Off-gas Adsorption Model Capabilities and Recommendations

Fuel Cycle Technology

Prepared for
U.S. Department of Energy
Material Recovery and Waste Form Development

Kevin Lyon, Amy Welty, Jack Law
Idaho National Laboratory
Austin Ladshaw, Sotira Yiacoumi
Georgia Institute of Technology
Costas Tsouris
Oak Ridge National Laboratory

March 1, 2016
INL/EXT-16-37975
FCRD-MRWFD-2016-000535

DISCLAIMER

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

SUMMARY

Off-gas treatment is required to reduce emissions from aqueous fuel recycling. Evaluating the products of innovative gas adsorption research requires increased computational simulation capability to more effectively transition from fundamental research to operational design.

Early modeling efforts produced the Off-Gas SeParation and REcoverY (OSPREY) model that, while efficient in terms of computation time, was of limited value for complex systems. However, the computational and programming lessons learned in development of the initial model were used to develop Discontinuous Galerkin OSPREY (DGOSPREY), a more effective model.

Initial comparisons between OSPREY and DGOSPREY show that while OSPREY does reasonably well to capture the initial breakthrough time, it displays far too much false numerical dispersion to accurately predict the actual shape of the breakthrough curves. DGOSPREY is a much better tool as it utilizes a more stable set of numerical methods. In addition, DGOSPREY has shown the capability to capture complex, multispecies adsorption behavior, while OSPREY currently only works for a single adsorbing species. This capability makes DGOSPREY ultimately a more practical tool for real world simulations involving many different gas species.

While DGOSPREY has initially performed very well, there is still need for improvement. The current state of DGOSPREY does not include any micro-scale adsorption kinetics and therefore assumes instantaneous adsorption. This is a major source of error in predicting breakthrough because the kinetics of the adsorption mechanism can be particularly slow. However, this deficiency can be remedied by building kinetic kernels into DGOSPREY. Another source of error in DGOSPREY stems from data gaps in single species, such as Kr and Xe, isotherms. Since isotherm data for each gas is currently available at a single temperature, the model is unable to predict adsorption at temperatures outside of the set of data currently available. Thus, in order to improve the predictive capabilities of the model, there is a need for more single-species adsorption isotherms at different temperatures, in addition to extending the model to include adsorption kinetics.

This report provides background information about the modeling process and recommendations for a path forward for further model improvement in terms of accuracy and user interface. User’s guides for OSPREY and DGOSPREY are also included.

CONTENTS

FIGURES

ACRONYMS

AgZ-PAN	Silver-modified mordenite in a polyacrylonitrile matrix
C	Carbon
CG	Continuous Galerkin
DG	Discontinuous Galerkin
DGOSPREY	Discontinuous Galerkin Off-gas SeParation and REcoverY
FEM	Finite Element Method
GIT	Georgia Institute of Technology
GPAST	Generalized Predictive Adsorbed Solution Theory
GSTA	Generalized Statistical Thermodynamic Adsorption
GUI	Graphical User Interface
H	Tritium
HPC	High Performance Computing
I	Iodine
INL	Idaho National Laboratory
Kr	Krypton
LDF	Linear Driving Force
MOOSE	Multiphysics Object-Oriented Simulation Environment
NEUP	Nuclear Energy University Program
ORNL	Oak Ridge National Laboratory
OSPREY	Off-gas SeParation and REcoverY
PDE	Partial Differential Equation
PJFNK	Preconditioned Jacobian-Free Newton-Krylov
Xe	Xenon

OFF-GAS ADSORPTION MODEL CAPABILITIES AND RECOMMENDATIONS

1. Introduction

Advanced simulation capabilities are essential for the design and prediction of complex unit operations such as those found in nuclear fuel recycling facilities. Fuel treatment head end processes generate fission product off-gas that requires efficient treatment technologies for the capture of krypton (^{85}Kr), iodine (^{129}I), tritium (^{3}H), carbon (^{14}C), and multiple xenon (Xe) isotopes. Innovation in gas adsorption technology requires increased computational capability in order to efficiently move from fundamental research to process design. Consequently, there has been a significant effort to develop modeling capabilities that solve fundamental transport equations for adsorption in a packed bed to facilitate development of separation processes for off-gas constituents.

Collaborative efforts among Georgia Institute of Technology (GIT), Idaho National Laboratory (INL), and Oak Ridge National Laboratory (ORNL) are ongoing to develop modeling capabilities for off-gas adsorption. The Off-gas SeParation and REcoverY (OSPREY) model was developed at INL to solve mass and energy balances for adsorption processes using the Multiphysics Object-Oriented Simulation Environment (MOOSE) framework (Rutledge 2013). While the transport equations solved in OSPREY are fundamentally sound, the Continuous Galerkin (CG) finite element method (FEM) utilized in OSPREY has been proven to show significant error in advection-dominate conservation problems (Pompelio et al 2014, Tavlarides et al 2015). Extensive modeling work has been performed at GIT to develop alternative numerical methods including Discontinuous Galerkin (DG) methods that can accommodate these types of physics problems (Tavlarides et al 2015). The Discontinuous Galerkin Off-gas SeParation and REcoverY (DGOSPREY) model is currently under development at GIT, essentially upgrading OSPREY from CG methods to DG methods to support advection-dominant processes and provide coupling with proven kinetic and equilibrium libraries developed at GIT.

The objective of the work presented herein is to provide an understanding of current off-gas model capabilities and recommendations for a path forward regarding future modeling efforts. Additionally, updated user guides for OSPREY and DGOSPREY are included for future use and development of gas adsorption modeling methods.

2. Background

The MOOSE framework is a finite element method (FEM) platform developed in C/C++ as well as some FORTRAN. MOOSE was built around the idea of developing the solution methodology first before considering any actual physics. By treating all problems anonymously in this fashion, MOOSE takes advantage of a set of generic linear and non-linear solvers to find solutions to systems of equations without having to know or understand where those problems came from. This anonymity gives MOOSE a large breadth of applications in science and engineering.

Physics within the MOOSE framework are applied on a need basis as kernels and additively placed into a residual. Each kernel comes from the weak form of a partial differential equation (PDE) that is used to describe a particular problem. For example, consider the following diffusion problem in Equation 1. The weak form of this problem is formulated by first moving all terms to the left hand side, multiplying the equation by a shape factor or test function (ψ) and subsequently applying the Gauss divergence theorem to lower the derivative order, as shown in Equation 2.

$$\frac{\partial u}{\partial t} = \nabla \bullet (\nabla u) \tag{1}$$

$$\underbrace{(\dot{u},\psi)}_{kernel} + \underbrace{(\nabla u,\nabla\psi)}_{kernel} - \underbrace{\langle \nabla u \bullet \vec{n},\psi \rangle}_{BC} = 0 \qquad (2)$$

The result of this transformation is a series of non-linear equations applied on each element in a mesh.

There are many advantages to using MOOSE for physics simulations, but perhaps the greatest advantage of MOOSE is how it handles the coupling of multiple physics objects and equations. Because of the modular design of MOOSE, the user codes each piece of physics separately when they are needed. These kernels feed into a non-linear residual. All non-linear residuals, for all physics and variables, are aggregated into a single residual vector. MOOSE then uses a Preconditioned Jacobian-Free Newton-Krylov (PJFNK) iterative method to solve all variables simultaneously using that aggregated residual vector as the basis. Therefore, all physics are inherently fully coupled and all variables are solved simultaneously.

A major advantage of the MOOSE framework is that the physics coded in MOOSE are valid on any spatial mesh object. Therefore, a kernel coded for diffusion physics is applicable in 1-D, 2-D, and 3-D geometries without having to change any source code. Additionally, MOOSE supports both regular and irregular mesh shapes, allowing the user to simulate physics on nearly any domain in space. This is the main reason that MOOSE is advantageous as the primary modeling platform for modeling off-gas adsorption.

By default, MOOSE employs CG methodology, whereby each element's solution is related to the next by a piecewise continuous series of basis functions. While this approach works very well for solid mechanics, thermal expansion, and groundwater hydrology, it performs very poorly for conservation law problems with non-linear or high-flux advection terms. Such problems typically arise in fluid dynamics for both compressible and incompressible flows. OSPREY's use of the default CG methodology has yielded some unsatisfactory results when encountering adsorption conservation problems (Tavlarides et al 2015). Therefore, simulating these types of problems in MOOSE requires special treatment of the physics within the MOOSE framework.

Advection-dominant conservation problems can be addressed in MOOSE using DG kernels. A detailed analysis of the numerical methods discussed here is available in the Nuclear Energy University Program (NEUP) Sorption Modeling and Verification for Off-Gas Treatment report (Tavlarides et al 2015). While MOOSE does support DG methods for advection-dominant or non-linear conservation law problems, support is currently limited. Additionally, documentation does not exist for creating and using DG kernels, consequently making development and troubleshooting in MOOSE very difficult. Regardless, modeling efforts at GIT have successfully implemented DG kernels for adsorption mass and energy balances into DGOSPREY. Additional work on DGOSPREY is necessary to couple the transport model with fundamental kinetic and equilibrium kernels. As additional support and capabilities expand within the MOOSE framework, issues relating to oscillatory behavior and error propagation can be improved upon to provide a robust adsorption model with increased reliability.

3. Model Evaluations

OSPREY models the adsorption of off-gas constituents for dispersed plug flow in a packed bed under non-isothermal and non-isobaric conditions. The model uses simple Linear Driving Force (LDF) kinetics that utilizes equilibrium data generated from Langmuir isotherms. Inputs to the model include gas composition, sorbent and column properties, equilibrium and kinetic data, and inlet conditions. The simulation outputs component concentrations along the column length as a function of time from which breakthrough data are obtained. It also outputs temperature and pressure drop along the column length as a function of time. The breakthrough data can be used to determine bed capacity, which in turn can be used to size columns.

Similarly, DGOSPREY is an improved model for the adsorption of off-gas constituents in a packed bed. The use of DG kernels in the MOOSE framework helps alleviate and eliminate the errors introduced by the numerical methods utilized by OSPREY. Errors in OSPREY, such as oscillations and false diffusion, are caused by the underlying methods (i.e., CG methods) utilized by the model. As the physics of the fixed-bed problem becomes more and more advectively dominated, those errors can become quite severe (Pompelio et al 2014). To demonstrate this, consider a simple conservation law problem shown in Equation 3. An analytical dimensionless solution exists for this PDE in 1D geometry (Kumar et al 2009), which will serve as a benchmark for the performance of the different numerical schemes utilized in MOOSE.

$$\frac{\partial u}{\partial t} + \nabla \bullet (vu) = \nabla \bullet (D\nabla u) \tag{3}$$

Figure 1 shows the simulation results for CG and DG methods compared to the analytical solution for Equation 3 (blue circles). The orders indicate the order of accuracy for the time integration technique used by MOOSE. Both results for the CG methods (red and green lines) show the two types of errors induced by these methods. At low order, CG methods will tend to develop false diffusion in order to stabilize the numerical result (red line), while at higher orders CG methods will often oscillate at the boundary of the wave (green line). Using DG methods to solve the same problem at the same level of accuracy will nearly eliminate all false diffusion and oscillations (orange line).

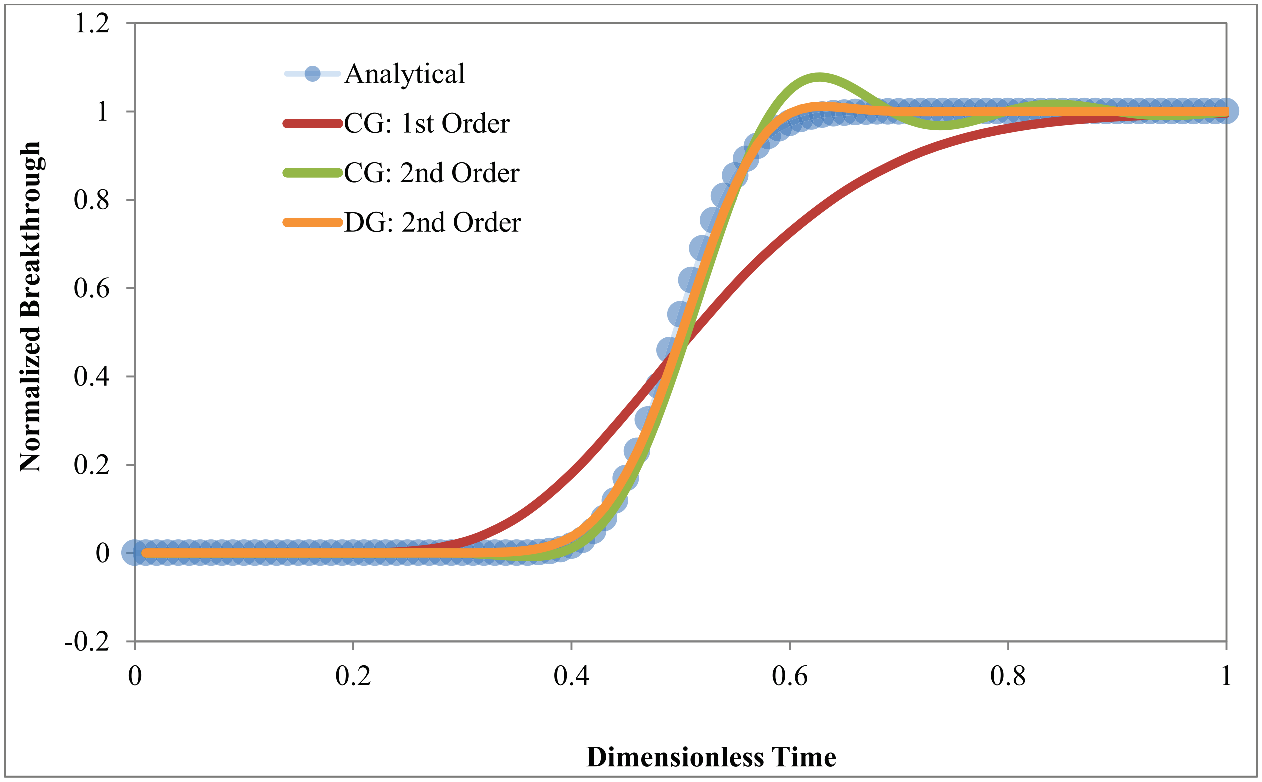


Figure 1: Comparison between Continuous Galerkin (CG) and Discontinuous Galerkin (DG) methods for solving an advectively dominant conservation law problem.

