ENVIRONMENTAL SCREENING OF FUTURE GASOLINE ADDITIVES: COMPUTATIONAL TOOLS TO ESTIMATE CHEMICAL PARTITIONING AND FORECAST WIDESPREAD GROUNDWATER CONTAMINATION

by
J. Samuel Arey

B.S., Public Policy and Environmental Science
Indiana University, Bloomington, 1998

M.S., Civil and Environmental Engineering
Massachusetts Institute of Technology, 2001

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the Degree of

DOCTOR OF PHILOSOPHY IN ENVIRONMENTAL CHEMISTRY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
September 2004

© Massachusetts Institute of Technology.
All rights reserved.

Signature of Author ______________________
Department of Civil and Environmental Engineering
September 1, 2004

Certified by ______________________
Philip M. Gschwend
Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by ______________________
Heidi M. Nepf
Professor of Civil and Environmental Engineering
Chairperson, Departmental Committee of Graduate Studies

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY
SEP 17 2004
LIBRARIES

BARKER

ENVIRONMENTAL SCREENING OF FUTURE GASOLINE ADDITIVES: COMPUTATIONAL TOOLS TO ESTIMATE CHEMICAL PARTITIONING AND FORECAST WIDESPREAD GROUNDWATER CONTAMINATION

by
J. Samuel Arey

Submitted to the Department of Civil and Environmental Engineering on
September 1, 2004 in partial fulfillment of the requirements for the
Degree of Doctor of Philosophy in Environmental Chemistry

Abstract

Fuel formulations evolve continually, and historical experience with the fuel additives tetraethyl lead and methyl-*tert*-butyl ether (MTBE) indicates that newly proposed additives should be screened for their potential to threaten environmental resources, before they are used widely.

A physical-chemical transport model was developed to forecast well water concentrations and transport times for gasoline components migrating from underground fuel tank releases to vulnerable community water supply wells. Transport calculations were parameterized using stochastic estimates of representative fuel release volumes and hydrogeologic characteristics, and were tailored to individual compounds based on their abundances in gasoline, gasoline-water partition coefficients, and organic matter-water partition coefficients. With no calibration, the screening model successfully captured the reported magnitude of MTBE contamination of at-risk community supply wells.

To estimate gasoline-water partition coefficients for unstudied solutes, we combined linear solvation energy relationships (LSERs) developed for pure 1:1 systems using linear solvent strength theory and a "solvent compartment" model. In this way, existing LSERs could be extended to treat solute partitioning from gasoline, diesel fuel, and similar mixtures into contacting aqueous mixtures. This allowed prediction of liquid-liquid partition coefficients in a variety of fuel-water systems for a broad range of dilute solutes. When applied to 37 polar and nonpolar solutes partitioning between an aqueous mixture and 12 different fuel-like mixtures (many including oxygenates), the estimated model error was a less than a factor of 2 in the partition coefficient. This was considerably more accurate than application of Raoult's law for the same set of systems.

An approach was developed which relates the empirical LSER solute polarity parameter, pi_2^H, to two more fundamental quantities: a polarizability term and a computed solvent accessible surface electrostatic term. Electrostatics computations employed dielectric field continuum models and a density functional theory (B3LYP) or efficient Hartree-Fock (HF/MIDI!) method for 90 polar and nonpolar organic solutes. Predicted pi_2^H values had a correlation coefficient of 0.95 and standard deviation of 0.11 relative to empirically measured values. The resulting model relies on only two fitted coefficients and has the additional advantage of potential applicability to any solute composed of C, H, N, O, S, F, Cl, and Br.

Thesis Supervisor: Dr. Philip M. Gschwend

Title: Professor of Civil and Environmental Engineering

Acknowledgments

This research was supported by:
the EPA/ONR/NCERCA Science to Achieve Results program, the
Alliance for Global Sustainability, the Martin Family Society of
Fellows for Sustainability, the Ralph M. Parsons Fellowship Foundation,
the Schoettler Scholarship Fund, and the Ford Scholarly Allowance.

My thesis committee – William Green Jr., Charlie Harvey, Bernhardt Trout, and Bruce Tidor – deserve my special thanks for providing insightful guidance. Additionally, Bill Green and his research group generously maintained the software and hardware (a nontrivial burden) that were used for all of the molecular *ab initio* calculations described herein.

Many colleagues and friends gave me critically important assistance with conceptual analysis, mathematical manipulations, writing revisions, computer support, access to key materials, and sometimes just a needed listening ear. They include: John MacFarlane, Elfatih Eltahir, Daniel Blankschtein, Katsuya Kawamoto (NIES, Japan), Torsten Schmidt (Tübingen, Germany), Michael Abraham (Univ. College, U.K.), Patricia Culligan (Columbia), Ken Beers, Federico san Martini, Lena Graber, Donnan Steele, Kai Udert, Julia Steinberger, Xin Fu, Farzan Parsinejad (Northeastern Univ.), Peter Huybers (WHOI), Dennis McLaughlin, Curt Stanley (API), Geralde Raabe (API), Bruce Bauman (API), John Brophy (EPA), Jed Goldstone (WHOI), and Stephen Grady (USGS), Al Leo (BioByte Corp.), Jim Landmeyer (USGS), Toby Avery (Mobil-Exxon), and not least, fellow members of the Gschwend laboratory group.

I am deeply indebted to my very supportive and caring (and neglected) parents.
I am particularly grateful for the spirited and patient counsel of Phil Gschwend.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction. Forecasting the environmental transport of gasoline additives and resulting human exposures

1.1. Anticipating the environmental impacts of synthetic chemicals

Societies have employed chemical technology to reinvent their relationship with nature for millennia. In the 20th century, we have witnessed unprecedented growth in the volume and variety of manufactured chemical materials. From 1930 to 2000, global production of synthetic chemicals increased from 1 million to 400 million metric tons annually (*1*). In the U.S., we generate about 28% of worldwide production value in synthetic chemicals (*1*) and currently track over 75000 chemical substances in registered commercial use (*2*). From plastics to pesticides to pharmaceuticals, it is difficult to overstate the perceived societal value of newly invented chemical technologies during the last several decades. For example, in *Scientific American* in 1951, the widely used insecticide, p,p'-dichlorodiphenyl-trichloroethane (DDT), was compared with steel and fuel as "one of the great world necessities" due to its record of effective malaria eradication in endemic areas (*3*). Incidentally, in the same year, U.S. Food and Drug Administration workers reported an average of 5 ppm (part-per-million) levels of DDT in the fatty tissues of 75 randomly tested California residents – none were occupational pesticide operators (*4*).

Mid-stride into this era of rapidly expanding synthetic chemical production and commerce, severe ecological and human toxicity of many popular chemicals became apparent. In 1962, Rachel Carson's best-seller, "Silent Spring," sparked wide controversy and initiated an important shift in the public's attitude towards synthetic chemicals. Carson starkly reported the ecological persistence and bioaccumulation of DDT and other pesticides, and the resulting pervasive and nonselective deaths of insects, fish, birds, and other wildlife in heavily treated areas (*5*). Following further scientific inquiry commissioned by President John Kennedy, DDT was eventually banned from use in the U.S. In response to pressure from the public and citizens' groups on these and related issues, President Richard Nixon created the Environmental Protection Agency (EPA) in 1970. Similarly, the 1972 Environmental Pesticide Control Act, the 1976 Toxic Substances Control Act (TSCA), and numerous other legislative actions established a formal programmatic framework for tracking, evaluating, and regulating the use of synthetic compounds, based on their established or suspected ecological and human health impacts.

Meanwhile, scientists continued to uncover the environmental persistence, bioaccumulation, and toxicity of certain commercial or industrial synthetic compounds. For example, widely used industrial dielectric fluids known as polychlorinated biphenyls (PCBs) were found in the fat and eggs of numerous wild English predatory birds in the 1960's (*6*). In areas near Saigon where the U.S. military herbicide mixture known as Agent Orange had been sprayed, the toxic chemical component 2,3,7,8-tetrachlorodibenzo-p-dioxin (TCDD) was found in commercial river fish at ~500 ppt (part-per-trillion) levels (*7*). The presence of PCBs and TCDD in high level carnivores strongly suggested that these compounds were bioaccumulating, i.e., distilling up the food chain and concentrating contaminants in the fatty tissues of animals at the highest trophic levels. Extensive evidence eventually confirmed the persistence (nondegradability) of PCBs in the environment (*8*). Similarly, several years after Agent Orange was applied, TCDD was still detectable in human breast milk at hundreds of ppt levels in heavily sprayed areas (*9,10*). As early as the 1930s, PCBs were known to have toxicity to laboratory animals and humans at low doses, including well-documented cases of severe chloracne and liver damage suffered by exposed Monsanto factory workers (*11*). Although the specific toxicity of TCDD to humans was not well established, food contaminated with TCDD was shown to be lethal to guinea pigs (*12*) at 10 ppb (part-per-billion) and cause cancers in laboratory rats at sub-ppb levels (*13*). Two cohorts of TCDD-exposed U.S. industrial workers were found to have significantly increased chloracne and cancer mortality rates relative to control groups, according to the company's own health records (*14,15*). The full evidence of damage to human and ecological health associated with exposure to these and many other synthetic compounds is far too extensive to review here. However, such threads of investigation illustrated a pattern typical among widely used synthetic compounds: persistence in the environment for months or years, accumulation in animal and human tissues, and demonstrated toxic effects at low levels.

More recently, regulatory and scientific progress has enabled us to ameliorate some environmental and health damage of the past, but not necessarily to avert future threats. Synthetic chemicals are still distributed and used widely before adequate environmental transport or toxicological assessments are performed. In the U.S., DDT, PCBs, Agent Orange, and other problematic synthetic substances have been banned from use, but we continue to find that many presently used chemicals pose ecological or human health threats. For example, in 2002 and 2003 several studies found that environmentally realistic concentrations (sub-ppm) of atrazine, an agricultural pesticide, cause severe endocrine disruption and resulting hermaphroditism and weakened immune response in frogs (*16-18*). Within the past five years, the widely used flame retardants, polybrominated diphenylethers (PBDEs), have been reported in human breast milk at 0.2 ppb levels (*19*) as well as in the tissues of fish and other wildlife (*20-22*). Although toxicological data on PBDEs is limited, recent studies have suggested that these compounds may contribute to cancer, hormone disruption, or neurodevelopmental problems (*23,24*). Many other such illustrative cases could be cited; synthetic chemicals are widely distributed before adequate environmental assessments are performed. Of the 2600 substances that are currently produced or imported in quantities greater than 1 million lbs/year in the U.S. (high production volume (HPV) chemicals), only a few hundred have been preliminarily evaluated for toxicological and environmental transport properties (*25*). Similarly, of the 75000 chemicals in registered commercial use in the U.S. and the 100000 registered in the European Union (EU), the vast majority are currently untested.

U.S. and EU agencies have recognized the need for data and models to forecast chemical threats. In a 1995 assessment, the EPA Science Advisory Board concluded that "EPA's traditional methods of identifying and solving environmental problems will not be adequate to protect against problems that may emerge several years or decades from now. They were not designed to determine the costs of future environmental problems or the benefits of actions taken today to avoid them" (*26*). The document then recommends that "EPA should establish a strong environmental futures capability that serves as an early-warning system for emerging environmental problems." A more recent EPA Office of Research and Development (ORD) Strategic Plan reaffirms the stated goal to "[s]earch for detectable early warning signals and extrapolate them into the future" in order to anticipate future environmental issues (*27*). In 1998, collaborative efforts between the EPA, the Environmental Defense Fund (a nonprofit watchdog group), and the Chemical Manufacturer's Association negotiated the EPA high production volume chemical testing program, an agreement to systematically evaluate HPV chemicals in the U.S. (*28*). The HPV testing program was intended to accelerate a sluggishly paced Organization for Economic Cooperation and Development (OECD) chemical testing program, which was founded in 1990 and had only completely evaluated about 160 chemicals regulated under TSCA (*28*). In 1998, the EPA also initiated the Endocrine Disruptor Screening Program, which was intended to evaluate practically all U.S. synthetic chemicals in commercial use for possible endocrine disruption effects (*29*). More recently, the EU finalized a more ambitious and contentious set of rules known as the Registration, Evaluation, and Authorization of Chemicals (REACH) program, which "transfers the burden of proof for a chemical's safety from the government to the manufacturer" (*30*). Continued development of such strategies will hopefully enable us to eventually "pre-act," rather than react, to chemical threats to the environment and human health. These efforts will require both data and forecasting models.

Since extensive testing of even a single chemical is time-consuming and expensive, regulators need to quickly identify chemicals which are either: (a) likely to be highly toxic; or (b) likely to expose people or the natural environment to high intake doses, as a result of their uses and chemical properties. The 2001 REACH program white paper argues that, "[g]iven the vast number of existing substances on the market, the European Commission proposes that first priority is given to substances that lead to a high exposure or cause concern by their known or suspected dangerous properties" (*1*). In other words, chemicals should first be ranked by the roughly estimated severity of threat that they pose. Chemicals initially identified as "high priority" suspects would then undergo more extensive testing on a rapid timetable. To conduct such ranking activities, regulators need efficient methods to estimate the exposure levels of commercial chemicals to humans and ecological endpoints. Such forecasting approaches are commonly referred to as chemical "priority-setting" or "screening" tools (*31*); ideally, they should be designed to make fast and reliable estimates for a wide range of synthetic compound types. With such forecasts readily available, regulators and industry could strategically avoid the severe environmental, health, and economic costs that often unintentionally result from specific chemical applications. **This thesis therefore focuses on the development of screening models for chemical exposure-forecasting and physical property estimation.** Toxicological evaluations are a critical accompaniment to such predictive assessments, but this is beyond the scope of the work described here.

Exposure estimates require an understanding of critical environmental transport pathways and the complementary physical-chemical properties of the chemical of interest. Screening models should therefore identify these routes to exposure and estimate the resulting compound concentrations at relevant human or ecological endpoints. Multiple environmental pathways of transport or exposure may require consideration, and stochastic approaches may be needed to adequately evaluate a range or distribution of likely exposure outcomes. Additionally, the expected production or use rate of a given compound may be needed to evaluate the compound source or emission terms. Finally, in cases where physical-chemical property data for a compound is cost-prohibitive or difficult to measure, Quantitative Structure Activity Relationships (QSAR) or other models may be used to estimate the properties of interest.

In this thesis, I have developed screening models and physical-chemical property estimation methods tailored to forecasting human exposures to chemicals used in fuels, particularly automotive gasoline. Gasoline is a highly motivating case study for environmental screening assessment, since it is widely used in high volumes. However, the philosophy underlying these approaches could be extended to other environmental transport problems for other chemicals. Hopefully, such screening models will eventually contribute to a comprehensive but efficient prioritization system for evaluating existing and new commercial synthetic chemicals.

1.2. Forecasting human exposures to chemicals in gasoline and fuel mixtures

Historical and present experience with gasoline additives such as tetra-ethyl lead (TEL) and methyl-*tert*-butylether (MTBE) strongly illustrate the need for fuel component exposure forecasting models. The use of TEL in U.S. gasolines has introduced millions of metric tons of lead into the environment, and this has been directly linked to child lead poisoning on a national scale (*32*). MTBE use has resulted in the closure of hundreds of contaminated drinking water supplies in the U.S. over only a few years, incurring enormous environmental and economic costs through lost water resources (*33,34*). Conceivably, environmental transport forecasts for future fuel constituents could avert similar (and possibly even entirely new) mistakes in the years ahead.

Exposures to different fuel constituents may involve different environmental transport pathways, depending on the physical-chemical properties of the component chemicals. Jo and Oh showed that gasoline service stations may significantly expose employees and nearby residents to (uncombusted) volatile gasoline constituents (*35*). Other studies have indicated that the emission of volatile fuel components in the lower troposphere may also cause nonnegligible exposures to urban residents (*36,37*). Extensive contamination of subsurface water supplies by fuel leaks or spills is well documented (*33,38,39*), and surface water supplies may also be contaminated by recreational boating (*40*). Additionally, atmospheric deposition of fuel compounds could widely contaminate surface soil and dust or water supplies (*32,41*). Appreciation for these particular transport pathways has evolved from our accumulated experience with fuel-related contamination, but these do not necessarily constitute a comprehensive list of possible exposure routes. In order to develop robust exposure forecasts, each of these scenarios should be considered separately, possibly in addition to others. Only two of the scenarios – the contamination of subsurface water supplies and the pollution of urban airsheds – are discussed in detail here.

As has been demonstrated by the use of MTBE, gasoline contamination of subsurface water supplies is potentially widespread and environmentally costly. This problem poses a significant challenge to gasoline design, since gasolines evolve continually and typically contain varying concentrations of several water-soluble components. From the standpoint of chemical fate forecasting, unresolved research questions arise; for example, can a simple model meaningfully capture contaminant transport behaviors at thousands of differing gasoline release sites? Conventional subsurface transport models estimate the fate of various contaminants in a somewhat characterized geologic formation, and these approaches have been extensively developed and tested. However, in order to prescribe policy actions for a specific chemical in gasoline (rather than for a specific contamination site), the screening model approach must somehow treat the ensemble of gasoline-contaminated sites. An additional difficulty exists: reliable approaches for estimating the gasoline-water equilibrium partitioning (i.e., the chemical mass distribution between gasoline and water) of new gasoline components are not currently available. The gasoline-water partition coefficient pivotally influences the extent to which an individual gasoline chemical component is expected to dissolve from a subsurface gasoline release into groundwater. Consequently, even an order-of-magnitude estimate of this physical-chemical property might allow rapid determination of whether a newly proposed gasoline amendment would be reasonably nonthreatening to subsurface water supplies. Finally, the prevalent subsurface (bio)degradability of gasoline constituents is difficult to predict or measure in laboratory trials. Because natural aquifers effectively act as bio-active filters which clean subsurface waters, a highly persistent chemical might pervasively contaminate water supplies. While additional uncertainties certainly pose challenges to effectively forecasting widespread damage to subsurface water supplies by gasoline, these are perhaps the most salient: how to treat an "ensemble" of gasoline contamination sites; how to estimate gasoline-water partition coefficients of gasoline components; and how to deal with the subsurface biodegradability of gasoline components. The first two of these will be treated in detail in subsequent chapters; however, subsurface degradability is difficult to reliably predict for a wide range of compounds, particularly if generality to many subsurface environments is desirable. Consequently, estimation approaches for subsurface degradability have been left for future work.

Although the advent of MTBE has focused a good deal of attention on water contamination, the urban atmosphere probably presents the most pervasive route of human exposure to gasoline compounds. Kawamoto, Arey, and Gschwend recently proposed a screening model for quickly estimating the expected order-of-magnitude urban air concentrations for gasoline additives in a particular urban setting (*37*). Possibly the most challenging aspect of this approach is estimating the emission rate of a volatilized gasoline compound to the urban airshed (either assuming that there are no other major emission sources or else accounting for these terms). Both the estimated emissions rate and expected subsequent environmental transport behaviors rely heavily on the equilibrium partitioning of a gasoline constituent between the gasoline mixture and air, between air and water, and between air and soil particles. Additionally, the compound's rates of degradation in air, water, and soil play an important role in both calculating expected urban airshed concentrations and assessing the eventual fate of the compound in the environment.

1.3. Relevant physical-chemical property estimation methods

Whether we are considering the fate of a gasoline constituent in subsurface water supplies or in an urban airshed, information about its equilibrium distribution in various environmental media is required. Measured or estimated chemical partition coefficients for gasoline-water, soil-water, and air-water systems (and possibly others) effectively distinguish the anticipated behaviors of one proposed gasoline additive from another. In fact, such physical-chemical properties are typically critical for assessing and differentiating the environmental transport of contaminants in almost any relevant context (*42-45*). Since measuring these properties in the laboratory is frequently difficult or costly, reliable estimation methods are a valuable asset to environmental exposure screening models.

Linear Solvation Energy Relationships (LSERs) have become a widely accepted approach for accurately estimating two-phase partition coefficients of organic compounds. LSERs were developed through the work of Kamlet and Abraham and coworkers (*46,47*) and propose that a solvation (or partitioning) free energy may be approximated as:

$$\log K = c + rR_2 + s\pi_2^{\mathrm{H}} + a\alpha_2^{\mathrm{H}} + b\beta_2^{\mathrm{H}} + mV_{\mathrm{x}} \qquad (1\text{-}1)$$

where the parameters R_2, π_2^{H}, α_2^{H}, β_2^{H}, and V_x describe the excess molar refraction (*48*), polarity/polarizability (*49*), hydrogen-bonding acidity (*50*), hydrogen-bonding basicity (*51-53*), and group-contributable molecular volume (*54*) of the solute, respectively. The fitted coefficients *c*, *r*, *s*, *a*, *b*, and *m* are calibrated to a specific two-phase system. This formulation is attractive to environmental scientists because it has been shown to accurately predict partition coefficients (*K* values) for a wide range of solute types in many different solvent and mixture systems (*48,55*).

From the standpoint of environmental fate analysis, LSERs have two notable limitations: first, the empirical fitting of LSER coefficients is data intensive and difficult to extrapolate to new systems in the absence of data; second, determination of the LSER polarity/polarizability (π_2^{H}) and hydrogen-bonding (α_2^{H} and β_2^{H}) solute parameters is data intensive, and these quantities have not been found to closely correlate with calculated or easily measurable properties. Essentially, development of LSER parameters and coefficients for new solutes and new systems is costly and time-consuming. Consequently, modeling approaches or approximations that extend LSERs into these knowledge gaps are desirable additions to a chemical exposure screening model "toolkit." For example, it is not clear how LSERs might apply to fuels, since fuels are variable mixtures and would technically require a set of separately fitted LSER coefficients for each significantly different fuel type (and therefore a new set of measurements of each fuel formulation). Additionally, newly proposed fuel additives may not have known (measured) LSER solute parameters; hence models to compute or estimate these values could be useful for rapid screening assessments. Such modeling strategies have broader implications than fuels assessment. Any chemical exposure or toxicological screening method reliant on chemical partitioning free energy information would benefit from modeling approaches that extrapolate existing LSERs to novel solutes and novel mixtures.

1.4. Main objective of the thesis

The preceding discussion has led us to the following questions. Given a hypothetical newly proposed gasoline additive, could one feasibly estimate its risk to community water supply wells and volatile exposures in urban air, *before the additive is used?* What physical or chemical properties of the compound are critical for making such forecasts, and could these properties be easily estimated in the absence of measurements? In the current work, some modeling tools are proposed to address these problems. This hopefully lays groundwork for future efforts to systematically identify fuel additives which would cause undesirable human or ecological exposures via groundwater, urban air, or other pathways. Regulators and industry could eventually optimize the results of such assessments along with other use criteria (e.g., manufacturing costs and combustion properties) to strategically choose the "best" among a range of proposed additive compounds.

1.5. Case example: is this hypothetical gasoline additive "safe"?

As an illustrative dilemma, let us suppose that n-pentyl nitrate (nPN) has been proposed by industry chemists as an octane-enhancing (i.e., anti-knock) gasoline amendment:

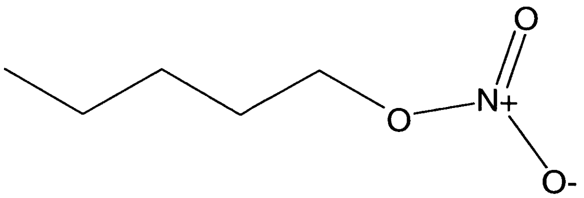

Figure 1-1. n-pentyl nitrate (nPN)

Given no prior knowledge of this chemical's environmental behavior, and little or no physical property information, how would one assess whether it is an acceptable amendment to fuels? At what gasoline concentration would it pose a widespread threat to community water supply wells due to leaking underground fuel tanks? How might it affect urban air quality via volatile emissions? What partitioning properties are important, and could they be reliably estimated? The methods proposed herein attempt to address such concerns, using computationally expedient modeling tools to produce practical decision-making information. Issues relating to tailpipe emissions or other routes of human or ecological exposure (such as impacts to surface water in recreational boating areas) have been ignored for the moment, since these concerns lie outside the scope of the current work. The nPN case study, detailed in chapter 5, demonstrates application of several of the physical-chemical estimation methods and screening models.

1.6. Outline of thesis

In *Chapter 2*, an exposure screening model is proposed for forecasting the widespread contamination of subsurface community water supplies by a new gasoline additive. The model development draws on information about the typical hydrogeologic setting of community water supply wells and the proximity and size of vicinal fuel releases to the land or subsurface environment in the U.S. Notably, the screening model is tuned to a specific gasoline constituent based only on its abundance in gasoline, its gasoline-water partition coefficient, and its organic matter-water partition coefficient. In other words, it does not require *a priori* calibration using field measurements or subsurface contamination data. The screening model forecasts an expected distribution of contamination levels in the population of affected wells, and estimates the time of subsurface transport. Based on this information, regulators or scientists may conclude what gasoline amendment levels would pose an acceptably low level of risk to subsurface community water supplies, and whether additional toxicological or biodegradability information is needed (i.e., requiring further testing). This screening model was evaluated based on comparisons to reported MTBE contamination of community supply wells in the U.S.

Chapter 3 presents a mixing rule for LSER coefficients using Linear Solvent Strength Theory (LSST). In other words, the LSST approximation was used to extrapolate known "pure phase" LSER coefficients to new mixture systems. Gasoline-water two-phase mixture data from the literature were assembled as a set of test cases. Mixing rule prediction results were compared to measured partition coefficients for 37 solutes in 17 two-phase systems involving fuels or fuel-like mixtures. Consequently, the LSER formulation could be applied to a range of gasoline mixture types, allowing gasoline-water partition coefficient estimates for a new set of systems.

In *Chapter 4*, I describe an electrostatic model and employ molecular orbital calculations to estimate the LSER solute polarity parameter, π_2^H. I suggest that the polarity parameter is related to two more fundamental physical quantities: a measured polarizability term and a calculated solute-solvent electrostatic interaction term. I evaluated various possible molecular "solvent accessible surfaces" that have been discussed in the literature. A diverse set of 90 semi-polar and highly polar solutes containing C, H, N, O, S, F, Cl, and Br were used to calibrate and test the model. The proposed method allows computational estimates of π_2^H values for novel (untested) solutes and will hopefully lead to further advances in the physical interpretation and modeling of LSERs.

Chapter 5 illustrates an example exposure screening model calculation for the hypothetical gasoline additive, nPN, as an instructive case study. The physical-chemical property estimation methods discussed in chapters 3 and 4 were used in conjunction with previously published modeling tools to estimate relevant partition coefficients for nPN. These physical-chemical parameters were then supplied as inputs for the community supply well screening model described in chapter 2 and the urban airshed exposure assessment model reported by Kawamoto, Arey, and Gschwend (*37*). Taking the study a step further, it was assumed that some toxicological information about tolerable exposure levels to nPN was given; consequently a range of environmentally acceptable fuel amendment levels of nPN could be recommended. These didactic exercises are only intended to illustrate the application of the physical-chemical property estimation and environmental screening models. In real cases, other environmental impacts, e.g., contamination of surface waters or interaction with other pollutants during combustion, should also be considered.

A summary of conclusions, and suggested areas of future work, are discussed in *Chapter 6*.

In *Appendix A*, I report a brief investigation of hydrogen-bonding parameter calculations using molecular orbital methods. *Appendix B* contains the C++ code which was used to construct the probability density functions and conduct the Monte Carlo analysis of subsurface transport forecasts reported in chapter 2. In *Appendix C*, I report an example Matlab code for iterative calculation of fuel-water system compositions based on mass balance constraints, as described in chapter 3. *Appendix D* contains the C++ and IDL (Interactive Data Language) codes which were used to: (a) generate Gaussian98 (molecular modeling) input files, (b) analyze and manipulate Gaussian98 output files and electron density data, (c) generate the numerical solute electron isodensity surface grids, and (d) perform the electrostatic energy integrations from Gaussian98 ouput which are discussed in chapter 4.

1.7. References

(1) Commission of the European Communities "White Paper. Strategy for a future chemicals policy," 2001.

(2) Office of Pollution Prevention & Toxics, Toxic Substances Control Act Chemical Substance Inventory, 2003 U.S. Environmental Protection Agency.

(3) Editorial staff In *Scientific American*, April 1951; p 32.

(4) Laug, E. P.; Kunze, F. M.; Prickett, C. S., *Occurrence of DDT in human fat and milk.* Industrial Hygiene and Occupational Medicine **1951**, *3*, 245.

(5) Carson, R. *Silent Spring*; Houghton Mifflin: Boston, **1962**.

(6) Holmes, D. C.; Simmons, J. H.; Tatton, J. O. G., *Chlorinated hydrocarbons in British wildlife.* Nature **1967**, *216*, 227-229.

(7) Shapley, D., *Herbicides: AAAS study find dioxin in Vietnamese fish.* Science **1973**, *180*, 285-286.

(8) Ballschmiter, K.; Zell, M.; Neu, H. J., *Persistence of PCBs in ecosphere - will some PCB components "never" degrade?* Chemosphere **1978**, *7*, 173-176.

(9) Schecter, A. J.; Ryan, J. J.; Constable, J. D., *Chlorinated dibenzo-p-dioxin and dibenzofuran levels in human adipose-tissue and milk samples from the north and south of Vietnam.* Chemosphere **1986**, *15*, 1613-1620.

(10) Schecter, A. J.; Dai, L. C.; Thuy, L. T. B.; Quynh, H. T.; Minh, D. Q.; Cau, H. D.; Phiet, P. H.; Phuong, N. T. N.; Constable, J. D.; Baughman, R.; Papke, O.; Ryan, J. J.; Furst, P.; Raisanen, S., *Agent orange and the Vietnamese: the persistence of elevated dioxin levels in human tissues.* American Journal of Public Health **1995**, *85*, 516-522.

(11) Drinker, C. K.; Warren, M. F.; Bennet, G. A., *The problem of possible systemic effects from certain chlorinated hydrocarbons.* The Journal of Industrial Hygiene and Toxicology **1937**, *19*, 283-310.

(12) Greig, J. B.; Jones, G.; Butler, W. H.; Barnes, J. M., *Toxic effects of 2,3,7,8-tetrachlorodibenzo-p-dioxin.* Food and Cosmetics Toxicology **1973**, *11*, 585-595.

(13) Kociba, R. J.; Keyes, D. G.; Beyer, J. E.; Carreon, R. M.; Wade, C. E.; Dittenber, D. A.; Kalnins, R. P.; Frauson, L. E.; Park, C. N.; Barnard, S. D.; Hummel, R. A.; Humiston, C. G., *Results of a two-year chronic toxicity and oncogenicity study of 2,3,7,8-tetrachlorodibenzo-p-dioxin in rats.* Toxicology and Applied Pharmacology **1978**, *46*, 279-303.

(14) Ott, M. G.; Olson, R. A.; Cook, R. R.; Bond, G. G., *Cohort mortality study of chemical workers with potential exposure to the higher chlorinated dioxins.* Journal of Occupational Medicine **1987**, *29*, 422-429.

(15) Fingerhut, M. A.; Halperin, W. E.; Marlow, D. A.; Piacitelli, L. A.; Honchar, P. A.; Sweeney, M. H.; Greife, A. L.; Dill, P. A.; Steenland, K.; Suruda, A. J., *Cancer mortality in workers exposed to 2,3,7,8-tetrachlorodibenzo-p-dioxin.* The New England Journal of Medicine **1991**, *324*, 212-218.

(16) Hayes, T. B.; Collins, A.; Lee, M.; Mendoza, M.; Noriega, N.; Stuart, A. A.; Vonk, A., *Hermaphroditic, demasculinized frogs after exposure to the herbicide atrazine at low ecologically relevant doses.* Proceedings of the National Academy of Sciences of the United States of America **2002**, *99*, 5476-5480.

(17) Hayes, T.; Haston, K.; Tsui, M.; Hoang, A.; Haeffele, C.; Vonk, A., *Atrazine-induced hermaphroditism at 0.1 ppb in American leopard frogs (Rana pipiens): laboratory and field evidence.* Environmental Health Perspectives **2003**, *111*, 568-575.

(18) Gendron, A. D.; Marcogliese, D. J.; Barbeau, S.; Christin, M. S.; Brousseau, P.; Ruby, S.; Cyr, D.; Fournier, M., *Exposure of leopard frogs to a pesticide mixture affects life history characteristics of the lungworm Rhabdias ranae.* Oecologia **2003**, *135*, 469-476.

(19) Meironyte, D.; Noren, K.; Bergman, A., *Analysis of polybrominated diphenyl ethers in Swedish human milk. A time-related trend study, 1972-1997.* Journal of Toxicology and Environmental Health - Part A **1999**, *58*, 329-341.

(20) Akutsu, K.; Obana, H.; Okihashi, M.; Kitigawa, M.; Nakazawa, H.; Matsui, Y.; Makino, T.; Oda, H.; Hori, S., *GC/MS analysis of polybrominated diphenyl ethers in fish collected from the Inland Sea of Seto, Japan.* Chemosphere **2001**, *44*, 1325-1333.

(21) Boon, J. P.; Lewis, W. E.; Tjoen-A-Choy, M. R.; Allchin, C. R.; Law, R. J.; de Boer, J.; ten Hallers-Tjabbes, C. C.; Zegers, B. N., *Levels of polybrominated diphenyl ether (PDBE) flame retardants in animals representing different trophic levels of the North Sea food web.* Environmental Science & Technology **2002**, *36*, 4025-4032.

(22) Law, R. J.; Alaee, M.; Allchin, C. R.; Boon, J. P.; Lebeuf, M.; Lepom, P.; Stern, G. A., *Levels and trends of polybrominated diphenylethers and other brominated flame retardants in wildlife.* Environment International **2003**, *29*, 757-770.

(23) Helleday, T.; Tuominen, K. L.; Bergman, A.; Jenssen, D., *Brominated flame retardants induce intragenic recombination in mammalian cells.* Mutation Research - Genetic Toxicology and Environmental Mutagenesis **1999**, *439*, 137-147.

(24) McDonald, T. A., *A perspective on the potential health risks of PBDEs.* Chemosphere **2002**, *46*, 745-755.

(25) Office of Pollution Prevention & Toxics, High Production Volume (HPV) Challenge Program, 2004 U.S. Environmental Protection Agency.

(26) Loehr, R.; Alm, A.; Conway, R.; Deisler, P.; Dickson, K.; Gordon, T. J.; Hansen, F.; Lippmann, M.; Matanoski, G. M.; Middleton, P.; Ray, V.; Yosie, T. "Beyond the Horizon: Using Foresight to Protect the Environmental Future," U.S. Environmental Protection Agency Science Advisory Board, 1995, EPA-SAB-EC-95-007.

(27) Office of Research and Development "Office of Research and Development Strategic Plan," U.S. Environmental Protection Agency, 2001, EPA/600/R-01/003.

(28) Johnson, J. In *Chemical & Engineering News*, 1999; Vol. 77, pp 23-26.

(29) Hileman, B., *Low-dose problem vexes endocrine testing plans.* Chemical & Engineering News **1999**, *77*, 27-31.

(30) Brown, V. J., *REACHing for chemical safety.* Environmental Health Perspectives **2003**, *111*, A766-A769.

(31) Office of Science Coordination and Policy, Endocrine Disruptor Screening Program, 2004 U.S. Environmental Protection Agency Office of Prevention, Pesticides, and Toxic Substances.

(32) Mielke, H. W.; Reagan, P. L., *Soil is an important pathway of human lead exposure.* Environmental Health Perspectives **1998**, *106*, 217-234.

(33) Squillace, P. J.; Zogorski, J. S.; Wilber, W. G.; Price, C. V., *Preliminary assessment of the occurrence and possible sources of MTBE in groundwater in the United States, 1993-1994.* Environmental Science & Technology **1996**, *30*, 1721-1730.

(34) Johnson, R.; Pankow, J. F.; Bender, D.; Price, C.; Zogorski, J. S., *MTBE, To what extent will past releases contaminate community water supply wells?* Environmental Science & Technology **2000**, *34*, 2A-9A.

(35) Jo, W.-K.; Oh, J.-W., *Exposure to methyl tertiary butyl ether and benzene in close proximity to service stations.* Journal of the Air & Waste Management Association **2001**, *51*, 1122-1128.

(36) Watson, J. G.; Chow, J. C.; Fujita, E. M., *Review of volatile organic compound source apportionment by chemical mass balance.* Atmospheric Environment **2001**, *35*, 1567-1584.

(37) Kawamoto, K.; Arey, J. S.; Gschwend, P. M., *Emission and fate assessment of methyl tertiary butyl ether in the Boston area airshed using a simple multimedia box model: comparison with urban air measurements.* Journal of the Air & Waste Management Association **2003**, *53*, 1426-1435.

(38) Freeze, R. A.; Cherry, J. A. *Groundwater*; Prentice-Hall, Inc: Englewood Cliffs, NJ, **1979**.

(39) Squillace, P. J.; Moran, M. J.; Lapham, W. W.; Price, C. V.; Clawges, R. M.; Zogorski, J. S., *Volatile organic compounds in untreated ambient groundwater of the United States.* Environmental Science & Technology **1999**, *33*, 4176-4187.

(40) Reuter, J. E.; Allen, B. C.; Richards, R. C.; Pankow, J. F.; Goldman, C. R.; Scholl, R. L.; Seyfried, J. S., *Concentrations, sources, and fate of the gasoline oxygenate methyl tert-butyl ether (MTBE) in a multiple use lake.* Environmental Science & Technology **1998**, *32*, 3666-3672.

(41) Pankow, J. F.; Thomson, N. R.; Johnson, R. L.; Baehr, A. L.; Zogorski, J. S., *The urban atmosphere as a non-point source for the transport of MTBE and other volatile organic compounds to shallow groundwater.* Environmental Science & Technology **1997**, *31*, 2821-2828.

(42) MacFarlane, S.; Mackay, D., *A fugacity-based screening model to assess contamination and remediation of the subsurface containing non-aqueous phase liquids.* Journal of Soil Contamination **1998**, *17*, 17-46.

(43) Mackay, D.; Shiu, W. Y.; Maijanen, A.; Feenstra, S., *Dissolution of non-aqueous phase liquids in groundwater.* Journal of Contaminant Hydrology **1991**, *8*, 23-42.

(44) Mackay, D.; Guardo, A. D.; Paterson, S.; Kicsi, G.; Cowan, C. E.; Kane, D. M., *Assessment of chemical fate in the environment using evaluative, regional and local-scale models: illustrative application to chlorobenzene and linear alkylbenzene sulfonates.* Environmental Toxicology and Chemistry **1996**, *15*, 1638-1648.

(45) Mackay, D.; Webster, E., *Linking emissions to prevailing concentrations - exposure on a local scale.* Environmetrics **1998**, *9*, 541-553.

(46) Kamlet, M. J.; Abboud, J.-L. M.; Abraham, M. H.; Taft, R. W., *Linear Solvation Energy Relationships. 23. A comprehensive collection of the solvatochromic parameters, π^*, α, and β and some methods for simplifying the generalized solvatochromic equation.* Journal of Organic Chemistry **1983**, *48*, 2877-2887.

(47) Abraham, M. H.; Chadha, H. S.; Whiting, G. S.; Mitchell, R. C., *Hydrogen-bonding. 32. An analysis of water-octanol and water-alkane partitioning and the delta-logP parameter of Seiler.* Journal of Pharmaceutical Sciences **1994**, *83*, 1085-1100.

(48) Abraham, M. H.; Poole, C. F.; Poole, S. K., *Classification of stationary phases and other materials by gas chromatography.* Journal of Chromatography A **1999**, *842*, 79-114.

(49) Abraham, M. H.; Whiting, G. S., *XVI. A new solute solvation parameter, π_2^H, from gas chromatographic data*. Journal of Chromatography **1991**, *587*, 213-228.

(50) Abraham, M. H.; Grellier, P. L.; Prior, D. V.; Duce, P. P.; Morris, J. J.; Taylor, P. J., *Hydrogen bonding. Part 7. A scale of solute hydrogen-bond acidity based on logK values for complexation in tetrachloromethane*. Journal of the Chemical Society. Perkin Transactions 2 **1989**, 699-711.

(51) Abraham, M. H.; Grellier, P. L.; Prior, D. V.; Morris, J. J.; Taylor, P. J., *Hydrogen bonding. Part 10. A scale of solute hydrogen-bond basicity using logK values for complexation in tetrachloromethane*. Journal of the Chemical Society. Perkin Transactions 2 **1990**, 521-529.

(52) Abraham, M. H., *Scales of solute hydrogen-bonding: their construction and application to physiochemical and biochemical processes*. Chemical Society Reviews **1993**, 73-83.

(53) Abraham, M. H., *Hydrogen bonding. 31. Construction of a scale of solute effective or summation hydrogen-bond basicity*. Journal of Physical Organic Chemistry **1993**, *6*, 660-684.

(54) Abraham, M. H.; McGowan, J. C., *The use of characteristic volumes to measure cavity terms in reversed phase liquid chromatography*. Chromatographia **1987**, *23*, 243-246.

(55) Goss, K. U.; Schwarzenbach, R. P., *Linear free energy relationships used to evaluate equilibrium partitioning of organic compounds*. Environmental Science & Technology **2001**, *35*, 1-9.

Chapter 2
A physical-chemical screening model for anticipating widespread contamination of community water supply wells by gasoline constituents

2.1. Introduction

2.1.1. Motivation

Less than ten years after the widely increased addition of methyl *tert*-butyl ether (MTBE) to U.S. gasolines, widespread contamination of thousands of drinking water supply wells caused enormous environmental and economic costs (*1,2*). Due to its abundance in gasoline, high aqueous solubility, and slow degradation rate in aquifers, MTBE has migrated in significant quantities from subsurface gasoline releases to many municipal and private water wells across the U.S. in only a few years (*3*). Although the U.S. Environmental Protection Agency (EPA) mandated oxygenated fuel use in many regions in 1990 (*4*), some investigators had used qualitative language to warn about the potential threat of MTBE to groundwater resources as early as 1986 (*5,6*). Despite such early warnings and subsequent experiences with MTBE, an approach for making an *a priori* quantitative evaluation of widespread water supply well contamination from underground fuel tank (UFT) related releases or other sources has not yet been developed. These events demonstrate the clear need for regulators and industry to pre-evaluate all future gasoline additives and/or adjust gasoline composition for the corresponding potential to contaminate subsurface water supplies.

Extensive surveys of U.S. water supplies over the last decade have consistently reported a high incidence of community supply well contamination at levels of ~1 to 20 μg/L MTBE. According to a U.S. EPA collection of reports, in high oxygenate use areas, 5 to 15% of community drinking water supply wells had MTBE concentrations of ≥0.1 μg/L, and about 1% had MTBE concentrations in the range of 5 to 20 μg/L or higher (*7*). In a more systematic national sampling program of 579 community supply wells by the U.S. Geological Survey (USGS), the survey sample was designed to reflect a random national distribution of well sizes and population density, finding 5.4% of wells to have ≥0.2 μg/L MTBE concentrations, all less than 20 μg/L (*8*). In a study of the northeastern states, where oxygenate use is more common, workers used a similarly "stratified" sampling strategy and found 7.8% of community supply wells to have MTBE levels of ≥1.0 μg/L (*9*). MTBE concentrations as high as 610 μg/L were reported in Santa Monica community supply wells as a result of contamination from nearby leaking underground fuel tanks (LUFTs) (*10*). MTBE contamination of public or private drinking water wells appears to result largely from UFT and local homeowner fuel releases or refueling spills (*2*). It is worth noting that non-point source transport of MTBE from the atmosphere to shallow groundwater in urban areas has also been suggested as a contributor to pervasive contamination of drinking water supplies at sub-μg/L concentrations (*11-13*).

A useful assessment for identifying which gasoline compounds could cause widespread contamination of water supplies must incorporate chemical property information relating to subsurface transport. In principle, any component of gasoline might contaminate drinking water resources as a result of transfer from released gasoline to groundwater, followed by advective (groundwater flow) transport to a public or private well. In practice, most gasoline constituents are sparingly soluble in water, highly sorptive to aquifer solids and therefore retarded with respect to groundwater flow, or substantially biodegraded in the subsurface before migration to drinking water wells. Therefore, to make an upper-bound estimate of the threat posed by gasoline components to water supply wells, one could use a transport model that neglects biodegradation but utilizes information on equilibrium distributions of the contaminants between phases: primarily water, gasoline, air, and aquifer solids (*14,15*). The corresponding chemical properties are partition coefficients, K_{12}, which quantify the chemical's equilibrium mass distribution between two phases:

$$K_{12} = \frac{\text{concentration of solute in phase 1}}{\text{concentration of solute in phase 2}} \tag{2-1}$$

The partition coefficients relevant to subsurface transport of gasoline constituents therefore include the gasoline-water partition coefficient (K_{gw}), the air-water partition coefficient (K_{aw}), and the organic matter-water partition coefficient (K_{om}).

In a regulatory context, gasolines are conventionally viewed as mixtures of nonpolar hydrocarbons and any added oxygenates (*4,16-19*). However, closer scrutiny reveals that gasolines contain a suite of polar constituents at levels between ten to hundreds of part-per-million (ppm, Table 2-1). These compounds are either present in the original petroleum, are byproducts of refining processes, or are intentionally added in order to improve engine performance, clean and lubricate valves, increase octane number, improve emissions quality, preserve fuels during storage, or perform other functions (*20*). Such compounds could threaten drinking water resources widely if they are sufficiently mobile in the subsurface. As with MTBE, it is useful to consider whether the contamination risk posed by these compounds could be anticipated *a priori*, in other words, independently of existing contamination data. An analogous screening method for estimating urban air contamination levels by volatile gasoline constituents such as MTBE was proposed recently (*21*).

2.1.2. Objective

The goal of this study was to develop and test a method for evaluating the plausibility of significant contamination of community supply wells (CSWs) resulting from the use of current or future gasoline additives. For this purpose, I chose to make the conservative assumptions that (a) gasoline constituents were not (bio)degraded in the subsurface, and that (b) gasoline constituents did not influence each other during subsurface transport, i.e., through cosolvent effects or competitive biochemical oxygen demand. If reasonable, such a screening model would allow regulators and industrial researchers developing fuels to easily and quickly identify proposed constituents which are likely to cause CSW contamination, given their chemical partitioning properties, *before they are introduced to gasoline on a wide scale*. This screening model result would specifically identify those compounds which should be rigorously tested in the more difficult areas of toxicity, subsurface degradability, and interactions with other fuel constituents during transport.

Table 2-1. The abundance, physical-chemical properties, and transport model predictions of 24 polar and nonpolar compounds found in gasoline

		solute properties				model predictions	
gasoline solute	expected conc in fuel [ppm]	pK_a[a]	K_{gw}[b]	K_{ow}[c]	(est.) K_{om}[d]	t_{arr} [yr]	C_{well} [ppb]
methyl *tert*-butyl ether (MTBE)	100000[e]	-	16[n]	8.7	8.1	7	20
ethyl *tert*-butyl ether (ETBE)	100000[f]	-	210[o]	33[r]	25	9	9
di-*iso*-propyl ether (DIPE)	100000[f]	-	560[o]	33[r]	25	9	5
tert-amyl methyl ether (TAME)	100000[f]	-	90[o]	35[s]	26	9	10
methanol	106000[f]	15.3[l]	0.0051[n]	0.17	0.33	6	20
ethanol	105000[f]	15.9[l]	0.015[p]	0.49	0.77	6	20
benzene	12000[g]	-	220[n]	130	27	9	1
toluene	162000[h]	-	690[n]	540	110	20	5
ethylbenzene	73000[h]	-	2200[n]	1400	290	40	0.9
naphthalene	33000[h]	-	1500[n]	2000	410	60	0.4
aniline	20[i]	4.6[l]	3.1[i]	7.9	7.8	7	0.003
p-toluidine	30[i]	5.2[l]	12[i]	25	19	8	0.004
o-toluidine	20[i]	4.5[l]	12[i]	21	17	8	0.003
3,4-dimethylaniline	15[i]	5.2[l]	29[i]	110	46	10	0.001
2,6-dimethylaniline	15[i]	3.9[l]	39[i]	69	42	10	0.001
phenol	150[i]	9.9[l]	3.2[i]	30	22	9	0.02
p-cresol	100[i]	10.3[l]	9.3[i]	87	54	10	0.009
o-cresol	100[i]	10.3[l]	14[i]	89	55	10	0.009
3,4-dimethylphenol	30[i]	10.4[l]	22[i]	170	93	20	0.002
2,6-dimethylphenol	30[i]	10.6[l]	44[i]	230	120	20	0.002
N,N'-disalicylidene-1,2-diaminopropane	12[j]	11.8[l]	940[i]	37	26	9	0.0004
thiophene	150[k]	-	110[i]	78	49	10	0.01
benzothiophene	300[k]	-	1700[i]	1300	500	70	0.003
di-*sec*-butyl-p-phenylenediamine	20[j]	6.2[m]	1.1×10^{-7}[q]	7500	2100	200	6×10^{-8}

"-" indicates that this data is not applicable. [a] acidity constant. [b] gasoline-water partition coefficient, molar basis. [c] octanol-water partition coefficient, molar basis. literature values were taken from the ClogP database or calculated using the ClogP fragment method where literature values were unavailable (N,N'-disalicylidene-1,2-diaminopropane and di-*sec*-butyl-p-phenylenediamine) (*22-24*). [d] organic matter-water partition coefficient [L/kg], estimated using published LFERs with K_{ow} (*15*). [e] (*2*). [f] corresponds to ~10% vol/vol. [g] corresponds to 1% vol/vol, as imposed by current legislation. [h] (*25*). [i] (*26*). [j] (*20*). [k] (*27*). [l] (*28*). [m] calculated using SPARC (*29,30*). [n] (*31*). [o] calculated using UNIFAC (*32*). [p] (*33*). [q] calculated from AQUAFAC (*34*) estimation of the aqueous activity coefficients and UNIFAC estimation of the gasoline activity coefficient, with 10% MTBE. [r] (*35*). [s] (*36*).

2.2. Model development

2.2.1. Modeling scope and variability analysis

Conventional transport models are designed to treat site-specific contamination problems. This differs from the goal of the screening approach, which seeks to evaluate the hazard associated with a particular product used in typical situations. Thus, the question is: how can a transport model capture the threat posed by a contaminant at literally thousands of subsurface sites? Since the specific hydrogeologic and contamination conditions of these sites are not individually known, it is impossible to model each case of contamination for at-risk CSWs and then aggregate the results. Instead, it may be useful to focus attention on a subset of the at-risk sites that are representative of a significant portion of affected sites. The screening calculations here use the assumption that judiciously chosen parameters can reflect a typical at-risk site and thereby adequately describe representative contamination scenarios for screening purposes. If the model predicts CSW concentrations that are significantly lower than a safe threshold value, then the compound should be considered unlikely to cause a widespread contamination problem for the set of drinking water sources considered here. However, if the model predicts CSW concentrations which are near or above an unacceptable value, then this constituent should be considered a potential widespread contaminant. In this eventuality, degradability and/or toxicity tests should be performed before the chemical is added to gasoline.

Appraisal of model input parameter variability was used to estimate the uncertainty of predicted contaminant levels in CSWs, which revealed the model's precision and usefulness. Input parameter probability distributions were assumed to be log normal and/or were calibrated directly from literature data. Using uniform sampling Monte Carlo simulation (*37*), a large number (10^6) of stochastic realizations were used to evaluate the variability of transport model predictions, assuming that the input parameters were statistically uncorrelated. Consequently, the screening model was mainly used to produce two types of information: the predicted severity of a typical contamination event for an at-risk CSW (i.e., contaminant concentration in the well, or time until the onset of contamination); and an estimate of uncertainty around this expectation value.

Only three model parameters were compound-specific: the gasoline-water partition coefficient (K_{gw}); the organic matter-water partition coefficient (K_{om}); and compound concentration in gasoline (C_g) (Table 2-1). Where appropriate, compound acidity constant (pK_a) values were additionally determined in order to assess whether ionic species might have a significant impact on phase partitioning processes. The model was intentionally designed so that minimal additional information is required to tailor the model to the behavior of a novel gasoline constituent.

In this chapter, available data on MTBE contamination of CSWs were used to evaluate screening model predictions, since MTBE is known to be poorly degradable in many subsurface environments. Domestic wells were excluded from the set of sites under consideration because it is more difficult to make generalizations about their hydrogeologic environment. I believe that this does not undermine the usefulness of model results, since CSWs account for ~85% of groundwater-derived drinking water in the U.S. (*38*) and since CSWs appear to have been affected as severely as domestic wells in most cases (*7*). Additionally, the model does not consider contamination of CSWs resulting from atmospheric deposition. Finally, it should be noted that surface water sources have been affected by MTBE at least as much as or more severely than subsurface drinking water supplies in the U.S. (*8,39*), and the assessment proposed here does not address risks to surface water resources.

2.2.2. Conceptual transport model

The transport model was used to calculate: contaminant gasoline-water transfer; advective transport via groundwater flow; sorption to aquifer organic matter (retardation); longitudinal dispersion; and finally, dilution in the CSW. A simplified conceptual model of transport of the gasoline constituent from a release site to a CSW is first briefly presented, as three sequential steps, in order to guide the reader. A mathematical development of the transport concepts and an evaluation of the model assumptions are presented in the subsequent sections. First, a gasoline constituent was assumed to leach out of the gasoline non-aqueous phase liquid (NAPL) pool into passing groundwater, creating an "initial" plume condition. Dissolution equilibrium was assumed at the water-NAPL interface. The rate of contaminant flux from the NAPL to the groundwater, and therefore the initial length of the plume, was considered limited by the access of passing groundwater to the NAPL pool and the gasoline-water equilibrium partition coefficient of the constituent. Highly water-soluble components (such as MTBE) would leach out of the NAPL quickly and create a relatively short initial plume(e.g., (*40,41*)). In this case, the eventual final length of the calculated MTBE plume depended almost entirely on transport-related longitudinal dispersion and was influenced very little by the plume initial condition. Conversely, a less water-soluble contaminant such as toluene would dissolve slowly into passing groundwater, resulting in a lengthy "steady-state" plume which would not experience as much subsequent longitudinal mixing. In other words, the effective flux of the constituent from the NAPL into the groundwater (presented mathematically as the initial length of the plume) reflected a contribution to longitudinal dilution of the plume, and this eventually influenced the calculated rate at which the plume was transported into the CSW. Second, the calculated initial plume was assumed to migrate towards the well influenced by advection (rate of groundwater flow) and retardation (sorption to sediments), and the plume would additionally lengthen during transport via longitudinal dispersion, as depicted in Figure 2-1. Dispersion in the transverse

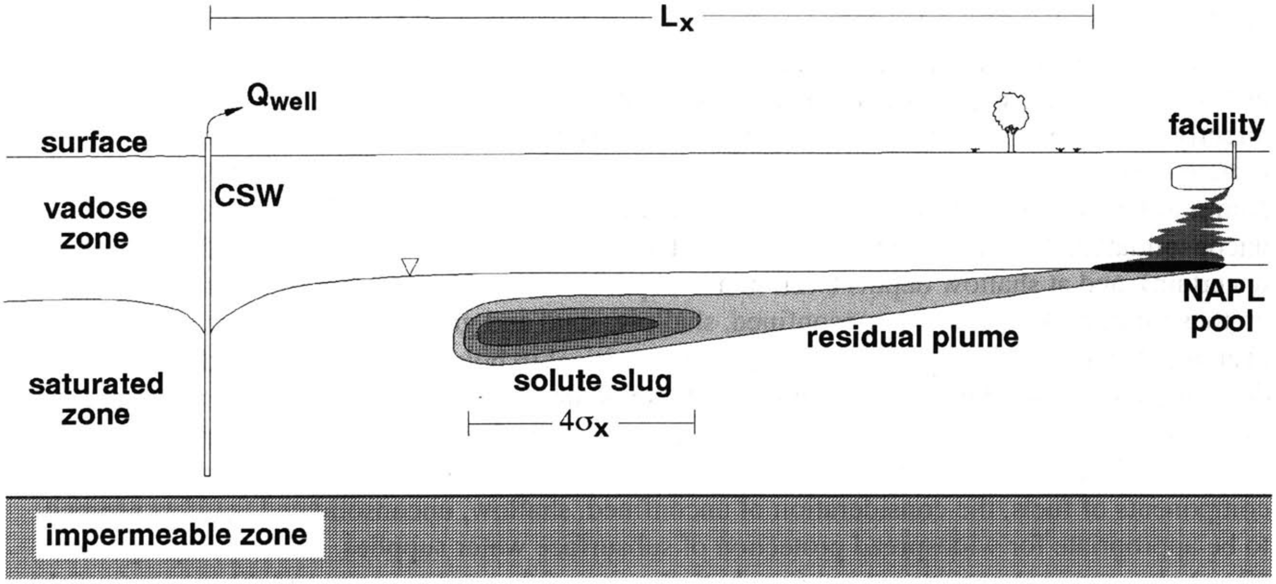

Figure 2-1. Depiction of a solute plume of length ~$4\sigma_x$ migrating from a gasoline release site to a CSW; the vertical scale is exaggerated to show detail

directions was considered unimportant since the entire plume was assumed to lie within the capture zone of the well. Consequently, the chemical flux into the well would be effectively decreased by dispersion only in the longitudinal coordinate, i.e., lengthening of the plume. Finally, the calculated plume arrived at and was drawn into a shallow CSW. The concentration of the constituent in the CSW water was thus determined by: (1) the final length of the plume when it arrived at the well (the result of both its initial length and dispersion-induced lengthening); (2) the rate at which the plume was fed into the well via ambient groundwater advection; and (3) the pumping rate of the well, which would effectively dilute the plume with the rest of the water in the capture zone.

2.2.3. Hydrogeologic setting

Before the screening model mathematical framework was developed, the appropriate scale and physical context of a gasoline component threat to a representative at-risk CSW were evaluated. First I considered hydrogeologic characteristics of CSW sites. While both consolidated and unconsolidated materials may contain productive aquifers (*42*), the large majority of CSWs in the U.S. are located in unconsolidated sand and/or gravel aquifers. Only a small fraction of wells are drilled into crystalline aquifers (*43*). Inspection of the geographic distribution of major productive aquifers throughout the coterminous U.S. (*44*) revealed that, while unconsolidated and consolidated aquifer regions appear roughly equally prevalent on an area basis, unconsolidated aquifers must serve by far the larger portion of the population which relies on groundwater. Productive unconsolidated aquifers predominate in the most populous regions, including most of the Atlantic seaboard and Gulf coastal plains, south Florida, much of the central plains (Ogallala aquifer) region, and most of the southwest alluvial basins and Pacific coast, as well as other areas. Additionally, even in areas that contain predominantly consolidated aquifers, the alluvial systems of river basins are probably relied upon heavily for community water supplies, due to their high yields. Using data collected across the U.S. in 23 different states during the early 1970's (*45*), 51% of public water supplies (27 out of 53 surveyed) had both screen depths of less than 85 m and were located in unconsolidated materials (sand or gravel). Separately, state agencies and other investigators have reported that significant portions of CSWs in several states (IL, NJ, MA, RI, MD, MT, and TX) or major regions therein draw directly from unconfined systems, ranging from ~30% to "virtually all" of them (*46-51*). In a more recent USGS nationwide survey of 575 CSW system managers, the large majority of communities reported CSW emplacement in unconsolidated lithology (63% consolidated, 7% unconsolidated, 29% uncertain), in unconfined aquifers (71% unconfined, 5% confined, 24% uncertain), and at shallow depths (well intake depth less than 76 m in 78% of cases) (*8*). Based on these studies, I assumed an unconfined, shallow, sand and gravel hydrogeologic system (Table 2-2) with a porosity (ϕ) of 0.25 (*52*). It is critical to note that many CSWs draw from deep or confined aquifers and consequently the screening model is not designed to represent such low-risk sites. Likewise, some communities may drill into fractured (e.g., karst) or otherwise highly conductive systems and thereby suffer increased risks. Nonetheless, for screening components of fuels, the consideration of unconfined, shallow, unconsolidated aquifers seemed to be appropriate for widespread protection of subsurface water supplies. CSW pumping rates (Q_{well}) also required consideration, since both the size of the capture zone and dilution rate of the plume as it is drawn into well both depend on well pumping rate. Grady's recent nationwide survey of 575 CSW systems revealed that CSW pumping rates follow a skewed distribution,

Table 2-2. Summary of model field transport parameters

			stochastic simulation statistics		
field parameter	symbol	expected value	M[a]	S[b]	constraints[c]
aquifer lithology	-	unconsol. sand, gravel	-	-	-
aquifer porosity	ϕ	0.25	-	-	-
aquifer fraction organic matter	f_{om}	0.003	-5.8	0.6	$f_{om} > 10^{-4}$
water table depth	d_w	7 m	-	-	-
aquifer saturated depth	H	25 m	-	-	-
well pumping rate	Q_{well}	2200 m^3/d (400 gpm)	7.7 (m^3/d)	1.0	$Q_{well} > 20$ gpm $Q_{well} < 5000$ gpm
distance from LUFT to CSW	L_x	1400 m	generated from data[d]		$L_x > 300$ m $L_x < 5000$ m
NAPL volume	V_g	1.65 m^3 (440 gal)	0.5 (m^3)	2.0	$V_g > 10$ gal
NAPL saturation	S_g	0.35	-1.0	0.2	$S_g > 0.05$ $S_g < 0.95$
NAPL lens thickness	h_g	0.2 m	-1.6 (m)	0.2	$h_g > 0.05$ m
longitudinal dispersivity	$a_{z,10}$	0.002 m	-6.0 (m)	0.9	-
ambient groundwater velocity	v_x	0.4 m/d	-0.9 (m/d)	0.5	$v_x > 0.01$ m/d
longitudinal dispersivity	a_x	20 m	3.0 (m)	0.5	$a_x > 2$ m $a_x \leq 10\%$ of L_x
aquifer solids density	ρ_s	2.5 kg/L	-	-	-
ground water temperature	T	15°C	-	-	-
ground water ionic strength	I	1 mM	-	-	-
ground water pH	pH	7	-	-	-

"-" indicates that a value or symbol was not assigned here. [a] stochastic mean of the log transformed parameter, assuming a log normal distribution. [b] standard deviation of the log transformed parameter, assuming a log normal distribution. For example, S = 2.3 indicates a standard deviation which corresponds to a factor of 10 in the non-transformed parameter. [c] constraints on the parameter values for stochastic simulations, so that unrealistically extreme or uninteresting parameter realizations are left out of the analysis. [d] stochastic estimates of CSW-LUFT separation distances were generated using equation 2-2, based on the LUFT density data of Johnson et al. (*2*) (not assumed log normal).

with 57% of wells having pumping rates of less than 70 gal/min and a significant number of wells being much larger systems (*8*). Based on this data and our own interviews with CSW managers (Table 2-3), a ln(Q_{well}) mean of 7.7 (units of m^3/day, corresponding to $Q_{well} \sim 400$ gal/min) and standard deviation of 1.0 were assumed.

Sediment organic matter content is known to have a significant influence on sorption and retardation of organic contaminants in the subsurface (*15*). Consequently an aquifer material organic matter (f_{om}) abundance of 0.3% was considered representative, since sand and gravel aquifers commonly contain relatively low levels of organic material, and this is also the risk-conservative choice for screening calculations (Table 2-2). In stochastic simulations, a ln(f_{om}) mean of -5.8 (corresponding to $f_{om} \sim 0.003$) and standard deviation of 0.6 were assumed.

Since transport in an unconfined sand and gravel aquifer could be comfortably established, ambient uniform flow of groundwater towards the well was assumed. In a review of field-study data, the median ambient groundwater velocity was 0.7 m/day in coarse-grained sand and/or gravel aquifers in the U.S. and Europe (n=16) (*53*). However, these observations were likely biased towards relatively high rates of groundwater flow, implying that more usual ambient groundwater velocities in sand and gravel aquifers are lower than 0.7 m/day (*54*). For example, the reported groundwater velocity on Cape Cod in a sand and gravel aquifer with relatively high recharge (~0.50 m/yr) was about 0.4 m/day (*55*). Tracer tests at the Borden site, a fine-grained sand aquifer, revealed an average groundwater velocity of about 0.09 m/day (*56*). Based on these considerations and the desire to remain risk-conservative in the screening assessment, a representative ambient groundwater velocity was chosen to be 0.4 m/day, with a typical range of 0.1 to 1 m/day reflected in the stochastic simulation statistics (Table 2-2). This groundwater velocity estimate did not include the drawdown effect of well pumping, which was accounted for separately as described in Section 2.6.

