HCToolkit/EOS interface: an open source, multi-platform phase equilibria framework for exploring phase behaviour of complex mixtures

PETER MEULBROEK,[1] LARRY CATHLES[2] & WILLIAM A. GODDARD[3]

[1] *Mc 139-74, Beckman Institute, California Institute of Technology, Pasadena, CA 91125, USA (e-mail: meulbroek@wag.caltech.edu)*
[2] *Department of Geological Sciences, Cornell University, Ithaca, NY 14853, USA (e-mail: cathles@geology.cornell.edu)*
[3] *Mc 139-74, Beckman Institute, California Institute of Technology, Pasadena, CA 91125, USA (e-mail: wag@wag.caltech.edu)*

Abstract: We present a computational framework that can be used to estimate phase equilibria, equation of state properties and composition-dependent viscosity, aimed at the geochemical community for modelling reservoir processes, and at the chemical community to quickly estimate continuum properties of known mixtures. The framework presented is an extensible, component-based set of modules that can be used in calculating phase properties (volumes and densities, compositions and conditions of phase separation, Henry's Law constants and viscosities) at a wide range of pressures, temperatures and starting compositions. There are two pieces to the model: the HCToolkit (a set of Perl modules that act as computational engines) and the EOSInterface (an ActiveX wrapper for the HCToolkit) which allow the models in the HCToolkit to be usable in Microsoft Office programs. An application hosted in Microsoft Excel is included within the distribution.

The EOS models are adaptable for mixtures of arbitrary complexity, with number and types of components only limited by hardware. Model runs with mixtures of over 100 components have been tested, and are perfectly feasible. Implemented within the software are four equations of state; additional equations of state can easily be added. Also implemented are G^E models and mixture viscosities. Finally, the EOS models provide liquid–vapour flash calculations, liquid–liquid flash (designed for petroleum–aqueous solutions) and the generation of phase diagrams.

Fluid-phase equilibria/equation of state (EOS) models are powerful tools used to predict phase properties of mixtures where gathering experimental data is difficult, expensive or impractical. These models have seen widespread application in both chemical engineering and petroleum engineering. Within the petroleum engineering community, EOS models are ubiquitously used to predict the phase behaviour and PVT properties of reservoir petroleum fluids during production. They are primarily used in reservoir simulation, testing the efficacy of different exploitation strategies. Increasingly, rapid phase equilibria calculations are becoming a useful tool of the reservoir geochemist for such tasks as production allocation, reservoir continuity (Meulbroek 2002), common source evaluation and determining fluid history (di Primio 2002; Losh *et al.* 2002). These calculations typically rely on a cubic equation of state, such as Soave–Redlich–Kwong (SRK) driving a fluid-phase equilibria model. Though current commercial software implementations of these models exist, no general, flexible model has been released to the community. We present here the HCToolkit, a flexible, extensible open source framework for performing general FPE calculations. Implemented within the software are four equations of state: Van der Waals, Mathias (Mathias 1983), Aasberg-Petersen (Aasberg-Petersen & Stenby 1991) and the PSRK (Holderbaum & Gmehling 1991). Additional equations of state can easily be added, as detailed in this work. The EOS models are adaptable for mixtures of arbitrary complexity, with number and types of components only limited by hardware. Model runs with mixtures of over 100 components are routinely executed. Also implemented are the G^E models UNIFAC (Fredenslund *et al.* 1975) and modified UNIFAC (Gmehling *et al.* 1993) models. Viscosities are calculated using the Chung *et al.* method (Chung *et al.* 1988). Finally, the EOS models provide liquid–vapour flash calculations, liquid–liquid flash (designed for petroleum–aqueous solutions), and the generation of phase diagrams (both *P-T* and *P-X*).

As an extension to the HCToolkit, we also release the EOSInterface, an MS Excel-hosted extension of the HCToolkit that allows the user access to all calculation abilities from within the Excel spreadsheet.

From: CUBITT, J. M., ENGLAND, W. A. & LARTER, S. (eds) 2004. *Understanding Petroleum Reservoirs: towards an Integrated Reservoir Engineering and Geochemical Approach.* Geological Society, London, Special Publications, **237**, 89–98. 0305-8719/04/$15.00

Problem area

The design principles that resulted in the EOS-Interface are aimed at developing a flexible framework for continuum fluid calculations for the problem solver and developer. The goal was to develop a framework that could be used for any phase equilibrium problem including subsurface and industrial processes. To provide this flexibility, the framework is equation-agnostic; that is, any calculation that the framework performs can be done with any of the embedded equations of state, and additional equation of state models can be easily added to the code base. Early on, we realized that the rapid development environment provided by a scripting language would speed code production and provide built-in extensibility. We chose Perl—the most popular multi-platform scripting language available—after considering and rejecting Visual Basic (limited to the Microsoft Windows platforms), Python (much smaller developer base) and Java (higher development overhead). Perl is an open source dynamic scripting language with good debugging and profiling capabilities, and an extensive library of freely-available extensions (the CPAN library). If execution speed becomes a bottleneck in any application, developers can also transparently link compiled code within a Perl code.