Note that these simulations are solving a relatively simple 1D conservation problem, and the error introduced with CG methods appears relatively small. The majority of the errors encountered in OSPREY are being caused by a propagation-of-error effect between the couplings of all variables. Small errors or small oscillations in one non-linear variable will propagate to other variables in the system, which then feedback on each other creating larger overall error. The continuous feedback of error causes the biggest problems in OSPREY, but the original error stems from the CG numerical method. The compounding effect of numerical method errors becomes more obvious when OSPREY output is compared to experimental data obtained from packed column gas adsorption experiments. Experimental data for xenon capture was obtained at INL studying the effect of channel velocities on sorbent capacity. These xenon adsorption tests were conducted at 295 K in a 101.6 cm long by 0.95 cm diameter fixed-bed column using an engineered form sorbent composed of silver-modified mordenite in a polyacrylonitrile matrix (AgZ-PAN). OSPREY was used to compare model output to experimentally determined breakthrough curves. The results, summarized in Figure 2, seem to indicate a fair amount of false numerical dispersion, as was seen from the example problem in Figure 1.

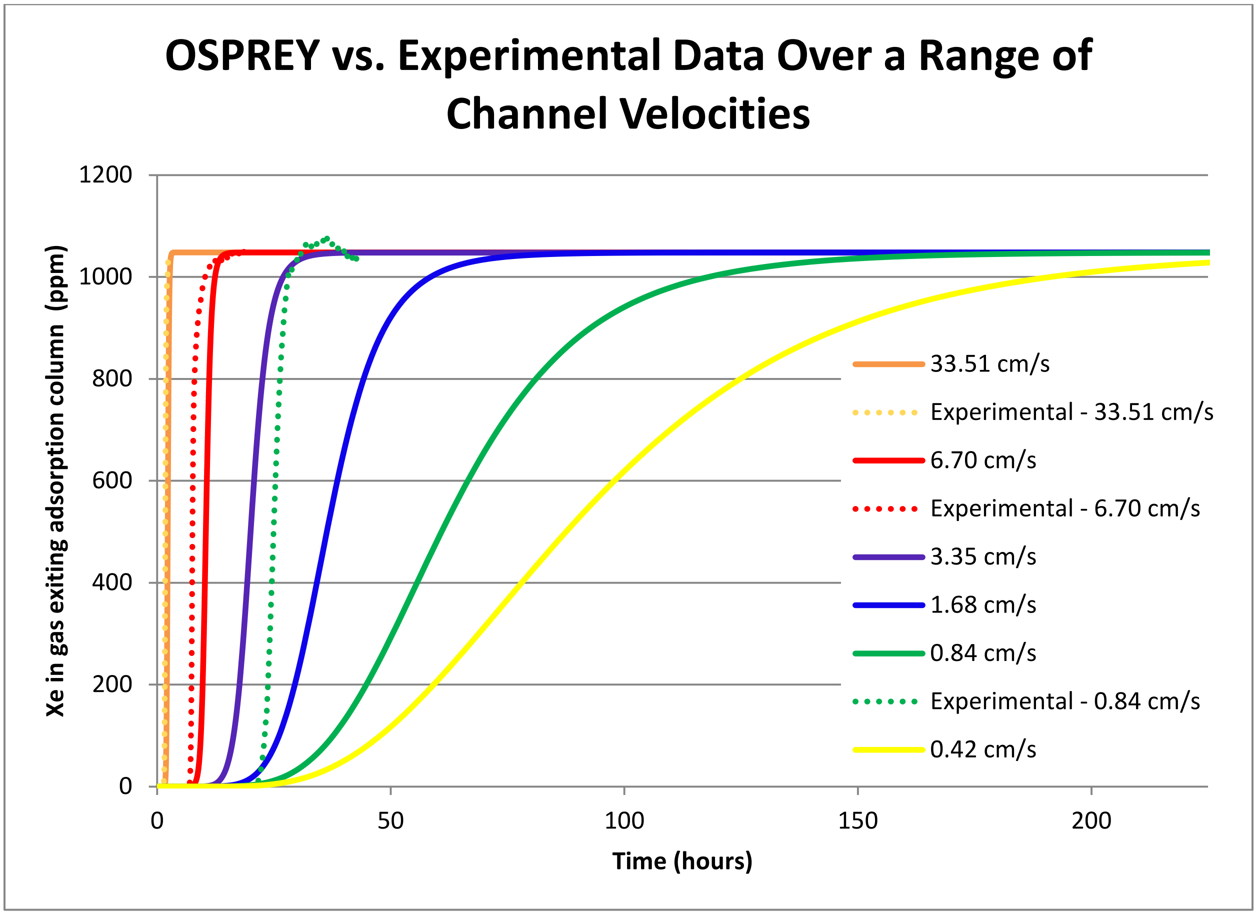


Figure 2: OSPREY output compared to experimental data at various channel velocities. All cases are for a 101.6 cm by 0.95 cm column tap-packed with engineered form AgZ-PAN sorbent.

Although OSPREY does predict initial breakthrough accurately, it is unable to accurately model the period between initial breakthrough and complete breakthrough, and therefore determines inaccurately high sorbent capacities. OSPREY provides exceptional agreement at high channel velocities, but false dispersion effects due to the CG methods cause significant deviation at lower channel velocities (compare the green line for OSPREY output and green dashes for experimental data). As a result, OPSREY is severely limited in its practical application for column design with the exception of accurate prediction of initial breakthrough.

As previously mentioned, DGOSPREY is currently under development and has not been fully coupled with the rigorous adsorption kinetic kernels developed at GIT. In its current state, DGOSPREY has upgraded PDE numerical schemes from CG to DG methods, and DGOSPREY has been coupled with mixed gas equilibria models developed at GIT (Ladshaw et al. 2015a, Ladshaw et al. 2015b). A graded approach is being utilized to couple DGOSPREY with kinetic uptake models for predicting breakthrough data. One of the main difficulties is coupling micro-scale kinetic and equilibria models with macro-scale transport equations to accurately determine column breakthrough data for column design. Because of the multi-scale nature of the problem, all of the governing equations cannot be fully coupled directly together, i.e., the smaller scale simulation cannot be directly coupled with the results of the larger scale simulation. Any attempt to fully couple multiscale physics often results in poor solution efficiencies or even complete failure to converge. However, by adopting a loose coupling scheme between the micro- and macro-scales of mass and energy transfer, adsorption problems can still be solved efficiently and accurately.

Under the MOOSE framework, loose coupling can be applied manually between different physical scales of the problem by making use of the Auxiliary kernel system and directing when the micro-scale problem is solved in conjunction with the macro-scale problem. The timing between each scale has to be dealt with in both the coded kernels and the simulation input files. One disadvantage to loose coupling is that some degree of accuracy is lost in the simulation results. However, significant gains in efficiency, such as faster convergence and overall runtime, are made possible when using loose coupling techniques. Multi-scale coupling implications were explored by comparing DGOSPREY simulation outputs to experimentally obtained data for water vapor adsorption on a MS3A type adsorbent. Experimental breakthrough data was obtained at ORNL at 298 K in a 12.7 cm long by 2.54 cm diameter fixed-bed column (Pompilio et al. 2014). The results are summarized in Figure 3.

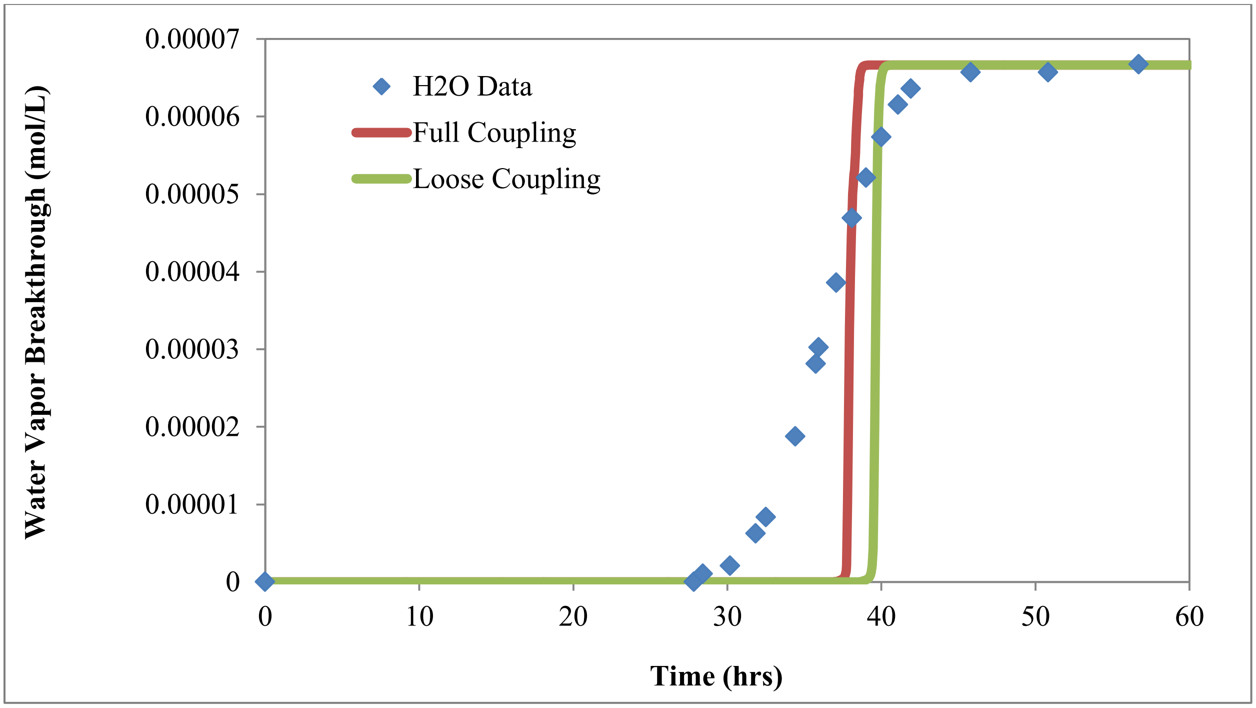


Figure 3: Comparison between DGOSPREY simulation results, with and without full coupling, and the water vapor breakthrough data from ORNL for adsorption at 298 K in MS3A (Pompilio et al. 2014). Extension of the model to include adsorption kinetics at the micro-scale will improve the predictive capability of the model.

In these modeling scenarios, the macro-scale mass and energy balances were coupled with the mixed gas isotherm models developed by GIT, but no kinetic model was used (this provides a more accurate representation of the effects caused by coupling techniques). As a result, the model assumes instantaneous adsorption (i.e., no pellet diffusion limitations). The simulated breakthrough curve is much sharper than what is observed in the data, which is likely the result of assuming instantaneous adsorption. This highlights the need to incorporate the micro-scale pellet diffusion kinetics into DGOPREY.

While the simulated breakthrough curves in Figure 3 do not match the data for water vapor breakthrough (due to the lack of kinetic model coupling), they do adequately represent the approximate breakthrough time. This is an improvement over the prior attempts to model this system in OSPREY. Additionally, only a small amount of accuracy is lost when using loose coupling (green line), as compared to full coupling (red line). Loose coupling is not necessarily required for this particular simulation since there is only one adsorbing species and kinetics is neglected. However, when multiple gas species are present and complex micro-scale kinetic kernels are incorporated, loose coupling will be required. Without loose coupling between multiple physical scales, simulation convergence cannot be guaranteed. Figure 3 demonstrates that the loose coupling scheme currently employed by DGOSPREY provides adequate results with minor losses in accuracy when compared to the same simulation with full coupling.

The same loose coupling scheme can also be applied in DGOSPREY to predict breakthrough curves for Kr and Xe. Currently, OSPREY is only capable of modeling one adsorbing gas species in a carrier gas. This provides limited applicability because multiple gas constituents are present in an actual off-gas process stream. DGOSPREY coupled with GIT's equilibria models is capable of handling mixed gas adsorption and predicting the breakthrough of multiple species simultaneously. Experimentally determined breakthrough data was obtained by INL at various temperatures using AgZ-PAN in a fixed column bed. First, consider adsorption of Kr and Xe in a 22.86 cm long by 1.73 cm diameter stainless steel column at 253 K. Figure 4 demonstrates that DGOSPREY model results (shown in red and blue lines) match the experimental breakthrough data fairly well and capture the general trends observed. The input file for this simulation is distributed with the DGOSPREY program. See Appendix B for information regarding gaining access to and running DGOSPREY.

