CHARACTER*(*) LINE(LN)
CHARACTER*(6) BLANK, PLUS, MINUS, ZERO, NINE, DCMP
DATA BLANK, PLUS, MINUS, ZERO, NINE, DCMP /* ' ', '+', '-', '0', '9', '.' */
  ICHZERO = ICHAR(ZERO)
ICHNINE = ICHAR(NINE)
ICHDCMP = ICHAR(DCMP)
ISIGN = 1
IMAG = -1
NUM = 0

1 IF (ICHAR(LINE(IPTR)) .NE. ICHAR(BLANK)) GO TO 2
   IPTR = IPTR+1
   IF (IPTR.GT.LN) GO TO 10
   GO TO 1
2 IF (ICHAR(LINE(IPTR)) .EQ. ICHAR(BLANK) .OR. IPTR.GT.LN) GO TO 10
   IF (ICHAR(LINE(IPTR)) .NE. ICHAR(MINUS) .OR. ICHAR(LINE(IPTR)).EQ. ICHAR(PLUS)) GO TO 3
   ISIGN = -ISIGN
   IPTR = IPTR+1
   GO TO 2
3 IF (ICHAR(LINE(IPTR)) .NE. ICHDCMP GOTO 4
   IMAG = IPTR
   IPTR = IPTR+1
   GOTO 2
4 IF (ICHAR(LINE(IPTR)) .LT. ICHZERO .OR. ICHAR(LINE(IPTR)).GT. ICHNINE) GO TO 5
   NUM = NUM * 10 + ICHAR(LINE(IPTR))-ICHZERO
   IPTR = IPTR+1
   GO TO 2
5 XLOP = 0.0
   RETURN
10 IF (IMAG.LT.1) IMAG = IPTR-1
   XLOP = FLOAT(ISIGN*NUM)/10.0**(IPTR-IMAG-1)
   RETURN
END

CHARACTER*(*) FUNCTION UCASE(STR)
CONVERT A STRING TO UPPER CASE LETTERS, BUT
IGNORE NON-ALPHABETIC CHARACTERS (E.G., 7,&,+)
DECLARE UCASE IN A CHARACTER STATEMENT, I.E.:
EXTERNAL UCASE
CHARACTER UCASE*5
CHARACTER STR*(*) , CH*1

UCASE = ' ' 
ILWA=ICHAR('a')
IUPA=ICHAR('A')
LGTH=LEN(STR)
DO 10 ICH=1, LGTH
   CH=STR(ICH:ICH)
   IF (CH.LT.'a'.OR. CH.GT.'z') GOTO 20
   CH=CHAR(ICHAR(CH) - ILWA+IUPA)
20 UCASE(ICH:ICH)=CH
10 CONTINUE
RETURN
END
PROGRAM BUILD

BUILD IS THE MAIN PROGRAM WHICH CALLS THE EMBED ROUTINE.
PROVIDED BY KUNTZ, HAVEL ET AL. AT UCSF.
WHAT IT DOES, IS JUST INITIALISE SOME PARAMETERS AND CONTROL
THE FLOW OF THE OPERATIONS, BY CALLING THE RIGHT ROUTINE AT
THE RIGHT TIME.
The required input is mainly the XXXSC1.CG communication file
issued by STRCHK. Its output is a set of Cartesian coordinates
to be used by the drawing package and by some other programs
as well (geometry optimisation programs for instance)

COPYRIGHT 1980, BOARD OF TRUSTEES, LEYLAND STANFORD JUNIOR
UNIVERSITY

COMMON BLOCKS

COMMON /XYZS/ NXYZ,XYZ(3,72)
XYZ contains the output coordinates
COMMON /BND/ NPNTS,BND(72,72)
NPNTS contains the number of geometric points = NATOMS for
our current application, and the BND matrix the upper bounds
above the main diagonal and lower bounds below. Note that
these values are returned squared after calling EMBED.
COMMON /EMBD/ DUMP,NDIM,NTRY,SMALL,DWK(72),EWK(72,72)
LOGICAL DUMP
COMMON /ATOMS/ NAT,ICONN(4,72),NCONN(72),NAMES(72),NCNT(72),
1 ITYPE(72)
CHARACTER*4 NAMES
NAT is the number of atoms in the connection table ICONN.
NCONN(I) is the number of entries in I-th row of the CT.
NAMES are atom names, with the name’s character count in NCNT
(usefull in the draw routines). ITYPE contains the expected
atom valences, in a first stage, then the actual number of
neighbors, hydrogens included.
Called BUILD
COMMON /LABEL/ TITLE(16),LTITLE
CHARACTER*5 TITLE
SOME general information: the title and its length.
COMMON /BONDS/ NBND, BNDAT(2,72), BTYPE(72)
INTEGER BNDAT,BTYPE
THE BOND DEFINITIONS
COMMON /STEREO/ ISTER(72)
COMMON /PARA/ GESFCT,STPMIN,STPMAX,CAPPA,DELTA,EPSSLN,
1 FAST,SLOW,ANGMAX,INTMAX,RESTART,ISEED
INTEGER RESTART
COMMON /CTRL/ SLFCOR,IFREQ,ITNLIM,FCTMIN,GRDMIN
LOGICAL SLFCOR
CHARACTER*9 DMPFIL,OUTFIL
CHARACTER*6 IDISP
LOGICAL LPUNCH,LPLOT
THE ABOVE ARE CONTROL PARAMETERS AD USUM EMBED AND SCCG. FOR
AN EXPLANATION, SEE EMBED.

SUGGESTED VALUES FOR PARAMETERS FOR OPTIMIZATION ROUTINE SCCG.

THESE ARE FOR COMMON BLOCK /PARA/
DATA GESFCF/5.E-1/, STPMIN/1.E-3/, STPMAX/1.E00/, CAPPA/5.E-1/,  
1 DELTA/1.E-6/, EPSLN/1.E-5/, FAST/5.E-1/, SLOW/1.E-3/,  
2 ANGMAX/88.00/, INTMAX/5/, RESTAR/100/  

C C TRIAL VALUES FOR COMMON BLOCK CTRL FOR SCCG.  
C C DATA SLFCOR/.TRUE./, IFREQ/1/, ITNLIM/100/, FCTMIN/0.01/,  
1 GRDMIN/0.002/  
C C TRIAL VALUE OF SMALL FOR EMBED - ON ADVICE THAT  
C FCTMIN = SMALL * GRDMIN. ALSO, SET NDIM = 72 AND NTRY = 10  
C C DATA SMALL/5.0/  
C C DATA OUTFIL /'00XYZ.MOL'/, DMPFIL '/000DMP.CG'/  
NDIM = 72  
NTRY = 10  
NSTRUC = 1  
1020 FORMAT (A10)  
DUMP = .TRUE.  
IF (DUMP) OPEN (UNIT=20,FILE=DMPFIL,STATUS=’NEW’)  
LPUNCH = .TRUE.  
OPEN (UNIT=21,FILE=OUTFIL,STATUS=’NEW’)  
1050 FORMAT (I)  
C C  
10 CALL PREL1(DUMP,NDIM,IERR)  
IF (IERR.GT.0) GO TO 9000  
NXXYZ = NAT  
NPNTS = NAT  
CALL PREL2(DUMP,NDIM)  
C C LOOP NSTRUC TIMES, TRYING TO GET THIS MUCH DIFFERENT STRUCTURES  
C  
KSTRUC = 0  
500 KSTRUC = KSTRUC + 1  
IF (KSTRUC.EQ.1) GOTO 501  
GOTO 502  
501 CALL EMBED (ISSUE)  
GO TO 503  
502 CALL EMBED2(ISSUE)  
503 IF (ISSUE) 520,530,540  
C C FAILURE EXIT  
C  
520 WRITE (6,5020) KSTRUC  
5020 FORMAT (’ EMBED FAILED AT KSTRUC =’,I3)  
GO TO 541  
C C EMBEDDING INDETERMINATE. WRITE OUT THE RESULT, AND  
C KEEP PUMPING...  
C  
silence for web version tws 8/14/99  
C 530 WRITE (6,5030) KSTRUC  
530 CONTINUE  
5030 FORMAT (’ EMBED SITS ON THE FENCE AT KSTRUC =’,I3)  
GO TO 541  
C C EMBEDDING SUCCESSFUL  

c 540 WRITE (6,5040) KSTRUC
540 CONTINUE
5040 FORMAT (' SUCCESS AT KSTRUC = ',I3,$)
c WRITE (6,5041)
5041 FORMAT ('+ LOOK AT THIS !!')
541  CALL MOLFILE()
     IF (KSTRUC.EQ.NSTRUC) GO TO 9000
     IF (KSTRUC.LT.NSTRUC) GO TO 500
C
9000  IF (DUMP) CLOSE(UNIT=20)
     IDISP = 'SAVE'
     IF (.NOT.LPUNCH) IDISP = 'DELETE'
     CLOSE (UNIT=21,DISPOSE=IDISP)
END
C
C
SUBROUTINE MOLFILE()
C WRITE OUT AN MOL MOLFILE FOR THE FINAL COORDINATES
C TWShattuck Colby 7/14/99
C COMMON BLOCKS
    COMMON /XYZS/ NXYZ, XYZ(3,72)
C XYZ CONTAINS THE OUTPUT COORDINATES
    COMMON /ATOMS/ NAT, ICONN(4,72), NCONN(72), NAMES(72), NCNT(72),
             1 ITYPE(72)
    COMMON /STEREO/ ISTER(72)
C NAT IS THE NUMBER OF ATOMS IN THE CONNECTION TABLE ICONN.
C NCONN(I) IS THE NUMBER OF ENTRIES IN I-TH ROW OF THE CT.
C NAMES ARE ATOM NAMES, WITH THE NAME'S CHARACTER COUNT IN NCNT
    COMMON /LABEL/ TITLE(16), LTITLE
C SOME GENERAL INFORMATION: THE TITLE AND ITS LENGTH.
    COMMON /BONDS/ NBNDs, BNDAT(2,72), BTYPE(72)
    INTEGER BNDAT, BTYPE, ICH(72), IIST(72)
    CHARACTER TITLE*5, NAMES*4, EL*4
C THE BOND DEFINITIONS
C C STRIP OUT CHARGE AND SET STEREO FLAG
DO 5 I = 1, NXYZ
    IF ( ISTER(I) .EQ. -1 ) IST(I) = 0
C PREL INVERTS THE STEREO FLAG
    IF ( ISTER(I) .EQ. 0 ) IST(I) = 2
    IF ( ISTER(I) .EQ. 1 ) IST(I) = 1
    ICH(I) = 0
    ICHARG = 0
    EL = NAMES(I)
DO 10 J = 4, 2, -1
    IF ( EL(J:J) .NE. '+' ) GOTO 15
10 CONTINUE
15 DO 20 J = J, 1, -1
    IF ( EL(J:J) .EQ. '-' ) ICHARG = ICHARG + 1
    IF ( EL(J:J) .EQ. '+' ) ICHARG = ICHARG - 1
    IF ( (EL(J:J) .EQ. '+' ) .AND. (EL(J:J) .NE. '-' ) ) GOTO 25
20 CONTINUE
25 IF ( ICHARG .EQ. 0 ) GOTO 5
    IF ( ICHARG .EQ. -3 ) ICH(I) = 7
    IF ( ICHARG .EQ. -2 ) ICH(I) = 6
    IF ( ICHARG .EQ. -1 ) ICH(I) = 5
    IF ( ICHARG .EQ.  1 ) ICH(I) = 3
    IF ( ICHARG .EQ.  2 ) ICH(I) = 2
    IF ( ICHARG .EQ.  3 ) ICH(I) = 1
    NAMES(I) = EL(1:J)
5 CONTINUE
500 WRITE (21,2000) (TITLE(I), I=1, LTITLE)
2000 FORMAT (16A5)
    WRITE(21,2001)
2001 FORMAT ("-ISIS-", 3D/)
    WRITE (21,2010) NAT, NBNDs
2010 FORMAT (2I3, ' 0 0 0 0 0 0 0 0 0')
    WRITE(21,3000) ((XYZ(I,J), I=1,3), NAMES(J), ICH(J),
             1 IIST(J), J=1,NXYZ)
3000 FORMAT (3F10.4,X, A3, ' 0', 2I3, ' 0 0')
NOW PRINT BONDS. BNDAT(1,I) AND BNDAT(2,I) ARE THE TWO ENDS OF
BOND I. BTYPE(I) IS THE MULTIPLICITY
DO 250 IBNDs = 1, NBNDs
C
WRITE(21,1234) BNDAT(1,IBNDS), BNDAT(2,IBNDS), BTYPE(1BNDS)
1234 FORMAT(3I3,' 0 0 0')
250 CONTINUE
WRITE(21,3030)
3030 FORMAT('M END')
C
  RETURN
END
integer function nxm(setA, istart)
c  find the next bit to the left of bit ibit that is 1
c  The right-most bit is bit 1
integer*8 setA
c
ibytes=8
ibits=8*ibytes
do 10 itst=istart, ibits-1
  if ( bktest(setA, itst) ) goto 200
10 continue
itst=-1
200 nxm = itst+1
return
end
SUBROUTINE PREL1 (DUMP, NDIM, IERR)
C PRELIMINARY OPERATIONS, BEFORE CALLING EMBED.
C PERFORMS ALL INPUT OPERATIONS ON XXXSC1.CG FILE
C CHECK AND RESET STEREOFLAGS ACCORDING TO THE NEW NUMBERING
C DEFINE ATOM TYPES, NUMBER OF NEIGHBORS
C DEFINE ALL 1,2 BONDS

