OPTIMIZING SOLVENT BLENDS FOR A QUINARY SYSTEM

A Thesis
Presented to
The Graduate Faculty of The University of Akron

In Partial Fulfilment
Of the Requirements for the Degree
Master of Science

Thomas Hoy
May, 2016
OPTIMIZING SOLVENT BLENDS FOR A QUINARY SYSTEM

Thomas Hoy

Thesis

Approved:  

Advisor  
Dr. David Simmons

Co-Advisor  
Dr. Kevin Cavicchi

Committee Member  
Dr. Mark D. Soucek

Accepted:  

Department Chair  
Dr. Sadhan C. Jana

Dean of the College  
Dr. Eric J. Amis

Dean of the Graduate School  
Dr. Chand Midha

Date
ABSTRACT

The Department of Polymer Engineering had received a grant from the Keck foundation to produce a process in which hundreds of copolymers can be formulated, synthesized, analyzed and then extrapolated to produce the optimal polymer for the desired material property. This process involves synthesizing approximately fifty polymers in a custom apparatus, none of which necessarily have the same monomer components. This leads to issue of which solvent or blend of solvents to use that would sufficiently dissolve both monomers and the resulting copolymer. If the monomers are not completely dissolved, a homogenous reaction will be impossible and if the copolymer is not soluble; it will precipitate out before the reaction is complete. My project is designing a computer program that will use the Hansen Solubility Parameters to predict the optimal blend of solvents that will sufficiently dissolve the monomers and resulting copolymers. The program will be written in C++ and executed in a Linux environment. The Hansen Solubility Parameters are derived purely from experimental data and are yet to have a strong theoretical basis. Despite this limitation, it is believed that these parameters will be sufficient for the needs of this project.

The Hansen Solubility Parameters (HSP) describe monomers, polymers, and solvents as three dimensional shapes. For the sake of clarity one may
consider these shapes as spheres, the center of which is described by HSP, and the radius is defined by the molar volume of the substance. All points that lie within these spheres describe a potential solvent for the given substance. The process of this optimization can take a human several minutes to complete, while a computer can accomplish the same task in fractions of a second. When designing a high throughput system, it is impractical to be spending multiple hours deciding which solvents to use for fifty polymer systems per iteration. This will eliminate an experimental bottleneck for the overall project. Lab results have shown that solvent blends calculated by the code sufficiently solvate the monomers for chosen copolymer systems.
DEDICATION

The author would like to dedicate this thesis to his friends and family who have shown him endless support throughout his education.
AKNOWLEDGEMENTS

The Author would like to thank his advisors Dr. David Simmons and Dr. Kevin Cavicchi for the guidance throughout the entirety of this Thesis. Also to Jui-Hsiang (Sean) Hung and Venkatesh Meenakshisundaram for their assistance in writing computer code.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>ix</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. LITERATURE</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Chemical Protective Clothing</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Organogels</td>
<td>5</td>
</tr>
<tr>
<td>2.3 Polymer Coatings</td>
<td>6</td>
</tr>
<tr>
<td>III. BACKGROUND</td>
<td>8</td>
</tr>
<tr>
<td>3.1 Hildebrand Solubility Parameter</td>
<td>8</td>
</tr>
<tr>
<td>3.2 Hansen Solubility Parameters</td>
<td>9</td>
</tr>
<tr>
<td>3.3 Group Contribution Theory</td>
<td>11</td>
</tr>
<tr>
<td>IV. APPLICATION OF HSP</td>
<td>13</td>
</tr>
<tr>
<td>4.1 Derivation of the Algorithm</td>
<td>13</td>
</tr>
<tr>
<td>4.2 Cases that Produce Valid Solvent Blends</td>
<td>15</td>
</tr>
<tr>
<td>4.2.1 Case 1</td>
<td>16</td>
</tr>
<tr>
<td>4.2.2 Case 2</td>
<td>17</td>
</tr>
<tr>
<td>4.2.2.1 Subcase 2.1</td>
<td>19</td>
</tr>
<tr>
<td>4.2.2.2 Subcase 2.2</td>
<td>19</td>
</tr>
<tr>
<td>4.2.3 Case 3</td>
<td>20</td>
</tr>
</tbody>
</table>
4.2.3.1 Subcase 3.1 ................................................................. 20
4.2.3.2 Subcase 3.2 ................................................................. 21
4.3 Solvent Bank ................................................................. 21
4.4 Experimental Results ...................................................... 22
V. CONCLUSION ................................................................. 25
REFERENCES ................................................................. 27
APPENDIX ................................................................. 30
LIST OF TABLES

Table  Page

1. Volumes of each solvent and monomer were calculated from the output of the procedure given in Figure 3.11. 3 mL total of monomer were used and 5 mL total of Solvent was used in each test.........................................................24
3.1 The polymer given in figure 2.1 has $\delta$ of 4 and an interaction radius of 4. Solvent one lies within the interaction radius, suggesting the two are miscible. Solvent two lies outside of the interaction radius, suggesting they are not miscible.

4.1 It can be seen that the HSP and the radius of the copolymers are taken as the weighted average of the homopolymers. The key is in the format of (percentage of homopolymer one) / (percentage of homopolymer two).

4.2 Monomer A is centered at (0,0,0) in Hansen Space with a molar volume of 194.8. Monomer B is centered at (6,4,0) with a molar volume of 304.5. Solvent A is centered at (-1,-6,0) and Solvent B centered at (4,7,0). The optimized blend is determined to be a 30.2:69.8 ratio of Solvent A to Solvent B. The area encompassed by the bold segments is the area where both monomers are soluble.

4.3 The area encompassed by the bold segments is the area where both monomers and the copolymer are soluble. Monomer A is centered at (0,0,0) in Hansen Space with a molar volume of 194.8. Monomer B is centered at (6,4,0) with a molar volume of 304.5. The CoPolymer is centered at (2,4,0) and has an $R_0$ of 4. Solvent A is centered at (-1,-6,0) and Solvent B centered at (4,7,0). The optimized blend is determined to be a 28.5:71.5 ratio of Solvent A to Solvent B.

4.4 One solvent is inside the common area, but the initial optimized blend is a represents a possible blend. The same monomers and copolymer are used as in Figure 4.3. Solvent A is located at (2.75,2.97,0) Slolvent B is located at (3.2,4.5,0) the calculated ratio of Solvent A to Solvent B is 91.9:8.1. It is important to note that this result produces the same point in Hansen Space as the conditions form Figure 4.3.
4.5 In this case, the initial optimized blend represented an impossible blend, thus the new optimization was calculated. Solvent A is centered at (2.675,3.555,0), the HSP of all other components are all equivalent to the HSP of Figure 4.4. It is important to note that the discarded solvent blend is in the same point of Hansen Space as the optimized blend of the previous two Figures.

4.6 Case 3.1 is an analogue of case 2.1, in that the initial optimized blend represents a possible solvent blend. The HSP of the monomers and copolymer are the same in this figure as in Figure 4.3. Solvent A is centered at (2.45,2.97,0) and Solvent B is centered at (3.75,2.95,0). The calculated blend is 58.2:41.8 ratio of Solvent A to Solvent B.

4.7 Represented is the only where both solvents are used as the endpoints of the optimized solvent search space. Also worth noting is that the procedures successfully discards the initial optimized blend. Solvent A is centered at (2.65,3.49,0) and Solvent B is centered at (2.8,3.88,0). The HSP of the monomers and copolymer are the same as Figure 4.3.

4.8 It is shown how multiple solutions can be obtained by testing multiple pairs of solvents. Areas not accessible by a single solvent blend are accessible by using multiple solvent pairs.

4.9 Two monomer HSP as well as their associated homopolymer HSP are given. The number in line three is the molar percentage of the first monomer in the input.

4.10 As many solvents available can be used. Each unique solvent pair will be tested producing many viable solvent blends.