In order for the code to be applicable to mixtures consisting of a wide and expandable number of compounds, we decided to store all data in a relational database and to provide a variety of access methods to that database. In fact, any database that can be accessed by Perl-DBI or ODBC methods (which include all major commercial and open source products) can be used to store compound data. Since many of these databases are network-accessible, the HCToolkit framework can be driven by a common institution-wide (or multi-institution) database to ensure consistency of properties and mixtures in a calculation. We also store mixture information in the database to ease collaboration in studying specific compositions. Hence, all researchers in a specific institution can connect to the same database, and all can use consistent fluid compositions (e.g. fluid '72') in numeric simulations.

Open source

The framework described here is open source (released under the GNU-GPL licence, http://www.gnu.org/licenses/gpl.html), a licensing method that has received a huge amount of publicity. Essentially, 'open source' means that the framework is released with source code, as a means of encouraging users to become active development partners (http://www.opensource.org/). All fair uses of the code including further distribution are allowed; however, any changes that a developer makes to the code must also be made available, in the spirit of collaboration. This is remarkably different from existing commercial licences, which forbid distribution and form a barrier to shared development.

In contrast to conventional wisdom, open source codes offer considerable advantages to software developers. If the goal is rapid framework improvement and extension, then an open source product advances much more quickly than a commercial framework, as all users of the framework are potentially developers. Open source codes also provide a guarantee to the user that the framework will not 'disappear'. Since the source code of a framework is available, if a development effort stalls at one location, another location can restart the development at a later date. Users can also customize the framework to their own needs and share those customizations.

Abilities of the framework

The HCToolkit provides implementations for four equations of state—the original Van der Waals model; the Mathias (Mathias 1983) model, which improves predictions for polar mixtures; the Aasberg-Petersen model (Aasberg-Petersen & Stenby 1991) which improves predictions around the critical point; and the predictive RKS model (Holderbaum & Gmehling 1991), which utilizes the UNIFAC model to eliminate the need for empirical fitting parameters and thus allows predictions of phase behaviour without additional PVT (pressure, temperature, composition) data. These models can each be used to predict molar volume, density and fugacity for mixtures at arbitrary PVT conditions. Also included are several excess Gibbs free energy models, which predict the Gibbs free energy of mixing and activity coefficients for arbitrary mixtures. The two models implemented are the original UNIFAC model (Fredenslund *et al.* 1975) and the modified UNIFAC model (Gmehling *et al.* 1993).

The HCToolkit also presents several algorithms for fluid-phase equilibria calculations. They include the Rijkers & Heidemann (1986) model, an unpublished liquid–liquid algorithm and the Bunz *et al.* (1991) three-phase model. The HCToolkit also includes the Chung *et al.*

model for predicting viscosity of both liquid and vapour mixtures.

Architecture

The HCToolkit is a set of Perl modules that implement four equations of state, two flash algorithms and a multi-component, multi-phase, temperature- and pressure-dependent viscosity prediction. The modules have been successfully run on MS Windows, Mac OSX and Redhat Linux. These modules can be called from another Perl code, or (via an ActiveX interface) from a front-end written in either Visual Basic or a VBA application such as Excel.

In open source software, every user is a potential developer. Hence, technical knowledge of program architecture becomes relevant to the individual user. The HCToolkit design utilizes 'plugin' architecture; wherever possible, families of related modules are written to instance abstract interfaces. The interfaces are: Phase::EquationofState, Phase::Database, Phase::Flash and Phase::Unifac, as shown in Figure 1.

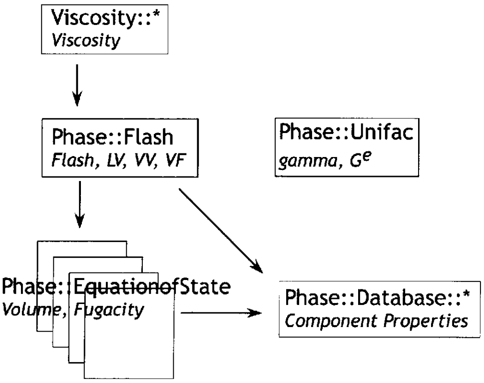


Fig. 1. Layout of the class structures in the HCToolkit.

An example of the application of abstract interfaces can be found in the Flash module. Figure 2 shows the class layout for the Flash 'engine', an interface to a flash calculation. The Flash engine takes two defined 'phases' (equation of state objects) plus a database that provides properties for the components of the phase, and calculates the equilibrium distribution of components among the phases. The equation of state used to calculate properties for each phase is specified at run-time via a 'Phase::EquationofState' object, and can even be different for different phases. All the EOS modules present the same interface to other modules by using abstract interfaces, allowing easy substitution of models for quick comparisons of calculation results. The Flash engine only expects the objects passed to it to conform to a generic interface (the 'Phase::EquationofState::Generic' interface, to be precise); any EOS object that is derived from that interface can be used in the flash calculation.