COMMON /ATOMS/ NAT, ICONN(4,72), NCONN(72), NAMES(72), NCNT(72),
1   ITYPE(72)
COMMON /LABEL/ TITLE(16), LTITLE
COMMON /ATSPEC/ NSPEC, ANAM(10), AVAL(10), IOLE(72), INEW(72)
COMMON /JUNK/ JJUNK, IADDH(72)
COMMON /PARA/ GESFC, STPMIN, STPMAX, CAPPA, DELTA, EPSLN,
1   FAST, SLOW, ANGMAX, INTMAX, RESTAR, ISEECHAND
COMMON /BONDS/ NBND, BNDAT(2,72), BTYPE(72)
INTEGER BNDAT,BTYPE
COMMON /CSTR/ NCSTR, IATCTR(10), JATCTR(10), CSTRMN(10), CSTRMX(10)
INTEGER AVAL
CHARACTER*4 ANAM, NAMES, LOPWRD, ITEMPC
CHARACTER*5 TITLE, BLANK
CHARACTER*16 FILNAM
EXTERNAL UCASE
CHARACTER*4 UCASE
EQUIVALENCE (IFN,FILNAM)
CHARACTER*1 LINE(160)
COMMON /STEREO/ ISTER(72)
INTEGER AFT(4), BEF(4)
LOGICAL DUMP, WORRY, DBOND, UNFLIP
DATA FILNAM / '000SC1.CG', BLANK, / '
DATA MAXLN /160/, MAXAT, MAXBND /72,72/

CALL ZEROM(ICONN, 8*NDIM)
NAT = 0
CALL ZEROM(BNDAT, 3*NDIM)
NBND = 0
CALL ZEROM(ISTER, NDIM)
CALL ZEROM(TITLE, 16)
LTITLE = 0
IERR = 0

C 1) SETUP FILE NAME AND OPEN
C     CALL MAXFIL (FILNAM)
OPEN(UNIT=1, FILE=FILNAM, STATUS='OLD')
C 2) READ NUMBER OF ATOM SPECS AND RANDOM NUMBER SEED
READ(1,1000) (LINE(I), I=1,160)
1000 FORMAT (160A1)
IPNT = 1
NSPEC = LOPNUM(LINE, IPNT, MAXLN)
C 3) READ ATOM DEFINITIONS (NAMES AND USUAL VALENCE)
READ(1,1000) (LINE(I), I=1,160)
IPNT = 1
DO 10 I = 1, NSPEC
  ANAM(I) = UCASE(LOPWRD(LINE, IPNT, MAXLN, ICNT))
  AVAL(I) = LOPNUM(LINE, IPNT, MAXLN)
10 CONTINUE
READ(1,1000) (LINE(I), I=1,160)
IPNT = 1
ISEED = LOPNUM(LINE,IPNT,MAXLN)

C 4) READ NUMBER OF ATOMS
100 READ(1,1000) (LINE(I),I=1,160)
   IPNT = 1
   NAT = LOPNUM(LINE,IPNT,MAXLN)
   IF (NAT.LE.0) GO TO 8300
   IF (NAT.GT.MAXAT) GO TO 8000

C 5) READ CONNECTION TABLE
   DO 30 I = 1,NAT
      DO 1020 J = 1,4
         IC = LOPNUM(LINE,IPNT,MAXLN)
         IF (IC.LE.O) GO TO 25
         NCONNI = NCONNI + 1
         ICONN(J,I) = IC
      1020 CONTINUE
      DO 22 J = 1,NSPEC
         VAL = AVAL(J)
         IF (NAMES(I).EQ.ANAM(J)) GO TO 23
      22 CONTINUE
      C ASSUME A DEFAULT VALENCE OF 4 FOR UNIDENTIFIED ATOMS
      C (THOU SHALT NOT HAVE UNIDENTIFIED ATOMS !!!!)
      VAL = 4
      23 ITYPE(I) = VAL
      NCONNI = 0
      DO 24 J = 1,4
         IC = LOPNUM(LINE,IPNT,MAXLN)
         IF (IC.LE.O) GO TO 25
         NCONNI = NCONNI + 1
         ICONN(J,I) = IC
      24 CONTINUE
      25 CONTINUE
      C MAKE SURE THAT THE NEIGHBORS OF I ARE LISTED IN ASCENDING ORDER
      NCM1 = NCONNI - 1
      IF (NCM1.LE.0) GO TO 29
      DO 27 J = 1,NCM1
         JJ = J + 1
         DO 26 K = JJ,NCONNI
            IF (ICONN(K,I).GE.ICONN(J,I)) GO TO 26
               ITEMP = ICONN(K,I)
               ICONN(K,I) = ICONN(J,I)
               ICONN(J,I) = ITEMP
            26 CONTINUE
      27 CONTINUE
      29 CONTINUE
      C NCONNI(I) IS THE NUMBER OF CONNECTIONS TO I
      NCONNI(I) = NCONNI
      C TO MAKE A STEREOCENTER, ONE MUST HAVE AT LEAST 3 NEIGHBORS ON A
      C CARBON ATOM, AND 3 ON NITROGEN AND PHOSPHORUS, TOO.
      C (NCONNI.GE.3) GO TO 30
      ISTER(I) = -1
      30 CONTINUE

   READ(1,1100) (TITLE(I),I=1,16)
1100 FORMAT (16A5)
C FIND LENGTH OF TITLE LINE (IN WORDS)
     LTITLE = 0
     DO 35 I = 1,16
        IF (TITLE(I) .EQ. BLANK) GO TO 36
        LTITLE = LTITLE + 1
     35 CONTINUE
     36 CONTINUE
C 6)  READ RENUMBERING TABLE
     DO 40 I = 1,NAT
        READ(1,1000) (LINE(J),J=1,160)
        IPNT = 1
        INEW(I) = LOPNUM(LINE,IPNT,MAXLN)
        IOLD(I) = LOPNUM(LINE,IPNT,MAXLN)
     40 CONTINUE
C TEST IF NEW NUMBERING GOES IN THE SAME SENSE AS THE OLD ONE. IF
C THIS IS THE CASE, SKIP THE NEXT SECTION.
     UNFLIP = ((IOLD(1) .LT. IOLD(NAT) .AND. INEW(1) .LT. INEW(NAT)) .OR.
                (IOLD(1) .GT. IOLD(NAT) .AND. INEW(1) .GT. INEW(NAT)))
     IF (UNFLIP) GO TO 175
C FLIP ALL PERTINENT ARRAYS IF RENUMBERING GOES BACKWARDS.
     NAT2 = NAT/2
     DO 120 I1 = 1,NAT2
         I2 = NAT+1 - I1
         ITEMPC = NAMES(I1)
         NAMES(I1) = NAMES(I2)
         NAMES(I2) = ITEMPC
         ITYPE(I1) = ITYPE(I2)
         ITYPE(I2) = ITYPE(I1)
         IADDH(I1) = IADDH(I2)
         IADDH(I2) = IADDH(I1)
         NCNT(I1) = NCNT(I2)
         NCNT(I2) = NCNT(I1)
         NCONN(I1) = NCONN(I2)
         NCONN(I2) = NCONN(I1)
         AFT(J) = ICONN(J,I1)
         110 CONTINUE
         120 CONTINUE
     DO 170 J = 1,4
         ICONN(J,I) = ICONN(J,I2)
     130 CONTINUE
     DO 140 J = 1,LIM
        JJ = LIM+1 - J
        ICONN(JJ,I) = NAT+1 - AFT(J)
     140 CONTINUE
     170 CONTINUE
READ EXTERNAL DISTANCE CONSTRAINTS
FOR NCSTR TOTAL CONSTRAINTS WITH ATOM(I)=IATCTR
AND ATOM(J)=JATCTR WITH CONSTRAINT MINIMUM CSTRMN
AND MAXIMUM CSTMX
175 READ(1,1000) (LINE(J),J=1,160)
   IPNT = 1
   NCSTR = LOPNUM(LINE,IPNT,MAXLN)
DO 50 ICSTR = 1,NCSTR
   READ(1,1000) (LINE(J),J=1,160)
   IPNT = 1
   IATCTR(ICSTR) = LOPNUM(LINE,IPNT,MAXLN)
   JATCTR(ICSTR) = LOPNUM(LINE,IPNT,MAXLN)
   CSTRMN(ICSTR) = XLOP(LINE,IPNT,MAXLN)
   CSTRMX(ICSTR) = XLOP(LINE,IPNT,MAXLN)
C MAKE SURE THAT JATCTR>IATCTR
   IF ( JATCTR(ICSTR) .GT. IATCTR(ICSTR) ) GOTO 50
      ITEMP = IATCTR(ICSTR)
      IATCTR(ICSTR) = JATCTR(ICSTR)
      JATCTR(ICSTR) = ITEMP
50 CONTINUE
C COMPLETE THE CONNECTION TABLE WITH HYDROGENS IF IADDH=0
C ORIGINAL VERSION ONLY ADDED H ON THE CHIRAL
C CENTERS AND ACHIRAL CENTERS WHOSE STEREOFLAG IS NOT -1 WILL BE
C COMPLETED TOO.
C
   NHYD = 0
DO 190 I = 1,NAT
   IF (IADDH(I) .NE.0) GO TO 190
      LIM = NCONN(I) + 1
      IVAL = ITYPE(I)
      IF (LIM.GT.IVAL) GO TO 190
   DO 180 J = LIM,IVAL
      NHYD = NHYD + 1
      INDXH = NAT + NHYD
      IF (INDXH.GT.MAXAT) GO TO 8100
      ICONN(J,I) = INDXH
      ICONN(1,INDXH) = I
      NCONN(I) = NCONN(I) + 1
      NAMES(INDXH) = 'H'
      NCNT(INDXH) = 1
      ISTER(INDXH) = -1
      IADDH(INDXH) = 1
      ITYPE(INDXH) = 1
180 CONTINUE
190 CONTINUE
   ONAT = NAT
   NAT = NAT + NHYD
C INVERT STEREOFLAGS ...
DO 195 I = 1,ONAT
   IF (ISTER(I) .LT.0) GO TO 195
C ... IF HYDROGEN ATOM HAS BEEN ADDED ON A TETRAVALENT ATOM...
      IF (ICONN(4,I).GT.ONAT) ISTER(I) = 1 - ISTER(I)
C ... OR IF TERTIARY TRIVALENT (N,P)
      IF (ITYPE(I).NE.3) GO TO 195
      IF (NAMES(I).EQ.'N'.OR.NAMES(I).EQ.'P') ISTER(I) = 1 - ISTER(I)
195 CONTINUE
C IF (.NOT.DUMP) GO TO 200
WRITE (20,2000) (TITLE(I),I=1,LTITLE)
2000 FORMAT (1H1,120(LH-)//' DATA HAS BEEN READ FOR ',16A5)
WRITE (20,2010) NAT
2010 FORMAT (' THERE ARE ',I2,' ATOMS IN THE MOLECULE'/
1 ' THE MODIFIED CONNECTION TABLE IS:')
   DO 198 I = 1,NAT
      LIM = NCONN(I)
      WRITE (20,2020) I,NAMES(I),ISTER(I),(ICONN(J, I) ,J=1,LIM)
198 CONTINUE
2020 FORMAT (2X,I3,2x,A4,2X,SI4)
      WRITE(20,2030)
2030 FORMAT(//' EXTERNAL CONSTRAINT MINIMUMS AND MAXIMUMS:')
      DO 55 1= 1,NCSTR
         WRITE(20,2035) IATCTR(I) ,JATCTR(I) ,CSTRMN(Il ,CSTRMX(I)
55 CONTINUE
2035 FORMAT (2X,I3,"-",I3,";",2FlO.4l
C C NOW DEFINE BONDS. BNDAT(1, I) AND BNDAT(2, I) ARE THE TWO ENDS OF
C BOND I. BTYPE(I) IS THE MULTIPLICITY
C C 200 NBNDS = 0
   DO 210 I = 1,MAXBND
      BNDAT(1, I) = 0
      BNDAT(2, I) = 0
      BTYPE(I) = 0
210 CONTINUE
   DO 250 J = 1,NAT
      LIM = NCONN(J)
      IPREV :: 0
      DO 240 K :: I,LIM
         IKJ = ICONN(K,J)
         IF (IKJ.LT.J) GO TO 240
         IF (IKJ.EQ.IPREV) GO TO 230
         NBNDS = NBNDS + 1
         IF (NBNDS.GT.MAXBND) GO TO 8200
         BNDAT(1,NBNDS) = J
         BNDAT(2,NBNDS) = IKJ
         BTYPE(NBNDS) = 1
         IPREV = IKJ
9
230 BTYPE(NBNDS) = BTYPE(NBNDS) + 1
C FOR EACH MULTIPLE BOND, DECREASE ITYPE. SO THAT ITYPE BECOMES
C THE NUMBER OF ATOMS CONNECTED TO J AND IKJ, HYDROGENS INCLUDED
   ITYPE(J) = ITYPE(J) - 1
   ITYPE(IKJ) = ITYPE(IKJ) - 1
240 CONTINUE
250 CONTINUE
C C CLOSE(UNIT=1)
RETURN
C
8000 IERR = 1
WRITE(6,18000)
18000 FORMAT (' SIZE OF MOLECULE EXCEEDS STORAGE CAPACITY'/)
   GO TO 9999
8100 IERR = 2
WRITE(6,18100)
18100 FORMAT (' STORAGE OVERFLOW DURING HYDROGEN ADDITION'/)
   GO TO 9999
8200 IERR = 3
WRITE(6,18200)
C
9999 CLOSE(UNIT=1)
   RETURN
END
SUBROUTINE PREL2(DUMP, NDIM)