4.11 Solvent blends can be adjusted after the fact for machine or human accuracy. All of these blends were tested to verify the results of the procedure.
CHAPTER I

INTRODUCTION

Choosing the correct solvent is a critical step in every polymerization. If a monomer is not fully soluble in the selected solvent, the reaction will take place in a heterogeneous fashion. Furthermore, if the resulting polymer is not fully soluble, the polymer will precipitate out. Whether a particular solvent will successfully solvate a monomer or polymer is based on the similarities in molecular composition and structure. Even if the molecular structure of solvent and solute are known, this observation is still too qualitative to make consistently correct decisions. The choice of solvents is further complicated when trying to synthesize a copolymer from multiple different monomers. Most solvent choices are based on personal experience or previously established standards, making it difficult to decide on a solvent for a completely new copolymer. These difficulties are compounded if the goal of an experiment is to make several new copolymers which have no previously established acceptable solvent.

The goal of the project funded by the W. M. Keck Foundation is to create a computer aided process which will produce upwards of fifty copolymers at a time, then systematically evaluate these polymers for a given material property. After evaluation a new batch of copolymers is created using the results from the previous experiment as basis for the new copolymers. With an extreme focus on
automation and quick turnaround, it is necessary to make the choice of solvents for each copolymer as quickly as possible. This emphasis makes the typical method of prior experience woefully inadequate for the choice of solvents. In order to ensure that picking solvents is not a rate-limiting step for each iteration of copolymer synthesis, it is necessary to create a computer program that can make solvent choices based on reliable and repeatable procedure.

Charles Hansen has created a set of parameters which mathematically describe solvents, monomers, and polymers in order to determine the quality of solvents.¹ The Hansen Solubility Parameters (HSP) take into account molecular structure, composition, and size in order to assign numeric values associated with these properties. While HSP do not have a strict theoretical basis, they are closely related to the Hildebrand Solubility Parameter which has an explicit thermodynamic explanation. While the Hildebrand Parameter gives a simple calculation for solubility, it fails to take into account molecular structure, and therefore is insufficient. HSP facilitate creating a numerical algorithm which can consistently determine the quality of solvents.
A frequent topic of discussion in the literature is determining the Hansen Solubility Parameters of polymers experimentally. To determine the HSP of a substance, one must observe the solubility of the given substance in a series of solvents that have wide variety of HSP. Then, by analyzing which solvents act as good quality and which solvents act as bad quality, one is able to triangulate the HSP of the desired substance. This process is not restricted to linear and highly crosslinked polymers, but also to molecules with highly ordered shapes such as fullerene. To be able to dissolve these carbon structures can lead to a wider range of uses. This highlights the usefulness of HSP for determination of solvent quality. This process is seen as so important that entire papers have been written on improvements made to the tools and methods used to determine HSP.

2.1 Chemical Protective Clothing

It is standard lab practice to wear Chemical Protective Clothing (CPC) while preforming any experiments in a lab setting. This is to prevent injury from the chemicals being used in an experiment or chemicals being used in other experiments in the same lab. Accidental exposure to chemicals can cause an acute condition known as contact dermatitis and accounts for almost ten percent
of all work occupational illnesses. Within the set of contact dermatitis injuries, hands are the most common area of the body to be affected. This makes research on gloves a very popular topic in the literature.

Viton gloves are made from a proprietary formulation which prevents researchers from being able to calculate their HSP with any method other than exposure to many solvents. To test these gloves, small samples are cut from a single glove and examined for defects. These samples are immersed in solvents for multiple weeks, measuring weight gain at regular intervals to determine solvent permeability. Examining the results of these experiments, the HSP of the gloves can be determined, which suggests what solvents are more likely to permeate the gloves without showing signs of damage. This method for determining solvent permeability has a clear advantage over other methods that test for diffusion coefficients. By finding the HSP, it is possible to make assessment about the permeability of solvents not tested in the experiment.

Butyl gloves are non-proprietary, and are produced by many companies. Studies similar to those of Viton gloves have been conducted to determine the HSP of butyl gloves. Samples of the butyl gloves were tested in a variety of solvents and their weight gained was measured. When testing different brands of the same product, it is found that the gloves have slight variations in their HSP. This information provided by the HSP can lead one to deduce that the formulations of these gloves are different for each brand. While the results from these studies showed that all of the gloves resisted similar solvents, one brand stood out among the rest as resisting solvent permeation at the highest level.
Similar studies have been conducted by the military to minimize chemical exposure to personnel, further exemplifying the wide uses of HSP.\textsuperscript{15}

2.2 Organogels

Gelation occurs when a solvent is trapped within a crosslinked polymer during polymerization.\textsuperscript{16} Organogels, while often a high percentage of solvent, can exhibit solid-like behavior without forming covalent crosslinks.\textsuperscript{17} Because solvents are contained within the crosslinks, drugs that are able to be dissolved by that solvent can also be contained within organogels. This had made organogels the subject of many studies involving drug delivery.\textsuperscript{18,19} These unique properties have made organogels, and their formation, a topic of great interest.

HSP are most applicable to organogels as they relate to formation of the gel networks. Instead of using the traditional solubility spheres, new gelation spheres are determined experimentally.\textsuperscript{20,21,22} While solubility spheres will be discussed more in-depth in a later chapter, gel spheres represent a volume in three dimensional space where the HSP of solvents will cause gelation to occur for a given substance.\textsuperscript{22,23} A simple example of gel spheres differ from solubility spheres involves two solvents of different chain length, but the with same functional group. For solubility you would expect the longer chain one would expect to see less of a dependence on polar, and hydrogen bonding parameters as van der Waals forces increase. The opposite is true for gel spheres as longer chains show I higher dependence on polar and hydrogen bonding parameters.\textsuperscript{22} This is an interesting use of HSP, but will not be utilized by the algorithm in this thesis.
2.3 Polymer Coatings

Polymer coatings cover a vast array of products, all of which depend on the quality of solvent used for the application process.\textsuperscript{24} Using HSP, good quality solvents can be found for each of the components of a coating, giving the highest chance that a good quality film will be formed.\textsuperscript{24,25} Because solvent quality is so important for producing a polymer coating, HSP can play a critical role in the choice of solvents.

A chemiresistor coating is used to detect organic vapors in given location. These sensors can provide simple electrical output that can be interpreted as easily understood data.\textsuperscript{26} There are many applications of chemiresistors including monitoring conditions, and industrial safety. In the study conducted by Eastman et al, monitoring for solvent spills was the main purpose for developing these chemiresistor coatings.\textsuperscript{26} The conclusion of this study found that similarities between solubility parameters of the coatings and target vapors resulted in more effective sensors.

Additives in polymer coatings are substances that do not contribute to the structural backbone of the coating, but do contribute to its end properties. Most commercial coatings employ additives and it is critical that these additives are universally dispersed throughout the film to ensure homogenous properties.\textsuperscript{27} Some additives serve cosmetic purposes, such as pigments and aluminum flakes in automotive coatings. Tetrabromobisphenol A is an additive that makes coatings more resistant to damage from open flames.\textsuperscript{27} Polymers are historically vulnerable to UV-radiation, such as sunlight. For coatings that are intended for
outdoor use it is highly advisable to include additives that mitigate these effects.\textsuperscript{27} In the case of any additive; HSP can be used to determine which solvents will be simultaneously appropriate for the coatings polymer backbone as well as the additives.\textsuperscript{27}

As similarly discussed earlier, HSP can also be used as a tool to find good quality solvents for new polymer coatings.\textsuperscript{28,29} Instead of completing an arduous trial and error process, HSP offers a good starting point to decide which solvents work best. In some cases, an analysis of HSP can reveal that certain intermolecular forces, such as hydrogen bonding forces, play a larger role in film formation than originally thought.\textsuperscript{30} HSP have been found to be useful in a plethora of applications, and have been extended far beyond there original use as tool to determine solubility.
CHAPTER III

BACKGROUND

3.1 Hildebrand Solubility Parameter

The Hildebrand Solubility Parameter is a measurable quantity of every molecule, solvents and polymers alike. The parameter $\delta$ can be calculated using equation (3.1). In equation (3.1), $\Delta H$ is the heat of vaporization, $R$ is the gas constant, $T$ is the absolute temperature, and $V_m$ is the molar volume.

$$\delta = \sqrt{\frac{\Delta H - RT}{V_m}} \quad (3.1)$$

When determining whether a polymer is miscible in a given solvent, equation (3.2) is used to see if the calculated $\chi$ is less than $\chi_c$ otherwise known as the critical $\chi$ value of a polymer solution. For an arbitrarily large polymer $\chi_c$ is .5.

$$\chi = \frac{V_m(\delta_1 - \delta_2)^2}{RT} \quad (3.2)$$

$$(\delta_1 - \delta_2)^2 = \frac{\chi RT}{V_m} = R_i \quad (3.3)$$

By rearranging equation (3.2) into (3.3), it can be seen that if $R_i$ is less than $\chi_c$, then the two substances are likely miscible. Using this equation we can create a one-dimensional interaction radius centered on the Hildebrand Parameter for a given polymer. If a solvent lies within the radius of a given polymer, the pair is likely to be miscible. An idealized example of this interaction...
radius is given in Figure 3.1. As stated before, the Hildebrand Parameter assumes all intermolecular interactions are van der Waals forces.