Using data describing the locations of reported LUFTs and 26000 CSWs in 31 states, Johnson et al. (*2*) estimated the distribution of CSW population as a function of LUFT density in the vicinity of the CSW. For stochastic simulations, the LUFT density histogram generated by Johnson et al. was used to construct an estimated cumulative frequency distribution, as shown in Figure 2-2. For a randomly selected realization of local LUFT density, the distance between the

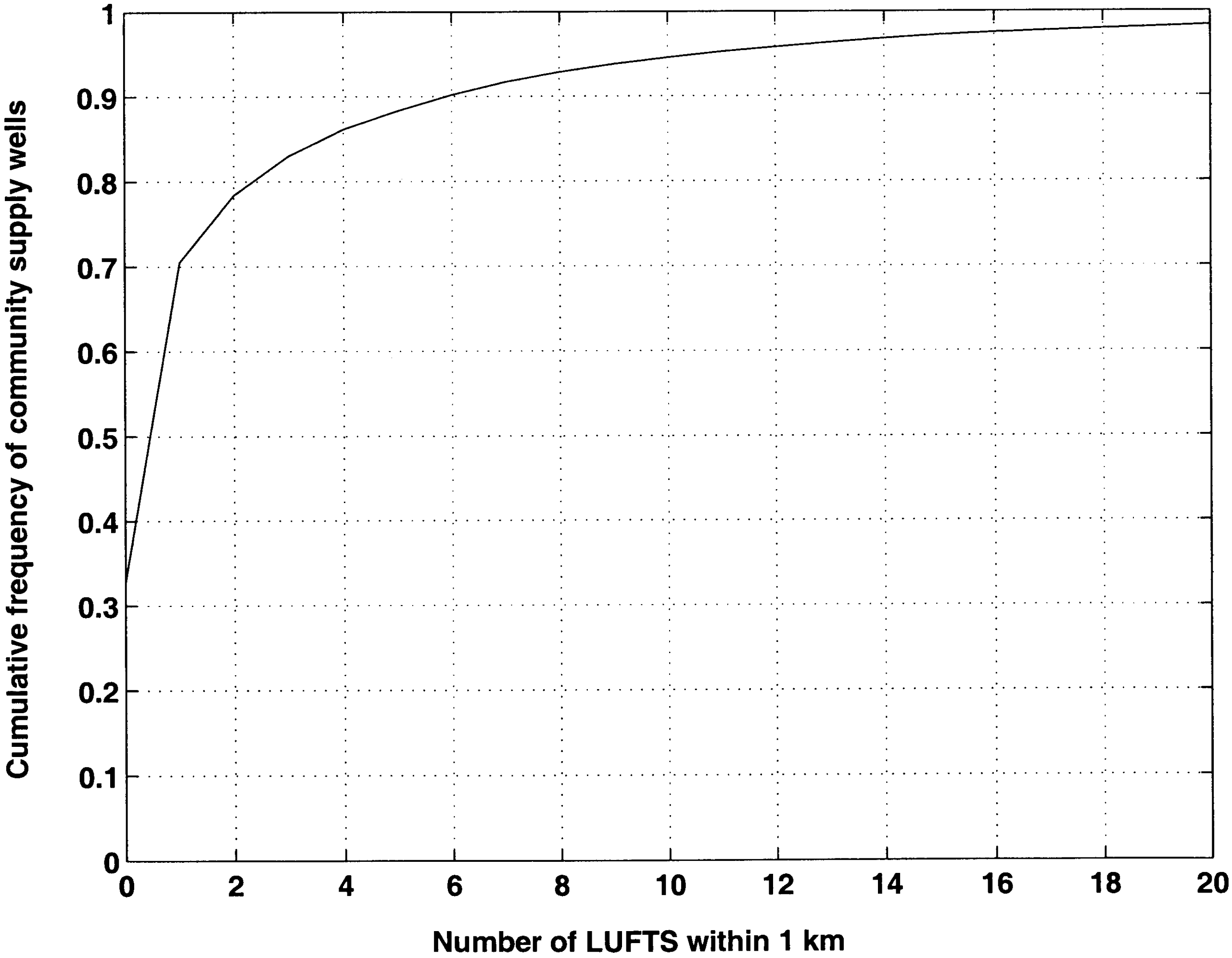

Figure 2-2. Estimated cumulative frequency distribution of LUFT density in the vicinity of CSWs, based on the histogram analysis provided by Johnson et al. (2000)

CSW and the nearest up-gradient LUFT was therefore determined as:

$$L_x \sim \frac{1}{2}\frac{(\text{area in which one LUFT will be found})}{(\text{capture zone width})} = \frac{1}{2}\frac{\left(\dfrac{1}{\rho_{\text{LUFTs}}}\right)}{\left(\dfrac{Q_{\text{well}}}{v_x \phi H}\right)} \tag{2-2}$$

where ρ_{LUFTs} is the (stochastic) local density of LUFTs [#/m^2], H is the saturated depth of the aquifer (assumed 25 m), and $Q_{\text{well}}/(v_x \phi H)$ is the capture zone width, assuming a uniform ambient flow field. The distance estimate in equation 2-2 assumes that the orientation of the capture zone is uncorrelated with LUFT density, which is likely to be biased for communities where wellhead protection efforts have guided the positioning of CSWs. Nevertheless, this distance estimate was considered adequate for screening purposes. For Monte Carlo simulations, a minimum L_x value of 300 m was imposed in order to avoid the spurious consideration of very close CSW-LUFT proximities. Additionally, in most populated regions, typical aquifer recharge (0.2 to 0.5 m/yr) will prevent the CSW capture zone from reaching more than a few km up-gradient under the conditions presented here. Consequently, a maximum L_x value of 5000 m was additionally imposed in stochastic simulations, since CSW contamination resulting from larger transport distances was considered unnecessarily unusual. Using these assumptions, the median stochastic L_x realization was about 1400 m and this value was considered representative for screening purposes.

The calculated kilometer-scale transport of contaminants to CSWs with screen depths of only tens of meters below the water table has an important modeling ramification. It has been shown that, except in systems of very low permeability, one may reasonably reduce unconfined 3-dimensional hydrology to a "flattened" 2-dimensional problem as long as the processes of interest involve a lateral scale which is at least a factor of 2 or 3 greater than the vertical scale (*57*). Since the screening model's lateral transport distances and capture zone lengths are generally an order of magnitude larger than the aquifer saturated thickness, the exact specifications of the well screen were therefore considered unimportant for the modeling presented here.

To assess whether the physical context established above was reasonable (Table 2-2), I obtained the specific data on the CSWs of eight communities distributed around the U.S. (Table 2-3). Three of the communities were located in geographical areas predominated by consolidated aquifer material (Chillicothe, OH; Idaho Falls, ID; Brush, CO), but they had drilled their community water supply wells in *un*consolidated alluvial areas. In six out of the eight cases, distances between CSWs and nearby UFTs were less than 1400 m. Reported CSW pumping rates ranged from 200 to 5000 gal/min, but clustered around 400 to 700 gal/min. In five of eight communities studied, well screening depths (d_{screen}) were relatively shallow ($\leq$130 m), suggesting vulnerability to contamination from UFT releases. The eight reported f_{om} values ranged from undetectable to 0.05, with a median value of 0.0015. Thus, these reports generally support the previously developed expectation that a significant fraction of sites across the U.S. reflect a CSW-to-LUFT separation scale (L_x) of about 1000-2000 m and CSW emplacement in shallow, unconfined, sand and gravel aquifers. The Chillicothe CSW site (Figure 2-3) provides a

Table 2-3. Summary of hydrogeologic and well data for eight randomly selected U.S. communities

location	aquifer material	L_x [m][a]	Q_{well} [gpm][b]	d_{screen} [m][c]	f_{om}[d]	references
Forestdale, MA	sand/gravel	< 200	200-350	10-18	0.0003	(*58,59*)
Guymon, OK	silt/sand/clay	300-700	80-900	130	0-0.01	(*60,61*)
Columbus, MS	sand/gravel	< 200	1400	300	0.0006	(*62,63*)
Chillicothe, OH	sand/gravel	700-900	900	20-30	0.002	(*64,65*)
Brush, CO	sand	~5000	600-1400	30-40	0.001	(*66,67*)
College Station, TX	sand	> 5000	200	1000	0-0.05	(*68,69*)
Idaho Falls, ID	sand	< 800	1000-5000	160	0.01	(*70,71*)
Toms River, NJ	sand/gravel	500	200-2000	30-70	0.0004	(*72,73*)

[a] distance from the CSW to the nearest known UFT, based on surveys of commercial service station locations. [b] CSW pumping rate. [c] approximate well screen depth. [d] aquifer material fraction of organic matter as measured at a regional location.

compelling illustrative example of the proposed model hydrogeologic context: underground fuel tanks lie within 0.5 to 1 km in almost any general direction, and the shallow wells are emplaced in a sand and gravel aquifer (Table 2-3). Actual contamination events or UFT releases were not investigated or reviewed in any of the communities listed here, and the reported results are not intended to imply any negligence or specific mismanagement on the part of planners in these areas.

2.2.4. NAPL release characteristics and contaminant plume initial conditions

Once the hydrologic setting and transport scale of CSW contamination by a gasoline release had been established, plume initial conditions were evaluated. First, I investigated gasoline release volumes (V_g). Inspection of several U.S. National Response Center (NRC) fuel release incident reports (*74*) revealed that terrestrial gasoline releases are frequently related to UFT refilling spills or tank/pipe leaks. A random survey of 50 NRC reports (1999-2003) of uncontained gasoline releases to land or soil environments, in which release volume estimates were recorded and were at least 10 gallons in size, was conducted. This sample of release reports reasonably matched a log normal distribution with a mean $\ln(V_g)$ value of 0.5 (units of ln m^3, corresponding to $V_g \sim 1.65$ m^3 or 440 gal) and a $\ln(V_g)$ standard deviation of 2.0 (Figure 2-4). The largest release in the randomly selected set was 16800 gal, although much larger recorded releases can be found in the database (up to ~10^5+ gal, based on our inspection). Nevertheless these estimates may be biased to low volumes, since reporting favors recognition of sudden release incidents. Many releases may be better described as slow leakages into the environment over long time frames (possibly years). Such cases do not need to be excluded from the scope of the screening model, since they can be treated as a series of smaller releases. Based on these considerations, a screening model release volume of 400 gal was assumed representative (Table 2-2).

Once released into the environment, the gasoline NAPL was assumed to percolate through the vadose zone and spread into a resting "pancake," or pooled lens, on the water table (Figure 2-5). The fraction of void space occupied by NAPL within the lens, referred to as the NAPL saturation (S_g), was assumed to average ~0.35 and could easily range from 0.2 to 0.5, based on a literature compilation of measured field residual saturation data for gasoline (*75*). The residual

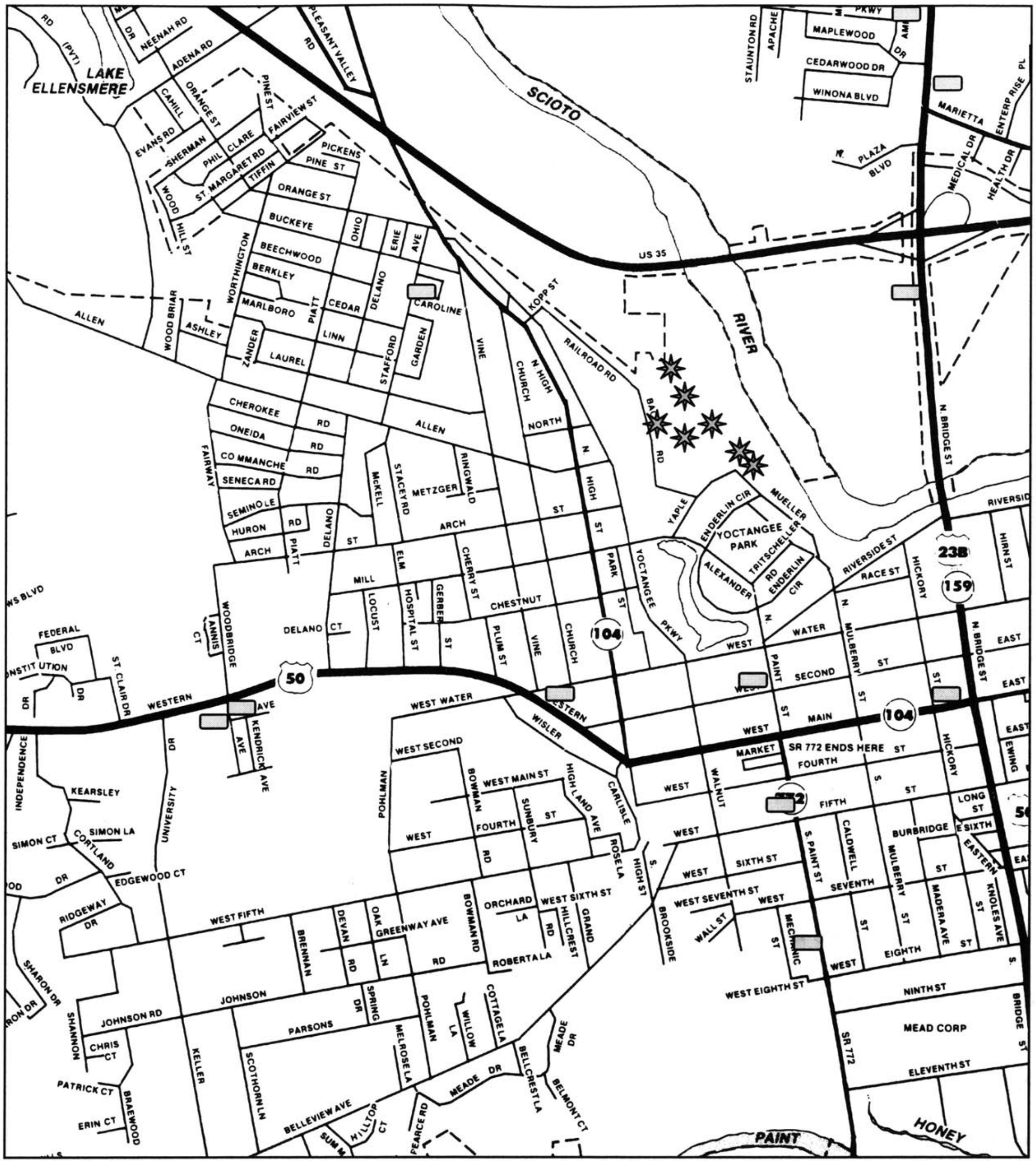

Figure 2-3. Map of the Chillicothe, OH, CSW site area, indicating the relative locations of CSWs (shaded stars) and service stations (shaded rectangles)

saturation data reflect a lower bound estimate of NAPL saturation at the water table. This is a risk conservative choice for screening purposes, and subsequent analysis (section 3.2) will show that forecasted community supply well contamination levels are relatively insensitive to the S_g value. The average height of the resting NAPL lens (h_g) was assumed to be 0.20 m. If the NAPL pool spreads in a circular fashion, then $V_g = h_g \pi r^2 S_g \phi$, and given a lens height of 0.2 m, NAPL saturation of 0.35, porosity of 0.25, and gasoline release volume of 1.65 m^3 (440 gal), the

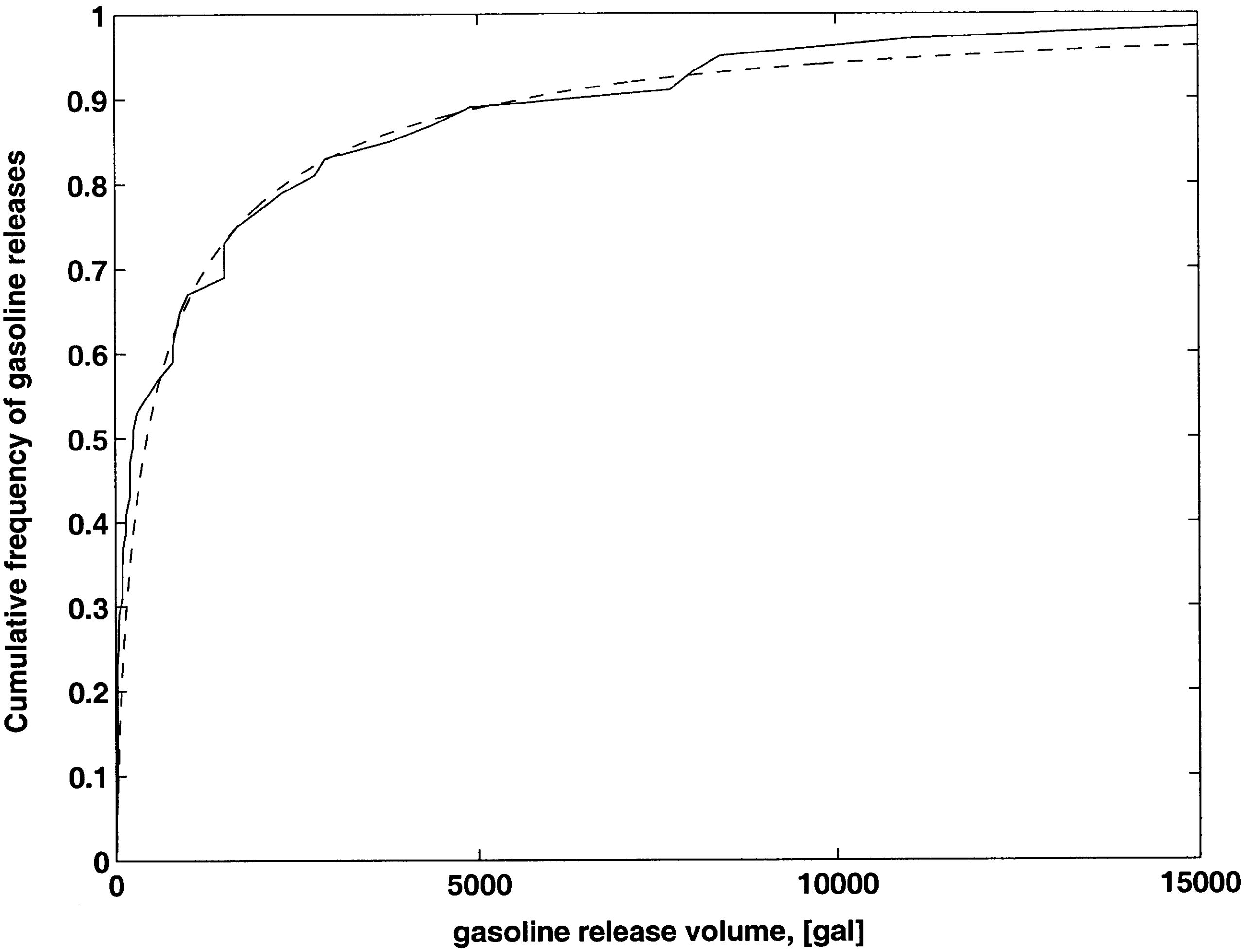

Figure 2-4. Cumulative frequency distribution of 50 reported uncontained gasoline releases to soil or subsurface environments based on National Response Center data (1999-2003), shown together with a suggested log normal distribution curve having M = 0.5 (m^3) and S = 2.0

corresponding lens radius would be r = 5.5 m.

Volatilization of gasoline components into the vadose zone and subsequent escape at the surface were considered, since this process might affect the contaminant's ability to impact a CSW. The rate of contaminant flux, $(dM_i/dt)_{volat}$, from the NAPL via vertical diffusion through pore gas (air) in the vadose zone was estimated as (*15*):

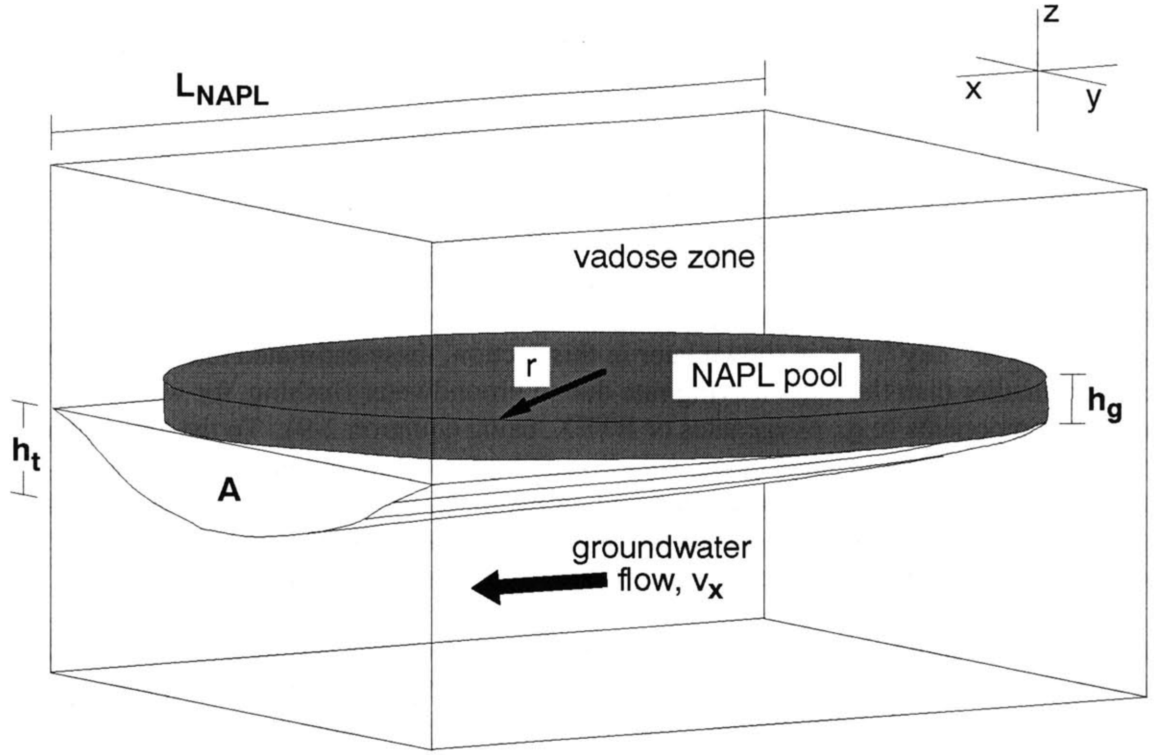

Figure 2-5. Conceptualization of the NAPL lens and contaminated groundwater as it flows beneath the pooled gasoline NAPL; the vertical scale is exaggerated to show detail

$$\left(\frac{dM_i}{dt}\right)_{volat} = \frac{D_a f_{i,a}}{\tau}\left(\frac{dC_a(z)}{dz}\right)\phi_a A_g = \frac{D_a f_{i,a}}{\tau}\frac{\left(\frac{C_g}{K_{ga}} - 0\right)}{(0 - d_w)}\phi_a A_g = -\frac{D_a f_{i,a}}{\tau}\frac{\left(\frac{M_i}{K_{ga}V_g}\right)\phi_a A_g}{d_w}$$

$$= -\frac{D_a}{\tau}\left(\frac{K_{aw}\phi_a}{K_{aw}\phi_a + \phi_w + K_{om}f_{om}\rho_s\phi_s}\right)\frac{\left(\frac{M_i\phi_a}{K_{ga}h_g S_g}\right)}{d_w} \qquad (2\text{-}3)$$

where M_i is the mass of the contaminant in the NAPL release, D_a is the diffusion coefficient of the contaminant in air, $f_{i,a}$ is the fraction of contaminant mass in the vadose material gaseous (air) phase (*15*), τ is the aquifer material tortuosity (assumed 1.5), $dC_a(z)/dz$ is the vertical vapor concentration gradient of the contaminant as it diffuses from the NAPL through vadose air to the land surface [kg/m^4], ϕ_a, ϕ_w, and ϕ_s are the air, water, and solids volume fractions of the vadose material, respectively (I assumed $\phi_a \sim 0.2$, $\phi_w \sim 0.1$, and $\phi_s \sim 0.7$), A_g is the area of the NAPL lens (plan view) [m^2], d_w is the distance from the NAPL pool at the water table to the land surface (taken as ~5 m for a shallow source), C_g is the abundance of the contaminant in gasoline [kg/m^3], K_{ga}, K_{aw}, and K_{om} are the gasoline-air partition coefficient (dimensionless), air-water partition coefficient (dimensionless), and organic matter-water partition coefficient [L/kg] of the contaminant, respectively, f_{om} is the vadose solids mass fraction of organic matter (assumed

0.003), and ρ_s is the vadose mineral solids density (assumed 2.5 kg/L). In order to evaluate the importance of volatilization relative to other loss mechanisms, I calculated the hypothetical fuel component loss rates due to diffusive escape to the surface for three volatile NAPL components: ethanol, MTBE, and ethylbenzene. Assuming $K_{ga} \sim K_{gw}/K_{aw}$, I estimated $K_{ga,ethanol} = 0.015/0.00020 = 75$, $K_{ga,MTBE} = 16/0.026 = 615$, and $K_{ga,ethylbenz} = 2200/0.34 = 6500$ (K_{gw} data from Table 2-1, K_{aw} data from (*12,31*)). Using correlations with molecular weight (*76*), I estimated $D_{a,ethanol} = 1.1$ m^2/day, $D_{a,MTBE} = 0.7$ m^2/day, and $D_{a,ethylbenz} = 0.7$ m^2/day. Taking these parameters and K_{om} predictions from Table 2-1, I estimated initial volatilization rate constants ($k_{volat} = -M_i^{-1}(dM_i/dt)_{volat}$) of $k_{volat,ethanol} \sim 2\times10^{-6}$ day^{-1}, $k_{volat,MTBE} \sim 2\times10^{-5}$ day^{-1}, and $k_{volat,ethylbenz} \sim 2\times10^{-6}$ day^{-1}. As is shown later in this section, these estimated rates of volatile escape are much smaller than the mass leaving rate due to groundwater flushing, for water-soluble gasoline components (e.g., oxygenates or BTEX, using equation 2-9). To the extent that this approximation is incorrect in some cases, volatilization was considered unlikely to change the contaminant mass available for groundwater contamination by more than a factor of 2. This uncertainty is small compared to our imprecise knowledge of subsurface gasoline release volumes; consequently gasoline volatilization was not included in the screening model.

Groundwater which could physically access the pooled gasoline was assumed to chemically equilibrate with the NAPL where the two fluids were in immediate contact, thereby creating an underlying plume of saturated water (Figure 2-5). This is a very reasonable assumption. Seagren and coworkers have used dimensional analysis to show that local equilibrium may be safely assumed when the product of the modified Sherwood number ($Sh' = L_{NAPL}\, k_l/D_z$) and Stanton number (k_l/v_x) is greater than 400 (*77*). Assuming a groundwater pore water velocity of $v_x = 0.4$ m/d, a vertical dispersion coefficient of $D_z = v_x a_{z,10} = 0.0008$ m^2/d, a mass transfer coefficient of $k_l = 1$ m/d (considered a reasonable value for an aquifer setting (*77*)), and NAPL pool length of $L_{NAPL} \sim 10$ m, the calculated $Sh'St \sim 30000 >> 400$. Thus the Sherwood-Stanton product would have to decrease by two orders of magnitude before conditions resulted in non-equilibrium mass transfer of gasoline components to passing groundwater. The depth of the groundwater equilibrated with the NAPL when it leaves the NAPL lense, $h_t(y)$, was estimated from the characteristic length of Fickian dispersion-induced vertical mixing (*78,79*):

$$h_t(y) = \sigma_z(z) = \sqrt{2a_{z,10}L_{NAPL}(y)} \qquad (2\text{-}4)$$

where $a_{z,10}$ is the vertical aquifer dispersivity on a ten-meter scale [m] and is assumed constant, and $L_{NAPL}(y)$ is the length of the NAPL pool [m]. Inspection of a review of literature vertical dispersivity data (*53*) suggested that $a_{z,10} \sim 0.002$ m was reasonable for a sand and gravel system, with a typical range of 0.0005 to 0.01 m reflected in the stochastic simulation statistics (Table 2-2). This results in a maximum $h_t(y)$ value of about 0.2 m.

From Figure 2-5 it is clear that if one could calculate the transverse area of the exiting plume, A_t, then the rate of mass loss of gasoline constituents leaving with passing groundwater could be estimated. If the gasoline NAPL spreads in an approximately circular fashion, the longitudinal length of NAPL to which the passing groundwater is exposed varies as $L_{NAPL}(y) = 2\sqrt{r^2 - y^2}$, where r is the radius of the NAPL pool. Consequently the transverse cross-sectional area of contamination leaving the NAPL pool could be found by integrating the depth of the plume as it leaves the NAPL over the width of the plume:

$$A_t = \int_{-r}^{r} h_t(y)dy = \int_{-r}^{r}\sqrt{2a_{z,10}L_{NAPL}(y)}dy = \int_{-r}^{r}\sqrt{2a_{z,10}2\sqrt{r^2 - y^2}}dy \qquad (2\text{-}5)$$

Since A_t is a symmetric function over –r to r, the integral limits can be simplified:

$$A_t = \int_{0}^{r} 2\sqrt{2a_{z,10}2\sqrt{r^2 - y^2}}dy = 4\sqrt{a_{z,10}}\int_{0}^{r}\sqrt[4]{r^2 - y^2}dy \qquad (2\text{-}6)$$

Equation 2-6 is a well-behaved function which can be numerically integrated with nominal error. In this work, the midpoint numerical integration method was used with N = 1000 intervals (*80*). An arbitrary fitting function was found (correlation coefficient = 1.000) which relates the computed value of A_t [m^2] to the NAPL pool radius, r [m]:

$$A_t = 3.5r^{1.5}\sqrt{a_{z,10}} \qquad (2\text{-}7)$$

Using the conditions suggested here for a 400 gal release ($r = 5.5$ m, $a_{z,10} = 0.002$ m), I calculated $A_t = 2.0\ m^2$.

The mass leaving rate of any compound from the NAPL pool was estimated as:

$$\left(\frac{dM_i}{dt}\right)_{flushing} = -Q_A C^{eq}_{water} = -A_t \phi v_x C^{eq}_{water} \qquad (2\text{-}8)$$

where Q_A refers to the flux of groundwater through area A_t at the NAPL edge [m^3/day], and C^{eq}_{water} is the aqueous contaminant concentration of the groundwater plume leaving the NAPL spill. Since contaminated water under the NAPL lense was assumed equilibrated with gasoline, $C^{eq}_{water} = C_g/K_{gw}$ and therefore:

$$\left(\frac{dM_i}{dt}\right)_{flushing} = \frac{-A_t \phi v_x C_g}{K_{gw}} = \frac{-A_t \phi v_x M_i}{K_{gw} V_g} \qquad (2\text{-}9)$$

Using the suggested parameters in Tables 2-2 and 2-1 and taking $A_t = 2.0\ m^2$, the initial rate constants of flushing ($k_{flushing} = -M_i^{-1}(dM_i/dt_{flushing})$) for three water-soluble fuel components were: $k_{flushing,ethanol} = 8\ day^{-1}$, $k_{flushing,MTBE} = 8\times10^{-3}\ day^{-1}$, and $k_{flushing,ethylbenz} = 6\times10^{-5}\ day^{-1}$. These mass leaving rates were compared to estimated loss rates due to volatilization (see the example calculations following equation 2-3). In all three cases, the ratio of $k_{flushing}$ to k_{volat} showed that mass loss to groundwater was the dominant process by at least an order of magnitude: $(k_{flushing}/k_{volat})_{ethanol} = 1\times10^6$, $(k_{flushing}/k_{volat})_{MTBE} = 5\times10^2$, and $(k_{flushing}/k_{volat})_{ethylbenz} = 3\times10^1$. This confirmed our assumption that volatilization could be neglected, for the purposes of screening CSW contamination by water-soluble gasoline constituents.

In order to integrate equation 2-9 and solve for $M_i(t)$, it was assumed that the NAPL volume does not change significantly as gasoline components are leached away from the release site. This is a reasonable screening model approximation for current gasolines: highly soluble components are fractionally abundant (20-40% v/v, including oxygenate and light aromatic constituents), and the assumption of constant NAPL volume may therefore reflect an error of order ~ $(1-\Delta V_g)^{-1}$ in dM_i/dt (a factor of 1.2 to 1.7). This is much smaller than the uncertainty due to variability in fuel release volumes. For hypothetical fuel mixtures which contain mostly (50%-75%) soluble components of interest, one could choose to explicitly treat changes in NAPL volume. Integration of equation 2-9 for $M_i(t)$ gives a first-order decay expression:

$$\frac{M_i(t)}{M_{i,0}} = \exp\left(\frac{-A_t \phi v_x t}{K_{gw} V_g}\right) \tag{2-10}$$

where $M_{i,0}$ is the total original mass of the contaminant in the gasoline. The amount of time required to deplete the gasoline of 80% of the compound (i.e., two to three half-lives), excluding a weak plume tail, is therefore:

$$t_{\text{depletion}} = \frac{K_{gw} V_g \ln(5)}{A_t \phi v_x} \tag{2-11}$$

Calculating $t_{\text{depletion}}$ allowed estimation of the initial length of the concentrated plume as it leaches away from the NAPL site. The initial plume length is important information because it reflects the extent of longitudinal dilution of the plume, and this will eventually affect the contaminant concentration in the CSW. Highly water-soluble components such as MTBE dissolved rapidly into groundwater, creating a short, concentrated plume. Conversely, less soluble components such as toluene leached slowly out of the NAPL, generating a long dilute plume. The initial length of the plume, approximated as a slug of uniform concentration, is then:

$$l_{x,\text{initial}} = \frac{v_x t_{\text{depletion}}}{R} = \frac{K_{gw} V_g \ln(5)}{R A_t \phi} \tag{2-12}$$

where R is the retardation factor of contaminant. The retardation factor reflects the decreased velocity of the contaminant as it is advected through the subsurface, due to sorption to aquifer organic matter (*81*):

$$R = \frac{v_x}{v_{\text{contaminant}}} = 1 + \frac{f_{om} K_{om} \rho_s (1-\phi)}{\phi} \tag{2-13}$$

where f_{om} is the aquifer material mass fraction of organic matter, K_{om} is the organic matter-water partition coefficient (L/kg), and ρ_s is the aquifer mineral solids density (kg/L).

2.2.5. Dispersion of the contaminant plume and dilution in the CSW

Once the initial plume length was estimated, transport-induced dispersion was expected to cause further dilution. Taking the plume to be a slug of uniform concentration as it leaves the NAPL site, its variance, $\sigma_{x,\text{ initial}}^2$, could be related to the 2nd moment of its length, $(l_{x,\text{initial}})^2/12$. Using the assumption of Fickian dispersion (*82*), the final longitudinal variance of the plume could be described by summing its initial variance and the dispersion-induced variance:

$$\sigma^2_{x,\text{final}} = \sigma^2_{x,\text{initial}} + \sigma^2_{x,\text{dispersion}} = \frac{l^2_{x,\text{initial}}}{12} + 2a_x L_x \tag{2-14}$$

where a_x is the subsurface longitudinal dispersivity. The final estimated length of the transported plume (expressed as 4 plume standard deviations) when it reaches the CSW was therefore:

$$l_{x,\text{final}} = 4\sigma_{x,\text{final}} = 4\sqrt{\frac{l^2_{x,\text{initial}}}{12} + 2a_x L_x} \qquad (2\text{-}15)$$

Literature data were consulted to estimate a_x. In a review of field studies (*53*), the median measured longitudinal dispersivity was 20 m (n=7) in sand and gravel systems having field transport scales of 500 to 2000 m. From these data and subsurface dispersivities used by the U.S. Environmental Protection Agency Composite Model for Landfills (EPACML) (*82*), a transport model longitudinal dispersivity of 20 m was considered representative. In stochastic simulations, a mean $\ln(a_x)$ value of 3.0 and standard deviation of 0.5 (Table 2-2) were assumed.

Having derived an estimate of the plume's length when it arrives at the CSW, the rate of contaminant flux into the CSW could be estimated directly as:

$$\left(\frac{dM_i}{dt}\right)_{\text{into well}} = \left(\frac{\text{plume mass}}{\text{plume length}}\right)(\text{plume velocity}) = \left(\frac{0.8M_{i,0}}{l_{x,\text{final}}}\right)\frac{v_x}{R} \qquad (2\text{-}16)$$

I remind the reader that the "0.8" coefficient to $M_{i,0}$ reflects the choice of mass depletion lifetime in the discussion following equation 2-10. Equation 2-16 implies that the capture zone of the CSW contains the entire plume in addition to dilution water. This is a realistic expectation. For an aquifer having a saturated depth (H) of 25 m, the calculated CSW capture zone width would be $Q_{well}/(v_x \phi H) \sim 900$ m. Conversely, the lateral spread of the plume may be of order ~100 m when it reaches the CSW, as calculated from a transverse dispersivity value of 1 m (*53*). Consequently, explicit consideration of transverse dispersion was not included in the model. The CSW concentration was estimated from the flux of contaminant into the well and the dilution resulting from pumping additional (uncontaminated) surrounding water:

$$C_{\text{well}} = \frac{\left(\frac{dM_i}{dt}\right)_{\text{into well}}}{Q_{\text{well}}} \qquad (2\text{-}17)$$

Equation 2-17 could be formulated in terms of measurable parameters (Tables 2-1, 2-2) by combining it with equations 2-16, 2-15, 2-12, and 2-7, and recognizing that $V_g = h_g \pi r^2 S_g \phi$ and $M_{i,0} = C_g V_g$. Thus, C_{well} was estimated as:

$$\boxed{C_{\text{well}} \approx \frac{0.2\left(\frac{v_x C_g V_g}{RQ_{\text{well}}}\right)}{\sqrt{0.1\left(\frac{K_{gw}}{R}\right)^2\left(\frac{V_g}{\phi}\right)^{\frac{1}{2}}\frac{(h_g S_g)^{\frac{3}{2}}}{a_{z,10}} + 2a_x L_x}}} \qquad (2\text{-}18)$$

where the contaminant retardation factor, R, is described by equation 2-13. The first term in the denominator of equation 2-18 describes dilution of the plume resulting from leaching out of the NAPL spill, whereas the second term in the denominator describes dilution of the plume related to dispersion during transport. Using the transport parameters suggested here (Table 2-2), the second term (transport-induced dispersion) will dominate and the first term can probably be ignored for contaminants having a $K_{gw}/\mathrm{R} << 150$; i.e., highly water-soluble contaminants such as ethanol. Conversely, for contaminants with a $K_{gw}/\mathrm{R} >>150$ (i.e., sparingly soluble contaminants such as ethylbenzene), the first term (the plume initial condition) will dominate and the second term can probably be ignored.

I briefly point out that log K_{gw} measurement error could be incorporated into the uncertainty analysis, without the need for stochastic simulations. Assuming that log K_{gw} and log $\mathrm{C_{well}}$ variability were both normally distributed, the K_{gw} contribution to uncertainty in $\mathrm{C_{well}}$ forecasts could be estimated from the propagated error of equation 2-18, as derived in chapter 5:

$$\sigma_{\log \mathrm{C_{well}}}\left(K_{gw}\right) \approx \sigma_{\log K_{gw}}\left[\frac{\partial \log \mathrm{C_{well}}}{\partial \log K_{gw}}\right] = -\sigma_{\log K_{gw}}\left(1+20\mathrm{a_x L_x a_{z,10}}\left(\frac{\mathrm{R}}{K_{gw}}\right)^{2}\left(\frac{\phi}{\mathrm{V_g}}\right)^{\frac{1}{2}}\left(\mathrm{h_g S_g}\right)^{-\frac{3}{2}}\right)^{-1} \qquad (2\text{-}19)$$

Using reported $\sigma_{\log K_{gw}}$ estimates (*31,33*) and equation 2-19, calculated $\sigma_{\log C_{well}}\left(K_{gw}\right)$ values were $(\pm 0.05)\times(-9\times10^{-9}) = \pm 5\times10^{-10}$ for ethanol; $(\pm 0.09)\times(-0.008) = \pm 7\times10^{-4}$ for MTBE; and $(\pm 0.06)\times(-0.8) = \pm 0.05$ for ethylbenzene. K_{gw} error therefore did not significantly affect $\mathrm{C_{well}}$ forecast variability, relative to other model parameters (section 3.2). In future applications of the model for other solutes, however, it may become necessary to explicitly include this term in the variability analysis.

2.2.6. Estimated contaminant time of arrival at the CSW

The estimated time of subsurface contaminant transport to the CSW was calculated from: (a) advection of the contaminant plume due to ambient groundwater flow; (b) retardation of plume advancement due to sorption on aquifer solids; (c) longitudinal dispersion, which effectively accelerates the plume front; and (d) the pumping action of the CSW, which accelerates groundwater velocity in the drawdown area surrounding the well. For a conservative (unretarded) solute, the expected time of plume arrival at the CSW ($\mathrm{t_{arr}}$) could be formulated as the time required for plume advection plus (acceleration) corrections for dispersion of the plume front and CSW drawdown:

$$\mathrm{t_{arr}^{conservative}} = \mathrm{t_{arr}^{advection}} - \Delta \mathrm{t_{dispersion}} - \Delta \mathrm{t_{well\ action}} \qquad (2\text{-}20)$$

$\Delta \mathrm{t_{dispersion}}$ was related to the characteristic length of longitudinal dispersion at the plume front, and $\Delta \mathrm{t_{well\ action}}$ was estimated from the analytical expression for groundwater travel time to a steady state pumping well in a uniform flow field with homogeneous hydraulic conductivity (*83*), so that:

$$t_{\mathrm{arr}}^{\mathrm{conservative}} = \frac{L_x}{v_x} - \frac{\sigma_{x,\mathrm{dispersion}}}{v_x} - \frac{\beta}{v_x^2}\ln\left(1+\frac{v_x\left(L_x - \sigma_{x,\mathrm{dispersion}}\right)}{\beta}\right) \tag{2-21}$$

where $\beta = Q_{\mathrm{well}}/(2\pi\phi H)$ and $\sigma_{x,\mathrm{dispersion}} = \sqrt{2a_xL_x}$. Correcting the calculated conservative transport velocity for plume retardation, the plume travel time was estimated as:

$$\boxed{t_{\mathrm{arr}} \approx \frac{R}{v_x}\left(L_x - \sqrt{2a_xL_x} - \frac{\beta}{v_x}\ln\left(1+\frac{v_x\left(L_x - \sqrt{2a_xL_x}\right)}{\beta}\right)\right)} \tag{2-22}$$

where the contaminant retardation factor, R, is described by equation 2-13.

2.2.7. Corrections for temperature, salinity, and acid-base chemistry

Using a sand and gravel hydrogeologic context, other aquifer properties relevant to partitioning and transport such as groundwater temperature, ionic strength, and pH were considered (Table 2-2). A nationwide survey of 100 sites in the U.S. revealed that groundwater usually reflects local annual mean temperatures, millimolar ionic strengths, and pHs suited to quartzitic (pH 5 to 7) to carbonate (pH 7 to 9) aquifer solids (*45*). Consequently, I chose screening solution conditions to be 15°C, 1 mM ionic strength, and pH of 7. The groundwater temperature (15°C) and ionic strength (1 mM) do not differ enough from conditions in which physical chemical properties are usually measured (i.e., 20-25°C and zero ionic strength) to warrant adjustments to the partition coefficients used for screening purposes. However, gasoline components with acidity constant (pK_a) values near groundwater pH may significantly ionize, resulting in either (a) enhanced transport due to greater aqueous solubility of charged species or (b) retarded transport due to ion-exchange with aquifer solid exchange sites. In the hypothetical case where a contaminant has been identified as a serious potential threat to CSWs and ion-exchange could clearly play a role, it may be advisable to include this effect. However since ion-exchange was generally expected to decrease contaminant risk to water resources, I opted not to include it in the screening model for sake of simplicity. Conversely, including acid-base chemistry corrections to the estimated gasoline-water partition coefficient was considered both a risk-conservative decision and a reasonably straightforward calculation. Potentially ionizable gasoline solutes were assumed to partition between the NAPL phase (phase 1; gasoline) and groundwater (phase 2) at chemical equilibrium as described by an "effective" partition coefficient ($K_{\mathrm{gw,\,eff}}$):

$$K_{\mathrm{gw,eff}} = f_{\mathrm{neutral}}K_{\mathrm{gw}} = \left(\frac{1}{1+10^{\pm(\mathrm{pH}\text{-}\mathrm{pKa})}}\right)K_{\mathrm{gw}} \tag{2-23}$$

The correction factor, f_{neutral}, reflects the fraction of solute present as the nonionic species in the aqueous phase. The adjusted partition coefficient ($K_{\mathrm{gw,\,eff}}$) was therefore defined to reflect mass distribution of the ionic plus nonionic solute in water versus the nonionic solute in the organic phase. This formulation of $K_{gw,\,eff}$ therefore assumed that the ionic species does not appreciably partition into the less polar phase (gasoline), again yielding a risk-conservative result. In equation 2-23 the sign of the term, (pH-pK_a), is positive if the pK_a corresponds to cases in which the neutral species acts as a proton-donor, and it is negative if the pK_a corresponds to situations in which the neutral species acts as a proton-acceptor. For the most of the gasoline additives examined here, solute proton transfer could be ignored because $10^{\pm(pH-pKa)} << 1$ and the term $f_{neutral}$ collapsed to unity.

2.3. Results

2.3.1. Model evaluation using MTBE data

The screening model formulation and parameterization was evaluated by comparing the model's predictions to reported MTBE contamination of CSWs. Since MTBE is believed to degrade slowly in most aquifers, neglecting biodegradation should not severely bias the screening model results. Using the deterministic parameterization (i.e., average or median parameter values), the model predicted a MTBE front arrival time of 7 years and CSW concentration of almost 20 µg/L for MTBE (Table 2-1, Figure 2-6). Expected model precision was evaluated using stochastic variation of model parameters with Monte Carlo simulations. Stochastic simulations produced a normal distribution of ln(C_{well}) predictions with a mean $M = 2.9$ for MTBE, corresponding to $C_{well} = e^M = 20$ µg/L. The stochastic ln(C_{well}) standard deviation was $S = 2.1$, corresponding to an $e^{M\pm S}$ range of 2 µg/L to 150 µg/L. In other words, a typically predicted C_{well} value for MTBE was ~20 µg/L and could easily vary by an order of magnitude in either direction. By comparison, various CSW surveys in high oxygenate use areas have found 8% of CSWs contaminated by MTBE at >1 µg/L levels, ~1% of CSWs contaminated by MTBE at >5 µg/L levels, and zero to ~0.4% of CSWs contaminated by MTBE at >20 µg/L levels, respectively (*7,9*). Since MTBE has only been widely used since about 1990, the screening model appears to have correctly captured the "rapid CSW response" and the order-of-magnitude MTBE concentrations for many at-risk CSWs (~20 µg/L in oxygenate high-use areas). Stochastic variability forecasts were also realistic. For example, while survey statistics suggest that hundreds of CSWs may have been contaminated at ~10 µg/L levels in the U.S., to our best knowledge, the most severe case recorded in the literature was only at 610 µg/L (a Santa Monica CSW (*10*)). Stochastic analysis of the screening model gave satisfactory agreement with this observation, predicting that C_{well} values greater than $e^{M+2S} = 1200$ µg/L should be unusual. Interpretation of predicted sub-µg/L CSW contamination was confounded by the fact that atmospheric deposition probably contributes to MTBE contamination of groundwater at sub-µg/L levels in many urban areas (*11-13*). Consequently it was difficult to relate the stochastic modeling forecast with observed incidences of low-level CSW contamination by MTBE. Had these screening results been available in the 1980s, they might have been used to trigger further study of MTBE degradability and toxicity before regulators and industry decided to utilize the additive in large quantities. However, here I conclude that the model formulation and parameter estimates appear to capture what history has shown to be the final result.

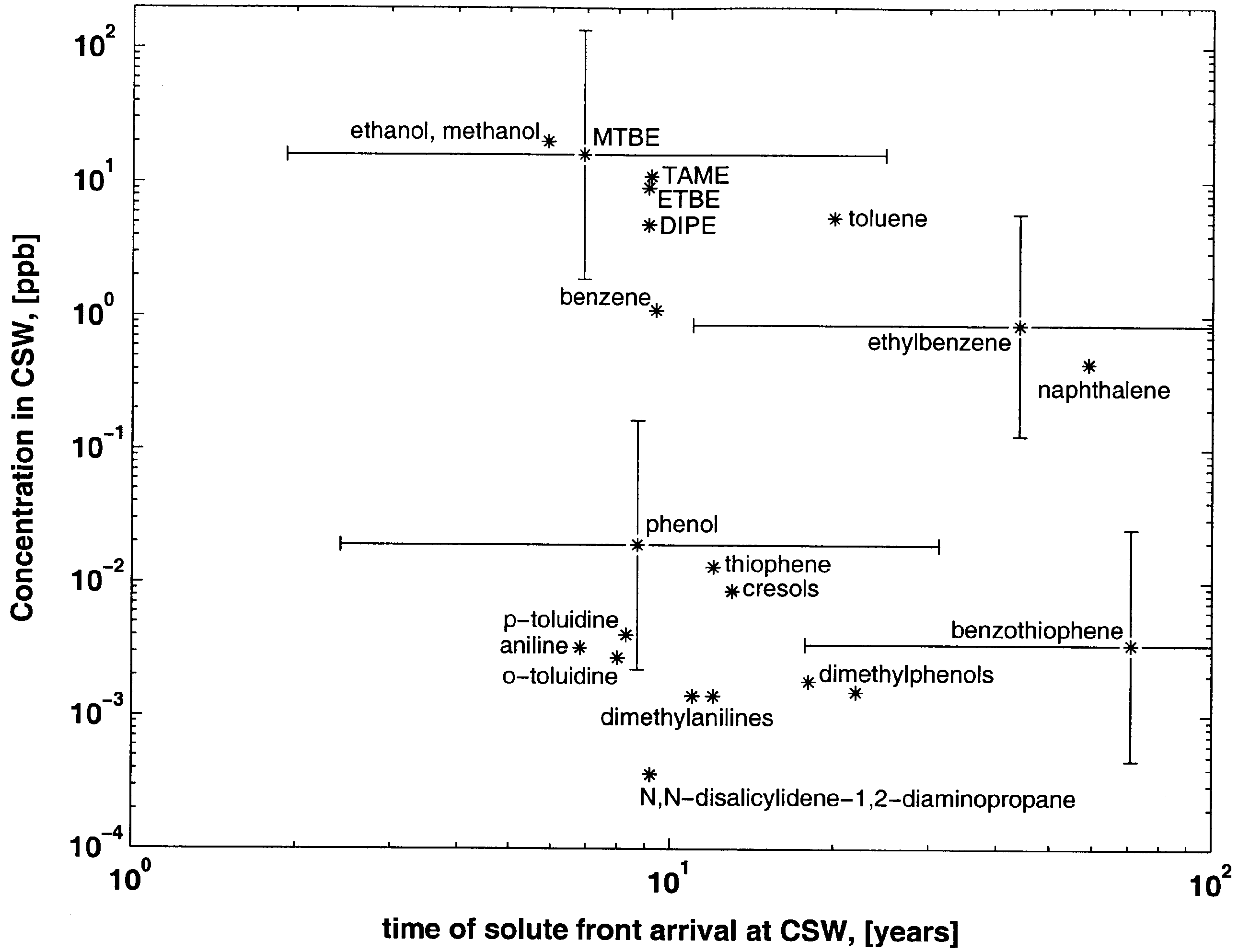

Figure 2-6. Predicted CSW concentration and front arrival time for 24 gasoline constituents. Expected variability is depicted for a few example compounds as $\pm e^{S}$ (one log normal standard deviation), based on Monte Carlo estimates

2.3.2. Interpretation and sensitivity analysis of model predictions

To demonstrate use of the screening model, the expected CSW concentrations and plume front arrival times of 24 gasoline constituents were calculated using equations 2-18 and 2-22 (Table 2-1, Figure 2-6). Solutes predicted in CSWs at substantial concentrations (C_{well} ~ 1 μg/L or more) were considered most likely to threaten municipal groundwater supplies in the absence of degradation processes. Monte Carlo simulations showed that an individual C_{well} realization could easily vary by an order of magnitude (×7 to ×8.5) from the expected value given by the deterministic model. Decision making analysis should take this breadth of the C_{well} distribution into account. For example, the expected CSW concentration of almost 20 μg/L for MTBE fell on the U.S. EPA advisory drinking water limit of 20 ppb, and the predicted concentration of 1 ppb for benzene was near the federal maximum drinking water threshold of 5 ppb. If such calculations had been pre-use forecasts, they would clearly indicate that work should be done to verify the prevalent and rapid degradability of MTBE and benzene in the subsurface before these compounds are allowed in gasoline. Conversely, the predicted 5 ppb well concentration for toluene falls far below its federal drinking water limit of 1 ppm, even neglecting biodegradation. Naphthalene and ethylbenzene also fell into this category (C_{well} ~ 0.5 ppb for naphthalene and ~1 ppb for ethylbenzene). Since the EPA-recommended long-term consumption threshold for naphthalene is 20 ppb and the ethylbenzene federal drinking water limit is 400 ppb, the screening model suggests these gasoline components would not deserve priority attention. Rulings for the permissible drinking water concentrations of DIPE, TAME, ETBE, methanol and ethanol have not yet been made by the EPA. But if there is concern for any of these compounds at CSW levels below ~200 ppb (expected $C_{well}\times10$), then the screening model output suggests that work is needed to confirm their prevalent degradability belowground before they are used in gasoline. Compounds that were expected to be at lower concentrations in CSWs (e.g., $C_{well} << 1$ ppb) might be considered as generally unlikely to create a widespread threat to drinking water supplies. Ultimately, however, toxicological expertise should be used to rigorously determine acceptable C_{well} levels for each case.

Expected plume arrival times at CSWs (t_{arr}) calculated using equation 2-22 varied from 6 to more than 200 years for the set of gasoline constituents considered here (Table 2-1). However, model parameter variability analysis suggested that arrival time estimates were highly uncertain (Table 2-4, Figure 2-6). Stochastic simulations suggested that the arrival times for individual CSWs may be only 1-2 years for relatively water-soluble gasoline constituents (e.g., oxygenates), and as short as ~10 years for some hydrophobic compounds which were retarded in the subsurface (e.g., ethylbenzene). From a decision making standpoint, I therefore suggest that pre-evaluations of gasoline constituents should generally verify subsurface degradability on a months-to-years time frame, depending on their predicted time of arrival at CSWs.

Table 2-4. Variability of C_{well} and t_{arr} for MTBE and ethylbenzene as related to stochastic transport parameters

	MTBE				ethylbenzene			
stochastic	C_{well}		t_{arr}		C_{well}		t_{arr}	
parameter	S	e^S	S	e^S	S	e^S	S	e^S
All parameters	2.14	8.5	1.23	3.6	1.92	6.8	1.37	4.0
V_g	1.86	6.4	-	-	1.52	4.6	-	-
Q_{well}	0.97	2.6	0.25	1.3	0.83	2.3	0.25	1.3
L_x	0.38	1.5	1.08	2.9	0.08	1.1	1.08	2.9
v_x	0.50	1.6	0.39	1.5	0.43	1.5	0.39	1.5
a_x	0.25	1.3	0.07	1.1	0.06	1.1	0.07	1.1
f_{om}	0.11	1.1	0.11	1.1	0.18	1.2	0.51	1.7
S_g	0.00	1.0	-	-	0.11	1.1	-	-
h_g	0.00	1.0	-	-	0.11	1.1	-	-
$a_{z,10}$	0.01	1.0	-	-	0.31	1.4	-	-

S indicates the estimated log normal standard deviation of the C_{well} or t_{arr} distribution resulting from fluctuations in the indicated stochastic field parameter or set thereof. e^S therefore expresses the multiple of C_{well} or t_{arr} which corresponds to one standard deviation of its log normal distribution. For example, ethylbenzene has an expected value C_{well} ~ 0.9 ug/L and a 1 standard deviation range around this value of $(e^{-1.92})0.9 = 0.1$ μg/L to $(e^{1.92})0.9 = 6.1$ μg/L for the case where all field parameters were treated as stochastic.

Additional analysis revealed that gasoline release volume, CSW pumping rate, and ambient groundwater velocity explained most of the variability of stochastic C_{well} realizations for a given gasoline constituent (Table 2-4). These three terms (V_g, Q_{well}, and v_x) were expected to control the uncertainty of C_{well} predictions, since they are linear with respect to C_{well} in equation 2-18 (whereas all other stochastic parameters are sub-linear in relation to C_{well}) and since they are all highly variable (Table 2-2). Subsurface parameters L_x and a_x accounted for second-order uncertainty in C_{well} predictions for relatively water-soluble compounds such as MTBE (Table 2-4), since these terms control the plume dilution of compounds with low K_{gw} values (as described in Section 2.5). Conversely, for less water-soluble compounds such as ethylbenzene, second-order C_{well} variability was linked to $a_{z,10}$ and f_{om}, that is, parameters that controlled the initial condition and subsurface retardation of a hydrophobic plume.

Sensitivity analysis results showed that CSWs serving small populations, or even small, private wells, appear likely to suffer higher contaminant concentrations due to their lower pumping rates and reduced plume dilution. It is important to bear in mind, however, that the smaller capture zones of small CSWs or private wells make it less likely they will encompass the entire contaminant plume. As a result of these conflicting factors, the predicted result for a smaller-flow well is unclear. This highlights the fact that the general screening treatment developed here does not necessarily directly apply to domestic wells.

2.3.3. Predicted behavior of BTEX and oxygenates

The screening model results were only partly consistent with reported behavior of BTEX (benzene, toluene, ethylbenzene, and xylenes) in field studies. Both benzene (1 ppb) and toluene (5 ppb) were estimated to appear in CSW water on a relatively short time frame (~10 years or less). The field parameter choices used here estimated a retardation factor of R ~ 1.6 for benzene, which is comparable to observed retardation factors of 1.2 to 1.3 for benzene in field studies (*41,84*). Hence, since these aromatic hydrocarbons have always been gasoline components, the screening model suggests that they should be already seen widely in CSWs. However, field surveys indicate that far fewer CSWs are contaminated by benzene or toluene than by MTBE (*2,44*). However, this disparity is not surprising, since several investigations have shown that BTEX components are biodegradable in many circumstances (*85-88*). I emphasize that degradation processes were not considered in the screening model, so these BTEX observations could not be captured. Presumably screening results like those for BTEX would be followed by additional work incorporating the degradability of these compounds, and corresponding decreased estimates of risk would be found.

The oxygenates ETBE, DIPE, TAME, ethanol, and methanol were predicted to behave similarly to MTBE in the subsurface. Like MTBE, they were predicted to partition preferentially from gasoline to water and migrate quickly (essentially unretarded) through aquifer sediments. Thus, in the absence of biodegradation processes, it is clear that these compounds would create a similar magnitude of CSW contamination as MTBE, depending on the extent and quantity of their use. As common natural products, methanol and ethanol would likely degrade quickly in the subsurface, and this expectation is consistent with observations (*89,90*). However, these alcohols may also need to be viewed as substantial sources of biochemical oxygen demand in the saturated zone. Thus, their inclusion in gasoline should inspire study of their ability to substantially change subsurface chemical properties and secondarily affect the transport of other solutes, such as benzene, whose biodegradability under anoxic conditions is far from certain (*91*). An ethanol-rich plume traveling away from a gasoline release site could quickly create an anaerobic wake through which other anaerobically persistent solutes may follow and substantially contaminate water supplies (*89,92*). The transport screening model results show that benzene could contaminate wells at levels (1 ppb) near the federal drinking water standard (5 ppb) within a few years, in the absence of subsurface biodegradation. Therefore, adding ethanol or methanol to fuels could severely upset this balance and conceivably result in benzene contamination of drinking water on a costly national scale.

2.3.4. The influence of acid-base chemistry on transport predictions

It should also be noted that some gasoline constituents with acid/base moieties (pK_as in Table 2-1) can significantly ionize under normal aquifer conditions (pH ~ 5 to 9). If a contaminant is partly ionized in groundwater, it may have an enhanced effective equilibrium aqueous concentration, since the ionic species will not appreciably partition into fuel (equation 2-23). The screening model did not address the potential effects of solute ion exchange reactions with aquifer solids. For example, using a cation-exchange sorption model described elsewhere (*15*), I estimated that the solids-water partition coefficient of di-*sec*-butyl-p-phenylenediamine may be substantially increased in subsurface materials with representative cation exchange capacities (CEC ~ 0.01 mol/kg) and groundwater ionic strengths ($[Na^+]$ ~ 1 mM). Since transport model calculations already suggested that di-*sec*-butyl-p-phenylenediamine poses little risk to CSWs, ion exchange calculations did not change the fundamental conclusions of the assessment. However, this illustrates the fact that assessments of other (future) fuel constituents may need to account for ion exchange and other sorptive mechanisms.

2.4. Implications and needs for future work

A transport screening model was developed for the forecasting of widespread contamination of at-risk CSWs by gasoline constituents. The model successfully captured the observed CSW contamination levels of a persistent contaminant (MTBE) in reformulated fuel use areas in the U.S., using only hydrogeologic data and fundamental hydrologic and chemical principles, and without any fitting procedures. Observed variability of hydrogeologic characteristics and gasoline release volumes across CSW sites implies that well contaminant concentrations predicted here are order-of-magnitude estimates. Additionally, prediction results do not accurately indicate the probable severity of unusually damaging cases that may result from large or multiple gasoline releases, different well pumping rates, or specific hydrogeology. Nevertheless, screening model forecasts are accurate and precise enough to be useful for policy making guidance. The model indicated typical contamination levels that will be found in CSWs located down gradient of UFT related releases, in the absence of degradation processes. Therefore, the prediction of a reasonably threatening result reflects the potential for extensive damage to water supplies on a costly scale. As shown by the MTBE case example, a calculation which implies high CSW contamination levels in only a few years should therefore be taken very seriously. Using the screening model, I suggest that any newly proposed gasoline additive or change in gasoline composition (e.g., alkylate enhancement) could be reliably screened for its potential to contaminate CSWs, thereby informing priority setting activities and indicating the need for further tests (e.g., for degradability or toxicity).

Gasoline release volume, CSW pumping rate, ambient groundwater velocity, and LUFT-CSW separation distance were the most influential field parameters responsible for C_{well} variability. Further study of these field parameter distributions and/or correlation with other parameters may improve the precision of this approach, and this would also produce insight into additional processes or effects which may need to be considered. Incorporation of uncertainty related to other field parameters had a secondary effect on the estimated variability of C_{well} predictions, demonstrating that these factors are less relevant to gaining an improved understanding of the problem of widespread CSW contamination. Estimated variability in predicted plume arrival times at CSWs was large, but it could effectively distinguish between gasoline constituents which may arrive on a <10 year versus 10+ year time frame. Consequently, those gasoline constituents which are predicted to contaminate at-risk CSWs at problematic levels should have confirmed biodegradabilities on a correspondingly shorter time frame over a wide range of subsurface conditions.

Biodegradation processes were not explicitly considered in this screening model, since these processes are much more difficult to generalize reliably *a priori* for individual gasoline constituents. However, model predictions indicated that some water-soluble, abundant gasoline components would normally cause high level contamination except that they may be quickly degradable under usual aquifer conditions (e.g., ethanol, methanol, toluene). Such compounds should be considered carefully, as they may generate toxic daughter products or consume sufficient subsurface oxygen to affect the fate of other gasoline components. For example, the highly biodegradable additive, ethanol, has been proposed as an alternative to MTBE (*93*). However, the widespread use of ethanol in fuels could create oxygen-starved subsurface zones in which anaerobically persistent compounds, such as benzene, might remain nondegraded and begin to cause significant CSW contamination.

2.5. Acknowledgments

This research was supported by the EPA/ONR/NCERQA Science to Achieve Results award R-82902301-0, the Alliance for Global Sustainability, the Martin Family Society of Fellows for Sustainability, the Ralph M. Parsons Fellowship Foundation, and the Schoettler Scholarship Fund. The authors would like to acknowledge Dr. Charles Harvey, Dr. Eric Adams, and Xin Fu for their assistance in development of the transport modeling approach and assistance with manuscript revisions, Drs. Katsuya Kawamoto and Dennis McLaughlin for helpful ideas on the model parameterization, Drs. Torsten Schmidt and Stephan Haderlein for measurement of gasoline-water partition coefficients and comments on early versions of the manuscript, Drs. Curt Stanley, Bruce Bauman, and Gerald Raabe at the American Petroleum Institute (API) for their critical insights regarding policy and industry needs relevant to this topic, John Brophy and Jim Caldwell at the Environmental Protection Agency (EPA) for their assistance with legislative interpretation and policy perspective, and Fred Hayslett, Carol Aamold, Bill Biza, Craig Crocker, Carl Goldapp, Dianna Grounds, Don Marymee, and Dr. Pete McMahon for supplying water supply and hydrogeologic data in several U.S. communities.

2.6. References

(1) Squillace, P. J.; Zogorski, J. S.; Wilber, W. G.; Price, C. V., *Preliminary assessment of the occurrence and possible sources of MTBE in groundwater in the United States, 1993-1994.* Environmental Science & Technology **1996**, *30*, 1721-1730.

(2) Johnson, R.; Pankow, J. F.; Bender, D.; Price, C.; Zogorski, J. S., *MTBE, To what extent will past releases contaminate community water supply wells?* Environmental Science & Technology **2000**, *34*, 2A-9A.

(3) Squillace, P. J.; Pankow, J. F.; Korte, N. E.; Zogorski, J. S., *Review of the environmental behavior and fate of methyl tert-butyl ether.* Environmental Toxicology and Chemistry **1997**, *16*, 1836-1844.

(4) Davis, J. M.; Brophy, J.; Hitzig, R.; Kremer, F.; Osinski, M.; Prah, J. D. "Oxygenates in Water: Critical Information and Research Needs," Office of Research and Development, U.S. Environmental Protection Agency, 1998, EPA-600/R-98/048.

(5) Garrett, P.; Moreau, M.; Lowry, J. D.; *MTBE as a ground water contaminant.* NWWA/API Conference on Petroleum Hydrocarbons and Organic Chemicals in Ground Water - Prevention, Detection and Restoration; Water Well Journal Publishing Co.: Houston, TX, 1986.

(6) Page, N. P.; *Gasoline leaking from underground storage tanks: Impact on drinking water quality.* Trace Substances in Environmental Health: Proceedings of the University of Missouri's 22nd Annual Conference on Trace Substances in Environmental Health; University of Columbia Press: Columbia, MO, 1989.