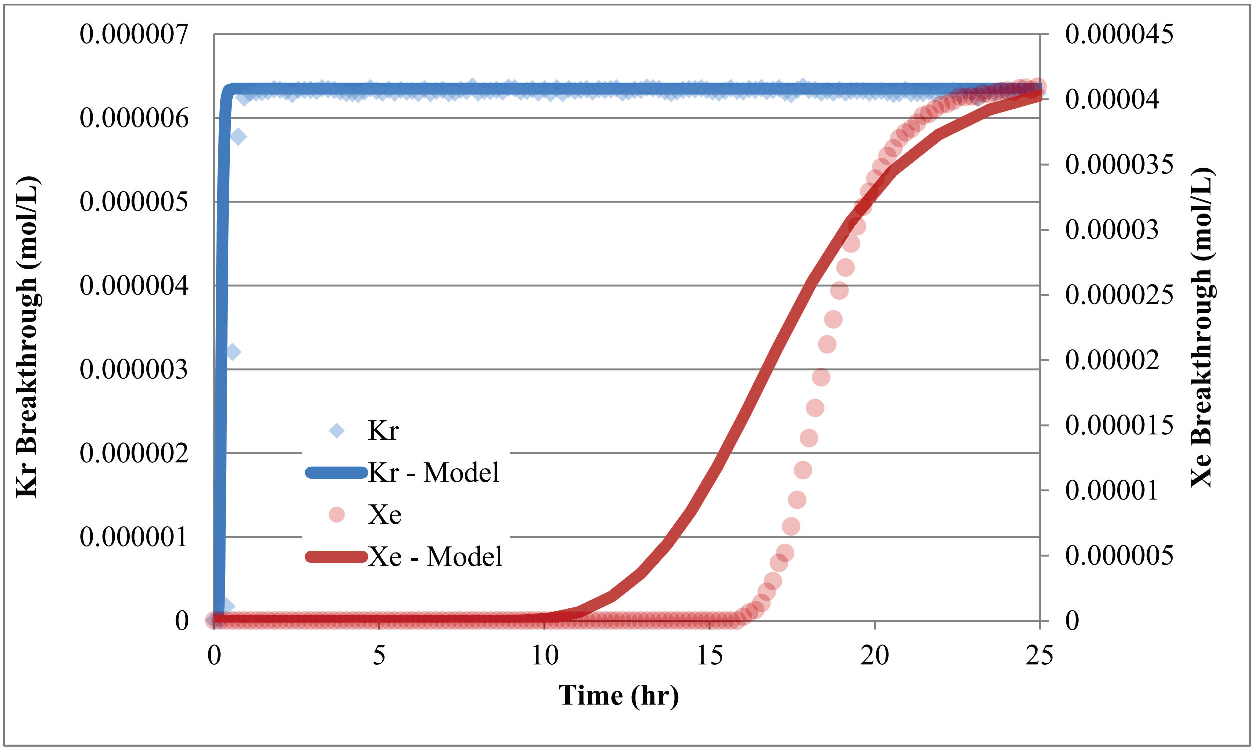


Figure 4: Comparison between DGOSPREY model predictions and the actual breakthrough data for Kr and Xe in a 22.86 cm by 1.73 cm stainless steel fixed-bed column containing AgZ-PAN at 253 K. Model predictions can be further improved by obtaining single-species isotherms at different temperatures for model parameter evaluation.

The initial breakthrough time predicted by DGOSPREY occurs earlier than what was observed experimentally, but provides a reasonable fit and a comparable sorbent capacity as determined by the overall breakthrough curve. Unfortunately, not all the Kr and Xe breakthrough curves predicted by DGOSPREY fit as well as those measured at 253 K. The breakthrough curves for Kr and Xe in the same size and type of column at 295 K are shown in Figure 5.

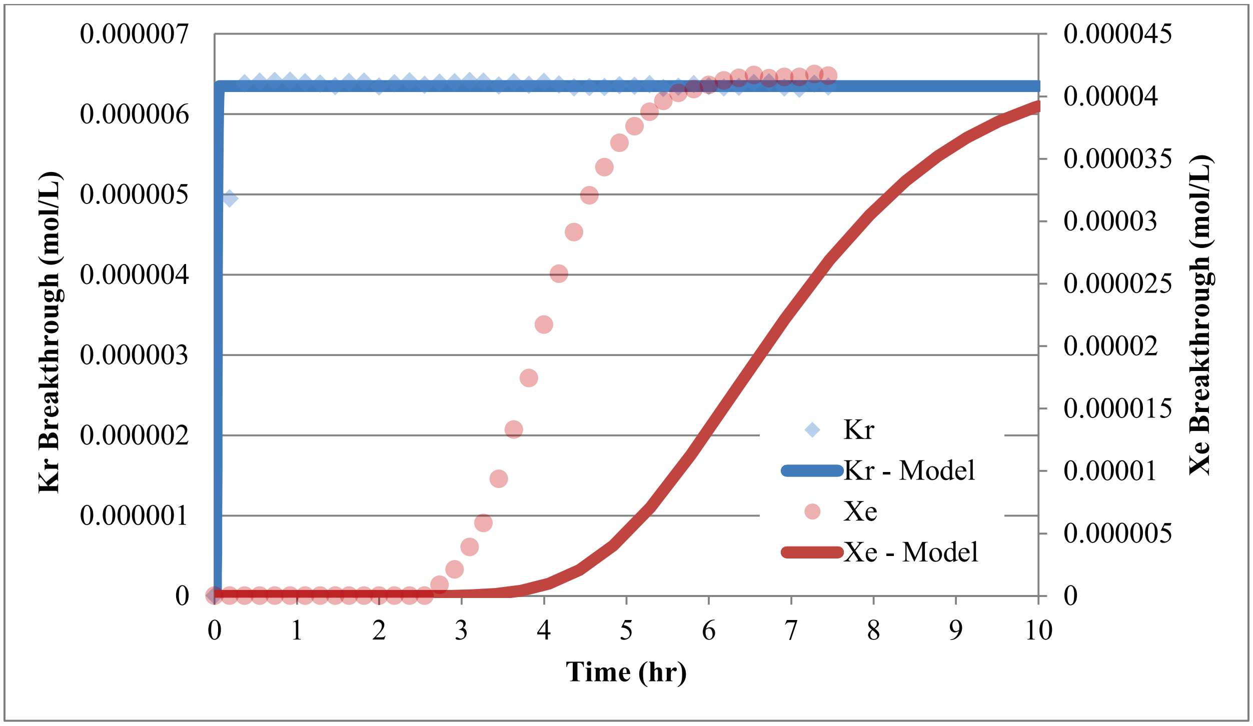


Figure 5: Comparison between DGOSPREY model predictions and the actual breakthrough data for Kr and Xe in a 22.86 cm by 1.73 cm stainless steel fixed-bed column containing AgZ-PAN at 295 K. Equilibrium data at different temperatures are needed to improve model predictions.

At 295 K, the overall trend in the breakthrough data for each species is predicted. The model predicts very fast Kr breakthrough with a delayed Xe breakthrough. However, the exact timing of breakthrough is not predicted well. The generic shape of the Xe breakthrough curve appears to be fairly accurate, but the actual initial breakthrough time occurs earlier than predicted by DGOSPREY. The errors observed in DGOSPREY simulations at 295 K are likely attributed to errors in the equilibria adsorption model predictions. Adsorption data were compared to the adsorption equilibria model coupled to DGOSPREY to determine the cause of the erroneous predicted breakthrough curves. Consequently, it was determined that the adsorption model predicts weaker Kr adsorption and stronger Xe adsorption than what the experimental data indicated. The adsorption data and predictions are shown in Figure 6. Those predictions are based on the single species GSTA isotherm parameters (Ladshaw et al. 2015a) coupled with the GPAST multispecies prediction model (Ladshaw et al. 201b). Errors in the isotherm parameters will occasionally propagate to larger errors in the multispecies model predictions. Eliminating these errors will require additional adsorption isotherm data for each individual species at multiple temperatures.

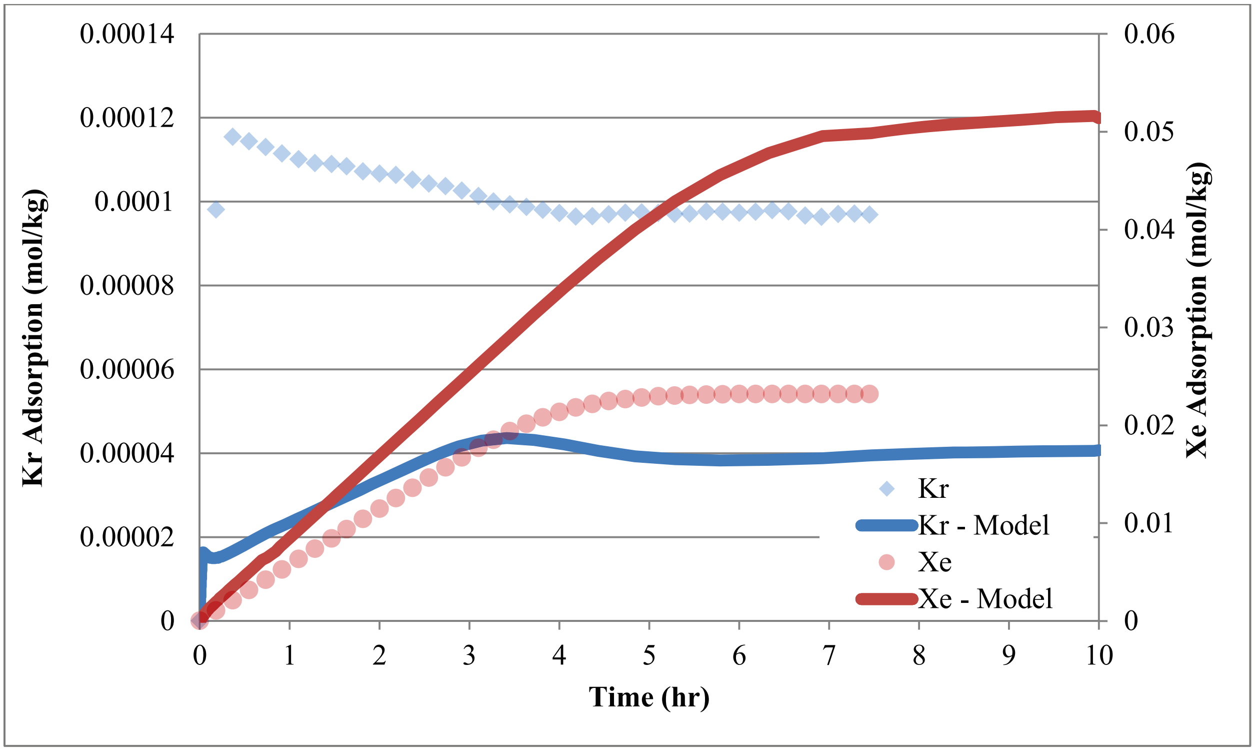


Figure 6: **Comparison between DGOSPREY model predictions and the actual adsorption data for Kr and Xe in a 22.86 cm by 1.73 cm stainless steel fixed-bed column at 295 K. Although the trends of breakthrough curves are captured, model predictions need improvement through addition of adsorption kinetics and use of isotherms at different temperatures.**

The average adsorption of each species in the column is under-estimated for Kr and over-estimated for Xe. Errors in predicted adsorption are caused mainly by errors in the adsorption isotherms and multi-component adsorption models. The multi-component adsorption model calculates adsorbed quantities using data collected from pure gas species isotherms, and therefore any error in isotherm data or mixed gas adsorption calculations propagates to DGOSPREY. This explains why the model shows earlier Kr breakthrough and later Xe breakthrough than what was observed in Figure 5. However, from a qualitative standpoint, the adsorption model behaves similarly to the adsorption data shown in Figure 6. There is a downward deflection shown in the adsorption of Kr both in the experimental data and the model predictions, which appears to be caused by the increased adsorption in Xe as breakthrough occurs. This is a promising result as it demonstrates that the detailed multi-species equilibria models that have been developed at GIT can be useful in describing and predicting the complex co-adsorption behavior of multiple species in the gas phase.

4. Conclusions and Recommendations

Off-gas adsorption models will require accurate solutions to complex, multi-scale mass and energy transport problems for the proper design of nuclear fuel recycling off-gas treatment systems. FEM schemes play an important role providing accurate solutions to the PDEs associated with advection-dominant transport problems, such as those frequently encountered during gas adsorption in packed bed columns. Consequently, modeling efforts have focused on using DG numerical schemes within the MOOSE platform to accurately predict breakthrough curves for adsorption processes.

In its current state, OSPREY's CG numerical scheme is not adequate for off-gas modeling and design and therefore there are no plans for future development of the OSPREY model. The OSPREY model has specific limited application for single component adsorption physics that are not dominated by advection. However, OSPREY does have valuable utility in the fact that it predicts initial breakthrough for a single species with a high degree of accuracy while consuming minimal computation time (model runtime is complete within seconds). Initial breakthrough is one of the most important parameters to consider for column design in nuclear fuel cycle applications- initial breakthrough predictions for gaseous radionuclides must be precise to prevent release to the atmosphere and potential exposure to the public.