Figure 3.1 The polymer given in figure 3.1 has $\delta$ of 4 and an interaction radius of 4. Solvent one lies within the interaction radius, suggesting the two are miscible. Solvent two lies outside of the interaction radius, suggesting they are not miscible.

3.2 Hansen Solubility Parameters

Charles Hansen, through experimental observation, observed that Hildebrand Solubility Parameter could be improved on by creating sub-parameters relating to the structure of given molecules. The $\delta_0$ parameter is the parameter associated with dispersion forces, or the non-polar contributions to solubility. $\delta_p$ refers to solubility contributions from polar portions of a molecule brought on by asymmetries in molecular structure. $\delta_h$ parameter is associated with hydrogen bonding forces, or intermolecular interactions brought on by the large differences in electronegativity. These three parameters are directly related to the Hildebrand parameter and can easily be substituted in for the equation of miscibility as shown in equations 3.4 and 3.5 respectively.

$$\delta^2 = \delta_D^2 + \delta_P^2 + \delta_H^2 \ (3.4)$$
\[ \chi = \frac{V_m[(\delta_D_1 - \delta_D_2)^2 + (\delta_P_1 - \delta_P_2)^2 + (\delta_H_1 - \delta_H_2)^2]}{RT} \quad (3.5) \]

For calculating the miscibility of polymers in a given solvent, Hansen introduced a deviation in the standard procedure. Observing that dispersion forces carry high weight of importance for the solubility of polymers, he introduced a new set of equations to determine miscibility.

Hansen Space is a three-dimensional plot with perpendicular axes of \( \delta_D \), \( \delta_P \), and \( \delta_H \). Instead of a line segment on a number line, the shape of an interaction radius of a molecule in Hansen Space can be represented by a sphere centered at the HSP of the given Molecule in a three-dimensional space.\(^1\)

\[ 4(\delta_D_1 - \delta_D_2)^2 + (\delta_P_1 - \delta_P_2)^2 + (\delta_H_1 - \delta_H_2)^2 = R_a^2 \quad (3.6) \]

\[ \text{RED} = \frac{R_a}{R_o} \quad (3.7) \]

Equation (3.6) determines the Relative Affinity (\( R_a \)) of a polymer and solvent. Then the Relative Energy Difference (RED) is calculated in equation (3.7) as the quotient of \( R_a \) and \( R_o \), where \( R_o \) is an experimentally derived parameter of a polymer for arbitrarily high molecular weight. If the RED is less than one, then the polymer and solvent are likely miscible. With this relationship, and combining equations (3.6) and (3.7), equation (3.8) is derived. Equation (3.8) describes all points in Hansen space that describe molecule miscible with a given polymer. As opposed to the non-polymer HSP equation, equation (3.8) describes an ellipsoid in Hansen Space.\(^1,3\)

\[ \frac{4(\delta_D_1 - \delta_D_2)^2}{(R_o)^2} + \frac{(\delta_P_1 - \delta_P_2)^2}{(R_o)^2} + \frac{(\delta_H_1 - \delta_H_2)^2}{(R_o)^2} < 1 \quad (3.8) \]
It is critical to mention that Hansen describes the HSP of a blend of solvents a simple molar weighted average between the HSP of the two solvent components.\textsuperscript{1,3} To calculate the HSP of the copolymers, a similar strategy will be employed. First, the HSP for the homopolymers of the constituent monomers will be obtained from Hansen's handbook.\textsuperscript{1} Then, the weighted average of the homopolymers HSP will be taken as the HSP for the copolymer with the weights being based on the molar percentages of the monomers.

3.3 Group Contribution Theory.

In order to take into account the effects that each functional group has on its molecules solubility, HSP employs Group Contribution Theory (GCT). GCT is the idea that every functional group has its own solubility properties and that a molecules solubility properties is a combination of its functional groups. Dispersion forces area simple some of each component divided by the molar volume of the molecule given in equation (3.9). $F_{Di}$ is the contribution of the $i_{th}$ functional group.\textsuperscript{31}

$$\delta_D = \frac{\Sigma F_{Di}}{V_m} \quad (3.9)$$

The calculation of the polar parameters and hydrogen bonding parameters differs slightly from the dispersion calculations and are given in equations (3.10) and (3.11) respectively. Additionally, if there are identical functional groups that are distributed symmetrically across the molecule, there are additional coefficients that need to be taken into account. For polar forces, there is a coefficient of .5 is for a single-plane symmetry, .25 for two-plane symmetry, and 0
for more than two planes of symmetry.\textsuperscript{31} Hydrogen forces only have an additional coefficient of 0 if there is more than one plane of symmetry.\textsuperscript{31}

\[ \delta_P = \sqrt{\frac{\sum F^2_{Pi}}{V_m}} \] (3.10)  \[ \delta_H = \sqrt{\frac{\sum F_{Hi}}{V_m}} \] (3.11)

Hansen provides an extensive list of HSP for solvents and polymers, but this list is far from comprehensive.\textsuperscript{1} With GCT, it is possible to calculate the HSP of molecule with a known composition and structure. Hansen’s handbook contains the solubility contributions of several common functional groups for this purpose. Further work could involve a program that determines the HSP of molecules based on its structure.
CHAPTER IV
APPLICATION OF HSP

4.1 Derivation of the Algorithm

The goal of the procedure is to determine an optimized blend of solvents that could simultaneously dissolve both monomers and the resulting copolymer in order to ensure that polymerization can begin and go to completion. A blend of solvents is necessary because it is highly unlikely that single point in Hansen space, the HSP of a single solvent, will be sufficient to dissolve every possible copolymer system. When using a blend of solvents, every possible HSP of the blend can be represented as points on a line segment between each solvent. The HSP lie on the line segment due to the simple nature of determining HSP for solvent blends, as described by Hansen.\(^1\) An example of how the HSP would be calculated is given in equation (4.1). The coefficient \(x_1\) represents the molar percentage of solvent one in the solvent blend and \((1 - x_1)\) represents the molar percentage of solvent two in the solvent blend.

\[
\delta_{D_{\text{total}}} = x_1 \delta_{D_{1}} + (1 - x_1) \delta_{D_{2}} \quad (4.1)
\]

\[
4 \left( \frac{\delta_{D} - (x_1 \delta_{D_{1}} + (1 - x_1) \delta_{D_{2}})^2}{(R_o)^2} \right) + \frac{\delta_{P} - (x_1 \delta_{P_{1}} + (1 - x_1) \delta_{P_{2}})^2}{(R_o)^2} + \frac{\delta_{H} - (x_1 \delta_{H_{1}} + (1 - x_1) \delta_{H_{2}})^2}{(R_o)^2} < 1 \quad (4.2)
\]

Per equations (3.8) and (4.1), equation (4.2) represents all of the possible points in a blend of solvents that are miscible with a given polymer. By expanding
the quadranomial of the dispersion forces in equation (4.2) we obtain equation (4.3a) and a simplified version in equation (4.3b).

\[ x_1^2(\delta_{D1}^2 - 2\delta_{D1}\delta_{D2} + \delta_{D2}^2) + x_1(-2\delta_{Dm}\delta_{D1} + 2\delta_{Dm}\delta_{D2} + 2\delta_{D1}\delta_{D2} - 2\delta_{D2}^2) + (\delta_{Dm}^2 - 2\delta_{Dm}\delta_{D2} + \delta_{D2}^2) \]  

(4.3a)

\[ x_1^2\Delta_{D2} + x_1\Delta_{D1} + \Delta_{D0} \]  

(4.3b)

By applying the same expansion to the polar and hydrogen bonding forces and summing up all of the analogous (4.3b) equations, equation (4.4) is produced.

\[ x_1^2(4\Delta_{D2} + \Delta_{P2} + \Delta_{H2}) + x_1(4\Delta_{D1} + \Delta_{P1} + \Delta_{H1}) + (4\Delta_{D0} + \Delta_{P0} + \Delta_{H0} - R_0^2) = 0 \]  

(4.4)

This equation is a quadratic with respects to \( x_1 \) and can be solved to find the two points of intersection between the solvent blend and the polymer ellipsoid. All of the points between the two intersections are miscible blends of solvent with the polymer. It is necessary to note that \( x_1 \) values that are not between zero and one do not make physical sense. Such an alpha would represent a solvent blend that consists of over one hundred percent of one solvent and a negative percentage of the other.