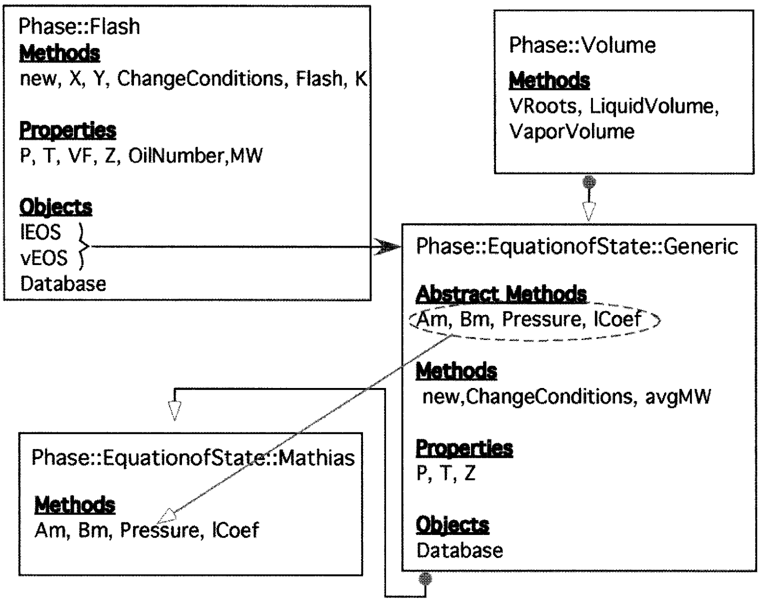


Fig. 2. Phase::Flash class layout.

Adding new equations of state to the framework is an easy and straightforward process. Two things tend to define most modern cubic equations of state: the mixing rules used to estimate incompressible volume (b) and attraction parameter (a), and the form of the attractive (second) term of the equation (see equation (1)). In order to add an additional EOS module, the user needs to supply an object-oriented Perl module that

(1) inherits from Phase::EquationofState::Generic;
(2) provides three functions: Am (the mixing rule for the attractive term coefficient from an equation of state), Bm (the mixing rule for the repulsive term coefficient from an equation of state) and Pressure (the pressure-explicit format for an equation of state);
(3) implements some functionality to return volume as a function of input variables composition, pressure and temperature (which is dependent on the form of the EOS).

For cubic equations of state, a pre-defined methodology allows easy implementation of (3), above. Most cubic equations of state are of the form

$$P = \frac{RT}{v - b} - \frac{a}{v^2 + uvb + wb^2} \tag{1}$$

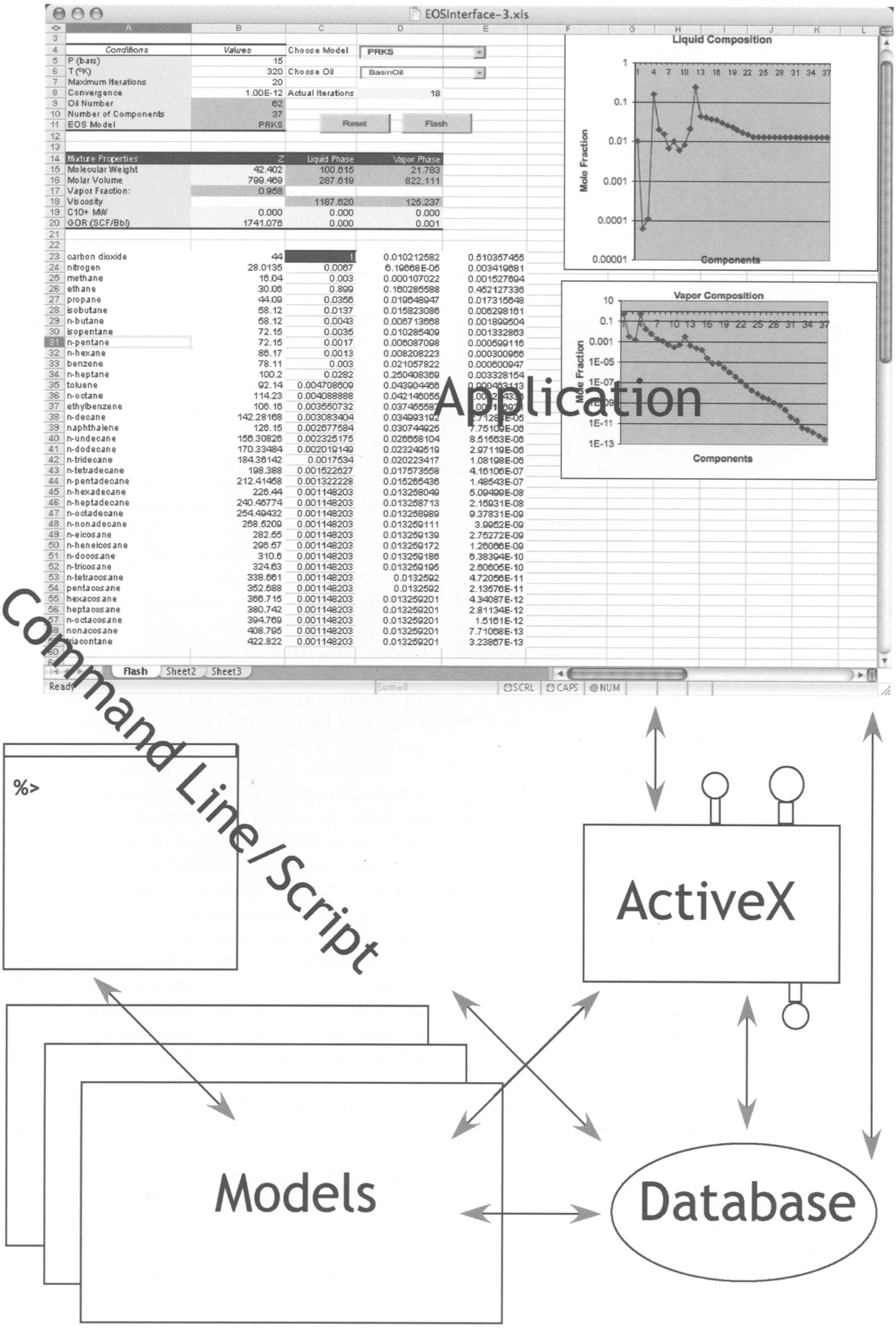
where P is pressure, R is the universal gas constant, v is molar volume, *lb* is the incompressible volume, a is the Van der Waals attractive parameter, and *lu* and *lw* are constants that define the equation of state (Poling 2000). For example, for the Van der Waals equation of state, $u = w = 0$, for the RKS equation of state, $u = 1$, $w = 0$. Equation (1) can be inverted to produce v as a dependant variable. An analytical solution to the general relationship $v = f(P, T, a, b, u, w)$ that returns one or three real roots is possible, and implemented within the HCToolkit framework in the module "Phase::-Volume". In order for a new cubic equation of state module to predict molar volume using these functions, it merely needs to implement a function called 'lCoef' in the new EOS module that returns a list of the appropriate u, w, P, T, a_{mix} and b_{mix} values. This function is called by "Phase::Volume" in its general solution of the relationship f, above.