The DG numerical schemes utilized in DGOSPREY provide accurate solutions for the physics problems encountered in off-gas adsorption. Although DGOSPREY has not been coupled to robust kinetic models, initial modeling results have shown promising predictions compared to experimentally obtained data. DGOSPREY will ultimately be a more robust and flexible model for off-gas adsorption processes covering a wide range of flows, pressures, temperatures, and gas compositions. Additionally, given only single-species isotherm data, DGOSPREY's equilibrium adsorption model accurately reflects interactions occurring among multiple adsorbing species, making it a valuable tool for predicting adsorption behavior of mixed gas streams.

User's guides for OSPREY and DGOSPREY are located in Appendix A and Appendix B of this report, respectively. An additional Doxygen-based pdf reference manual is available within the DGOSPREY repository that provides valuable information about the source code for individuals seeking to develop new code or debug and troubleshoot existing code. Instructions to locate and obtain this manual are located in Appendix B. The user guides provide instructions for setting up the MOOSE platform and accessing the OSPREY/DGOSPREY repositories. Additional instructions are provided for maintaining and updating the different repositories as the models are updated and expanded. Finally, each user's guide provides an example that utilizes actual column conditions and walks the user through the necessary steps to run the model and evaluate the model output. The examples provide actual model output that the user can reference to confirm proper operation and functionality of the model.

DGOSPREY is the recommended modeling platform for future adsorption work. There are several additional recommendations that will serve as guidance to develop a path forward for future off-gas modeling efforts. These recommendations provide a framework for modeling goals throughout the remainder of FY16.

First, the coupling and inclusion of kinetic models into DGOPSREY should be pursued. A graded approach for kinetic coupling will be used as this will provide easier transitions for model development and provide several kinetic kernels that can be used interchangeably for a wide variety of adsorption processes. Kinetic kernels will include simple kinetic models such as LDF, and eventually incorporate more complex surface and pore diffusion kinetic models. Providing several kinetic models allows the user greater flexibility to balance the required level of rigor to accurately represent an adsorption process while minimizing computational requirements. Unit tests for kinetic kernels should be developed and the new source code committed to the repository.

Second, future management of the DGOSPREY source code should be considered to streamline distribution and functionality. The DGOSPREY repository is hosted on GitHub and it is recommended that the repository remains on GitHub. OSPREY is hosted on INL's internal High Performance Computing (HPC) GitLab repository; past experience has shown that it is extremely difficult to provide access to off-site users, requiring a case-by-case evaluation to access the OSPREY code and in some instances users cannot obtain a local copy of the source code on their personal computers and must work remotely. Use of GitHub alleviates these internal access issues. However, as the number of people interested in using DGOSPREY grows, a distribution system will need to be organized to protect original source code and allow flexibility for developing new source code. This may be accomplished by creating multiple "forks" of the primary DGOSPREY GitHub repository. Each individual "fork" could then have multiple branches and would need to be updated with the primary repository periodically. Additionally, a legacy plan should be put in place to ensure continued maintenance of the program to keep pace with periodic changes in the MOOSE platform as well as any personnel changes that may occur in future. Collaboration with INL MOOSE personnel should be implemented to ensure long-term viability of the work.

Third, the user interface for MOOSE and DGOSPREY is currently geared toward users with some level of programming skills. The MOOSE platform does include a graphical user interface (GUI), PEACOCK, but this GUI has limited flexibility for running simulations and still requires a certain degree of knowledge about how the source code and input files are structured to operate it efficiently. Whether through the MOOSE framework or within DGOSPREY itself, a more user-friendly interface will greatly improve usability. Initial setup and running simulations with the model can be quite cumbersome for those not well versed in C++/MOOSE. Furthermore, any future development of new source code or model improvements requires an extensive knowledge of C++ and FEM, which may be difficult for scientists and engineers that have limited programming experience and limited time resources.

Finally, collaborative efforts should continue with the Off-Gas Sigma Team for model validation and experimental support. Experimental data provides a vital role for model validation by identifying discrepancies and deficiencies in the model that can be corrected to increase reliability and accuracy. Additionally, as simulation reliability and confidence increases, DGOSPREY can provide a support role for column design and test planning for future experimental work within the Sigma Team. These collaborative efforts ultimately support the overarching goal to develop a modeling tool that can be used for column design for efficient off-gas treatment technologies in nuclear fuel cycle applications.

5. References

Kumar, A., Jaiswal, D.K., and Kumar, N., “Analytical solutions of one-dimensional advection-diffusion equations with variable coefficients in a finite domain”, *J. Earth Syst. Sci.*, 118, 539-549, 2009.

Ladshaw, A.; Yiacoumi, S.; Tsouris, C.; DePaoli, D.W., “Generalized Gas-Solid Adsorption Modeling: Single-Component Equilibria”, *Fluid Phase Equilibria*, 388, 169-181, 2015a.

Ladshaw, A.; Yiacoumi, S.; Tsouris, C., “A generalized procedure for the prediction of multicomponent adsorption equilibria”, *AIChE J.*, 61, 8, 2600-2610, 2015b.

Pompilio, L.M., DePaoli, D.W., Spencer, B.B., “Evaluation of INL Supplied MOOSE/OSPREY Model: Modeling Water Adsorption on Type 3A Molecular Sieve”, FCRD-SWF-2014-000459, ORNL/LTR-2014/340, August, 2014.

Rutledge, V. J., “OSPREY Model”, FCRD-SWF-2013-000086, INL/EXT-13-28150, January 2013.

Tavlarides, L. L., et.al. “Sorption Modeling and Verification for Off-Gas Treatment”, U.S. Department of Energy Nuclear Energy University Program, February 15, 2015.

APPENDIX A: OSPREY USER'S GUIDE

NOTICE: This computer software was prepared by Battelle Energy Alliance, LLC, hereinafter the Contractor, under Contract No. DE-AC07-05ID14517 with the United States (U.S.) Department of Energy (DOE). NEITHER THE UNITED STATES GOVERNMENT, NOR DOE, NOR THE CONTRACTOR MAKE ANY WARRANTY, EXPRESSED OR IMPLIED, OR ASSUMES ANY LIABILITY OR RESPONSIBILIITY FOR THE USE, ACCURACY, COMPLETENESS, OR USEFULNESS OR ANY INFORMATION, APPARATUS, PRODUCT, OR PROCESS DISCOLSED, OR REPRESENTS THAT ITS USE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS.

EXPORT RESTRICTIONS. The provider of this computer software and its employees and its agents are subject to U.S. export control laws that prohibit or restrict (i) transactions with certain parties, and (ii) the type and level of technologies and services that may be exported. You agree to comply fully with all laws and regulations of the United States and other countries (Export Laws) to assure that neither this computer software, nor any direct products thereof are (1) exported, directly or indirectly, in violation of Export Laws, or (2) are used for any purpose prohibited by Export Laws, including without limitation, nuclear, chemical, or biological weapons proliferation.

None of this computer software or underlying information or technology may be downloaded or otherwise exported or re-exported (i) into (or to a national or resident of) Cuba, North Korea, Iran, Sudan, Syria or any other country to which the U.S. has embargoed goods; or (ii) to anyone on the U.S. Treasury Department's List or Specially Designated National or the U.S. Commerce Department's Denied Persons List, Unverified List, Entity List, Nonproliferation Sanctions or General Orders. By downloading or using this computer software, you are agreeing to the foregoing and you are representing and warranting that you are no located in, under the control of, or a national or resident of any such country or on any such list, and that you acknowledge you are responsible to obtain any necessary U.S. government authorization to ensure compliance with U.S. law.

1. Introduction

OSPREY models the adsorption of off-gas constituents for dispersed plug flow in a packed bed under non-isothermal and non-isobaric conditions. Inputs to the model include gas, sorbent and column properties, equilibrium and kinetic data, and inlet conditions. The simulation outputs component concentrations along the column length as a function of time from which breakthrough data are obtained. It also outputs temperature along the column length as a function of time and pressure drop along the column length. The breakthrough data can be used to determine bed capacity, which in turn can be used to size columns.

Currently, OSPREY is available on INL's HPC gitlab repository.

Note that OSPREY is only valid for laminar flow through a packed bed, where the packed bed Reynold's number $N_{Re,p} < 10$, according to the following equation:

$$N_{Re,p} = \rho_f \bar{V}_0 \frac{D_p}{(1-\varepsilon)\mu}$$

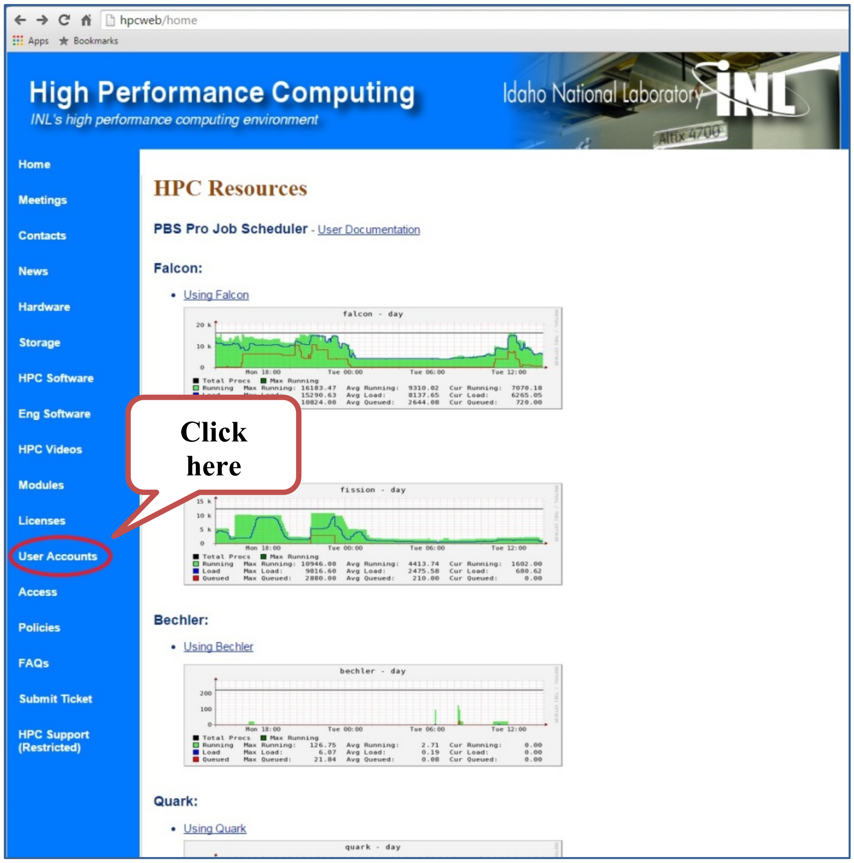
where:	$N_{Re,p}$	=	Packed bed Reynold's number
	P	=	Feed gas density at operating temperature
	$\bar{V}_0$	=	Channel velocity
	D_p	=	Particle equivalent spherical diameter
	ε	=	Bulk void fraction in packed bed
	μ	=	Dynamic viscosity of feed gas

$$D_p = \frac{6v_p}{s_p \Phi_s}$$

where:	v_p	=	particle volume
	s_p	=	particle surface area
	Φ_s	=	particle sphericity

2. Gaining access to OSPREY

1. Before accessing OSPREY users must first set up MOOSE and check out the MOOSE repository. INL employees must have a High Performance Computing (HPC) account for MOOSE to work properly. HPC home page (http://hpcweb/home). To obtain an account, click on "User Accounts" in the list on the left side of the page and follow the instructions (Figure 1). It may take up to two weeks for the account to be set up and accessible.


hpcweb/home
Apps
Bookmarks
High Performance Computing
INL's high performance computing environment
Idaho National Laboratory
INL
Altix 4700
Home
Meetings
Contacts
News
Hardware
Storage
HPC Software
Eng Software
HPC Videos
Modules
Licenses
User Accounts
Access
Policies
FAQs
Submit Ticket
HPC Support (Restricted)
Click here
falcon - day
20 k
10 k
0
Mon 18:00
Tue 00:00
Tue 06:00
Tue 12:00
■ Total Procs ■ Max Running
🟩 Running Max Running: 16183.47 Avg Running: 9310.02 Cur Running: 7070.18
Max Load: 15290.63 Avg Load: 8137.65 Cur Load: 6265.05
10824.00 Avg Queued: 2644.08 Cur Queued: 720.00
fission - day
15 k
10 k
5 k
0
Mon 18:00
Tue 00:00
Tue 06:00
Tue 12:00
■ Total Procs ■ Max Running
🟩 Running Max Running: 10946.00 Avg Running: 4413.74 Cur Running: 1602.00
🟦 Load Max Load: 9816.60 Avg Load: 2475.58 Cur Load: 680.62
🟥 Queued Max Queued: 2880.00 Avg Queued: 210.00 Cur Queued: 0.00
bechler - day
200
100
0
Mon 18:00
Tue 00:00
Tue 06:00
Tue 12:00
■ Total Procs ■ Max Running
🟩 Running Max Running: 126.75 Avg Running: 2.71 Cur Running: 0.00
🟦 Load Max Load: 6.07 Avg Load: 0.19 Cur Load: 0.00
🟥 Queued Max Queued: 21.84 Avg Queued: 0.08 Cur Queued: 0.00
quark - day