(7) Greenbaum, D.; Beuhler, M.; Campbell, R.; Ellis, P.; Greer, L.; Grumet, J.; Happel, A.; Henry, C.; Kenny, M.; Sawyer, R.; Sneller, T.; Starnes, D.; White, R. "Achieving Clean Air and Clean Water: The Report of the Blue Ribbon Panel on Oxygenates in Gasoline," U.S. Environmental Protection Agency, 1999, EPA-420-R-99-021.

(8) Grady, S. J. "A national survey of methyl *tert*-butyl ether and other volatile organic compounds in drinking water sources: results of the random survey," U.S. Geological Survey, 2002, Water-Resources Investigations Report 02-4079.

(9) Grady, S. J.; Casey, G. D. "Occurrence and distribution of methyl *tert*-butyl ether and other volatile organic compounds in drinking water in the Northeast and mid-Atlantic regions of the United States.," U.S. Geological Survey, 2001, Water-Resources Investigations Report 00-4228.

(10) Brown, A.; Farrow, J. R. C.; Rodriguez, R. A.; Johnson, B. J.; Bellomo, A. J.; *Methyl tertiary butyl ether (MtBE) contamination of the city of Santa Monica drinking water supply.* Proceedings of the 1997 Petroleum Hydrocarbons & Organic Chemicals in Ground Water Prevention, Detection, and Remediation Conference; Ground Water Publishing Co.: Houston, TX, 1997.

(11) Pankow, J. F.; Thomson, N. R.; Johnson, R. L.; Baehr, A. L.; Zogorski, J. S., *The urban atmosphere as a non-point source for the transport of MTBE and other volatile organic compounds to shallow groundwater.* Environmental Science & Technology **1997**, *31*, 2821-2828.

(12) Baehr, A. L.; Stackelberg, P. E.; Baker, R. J., *Evaluation of the atmosphere as a source of volatile organic compounds in shallow groundwater.* Water Resources Research **1999**, *35*, 127-136.

(13) Baehr, A. L.; Charles, E. G.; Baker, R. J., *Methyl tert-butyl ether degradation in the unsaturated zone and the relation between MTBE in the atmosphere and shallow groundwater.* Water Resources Research **2001**, *37*, 223-233.

(14) MacFarlane, S.; Mackay, D., *A fugacity-based screening model to assess contamination and remediation of the subsurface containing non-aqueous phase liquids.* Journal of Soil Contamination **1998**, *17*, 17-46.

(15) Schwarzenbach, R. P.; Gschwend, P. M.; Imboden, D. M. *Environmental Organic Chemistry*; 2 ed.; John Wiley & Sons: New York, NY, **2003**.

(16) Gardner, S.; Moore, B. "Case Studies in Wellhead Protection Area Delineation and Monitoring," U.S. Environmental Protection Agency, 1993, EPA-600/R-93/107.

(17) Belk, T.; Smith, J. J.; Trax, J. "Wellhead Protection, a Guide for Small Communities," U.S. Environmental Protection Agency, 1993, EPA-625/R-93/002.

(18) Hoffer, R. "Guidelines for Delineation of Wellhead Protection Areas," U.S. Environmental Protection Agency, 1987, EPA-4405/93/001.

(19) EPA; U.S. Environmental Protection Agency: Washington, DC, 1987.

(20) Owen, K. *Gasoline and Diesel Fuel Additives*; John Wiley & Sons, **1989**.

(21) Kawamoto, K.; Arey, J. S.; Gschwend, P. M., *Emission and fate assessment of methyl tertiary butyl ether in the Boston area airshed using a simple multimedia box model: comparison with urban air measurements.* Journal of the Air & Waste Management Association **2003**, *53*, 1426-1435.

(22) Leo, A. J.; Hansch, C., *Role of hydrophobic effects in mechanistic QSAR.* Perspectives in Drug Discovery and Design **1999**, *17*, 1-25.

(23) Leo, A. J.; Hoekman, D., *Calculating logP(oct) with no missing fragments; the problem of estimating new interaction parameters.* Perspectives in Drug Discovery and Design **2000**, *18*, 19-38.

(24) Hansch, C.; Leo, A. J. *Substituent constants for correlation analysis in chemistry and biology*; John Wiley & Sons: New York, NY, **1979**.

(25) Schubert, A. J.; Johansen, N. G., *Cooperative study to evaluate a standard test method for the speciation of gasolines by capillary gas chromatography.* Society of Automotive Engineers **1993**, *930144.*

(26) Schmidt, T. C.; Kleinert, P.; Stengel, C.; Goss, K. U.; Haderlein, S. B., *Polar fuel constituents - compound identification and equilibrium partitioning between non-aqueous phase liquids and water.* Environmental Science & Technology **2002**.

(27) Quimby, B. D.; Giarrocco, V.; Sullivan, J. J.; McCleary, K. A., *Fast analysis of oxygen and sulfur compounds in gasoline.* Journal of High Resolution Chromatography **1992**, *15*, 705-709.

(28) Howard, P. H.; Meylan, W. M. *Handbook of Physical Properties of Organic Chemicals*; CRC Press: Boca Raton, **1997**.

(29) Hilal, S. H.; El-Shabrawy, Y.; Carreira, L. A.; Karickhoff, S. W.; Toubar, S. S.; Rizk, M., *Estimation of the ionization of pKa of pharmaceutical substances using the computer program SPARC.* Talanta **1996**, *43*, 607.

(30) Hilal, S.; Karickhoff, S. W.; Carreira, L. A., *A rigorous test for SPARC's chemical reactivity models:*
estimation of more than 4300 ionization pKa's. Quantitative Structure Activity Relationships **1995**, *14*, 348.

(31) Cline, P. V.; Delfino, J. J.; Rao, P. S. C., *Partitioning of aromatic constituents into water from gasoline and other complex solvent mixtures*. Environmental Science & Technology **1991**, *25*, 914-920.

(32) Gmehling, J.; Lohmann, J.; Jakob, A.; Li, J. D.; Joh, R., *A modified UNIFAC (Dortmund) model. 3. Revision and extension*. Industrial & Engineering Chemistry Research **1998**, *37*, 4876-4882.

(33) Heermann, S. E.; Powers, S. E., *Modeling the partitioning of BTEX in water-reformulated gasoline systems containing ethanol*. Journal of Contaminant Hydrology **1998**, *34*, 315-341.

(34) Myrdal, P. B.; Manka, A. M.; Yalkowsky, S. H., *AQUAFAC 3: aqueous functional group activity coefficients; application to the estimation of aqueous solubility*. Chemosphere **1995**, *30*, 1619-1637.

(35) Rathburn, R. E. "Transport, Behavior, and Fate of Volatile Organic Carbon Compounds in Streams," U.S. Geological Survey, 1998, Professional Paper 1589.

(36) Huttunen, H.; Wyness, L. E.; Kalliokoski, P., *Identification of environmental hazards of gasoline oxygenate tert-amyl methyl ether (TAME)*. Chemosphere **1997**, *35*, 1199-1214.

(37) Gentle, J. E. *Random number generation and Monte Carlo methods*; 2nd ed.; Springer: New York, **2003**.

(38) EPA "Water on Tap: A Consumer's Guide to the Nation's Drinking Water," U.S. Environmental Protection Agency, Office of Water, 1997, EPA-815-K-97-002.

(39) Reuter, J. E.; Allen, B. C.; Richards, R. C.; Pankow, J. F.; Goldman, C. R.; Scholl, R. L.; Seyfried, J. S., *Concentrations, sources, and fate of the gasoline oxygenate methyl tert-butyl ether (MTBE) in a multiple use lake*. Environmental Science & Technology **1998**, *32*, 3666-3672.

(40) Weaver, J. W.; Haas, J. E.; Wilson, J. T.; *Analysis of the Gasoline Spill at East Patechoque, New York*. Conference on Non-aqueous Phase Liquids in the Subsurface Environment: Assessment and Remediation; Proceedings of the American Society of Civil Engineers: Washington, DC, 1996.

(41) Landmeyer, J. E.; Chapelle, F. H.; Bradley, P. M.; Pankow, J. F.; Church, C. D.; Tratnyek, P. G., *Fate of MTBE relative to benzene in a gasoline-contaminated aquifer*. Groundwater Monitoring and Remediation **1998**, 93-102.

(42) Delleur, J. W., *Geological Occurence of Groundwater*, In *The Handbook of Groundwater Engineering*; CRC Press: Boca Raton, **1999**; pp 4-11.

(43) Campbell, M. D.; Lehr, J. H. *Water Well Technology, Field Principles of Exploration Drilling and Development of Ground Water and Other Selected Minerals*; McGraw Hill Book Company: New York, NY, **1973**.

(44) Squillace, P. J.; Moran, M. J.; Lapham, W. W.; Price, C. V.; Clawges, R. M.; Zogorski, J. S., *Volatile organic compounds in untreated ambient groundwater of the United States*. Environmental Science & Technology **1999**, *33*, 4176-4187.

(45) Leenheer, J. A.; Malcolm, R. L.; McKinley, P. W.; Eccles, L. A., *Occurrence of dissolved organic carbon in selected ground water samples in the United States*. Journal of Research of the U.S. Geological Survey **1974**, *2*, 361-369.

(46) Kanerva, R.; Baker, D.; Clark, G.; Wheeler, D.; Antonacci, D.; McClasin, J.; Goetsch, W.; Horton, J.; Schrodt, S.; Ed, D., *Chapter 4, Section 4. Continue to Utilize Innovative and Cost Effective Methods to Implement Statewide Groundwater Quality Monitoring*, In

Biennial Comprehensive Status and Self-Assessment Report, 1996-1997; IL Environmental Protection Agency, IL Groundwater Protection Program., **1997**; pp 26-38.

(47) Fryar, A. E.; Mullican, W. F.; Macko, S. A., *Groundwater recharge and chemical evolution in the southern High Plains of Texas, USA*. Hydrogeology Journal **2001**, *9*, 522-542.

(48) MD "Maryland's Source Water Assessment Program," MD Department of the Environment, 1999.

(49) MT "Source Water Protection Program Newsletter, Spring 2000. Montana Source Water Assessment Program Update: Source Water Assessment Program Implementation," MT Department of Environmental Quality, 2000.

(50) Lent, R. M.; Waldron, M. C.; Rader, J. C. "Public Water Supplies in Massachusetts and Rhode Island: Investigations of Processes Affecting Source-Water Quality," U.S. Geological Survey, 1997, FS-054-97.

(51) Spayd, S. E. "Draft Guidance for Wellhead Protection Area Delineations in New Jersey," NJ Department of Environmental Protection, 1998.

(52) Detay, M. *Water Wells, Implementation, Maintenance, and Restoration*; John Wiley & Sons: Chichester, **1997**.

(53) Gelhar, L. W.; Welty, C.; Rehfeldt, K. R., *A critical review of data on field-scale dispersion in aquifers*. Water Resources Research **1992**, *28*, 1955-1974.

(54) Gelhar, L. W., Ambient ground water velocities in sand and gravel aquifers, *pers. comm.* **2002**.

(55) LeBlanc, D. R.; Garabedian, S. P.; Hess, K. M.; Gelhar, L. W.; Quadri, R. D.; Stollenwerk, K. G.; Wood, W. W., *Large-scale natural gradient tracer test in sand and gravel, Cape Cod, Massachusetts 1. Experimental design and observed tracer movement*. Water Resources Research **1991**, *27*, 895-910.

(56) Rajaram, H.; Gelhar, L. W., *Three-dimensional spatial moments analysis of the Borden tracer test*. Water Resources Research **1991**, *27*, 1239-1251.

(57) Haitjema, H. M. *Analytic Element Modeling of Groundwater Flow*; Academic Press: San Diego, **1995**.

(58) Crocker, C., Centerville water supply data, *pers. comm.* **1999**.

(59) Barber, L. B., *Sorption of chlorobenzenes to Cape Cod aquifer sediments*. Environmental Science & Technology **1994**, *28*, 890-897.

(60) Grounds, D., Guymon water supply data, *pers. comm.* **1999**.

(61) McMahon, P. B., Southwest Kansas aquifer sediment organic carbon data, *pers. comm.* **1999**.

(62) Hayslett, F., Columbus water supply data, *pers. comm.* **1999**.

(63) MacIntyre, W. G.; Antworth, C. P.; Stauffer, T. B.; Young, R. G., *Heterogeneity of sorption and transport-related properties in a sand-gravel aquifer at Columbus, Mississippi*. Journal of Contaminant Hydrology **1998**, *31*, 257-274.

(64) Biza, B., Chillicothe water supply well data, *pers. comm.* **1999**.

(65) Springer, A. E.; Bair, E. S., *Natural-gradient transport of bromide, atrazine, and alachlor in an organic carbon-rich aquifer*. Journal of Environmental Quality **1998**, *27*, 1200-1208.

(66) Marymee, D., Brush water supply data, *pers. comm.* **1999**.

(67) McMahon, P. B.; Bohlke, J. K.; Bruce, B. W., *Denitrification in marine shales in northeastern Colorado*. Water Resources Research **1999**, *35*, 1629-1642.

(68) Goldapp, C., College Station water supply data, *pers. comm.* **1999**.

(69) Martino, D. P.; Grossman, E. L.; Ulrich, G. A.; Burger, K. C.; Schlichenmeyer, J. L.; Suflita, J. M.; Ammerman, J. W., *Microbial abundance and activity in a low-conductivity aquifer system in east-central Texas*. Microbial Ecology **1998**, *35*, 224-234.

(70) McKinley, J. P.; Stevens, T. O.; Frederickson, J. K.; Zachara, J. M.; Colwell, F. S.; Wagnon, K. B.; Smith, S. C.; Rawson, S. A.; Bjornstad, B. N., *Biogeochemistry of anaerobic lacustrine and paleosol sediments within an aerobic unconfined aquifer*. Geomicrobiology Journal **1997**, *14*, 23-39.

(71) Aamold, C., Idaho Falls water supply data, *pers. comm.* **1999**.

(72) Carmichael, L. M.; Smith, T. G.; Pardieck, D. L., *Site-specific sorption values for mixtures of volatile and semivolatile organic compounds in sandy soils*. Journal of Environmental Quality **1999**, *28*, 888-897.

(73) Maslia, M. L.; Sautner, J. B.; Aral, M. M. "Analysis of the water-distribution system serving the Dover Township area, New Jersey: field-data collection activities and water-distribution system modeling," Agency for Toxic Substances and Disease Registry, 2000.

(74) NRC "Standard Incident Report database," National Response Center, 2004.

(75) Mercer, J. W.; Cohen, R. M., *A review of immiscible fluids in the subsurface: properties, models, characterization and remediation*. Journal of Contaminant Hydrology **1990**, *6*, 107-163.

(76) Fuller, E. N.; Schettler, P. D.; Giddings, J. C., *A new method for prediction of binary gas-phase diffusion coefficient*. Industrial & Engineering Chemistry **1966**, *58*, 19-27.

(77) Seagren, E. A.; Rittman, B. E.; Valocchi, A. J., *A critical evaluation of the local-equilibrium assumption in modeling of NAPL-pool dissolution*. Journal of Contaminant Hydrology **1999**, *39*, 109-135.

(78) Weaver, J. W.; Charbeneau, R. J.; Tauxe, J. D.; Lien, B. K.; Provost, J. B. "The Hydrocarbon Spill Screening Model," Robert S Kerr Environmental Research Laboratory, Office of Research and Development, U.S. EPA, 1994, EPA-600/R-94/039a.

(79) Domenico, P. A.; Schwartz, F. W. *Physical and Chemical Hydrogeology*; 2nd ed.; Wiley: New York, **1998**.

(80) Stewart, J. *Calculus, Early transcendentals*; 3 ed.; Brooks/Cole: Pacific Grove, CA, **1995**.

(81) Freeze, R. A.; Cherry, J. A. *Groundwater*; Prentice-Hall, Inc: Englewood Cliffs, NJ, **1979**.

(82) Charbeneau, R. J. *Groundwater Hydraulics and Pollutant Transport*; Prentice Hall: Saddle River, NJ, **2000**.

(83) Bear, J.; Jacobs, M., *On the movement of water bodies injected into aquifers*. Journal of Hydrology **1965**, *3*, 37-57.

(84) Knox, R. C.; Sabatini, D. A.; Canter, L. W. *Subsurface Fate and Transport Processes*; Lewis Publishers: Boca Raton, FL, **1993**.

(85) Rifai, H. S.; Borden, R. C.; Wilson, J. T.; Ward, C. H., *Intrinsic bioremediation for subsurface restoration*, In *Intrinsic Bioremediation*; Downey, D., Ed.; Batelle Press: Columbus, OH, **1995**; pp 1-29.

(86) Yeh, C. K.; Novak, J. T., *Anaerobic biodegradation of gasoline oxygenates in soils*. Water Environment Research **1994**, *66*, 744-752.

(87) Horan, C. M.; Brown, E. J.; *Biodegradation and inhibitory effects of methyl-tertiary-butyl ether (MTBE) added to microbial consortia.* 10th Annual Conference on Hazardous Waste Research: Kansas State University, Manhattan, KS, 1995.

(88) Hubbard, C. E.; Barker, J. F.; O'Hannesin, S. F.; Vandegriendt, M.; Gillham, R. W. "Transport and Fate of Dissolved Methanol, Methyl-tertiary-butyl Ether, and Monoaromatic Hydrocarbons in a Shallow Sand Aquifer," American Petroleum Institute, 1994, 4601.

(89) Powers, S. E.; Rice, D.; Dooher, B.; Alvarez, P. J. J., *Will ethanol-blended gasoline affect groundwater quality?* Environmental Science & Technology **2001**, *35*, 24A-30A.

(90) Novak, J. T.; Goldsmith, C. D.; Benoit, R. E.; OBrien, J. H., *Biodegradation of methanol and tertiary butyl alcohol in subsurface systems.* Water Science and Technology **1985**, *17*, 71-85.

(91) Boethling, R. S.; Howard, P. H., *Factors for intermedia extrapolation in biodegradability assessment.* Chemosphere **1995**, *30*, 741-752.

(92) Ruiz-Aguilar, G. M. L.; O'Reilly, K.; Alvarez, P. J. J., *A comparison of benzene and toluene plume lengths for sites contaminated with regular vs. ethanol-amended gasoline.* Ground Water Monitoring & Remediation **2003**, *23*, 48-53.

(93) Hogue, C., *News of the Week - good-bye MTBE: Congress is asked to drop requirement for water-polluting gasoline in favor of ethanol.* Chemical & Engineering News **2000**, *78*, 6.

Chapter 3
Estimating partition coefficients for fuel-water systems: extending pure phase Linear Solvation Energy Relationships to mixtures

3.1. Introduction

The discovery of nationwide contamination of subsurface drinking water sources by methyl-*tert*-butyl ether (MTBE) has demonstrated that gasoline constituents can seriously threaten thousands of community water supplies in the U.S. (*1-4*). Subsequent work has scrutinized the danger posed by other less abundant gasoline components, such as phenol and aniline derivatives, that are also polar and relatively water-soluble (*5*). In the wake of these activities, the need for environmental transport modeling of existing and future fuel constituents has become increasingly apparent (*3*).

Environmental fate assessment of fuel components relies heavily on fuel-water partition coefficient ($K_{\mathrm{i,fw}}$) values (*4,6,7*), defined as:

$$K_{\mathrm{i,fw}} = \frac{\text{solute concentration in the fuel phase}}{\text{solute concentration in the aqueous phase}} \tag{3-1}$$

for solute *i* distributed between a fuel (*f*) phase and an aqueous (*w*) phase. More generally, organic contaminant transport modeling requires information on solvation energetics in a wide variety of environmental media. For example, nonionic organic solutes typically have substantially different solvation energies in diverse organic phases such as wood (lignin) (*8*), soil organic matter (*9*), sediment organic matter (*9*), humic acid (*10*), and gasoline (*5*). Conventional approaches to calculating solvation behavior (UNIQUAC (*11*), UNIFAC (*12-14*)) cannot treat mixtures having solute-solvent interactions between a broad range of heteroatomic moieties. Traditional empirical methods (Linear Solvation Energy Relationships (*15*), Linear Free Energy Relationships (*16*), fragment methods (*17-19*)) are usually selective to particular solvent systems and require substantial parameter fitting from existing data. No single approach can estimate solvation energetics of organic contaminants in a wide range of multicomponent environmental mixtures. Until more generalizable solvation theories have been validated, it is useful to draw on a suite of methods, depending on the application.

In the present work, I develop and test empirical methods which are tailored to the challenge of estimating solute partitioning in fuel-water mixtures. This problem does not easily fall into the scope of previous models or estimation methods. Typically, the fuel-water equilibria of hydrocarbons are estimated assuming ideal solution conditions (Raoult's law) in the fuel mixture (*20-26*). However Raoult's law is likely to be inaccurate for polar fuel constituents, since these compounds may experience very different solution conditions in their pure liquid (ideal) phase. Additionally, polar fuel constituents (e.g., phenols and anilines) may be especially sensitive to the presence of other polar additives like oxygenates. Consequently it is desirable to generate a solvation model for fuels which could be accurately applied to co-existing polar and nonpolar fuel constituents.

Estimation of fuel-water equilibria poses the additional challenge that fuel and oil compositions vary widely by type. Retail and industrial fuel formulations are adjusted regionally and seasonally, as well as in response to new regulatory requirements, engine advances, and market influences. Liquid fuels are usually composed of mostly hydrocarbons, but they typically include additives or processing byproducts which contain heteroatoms (i.e., O, N, or S); such compounds may therefore affect the solvation properties of the fuel mixture (*27*). Conventional (not oxygenated) automotive retail gasolines contain C_4-C_{12} alkanes (45% to 65% by mass), low molecular weight aromatic hydrocarbons (20% to 40% by mass), and low molecular weight olefins (5% to 15% by mass) (*28*). As of this writing, oxygenated gasolines in many regions are required to contain more than 10% MTBE or ethanol by volume, and these regulations are likely to change in the near future (*29*). Diesel (*30*) and aviation (*31*) fuels generally include higher molecular weight components than gasoline and tend to be enriched in aliphatic compounds relative to gasoline. Motor oils have yet higher average molecular weights (250 to 1000 Daltons), and they contain significant quantities of both aromatic and aliphatic components as well as numerous additives (*24*). Finally, many other liquids of concern such as coal tar, a waste product of coal gasification, contain a highly variable and unrefined mixture of large molecular weight hydrocarbon compounds, often having substantial levels of oxygen-containing (up to 33% by mass) and sulfur-containing (up to 4% by mass) impurities (*22,32*). These examples of organic liquid mixtures have varied compositions, but they all are predominantly made up of hydrocarbons and sometimes have significant quantities of polar constituents.

3.2. Partitioning model theory for mixtures

Linear Solvation Energy Relationships (LSERs) allow accurate estimation of partition coefficients for a wide range of organic solutes in various solvents and organic mixtures ($\sigma_{\log K,\,\mathrm{LSER}} = 0.10$ to 0.25 (*33-38*)). The general LSER treatment relates a solute's partition coefficient to five independent solvation parameters of that solute and five coefficients specific to a given two-phase system, plus an intercept (*c*):

$$\log K_{\mathrm{i,pq}} = c + rR_2 + s\pi_2^{\mathrm{H}} + a\alpha_2^{\mathrm{H}} + b\beta_2^{\mathrm{H}} + mV_{\mathrm{x}} \qquad (3\text{-}2)$$

where $K_{\mathrm{i,\,pq}}$ is the partition coefficient of solute *i* between liquid or gas phases, *p* and *q*. The parameters R_2, π_2^{H}, α_2^{H}, β_2^{H} and V_{x} describe the excess molar refraction (*15*), polarity/ polarizability (*39*), hydrogen-bonding acidity (*40*), hydrogen-bonding basicity (*41,42*), and group-contributable molecular volume (*43*), respectively, of solute *i*, and *c*, *r*, *s*, *a*, *b*, and *m* are adjusted coefficients specific to the two-phase system, *p-q*. A key limitation to using LSERs for assessing fuels has been the need for copious partitioning data to calibrate the LSER coefficients for individual fuel-environmental media (e.g., water) combinations.

However, the linear solvent strength approximation (often referred to as the log-linear cosolvency model (*44*)) may be used to develop a mixing rule for use of LSERs deduced for 1:1 immiscible liquids, thereby allowing straightforward construction of estimated LSERs for novel mixtures. According to linear solvent strength theory (LSST), the solubility of solute *i* in a mixture phase is given by (*45*):

$$\log S_{\mathrm{i,p}} = \left(1 - \sum_{\mathrm{j}} \phi_{\mathrm{j}}^{\mathrm{p}}\right) \log S_{\mathrm{i,w}} + \sum_{\mathrm{j}} \phi_{\mathrm{j}}^{\mathrm{p}} \log S_{\mathrm{i,j}} \tag{3-3}$$

where $S_{\mathrm{i,p}}$ is the solubility of solute *i* in mixture phase *p*; $S_{\mathrm{i,w}}$ is the solubility of solute *i* in pure water; $S_{\mathrm{i,j}}$ is the solubility of solute *i* in pure cosolvent *j* in the mixture; and $\phi_{\mathrm{j}}^{\mathrm{p}}$ is the volume fraction of each cosolvent *j* in mixture *p*. Equation 3-3 is formulated with water as a reference solubility because LSST has conventionally been used to model organic solutes in aqueous/organic cosolvent mixtures, including applications in drug development (*45,46*), reverse phase liquid chromatography (*47,48*), and environmental fate modeling (*49,50*). But one may write this equation substituting any reference phase, including a gas phase, in place of water (i.e., $S_{\mathrm{i,g}}$ for $S_{\mathrm{i,w}}$). LSST is most applicable to cases in which the mixed solvent is polar (*46*), making it a simple and powerful approach for extrapolating the solubilities of a wide range of solutes in polar cosolvent-aqueous mixtures (*48,49,51*). Li (*52*) applied LSST to common aqueous/cosolvent binary mixtures for over 1000 solutes and found $\sigma_{\log S,\,\mathrm{LSST}} \sim 0.1$ to 0.4, depending on the cosolvent. In addition to contrived (laboratory) ternary mixtures, relevant environmental systems such as fuel-water/cosolvent and natural sorbent-water/cosolvent systems have been characterized effectively using LSST. Several workers have used LSST to model the partitioning of polycyclic aromatic hydrocarbon (PAH) compounds in fuel-water or similar systems containing methanol, ethanol, isopropanol, acetonitrile, or MTBE cosolvents (*23,25,44,53,54*). Fu and Luthy demonstrated that LSST accurately described soil-water/methanol partitioning equilibria of naphthalene, naphthol, quinoline, and 3,5-dichloroaniline (*50*). Spurlock and Biggar verified LSST application to soil-water partitioning of several phenylurea herbicides in the presence of methanol or dimethylsulfoxide solvent (*55*). Lee and coworkers showed that LSST could explain pentachlorophenol equilibria between soils and methanol/water, acetonitrile/water, and dimethylsulfoxide/water mixtures; however LSST failed to fit benzoic acid sorption behaviors under similar conditions (*56,57*). Hayden and coworkers recently showed that LSST could accurately explain tetrachloroethylene partitioning between isopropanol/water or ethanol/water mixtures and an activated carbon surface (*58*). Hence, I surmised that LSST may provide the basis for a simple mixing rule for LSERs.

To extend LSERs to new 1:1 phase partitioning mixtures, I combined these models. From equation 3-3, partitioning of solute *i* between two phases *p* and *q* can be described generally as (using the subscript, *g*, to reflect any reference phase):

$$\begin{aligned}
\log K_{\mathrm{i,pq}} &\equiv \log S_{\mathrm{i,p}} - \log S_{\mathrm{i,q}} \\
&= \left(1 - \sum_{\mathrm{j}} \phi_{\mathrm{j}}^{\mathrm{p}}\right) \log S_{\mathrm{i,g}} + \sum_{\mathrm{j}} \phi_{\mathrm{j}}^{\mathrm{p}} \log S_{\mathrm{i,j}} - \left(1 - \sum_{\mathrm{k}} \phi_{\mathrm{k}}^{\mathrm{q}}\right) \log S_{\mathrm{i,g}} - \sum_{\mathrm{k}} \phi_{\mathrm{k}}^{\mathrm{q}} \log S_{\mathrm{i,k}} \\
&= \sum_{\mathrm{j}} \phi_{\mathrm{j}}^{\mathrm{p}} \log\left(\frac{S_{\mathrm{i,j}}}{S_{\mathrm{i,g}}}\right) - \sum_{\mathrm{k}} \phi_{\mathrm{k}}^{\mathrm{q}} \log\left(\frac{S_{\mathrm{i,k}}}{S_{\mathrm{i,g}}}\right)
\end{aligned}$$

and setting $(S_{\mathrm{i,j}})/(S_{\mathrm{i,g}}) = K_{\mathrm{i,jg}}$ and $(S_{\mathrm{i,k}})/(S_{\mathrm{i,g}}) = K_{\mathrm{i,kg}}$, one finds:

$$\log K_{i,pq} = \log\left(\frac{S_{i,p}}{S_{i,q}}\right) = \sum_{j} \phi_j^p \log K_{i,jg} - \sum_{k} \phi_k^q \log K_{i,kg} \tag{3-4}$$

where log $K_{i,,jg}$ and log $K_{i,kg}$ may be related to a reference phase, *g*; and *j* and *k* are the solvent components of mixtures, *p* and *q*, respectively. Employing the LSER formulation (equation 3-2), the solvent volume-fraction additivity of log *K* values for mixture components therefore implies that:

$$\begin{aligned}\log K_{i,pq} = &\sum_{j} \phi_j^p \left(c_{jg} + r_{jg}R_2 + s_{jg}\pi_2^H + a_{jg}\alpha_2^H + b_{jg}\beta_2^H + m_{jg}V_x\right) \\ &-\sum_{k} \phi_k^q \left(c_{kg} + r_{kg}R_2 + s_{kg}\pi_2^H + a_{kg}\alpha_2^H + b_{kg}\beta_2^H + m_{kg}V_x\right)\end{aligned} \tag{3-5}$$

This equation suggests that the set of system LSER coefficients ([c], [r], [s], ...) may be specified using established pure-phase/reference-phase LSER coefficients.

Wang et al. (*59*) applied equation 3-5 to a set of reverse phase liquid chromatographic systems. They investigated a broad range of solutes partitioning between a stationary saturated C_8 hydrocarbon phase and an aqueous binary phase having an eluent modifiers (cosolvent) of methanol, acetonitrile, or tetrahydrofuran. With respect to equation 3-6, this means that partitioning was between a stationary phase (*p*) and mixtures (*q*) of water (*w*) and single organic cosolvents (*z*). Over a range of mixture compositions (ϕ_z^q), the partition coefficient could be described using:

$$\begin{aligned}\log K_{i,pq} = &\left(c_{pw} + r_{pw}R_2 + s_{pw}\pi_2^H + a_{pw}\alpha_2^H + b_{pw}\beta_2^H + m_{pw}V_x\right) \\ &-\phi_z^q \left(c_{zw} + r_{zw}R_2 + s_{zw}\pi_2^H + a_{zw}\alpha_2^H + b_{zw}\beta_2^H + m_{zw}V_x\right)\end{aligned} \tag{3-6}$$

where $\phi_p^p \equiv 1$ since the stationary phase is taken to be pure and where water was chosen as a reference phase so that $\phi_w^q \log K_{i,ww} = 0$. These workers fitted so-called "global" LSER coefficients to all three systems with modifier concentrations of up to 50 percent by volume. In other words, the set of twelve LSER coefficients of equation 3-6 were fitted to solute partitioning data which reflected a range of ϕ_z values ($\phi_z = 0.10$ to $\phi_z = 0.50$) for each binary (water plus cosolvent) system. This approach gave calculated solute partitioning free energy values as accurate as results found for LSERs that have been derived for systems of fixed composition ($\sigma_{\log K, \text{Wang}} \sim 0.10$), showing that the LSST approximation is very useful for the conditions that Wang et al. considered.

A contrasting approach, based on a "solvent compartments" mixing rule suggested by Schmidt et al. (*5*), was also developed to model the fuel phase. Schmidt et al. proposed that partition coefficients of a solute (*i*) partitioning between a fuel mixture (*p*) and relatively pure water phase (*w*) could be modeled as:

$$K_{i,pw} = \sum_{j} \phi_j^p K_{i,jw} \tag{3-7}$$

where multiple solvent components (*j*) constitute the fuel phase. This approach is mathematically equivalent to a hypothetical system in which the solute partitions between pure water and a composite of pure solvent compartments that constitute the total fuel mixture (*p*), where ϕ_j^p represents the volume of each fuel component (*j*) compartment as a fraction of the total fuel volume. This approach was very useful for the set of systems that Schmidt et al. considered; however it is not clear how equation 3-7 should extend to systems in which the aqueous (reference) phase also contains abundant cosolvent(s). By switching to a gas reference phase, we could treat the aqueous mixture (*q*) using LSST (similar to equation 3-4) and apply the compartment solvent model to the fuel phase (*p*) only:

$$\log K_{i,pq} = \log\left(\sum_j \phi_j^p K_{i,jg}\right) - \sum_k \phi_k^q \log K_{i,kg} \qquad (3\text{-}8)$$

Employing LSERs, as before, to calculate the pure phase partition coefficients, this "combined solvent compartment / LSST" (CSCLSST) mixing rule could therefore be expressed as:

$$\begin{aligned}\log K_{i,pq} &= \log\left(\sum_j \phi_j^p 10^{\left(c_{jg}+r_{jg}R_2+s_{jg}\pi_2^H+a_{jg}\alpha_2^H+b_{jg}\beta_2^H+m_{jg}V_x\right)}\right) \\ &\quad -\sum_k \phi_k^q\left(c_{kg}+r_{kg}R_2+s_{kg}\pi_2^H+a_{kg}\alpha_2^H+b_{kg}\beta_2^H+m_{kg}V_x\right)\end{aligned} \qquad (3\text{-}9)$$

In cases where there is not sufficient data to fit the LSER coefficients of equations 3-5 or 3-9, one may simply apply known pure-component LSER coefficients, and thereby estimate the partitioning properties of solutes in mixtures. I refer to mixture K_{fw} values estimated in this way as "LSST-LSER" estimates (equation 3-5) or "CSCLSST-LSER" estimates (equation 3-9). In this paper, I evaluated the efficacy of the solvent compartment model and LSST for extending pure phase LSER coefficients to new mixture systems. Fuel mixtures with both nonpolar and polar constituents were used as a set of test cases. It is worth noting that previous investigators have suggested that LSST applies best to systems in which the solute is less polar than the solution mixture (*46,52*). However most systems under consideration here contain a nonpolar (fuel) phase in which polar solutes are dissolved into moderately polar or nonpolar mixtures. Therefore a second objective was to explore whether this potential limitation undermined the usefulness of LSST-LSER fuel-water equilibria predictions, in comparison to other commonly used approaches for modeling the environmental fate of organic pollutants in fuels. If successful, this method would allow engineers and regulators to estimate the fuel-water partition coefficients of novel fuel constituents based on existing pure phase LSERs. As a result, LSST-LSER or CSCLSST-LSER estimates may enable the effective screening of the environmental transport behavior of proposed fuel additives, including the potential to threaten water (*4*) and urban air (*60*). Additionally, this investigation illuminates a general method whereby the predictive power of previously resolved LSERs for solvent systems might be extended to many multicomponent mixtures of environmental relevance.

3.3. Methods

Literature compositions were collected for several fuels and fuel-like mixtures (Tables 3-1 and 3-2) and related ternary two-phase organic-water systems (Table 3-3). Individual phase mixture compositions were converted to volume fractions, assuming $\Delta V_{mixing} = 0$. Only mixture components which contributed greater than 0.1 vol.% to either phase were included in the subsequent solvation modeling. In other words, solutes of less than 0.1 vol.% concentration in a given mixture were considered too dilute to influence the overall solvation properties of that phase. Where specific characterizations of fuels were not given, I assumed average gasoline or diesel compositions found in surveys (*28,30*).

Table 3-1. Compositions of synthetic and retail fuel mixtures

index	mixture type	composition (by volume percent)
1.	synthetic gasoline (*61*)	24% hexane, 32% 2,2,4-trimethylpentane, 3% benzene, 7% toluene, 24% xylenes
2.	synthetic gasoline (*62*)	83.1% 2,2,4-trimethylpentane, 0.8% benzene, 5.8% toluene, 2.6% ethylbenzene, 7.7% xylenes, + variable ethanol amendment
3.	retail gasolines (*5*)	52% aliphatic hydrocarbons, 34% aromatic hydrocarbons, 5.3% olefins, 6.5% methyl-*tert*-butyl ether
4.	isooctane-MTBE mix (*5*)	95% isooctane, 5% methyl-*tert*-butyl ether
5.	isooctane-MTBE mix (*5*)	85% isooctane, 15% methyl-*tert*-butyl ether
6.	isooctane-MTBE mix (*5*)	70% isooctane, 30% methyl-*tert*-butyl ether
7.	toluene-MTBE mix (*5*)	95% toluene, 5% methyl-*tert*-butyl ether
8.	toluene-MTBE mix (*5*)	85% toluene, 15% methyl-*tert*-butyl ether
9.	toluene-MTBE mix (*5*)	70% toluene, 30% methyl-*tert*-butyl ether
10.	diesel fuel survey (*30*)	83% aliphatic hydrocarbons, 15.3% aromatic hydrocarbons, 1.4% olefins

Table 3-2. Estimated compositions of retail gasoline mixtures based on survey averages (reported as mass percent) (*28,63*)

gasoline	conventional		oxygenated	
component	mass%	calc. vol.%	mass%	calc. vol.%
butane	8.3	9.4	7.5	8.5
pentane	7.5	8.3	6.7	7.4
hexane	5.8	6.3	5.2	5.6
heptane	2.2	2.4	2.0	2.1
octane	2.0	2.1	1.8	1.9
2-methylpentane	5.8	6.3	5.2	5.6
2,3-dimethylbutane	4.0	4.3	3.6	3.9
2,2,4-trimethylpentane	10.6	11.2	9.5	10.1
methylcyclopentane	2.1	2.1	1.9	1.9
2,-methyl-2-butene	1.8	1.9	1.6	1.7
1-hexene	3.4	3.6	3.1	3.3
benzene	4.3	3.9	3.9	3.5
toluene	16.2	14.7	14.6	13.3
xylenes	6.2	5.7	5.6	5.2
ethylbenzene	7.3	6.7	6.6	6.1
1,2,3-trimethylbenzene	9.2	8.5	8.3	7.7
naphthalene	3.3	2.8	3.0	2.5
methyl-*tert*-butyl ether	0.0	0.0	10.0	9.7

Table 3-3. Ternary mixture composition ranges (reported as mass percent) (*61*)

	water (A) benzene (B) isobutanol (C)		water (A) toluene (B) isobutanol (C)		water (A) benzene (B) pentanol (C)		water (A) benzene (B) hexanol (C)		water (A) benzene (B) MTBE (C)	
	aqueous	organic	aqueous	organic	aqueous	organic	aqueous	organic	aqueous	organic
A	91.8-99.8	0.2-19.0	91.5-99.9	0.1-17.6	97.9-99.8	0.1-11.0	99.5-99.8	0.1-7.3	95.8-100	0.1-1.4
B	0.0-0.2	0.0-99.9	0.0-0.1	0.0-99.9	0.0-0.2	0.0-99.9	0.0-0.2	0.0-99.9	0.0-0.2	3.9-99.9
C	0.0-8.2	0.0-81.0	0.0-8.5	0.0-82.4	0.0-2.1	0.0-89.0	0.0-0.5	0.0-92.7	0.0-4.2	0.0-94.7

In order to conduct LSST-LSER and CSCLSST-LSER calculations for fuel and fuel-like mixture systems of interest, pure liquid-gas LSERs were used to describe representative individual mixture components (i.e., the gas phase was chosen as the reference). Fitted LSER coefficients for several pure solvent-water systems were directly available in the literature (*35-37,42,64*); these coefficient values were added to water-air LSER coefficients (*35*) to estimate the corresponding solvent-air LSER coefficient values (Table 3-4). Since individual solvent-water or solvent-air LSERs were not found for all of the fuel mixture components considered here, many components were grouped into categories as follows. Normal and branched alkanes were grouped and treated using the alkane solvent LSER; additionally, olefins were considered treatable using the alkane-air LSER, as long olefin was not a dominant component (less than ~10 vol.%). Methylcyclopentane was treated using the cyclohexane LSER; and alkyl-substituted aromatic hydrocarbons and naphthalene were grouped together and modeled using the toluene LSER. Since a LSER was not available to describe MTBE-air systems, the diethylether-air LSER was used as a substitute. Using these assumptions, enough applicable solvent-air LSERs were drawn from the literature or estimated to represent all the major components for several relevant fuel formulations.

Table 3-4. Solvent-air LSER coefficients estimated from published solvent-water LSERs (*35-37,42,64*)

solvent system[†]	c	r	s	a	b	v
water-air	-0.99	0.58	2.55	3.81	4.84	-0.90
alkane-air	-0.71	1.23	0.89	0.30	0.02	3.39
cyclohexane-air	-0.87	1.39	0.82	0.04	-0.06	3.75
toluene-air	-0.98	1.17	1.77	0.90	0.27	3.64
benzene-air	-0.98	1.07	1.95	0.80	0.21	3.72
diethylether-air	-0.53	1.15	1.51	3.79	-0.67	3.45
hexanol-air	-0.95	1.05	1.40	3.90	0.78	3.35
pentanol-air	-0.91	1.10	1.26	4.02	0.93	3.31
isobutanol-air	-0.77	1.09	1.86	3.83	2.58	1.88
ethanol-air	-0.79	0.99	1.59	4.00	1.20	3.03
alkane-water	0.29	0.65	-1.66	-3.52	-4.82	4.28

[†] organic solvent-air LSER coefficients shown here were not regressed directly from data; they were estimated by adding the water-air LSER coefficients (shown) to corresponding organic solvent-water LSER coefficients found in the literature (not shown, except for the alkane-water coefficients).

Mixture volume fraction data were combined with estimated pure component solvent-gas LSER coefficients to formulate liquid-liquid mixture LSER coefficients using equations 3-5 and 3-9.

The liquid-liquid mixture LSER coefficients were then linearly combined with the solvatochromic parameters of 37 polar and nonpolar solutes (Table 3-5) to calculate 123 synthetic and retail fuel-water partition coefficients and 156 ternary system partition coefficients (Table 3-6).

Table 3-5. LSER solute parameters and hypothetical pure liquid vapor pressures (*16,35,37,65*)

solute	R_2	π_2^H	α_2^H	β_2^H	V_x	log P°_L [bar]
water	0.000	0.45	0.82	0.35	0.167	-1.50
aniline	0.955	0.96	0.26	0.41	0.816	-3.08
p-toluidine	0.923	0.95	0.23	0.45	0.957	-1.76
o-toluidine	0.966	0.92	0.23	0.45	0.957	-3.45
2,6-dimethylaniline	0.972	0.89	0.20	0.46	1.098	-3.70
phenol	0.805	0.89	0.60	0.30	0.775	-3.14
p-cresol	0.820	0.87	0.57	0.31	0.916	-3.59
o-cresol	0.840	0.86	0.52	0.30	0.916	-3.20
3,4-dimethylphenol	0.830	0.86	0.56	0.39	1.057	-4.06
2,6-dimethylphenol	0.860	0.79	0.39	0.39	1.057	-3.26
3,4,5-trimethylphenol	0.830	0.88	0.55	0.44	1.198	-4.02[a]
2,4,6-trimethylphenol	0.860	0.79	0.37	0.44	1.198	-4.02[a]
methanol	0.278	0.44	0.43	0.47	0.308	-0.76
ethanol	0.246	0.42	0.37	0.48	0.449	-1.09
isopropanol	0.212	0.36	0.33	0.56	0.590	-1.21
tert-butanol	0.180	0.30	0.31	0.60	0.731	-1.25
isobutanol	0.219	0.39	0.37	0.48	0.872	n.a.
n-pentanol	0.219	0.42	0.37	0.48	0.872	n.a.
n-hexanol	0.210	0.42	0.37	0.48	1.013	n.a.
methyl-*tert*-butyl ether	0.024	0.19	0.00	0.45	0.872	-0.49
ethylacetate	0.106	0.62	0.00	0.45	0.747	-0.90
thiophene	0.687	0.56	0.00	0.15	0.641	-0.96
benzo[b]thiophene	1.323	0.88	0.00	0.20	1.010	-2.79[b]
benzene	0.610	0.52	0.00	0.14	0.716	-0.90
toluene	0.601	0.52	0.00	0.14	0.857	-1.43
ethylbenzene	0.613	0.51	0.00	0.15	0.998	-1.91
n-propylbenzene	0.604	0.50	0.00	0.15	1.139	-2.35
m-xylene	0.623	0.52	0.00	0.16	0.998	-1.96
o-xylene	0.663	0.56	0.00	0.16	0.998	-2.05
p-xylene	0.613	0.52	0.00	0.16	0.998	-1.93
1,2,3-trimethylbenzene	0.728	0.61	0.00	0.19	1.139	-2.70
4-ethyltoluene	0.630	0.51	0.00	0.18	1.139	-2.40
naphthalene	1.340	0.92	0.00	0.20	1.085	-3.33
1-methylnaphthalene	1.344	0.90	0.00	0.20	1.226	-4.08
2-methylnaphthalene	1.304	0.88	0.00	0.20	1.226	-3.95[a]
acenaphthene	1.604	1.05	0.00	0.20	1.259	-4.67
fluorene	1.588	1.06	0.00	0.20	1.357	-4.75
phenanthrene	2.055	1.29	0.00	0.26	1.454	-5.36
anthracene	2.290	1.34	0.00	0.26	1.454	-5.35
fluoranthene	2.377	1.55	0.00	0.20	1.585	-7.15[a]

[n.a.] not applicable. [a] log P°_L value estimated using (*66*). [b] log P°_L value taken from (*67*).

There were two exceptions to this straightforward protocol. For Heerman and Powers's synthetic fuel-water/ethanol system (*62*), the aqueous phase was set to their reported ethanol/water composition. But since the system composition depended on the calculated $K_{\text{ethanol,fw}}$ values, ethanol concentrations in the synthetic fuel phase were iteratively varied until $\phi^{f}_{\text{ethanol}}$ values and $K_{\text{ethanol,fw}}$ values were self-consistent using equation 3-5 or equation 3-9. This generated an estimated ethanol partition coefficient which could be evaluated against measured values. In the case of the fuel/MTBE-water mixture data of Schmidt et al. (*5*), MTBE mixture concentrations were reported in terms of initial fuel levels rather than measured post-equilibration solution concentrations in either phase. For this set, an overall system mass balance of MTBE was therefore constrained while, simultaneously, MTBE concentrations in both phases were determined via iterative calculation of ϕ^{f}_{MTBE}, ϕ^{w}_{MTBE}, and $K_{\text{MTBE,fw}}$ until equation 3-5 (or equation 3-9) achieved self-consistency.

3.4. Results and discussion

Three categories of fuel-water systems were separately considered. First, I report results for which the fuel phase has multiple components, but the aqueous phase is relatively pure. Second, I consider systems in which the fuel phase is a nonpolar mixture and the aqueous phase includes substantial ethanol. Finally, I describe results for a few ternary systems in which polar and nonpolar solvent components are significantly abundant in both the aqueous and organic phase.

3.4.1. Partitioning of polar and nonpolar compounds between synthetic or retail fuels and water

In both simulated and realistic fuel-water systems, both LSST-LSER and CSCLSST-LSER predicted partitioning of 88 polar and nonpolar solutes were within a factor of 2 to 3 of measured K_{fw} values, despite simplifications (e.g., assuming alkene solvent behaved like alkane solvent) and without the aid of any fitting procedures (Figures 3-1A and 3-1B; Table 3-6). Reported or estimated experimental uncertainty in partition coefficient measurements was significantly lower, ranging from 0.05 to 0.12 in the log K_{fw} (*5,20,61,62*). Both models predicted partitioning of nonpolar solutes better than polar solutes, on average ($\sigma_{\log K} \sim 0.2$ for nonpolar solutes using either model). Using the LSST-LSER model, estimated partition coefficients of phenols and methyl-substituted phenols were the most inaccurate ($\sigma_{\log K} = 0.61$), especially in systems containing MTBE. Across all other solute families under consideration (i.e., not phenols), including nonpolar aromatic hydrocarbons, anilines, aliphatic alcohols, methyl-*tert*-butyl ether, ethylacetate, thiophenes, and water, the LSST-LSER approach gave predictions that were almost as good as predictions for the nonpolar solute set alone ($\sigma_{\log K} = 0.23$). Errors of the CSCLSST-LSER model did not appear to strongly relate to solute polarity or trends in solute-solvent hydrogen-bonding interactions; the largest errors were for phenol solutes in isooctane/MTBE-water systems ($\sigma_{\log K} = 0.48$).

Failure of the LSST approximation for polar solutes in the organic phase could largely explain LSST-LSER deviations. In this set of systems, major fuel components are sufficiently hydrophobic that the solvation properties of the aqueous phase were likely to be unaffected by the presence of fuel phase constituents. The most abundant aqueous phase organic component was MTBE in system 6, Table 3-1, having aqueous phase concentrations as high as 0.067 M

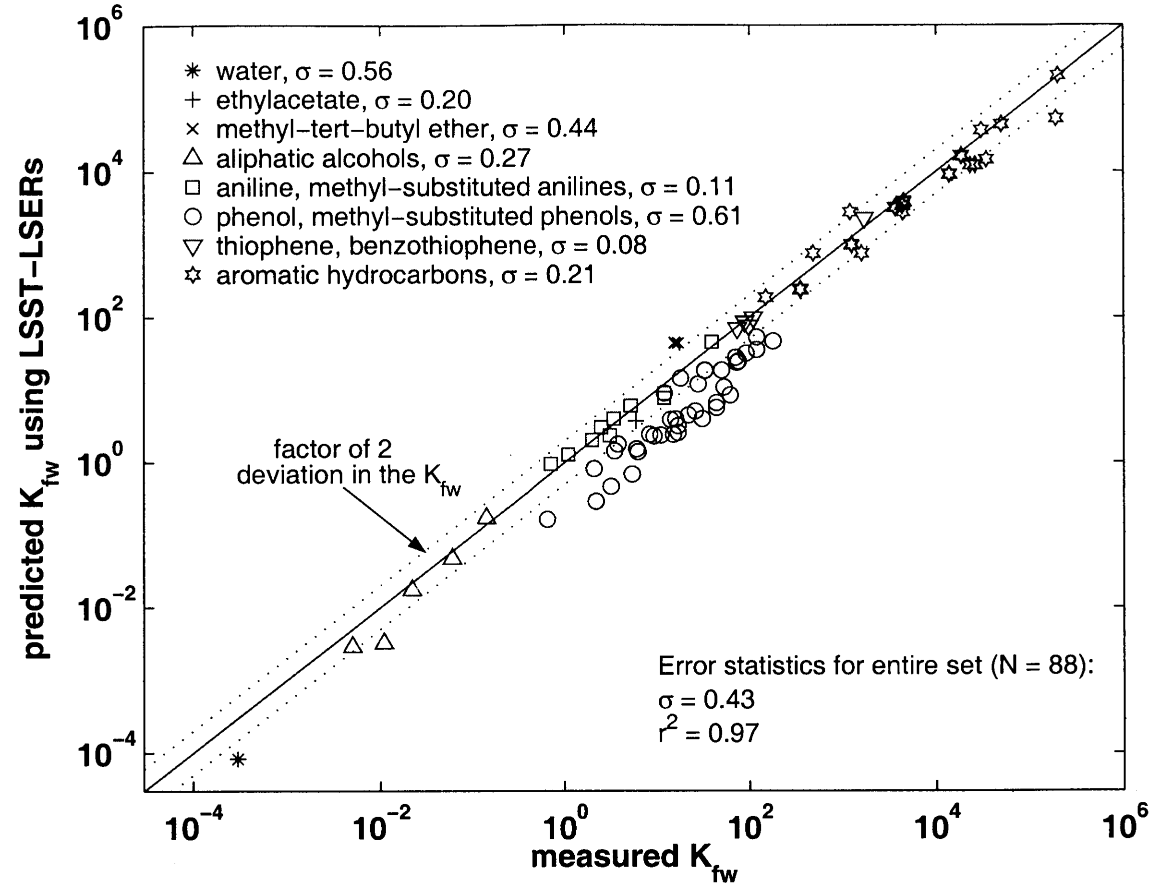

Figure 3-1A. Predicted K_{fw} values of polar and nonpolar compounds in simulated and retail fuel-water systems using the LSST-LSER model (equation 3-5)

(ϕ^{w}_{MTBE} = 0.008) in this case. Since LSERs can model pure water-air partitioning accurately ($\sigma_{\log K}$ = 0.15 (*35*)), I inferred that model error for these systems was primarily related to treatment of the organic phase. Additionally, consistent trends were detectable in model residuals. The LSST-LSER standard error for phenol log K_{fw} predictions was $\sigma_{\log K}$ = 0.45 in 5% MTBE mixtures, $\sigma_{\log K}$ = 0.63 in 15% MTBE mixtures, and $\sigma_{\log K}$ = 0.68 in 30% MTBE mixtures (N = 10 for all three cases); thus the bias of phenol K_{fw} predictions systematically increased with increasing MTBE content in the organic phase, other considerations held equal. Additionally, across predictions for all solutes, a significant correlation (r^2 = 0.60) was found between model residuals and solute α_2^H (hydrogen-bond donating parameter) values. Since LSST becomes less accurate in cases where solute polarity exceeds that of the solvent, I surmised that strong hydrogen-bond solute-solvent interactions (e.g., phenol-MTBE complexes) in the organic phase were primarily responsible for LSST-LSER deviations.

I separately modeled the organic phase as an ideal solution (Raoult's law) to assess whether LSST-LSER and CSCLSST-LSER estimates could meaningfully capture organic phase mixture solvation of both nonpolar and polar solutes. Raoult's law (the ideal solution assumption) is

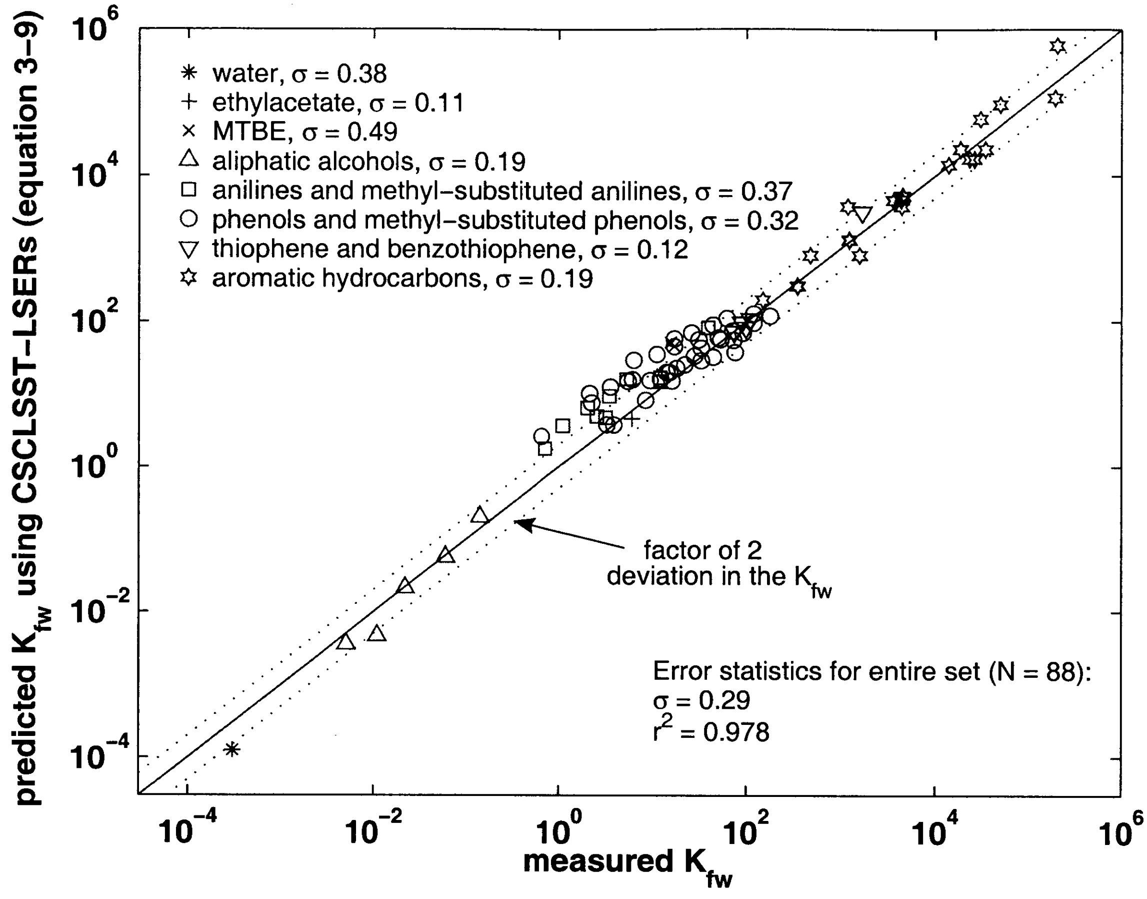

Figure 3-1B. Predicted K_{fw} values of polar and nonpolar compounds in simulated and retail fuel-water systems using the CSCLSST-LSER model (equation 3-9)

frequently used to model nonpolar solutes in fuels and related mixtures (*20-26*). LSST-LSER computed partition coefficients were therefore compared to those using Raoult's law for the nonpolar (fuel) phase. In this case, the fuel-gas partition coefficient may be expressed as (*16,68*):

$$K_{i,fg} = \frac{RT}{V_f P^{\circ}_{L,i}} \qquad (3\text{-}10)$$

where R is the molar gas constant, T is temperature, V_f is the molar volume of the fuel phase, and $P^{\circ}_{L,i}$ is the (hypothetical) liquid vapor pressure of solute i. Raoult's law would be inappropriate for modeling the aqueous phase, since nonpolar organic solutes are known to experience significant nonideality in aqueous conditions (*69*). Applying Raoult's law to the fuel phase, but continuing to use LSST to treat the aqueous phase, the fuel-gas subsystem of equation 3-5 could be substituted with equation 3-10, so that:

$$\log K_{i,fq} = \log\left[\frac{RT}{V_f P^{\circ}_{L,i}}\right] - \sum_k \phi^q_k \left(c_{kg} + r_{kg}R_2 + s_{kg}\pi^H_2 + a_{kg}\alpha^H_2 + b_{kg}\beta^H_2 + m_{kg}V_x\right) \qquad (3\text{-}11)$$

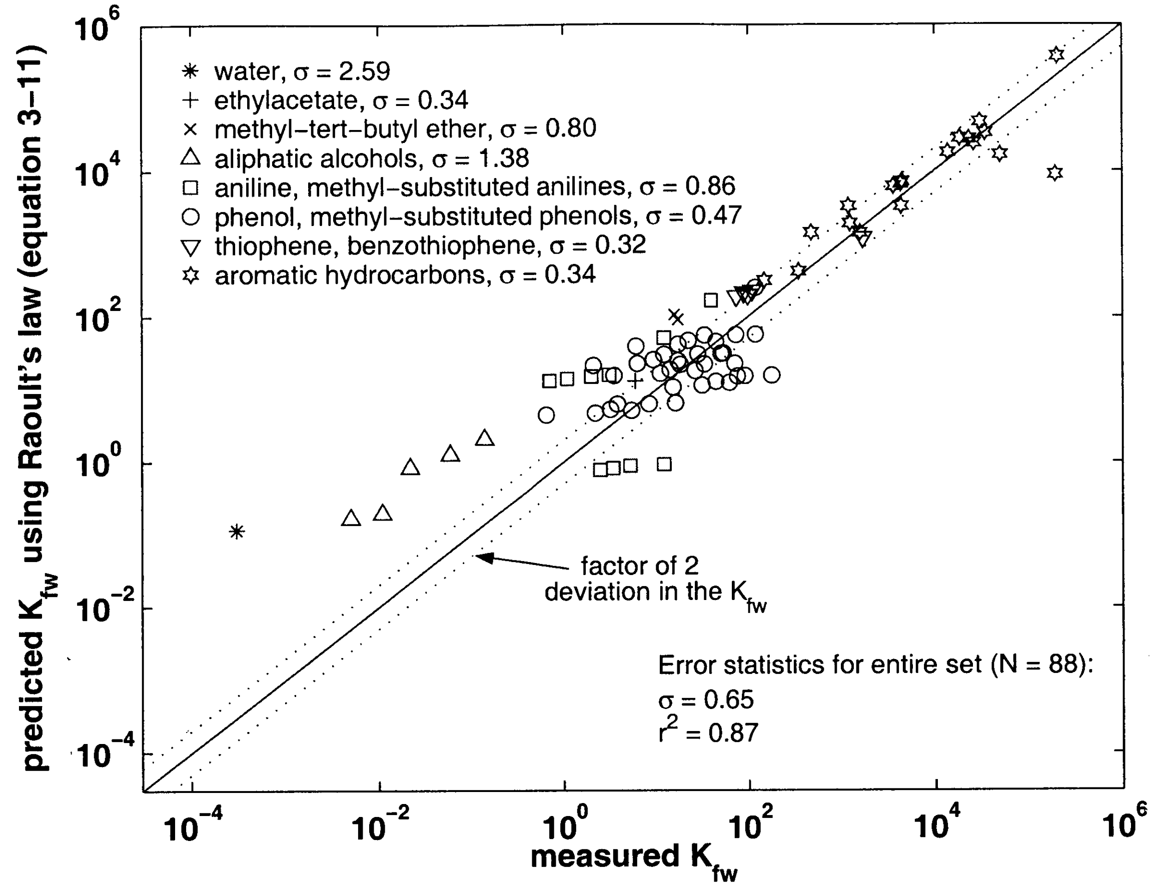

Figure 3-1C. Predicted K_{fw} values of polar and nonpolar compounds in simulated and retail fuel-water systems assuming Raoult's law (ideality) in the fuel phase

where the solvent components, *k*, constitute the mixture in the aqueous phase, *q*. Equation 3-11 accurately predicted fuel-water partitioning of a wide range of nonpolar hydrocarbon and thiophene compounds (Figure 3-1C), in agreement with previous findings (*20,21,24,25*). However, predictions were highly unreliable for polar solutes, since nonideal solvation typically occurs in cases where the solute and solution differ significantly in polarity and/or hydrogen-bonding capabilities (*68*). Hence these results were consistent with my expectation that Raoult's law is an inadequate model for describing the behavior of polar solutes in fuels.

Additional calculations were conducted to test whether the CSCLSST-LSER and LSST-LSER models could effectively distinguish between solvation in real fuel mixtures and solvation controlled only by London dispersion interactions with alkanes. Across 7 out of 8 solute families, fuel-water partitioning calculations using the alkane-water LSER (Table 3-1, (*37*)) gave significantly poorer predictions (Figure 3-1D) than the CSCLSST-LSER or LSST-LSER approach. Using the alkane-water LSER, solvation of polar solutes in the organic phase, particularly for phenols and anilines, was underpredicted by one to two orders of magnitude in the partition coefficient, K_{fw}. Additionally, alkane-water LSER predictions of nonpolar solute partitioning between gasolines or diesel fuel and water was significantly biased

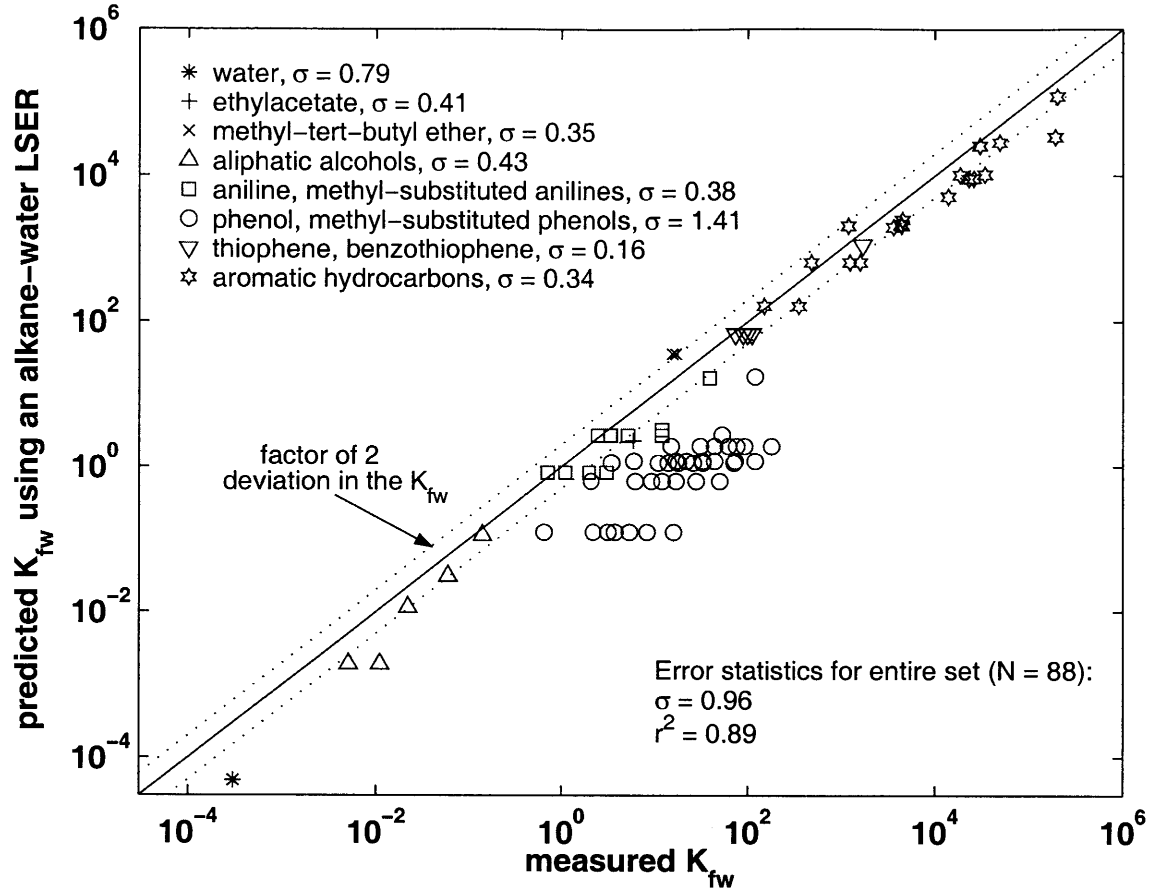

Figure 3-1D. Predicted K_{fw} values of polar and nonpolar compounds in simulated and retail fuel-water systems using the alkane-water LSER of Abraham et al. (*37*)

low compared to CSCLSST-LSER or LSST-LSER predictions. The alkane-water LSER biases found for both polar solutes and non-hydrogen bonding solutes demonstrated that, to an important extent, both CSCLSST-LSERs and LSST-LSERs captured the increased solvency of the fuel phase resulting from the presence of both aromatic hydrocarbons and MTBE.