The EOSInterface Worksheet

EOSInterface, an ActiveX/Excel interface for the HCToolkit, has a similar design philosophy to that of the HCToolkit. The EOSInterface Worksheet is a Microsoft Excel spreadsheet designed to interface with and form a front-end to the HCToolkit. The HCToolkit is completely functional without the EOSInterface: the EOSInterface sits on top of the HCToolkit, and exports most of the HCToolkit functionality in a more user-friendly environment. It is designed for the practical thermodynamicist who is content with simple flash calculations, and who is more comfortable with using an Excel spreadsheet than writing code in Perl. The EOSInterface interacts with the HCToolkit via an ActiveX gateway, compiled using ActiveState's PerlCtrl program. Since it is built on top of the HCToolkit, the EOSInterface provides the ability to do flash calculations using the four equations of state mentioned above. Figure 3 shows the object layout of the EOSInterface. Note that command-line scripting (via VBS) can also call the ActiveX object. This object shares with the HCToolkit the ability to interface with a large (>2500 species) database of compounds, including hydrocarbons, fluorocarbons and non-hydrocarbon species of interest to the geochemical community

PSRK equation of state

Of particular interest to the users of the HCToolkit and EOSInterface is the implementation of the PSRK EOS (Holderbaum & Gmehling 1991). In traditional equations of state that are used in petroleum engineering, models take as input pure species critical properties and empirical binary adjustment parameters (the '*kij*' parameters) that adjust for non-ideal interactions between species. These *kij* parameters are typically derived from PVT experiments, and have both temperature and composition dependence, which make them impractical for complex systems or systems with unusual components. However, the PSRK EOS uses a new mixing rule that depends on a group-additivity G^E model (UNIFAC) to estimate non-idealities, instead of using system/component-specific empirical data. UNIFAC considers molecules to be sets of interacting functional groups, the interactions of which are well known from a large number of studies (Fredenslund *et al.* 1975). To calibrate the PSRK EOS for any system only requires knowledge of molecular structure of all species to functional group resolution. Hence, this can be applied to complex systems with species whose interactions are not known. In the context of reservoir geochemistry, this means that, e.g. large biomarkers can be modelled as easily as well-known species such