DEFINE CHIRAL CENTERS
SET-UP BND ARRAY WITH 1-2, 1-3 AND 1-4 DISTANCES

COMMON /ATOMS/ NAT, ICONN(4,72), NCONN(72), NAMES(72), NCNT(72),
1 1 ITYPE(72)
CHARACTER*4 NAMES, NAME1, NAME2, CTEMP
COMMON /CHIR/ NCCHIR, LIG(4,120), VAL(120)
COMMON /Bonds/ NBND, BNDAT(2,72), BTYPE(72)
INTEGER BNDAT, BTYPE
COMMON /CSTR/ NCSTR, IATCTR(10), JATCTR(10), CSTRMN(10), CSTRMX(10)
COMMON /STEREO/ ISTER(72)
DIMENSION BANGL(72)
LOGICAL DUMP, WORRY, DBOND
COMMON /NPTS/ BND(72,72)
DIMENSION RSIZE(20), RNGAT(10,20), ISINRG(72)
INTEGER RSIZE, RNGAT
C
DATA TABLE FOR BOND LENGTHS IN THE FORM Lmn(I,J) WHERE L IS
S, D OR T FOR SINGLE, DOUBLE TRIPLE BOND; MN IS ATOM NAME M
AND N (LIMITED TO C, N AND O) AND I, J ARE THE "CONNECTIVI-
TIES", OR THE NUMBER OF ATTACHED ATOMS, RESPECTIVELY. THUS
SCC(4,4) IS TWO SINGLY CONNECTED SP3 CARBONS, WHILE TCN(2,1)
IS CN TRIPLE BOND.
BOND COUPLES WHICH ARE INDETERMINATE, E.G., CC(1,1) ARE
SET TO 0.0

DIMENSION SCC(4,4), SCO(4,2), SCN(4,3), SNN(3,3), SOP(2,3),
1 SNO(3,2), DCC(3,3), DCO(3,1), DCM(3,2), DCP(3,2),
2 DNO(2,1), DNN(2,2), DCS(3)
DIMENSION SHC(4), SHN(3), SHP(3), SHSI(4)
DIMENSION SCF(4), SCCL(4), SCBR(4), SCI(4)
DIMENSION SCP(4,3), SC(4,2), SCI(4)
DIMENSION CBANGL(4), NBANGL(3), OBANGL(2)
DIMENSION SIBANGL(4), PIBANGL(3), SIBANGL(2)
REAL NBANGL, OBANGL
CHARACTER*2 HYDR, CARB, NITR, OXYG, SILL, PHOS, SULF, NAME
EQUIVALENCE (HYDR, MENDEL(1))
EQUIVALENCE (CARB, MENDEL(6)), (NITR, MENDEL(7)), (OXYG, MENDEL(8))
EQUIVALENCE (SILI, MENDEL(14)), (PHOS, MENDEL(15)), (SULF, MENDEL(16))

DIMENSION NNAT(72)
CHARACTER*2 MENDEL /54/ DATA NMEND /54/,
1 DATA MENDEL /'H ', ' ', ' ', 'C ', 'N ', 'O ', 'F ', ' ',
2 /' ', ' ', 'SI', 'P ', 'S ', 'CL', ' ',
3 16/*', ' ', 'BR', ' ',
4 16/*', 'I ', '/

SINGLE BONDS

DATA (SCC(1,1), I=1,4)/0.0, 0.0, 0.0, 0.0/
DATA (SCC(1,1), I=1,4)/0.0, 0.0, 0.0, 0.0/
DATA (SCC(1,1), I=2,4)/ 0.0, 0.0, 0.0/
DATA  SCC(2,2)/1.38/, SCC(3,2)/1.45/, SCC(4,2)/1.46/
DATA  SCC(2,3)/1.45/, SCC(3,3)/1.46/, SCC(4,3)/1.52/
DATA  SCC(2,4)/1.46/, SCC(3,4)/1.52/, SCC(4,4)/1.54/
DATA (SCO(1,1), I=1,2) /0.0, 0.0/ 
      C       c       6/9/99
DATA (SCO(I,1), I=1,4) /0.0, 0.0, 0.0, 0.0/
DATA (SCO(I,1), I=2,4) /0.0, 0.0, 0.0, 0.0/
DATA  SCO(2,2)/1.36/, SCO(3,2)/1.36/, SCO(4,2)/1.43/
DATA (SCN(1,1), I=1,3) /0.0, 0.0, 0.0/ 
      C       c       6/9/99
DATA (SCN(I,1), I=1,4) /0.0, 0.0, 0.0, 0.0/
DATA (SCN(I,1), I=2,4) /0.0, 0.0, 0.0, 0.0/
DATA  SCN(2,2)/1.33/, SCN(3,2)/1.33/, SCN(4,2)/1.36/
DATA (SCN(1,1), I=1,3) /0.0, 0.0, 0.0/ 
      C       c       6/9/99
DATA (SCN(I,1), I=1,3) /0.0, 0.0, 0.0, 0.0/
DATA (SCN(I,1), I=2,4) /0.0, 0.0, 0.0, 0.0/
DATA  SCN(2,2)/1.36/, SCN(3,2)/1.36/, SCN(4,2)/1.47/
DATA (SCN(1,1), I=1,3) /0.0, 0.0, 0.0/ 
      C       c       6/9/99
DATA (SCN(I,1), I=1,3) /0.0, 0.0, 0.0, 0.0/
DATA (SCN(I,1), I=2,4) /0.0, 0.0, 0.0, 0.0/
DATA  SCN(2,2)/1.33/, SCN(3,2)/1.33/, SCN(4,2)/1.36/
DATA  SCF /1.38/, SOS /1.76/, SSS /2.04/
DATA  SCF /0.0, 0.0, 1.33, 1.38/
DATA  SCCL /0.0, 1.64, 1.71, 1.77/
DATA  SCBR /0.0, 1.79, 1.87, 1.94/
DATA  SCI /0.0, 1.99, 2.07, 2.17/
DATA  SCP /10*0.0, 1.82, 1.84/
DATA  SCF /6*0.0, 1.72, 1.82/
DATA  SOSI /1.61/, SNSI /1.74/, SPSI /1.58/, SSICL /2.62/
DATA  SMC /0.0, 1.06, 1.08, 1.09/
DATA  SHN /0.0, 0.99, 1.01/
DATA  SHO /0.96/.SHS /1.33/
DATA  SHP /0.0, 0.0, 1.44/
DATA  SHSI /0.0, 0.0, 0.0, 1.48/

C       C SPECIAL BOND DISTANCE FOR AMIDE C-N BOND

C       C DOUBLE BONDS

DATA (DCC(1,1), I=1,3) /0.0, 0.0, 0.0, 0.0/ 
      C       c       6/9/99
DATA (DCC(I,1), I=1,3) /0.0, 0.0, 0.0, 0.0/
DATA (DCC(I,1), I=2,3) /0.0, 0.0, 0.0/
DATA  DCC(2,2)/1.28/, DCC(3,2)/1.31/
DATA  DCC(2,3)/1.31/, DCC(3,3)/1.34/
DATA  DCO(1,1)/0.0/, DCO(2,1)/1.16/, DCO(3,1)/1.22/
DATA  DCN(1,1)/0.0/, DCN(2,1)/0.0/, DCN(3,1)/0.0/
DATA  DCN(2,2)/0.0/, DCN(2,2)/1.32/, DCN(2,2)/1.32/
DATA  DNN(1,1)/0.0/, DNN(2,1)/0.0/
DATA  DNN(1,2)/0.0/, DNN(2,2)/1.25/
DATA  DNO(1,1)/0.0/, DNO(2,1)/1.22/
DATA  DCS /0.0, 1.56, 1.71/
DATA  DCP /6*0.0/

C       C TRIPLE BONDS

DATA  TCC /1.20/, TCN /1.16/
NOW FOR BOND ANGLES. FOR NOW ASSUME STANDARD BOND ANGLES
DEPENDENT ONLY ON THE HYBRIDIZATION OF THE CENTRAL ATOM.
IN THE FOLLOWING DATA TABLE, THE INDEX USED REFLECTS THE
NUMBER OF NEIGHBORS AND THUS THE HYBRIDIZATION...

```
DATA (CBANGL(I), I=1,4)/0.0, 180.0, 120.0, 109.47/
DATA (NBANGL(I), I=1,3)/180.0, 120.0, 109.47/
DATA (OBANGL(I), I=1,2)/180.0, 120.0/
DATA (SIBANGL(I), I=1,4) /0.0, 180.0, 120.0, 109.47/
DATA (PBANGL(I), I=1,3) /180.0, 120.0, 109.47/
DATA (SBANGL(I), I=1,2) /180.0, 120.0/
```

CALL ZEROM(BND,NDIM*NDIM)
CALL ZEROM(LIG,600)
CALL ZEROM(ISINRG,NDIM)

DEFINE ATOMIC NUMBERS

```
DO 220 I = 1,NAT
  NNAT(I) = 0
DO 205 J = 1,NMEND
  JJ = J
  NAME = NAMES(I)(1:2)
  IF (NAME.EQ.MENDEL(J)) GO TO 210
  IF ((NAME(2:2).EQ."+").AND.(NAME(1:1).EQ.MENDEL(J)(1:1))) GO TO 210
  IF ((NAME(2:2).EQ."-").AND.(NAME(1:1).EQ.MENDEL(J)(1:1))) GO TO 210
205  CONTINUE
210  NNAT(I) = JJ
220  CONTINUE
```

DEFINE CHIRAL CENTERS

```
IF (DUMP) WRITE(20,4010)
4010  FORMAT (' CHIRAL CENTERS DEFINITIONS: ')
  NCHIR = 0
DO 490 I = 1,NAT
  IF (ISTER(I).LT.0.OR.ITYPE(I).LE.2) GO TO 490
  IF (NAMES(I)(1:1).EQ.'C'.AND.ITYPE(I).EQ.1) GO TO 450
  IF ((NAMES(I)(1:1).EQ.'N'.OR.NAMES(I)(1:1).EQ.'P').AND.
      ITYPE(I).LT.3) GO TO 490
  1  NCHIR = NCHIR + 1
    LIG(4,NCHIR) = I
  NLIG = 0
  DO 410 ILIG = 1,4
    IF (ILIG.EQ.IP) GO TO 410
    NLIG = NLIG + 1
    LIGINLIG,NCHIR) = ICONN(ILIG,I)
410  CONTINUE
  IF ((IP/2)*2.EQ.IP) GO TO 420
  ITEMP = LIG(2,NCHIR)
  LIG(2,NCHIR) = LIG(3,NCHIR)
  LIG(3,NCHIR) = ITEMP
```

A) SP3 CARBON: PUT NEIGHBORS NUMBERS IN LIG(1-4,N) IN THE
APPROPRIATE ORDER.