Similar calculations can be used to determine where a solvent blend will be miscible with the monomers. By combining equations (3.5) and (4.1), an analogous calculation will be produced and solved in the same fashion. In order to find the section of the solvent blend that is miscible with the monomers and the resulting copolymer, each intersection point will be checked if it within the other two volumes in Hansen Space. For every real solution, two of these intersection points will be found and the midpoint between them will be used as the optimal blend of solvents to solvate the system.
4.2 Cases that Produce Valid Solvent Blends

For the purposes of all examples, the $\delta_H$ value of all components will be zero. Setting $\delta_H$ to zero will have no effect on the accuracy of the procedure. The purpose of setting all $\delta_H$ values at zero is to make visualization of the procedure easier to comprehend as it makes the process effectively two-dimensional. The $\delta_P-\delta_D$ plane will be displayed in all following figures. Also the temperature is assumed to be 293 K. Figure 4.1 illustrates how this simplification makes the procedure easier to interpret as well as gives a visualization of the aforementioned process of determining the HSP for copolymers.

![Figure 4.1](image)

**Figure 4.1** It can be seen that the HSP and the radius of the copolymers are taken as the weighted average of the homopolymers. The key is in the format of (percentage of homopolymer one) / (percentage of homopolymer two).

**Figure 4.2** illustrates a ternary system consisting of a monomer and two solvents. The ternary system is a precursor to the quinary system, and depicts a
how the miscible solvent blends are chosen. This process is repeated for the second monomer as well as the polymer. The calculations are combined to determine where the solvent blend intersects the shared volume.

![Diagram](image)

**Figure 4.2** Monomer A is centered at (0,0,0) in Hansen Space with a molar volume of 194.8, Solvent A is centered at (-1,-6, 0) and Solvent B centered at (4,7, 0). The area encompassed by the bold segments is the area where monomer is soluble.

4.2.1 Case 1

The simplest case is when the two solvents lie outside of the area where the two monomers and the resulting copolymer are all soluble. If the line segment of the solvent blend intersects the common area, then the optimized blend is chosen as the midpoint between the two points of intersection with the common area. Whether the points of intersection with the common area are on both monomers, or one monomer, has no bearing on the determination of the optimal solvent blend. Figure 4.3 represents the simplest, quinary case.
Figure 4.3 The area encompassed by the bold segments is the area where both monomers and the copolymer are soluble. Monomer A is centered at (0,0,0) in Hansen Space with a molar volume of 194.8. Monomer B is centered at (6,4,0) with a molar volume of 304.5, The CoPolymer is centered at (2,4,0) and has an $R_o$ of 4. Solvent A is centered at (-1,-6,0) and Solvent B centered at (4,7,0). The optimized blend is determined to be a 28.5:71.5 ratio of Solvent A to Solvent B.

4.2.2 Case 2

Case 2 considers the scenario where only one of the solvents lies outside of the area of common solubility. Case 2 is broken up into two subcases based on how close the solvent within the common area of solubility is to the solute surfaces. With one solvent inside of the common area, one of the calculated points of intersection will represent an impossible blend of solvents that is over one hundred percent of one solvent and a negative percent of the other. Using the midpoint between the two intersection points may or may not yield an impossible blend, creating two distinct sub cases.
4.2.2.1 Case 2.1

When the procedure encounters a scenario that falls under case 2, it initially does not apply any additional steps in determining the optimal solvent blends, even though there is the potential to produce an impossible blend. That is because there are cases such as subcase 2.1 where the normal procedure produces an acceptable result. This result will be farther from the common surface than a midpoint between the inclusive solvent and the intersection with the surface. Figure 4.4 illustrates such a subcase. In order to emphasize the important differences between sub case 2.1 and other subcases of case 2, the window of this figure is much smaller and encompasses only the common area.

![Diagram](image)

Figure 4.4 One solvent is inside the common area, but the initial optimized blend is a represents a possible blend. The same monomers and copolymer are used as in Figure 4.3. Solvent A is located at (2.75,2.97,0) Solvent B is located at (3.2,4.5,0) the calculated ratio of Solvent A to Solvent B is 91.9:8.1. It is important to note that this result produces the same point in Hansen Space as the conditions form Figure 4.3.
4.2.2.2 Subcase 2.2

Subcase 2.2 differs from subcase 2.1 in that the midpoint between points of intersection is not a possible blend of solvents. The procedure has subroutine built in for just such a case. The procedure is able to detect that a non-valid blend has been calculated, and recalculate the optimal solvent as the pure solvent that lies inside the shared volume. This subcase is illustrated in Figure 4.5.

Figure 4.5 In this case, the initial optimized blend represented an impossible blend, thus the new optimization was calculated. Solvent A is centered at (2.675, 3.555, 0), the HSP of all other components are all equivalent to the HSP of Figure 4.4. It is important to note that the discarded solvent blend is in the same point of Hansen Space as the optimized blend of the previous two Figures.
4.2.3 Case 3

Case 3 represents the scenario when both solvents are included inside the common area. As in case 2, case 3 is broken up into two subcases, depending on if the initial optimized blend lies within the range of possible solvent blends.

4.2.3.1 Subcase 3.1

In this subcase the initial optimized blend lies within the possible blend of solvents. The initial optimized blend will be better suited to dissolve the system throughout polymerization as it lies as far away from the common area surface as possible along the solvent blend line segment. This case is illustrated in Figure 4.6.

Figure 4.6 Case 3.1 is an analogue of case 2.1, in that the initial optimized blend represents a possible solvent blend. The HSP of the monomers and copolymer are the same in this figure as in Figure 4.3. Solvent A is centered at (2.45,2.97,0) and Solvent B is centered at (3.75,2.95,0). The calculated blend is 58.2:41.8 ratio of Solvent A to Solvent B.
4.2.3.2 Subcase 3.2

This subcase is similar to subcase 2.2 in that the initial calculation for the optimized blend is outside of the possible blend of solvents. Because there are two solvents in the shared volume, the subroutine from case 2.2 is insufficient. The distance between the discarded blend and each pure solvent will be calculated and the pure solvent with the shortest distance will be chosen as the optimal solvent. This case is illustrated in Figure 4.7.

Figure 4.7 Represented is the only where both solvents are used as the endpoints of the optimized solvent search space. Also worth noting is that the procedures successfully discards the initial optimized blend. Solvent A is centered at (2.65, 3.49, 0) and Solvent B is centered at (2.8, 3.88, 0). The HSP of the monomers and copolymer are the same as Figure 4.3.

4.3 Solvent Bank

It is likely that two solvents and their resulting blends will not be sufficient to solvate every possible copolymer system. In order to increase the likelihood
that an optimal solvent blend will be obtained, multiple pairs of solvents will be tested with the procedure for each copolymer system. The selection of solvents will be referred to as the solvent bank. The first step in this system will be deciding which solvent pairs are miscible with each other. Each miscible pair will then be used in the normal procedure and all optimal blends will be output. This procedure is illustrated in Figure 4.8

![Diagram showing solvent miscibility](image)

Figure 4.8 It is shown how multiple solutions can be obtained by testing multiple pairs of solvents. Areas not accessible by a single solvent blend are accessible by using multiple solvent pairs.

4.4 Experimental Results

With all of the cases mapped out and tested the last step in verifying the procedure is in lab experimental results. Two commonly available monomers were chosen and a bank of commonly available solvents was used to increase the odds that an acceptable blend was produced. Each unique solvent pair is
tested with the two monomers making it possible that multiple solvent blends are outputted by the program. Figures 3.9. and 4.10 are examples of the documents used as inputs for the program. Figure 4.11 is an example output.