3.4.2. Partitioning of aromatic hydrocarbons and ethanol in synthetic fuel-water/ethanol mixtures

Synthetic fuel-water systems containing 5 to 50 vol.% ethanol in the aqueous phase (*62*) reflect a more typical application of LSST, in which LSST is used to extrapolate the solubility of solutes in water modified with a miscible organic cosolvent (*51*). The partitioning behavior of nonpolar aromatic solutes and a polar solute, ethanol, were predicted using the LSST-LSER model ($\sigma_{\log K} = 0.17$; Figure 3-2A; Table 3-6) and the CSCLSST-LSER approach ($\sigma_{\log K} = 0.19$; Figure 3-2B; Table 3-6). These results confirm previous applications of LSST to the cosolvent effect on solute partitioning in fuel-water systems (*25,53,54*). In addition, fuel-water/ethanol mixture calculations showed that both the CSCLSST-LSER and LSST-LSER approach could treat mixtures in the organic phase, consistent with results discussed in the previous section.

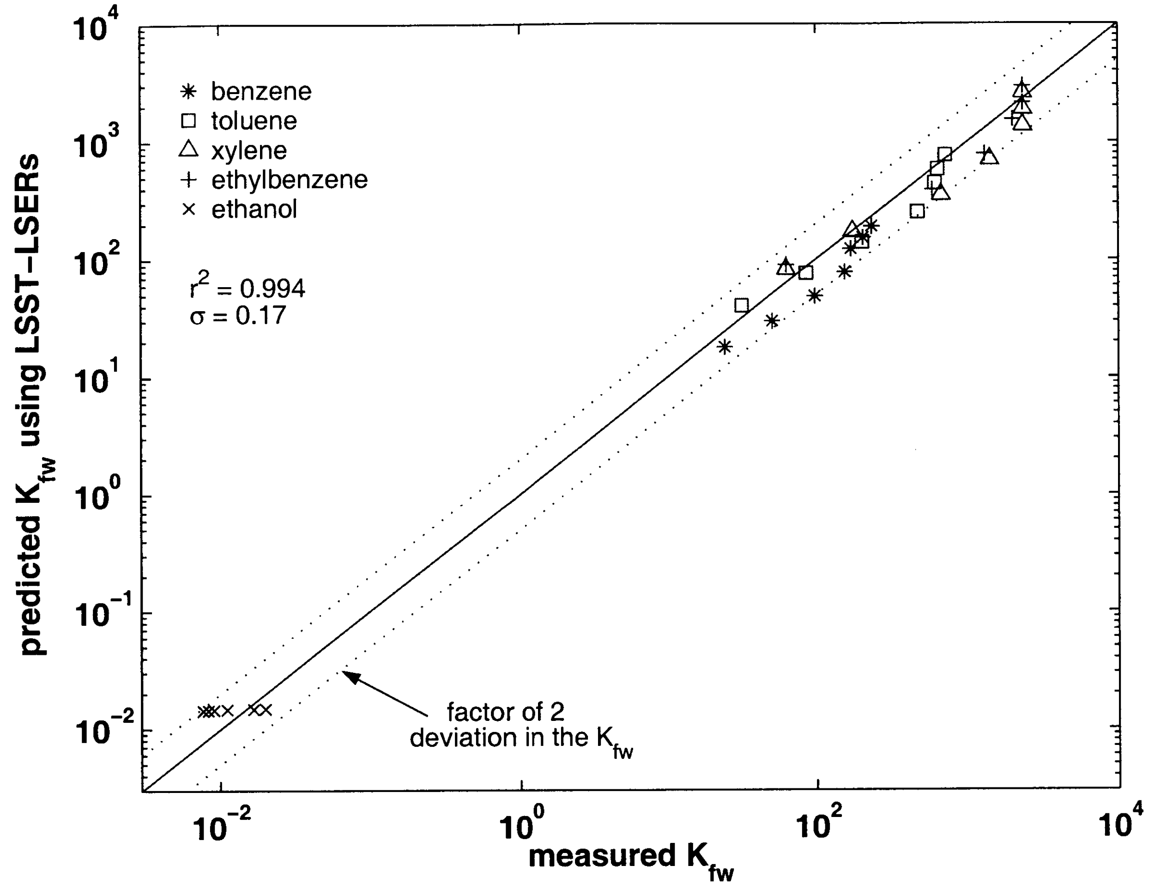

Figure 3-2A. Predicted K_{fw} values of BTEX and ethanol in synthetic fuel systems with ethanol amendments using the LSST-LSER model

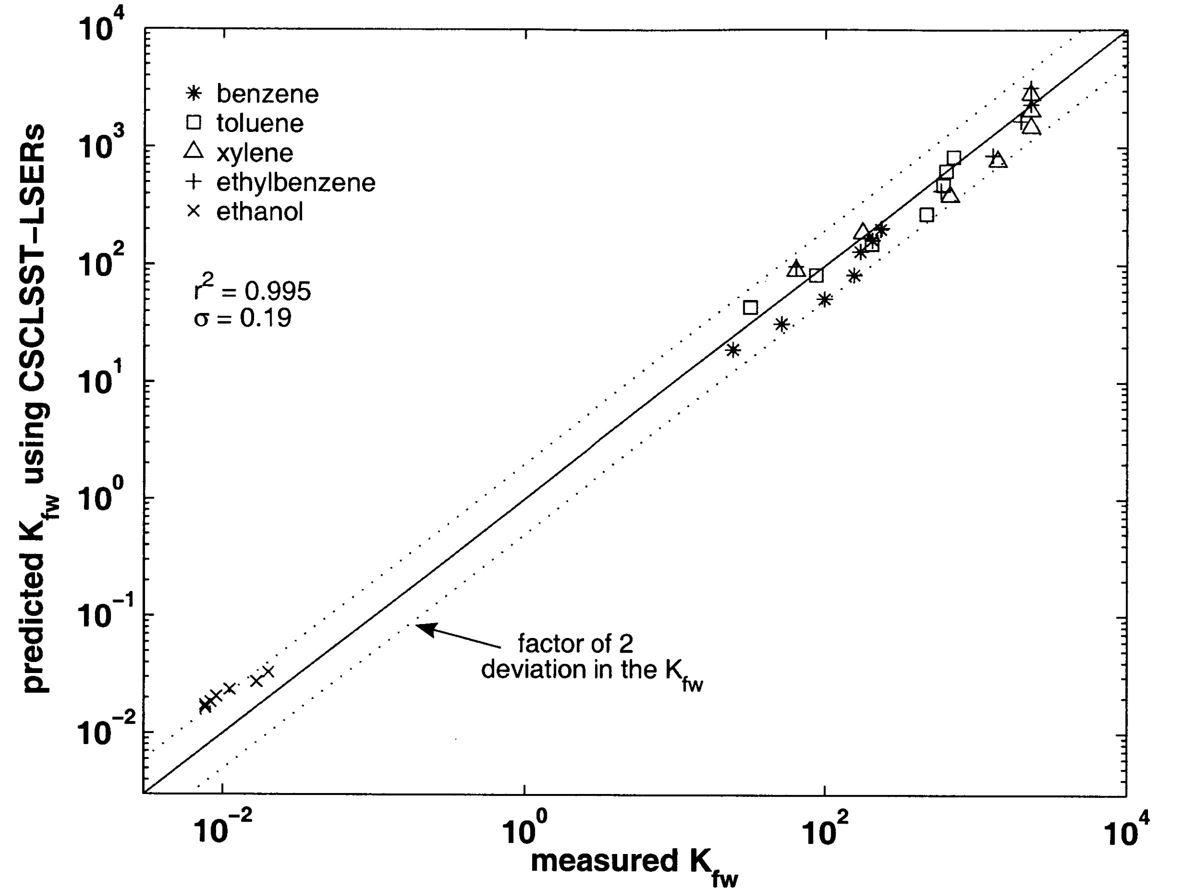

Figure 3-2B. Predicted K_{fw} values of BTEX and ethanol in synthetic fuel systems with ethanol amendments using the CSCLSST-LSER model

Table 3-6. Predicted and experimental partition coefficient values of polar and nonpolar compounds in various fuel-water systems (molar units)

		eqn 3-5	eqn 3-9	fuel system	
solute	K_{fw} meas.	K_{fw} pred.	K_{fw} pred.	(table-type)	experiment
methyl-*tert*-butylether	17	44	46	1-1	(*61*)
methyl-*tert*-butylether	15.5	50	52	2-conv.	(*20*)
methyl-*tert*-butylether	15.5	49	51	2-oxyg.	(*20*)
ethylacetate	5.9	3.7	4.6	1-1	(*61*)
water	0.0003	0.000082	0.00012	1-1	(*61*)
methanol	0.005	0.0029	0.0036	1-1	(*61*)
methanol	0.011	0.0039	0.0047	2-conv.	(*61*)
ethanol	0.022	0.017	0.021	1-1	(*61*)
ethanol	0.008	0.014	0.017	1-2	(*62*)
ethanol	0.008	0.014	0.017	1-2 (5% aq. ethanol)[a]	(*62*)
ethanol	0.008	0.014	0.018	1-2 (10% aq. ethanol)[a]	(*62*)
ethanol	0.009	0.015	0.021	1-2 (20% aq. ethanol)[a]	(*62*)
ethanol	0.01	0.015	0.024	1-2 (30% aq. ethanol)[a]	(*62*)
ethanol	0.02	0.015	0.028	1-2 (40% aq. ethanol)[a]	(*62*)

ethanol	0.02	0.015	0.033	1-2 (50% aq. ethanol)[a]	*(62)*
isopropanol	0.06	0.047	0.057	1-1	*(61)*
tert-butanol	0.14	0.17	0.20	1-1	*(61)*
aniline	3.1	2.4	4.7	1-3	*(5)*
aniline	0.71	0.96	1.8	1-4	*(5)*
aniline	1.1	1.3	3.6	1-5	*(5)*
aniline	2.0	2.0	6.4	1-6	*(5)*
p-toluidine	12	7.7	15	1-3	*(5)*
p-toluidine	2.5	3.1	4.9	1-4	*(5)*
p-toluidine	3.4	4.0	9.4	1-5	*(5)*
p-toluidine	5.2	6.1	16	1-6	*(5)*
o-toluidine	12	9.0	17	1-3	*(5)*
2,6-dimethylaniline	39	45	82	1-3	*(5)*
phenol	3.2	0.47	3.8	1-3	*(5)*
phenol	0.65	0.17	2.6	1-4	*(5)*
phenol	2.2	0.29	7.6	1-5	*(5)*
phenol	5.4	0.69	15	1-6	*(5)*
phenol	3.8	1.8	3.8	1-7	*(5)*
phenol	8.3	2.5	8.3	1-8	*(5)*
phenol	16	4.0	15	1-9	*(5)*
p-cresol	9.3	2.3	15	1-3	*(5)*
p-cresol	2.1	0.82	10	1-4	*(5)*
p-cresol	6.2	1.4	29	1-5	*(5)*
p-cresol	17	3.2	58	1-6	*(5)*
p-cresol	12	9.0	16	1-7	*(5)*
p-cresol	28	12	34	1-8	*(5)*
p-cresol	50	19	60	1-9	*(5)*
o-cresol	14	3.9	20	1-3	*(5)*
o-cresol	3.5	1.4	13	1-4	*(5)*
o-cresol	11	2.4	35	1-5	*(5)*
o-cresol	26	5.1	70	1-6	*(5)*
o-cresol	18	14	23	1-7	*(5)*
o-cresol	33	19	43	1-8	*(5)*
o-cresol	71	28	74	1-9	*(5)*
3,4-dimethylphenol	22	4.5	26	1-3	*(5)*
3,4-dimethylphenol	6.0	1.5	16	1-4	*(5)*
3,4-dimethylphenol	17	2.6	46	1-5	*(5)*
3,4-dimethylphenol	44	5.7	90	1-6	*(5)*
3,4-dimethylphenol	33	18	29	1-7	*(5)*
3,4-dimethylphenol	73	24	56	1-8	*(5)*
3,4-dimethylphenol	120	36	96	1-9	*(5)*
2,6-dimethylphenol	44	6.7	32	1-3	*(5)*
2,6-dimethylphenol	15	2.4	20	1-4	*(5)*
2,6-dimethylphenol	31	4.0	56	1-5	*(5)*
2,6-dimethylphenol	62	8.5	110	1-6	*(5)*
2,6-dimethylphenol	76	25	38	1-7	*(5)*
2,6-dimethylphenol	92	32	70	1-8	*(5)*
2,6-dimethylphenol	180	47	120	1-9	*(5)*
3,4,5-trimethylphenol	53	11	57	1-3	*(5)*
2,4,6-trimethylphenol	120	53	130	1-3	*(5)*
thiophene	110	99	110	1-3	*(5)*
thiophene	74	70	72	1-4	*(5)*
thiophene	99	76	83	1-5	*(5)*
thiophene	89	86	98	1-6	*(5)*
benzo[b]thiophene	1700	2300	3200	1-3	*(5)*
benzene	230	190	200	1-2	*(62)*

benzene	210	150	160	1-2 (5% aq. ethanol)[a]	(*62*)
benzene	170	120	130	1-2 (10% aq. ethanol)[a]	(*62*)
benzene	160	78	83	1-2 (20% aq. ethanol)[a]	(*62*)
benzene	99	49	52	1-2 (30% aq. ethanol)[a]	(*62*)
benzene	51	30	32	1-2 (40% aq. ethanol)[a]	(*62*)
benzene	25	18	19	1-2 (50% aq. ethanol)[a]	(*62*)
benzene	150	190	200	1-10	(*20*)
benzene	350	280	300	2-conv.	(*20*)
benzene	350	280	310	2-oxyg.	(*20*)
toluene	710	760	830	1-2	(*62*)
toluene	640	580	630	1-2 (5% aq. ethanol)[a]	(*62*)
toluene	610	440	480	1-2 (10% aq. ethanol)[a]	(*62*)
toluene	470	250	270	1-2 (20% aq. ethanol)[a]	(*62*)
toluene	200	140	150	1-2 (30% aq. ethanol)[a]	(*62*)
toluene	87	76	82	1-2 (40% aq. ethanol)[a]	(*62*)
toluene	32	41	44	1-2 (50% aq. ethanol)[a]	(*62*)
toluene	480	750	810	1-10	(*20*)
toluene	1250	1200	1300	2-conv.	(*20*)
toluene	1250	1200	1300	2-oxyg.	(*20*)
m-xylene	2300	2600	2800	1-2	(*62*)
m-xylene	2300	1900	2100	1-2 (5% aq. ethanol)[a]	(*62*)
m-xylene	2300	1400	1500	1-2 (10% aq. ethanol)[a]	(*62*)
m-xylene	1400	700	770	1-2 (20% aq. ethanol)[a]	(*62*)
m-xylene	670	350	390	1-2 (30% aq. ethanol)[a]	(*62*)
m-xylene	180	170	190	1-2 (40% aq. ethanol)[a]	(*62*)
m-xylene	64	82	90	1-2 (50% aq. ethanol)[a]	(*62*)
m-xylene	4350	4100	4600	2-conv.	(*20*)
m-xylene	4350	4100	4700	2-oxyg.	(*20*)
o-xylene	3630	3900	4500	2-conv.	(*20*)
o-xylene	3630	3900	4600	2-oxyg.	(*20*)
ethylbenzene	2300	2900	3200	1-2	(*62*)
ethylbenzene	2300	2100	2300	1-2 (5% aq. ethanol)[a]	(*62*)
ethylbenzene	2000	1500	1700	1-2 (10% aq. ethanol)[a]	(*62*)
ethylbenzene	1300	780	860	1-2 (20% aq. ethanol)[a]	(*62*)
ethylbenzene	580	390	430	1-2 (30% aq. ethanol)[a]	(*62*)
ethylbenzene	230	190	210	1-2 (40% aq. ethanol)[a]	(*62*)
ethylbenzene	64	90	98	1-2 (50% aq. ethanol)[a]	(*62*)
ethylbenzene	4500	4600	5200	2-conv.	(*20*)
ethylbenzene	4500	4600	5300	2-oxyg.	(*20*)
n-propylbenzene	18500	20000	23000	2-conv., 2-oxyg.	(*20*)
1,2,3-trimethylbenzene	13800	11000	14000	2-conv., 2-oxyg.	(*20*)
naphthalene	1180, 4400, 4800	2800	3800	1-10	(*20,21,70*)
1-methylnaphthalene	23000, 20000	12000	17000	1-10	(*21,70*)
2-methylnaphthalene	26000	12000	17000	1-10	(*21*)
acenaphthene	34000	15000	23000	1-10	(*21*)
fluorene	30000	37000	60000	1-10	(*21*)
phenanthrene	49000	44000	94000	1-10	(*21*)
anthracene	190000	53000	120000	1-10	(*21*)
fluoranthene	200000	210000	600000	1-10	(*21*)

[a] amendment resulting in this level of ethanol measured in the aqueous phase at equilibrium, by volume.

3.4.3. Partitioning of benzene/alcohol-water, toluene/alcohol-water, and benzene/MTBE-water systems

In ternary systems containing either benzene or toluene, water, and a C_4 to C_6 aliphatic alcohol (Table 3-3; Figure 3-3A; Figure 3-3B), partitioning of all three components (including water) was calculated to within a factor of 2 to 4 of observed K_{fw} values using the LSST-LSER approach ($\sigma_{\log K} = 0.41$ overall). The CSCLSST-LSER model gave more accurate estimates for organic system components ($\sigma_{\log K} = 0.25$) but predicted water partitioning poorly ($\sigma_{\log K} = 0.76$). The aliphatic alcohols partitioned primarily into the organic phase in all of the systems studied (i.e., $K_{fw} > 1$ for these solutes). Both CSCLSST-LSER and LSST-LSER-calculated K_{fw} values for aromatic hydrocarbons were consistently biased low: as the abundance of aliphatic alcohol was increased, solvation in the organic phase was more favorable for the water and aromatic hydrocarbon solutes than was indicated by either model. Additionally, the LSST-LSER approach usually underpredicted water K_{fw} values, whereas the CSCLSST-LSER model consistently overpredicted K_{fw} for water. By comparison, in benzene/MBTE-water systems, the LSST-LSER model accurately calculated the partitioning of all three components over the entire

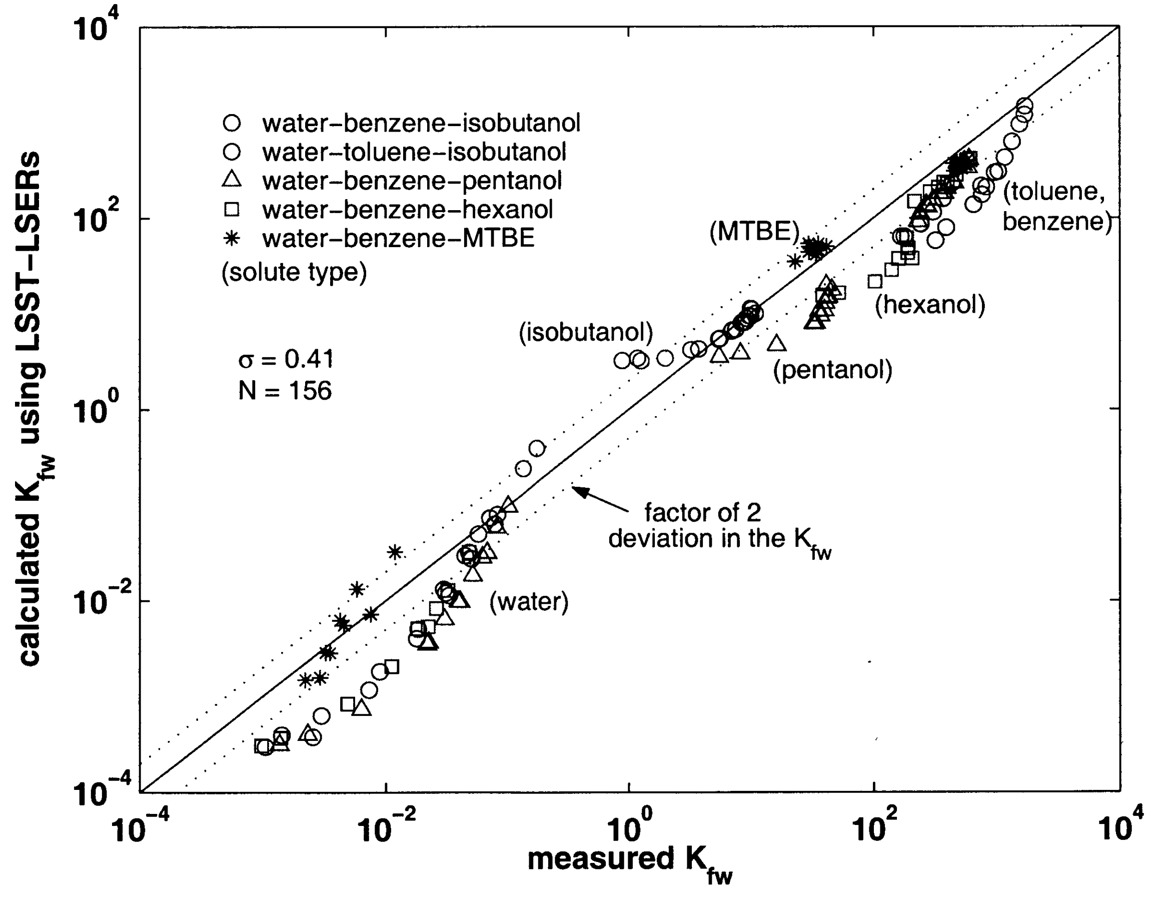

Figure 3-3A. Calculated K_{fw} values for ternary benzene/toluene-aliphatic alcohol-water mixtures and benzene-MTBE-water mixtures using the LSST-LSER model. Symbols indicate type of system; parenthetical labels indicate type of solute (e.g., K_{fw} values of the water solute in water-benzene-pentanol mixture systems are shown as "Δ" points near the "(water)" label)

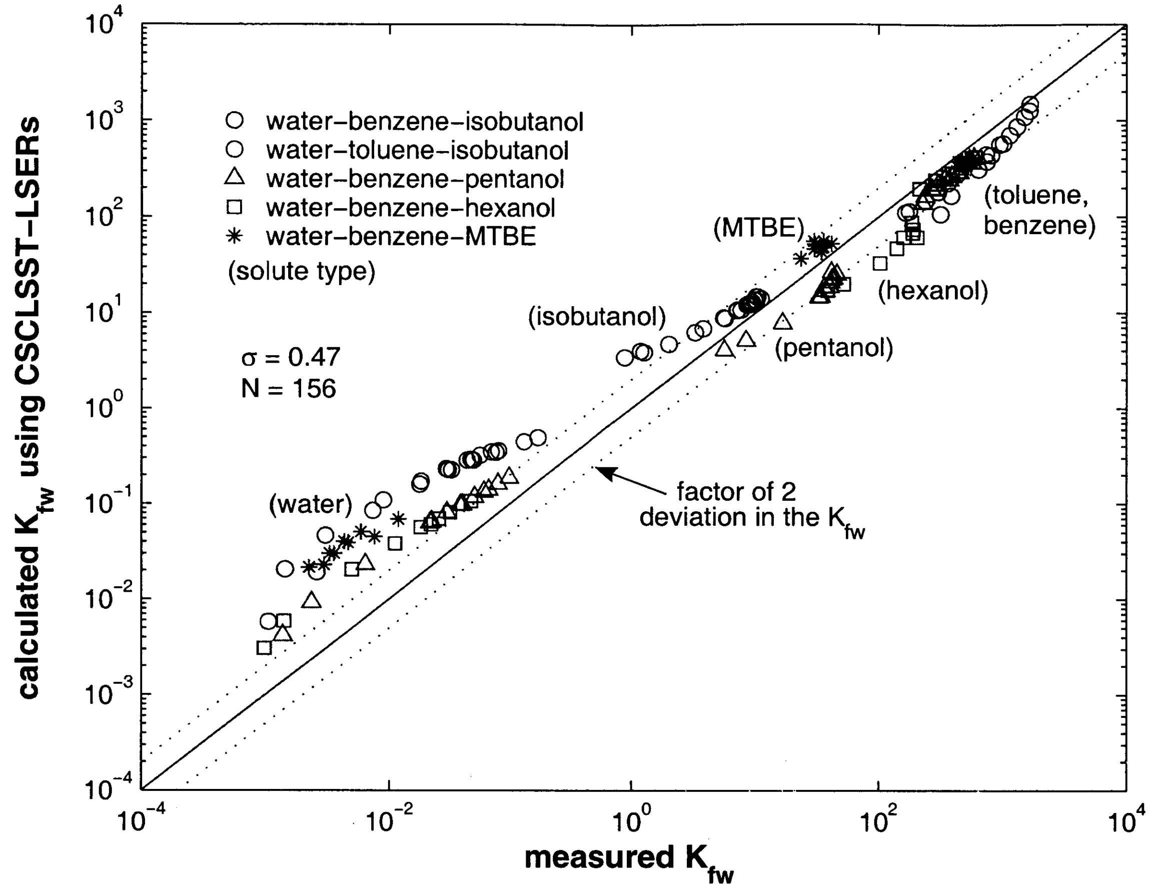

Figure 3-3B. Calculated K_{fw} values for ternary benzene/toluene-aliphatic alcohol-water mixtures and benzene-MTBE-water mixtures using the CSCLSST-LSER model

range of compositions considered ($\sigma_{\log K}$ = 0.18). In agreement with previously discussed results, LSST-LSERs could make meaningful partitioning predictions ($\sigma_{\log K}$ ~ 0.4) of both polar and nonpolar solutes, given prior knowledge of the ternary system compositions.

However if mixture composition information was not provided to set the volume fractions needed in the LSST-LSER or CSCLSST-LSER estimates for ternary systems, computed results were considerably worse. Iterative calculation of equation 3-5 or 3-9 for ϕ and K_{fw} values to self-consistency under mass conservation constraints produced only order-of-magnitude accuracy for K_{fw} predictions of ternary mixture components ($\sigma_{\log K}$ = 0.98 for LSST-LSER estimates, data not shown). Both approaches made inadequate predictions using mass conservation calculations for these ternary systems because the organic phase was typically composed of a high water content (5 to 20 vol.%). Moderate errors in the calculated water content of the organic phase propagated to considerable changes in the mixture LSER coefficients. This in turn led to exponentially magnified errors in predictions of both water partitioning and that of other components. I therefore expect either the CSCLSST-LSER or LSST-LSER model to make inaccurate composition predictions for systems in which highly polar compounds (e.g., water) are significant constituents of the organic phase.

3.4.4. Synthesis of results

Applicability of the LSST-LSER and CSCLSST-LSER approaches rests on the conditions that: (a) LSERs are an adequate model for characterizing solute partitioning in mixture systems; (b) LSST is a reliable mixing rule for the aqueous phase; (c) LSST and the solvent compartment model are reliable mixing rules for the organic phase; and (d) the mixture solvation properties can be accurately extrapolated from dilute solute conditions. Of these three assumptions, the first was considered unlikely to contribute dominantly to model failure, since the accuracy of LSER-predicted solvation energies is about $\sigma_{\log K,\ \mathrm{LSER}} = 0.16$ for a wide range of solvents and mixtures. Model error related specifically to the LSST or CSCLSST mixing rule assumption could be estimated using first-order error propagation analysis (*71*), assuming that uncertainties associated with the LSER model, K_{fw} measurements, and the LSST mixing rule were uncorrelated. K_{fw} measurement errors were considered $\sigma_{\log K,\ \mathrm{meas}} \sim 0.08$ for the studies considered here. Given $\sigma_{\log K} = 0.4$ overall for LSST-LSER estimates, the LSST mixing rule error was thus estimated to be $\sigma_{\log K,\ \mathrm{LSST}} = 0.3$ (which includes inaccuracy due to extrapolation from infinite dilution conditions). This is comparable to the previous results of Li (*52*), in which the LSST approximation was shown to have standard deviations ranging from 0.1 to 0.4 log *K* units for binary mixtures (depending on the type of cosolvent). For the systems studied by Li, LSST tended to perform best when the mixture was dominated by one solvent ($\phi \sim 0.80$ or more) and when the solute was less polar than the mixture (*52*). CSCLSST-LSER model error was also $\sigma_{\log K} = 0.4$ over all data considered in this study, so the CSCLSST mixing rule error was estimated to also be $\sigma_{\mathrm{CSCLSST}} = 0.3$ using the analysis described above.

The results shown here suggest that both the CSCLSST-LSER and LSST-LSER approaches captured the effect of most polar and aromatic cosolvent amendments, relative to an alkane solvent. However, we concluded that the likelihood of hydrogen-bonding solutes to be solvated by complementary solvent components in the nonpolar phase was usually underpredicted by LSST and overpredicted by the solvent compartment model. For example, LSST-LSER K_{fw} predictions were generally biased low for hydrogen-bond donors such as phenol and water in synthetic fuel-water systems, whereas CSCLSST-LSER predictions were usually biased high for these solutes. Consequently I proposed a logarithmic average of these two models, defined as:

$$\log\left(K_{\mathrm{fw,HSCLSST-LSER}}\right) = \frac{1}{2}\log\left(K_{\mathrm{fw,CSCLSST-LSER}}\right) + \frac{1}{2}\log\left(K_{\mathrm{fw,LSST-LSER}}\right) \qquad (3\text{-}12)$$

which I refer to as the "half solvent compartment / LSST-LSER" model. When applied to all systems discussed in this work, equation 3-12 calculated K_{fw} values with an overall standard deviation of $\sigma_{\log K} = 0.26$ (Figure 3-4). This was a considerable improvement over either the CSCLSST-LSER or LSST-LSER model, primarily due to cancellation of errors for polar solutes.

A fuel-water mixtures partitioning model which can be applied to wide range of environmental solutes is needed. Although Raoult's law is commonly applied for estimating solute partitioning into environmental mixtures, it is inadequate when the solute structure implies hydrogen-bond donation and components of the mixture can serve as hydrogen-bond acceptors. The LSST-LSER and CSCLSST-LSER approaches do not calculate solute partitioning behavior

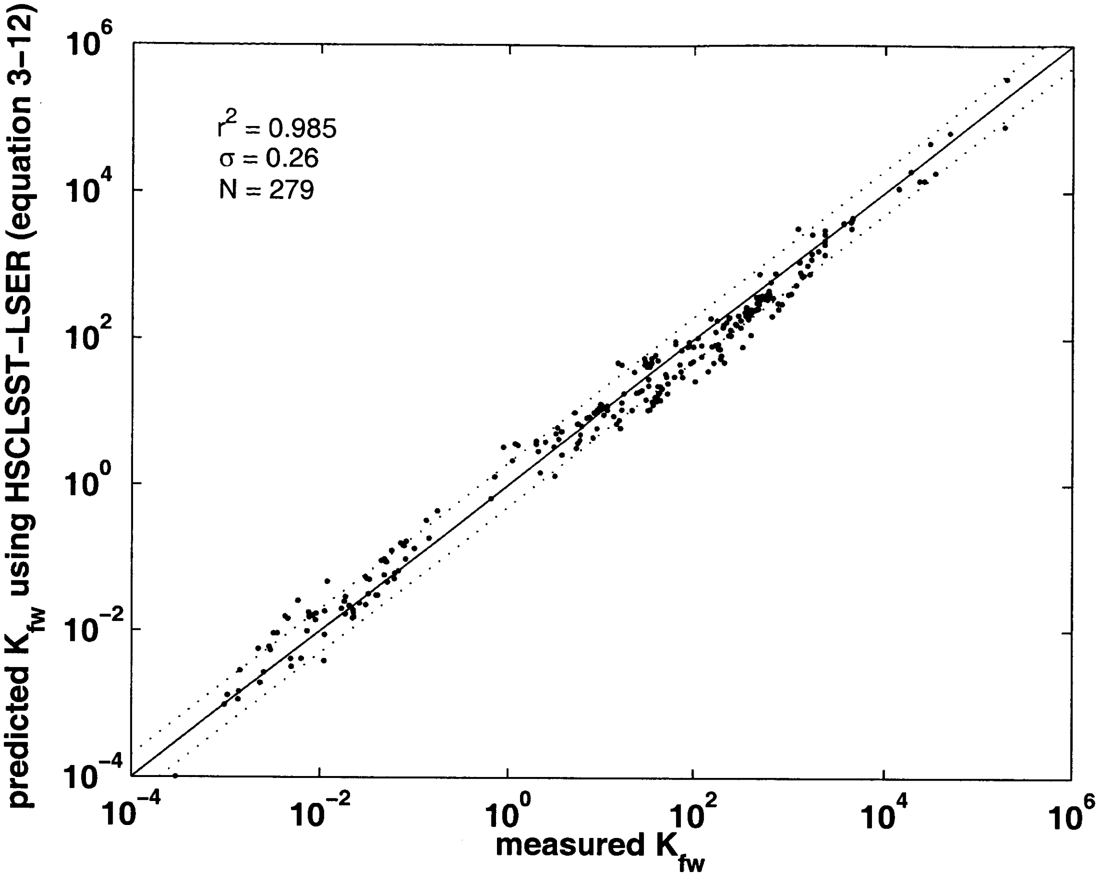

Figure 3-4. HSCLSST-LSER calculated K_{fw} values for all systems

in mixtures as accurately as LSERs derived from original data. Additionally, the systems considered here do not to reflect a rigorous evaluation of these models for organic and aqueous mixtures in general. However, this work suggests that equations 3-5, 3-9, and 3-12 are suited for common fuel mixtures, and the results may inspire evaluation of these approaches for other environmentally relevant media. Once a mixture LSER for a fuel-water system has been estimated, solvation energies for a broad set of solutes may be estimated. This is not true of other conventional approaches such as UNIFAC (*12-14*), which frequently lack the interaction parameters necessary to estimate the behaviors of solutes in fuel mixtures (*63*).

3.5. Conclusions

Without any additional fitting, several pure-phase LSERs were combined using linear solvent strength theory and the solvent compartment model to estimate the partitioning of 37 different solutes between various fuel or fuel-like mixtures and an aqueous phase. Previous research suggests that LSERs can accurately model partitioning in many organic mixtures (*33,34,38,59*). The challenge, then, lies in finding a general method to estimate the best LSER coefficients for such mixtures, in the absence of the copious data required for a conventional regression analysis. In this study, the linear solvent strength approximation and solvent compartment model were used to generalize the application of pure-phase LSER coefficients to a range of mixtures. The resulting model predictive standard errors for solutes in synthetic and realistic fuel systems composed of nonpolar hydrocarbons and MTBE or alcohols were estimated to be ~0.4 in the log K, using either approach. Noticing that these model descriptions of the fuel phase had systematic and opposite errors, I proposed a logarithmic average of the two approaches (equation 3-12). The resulting empirical average gave considerably improved K_{fw} estimates, having a standard deviation of 0.26 in the log K across the entire set of polar and nonpolar solute partitioning data reviewed here (N = 279). This level of accuracy is suitable for many applications in environmental fate analysis of organic pollutants.

3.6. Acknowledgments

The authors extend thanks to Drs. Torsten Schmidt and Kai-Uwe Goss for helpful conversations concerning measurement of gasoline solute partitioning and the solvent compartment model; to Farzan Parsinejad for his assistance in reviewing the composition of fuels; and to Professors William H. Green Jr. and Bernhardt Trout for their comments on the manuscript. This work was supported by the Environmental Protection Agency/Office of Naval Research/National Center for Environmental Research and Quality Assurance Science to Achieve Results award R-82902301-0, the Alliance for Global Sustainability, the Martin Family Society of Fellows for Sustainability, the Ralph M. Parsons Fellowship Foundation, and the Schoettler Scholarship Fund.

3.7. References

(1) Squillace, P. J.; Zogorski, J. S.; Wilber, W. G.; Price, C. V., *Preliminary assessment of the occurrence and possible sources of MTBE in groundwater in the United States, 1993-1994*. Environmental Science & Technology **1996**, *30*, 1721-1730.

(2) Davis, J. M.; Brophy, J.; Hitzig, R.; Kremer, F.; Osinski, M.; Prah, J. D. "Oxygenates in Water: Critical Information and Research Needs," Office of Research and Development, U.S. Environmental Protection Agency, 1998, 600/R-98/048.

(3) Greenbaum, D.; Beuhler, M.; Campbell, R.; Ellis, P.; Greer, L.; Grumet, J.; Happel, A.; Henry, C.; Kenny, M.; Sawyer, R.; Sneller, T.; Starnes, D.; White, R. "Achieving Clean Air and Clean Water: The Report of the Blue Ribbon Panel on Oxygenates in Gasoline," U.S. Environmental Protection Agency, 1999, EPA420-R-99-021.

(4) Johnson, R.; Pankow, J. F.; Bender, D.; Price, C.; Zogorski, J. S., *MTBE, To what extent will past releases contaminate community water supply wells?* Environmental Science & Technology **2000**, *34*, 2A-9A.

(5) Schmidt, T. C.; Kleinert, P.; Stengel, C.; Goss, K.-U.; Haderlein, S. B., *Polar fuel constituents: compound identification and equilibrium partitioning between nonaqueous phase liquids and water*. Environmental Science & Technology **2002**, *36*, 4074-4080.

(6) MacFarlane, S.; Mackay, D., *A fugacity-based screening model to assess contamination and remediation of the subsurface containing non-aqueous phase liquids*. Journal of Soil Contamination **1998**, *17*, 17-46.

(7) Powers, S. E.; Rice, D.; Dooher, B.; Alvarez, P. J. J., *Will ethanol-blended gasoline affect groundwater quality?* Environmental Science & Technology **2001**, *35*, 24A-30A.

(8) Mackay, A. A.; Gschwend, P. M., *Sorption of monoaromatic hydrocarbons to wood.* Environmental Science & Technology **2000**, *34*, 839-845.

(9) Kile, D. E.; Chiou, C. T.; Zhou, H.; Li, H.; Xu, O., *Partition of nonpolar organic pollutants from water to soil and sediment organic matters.* Environmental Science & Technology **1995**, *29*, 1401-1406.

(10) Herbert, B. E.; Bertsch, P. M.; Novak, J. M., *Pyrene sorption by water-soluble organic carbon.* Environmental Science & Technology **1993**, *27*, 398-403.

(11) Abrams, D. S.; Prausnitz, J. M., *Statistical thermodynamics of liquid mixtures: a new expression for the excess Gibbs energy of partly or completely miscible systems.* AIChE J. **1975**, *21*, 116-128.

(12) Fredenslund, A.; Jones, R. J.; Praustnitz, J. M., *Group-contribution estimation of activity coefficients in nonideal liquid mixtures.* AIChE J. **1975**, *21*, 1086-1099.

(13) Gmehling, J.; Li, J.; Schiller, M., *A modified UNIFAC model. 2. Present parameter matrix and results for different thermodynamic properties.* Industrial & Engineering Chemistry Research **1993**, *32*, 178-193.

(14) Gmehling, J.; Lohmann, J.; Jakob, A.; Li, J. D.; Joh, R., *A modified UNIFAC (Dortmund) model. 3. Revision and extension.* Industrial & Engineering Chemistry Research **1998**, *37*, 4876-4882.

(15) Abraham, M. H.; Poole, C. F.; Poole, S. K., *Classification of stationary phases and other materials by gas chromatography.* Journal of Chromatography A **1999**, *842*, 79-114.

(16) Schwarzenbach, R. P.; Gschwend, P. M.; Imboden, D. M. *Environmental Organic Chemistry*; 2nd ed.; John Wiley & Sons: New York, NY, **2003**.

(17) Hine, J.; Mookerjee, P. K., *The intrinsic hydrophilic character of organic compounds. Correlations in terms of structural contributions.* Journal of Organic Chemistry **1975**, *40*, 292-298.

(18) Pinsuwan, S.; Myrdal, P. B.; Lee, Y. C.; Yalkowsky, S. H., *AQUAFAC 5: aqueous functional group activity coefficients; application to alcohols and acids.* Chemosphere **1997**, *35*, 2503-2513.

(19) Hansch, C.; Leo, A. J. *Substituent constants for correlation analysis in chemistry and biology*; John Wiley & Sons: New York, NY, **1979**.

(20) Cline, P. V.; Delfino, J. J.; Rao, P. S. C., *Partitioning of aromatic constituents into water from gasoline and other complex solvent mixtures.* Environmental Science & Technology **1991**, *25*, 914-920.

(21) Lee, L. S.; Hagwall, M.; Delfino, J. J.; Rao, P. S. C., *Partitioning of polycyclic aromatic hydrocarbons from diesel fuel into water.* Environmental Science & Technology **1992**, *26*, 2104-2110.

(22) Lee, L. S.; Rao, P. S. C.; Okuda, I., *Equilibrium partitioning of polycyclic aromatic hydrocarbons from coal tar into water.* Environmental Science & Technology **1992**, *26*, 2110-2115.

(23) Lane, W. F.; Loehr, R. C., *Estimating the equilibrium aqueous concentrations of polynuclear aromatic hydrocarbons in complex mixtures.* Environmental Science & Technology **1992**, *26*, 983-990.

(24) Chen, C. H.-S.; Delfino, J. J.; Rao, P. S. C., *Partitioning of organic and inorganic components from motor oil into water.* Chemosphere **1994**, *28*, 1385-1400.

(25) Reckhorn, S. B. F.; Zuquette, L. V.; Grathwohl, P., *Experimental investigations of oxygenated gasoline dissolution*. Journal of Environmental Engineering **2001**, *127*, 208-216.

(26) Mukherji, S.; Peters, C. A.; Weber, W. J. J., *Mass transfer of polynuclear aromatic hydrocarbons from complex DNAPL mixtures*. Environmental Science & Technology **1997**, *31*, 416-423.

(27) Owen, K. *Gasoline and Diesel Fuel Additives*; John Wiley & Sons, **1989**.

(28) Schubert, A. J.; Johansen, N. G., *Cooperative study to evaluate a standard test method for the speciation of gasolines by capillary gas chromatography*. Society of Automotive Engineering **1993**, *930144*.

(29) Hogue, C., *Rethinking Ethanol. Court orders EPA to revisit California requeston gasoline standard*. Chemical & Engineering News **2003**, *81*, 12.

(30) Sjogren, M.; Li, H.; Rannug, U.; Westerholm, R., *A multivariate statistical analysis of chemical composition and physical characteristics of ten diesel fuels*. Fuel **1995**, *74*, 983-989.

(31) Edwards, T.; *"Kerosene" fuels for aerospace propulsion - composition and properties.* 38th AIAA/ASME/AE/ASEE Joint Propulsion Conference and Exhibit; American Institute of Aeronautics and Astronautics: Indianapolis, IN, 2001.

(32) Peters, C. A.; Luthy, R. G., *Coal tar dissolution in water-miscible solvents: experimental evaluation*. Environmental Science & Technology **1993**, *27*, 2831-2843.

(33) Abraham, M. H.; Priscilla, L. G.; McGill, R. A., *Determination of olive oil-gas and hexadecane-gas partition coefficients, and calculation of the corresponding olive oil-water and hexadecane-water partition coefficients*. Journal of the Chemical Society - Perkins Transactions 2 **1987**, 797-803.

(34) Abraham, M. H.; Whiting, G. S., *Hydrogen-bonding. 22. Characterization of soybean oil and prediction of activity coefficients in soybean oil from inverse gas-chromatographic data*. Journal of the American Oil Chemists Society **1992**, *69*, 1236-1238.

(35) Abraham, M. H.; J, A.-H.; Whiting, G. S.; Leo, A.; Taft, R. S., *Hydrogen bonding. Part 34. The factors that influence the solubility of gases and vapours in water at 298 K, and a new method for its determination*. Journal of the Chemical Society - Perkins Transactions 2 **1994**, 1777-1791.

(36) Abraham, M. H.; Chadha, H. S.; Dixon, J. P.; Leo, A. J., *Hydrogen-bonding. 39. The partition of solutes between water and various alcohols*. Journal of Physical Organic Chemistry **1994**, *7*, 712-716.

(37) Abraham, M. H.; Chadha, H. S.; Whiting, G. S.; Mitchell, R. C., *Hydrogen-bonding. 32. An analysis of water-octanol and water-alkane partitioning and the delta-logP parameter of Seiler*. Journal of Pharmaceutical Sciences **1994**, *83*, 1085-1100.

(38) Goss, K. U.; Schwarzenbach, R. P., *Linear free energy relationships used to evaluate equilibrium partitioning of organic compounds*. Environmental Science & Technology **2001**, *35*, 1-9.

(39) Abraham, M. H.; Whiting, G. S., *XVI. A new solute solvation parameter, π_2^H, from gas chromatographic data*. Journal of Chromatography **1991**, *587*, 213-228.

(40) Abraham, M. H.; Grellier, P. L.; Prior, D. V.; Duce, P. P.; Morris, J. J.; Taylor, P. J., *Hydrogen bonding. Part 7. A scale of solute hydrogen-bond acidity based on logK values for complexation in tetrachloromethane*. Journal of the Chemical Society - Perkins Transactions 2 **1989**, 699-711.

(41) Abraham, M. H.; Grellier, P. L.; Prior, D. V.; Morris, J. J.; Taylor, P. J., *Hydrogen bonding. Part 10. A scale of solute hydrogen-bond basicity using logK values for complexation in tetrachloromethane*. Journal of the Chemical Society - Perkins Transactions 2 **1990**, 521-529.

(42) Abraham, M. H., *Hydrogen bonding. 31. Construction of a scale of solute effective or summation hydrogen-bond basicity*. Journal of Physical Organic Chemistry **1993**, *6*, 660-684.

(43) Abraham, M. H.; McGowan, J. C., *The use of characteristic volumes to measure cavity terms in reversed phase liquid chromatography*. Chromatographia **1987**, *23*, 243-246.

(44) Pinal, R.; Rao, P. S. C.; Lee, L. S.; Cline, P. V.; Yalkowsky, S. H., *Cosolvency of partially miscible organic solvents on the solubility of hydrophobic organic chemicals*. Environmental Science & Technology **1990**, *24*, 639-647.

(45) Yalkowsky, S. H.; Flynn, G. L.; Amidon, G. L., *Solubility of nonelectrolytes in polar solvents*. Journal of Pharmaceutical Sciences **1972**, *61*, 983-984.

(46) Yalkowsky, S. H.; Valvani, S. C.; Amidon, G. L., *Solubility of nonelectrolytes in polar solvents IV: nonpolar drugs in mixed solvents*. Journal of Pharmaceutical Sciences **1976**, *65*, 1488-1494.

(47) Synder, L. R., *Gradient elution in high-performance liquid chromatography 1. Theoretical basis for reversed-phase systems*. Journal of Chromatography **1979**, *165*, 3-30.

(48) Snyder, L. R., *Gradient Elution*, In *High Performance Liquid Chromatography*; Horvath, C., Ed.; Academic Press: New York, **1980**; Vol. 1, pp 207-316.

(49) Fu, J.-K.; Luthy, R. G., *Aromatic compound solubility in solvent/water mixtures*. Journal of Environmental Engineering **1986**, *112*, 328-345.

(50) Fu, J.-K.; Luthy, R. G., *Effect of organic solvent on sorption of aromatic solutes onto soils*. Journal of Environmental Engineering **1986**, *112*, 346-366.

(51) Morris, K. R.; Abramowitz, R.; Pinal, R.; Davis, P.; Yalkowsky, S. H., *Solubility of aromatic pollutants in mixed solvents*. Chemosphere **1988**, *17*, 285-298.

(52) Li, A., *Predicting cosolvency. 3. Evaluation of the extended log-linear model*. Industrial & Engineering Chemistry Research **2001**, *40*, 5029-5035.

(53) Poulsen, M.; Lemon, L.; Barker, J. F., *Dissolution of monoaromatic hydrocarbons into groundwater from gasoline-oxygenate mixtures*. Environmental Science & Technology **1992**, *26*, 2483-2489.

(54) Chen, C. H.-S.; Delfino, J. J., *Cosolvent effects of oxygenated fuels on PAH solubility*. Journal of Environmental Engineering **1997**, *123*, 354-363.

(55) Spurlock, F. C.; Biggar, J. W., *Thermodynamics of organic chemical partition in soils. 3. Nonlinear partition from water-miscible cosolvent solutions*. Environmental Science & Technology **1994**, *28*, 1003-1009.

(56) Lee, L. S.; Bellin, C. A.; Pinal, R.; Rao, P. S. C., *Cosolvents effects on sorption of organic acids by soils from mixed solvents*. Environmental Science & Technology **1993**, *27*, 165-171.

(57) Lee, L. S.; Rao, P. S. C., *Impact of several water-miscible organic solvents on sorption of benzoic acid by soil*. Environmental Science & Technology **1996**, *30*, 1533-1539.

(58) Hayden, N. J.; Brooks, M. C.; Annable, M. D.; Zhou, H., *Activated carbon for removing tetrachloroethylene from alcohol solutions*. Journal of Environmental Engineering **2001**, *127*, 1116-1123.

(59) Wang, A.; Tan, L. C.; Carr, P. W., *Global linear solvation energy relationships for retention prediction in reversed-phase liquid chromatography.* Journal of Chromatography **1999**, *848*, 21-37.

(60) Kawamoto, K.; Arey, J. S.; Gschwend, P. M., *Emission and fate assessment of methyl tertiary butyl ether in the Boston area airshed using a simple multimedia box model: comparison with urban air measurements.* Journal of the Air & Waste Management Association **2003**, *53*, 1426-1435.

(61) Stephenson, R. M., *Mutual solubilities: water-ketones, water-ethers, and water-gasoline-alcohols.* Journal of Chemical Engineering Data **1992**, *37*, 80-95.

(62) Heermann, S. E.; Powers, S. E., *Modeling the partitioning of BTEX in water-reformulated gasoline systems containing ethanol.* Journal of Contaminant Hydrology **1998**, *34*, 315-341.

(63) Arey, J. S. M.S. Thesis Massachusetts Institute of Technology, 2001.

(64) Abraham, M. H.; Le, J.; Acree, W. E., *The solvation properties of the aliphatic alcohols.* Collection of Czechoslovak Chemical Communications **1999**, *64*, 1748-1760.

(65) CRC, *Physical Constants of Organic Compounds*, In *Handbook of Chemistry and Physics*; 77 ed.; Lide, D., Ed.; CRC Press, Inc: Boca Raton, FL, **1997**.

(66) Myrdal, P. B.; Yalkowsky, S. H., *Estimating pure component vapor pressures of complex organic molecules.* Industrial & Engineering Chemistry Research **1997**, *36*, 2494-2499.

(67) Chirico, R. D.; Knipmeyer, S. E.; Nguyen, A.; Steele, W. V., *The thermodynamic properties of benzo[b]thiophene.* Journal of Chemical Thermodynamics **1991**, *23*, 759-779.

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

(69) Yalkowsky, S. H. *Solubility and Solubilization in Aqueous Media*; Oxford University Press: New York, NY, **1999**.

(70) Yang, Y.; Miller, D. J.; Hawthorne, S. B., *Toluene solubility in water and organic partitioning from gasoline and diesel fuel into water at elevated temperatures and pressures.* Journal of Chemical Engineering Data **1997**, *42*, 908-913.

(71) Harris, D. C., *Appendix C: A Detailed Look at Propagation of Uncertainty*, In *Quantitative Chemical Analysis*; 2nd Ed. ed.; W.H. Freeman and Co.: New York, NY, **1987**; pp 712-714.

Chapter 4
Use of electrostatic computations to estimate the empirical solute polarity parameter, π_2^H

4.1. Introduction

4.1.1. Motivation

The development of Linear Solvation Energy Relationships (LSERs) has contributed significant insight into the physical chemical processes governing solute-solvent interactions (*1*). Additionally, LSERs have been shown to accurately predict solvation free energies for a wide range of dilute solutes across different solvent environments (*2, 3*). Consequently, LSERs have potential applicability in diverse separation sciences, environmental toxicological screening, environmental engineering, and pharmacology, among others. Despite these successes, the determination of most LSER parameter values remains essentially empirical. In particular, the solute polarity scales require a considerable amount of experimental data to fit (*4, 5*), and they continue to elude satisfactory correlations with more fundamental quantities. To date, the most extensively developed empirical solute polarity parameter is that of Abraham and co-workers, π_2^H. The main aim of this work was to develop a competitive method for estimating π_2^H values using molecular orbital calculations, which would in turn lead to accurate solvation energy estimates for unstudied solutes in many liquid-liquid and liquid-gas systems. Additionally, such an investigation could shed light on the physical origins of this highly empirical free energy parameter.

A physical understanding of π_2^H must be placed in context of the development of LSERs. The LSER equation formulated by Abraham and co-workers is (*2*):

$$\log P = mV_x + rR_2 + s\pi_2^H + a\Sigma\alpha_2^H + b\Sigma\beta_2^H + c \tag{4-1}$$

where P is a partitioning property of a solute between two bulk phases of interest, and m, r, s, a, b, and c are fitted coefficients, characteristic to a given two-phase system. V_x is a group contributable solute volume which accounts for both the solvent cavitation energy and part of the solute-solvent London dispersion interaction (which increases with solute size) (*6*). R_2 is the "excess molar refraction" of a solute, i.e., the measured molar refraction minus that of a hypothetical alkane of identical volume (*2*). The rR_2 term is intended to capture solute-solvent interactions which involve an induced dipole (polarization) on the solute beyond what is accounted for by the mV_x term. R_2 and V_x are therefore independently derived and clearly physically interpretable, unlike the remaining solute parameters in equation 4-1. $\Sigma\alpha_2^H$ and $\Sigma\beta_2^H$ refer to the total hydrogen bond donating and hydrogen bond accepting capacities of the solute, respectively. Finally, the polarity/polarizability parameter, π_2^H, is believed to reflect the interactions associated with both induced and stable polarity on the solute. Physical explanation of $\Sigma\alpha_2^H$, $\Sigma\beta_2^H$, and π_2^H relies on the assumed linear separability of the underlying processes that they are intended to represent. To the extent that this assumption is invalid, the LSER parameters must reflect some blending of processes and may nevertheless give mathematically convenient results. Hence, in order to understand what is known about the π_2^H scale, I first examine its evolution among concurrently developed LSER parameters.

4.1.2. The original development of π_2^H

The parameters $\Sigma\alpha_2^H$, $\Sigma\beta_2^H$, and π_2^H represent "updated" parameters from a previous set of solvatochromic trial descriptors, α_2^H (*7*), β_2^H (*8*), and π^* (*4*). These parameters have previously been related to solvation properties by fitting the coefficients of the equation:

$$\log P = l \log L^{16} + rR_2 + s\pi^* + a\alpha_2^H + b\beta_2^H + c \tag{4-2}$$

where L^{16} is the air-hexadecane partition coefficient (*9*). The development of these original parameters must be briefly reviewed in order to understand the basis of π_2^H. The hydrogen bonding parameters, α_2^H and β_2^H, were linear free energy scales of 1:1 hydrogen-bond complexation equilibrium constants in tetrachloromethane solvent. Conversely, the original polarity/polarizability parameter of Kamlet and coworkers, π^*, was a scale of solvent-induced spectral (frequency) shifts of the electronic transitions (p $\rightarrow \pi^*$ and $\pi \rightarrow \pi^*$) of characteristic solutes which were not believed to engage in significant hydrogen-bonding with the selected solvents (*1, 4, 10*). Although π^* was used as a solute descriptor, it actually reflected a *solvent* property: a measure of a solvent's ability to alter the spectral transition of a characteristic set of solutes. With these assumptions, s values could be calibrated as two-phase system coefficients, taking π^* to be a solute parameter. Subsequently, Abraham et al. drew on these developments by proposing a LSER of gas chromatography retention times for several hundred solutes on 75 non-hydrogen-bond donating stationary phases (so that $b = 0$) using:

$$\log V^0 = l \log L^{16} + rR_2 + s\pi^* + a\alpha_2^H + c \tag{4-3}$$

where V^0 is a retention capacity (*5*). This resulted in a set of stationary phase fitted coefficients (l, r, s, a, and c) obtained via multiple linear regression. These workers then kept the stationary phase coefficients fixed and used the same set of data to *reverse fit* the polarity/polarizability and solute hydrogen-bond donating scales, thereby producing an updated set of solute parameters, π_2^H and $\Sigma\alpha_2^H$ (the $l \log L^{16}$ and rR_2 terms were first subtracted from the dependent variable). Having determined π_2^H and $\Sigma\alpha_2^H$, values for $\Sigma\beta_2^H$ were similarly obtained, as follows. After using β_2^H as a trial descriptor to parameterize 16 water-organic solvent system LSER coefficients, these coefficients were fixed in order to isolate $\Sigma\beta_2^H$ in a reverse fit, with the contributions of other terms first subtracted out of the dependent variable (*11*).

Abraham and coworkers rationalized these reverse fit updates of the hydrogen bonding and polarity/polarizability descriptors as a way of "correcting" 1:1 complex solute parameters to reflect a more realistic set of interactions of the solute with multiple solvent molecules. It is difficult to assess what physical meaning may have been inserted into the parameters as a result of the updating procedure. In particular, the $\Sigma\alpha_2^H$ and $\Sigma\beta_2^H$ scales typically differed little, if at all, from the trial descriptors, α_2^H and β_2^H, for most solutes. Nevertheless, this paved the way for characterizing new solutes, since π_2^H, $\Sigma\alpha_2^H$, and $\Sigma\beta_2^H$ values could subsequently be fitted via reverse regression from a large set of retention data in gas or liquid chromatography systems which had established system coefficients (i.e., r, s, a, b, m, and c values) (*12*).

4.1.3. Previous correlations of the polarity parameters with other descriptors

In light of its somewhat complicated history, one may suspect that π_2^H is strongly related to both solute polarizability and solute charge density on the solvent accessible surface (hereafter referred to as "SAS"). Further, recognizing that Abraham and co-workers specifically attempted to separate the hydrogen-bonding and polarity contributions to ΔG_{solv}, π_2^H may still reflect some unclear amount of mixing or interference with the hydrogen-bonding terms. Several workers have previously attempted to correlate the polarity scales π^* and π_2^H with theoretically conceived or calculated quantites. Brink et al. observed that "local charge separation" on the SAS of a solute may more accurately describe the solute's ability to create electrostatic interactions than do its dipole or multipole moments (*13, 14*). For example, some symmetric molecules (e.g., carbon dioxide, para-dinitrobenzene) have net zero dipole moments, but they exhibit significant charge separation at the SAS which can interact with surrounding solvent molecules. Brink et al. found a limited correlation between the π^* scale and a calculated parameter, Π, defined as the area-normalized summation of local charge separation on an operationally designated SAS of the solute. Other workers have investigated correlations between π^* or π_2^H and area-normalized summations of solute SAS charge or its square (*15*), measured and calculated dipole moments (*16*), and various related quantities including summed atomic charges, calculated HOMO-LUMO energy gaps, and topological indices (*17, 18*). Still other investigators have proposed mixing computed and empirically derived solute descriptors to generate revised LSERs (*19, 20*). Most successfully, group contribution approaches have also been applied to the problem of estimating π_2^H. Platts et al. developed a comprehensive group contribution method consisting of 81 functional fragments for predicting π_2^H ($r^2 = 0.92$ and $\sigma = 0.16$ for the regression set) (*21*). Abraham additionally showed that for several families of aromatic compounds, π_2^H can be accurately estimated from a regression including both solute dipole moment and empirically fitted group-contribution parameters (*12*). Such group-additivity estimation methods are practical and useful, as long as the group values are available. However a more fundamentally based approach might allow estimates of π_2^H in cases where functional group values exhibit poor additivity or have not yet been defined. It should finally be noted that Weckwerth et al. initiated the development of a separate LSER system based on reference solutes, which they contended may provide "purer" solute descriptors (*22*). From the accomplishments of these groups, I concluded that the Debye and Keesom-type contributions to π_2^H may be related to electrostatic properties at the solute SAS. Since π_2^H exhibits reasonable functional additivity, I hypothesized that a correlation which relies on SAS area-aggregated, rather than area-normalized, charge descriptors was appropriate. Additionally, because π_2^H is partly rooted in dispersion interactions (*4, 12*), I considered it useful to separate out a partial dependence of π_2^H on solute polarizability (measured directly using the index of refraction).

Based on these considerations, the goals of this work were: (1) to relate π_2^H with a computed solute electrostatic component and the solute excess polarizability scale, R_2; and (2) to attempt to rationalize the contributions of these more fundamental quantities to the $s\pi_2^H$ solvation free energy term of a LSER. I therefore present a methodology for calculating an appropriate solute electrostatic descriptor. I then discuss the extent to which the computed electrostatic term and R_2 can explain π_2^H for a diverse set of solutes. Finally, I compare these results to some other approaches which have been previously suggested for calculating π_2^H or π^*.

4.2. Method

4.2.1. Development of a computed electrostatic descriptor

Any plausible computational estimate of π_2^H should, at the very least, incorporate the simplifying assumptions inherent in the LSER formulation. Most notably, LSERs presuppose that the physical processes governing the solute and the solvent are mathematically separable. For example, a π_2^H value is considered a constant property of the solute, regardless of the solvent. All information about the solvent relating to π_2^H is reflected by the LSER multiplying coefficient, s. This approximation implies that a computed analogue of π_2^H should be linear with respect to any solvent properties that may enter the calculation; in fact it would be difficult to justify otherwise. Fortunately, classical electrostatic theory offers such an approach.

In classical electrostatics, the interaction energy, ΔU_e, between a solute, 1, and the surrounding solvent medium, s, can be described as (*23*):

$$\Delta U_e = \frac{1}{2}\int_{allspace} \phi_1(\mathbf{r})\rho_s(\mathbf{r})d\tau \tag{4-4}$$

where $\rho_s(\mathbf{r})$ corresponds to the charge density of the solvent and $\phi_1(\mathbf{r})$ is the solute electrostatic potential at any point, $\mathbf{r}$, in infinitesimal volume, $d\tau$. In order to facilitate the evaluation of ΔU_e, it is frequently assumed that a meaningful solute-solvent boundary may be reliably defined (although this is a disputed concept (*24*)). In this case, equation 4-4 is recast as an integral over so-called "virtual" surface charges at the SAS of the solute, $\sigma_s(\mathbf{r})$, which reflect the solvent's electrostatic interaction with the solute (*25*):

$$\Delta U_e = \frac{1}{2}\oint_{SAS} \phi_1(\mathbf{r})\sigma_s(\mathbf{r})dA \tag{4-5}$$

Solvent charge density at the solute-solvent boundary (the SAS) may be approximated by treating the solvent as a continuous linear dielectric medium (*25*). In other words, $\sigma_s(\mathbf{r})$ "mirrors" the electrostatic field of the solute with a proportional field damped by the solvent dielectric constant, ϵ_s. Consequently, the solvent charge at the solute surface may be related to the normal component of the solute electrostatic field, $E_{1,n}(\mathbf{r})$, at the solute-solvent boundary as (*23*):

$$\frac{\sigma_s(\mathbf{r})}{\epsilon_o} = \left(\frac{1}{\epsilon_s} - 1\right) E_{1,n}(\mathbf{r}) \tag{4-6}$$

where ϵ_s is a constant scalar if the dielectric medium is assumed homogeneous and isotropic. The validity of linear solvent response for nonionic organic solvents has been evaluated in various Monte Carlo and molecular dynamics simulations (*26–29*). These studies suggest that the linear solvent response approximation is most accurate when solute polarity exceeds that of the solvent. Linear solvent response may nevertheless provide useful estimates of electrostatic effects for both semipolar and polar solutes, since larger relative errors in ΔU_e are tolerable in cases where electrostatic effects make a smaller contribution to the total solvation partial free energy. The linear response approximation additionally implies that the electrostatic component of the partial free energy of solvation is given by the solute-solvent interaction energy (*26*). Combining equations 4-5 and 4-6,

$$\Delta U_e \equiv \frac{1}{2}\left(\frac{1}{\epsilon_s} - 1\right) \epsilon_o \oint_{SAS} \phi_1(\mathbf{r}) E_{1,n}(\mathbf{r}) \; dA \;\; [\text{kcal/mol}] \tag{4-7}$$

ΔU_e can be expressed directly in terms of the solute properties, $\phi_1(\mathbf{r})$ and $E_{1,n}(\mathbf{r})$. This computed electrostatic scale therefore suits the separability assumption of the corresponding LSER term, π_2^H. The computational realization of ΔU_e could be carried out in a few different ways, depending on what further assumptions are used (as described in the subsequent section).