HPC Resources

PBS Pro Job Scheduler - User Documentation

Falcon:

- Using Falcon

Bechler:

- Using Bechler

Quark:

- Using Quark

Figure 1: HPC Home Page

a. Open bash profile (Terminal) and create a directory named "projects" and clone MOOSE
   i. Type the following commands in Terminal. If this does not work, continue on to steps ii, iii.
      1. mkdir ~/projects
      2. cd ~/projects
      3. git clone https://github.com/idaholab/moose.git
      4. cd ~/projects/moose
      5. git checkout master

        ii. The MOOSE home page is the gateway for MOOSE information, instructions, training and documentation (http://www.mooseframework.org/)
        iii. The MOOSE wiki has instructions for setting up MOOSE on a variety of operating sytems (http://www.mooseframework.org/getting-started/). Follow the applicable instructions.
    b. Compile libMesh
        i. cd ~/projects/moose
        ii. scripts/update_and_rebuild_libmesh.sh
    c. Test the clone
        i. cd ~/projects/moose/test
        ii. make –j8 [NOTE: 8 is the number of functional cores – change this number to reflect the available cores]
        iii. ./run_tests –j8 [NOTE: 8 is the number of functional cores – change this number to reflect the available cores]

2. Peacock must be added to the .bash_profile Path in order for it to be enabled for use with OSPREY.
    a. Open terminal
    b. Navigate to moose/gui directory. Should be: cd ~/projects/moose/gui
    c. Type "pwd" - The path to Peacock will be printed on the screen
    d. Type "export PATH=*pwd*:$PATH" where *pwd* is what was printed out in the previous step.
    e. The Peacock wiki contains additional information (http://mooseframework.com/wiki/peacock/)
3. Clone OSPREY.
    a. Go to "projects" directory (cd ~/projects)
    b. Type: git clone git@hpcgitlab.inl.gov:idaholab/osprey.git

3. Using OSPREY

1. Open bash profile
2. Open the trunk directory
    a. The trunk directory is the folder in which the MOOSE repository, checked out previously, is located
    b. Type: cd ~/<*address*>
        i. Should be: cd ~/projects
3. Update MOOSE repository
    a. If you are using a clone of the official MOOSE repository (idaholab/moose) from GitHub, enter your MOOSE directory and type: git pull --rebase origin master.
    b. If you are using a Fork of MOOSE from GitHub, then type: git pull --rebase upstream devel

   c. MOOSE is modified periodically. If the above instructions do not work, updated MOOSE instructions may be found on the mooseframework web page (http://www.mooseframework.org/blog/time-for-libmesh-update/)

4. Update libMesh
   a. After updating MOOSE, enter the MOOSE directory and update libMesh
      i. Type: scripts/update_and_rebuild_libmesh.sh [libMesh can take a long time to build, depending upon server load and your connection speed]
      ii. Some user information may be found in the libMesh wiki (http://www.mooseframework.org/wiki/libMesh/)
      iii. The libMesh home page provides access to full instructions, documentation and examples (http://libmesh.github.io/)
5. Build OSPREY
   a. OSPREY should be rebuilt prior to use each day as modifications may have been made to the repository
   b. Type: cd ../osprey to go to the osprey directory/folder
   c. Type: make cleanall
   d. Type: make –j8
      i. 8 is the number of functional cores – change this number to reflect the available cores.
      ii. The OSPREY application will be built [This will take several minutes, approximately the same amount of time as building libMesh]
      iii. Final line of the build should be: Linking Executable /Users/<address>/projects/osprey/osprey-opt…
   e. Run tests to ensure the new build is working properly
      i. Type: ./run_tests
      ii. Some tests may be skipped. The mass balance test (mass_bal_test.test) may fail. This is due to changes in the way MOOSE interacts and does not necessarily indicate an OSPREY failure.
      iii. If other tests fail, contact Kevin Lyon (kevin.lyon@inl.gov) or Amy Welty (amy.welty@inl.gov)
6. Open input file in PEACOCK
   c. Input files bring together all equations, parameters and information needed to run a particular simulation
   d. Go to OSPREY gas adsorption problems directory
      i. Type: cd ~/projects/osprey/problems/osprey/gas_adsorption
   e. List all the files in the directory to see available input files (.i files)
      i. Type: ls
   f. Open the desired input file to run in PEACOCK
      i. Type: peacock –i <*input_file.i*>

ii. As modifications are made to the model to include other components-specific simulations, new input files will be created and made available in this folder.

1. There are four main input files of interest

a. GasAdsorption.i is set up with parameters to simulate krypton sorption

b. GasAdsorption_single_comp.i

i. Equations and parameters for a single component in a carrier gas

ii. The same as GasAdsorption.i

c. GasAdsorption_multi_comp.i

i. Multiple component version of GasAdsorption_single_comp.i

ii. User specifies the number of components and whether or not each component adsorbs

d. GasAdsorption_skua.i

i. Set up for integration with fundamental level models to describe equilibrium and kinetics

ii. There are currently no plans to complete integration. Users may program the integrations if desired. DG-OSPREY is a better choice for more complex modelling problems.

iii. The command in step 6.d.i will open a Peacock window (Figure 2)

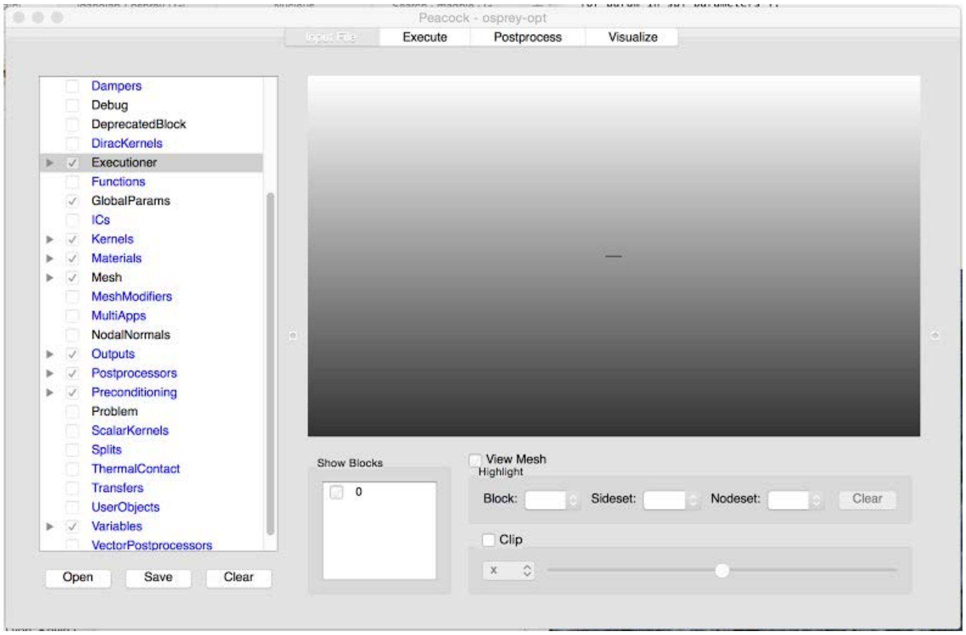
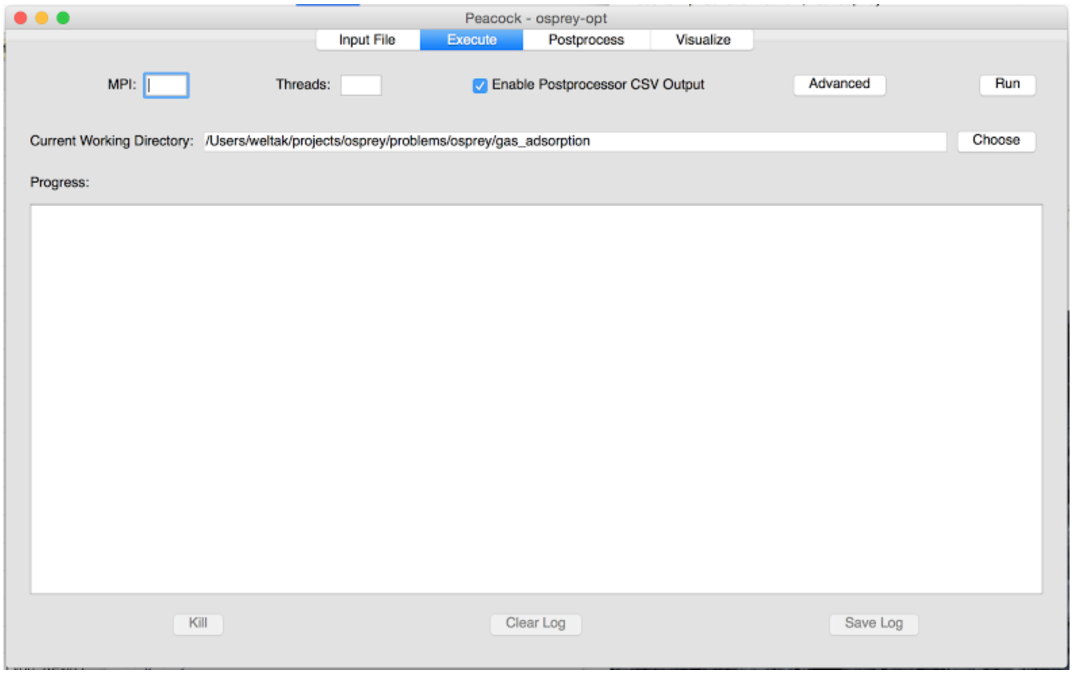


Figure 2: Peacock initial screen upon opening

4. Modify Parameter

   a. Double click on GlobalParams in the list on the left side of the PEACOCK window
   b. Go through the list of parameters and change Value to system specific parameters
      i. Comment section shows units used and has a brief description of parameter
   c. Click on "Apply" button
5. The input file for multiple components is set up for two components, one adsorbing and one not adsorbing.
   a. When this changes, variables, initial conditions, kernels and parameters for each species will need to be added.
   b. To do this, click on each corresponding label on the left side of the window and click add.
      i. Go through each box that comes up and input system information.
      ii. With multiple components, each species has a comp_index, starting with 0. This is how each component is identified within the model so any time asked for the comp_index, indicate with the number that corresponds to the species of interest.
6. Modify xmax in Mesh
   a. If the bed length is changed in GlobalParams, xmax must also be changed
      i. Xmax=bed length.
   b. Double click on Mesh in list on the left
   c. Scroll to xmax and input Value of bed length
   d. The number of spatial steps (in the x direction) can also be modified by changing nx value. A value of 50 works well and should not need to be changed unless the simulation fails to run.
   e. Click "Apply"
7. Modify initial conditions
   a. Click on arrow next to Variables in the list on the left side of the window
   b. Double click on a variable to set the initial condition for that particular variable
   c. Input initial_condition in Value column
   d. Click "Apply"
   e. Go through and set initial conditions for each variable
8. Modify simulation runtime
   a. Double click Executioner in the list on the left side of the window
   b. Locate end_time and change Value to desired simulation time
   c. Time steps can be changed by changing the value for dt
   d. Click "Apply"
9. Running the simulation

a. Click on Execute tab at the top of the window (Figure 3)



☑ Enable Postprocessor CSV Output

Current Working Directory: /Users/weltak/projects/osprey/problems/osprey/gas_adsorption

Figure 3: Peacock Execute screen

b. Ensure "Enable Postprocessor CSV Output" is checked

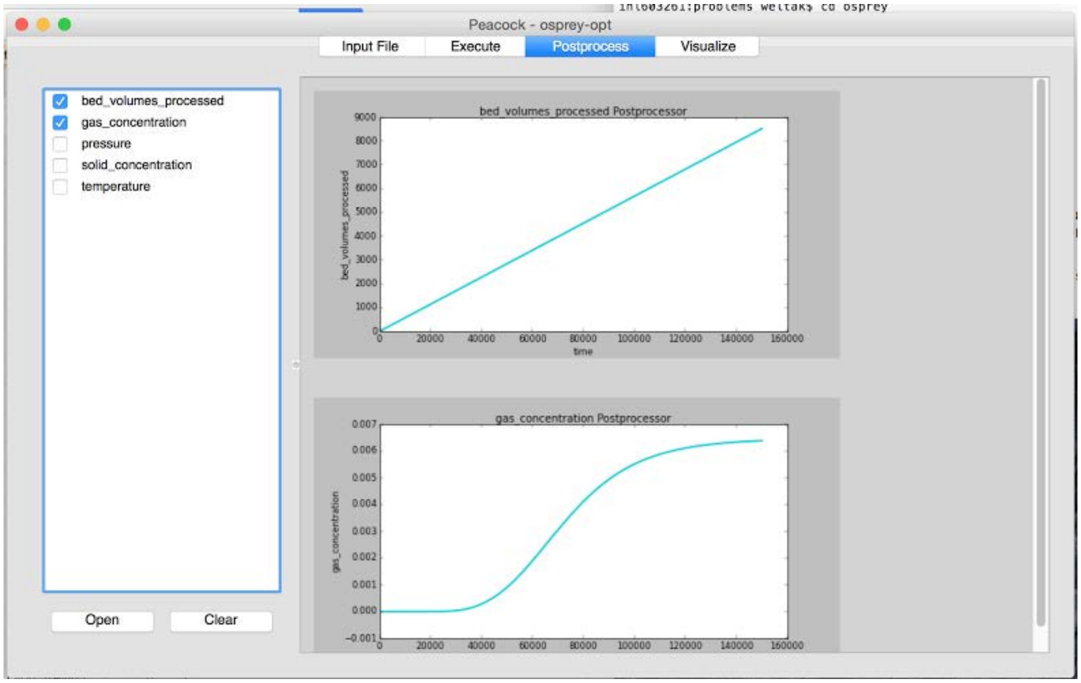
c. Click "Run" button

   i. Simulation prior to any parameter modification will run in a few seconds

d. To end the simulation before completion, click "Kill".