Figure 4.9 Two monomer HSP as well as their associated homopolymer HSP are given. The number in line three is the molar percentage of the first monomer in the input.

Figure 4.10 As many solvents available can be used. Each unique solvent pair will be tested producing many viable solvent blends.

Figure 4.11 Solvent blends can be adjusted after the fact for machine or human accuracy. All of these blends were tested to verify the results of the procedure.

The results from the lab experiment were a resounding success. With this final piece of verification, it is observed that not only is the procedure successful in idealized test cases, but also in real life examples. The experimental results are summarized in Table 1.
Table 1 Volumes of each solvent and monomer were calculated from the output of the procedure given in Figure 4.11. 3 mL total of monomer were used and 5 mL total of Solvent was used in each test.

<table>
<thead>
<tr>
<th>Solutions</th>
<th>Styrene</th>
<th>Vinyl acetate</th>
<th>Cholorobenzene</th>
<th>Dioaxane</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume (mL)</td>
<td>1.5</td>
<td>1.5</td>
<td>3.6</td>
<td>1.4</td>
</tr>
<tr>
<td>Results</td>
<td></td>
<td></td>
<td>No Precipitate</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solutions</th>
<th>Styrene</th>
<th>Vinyl acetate</th>
<th>Cholorobenzene</th>
<th>Toluene</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume (mL)</td>
<td>1.5</td>
<td>1.5</td>
<td>0.3</td>
<td>4.7</td>
</tr>
<tr>
<td>Results</td>
<td></td>
<td></td>
<td>No Precipitate</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solutions</th>
<th>Styrene</th>
<th>Vinyl acetate</th>
<th>Cholorobenzene</th>
<th>Hexane</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume (mL)</td>
<td>1.5</td>
<td>1.5</td>
<td>4.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Results</td>
<td></td>
<td></td>
<td>No Precipitate</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solutions</th>
<th>Styrene</th>
<th>Vinyl acetate</th>
<th>Toluene</th>
<th>Dioaxane</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume (mL)</td>
<td>1.5</td>
<td>1.5</td>
<td>4.8</td>
<td>0.2</td>
</tr>
<tr>
<td>Results</td>
<td></td>
<td></td>
<td>No Precipitate</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER V
CONCLUSION

The goal of creating an Algorithm using the Hansen Solubility parameters has been derived and verified. With this procedure in place it will be possible to quickly and accurately determine optimal solvent blends. By providing a mathematical description based on molecular structure, HSP has facilitated the development of a consistent and successful procedure. Despite the lack of a theoretical basis for HSP, it is shown that they are effective and accurate in determining the miscibility of solvents and polymers.

Automating this process will eliminate one of the many rate limiting steps of the high throughput Keck Foundation project. The end product of the Keck project will be able to create and analyze new copolymers at unprecedented speeds. To contribute to this project requires a novel approach to determining solvent blends, as given in this thesis. Without this procedure, it would not be possible to run through an iteration of copolymers quickly and efficiently.

It has been shown that HSP have a wide range of applications across the field of polymers. From coatings to gloves, HSP plays a relevant role in determining key characteristics of these systems\textsuperscript{12,27}. The HSP provides a much needed quantification of molecular structure as it relates to solubility.
There is a known improvement to the procedure that can be implemented but is left out to lower the time it takes for the procedure to produce results. These improvements involve calculating distances from the surface of the polymer ellipsoids in Hansen Space which is computationally expansive. Future work can be done to improve this procedure when speed of results is not a primary focus.
REFERENCES


APPENDIX

FULL ALGORITHM WRITTEN IN C++, COMPILED IN A LINUX ENVIRONMENT

```cpp
#include<iostream>
#include<math.h>
#include<vector>
#include<string>
#include<fstream>
#include<sstream>
using namespace std;

/**********************************************************
* 1. Creating classes for each coponent    *
*       *
*       *
*       *
*       *
/**********************************************************/

// Monomers contain the HSP for a Monomer as well as its homopolymer.
// This allows us to create a variable to store all of the data.
class Monomer{
    double MonoDel_D,MonoDel_H,MonoDel_P,Mono_Volm,PolyDel_D,PolyDel_H,PolyDel_P,PolyR_not;
    string Mono_name,Poly_name;
    public:
        void Mono_assign(string,double,double,double,double,string,double,double,double,double);
        string Give_Mono_N () {return Mono_name;}
        double Give_Mono_D () {return MonoDel_D;}
        double Give_Mono_H () {return MonoDel_H;}
        double Give_Mono_P () {return MonoDel_P;}
        double Give_Mono_V () {return Mono_Volm;}
        string Give_Poly_N () {return Poly_name;}
        double Give_Poly_D () {return PolyDel_D;}
        double Give_Poly_H () {return PolyDel_H;}
        double Give_Poly_P () {return PolyDel_P;}
        double Give_Poly_R () {return PolyR_not;}
};

// The HSP of solvent is stored similarly to a monomer
class Solvent{
    double SolvDel_D,SolvDel_H,SolvDel_P,Solv_Volm;
};
```
string Solv_name;
public:
    void Solv_assign (string,double,double,double,double);
string Give_Solv_N () {return Solv_name;}
double Give_Solv_D () {return SolvDel_D;}
double Give_Solv_H () {return SolvDel_H;}
double Give_Solv_P () {return SolvDel_P;}
double Give_Solv_V () {return Solv_Volm;}
};

// The HSP of polymer is stored similarly to a monomer
class Polymer{
    double PolyDel_D,PolyDel_H,PolyDel_P,PolyR_not;
    string Poly_name;
public:
    void Poly_assign (string,double,double,double,double);
    string Give_PolyN () {return Poly_name;}
    double Give_PolyD () {return PolyDel_D;}
    double Give_PolyH () {return PolyDel_H;}
    double Give_PolyP () {return PolyDel_P;}
    double Give_PolyR () {return PolyR_not;}
};

/*****************************************
**                                      *
*    2. Prototypes for each function    *
*                                      *
*****************************************/
void QuinaryMixer(Monomer,Monomer,Solvent,Solvent,double);
bool SolventMiscibility(Solvent,Solvent);
Polymer HomoPolymerize (Monomer);
Polymer CoPolymerize (Polymer,Polymer,double);
double CapDelTwo(double,double);
double CapDelOne(double,double,double);
double CapDelNot(double,double);
double MCheck(double,Monomer,Solvent,Solvent);
double PCheck(double,Polymer,Solvent,Solvent);

/*******************************************
3. Main Function
   Takes inputs for file names
   Initializes all monomers and solvents
   Runs all solvent pairs

```cpp
int main(){
    int Num_of_Solv;
    string InputOne,InputTwo;
    Monomer Chosen_Mono_one,Chosen_Mono_two;
    double Mono_one_Percent,Mix;

    // location of monomer file
    string Mono_file = "/home/thomashoy/Documents/";

    // acquire file name
    cout << "Monomer File name:";
    cin >> InputOne;
    Mono_file = Mono_file + InputOne;

    ifstream ReadMonoFile(Mono_file.c_str());
    string lineContentOne;
    string Mono_N,Poly_N;
    int n = 1;

    // loop that initializes monomers form input file
    if (ReadMonoFile.is_open()) {
        while(getline(ReadMonoFile,lineContentOne)) {
            istringstream iss(lineContentOne);
            if(n == 1){
                iss >> Mono_N;
                iss >> Mono_D;
                iss >> Mono_P;
                iss >> Mono_H;
                iss >> Mono_V;
                iss >> Poly_N;
                iss >> Poly_D;
                iss >> Poly_P;
                iss >> Poly_H;
                iss >> Poly_R;
            }
        }
    }
```

else if(n == 2) {
    iss >> Mono_N;
    iss >> Mono_D;
    iss >> Mono_P;
    iss >> Mono_H;
    iss >> Mono_V;
    iss >> Poly_N;
    iss >> Poly_D;
    iss >> Poly_P;
    iss >> Poly_H;
    iss >> Poly_R;
    Chosen_Solvent.Mono_assign
        (Mono_N,Mono_D,Mono_P,Mono_H,Mono_V,Poly_N, Poly_D,Poly_P,Poly_H,Poly_R);
}
else if(n == 3) {
    iss >> Mix;
}
else {