	A	B	C	D	E
4	*Conditions*	*Values*	Choose Model	PRKS	
5	P (bars)	15			
6	T (°K)	320	Choose Oil	BasinOil	
7	Maximum Iterations	20			
8	Convergence	1.00E-12	Actual Iterations	18	
9	Oil Number	62			
10	Number of Components	37			
11	EOS Model	PRKS			
14	Mixture Properties	Z	Liquid Phase	Vapor Phase	
15	Molecular Weight	42.402	100.615	21.783	
16	Molar Volume	799.469	287.619	822.111	
17	Vapor Fraction:	0.958			
18	Viscosity		1187.620	125.237	
19	C10+ MW	0.000	0.000	0.000	
20	GOR (SCF/Bbl)	1741.076	0.000	0.001	
23	carbon dioxide	44	1	0.010212582	0.510357455
24	nitrogen	28.0135	0.0067	6.19668E-05	0.003419681
25	methane	16.04	0.003	0.000107022	0.001527694
26	ethane	30.06	0.899	0.160285588	0.452127336
27	propane	44.09	0.0356	0.019648947	0.017315648
28	isobutane	58.12	0.0137	0.015823086	0.006298161
29	n-butane	58.12	0.0043	0.006713668	0.001899504
30	isopentane	72.15	0.0035	0.010285409	0.001332863
31	n-pentane	72.15	0.0017	0.006087098	0.000599116
32	n-hexane	86.17	0.0013	0.008208223	0.000300966
33	benzene	78.11	0.003	0.021057822	0.000600947
34	n-heptane	100.2	0.0282	0.250408369	0.003328154
35	toluene	92.14	0.004708609	0.043904466	0.000463113
36	n-octane	114.23	0.004088888	0.042146055	[illegible]
37	ethylbenzene	106.16	0.003550732	0.037465587	[illegible]
38	n-decane	142.28168	0.003083404	0.034993192	[illegible]
39	naphthalene	126.15	0.002677584	0.030744925	7.75109E-06
40	n-undecane	156.30826	0.002325175	0.026658104	8.51563E-06
41	n-dodecane	170.33484	0.002019149	0.023249519	2.97119E-06
42	n-tridecane	184.36142	0.0017534	0.020223417	1.08198E-06
43	n-tetradecane	198.388	0.001522627	0.017573558	4.16106E-07
44	n-pentadecane	212.41458	0.001322228	0.015265436	1.48543E-07
45	n-hexadecane	226.44	0.001148203	0.013258049	5.09499E-08
46	n-heptadecane	240.46774	0.001148203	0.013258713	2.15931E-08
47	n-octadecane	254.49432	0.001148203	0.013258989	9.37831E-09
48	n-nonadecane	268.5209	0.001148203	0.013259111	3.9952E-09
49	n-eicosane	282.55	0.001148203	0.013259139	2.75272E-09
50	n-heneicosane	296.57	0.001148203	0.013259172	1.26066E-09
51	n-docosane	310.6	0.001148203	0.013259186	6.38394E-10
52	n-tricosane	324.63	0.001148203	0.013259195	2.60605E-10
53	n-tetracosane	338.661	0.001148203	0.0132592	4.72056E-11
54	pentacosane	352.688	0.001148203	0.0132592	2.13576E-11
55	hexacosane	366.715	0.001148203	0.013259201	4.34087E-12
56	heptacosane	380.742	0.001148203	0.013259201	2.81134E-12
57	n-octacosane	394.769	0.001148203	0.013259201	1.5161E-12
58	nonacosane	408.795	0.001148203	0.013259201	7.71068E-13
59	triacontane	422.822	0.001148203	0.013259201	3.23867E-13

Fig. 3. Layout of the parts of the EOSInterface.

```
# initialize the database
$db = Phase::Database::PostgreSQL->new(
      UserName => 'guest',
      Password => 'guest',
);

# load a fluid
$db->LoadOilByNumber(75);

# initialize the flash obj.
my $flash = Phase::Flash->new(
      P => $P, T => $T,
      Z => Standard::List->new($x, 1-$x),
      Database=>$db,
      lEOS => Phase::EquationofState::PSRK->new(  ),
      vEOS => Phase::EquationofState::PSRK->new(  ),
);

$flash->ChangeConditions(P => $p);

# do the flash calculations
$flash->Flash(MaxIterations => 20, Precision => 1e-9);
```

Fig. 4. Example HCToolkit code.

as *n*-alkanes. The model is also adaptable to pseudo-components, if the composition of the pseudo-component is known to the level of functional groups (i.e., fraction methyl groups, fraction aromatic carbon etc.).

The Perl scripting front-end

One of the most powerful aspects of the HCToolkit is its extensibility in Perl. This allows rapid development of new applications for specific tasks. For example, an included extension quickly calculates and plots a pressure–temperature phase diagram at user-specified conditions (PT-PhaseDiagram.pl, found in the /apps subdirectory of the distribution tree). In fact, a scripting application that does some sort of equilibrium calculation needs only do four things: create a database connection, use that connection to load a pre-defined fluid, initialize the flash object and do the flash calculation. These steps are shown in Figure 4.

Calculation examples

The implementation of the EOS components allows very tight control over the accuracy of calculations. As proof of concept, two examples of calculations are included. The first is a prediction of vapour–liquid equilibrium in a binary CO_2–cyclopentane system. This system is somewhat non-ideal, due to the quadrupole movement of CO_2. The predicted phase envelope is shown in Figure 5.

Figure 5 shows a (P, X) phase diagram for CO_2–cyclopentane at a constant temperature. The calculation shows predicted isofractions compared with experimental data (Keshtkar *et al.* 1997). Overall fits are very good. The actual algorithm chosen to predict the phase envelope is actually quite naïve; a grid-search calculation was performed over (composition, P) space using the example code 'apps/PX-PhaseDiagram.pl'. Flash calculations were performed on each grid node of a 50 × 50 node grid, and the location of the phase envelope was interpolated. Though much more sophisticated methods are available (various bubble/dew point methods and phase envelope 'walking' methods) from the literature, the point of showing the calculations is that a quick-and-dirty solution for questions of interest that produces very good answers can be coded within the HCToolkit, and calculated fairly rapidly (under 5 min on an Apple Powerbook G4 laptop). Furthermore, these more sophisticated methods can also be implemented within the HCToolkit, if desired.