SP3 N AND P: THE THREE NEIBOURS NUMBERS IN LIG(1-3,N) AND
THE ATOM NUMBER IN LIG(4,N)

```
DO 430 IP = 1,4
  NCHIR = NCHIR + 1
  LIG(4,NCHIR) = I
  NLIG = 0
  DO 410 ILIG = 1,4
    IF (ILIG.EQ.IP) GO TO 410
    NLIG = NLIG + 1
    LIG(NLIG,NCHIR) = ICONN(ILIG,I)
410  CONTINUE
  IF ((IP/2)*2.EQ.IP) GO TO 420
  ITEMP = LIG(2,NCHIR)
  LIG(2,NCHIR) = LIG(3,NCHIR)
  LIG(3,NCHIR) = ITEMP
```
420 IF (ISTER(I).EQ.0) GO TO 430
   ITEM = LIG(1,NCHIR)
   LIG(1,NCHIR) = LIG(2,NCHIR)
   LIG(2,NCHIR) = ITEM
430 IF (DUMP) WRITE(20,4022) NCHIR,NAMES(I),(LIG(J,NCHIR),J=1,4)
   GO TO 490
450 CONTINUE
C B) SP2 CARBON ATOMS: CONSIDER THEM AS STEREOCENTERS, IT WILL
C HELP KEEP THEM FLAT DURING THE EMBED ITERATION. IN THIS
C CASE THE ORDERING OF THE NEIGHBORS IS NOT IMPORTANT, BECAUSE
C THE SPANNED VOLUME WILL BE ZERO ANYWAY.
   LIM = NCONN(I)
   IF (LIM.LE.3) GO TO 490
   NCHIR = NCHIR + 1
   LIG(4,NCHIR) = I
   IPREV = 0
   NLIG = 0
   DO 460 J = 1,LIM
      NEXT = ICONN(J,I)
      IF (NEXT.EQ.IPRED) GO TO 460
      NLIG = NLIG + 1
      LIG(NLIG,NCHIR) = NEXT
      IPRED = NEXT
460 CONTINUE
   IF (DUMP) WRITE(20,4021)
490 CONTINUE
4021 FORMAT (IS, IX, 'DOUBLE BOND', 1X, A4, 3H - , 4I4)
4022 FORMAT (IS, IX, 'TETRAHEDRON', 1X, A4, JH - , 4I4)
C C FILL IN ALL ROWS AND COLUMNS OF BND ARRAY WITH DSTMAX
C (ESTIMATED FROM FORMULA DUE TO HAVEL/KUNTZ - UCSF)
C FOR UPPER HALF AND 2.0 ANGSTROMS FOR LOWER HALF
C THIS WILL HELP TO CHECK PROGRESS OF FILLING IN REAL
C DISTANCES AND PROVIDE MAX AND MIN VALUES FOR USE BY
C EMBED. SET DIAGONAL ELEMENTS TO -1.0
C
   NPTS = NAT
   DSTMAX = 10.0 - FLOAT(NAT)**(1.0/3.)
   DSTMIN = 2.0
   DO 500 I = 1,NAT-1
      BND(I,I) = -1.0
   DO 500 J = I+1, NAT
      BND(I,J) = DSTMAX
      BND(J,I) = DSTMIN
500 CONTINUE
   BND(NAT,NAT) = -1.0
C C NOW, BEGIN FILING THE DIST MATRIX AS NEEDED BY EMBED.
C PROCEED AS FOLLOW: FIRST LOOP THROUGH THE BOND LIST TO
C SET UP DISTANCES BETWEEN DIRECTLY CONNECTED ATOMS.
C SECOND, LOOP THROUGH TRIPLET DEFINITIONS TO DEFINE ALL
C 1,3 DISTANCES, USING ALREADY DEFINED 1,2 DISTANCES AND
C THE LAW OF COSINES.
C FOR 1-4 DISTANCES, COMPUTE MIN AND MAX DISTANCES BASED ON A
C TORSION ANGLE OF 0 AND 180 DEGREES RESPECTIVELY. IF ATOMS 1
C AND 4 ARE ON A DOUBLE BOND, DMIN = DMAX, COMPUTED ACCORDING
C TO THE CONFIGURATION (CIS OR TRANS) OF THE DOUBLE BOND.
C PASS 1: SET ALL NEIGHBORING DISTANCES SYMMETRICALLY IN BND
IF (DUMP) WRITE(20,5010)
5010 FORMAT ('// PASS 1: SET 1,2 DISTANCES.')
1 ' NAMEI,TYEPI,I.,NAMEJ,TYPEJ,J,NBORD,DISTANCE ARE:'
DO 599 I = 1,NBNDs
NDX1 = BNDAT(I,1)
NDX2 = BNDAT(I,2)
BND(NDX1,NDX1) = 0.0
BND(NDX2,NDX2) = 0.0
ITYPE1 = ITYPE(NDX1)
ITYPE2 = ITYPE(NDX2)
NAME1 = NAMES(NDX1)
NAME2 = NAMES(NDX2)
C MAKE SURE ATOMS ARE IN CORRECT ORDERING
NN1 = NNAT(NDX1)
NN2 = NNAT(NDX2)
IF (NN1.EQ.0.OR.NN2.EQ.0) GO TO 598
IF (NN1.LE.NN2) GO TO 530
CTEMP = NAME1
NAME1 = NAME2
NAME2 = CTEMP
ITEMP = NDX1
NDX1 = NDX2
NDX2 =ITEMP
ITEMP = ITYPE1
ITYPE1 = ITYPE2
ITYPE2 = ITEMP
ITEMP = NN1
NN1 = NN2
NN2 = ITEMP
530 CONTINUE
NBORD = BTYPE(I)
DIST = 0
10000 IF (NBORD.NE.1) GO TO 20000
10100 IF (NN1.NE.1) GO TO 10600
10106 IF (NN2.NE.6) GO TO 10107
10107 IF (NN2.NE.07) GO TO 10108
10108 IF (NN2.NE.08) GO TO 10114
10114 IF (NN2.NE.14) GO TO 10115
10115 IF (NN2.NE.15) GO TO 10116
10116 IF (NN2.NE.16) GO TO 598
10600 IF (NN1.NE.06) GO TO 10700
10606 IF (NN2.NE.06) GO TO 10607
10607 IF (NN2.NE.07) GO TO 10608
10608 IF (NN2.NE.08) GO TO 10609
    DIST = SCO(ITYPE1,ITYPE2)
    GO TO 598
10609 IF (NN2.NE.09) GO TO 10614
    DIST = SCF(ITYPE1)
    GO TO 598
10614 IF (NN2.NE.14) GO TO 10615
    DIST = SCSI(ITYPE1)
    GO TO 598
10615 IF (NN2.NE.15) GO TO 10616
    DIST = SCP(ITYPE1,ITYPE2)
    GO TO 598
10616 IF (NN2.NE.16) GO TO 10617
    DIST = SCS(ITYPE1,ITYPE2)
    GO TO 598
10617 IF (NN2.NE.17) GO TO 10635
    DIST = SCCL(ITYPE1)
    GO TO 598
10635 IF (NN2.NE.35) GO TO 10653
    DIST = SCBR(ITYPE1)
    GO TO 598
10653 IF (NN2.NE.53) GO TO 598
10700 IF (NN1.NE.07) GO TO 10800
10707 IF (NN2.NE.07) GO TO 10708
    DIST = SNN(ITYPE1,ITYPE2)
    GO TO 598
10708 IF (NN2.NE.08) GO TO 598
    DIST = SNO(ITYPE1,ITYPE2)
    GO TO 598
10800 IF (NN1.NE.08) GO TO 10900
10808 IF (NN2.NE.08) GO TO 10814
    DIST = SOO
    GO TO 598
10814 IF (NN2.NE.14) GO TO 10815
    DIST = SOSI
    GO TO 598
10815 IF (NN2.NE.15) GO TO 10816
    DIST = SOP(ITYPE1,ITYPE2)
    GO TO 598
10816 IF (NN2.NE.16) GO TO 598
    DIST = SSS
    GO TO 598
10900 IF (NN1.NE.09) GO TO 11400
10914 IF (NN2.NE.14) GO TO 598
    DIST = SFSI
    GO TO 598
11400 IF (NN1.NE.14) GO TO 11600
11417 IF (NN2.NE.17) GO TO 598
    DIST = SSICL
    GO TO 598
11600 IF (NN1.NE.16) GO TO 598
11616 IF (NN2.NE.16) GO TO 598
    DIST = SSS
    GO TO 598
20000 IF (NBORD.NE.2) GO TO 30000
20600 IF (NN1.NE.06) GO TO 20700
20606 IF (NN2.NE.06) GO TO 20607
    DIST = DCC(ITYPE1,ITYPE2)
GO TO 598
20607 IF (NN2.NE.07) GO TO 20608
  DIST = DCN(ITYPE1,ITYPE2)
  GO TO 598
20608 IF (NN2.NE.08) GO TO 20615
  DIST = DCO(ITYPE1,ITYPE2)
  GO TO 598
20615 IF (NN2.NE.15) GO TO 20616
  DIST = DCP(ITYPE1,ITYPE2)
  GO TO 598
20616 IF (NN2.NE.16) GO TO 598
  DIST = DCS(ITYPE1)
  GO TO 598
20700 IF (NN1.NE.07) GO TO 598
20708 IF (NN2.NE.OBl GO TO 598
  DIST = DNO(ITYPE1,ITYPE2)
  GO TO 598
30000 IF (NBORD.NE.)) GO TO 40000
30600 IF (NN1.NE.06) GO TO 598
30606 IF (NN2.NE.06) GO TO 30607
  DIST = TCC
  GO TO 598
30607 IF (NN2.NE.07) GO TO 598
  DIST = TCN
  GO TO 598
40000 CONTINUE
  C DEFAULT BOND DISTANCE tws 8/14/99
598 IF ( DIST .EQ. 0 ) DIST = 1.0
  BND(NDX1,NDX2) = DIST
  IF (DUMP) WRITE(20,5020) NAME1,ITYPE1,NDX1,NAME2,ITYPE2,
  1 NDX2,NBORD,DIST
5020 FORMAT (2(2X,A4,2I4},I4,F12.5)
  BND(NDX2,NDX1) = DIST
599 CONTINUE
C
C PASS TWO:
C NOW COMPUTE ALL 1-3 DISTANCES USING BND ARRAY FOR 1,2 DISTANCES
C AND PERTINENT BOND ANGLE, SELECTED ACCORDING TO ATOM TYPE.
C
  IF (DUMP) WRITE(20,3010)
3010 FORMAT (/* PASS 2: COMPUTE 1,3 DISTANCES. */
  1 ' I,D1,J,D2,K,BANGLE,DISTANCE ARE: ')
C
C IT IS TIME NOW TO FIND THE RINGS OF THE MOLECULE, AS THE 3 AND
C FOUR MEMBERED ONES DESERVE A SPECIAL TREATMENT
C
  CALL RINGS (NRG,RNGAT,RSIZE)
  IF (NRG.LE.0) GO TO 300
  IF (DUMP) WRITE(20,8010)
8010 FORMAT (' RINGS:')
  DO 830 I = 1,NRG
    NRS = RSIZE(I)
    IF (DUMP) WRITE (20,8020) I,NRS,(RNGAT(J,I),J=1,NRS)
8020 FORMAT (2X,12,' NRS = ',I2,' ,ATOMS: ',10I3)
  DO 810 J = 1,NRS
    ISINRG(RNGAT(J,I)) = NRS
810 CONTINUE
  IF (NRS.GT.4) GO TO 830
  CALL CORNER (DUMP,NRS,RNGAT(1,I))
830 CONTINUE
300 DO 330 J = 1,NAT
   IF (ISINRG(J) .EQ.3 .OR. ISINRG(J) .EQ.4) GO TO 330
   LIM = NCONN(J)
   IF (LIM.LE.1) GO TO 330
   IPREV2 = 0
   DO 320 K = 2,LIM
      NEIB2 = ICONN(K,J)
      IF (NEIB2.EQ.IPREV2) GO TO 320
      KM1 = K - 1
      IPREV1 = 0
      DO 310 I = 1,KM1
         NEIB1 = ICONN(I,J)
         IF (NEIB1.EQ.IPREV1) GO TO 310
         NEIB1 = ICONN(I,J)
         IF (NEIB1.EQ.NEIB2) GO TO 310
         DIST1 = BND(NEIB1,J)
         DIST2 = BND(J,NEIB2)
         NAME = NAMES(J) (1:2)
         ITYPE2 = ITYPE(J)
         IF (NAME.EQ.CARB) BANG = CBANGL(ITYPE2)
         DEFAULT BOND ANGLE tws 8/14/99
         BANG = CBANGL(ITYPE2)
         IF (NAME.EQ.NITR) BANG = NBANGL(ITYPE2)
         IF (NAME.EQ.OXYG) BANG = OBANGL(ITYPE2)
         IF (NAME.EQ.SILI) BANG = SIBANGL(ITYPE2)
         IF (NAME.EQ.PHOS) BANG = PBANGL(ITYPE2)
         IF (NAME.EQ.SULF) BANG = SBANGL(ITYPE2)
         BANGL(J) = BANG
         DIST = SQRT(DIST1**2 + DIST2**2 - 2.0*DIST1*DIST2*COSD(BANG))
         BND(NEIB1,NEIB2) = DIST
         BND(NEIB2,NEIB1) = DIST
         IPREV1 = NEIB1
         IF (DUMP) WRITE(20,3020) NAME,NEIB1,DIST1,J,DIST2,NEIB2,
                        1 BANG,DIST
            3020 FORMAT (2X,A4,3(I4,F12.5),F12.5)
      310 CONTINUE
   IPREV2 = NEIB2
   320 CONTINUE
   330 CONTINUE
C  PASS 3: FILL BND MATRIX WITH MINIMUM AND MAXIMUM 1-4 DISTANCES.
C  THESE ARE COMPUTED USING ALREADY DEFINED QUANTITIES LIKE 1-2
C  AND 1-3 DISTANCES AND BOND ANGLES, AND ASSUMING A MINIMUM AND
C  MAXIMUM TORSION ANGLE OF 0 AND 180 DEGREES RESPECTIVELY. THE
C  FORMULA IS THAT OF HENDRIKSON, JACS 83, 4537 (1961)
C
   IF (DUMP) WRITE(20,6010)
      6010 FORMAT (/*' PASS 3: COMPUTE 1,4 DISTANCES'/*
            'I, D1,J,D2,K,D3,L,DMIN,DMAX ARE:')
   DO 690 NB = 1,NBNDS
      JJ = BNDAT(1,NB)
      KK = BNDAT(2,NB)
      DBOND = BTYPE(NB) .EQ. 2
      WORRY = ISTER(JJ) .GE. 0 .AND. ISTER(KK) .GE. 0
      TORS = 180.0
      IF (ISTER(JJ) .NE. ISTER(KK)) TORS = 0.0
      LIM1 = NCONN(JJ)
      LIM2 = NCONN(KK)
      IF (LIM1.EQ.1 .OR. LIM2.EQ.1) GO TO 690
      DIST2 = BND(JJ,KK)
COS1 = COSD(BANGL(JJ))
SIN1 = SIND(BANGL(JJ))
COS2 = COSD(BANGL(KK))
SIN2 = SIND(BANGL(KK))
IPREV1 = 0
DO 670 I = 1,LIM1
II = ICONN(I,JJ)
IF (II.EQ.KK) GO TO 670
IF (II.EQ.IPREV1) GO TO 670
DIST1 = BND(II,JJ)
IF (ISINRG(JJ).NE.3.AND.ISINRG(JJ).NE.4) GO TO 604
C COMPUTE COSINE, SINE OF NON-STANDARD ANGLES
DISTC = BND(II,KK)
COS1 = (DIST1**2+DIST2**2-DISTC**2)/(2.0*DIST1*DIST2)
SIN1 = SQRT(1.0-COS1**2)
604 IPREV2 = 0
DO 650 L = 1,LIM2
LL = ICONN(L,KK)
IF (LL.EQ.IPREV2) GO TO 650
IF (JJ.EQ.LL) GO TO 650
IF (BND(II,LL).NE.DSTMIN.AND.BND(II,LL).NE.DSTMAX) GO TO 649
DIST3 = BND(KK,LL)
IF (ISINRG(KK).NE.3.AND.ISINRG(KK).NE.4) GO TO 608
DISTC = BND(JJ,LL)
COS2 = (DIST2**2+DIST3**2-DISTC**2)/(2.0*DIST2*DIST3)
SIN2 = SQRT(1.0-COS2**2)
608 OMGMIN = 0.0
OMGMAX = 180.0
IF (.NOT.WORRY.OR..NOT.DBOND) GO TO 610
OMGMIN = TORS
OMGMAX = TORS
610 PART1 = DIST1**2 + DIST2**2 + DIST3**2 -
1 2.0*DIST1*DIST2*COS1 - 2.0*DIST2*DIST3*COS2 +
2 2.0*DIST1*DIST3*COS1*COS2
PART2 = 2.0*DIST1*DIST3*SIN1*SIN2
DISMAX = SQRT(PART1-PART2*COSD(OMGMAX))
DISMIN = SQRT(PART1-PART2*COSD(OMGMIN))
BND(II,LL) = DISMAX
BND(LL,II) = DISMIN
IF (II.LT.LL) GO TO 620
BND(LL,II) = DISMIN
BND(LL,II) = DISMAX
620 CONTINUE
IPREV2 = LL
IF (DUMP) WRITE(20,6020) II,DIST1,JJ,DIST2,KK,DIST3,LL,
1 DISMIN,DISMAX
6020 FORMAT (I4,3(F12.5,I4),2F12.5)
649 TORS = TORS + 180.0
IF (TORS.GE.360.0) TORS = TORS - 360.0
650 CONTINUE
IPREV1 = II
IF (.NOT.DBOND) GO TO 670
TORS = TORS + 180.0
IF (TORS.GE.360.0) TORS = TORS - 360.0
670 CONTINUE
690 CONTINUE
C C NOW ADD EXTERNAL DISTANCE CONSTRAINTS
DO 900 ICSTR=1,NCSTR
BND(IATCTR(ICSTR),JATCTR(ICSTR)) = CSTRMX(ICSTR)
BND(JATCTR(ICSTR),IATCTR(ICSTR)) = CSTRMN(ICSTR)

