

A Web Service Infrastructure and its Application for Distributed Chemical Equilibrium Computation

Subrata Bhattacharjee
San Diego State University
5500 Campanile Drive
San Diego, CA 92182-1323
+1 (619) 594-6080
subrata@thermo.sdsu.edu

Christopher P. Paolini
San Diego State University
5500 Campanile Drive
San Diego, CA 92182-1323
+1 (619) 594-7159
paolini@attila.sdsu.edu

Mark Patterson
San Diego State University
5500 Campanile Drive
San Diego, CA 92182-1323
+1 (858) 735-5769
markooo@gmail.com

ABSTRACT

W3C standardized Web Services are becoming an increasingly popular middleware technology used to facilitate the open exchange of data and perform distributed computation. In this paper we propose a modern alternative to commonly used software applications such as STANJAN and NASA CEA for performing chemical equilibrium analysis in a platform-independent manner in combustion, heat transfer, and fluid dynamics research. Our approach is based on the next generation style of computational software development that relies on loosely-coupled network accessible software components called Web Services. While several projects in existence use Web Services to wrap existing commercial and open-source tools to mine thermodynamic data, no Web Service infrastructure has yet been developed to provide the thermal science community with a collection of publicly accessible remote functions for performing complex computations involving reacting flows. This work represents the first effort to provide such an infrastructure where we have developed a remotely accessible software service that allows developers of thermodynamics and combustion software to perform complex, multiphase chemical equilibrium computation with relative ease. Coupled with the data service that we have already built, we show how the use of this service can be integrated into any numerical application and invoked within commonly used commercial applications such as Microsoft Excel™ and MATLAB® for use in computational work. A rich internet application (RIA) is presented in this work to demonstrate some of the features of these newly created Web Services.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright ©JOCSE, a supported publication of the Shodor Education Foundation Inc.

1. INTRODUCTION

Numerical determination of the equilibrium state mass fractions of gaseous and condensed matter is frequently needed in combustion simulations that model chemically reacting flows. The numerical method most often used to calculate an equilibrium distribution is based on minimizing a system's Gibbs free energy function with constraints. Several techniques for Gibbs energy minimization have appeared in the literature such as the tangent line/plane procedure suggested by Michelsen [1, 2], the maximum area method developed by Eubank et al. [3] and Elhassan et al. [4, 5], and the equal area method of Eubank and Hall [6], Shyu et al. [7, 8], and Hanif et al. [9, 10].

The total Gibbs energy G of a system composed of m species is given by

$$G = \sum_{j=1}^m \mu_j n_j \quad (1)$$

where μ_j is the chemical potential of the j^{th} species and a function of the system temperature T , pressure p , and number of moles of each component species, $n_{i \neq j}$, $\forall i$. From (1) and using the definition of the chemical potential for an ideal gas species j , we have

$$\mu_j = \mu_j^\circ + RT \ln \left(\frac{p_j}{p^\circ} \right) \quad (2)$$

where p° is the standard state pressure of 1 bar and p_j is the partial pressure of species j . The minimum stationary point of (1) will be the vector of species molar values \vec{n} where dG vanishes. Differentiating (1), we obtain

$$dG = \sum_{j=1}^m n_j d\mu_j + \sum_{j=1}^m \mu_j dn_j \quad (3)$$

From the isothermal, isobaric Gibbs-Duhem equation we know that

$$\sum_{j=1}^m n_j d\mu_j = 0 \quad (4)$$

and so we seek the unique vector \vec{n} such that

$$\sum_{j=1}^m \left[\frac{\mu_j^\circ}{RT} + \ln n_j - \ln n + \ln \left(\frac{p}{p^\circ} \right) \right] dn_j = 0 \quad (5)$$

where n_j is the number of moles of species j and

$n = \sum_{j=1}^m n_j$ is the total number of moles in the

equilibrium composition. Solving (5) amounts to solving a nonlinear constrained minimization problem. The traditional numerical method used for solving optimization problems [11] of this type is the method of Lagrange multipliers using an iterative Newton-Raphson technique for solving the resulting set of nonlinear equations. Both NASA CEA [12] and STANJAN [13] employ this method of element potentials. Both of these codes were written in Fortran and cannot be easily incorporated into other codes. Paolini and Bhattacharjee [14] modified the CEA algorithm and developed an object-oriented Java code for computing an equilibrium distribution by minimizing a system's Gibbs function. Front-end Java applets with easy-to-use graphical user interface were developed to run this equilibrium code through a browser.

In this work, we extend our object oriented code into a publicly accessible Web Service based module that can be called from most contemporary programming environments and integrated into many applications written in a variety of computer languages. This service can be invoked as a reusable third party software component by a thermal science researcher when developing custom applications. As a result, the researcher is freed from having to worry about implementing his or her own code for computing chemical equilibrium. When a desired equilibrium distribution is needed, the developer need only insert the requisite code to remotely discover and dynamically invoke the Web Service.