The second example shows the solubility of methane in water. In general, cubic equations of state do quite poorly with aqueous solutions, since most are tuned to non-polar compositions such as hydrocarbon mixtures. The HCToolkit allows one to quickly compare the abilities of

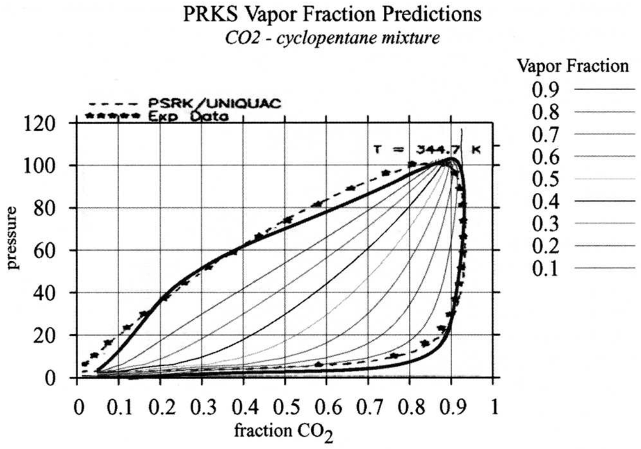


Fig. 5. Vapour–liquid equilibrium predictions using the HCToolkit. This figure shows the experimental data from (Keshtkar *et al.* 1997), an accurate binary calculation (UNIQUAC), the phase envelope (in bold) and vapour phase isofractions (shaded).

different equations of state. Figure 6 shows the prediction of methane solubility in water for two cubic equations of state—the Aasberg-Petersen & Stenby (1991) and the Mathias (1983)—compared with experimental data. The experimental data are taken from Li *et al.* (1997). At all temperatures, the Mathias EOS does a better job of predicting the solubility of methane in water (i.e. the amount of methane, by mole fraction, that can be dissolved in water as a function of pressure and temperature). This is not surprising, as the APS EOS is tuned for hydrocarbons and the Mathias EOS has a polar correction term for water mixtures. Unfortunately, the Mathias EOS only works well for the lowest temperature studied and works increasingly badly for higher temperatures and higher concentrations of CH_4.

Field example

Unlike a monolithic code, the HCToolkit is designed to ease the development of tools to

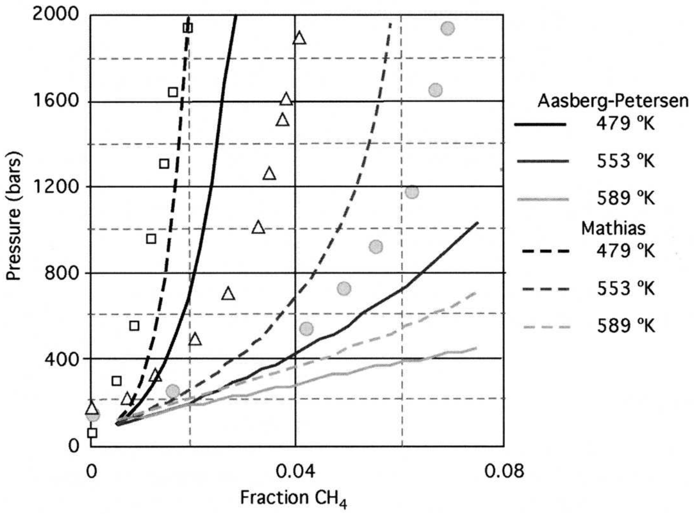


Fig. 6. Predictions of methane solubility in water. Shown is the bubble point pressure as a function of mixture composition (in mole fraction methane) at three different temperatures for two equations of state compared with experimental data.

solve phase-related problems in chemistry and geochemistry. HCToolkit comes with several example programs that are general, but can be used to address field-related questions. One such script is PT-PhaseDiagram.pl (found in the /apps directory), a script that calculates the phase distribution of a given mixture. These examples are not meant to be extensive; the toolkit is designed for the developer/user, giving the ability to quickly *develop* scripts to handle difficult problems.

In petroleum engineering and exploration geochemistry, equations of state are applied to 'PVT'-type compositions, which consist of several real components and a collection of 'pseudo-components' that are mixtures of undifferentiated compounds. Actually doing phase calculations with pseudo-components is only accurate for closed systems such as a reservoir (Meulbroek 2003), since pseudo-components are typically constructed of components that might have a similar molecular weight, but different partial pressures (fugacities). To better predict the complex partitioning and fractionation possible in an open system such as a migration conduit or a producing reservoir, one would prefer to use a more detailed specification, such as a GC analysis, that allows a more chemically-sensible lumping scheme. There are many problems with this approach, however. GC analyses are performed on 'dead' oils, where a large fraction of volatiles have separated and been lost from the fluid whose composition one is measuring. These compounds are precisely