}
n++;
}
ReadMonoFile.close();

// Solvent Bank is stored as a vector to ensure any amount of solvents can be used
vector<Solvent> Solv_Bank;
Solvent Solv_hold;

// location of solvent file
string Solv_file = "/home/thomashoy/Documents/";

// location of Solvent file
cout << "Solvent File Name:";
cin >> InputTwo;
Solv_file = Solv_file + InputTwo;

ifstream ReadSolvFile(Solv_file.c_str());
if (ReadSolvFile.is_open()) {
    string lineContentTwo;
string Solv_N;
double Solv_D,Solv_P,Solv_H,Solv_V;

// loop that initializes solvents
while(getline(ReadSolvFile,lineContentTwo)) {
    istringstream iss(lineContentTwo);
    iss >> Solv_N;
    iss >> Solv_D;
    iss >> Solv_P;
    iss >> Solv_H;
    iss >> Solv_V;

    Solv_hold.Solv_assign(Solv_N,Solv_D,Solv_P,Solv_H,Solv_V);
    Solv_Bank.push_back(Solv_hold);
}

ReadSolvFile.close();
else {
    cout << "Solvent file not open." << "n";
}

Mono_one_Percent = Mix/100;
cout << endl;

Num_of_Solv = Solv_Bank.size();

// nested loop runs all pairs of solvent bank
for(int i = 0;i < Num_of_Solv-1;i++){
    for(int j = i+1;j < Num_of_Solv;j++){
        if(SolventMiscibility(Solv_Bank[i],Solv_Bank[j])){
            QuinaryMixer
            (Chosen_Mono_one,Chosen_Mono_two,Solv_Bank[i],Solv_Bank[j],Mono_one_Percent);
        }
    else{

        }
    }
}

/******************************************************************************/
void QuinaryMixer(Monomer Chosen_Mono_one, Monomer Chosen_Mono_two, Solvent Solv_one, Solvent Solv_two, double Mono_one_Percent) {
    bool FinalOne;
    double
    Chosen_Mono_one_D, Chosen_Mono_one_H, Chosen_Mono_one_P, Chosen_Mono_one_V;
    double
    Chosen_Mono_two_D, Chosen_Mono_two_H, Chosen_Mono_two_P, Chosen_Mono_two_V;

    // obtaining the HSP of all the monomers
    Chosen_Mono_one_D = Chosen_Mono_one.Give_Mono_D();
    Chosen_Mono_one_H = Chosen_Mono_one.Give_Mono_H();
    Chosen_Mono_one_P = Chosen_Mono_one.Give_Mono_P();
    Chosen_Mono_one_V = Chosen_Mono_one.Give_Mono_V();

    Chosen_Mono_two_D = Chosen_Mono_two.Give_Mono_D();
    Chosen_Mono_two_H = Chosen_Mono_two.Give_Mono_H();
    Chosen_Mono_two_P = Chosen_Mono_two.Give_Mono_P();
    Chosen_Mono_two_V = Chosen_Mono_two.Give_Mono_V();

    double Solv_one_D, Solv_one_H, Solv_one_P;
    double Solv_two_D, Solv_two_H, Solv_two_P;
    string Solv_one_N, Solv_two_N;

    // obtaining the HSP of all solvents
    Solv_one_D = Solv_one.Give_Solv_D();
    Solv_one_H = Solv_one.Give_Solv_H();
    Solv_one_P = Solv_one.Give_Solv_P();
    Solv_one_N = Solv_one.Give_Solv_N();

    Solv_two_D = Solv_two.Give_Solv_D();
    Solv_two_H = Solv_two.Give_Solv_H();
    Solv_two_P = Solv_two.Give_Solv_P();
    Solv_two_N = Solv_two.Give_Solv_N();

    double Poly_D, Poly_H, Poly_P, Poly_R;
    Polymer HomoPoly_one, HomoPoly_two, CoPoly;
    HomoPoly_one = HomoPolymerize(Chosen_Mono_one);
    HomoPoly_two = HomoPolymerize(Chosen_Mono_two);

    // initializing copolymer HSP
    if (Mono_one_Percent == 1) {
        CoPoly = HomoPoly_one;
    }
}
else{
    CoPoly = CoPolymerize(HomoPoly_one,HomoPoly_two,Mono_one_Percent);
}

//obtaining copolymer HSP
Poly_D = CoPoly.Give_PolyD();
Poly_H = CoPoly.Give_PolyH();
Poly_P = CoPoly.Give_PolyP();
Poly_R = CoPoly.Give_PolyR();

// calculating coefficients for quadratic formula
double Cap_Del_Dtwo_Mone,Cap_Del_Done_Mone,Cap_Del_Dnot_Mone;
double Cap_Del_Dtwo_Mtwo,Cap_Del_Done_Mtwo,Cap_Del_Dnot_Mtwo;
double Cap_Del_Dtwo_P,Cap_Del_Done_P,Cap_Del_Dnot_P;

double Cap_Del_Dtwo_Mone = CapDelTwo(Solv_one_D,Solv_two_D);
double Cap_Del_Done_Mone = CapDelOne(Solv_one_D,Solv_two_D,Chosen_Mono_one_D);
double Cap_Del_Dnot_Mone = CapDelNot(Solv_two_D,Chosen_Mono_one_D);
double Cap_Del_Dtwo_Mtwo = CapDelTwo(Solv_one_D,Solv_two_D);
double Cap_Del_Done_Mtwo = CapDelOne(Solv_one_D,Solv_two_D,Poly_D);
double Cap_Del_Dnot_Mtwo = CapDelNot(Solv_two_D,Poly_D);

double Cap_Del_Dtwo_P = CapDelTwo(Solv_one_D,Solv_two_D);
double Cap_Del_Done_P = CapDelOne(Solv_one_D,Solv_two_D, Poly_D);
double Cap_Del_Dnot_P = CapDelNot(Solv_two_D, Poly_D);
\[
\text{Cap}_\text{Del}_\text{Htwo}_\text{P} = \text{CapDelTwo}(\text{Solv}_\text{one}_\text{H}, \text{Solv}_\text{two}_\text{H}); \\
\text{Cap}_\text{Del}_\text{Hone}_\text{P} = \text{CapDelOne}(\text{Solv}_\text{one}_\text{H}, \text{Solv}_\text{two}_\text{H}, \text{Poly}_\text{H}); \\
\text{Cap}_\text{Del}_\text{Hnot}_\text{P} = \text{CapDelNot}(\text{Solv}_\text{two}_\text{H}, \text{Poly}_\text{H}); \\
\text{Cap}_\text{Del}_\text{Ptwo}_\text{P} = \text{CapDelTwo}(\text{Solv}_\text{one}_\text{P}, \text{Solv}_\text{two}_\text{P}); \\
\text{Cap}_\text{Del}_\text{Pone}_\text{P} = \text{CapDelOne}(\text{Solv}_\text{one}_\text{P}, \text{Solv}_\text{two}_\text{P}, \text{Poly}_\text{P}); \\
\text{Cap}_\text{Del}_\text{Pnot}_\text{P} = \text{CapDelNot}(\text{Solv}_\text{two}_\text{P}, \text{Poly}_\text{P}); \\
\]