As a demonstration of the power of this newly created service, we present a few case studies: (a) Use of this service from third party applications such as Microsoft Excel or MATLAB, and (b) Use of this

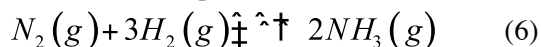
service to create a user friendly rich internet application (RIA) to simulate a steady-state combustion chamber.

Web Services: Web Services extend the paradigm of object-oriented programming to the network whereby a single software application is composed of loosely-coupled modules that execute on autonomous networked systems. Recent efforts by Dong et al. [15], Truong et al. [16], and Paolini and Bhattacharjee [17] have shown the strength of using Web Service technology in chemoinformatics applications to facilitate the organization and retrieval of chemical data. Furthermore, advances in collaborative cyberinfrastructure for developing predictive models for chemically reacting systems have been spearheaded by Frenklach et al. through the open-source Process Informatics Model (PrIme) project [18]. Additionally, the Cantera [19] package by Goodwin provides an open-source, object-oriented suite of software tools to aid in simulating problems in combustion and can be called to solve equilibrium problems from within FORTRAN, MATLAB, and Python scripts. Conceptually, Web Services can be thought of as a middleware technology that provides platform independent methods of facilitating machine-to-machine communication. This communication is accomplished through the use of Web application servers that deliver software services to client computers over a computer network. Web application servers are different than traditional Web servers in that an application server will manage and invoke user supplied code when requested to do so from a client system. Application servers publish a set of publicly available or exposed operations available to client applications using a standardized interface language called WSDL or *Web Services Description Language*. A WSDL specification is a structured XML document, often publicly accessible on the Internet, that client applications access remotely to determine the exact name, argument specification, and return type of a particular operation. The interface of a Web Service is separate from its actual implementation and the practice of separating interface from implementation is a core characteristic of all Web Services. While the interface of every Web Service is specified in standardized WSDL format, the implementation can be in any programming language. This separation allows Web Services to be platform-independent and provide transparent and modular access to pre-existing software services.

As a result of our work, there is now no need for a researcher to design his or her own equilibrium solver or locate a suitable third-party library. Our equilibrium Web Service can essentially be thought of as code that “plugs in” to existing software and takes advantage of distributed computational resources over the Internet. This style of software development based on orchestrating loosely-coupled and distributed software services is called a *Service Oriented Architecture* or SOA. Adopting an SOA approach to building combustion applications will have a sweeping impact on research and teaching in the thermal sciences as developers are able to construct new software tools that build upon an ever expanding collection of independent and modular Web Services.

2. EQUILIBRIUM COMPUTATIONS

We developed a chemical equilibrium Web Service that exposes an operation to calculate and return the equilibrium distribution of the products of an arbitrary reaction at a defined temperature and pressure. The input parameters are the reaction temperature in Kelvin, pressure in kilopascals, a comma and colon delimited list of reactants, and a comma delimited list of allowable products. The list of reactant species is specified using the format *moles:formula* where *formula* is a chemical formula of a reactant species using the Hill naming system and *moles* is the quantity of the respective species in the reactant mixture. To illustrate how our Web Service can be used, consider the standard problem of ammonia synthesis by means of the well known Haber process,



The Haber process is carried out at about 520°C and 500 atm in the presence of an iron-molybdenum catalyst. The catalyst increases the rate of the reaction but does not affect the reaction stoichiometry. At equilibrium, the mole fractions of nitrogen, hydrogen, and ammonia are approximately 17%, 50%, and 33%.

Our Web Service *solve* operation returns the unique distribution of product species that corresponds to a reaction at a fixed temperature *T* in Kelvin and pressure *p* in kilopascals. To invoke the solve operation, a SOAP message body is constructed by a client process that includes a definition of the four required input parameters shown in Figure 0. In Figure 0 we see four XML elements with names that correspond to the four parameters required to invoke the solve operation. Note the format used to specify reactants and products: reactants are given as a comma delimited list of

number-formula pairs, each delimited by a colon, and the products are given as a comma delimited list of just chemical formulas. The solution returned by the Web Service upon calling the solve operation is encoded in JavaScript Object Notation (JSON). Web Services can be invoked within most

```
double solve(double T, double P,
             string reactants,
             string products)
```

Figure 0. The chemical equilibrium Web Service exposes an operation named solve that returns the equilibrium state distribution of product species in JSON format.

```
<equ:solve
  xmlns:equ="http://cheqs/">
  <temperature>773.15</temperature>
  <pressure>50662.5</pressure>
  <reactants>1:N2,3:H2</reactants>
  <products>N2,H2,NH3</products>
</equ:solve>
```

Figure 0. The body of a SOAP message corresponding to the Haber process for ammonia synthesis given in reaction (6).

contemporary programming languages such as FORTRAN, C++, and Java. Programs written in these languages invoke the solve operation to compute the equilibrium distribution. The code uses the dynamic dispatch interface (DDI) provided by JAX-WS. Complete example code showing how to parse the SOAP response using the Simple API for XML (SAX) is available from the Tools section of <http://cheqs.sdsu.edu/>. As discussed in the previous section on Web Services, the DDI can be used in conjunction with a service registry to dynamically discover Web Services and invoke them on-the-fly. In the Java environment, Web Service discovery is frequently accomplished using the Java API for XML Registries (JAXR). Complete Java code that shows how software can dynamically discover our chemical equilibrium Web Service using a classification name and then call the service using the DDI technique is available from the Tools section of <http://cheqs.sdsu.edu/>.