900    CONTINUE
    IF (.NOT. DUMP) RETURN
    WRITE(20,7010)

7010   FORMAT (' FINAL BOUNDS MATRIX')
    CALL MATPRT(BND,NDIM,NAT)
    IF (DUMP) WRITE(20,7110)

7110   FORMAT (1X,120(1H-))

C

RETURN
END
SUBROUTINE RINGS (NRG, RNGAT, RSIZE)

IMPLICIT INTEGER (A-R, T-Z), INTEGER*8 (S)
INTEGER SETLEN
INTEGER CNT
LOGICAL DELFLG, LOGIC
INTEGER*8 AND, OR, XOR, NOT, ATLEFT, BDLEFT, INC(72)
INTEGER*8 QSETS(144), CLBASR(50), SXOR, SRBD(50)
INTEGER*8 BASR(50)
DIMENSION SRAT(50), LABEL(72), SCLRG(20)
DIMENSION RSIZE(20), RNGAT(10, 20)

COMMON /STLN/ SETLEN
COMMON /BONDS/ NBNDS, BNDAT(2, 72), BTYPE(72)
COMMON /ATOMS/ NAT, ICONN(4, 72), NCONN(72), NAMES(72), MCNT(72),
ITYPE(72)
EQUIVALENCE (INC, QSETS), (LABEL(1), QSETS(73)),
(BASR, SRAT)

SETLEN = 2
NRGMAX = 20
NRG = 0
SNUL = 0
SMSR = SNUL
SALLR = SNUL
STRINGA = SNUL
STRINGB = SNUL
SRBDHA = SNUL
SOCUPA = SNUL
SOCUPB = SNUL
CALL ZEROM(SRAT, 100)
CALL ZEROM(SRBD, 100)
CALL ZEROM(RNGAT, 200)
CALL ZEROM(RSIZE, 20)

IF NO RING POSSIBLE, RETURN

IF (NBNDS.EQ.NAT-1) RETURN

FILL INC, THE INCIDENCE MATRIX

CALL ZEROM(INC, 144)
DO 10 NB = 1, NBNDS
CALL ON(SOCUPB, NB)
CALL ON(INC(BNDAT(1, NB)), NB)
CALL ON(INC(BNDAT(2, NB)), NB)
10 CONTINUE
DO 20 NA = 1, NAT
CALL ON(SOCUPA, NA)
20 CONTINUE
ATLEFT = SOCUPA
BDLEFT = SOCUPB

WELCH'S ALGORITHM

BOND = 0
30 BOND = NXM(SCCUPB,BOND)
    IF (BOND.LE.0) GO TO 60

    LOOK FOR AN UNCHOSEN ATOM ON BOND. IF THERE IS NONE, THE BOND
    IS A RING CLOSURE BOND.

    ATOM = 0
40 ATOM = NXM(ATLEFT,ATOM)
    IF (ATOM.LE.0) GO TO 30
        DELFLG = ISM(INC(ATOM),BOND)
        IF (ISM(NOT(INC(ATOM)),BOND)) GO TO 40

    FOUND AN ATOM ON THIS BOND WHICH HAS NOT YET BEEN CHOSEN.
    SO EXTEND THE SPANNING TREE OUT OF THAT ATOM. THIS IS DONE
    BY LABELING THE ATOM WITH THE BOND NUMBER.

    LABEL(ATOM) = BOND

    REMOVE THIS ATOM AND BOND FROM THE SETS OF THOSE UNCHOSEN.

    CALL OFF(ATLEFT,ATOM)
    CALL OFF(BDLEFT,BOND)
    ATOM2 = 0
50 ATOM2 = NXM(SCCUPA,ATOM2)
    IF (ATOM2.LE.0) GO TO 30

    IF THIS IS THE ATOM JUST CHOSEN, DO NOTHING
    IF (ATOM.EQ.ATOM2) GO TO 50

    IF ATOMS ARE INCIDENT TO THE SAME EDGE, XOR THEM
    IF (ISM(INC(ATOM2),BOND)) INC(ATOM2) = XOR(INC(ATOM2),INC(ATOM))
    GO TO 50

    COLLECT BASIS RINGS. THE BONDS LEFT ARE THE RING-CLOSURES

60 NBR = 0
    BOND = 0
70 BOND = NXM(BDLEFT,BOND)
    IF (BOND.LE.0) GO TO 90

    START A NEW BASIS RING WITH THIS RING-CLOSURE

    NBR = NBR + 1
    CALL ON(BASR(NBR),BOND)
    ATOM = 0
80 ATOM = NXM(SCCUPA,ATOM)
    IF (ATOM.LE.0) GO TO 85

    IF ATOM IS ON BOND, PUT THE BOND LABELING THE ATOM IN RING SET
    IF (ISM(INC(ATOM),BOND)) CALL ON(BASR(NBR),LABEL(ATOM))
    GO TO 80

    THIS BASIS RING IS COMPLETE. PUT IT INTO THE SET OF RING BONDS

85 SRINGB = OR(SRINGB,BASR(NBR))
    GO TO 70
END OF WELCH'S ALGORITHM

90 CONTINUE

THIS SECTION ATTEMPTS TO FIND RING CLUSTERS.