10. Viewing graphical results as output by simulation

a. Click on Postprocess tab at top of window (Figure 4)



Peacock - osprey-opt

☑ bed_volumes_processed
☑ gas_concentration
□ pressure
□ solid_concentration
□ temperature

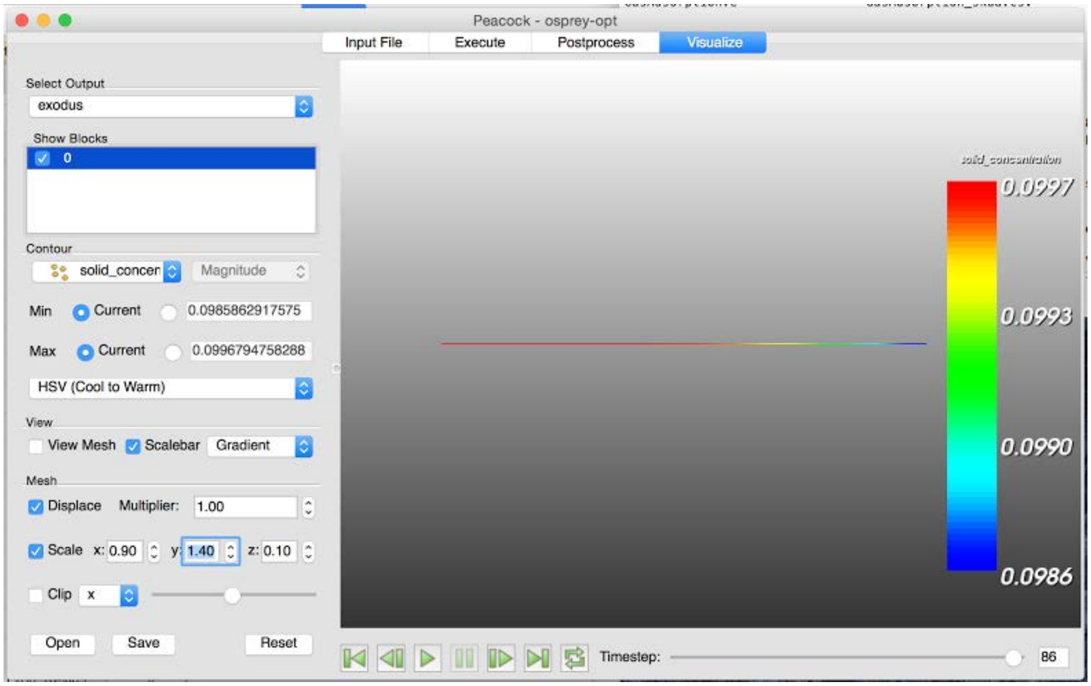
Figure 4: Peacock Postprocess screen

b. To view plots of variables, check the box next to the variable on the left. Plots will be generated in the gray area.

i. All plots are a function of time.

11. Viewing the gradient along the bed length with time.

a. Click on the visualize tab at the top of the window (Figure 5)



Peacock - osprey-opt

Select Output: exodus

Show Blocks: ☑ 0

Contour: solid_concen | Magnitude

Min ● Current ○ 0.0985862917575

Max ● Current ○ 0.0996794758288

HSV (Cool to Warm)

View: □ View Mesh ☑ Scalebar Gradient

Mesh: ☑ Displace Multiplier: 1.00

☑ Scale x: 0.90 y: 1.40 z: 0.10

□ Clip x

Timestep: 86

Figure 5: Peacock Visualize screen

   b. Choose the variable to visualize from the drop down list in the Contour box on the left side
   c. The min and max can also be changed in the Contour box
   d. Click on the Play button at the bottom of the window to watch the gradient move along with bed length with each time step.
   e. This can be done while the simulation is running.

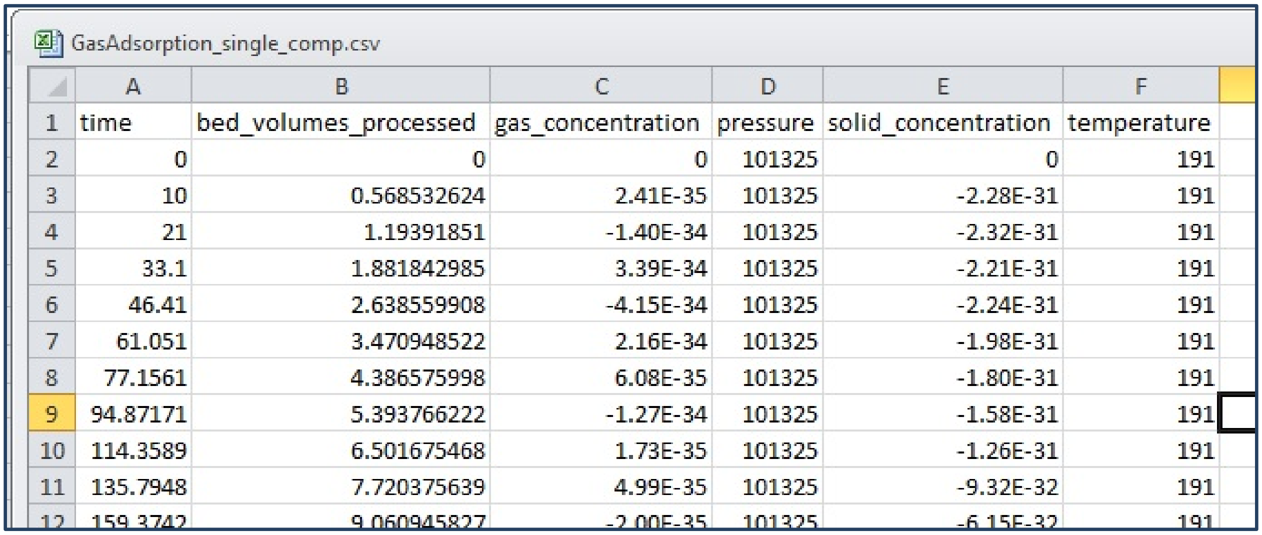
12. Viewing results in tabular form
    a. Results shown in graphical form in the postprocess tab are saved as a .csv file that can be opened in Excel. The .csv file will be located in the same folder where the input file is located.
13. Any modifications made to the input file by changing parameters and other values can be saved as a new input file.
    a. Go to the input file tab in Peacock.
    b. Change the file name if desired
    c. Click the "save" button
    d. Choose name and directory
       i. In order to open the new input file in Peacock at a later time, the file must be saved in the OSPREY folder in the problems directory.
    e. Click "save".
    f. The name of the output file will also need to be changed in order to keep outputs from different runs distinct.
       i. Double click on "Outputs"
       ii. Scroll down ton "file_base" and change the file name
       iii. Click "Apply" and the new file name will be saved and the user input box will close automatically.
14. A different input file can be opened directly from Peacock.
    a. Go to input file tab.
    b. Click the "open" button.
    c. Locate the input file to open and run in Peacock.
15. For assistance running OSPREY, contact kevin.lyon@inl.gov or amy.welty@inl.gov

4. Example:

New OSPREY users are encouraged to test their build using the input files that are included. Figures 6 through 8 of this manual were generated using "GasAdsorption_single_comp.i" which is included with every OSPREY build. The test case is for the following conditions:

Gas feed rate:	69.6 cm^3/min
Gas composition:	154.3 ppm krypton in helium balance
Adsorption temperature:	191 K
Sorbent:	HZ-PAN
Sorbent mass:	5.02 g
Column material:	Stainless steel tubing
Column length:	16.76 cm
Column OD:	1.245 cm
Column wall thickness:	0.0889 cm

In addition to the graphical output in Peacock, OSPREY will generate an Excel-readable output file with the same name as the input file, in this case, "GasAdsorption_single_comp.csv".
When the .csv file is opened in Excel, it should look like Figure 6.



GasAdsorption_single_comp.csv

	A	B	C	D	E	F
1	time	bed_volumes_processed	gas_concentration	pressure	solid_concentration	temperature
2	0	0	0	101325	0	191
3	10	0.568532624	2.41E-35	101325	-2.28E-31	191
4	21	1.19391851	-1.40E-34	101325	-2.32E-31	191
5	33.1	1.881842985	3.39E-34	101325	-2.21E-31	191
6	46.41	2.638559908	-4.15E-34	101325	-2.24E-31	191
7	61.051	3.470948522	2.16E-34	101325	-1.98E-31	191
8	77.1561	4.386575998	6.08E-35	101325	-1.80E-31	191
9	94.87171	5.393766222	-1.27E-34	101325	-1.58E-31	191
10	114.3589	6.501675468	1.73E-35	101325	-1.26E-31	191
11	135.7948	7.720375639	4.99E-35	101325	-9.32E-32	191
12	159.3742	9.060945827	-2.00E-35	101325	-6.15E-32	191

Figure 6: OSPREY .csv file opened in Excel

Notice there are no units associated with the columns of data. Units are as follows:

1. Time: seconds
2. Bed_volumes_processed: dimensionless
3. Gas_concentration: mol/m^3
4. Pressure: pascal
5. Solid_concentration: mol/kg
6. Temperature: K

Breakthrough curves are easily generated in Excel after a few simple unit conversions (Figure 7 and Figure 8):

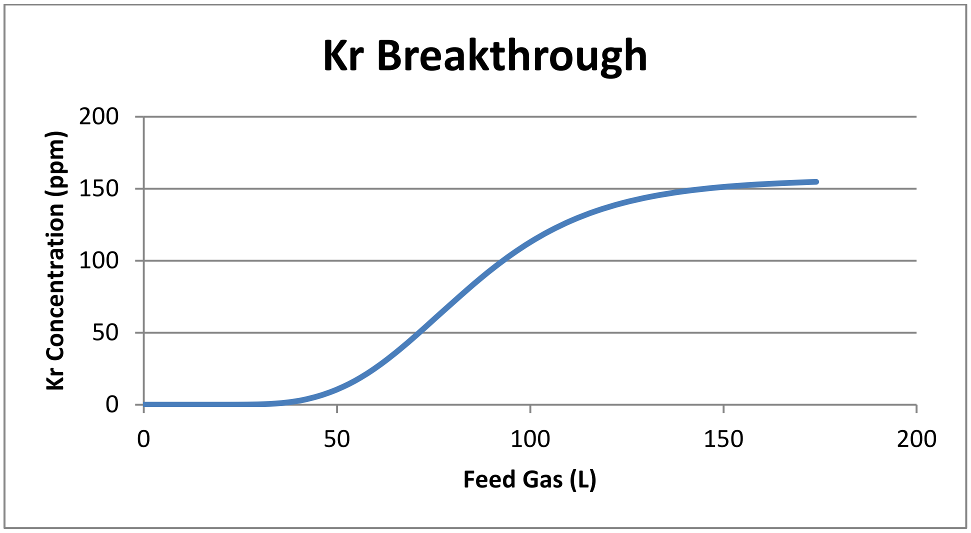


Figure 7: Kr breakthrough vs feed gas volume.

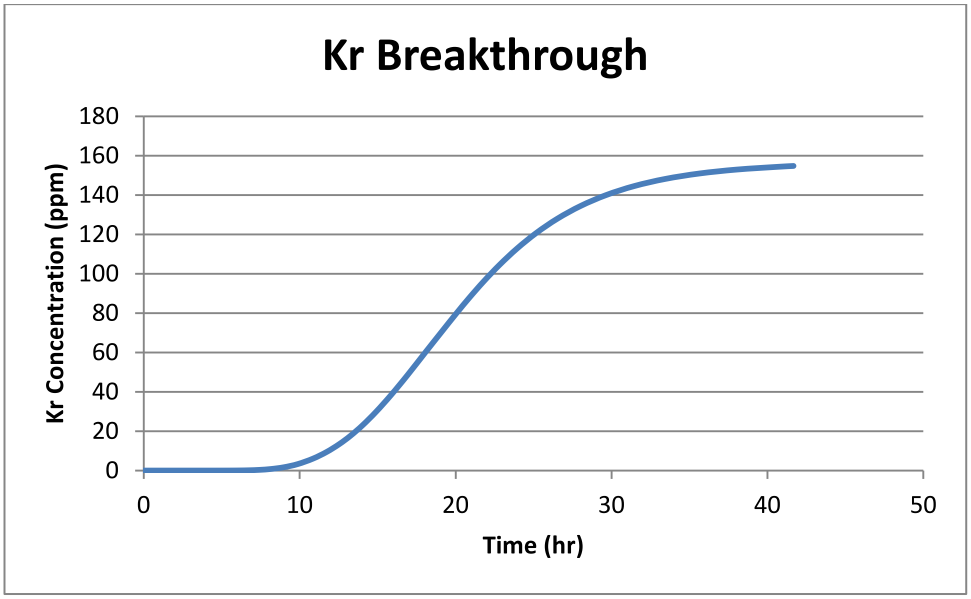


Figure 8: Kr breakthrough vs time.