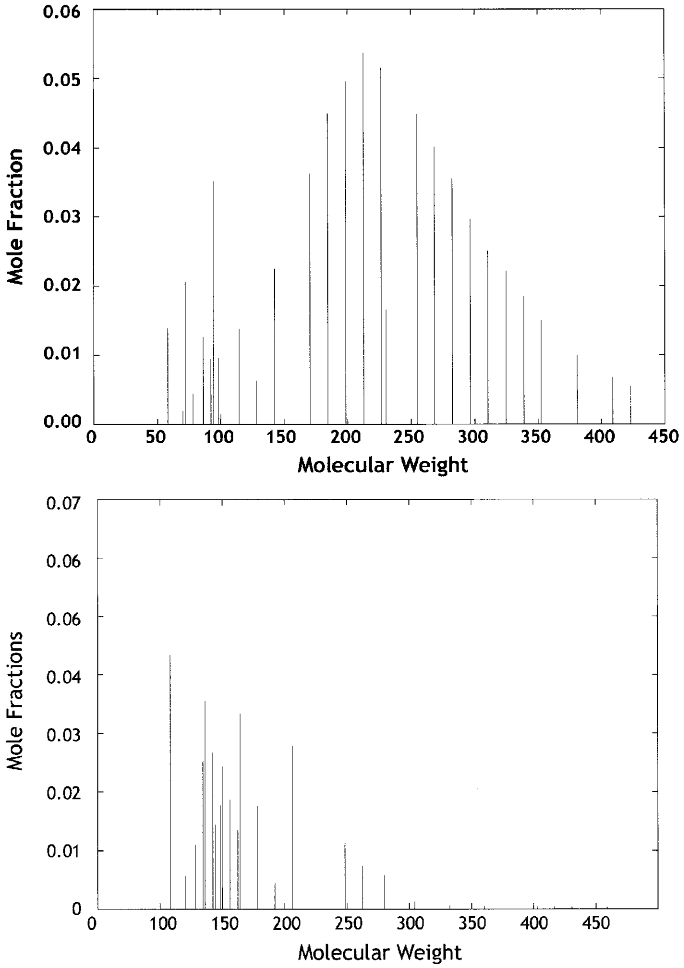


Fig. 7. Dead oil compositions, Vermillion area. The left figure shows a typical light oil (API gravity, 53°), while the right figure shows a condensate (original GOR, 15 K bbl/scf).

those that strongly influence phase behaviour, so estimating their loss is critical. Also, a great deal of oil never makes it through the GC column, including polar and asphaltene fractions. This is of lesser importance for vapour–liquid equilibrium, as these compounds have less of an effect on VLE (though a greater effect on aqueous–hydrocarbon equilibria). Hence, calculation of 'reincarnated' phase diagrams (using compositions where the volatile gasses are numerically reintroduced) is necessary. As an example of using GC for PVT calculations, several examples (one condensate, one light oil) are presented.

The examples presented here are dead oil compositions gleaned from GC data for several oils from the UNOCAL Vermillion area of offshore Louisiana, USA (Curiale & Bromley 1996*a*, *b*). The pseudo-GCs are presented in Figure 7. If we examine the composition of several of the fluids from that field, each shows the typical 'dead' oil profile of massive light-end loss (e.g. no methane, minimal ethane). In actuality, a great deal of methane and other light gases are associated with the original pre-separator fluids. For example, several PVT analyses from the area (unpublished UNOCAL data) shows a typical methane fraction of 0.55 for oils, 0.93 for a condensate. In order to use FPE models to analyse the fluid, an estimate of the live oil composition needs to be made. As an example of the profound phase behaviour effect of degassing, Figure 8 shows calculated phase envelopes for the oil from Figure 7a, where increasing amounts of methane have been numerically added. Each envelope (contiguous colour) is labelled with the mole percent methane present in the mixture. Note that the bottom phase boundary for each phase envelope lie very close to the 0 MPa pressure axis. By adding increasing amounts of methane (from 10 to 50% by mole), the maximum bubble point pressure moves from approximately 1.5 MPa to off the chart (over 12 MPa). A similar simulation can be performed for the condensate shown in Figure 7b, and is presented in Figure 9. Here, the phase diagram reaches much higher pressure (very close to reservoir pressures).

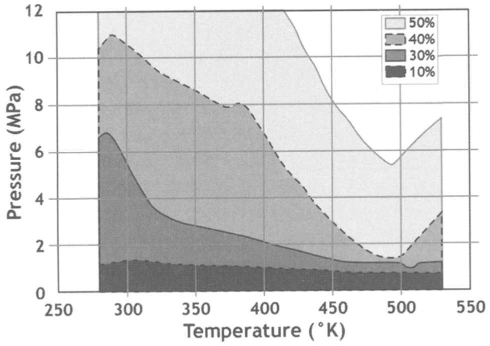


Fig. 8. Calculated phase envelopes for Figure 7 gas with added methane. Labelled are the fraction methane of the mixture whose phase envelope is shown. Note that a 'typical' reservoir has ($P = 30.3$ MPa, $T = 370.9$ K).

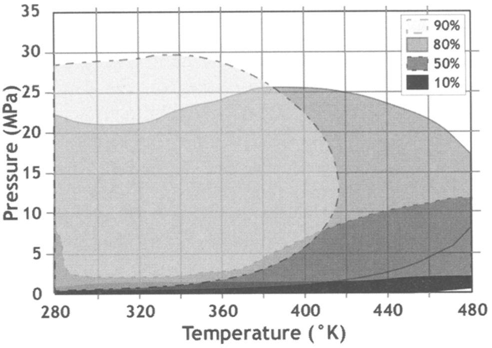


Fig. 9. Calculated phase envelopes for condensate with increasing amounts of added methane. Note that a 'typical' resevoir has ($P = 30.3$ MPa, $T = 370.9$ K).