STEMP = SNULL
DO 220 I = 1, NBR
CALL ON(STEMP, I)
220 CONTINUE

I = 0
NCL = 0

TAKE A BASIS RING

222 I = NXM(STEMP, I)
IF (I.LE.0) GO TO 226
NCL = NCL + 1

BEGIN A NEW CLUSTER

SCLRG(NCL) = SNULL
CALL ON(SCLRG(NCL), I)
SCLBD = BASR(I)
223 J = I
DELFGLG = .FALSE.
224 J = NXM(STEMP, J)
IF (J.LE.0) GO TO 225
IF (AND(SCLBD, BASR(J)).EQ.SNUL) GO TO 224

ADD THIS RING TO THE CLUSTER

DELFGLG = .TRUE.
CALL ON(SCLRG(NCL), J)
CALL OFF(STEMP, J)
SCLBD = OR(SCLBD, BASR(J))
GO TO 224

IF ADDED MORE RINGS TO CLUSTER REPEAT

225 IF (DELFGLG) GO TO 223
GO TO 222

NOW, ALL CLUSTERS HAVE BEEN FOUND. PROCESS EACH OF THEM

226 DO 1630 INDEX = 1, NCL
SETLEN = 2
RNCL = CNT(SCLRG(INDEX))

THE MAXIMUM IS TEN BASIS RINGS PER CLUSTER

IF (RNCL.GT.10) RNCL = 10
I = 0
J = 0
230 I = NXM(SCLRG(INDEX), I)
IF (I.LE.0) GO TO 235
J = J + 1
CLBASR(J) = BASR(I)
GO TO 230
235 CONTINUE
THIS SECTION IS USED TO CALCULATE THE REDUCED BASIS SET FROM
THE ARBITRARY BASIS SET THAT WE JUST OBTAINED.

THIS IS NOT NECESSARY IF THERE IS ONLY ONE RING.

IF (RNCL.LE.1) GO TO 280
N2 = RNCL - 1
245 DELFLG = .FALSE.
DO 270 J = 1,N2
   CNTRJ = CNT(CLBASR(J))
   I2 = J + 1
   DO 270 I = I2,RNCL
   CNTRI = CNT(CLBASR(I))
   STEMP = XOR(CLBASR(I),CLBASR(J))
   CNTRIJ = CNT(STEMP)
   IF THE XOR IS SMALLER THAN EITHER OF THE ORIGINAL TWO BASIS
   RINGS, REPLACE THE LARGER OF THE TWO BASIS RINGS WITH IT.
   IF (CNTRIJ.GE.CNTRI) GO TO 250
   DELFLG :: .TRUE.
   TEMP :: CNTRI
   CNTRI = CNTRIJ
   CNTRIJ = TEMP
   STEMP2 :: CLBASR(I)
   CLBASR(I) :: STEMP
   STEMP = STEMP2
250 IF (CNTRIJ.GE.CNTRJ) GO TO 270
DELFLG = .TRUE.
CNTRJ = CNTRIJ
CLBASR(J) = STEMP
270 CONTINUE
IF (DELFLG) GO TO 245

GIBB'S ALGORITHM.
QSETS ARE ALL OF THE COMBINATIONS OF CLBASR BOND SETS.
QFREE IS A POINTER TO THE NEXT FREE LOCATION IN QSETS TABLE
RNCL IS THE NUMBER OF BASIS RINGS IN THE CLUSTER
S IS THE SET THAT EVENTUALLY CONTAINS ALL OF THE RINGS

280 LENSET = (2**RNCL)/36 + 1
SETLEN = LENSET
CALL ZEROM(S,LENSET)
STEMP = SNULL
QFREE = 1
DO 120 I = 1,RNCL
   IF (I.EQ.1) GO TO 115
   LAST = QFREE - 1
   DO 110 J = 1,LAST
      SXOR = XOR(QSETS(J),CLBASR(I))
      IF NO INTERSECTION, THEN NOT A NEW RING
      IF (AND(CLBASR(I),QSETS(J)).EQ.SNULL) GO TO 106
      IF ANOTHER SET IS A SUBSET OF THIS SET, THEN NOT NEW RING.
      IF THIS SET IS A SUBSET OF ANOTHER SET, REMOVE THE OTHER.
K = LAST
102 K = NXM(STEMP,K)
   IF (K.LE.0) GO TO 104
   CALL sub(qsets(k),sxor,logic)
   IF (logic) go to 106
   CALL (SUB(SXOR,QSETS(K))) S = ibclr(S,K)
   GO TO 102
104 CALL ON(STEMP,QFREE)
GO TO 108
106 IF (I.EQ.RNCL) GO TO 110
108 QSETS(QFREE) = SXOR
   QFREE = QFREE + 1
110 CONTINUE
C ADD THE CLBASR TO THE S SET NOW, TOO
115 QSETS(QFREE) = CLBASR(I)
   CALL ON(STEMP,QFREE)
120 QFREE = QFREE + 1
C THE ALGORITHM IS NOW FINISHED.
I = 0
LOWNRG = NRG + 1
152 SETLEN = LENSET
   I = NXM(STEMP,I)
   SETLEN = 2
   IF (I.LE.0) GO TO 162
C ONLY KEEP THE RING IF IT IS IN THE REDUCED BASIS SET OR
C SMALLER THAN 10.
C DO 154 K = 1,RNCL
   IF (QSETS(I).EQ.CLBASR(K)) GO TO 155
154 CONTINUE
C NOT A REDUCED BASIS SET RING
C IF (CNT(QSETS(I)).LE.10) GO TO 156
C RING LARGER THAN ALLOWED FOR A NON-BASIS RING.
C GO TO 152
C PUT RING BOND SET IN FINAL ARRAY.
155 CALL ON(SMSPR, NRG+1)
156 NRG = NRG + 1
   SRBD(NRG) = QSETS(I)
GO TO 152
C LOOK FOR BRIDGE-HEAD ATOMS IN THIS CLUSTER
162 NRGM1 = NRG - 1
DO 163 I = LOWNRG,NRGM1
SATR1 = SAASB(SRBD(I))
DO 163 K = I+1,NRG
SATR2 = SAASB(SRBD(K))
Sboth = AND(SATR1,SATR2)
IF (CNT(Sboth).LE.3) GO TO 163
SATOR1 = AND(SATR1,NOT(Sboth))
SATOR2 = AND(SATR2,NOT(Sboth))
IF (SATOR1.EQ.SNULL.OR.SATOR2.EQ.SNULL) GO TO 163
SBRAT = AND(Sboth,SAASA(SATOR1))
SBRHDA = OR(SBRHDA,SBRAT)
163 CONTINUE
1630 CONTINUE
C
C NOW, GET THE ATOM SETS AND LISTS OF ATOMS IN ORDER FOR THE RINGS
C
C TO AVOID DIMENSION MISTAKES
IF ( NRG .GT. NRGMAX ) WRITE(20, 1200) NRG,NRGMAX
1200 FORMAT(' TOO MANY RINGS:',I3,' REDUCED TO:',I3)

164 LERR = 0
SETLEN = 2
DO 175 I = 1,NRG
 TEMP = 0
STEMP = SRBD(I)
GET FIRST BOND OF THIS RING
BND1 = NXM(STEMP,0)
REMOVE FROM RING BOND SET
CALL OFF(STEMP,BND1)
GET SECOND BOND
BND = NXM(AND(SBAB(BND1),STSM?),0)
GET FIRST ATOM AT INTERSECTION OF FIRST TWO BONDS
SRAT(I) = AND(SAAB(BND),SAAB(BND1))
GET NUMBER OF FIRST ATOM
TATM = NXM(SRAT(I),0)
PUT ON LIST OF ATOMS
TEMP = TEMP + 1
RNGAT(TEMP,I) = TATM
GET NEXT ATOM
170 TATM = ADJAT(TATM,BND)
TURN ON IN SET
CALL ON(SRAT(I),TATM)
PUT ON LIST OF ATOMS
TEMP = TEMP + 1
RNGAT(TEMP, I) = TATM

REMOVE CURRENT BOND FROM SET OF RING BONDS LEFT

CALL OFF(STEMP, BND)

GET NEXT BOND

BND = NXM(AND(SBAB(BND), STEMP), 0)

DONE WITH ALL BONDS ON ATOM YET

IF (BND.GT.0) GO TO 170

175 CONTINUE

GET SET OF RING ATOMS AND RING SIZE SETS

DO 176 I = 1, NRG
SRINGA = OR(SRINGA, SRAT(I))
RS = CNT(SRAT(I))
RSIZE(I) = RS

176 CONTINUE
RETURN
END
SUBROUTINE SCCG (FCALC,NR,X,NFCT,NITN,FCTVAL,GRDNRM,STATUS,DUMP)

C HAVEL'S IMPLEMENTATION OF THE SELF-CORRECTING CONJUGATE GRADIENT
C OPTIMIZER OF A. PERRY (INTNL. J. COMPUTER MATH. (1978) B6 327-33)

MODIFIED FOR DEC-10 BY D. SMITH, 10/15/80

C COMMON /CTRL/ SLFCOR,IFREQ,ITNLIM,FCTMIN,GRDMIN
COMMON /PARA/ GESFCT,STPMIN,STPMAX,CAPPA,DELTA,EPSLN,
1 FAST,SLOW,ANGMAX,INTMAX,RESTAR,ISEED
COMMON /SCCGC/ P(3,72),Q(3,72),C(3,72),G(3,72)
DIMENSION X(3,1)

PARAMETERS: NR - NO. OF RESIDUES
FCALC - NAME OF THE FUNCTION TO BE MINIMIZED
SLFCOR - SCCG IF TRUE; HSCG IF FALSE
GESFCT - MIN. FRACTION OF OLD FCTN DECREASE EXPECTED
STPMIN - MINIMUM ALLOWED LINE SEARCH STEP SIZE
STPMAX - MAXIMUM ALLOWED LINE SEARCH STEP SIZE
CAPPA - 0 < ACCURACY OF LINE SEARCH CONTROL < 1
DELTA - ERROR IF RELATIVE FUNCTION INCREASE EXCEEDS
EPSLN - ERROR IF TOTAL MOVE IS LESS THAN THIS
FAST - STEEPEST DESCENT WHILE FCTN DECREASE EXCEEDS
SLOW - RESTART IF RELATIVE FCTN DECREASE DROPS BELOW
ANGMAX - RESTART IF ANGLE BTN GRAD & CONJ GRAD EXCEEDS
INTMAX - MAXIMUM NO. OF INTERPOLATIONS IN LINE SEARCH
RESTAR - PERIODIC RESTART EVERY RESTAR ITERATIONS
FCTMIN - NORMAL EXIT IF FUNCTION GETS LESS THAN THIS
GRDMIN - NORMAL EXIT IF GRADIENT GETS LESS THAN THIS
ITNLIM - ERROR RETURN IF NO. OF ITERATIONS EXCEEDS THIS
IFREQ - IF NONZERO, OUTPUT EVERY IFREQ'D ITERATION
   IF NEGATIVE, OUTPUT INCLUDES PRESENT COORDS.
NFCT - NO. OF FUNCTION CALLS
NITN - NO. OF ITERATIONS
STATUS - CHARACTER STRING CONTAINING ERROR DESCRIPTION
FCTVAL - FINAL FUNCTION VALUE
GRDNRM - FINAL GRADIENT NORM
X & G - COORDINATES AND GRADIENT
   C - CONJUGATE GRADIENT AND ESTIMATED PRIOR GRADIENT
P & Q - DIFFERENCES BETWEEN CURRENT AND ESTIMATED CO-
   ORDINATES AND GRADIENTS; ALSO CONTAIN OLD CO-
   ORDINATES AND GRADIENTS PART OF THE TIME

EXTERNAL FCALC,COEFCT,SEARCH
LOGICAL SLFCOR, DUMP, NOGRAD
tws 6/9/99
INTEGER BLANK,STATUS(2),RESTAR
   CHARACTER*5 STATUS(2),BLANK
   INTEGER RESTAR
DATA BLANK /5H /
STATUS(1) = BLANK
STATUS(2) = BLANK
N = NR
NITN = 0
NFCT = 0
NOGRAD=.FALSE.
IF (DUMP .AND. IFREQ .NE. 0) WRITE(20,1) GESFCT, STPMIN, STPMAX, CAPPA,
1 DELTA, EPSLN, FAST, SLOW, ANGMAX, INTMAX, RESTAR, FCTMIN, GRDMIN,
2 ITNLIM, SLFCOR