The π_2^H parameter is believed to include both electrostatic effects and some solute polarization information; therefore I proposed that a linear combination of the measured solute excess polarizability and the computed electrostatic energy may explain π_2^H, to first order:

$$\pi_{2,fit}^H = \lambda_p R_2 + \lambda_e \Delta U_e \tag{4-8}$$

where λ_p and λ_e are characteristic coefficients, optimized via multiple linear regression using literature (measured) R_2 and π_2^H values and computed ΔU_e values to produce $\pi_{2,fit}^H$ values for 90 organic solutes.

4.2.2. Molecular orbital computations of ΔU_e

All solute geometry optimizations and electrostatic energy computations were performed using Gaussian98 (*30*), with "tight" SCF (self-consistent field) convergence criteria for the wavefunction computation. 90 solutes were optimized to an energetically minimized nuclear geometry using both (1) the hybrid HF-DFT method B3LYP (*31*) with the 6-31G(d,p) basis set and (2) the HF method using the smaller MIDI! basis set (*32*). The popular B3LYP method was chosen because it has been found to predict geometries, energetics, and electrostatic interactions more accurately than some other DFT and *ab initio* methods

(*33, 34*), and the MIDI! basis set was employed because it has been optimized specifically for charge-property calculations. Since the role of π_2^H is most prominent in the presence of strongly polar solvents, all geometry optimizations utilized a dielectric continuum field corresponding to aqueous solution ($\varepsilon_s = 78.3$) using the Polarizable Continuum Model (PCM (*35, 36*)). At B3LYP/6-31G(d,p) or HF/MIDI! optimized solute geometries, several single point electrostatic energy calculations were performed, using SAS virtual solvent charges, $\sigma_s(\mathbf{r})$, of either the Polarizable Continuum Model or the Self-Consistent Isodensity Polarizable Continuum Model (again assuming $\varepsilon_s = 78.3$). In single point computations, equation 4-7 was integrated over either a fixed-atomic radii surface (ΔU_e^B) or a solute electron isodensity surface (ΔU_e^I), described as follows (Table 4-1).

ΔU_e^B [kcal/mol] values were computed in the presence of a PCM-computed solvent charges, using both the B3LYP/6-311G(2df,2p) and HF/MIDI! methods. The Polarizable Continuum Model (PCM) (*35, 36*) is a widely used solvation model which approximates $\sigma_s(\mathbf{r})$ as a set of discrete charges at the SAS_B, where SAS_B is defined as the outer surface carved by Bondi atomic radii (*37*) multiplied by 1.2. The SAS_B charges act to stabilize and distend the solute wavefunction. The PCM also incorporates the polarization response of SAS_B charges to each other (i.e., "self-polarization" of the surface charges) and includes $E_1(\mathbf{r})$ corrections for SAS_B curvature. Using these approximations, the PCM self-consistently calculates SAS_B charges, polarizes the solute electronic wavefunction in response, and (optionally) relaxes the solute geometry (*36*) in the dielectric bath. Notably, the PCM and most related models do not include corrections for solute-solvent hydrogen-bonding effects. Since continuum solvation models were designed for prediction of solvation energies in a variety of systems, neglect of hydrogen-bonding has usually been considered a shortcoming of the approach (*38*). However, predictions of the π_2^H parameter may be well suited by the continuum model approximations, since π_2^H was intentionally designed to be independent of hydrogen-bonding effects.

It was desirable to evaluate the electrostatic energy at an electron isodensity SAS (thus denoted ΔU_e^I), to allow comparisons with electrostatics computed using the fixed atomic radii surface (SAS_B). ΔU_e^I was calculated in the presence of SCIPCM or IPCM solvent charges, using the HF/MIDI! method as follows. The solute wavefunction was first relaxed in a dielectric bath using the Self-Consistent Isodensity Polarizable Continuum Model (SCIPCM (*39*)). For cases in which the SCIPCM did not converge (10 out of 90 solutes), the Isodensity Polarizable Continuum Model (IPCM) (*39*) was applied. The SCIPCM and IPCM formulations of solute-dielectric field interactions are similar to that of the PCM as described previously; however the SCIPCM and IPCM place virtual solvent charges at a solute electron isodensity surface of 0.0004 e^-/bohr^3 (SAS_F). The SAS_F may be more favorable than the conventional SAS_B, because an isodensity surface reflects the extent of solvent access to the solute expected from electron cloud repulsions between molecules. Since the location of the SAS_F is itself a function of calculated SAS_F charges (unlike the fixed SAS_B), the SAS_F charges are incorporated into the solute Hamiltonian potential expression and the wavefunction is iteratively calculated until the charge updates converge. The IPCM computes SAS_F charges in between SCF convergence cycles, whereas the

Table 4-1. Methods used to compute the solute electrostatic descriptor (ΔU_e in kcal/mol; ΣV_s^2 in kcal Å/mol)

electrostatic descriptor	solvent charges[a]	single point method	SAS used for equation 4-7
ΔU_e^B	PCM	B3LYP/6-311G(2df,2p)	1.2 Bondi radii
ΔU_e^B	PCM	HF/MIDI!	1.2 Bondi radii
ΔU_e^I	SCIPCM	HF/MIDI!	0.0004 e^-/bohr^3
ΔU_e^I	PCM	B3LYP/6-311G(2df,2p)	0.0004 e^-/bohr^3
ΔU_e^I	PCM	HF/MIDI!	0.0004 e^-/bohr^3
ΔU_e^I	PCM	HF/MIDI!	0.0001 e^-/bohr^3
ΣV_s^2	SCIPCM	HF/MIDI!	0.0004 e^-/bohr^3
ΣV_s^2	PCM	B3LYP/6-311G(2df,2p)	0.0004 e^-/bohr^3
ΣV_s^2	PCM	HF/MIDI!	0.0004 e^-/bohr^3
ΣV_s^2	PCM	HF/MIDI!	0.0001 e^-/bohr^3

[a] Using $\varepsilon_s = 78.3$.

SCIPCM embeds SAS$_F$ charge computations directly into the SCF procedure. Although the SCIPCM or IPCM may provide a more realistic SAS than the PCM, there are practical disadvantages to their use. The SCIPCM and IPCM are more computationally expensive than the PCM (*35*) and they are less numerically stable than the PCM (*24*). The resulting solute wavefunction was used to generate a fine grid discretized electron density (output with the Gaussian98 "cube" keyword). An isodensity 0.0004 e^-/bohr^3 surface (SAS$_I$) was numerically interpolated from the calculated grid of the electron density. The resulting surface had uniform 0.04 Å^2 resolution, corresponding to about 1700 surface elements for a single molecule of water. Subsequently, solute $E_1(\mathbf{r})$ and $\phi_1(\mathbf{r})$ values were found at the SAS$_I$ element centers. The $E_1(\mathbf{r})$ and $\phi_1(\mathbf{r})$ values reflected only the solute wavefunction. In other words, the computed field and potential values at the SAS$_I$ elements did not include the field and potential contribution of the SCIPCM or IPCM charges, although the solute wavefunction had been optimized in the SCIPCM or IPCM dielectric environment. The outward normal vector at each SAS$_I$ element was approximated using the locations of several adjacent SAS$_I$ element centers. Computed SAS$_I$ normal vectors were then combined with $E_1(\mathbf{r})$ values to arrive at $E_{1,n}(\mathbf{r})$ estimates. ΔU_e^I could subsequently be numerically integrated over the SAS$_I$ of each solute. Since the SCIPCM and IPCM were computationally expensive, it was desirable to also apply the PCM prior to integration of ΔU_e^I, in a new set of calculations using both the B3LYP/6-311G(2df,2p) and HF/MIDI! methods (Table 4-1). In other words, since application of the dielectric continuum was a separate step from integration of ΔU_e^I, it was possible to use the PCM to optimize the solute wavefunction in an aqueous dielectric (ε_s = 78.3) and subsequently integrate equation 4-7 at the SAS$_I$. Finally, in an additional set of calculations, ΔU_e^I was integrated over a fine-grid 0.0001 e^-/bohr^3 SAS$_I$ in order to test the sensitivity of results to the isodensity surface location. HF/MIDI! was used for most of the sets of ΔU_e^I calculations because it was computationally expensive to use the SCIPCM or IPCM together with the B3LYP/6-311G(2df,2p) method for the largest solutes. Since SAS$_I$ curvature corrections were not made for $E_{1,n}(\mathbf{r})$ estimates, the numerical integration of ΔU_e^I was additionally evaluated using several levels of SAS$_I$ resolution with water as a test solute.

As outlined above, computation of $E_{1,n}(\mathbf{r})$ required numerical evaluation of the normal electrostatic field component at the solute SAS$_I$ for a large number of points. This procedure added complexity to the method and may be susceptible to errors; hence it was desirable to generate a more tractable form of the integral in equation 4-7. As a simplification, integration over the normal solute field was thus assumed proportional to integration over the solute potential at the solute SAS$_I$:

$$\oint_{SAS_I} \phi_1(\mathbf{r}) E_{1,n}(\mathbf{r})\, dA = -\oint_{SAS_I} \phi_1(\mathbf{r}) \left[\frac{d\phi_1(\mathbf{r})}{dn(\mathbf{r})}\right] dA \propto -\oint_{SAS_I} \phi_1^2(\mathbf{r})\, dA \qquad (4\text{-}9)$$

where $E_{1,n}(\mathbf{r})$ is given by $d\phi_1(\mathbf{r})/dn(\mathbf{r})$ (the gradient of the solute electrostatic potential along the SAS normal vector, $n(\mathbf{r})$), and $d\phi_1(\mathbf{r})/dn(\mathbf{r})$ is assumed proportional to $\phi_1(\mathbf{r})$. Equation 4-9 was evaluated by comparison of calculated $E_{1,n}(\mathbf{r})$ and $\phi_1(\mathbf{r})$ values at 9000 randomly selected SAS$_I$ points on the set of studied solutes (100 SAS$_I$ points on each solute). The validity of equation 4-9 is further discussed in the Results section. Equation 4-9 led to a new electrostatic scale proportional to ΔU_e^I which could now be defined (from equation 4-7) as:

$$\Sigma V_s^2 \equiv -\frac{1}{2}\left(\frac{1}{\epsilon_s} - 1\right) \epsilon_o \oint_{SAS_I} \phi_1^2(\mathbf{r})\, dA \ \ [\text{kcal Å/mol}] \qquad (4\text{-}10)$$

In a new set of calculations, the validity of equation 4-9 was additionally judged by the observed correlation of ΣV_s^2 with ΔU_e^I for 90 solutes. Finally, regressions of equation 4-8 were also evaluated using ΣV_s^2 as a substitute for ΔU_e (Table 4-1).

It is important to note that computed ΔU_e values did not include the computed change in solute energy caused by polarization of the solute wavefunction in the dielectric field. It was considered advantageous to exclude PCM or SCIPCM/IPCM-induced solute polarization energies and independently fit the solute polarizability contribution (R_2) in equation 4-8 for at least two reasons: first, the PCM and SCIPCM/IPCM-calculated solute polarization energies did not correlate well with the measured polarizability scale. Second, the precise origin and magnitude of the solute polarizability contribution to π_2^H is unclear, since the LSER formulation already incorporates a R_2 component in empirical fits of solvation data (as discussed in the *Introduction*). Investigators wishing to reproduce the computational method described herein should note that, by default, the PCM-computed ΔG_{solv} includes an estimated solvent cavitation energy term, an estimated solute-solvent dispersion-repulsion interaction energy term, a dielectric field-induced solute polarization energy term, and finally ΔU_e^B. These first three terms were not included in the analysis I conducted. Similarly, the default SCIPCM and IPCM-computed ΔG_{solv} output implicitly includes a solute polarization energy associated with the dielectric charges. Consequently, I decided to use the calculated electronic population to define an electron isodensity surface for explicit integration of equation 4-7 (as described previously), in order to provide a more direct estimate of ΔU_e^I.

4.2.3. Gas phase dipole moment computations

As a validation of the computational accuracy of the molecular orbital computation methods for charge distribution properties, gas phase dipole moments were computed and compared to literature data. The B3LYP/6-31G(d,p) and HF/MIDI! approaches were used to optimize the geometry of 45 solutes from the set of 90 considered for this study, *in vacuo*. The B3LYP/6-311G(2df,2p) and HF/MIDI! methods were then used to compute the dipole moments of these solutes (also *in vacuo*) for comparison with measured gas phase dipole moments (*40*).

4.2.4. Selection of π_2^H data

I selected published π_2^H values (*41, 42*) which represented a range of solute types. N-alkanes were included as reference compounds, since Abraham set $\pi_{2,alkane}^H = 0$. A range of small to moderately sized aliphatic compounds (1 to 11 heavy atoms), often containing multiple moieties, composed the first subset of the list. A few homologous series were included to evaluate the extension of nonpolar chains. Aromatic compounds composed the second subset (ranging from 4 to 16 heavy atoms), some of which contained N, S, or O as ring members. Multiple moieties and some flexible chain substituents characterized many of the aromatic compounds. In both aliphatic and aromatic sets, a range of semipolar (e.g., olefin, amino, halogen) to highly polar (e.g., sulfone, sulfoxide, amide, nitro) groups were tested. Additionally, in some cases single or multiple electron-withdrawing groups (such as halogens) were proximate to polar groups, inducing especially electron-deficient protons (e.g., 3-bromophenol and 2,2,2-trifluoroethanol). This range of compounds was considered a robust test of model applicability to small and moderately sized organic compounds containing C, H, N, O, S, F, Cl, and Br (only one compound containing P was included in the set).

4.3. Results and discussion

4.3.1. Computation of gas phase dipole moments

It was desirable to evaluate the reliability of the HF/MIDI! and B3LYP/6-311G(2df,2p) methods against an independently measurable molecular charge distribution property. Reported gas phase dipole moments (*43*) compared favorably to those calculated *in vacuo* using both methods, finding $r^2 = 0.96$, $\sigma_\mu = 0.25$ for HF/MIDI! computations (Figure 4-1) and $r^2 = 0.975$, $\sigma_\mu = 0.19$ for B3LYP/6-311G(2df,2p) computations (not shown) with 45 compounds among the set of 90 considered in this work. It was reasonable to assume that fits of equation 4-8 would reflect this limitation in accuracy; that is, I did not expect to

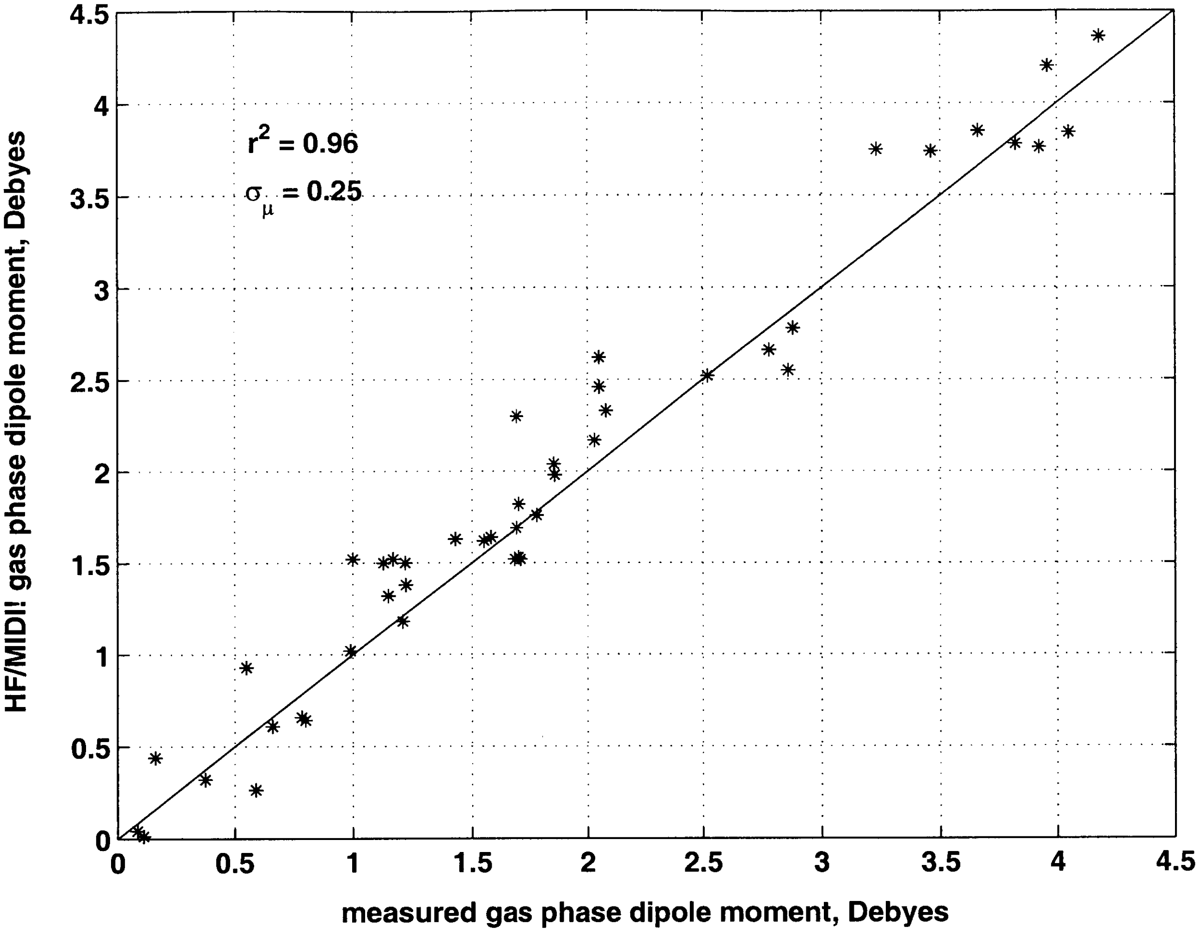

Figure 4-1. HF/MIDI! computed *in vacuo* dipole moments for 45 compounds

generate a π_2^H model of significantly better predictive quality than that found for gas phase dipole moment calculations.

4.3.2. π_2^H regressions

Regressions using 90 solutes (Table 4-2) showed that the combination of the computed solute electrostatics term and measured solute polarizability scale fit π_2^H with considerable accuracy (Table 4-3). HF/MIDI! electrostatic energies computed at the 0.0004 e^-/bohr^3 SAS_I in the presence of a SCIPCM/IPCM dielectric produced the best correlation with π_2^H values (Table 4-2):

$$\pi_{2,fit}^H = 0.49R_2 - 0.116\Delta U_e^I \qquad (4\text{-}11)$$

$$r^2 = 0.95,\ \sigma_\pi = 0.12$$

Table 4-2. Regression results for equation 4-11; comparison of calculated and literature π_2^H values (ΔU_e^I in [kcal/mol])

			calc.	meas.
Solute	R_2	$-\Delta U_e^I$	$\pi_{2,fit}^H$	π_2^H
propane	0.000	0.17	0.02	0.00
pentane	0.000	0.22	0.03	0.00
cyclohexane	0.305	0.20	0.17	0.10
1-hexene	0.078	0.77	0.13	0.08
propyne	0.183	2.05	0.33	0.25
1-butyne	0.178	2.02	0.32	0.23
fluoromethane	0.066	3.00	0.38	0.35
1-fluorobutane	0.017	2.74	0.33	0.35
1-fluoropentane	0.002	2.79	0.33	0.35
tetrafluoromethane	-0.280	0.71	-0.05	-0.20
hexafluorosulfide	-0.600	0.29	-0.26	-0.20
chloroethane	0.227	2.54	0.41	0.40
1-chlorobutane	0.210	2.53	0.40	0.40
1-chlorooctane	0.191	2.43	0.38	0.40
carbon tetrachloride	0.458	0.45	0.28	0.38
trichloroethane	0.499	3.77	0.68	0.68
hexachloroethane	0.680	0.61	0.40	0.22
bromoethane	0.366	2.51	0.47	0.40
1-bromobutane	0.360	2.50	0.47	0.40
1-bromooctane	0.339	2.69	0.48	0.40
dibromomethane	0.714	2.87	0.68	0.67
tribromomethane	0.974	2.27	0.74	0.68
diethylether	0.041	1.78	0.23	0.25
dipropylether	0.008	1.87	0.22	0.25
tetrahydrofuran	0.289	2.44	0.42	0.52
carbon monoxide	0.000	0.62	0.07	0.00
carbon dioxide	0.150	2.80	0.40	0.42
carbon disulfide	0.877	0.24	0.46	0.21
dioxygen	0.000	0.06	0.01	0.00
nitrous oxide	0.068	1.89	0.25	0.35
ethylamine	0.236	2.76	0.44	0.35
propylamine	0.225	3.03	0.46	0.35
butylamine	0.224	3.05	0.46	0.35
water	0.000	6.23	0.72	0.45
methanol	0.278	4.24	0.63	0.44
ethanol	0.246	3.90	0.57	0.42
1-propanol	0.236	3.91	0.57	0.42

isopropanol	0.212	3.84	0.55	0.36
1-decanol	0.191	4.15	0.58	0.42
acetone	0.179	5.07	0.68	0.70
butanone	0.166	4.50	0.60	0.70
propanal	0.196	4.23	0.59	0.65
acetonitrile	0.237	6.37	0.86	0.90
propionitrile	0.162	6.01	0.78	0.90
nitromethane	0.313	7.39	1.01	0.95
nitroethane	0.270	6.86	0.93	0.95
nitropropane	0.242	6.62	0.89	0.95
ethylacetate	0.106	4.43	0.57	0.62
acetic acid	0.265	6.57	0.89	0.65
2,2,2-trifluoroethanol	0.015	6.60	0.82	0.60
enflurane	-0.230	4.91	0.46	0.40
isoflurane	-0.240	5.86	0.56	0.50
trimethylphosphate	0.113	10.69	1.30	1.10
propionamide	0.440	8.53	1.21	1.30
N-methylformamide	0.405	9.09	1.25	1.30
dimethylsulfone	0.590	13.09	1.81	1.70
N,N-dimethylacetamide	0.363	6.91	0.98	1.33
N,N-dimethylformamide	0.367	7.47	1.05	1.31
dimethylsulfoxide	0.522	10.16	1.44	1.74
benzene	0.610	1.60	0.48	0.52
toluene	0.601	1.55	0.47	0.52
ethylbenzene	0.613	1.57	0.48	0.51
naphthalene	1.340	2.37	0.93	0.92
phenanthrene	2.055	3.15	1.37	1.29
pyrene	2.808	3.43	1.77	1.71
chlorobenzene	0.718	2.46	0.64	0.65
1,2,4-trichlorobenzene	0.980	2.98	0.82	0.81
1,2-dibromobenzene	1.190	3.02	0.93	0.96
aniline	0.955	4.24	0.96	0.96
N-methylaniline	0.948	3.38	0.86	0.90
N,N-dimethylaniline	0.957	2.60	0.77	0.84
phenol	0.805	5.36	1.02	0.89
m-cresol	0.820	5.15	1.00	0.87
benzylalcohol	0.803	4.54	0.92	0.87
benzaldehyde	0.820	5.18	1.00	1.00
benzonitrile	0.742	6.21	1.08	1.11
thiophene	0.687	2.03	0.57	0.56
benzothiophene	1.323	2.70	0.96	0.88
thiazole	0.800	4.09	0.87	0.80
pyrazole	0.620	6.57	1.07	1.00

benzophenone	1.447	5.25	1.32	1.50
4-cyanophenol	0.940	10.61	1.69	1.55
diethylphthalate	0.729	7.72	1.25	1.40
benzotrifluoride	0.225	3.00	0.46	0.48
3-bromophenol	1.060	6.17	1.24	1.15
benzamide	0.990	8.75	1.50	1.50
benzenesulfonamide	1.130	6.68	1.33	1.55
methylphenylsulfone	1.080	11.48	1.86	1.85
diphenylsulfone	1.570	10.46	1.98	2.15
methylphenylsulfoxide	1.080	9.27	1.61	1.85

Using the HF/MIDI! method, $\pi^H_{2,fit}$ was relatively insensitive to changes in the dielectric field method (PCM versus SCIPCM/IPCM) or location of the SAS$_I$ (0.0004 versus 0.0001 e^-/bohr^3). This indicated that ΔU^I_e is probably a robust and physically meaningful parameter for π^H_2. By comparison, $\pi^H_{2,fit}$ values calculated from electrostatic energies at the 120% Bondi radii solute SAS (ΔU^B_e) showed considerably weaker agreement with π^H_2 ($r^2 < 0.80$ and $\sigma_\pi > 0.20$; Table 4-3). This suggested that an SAS based on electron isodensity, rather than the 120% Bondi radii SAS, is an appropriate physical surface for π^H_2.

In all regressions of equation 4-8 (Table 4-3), the measured excess polarizability and computed electrostatics term contributed about 1/3 and 2/3 of the $\pi^H_{2,fit}$ scale, respectively, showing that both of these terms are important components of π^H_2. This indicated that while stable charge density at the solute surface generally dominates the π^H_2 term, solute polarizability also contributes significantly. These results lend substantial credibility to the contention that Abraham and coworkers have indeed isolated a LSER term which quantitatively reflects mainly polarity and polarizability character of the solute.

A data-withholding test of each variant of equation 4-8 was additionally conducted in order to evaluate its robustness for novel compounds not included in the regression set, as follows. In a new set of regressions, each $\pi^H_{2,pred}$ values were calculated based on fitted coefficients derived from the remaining 89 solutes. This generated π^H_2 estimates which were independent of the regression procedures. All regressions and parameter statistics were calculated using singular value decomposition (*44*). Data-withholding tests suggested that the expected uncertainty of $\pi^H_{2,pred}$ values for solutes outside of the regression sets were similar to regression statistics (Table 4-3).

Regression outliers for equation 4-8 using either of the computed ΔU^B_e or ΔU^I_e terms showed systematic bias. The most egregious π^H_2 overestimates were generally strong hydrogen-bond donors (e.g., water, acetic acid, 2,2,2-trifluoroethanol). Conversely, underestimated π^H_2 outliers were consistently composed of non-acidic strong hydrogen-bond acceptors with highly electronegative moieties (e.g., N,N-dimethylformamide, N,N-dimethylacetamide, dimethylsulfoxide). This pattern of model prediction error could result from hydrogen-bonding interference during the development of π^H_2 parameter values from data. As discussed in the *Introduction*, Abraham et al.'s development of π^H_2 involved

Table 4-3. Regression statistics for equation 4-8 using 90 solutes (ΔU_e in kcal/mol; ΣV_s^2 in kcal Å/mol)

				equation 4-8 regression		regression		data-withholding	
	dielectric		SAS used for	best fit coefficients		statistics		test statistics	
ΔU_e	model[a]	method[b]	electrostatics[c]	$\lambda_p \pm \sigma_\lambda$ (weight)	$\lambda_e \pm \sigma_\lambda$ (weight)	r^2	σ_π	r^2	σ_π
ΔU_e^B	PCM	B3LYP	1.2 Bondi rad.	0.45 ± 0.22 (33%)	-0.099 ± 0.039 (67%)	0.72	0.27	0.71	0.28
ΔU_e^B	PCM	MIDI!	1.2 Bondi rad.	0.45 ± 0.22 (31%)	-0.113 ± 0.042 (69%)	0.79	0.23	0.77	0.24
ΔU_e^I	SCIPCM	MIDI!	0.0004 a.u.	0.49 ± 0.20 (33%)	-0.116 ± 0.039 (67%)	0.95	0.12	0.94	0.12
ΔU_e^I	PCM	B3LYP	0.0004 a.u.	0.45 ± 0.21 (31%)	-0.138 ± 0.046 (69%)	0.93	0.13	0.92	0.14
ΔU_e^I	PCM	MIDI!	0.0004 a.u.	0.49 ± 0.20 (33%)	-0.118 ± 0.040 (67%)	0.93	0.13	0.93	0.14
ΔU_e^I	PCM	MIDI!	0.0001 a.u.	0.50 ± 0.20 (34%)	-0.170 ± 0.056 (66%)	0.94	0.12	0.94	0.12
ΣV_s^2	SCIPCM	MIDI!	0.0004 a.u.	0.47 ± 0.20 (32%)	-0.092 ± 0.030 (68%)	0.96	0.10	0.95	0.11
ΣV_s^2	PCM	B3LYP	0.0004 a.u.	0.44 ± 0.21 (31%)	-0.098 ± 0.033 (69%)	0.93	0.14	0.92	0.15
ΣV_s^2	PCM	MIDI!	0.0004 a.u.	0.46 ± 0.20 (31%)	-0.095 ± 0.031 (69%)	0.95	0.11	0.94	0.12
ΣV_s^2	PCM	MIDI!	0.0001 a.u.	0.48 ± 0.20 (33%)	-0.116 ± 0.038 (67%)	0.95	0.11	0.95	0.11

[a] Using $\varepsilon_s = 78.3$. [b] Indicates either a B3LYP/6-311G(2df,2p) or HF/MIDI! single point wavefunction computation.

[c] 1 a.u. = 1 e^-/bohr^3 for an isodensity SAS.

reverse fits which simultaneously updated both $\Sigma\alpha_2^H$ and π_2^H values from trial descriptors. It is difficult to ascertain that these earlier investigations successfully removed all of the Debye and Keesom contributions to $\Sigma\alpha_2^H$. Or, particularly for highly polar solutes, blending of some of the $\Sigma\beta_2^H$ character into π_2^H values may have occurred, even assuming that these contributions to the solvation free energy are linearly separable. Additionally, I suspected that failure of the computational methods to evaluate properly surface potential on some highly charged moieties might explain some bias and error in π_2^H regressions. However, a comparison of HF/MIDI! calculated dipole moment residuals with $\pi_{2,fit}^H$ residuals from equation 4-11 showed no correlation at all ($r^2 = 0.03$). Additionally, the accuracy of gas phase dipole moment computations did not correlate with solute polarity (Figure 4-1). Consequently, I concluded that the computational methods were not responsible for the observed $\pi_{2,fit}^H$ error bias towards highly charged or hydrogen-bonding moieties.

The best π_2^H regression overall was found with the HF/MIDI! computed ΣV_s^2 electrostatic descriptor:

$$\pi_{2,fit}^H = 0.47R_2 - 0.092\Sigma V_s^2 \qquad (4\text{-}12)$$

$$r^2 = 0.96,\ \sigma_\pi = 0.10$$

employing the SCIPCM/IPCM at the 0.0004 e^-/bohr^3 SAS$_I$ (Figure 4-2). In fact, substitution of the computed $E_{1,n}(\mathbf{r})$ by the $\phi_1(\mathbf{r})$ in SAS$_I$ integrals (equation 4-9) produced consistently improved correlations between the electrostatic descriptor and π_2^H (using HF/MIDI!, Table 4-3). This surprising result could be explained as a cancelling of errors between equations 4-9 and 4-11. Using the HF/MIDI! method with the SCIPCM/IPCM at the 0.0004 e^-/bohr^3 SAS$_I$, it was found that:

$$\Sigma V_{s,fit}^2 = (1.29\ \AA)\Delta U_e^I\ \ [\text{kcal Å/mol}] \qquad (4\text{-}13)$$

$$r^2 = 0.976,\ \sigma = 0.50\ \text{kcal}\ \AA/\text{mol}$$

(Figure 4-3). Solutes which had the most overestimated ΣV_s^2 values in this correlation were strong hydrogen-bonding donors (e.g., water, acetic acid), whereas strong hydrogen-bond acceptors (e.g., diphenylsulfone) were underestimated. The outlier bias of equation 4-13 therefore mirrors the outlier bias found for equation 4-11 using the HF/MIDI! method, creating offsetting errors. This explains how ΣV_s^2 was apparently the most successful electrostatic variable for predicting π_2^H values for the set of solutes considered here.

4.3.3. On correlating solute electric field with electric potential at the SAS

It would be reasonable to conclude that since equation 4-13 produces a good correlation, it is additionally the case that $E_{1,n}(\mathbf{r})$ correlates well with $\phi_1(\mathbf{r})$ locally at the SAS$_I$. However, testing a set of 9000 points along the HF/MIDI! SCIPCM/IPCM-computed 0.0004 e^-/bohr^3 SAS$_I$ on the 90 solute set (100 points on each solute surface), it was found that:

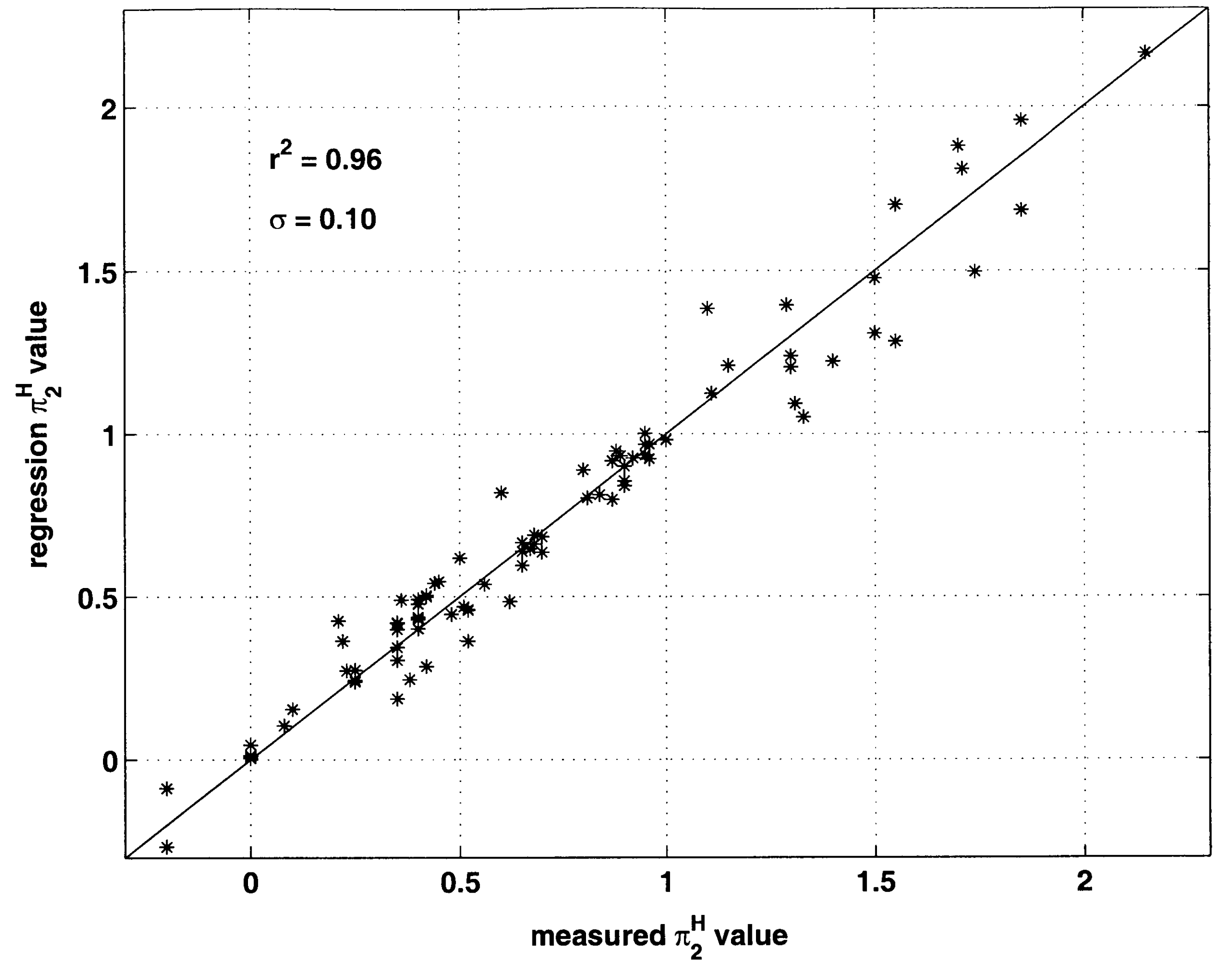

Figure 4-2. A plot of π_2^H regression results for equation 4-12

$$\phi_{1,fit}(\mathbf{r}) = (-0.94\ \mathring{A})E_{1,n}(\mathbf{r}) \quad [\text{volts}] \tag{4-14}$$
$$r^2 = 0.83$$

An apparent incongruity arises here: equation 4-13 suggests that $\phi_1(\mathbf{r})$ is a robust substitute for $E_{1,n}(\mathbf{r})$ in the SAS$_I$ electrostatic energy integral, but in equation 4-14, $E_{1,n}(\mathbf{r})$ is only a roughly accurate ($r^2 = 0.83$) explanatory variable for $\phi_1(\mathbf{r})$ at local points on the SAS$_I$. This contradiction could be explained in the following way: for a collection of SAS$_I$ points on an individual solute, the slope of the correlation between $\phi_1(\mathbf{r})$ and $E_{1,n}(\mathbf{r})$ was usually found to be close to that for equation 4-14. However, if an individual solute surface gave a $\phi_1(\mathbf{r})$ versus $E_{1,n}(\mathbf{r})$ slope that differed from equation 4-14, this deviation usually corresponded to the solute residual in equation 4-13 (a positive correlation of $r^2 = 0.43$ for the solutes having a significant electrostatic term, i.e., $\Delta U_e^I > 0.3$ kcal/mol). Consequently, I concluded that the scatter in the relationship of $\phi_1(\mathbf{r})$ versus $E_{1,n}(\mathbf{r})$ usually counterbalanced in the SAS integral for individual solutes; this scatter did not

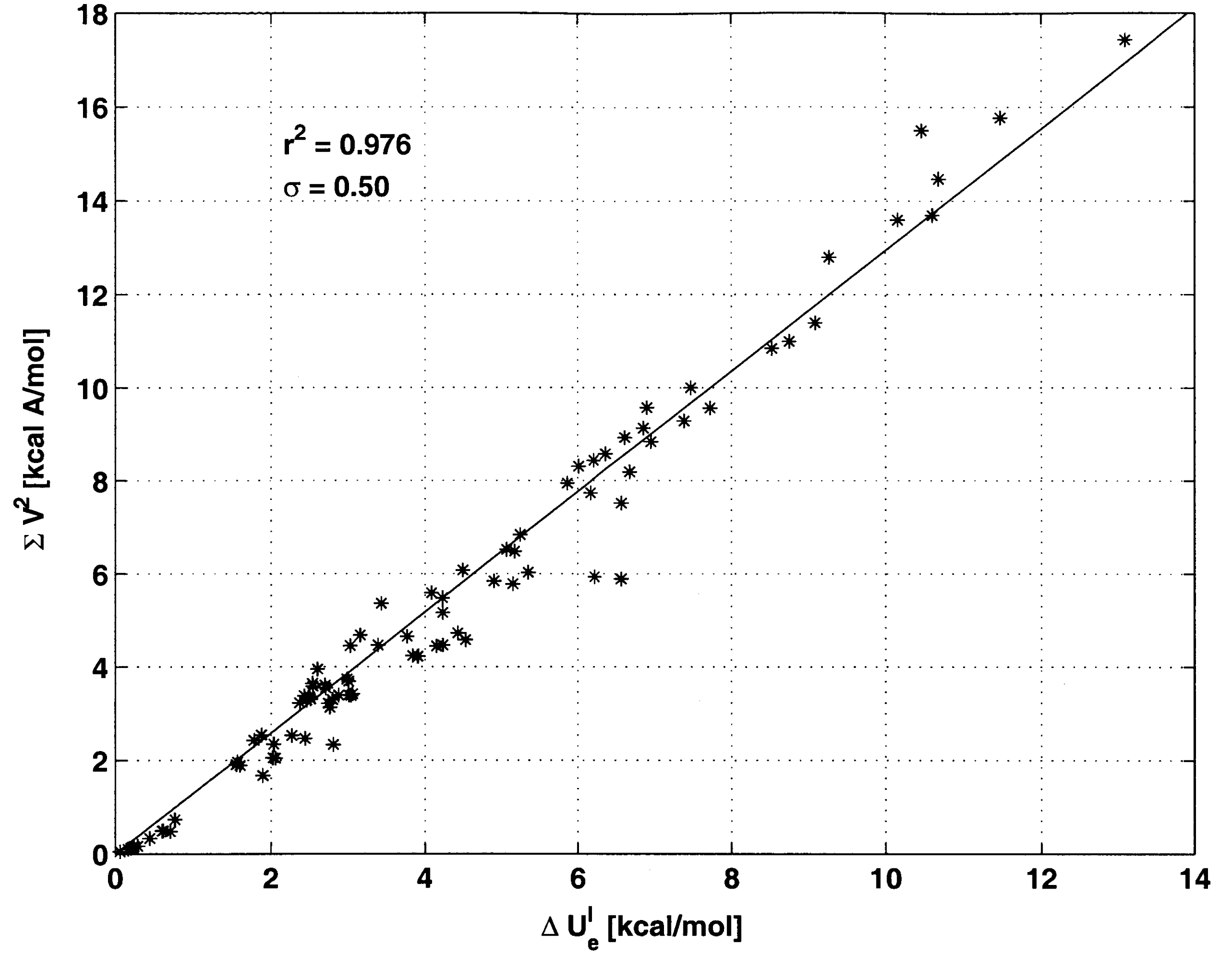

Figure 4-3. Correlation between ΔU_e^I and ΣV^2 for 90 solutes using the HF/MIDI! method with SCIPCM/IPCM at the 0.0004 e^-/bohr^3 SAS$_I$

effectively propagate to equation 4-13 except for solute cases in which the slope of $\phi_1(\mathbf{r})$ versus $E_{1,n}(\mathbf{r})$ differed significantly from equation 4-14. For completeness it is worth noting that there should be an additional term included in equation 4-14 which corresponds to the average $\phi_1(\mathbf{r})$ over the solute SAS$_I$ ($\oint \phi_1(\mathbf{r})dA$ does not sum to zero, as does $\oint E_{1,n}(\mathbf{r})dA$ by Gauss' Law). The $\bar{\phi}_1$ term was neglected since it was small and estimated to contribute to only about 3% of the deviation of equation 4-14. In summary, for a solute surface $\phi_1(\mathbf{r})$ versus $E_{1,n}(\mathbf{r})$ slope which deviates from equation 4-14, one may usually expect a $\Sigma V_{s,fit}^2$ deviation of the same sign in equation 4-13.

4.3.4. Comparison to alternative models

I compared the results of equations 4-11 and 4-12 to previous correlations that have been developed for π^* or π_2^H, using the set of compounds considered here. Lewis suggested a correlation of π^* with calculated dipole moments plus an intercept, finding a reasonable fit for 14 solutes ($r^2 = 0.91$) (*16*). Lamarche et al. suggested π_2^H fits with linear combinations of solute dipole moment, polarizability, and other quantities such as calculated atomic charges and HOMO-LUMO gap, finding correlations ranging from $r^2 = 0.76$ to $r^2 = 0.85$ for a set of 58 solutes. In this vein, I used the efficient HF/MIDI! method with a PCM dielectric field ($\varepsilon_s = 78.3$) to compute solute dipole moments, finding a correlation with π_2^H for the 90 solutes considered here:

$$\pi_{2,fit}^H = 0.59R_2 + 0.17\mu_{calc} \tag{4-15}$$

$$r^2 = 0.85,\ \sigma_\pi = 0.19$$

where the dipole moment is expressed in Debyes. A similar regression of π_2^H with μ_{calc}^2, which corresponds to the pairwise free energy of interaction between freely rotating dipoles (*45*), yielded a comparable fit. Although convenient to compute, these regressions are biased against symmetric molecules, since such solutes may exhibit substantial solvent-accessible charge separation not reflected in their dipole moments (e.g., carbon dioxide, benzene). It is worth noting that although the solute dipole moment is commonly relied upon as an indicator of solute polarity in solvent environments, higher multipoles contribute significantly to the solute-solvent interaction energy. In fact, the marginal contributions of higher multipoles may be slowly convergent, and they may still be significant well beyond the 20th term in the multipole expansion (*24*). Recognizing this deficiency of the dipole moment as an electrostatic descriptor, Brink et al. developed Π, an area-normalized summation of absolute elecrostatic surface potential (*13*):

$$\Pi \equiv \frac{1}{A}\oint_{SAS} |\phi_1(\mathbf{r}) - \bar{\phi}_1|\, dA \tag{4-16}$$

and an area-normalized summation of squared electrostatic surface potential, which they termed σ_{tot}^2 (*14*). Brink et al. found a limited correlation between π^* and Π plus a polarizability parameter plus an intercept (statistics were not given). Zou et al. recently improved this π^* correlation by including σ_{tot}^2 as an additional term ($r^2 = 0.93$ for 50 solutes) (*15*). Using HF/MIDI! calculations with the SCIPCM/IPCM dielectric field and a 0.0004 e^-/bohr^3 solute isosurface, I found the following correlation for π_2^H using the solute set presented here:

$$\pi_{2,fit}^H = 0.55R_2 + 1.02\Pi \tag{4-17}$$

$$r^2 = 0.80 \text{ and } \sigma_\pi = 0.23$$

where Π is given in volts. Adding a σ_{tot}^2 term to the equation 4-17 regression undermined the statistical interpretability of the parameters and improved the fit little ($r^2 = 0.84$). Floating a constant failed to improve the correlation. The disparity in goodness of fit found between equation 4-17 and equation 4-11 is consistent with the notion that π_2^H reflects area-aggregated, rather than area-normalized, charge density on the solute surface.

4.4. Conclusions

A method has been developed to estimate the polarity/polarizability parameter, π_2^H, for new solutes. This empirical parameter has been purported to capture solute electrostatic contributions to the solvation free energy, with minimal interference from solute-solvent hydrogen-bonding interactions. Nevertheless, π_2^H has conventionally eluded reliable correlations with more fundamental quantities. Moreover, its ambiguous physical origin has been presumed to reflect a conserved solute property over a wide range of solvent environments. Despite its complicated inception, π_2^H appears to be accurately explained by two solute properties: a polarizability term and a computed solvent accessible surface electrostatic term. This result supports Abraham et al.'s contention that solute-solvent interaction free energies are mostly separable into solute-solvent hydrogen-bonding, solvent cavitation, solute polarization, and solute-solvent electrostatic interactions, to a substantial extent. Additionally, correlations found between π_2^H and electrostatic descriptors consistently indicate that a solute electron isodensity surface is a better basis for electrostatic computations than a Bondi fixed atomic radii surface. Results here show that π_2^H is not very sensitive to the choice of isodensity surface in the 0.0001 to 0.0004 e^-/bohr^3 range, additionally corroborating the robustness of this particular type of surface. This directed the development of a model for π_2^H; however it may additionally inform the ongoing debate over what type of solvent accessible surface is most appropriate for continuum solvation free energy computations more generally.

For practical applications of π_2^H estimation, I recommend equation 4-12, which has an estimated $\pi_{2,pred}^H$ standard error of about 0.11; i.e, using the efficient HF/MIDI! method for computation of ΣV_s^2 at a 0.0004 e^-/bohr^3 SAS$_I$ in the presence of a SCIPCM or IPCM dielectric field with $\varepsilon_s = 78.3$. However in (not unusual) cases where SCIPCM or IPCM may be computationally expensive or poorly convergent, similar results may be obtained by using PCM to generate the dielectric field, followed by computation of ΣV_s^2 or ΔU_e^I at the 0.0004 or 0.0001 e^-/bohr^3 SAS$_I$. Unlike previous group contribution approaches, the model could be practically applied to any moderately small ($\leq$ 20 non-hydrogen atoms) molecule containing C, H, N, O, S, F, Cl, and Br.

The uncertainty in predicted ΔG_{solv} values propagated from error in $\pi_{2,pred}^H$ calculations depends on the magnitude of the LSER coefficient, s, in equation 4-1. The s term indicates the change in electrostatic interaction that the solute will experience in going between the two solvation environments, as defined by ΔG_{solv}. The largest documented s value is probably that for air-water partitioning, where $s = 2.55$ (*42*). In this limiting case one may therefore expect a typical log P error of $\approx s\sigma_\pi = 2.55 \times 0.11 = 0.28$, or a factor of 1.9 in the partition coefficient, as a result of the uncertainty in the π_2^H model proposed here.

HF/MIDI! and B3LYP/6-311G(2df,2p) molecular orbital computations of gas phase dipole moments compared favorably to measurement data with correlation coefficients of 0.96 and 0.975, respectively. Given this performance for charge distribution estimates of small and medium sized molecules, I do not expect significantly better results for prediction of the electrostatic variable in $\pi_{2,pred}^H$. In addition to molecular orbital model limitations, the largest source of error in $\pi_{2,pred}^H$ values is probably contamination by solute-solvent hydrogen-bonding interactions inherent in the original development of solute π_2^H values. This small amount of blending of π_2^H with other physical processes probably also reflects the extent to which the LSER assumption of linearly separable physical processes is simply inappropriate.

In future work, extension and validation of the model using a wider range of elements, such as Si, P, and I (to which the MIDI! basis set has been extended (*46*)) would contribute added insight and utility to this investigation. Additionally, development of general LSER approaches which rely on more physically transparent parameters such as computed electrostatics descriptors may offer incisive insights. Such studies could more deeply evaluate the assumptions and limitations of the LSER approximation, thereby leading to a better understanding of these highly successful but preponderantly empirical models.

4.5. Acknowledgments

This research was supported by the EPA/ONR/NCERQA Science to Achieve Results award R-82902301-0, the Alliance for Global Sustainability, the Martin Family Society of Fellows for Sustainability, and the Ralph M. Parsons Fellowship Foundation. The authors also acknowledge Drs. Bernhardt L. Trout, Bruce Tidor, Daniel Blankschtein, Elfatih A. B. Eltahir and Kenneth Beers for their insightful comments, modeling guidance, and support with molecular computations. Additional thanks goes to Donnan Steele for helpful comments on the manuscript.

4.6. References

(1) Kamlet, M. J.; Abboud, J.-L. M.; Abraham, M. H.; Taft, R. W., *Linear solvation energy relationships. 23. A comprehensive collection of the solvatochromic parameters, pi*, alpha, and beta and some methods for simplifying the generalized solvatochromic equation.* Journal of Organic Chemistry **1983**, *48*, 2877-2887.

(2) Abraham, M. H.; Poole, C. F.; Poole, S. K., *Classification of stationary phases and other materials by gas chromatography.* Journal of Chomatography A **1999**, *842*, 79-114.

(3) Goss, K.-U.; Schwarzenbach, R. P., *Linear free energy relationships used to evaluate equilibrium partitioning of organic compounds.* Environmental Science & Technology **2001**, *35*, 1-9.

(4) Kamlet, M. J.; Abboud, J. L.; Taft, R. W., *The solvatochromic comparison method. 6. The pi(*) scale of solvent polarities.* Journal of the American Chemical Society **1977**, *99*, 6027-6038.

(5) Abraham, M. H.; Whiting, G. S.; Doherty, R. M.; Shuely, W. J., *XVI. A new solute solvation parameter, pi^H_2, from gas chromatographic data.* Journal of Chromatography **1991**, *587*, 213-228.

(6) Abraham, M. H.; McGowan, J. C., *The use of characteristic volumes to measure cavity terms in reversed phase liquid chromatography.* Chromatography **1987**, *23*, 243-246.

(7) Abraham, M. H.; Grellier, P. L.; Prior, D. V.; Duce, P. P.; Morris, J. J.; Taylor, P. J., *Hydrogen bonding. Part 7. A scale of solute hydrogen-bond acidity based on log K values for complexation in tetrachloromethane.* Journal of the Chemical Society Perkins Transactions 2 **1989**, 699-711.

(8) Abraham, M. H.; Grellier, P. L.; Prior, D. V.; Morris, J. J.; Taylor, P. J., *Hydrogen bonding. Part 10. A scale of solute hydrogen-bond basicity using log K values for complexation in tetrachloromethane.* Journal of the Chemical Society Perkins Transactions 2 **1990**, 521-529.

(9) Abraham, M. H.; Whiting, G. S.; Doherty, R. M.; Shuely, W. J., *Hydrogen bonding. 13. A new method for the characterization of GLC stationary phases - the Laffort data set.* Journal of the Chemical Society Perkins Transactions 2 **1990**, 1451-1460.

(10) Taft, R. W.; Abraham, M. H.; Famini, G. R.; Doherty, R. M.; Abboud, J.-L. M.; Kamlet, M. J., *Solubility properties in polymers and biological media. 5. An analysis of the factors that influence adsorption of organic-compounds on activated carbon.* Journal of Pharmaceutical Science **1985**, *74*, 807-814.

(11) Abraham, M. H., *Hydrogen bonding. 31. Construction of a scale of solute effective or summation hydrogen-bond basicity.* Journal of Physical Organic Chemistry **1993**, *6*, 660-684.

(12) Abraham, M. H., *Hydrogen-bonding. 27. Solvation parameters for functionally-substituted aromatic-compounds and heterocyclic-compounds, from gas-liquid-chromatographic data.* Journal of Chromatography **1993**, *644*, 95-139.

(13) Brinck, T.; Murray, J. S.; Politzer, P., *Quantitative determination of the total local polarity (charge separation) in molecules.* Molecular Physics **1992**, *76*, 609-617.

(14) Murray, J. S.; Politzer, P., *Statistical analysis of the molecular surface electrostatic potential: an approach to describing noncovalent interactions in condensed phases*. Journal of Molecular Structure **1998**, *425*, 107-114.

(15) Zou, J.; Yu, Q.; Shang, Z., *Correlation between empirical solvent polarity scales and computed quantities derived from molecular surface electrostatic potentials*. Journal of the Chemical Society Perkins Transactions 2 **2001**, 1439-1443.

(16) Lewis, D. F. V., *Molecular orbital calculations on solvents and other small molecules: correlation between electronic and molecular properties, mu, alpha(mol), pi(*), and beta*. Journal of Computational Chemistry **1987**, *8*, 1084-1089.

(17) Lamarche, O.; Platts, J. A.; Hersey, A., *Theoretical prediction of the polarity/polarizability parameter pi^H_2*. Physical Chemistry Chemical Physics **2001**, *3*, 2747-2753.

(18) Svozil, D.; Sevcik, J. G. K.; Kvasnicka, V., *Neural network prediction of the solvatochromic polarity/polarizability parameter pi^H_2*. Journal of Chemical Information and Computer Science **1997**, *37*, 338-342.

(19) Cramer, C. J.; Famini, G. R.; Lowrey, A. H., *Use of calculated quantum chemical properties as surrogates for solvatochromic parameters in structure-activity relationships*. Accounts of Chemical Research **1993**, *26*, 599-605.

(20) Lowrey, A. H.; Cramer, C. J.; Urban, J. J.; Famini, G. R., *Quantum chemical descriptors for linear solvation energy relationships*. Computers in Chemistry **1995**, *19*, 209-215.

(21) Platts, J. A.; Butina, D.; Abraham, M. H.; Hersey, A., *Estimation of molecular free energy relationship descriptors using a group contribution approach*. Journal of Chemical Information and Computer Science **1999**, *39*, 835-845.

(22) Weckwerth, J. D.; Vitha, M. F.; Carr, P. W., *The development and determination of chemically distinct solute parameters for use in linear solvation energy relationships*. Fluid Phase Equilibria **2001**, *183-184*, 143-157.

(23) Wangsness, R. K. *Electromagnetic Fields*; John Wiley & Sons, Inc.: New York, NY, **1979**.

(24) Cramer, C. J.; Truhlar, D. G., *Implicit solvation models: equilibria, structure, spectra, and dynamics*. Chemical Reviews **1999**, *99*, 2161-2200.

(25) Miertus, S.; Scrocco, E.; Tomasi, J., *Electrostatic interaction of a solute with a continuum. A direct utilization of ab initio molecular potentials for the prevision of solvent effects*. Chemical Physics **1981**, *55*, 117-129.

(26) Aqvist, J.; Hansson, T., *On the validity of electrostatic linear response in polar solvents*. Journal of Physical Chemistry **1996**, *100*, 9512-9521.

(27) Milischuk, A.; Matyushov, D. V., *Dipole solvation: nonlinear effects, density reorganization, and the breakdown of the Onsager saturation limit*. Journal of Physical Chemistry A **2002**, *106*, 2146-2157.

(28) Milischuk, A.; Matyushov, D. V., *On the validity of dielectric continuum models in application to solvation in molecular solvents*. Journal of Chemical Physics **2003**, *118*, 1859-1862.

(29) Matyushov, D. V., *Dipole solvation in dielectrics*. Journal of Chemical Physics **2004**, *120*, 1375-1382.

(30) Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Zakrzewski, V. G.; Montgomery, J. A., Jr.; Stratmann, R. E.; Burant, J. C.; Dapprich, S.; Millam, J. M.; Daniels, A. D.; Kudin, K. N.; Strain, M. C.; Farkas, O.; Tomasi, J.; Barone, V.; Cossi, M.; Cammi, R.; Mennucci, B.; Pomelli, C.; Adamo, C.; Clifford, S.; Ochterski, J.; Petersson, G. A.; Ayala, P. Y.; Cui, Q.; Morokuma, K.; Malick, D. K.; Rabuck, A. D.; Raghavachari, K.; Foresman, J. B.; Cioslowski, J.; Ortiz, J. V.; Stefanov, B. B.; Liu, G.; Liashenko, A.; Piskorz, P.; Komaromi, I.; Gomperts, R.; Martin, R. L.; Fox, D. J.; Keith, T.; Al-Laham, M. A.; Peng, C. Y.; Nanayakkara, A.; Gonzalez, C.; Challacombe, M.; Gill, P. M. W.; Johnson, B.; Chen, W.; Wong, M. W.; Andres, J. L.; Gonzalez, C.; Head-Gordon, M.; Replogle, E. S.; Pople, J. A., *Gaussian 98, Revision A.6.* Gaussian, Inc., Pittsburgh PA **1998**.

(31) Becke, A. D., *Density-functional thermochemistry. III. The role of exact exchange.* Journal of Chemical Physics **1993**, *98*, 5648-5652.

(32) Easton, R. E.; Giesen, D. J.; Welch, A.; Cramer, C. J.; Truhlar, D. G., *The MIDI! basis set for quantum mechanical calculations of molecular geometries and partial charges.* Theoretica Chimica Acta **1996**, *93*, 281-301.

(33) Green, D. F.; Tidor, B., *Evaluation of ab initio charge determination methods for use in continuum solvation calculations.* Journal of Physical Chemistry B **2003**, *107*, 10261-10273.

(34) Cramer, C. J. *Essentials of Computational Chemistry. Theories and Models.*; John Wiley & Sons, Ltd.: New York, **2002**.

(35) Cossi, M.; Barone, V.; Cammi, R.; Tomasi, J., *Ab initio study of solvated molecules: a new implementation of the Polarizable Continuum Model.* Chemical Physics Letters **1996**, *255*, 327-335.

(36) Barone, V.; Cossi, M.; Tomasi, J., *Geometry optimization of molecular structures in solution by the polarizable continuum model.* Journal of Computational Chemistry **1998**, *19*, 404-417.

(37) Bondi, A., *van der Waals volumes and radii.* Journal of Physical Chemistry **1964**, *68*, 441-451.

(38) Marten, B.; Kim, K.; Cortis, C.; Freisner, R. A.; Murphy, R. B.; Ringnalda, M. N.; Sitkoff, D.; Honig, B., *New model for calculation of solvation free energies: correction of self-consistent reaction field continuum dielectric theory for short-range hydrogen-bonding effects.* Journal of Physical Chemistry **1996**, *100*, 11775-11788.

(39) Foresman, J. B.; Keith, T. A.; Wiberg, K. B.; Snoonian, J.; Frisch, M. J., *Solvent effects 5. The influence of cavity shape, truncation of electrostatics and electron correlation on ab initio reaction field calculations.* Journal of Physical Chemistry **1996**, *100*, 16098-16104.

(40) Lide, D. R.; Frederikse, H. P. R. *CRC Handbook of Chemistry and Physics, 76^th Ed.*; CRC Press, Inc.: Boca Raton, **1995**.

(41) Abraham, M. H.; Chadha, H. S.; Whiting, G. S.; Mitchell, R. C., *Hydrogen-bonding. 32. An analysis of water-octanol and water-alkane partitioning and the delta-logP parameter of Seiler.* Journal of Pharmaceutical Science **1994**, *83*, 1085-1100.

(42) Abraham, M. H.; Andonian-Haftvan, J.; Whiting, G. S.; Leo, A., *Hydrogen bonding. Part 34. The factors that influence the solubility of gases and vapours in water at 298 K, and a new method for its determination*. Journal of the Chemical Society Perkins Transactions 2 **1994**, 1777-1790.

(43) Weast, R. C.; Astle, M. J. *CRC Handbook of Chemistry and Physics*; Chemical Rubber Company Press, Inc.: Boca Raton, **1981**.

(44) Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical Recipes in C: The Art of Scientific Computing*; Cambridge University Press: Cambridge, **1993**.

(45) Israelachvili, J. N. *Intermolecular and Surface Forces, 2^nd Ed.*; Academic Press: New York, **1991**.

(46) Li, J.; Cramer, C. J.; Truhlar, D. G., *MIDI! basis set for silicon, bromine, and iodine*. Theoretical Chemistry Accounts **1998**, *99*, 192-196.

Chapter 5
Illustrative environmental transport calculations for the hypothetical gasoline additive, n-pentyl nitrate

5.1. Introduction

Preceding chapters have described methods that could be used to either forecast chemical environmental fate or estimate environmentally relevant chemical thermodynamic properties. In realistic applications, however, several modeling tools would probably be applied in tandem. It is illuminating to consider combined environmental fate forecasts for a "novel" compound, based on physical properties which are unknown and therefore must be estimated. In the following chapter, a set of example calculations has been assembled, in order to: (a) demonstrate how the previously described modeling tools would be applied to a new compound; (b) identify gaps in information that arise; (c) assess propagation of uncertainty through the linked models; and (d) discuss the ultimate utility and limitations of the entire analysis to decision-makers.

As an illustrative case study, I will presume that n-pentyl nitrate (nPN) has been proposed by industry scientists as a novel octane-enhancing gasoline additive:

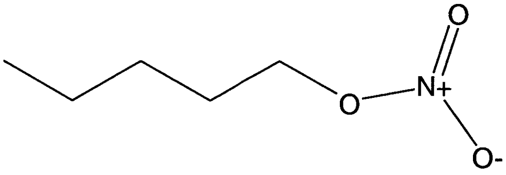

Figure 5-1. n-pentyl nitrate (nPN)

Available physical-chemical data of nPN in the scientific literature were limited (Table 5-1). Several properties of nPN relevant to atmospheric fate have been measured: i.e., reaction rates with light and hydroxyl radical; air-water partition coefficient. But properties pertaining to nPN in fuel mixtures, such as the gasoline-water partition coefficient or volatility in gasoline (gasoline-air partition coefficient) were not found, and properties relevant to nPN transport in the subsurface (organic matter-water partition coefficient) were not found. Additionally, the

Table 5-1. Known physical-chemical properties of nPN

Excess molar refraction (*1*)	0.15
Boiling point at 1 atm (*2*)	157 °C
Liquid vapor pressure at 25 °C (*2*)	6.7×10^{-3} atm
Air-water partition coefficient at 25 °C (*2*)	0.07 (dimensionless)
Tropospheric photodissociation rate[a] (*3*)	0.17 day^{-1}
Hydroxyl radical reaction rate constant (*4*)	3.3×10^{-12} cm^3 molecule^{-1} s^{-1}

[a] Clemitshaw and co-workers measured quantum yields and photodissociation rates of nPN over a range of light wavelengths, and then used these data to extrapolate diurnally averaged rates of nPN dissociation due to exposure to sunlight at 40° N latitude in the lower troposphere during July.

literature search revealed no toxicity data. Given the limited information about this compound, how would one assess its behavior in the environment resulting from use in gasolines? We would like to determine whether it is an environmentally acceptable amendment to gasoline, and at what levels, based on physically meaningful forecasts. In order to assess whether significant human exposures to nPN could result from its use in gasoline, two relevant environmental transport scenarios will be considered. First, I will examine the likely threat that this compound would pose to subsurface community water supplies, assuming that it is not rapidly degradable in the subsurface. Second, I will forecast the urban air quality consequences of volatile emissions of nPN from automobiles in a typical U.S. airshed (Boston, MA). Much of the physical property data required for such forecasts are not included in Table 5-1; hence these parameters will be estimated (partly) using tools developed in the thesis.

Although federal agencies have not (to my knowledge) set any exposure limits for nPN, I hypothetically supposed that toxicological assessments suggest an average human nPN dose of ≤ 2 μg/day to be safe, regardless of the route of exposure. Hence one could conjecture the following exposure thresholds: ≤ 1 ppb in drinking water, assuming a daily drinking water intake of 2 L; and ≤ 100 ng/m^3 in ambient urban air, assuming a daily air intake of 20 m^3 (*5*). I emphasize that these presumed human toxicity thresholds are purely fictitious; they are suggested only for didactic purposes, in order to provide a target of "acceptable" chemical exposure levels for the example calculations shown here. However, this supposition highlights an important data need: in order for exposure assessment models to give meaningful results, we require some kind of outside information about what exposure levels would be tolerable.