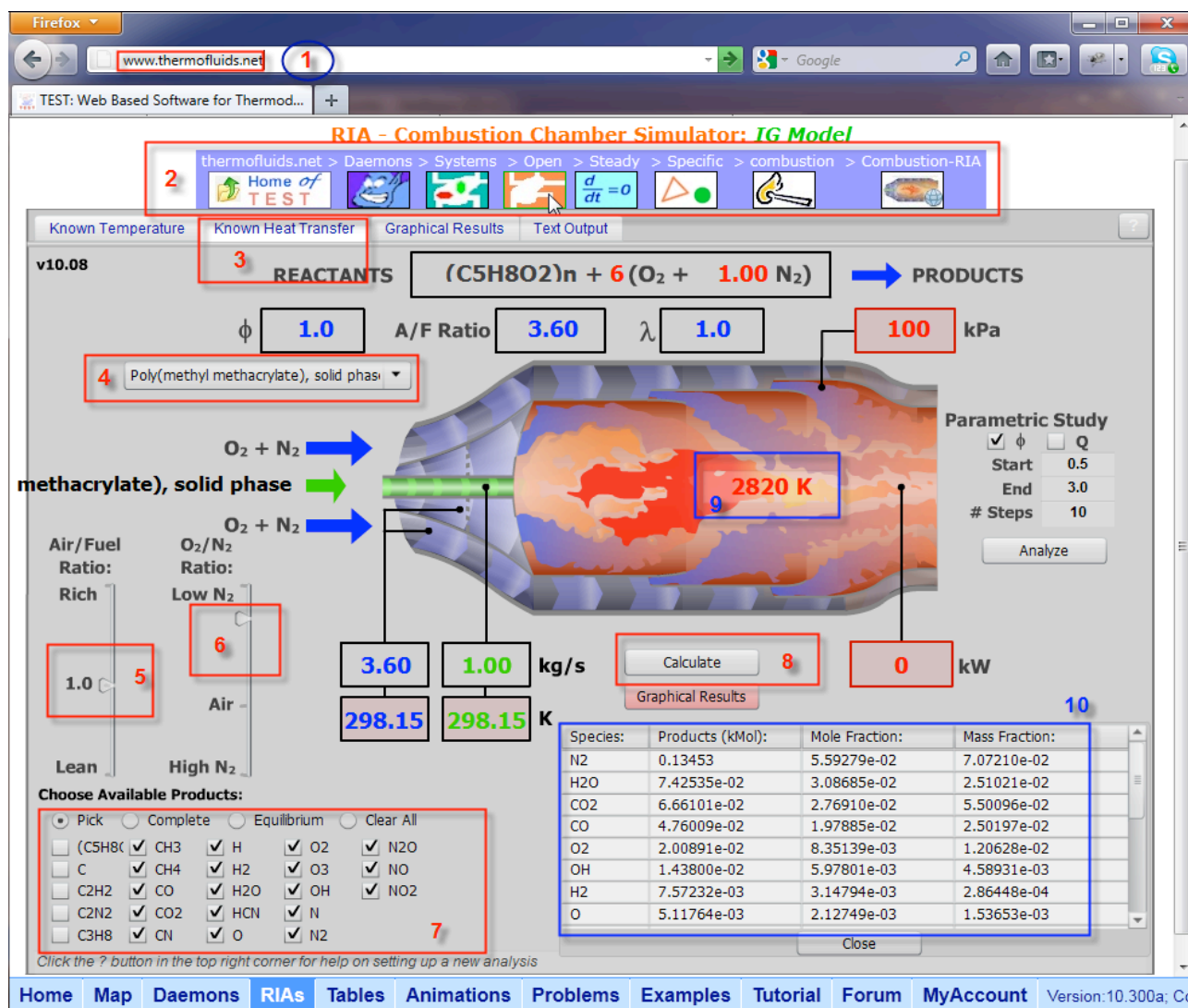


Figure 0. Screenshot of the combustion chamber simulator calculating the equilibrium flame temperature of PMMA combustion.

As previously stated, Web Services can be invoked from other languages besides Java such as FORTRAN, C, and C++. Because many present day software packages developed for combustion applications are written in FORTRAN, there is much interest in being able to invoke our chemical equilibrium Web Service from within a preexisting FORTRAN program. The two basic ways this can be accomplished is to use a FORTRAN compiler that provides native support for SOAP/XML Web Services or use a third party SOAP library. The former is provided by compilers such as the popular Lahey/Fujitsu Fortran v7.1 [20] while the later can be accomplished through the use of libraries like gSOAP [21].