1 FORMAT(/,' ENTER SCCG: PARAMETERS:/', GESFCT, STPMIN, STPMAX,
1 CAPPA, DELTA, EPSLN, FAST, SLOW, ANGMAX, INTMAX, RESTAR,
2 FCTMIN, GRDMIN, ITNLIM, SLFCOR/, /9E9.2, 2I9, 2E9.2, 19.19, /)

C
NFCT = NFCT+1
FCT = FCALC (N,X,G,NOGRAD)
OLDFCT = 2.*FCT
GDG = 0.
DO 2 I=1,3
DO 2 J=1,N
GDG = GDG + G(I,J)**2
P(I,J) = 0.
Q(I,J) = 0.
C(I,J) = 0.
2 CONTINUE

GNM = SQRT(GDG)

C
IF (.NOT.DUMP) GO TO 10
IF (IFREQ .NE. 0) WRITE(20,3) FCT, GNM
3 FORMAT(' BEGIN SCCG WITH', 18X, E9.3, ' FCT VAL; .. E9.3.
1' GRAD NORM', /)
4 FORMAT(18X, 'added to prevent run away jobs tws 8/17/99
4 IF (GNM .EQ. 0.0) GO TO 3000
5 IF (IFREQ .GE. 0) GO TO 10
WRITE(20,4)

5 FORMAT(3(I5, 1X, 3F12.3))
WRITE(20,6)
6 FORMAT(/)

******************************************************************************
BEGIN: SET COEF TO ZERO AND GO TO CONJUGATE GRADIENT CALCULATION
******************************************************************************
THIS (RE)STARTS THE PROCEDURE AND IS DONE AT LEAST EVERY N ITER-
ATIONS. IN NORMAL RESTARTS, WE PROCEED WITH LATEST COORDINATES;
IN ERROR RESTARTS, WE RESTORE THE OLD COORDINATES.
10 COEF = O.

******************************************************************************
MAIN ITERATION LOOP FOR SELF-CORRECTING CONJUGATE GRADIENT SEARCH
******************************************************************************
IF ITN=1, WE ARE GOING IN STEEPEST DECENT DIRECTION

DO 1000 ITN = 1, RESTAR
STATUS(1) = BLANK
STATUS(2) = BLANK
IF (ITN .EQ. 1) GO TO 100

Determine the coefficient for computing the new conjugate direction

COEF = COEFC (FCALC, N, X, FCT, GDG, NFCT, STATUS, SLFCOR)

Compute the new search vector conjugate to the (corrected) prior
search vector, and store current coordinates and gradient
100 CDC = 0.
DO 101 I=1,3
DO 101 J=1,N
C(I,J) = - G(I,J) + COEF * P(I,J)
CDC = CDC + C(I,J)**2
P(I,J) = X(I,J)
Q(I,J) = G(I,J)
101 CONTINUE
CNM = SQRT(CDC)

C NORMALIZE THE RESULTANT VECTOR AND CHECK THE NEW SEARCH
DIRECTION TO MAKE SURE IT IS NOT TOO FAR FROM THE GRADIENT

SLOPE = 0.
DO 110 I=1,3
DO 110 J=1,N
C(I,J) = C(I,J) / CNM
SLOPE = SLOPE + G(I,J) * C(I,J)
110 CONTINUE

COSANG = -SLOPE/GNM
IF (COSANG.GT.1.0) COSANG = 1.0
AACCOS = ACOS(COSANG)
ANGLE = 57.2958 * AACCOS
IF (ANGLE.LT.ANGMAX) GO TO 111
STATUS(1) = 5H WIDE
STATUS(2) = 5HANGLE
GO TO 303

C PERFORM LINE SEARCH ALONG THE NEW CONJUGATE DIRECTION

111 FCTLOW = SEARCH(FCALC,NFCT,N,X,FCT,OLDFCT,SLOPE,STATUS)
IF ((STATUS(1).EQ.5H INTPI) .AND. (STATUS(2).EQ.5H LINERR)) GO TO 200

C CHECK THAT THE FUNCTION DID NOT INCREASE SIGNIFICANTLY AND
THAT THE MAGNITUDE OF OUR TOTAL MOVE WAS SIGNIFICANT

RATIO = (FCT-FCTLOW) / AMAX1(ABS(FCT),ABS(FCTLOW))
IF (-RATIO.LE.DELTA) GO TO 200
STATUS(1) = 5H INCR
STATUS(2) = 5H EASE
GO TO 200

200 TMPGDG = 0.
TOTMVE = 0.
DO 202 I=1,3
DO 202 J=1,N
TOTMVE = TOTMVE + (P(I,J)-X(I,J))**2
TMPGDG = TMPGDG + G(I,J)**2
202 CONTINUE
TOTMVE = SQRT(TOTMVE)
IF (TOTMVE.GE.EPSLN) GO TO 220
STATUS(1) = 5H SMAL
STATUS(2) = 5H LMOVE
GO TO 200

C SUCCESSFUL ITERATION COMPLETE: UPDATE VARIABLES

220 NITN = NITN+1
OLDFCT = FCT  
FCT = FCTLOW  
GDG = TMPGDG  
GNM = SQRT(GDG)  

OUTPUT INTERMEDIATE RESULTS EVERY IFREQ ITERATIONS  

IF (IFREQ.NE.0 .AND. ((ITN-1)/IFREQ)*IFREQ.EQ.ITN-1 .AND. DUMP)  
1 CALL PRINT (N,X,ITN,FCT,GNM,ANGLE,TOTMVE,NFCT,NITN,IFREQ)  

TEST FOR NORMAL TERMINATION OR TOO MANY ITERATIONS HERE  

IF (FCT.LE.FCTMIN .OR. GNM.LE.GRDMIN) GO TO 3000  
IF (NITN.LE.ITNLIIM) GO TO 300  
STATUS(1) = 5H OVER  
STATUS(2) = 5HLIMIT  
GO TO 2020  

TEST TO SEE IF WE SHOULD RESTART (SOME MINOR GLITCH, TOO MANY CONJUGATE STEPS. PROGRESS FAST IN STEEP DIRECTION OR SLOW IN CG DIRECTION)  

300 STATUS(1) = 5H FAST  
STATUS(2) = 5HDEC  
IF (ITN.EQ.1 .AND. RATIO.GE.FAST) GO TO 303  
STATUS(1) = 5H SLOW  
STATUS(2) = 5HDEC  
IF (ITN.GT.(3*N/10).AND. RATIO.LE.SLOW) GO TO 303  
GO TO 1000  

NORMAL RESTART: PROCEED FROM CURRENT POINT IN STEEPEST DIRECTION  

303 IF (IFREQ.NE.0.AND.DUMP) WRITE(20,330) ITN,(STATUS(NNN),NNN=1,2)  
330 FORMAT(/'*** NORMAL RESTART AFTER ' , I3, ' CONJUGATE STEPS BECAUSE 1 OF', 2A5, '***',/)  
GO TO 10  

END OF MAIN LOOP END OF MAIN LOOP END OF MAIN LOOP END OF MAIN LOOP  

1000 CONTINUE  
STATUS(1) = 5H PERI  
STATUS(2) = 5HODIC  
IF (IFREQ.NE.0.AND.DUMP) WRITE(20,330) ITN,(STATUS(NNN),NNN=1,2)  
GO TO 10  

ERROR CONTROL SECTION: RESTART OR RETURN IF WE HAVE JUST RESTARTED  

2000 DO 2002 I=1,N  
DO 2002 J=1,N  
X(I,J) = P(I,J)  
G(I,J) = Q(I,J)  
2002 CONTINUE  
IF (ITN.NE.1) GO TO 2200  
2020 IF (IFREQ.NE.0.AND.DUMP) WRITE(20,2022) (STATUS(NNN),NNN=1,2)  
2022 FORMAT(/'*** ERROR RETURN BECAUSE OF ', 2A5, '***',/)  
FCTVAL = FCT  
GRDNRM = GNM
RETURN
2200 IF (IFREQ.NE.O.AND.DUMP) WRITE(20,2202) ITN,(STATUS(NNN),NNN=1,2)
2202 FORMAT(,,** ERROR RESTART AFTER 'I3,' CONJUGATE STEPS BECAUSE
1 OF .2AS, '*** ///)
   GO TO 10

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

NORMAL TERMINATION FROM HERE NORMAL TERMINATION FROM HERE NORMAL
******************************************************************************

3000 IF (IFREQ.NE.O.AND.DUMP) WRITE(20,3003)
3003 FORMAT(,,** NORMAL TERMINATION SCCG ***,///)
   FCTVAL = FCT
   GRDNRM = GNM
   STATUS(1) = BLANK
   STATUS(2) = BLANK
   RETURN
END

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

REAL FUNCTION COEFCT (FCALC,NP,X,FCT,GDG,NFCT,STATUS,SLFCOR)
******************************************************************************

RETURNS THE COEFFICIENT TO BE USED IN COMPUTING THE CONJUGATE
DIRECTION, AS WELL AS THE DISPLACEMENT VECTOR P IN /SCCG/.

EXTERNAL FCALC
LOGICAL SLFCOR,NOGRAD
CHARACTER*5 STATUS(2)
COMMON /SCCG/ P(3,72),Q(3,72),C(3,72),G(3,72)
DIMENSION X(3,NP)

NOGRAD=.FALSE.
N = NP

BEGIN BY COMPUTING THE UNCORRECTED HESTENES-STIEFEL COEFFICIENT
AND THE GRAM-SCHMIDT ORTHOGONALIZATION COEFFICIENT FOR P & G.

GDQ = 0.
PDQ = 0.
PDG = 0.
DO 1 I=1,3
DO 1 J=1,N
   P(I,J) = X(I,J) - P(I,J)
   Q(I,J) = G(I,J) - Q(I,J)
   PDQ = PDQ + P(I,J) * Q(I,J)
   GDQ = GDQ + G(I,J) * Q(I,J)
1   CONTINUE
   HSCOEF = GDQ/PDQ
   IF (.NOT.SLFCOR) GO TO 110
   GSCOEF = PDG/GDG

COMPUTE THE CORRECTED DISPLACEMENT VECTOR TO BE ORTHOGONAL TO THE
CURRENT GRADIENT AS WOULD BE EXPECTED FROM A PERFECT LINE SEARCH

DO 10 I=1,3
DO 10 J=1,N
   P(I,J) = P(I,J) - GSCOEF * G(I,J)
   Q(I,J) = X(I,J) - P(I,J)
10  CONTINUE
CONTINUE
C FIND THE CORRECTED GRADIENT C AND CORRESPONDING FUNCTION VALUE AT
C THE CORRECTED PRIOR POINT, AND CHECK THAT THIS GIVES A DECREASE
C
NFCT = NFCT + 1
IF (FCALC(N,Q,C,NOGRAD).LE.FCT) GO TO 100
C
COMPUTE THE CORRECTED HESTENES-STIEFEL COEFFICIENT AND DOT PRODS
C
QDG = 0.
PDC = 0.
CDC = 0.
PDP = 0.
DO 11 I=1,3
   DO 11 J=1,N
      QCOR = G(I,J) - C(I,J)
      QDG = QDG + QCOR * G(I,J)
      PDC = PDC + P(I,J)*C(I,J)
      CDC = CDC + C(I,J)**2
      PDP = PDP + P(I,J)**2
11 CONTINUE
SCCOEF = - QDG/PDC
PNM = SQRT(PDP)
CNM = SQRT(CDC)
C
CHECK THE ANGLE BETWEEN THE CORRECTED DISPLACEMENT AND GRADIENT
C
COSANG = -PDC/(PNM*CNM)
IF (COSANG.GT.1.0) COSANG = 1.0
ANGLE = 57.2958 * ACOS(COSANG)
IF (ANGLE.GT.89.9) GO TO 100
C
RETURN WITH CORRECTED COEFFICIENT AND DISPLACEMENT VECTOR
C
COEFC = SCCOEF
RETURN
C
IF THE CORRECTED PRIOR POINT IS UNACCEPTABLE, RESTORE THE OLD
C DISPLACEMENT VECTOR AND RETURN THE HESTENES-STIEFEL COEFFICIENT
C
100 DO 101 I=1,3
   DO 101 J=1,N
      P(I,J) = P(I,J) + GSCEOF * G(I,J)
101 CONTINUE
STATUS(1) = 5H NOCO
STATUS(2) = 5HRRCTN
110 COEFC = HSCCEOF
RETURN
END
C
REAL FUNCTION SEARCH (FCALC,NFCT,NO,X,FCT,OLDFCT,SLOPE,STATUS)
*****************************************************************************
C THIS IS THE PARABOLIC EXTRAPOLATION / CUBIC INTERPOLATION PROCEDURE DUE TO
C DIXON, POWELL, WEINER AND HAVEL (1980).
C
EXTERNAL FCALC
LOGICAL NOGRAD
INTEGER RESTAR
PARABOLIC EXTRAPOLATION FROM PREVIOUS FUNCTION DECREASE INITIALIZES

\[ \text{GUESS} = FCT - \text{GESFCT} \times (\text{OLDFCT} - FCT) \]
\[ \text{STEP} = 2. \times (\text{GUESS} - FCT) / \text{SLOPE} \]