\[
\text{double CorrectA, CorrectB, CorrectC;}
\text{double A\_one, A\_two, A\_P, B\_one, B\_two, B\_P, C\_one, C\_two, C\_P, D\_one, D\_two, D\_P;}
\text{double Alpha\_one, Alpha\_two, Alpha\_thr, Alpha\_for, Alpha\_fiv, Alpha\_six;}
\text{double intercept\_one, intercept\_two, temp\_comp, final\_comp;}
\text{intercept\_one = 1;}
\text{intercept\_two = 0;}
\text{double Chosen\_Mono\_one\_R = 2*8.314*293/Chosen\_Mono\_one\_V;}
\text{double Chosen\_Mono\_two\_R = 2*8.314*293/Chosen\_Mono\_two\_V;}
\text{A\_one = Cap\_Del\_Dtwo\_Mone + Cap\_Del\_Htwo\_Mone + Cap\_Del\_Ptwo\_Mone;}
\text{B\_one = Cap\_Del\_Done\_Mone + Cap\_Del\_Hone\_Mone + Cap\_Del\_Pone\_Mone;}
\text{C\_one = Cap\_Del\_Dnot\_Mone + Cap\_Del\_Hnot\_Mone + Cap\_Del\_Pnot\_Mone - Chosen\_Mono\_one\_R;}
\text{// continuing to calculate coefficients for quadratic formula}
\text{A\_two = Cap\_Del\_Dtwo\_Mtwo + Cap\_Del\_Htwo\_Mtwo + Cap\_Del\_Ptwo\_Mtwo;}
\text{B\_two = Cap\_Del\_Done\_Mtwo + Cap\_Del\_Hone\_Mtwo + Cap\_Del\_Pone\_Mtwo;}
\text{C\_two = Cap\_Del\_Dnot\_Mtwo + Cap\_Del\_Hnot\_Mtwo + Cap\_Del\_Pnot\_Mtwo - Chosen\_Mono\_two\_R;}
\text{A\_P = 4*Cap\_Del\_Dtwo\_P + Cap\_Del\_Htwo\_P + Cap\_Del\_Ptwo\_P;}
\text{B\_P = 4*Cap\_Del\_Done\_P + Cap\_Del\_Hone\_P + Cap\_Del\_Pone\_P;}
\text{C\_P = 4*Cap\_Del\_Dnot\_P + Cap\_Del\_Hnot\_P + Cap\_Del\_Pnot\_P - Poly\_R;}
\text{// results of quadratic formula}
\text{D\_one = B\_one*B\_one - 4*A\_one*C\_one;}
\text{if (D\_one >= 0){}
\text{Alpha\_one = (-B\_one + sqrt(D\_one))/(2*A\_one);}
\text{Alpha\_two = (-B\_one - sqrt(D\_one))/(2*A\_one);}
\text{}}
\text{else{}
\text{return ;}
\text{}}
\text{D\_two = B\_two*B\_two -4*A\_two*C\_two;}
\text{if (D\_two >= 0){}
\text{Alpha\_thr = (-B\_two + sqrt(D\_two))/(2*A\_two);}
\text{Alpha\_for = (-B\_two - sqrt(D\_two))/(2*A\_two);}
\text{}}
\text{else{}
\text{return ;}
\text{}}
\[ D_P = B_P \cdot B_P - 4 \cdot A_P \cdot C_P; \]

if \((D_P \geq 0)\) {
    \[
    \text{Alpha_fiv} = \left(-B_P + \sqrt{D_P}\right)/(2 \cdot A_P); \\
    \text{Alpha_six} = \left(-B_P - \sqrt{D_P}\right)/(2 \cdot A_P); 
    \]
} 
else{
    return ;
}

// finding intersection of solvent blend with shared volume
for(int i = 0; i < 6; i++){
    double CheckOne = 100;
    double CheckTwo = 100;
    double CheckP = 100;
    if(i == 0){
        CheckTwo = MCheck(Alpha_one, Chosen_Mono_two, Solv_one, Solv_two);
        CheckP = PCheck(Alpha_one, CoPoly, Solv_one, Solv_two);
        if(CheckTwo < 1 && CheckP < 1){
            intercept_one = Alpha_one;
        } 
    } 
    else{
    }
}
if(i == 1){
    CheckTwo = MCheck(Alpha_two, Chosen_Mono_two, Solv_one, Solv_two);
    CheckP = PCheck(Alpha_two, CoPoly, Solv_one, Solv_two);
    if(CheckTwo < 1 && CheckP < 1 && intercept_one == 1){
        intercept_one = Alpha_two;
    }
    else if(CheckTwo < 1 && CheckP < 1){
        intercept_two = Alpha_two;
    }
    else{
    }
}
if(i == 2){
    CheckOne = MCheck(Alpha_thr, Chosen_Mono_one, Solv_one, Solv_two);
    CheckP = PCheck(Alpha_thr, CoPoly, Solv_one, Solv_two);
    if(CheckOne < 1 && CheckP < 1 && intercept_one == 1){
    }
intercept_one = AlphaThr;
}
else if(CheckOne < 1 && CheckP < 1){
    intercept_two = AlphaThr;
}
else{
    }
}
if(i == 3){
    CheckOne = MCheck(Alpha_for,Chosen_Mono_one,Solv_one,Solv_two);
    CheckP = PCheck(Alpha_for,CoPoly,Solv_one,Solv_two);
    if(CheckOne < 1 && CheckP < 1 && intercept_one == 1){
        intercept_one = Alpha_for;
    }
    else if(CheckOne < 1 && CheckP < 1){
        intercept_two = Alpha_for;
    }
    else{
    }
}
if(i == 4){
    CheckOne = MCheck(Alpha_fiv,Chosen_Mono_one,Solv_one,Solv_two);
    CheckTwo = MCheck(Alpha_fiv,Chosen_Mono_two,Solv_one,Solv_two);
    if(CheckOne < 1 && CheckTwo < 1 && intercept_one == 1){
        intercept_one = Alpha_fiv;
    }
    else if(CheckOne < 1 && CheckTwo < 1){
        intercept_two = Alpha_fiv;
    }
    else{
    }
}
if(i == 5){
    CheckOne = MCheck(Alpha_six,Chosen_Mono_one,Solv_one,Solv_two);
    CheckTwo = MCheck(Alpha_six,Chosen_Mono_two,Solv_one,Solv_two);
    if(CheckOne < 1 && CheckTwo < 1 && intercept_one == 1){
        intercept_one = Alpha_six;
    }
    else if(CheckOne < 1 && CheckTwo < 1){
        intercept_two = Alpha_six;
    }
    else{
    }
}
if(intercept_one != 1 && intercept_two == 0){
  CorrectA = MCheck(1,Chosen_Mono_one,Solv_one,Solv_two);
  CorrectB = MCheck(1,Chosen_Mono_one,Solv_one,Solv_two);
  CorrectC = PCheck(1,CoPoly,Solv_one,Solv_two);

  if(CorrectA < 1 && CorrectB < 1 && CorrectC < 1){
    intercept_two = 1;
  }
  else{
  }
}
else{
}

temp_comp = (intercept_one + intercept_two)/2;

// failsafe to prevent suggestion of impossible solvent blend
if(temp_comp < 1 && temp_comp > 0){
  final_comp = temp_comp;
}
else{

double SolvOneCheckOne,SolvOneCheckTwo,SolvOneCheckPoly;
double SolvTwoCheckOne,SolvTwoCheckTwo,SolvTwoCheckPoly;
bool OneCheck,TwoCheck;

SolvOneCheckOne = MCheck(1,Chosen_Mono_one,Solv_one,Solv_two);
SolvOneCheckTwo = MCheck(1,Chosen_Mono_one,Solv_one,Solv_two);
SolvOneCheckPoly = PCheck(1,CoPoly,Solv_one,Solv_two);

SolvTwoCheckOne = MCheck(0,Chosen_Mono_one,Solv_one,Solv_two);
SolvTwoCheckTwo = MCheck(0,Chosen_Mono_one,Solv_one,Solv_two);
SolvTwoCheckPoly = PCheck(0,CoPoly,Solv_one,Solv_two);

if(SolvOneCheckOne < 1 && SolvOneCheckTwo < 1 && SolvOneCheckPoly < 1){
  OneCheck = true;
}
else{
  OneCheck = false;
}

if(SolvTwoCheckOne < 1 && SolvTwoCheckTwo < 1 && SolvTwoCheckPoly < 1){
  TwoCheck = true;
}
else{
  TwoCheck = false;
}
if(OneCheck && TwoCheck){
    if(temp_comp <= 0){
        final_comp = 0;
    } else{
        final_comp = 1;
    }
}
else if(OneCheck){
    final_comp = 1;
} else{
    final_comp = 0;
}
}

double FinalOneA, FinalOneB, FinalOneC;

// final fail safe to make sure chosen blend is within shared volume
FinalOneA = MCheck(final_comp, Chosen_Mono_one, Solv_one, Solv_two);
FinalOneB = MCheck(final_comp, Chosen_Mono_one, Solv_one, Solv_two);
FinalOneC = PCheck(final_comp, CoPoly, Solv_one, Solv_two);

if (FinalOneA < 1 && FinalOneB < 1 && FinalOneC < 1){
    FinalOne = true;
} else{
    FinalOne = false;
}