When several models are successively linked, their respective errors add together, and this may attribute to significant uncertainties in the final desired forecast or analysis. In order to track parameter uncertainties through the calculations presented here, first order error-propagation analysis was applied (*6*), described briefly as follows. It was always assumed that model parameters (or in some cases, the log-transformed parameters) had errors which were normally distributed and uncorrelated with the error distributions of other parameters. Under these conditions, the propagated variances of the dependent variable (model prediction) relating to each uncertain independent variable (model parameter) were superpositionable, or additive (*7*). Each model parameter's contribution to the uncertainty of the dependent variable was approximated using the first Taylor series term of the dependent variable variance with respect to the variance of the model parameter. Consequently, the summed contributions of model input parameters to the model prediction uncertainty could be described as (*7*):

$$\sigma_y^2(x_1, x_2, x_3, \ldots) = \left(\sigma_{x_1}\frac{\partial y}{\partial x_1}\right)^2 + \left(\sigma_{x_2}\frac{\partial y}{\partial x_2}\right)^2 + \left(\sigma_{x_3}\frac{\partial y}{\partial x_3}\right)^2 + \ldots \qquad (5\text{-}1)$$

where y is the model predicted variable, x_1, x_2, and x_3 are the model input parameters, and σ_i^2 is the variance of parameter i. In this way, parameter uncertainties could be added and propagated through the linked estimation models – the terms "uncertainty" or "error" as used in this chapter always refers to $\pm\sigma$ (one standard deviation) of the input parameter or predicted variable of interest.

5.2. Physical property estimation I: LSER solute parameters of nPN

Experimentally determined LSER solute parameters were not available for nPN. However, they could be estimated using the Absolv™ (*1*), which employs the fragment contribution method of Platts et al. (*8*). Separately, the solute polarity parameter, π_2^H, could be estimated with the approach described in Chapter 4. Integration of equation 4-10 along an HF/MIDI!-computed 0.0004 e^-/bohr^3 isosurface in the presence of a SCIPCM dielectric gave $\Sigma V_s^2 = 8.24$ kcal Å/mol for nPN. The measured R_2 was 0.15 (*1*); hence, application of equation 4-12 gave a π_2^H estimate of:

$$\pi_2^H = 0.47R_2 + 0.092\Sigma V_s^2 \pm 0.11 = 0.47\times 0.15 + 0.092\times 8.24 \pm 0.11 = 0.83 \pm 0.11 \qquad (5\text{-}2)$$

which agreed with the π_2^H and R_2 predictions of the fragment method of Platts et al. (Table 5-2).

Table 5-2. LSER solute parameter estimates and uncertainties for nPN

solute parameter	Absolv™ prediction	Equation 5-2
R_2	0.22 ± 0.09	-
π_2^H	0.83 ± 0.19	0.83 ± 0.11
α_2^H	0.00	-
β_2^H	0.39 ± 0.15	-
V_x	1.046[a]	-

[a] The solute term V_x is defined as a fragment-contributable quantity (*9*), hence uncertainty estimates do not apply.

5.3. Physical property estimation II: the gasoline-water partitioning of nPN

Given the LSER solute parameters (Table 5-2), one could estimate hypothetical gasoline-water partition coefficients (K_{gw}) of nPN using the LSER mixing rule described in Chapter 3. Let us suppose that nPN has been proposed as an amendment to the conventional gasoline reported in Table 3-2, at possible nPN addition rates ranging from <0.01 vol% to 25 vol%.

First, the case of a dilute (< 0.01 vol%) nPN amendment to the gasoline was considered. It was assumed that the gasoline was equilibrated with a contacting aqueous phase; that negligible water entered the gasoline phase; and that organic components in the aqueous phase were at insufficient concentrations to affect the aqueous solvent properties. In this case, equation 3-5 could be applied directly to the gasoline-water system, using the solvent component substitutions described in Chapter 3 and the corresponding LSER coefficients (Table 3-4), resulting in the following gasoline-water LSER:

$$\log K_{gw} = -0.171 + 0.625R_2 - 1.283\pi_2^H - 3.272\alpha_2^H - 4.715\beta_2^H + 4.393V_x \pm 0.4 \qquad (5\text{-}3)$$

The reader is reminded that the water-air LSER coefficients in Table 3-4 could be subtracted from a set of solvent-air LSER coefficients, to give the corresponding solvent-water LSER coefficients. I applied the *ab initio* π_2^H estimate (equation 5-2) and other Absolv™-predicted

LSER solute parameters (Table 5-2) to equation 5-3 and propagated the uncertainty of the solute parameter predictions together with the estimated LSST-LSER uncertainty. Thus I estimated:

$$\log K_{\text{gw, nPN}} = 2.00 \pm 0.85 \qquad (5\text{-}4)$$

for the dilute nPN-amendment case. The uncertainty of this log K_{gw} estimate is dominated by error of the β_2^H prediction and error of the LSST-LSER method. This highlights the need for improved approaches to estimate LSER solute parameters and continuing work on mixtures modeling.

Next, increased nPN additions to gasoline were considered. With higher amendment rates, nPN becomes an important solvent component and thereby contributes additional uncertainty to the log K_{gw} estimate. According to the LSST-LSER mixing rule, the LSER coefficients for nPN solvent are needed in order to solve equation 3-5 for gasolines containing significant concentrations of the additive. Since LSER coefficients describing nPN solvent were not available, I bounded the problem by considering two limiting cases. A "nonpolar solvent" limiting case was defined by the alkane-air LSER coefficients (Table 5-3). Based on inspection of other organic solvent LSER coefficients (Table 3-4), a "polar solvent" limiting case was also described, using the alkane-air LSER with modifications to coefficients *s* (+1.0), *a* (+5.0), and *b* (+1.0). nPN exhibits electrostatic surface potential comparable to that of an alkyl alcohol, has an excess polarizability similar to that of cyclohexane, and is not a hydrogen-bond donor; consequently these LSER solvent coefficient estimates were considered 95% confidence limits.

Table 5-3. Two suggested bounding cases of LSER solvent coefficients for nPN

	c	*r*	*s*	*a*	*b*	*m*
nonpolar	-0.71	1.23	0.89	0.30	0.02	3.39
polar	-0.71	1.23	1.89	5.30	1.02	3.39

Fuel mixtures having abundant nPN additions might be sufficiently polar to cause significant mixing between the aqueous phase and the gasoline phase, so the assumption of constant system composition was relaxed for these cases. In order to solve these mixture systems, components of the gasoline and aqueous phase were allowed to exchange between the two phases by iterative calculation of ϕ_i^g, ϕ_i^w, and $K_{gw,i}$ for each component *i*, until these parameters were self-consistent in successive iterations, according to the LSST mixing rule (equation 3-5). Consistent with equation 3-5, a gas (air) phase was taken to be the reference phase. The organic phase was assumed equilibrated with a much smaller volume water, so that dissolution of gasoline components into the aqueous phase did not significantly diminish their abundance in the organic phase; this was considered an appropriate approximation near the gasoline-water interface of a subsurface release. Specifically, the procedure was:

Step 1 (a) Assume pure aqueous phase ($\phi_w^w = 1$; $\phi_i^w = 0$ for all $i \neq w$).
(b) Assume gasoline composition as conventional (Table 3-2) + nPN amendment.
(c) Go to step 2.

Step 2 (a) Calculate $K_{gw,i}$ for all gasoline and aqueous components using equation 3-5, based on the system composition (ϕ_i^w and ϕ_i^g values) of the previous step.

(b) If this is the first iteration of the procedure, go to step 3.

(c) Otherwise, check to see if the relative change in $K_{gw,i}$ value for any system component *i* exceeded 10^{-9} during this update step. If yes, then go to step 3; if no, then the system is considered equilibrated (end of calculation).

Step 3 (a) Compute new ϕ_i^w values of each organic (non-water) component resulting from transfer of gasoline components into the aqueous phase, based on the organic phase composition (ϕ_i^g values) and $K_{gw,i}$ values of step 2.

(b) Update the water content of the aqueous phase to account for the added gasoline components: $\phi_w^w = 1-\Sigma_i\phi_i^w$ for all $i \neq w$.

(c) Compute a new water concentration in the organic phase (ϕ_w^g value) using the gasoline-water partition coefficient for water ($K_{gw,w}$) given in step 2 and the ϕ_w^w value calculated in step 3(b).

(d) Update the concentration of each organic phase component to account for the water that has been added to the organic phase; $\phi_i^g{}_{new} = \phi_i^g /(\Sigma_j\phi_j^g)$.

(e) Go to step 2.

where convergence was defined by step 2(c). For nPN amendments of up to 25 vol%, the LSST-LSER model predicted negligible amounts of water in the gasoline phase, using either set of the bounding nPN solvent LSER coefficients in Table 5-3. The uncertainties generated by the lack of good nPN LSER coefficients (Table 5-4) were tolerable – although the solvent properties of nPN were highly uncertain (Table 5-3), this did not significantly affect the total estimated uncertainty of log K_{gw} calculations. In this analysis, the uncertainty of predicted log K_{gw} values was dominated by the uncertainty in the $b\beta_2^H$ term of equation 5-3 ($b\times\sigma_\beta = -4.72\times(\pm0.15)$ ~ ±0.7) and the anticipated error of the LSST-LSER mixing rule (σ_{LSST} ~ ±0.4).

Table 5-4. Estimated nPN gasoline-water partition coefficient values at varying gasoline amendment rates

nPN addition[a]	calc. log $K_{gw,nPN}$	nPN solvency error[b]	total uncertainty[c]
infinite dilution	2.00	0.00	± 0.85
0.01 vol%	2.00	~0.00	± 0.85
0.05 vol%	2.00	~0.00	± 0.85
0.2 vol%	2.00	~0.00	± 0.85
1 vol%	2.00	~0.00	± 0.85
5 vol%	2.02	± 0.02	± 0.85
25 vol%	2.08	± 0.11	± 0.86

[a] The nPN concentration in gasoline. [b] The estimated uncertainty of the calculated log $K_{gw,nPN}$ propagated from the assumption that the Table 5-3 limiting cases describe a 95% confidence interval of nPN solvent LSER coefficients. [c] The total estimated log $K_{gw,nPN}$ uncertainty, including propagated solute parameter prediction error (Table 5-2), LSST-LSER error (~0.4), and nPN solvency error.

5.4. Physical property estimation III: organic matter-water partitioning of nPN

The organic matter-water partition coefficient (log K_{om}) of nPN provides instrumental information about its subsurface transport behavior. The log K_{om} could be readily estimated from a log K_{ow} linear free energy relationship (LFER), consequently I first estimated the latter property for nPN. Applying the log K_{ow} LSER (*10*):

$$\log K_{\mathrm{ow}} = 0.088 + 0.562R_2 - 1.054\pi_2^{\mathrm{H}} + 0.034\alpha_2^{\mathrm{H}} - 3.460\beta_2^{\mathrm{H}} + 3.814V_{\mathrm{x}} \pm 0.12 \qquad (5\text{-}5)$$

and using the estimated solute parameters for nPN, I found:

$$\log K_{\mathrm{ow,\,nPN}} = 2.0 \pm 0.6 \qquad (5\text{-}6)$$

Similar to the case of the log $K_{gw,nPN}$ estimate, the uncertainty of the log $K_{ow,nPN}$ estimate was dominated by error in the $b\beta_2^H$ term. The log $K_{ow,nPN}$ estimate did not compare very favorably to the measured log K_{ow} value of its structurally related isomer, isopentyl nitrate (log K_{ow} = 2.84 (*11*) for the structure $(CH_3)_2CHCH_2CH_2ONO_2$). This suggests that the error of the predicted log $K_{ow,nPN}$ may be greater than that given in equation 5-6. Taking the equation 5-6 log $K_{ow,nPN}$ estimate, a log K_{om}-log K_{ow} LFER (*12*) for polar solutes was employed:

$$\log K_{\mathrm{om}} = 0.59 \log K_{\mathrm{ow}} + 0.78 \pm 0.2 \qquad (5\text{-}7)$$

Consequently:

$$\log K_{\mathrm{om,nPN}} = 1.9 \pm 0.4 \qquad (5\text{-}8)$$

5.5. Impact of nPN on subsurface community water supplies

Having established estimates of the gasoline-water and organic matter-water partitioning behavior of nPN, we are in a position to consider the behavior of this compound in the subsurface. The physical-chemical community supply well screening model (chapter 2) was applied to nPN, using the $K_{om,nPN}$ estimate of equation 5-8, the nPN gasoline concentrations and $K_{gw,nPN}$ estimates given in Table 5-4, and the hydrogeologic parameters of Table 2-2. I thereby forecasted water concentrations ranging from 2 to 40 ppb in vulnerable community supply wells, expected to occur within less than a decade (Table 5-5).

Table 5-5. Expected community supply well contamination levels and arrival times as forecasted by the physical-chemical well screening model (equations 2-18, 2-22)

nPN addition	C_{well} forecast [ppb]	$t_{arrival}$ forecast [yrs]
0.01 vol%	0.02	6
0.05 vol%	0.1	6
0.2 vol%	0.4	6
1 vol%	2	6
5 vol%	10	6
25 vol%	44	6

Before interpreting these forecasts, let us first address their expected uncertainty. In chapter 2, variability in the model hydrogeologic parameters was shown to result in an uncertainty ranging from ± 0.83 to ± 0.93 in the log C_{well} estimate, for ethylbenzene and MTBE, respectively (I will assume $\sigma_{\log C_{well}}(\text{hydrogeol.}) = \pm 0.90$ for nPN). It would be additionally informative to evaluate the contribution of $K_{gw,nPN}$ to uncertainty in the community supply well screening model forecast. I assumed that ln K_{gw} uncertainty is normally distributed and that ln C_{well} variability is normally distributed (which is supported by the analysis in chapter 2). In this case, the propagation of ln K_{gw} uncertainty in the ln C_{well} forecast could be approximated vis-a-vis equation 5-1:

$$\sigma_{\ln C_{well}}\left(\ln K_{gw}\right) = \sigma_{\ln K_{gw}}\left[\frac{\partial \ln C_{well}}{\partial \ln K_{gw}}\right] \tag{5-9}$$

From equation 2-18, we can express the derivative in equation 5-9 in terms of measurable input parameters, as:

$$\frac{\partial \ln C_{well}}{\partial \ln K_{gw}} = \frac{\partial}{\partial \ln K_{gw}} \ln\left[\frac{\left(\dfrac{0.2 v_x C_g V_g}{R Q_{well}}\right)}{\sqrt{0.1\left(\dfrac{K_{gw}}{R}\right)^2\left(\dfrac{V_g}{\phi}\right)^{\frac{1}{2}}\dfrac{\left(h_g S_g\right)^{\frac{3}{2}}}{a_{z,10}} + 2a_x L_x}}\right] \tag{5-10}$$

Expanding the log-transformed term of equation 5-10, it was found that:

$$\frac{\partial \ln C_{well}}{\partial \ln K_{gw}} = \frac{\partial}{\partial \ln K_{gw}}\left[\ln\left(\frac{0.2 v_x C_g V_g}{R Q_{well}}\right) - \frac{1}{2}\ln\left(0.1\left(\frac{K_{gw}}{R}\right)^2\left(\frac{V_g}{\phi}\right)^{\frac{1}{2}}\frac{\left(h_g S_g\right)^{\frac{3}{2}}}{a_{z,10}} + 2a_x L_x\right)\right] \tag{5-11}$$

Invoking the right-hand-side substitution $K_{gw}{}^2 = \exp(2 \ln K_{gw})$, equation 5-11 was solved analytically with little difficulty, giving:

$$\frac{\partial \ln C_{well}}{\partial \ln K_{gw}} = -\frac{1}{2}(2)\frac{0.1\left(\dfrac{K_{gw}}{R}\right)^2\left(\dfrac{V_g}{\phi}\right)^{\frac{1}{2}}\dfrac{\left(h_g S_g\right)^{\frac{3}{2}}}{a_{z,10}}}{\left(0.1\left(\dfrac{K_{gw}}{R}\right)^2\left(\dfrac{V_g}{\phi}\right)^{\frac{1}{2}}\dfrac{\left(h_g S_g\right)^{\frac{3}{2}}}{a_{z,10}} + 2a_x L_x\right)} \tag{5-12}$$

which was simplified to the final expression:

$$\frac{\partial \ln \mathrm{C}_{\mathrm{well}}}{\partial \ln K_{\mathrm{gw}}} = \frac{-1}{\left(1 + 20\mathrm{a}_{\mathrm{x}}\mathrm{L}_{\mathrm{x}}\mathrm{a}_{\mathrm{z},10}\left(\frac{\mathrm{R}}{K_{\mathrm{gw}}}\right)^{2}\left(\frac{\phi}{\mathrm{V}_{\mathrm{g}}}\right)^{\frac{1}{2}}\left(\mathrm{h}_{\mathrm{g}}\mathrm{S}_{\mathrm{g}}\right)^{-\frac{3}{2}}\right)} \tag{5-13}$$

Using equations 5-9 and 5-13 and the previously estimated uncertainties of log $K_{\mathrm{gw,nPN}}$ (Table 5-4), the contribution of $K_{\mathrm{gw,nPN}}$ to the $\mathrm{C}_{\mathrm{well}}$ forecast variability was estimated (Table 5-6). The cumulative model forecast variability was not found to be substantially increased by uncertainty in the log $K_{\mathrm{gw,nPN}}$. I did not explicitly analyze the influence of $K_{\mathrm{om,nPN}}$ uncertainty on the variability of $\mathrm{C}_{\mathrm{well}}$ forecasts. Inspection of equation 2-18 reveals that $\mathrm{C}_{\mathrm{well}}$ is a considerably weaker function of K_{om} than of K_{gw}; hence the $K_{\mathrm{om,nPN}}$ was not expected to affect the $\mathrm{C}_{\mathrm{well}}$ forecast variability in an important way.

Table 5-6. Estimated uncertainty of forecasted community supply well nPN contamination levels as related to hydrogeologic parameters and the K_{gw} parameter

nPN addition	error[a] in log K_{gw}	$\frac{\partial \ln \mathrm{C}_{\mathrm{well}}}{\partial \ln K_{\mathrm{gw}}}$	log K_{gw}-induced error[b] in log $\mathrm{C}_{\mathrm{well}}$	hydrogeologic error[c] in log $\mathrm{C}_{\mathrm{well}}$	log $\mathrm{C}_{\mathrm{well}}$ [ppb][d] + total error[e]
0.01 vol%	± 0.85	-0.28	± 0.24	± 0.9	-1.70 ± 0.93
0.05 vol%	± 0.85	-0.28	± 0.24	± 0.9	-1.0 ± 0.93
0.2 vol%	± 0.85	-0.28	± 0.24	± 0.9	-0.40 ± 0.93
1 vol%	± 0.85	-0.28	± 0.24	± 0.9	0.30 ± 0.93
5 vol%	± 0.85	-0.30	± 0.26	± 0.9	1.00 ± 0.94
25 vol%	± 0.86	-0.36	± 0.31	± 0.9	1.64 ± 0.95

[a] Standard error estimates from Table 5-4. [b] Standard error of log $\mathrm{C}_{\mathrm{well}}$ resulting from K_{gw} variability, calculated using equations 5-9 and 5-13. [c] Standard error of log $\mathrm{C}_{\mathrm{well}}$ resulting from hydrogeologic parameter variability (Table 2-2). [d] Forecasted log $\mathrm{C}_{\mathrm{well}}$ averages (Table 5-5). [e] Combined standard error of log $\mathrm{C}_{\mathrm{well}}$ resulting from hydrogeologic variability plus K_{gw} uncertainty (equation 5-1).

Based on the drinking water exposure criteria that I supposed in section 5-1 (an nPN drinking water limit of 1 ppb), most of the suggested gasoline formulations fail to adequately protect community water supply resources (Table 5-6). Unless nPN is shown to be rapidly and prevalently biodegradable in subsurface environments, the transport forecasts described here suggest that gasoline formulations amended with >0.05 vol% nPN could threaten many community water supplies in high-use areas. If additional work demonstrated that nPN persists in typical subsurface environments for years, a strict nPN amendment limit of 0.05 vol% should be considered and potential health impacts of nPN exposure should be rigorously investigated.

5.6. Impact of nPN volatile emissions on urban air quality

In addition to contamination of subsurface water supplies, nPN could affect urban airsheds due to volatile losses from automobiles. I conducted a screening estimate of nPN levels in the

Boston primary metropolitan area, assuming a 0.05 vol% nPN amendment to gasoline (based on the restriction set by the previous calculation). Based on the work of Kawamoto, Arey, and Gschwend (*13*), it was proposed that, for screening purposes, the urban air concentration of a volatile gasoline additive could be conservatively estimated as:

$$C_{\text{air}}^{\text{steady state}} \approx \frac{\dfrac{\text{NPN emission rate}}{V_{\text{air}}} + \dfrac{C_{\text{air.in}}}{V_{\text{air}}}\dfrac{dV_{\text{air}}}{dt}}{\dfrac{1}{V_{\text{air}}}\dfrac{dV_{\text{air}}}{dt} + k_{\text{OH,nPN}}\{\cdot\text{OH}\} + k_{\text{photo,nPN}}} \qquad (5\text{-}14)$$

where V_{air} is the volume of the lower troposphere mixed layer of the Boston, Massachusetts, primary metropolitan area (5×10^{12} m^3), assuming a mixing height of 1000 m, dV_{air}/dt is the atmospheric flushing rate due to wind (4×10^{13} m^3/day), assuming an average wind speed of 3.5 m/sec, $C_{air,in}$ is the upwind boundary concentration of nPN in the atmosphere (assumed zero), $k_{OH,nPN}$ is the rate of reaction of hydroxyl radical (·OH) with nPN (estimated as 4.6×10^{-12} cm^3/molec/sec at 298 °K using the fragment contribution of Kwok and Atkinson (*14*), in reasonable agreement with the reported value of 3.3×10^{-12} cm^3/molec/sec in Table 5-1), $\{\cdot OH\}$is an estimate of the diurnally averaged concentration of hydroxyl radical in the lower troposphere during the summer months (~2×10^6 molec/cm^3), and k_{photo} is the photolytic dissociation rate in the lower troposphere (0.17 day^{-1}; Table 5-1). Based on the chemical of nPN, it may also be advisable to consider the possibility of hydrolysis in the atmosphere, however neither data nor computation methods were readily found to estimate the hydrolysis rate constant(s).

Precipitation was not expected to significantly contribute to nPN removal from the Boston urban airshed. Heavy summer rain in Boston may amount to accumulation of a few cm during a day. If I assumed that a 5 cm rainfall was completely saturated with nPN scavenged from the lower mixing layer of the troposphere (i.e., the layer between ground level and about ~1000 m height), then the mass ratio of nPN in the atmosphere relative to that in the rainfall could be estimated as $R_{a/w} = K_{aw,nPN}*V_{air}/V_{water} = 0.07*1000/0.05 = 1400$. Since $R_{a/w} >> 1$, this calculation indicates that rainfall would not significantly affect the atmospheric reservoir of nPN.

In order to calculate the emission rate of nPN to the Boston primary metropolitan area, an estimate of nPN gasoline-air partitioning was needed. The gasoline-air partition coefficient of nPN could be estimated by adding the gasoline-water partitioning LSER for nPN (equation 5-3) with the water-air partitioning LSER (*15*), assuming $K_{ga,nPN} = K_{gw,nPN}*K_{wa,nPN}$. This approximation was considered reasonable, since the LSST-LSER model did not predict much entrainment of water into the gasoline phase, for the gasoline composition reflected by equation 5-3. Using the nPN solute parameters in Table 5-1, the water-air partition coefficient LSER was used (*15*):

$$\log K_{\text{wa}} = -0.994 + 0.577R_2 + 2.549\pi_2^{\text{H}} + 3.813\alpha_2^{\text{H}} + 4.841\beta_2^{\text{H}} - 0.869V_{\text{x}} \pm 0.15 \qquad (5\text{-}15)$$

giving a predicted log $K_{wa,nPN} = 1.9 \pm 0.8$. This was consistent with the measured log $K_{wa,nPN}$ value of 1.15 (Table 5-1). As with the $K_{gw,nPN}$ estimate, the $K_{wa,nPN}$ prediction uncertainty largely owed to the significant contribution of the $b\beta_2^H$ term of equation 5-15 ($b\times\sigma_\beta = 4.84\times(\pm0.15)$

~ ±0.7). The gasoline-air LSER was therefore estimated as:

$$\begin{aligned}\log K_{ga} &= -0.171 + 0.625R_2 - 1.283\pi_2^H - 3.272\alpha_2^H - 4.715\beta_2^H + 4.393V_x \pm 0.4 \\ &\quad - 0.994 + 0.577R_2 + 2.549\pi_2^H + 3.813\alpha_2^H + 4.841\beta_2^H - 0.869V_x \pm 0.15\end{aligned}$$

$$\log K_{ga} = -1.16 + 1.21R_2 + 1.27\pi_2^H + 0.54\alpha_2^H + 0.13\beta_2^H + 3.49V_x \pm 0.43 \qquad (5\text{-}16)$$

This resulted in a calculated gasoline-air partition coefficient of:

$$\log K_{ga,nPN} = 3.86 \pm 0.50 \qquad (5\text{-}17)$$

using the Absolv^{TM}-predicted parameters (Table 5-1), and:

$$\log K_{ga,nPN} = 3.86 \pm 0.47 \qquad (5\text{-}18)$$

using the *ab initio* π_2^H estimate (equation 5-2) with the remaining Absolv^{TM}-predicted parameters. The LSST-LSER approximation was the largest source of uncertainty in log $K_{ga,nPN}$ estimates. The estimated error of the log $K_{ga,nPN}$ predictions were considerably smaller than that of the log $K_{gw,nPN}$ prediction: this was largely because β_2^H, a highly uncertain solute parameter, is critical to the log $K_{gw,nPN}$ estimate (equation 5-3), but β_2^H is an insignificant parameter in the log $K_{ga,nPN}$ estimate (equation 5-16). The partial pressure of nPN in gasoline vapor implied by the gasoline-air partition coefficient, at 298 °K is given by:

$$\begin{aligned}p_{NPN} &= \gamma_{NPN,g} X_{NPN,g} p^*_{NPN} \\ &= \frac{X_{NPN,g} RT}{K_{ga} V_g}\end{aligned} \qquad (5\text{-}19)$$

where $\gamma_{nPN,g}$ is the activity coefficient of nPN in the gasoline mixture (using the pure nPN liquid reference state), $X_{nPN,g}$ is the mole fraction of nPN in gasoline, p^*_{nPN} is the pure liquid vapor pressure of nPN at 298 °K [atm], R is the molar gas constant (0.08206 L atm mol^{-1} °K^{-1}), T is temperature [°K], and V_g is intensive (molar) volume of the gasoline [L mol^{-1}]. The molar volume of gasoline was related to the mole fractions and molar volumes of the gasoline constituents, assuming that the partial molar volume of mixing was negligible:

$$V_g = \sum_i X_i V_i \qquad (5\text{-}20)$$

The gasoline composition was, as before, taken to be the conventional gasoline described in Table 3-2, plus an nPN amendment of 0.05 vol%, resulting in a calculated V_g = 0.12 L/mol. Using this composition, the partial pressure of nPN was calculated using equation 5-19, and the partial pressures of all other gasoline components were computed using UNIFAC-calculated activity coefficients for the unamended base gasoline (*16*) (i.e., it was assumed that the presence of nPN did not significantly change the nonideality coefficients of other gasoline components). The nPN mass fraction in the gasoline vapor at 298 °K could then be estimated as:

$$F_{nPN}^{vap} = \frac{p_{nPN} m_{nPN}}{\sum_i p_i m_i} \tag{5-21}$$

where m_i is the molar mass of gasoline component i, and the denominator represents a sum of weighted partial pressures for all gasoline components. Using these approximations, the total vapor pressure of the gasoline mixture was found to be 0.49 atm, and nPN was estimated to compose 50 ± 30 ppm (by mass) of the gasoline vapor. The F_{NPN}^{vap} uncertainty estimate was based only on the propagated uncertainty of the log $K_{ga,nPN}$. The automobile volatile emission rate of nPN into the Boston primary metropolitan area airshed could thus be estimated (*13*) as:
(86 g gasoline vapor/day/vehicle)×(5×10^{-5} g nPN/g gasoline vapor)×(2.8×10^{6} vehicles) =
12 kg nPN/day.

Applying the source function estimate and the previously suggested transport parameters to equation 5-14, the steady state nPN concentration on a typical summer day in Boston was forecast to be:

$$C_{air}^{steady\ state} \approx 0.2 \pm 0.2\,\text{ng/m}^3 \tag{5-22}$$

where the uncertainty estimate accounts for the variability in the emission rate and the estimated variability of the diurnally averaged physical parameters during a four-day experiment in Boston during September of 2000 (*13*). In the event that an inversion layer prevented significant movement of the lower troposphere air mass over the urban area, air flushing might be negligible, and the new steady state concentration of nPN (in which the only removal process is reaction with ·OH) was estimated to be an order of magnitude higher in concentration:

$$C_{air}^{steady\ state}(\text{no wind}) \approx 2 \pm 2\,\text{ng/m}^3 \tag{5-23}$$

If such conditions are additionally exacerbated by unusually low ·OH and light levels (e.g., due to dense cloud cover), nPN concentration forecasts would be even higher.

Although variable conditions create widely ranging forecasts of nPN concentration in the Boston urban atmosphere, these levels are still two orders of magnitude below air concentrations which are (hypothetically) expected to result in potentially unhealthy exposures (>100 ng/m^3). These screening assessments ignored many other factors, for example: other mechanisms of transport such as nPN air-water exchange into surface waters or precipitation, diurnal variability of traffic, other sources of nPN in the atmosphere (*17*), and possibly increased gasoline volatilization rates associated with higher ambient temperatures (I have assumed T = 25 °C here). However, these effects were not expected to change forecasted nPN diurnal average levels in urban air by more than an order of magnitude.

5.7. Conclusions and recommendations

In this chapter, I used the nPN case study to: illustrate the usefulness and limitations of forecasting models designed to screen the environmental impacts of proposed gasoline additives; highlight information needs; and identify major sources of uncertainty in such approaches. In order to conduct a meaningful forecasting assessment, some information about health effects of nPN was required, i.e., preliminary acceptable levels of human exposure to nPN in water and air. Several sources of error or uncertainty produced challenges to the thread of analysis, and these illuminate areas for future improvements to the approaches. First, LSER parameters were not available for nPN. The Absolv^{TM} and *ab initio* methods were used to estimate a set of nPN LSER solute parameters, and uncertainty in the β_2^H parameter subsequently produced significant uncertainty in the predicted log $K_{gw,nPN}$. The other major origin of uncertainty in log $K_{gw,nPN}$ predictions was the error of the LSST-LSER approach. As a result, use of the π_2^H estimate given by equation 5-2 did not significantly improve the uncertainty of the log $K_{gw,nPN}$ estimate – other sources of error simply dominated the $K_{gw,nPN}$ prediction. Although measured LSER solvent coefficients of nPN were not available, consideration of bounding cases suggested that this information gap was unlikely to cause significant (additional) inaccuracies in log $K_{gw,nPN}$ predictions. However, the reader is reminded that LSST-LSERs may be inaccurate for gasoline mixtures which are radically different (i.e., not composed mostly of aromatic and aliphatic hydrocarbon components), particularly if pure phase LSER coefficients of the major components are not available. For log $K_{ga,nPN}$ estimates, the LSST-LSER approximation was the major source of uncertainty. It is worth considering whether better log $K_{gw,nPN}$ and log $K_{ga,nPN}$ estimates could have been made using UNIQUAC (*18*) or UNIFAC (*16*): unfortunately, the solute-solvent interaction parameters needed for the nitrate ($-ONO_2$) functionality are not currently available for these methods. This highlights the motivation for the methods described in Chapters 2 and 3, which are designed to handle a broad set of solute-solvent interactions that may not be treatable using more traditional approaches such as UNIFAC or UNIQUAC. In spite of the significant uncertainties of log $K_{gw,nPN}$ and log $K_{ga,nPN}$ estimates, the expected variability of C_{well} and C_{air} forecasts were dominated by other (physical) parameter variabilities. These order-of-magnitude environmental concentration forecasts then generated the basis for useful comparisons with (probably equally uncertain) exposure limits. Consequently, decision-making criteria were attainable in the face of substantial uncertainties.

It should be emphasized that these screening model results suggest that, at 0.05 vol% addition levels to gasoline, nPN would be an acceptable additive in terms of contamination of community subsurface water supplies and nonpoint-source contamination of the Boston urban airshed (given the arbitrary air and water exposure thresholds that have been chosen). The work outlined here reflects only a subset of the preliminary studies that should be performed for nPN, among many other additives that might be hypothetically considered. Other similar screening models should be developed to: address potential nPN exposures of service station workers; consider nPN impacts on surface water supplies; evaluate the effects of nPN on wildlife and its tendency to bioaccumulate in biota, particularly if it is persistent; and so forth. Emphasis has been placed on developing computationally streamlined forecasting approaches which estimate order-of-magnitude outcomes. If, among many considered gasoline additives, such screening evaluations prioritize nPN as a highly likely candidate, additional work should be done before nPN is actually added to gasolines: e.g., to extensively study the potential health effects of exposure to nPN; to understand its interaction with urban atmospheric contaminants (including reaction daughter products); and to evaluate its degradability in a wide set of environmental compartments (e.g., in the troposphere, in the subsurface, in surface waters, and in humans and other biota), among other environmental and health considerations.

5.8. References

(1) Advanced Pharma Algorithms Inc., Absolv^{TM}, 2001 ADME Boxes version 2.2.

(2) Hauff, K.; Fischer, R. G.; Ballschmiter, K., *Determination of C1-C5 alkyl nitrates in rain, snow, white frost, lake, and tap water by a combined codistillation head-space gas chromatography technique. Determination of Henry's law constants by head-space GC.* Chemosphere **1998**, *37*, 2599-2615.

(3) Clemitshaw, K. C.; Williams, J.; Rattigan, O. V.; Shallcross, D. E.; Law, K. S.; Cox, R. A., *Gas-phase ultraviolet absorption cross-sections and atmospheric lifetimes of several C2-C5 alkyl nitrates.* Journal of Photochemistry and Photobiology A: Chemistry **1997**, *102*, 117-126.

(4) Nielsen, O. J.; Sidebottom, H. W.; Donlon, L.; Treacy, J., *An absolute-rate and relative-rate study of the gas-phase reaction of OH radicals and Cl atoms with normal-alkyl nitrates.* Chemical Physics Letters **1991**, *178*, 163-170.

(5) Masters, G. M., *Chapter 4. Risk Assessment*, In *Introduction to Environmental Engineering and Science*; 2nd ed.; Prentice-Hall, Inc.: Upper Saddle River, NJ, **1998**; pp 117-162.

(6) Haefner, J. W., *Chapter 9. Model Analysis*, In *Modeling Biological Systems. Principles and Applications*; Chapman & Hall: New York, NY, **1996**; pp 179-211.

(7) Fischer, H. B.; List, J. E.; Koh, R. C. Y.; Imberger, J.; Brooks, N., *Chapter 2. Fickian Diffusion*, In *Mixing in Inland and Coastal Waters*; Academic Press: San Diego, **1979**; pp 30-54.

(8) Platts, J. A.; Abraham, M. H.; Butina, D.; Hersey, A., *Estimation of molecular linear free energy relationship descriptors by a group contribution approach.* Journal of Chemical Information and Computer Sciences **1999**, *39*, 835-845.

(9) Abraham, M. H.; McGowan, J. C., *The use of characteristic volumes to measure cavity terms in reversed phase liquid chromatography.* Chromatographia **1987**, *23*, 243-246.

(10) Abraham, M. H.; Chadha, H. S.; Whiting, G. S.; Mitchell, R. C., *Hydrogen-bonding. 32. An analysis of water-octanol and water-alkane partitioning and the delta-logP parameter of Seiler.* Journal of Pharmaceutical Sciences **1994**, *83*, 1085-1100.

(11) Hansch, C.; Leo, A. J. *Exploring QSAR. Hydrophobic, Electronic, and Steric Constants*; American Chemical Society: Washington D.C., **1995**.

(12) Schwarzenbach, R. P.; Gschwend, P. M.; Imboden, D. M., *Chapter 9. Sorption I: Sorption Processes Involving Organic Matter*, In *Environmental Organic Chemistry*; 2nd ed.; John Wiley & Sons, Inc.: Hoboken, NJ, **2003**; pp 275-330.

(13) Kawamoto, K.; Arey, J. S.; Gschwend, P. M., *Emission and fate assessment of methyl tertiary butyl ether in the Boston area airshed using a simple multimedia box model: comparison with urban air measurements.* Journal of the Air & Waste Management Association **2003**, *53*, 1426-1435.

(14) Kwok, E. S. C.; Atkinson, R., *Estimation of hydroxyl radical reaction rate constants for gas-phase organic compounds using a structure activity relationship: an update.* Atmospheric Environment **1995**, *29*, 1685-1695.

(15) Abraham, M. H.; J, A.-H.; Whiting, G. S.; Leo, A.; Taft, R. S., *Hydrogen bonding. Part 34. The factors that influence the solubility of gases and vapours in water at 298 K, and a new method for its determination.* Journal of the Chemical Society. Perkin Transactions 2 **1994**, *2*, 1777-1791.

(16) Fredenslund, A.; Jones, R. J.; Praustnitz, J. M., *Group-contribution estimation of activity coefficients in nonideal liquid mixtures.* AIChE J. **1975**, *21*, 1086-1099.

(17) Bertman, S. B.; Roberts, J. M.; Parrish, D. D.; Buhr, M. P.; Goldan, P. D.; Kuster, W. C.; Fehsenfeld, F. C., *Evolution of alkyl nitrates with air mass age.* Journal of Geophysical Research **1995**, *100*, 22805-22813.

(18) Abrams, D. S.; Prausnitz, J. M., *Statistical thermodynamics of liquid mixtures: a new expression for the excess Gibbs energy of partly or completely miscible systems.* AIChE J. **1975**, *21*, 116-128.

Chapter 6
Summary and conclusions

The goal of this work was to develop practical models for anticipating the environmental impacts of synthetic organic chemicals. First, the scope of the problem was emphasized – decision makers need tools which will enable them to prioritize the health and environmental testing of currently used commercial chemicals. Likewise, approaches to pre-evaluate newly proposed substances could focus industrial research resources towards those chemicals which are likely to be environmentally benign, thereby reaping cost avoidance benefits both monetarily and environmentally.

Gasoline mixtures provided a motivating case study, since quantitative strategies to anticipate widespread gasoline contamination of urban air and subsurface water resources have not been systematically developed. In response to these challenges, I developed and tested methods designed to: (a) forecast environmental threats posed by newly proposed gasoline additives; and (b) estimate environmentally relevant physical chemical properties of such compounds, in the event that these properties are not previously determined.

In chapter 2, a model to forecast the widespread contamination of community subsurface water supplies by new gasoline additives was presented. Conventional subsurface contamination models have been largely developed to address the needs of local site assessments; consequently they are tailored to specific site conditions rather than to specific fuel formulations. It was therefore necessary to develop an approach which accounted for variability in hydrogeologic and gasoline contamination conditions at sites near community water supply wells. As a result of the model development, it was found that some hydrogeologic conditions are widely generalizable at many or most community supply wells in the U.S.: unconsolidated and coarse grained aquifer material, low sediment organic matter content, and shallow, unconfined saturated zones. Consequently, I concluded that a representative physical regime with stochastic hydrologic and gasoline release parameters could effectively capture the contamination risks borne by vulnerable community supply wells. The forecasting model correctly predicted the range of methyl-*tert*-butyl ether (MTBE) contamination levels (~ 1 to 100 ppb) observed in the set of most severely affected U.S. community supply wells (about 8% of community supply wells in sub-regions where MTBE is widely used).

The community supply well contamination forecasting model showed that other oxygenates would behave similarly to MTBE, contaminating many wells at high concentrations (tens of ppb) within a short period of time (a few years). Additionally, model calculations suggested that benzene could widely contaminate drinking water supplies at levels near or exceeding drinking water standards (5 ppb) if benzene was not biodegradable in the subsurface. This is a timely forecasting result, because a degradable oxygenate additive such as ethanol could rapidly consume dissolved oxygen down-gradient of the gasoline release. As a result of ethanol addition to fuels, benzene (which is typically aerobically degraded) could experience prolonged persistence in the subsurface, thereby creating a new threat of widespread water supply contamination.

In chapter 3, mixing rules for linear solvation energy relationships (LSERs) were used to estimate the fuel-water partition coefficients of fuel solutes. Conventional LSERs are generally fitted for solute partitioning between two relatively pure liquid phases. Consequently, it was not clear how one would extrapolate existing LSERs to mixture systems such as fuels. Linear solvent strength theory (LSST) and a solvent compartment model were used to generate mixture system predictions of solute partitioning between various aqueous phase mixtures and fuels. Without any fitting, both LSST and solvent compartment model extension of LSERs resulted in accurate predictions of gasoline-water partition coefficients of both polar and nonpolar solute (standard error ~ 0.4 log *K*) for systems in which the gasoline phase remained relatively uncontaminated by water (a few vol% or less). This was much more accurate than using Raoult's law to model the fuel phase for the same set of systems. Raoult's law has traditionally been used to describe fuel-water partitioning of nonpolar chemicals of environmental concern, but this approach fails for polar solutes in fuels. The LSST-LSER or CSCLSST-LSER estimation methods could be used to treat both polar and nonpolar solutes in future fuel formulations which resemble the systems considered here. The expected resulting errors (~ 0.4 log *K*) were considered tolerable for the purposes of environmental fate assessments, and the logarithmic average of these two models (which I dubbed HSCLSST-LSER) gave considerably improved predictions (~ 0.26 log *K*), due to a systematic offsetting of errors.

In chapter 4, a method was proposed for calculating the LSER solute polarity/polarizability parameter, π_2^H . Of the five solute parameters that have been developed by Abraham and co-workers for use in LSERs (*1*), π_2^H is the least physically interpretable. Additionally, the current fragment contribution estimation method (*2*) for π_2^H is not very accurate (standard error of 0.19 in the unitless π_2^H scale), and it cannot treat solutes having previously unstudied moieties. Using molecular orbital computational methods, I developed a computed electrostatic interaction scale which, when combined with a polarizability parameter, R_2, correlates closely with π_2^H for 90 solutes ($r^2 = 0.93$ to 0.96, depending on the parameterization of the method). Correlations suggested that about 1/3 of π_2^H variability could be explained by solute polarizability, and about 2/3 of π_2^H variability could be related to electrostatic potential at the solute surface. Additionally, calculations suggested that electron density may be a better indicator of solvent accessibility to the solute than is the traditional fixed-atomic radii van der Waals surface proposed by Bondi (*3*). The resulting model appeared to predict π_2^H values for new solutes with a standard error of 0.11 or 0.12 in the unitless π_2^H scale – much more accurately than the fragment contribution method. As a result of these efforts, π_2^H values can be estimated for new chemicals which have not been previously studied, as long as these compounds are small to moderately sized (less than ~30 heavy atoms), nonionic, and composed of C, H, N, O, S, F, Cl, or Br.

In chapter 5, I conducted example calculations to illustrate combined application of the models for a hypothetical uncharacterized gasoline additive, n-pentyl nitrate (nPN). Molecular orbital calculations and fragment contributions were used to estimate the LSER solute parameters for nPN. Estimated uncertainty in the subsequently predicted gasoline-water partition for nPN (a standard error of about 0.85 in the log K_{gw}) was mostly due to error in the

β_2^H LSER parameter estimate and the LSST mixing rule approximation. The organic matter-water partition coefficient of nPN was estimated with acceptable error (an estimated standard error of 0.4 in the log K_{om}). By using the community supply well contamination screening model to evaluate fuel formulations with different nPN amendment levels, it was found that fuel mixtures of only 0.5 vol% nPN are expected to widely contaminate community supply wells at drinking water threshold concentrations (fictitiously assumed ~ 1 ppb). Therefore it was determined that, unless nPN is shown to be prevalently and rapidly degradable in the subsurface, nPN should not be added to gasoline in amounts exceeding 0.05 vol%. Additional modeling was conducted to forecast the expected volatile emission rate of nPN from automobiles and the resulting impact on urban airsheds. At a 0.05 vol% amendment rate, nPN was expected to typically contaminate urban air in the Boston metropolitan airshed at < 1 ng/m^3, which was well within (fictitiously presumed) acceptable exposure thresholds.

These investigations demonstrated that while significant uncertainties may enter modeling calculations (e.g., 0.85 standard error in the log K_{gw} of nPN), environmental exposure forecasts nevertheless provide very useful decision making criteria. Realistically forecasted exposure outcomes exhibit order-of-magnitude variability as a reflection of physical variability in the environment. Additionally, other factors could change or diminish risks not accounted for here – e.g., other industrial or natural sources of nPN could exacerbate urban air concentrations, or degradation daughter products could pose unforeseen threats. Consequently such preliminary modeling does not *substitute* for rigorous environmental risk assessment. However, screening model predictions provide a rational and quantitative basis to calibrate the risks associated with commercial chemical use and to direct the further study of compounds that would most likely threaten air and water supplies. Similar efficient approaches should be developed to estimate the chemical exposures of service station workers, consider the impacts of proposed gasoline additives to surface water supplies, and evaluate large scale transport and long term persistence of proposed gasoline additives in soils and biota of the natural environment. A suite of well-designed screening models would enable future industry and regulatory decision makers to rapidly, effectively identify fuel additives which are likely to be benign to the environment and human health. As a consequence, severe environmental and societal costs associated with the use of such fuel additives as tetra-ethyl lead and MTBE might be avoided in the future.

References

(1) Abraham, M. H.; Poole, C. F.; Poole, S. K., *Classification of stationary phases and other materials by gas chromatography.* Journal of Chromatography A **1999**, *842*, 79-114.

(2) Advanced Pharma Algorithms Inc., Absolv^{TM}, 2001 ADME Boxes version 2.2.

(3) Bondi, A., *van der Waals volumes and radii.* Journal of Physical Chemistry **1964**, *68*, 441-451.

Appendix A
Estimation of hydrogen-bond complex energies using *ab initio* computational methods: current strategies

A.1. Introduction

As described in the main thesis, linear solvation energy relationships (LSERs) are powerful tools for environmental fate modeling, as long as contaminant solute parameters are available or estimable. The LSER solute parameters V_x and R_2 can be estimated using group contribution methods (*1,2*), and a method to estimate the LSER solute π_2^H parameter was developed in chapter 4. Consequently, methods to accurately estimate the only remaining LSER solute parameters, α_2^H and β_2^H, would valuably extend LSERs to novel solutes. As discussed in chapter 4, the α_2^H and β_2^H parameters were developed directly from measured 1:1 hydrogen-bond (H-bond) complexation free energies using linear free energy relationships. Consequently, in Appendix A I examine the feasibility of predicting H-bond complex energies, which might then be used to extrapolate estimates of α_2^H and β_2^H. We will find that computing H-bonding interactions between molecules either in solution or a gas phase poses a current research challenge.

Following Abraham and coworkers (*3*), a 1:1 H-bond complexation reaction in tetrachloromethane solvent was defined as:

$$\mathrm{AH} + \mathrm{B} \xleftrightarrow{K^H} \mathrm{AH:B} \qquad \text{(A-1)}$$

where AH:B is a complex of the H-bond donor, AH, and the H-bond acceptor, B. The Gibbs free energy of the H-bond complexation reaction, ΔG^H_{rxn}, was defined as:

$$\Delta G^H_{rxn} = -RT\ln\left(K^H\right) \equiv -RT\ln\left(\frac{[\mathrm{AH:B}]}{[\mathrm{AH}][\mathrm{B}]}\right) \qquad \text{(A-2)}$$

where R is the molar gas constant (J mol^{-1} K^{-1}), T is temperature in Kelvins, and [AH:B], [AH], and [B] are the H-bonded complex and monomeric species liquid phase concentrations (mol/L), and K^H is the complexation equilibrium constant. By devising linear free energy relationships (LFERs) between sets of measured log K^H values, Abraham and coworkers constructed an empirical solute H-bond acidity parameter, α_2^H (*3*). In LSERs, the α_2^H scale effectively describes solute-solvent H-bonding contributions to gas-liquid partitioning free energies for a wide set of nonionic H-bond donor solutes in a range of solvent media (*4*). Consequently, computational approaches to predict K^H values for untested H-bond donors with a reference H-bond acceptor would allow α_2^H estimates to be made for novel solutes; and this would significantly aid efforts to model the environmental fate of unstudied compounds.

Quantitative predictions of H-bond complexation energies are partly difficult because electrostatics, electron exchange, and electron correlation all contribute importantly to the weak electronic binding of the complex (*5*). In particular, correct treatment of electron correlation effects is computationally expensive (*6*), and currently available methods might not predict such weak intermolecular interactions (i.e., 5 to –30 kJ/mol for nonionic solutes in tetrachloromethane solvent (*3*)) with very good accuracy unless one resorts to prohibitively expensive levels of theory. For example, computed atomization energies of small molecules had errors of order ~9 kJ/mol using the B3LYP/6-311+G(3df,2p) method or ~5 kJ/mol using G2 theory with the same test set (*7*). Similarly, typical errors in computed proton affinities of a test set of small molecules were ~5 kJ/mol using either B3LYP/6-311+G(3df,2p) or G2 theory (*6*). By comparison, accurate prediction of nonionic H-bond complexation energies would require standard errors of 1-2 kJ/mol.

In addition to the computationally challenging panoply of electronic effects, rotational and vibrational mode energy corrections may be important. The vibrational energetic changes associated with H-bond complexation may be several kJ/mol (*8-10*). Additionally, non-negligible anharmonicity in the H-bond modes generally needs to be considered (*5,8,11,12*). Molecular rotational and vibrational partition functions are difficult to estimate for (liquid phase) solvated species, so modeling these aspects of the problem may be nontrivial.

Finally, a computational artifact known as basis set superposition error (BSSE) may require consideration. BSSE arises from artificial refinement of the wavefunction estimate in regions where molecular orbital basis functions of the two-molecule complex overlap. Depending on the method and basis set size, BSSE can easily lead to several kJ/mol inaccuracies in H-bond complexation calculations (*5*). A standard correction to BSSE is the counterpoise method (*6*), which usually increases the required computational effort several-fold. Consequently, if high level methods are employed to model the H-bond complex, the counterpoise approach constitutes an expensive correction.

My objective was to develop computational estimates of K^H for nonionic complexes in tetrachloromethane solvent. Using calculated H-bond equilibrium constants for a set of proton donor species to a small reference proton acceptor, one might generate a LFER to extrapolate α_2^H values for novel (proton donor) solutes. In the interest of illuminating research needs which could focus future efforts, I have presented a set of illustrative test calculations which are practically implementable for small to medium-sized nonionic molecules (i.e., up to ~20 heavy atoms).

A.2. Theoretical considerations

Following Ben-Naim and Marcus (*13*), the chemical potential, μ_i^s, of a solvated nonionic solute, i, in liquid, s, can be described as:

$$\mu_i^s = \Delta g_{solv}(i,s) + kT\ln\rho_i + kT\ln\Lambda_i^3 - kT\ln q_i \qquad \text{(A-3)}$$

where $\Delta g_{solv}(i,s)$ is the Gibbs free energy of transferring the solute from a fixed position in vacuum to a fixed position in the liquid phase, ρ_i is the number of the solute molecules per unit volume, Λ_i is the de Broglie wavelength of the solute, and q_i is the internal partition function of the solute *in vacuo*. The kTln q_i free energy term therefore relates the electronic, vibrational, and rotational energy contributions of the solute molecule in an ideal gas. It was assumed that the vibrational and rotational states of the solute did not change significantly upon placement in solution; consequently rotational and vibrational energies were treated using gas phase approximations.

Recall that the Gibbs free energy (equation A-2) for a 1:1 complexation reaction in carbon tetrachloride solvent was:

$$\Delta g^{H}_{rxn} = -kT\ln\left(\frac{\rho_{AH:B}}{\rho_{AH}\,\rho_{B}}\right) \tag{A-4}$$

where $N_A \Delta g^{H}_{rxn} = \Delta G^{H}_{rxn}$. Employing equation A-4 with the chemical potential expression (equation A-3) results in an equilibrium formula for Δg^{H}_{rxn} in terms of the energy differences of more fundamental processes:

$$\begin{aligned}\Delta g^{H}_{rxn} = {} & \left(\Delta g_{solv}(AH:B,s) - \Delta g_{solv}(AH,s) - \Delta g_{solv}(B,s)\right) + \\ & \left(kT\ln\Lambda^{3}_{AH:B} - kT\ln\Lambda^{3}_{AH} - kT\ln\Lambda^{3}_{B}\right) - \left(kT\ln q_{AH:B} - kT\ln q_{AH} - kT\ln q_{B}\right)\end{aligned} \tag{A-5}$$

Each of these energy terms will be considered in turn. The de Broglie wavelength is given by the usual expression *(14)*:

$$\Lambda_i = \left(\frac{h^2}{2m_i kT\pi}\right)^{\frac{1}{2}} \tag{A-6}$$

where h is Planck's constant, and m_i is the mass of the solute. The solute internal partition function can be expressed in terms of electronic, vibrational, and rotational partition functions, which are assumed separable:

$$q_i = q_{i,elec}\,q_{i,vib}\,q_{i,rot} \tag{A-7}$$

The electronic partition function was assumed to reflect only the energy of the ground state (ε_o):

$$q_{i,elec} = \Omega(\varepsilon_o)e^{-\frac{\varepsilon_o}{kT}} \tag{A-8}$$

where $\Omega(\varepsilon_o)$ is the ground state degeneracy *(14)*. The vibrational partition function expression assumed that nuclear motion could be treated as independent harmonic oscillations *(14)*:

$$q_{i,vib} = \prod_{j=1}^{n_{modes}} \frac{e^{-\frac{h\upsilon_j}{2kT}}}{1-e^{-\frac{h\upsilon_j}{kT}}} \tag{A-9}$$

where the ν_j represents a vibrational mode, and the zero-point energy correction is implicitly included. The harmonic oscillator approximation was highly suspect, since hydrogen-bonded complexes are known to exhibit significant anharmonicity. The rotational partition function was treated using the classical expression (*14*):

$$q_{i,rot} = \left(\frac{I_a I_b I_c \pi}{\sigma_{sym}^2}\right)^{\frac{1}{2}} \left(\frac{8\pi^2 kT}{h^2}\right)^{\frac{3}{2}} \tag{A-10}$$

where I_a, I_b, and I_c represent the moments of inertia of the solute molecule along the three principle axes, and σ_{sym} is the rotational symmetry number of the molecule.

These modeling approximations (equations A-6 to A-10) allowed electronic, vibrational, rotational, translational, and solvation energetic effects to be practically computed and applied to the Δg_{rxn}^{H} term (equation A-5).

A.3. Computational approach

Twelve hydrogen-bond donor solutes which exhibit a range of hydrogen-bonding strengths and moiety types were selected. Dimethylsulfoxide (DMSO) was chosen as a reference hydrogen-bond acceptor, since it is a strong hydrogen-bond base of relatively small size. Computation of AH:DMSO complex binding energies should generate a better signal to error result than complexes using weaker bases that could have been chosen (e.g., water). Since DMSO is also reasonably small, it added affordable computational overhead to wavefunction calculations of the hydrogen-bonded complex. Measured AH:DMSO hydrogen-bond complexation energies were estimated from the linear free energy relationship between solute α_2^H values and log K^H for DMSO devised by Abraham et al. (*3*):

$$\log K_{i,DMSO}^{H} = 5.748\alpha_{A}^{H} - 1.098 \tag{A-11}$$

where log K_A^H is a general solute hydrogen-bond donor scale, and had an estimated error of 0.096 in the log $K_{i,DMSO}^{H}$ values (or 0.55 kJ/mol in the ΔG_{rxn}) for the set of data (n = 51) which these workers considered. Δg_{rxn} values were computed for each hydrogen-bond acid complexed with DMSO in tetrachloromethane solvent by application of equation A-5; the relevant energy and partition function terms were evaluated as follows.

All wavefunction and partition function calculations were performed with Gaussian98 (*15*) using "tight" SCF (self-consistent field) convergence criteria. Solute reactant and complex nuclear geometries were optimized to a low energy structure using the hybrid density functional theory approach, B3LYP/6-31+G(d,p). Frequency calculations were also conducted at this level of theory in order to verify that a stable structure had been found, and to calculate solute frequencies, rotational energy, and translational energy. Single point calculations at these fixed conformations were conducted using B3LYP/6-311++G(2df,2pd), in the presence of a PCM (*16*) dielectric field with dielectric constant = 2.23 (corresponding to the measured dielectric constant of carbon tetrachloride solvent (*17*)). Although they added significant computational overhead, diffuse basis functions were considered necessary to capture the long range electronic interactions of the hydrogen bond (*6*). PCM-computed solvation energies were considered a suitable estimate of Δg_{solv}(solute, carbon tetrachloride), since carbon tetrachloride is a nonpolar and non-hydrogen-bonding solvent (*18,19*). BSSE corrections were not employed due to time constraints; however this effect is diminished for large basis sets (*6*). For example, Zhou and coworkers found BSSE corrections of 1 to 4 kJ/mol using B3LYP with the 6-31++G(2d,2p) or larger basis sets for hydrogen-bonding systems (*20,21*).

A.4. Results and conclusions

Computed AH:DMSO complexation ΔG_{rxn} values overestimated measured values by an average of +25 ± 9 kJ/mol for the solute set (Table A-1), that is, computed energies did not predict that complexation is a very favorable process. Additionally, the correlation between computed and measured ΔG_{rxn} values was poor ($r^2 = 0.61$). Not only the magnitude, but also the ordering of ΔG_{rxn} magnitudes for different solutes was incorrectly predicted. For example, cyanic acid was predicted to have the most favorable complexation energy with DMSO, but it ranks as only the 8th lowest measured ΔG_{rxn} out of the set of 12 H-bond donor solutes. The binding energy, defined as:

$$\Delta E_{bind} = -RT\ln\frac{q_{AH:B,elec}}{q_{AH,elec}q_{B,elec}} \tag{A-12}$$

$$= -RT\ln\left(\frac{\Omega(\varepsilon_{o,AH:B})}{\Omega(\varepsilon_{o,AH})\Omega(\varepsilon_{o,B})}\right) + (\varepsilon_{o,AH:B} - \varepsilon_{o,AH} - \varepsilon_{o,B})$$

was a better indicator of measured ΔG_{rxn} values (correlation $r^2 = 0.76$). This correlation might result from the fact that inclusion of temperature corrections may invite significant errors due to anharmonic effects. ΔG_{therm} values, which reflected the net vibrational, translational, and rotational free energy changes associated with formation of the complex:

$$\Delta G_{therm} = -RT\ln\frac{q_{AH:B,vib}q_{AH:B,trans}q_{AH:B,rot}}{q_{AH,vib}q_{AH,trans}q_{AH,rot}q_{B,vib}q_{B,trans}q_{B,rot}} \tag{A-13}$$

consistently reflected an energetic cost ($\Delta G_{therm} > 0$). The change in solvation energy on complexation, defined as:

$$\Delta\Delta G_{solv} = \Delta G_{solv}(AH:B,s) - \Delta G_{solv}(AH,s) - \Delta G_{solv}(B,s) \tag{A-14}$$

was significant and positive (+3 to 13 kJ/mol) for all complexes under consideration. This was an unexpected result, since complex formation should decrease the size of the required solvent cavity, which is energetically favorable.

Future work would probably benefit from explicit treatment of anharmonicity energies and BSSE corrections. Huang and MacKerrell suggest that these corrections, together with theoretical

Table A-1. Measured and calculated AH:DMSO complexation energies for 12 test solutes (kJ/mol)

	measured[a]	calculated[b]						
H-bond donating solute	ΔG_{rxn}	ΔG_{rxn}	ΔE_{bind}	$\Delta\Delta G_{solv}$	ΔG_{therm}	ΔG_{vib}	ΔG_{tran}	ΔG_{rot}
propyl hydrosulfide	6.3	25.4	-10.8	7.9	28.3	-31.5	42.3	20.0
thiophenol	4.0	25.0	-17.3	8.2	34.1	-28.2	42.9	21.9
dichloromethane	2.0	31.7	-19.8	12.1	39.4	-20.3	42.5	19.7
aniline	-2.3	33.1	-21.1	13.4	40.8	-21.3	42.6	21.9
ethanol	-4.6	22.7	-33.4	10.7	45.4	-12.0	41.2	18.7
water	-5.2	18.1	-34.9	6.8	46.2	1.5	38.7	8.5
ammonia	-7.8	29.8	-17.4	6.4	40.8	-5.8	38.5	10.5
cyanic acid	-12.1	-7.7	-56.2	2.9	45.6	-7.7	41.1	14.7
phenol	-13.4	14.2	-40.6	11.2	43.7	-18.4	42.7	21.8
thiocyanic acid	-18.3	14.1	-38.2	8.6	43.7	-12.4	41.8	16.8
4-nitrophenol	-20.6	1.1	-49.1	8.1	42.1	-21.4	43.2	22.8
2,2,2-trifluoroacetic acid	-24.9	-6.8	-62.6	9.3	46.6	-16.1	43.0	22.2

[a] "Measured" ΔG_{rxn} values were found using equation A-2, based on log K^H values given by equation A-11.

[b] Calculated ΔG_{rxn} and other listed (calculated) quantities were determined using equations A-5, A-12, A-13, and A-14.

treatment of electron correlation effects, could result in gas phase hydrogen bond complexation energy estimates of 4 kJ/mol accuracy (*5*). Based on the promising results of this previous work, it is reasonable to expect that accurate calculation of log K^H values for nonionic solutes in organic solvents (for which there is a larger body of accurate validation data) could be developed.

A.5. References

(1) Abraham, M. H.; McGowan, J. C., *The use of characteristic volumes to measure cavity terms in reversed phase liquid chromatography.* Chromatographia **1987**, *23*, 243-246.

(2) Platts, J. A.; Abraham, M. H.; Butina, D.; Hersey, A., *Estimation of molecular linear free energy relationship descriptors by a group contribution approach.* Journal of Chemical Information and Computer Sciences **1999**, *39*, 835-845.

(3) Abraham, M. H.; Grellier, P. L.; Prior, D. V.; Duce, P. P.; Morris, J. J.; Taylor, P. J., *Hydrogen bonding. Part 7. A scale of solute hydrogen-bond acidity based on logK values for complexation in tetrachloromethane.* Journal of the Chemical Society. Perkin Transactions 2 **1989**, 699-711.

(4) Abraham, M. H.; Poole, C. F.; Poole, S. K., *Classification of stationary phases and other materials by gas chromatography.* Journal of Chromatography A **1999**, *842*, 79-114.

(5) Huang, N.; MacKerell, A. D., *An ab initio quantum mechanical study of hydrogen-bonded complexes of biological interest.* Journal of Physical Chemistry A **2002**, *106*, 7820-7827.

(6) Cramer, C. J. *Essentials of Computational Chemistry. Theories and Models*; John Wiley & Sons, Inc.: New York, NY, **2002**.

(7) Curtiss, L. A.; Raghavachari, K.; Trucks, G. W.; Pople, J. A., *Gaussian-2 theory for molecular-energies of 1st-row and 2nd-row compounds.* Journal of Chemical Physics **1991**, *94*, 7221-7320.

(8) Sim, F.; St-Amant, A.; Papai, I.; Salahub, D. R., *Gaussian density functional calculations on hydrogen-bonded systems*. Journal of the American Chemical Society **1992**, *114*, 4391-4400.

(9) Rappe, A. K.; Bernstein, E. R., *Ab initio calculations of nonbonded interactions: are we there yet?* Journal of Physical Chemistry A **2000**, *104*, 6117-6128.

(10) Yilgor, E.; Yilgor, I.; Yurtsever, E., *Hydrogen bonding and polyurethane morphology. I. Quantum mechanical calculations of hydrogen bond energies and vibrational spectroscopy of model compounds*. Polymer **2002**, *43*, 6551-6559.

(11) Schreiber, V. M.; Shchepkin, D. N., *Solvent effect on the vibrational spectrum of a hydrogen-bonded complex*. Journal of Molecular Structure **1992**, *270*, 481-490.

(12) Feyereisen, M. W., D; Dixon, D. A., *Hydrogen bond energy of the water dimer*. Journal of Physical Chemistry **1996**, *100*, 2993-2997.

(13) Ben-Naim, A.; Marcus, Y., *Solvation thermodynamics of nonionic solutes*. Journal of Physical Chemistry **1984**, *81*, 2016-2027.

(14) Hill, T. L., *Chapter 9. Ideal Polyatomic Gas*, In *An Introduction to Statistical Thermodynamics*; Dover Publications, Inc.: New York, NY, **1986**; pp 161-176.