The Combustion Chamber Simulator: An interactive rich internet application is created to

demonstrate the usefulness of the Web Service in making complex calculations simple and user friendly. The front-end of the simulator is written in AdobeTM flash, which calls Web Services in the background. The simulator is integrated with our thermodynamic Web portal TEST (The Expert System for Thermodynamics) www.thermofluids.net, is freely accessible, and used by more than 25,000 registered users.

To illustrate the use of the RIA, suppose we are interested in the combustion of PMMA (poly methyl methacrylate), specifically, the stoichiometric equilibrium flame temperature when condensed PMMA is burned with a 50-50 mixture (by volume) of oxygen and nitrogen at 1 atm. Using any browser (IE, Firefox, Chrome, Safari, etc.) equipped with the Flash

plug-in, launch the simulator (see Figure 0) by clicking the RIAs tab on the task bar and selecting the Combustion Chamber RIA (Label-1 in Figure 0). A Flash animation of a combustion chamber is loaded with its address displayed in a hierarchical manner relative to other resources (Label-2). For calculation of adiabatic temperature, we select the *Known Heat Transfer* tab (Label-3). Note that the pressure in the combustion chamber is set at 100 kPa by default and the heat transfer is set at 0, each of which can be edited as desired. From the fuel selector (Label-4), choose PMMA (solid phase). The reactants side of the reaction is displayed for the default reaction of the fuel burning with theoretical amount of air. Using the first slider bar (Label-4) you can change the equivalence ratio and see its effect on the reaction as it is made leaner or richer. Using the second slider bar (Label-5) you can change the makeup of the oxidizer from pure oxygen to very low level of oxygen in an oxygen-nitrogen mixture. The stoichiometric coefficient of nitrogen or the oxidizer (in red) can also be directly edited. For each settings, reaction parameters such as the equivalence ratio Φ , oxidizer/fuel ratio, percent theoretical oxidizer used λ , amount of oxidizer (in kg/s) entering the combustion chamber for a fuel mass flow rate of 1 kg/s,

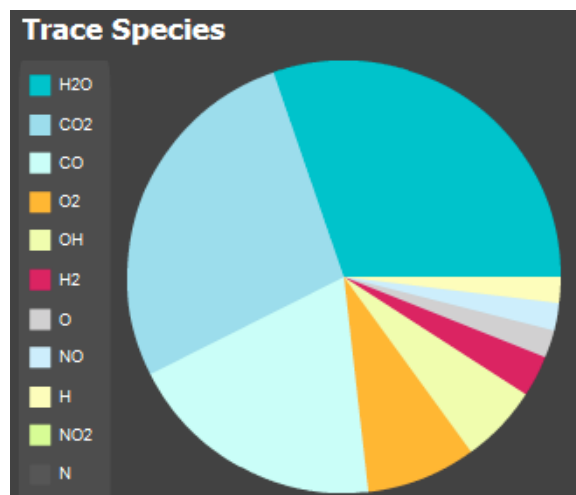


Figure 0. Graphical distribution of the trace species in the equilibrium composition of Figure 0.

etc., are dynamically updated.

Once the reactants mixture is set – in this case PMMA reacting with a theoretical amount of a 50-50 oxygen-nitrogen mixture, we need to specify the nature of the reaction. The RIA offers several options (see Label 7): complete combustion whereby all carbon and hydrogen atoms are fully oxidized, equilibrium combustion where most common combustion products are

automatically selected, and a customized selection of products species. If the *Complete* radio-button is selected, the adiabatic flame temperature is calculated as 3998 K. When the *Equilibrium* button is selected, the 23 most common combustion species are automatically selected. By clicking the *Pick* button, the list can be customized. With the selected species shown in Figure 0, when the *Calculate* button (Label 8) is pressed, a Web Service call is made to the CHEQS server. Based on the number of products species chosen, the calculations take anywhere between 5 s to 20 s. The calculated equilibrium flame temperature, 2820 K, and the composition (mass and mole fraction) of the products are displayed (Label 9 and 10). Clicking the *Graphical Results* button displays the composition in a graphical pie chart as shown in Figure 0.