IF (\text{STEP} \geq \text{STPMAX}) \text{ STEP} = \text{STPMAX}
IF (\text{STEP} \leq \text{STPMIN}) \text{ STEP} = \text{STPMIN}

\text{FCTB} = FCT
\text{CDGO} = \text{SLOPE}

BEGIN EXTRAPOLATION

1 \text{ FCTB} = FCT0
\text{CDGB} = \text{CDGO}
\text{INTPLN} = 0

REPLACE LAST POINT BY LATEST AND TAKE ANOTHER STEP

2 \text{ FCTA} = \text{FCTB}
\text{CDGA} = \text{CDGB}
DO 3 \text{ I=1,3}
DO 3 \text{ J=1,N}
\text{X(I,J)} = \text{X(I,J)} + \text{STEP} \times \text{C(I,J)}
3 \text{ CONTINUE}

NFCT = NFCT+1
\text{FCTB} = \text{FCALC} (\text{N,X,G,NOGRAD})
\text{CDGB} = 0.
DO 4 \text{ I=1,3}
DO 4 \text{ J=1,N}
\text{CDGB} = \text{CDGB} + \text{C(I,J)} \times \text{G(I,J)}
4 \text{ CONTINUE}

IF THE FINITE DIFFERENCE CURVATURE IS NEGATIVE DOUBLE THE STEP;
ELSE IF \text{STEP} < \text{PARABOLIC ESTIMATE} \times 4*\text{STEP} USE THIS ESTIMATE;
ELSE TRUNCATE TO \text{STEP} OR 4*\text{STEP}, RESP.

IF (ABS(\text{CDGB}/\text{SLOPE}) \leq \text{CAPP} A \AND \text{FCTB} \LT \text{FCTA}) \text{ GO TO 1003}
IF (\text{CDGA} \times \text{CDGB} \LT 0. \OR \text{FCTA} \LT \text{FCTB}) \text{ GO TO 11}
\text{STEP} = 2.0 \times \text{STEP}
IF (\text{CDGB} - \text{CDGA} \LE 0.) \text{ GO TO 5}
\text{PARAB} = (\text{FCTA} - \text{FCTB}) / (\text{CDGB} - \text{CDGA})
IF (\text{PARAB} \GT 2.0 \times \text{STEP}) \text{ PARAB} = 2.0 \times \text{STEP}
IF (\text{PARAB} \LT \text{STEP}/2.0) \text{ PARAB} = \text{STEP}/2.0
\text{STEP} = \text{PARAB}
5 \text{ IF} (\text{STEP} \GE \text{STPMAX}) \text{ STEP} = \text{STPMAX}
\text{ GO TO 2}

IF SLOPES CHANGE OR FUNCTION INCREASES BEGIN CUBIC INTERPOLATION

11 \text{ CUBE} = 0.
\text{FCTC} = 1.0E35
\text{RRR} = 3.0 \times (\text{FCTB} - \text{FCTA}) / \text{STEP}
SSS = RRR - CDGA - CDGB
TSQ = SSS*SSS - CDGA*CDGB
IF (TSQ.LT.0.) GO TO 1002

TTT = SQRT(TSQ)
CUBE = STEP * (CDGB+TTT+SSS) / (CDGB-CDGA+2.0*TTT)
IF (CUBE.LT.0. .OR. CUBE.GT.STEP) GO TO 1002
DO 12 I=1,3
DO 12 J=1,N
X(I,J) = X(I,J) - CUBE * C(I,J)
12 CONTINUE

NFCT = NFCT+1
FCTA = FCALC (N,X,G,NOGRAD)
CDGA = 0.
DO 13 I=1,3
DO 13 J=1,N
CDGA = CDGA + C(I,J) * G(I,J)
13 CONTINUE

MAYDAY = 0
IF (FCTA.GT.FCTC) MAYDAY = MAYDAY + 1
IF (FCTB.GT.FCTA) MAYDAY = MAYDAY + 1
INTPLN = INTPLN + 1

IF THE BRACKETTING POINTS HAVE OPPOSITE SLOPE REPLACE THE INTERPOLATED POINT BY THE BRACKET WHOSE SLOPE HAS THE SAME SIGN & REPEAT;
ELSE IF THE INTERPOLATED AND BRACKETS HAVE OPPOSITE SLOPE REPLACE THE FAR BRACKET (B); ELSE IF THE INTERPOLATED HAS LOWEST FUNCTION VALUE REPLACE THE NEAR BRACKET (A); ELSE REPLACE B; REPEAT

IF (ABS(CDGA/SLOPE) .LE.CAPPAI GO TO 1004
CUBSTP = AMIN1( ABS(CUBE),ABS(STEP-CUBE) )
IF (INTPLN.GE.INTMAX .OR. CUBSTP.LT.STPMIN .OR. MAYDAY.EQ.2) GO TO 101
IF (CDGA*CDGB.GE.0.) GO TO 14
IF (CDGA*CDGC.LT.0.) GO TO 15
GO TO 16
14 IF (CDGA*CDGC.LT.0.) GO TO 15
IF (FCTC.LT.FCTA) GO TO 16

15 FCTB = FCTC
CDGB = CDGC
STEP = STEP-CUBE
GO TO 11

16 FCTA = FCTC
CDGA = CDGC
STEP = CUBE
DO 17 I=1,3
DO 17 J=1,N
X(I,J) = X(I,J) + CUBE * C(I,J)
17 CONTINUE
GO TO 11

IF INTERPOLATION LIMIT EXCEEDED OR FUNCTION INCREASES OR THE INTERPOLATION CHANGE IS INSIGNIFICANT RESTART THE LINE SEARCH FROM THE LOWEST AVAILABLE POINT WITH A MUCH SMALLER STEP

101 FCT1 = AMIN1( FCTA,FCTB,FCTC )
IF (FCT1.LT.FCT0) GO TO 1002
IF (FCT1.EQ.FCTA) GO TO 103
IF (FCT1.EQ.FCTB) GO TO 105
CDG1 = CDGC
SGN = - SIGN(1.0,CDG1)
DO 102 I=1,3
DO 102 J=1,N
C(I,J) = SGN * C(I,J)
102 CONTINUE
GO TO 107

103 CDG1 = CDGA
SGN = - SIGN(1.0,CDG1)
DO 104 I=1,3
DO 104 J=1,3
X(I,J) = X(I,J) + (CUBE-STEP) * C(I,J)
C(I,J) = SGN * C(I,J)
104 CONTINUE
GO TO 107

105 CDG1 = CDGB
SGN = - SIGN(1.0,CDG1)
DO 106 I=1,3
DO 106 J=1,N
X(I,J) = X(I,J) + CUBE * C(I,J)
C(I,J) = SGN * C(I,J)
106 CONTINUE
GO TO 107

107 STEP = AMAX1( CUBE,STEP-CUBE ) / 10.0
IF (STEP.LT.STPMIN) STEP = STPMIN

1001 STATUS(1) = 5H BADI
STATUS(2) = 5HINTPLN
SEARCH = FCT1
RETURN

1002 SEARCH = FCT
CALL SPIT(FCT0,FCTA,FCTB,FCTC,CDGO,CDGA,CDGB,CDGC,STEP,CUBE,RRR,
1 SSS,TSQ,TTP,X(1,1),G(1,1),CUBSTP,INTPLN,MAYDAY,STATUS)
STATUS(1) = 5HINTP
STATUS(2) = 5HLNERR
RETURN

1003 SEARCH = FCTB
RETURN

1004 SEARCH = FCTC
RETURN
END

SUBROUTINE SPIT(FCT0,FCTA,FCTB,FCTC,CDGO,CDGA,CDGB,CDGC,STEP,CUBE,
1 RRR,SSS,TSQ,TTP,X,G,CUBSTP,INTPLN,MAYDAY,STATUS)

INTEGER STATUS(2)
character*5 STATUS(2)
WRITE(20,9999) (STATUS(NNN), NNN=1,2),FCT0,FCTA,FCTB,
SUBROUTINE PRINT(N,X,ITN,FCT,GNM,ANG,TMV,NFCT,NITN,IFREQ)
C
C OUTPUT ROUTINE FOR SCCG; DUMPS SPECIFICS ON OPTIMIZATION,
C ALONG WITH COORDINATES IF IFREQ IS LESS THAN ZERO
C
DIMENSION X(3,N)
WRITE(20,1) NITN,NFCT,FCT,GNM,ANG,TMV,ITN
1 FORMAT(' ITRN ',13,' OF SCCG: ',14,' FCT CALLS; ',E9.3,
1 ' FCT VAL; ',E9.3,' GRAD NORM; ',F6.2,' ANGLE; ',E9.3,
2 ' TOT MOVE; ',13,' CONJ STEPS. ')
IF (IFREQ.GT.0) RETURN
WRITE(20,2)
2 FORMAT(/,' COORDINATES: /)
WRITE(20,3) (J, (X(I,J),I=1,3),J=1,N)
3 FORMAT((I5,1X,3F12.3))
WRITE(20,4)
RETURN
END
subroutine sub(setA, setB, yes)
    c sub is a subroutine used in rings - it takes two sets and
determines if one set is a subset of the second
    integer*8 setA, setB, setBcp, combnd
    logical yes
    yes = .FALSE.
    setBcp = not(setB)
    combnd = iand(setA, setBcp)
    if (combnd.eq.0) yes = .TRUE.
    return
end

subroutine on(setA, nbit)
    c on is a subroutine used in rings - it sets bit nbit on
    c the right most bit is bit 1
    integer*8 setA
    c
    setA = ibset(setA, nbit-1)
    return
end

subroutine off(setA, nbit)
    c off is a subroutine used in rings - it sets bit nbit off
    c the right most bit is bit 1
    integer*8 setA
    c
    setA = ibclr(setA, nbit-1)
    return
end

logical function ism(setA, nbit)
    c ism is a function used in rings - it tests bit nbit
    c the right most bit is bit 1
    integer*8 setA
    ism = bktest(setA, nbit-1)
    return
end

integer function cnt(setA)
    c cnt is a function used in rings - it counts set bits
    c the right most bit is bit 1
    integer*8 setA
    c
    ibytes = 8
    nbits = 8*ibytes
    icnt = 0
    do 10 itst = 0, nbits-1
       if (bktest(setA, itst)) icnt = icnt + 1
       continue
    10 cnt = icnt
    return
end
subroutine zerom(iarray,n)
  ZEROM(ARRAY, # of words) ZEROS N WORDS BEGINNING AT ARRAY.
  example: CALL ZEROM(ARRAY,NWORDS) for NWORDS of INTEGER TYPE
  This routine replaces a machine language routine in the original.
  For regular INTEGER type NWORDS=dimensioned length
  For DOUBLE PRECISION type NWORDS=2*dimensioned length
  tws 6/11/99
  integer iarray
  dimension iarray(*)
  do 100 i=1,n
    iarray(i)=0
  100 continue
  return
end
References


11. V. M. Indivero; T. A. Stephenson; Physical Chemistry - Developing a Dynamic Curriculum. ACS, 1993, Chapter 16.