(15) Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Zakrzewski, V. G.; Montgomery Jr., J. A.; Stratmann, R. E.; Burant, J. C.; Dapprich, S.; Millam, J. M.; Daniels, A. D.; Kudin, K. D.; Strain, M. C.; Farkas, O.; Tomasi, J.; Bar, V.; Cossi, M.; Cammi, R.; Mennucci, B.; Pomelli, C.; Adamo, C.; Clifford, S.; Ochterski, J.; Petersson, G. A.; Ayala, P. Y.; Cui, Q.; Morokuma, K.; Malick, D. K.; Rabuck, A. D.; Raghavachari, K.; Foresman, J. B.; Cioslowski, J.; Ortiz, J. V.; Stefanov, B. B.; Liu, G.; Liashenko, A.; Piskorz, P.; Komaromi, I.; Gomperts, R.; Martin, R. L.; Fox, D. J.; Keith, R.; Al-Laham, M. A.; Peng, C. Y.; Nanayakkara, A.; Gonzalez, C.; Challacombe, M.; Gill, P. M. W.; Johnson, B.; Chen, W.; Wong, M. W.; Andres, J. L.; Gonzalez, C.; Head-Gordon, M.; Replogle, E. S.; Pople, J. A., *Gaussian 98, revision a.6*; Gaussian, Inc.: Pittsburgh, PA, **1998**.

(16) Cossi, M.; Barone, V.; Cammi, R.; Tomasi, J., *Ab initio study of solvated molecules: a new implementation of the Polarizable Continuum Model*. Chemical Physics Letters **1996**, *255*, 327.

(17) Chemical Rubber Company, *Physical Constants of Organic Compounds*, In *Handbook of Chemistry and Physics*; 77th ed.; Lide, D., Ed.; CRC Press, Inc: Boca Raton, FL, **1997**.

(18) Milischuk, A.; Matyushov, D. V., *Dipole solvation: nonlinear effects, density reorganization, and the breakdown of the Onsager saturation limit*. Journal of Physical Chemistry A **2002**, *106*, 2146-2157.

(19) Milischuk, A.; Matyushov, D. V., *On the validity of dielectric continuum models in application to solvation in molecular solvents*. Journal of Chemical Physics **2003**, *118*, 1859-1862.

(20) Fu, A.; Du, D.; Zhou, Z., *Density functional theory study of the hydrogen bonding interaction of 1:1 complexes of formamide with water*. Journal of Molecular Structure **2003**, *623*, 315-325.

(21) Zhou, Z.; Shi, Y.; Zhou, X., *Theoretical studies on the hydrogen bonding interactions of complexes of formic acid with water*. Journal of Physical Chemistry A **2004**, *108*, 813-822.

Appendix B
Monte Carlo simulation C++ code used for the stochastic subsurface transport calculations

The main program code **montecarlo.c** and ancillary codes (included files) are given here. Note that the **model.c** and **lustmodule.c** must be compiled together with **montecarlo.c** in order to create a viable executable. Otherwise, standard libraries have been used and the coding is fairly simple. The **montecarlo.c** program is designed for flexibility and could be readily adapted to stochastic analysis of other models having different stochastic properties.

Required input data files are as follows:
LUSTs.dat lists the LUFT pdf (probability distribution function) data corresponding to Figure 2-2.
parm_var.dat lists the type of sampling distribution associated with each stochastic parameter:

$flag_1$
$flag_2$
$flag_3$
...

where each $flag_i$ corresponds to a stochastic parameter, i, and is designated as either: constant (0), uniform (1), normal (2), lognormal (3), exponential (4), or a designed separate module (5+)
parms.dat lists the properties of each stochastic variable on a line, as:

μ_1	M_1	S_1
μ_2	M_2	S_2
μ_3	M_3	S_3

...

The interpretation of μ_i, M_i, and S_i depend on the corresponding **parm_var.dat** flag for each parameter:
If $flag_i = 0$, then μ_i = constant value of i, and M_i and S_i are not used.
If $flag_i = 1$, then μ_i is not used, M_i is the cdf (cumulative distribution function) floor, and S_i is the cdf ceiling.
If $flag_i = 2$, then μ_i is not used, M_i = mean of i, S_i = standard deviation of i.
If $flag_i = 3$, then μ_i is not used, M_i = log normal mean of i, S_i = log normal std deviation of i.
If $flag_i = 4$, then μ_i is not used, M_i = exp lambda parameter, and S_i is not used.
If $flag_i$ = 5+, then μ_i, M_i, and S_i are tailored to whatever the designed module takes.
The reason I have included the μ_i parameter (which is not used for stochastic cases) is that this allows the user to quickly switch an interesting parameter from "constant" to "stochastic", or vice versa, by simply adjusting the corresponding **parm_var.dat** flag.

Output files include the following:
model_complete_cdf.out is the raw cdf of the model output (e.g., predicted C_{well} value), having a total length = Nsamples.
model _cdf.out is a smoothed cdf of the model output, having length = Nout.

model _pdf.out is a smoothed pdf of the model output, having length = Nout.

parminvcdfs.out is a computed table of inverse cdfs corresponding to the set of stochastic parameters as designated by **parms.dat** and **parm_var.dat**. Once this table has been generated for a given set of parameters at the resolution set by Ncdf, it does not need to be recomputed in subsequent runs using the same parameters. Computational generation of the table is slow, but once the table has been created, subsequent sampling calculations are a fairly efficient table look-up procedure. The user may also examine the parminvcdfs.out file to check whether stochastic input parameter distributions are being correctly generated.

montecarlo.c

```
// Straight Monte Carlo. Input a function called MODEL and systematically
// vary PARMS, each having a pdf of one of the types listed. All that is
// needed to consider a new pdf is a file or function mapping a uniform cdf
// to the inverse cdf of interest, for 10000 points (0.00005 to 0.99995) of
// the uniform cdf to 4 significant figures.

#include <iostream>
#include <iomanip>
#include <string>
#include <cstdlib>      // for srand() fn
#include <math.h>
#include <time.h>       // for time() fn
#include <fstream>      // file io
#include "model.h"      // output model function file
#include "lustmodule.h"  // add module to estimate LUST density inv cdf from data
#define PI 3.14159265

using namespace std;

void READ_CDF(string, int, double *);
int COUNT_NPARMS(string);
void READ_PARMS(string, int, double **);
void READ_PARMTRANS(string, int, int *);
void WRITE_PARMICDF(string, double **, int, int);
double ERF(double);
double DERFDX(double);
double INVERF(double, double);
void CDF2PDF(int, int, double *, double **, double **);
void QUICKSORT(int, int, double *);
int PARTITION(int, int, double *);
void SWAP(double *, double *);
void TRISMOOTH(int, double *);
void CALC_DERIVATIVES(int, int, double *, double **, double **);
void CALC_LOG_DERIVATIVES(int, int, double *, double **, double **);
void WRITE_DF(string, int, double *);
void WRITE_DF(string, int, double **);

int main()
{
```

```
int Nsamples = 1000000; // number of samples to be drawn, always divisible by 1E2
int Ncdf = 10000;          // resolution of individual parameter cdfs, always < 1E6
int Nout = 200;            // resolution of output pdf
int Nchunk = Nsamples/10; // size of double vector "chunks" written to output
int Nprint_freq = Nchunk;    // frequency of on-screen sampling reports

int timestart = time(0);

string outputfile = "model_complete_cdf.out"; // model whole cdf results
string outputfile2 = "model_cdf.out";          // model smoothed cdf results
string outputfile3 = "model_pdf.out";          // model smoothed pdf results
int Nparms = 0;
string parmfile = "parms.dat";                 // mu, sigma vals for each parm
string parmvarfile = "parm_var.dat";           // vector of integer switches
  // indicating whether parms are constant (0), uniform (1), normal (2),nm
  // lognormal (3), or exponential (4), or designed separate module (5+)
string parmicdffile = "parminvcdfs.out";       // matrix of cdf vectors for parms

Nparms = COUNT_NPARMS(parmfile);
cout << "\nFound " << Nparms << " parameters."
    << "\n\nSetting up input parameter inverse cumulative"
    << " distribution functions...";
double **pstat;                                // parm statistics
pstat = new double*[Nparms];
int *v;                                        // parm switch values (0-3)
v = new int[Nparms];
for (int i = 0; i < Nparms; i++)
{
  v[i] = 0;
  pstat[i] = new double[3];
  for (int j = 0; j < 3; j++)                  // 1st column, mu; 2nd column, sigma
    pstat[i][j] = 0.0;
}

READ_PARMS(parmfile, Nparms, pstat);           // assign parm values
READ_PARMTRANS(parmvarfile, Nparms, v); // assign switches, "transmission"

double **parmicdf;                             // create parm inv cdf matrix
parmicdf = new double*[Nparms];
for (int i = 0; i < Nparms; i++)               // rows, parm type
{
  cout << "\nCreating parameter " << i+1 << " inverse cdf.";
  parmicdf[i] = new double[Ncdf];
  for (int j = 0; j < Ncdf; j++)               // columns, inv cdf value
    parmicdf[i][j] = 0.0;                      // (initialize icdfs for safety)
  if (v[i] == 0)                               // const parm case
  {
    for (int j = 0; j < Ncdf; j++)
        parmicdf[i][j] = pstat[i][0];          // always take mean val
  }
  else if (v[i] == 1)                          // uniform distribution case
  {
    for (int j = 0; j < Ncdf; j++)
          parmicdf[i][j] = pstat[i][1] + ((j+0.5)/Ncdf)*(pstat[i][2] - pstat[i][1]);
  }
  else if (v[i] == 2)                                  // normal distribution case
  {
    for (int j = 0; j < Ncdf; j++)
          parmicdf[i][j] = pstat[i][1] + sqrt(2.0)*pstat[i][2]*INVERF(2.0*(j+0.5)/Ncdf - 1.0, 0.5);
  }
  else if (v[i] == 3)                                  // log normal distribution case
  {
    for (int j = 0; j < Ncdf; j++)
    {
          parmicdf[i][j] = exp(pstat[i][1] + sqrt(2.0)*pstat[i][2]*INVERF(2.0*(j+0.5)/Ncdf - 1.0, 0.5));
    }
  }
  else if (v[i] == 4)                                  // exponential distribution case
  {
    for (int j = 0; j < Ncdf; j++)
          parmicdf[i][j] = - (pstat[i][1])*log(1.0 - (j+0.5)/Ncdf);
  }
  else if (v[i] == 5)                                  // "designed module" case
  {
    LUSTMODULE(parmicdf, i, Ncdf);
  }
  else
  {
    cerr << "\nERROR. Parameter switch " << i << " in file "
             << parmvarfile.c_str() << " is invalid.\nExiting.\n\n";
    exit(1);
  }
}
cout << "\nDone.";

cout << "\n\nWriting parameter inverse cdf values to file "
     << parmicdffile.c_str() << ".";
WRITE_PARMICDF(parmicdffile, parmicdf, Nparms, Ncdf);

int *MCunipdf;                              // MC uniform pdf,
MCunipdf = new int[Ncdf];                   // arranged as bins of the cdf
for (int i = 0; i < Ncdf; i++)
  MCunipdf[i] = 0;
double **MCparmpdf;                          // MC parm pdf arranged as bins
MCparmpdf = new double*[Nparms];
for (int i = 0; i < Nparms; i++)
{
  MCparmpdf[i] = new double[Ncdf];
  for (int j = 0; j < Ncdf; j++)
    MCparmpdf[i][j] = 0.0;
}
```

```
double *MCoutcdf;                      // MC output cdf vector
MCoutcdf = new double[Nchunk];
for (int i = 0; i < Nchunk; i++)
 MCoutcdf[i] = 0.0;
double *modelparms;                    // MODEL input vector
modelparms = new double[Nparms];
for (int i = 0; i < Nparms; i++)
 modelparms[i] = 0.0;

cout << "\n\nSampling random configurations..";
srand(time(0));                        // seed the rand fn
for (int i = 0; i < Nsamples; i++)     // Monte Carlo simulation start
{
 int sample = 0;

 if (i%Nprint_freq == Nprint_freq-1)
  cout << "\nNow drawing sample " << i+1 << ".";

 int j = i%Nchunk;                     // counter from [0 to Nchunk-1]

 for (int k = 0; k < Nparms; k++)
 {
  while (v[k] == 0)                    // skip over "uniform distribution" cases
     k++;
  if (k >= Nparms)
     break;
                    // generate an integer [0 to Ncdf-1] but note that RAND_MAX = 32767,
                    // so if Ncdf ~ 10000 or more, must split into "big" and "small" parts:
  sample = (Ncdf/100)*(rand()%100) + rand()%(Ncdf/100);
  modelparms[k] = parmicdf[k][sample];
                                       // PARM SPECIAL CONDITIONS FROM TABLE 2-2
  if (k == 0)                          // Vf
     while (modelparms[k] < 0.0379)   // if less than 10 gal
     {                                 // then resample
      sample = (Ncdf/100)*(rand()%100) + rand()%(Ncdf/100);
      modelparms[k] = parmicdf[k][sample];
     }
  if (k == 1)                          // Qwell
     while (modelparms[k] > 27000.0 || modelparms[k] < 108.0)
     {                                 // if > 5000 gal/min or
      sample = (Ncdf/100)*(rand()%100) + rand()%(Ncdf/100);
      modelparms[k] = parmicdf[k][sample];  // < 20 gal/min, resample
     }
  if (k == 2)                          // v
     while (modelparms[k] < 0.01)      // if less than 0.01 m/day
     {                                 // then resample
      sample = (Ncdf/100)*(rand()%100) + rand()%(Ncdf/100);
      modelparms[k] = parmicdf[k][sample];
     }
  if (k == 3)                           // fom
     while (modelparms[k] < 0.0001)     // if less than 0.0001
```

```
       {                                 // then resample
         sample = (Ncdf/100)*(rand()%100) + rand()%(Ncdf/100);
         modelparms[k] = parmicdf[k][sample];
       }
    if (k == 4)                           // S
       while (modelparms[k] < 0.05 || modelparms[k] > 0.95) // if less than
       {                                  // 0.05 or greater than 0.95, resample
         sample = (Ncdf/100)*(rand()%100) + rand()%(Ncdf/100);
         modelparms[k] = parmicdf[k][sample];
       }
    if (k == 5)                           // h
       while (modelparms[k] < 0.05)       // if less than 0.05 m
       {                                  // then resample
         sample = (Ncdf/100)*(rand()%100) + rand()%(Ncdf/100);
         modelparms[k] = parmicdf[k][sample];
       }
    if (k == 7)                           // LUST density
       while (5000.0 <
(1000000.0*PI*25.0*modelparms[2]*0.25)/(2.0*modelparms[7]*modelparms[1]) || 300.0 >
(1000000.0*PI*25.0*modelparms[2]*0.25)/(2.0*modelparms[7]*modelparms[1]))
                                          // if Lx < 300 m or Lx > 5000
       {                                  // then resample
         sample = (Ncdf/100)*(rand()%100) + rand()%(Ncdf/100);
         modelparms[k] = parmicdf[k][sample];  // assumes H = 25 m, phi = 0.25
       }
    if (k == 8)                            // ax
       while (modelparms[k] >
0.10*(1000000.0*PI*25.0*modelparms[2]*0.25)/(2.0*modelparms[7]*modelparms[1]) ||
modelparms[k] < 2.0)                      // if greater
       {                                  // than 10% of Lx or less than 2 m,
         sample = (Ncdf/100)*(rand()%100) + rand()%(Ncdf/100); // then resample
         modelparms[k] = parmicdf[k][sample];
       }
  }

  for (int k = 0; k < Nparms; k++)
    if (v[k] == 0)
       modelparms[k] = parmicdf[k][0];     // assign uniform distribution cases

  MCoutcdf[j] = MODEL(modelparms);        // run MODEL fn
  if (j == Nchunk-1)
  {
    ofstream outfile;
    if (j == i)
       outfile.open(outputfile.c_str(), ios::out);  // writing first set
    else
       outfile.open(outputfile.c_str(), ios::app);  // subsequent sets
    if (!outfile)
    {
       cerr << "\nFile '" << outputfile.c_str() << "' could not be opened."
          << "\nExiting.\n\n";
```

```
      exit(1);
    }

    outfile.precision(5);
    outfile.setf(ios::showpoint);
    for (int k = 0; k < Nchunk; k++)
      outfile << MCoutcdf[k] << endl;
    outfile.close();
    for (int k = 0; k < Nchunk; k++)                 // reinitialize for safety
      MCoutcdf[k] = 0.0;
  }
}
cout << "\nDone.";
cout << "\n\nSorting model output and writing to files "
   << outputfile2 << " and " << outputfile3 << ".";

for (int i = 0; i < Nparms; i++)
{
  delete [] pstat[i];
  delete [] parmicdf[i];
  delete [] MCparmpdf[i];
}
delete [] v;
delete [] MCunipdf;
delete [] MCoutcdf;
delete [] modelparms;

double *modelcdf;                                  // MC entire output cdf vector
modelcdf = new double[Nsamples];
for (int i = 0; i < Nsamples; i++)
  modelcdf[i] = 0.0;

double **red_modelcdf;                             // reduced (smoothed) output cdf vector
red_modelcdf = new double*[2];
double **red_modelpdf;                             // reduced (smoothed) output pdf vector
red_modelpdf = new double*[2];
for (int i = 0; i < 2; i++)
{
  red_modelcdf[i] = new double[Nout];
  red_modelpdf[i] = new double[Nout];
  for (int j = 0; j < Nout; j++)
  {
    red_modelcdf[i][j] = 0.0;
    red_modelpdf[i][j] = 0.0;
  }
}
cout << "\n";
READ_CDF(outputfile, Nsamples, modelcdf); // now read in entire model cdf

// sort entire model output cdf and convert to reduced cdf and reduced pdf
CDF2PDF(Nsamples, Nout, modelcdf, red_modelcdf, red_modelpdf);
```

```
  WRITE_DF(outputfile, Nsamples, modelcdf);  // write sorted model cdf
  WRITE_DF(outputfile2, Nout, red_modelcdf);   // write smoothed model cdf
  WRITE_DF(outputfile3, Nout, red_modelpdf);   // write smoothed model pdf
  cout << "Done.";

  delete [] modelcdf;
  for (int i = 0; i < 2; i++)
  {
    delete [] red_modelcdf[i];
    delete [] red_modelpdf[i];
  }

  int timeend = time(0);
  cout.precision(3);
  cout << "\n\nTotal run time = " << timeend-timestart << " seconds = "
      << static_cast<double>(timeend-timestart)/60.0 << " minutes = "
      << static_cast<double>(timeend-timestart)/3600.0 << " hours.\n";
  cout << "\a\a\a";                                    // sound system bell “Done!”

  return 0;
}

void READ_CDF(string filename, int N, double *df)
{
  ifstream infile(filename.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere is file '" << filename.c_str() << "'? "
        << "I can't find it.\nExiting.\n\n";
    exit(1);
  }
  for (int i = 0; i < N; i++)
  {
    infile >> df[i];
    if (infile.fail() != 0)                   // check failbit of infile
    {
      cout << "infile.fail() = " << infile.fail() << " at point " << i << endl;
      exit(1);
    }
  }
  infile.close();
}

int COUNT_NPARMS(string filename)
{
  int N = 0;
  ifstream infile(filename.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << filename.c_str() << "'?? "
        << "I can't find it.\nExiting.\n\n";
    exit(1);
  }
  string dummy;
  while (infile >> dummy)
    N++;
  return static_cast<int>(static_cast<float>(N)/3);
}

void READ_PARMS(string filename, int N, double **parms)
{
  ifstream infile(filename.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere is file '" << filename.c_str() << "'?? "
         << "I can't find it.\nExiting.\n\n";
    exit(1);
  }
  for (int i = 0; i < N; i++)
  {
    infile >> parms[i][0];
    infile >> parms[i][1];
    infile >> parms[i][2];
  }
}

void READ_PARMTRANS(string filename, int N, int *trans)
{
  ifstream infile(filename.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere is file '" << filename.c_str() << "'?? "
         << "I can't find it.\nExiting.\n\n";
    exit(1);
  }
  for (int i = 0; i < N; i++)
    infile >> trans[i];
}

void WRITE_PARMICDF(string filename, double **cdf, int Nparms, int N)
{
  ofstream outfile(filename.c_str(), ios::out);
  if (!outfile)
  {
    cerr << "\nFile '" << filename.c_str() << "' could not be opened."
            << "\nExiting.\n\n";
    exit(1);
  }
  outfile.precision(4);
  outfile.setf(ios::showpoint);
  for (int j = 0; j < N; j++)
  {
    for (int i = 0; i < Nparms; i++)
      outfile << setw(12) << cdf[i][j];
    outfile << endl;
  }
  outfile.close();
}

double ERF(double x)          // "erf": error function (integral of gaussian)
{
  int n = 0;
  double logfact_n = 0.0;
  double value = 0.0;
  int series_precision = 30;  // default for most cases
  double sign = fabs(x)/x;
  x = fabs(x);
  if (x > 3.0/sqrt(2.0))
    series_precision = 50;
  // first term of series has factorial(0) = 1
  value = pow(-1.0,n)*pow(x,2.0*n+1.0)/((exp(logfact_n))*(2.0*n+1.0));
  for (n = 1; n < series_precision; n++)
  {
    logfact_n += log(static_cast<double>(n));          // factorial(n)
    value += pow(-1.0,n)*exp((2.0*n+1.0)*log(x) - logfact_n - log(2.0*n+1.0));
  }

  value *= sign*(2.0/sqrt(PI));
  assert(value < 1.0);
  return value;
}

double DERFDX(double x)                // d(erf(x))/dx
{ return exp(-pow(x,2.0))*2.0/sqrt(PI);}

double INVERF(double var, double x0) // find inverse erf, using Newton's method
{
  double dmin = 0.0;                   // min allowable derivative of erf, typically dmin = 0
  double tol = 1.0E-9;
  double delx = 2.0*tol;
  int err = 0;                         // set to "false"
  while (!(err) && fabs(delx) > tol)
    if (fabs(DERFDX(x0)) > dmin)
    {
      delx = (ERF(x0) - var)/DERFDX(x0);
      x0 = x0 - delx;
    }
    else err = 1;
  assert(err == 0);
  return x0;
}

void CDF2PDF(int N, int Nout, double *bigcdf, double **cdf, double **pdf)
```

```
{
 QUICKSORT(0, N-1, bigcdf);
 // TRISMOOTH(N, bigcdf);          // optionally, first smooth total cdf using a triangular window
 CALC_LOG_DERIVATIVES(N, Nout, bigcdf, cdf, pdf); // pdf output in log space
 TRISMOOTH(Nout, pdf[0]); // optionally smooth pdf using a triangular window
}

void QUICKSORT(int L, int R, double *s) // efficient sort algorithm, ~ N*log(N)
{
 int part;
 if (R <= L)
  return;
 part = PARTITION(L, R, s);
 QUICKSORT(L, part-1, s);
 QUICKSORT(part+1, R, s);
}

int PARTITION(int L, int R, double *s)
{
 int i, j;
 double value;
 i = L-1;
 j = R;
 value = s[R];
 for ( ; ; )
 {
  while (s[++i] < value);      // scan L to R for larger element
  while (s[--j] > value);      // scan R to L for larger element
  if (i >= j)                  // if pointers have crossed, end scans
   break;
  SWAP(s+i, s+j);              // exchange out of place elements
 }
 SWAP(s+i, s+R);                // place pivot element
 return i;                      // return pivot index
}

void SWAP(double *a, double *b)
{
 double temp = *a;
 *a = *b;
 *b = temp;
}

void TRISMOOTH(int N, double *dist)   // triangular window smoother
{
 double temp[3];
 temp[2] = dist[0];
 temp[1] = dist[1];
 for (int i = 2; i < N-3; i++)
 {
  temp[0] = dist[i];
  dist[i] = (temp[2] + 2.0*temp[1] + 3.0*dist[i] + 2.0*dist[i+1] + dist[i+2])/9.0;
  temp[2] = temp[1];
  temp[1] = temp[0];
 }
}

void CALC_DERIVATIVES(int N, int Nout, double *bigcdf, double **cdf, double **pdf)
{
 // use the midpoint approximation to reduce bigcdf (assume well-behaved fns)
 double del = 1.0/static_cast<double>(N);   // delta y
 int window = static_cast<int>(N/(2*Nout)); // window is deriv smoother
 //  double span = bigcdf[N-1] - bigcdf[0];     // range of cdf x values
 for (int i = 0; i < Nout; i++)
 {
  double numer = 0;
  double denom = 0;
  double avgcdfval = 0;
  for (int j = (2*i)*window; j < (2*i+2)*window; j++) // find "average
   avgcdfval += bigcdf[j]/(2.0*window);        // center" of window
  for (int j = (2*i)*window; j < (2*i+2)*window; j++) // regress from avg
  {                                            // center of window
   numer += (del*static_cast<double>(j - (2*i+1)*window))*(bigcdf[j] - avgcdfval);
   denom += pow((bigcdf[j] - avgcdfval),2.0);
  }
  cdf[0][i] = avgcdfval;
  cdf[1][i] = static_cast<double>((i+0.5)/static_cast<double>(Nout));
  pdf[0][i] = avgcdfval;
  pdf[1][i] = numer/denom;                     // slope
 }
 //  for (int i = 0; i < N; i++)                     // optional pdf normalization
 //    pdf[i] = pcf[i]*span/static_cast<double>(N);
}

void CALC_LOG_DERIVATIVES(int N, int Nout, double *bigcdf, double **cdf, double **pdf)
{
 // one may generate an output pdf as derivative of ln(x) rather than of x directly.
 // otherwise this function is identical to CALC_DERIVATIVES().
 double del = 1.0/static_cast<double>(N);        // delta y
 int window = static_cast<int>(N/(2*Nout));      // deriv smoother
 for (int i = 0; i < Nout; i++)
 {
  double numer = 0;
  double denom = 0;
  double avgcdfval = 0;
  for (int j = (2*i)*window; j < (2*i+2)*window; j++) // find "average
   avgcdfval += log(bigcdf[j])/(2.0*window);        // center" of window
  for (int j = (2*i)*window; j < (2*i+2)*window; j++) // regress from avg
  {                                            // center of window
   numer += (del*static_cast<double>(j - (2*i+1)*window))*(log(bigcdf[j]) - avgcdfval);
   denom += pow((log(bigcdf[j]) - avgcdfval),2.0);
  }
```

```
    cdf[0][i] = exp(avgcdfval);
    cdf[1][i] = (static_cast<double>(i)+0.5)/static_cast<double>(Nout);
    pdf[0][i] = exp(avgcdfval);
    pdf[1][i] = numer/denom;                    // from Fatih's notes: slope
  }
}

void WRITE_DF(string filename, int N, double *df)
{
  ofstream outfile(filename.c_str(), ios::out);
  if (!outfile)
  {
    cerr << "\nFile '" << filename.c_str() << "' could not be opened."
         << "\nExiting.\n\n";
    exit(1);
  }
  outfile.precision(5);
  outfile.setf(ios::showpoint);
  for (int i = 0; i < N; i++)
    outfile << df[i] << endl;
  outfile.close();
}

void WRITE_DF(string filename, int N, double **df)    // overloaded method
{
  ofstream outfile(filename.c_str(), ios::out);
  if (!outfile)
  {
    cerr << "\nFile '" << filename.c_str() << "' could not be opened."
         << "\nExiting.\n\n";
    exit(1);
  }
  outfile.precision(4);
  outfile.setf(ios::showpoint);
  for (int i = 0; i < N; i++)
    outfile << df[0][i] << setw(12) << df[1][i] << endl;
  outfile.close();
}
```

lustmodule.h

```
void LUSTMODULE(double **, int, int);
void READLUSTDATA(std::string, double **, int);
```

lustmodule.c

```
// Specialized module to generate the LUST density cdf for parminvcdfs.dat
#include <math.h>
#include <fstream>
```

```
#include <iostream>
#include <string>
#include "lustmodule.h"

using namespace std;

void LUSTMODULE(double **parmicdf, int col, int Ncdf)
{
 int Nlustdat = 50;
 double **lustdat;                    // MC parm pdf arranged as bins
 lustdat = new double*[2];
 for (int i = 0; i < 2; i++)
 {
  lustdat[i] = new double[Nlustdat];
  for (int j = 0; j < Nlustdat; j++)
   lustdat[i][j] = 0.0;
 }
 string lustfile = "LUSTs.dat";
 READLUSTDATA(lustfile, lustdat, Nlustdat);

 double *datacdf;
 double totlust = 0;
 datacdf = new double[Nlustdat];
 for (int i = 0; i < Nlustdat; i++)
 {
  datacdf[i] = 0.0;
  totlust += lustdat[1][i];
 }
 datacdf[0] = 0.5*lustdat[1][0]/totlust;
 for (int i = 1; i < Nlustdat; i++)
  datacdf[i] = datacdf[i-1] + 0.5*(lustdat[1][i] + lustdat[1][i-1])/totlust;       // "midpoint" assumption
 //   datacdf[i] = datacdf[i-1] + lustdat[1][i]/totlust;
 //  for (int i = 0; i < Nlustdat; i++)
 //   cout << datacdf[i] << endl;

 double P = 0;
 for (int i = 0; i < Ncdf; i++)
 {
  P = static_cast<double>(static_cast<double>(i)/Ncdf);
  int j = 0;
  while (P > datacdf[j] && j < Nlustdat)
   j++;
  if (j == 0)                     // special case
   parmicdf[col][i] = 0;
  else
   parmicdf[col][i] = static_cast<double>(j-1) + (P - datacdf[j-1])/(datacdf[j] - datacdf[j-1]);
 }

 for (int i = 0; i < 2; i++)
  delete [] lustdat;
 delete [] datacdf;
}

void READLUSTDATA(string filename, double **data, int N)
{
  ifstream infile(filename.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere did you stick '" << filename.c_str() << "' you bozo?? "
        << "I can't find it.\nExiting.\n\n";
    exit(1);
  }
  for (int j = 0; j < N; j++)
    for (int i = 0; i < 2; i++)
      infile >> data[i][j];
  infile.close();
}
```

model.h

```
double MODEL(double *);
```

model.c

```
#include <iostream>
#include <math.h>
#define PI 3.14159265

using namespace std;

double MODEL(double *p)
{
  double Kfw = 16.0;        // 16.0 MTBE, 2200 ebz, 3.2 phenol, 1700 bzthio
  double Kom = 8.1;         // 8.1 MTBE, 290 ebz, 22 phenol, 500 bzthio
  double Cf = 7.5E7;        // ug/L, 7.5E7 MTBE, 5.48E7 ebz,
                            // 1.13E5 phenol, 2.25E5 bzthio
  double rho = 2.5;
  double phi = 0.25;
  double H = 25.0;          // aquifer saturated depth, m
  //  return p[0];          // toggle to monitor parameter distribution

  double w = p[1]/(H*p[2]*phi);  // capture zone width, m

  double Lx = (1000000.0*PI)/(2.0*p[7]*w);
  // return Lx;

  double R = 1.0 + p[3]*Kom*rho*(1 - phi)/phi;
  double term1 = pow(log(5.0)*Kfw/(12.0*R),2.0)*pow(p[4]*p[5]*PI,1.5)*sqrt(p[0]/phi)/p[6];
  double term2 = 2.0*p[8]*Lx;
  double Cwell = Cf*p[0]*p[2]/((p[1]*5.0*R)*sqrt(term1 + term2));
```

```
 //  double beta = 2.0*PI*phi*p[2]*H/(p[1]);
 //  double Tarr = (R/p[2])*(Lx - sqrt(2.0*p[8]*Lx) - log(1.0 + beta*(Lx - sqrt(2.0*p[8]*Lx)))/beta);

 if (Cwell != Cwell)                         // test for NaN
 {
   cerr << "\nCwell = NaN!" << endl << "Exiting.\n";
   for (int k = 0; k < 9; k++)
     cout << "p[" << k << "] = " << p[k] << endl;
   cout << "Lx = " << Lx << endl;
   cout << "w = " << w << endl;
   cout << "term1 = " << term1 << endl;
   cout << "term2 = " << term2 << endl;
   exit(1);
 }

 return Cwell;
 //  return Tarr;
}
```

LUSTs.dat

```
0       16980
1       2700
2       1400
3       1000
4       630
5       500
6       480
7       315
8       280
9       200
10      190
11      166
12      126
13      140
14      112
15      91
16      59
17      56
18      59
19      59
20      44
21      45
22      34
23      45
24      32
25      30
26      23
27      24
28      18
```

```
29   11
30   16
31   16
32   13
33   13
34   8
35   17
36   8
37   3
38   11
39   6
40   6
41   3
42   5
43   6
44   3
45   5
46   2
47   2
48   2
49   6
```

parm_var.dat

```
3
3
3
3
3
3
3
5
3
```

parms.dat

```
1.65   0.5    2.0
2208   7.7    1.0
0.4    -0.9   0.5
0.003  -5.8   0.6
0.35   -1.05  0.2
0.20   -1.6   0.2
0.002  -6.0   0.9
1.0    0.0    0.0
20     3.0    0.5
```

Appendix C
Self-consistent LSST-LSER calculations for MTBE in fuel-water systems using mass balance constraints; Matlab code

The following Matlab script, **LSSTLSER.m**, iteratively calculates MTBE K_{fw} values in a fuel-water system, based on mass balance constraints. Required input files include:
prop.dat, a data matrix of solvent property information,
coeff.dat, a data matrix of solvent LSER liquid-air coefficients, and
solparms.dat, a data matrix of solute LSER parameters.
These files are also given below.

LSSTLSER.m

```
% LSST-LSER calculations for gasoline

clear
warning off MATLAB:colon:operandsNotRealScalar

% McGowan-based volume fractions if given vol/vol percent mixture
% data (based on pure-phase component densities). Density data taken
% from CRC handbook (1996).
load prop.dat;

% mixture and solute component property lists
load coeff.dat
load solparms.dat

% pure-phase component density based volume fraction data (using prop index):
Fvd = [ ...
0 0 0 0.24 0 0 0 0 0.32 0.10 0 0 0.03 0.07 0.24 0 0 0 0 0; ... % synth gas <6>
0 0 0 0 0 0 0 0 0.52 0 0 0.053 0.019 0.343 0 0 0 0 0.065 0; ... % gas RON98 <9>
0 0 0 0 0 0 0 0 0.95 0 0 0 0 0 0 0 0 0 0.05 0; ... % isooctane-MTBE <9>
0 0 0 0 0 0 0 0 0.85 0 0 0 0 0 0 0 0 0 0.15 0; ... % isooctane-MTBE <9>
0 0 0 0 0 0 0 0 0.70 0 0 0 0 0 0 0 0 0 0.30 0; ... % isooctane-MTBE <9>
0 0 0 0 0 0 0 0 0 0 0 0 0 0.95 0 0 0 0 0.05 0; ... % toluene-MTBE <9>
0 0 0 0 0 0 0 0 0 0 0 0 0 0.85 0 0 0 0 0.15 0; ... % toluene-MTBE <9>
0 0 0 0 0 0 0 0 0 0 0 0 0 0.70 0 0 0 0 0.30 0; ... % toluene-MTBE <9>
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0; ... % pure toluene <9>
0 0 0 0 0 0 0 0 0.833 0 0 0.014 0 0.153 0 0 0 0 0 0; ... % retail diesel <11>
]';

% pure-phase component mass fraction data:

Fmd = [ ...
0 0.083 0.075 0.058 0.022 0.020 0.058 0.040 0.106 0.021 0.018 0.034 0.043 0.162 0.062 ...
0.073 0.092 0.033 0 0; ... % conventional syngas <7>
0 0.075 0.067 0.052 0.020 0.018 0.052 0.036 0.095 0.019 0.016 0.031 0.039 0.146 0.056 ...
0.066 0.083 0.030 0.100 0; ... % oxygenated syngas <7>
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0; ... % pure MTBE <10>
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0; ... % pure alkane (hypthetical)
]';

% mixture weight formulations written as McGowan volume fractions of
% solvent components, NOT as density-based volume fractions.

for i = 1:size(Fvd,2)
 Fv(:,i) = [Fvd(:,i).*prop(:,3).*prop(:,4)./prop(:,2)]/sum(Fvd(:,i).*prop(:,3).*prop(:,4)./prop(:,2));
end

for i = (size(Fvd,2)+1):(size(Fvd,2)+size(Fmd,2))
 Fv(:,i) =[Fmd(:,i-size(Fvd,2)).*prop(:,4)./prop(:,2)]/sum(Fmd(:,i-size(Fvd,2)).*prop(:,4)./prop(:,2));
end

Nmix = size(Fv,2);

% SUBSTITUTE PURE SOLVENTS
aromFcomp11 = 0; % "olefinic" McGowan volume fraction of component 11
aromFcomp12 = 0; % "olefinic" McGowan volume fraction of component 12
deeFcomp19 = 1; % ether McGowan volume fraction of component 19

for i = 1:Nmix
 Falk(i,:) = sum(Fv(2:9,i)) + (1-aromFcomp11)*Fv(11,i) + (1-aromFcomp12)*Fv(12,i) + ...
(1-deeFcomp19)*Fv(19,i);
 Fcyc(i,:) = Fv(10,i);
 Fbenz(i,:) = Fv(13,i);
 Ftol(i,:) = sum(Fv(14:18,i)) + aromFcomp11*Fv(11,i) + aromFcomp12*Fv(12,i);
 Fdee(i,:) = deeFcomp19*Fv(19,i);
 Feth(i,:) = Fv(20,i);
end

for i = 1:Nmix
 Fmix(i,:) = Falk(i).*coeff(2,:) + Fcyc(i)*coeff(3,:) + Fbenz(i)*coeff(4,:) + Ftol(i)*coeff(5,:) + ...
Fdee(i)*coeff(6,:) + Feth(i)*coeff(8,:);
end

% ---------------------------------------------------------------------
% ITERATIVELY CALCULATE MTBE PARTITIONING BETWEEN AQUEOUS PHASE AND
% FUEL, MAINTAINING MASS BALANCE AND ASSUMING V_f = V_w (as in Schmidt, 2002).
% (COMMENT OUT THIS SECTION TO ASSUME NEGLIGIBLE PRESENCE OF MTBE IN
% WATER)

% setup parms
 Fdee_f = Fdee;            % Fdee is actually init MTBE vol frac in fuel phase
 Fdee_w = zeros(Nmix,1);   % initial MTBE frac in aqueous phase is zero

for i = 1:Nmix
 Vw_old(i) = 1;   % init vol of water phase (units are not important)
 Vf_old(i) = 1;   % init vol of fuel phas
 Vw_notMTBE(i) = (1 - Fdee_w(i))*Vw_old(i);   % const values for each mixture
 Vf_notMTBE(i) = (1 - Fdee_f(i))*Vf_old(i);   % const values for each mixture
 Kfw_MTBE_old(i) = 10^sum(Fmix(i,:).*solparms(17,:));  % calc init MTBE Kfw
end

% iteratively move MTBE into water phase
% first iteration requires particular settings to avoid div by zero
for i = 1:Nmix
 Fdee_f(i) = Fdee_f(i)*(1 - Vw_old(i)/(Vw_old(i) + Vf_old(i)*Kfw_MTBE_old(i)));
 Fdee_w(i) = Fdee_f(i)/Kfw_MTBE_old(i);

 Vf(i) = Vf_notMTBE(i)/(1 - Fdee_f(i));
 Vw(i) = Vw_notMTBE(i)/(1 - Fdee_w(i));

 Ffuel(i,:) = (Falk(i).*coeff(2,:) + Fcyc(i)*coeff(3,:) + Fbenz(i)*coeff(4,:) + Ftol(i)*coeff(5,:) + ...
Fdee_f(i)*coeff(6,:) + Feth(i)*coeff(8,:))/Vf(i);
 Fwater(i,:) = Fdee_w(i)*coeff(6,:)./Vw(i);
 Fmix(i,:) = Ffuel(i,:) - Fwater(i,:);

 Kfw_MTBE(i) = 10^sum(Fmix(i,:).*solparms(17,:));  % calc init MTBE Kfw
end

iterations = ones(Nmix,1);
tolerance = ones(Nmix,1);
for i = 1:Nmix
 while (tolerance(i) > 1E-9)
  iterations(i) = iterations(i) + 1;

  Fdee_f(i) = Fdee_f(i)*(1 - Vw(i)/(Vw(i) + Vf(i)*Kfw_MTBE(i)) + Vw_old(i)/(Vw_old(i) + ...
Vf_old(i)*Kfw_MTBE_old(i)));            % calc new Fdee_f variable
  Fdee_w(i) = Fdee_f(i)/Kfw_MTBE(i);

  Vf_old(i) = Vf(i);                        % update "old" Vf variable
  Vw_old(i) = Vw(i);
  Vf(i) = Vf_notMTBE(i)/(1 - Fdee_f(i));       % calc new Vf variable
  Vw(i) = Vw_notMTBE(i)/(1 - Fdee_w(i));

  Ffuel(i,:) = (Falk(i).*coeff(2,:) + Fcyc(i)*coeff(3,:) + Fbenz(i)*coeff(4,:) + Ftol(i)*coeff(5,:) + ...
Fdee_f(i)*coeff(6,:) + Feth(i)*coeff(8,:))./Vf(i);
  Fwater(i,:) = Fdee_w(i)*coeff(6,:)/Vw(i);
  Fmix(i,:) = Ffuel(i,:) - Fwater(i,:);         % calc new Fmix variable

  Kfw_MTBE_old(i) = Kfw_MTBE(i);              % update old Kfw variable
  Kfw_MTBE(i) = 10^sum(Fmix(i,:).*solparms(17,:));  % calc new MTBE Kfw

  tolerance(i) = abs(Vw(i)/(Vw(i) + Vf(i)*Kfw_MTBE(i)) - Vw_old(i)/(Vw_old(i) + ...
Vf_old(i)*Kfw_MTBE_old(i)));
 end
end

% -------------------------------------------------------------------
% Given these system compositions, calculate gasoline-water partition coefficients for
% many other solutes

Kgas2 = [ ... % data LSERpredicted
3E-4 10.^sum(solparms(1,:).*Fmix(1,:)); ... % water, synth gas <6>
5.9 10.^sum(solparms(18,:).*Fmix(1,:)); ... % ethylacetate, synth gas <6>
17 10.^sum(solparms(17,:).*Fmix(1,:)); ... % MTBE, synth gas <6>
15.5 10.^sum(solparms(17,:).*Fmix(11,:)); ... % MTBE, conv ret gas <10>
15.5 10.^sum(solparms(17,:).*Fmix(12,:)); ... % MTBE, oxyg ret gas <10>
0.005 10.^sum(solparms(13,:).*Fmix(1,:)); ... % methanol, synth gas <6>
0.011 10.^sum(solparms(13,:).*Fmix(11,:)); ... % methanol, retail conv gas <6>
0.022 10.^sum(solparms(14,:).*Fmix(1,:)); ... % ethanol, synth gas <6>
0.06 10.^sum(solparms(15,:).*Fmix(1,:)); ... % isopropanol, synth gas <6>
0.14 10.^sum(solparms(16,:).*Fmix(1,:)); ... % tert-butanol, synth gas <6>
3.1 10.^sum(solparms(2,:).*Fmix(2,:)); ... % aniline, retail gas <9>
12 10.^sum(solparms(3,:).*Fmix(2,:)); ... % p-toluidine, retail gas <9>
12 10.^sum(solparms(4,:).*Fmix(2,:)); ... % o-toluidine, retail gas <9>
39 10.^sum(solparms(5,:).*Fmix(2,:)); ... % 2,6-dimethylaniline, retail gas <9>
0.71 10.^sum(solparms(2,:).*Fmix(3,:)); ... % aniline, isooct-MTBE5 <9>
2.5 10.^sum(solparms(3,:).*Fmix(3,:)); ... % p-toluidine, isooct-MTBE5 <9>
1.1 10.^sum(solparms(2,:).*Fmix(4,:)); ... % aniline, isooct-MTBE15 <9>
3.4 10.^sum(solparms(3,:).*Fmix(4,:)); ... % p-toluidine, isooct-MTBE15 <9>
2.0 10.^sum(solparms(2,:).*Fmix(5,:)); ... % aniline, isooct-MTBE30 <9>
5.2 10.^sum(solparms(3,:).*Fmix(5,:)); ... % p-toluidine, isooct-MTBE30 <9>
3.2 10.^sum(solparms(6,:).*Fmix(2,:)); ... % phenol, retail gas <9>
9.3 10.^sum(solparms(7,:).*Fmix(2,:)); ... % p-cresol, retail gas <9>
14 10.^sum(solparms(8,:).*Fmix(2,:)); ... % o-cresol, retail gas <9>
22 10.^sum(solparms(9,:).*Fmix(2,:)); ... % 3,4-dimethylphenol, retail gas <9>
44 10.^sum(solparms(10,:).*Fmix(2,:)); ... % 2,6-dimethylphenol, retail gas <9>
53 10.^sum(solparms(11,:).*Fmix(2,:)); ... % 3,4,5-trimethylphenol, ret gas <9>
120 10.^sum(solparms(12,:).*Fmix(2,:)); ... % 2,4,6-trimethylphenol, retgas <9>
0.65 10.^sum(solparms(6,:).*Fmix(3,:)); ... % phenol, isooct-MTBE5 <9>
2.1 10.^sum(solparms(7,:).*Fmix(3,:)); ... % p-cresol, isooct-MTBE5 <9>
3.5 10.^sum(solparms(8,:).*Fmix(3,:)); ... % o-cresol, isooct-MTBE5 <9>
6.0 10.^sum(solparms(9,:).*Fmix(3,:)); ... % 3,4-dimethylphenol, isoO-MTBE5 <9>
15 10.^sum(solparms(10,:).*Fmix(3,:)); ... % 2,6-dimethylphenol, isoO-MTBE5 <9>
2.2 10.^sum(solparms(6,:).*Fmix(4,:)); ... % phenol, isooct-MTBE15 <9>
6.2 10.^sum(solparms(7,:).*Fmix(4,:)); ... % p-cresol, isooct-MTBE15 <9>
11 10.^sum(solparms(8,:).*Fmix(4,:)); ... % o-cresol, isooct-MTBE15 <9>
17 10.^sum(solparms(9,:).*Fmix(4,:)); ... % 3,4-dimethphenol, isooct-MTBE15 <9>
31 10.^sum(solparms(10,:).*Fmix(4,:)); ... % 2,6-dimethphenol, isoO-MTBE15 <9>
5.4 10.^sum(solparms(6,:).*Fmix(5,:)); ... % phenol, isooct-MTBE30 <9>
17 10.^sum(solparms(7,:).*Fmix(5,:)); ... % p-cresol, isooct-MTBE30 <9>
26 10.^sum(solparms(8,:).*Fmix(5,:)); ... % o-cresol, isooct-MTBE30 <9>
44 10.^sum(solparms(9,:).*Fmix(5,:)); ... % 3,4-dimethphenol, isooct-MTBE30 <9>
62 10.^sum(solparms(10,:).*Fmix(5,:)); ... % 2,6-dimethphenol, isoO-MTBE30 <9>
3.8 10.^sum(solparms(6,:).*Fmix(6,:)); ... % phenol, toluene-MTBE5 <9>
12 10.^sum(solparms(7,:).*Fmix(6,:)); ... % p-cresol, toluene-MTBE5 <9>
18 10.^sum(solparms(8,:).*Fmix(6,:)); ... % o-cresol, toluene-MTBE5 <9>
33 10.^sum(solparms(9,:).*Fmix(6,:)); ... % 3,4-dimethphenol, tol-MTBE5 <9>
76 10.^sum(solparms(10,:).*Fmix(6,:)); ... % 2,6-dimethphenol, tol-MTBE5 <9>
8.3 10.^sum(solparms(6,:).*Fmix(7,:)); ... % phenol, toluene-MTBE15 <9>
28 10.^sum(solparms(7,:).*Fmix(7,:)); ... % p-cresol, toluene-MTBE15 <9>
33 10.^sum(solparms(8,:).*Fmix(7,:)); ... % o-cresol, toluene-MTBE15 <9>
73 10.^sum(solparms(9,:).*Fmix(7,:)); ... % 3,4-dimethphenol, tol-MTBE15 <9>
92 10.^sum(solparms(10,:).*Fmix(7,:)); ... % 2,6-dimethphenol, tol-MTBE15 <9>
16 10.^sum(solparms(6,:).*Fmix(8,:)); ... % phenol, toluene-MTBE30 <9>
50 10.^sum(solparms(7,:).*Fmix(8,:)); ... % p-cresol, toluene-MTBE30 <9>
71 10.^sum(solparms(8,:).*Fmix(8,:)); ... % o-cresol, toluene-MTBE30 <9>
120 10.^sum(solparms(9,:).*Fmix(8,:)); ... % 3,4-dimethphenol, tol-MTBE30 <9>
180 10.^sum(solparms(10,:).*Fmix(8,:)); ... % 2,6-dimethphenol, tol-MTBE30 <9>
110 10.^sum(solparms(19,:).*Fmix(2,:)); ... % thiophene, retail gas <9>
1700 10.^sum(solparms(20,:).*Fmix(2,:)); ... % benzothiophene, retail gas <9>
74 10.^sum(solparms(19,:).*Fmix(3,:)); ... % thiophene, isooct-MTBE5 <9>
99 10.^sum(solparms(19,:).*Fmix(4,:)); ... % thiophene, isooct-MTBE15 <9>
89 10.^sum(solparms(19,:).*Fmix(5,:)); ... % thiophene, isooct-MTBE30 <9>
350 10.^sum(solparms(21,:).*Fmix(11,:)); ... % benzene, conv ret gas <10>
1250 10.^sum(solparms(22,:).*Fmix(11,:)); ... % toluene, conv ret gas <10>
4500 10.^sum(solparms(23,:).*Fmix(11,:)); ... % ethylbenzene, conv ret gas <10>
4350 10.^sum(solparms(25,:).*Fmix(11,:)); ... % m-xylene, conv ret gas <10>
3630 10.^sum(solparms(26,:).*Fmix(11,:)); ... % o-xylene, conv ret gas <10>
18500 10.^sum(solparms(24,:).*Fmix(11,:)); ... % n-propylbenzene, conv gas <10>
13800 10.^sum(solparms(28,:).*Fmix(11,:)); ... % 1,2,3-trimthbenz, convgas <10>
350 10.^sum(solparms(21,:).*Fmix(12,:)); ... % benzene, oxyg ret gas <10>
1250 10.^sum(solparms(22,:).*Fmix(12,:)); ... % toluene, oxyg ret gas <10>
4500 10.^sum(solparms(23,:).*Fmix(12,:)); ... % ethylbenzene, oxyg ret gas <10>
4350 10.^sum(solparms(25,:).*Fmix(12,:)); ... % m-xylene, oxyg ret gas <10>
3630 10.^sum(solparms(26,:).*Fmix(12,:)); ... % o-xylene, oxyg ret gas <10>
18500 10.^sum(solparms(24,:).*Fmix(12,:)); ... % n-propylbenzene, oxyg gas <10>
13800 10.^sum(solparms(28,:).*Fmix(12,:)); ... % 1,2,3-trimthbenz, oxygas <10>
150 10.^sum(solparms(21,:).*Fmix(10,:)); ... % benzene, diesel <10>
480 10.^sum(solparms(22,:).*Fmix(10,:)); ... % toluene, diesel <10>
1200 10.^sum(solparms(30,:).*Fmix(10,:)); ... % naphthalene, diesel <10>
1600 10.^sum(solparms(22,:).*Fmix(10,:)); ... % toluene, diesel <12>
4400 10.^sum(solparms(30,:).*Fmix(10,:)); ... % naphthalene, diesel <12>
23000 10.^sum(solparms(31,:).*Fmix(10,:)); ... % 1-methnaphthalene, diesel <12>
26000 10.^sum(solparms(32,:).*Fmix(10,:)); ... % 2-methnaphthalene, diesel <12>
34000 10.^sum(solparms(33,:).*Fmix(10,:)); ... % acenaphthene, diesel <12>
30000 10.^sum(solparms(34,:).*Fmix(10,:)); ... % fluorene, diesel <12>
49000 10.^sum(solparms(35,:).*Fmix(10,:)); ... % phenanthrene, diesel <12>
190000 10.^sum(solparms(36,:).*Fmix(10,:)); ... % anthracene, diesel <12>
200000 10.^sum(solparms(37,:).*Fmix(10,:)); ... % fluoranthrene, diesel <12>
];

figure
loglog(Kgas2(1,1),Kgas2(1,2),'*') % water
hold on
loglog(Kgas2(2,1),Kgas2(2,2),'+') % ethylacetate
loglog(Kgas2(3:5,1),Kgas2(3:5,2),'x') % MTBE
loglog(Kgas2(6:10,1),Kgas2(6:10,2),'^') % aliphatic alcohols
loglog(Kgas2(11:20,1),Kgas2(11:20,2),'s') % anilines
loglog(Kgas2(21:57,1),Kgas2(21:57,2),'o') % phenols
loglog(Kgas2(58:62,1),Kgas2(58:62,2),'v') % thiophenes
loglog(Kgas2(63:88,1),Kgas2(63:88,2),'h') % aromatic HCs

logKgas2 = log10(Kgas2);

plot(3e-5:1e4:1e6,3e-5:1e4:1e6)
plot(3e-5:1e4:1e6,[3e-5:1e4:1e6]*2,':')
plot(3e-5:1e4:1e6,[3e-5:1e4:1e6]/2,':')
axis([3e-5 1e6 3e-5 1e6])

plot(10^-4, 10^5.4, '*')
text(10^-3.75, 10^5.4, 'water, \sigma = 0.56')
plot(10^-4, 10^5.0, '+')
text(10^-3.75, 10^5.0, 'ethylacetate, \sigma = 0.20')
plot(10^-4, 10^4.6, 'x')
text(10^-3.75, 10^4.6, 'methyl-tert-butyl ether, \sigma = 0.44')
plot(10^-4, 10^4.2, '^')
text(10^-3.75, 10^4.2, 'aliphatic alcohols, \sigma = 0.27')
plot(10^-4, 10^3.8, 's')
text(10^-3.75, 10^3.8, 'aniline, methyl-substituted anilines, \sigma = 0.11')
plot(10^-4, 10^3.4, 'o')
text(10^-3.75, 10^3.4, 'phenol, methyl-substituted phenols, \sigma = 0.61')
plot(10^-4, 10^3.0, 'v')
text(10^-3.75, 10^3.0, 'thiophene, benzothiophene, \sigma = 0.08')
plot(10^-4, 10^2.6, 'h')
text(10^-3.75, 10^2.6, 'aromatic hydrocarbons, \sigma = 0.21')
text(10^1, 10^-2.8, 'Error statistics for entire set (N = 88):')
text(10^1, 10^-3.6, '\sigma = 0.43')
text(10^1, 10^-4.0, 'r^2 = 0.97')
hold off
xlabel('log K_f_w, measured')
ylabel('log K_f_w, predicted using LSST-LSERs')

% 1. Abraham et al., J Chem. Soc. Perk. Trans. 2 (1994) p. 1777-1791.
% 2. Abraham et al., J Chrom. A. 842 (1999) p. 79-114.
% 3. Abraham et al., J Phys. Org. Chem. 6 (1993) p. 660-684.
% 4. Pagliara et al., J Chem. Soc. Perk. Trans. 2 (1997) p. 2639-2643.
% 5. Abraham et al., Collect. Czech. Chem. Commun. 64 (1999) p. 1749-1760.
% 6. Stephenson, J Chem. Eng. Data 37 (1992) p. 80-95.
% 7. Arey, MS thesis (2001).
% 8. Heerman et al., J Cont. Hydr. 34 (1998) p. 315-341.
% 9. Schmidt et al., ES&T, in prep. (2002)
% 10. Cline et al., ES&T (1991) p. 914-920.
% 11. Sjogren et al., Fuel (1995) p. 983-989.
% 12. Yang et al., J Chem. Eng. Data (1997) p. 908-913.
```

prop.dat

```
% index molecweight density McGowanvolume
1 18.015 1.000 0.1673   % water
2 58.123 0.573 0.6722   % butane
```

```
3 72.150 0.6262 0.8131   % pentane
4 86.177 0.6548 0.9540   % hexane
5 100.204 0.6837 1.0949  % heptane
6 114.231 0.6986 1.2358  % octane
7 86.177 0.650 0.9540    % 2-methylpentane
8 86.177 0.6616 0.9540   % 2,3-dimethylbutane
9 114.231 0.6877 1.2358  % 2,2,4-trimethylpentane
10 84.161 0.7486 0.8454  % methylcyclopentane
11 70.134 0.6623 0.7701  % 2-methyl-2-butene
12 84.161 0.6731 0.9110  % 1-hexene
13 78.114 0.8765 0.7164  % benzene
14 92.141 0.8669 0.8573  % toluene
15 106.167 0.8611 0.9982 % xylene
16 106.167 0.8670 0.9982 % ethylbenzene
17 120.194 0.8944 1.1391 % 1,2,3-trimethylbenzene
18 128.174 1.0253 1.0954 % naphthalene
19 88.150 0.7405 0.8718  % MTBE
20 46.069 0.7893 0.4491  % ethanol
```

coeff.dat

```
% c r s a b m
-0.994 0.577 2.549 3.813 4.841 -0.869  % 1 water-air <1>
0.29 0.65 -1.66 -3.52 -4.82 4.28       % 2 alkanes-water <2>
0.13 0.82 -1.73 -3.78 -4.90 4.65       % 3 cyclohexane-water <2>
0.017 0.490 -0.604 -3.013 -4.628 4.587  % 4 benzene-water <3>
0.015 0.594 -0.781 -2.918 -4.571 4.533  % 5 toluene-water <3>
0.462 0.571 -1.035 -0.024 -5.508 4.346  % 6 diethylether-water <3>
0.18 0.82 -1.50 -0.83 -5.09 4.69       % 7 di(n)butylether-water <4>
0.208 0.409 -0.959 0.186 -3.645 3.928          % 8 ethanol-water <5>
0.249 0.480 -0.639 -0.050 -2.284 2.758  % 9 isobutanol-water <3>
```

solparms.dat

```
% c_variable  R2  piH  alphaH  betaH  Vx
 1 0.000 0.45 0.82 0.35 0.167  % 1 water
 1 0.955 0.96 0.26 0.41 0.816  % 2 aniline
 1 0.923 0.95 0.23 0.45 0.957  % 3 p-toluidine
 1 0.966 0.92 0.23 0.45 0.957  % 4 o-toluidine
 1 0.972 0.89 0.20 0.46 1.098  % 5 2,6-dimethylaniline
 1 0.805 0.89 0.60 0.30 0.775  % 6 phenol
 1 0.820 0.87 0.57 0.31 0.916  % 7 p-cresol
 1 0.840 0.86 0.52 0.30 0.916  % 8 o-cresol
 1 0.830 0.86 0.56 0.39 1.057  % 9 3,4-dimethylphenol
 1 0.860 0.79 0.54 0.39 1.057  % 10 2,6-dimethylphenol
 1 0.830 0.88 0.55 0.44 1.198  % 11 3,4,5-trimethylphenol
 1 0.860 0.79 0.37 0.44 1.198  % 12 2,4,6-trimethylphenol
 1 0.278 0.44 0.43 0.47 0.308  % 13 methanol
```

```
1 0.246 0.42 0.37 0.48 0.449  % 14 ethanol
1 0.212 0.36 0.33 0.56 0.590  % 15 isopropanol
1 0.180 0.30 0.31 0.60 0.731  % 16 tert-butanol
1 0.024 0.19 0.00 0.45 0.872  % 17 MTBE
1 0.106 0.62 0.00 0.45 0.747  % 18 ethylacetate
1 0.687 0.56 0.00 0.15 0.641  % 19 thiophene
1 1.323 0.88 0.00 0.20 1.010  % 20 benzo[b]thiophene
1 0.610 0.52 0.00 0.14 0.716  % 21 benzene
1 0.601 0.52 0.00 0.14 0.857  % 22 toluene
1 0.613 0.51 0.00 0.15 0.998  % 23 ethylbenzene
1 0.604 0.50 0.00 0.15 1.139  % 24 n-propylbenzene
1 0.623 0.52 0.00 0.16 0.998  % 25 m-xylene
1 0.663 0.56 0.00 0.16 0.998  % 26 o-xylene
1 0.613 0.52 0.00 0.16 0.998  % 27 p-xylene
1 0.728 0.61 0.00 0.19 1.139  % 28 1,2,3-trimethylbenzene
1 0.630 0.51 0.00 0.18 1.139  % 29 4-ethyltoluene
1 1.340 0.92 0.00 0.20 1.085  % 30 naphthalene
1 1.344 0.90 0.00 0.20 1.2263  % 31 1-methylnaphthalene
1 1.304 0.88 0.00 0.20 1.226  % 32 2-methylnaphthalene
1 1.604 1.05 0.00 0.20 1.259  % 33 acenaphthene
1 1.588 1.06 0.00 0.20 1.357  % 34 fluorene
1 2.055 1.29 0.00 0.26 1.454  % 35 phenanthrene
1 2.290 1.34 0.00 0.26 1.454  % 36 anthracene
1 2.377 1.55 0.00 0.20 1.585  % 37 fluoranthrene
```

Appendix D
C++ and IDL codes used to manipulate and analyze Gaussian98 molecular orbital computations

Several independent programs which perform different functions are listed here. The codes are tailored for use with Gaussian98 Revision A.6 and may not necessarily work correctly with other versions. Gaussian98 output is designed for handling with Fortran, and these utilities would probably have been more robust to updates in Gaussian98 output if they had been written in Fortran. The codes perform crude exception handling. The user is encouraged to use them with an understanding of how they perform, and modify them for his/her needs. As written here, the programs reflect the resources that I have had at my convenient disposal while at M.I.T. (e.g., C++, IDL, Matlab), but other (more computationally efficient) approaches could have been taken if other software had been available (e.g., a working "cubman" utility). I did not rely heavily on Gaussian98 checkpoint files, except to solve occasionally difficult Gaussian98 numerical hang-ups during geometry optimizations or self-consistent field calculations. Checkpoint files require substantial hard disk memory, and for the large number of calculations that I performed (over 2000 calculations for 90 types of molecules), checkpoint files would have used far more disk space than was available. However, future users are encouraged to use checkpoint files where possible, since this greatly improves the speed of repeated single point evaluations. I have not reviewed the syntax of Gaussian98 input – the reader should independently gain familiarity with Gaussian98 input syntax before attempting to repeat calculations discussed here. A quick description of the codes, and their uses, follows.

All of the C++ codes implicitly require an input file called **filelist.dat**, which gives a list of the root filenames that the user wants to analyze, i.e.:

```
benz
h2o
toln
...
```

For example, the first entry, "benz", represents the filename root that is specific to a certain kind of molecule, whereas the filename suffix will reflect a specific kind of input or output file. The syntax is always: *moleculename dot filetype*, e.g., the file "h2o.opt" would contain the Gaussian98 output of a water molecular geometry optimization. The **filelist.dat** list therefore allows the user to manipulate, analyze, or generate information for several different molecules at once.

The data file **comment.txt** gives a list of strings with short descriptions corresponding to the filename root, and these are generally included in Gaussian98 input as a way of aiding the user. The syntax of **comment.txt** is

```
alk3      propane
mtbe      methyltertbutylether
mtbe2     methyltertbutylether-alternate-geometric-configuration
h2o       water
...
```

where the first column is intended to be a short mnemonic used as the root string for filenames, and the second column is a single string intended to give a little more explanation.

The file **header.txt** contains the Gaussian98 header instructions for a particular calculation, e.g.:

```
    B3LYP/G-31+G(d,p) OPT SCF=TIGHT NOSYMM TEST
```

By editing the **header.txt** file, the user can dictate the type of computation which will then be applied to a large number of input files with a single command such as **makeinputset.c.** Together, **filelist.dat** and **header.txt** allow the user to processively calculate properties for a "fleet" of molecules, which can contribute to considerable user input time savings if the molecule set is large.

I have used a slightly different naming convention for "jobfiles," that is, files which are used to submit an input file of interest to Gaussian98 and indicate the name of the corresponding output file. Jobfiles always use the syntax *j moleculename*, e.g., the file "jmtbe" is a jobfile for mtbe. Sometimes a "k" prefix has been used instead (in cases where I want to preserve two types of jobfile for the same set of molecules at the same time).