One of the strengths of the RIA approach to computing is the simplicity with which one can explore fundamentals of chemical equilibrium. For example, the chamber pressure can be changed to 1 MPa and the equilibrium temperature can be recalculated as 3033 K without having to set up the problem again. Similarly, the effect of a specific reaction, say participation of N_2 can be evaluated by de-selecting all species that contains the N atom, except N_2 . The equilibrium temperature can be shown to remain relatively unchanged, changing slightly from 2820 K to 2831 K, showing the relatively weak effect of nitrogen participation on the energetics of

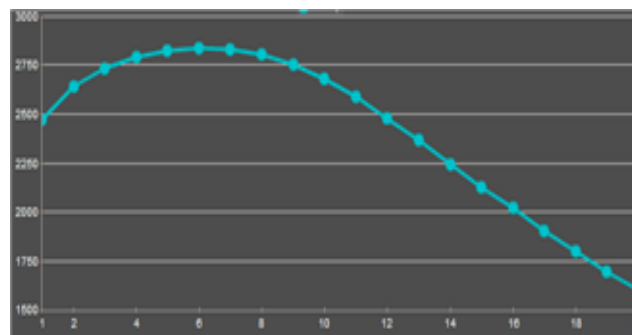


Figure 0. Equilibrium temperature against run number (equivalence ratio) is updated in real time as the web service calls returns.

PMMA combustion. In fact, it can be shown that including only the eight components - N_2 , H_2O , CO_2 , CO , O_2 , OH , H_2 , and O_2 - produces sufficient accuracy in the prediction of equilibrium temperature in this specific case.

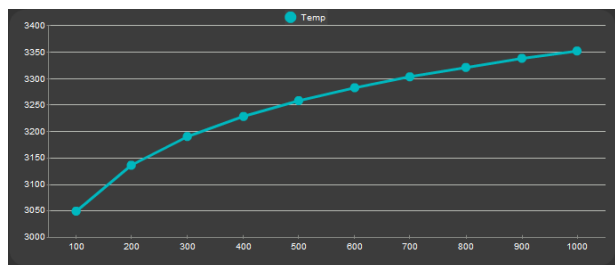


Figure 1. Effect of increasing pressure on the adiabatic flame temperature of methane combustion. Pressure is varied from 1 bar to 10 bar.

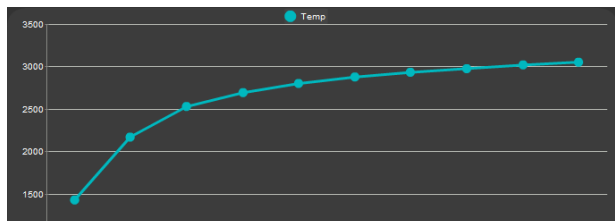


Figure 2. Parametric study of diluent (N₂) addition on adiabatic flame temperature in constant-pressure methane combustion. O₂ varies from 10% (by volume) in the oxidizer to 100%.

Arguably, the most valuable tool the RIA offers is its one-click approach to performing a parametric study. Simply select the range and number of steps in the parametric study block and click the *Analyze* button. The equilibrium solver Web Service is called repeatedly and the products composition table is updated continually. On the *Graphical Results* panel, the equilibrium temperature is plotted against the run number in real-time as the equivalence ratio is varied from 0.5 to 3 through 20 steps (see Figure 0). Detailed output for each set of calculations for a given equivalence ratio is produced in the *Text Output* panel. In addition to varying the equivalence ratio, the parametric study facility allows one to vary chamber pressure and oxygen mole fraction in the oxidizer. For example, one can evaluate the effect of diluent (N₂) addition on adiabatic flame temperature in the constant-pressure combustion of methane by choosing *Oxygen Mole Fraction* in the parameter drop-down menu of the *Known Heat Transfer* tab. By default, the volume of O₂ is configured to vary from 10% (mole fraction of 0.1) in the oxidizer to 100% (mole fraction of 1.0), although these limits are user adjustable by modifying the parameter *Start* and

End values. As one can see from Figure 2, diluent addition decreases adiabatic flame temperature, with a maximum temperature of 3049 K reached from an oxidizer consisting of pure oxygen. A similar analysis can be conducted by varying pressure. In Figure 1 we see how adiabatic flame temperature in methane combustion changes as pressure is varied from 1 bar to 10 bar. From the resulting graph, adiabatic flame temperature is shown to increase with pressure.

The *Known Temperature* panel can be used to calculate the heat transfer if the exit temperature is known. For example, the enthalpy of combustion for PMMA(s), Δh_C° , can be calculated by setting the exit temperature as 298 K and pressure as 100 kPa. For complete combustion, the heat transfer for a 1 kg/s fuel flow rate is calculated as -25,983 kW, that is, $\Delta h_C^\circ = 25.98 \text{ MJ}$.

Although illustrated for a specific fuel, the RIA can be used with any of the 24 fuels listed in the fuel selector. One of the advantages of the Web Service architecture is that the fuel list can be expanded on the server side as more data is available without having to recompile and redeploy the front-end application. The thermochemical data for fuels and combustion product species accessible through CHEQS Web Services is an aggregation of data provided by NASA [22], NIST [23], and Professor Alexander Burcat [24].