APPENDIX B: DGOSPREY USER'S GUIDE

1. Introduction

Discontinuous Galerkin Off-gas Separation and REcoverY (DGOSPREY) models the adsorption of off-gas constituents for dispersed plug flow in a packed bed. Inputs to the model include gas, sorbent and column properties, equilibrium and kinetic data, and inlet conditions. The simulation outputs component concentrations along the column length as a function of time from which breakthrough data are obtained. It also outputs temperature along the column length as a function of time and pressure drop along the column length. The breakthrough data can be used to determine bed capacity, which in turn can be used to size columns.

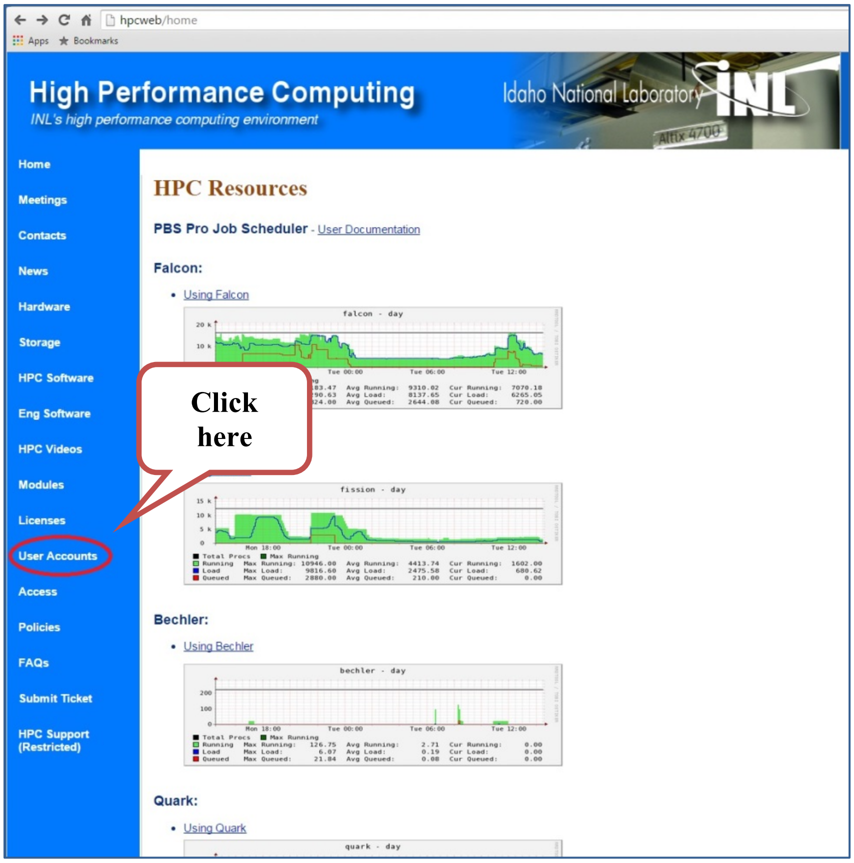
Currently, DGOSPREY is available on GitHub under a private git repository. Access to that repository can be given to any project participant who requests it. Once access to DGOSPREY is acquired, the user will be able to find a comprehensive Reference Manual (refman.pdf) under the /doc folder of the main project folder. That manual will contain all information regarding the structure of each class and/or object, as well as the functions and subroutines that each file contains. That manual is to be used as a guide tool for individuals wanting to develop DGOSPREY further, design new kernels, or understand how each action is performed and why. It is not a guide for how MOOSE is structured and does not contain information on how to run or debug the software.

DGOSPREY is run just like any MOOSE application. It requires a single input file that defines all variables, kernels, and processes to be performed. Check the MOOSE manual for how input files are built and structured. In addition to the Reference Manual, the DGOSPREY project comes bundled with several input files already structured and ready to run. These sample input files may be used as a reference for making input files for other simulations. The sample input files can be found in the main project folder under the sub-folder /input_files. This folder contains several other sub-folders that contain the sample input files.

The location of one of those sample input files is provided in section 2 of this appendix. This is the input file used to produce the breakthrough prediction in Figure 3 of the Model Evaluations section of this report. This file provides the breakdowns of each individual kernel being used in DGOSPREY to solve a multi-species adsorption problem in a 2D cylindrical column. Users can run this input file in DGOSPREY to confirm proper model operation and output. (Note: This simulation case was run using MPI on 8 cores. If this simulation is not run in parallel, results may vary slightly, but should be more or less the same.) For more information on input file structure, please consult the MOOSE Manual (http://mooseframework.org/static/media/uploads/docs/main.pdf).

2. Gaining access to DGOSPREY

1. Before accessing DGOSPREY, users must first set up MOOSE and check out the MOOSE repository. INL employees must have a High Performance Computing (HPC) account for MOOSE to work properly. HPC home page (http://hpcweb/home). To obtain an account, click on "User Accounts" in the list on the left side of the page and follow the instructions. It may take up to two weeks for the account to be set up and accessible.



Click here

Figure 1: HPC Home Page

a. Open bash profile (Terminal) and create a directory named "projects" and clone MOOSE

i. Type the following commands in Terminal. If this does not work, continue on to steps ii, iii.

1. mkdir ~/projects
2. cd ~/projects
3. git clone https://github.com/idaholab/moose.git
4. cd ~/projects/moose
5. git checkout master

ii. The MOOSE home page is the gateway for MOOSE information, instructions, training and documentation (http://www.mooseframework.org/)

iii. The MOOSE wiki has instructions for setting up MOOSE on a variety of operating sytems (http://www.mooseframework.org/getting-started/). Follow the applicable instructions.

b. Compile libMesh

i. cd ~/projects/moose

ii. scripts/update_and_rebuild_libmesh.sh

c. Test the clone

i. cd ~/projects/moose/test

ii. make –j8 [NOTE: 8 is the number of functional cores – change this number to reflect the available cores]

iii. ./run_tests –j8 [NOTE: 8 is the number of functional cores – change this number to reflect the available cores]

2. Peacock must be added to the .bash_profile Path in order for it to be enabled for use with DGOSPREY.

a. Open terminal

b. Navigate to moose/gui directory. Should be: cd ~/projects/moose/gui

c. Type "pwd" - The path to Peacock will be printed on the screen

d. Type "export PATH=*pwd*:$PATH" where *pwd* is what was printed out in the previous step.

e. The Peacock wiki contains additional information (http://mooseframework.com/wiki/peacock/)

3. Clone DGOSPREY

a. DGOSPREY resides in a private git repository http://github.com/aladshaw3/dgosprey.git

b. Go to "projects" directory (cd ~/projects)

c. Clone DGOSPREY – Type: git clone https://github.com/aladshaw3/dgosprey.git

d. Go to "dgosprey" directory – Type: cd dgosprey

e. Build DGOSPREY – Type: make –j8 [note: the 8 is the number of functional cores – change this number to reflect the available cores]

f. Test the build – Type: ./run_tests [note: all tests should pass]

3. Using DGOSPREY

7. Open bash profile (terminal)

8. Open the trunk directory

a. The trunk directory is the folder in which the MOOSE repository, checked out previously, is located

b. Type: cd ~/<*address*>

i. Should be: cd ~/projects

9. Update MOOSE repository
   a. If you are using a clone of the official MOOSE repository (idaholab/moose) from GitHub, enter your MOOSE directory and type: git pull --rebase origin master.
   b. If you are using a Fork of MOOSE from GitHub, then type: git pull --rebase upstream devel
   c. MOOSE is modified periodically. If the above instructions do not work, updated MOOSE instructions may be found on the mooseframework web page (http://www.mooseframework.org/blog/time-for-libmesh-update/)
10. Update libMesh
   a. After updating MOOSE, enter the MOOSE directory and update libMesh
      i. Type: scripts/update_and_rebuild_libmesh.sh [libMesh can take a long time to build, depending upon server load and your connection speed]
      ii. Some user information may be found in the libMesh wiki (http://www.mooseframework.org/wiki/libMesh/)
      iii. The libMesh home page provides access to full instructions, documentation and examples (http://libmesh.github.io/)
11. Update DGOSPREY
   a. DGOSPREY should be updated prior to use each day as modifications may have been made to the repository
   b. Go to your "dgosprey" directory – Type: cd ~/projects/dgosprey
   c. Switch to the master branch – Type: git checkout master
   d. Pull down changes – Type: git pull
      i. From INL intranet, "git pull" may not work. It may be necessary to type: make cleanall
   e. Rebuild if necessary – Type: make –j8
      i. 8 is the number of processors being used. Adjust this number to reflect the number of processors available.
      ii. The DGOSPREY application will be built (This may take a long time)
      iii. Final line of the build should be: Linking Executable /Users/<address>/projects/dgosprey/dgosprey-opt…
   f. Run tests to ensure the new build is working properly
      i. Type: ./run_tests
      ii. Some tests may be skipped. All tests that run should pass.
      iii. If tests fail, contact Kevin Lyon (kevin.lyon@inl.gov) or Amy Welty (amy.welty@inl.gov)
12. Adding/modifying source code (Note: ADVANCED USERS ONLY)
   a. Create a unique branch of DGOSPREY

        i. Create a branch: git branch <name_branch> (Individual branch names are user-defined. Maintaining dgosprey as part of the name may help with clarity. ie. <dgosprey_uniquename>
        ii. Move to the new branch: git checkout <name_branch>
    b. Ensure you are NOT on the master branch of DGOSPREY before editing source code.
        i. See what branch you are on: git branch
        ii. Current branch will be marked with an asterisk (*). If you have colors activated in your terminal, it will also be green. In the example below,

```
[<user>][~/projects/dgosprey]> git branch
   dgosprey_uniquename1
* dgosprey_uniquename2
    master
```

the user is in branch "dgosprey_uniquename2"

    c. Commit changes locally after editing, building, and testing.
        i. git commit –a –m "user message about the commit"
        ii. All commits require a message.
    d. Activate git tracking on new source code files
        i. git add include/NewFile.h
        ii. git add src/NewFile.C
    e. New branches are local. Save a new branch globally by pushing to the git repository.
        i. git push –u origin <branch_name>
        ii. Only changes that have been committed may be pushed.
        iii. DO NOT push to "origin master"

13. Running DGOSPREY
    a. Run with terminal command line, from within dgosprey directory
        i. Run in serial: ./dgosprey-opt –i path/to/input/file.i
        ii. Run in parallel: mpiexec –n 8 ./dgosprey-opt –i path/to/input/file.i
    b. Run with Peacock
        i. From terminal, within dgosprey directory: peacock –i path/to/input/file.i
        ii. Run directly from Peacock
            1. In terminal, within dgosprey directory: Peacock

2. Once the Peacock interface is running, click on "Open" and navigate to the input file.

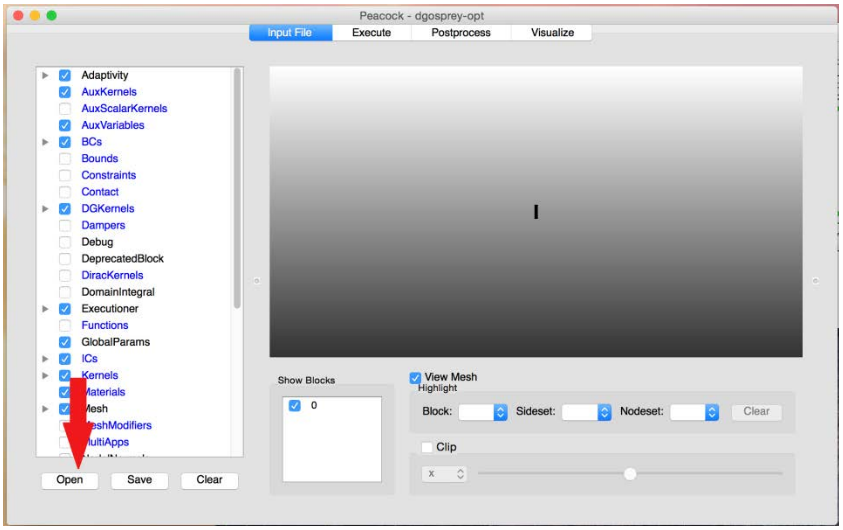


Figure 2: Peacock initial screen upon opening

14. Viewing results in tabular form
    g. Results shown in graphical form in the postprocess tab are saved as a .csv file that can be opened in Excel. The .csv file will be located in the same folder where the input file is located.
15. Any modifications made to the input file by changing parameters and other values can be saved as a new input file.
    h. Go to the input file tab in Peacock.
    i. Click the "save" button
    j. Choose name and directory
    k. Click "save".
16. For assistance running DGOSPREY, contact Kevin Lyon (kevin.lyon@inl.gov), Amy Welty (amy.welty@inl.gov), or Austin Ladshaw (aladshaw3@gatech.edu).