Getting the code

The code for both the HCToolkit and the EOSInterface is available for download from http://ruby.wag.caltech.edu/EOSInterface. As the code is open source, all source code is published at this site, and users are encouraged to download and submit modifications to the code. An MS Windows installer program is available for the EOSInterface, making installation trivial. The EOSInterface program only works on the MS Windows platform. MS Office (MS Excel and MS Access) is required for standard installations. The HCToolkit works on most platforms and requires Perl (http://www.Perl.com) to be installed.

Conclusions

Presented here is a new open source software framework for doing phase equilibria calculations. The framework consists of two components: the HCToolkit (a set of Perl classes that implement EOS calculations and flash calculations) and the EOSInterface (an ActiveX/MS Excel front end to the HCToolkit). Several calculations are also shown that demonstrate the flexibility of the programs presented here. A (composition, pressure) phase diagram for the system (CO_2–cyclopentane) demonstrates the usefulness of the PSRK EOS (Holderbaum & Gmehling 1991). A phase boundary calculation for the methane–water system shows the ability of the framework for comparing different equations of state in challenging calculations. The applicability to hydrocarbons is shown by phase diagrams generated from GC analyses. The framework is flexible, extensible and available for download at http://ruby.wag.caltech.edu/

The authors gratefully acknowledge the support of the US DOE (contract number DE-AC26-99BC15217) and MSC industrial partners.

References

Aasberg-Petersen, K. & Stenby, E. 1991. Prediction of thermodynamic properties of oil and gas condensate mixtures. *Industrial & Engineering Chemistry Research*, **30**(1), 248–254.

Bunz, A. P., Dohrn, R. & Prausnitz, J. M. 1991. 3-Phase flash calculations for multicomponent systems. *Computers & Chemical Engineering*, **15**(1), 47–51.

Chung, T. H., Ajlan, M., Lee, L. L. & Starling, K. E. 1988. Generalized multiparameter correlation for nonpolar and polar fluid transport-properties. *Industrial & Engineering Chemistry Research*, **27**(4), 671–679.

Curiale, J. A. & Bromley, B. W. 1996a. Migration induced compositional changes in oils and condensates of a single field. *Organic Geochemistry*, **24**(12), 1097–1113.

Curiale, J. A. & Bromley, B. W. 1996b. Migration of petroleum into Vermillion 14 field, Gulf Coast, USA—Molecular evidence. *Organic Geochemistry*, **24**(5), 563–579.

di Primio, R. 2002. Unraveling secondary migration effects through the regional evaluation of PVT data: a case study from Quadrant 25, NOCS. *Organic Geochemistry*, **33**(6), 643–653.

Fredenslund, A., Jones, R. L. & Prausnitz, J. M. 1975. Group-contribution estimation of activity-coefficients in nonideal liquid-mixtures. *American Institute of Chemical Engineering Journal*, **21**(6), 1086–1099.

Gmehling, J., Li, J. D. & Schiller, M. 1993. A modified Unifac model. 2. Present parameter matrix and results for different thermodynamic properties. *Industrial & Engineering Chemistry Research*, **32**(1), 178–193.

Holderbaum, T. & Gmehling, J. 1991. PSRK—a group contribution equation of state based on Unifac. *Fluid Phase Equilibria*, **70**(2/3), 251–265.

Keshtkar, A., Jalali, F. & Moshfeghian, M. 1997. Evaluation of vapor–liquid equilibrium of CO_2 binary systems using UNIQUAC-based Huron-Vidal mixing rules. *Fluid Phase Equilibria*, **140**(1/2), 107–128.

Li, J. D., Vanderbeken, I., Ye, S. Y., Carrier, H. & Xans, P. 1997. Prediction of the solubility and gas-liquid equilibria for gas–water and light hydrocarbon–water systems at high temperatures and pressures with a group contribution equation of state. *Fluid Phase Equilibria*, **131**(1/2), 107–118.

Losh, S., Cathles, L. & Meulbroek, P. 2002. Gas washing of oil along a regional transect, offshore Louisiana. *In*: Meulbroek, P. & MacLeod, G. (eds) *Organic Geochemistry*. Pergamon, Oxford & New York, **33**, 655–663.

Mathias, P. M. 1983. A versatile phase-equilibrium equation of state. *Industrial & Engineering Chemistry Process Design and Development*, **22**(3), 385–391.

Meulbroek, P. 2002. Equations of state in exploration. *In*: Meulbroek, P. & MacLeod, G. (eds) *Organic Geochemistry*. Pergamon, Oxford & New York, **33**, 613–634.

Poling, B. E. 2000. *Properties of Gases and Liquids*. McGraw-Hill, New York.

Rijkers, M. P. W. & Heidemann, R. A. 1986. *Convergence behavior of single-stage flash calculations.* American Chemistry Society Symposium Series, **300**, 476–493.