Emission Calculations through Web Services:

The combustion chamber simulator allows up to 23 pre-selected species. For a more general emission calculation, many more species are necessary. An applet based interface [6] has been created for this purpose and integrated with our thermodynamic Web portal: www.thermofluids.net at the location Daemons> Systems> Open> Steady> Specific> Combustion> ChemEqIWS. To illustrate the use of our Web Service to solve a rather complex equilibrium problem, consider the combustion of isooctane and air at $p = 50 \text{ bar}$ and $T = 3000 \text{ K}$ producing the 18 possible products for a given equivalence ratio ϕ and molar stoichiometric ratio s . The equilibrium reaction is

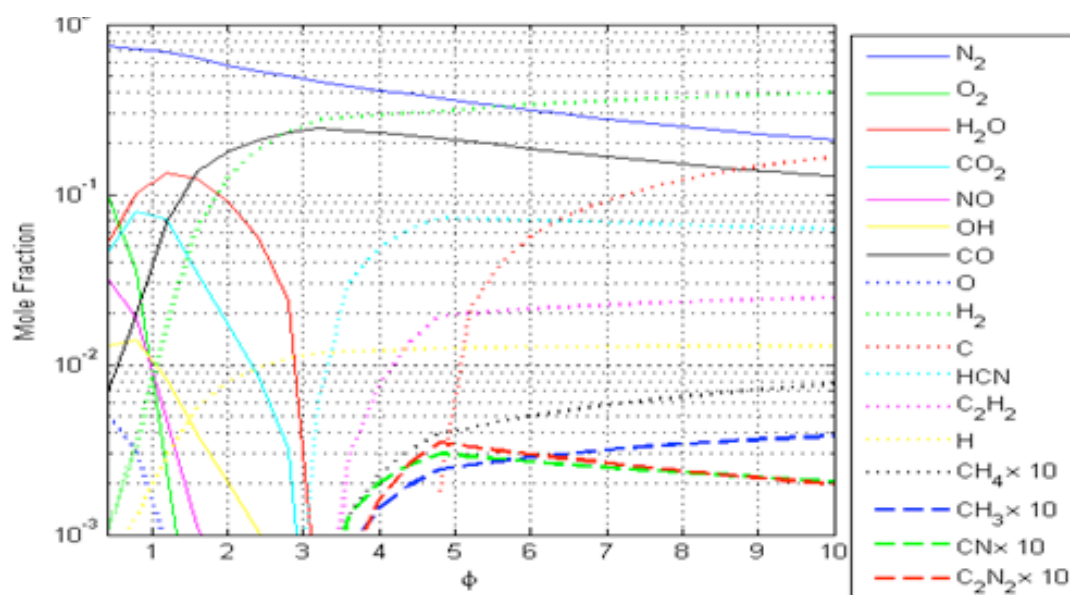
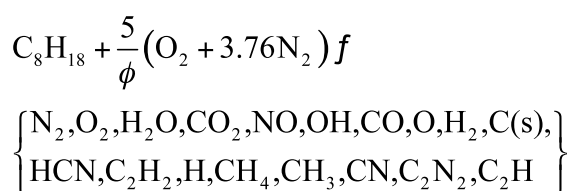


Figure 2. Equilibrium distribution of emissions from isooctane burning with air at 50 bar and 3000K for $0.4 \leq \phi \leq 10$.



The molar stoichiometric ratio s for octane is 12.5. A plot of equilibrium distributions obtained from invoking our Web Service for values $0.4 \leq \phi \leq 10$ is shown in Figure 2. The results presented compare very

Similar comparisons for other published data have produced consistent validation for the Web Service based equilibrium computation.

Integration with Third Party Applications: One of the primary advantages of standardized Web Services is platform independent accessibility. This means one should be able to invoke a Web Service as if it were a third party software component. To demonstrate this concept, we have developed a Microsoft Excel™ macro package and a MATLAB™ toolbox, both