// final result of Mixer
if(FinalOne){
    cout << final_comp*100 << "% of " << Solv_one_N << " and " << 100 - final_comp*100 << "% of " << Solv_two_N << endl;
} else{
}
return;
}

/*****************************************
* 5. Functions  *
* *
* *
* *
* *
*******************************************/

// checking if the the HSP of one solvent lies inside the volume of another
bool SolventMiscibility(Solvent Solv_one,Solvent Solv_two){

double Solv_one_D,Solv_one_H,Solv_one_P,Solv_one_V;
double Solv_two_D,Solv_two_H,Solv_two_P,Solv_two_V;
double D_Diff,H_Diff,P_Diff,Total;
Solv_one_D = Solv_one.Give_Solv_D();
Solv_one_H = Solv_one.Give_Solv_H();
Solv_one_P = Solv_one.Give_Solv_P();
Solv_one_V = Solv_one.Give_Solv_V();
Solv_two_D = Solv_two.Give_Solv_D();
Solv_two_H = Solv_two.Give_Solv_H();
Solv_two_P = Solv_two.Give_Solv_P();
Solv_two_V = Solv_two.Give_Solv_V();

D_Diff = Solv_one_D - Solv_two_D;
H_Diff = Solv_one_H - Solv_two_H;
P_Diff = Solv_one_P - Solv_two_P;
Total = D_Diff*D_Diff + H_Diff*H_Diff + P_Diff*P_Diff;

if (Total < Solv_one_V || Total < Solv_two_V){
    return true;
}
else{
    return false;
}
}

// taking info from files and intializes monomer HSP
void Monomer::Mono_assign(string Mname, double s,double t,double u,double v,string Pname,double w,double x,double y,double z){
    Mono_name = Mname;
    MonoDel_D = s;
    MonoDel_P = t;
    MonoDel_H = u;
    Mono_Volm = v;
    Poly_name = Pname;
    PolyDel_D = w;
    PolyDel_P = x;
    PolyDel_H = y;
    PolyR_not = z;
}
// taking info from files and initializes solvent HSP
void Solvent::Solv_assign (string Sname,double w,double x,double y,double z){
    Solv_name = Sname;
    SolvDel_D = w;
    SolvDel_P = x;
    SolvDel_H = y;
    Solv_Volm = z;
}

// initializing polymer HSP
void Polymer::Poly_assign (string Pname,double w,double x,double y,double z){
    Poly_name = Pname;
    PolyDel_D = w;
    PolyDel_P = x;
    PolyDel_H = y;
    PolyR_not = z;
}

// intermediate step in copolymerization to gain HSP of homopolymers
Polymer HomoPolymerize(Monomer Mono){
    Polymer output;
    string Poly_N;
    double Poly_D,Poly_H,Poly_P,Poly_R;
    Poly_N = Mono.Give_Poly_N ();
    Poly_D = Mono.Give_Poly_D ();
    Poly_H = Mono.Give_Poly_H ();
    Poly_P = Mono.Give_Poly_P ();
    Poly_R = Poly_R*Poly_R;
    output.Poly_assign(Poly_N,Poly_D,Poly_P,Poly_H,Poly_R);
    return output;
}

// taking average of homopolymer HSP to find HSP of copolymer
Polymer CoPolymerize(Polymer Poly_one,Polymer Poly_two,double P){
    Polymer output;
    string output_N = "copoly";
    double Poly_one_D,Poly_one_H,Poly_one_P,Poly_one_R;
    double Poly_two_D,Poly_two_H,Poly_two_P,Poly_two_R;
    double output_D,output_H,output_P,output_R;
    Poly_one_D = Poly_one.Give_PolyD ();
    Poly_one_H = Poly_one.Give_PolyH ();
    Poly_one_P = Poly_one.Give_PolyP ();
    Poly_one_R = Poly_one.Give_PolyR ();
    Poly_two_D = Poly_two.Give_PolyD ();
    Poly_two_H = Poly_two.Give_PolyH ();
    Poly_two_P = Poly_two.Give_PolyP ();
    Poly_two_R = Poly_two.Give_PolyR ();
    output_D = P*Poly_one_D + (1-P)*Poly_two_D;
    output_H = P*Poly_one_H + (1-P)*Poly_two_H;
    output_P = P*Poly_one_P + (1-P)*Poly_two_P;
    output_R = P*Poly_one_R + (1-P)*Poly_two_R;
}
output.Poly_assign(output_N,output_D,output_P,output_H,output_R);
    return output;
}

// Calculations for quadratic formula coefficients
double CapDelTwo (double Solv_one,double Solv_two){
    double output;
    output = Solv_one*Solv_one-2*Solv_one*Solv_two + Solv_two*Solv_two;
    return output;
}

double CapDelOne(double Solv_one,double Solv_two,double Chosen_Mono ){
    double output;
    output = 2*Solv_one*Solv_two-2*Solv_one*Chosen_Mono-2*Solv_two*Solv_two + 2*Solv_two*Chosen_Mono;
    return output;
}

double CapDelNot(double Solv_two,double Chosen_Mono){
    double output;
    output = Chosen_Mono*Chosen_Mono-2*Chosen_Mono*Solv_two + Solv_two*Solv_two;
    return output;
}

// determining if a point in hansen space lies within a monomer
double MCheck(double Alpha,Monomer Chosen_Mono,Solvent Solv_one,Solvent Solv_two){
    double Check_D,Check_H,Check_P,output;
    double Solv_one_D,Solv_one_H,Solv_one_P;
    double Solv_two_D,Solv_two_H,Solv_two_P;
    double Mono_D,Mono_H,Mono_P,Mono_V,Mono_R;
    Solv_one_D = Solv_one.Give_Solv_D ();
    Solv_one_H = Solv_one.Give_Solv_H ();
    Solv_one_P = Solv_one.Give_Solv_P ();
    Solv_two_D = Solv_two.Give_Solv_D ();
    Solv_two_H = Solv_two.Give_Solv_H ();
    Solv_two_P = Solv_two.Give_Solv_P ();
    Mono_D = Chosen_Mono.Give_Mono_D ();
    Mono_H = Chosen_Mono.Give_Mono_H ();
    Mono_P = Chosen_Mono.Give_Mono_P ();
    Mono_V = Chosen_Mono.Give_Mono_V ();
    Mono_R = 2*8.314*293/Mono_V;
    Check_D = Alpha*Solv_one_D + (1-Alpha)*Solv_two_D;
    Check_H = Alpha*Solv_one_H + (1-Alpha)*Solv_two_H;
    Check_P = Alpha*Solv_one_P + (1-Alpha)*Solv_two_P;

    output  = (Check_D-Mono_D)*(Check_D-Mono_D)/Mono_R + (Check_H-Mono_H)*(Check_H-Mono_H)/Mono_R + (Check_P-Mono_P)*(Check_P-Mono_P)/Mono_R;
    return output;
}

// determining if a point in hansen space lies within a polymer
double PCheck(double Alpha,Polymer Poly,Solvent Solv_one,Solvent Solv_two){

}
double Check_D, Check_H, Check_P, output;
double Solv_one_D, Solv_one_H, Solv_one_P;
double Solv_two_D, Solv_two_H, Solv_two_P;
double Poly_D, Poly_H, Poly_P, Poly_R;
Solv_one_D = Solv_one.Give_Solv_D();
Solv_one_H = Solv_one.Give_Solv_H();
Solv_one_P = Solv_one.Give_Solv_P();
Solv_two_D = Solv_two.Give_Solv_D();
Solv_two_H = Solv_two.Give_Solv_H();
Solv_two_P = Solv_two.Give_Solv_P();
Poly_D = Poly.Give_PolyD();
Poly_H = Poly.Give_PolyH();
Poly_P = Poly.Give_PolyP();
Poly_R = Poly.Give_PolyR();

Check_D = Alpha*Solv_one_D + (1-Alpha)*Solv_two_D;
Check_H = Alpha*Solv_one_H + (1-Alpha)*Solv_two_H;
Check_P = Alpha*Solv_one_P + (1-Alpha)*Solv_two_P;

output = 4*(Check_D - Poly_D)*(Check_D - Poly_D)/Poly_R + (Check_H - Poly_H)/(Check_H - Poly_H) + (Check_P - Poly_P)/(Poly_P - Poly_P) * Poly_R;

return output;