File suffix conventions I have used include the following. Some of the listings refer to Gaussian98 direct output (G98 refers to Gaussian98), but others simply correspond to input or archived analysis results:

molec.inp	any input file for a G98 calculation
molec.out	generic (unspecified) G98 job output file
molec.opt	gas phase G98 geometry optimization using B3LYP
molec.xopt	gas phase G98 geometry optimization using HF/MIDI!
molec.sopt	G98 geometry optimization in the presence of a B3LYP PCM field
molec.copt	G98 geometry optimization in the presence of a HF/MIDI! PCM field
molec.gnc	saved optimized nuclear coordinates (gas phase B3LYP)
molec.xnc	saved optimized nuclear coordinates (gas phase HF/MIDI!)
molec.snc	saved optimized nuclear coordinates (B3LYP with PCM)
molec.cnc	saved optimized nuclear coordinates (HF/MIDI! with PCM)
molec.gsp	gas phase G98 single point calculation using B3LYP
molec.xsp	gas phase G98 single point calculation using HF/MIDI!
molec.ssp	G98 single point calculation using B3LYP with PCM
molec.csp	G98 single point calculation using HF/MIDI! with PCM
molec.fsp	G98 single point calculation using HF/MIDI! with IPCM or SCIPCM
molec.gcub	G98 cube file, B3LYP gas phase
molec.xcub	G98 cube file, HF/MIDI! gas phase
molec.scub	G98 cube file, B3LYP with PCM
molec.ccub	G98 cube file, HF/MIDI! with PCM
molec.ccib	G98 cube file, HF/MIDI! with IPCM or SCIPCM
molec.sic	saved electron isosurface coordinates at 0.0004 e^-/bohr^3, B3LYP with PCM
molec.sic1	saved electron isosurface coords at 0.0001 e^-/bohr^3, B3LYP with PCM
molec.cic	saved electron isosurface coords at 0.0004 e^-/bohr^3, HF/MIDI! with PCM
molec.cic1	saved electron isosurface coords at 0.0001 e^-/bohr^3, HF/MIDI! with PCM
molec.fic	saved electron isosurface coords at 0.0004 e^-/bohr^3, HF/MIDI! with SCIPCM
molec.iic	saved electron isosurface coords at 0.0004 e^-/bohr^3, HF/MIDI! with IPCM

molec.sts	saved PCM tesserae coordinates, calculated from B3LYP with PCM
molec.gfd	G98 calculated field, potential at 0.0004 e^-/bohr^3, B3LYP gas phase
molec.ifd	G98 calculated field, potential at 0.0004 e^-/bohr^3, B3LYP with PCM
molec.ifd1	G98 calculated field, potential at 0.0001 e^-/bohr^3, B3LYP with PCM
molec.cfd	G98 calculated field, potential at 0.0004 e^-/bohr^3, HF/MIDI! with PCM
molec.ffd	G98 calc'd field, potential at 0.0004 e^-/bohr^3, HF/MIDI! with IPCM or SCIPCM

The uses of the different codes are as follows:

savegeomset.c takes **molec.*out** files as input, attempts to find a stationary point, and returns the (cartesian) coordinates as **molec.*nc** files, and also generates a set of input files with these optimized coordinates using **header.txt** (**filelist.dat** is also required). The file **savegeomset.log** reports the progress of the search, and a flag notifies the user if optimized results cannot be found for a particular molecule. A nuclear geometry file for water, **h2o.cnc**, might look like:

```
3
   O    0.000000   0.000000  -0.123148
   H    0.000000   0.753468   0.492594
   H    0.000000  -0.753468   0.492594
```

where the first line designates the total number of nuclei, and subsequent lines give the atom types and locations (Å).

makeinputset.c is an "all-purpose" code which simply generates a set of input files from the user-designated set of **molec.*nc** files, applying the Gaussian98 header found in **header.txt** (**filelist.dat** is also required).

makecubset.c modifies an existing set of **molec.inp** files, by appending the input instructions necessary to generate a cube file of the appropriate dimensions to encapsulate an electron isodensity surface of 0.0001 e^-/bohr^3. Specifically, the **makecubset.c** utility parameters tell Gaussian98 to output a 3-dimensional grid of electron density (called a "cube file") such that the box edges are at least *r* distance away from any nucleus, where *r* is adjusted by the user in terms of Van der Waals radii, using the "VdW scale factor." Finally, the box has a uniform spacing between points as set by the user. Recommended inputs are: VdW scale factor = 2.0, and grid resolution = 0.2 Å (**filelist.dat** is also required).

getiso.pro is an Interactive Data Language (IDL) code which takes a cube file as input, and generates and rotates on-screen rendered electron isodensity surfaces of 0.0004 e^-/bohr^3 and 0.0001 e^-/bohr^3, and then records these vectors of isodensity vertex locations in **molec.*ic** and **molec.*ic1** files. These vertex vectors can subsequently be appended to Gaussian98 input, in order to evaluate the electrostatic field and potential at these points. **getiso.pro** also overwrites the **molec.*nc** files from the cube files, in case the nuclei standard orientations have changed.

intfieldset.c takes a set of Gaussian98 output files containing electrostatic field and potential evaluations along the set of SAS (solvent accessible surface) points as input, and integrates equation 4-7 along these points, producing an on-screen output vector of ΔU_e and ΣV^2 values (in kcal/mol and kcal Å/mol, respectively) readable by Matlab. This code must be compiled with **point.c** (and it also requires **filelist.dat**).

dipoleset.c takes a set of (practically any type of) Gaussian98 output file as input, and reads the calculated dipole moment tensors, which are then output to the screen and recorded in **dipoleset.log** (the file **filelist.dat** is also required).

point.h and **point.c** are a C++ point class.

An example sequence of calculations may be as follows:

(1) Perform a geometry optimization of a set of molecules, e.g., using the header:
    MIDIX OPT SCF=TIGHT SCRF=(PCM,SOLVENT=WATER) TEST
  to give output file set **molec.copt**.

(2) Use **savegeomset.c** to record the optimized nuclear positions of this molecule set in the files called **molec.cnc**, generating a new set of input files (at these geometries) with the header:
    MIDIX SCF=TIGHT CUBE=(DENSITY,CARDS)
    SCRF=(PCM,SOLVENT=WATER) TEST

(3) Run the **makecubset.c** utility, which will append the input files with appropriate Gaussian98 input stream instructions for tailored cube files of appropriate dimension and uniform resolution, for each molecule.

(4) Use Gaussian98 to perform these single point calculations, generating a set of **molec.ccub** files.

(5) Run **getiso.pro** on each of the cube files, generating a set of **molec.cic** and **molec.cicl** files.

(6) Use **makeinputset.c** to generate a new set of input files having the header
    MIDIX SCF=TIGHT PROP=(READ,FIELD) IOP(6/33=2)
    SCRF=(PCM,SOLVENT=WATER) TEST

(7) Append each **molec.inp** file with a **molec.cic** file – this can be rapidly achieved for a large set of files using the UNIX "foreach" command, as:

```
foreach f (`more filelist.dat`)
mv $f.inp scratch
cat scratch $f.cic > $f.inp
end
```

(8) Use Gaussian98 to perform these calculations, generating a set of **molec.cfd** files.

(9) Run **intfieldset.c** on the set of **molec.cfd** files.

savegeomset.c

```
#include <iostream>
#include <iomanip>
#include <cstdlib>
#include <string>
#include <fstream>
#include <math.h>

// This code saves the standard orientation geometry of a gaussian
// output file in cartesian coordinates, good for the geometry input
// of another gaussian input file.

using namespace std;

int GET_N_MOLEC(string);
void READ_FILENAMES(int, string, string *);
int FIND_N_POINTS(char []); // find the number of nuclei in the opt file
int FIND_N_OPTS(char []);  // find number of optimizations in the opt file
void READ_XYZ_DATA(int, int, double **, int *, char [], int index);
void WRITE(char [], double **, int *, int);
void WRITE_INPUT(const char [], double **, int *, int);
void WRITE_JOBFILE(const char [], string suffix);

int main()
{
  int N_points = 0;              // number of data points
  int N_opts = 0;                // number of optimization cycles
  int N_molec = 0;               // number of molec data files to read
  char inputfile[16];
  char geomfile[16];
  char outfile[16];

  string filelist = "filelist.dat";       // filename of molec name data
  N_molec = GET_N_MOLEC(filelist);

  string *basefiles;                       // vector of molec names
  basefiles = new string[N_molec];
  READ_FILENAMES(N_molec, filelist, basefiles);

  string appstr;
  cout << "\n\nenter gaussian output file extension, eg .xopt suffix:\n? ";
  cin >> appstr;

  string appstr_g;
  cout << "\nenter gaussian geometry file extension, eg .xnc suffix:\n? ";
  cin >> appstr_g;

  string suffix;
  cout << "\nenter anticipated output file suffix:\n? ";
  cin >> suffix;

  for (int k = 0; k < N_molec; k++)
  {
    strcpy(inputfile,basefiles[k].c_str());
    strcat(inputfile,".");
    strcat(inputfile,appstr.c_str());

    N_points = FIND_N_POINTS(inputfile);
    N_opts = FIND_N_OPTS(inputfile);

    double **xyz;            // declare dynamic mem 3d coordinate data
    xyz = new double*[N_points];
    for (int i = 0; i < N_points; i++)
      xyz[i] = new double[3];

    int *atoms;
    atoms = new int[N_points];

    for (int i = 0; i < N_points; i++)
      for (int j = 0; j < 3; j++)
            xyz[i][j] = 0;
    READ_XYZ_DATA(N_points, N_opts, xyz, atoms, inputfile, k);

    strcpy(outfile, basefiles[k].c_str());
    strcat(outfile, ".");
    strcat(outfile, appstr_g.c_str());

    WRITE(outfile, xyz, atoms, N_points);
    WRITE_JOBFILE(basefiles[k].c_str(), suffix);
    WRITE_INPUT(basefiles[k].c_str(), xyz, atoms, N_points);

    cout << "\nStd orientation cartesian geometry file "
         << outfile << " produced (angstroms).\n";
    cout << " ---------------------------------------------------------------------\n\n";

    for (int i = 0; i < N_points; i++)
     delete [] xyz[i];               // memory management
   }

  delete [] basefiles;               // memory management

return 0;
}

int GET_N_MOLEC(string filelist)
{
  int N = 0;
  ifstream infile(filelist.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << filelist << "'?? "
        << "I can't find it.\nExiting.\n\n";
    exit(1);
  }

  string dummy;
  while (infile >> dummy)
    N++;
  cout << "\nN molec = " << N;
  return N;
}

void READ_FILENAMES(int N, string filelist, string *filenames)
{
  ifstream infile(filelist.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << filelist << "'?? "
        << "I can't find it.\nExiting.\n\n";
    exit(1);
  }
  for (int i = 0; i < N; i++)
    infile >> filenames[i];
}

int FIND_N_POINTS(char data[])
{
  int N = 0;
  ifstream infile(data, ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << data << "'?!!  "
         << "I can't find it.\n"
         << "\nExiting.\n\n";
    exit(1);
  }

  string dummy1, dummy2;

  while (infile >> dummy1)
  {
    if (dummy1 == "------------------------------------------------------------------" && dummy2 == "Z")
    {
      infile >> dummy1;
      while (dummy1 != "------------------------------------------------------------------")
      {
          for (int j = 0; j < 6; j++)
          {
            infile >> dummy1;
          }
          N++;
      }
      break;
    }
    dummy2 = dummy1;
  }
  cout << "N = " << N << " nuclei";
  return N;
}

int FIND_N_OPTS(char data[])
{
  int N = 0;
  ifstream infile(data, ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << data << "'?!!  "
         << "I can't find it.\n"
         << "\nExiting.\n\n";
    exit(1);
  }
```

```
 string dummy1, dummy2;

 while (infile >> dummy1)
 {
  if (dummy1 == "orientation:" && dummy2 == "Input")
   N++;
  dummy2 = dummy1;
 }
 if (N > 2)
  cout << endl <<  N-1 << " optimization cycles were found.";
 else if (N == 2)
  cout << "\n1 optimization cycle was found.";
 else if (N == 1)
  cout << "\nNo optimization cycles were found.";
 else
 {
  cerr << "\nERROR. NO STANDARD ORIENTATION TABLE FOUND.\nExiting.\n";
  exit(1);
 }
 return N;
}

void READ_XYZ_DATA(int N_points, int N_opts, double **xyz, int *atom, char data[], int index)
{
 ifstream infile(data, ios::in);
 if (!infile)
 {
  cerr << "\nWhere is '" << data << "'?!!  "
      << "I can't find it.\n"
      << "\nExiting.\n\n";
  exit(1);
 }

 string dummy1, dummy2;
 int N=0;
 int optflag=0;
 int messageflag=0;
 int NOSYMflag=0;
 int readflag=0;

 while (infile >> dummy1)
 {
  if (dummy1 == "#" && messageflag == 0)
  {
   messageflag = 1;
   cout << "\nCommand line: # ";
   int counter = 0;
   while (infile >> dummy1)
   {
    if (dummy1 == "NOSYM" || dummy1 == "NOSYMM")
```

```
        NOSYMflag = 1;
          cout << dummy1 << " ";
          counter++;
          if (dummy1 == "TEST" || counter > 8)
            break;
    }
  }

  if (dummy1 == "orientation:" && dummy2 == "Input")
    N++;
  if (dummy1 == "Parameters" && dummy2 == "Optimized")
    optflag=1;

  if (dummy1 == "orientation:" && dummy2 == "Standard" && N == N_opts)
    readflag = 1;
  if (dummy1 == "orientation:" && dummy2 == "Input" && N == N_opts && NOSYMflag == 1)
    readflag = 1;
  if (readflag == 1)
  {
    while (infile >> dummy1)
    {
      if (dummy1 == "Parameters" && dummy2 == "Optimized")
        optflag=1;
          if (dummy1 == "---------------------------------------------------------------------" && dummy2 ==
"Z")
      {
            int i = 0;
            for (i = 0; i < N_points; i++)
            {
              infile >> dummy1;
              infile >> atom[i];
              infile >> dummy1;
              for (int j = 0; j < 3; j++)
                infile >> xyz[i][j];
            }
          }
          dummy2 = dummy1;
    }
  }
  dummy2 = dummy1;
}

if (optflag == 0)
  cout << "\nWARNING: 'OPTIMIZED PARAMETERS' TABLE NOT FOUND.";
else
  cout << "\n'Optimized Parameters' table found.";

// WRITE A SUMMARY TO THE .LOG FILE
if (index == 0)
{
  ofstream logfile("savegeom3set.log", ios::out);
  if (logfile == 0)
    exit(1);
  if (optflag == 0)
    logfile << "File " << data << ": 'OPTIMIZED PARAMETERS' TABLE NOT FOUND.\n";
  else
    logfile << "File " << data << ": 'Optimized Parameters' table found.\n";
  logfile << N_opts << " optimizations found.\n";
  for (int i = 0; i < N_points; i++)
  {
    for (int j = 0; j < 3; j++)
    {
        if (xyz[i][j] == 0)
          logfile << "    " << "0.000000";
        else if (xyz[i][j] < 0)
          logfile << "   " << xyz[i][j];
        else
          logfile << "    " << xyz[i][j];
    }
    logfile << endl;
  }
  logfile << " ----------------------------------------------------------------\n\n";
 }
 else
 {
  ofstream logfile("savegeomset.log", ios::app);
  if (logfile == 0)
    exit(1);
  if (optflag == 0)
    logfile << "File " << data << ": 'OPTIMIZED PARAMETERS' TABLE NOT FOUND.\n";
  else
    logfile << "File " << data << ": 'Optimized Parameters' table found.\n";
  logfile << N_opts << " optimizations found.\n";
  for (int i = 0; i < N_points; i++)
  {
    for (int j = 0; j < 3; j++)
    {
        if (xyz[i][j] == 0)
          logfile << "    " << "0.000000";
        else if (xyz[i][j] < 0)
          logfile << "   " << xyz[i][j];
        else
          logfile << "    " << xyz[i][j];
    }
    logfile << endl;
  }
  logfile << " ----------------------------------------------------------------\n\n";
 }
}

void WRITE(char filename[], double **xyzdata, int *atom, int N)
{
```

```
ofstream outfile(filename, ios::out);
if (outfile == 0)
  exit(1);

outfile.setf(ios::fixed);
cout.setf(ios::fixed);
cout << endl << "  " << N << endl << endl;
outfile << "  " << N << endl << endl;

for (int i = 0; i < N; i++)
{
  outfile << setiosflags(ios::left) << setw(6);
  cout << setiosflags(ios::left) << setw(6);
  switch (atom[i])
  {
  case 1:
    outfile << "H";
    cout << "H";
    break;
  case 6:
    outfile << "C";
    cout << "C";
    break;
  case 7:
    outfile << "N";
    cout << "N";
    break;
  case 8:
    outfile << "O";
    cout << "O";
    break;
  case 9:
    outfile << "F";
    cout << "F";
    break;
  case 15:
    outfile << "P";
    cout << "P";
    break;
  case 16:
    outfile << "S";
    cout << "S";
    break;
  case 17:
    outfile << "Cl";
    cout << "Cl";
    break;
  case 35:
    outfile << "Br";
    cout << "Br";
    break;
  case 53:
   outfile << "I";
   cout << "I";
   break;
  }
  outfile.setf(ios::right);
  cout.setf(ios::right);
  outfile.precision(6);
  cout.precision(6);
  for (int j = 0; j < 3; j++)
  {
   outfile << setw(12) << xyzdata[i][j];
   cout << setw(12) << xyzdata[i][j];
  }
  outfile << endl;
  cout << endl;
 }
}

void WRITE_INPUT(const char basefile[], double **xyzdata, int *atom, int N)
{
 char headertext[] = "header.txt";
 char commenttext[] = "comment.txt";
 char footertext[] = "footer.txt";
 char Ginput[16];
 char append[] = ".inp";
 strcpy(Ginput,basefile);
 strcat(Ginput,append);

 ifstream infile(headertext, ios::in);
 if (!infile)
 {
  cerr << "\nWhere the heck is '" << headertext << "'?!!  "
     << "I can't find it.\nA program needs data to run, you know..."
     << "\nExiting.\n\n";
  exit(1);
 }

 ofstream outfile(Ginput, ios::out);
 if (outfile == 0)
  exit(1);

 outfile.setf(ios::fixed);
 int fourflag = 0;
 string header[10];
 for (int i = 0; i < 10; i++)
 {
  if (i == 4 || i == 8)
   outfile << endl;
  infile >> header[i];
  if (header[i] == "NEWLINE")
  {
    outfile << endl;
    i = 0;
  }
  else
    outfile << header[i] << " ";
  if (infile == 0)
  {
    fourflag = i;
    break;
  }
}

outfile << endl;
if (fourflag != 4 && fourflag != 8)
  outfile << endl;
infile.close();
ifstream infile2(commenttext, ios::in);
if (!infile2)
{
  cerr << "\nWhere the heck is '" << commenttext << "'?!!  "
       << "I can't find it.\nA program needs data to run, you know..."
       << "\nExiting.\n\n";
  exit(1);
}

string comment;
int commentflag = 0;
while (infile2 >> comment)
{
  if (comment == basefile)
  {
    infile2 >> comment;
    commentflag = 1;
    break;
  }
}
if (commentflag == 0)
{
  cerr << "ERROR. Comment file needs to be updated. "
          << "Input file left incomplete.\nExiting.\n";
  exit(1);
}
outfile << comment << endl << endl << "0 1" << endl;

for (int i = 0; i < N; i++)
{
  outfile << setw(6);
  switch (atom[i])
  {
  case 1:
```

```
  outfile << "H   ";
  break;
 case 6:
  outfile << "C   ";
  break;
 case 7:
  outfile << "N   ";
  break;
 case 8:
  outfile << "O   ";
  break;
 case 9:
  outfile << "F   ";
  break;
 case 15:
  outfile << "P   ";
  break;
 case 16:
  outfile << "S   ";
  break;
 case 17:
  outfile << "Cl  ";
  break;
 case 35:
  outfile << "Br  ";
  break;
 case 53:
  outfile << "I   ";
  break;
 }
 outfile.setf(ios::right);
 outfile.precision(6);
 for (int j = 0; j < 3; j++)
  outfile << setw(12) << xyzdata[i][j];
 outfile << endl;
}
outfile << endl;

ifstream infile3(footertext, ios::in);
if (infile3)
{
 string footer[10];
 for (int i = 0; i < 10; i++)
 {
  infile3 >> footer[i];
  outfile << footer[i] << " ";
 }
 outfile << endl << endl;
 infile3.close();
}
cout << "\nGaussian input file '" << Ginput << "' returned.";
```

```
}

void WRITE_JOBFILE(const char basefile[], string suffix)
{
 char jobfile[] = "j";
 strcat(jobfile,basefile);
 ofstream outfile(jobfile, ios::out);
 if (outfile == 0)
   exit(1);
 outfile << "#!/bin/csh" << endl << endl
         << "/usr2/g98/g98 < /usr/people/sarey/jobsDx/" << basefile
         << ".inp > /usr/people/sarey/jobsDx/" << basefile
         << "." << suffix << endl << endl;
 cout << "\nGaussian job file '" << jobfile << "' returned.";
}
```

makeinputset.c

```
#include <iostream>
#include <iomanip>
#include <stdlib.h>
#include <string>
#include <fstream>
#include <math.h>

// This code creates an input file + jobfile simply from the saved nuclear
// geometry file (molec.snc). It is good for preserving the nuclear
// orientation chosen by Gaussian in a cube calculation. Coordinates in Ang.

using namespace std;

int GET_N_MOLEC(string);
void READ_FILENAMES(int, string, string *);
int FIND_N_POINTS(char []); // find the number of nuclei in the opt file
void READ_XYZ_DATA(double **, string *, char []);
void WRITE_INPUT(const char [], double **, string *, int);
void WRITE_JOBFILE(const char [], char [], char);

int main()
{
 int i = 0;                // counter variable
 int N_points = 0;             // number of data points
 int N_molec = 0;              // number of molec data to read
 char geomfile[16];
 char outfile[16];

 string filelist = "filelist.dat";    // filename of molec name data
 N_molec = GET_N_MOLEC(filelist);
 string *basefiles;              // vector of molec names
 basefiles = new string[N_molec];
 READ_FILENAMES(N_molec, filelist, basefiles);

 string app_nc;
 cout << "enter the nuclear geometry file extension (e.g. 'gnc')\n? ";
 cin >> app_nc;
 string app_out;
 cout << "enter the desired gaussian output file extension (e.g. 'gpot')\n? ";
 cin >> app_out;
 char jobflag;
 cout << "choose an output directory: is this a (b)3lyp calc or a (m)idi calc\n? ";
 cin >> jobflag;

 for (int k = 0; k < N_molec; k++)
 {
   strcpy(geomfile, basefiles[k].c_str());
   strcat(geomfile, ".");
   strcat(geomfile, app_nc.c_str());
   strcpy(outfile, basefiles[k].c_str());
   strcat(outfile, ".");
   strcat(outfile, app_out.c_str());

   N_points = FIND_N_POINTS(geomfile);
   double **xyz;           // declare dynamic mem 3d coordinate data
   xyz = new double*[N_points];
   for (i = 0; i < N_points; i++)
     xyz[i] = new double[3];
   string *atoms;
   atoms = new string[N_points];
   for (i = 0; i < N_points; i++)
     for (int j = 0; j < 3; j++)
          xyz[i][j] = 0;

   READ_XYZ_DATA(xyz, atoms, geomfile);
   WRITE_INPUT(basefiles[k].c_str(), xyz, atoms, N_points);
   WRITE_JOBFILE(basefiles[k].c_str(), outfile, jobflag);

   for (i = 0; i < N_points; i++)
     delete [] xyz[i];            // memory management
   delete [] atoms;
  }
  cout << endl;
  return 0;
}

int GET_N_MOLEC(string filelist)
{
  int N = 0;
  ifstream infile(filelist.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << filelist << "'?? "
```

```
        << "I can't find it.\nExiting.\n\n";
    exit(1);
  }
  string dummy;
  while (infile >> dummy)
    N++;
  cout << "\nN molec = " << N << endl;
  return N;
}

void READ_FILENAMES(int N, string filelist, string *filenames)
{
  ifstream infile(filelist.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << filelist << "'?? "
        << "I can't find it.\nExiting.\n\n";
    exit(1);
  }
  for (int i = 0; i < N; i++)
    infile >> filenames[i];
}

int FIND_N_POINTS(char data[])
{
  int N = 0;
  ifstream infile(data, ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << data << "'??  "
        << "I can't find it.\nExiting.\n\n";
    exit(1);
  }
  infile >> N;
  infile.close();
  return N;
}

void READ_XYZ_DATA(double **xyz, string *atom, char data[])
{
  ifstream infile(data, ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << data << "'??  "
        << "I can't find it.\nExiting.\n\n";
    exit(1);
  }
  int N_points = 0;
  infile >> N_points;
  for (int i = 0; i < N_points; i++)
  {
    infile >> atom[i];
    for (int j = 0; j < 3; j++)
      infile >> xyz[i][j];
  }
}

void WRITE_INPUT(const char basefile[], double **xyzdata, string *atom, int N)
{
  char headertext[] = "header.txt";
  char commenttext[] = "comment.txt";
  char footertext[] = "footer.txt";

  char Ginput[16];
  char append[] = ".inp";
  strcpy(Ginput,basefile);
  strcat(Ginput,append);
  ifstream infile(headertext, ios::in);
  if (!infile)
  {
    cerr << "\nWhere is file '" << headertext << "'??"
         << "\nI can't find it..."
         << "\nExiting.\n\n";
    exit(1);
  }
  ofstream outfile(Ginput, ios::out);
  if (outfile == 0)
    exit(1);
  outfile.setf(ios::fixed);
  cout.setf(ios::fixed);
  string header[10];
  int iterflag = 0;
  for (int i = 0; i < 10; i++)
  {
    infile >> header[i];
    if (infile == 0)
      break;
    if (header[i] == "NEWLINE")
    {
      outfile << endl;
      i = 0;
    }
    else
      outfile << header[i] << " ";
    if (i == 4 || i == 8)
      outfile << endl;
    iterflag = i;
  }
  if (iterflag == 4 || iterflag == 8)
    outfile << endl;
  else
    outfile << endl << endl;
```

```
infile.close();
ifstream infile2(commenttext, ios::in);
if (!infile2)
{
  cerr << "\nWhere the heck is '" << commenttext << "'?!!  "
       << "I can't find it.\nExiting.\n\n";
  exit(1);
}
string comment;
int commentflag = 0;
while (infile2 >> comment)
{
  if (comment == basefile)
  {
    infile2 >> comment;
    commentflag = 1;
    break;
  }
}
if (commentflag == 0)
{
  cerr << "ERROR. Comment file needs to be updated. "
         << "Input file left incomplete.\nExiting.\n";
  exit(1);
}
if (comment == "dioxygen" || comment == "oxygen")
{
  outfile << comment << endl << endl << "0 3" << endl;
  cout << "\nSETTING O2 MULTIPLICITY TO 3.";
}
else
  outfile << comment << endl << endl << "0 1" << endl;
for (int i = 0; i < N; i++)
{
  outfile.setf(ios::left);
  outfile << setw(6) << atom[i].c_str();
  outfile.unsetf(ios::left);
  outfile.setf(ios::right);
  outfile.precision(6);
  for (int j = 0; j < 3; j++)
    outfile << setw(12) << xyzdata[i][j];
  outfile << endl;
  outfile.unsetf(ios::right);
}
outfile << endl;
ifstream infile3(footertext, ios::in);
if (infile3)
{
  string footer[10];
  for (int i = 0; i < 10; i++)
  {
      infile3 >> footer[i];
      outfile << footer[i] << " ";
    }
    outfile << endl << endl;
    infile3.close();
  }
  cout << "\nGaussian input file '" << Ginput << "' returned.";
}

void WRITE_JOBFILE(const char basefile[], char outfilename[], char jflag)
{
  char jobfile[] = "k";
  strcat(jobfile,basefile);
  ofstream outfile(jobfile, ios::out);
  if (outfile == 0)
    exit(1);
  if (jflag == 'b')
  {
  outfile << "#!/bin/csh" << endl << endl
          << "/usr2/g98/g98 < /usr/people/sarey/jobsDx/" << basefile
          << ".inp > /usr/people/sarey/jobsDx/" << outfilename
          << endl << endl;
  }
  else if (jflag == 'm')
  {
  outfile << "#!/bin/csh" << endl << endl
          << "/usr2/g98/g98 < /usr/people/sarey/midi/" << basefile
          << ".inp > /usr/people/sarey/midi/" << outfilename
          << endl << endl;
  }
  else
  {
    cerr << "ERROR. Incorrect job type input.\nExiting.\n";
    exit(1);
  }
  cout << "\nGaussian job file '" << jobfile << "' returned.";
}
```

makecubset.c

```
#include <iostream>
#include <iomanip>
#include <string>
#include <fstream>
#include <math.h>

using namespace std;

int GET_N_MOLEC(string);
void READ_FILENAMES(int, string, string *);
```

```
int FIND_N_PTS(char []);
void READ_XYZ_DATA(double **, string *, char []);

int main()
{
 int N_pts = 0;
 int N_molec = 0;           // number of molec data files to read
 double min[3];             // lowest xyz point
 double max[3];             // highest xyz point
 int Nsteps[3];             // stepsize in each dimension xyz
 double stepsize = 0;
 char solvflag;
 char ipcmflag;
 char geomfile[16];
 char inputfile[16];

 cout << "enter (g)as, (s)olvated, midi(x), or midix solvated (c) solute\n? ";
 cin >> solvflag;
 cout << "enter (i)pcm calculation (0.0004 e-/bohr3 assumed) or (n)ot an ipcm calculation\n? ";
 cin >> ipcmflag;
 double scalefactor = 0;
 cout << "enter scale factor on the Van der Waals estimated radii\n? ";
 cin >> scalefactor;
 cout << "enter the grid resolution (step size) in angstroms\n? ";
 cin >> stepsize;

 string filelist = "filelist.dat";      // filename of molec name data
 cout << "\nReading input file names from " << filelist << endl;
 N_molec = GET_N_MOLEC(filelist);
 cout << "\n\nThe following input files have been modified:\n";
 string *basefiles;                     // vector of molec names
 basefiles = new string[N_molec];
 READ_FILENAMES(N_molec, filelist, basefiles);

 char append1[] = ".gnc";
 if (solvflag == 's')
   append1[1] = 's';
 if (solvflag == 'x')
   append1[1] = 'x';
 if (solvflag == 'c')
   append1[1] = 'c';
 char append2[] = ".inp";

 for (int k = 0; k < N_molec; k++)
 {
   strcpy(geomfile, basefiles[k].c_str());
   strcpy(inputfile, basefiles[k].c_str());
   strcat(geomfile, append1);
   strcat(inputfile, append2);

   N_pts = FIND_N_PTS(geomfile);
```

```
double **xyz;                  // declare dynamic mem 3d coordinate data
xyz = new double*[N_pts];
for (int i = 0; i < N_pts; i++)
  xyz[i] = new double[3];
string *atoms;
atoms = new string[N_pts];

READ_XYZ_DATA(xyz, atoms, geomfile);

cout << inputfile << setw(15) << "N points = " << N_pts << endl;

// MAIN ALGORITHM

for (int i = 0; i < 3; i++)
{
  min[i] = xyz[0][i];          // set first point
  max[i] = xyz[0][i];
}
for (int i = 0; i < N_pts; i++)
{
  double atomicradius=0;
  switch (atoms[i][0])
  {
  case 'H':
       atomicradius = 1.06;       // Bondi calculated radii in angstroms
       break;
  case 'C':
       atomicradius = 1.53;
       if (atoms[i][1] == 'l')
         atomicradius = 1.75;
       break;
  case 'N':
       atomicradius = 1.46;
       break;
  case 'O':
       atomicradius = 1.42;
       break;
  case 'F':
       atomicradius = 1.40;
       break;
  case 'S':
       atomicradius = 1.80;
       break;
  case 'B':
       atomicradius = 1.65;
       if (atoms[i][1] == 'r')
         atomicradius = 1.87;
       break;
  case 'I':
       atomicradius = 2.04;
       break;
  case 'P':
      atomicradius = 1.86;
      break;
  }

  if (atomicradius == 0)
  {
      cerr << " ERROR.  One of the atoms is not recognized.\nExiting.\n";
      exit(1);
  }

  for (int j = 0; j < 3; j++)
  {
      if (min[j] > scalefactor*(xyz[i][j] - atomicradius))
        min[j] = scalefactor*(xyz[i][j] - atomicradius);
      if (max[j] < scalefactor*(xyz[i][j] + atomicradius))
        max[j] = scalefactor*(xyz[i][j] + atomicradius);
  }
}

for (int i = 0; i < 3; i++)
  Nsteps[i] = int(ceil((max[i]-min[i])/stepsize));
char cubefile[16];
strcpy(cubefile,basefiles[k].c_str());
char append3[] = ".gcub";
if (solvflag == 's')
  append3[1] = 's';
if (solvflag == 'x')
  append3[1] = 'x';
if (solvflag == 'c')
  append3[1] = 'c';
if (ipcmflag == 'i')
  append3[3] = 'i';

strcat(cubefile, append3);
ofstream ofileobj(inputfile, ios::app);
if (ofileobj == 0)
  exit(1);
ofileobj.precision(6);
ofileobj.setf(ios::fixed);
if (solvflag == 'x' || solvflag == 'c')
ofileobj << "midi/" << cubefile << endl;
else
  ofileobj << "jobsDx/" << cubefile << endl;
ofileobj << setw(5) << "07" << setw(12) << min[0] << setw(12) << min[1]
          << setw(12) <<  min[2] << endl;
ofileobj << setw(5) << Nsteps[0] << setw(12) <<  stepsize << setw(12)
          << 0.0 << setw(12) <<  0.0 << endl
          << setw(5) << Nsteps[1] << setw(12) << 0.0 << setw(12)
          << stepsize << setw(12) << 0.0 << endl
          << setw(5) << Nsteps[2] << setw(12) << 0.0 << setw(12) << 0.0
```

```
            << setw(12) << stepsize << endl << endl;

    for (int i = 0; i < N_pts; i++)
      delete [] xyz[i];             // memory management
    delete [] atoms;
  }
  delete [] basefiles;              // memory management

  return 0;
}

int GET_N_MOLEC(string filelist)
{
  int N = 0;
  ifstream infile(filelist.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << filelist << "'?? "
         << "I can't find it.\nExiting.\n\n";
    exit(1);
  }
  string dummy;
  while (infile >> dummy)
    N++;
  cout << "\nN molec = " << N;
  return N;
}


void READ_FILENAMES(int N, string filelist, string *filenames)
{
  ifstream infile(filelist.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << filelist << "'?? "
         << "I can't find it.\nExiting.\n\n";
    exit(1);
  }
  for (int i = 0; i < N; i++)
    infile >> filenames[i];
}


int FIND_N_PTS(char filename[])
{
  int N=0;
  ifstream infile(filename, ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << filename << "'??  "
         << "\nExiting.\n\n";
    exit(1);
  }
  infile >> N;
  return N;
}

void READ_XYZ_DATA(double **xyz, string *atoms, char filename[])
{
  ifstream infile(filename, ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << filename << "'??  "
         << "\nExiting.\n\n";
    exit(1);
  }

  int N = 0;
  infile >> N;
  for (int i = 0; i < N; i++)
  {
    infile >> atoms[i];
    for (int j = 0; j < 3; j++)
      infile >> xyz[i][j];
  }
}
```

getiso.pro

```
device, retain=2, decomposed=0
isodensity = 0.0004     ;units = [electrons/bohr^3]
isodensity1 = 0.0001     ;units = [electrons/bohr^3]

filename = ''
read, filename, prompt='Enter filename without the .cub suffix: '
cubefile = filename + '.ccub'
ivfile = ''
ivfile = filename + '.cic'
ivfile1 = ''
ivfile1 = filename + '.cic1'
ncfile = ''
ncfile = filename + '.cnc'

openr, 1, cubefile
header = strarr(2)
readf, 1, header  ;put 2 comment lines in var header
print, '"' + header + '"'

readf, 1, n_atoms, z0, y0, x0
n_atoms = fix(n_atoms)     ; conversion to integer
readf, 1, zdim, delz
```

```
readf, 1, ydim, dummy, dely
readf, 1, xdim, dummy, dummy, delx
nuclei = make_array(5, n_atoms, /double, value = 0)
readf, 1, nuclei

for i = 0, n_atoms-1 do begin $  ;convert to angstroms
 for j = 2,4 do nuclei(j,i) = nuclei(j,i)*0.529177
endfor

bigcube = make_array(xdim,ydim,zdim, /double, value = 0)
readf, 1, bigcube      ;cube units, distance = [bohr]
close, 1             ;density = [electrons/bohr^3]

xd = xdim       ;look at "slices"
yd = ydim
zd = zdim

cube = make_array(xd,yd,zd, /double, value = 0)
cube = bigcube[0:xd-1, 0:yd-1, 0:zd-1]
maxdim = xdim
if ydim > maxdim then maxdim = ydim
if zdim > maxdim then maxdim = zdim
s = size(cube)
scale3, xrange=[0,maxdim], yrange=[0,maxdim], $
  zrange=[0,maxdim], ax=0, az=45

 ;find 1st isosurface

shade_volume, cube, isodensity, v, p, /low
y_image = polyshade(v, p, /t3d)
tv, y_image
nframes = 50
for i = 0, nframes-1 do begin & $
 t3d, tr=[-.5,-.5,-.5], rot=[-360./nframes, -180./nframes, 120./nframes] & $
 t3d, tr=[.5,.5,.5] & $
 tv, polyshade(v,p,/t3d) & $
endfor

size_v = size(v)
v_bohr = make_array(3, size_v(2), /double, value = 0)
v_ang = make_array(3, size_v(2), /double, value = 0)

 ;in the following for loop, the indices for the
 ;vertices are corrected to correspond to the
 ;standard orientation of the nuclear positions
for i = 0, size_v(2)-1 do begin  ;convert to bohr
 v_bohr(2,i) = v(0,i)*delx + x0
 v_bohr(1,i) = v(1,i)*dely + y0
 v_bohr(0,i) = v(2,i)*delz + z0
 v_ang(0,i) = v_bohr(0,i)*0.529177 ;convert to angstroms
 v_ang(1,i) = v_bohr(1,i)*0.529177
 v_ang(2,i) = v_bohr(2,i)*0.529177
endfor

length_v = string(size_v(2), format='(i6)')
openw, 2, ivfile
printf, 2, v_ang, format='(3f12.6)'
close, 2
print, length_v + ' isosurface vertices printed to file ' + ivfile

 ;find 2nd isosurface

shade_volume, cube, isodensity1, v1, p1, /low
y_image1 = polyshade(v1, p1, /t3d)
tv, y_image1

for i = 0, nframes-1 do begin & $
 t3d, tr=[-.5,-.5,-.5], rot=[-180./nframes, -120./nframes,0] & $
 t3d, tr=[.5,.5,.5] & $
 tv, polyshade(v1,p1,/t3d) & $
endfor

size_v1 = size(v1)
v_bohr1 = make_array(3, size_v1(2), /double, value = 0)
v_ang1 = make_array(3, size_v1(2), /double, value = 0)

 ;in the following for loop, the indices for the
 ;vertices are corrected to correspond to the
 ;standard orientation of the nuclear positions
for i = 0, size_v1(2)-1 do begin  ;convert to bohr
 v_bohr1(2,i) = v1(0,i)*delx + x0
 v_bohr1(1,i) = v1(1,i)*dely + y0
 v_bohr1(0,i) = v1(2,i)*delz + z0
 v_ang1(0,i) = v_bohr1(0,i)*0.529177 ;convert to angstroms
 v_ang1(1,i) = v_bohr1(1,i)*0.529177
 v_ang1(2,i) = v_bohr1(2,i)*0.529177
endfor

length_v1 = string(size_v1(2), format='(i6)')
openw, 2, ivfile1
printf, 2, v_ang1, format='(3f12.6)'
close, 2
print, length_v1 + ' isosurface vertices printed to file ' + ivfile1

; the following function rewrites the nuc coord file

length_a = string(n_atoms, format='(i3)')
openw, 3, ncfile
printf, 3, n_atoms
printf, 3
for i = 0, n_atoms-1 do begin $
 if nuclei(0,i) eq 1 then printf, 3, format='("H  ", 3f12.6)', nuclei(2,i), nuclei(3,i), nuclei(4,i)
 if nuclei(0,i) eq 6 then printf, 3, format='("C  ", 3f12.6)', nuclei(2,i), nuclei(3,i), nuclei(4,i)
 if nuclei(0,i) eq 7 then printf, 3, format='("N  ", 3f12.6)', nuclei(2,i), nuclei(3,i), nuclei(4,i)
 if nuclei(0,i) eq 8 then printf, 3, format='("O  ", 3f12.6)', nuclei(2,i), nuclei(3,i), nuclei(4,i)
 if nuclei(0,i) eq 9 then printf, 3, format='("F  ", 3f12.6)', nuclei(2,i), nuclei(3,i), nuclei(4,i)
 if nuclei(0,i) eq 15 then printf, 3, format='("P  ", 3f12.6)', nuclei(2,i), nuclei(3,i), nuclei(4,i)
 if nuclei(0,i) eq 16 then printf, 3, format='("S  ", 3f12.6)', nuclei(2,i), nuclei(3,i), nuclei(4,i)
 if nuclei(0,i) eq 17 then printf, 3, format='("Cl ", 3f12.6)', nuclei(2,i), nuclei(3,i), nuclei(4,i)
 if nuclei(0,i) eq 35 then printf, 3, format='("Br ", 3f12.6)', nuclei(2,i), nuclei(3,i), nuclei(4,i)
 if nuclei(0,i) eq 53 then printf, 3, format='("I  ", 3f12.6)', nuclei(2,i), nuclei(3,i), nuclei(4,i)
endfor
close, 3
print, length_a + ' nuclei positions printed to file ' + ncfile
print, ' (all output units are in angstroms)'

end
```

intfieldset.c

```
// compile with point.c using, e.g.:
// g++ -o intfieldset intfieldset.c point.c

#include <iostream>
#include <string>
#include <fstream>
#include <math.h>
#include <iomanip>
#include <cstdlib>      // the updated assert() fn
#include "point.h"

using namespace std;

int GET_N_MOLEC(string);
void READ_FILENAMES(int, string, string *);
int FIND_N_TESS(string);
int FIND_N_NUC(string);
void READ_NUC_DATA(int, point *, string *, string);
void READ_TESS_DATA(int, point *, string);
void READ_U_E_DATA(int, double *, point *, string);
void DISTANCE_TEST(int, int, point *, string *, point *, double *, point *);
void CALC_NORMAL(int, int, point *, int, point *, point &, double &);
void WRITE_LOG(string, string, int, int, point *, double *, point *, point *, double *, double *);

int main()
{
  int N_molec = 0;           // number of molecules
  int N_nuc = 0;            // number of nuclei
  int N_tess = 0;           // number of isosurface tessarae points
  double bohr2ang = 0.529177; // convert distances in au to Angstroms
  cout.setf(ios::fixed);
  string filelist = "filelist.dat";      // filename of molec name data
string logfile = "intfieldset.log";     // filename of diagnostic log
string outfile = "intfieldset.out";     // filename for output

N_molec = GET_N_MOLEC(filelist);
string app_nuc;
cout << "\nenter the nuclear geometry file suffix (e.g. 'cnc')\n? ";
cin >> app_nuc;
string app_fd;
cout << "\nenter the field data file suffix (e.g. 'cfd')\n? ";
cin >> app_fd;

string *basefiles;                      // vector of molec names
basefiles = new string[N_molec];
READ_FILENAMES(N_molec, filelist, basefiles);
ofstream logfileobj(logfile.c_str(), ios::out); // clear the current logfile
logfileobj.close();
ofstream outfileobj(outfile.c_str(), ios::out); // clear the current outfile
outfileobj << "\nEscale = [ ...  % [Ntess |E*U| E*U U*U]\n";
outfileobj.close();

cout << "\nEscale = [ ...  % [Ntess |E*U| E*U U*U]\n";
for (int k = 0; k < N_molec; k++)
{
 string nucfile, fdfile;
 nucfile = basefiles[k];           // define nuclear data filename
 nucfile.append(".");
 nucfile.append(app_nuc);
 fdfile = basefiles[k];          // define field data filename
 fdfile.append(".");
 fdfile.append(app_fd);

 N_tess = FIND_N_TESS(fdfile);  // find number of solv surface tess
 N_nuc = FIND_N_NUC(nucfile);      // find number of nuclei
 // allocate a bunch of dynamic mem; points are initialized to 0 by default
 point *nuc;
 nuc = new point[N_nuc];        // cartesian locations of nuclei
 string *nuctype;
 nuctype = new string[N_nuc];  // names of nuclei
 point *tess;
 tess = new point[N_tess];      // cartesian locations of tesserae
 point *n;
 n = new point[N_tess];         // unit normal vectors of tesserae
 double *ane;
 ane = new double[N_tess];      // index of active (counted) normal
                           // estimates for each normal average
 double *U;
 U = new double[N_tess];        // scalar potential, au
 point *E;
 E = new point[N_tess];         // vector coordinates of E field, au
 double *El;
 El = new double[N_tess];       // normal E field vector lengths, au
```

```
for (int i = 0; i < N_tess; i++)   // initialization of U
{
  U[i] = 0.0;
  ane[i] = 20;
}

READ_NUC_DATA(N_nuc, nuc, nuctype, nucfile);
READ_TESS_DATA(N_tess, tess, fdfile);
READ_U_E_DATA(N_tess, U, E, fdfile);

DISTANCE_TEST(N_nuc, N_tess, nuc, nuctype, tess, U, E);

for (int i = 0; i < N_tess; i++)
{
  if (E[i] != 0)
       CALC_NORMAL(i, N_tess, tess, N_nuc, nuc, n[i], ane[i]);
  El[i] = n[i].dot(E[i]);          // scalar projection of E onto n
}

// write some diagnostic stats to a log file
WRITE_LOG(logfile, basefiles[k], N_nuc, N_tess, nuc, ane, n, E, El, U);

// integrate over SAS
double Pla = 0.0;
double Plb = 0.0;
double Plc = 0.0;
double Pld = 0.0;
double Ubar = 0.0;

for (int i = 0; i < N_tess; i++)
  Ubar = Ubar + U[i];
Ubar = Ubar/N_tess;

for (int i = 0; i < N_tess; i++)
{
  Pla = Pla + sqrt(pow(El[i]*U[i],2.0));
  //     Plb = Plb + sqrt(pow(U[i] - Ubar,2.0));  // Brink method
  Plb = Plb + El[i]*U[i];
  Plc = Plc + U[i]*U[i];
}

cout.precision(2);

ofstream outfileobj(outfile.c_str(), ios::app);
outfileobj.setf(ios::fixed);
outfileobj.precision(2);

cout << "[" << N_tess << setw(8) << 3.1462*Pla << setw(8) << 3.1462*Plb
       << setw(8) << 1.8631*Plc;
outfileobj << "[" << N_tess << setw(8) << 3.1462*Pla << setw(8) << 3.1462*Plb
           << setw(8) << 1.8631*Plc;
  if (k < N_molec-1)
  {
   cout << "]; ..." << setw(3) << "% " << basefiles[k] << endl;
   outfileobj << "]; ..." << setw(3) << "% " << basefiles[k] << endl;
  }
  else
  {
   cout << "]]; " << setw(3) << "% " << basefiles[k] << endl;
   outfileobj << "]]; " << setw(3) << "% " << basefiles[k] << endl;
  }
  outfileobj.close();

  delete [] nuc;                // free allocated memory
  delete [] tess;
  delete [] U;
  delete [] E;
  delete [] El;
  delete [] n;
 }
 delete [] basefiles;

 return 0;
}

int GET_N_MOLEC(string filelist)
{
 int N = 0;
 ifstream infile(filelist.c_str(), ios::in);
 if (!infile)
 {
  cerr << "\nWhere is '" << filelist << "'?? "
     << "I can't find it.\nExiting.\n\n";
  exit(1);
 }
 string dummy;
 while (infile >> dummy)
  N++;
 cout << "\nN molec = " << N << " found in database " << filelist << endl;
 return N;
}

void READ_FILENAMES(int N, string filelist, string *filenames)
{
 ifstream infile(filelist.c_str(), ios::in);
 if (!infile)
 {
  cerr << "\nWhere is '" << filelist << "'?? "
     << "I can't find it.\nExiting.\n\n";
  exit(1);
 }
```

```
  for (int i = 0; i < N; i++)
    infile >> filenames[i];
}

int FIND_N_TESS(string data)
{
  int N = 0;
  ifstream infile(data.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere the heck is '" << data.c_str() << "'?!!  "
         << "I can't find it.\nA program needs data to run, you know..."
         << "\nExiting.\n\n";
    exit(1);
  }
  string dummy1, dummy2;
  while (infile >> dummy1)
  {
    if (dummy1 == "Properties" && dummy2 == "Electrostatic")
    {
      while (infile >> dummy1 && dummy1 != "Calculate")
        if (dummy1 == "Read-in")
          N++;
    }
    dummy2 = dummy1;
  }
  infile.close();
  return N;
}

int FIND_N_NUC(string filename)
{
  ifstream infile(filename.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << filename.c_str() << "'?? "
         << "I can't find it.\nExiting.\n\n";
    exit(1);
  }
  int N;
  infile >> N;
  infile.close();
  return N;
}

void READ_NUC_DATA(int N, point *nuc, string *nucname, string filename)
{
  ifstream infile(filename.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << filename.c_str() << "'?? "
```

```
        << "I can't find it.\nExiting.\n\n";
    exit(1);
  }
  string dummy;
  infile >> dummy;
  for (int i = 0; i < N; i++)
  {
    infile >> nucname[i];
    infile >> nuc[i];
  }
  infile.close();
}

void READ_TESS_DATA(int N_tess, point *tess, string icfile)
{
  ifstream infile(icfile.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << icfile.c_str() << "'?? "
        << "I can't find it.\nExiting.\n\n";
    exit(1);
  }

  string dummy1, dummy2;
  while (infile >> dummy1)
  {
    if (dummy2 == "Read-in" && dummy1 == "Center")
    {
      int i = 0;
      while (i < N_tess)
      {
        infile >> dummy1;
        if (dummy1 == "at")
        {
          infile >> tess[i];          // already in units of angstroms
          i++;
        }
      }
      if (i == N_tess-1)
        infile.close();
    }
    dummy2 = dummy1;
  }
  infile.close();
}

void READ_U_E_DATA(int N_tess, double *U, point *E, string fdfile)
{
  ifstream infile(fdfile.c_str(), ios::in);
  if (!infile)
  {
   cerr << "\nWhere is file '" << fdfile.c_str() << "'?? "
      << "I can't find it.\nExiting.\n\n";
   exit(1);
 }
 string dummy1, dummy2;
 while (infile >> dummy1)
 {
   if (dummy1 == "---------------------------------------------------------" && dummy2 == "Z")
   {
     int i = 0;
     int flag = 0;
     infile >> dummy1;
     while (i < N_tess)
     {
          if (flag == 0)
          {
            infile >> dummy1;
            if (dummy1 == "Atom")
            {
              infile >> dummy1;
              infile >> dummy1;
              infile >> dummy1;
              infile >> dummy1;
              infile >> dummy1;
            }
            else
            {
              flag = 1;
              U[i] = atof(dummy1.c_str());
              infile >> E[i];
              infile >> dummy1;
 //         cout << "\nU[" << i << "] = " << U[i];   // first point
 //         cout << "\nE[" << i << "] = " << E[i];
              i++;
            }
          }
          if (flag == 1)
          {
            infile >> U[i];
            infile >> E[i];
            infile >> dummy1;
            i++;
          }
          if (i == N_tess)
            infile.close();
     }
   }
   dummy2 = dummy1;
 }
 infile.close();
}
```

```
void DISTANCE_TEST(int N_nuc, int N_tess, point *nuc, string *nuctype, point *tess, double *U,
point *E)
{
  point origin;
  for (int i = 0; i < N_nuc; i++)
  {
    double atomicradius = 0;

    if (nuctype[i] == "H")
      atomicradius = 1.06;          // Bondi calculated radii in angstroms
    else if (nuctype[i] == "C")
      atomicradius = 1.53;
    else if (nuctype[i] == "Cl")
      atomicradius = 1.75;
    else if (nuctype[i] == "N")
      atomicradius = 1.46;
    else if (nuctype[i] == "O")
      atomicradius = 1.42;
    else if (nuctype[i] == "F")
      atomicradius = 1.40;
    else if (nuctype[i] == "S")
      atomicradius = 1.80;
    else if (nuctype[i] == "B")
      atomicradius = 1.65;
    else if (nuctype[i] == "Br")
      atomicradius = 1.87;
    else if (nuctype[i] == "I")
      atomicradius = 2.04;
    else if (nuctype[i] == "P")
      atomicradius = 1.86;
    else
    {
      cerr << " ERROR. One of the atoms is not recognized.\nExiting.\n";
      exit(1);
    }
    double distance = 0.0;
    for (int j = 0; j < N_tess; j++)
    {
      if (tess[j].dist(nuc[i]) < 0.5*atomicradius)
      {
          cout << "Tessera " << j << " is " << tess[j].dist(nuc[i])
              << " Ang from " << nuctype[i] << ", U = " << U[j]
              << ". U and E values rejected." << endl;
          U[j] = 0.0;          // suppress potential to zero
          E[j] = 0.0;          // suppress elect field to zero
      }
    }
  }
}
```

```
void CALC_NORMAL(int k, int N_tess, point *tess, int N_nuc, point *nuc, point &n, double
&estindex)
{
 point origin;              // default constructor assigns 0,0,0
 double pi = 3.14159265;
 // first find points which define the local surface
 point a1 = 2*tess[k];      // to be closest local grid points
 point a2 = 2*tess[k];
 point a3 = 2*tess[k];
 point a4 = 2*tess[k];
 point a5 = 2*tess[k];
 point a6 = 2*tess[k];
 for (int i = 0; i < N_tess; i++)
 {
  if (tess[i].dist(tess[k]) < tess[k].dist(a1) && i != k)
  {
   a6 = a5;
   a5 = a4;
   a4 = a3;
   a3 = a2;
   a2 = a1;
   a1 = tess[i];
  }
  else if (tess[i].dist(tess[k]) < tess[k].dist(a2) && i != k)
  {
   a6 = a5;
   a5 = a4;
   a4 = a3;
   a3 = a2;
   a2 = tess[i];
  }
  else if (tess[i].dist(tess[k]) < tess[k].dist(a3) && i != k)
  {
   a6 = a5;
   a5 = a4;
   a4 = a3;
   a3 = tess[i];
  }
  else if (tess[i].dist(tess[k]) < tess[k].dist(a4) && i != k)
  {
   a6 = a5;
   a5 = a4;
   a4 = tess[i];
  }
  else if (tess[i].dist(tess[k]) < tess[k].dist(a5) && i != k)
  {
   a6 = a5;
   a5 = tess[i];
  }
  else if (tess[i].dist(tess[k]) < tess[k].dist(a6) && i != k)
   a6 = tess[i];
```

```
}

// locate nearest nucleus
point near_nuc = nuc[0];
for (int i = 1; i < N_nuc; i++)
  if (tess[k].dist(nuc[i]) < tess[k].dist(near_nuc))
    near_nuc = nuc[i];
//  cout << "nearest nuc = " << near_nuc << endl;
// evaluate normal vector, n, of point k
int N_nest = 20;
point n_est[20];
n_est[0] = xmult(a1-a2,a2-a3);
n_est[1] = xmult(a1-a2,a2-a4);
n_est[2] = xmult(a1-a2,a2-a5);
n_est[3] = xmult(a1-a2,a2-a6);
n_est[4] = xmult(a1-a3,a3-a4);
n_est[5] = xmult(a1-a3,a3-a5);
n_est[6] = xmult(a1-a3,a3-a6);
n_est[7] = xmult(a1-a4,a4-a5);
n_est[8] = xmult(a1-a4,a4-a6);
n_est[9] = xmult(a1-a5,a5-a6);
n_est[10] = xmult(a2-a3,a3-a4);
n_est[11] = xmult(a2-a3,a3-a5);
n_est[12] = xmult(a2-a3,a3-a6);
n_est[13] = xmult(a2-a4,a4-a5);
n_est[14] = xmult(a2-a4,a4-a6);
n_est[15] = xmult(a2-a5,a5-a6);
n_est[16] = xmult(a3-a4,a4-a5);
n_est[17] = xmult(a3-a4,a4-a6);
n_est[18] = xmult(a3-a5,a5-a6);
n_est[19] = xmult(a4-a5,a5-a6);

// make sure each normal vector estimate is pointing "outwards"
for (int i = 0; i < N_nest; i ++)
//  if (n_est1.dot(tess[k]-near_nuc) < 0)
  if (origin.dist(n_est[i]+tess[k]-near_nuc) < origin.dist(tess[k]-near_nuc))
    n_est[i] = -n_est[i];

for (int i = 0; i < N_nest; i ++)
  n_est[i] = (1/n_est[i].dist(origin))*n_est[i];    // normalize normal est's
for (int i = 0; i < N_nest; i ++)
  n = n + n_est[i];                      // take prelim average of normals
assert(n != origin);                     // make sure something is there
n = (1/n.dist(origin))*n;                // normalize prelim avg

// if any of the vector estimates differs from avg by > 20 deg, bury it.
for (int i = 0; i < N_nest; i ++)
  if (acos(n.dot(n_est[i])/(n.dist(origin)*n_est[i].dist(origin))) > 2*pi/18)
  {
    n_est[i] = origin;
    estindex = estindex - 1;
  }

//  for (int i = 0; i < N_nest; i ++)        // how many did we bury?
//    cout << "n_est[" << i << "] = " << n_est[i] << endl;
 for (int i = 0; i < N_nest; i ++)
   n = n + n_est[i];                      // take average of normal estimates
 assert(n != origin);                      // make sure something is there
 n = (1/n.dist(origin))*n;                  // normalize the final normal vector
}

void WRITE_LOG(string logfile, string molecname, int N_nuc, int N_tess, point *nuc, double
*ane, point *n, point *E, double *El, double *U)
{
 ofstream outfile(logfile.c_str(), ios::app);
 if (outfile == 0)
  exit(1);
 outfile.setf(ios::fixed);
 outfile << "INTEGRATION SUMMARY FOR " << molecname.c_str() << "." << endl;
 point origin;
 int nucflag = 1;
 for (int i = 0; i < N_nuc; i++)
 {
  if (nuc[i] != origin);
  nucflag = 0;
 }
 if (nucflag == 1)
  outfile << "WARNING. No nuclei were found for this solute.\n";
 cout.precision(4);
 int anetensum = 0;
 int anefivesum = 0;
 int Enullflag = 0;
 int Elnullflag = 0;
 int Elbigflag = 0;
 int Elnanflag = 0;
 int Unullflag = 0;
 for (int i = 0; i < N_tess; i++)
 {
  if (ane[i] < 10)
   anetensum++;
  if (ane[i] < 5)
   anefivesum++;
  if (E[i] == origin)
   Enullflag++;
  if (El[i] == 0)
   Elnullflag++;
  if (El[i] > 1)
   Elbigflag++;
  if (El[i] != El[i])      // check for NaN
  {
   Elnanflag++;
```

```
    cout << "Failed projection of E[" << i << "] = " << E[i] << "\nis at n = " << n[i] << ". E
projection rejected at this point." << endl;
    El[i] = 0;
   }
   if (U[i] == 0)
    Unullflag++;
  }

  outfile << endl << anetensum
          << " normals contained fewer than 10 estimates.\n"
          << anefivesum << " normals contained fewer than 5 estimates.\n";

  if (Enullflag != 0 || Elnullflag != 0 || Unullflag != 0)
  {
   outfile << "WARNING.\n" << Enullflag << " E vectors were found to be zero."
           << endl << Elnullflag << " E projections were found to be zero."
           << endl << Unullflag << " U points were found to be zero.\n";
  }
  if (Elbigflag != 0)
   outfile << "WARNING.\n" << Elbigflag << " E projections were greater than 1.\n";
  if (Elnanflag != 0)
   outfile << "WARNING.\n" << Elnanflag << " E projections were NaN.\n";
  outfile << "
======================================================================\n\n";

  outfile.close();
}
```

dipoleset.c

```
#include <iostream>
#include <string>
#include <fstream>

// This code simply extracts calculated dipole moments from gaussian output files.

using namespace std;

int GET_N_MOLEC(string);
void READ_FILENAMES(int, string, string *);
void READ_DIPOLE(string, string, char, double &);

int main()
{
 int N_molec = 0;           // number of molecules
 string filelist = "filelist.dat";     // filename of molec name data
 N_molec = GET_N_MOLEC(filelist);
 string logfile = "dipoleset.log";       // logged output filename
 string suffix;
 cout << "\n\nEnter the file suffix (e.g. spot)\n? ";
```

```
  cin >> suffix;
  char sflag;
  cout << "\nSelect (g)as or (s)olvent phase calculation\n? ";
  cin >> sflag;
  string *basefiles;                    // vector of molec names
  basefiles = new string[N_molec];
  READ_FILENAMES(N_molec, filelist, basefiles);

  cout << "\ndipole = [ ... % (Debyes)" << endl;
  for (int k = 0; k < N_molec; k++)
  {
    double dipole = 0;
    READ_DIPOLE(basefiles[k], suffix, sflag, dipole);
    cout.precision(3);
    cout << "[" << dipole << "]; ... % " << basefiles[k] << endl;
  }
  return 0;
}

int GET_N_MOLEC(string filelist)
{
  int N = 0;
  ifstream infile(filelist.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << filelist << "'?? "
         << "I can't find it.\nExiting.\n\n";
    exit(1);
  }
  string dummy;
  while (infile >> dummy)
    N++;
  cout << "\nN molec = " << N;
  return N;
}

void READ_FILENAMES(int N, string filelist, string *filenames)
{
  ifstream infile(filelist.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << filelist << "'?? "
         << "I can't find it.\nExiting.\n\n";
    exit(1);
  }
  for (int i = 0; i < N; i++)
    infile >> filenames[i];
}

void READ_DIPOLE(string basefile, string suffix, char sflag, double &dipole)
{
  string molecfile;
  molecfile = basefile;
  molecfile.append(".");
  molecfile.append(suffix);
  ifstream infile(molecfile.c_str(), ios::in);
  if (!infile)
  {
    cerr << "\nWhere is '" << molecfile.c_str() << "'? "
             << "I can't find it.\n\nExiting.\n\n";
    exit(1);
  }
  string dummy1, dummy2, dummy3;
  if (sflag == 'g')
  {
    while (infile >> dummy1)
    {
      if (dummy2 == "Dipole" && dummy1 == "moment")
      {
            while (infile >> dummy1)
        {
              if (dummy1 == "Tot=")
              {
                infile >> dipole;
                break;
              }
            }
      }
      dummy2 = dummy1;
    }
  }

  if (sflag == 's')
  {
    while (infile >> dummy1)
    {
      if (dummy3 == "SOLUTION" && dummy2 == "Dipole" && dummy1 == "moment")
      {
            while (infile >> dummy1)
        {
              if (dummy1 == "Tot=")
              {
                infile >> dipole;
                break;
              }
            }
      }
      dummy3 = dummy2;
      dummy2 = dummy1;
    }
  }
}
```

point.h

```
// file point.h

#include <iostream>
#include <assert.h>
using namespace std;      // to define function names of std c++

class point
{
friend class rect;
friend ostream &operator<<(ostream &out, const point &p);
friend istream &operator>>(istream &in, point &p);
friend point operator*(const double &s, const point &p);  // scalar mult
friend point operator*(const point &p, const double &s);  // scalar mult
friend point xmult(const point &n, const point &p);       // cross mult
public:
  point (double a = 0.0, double b = 0.0, double c = 0.0);  // default constructor
  point (const point &p);
  ~point(){};
  double getx();
  double gety();
  double getz();
  void print();
  point operator+(const point &p);             // + operator
  point operator-();                        // "unary minus" operator
  point operator-(const point &p);             // subtraction operator
  point &operator=(const point &p);             // = operator
  bool operator==(const point &p) const;        // == operator
  bool operator!=(const point &p) const;        // != operator
  double dist(const point &p);                 // find distance to point2
  double dot(const point &p);                  // dot product
private:
  double x, y, z;
};
```

point.c

```
// file point.c
// a pretty good point class. it is designed to hold
// and manipulate 3-d (cartesian) vectors and points.
// compile syntax is> g++ -o foo foo.c point.c

#include <iostream>
#include <iomanip>
#include <math.h>
#include "point.h"

point::point(double a, double b, double c) { x = a; y = b; z = c; }
```

```
point::point(const point &p) { x = p.x; y = p.y; z = p.z; }
double point::getx() { return x; }
double point::gety() { return y; }
double point::getz() { return z; }
void point::print() { cout << "(" << x << ", " << y << ", " << z << ")\n"; }

// math member functions
point point::operator+(const point &p2)    // input is const (see below)
{ return point(x+p2.x, y+p2.y, z+p2.z); }
point point::operator-() { return point(-x, -y, -z); }
point point::operator-(const point &p2)    // input is const (see below)
{ return point(x-p2.x, y-p2.y, z-p2.z); }
point &point::operator=(const point &p2)   // input to member fn is CONST so
{                                          // that member fns can be nested.
 if (this != &p2)
 {
 x = p2.x;
 y = p2.y;
 z = p2.z;
 }
 return *this;
}
bool point::operator==(const point &p_sim) const
{
  if (p_sim.x == x && p_sim.y == y && p_sim.z == z)
    return true;
  else
    return false;
}
bool point::operator!=(const point &p_sim) const
{ return ! (*this == p_sim); }
double point::dist(const point &p2)
{ return ( sqrt(pow((x - p2.x),2) + pow((y - p2.y),2) + pow((z - p2.z),2)) ); }
double point::dot(const point &p2) { return (x*p2.x + y*p2.y + z*p2.z); }

// friend functions

point operator*(const double &s, const point &p)  // scalar mult from front
{
  point h;
  h.x = s*p.x;
  h.y = s*p.y;
  h.z = s*p.z;
  return h;
}

point operator*(const point &p, const double &s)  // scalar mult from back
{
  point h;
  h.x = s*p.x;
  h.y = s*p.y;
 h.z = s*p.z;
 return h;
}

point xmult(const point &p1, const point &p2)     // cross mult
{
 point h;
 h.x = p1.y*p2.z - p1.z*p2.y;
 h.y = p1.z*p2.x - p1.x*p2.z;
 h.z = p1.x*p2.y - p1.y*p2.x;
 return h;
}

ostream &operator<<(ostream &out, const point &p)
{
 out.precision(4);
 out << p.x << setw(9) << p.y << setw(9) << p.z;
 return out; }

istream &operator>>(istream &in, point &p)
{ in >> p.x >> p.y >> p.z; return in; }
```