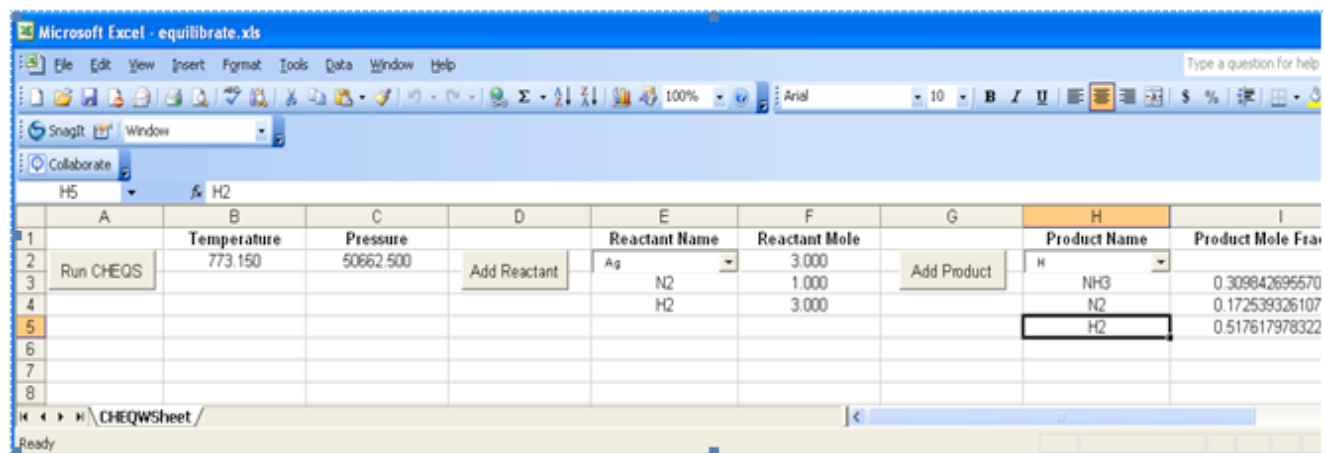


Figure 2. Solving the Haber process example using the Excel interface to our chemical equilibrium Web Service.

well with the results of NASA CEA calculations [25]. downloadable via the Tools section of

<http://cheqs.sdsu.edu/> that accesses our chemical equilibrium Web Service to compute the equilibrium distribution of an arbitrary multiphase reaction. Figure 2 shows a screenshot of running our Excel™ macro package to solve for the mole fraction of NH_3 in the Haber process for ammonia synthesis given by equation (6). Using the Excel spreadsheet, one can quickly specify additional reactants and products using the drop-down list and perform *what-if?* scenarios quite easily. For example, one might be interested to know if any monatomic species are present once equilibrium is reached during the Haber process. To find out, one simply adds N and H to the products list using the *Add Product* button and then recalculates the distribution by clicking the *Run CHEQS* button. As shown in Figure 4, one can see no appreciable amount of monatomic species are produced.

In addition to Microsoft Excel™, the numerical computing package MATLAB is often used by researchers to perform computational work. MATLAB possesses a powerful interpretive scripting engine that allows researchers to develop computational codes with relative ease, compared to development using a compiled language such as FORTRAN or C. Introduced in version 7, MATLAB contains built in functionality to generate SOAP messages and invoke Web Services. This capability allows researchers who use MATLAB for solving combustion problems to interface with our equilibrium Web Service.

To illustrate, Figure 3 shows an example using the MATLAB chemical equilibrium toolbox we developed to solve the ammonia synthesis problem from the MATLAB command line. Example MATLAB scripts are downloadable from our chemical equilibrium Web Service JSP cover page via URL <http://cheqs.sdsu.edu:8080/CHEQS/>.

3. CONCLUSIONS

The current trend in software development is to make use of distributed software components hosted on remote systems accessible through the Internet. Combustion applications can make use of these distributed components by calling Web Services in client code. In this paper we presented a modern alternative to software applications frequently used in combustion research, such as STANJAN and NASA CEA, for performing chemical equilibrium analysis. Adapting custom applications to libraries supplied with tools like STANJAN and CEA is possible but cumbersome and time-consuming. Instead, we have proposed a unique and different approach that

essentially “outsources” computational responsibility to network accessible Web Services. We developed one such Web Service for computing the equilibrium distribution through Gibbs free energy minimization and provided several examples of how this service can be invoked in Flash, Java, MATLAB, and Excel. Our Web Service vastly expands the scope of equilibrium computation and is especially advantageous in the simulation of chemically reacting flows. Educators from universities around the world have registered in our web portal to use this rich internet application in graduate level combustion classes.

Work is currently underway to demonstrate how Web Service based equilibrium calculations can be loosely coupled to existing CFD codes. Capabilities such as automatic load-balancing and failover protection being introduced in newer Web application servers like Glassfish will allow us to distribute the invocation of Web Services in parallel without the need for an additional high performance computing infrastructures. We will also investigate how a clustered configuration of Web application servers can facilitate parallel equilibrium computation. We will be extending our equilibrium Web Service to allow users to specify temperature and volume (T, V), entropy and pressure (S, P), or entropy and volume (S, V) and then calculate the equilibrium composition by minimizing the corresponding thermodynamic potential.

```
>> service =
    ChemicalEquilibriumService;
>> products = jsonread(
    solve(service, 520 + 273.15,
    500 * 101.325, '1:N2,3:H2',
    'N2,H2,NH3'))

products =

    H2: 0.54282415739298
    NH3: 0.27623445680936
    N2: 0.18094138579766
```

Figure 3. Invoking our chemical equilibrium Web Service from the MATLAB environment.

4. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 0753283

5. REFERENCES

- [1] Michelsen, M. L., Fluid Phase Equilib. 9, pp. 1–20, 1982.
- [2] Michelsen, M. L., Fluid Phase Equilib. 9, pp. 21–35, 1982.
- [3] Eubank, P. T., Elhassan, A. E., Barrufet, M. A., Whiting, W. B., Ind. Eng. Chem. Res., 31, pp. 942–949, 1992.
- [4] Elhassan, A. E., Lopez, A. A., Craven, R. J. B., J. Chem. Soc., Faraday Trans., 92, pp. 4419–4433, 1996.
- [5] Elhassan, A. E., Tsvetkov, S. G., Craven, R. J. B., Stateva, R. P., Wakeham, W. A., Ind. Eng. Chem. Res., 37, pp. 1483–1489, 1998.
- [6] Eubank, P. T., Hall, K. R., AIChE J., 41, pp. 924–927, 1995.
- [7] Shyu, G. S., Hanif, N. S. M., Alvarado, J. F. J., Hall, K. R., Eubank, P. T., Ind. Eng. Chem. Res., 34, pp. 4562–4570, 1995.
- [8] Shyu, G. S., Hanif, N. S. M., Hall, K. R., Eubank, P. T., Ind. Eng. Chem. Res., 35, pp. 4348–4353, 1996.
- [9] Hanif, N. S. M., Shyu, G. S., Hall, K. R., Eubank, P. T., Ind. Eng. Chem. Res., 35, pp. 2431–2437, 1996.
- [10] Hanif, N. S. M., Shyu, G. S., Hall, K. R., Eubank, Fluid Phase Equilib., 126, pp. 53–70, 1996.
- [11] Kramlich, K., General Gibbs Minimization as an Approach to Equilibrium, Course notes for ME 430 Advanced Energy Systems, University of Washington, August 2005.
- [12] Gordon, S. and McBride, B. J., Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications – I. Analysis, NASA Reference Publication 1311, October 1994.
- [13] Reynolds, W. C., The Element-Potential Method for Chemical Equilibrium Analysis: Implementation in the Interactive program STANJAN. Technical Report A-3391, Department of Mechanical Engineering, Stanford University, Palo Alto, CA, 1986.
- [14] Paolini, C. P. and Bhattacharjee, S., An Object-Oriented Online Tool for Solving Generalized Chemical Equilibrium Problems, Proceedings of the 2008 ASME International Mechanical Engineering Congress and Exposition, Boston, Massachusetts, October 31 – November 6, 2008.
- [15] Dong, X., Gilbert, K.E., Guha, R., Heiland, R., Kim, J., Pierce, M.E., Fox, G.C., and Wild, D.J. Web Service Infrastructure for Chemoinformatics. J. Chem. Inf. Model., 2007, 47(4), 1303 – 1307.
- [16] Truong, T.N.; Nayak, M.; Huynh, H.H.; Cook, T.; Mahajan, P.; Tran, L.T.; Bharath, J.; Jain, S.; Pham, H.B.; Boonyasirawat, C.; Nguyen, N.; Andersen, E., Kim, Y.; Choe, S.; Choi, J.; Cheatham, T.E.; and Facelli, J.C.; Computational Science and Engineering Online (CSE-Online): A Cyber-Infrastructure for Scientific Computing, J. Chem. Inf. Model., 2006, (46), 3, 971 - 984.
- [17] Paolini, C. P. and Bhattacharjee, S., A Web Service Infrastructure for Thermochemical Data, J. Chem. Inf. Model. 2008; 48(7); 1511-1523.
- [18] Frenklach, M.; Packard, A.; Seiler, P.; and Feeley, R. Collaborative Data Processing In Developing Predictive Models Of Complex Reaction Systems. Int. J. Chem. Kinetics, 2004, (36), 57 – 66.
- [19] Goodwin, D. G.; CANTERA: An Open-Source, Object-Oriented Software Suite for Combustion, NSF Workshop on Cyber-based Combustion Science, National Science Foundation, NSF Headquarters, Arlington, VA, April 19–20, 2006.
- [20] Lahey/Fujitsu Fortran Enterprise v7.1, Lahey Computer Systems, Inc., P.O. Box 6091, Incline Village, NV 89450 USA.
- [21] van Engelen, R. A. and Gallivan, K., The gSOAP Toolkit for Web Services and Peer-To-Peer Computing Networks, in the proceedings of the 2nd IEEE International Symposium on Cluster Computing and the Grid (CCGrid2002), pages 128-135, May 21-24, 2002, Berlin, Germany.
- [22] Gordon, S. and McBride, B. J., Thermodynamic Properties of Chemical Substances to 6000 K, NASA Report SP-3001, NASA Glenn Research Center, Cleveland, OH, 1963.
- [23] NIST Chemistry WebBook, NIST Standard Reference Database Number 69, National Institute of Standards and Technology. <http://webbook.nist.gov/chemistry/>, June 2005 Release.
- [24] Burcat, A., Third Millennium Thermodynamic Database for Combustion and Air-Pollution Use with updates from Active Thermochemical Tables, <ftp://ftp.technion.ac.il/pub/supported/aetdd/thermodynamic/s/>, <http://garfield.chem.elte.hu/Burcat/burcat.html>.
- [25] Ferguson, C. R. and Kirkpatrick, A. T., Internal Combustion Engines Applied Thermosciences, 2nd Edition, John Wiley & Sons, Inc., New York, 2001, p. 71.