4. Example

New DGOSPREY users are encouraged to test their build using the input file(s) included. After building DGOSPREY, the build should be tested first by running its unit tests (e.g. type "./run_tests"). For larger scale simulations, several example and simulation files have been included in the DGOSPREY repository. A file named Kr_Xe_Mix_253K.i, included with the DGOSPREY build under input_files/examples, was used to produce the output seen in Figures 3-6 and uses the following parameters:

Gas feed rate:	2994.06 cm^3/hr

Gas composition:	Kr = 0.01336 kPa, Xe = 0.08748 kPa,
	He = 101.249 kPa

Adsorption temperature:	253.15 K
Sorbent:	AgZ-PAN
Sorbent mass:	18.26 g
Column material:	Stainless Steel
Column length:	22.86 cm
Column OD:	1.905 cm
Column wall thickness:	0.1778 cm

All output from DGOSPREY is placed in the same directory as the input file that was just run. For the example shown above, the output files would be found in input_files/examples/. 2D graphical outputs may be obtained in Peacock or ParaView, which is free visualization software that can read exodus (.e) files (http://www.paraview.org/). Figures 3 and 4 show some graphical output generated by DGOSPREY by opening the Kr_Xe_Mix_253K_out.e file with ParaView. The plot in Figure 3 shows the Xe concentration at a particular time as a function of bed length. Figure 4 shows the changes in column pressure as a result of changing temperature and mass transfer in the bed. It is also plotted at the same time stamp as the Xe curve in Figure 3 and as a function of the bed length.

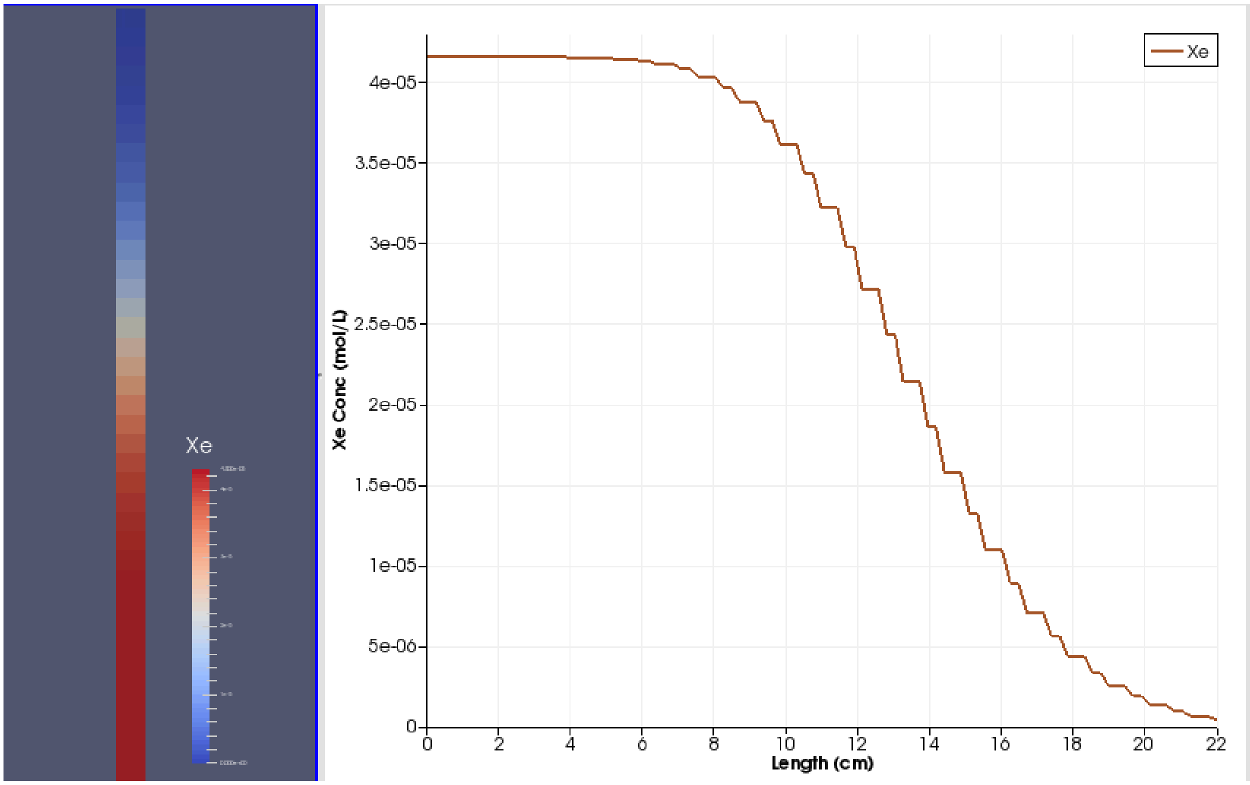


Figure 3: Xe concentration in the bed taken at a particular time stamp. This image was generated in ParaView.

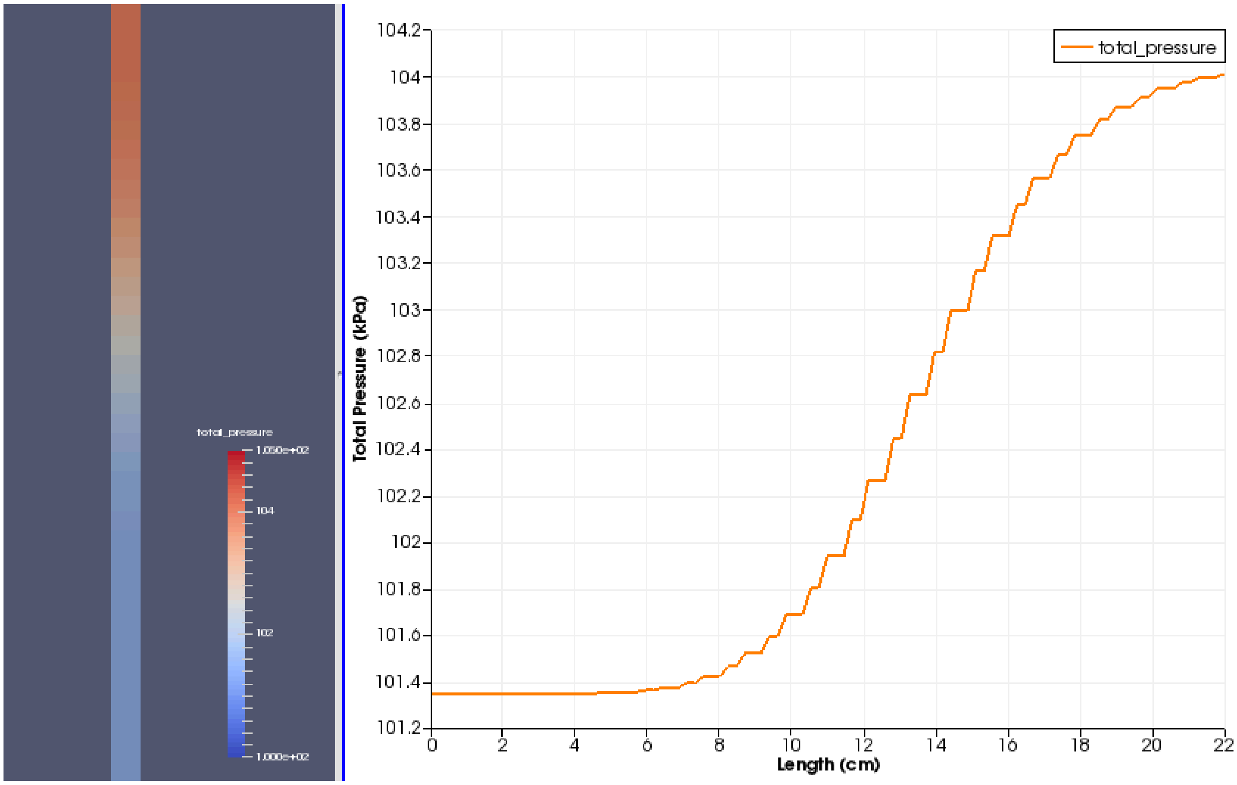
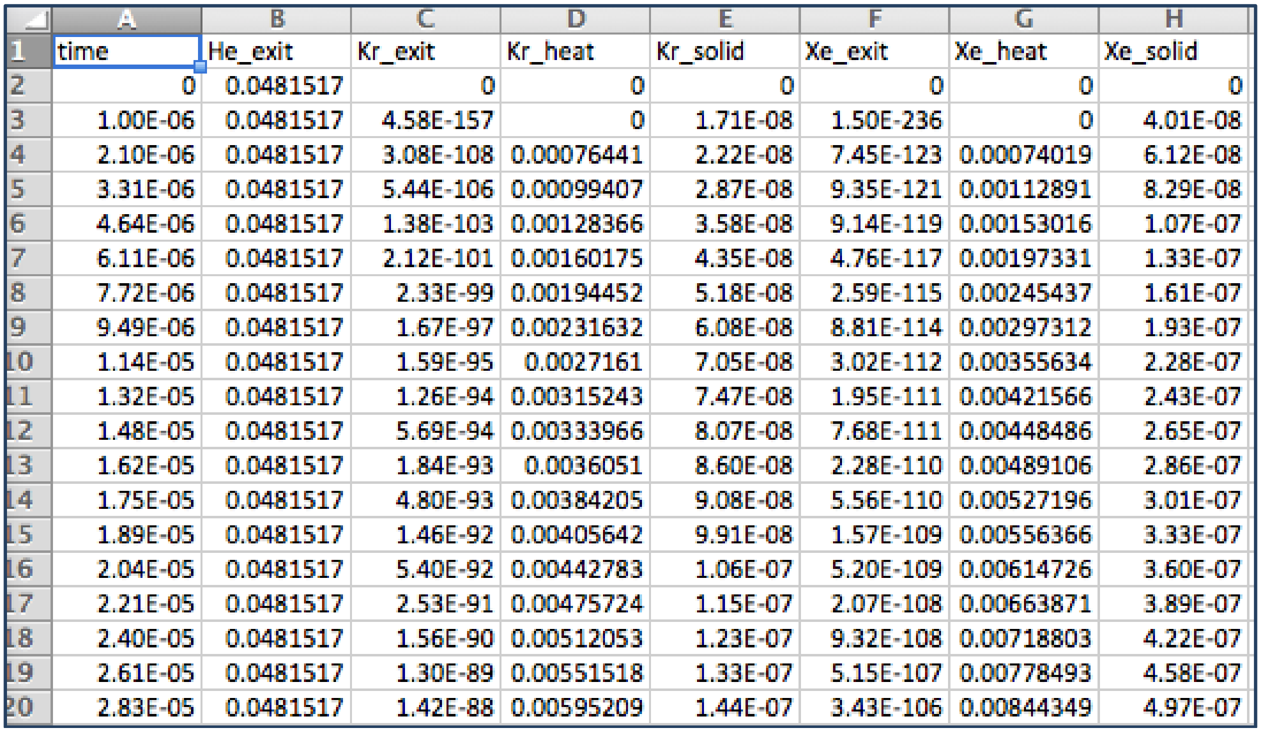


Figure 4: Pressure change in the bed taken at the same time stamp as in Figure 3. This image was generated in ParaView.

In addition to the exodus file produced by DGOSPREY, a CSV file is also produced for plotting particular information about the simulation over time. The CSV file can be found in the same location as the exodus file (input_files/examples/Kr_Xe_Mix_253K_out.csv). Figure 5 shows what the output in the CSV file should look like when opened in excel (Note: The time stamps may be different in the output file when running the simulation with time adaptivity enabled). The CSV file can be used to plot specific information, such as concentration breakthrough curves or average column adsorption. Figure 6 shows the breakthrough curve for Xe in this particular simulation. Users should expect the same results upon running this simulation.

Note that neither ParaView nor the CSV outputs give any units. The units are specified by the individual MOOSE applications and are never explicitly stated on output. For DGOSPREY, time is always in hours, concentrations are in mol/L, adsorption is in mol/kg, lengths are in cm, and pressure is in kPa. Additional information regarding the units can be found in the DGOSPREY source code or the reference manual.



	A	B	C	D	E	F	G	H
1	time	He_exit	Kr_exit	Kr_heat	Kr_solid	Xe_exit	Xe_heat	Xe_solid
2	0	0.0481517	0	0	0	0	0	0
3	1.00E-06	0.0481517	4.58E-157	0	1.71E-08	1.50E-236	0	4.01E-08
4	2.10E-06	0.0481517	3.08E-108	0.00076441	2.22E-08	7.45E-123	0.00074019	6.12E-08
5	3.31E-06	0.0481517	5.44E-106	0.00099407	2.87E-08	9.35E-121	0.00112891	8.29E-08
6	4.64E-06	0.0481517	1.38E-103	0.00128366	3.58E-08	9.14E-119	0.00153016	1.07E-07
7	6.11E-06	0.0481517	2.12E-101	0.00160175	4.35E-08	4.76E-117	0.00197331	1.33E-07
8	7.72E-06	0.0481517	2.33E-99	0.00194452	5.18E-08	2.59E-115	0.00245437	1.61E-07
9	9.49E-06	0.0481517	1.67E-97	0.00231632	6.08E-08	8.81E-114	0.00297312	1.93E-07
10	1.14E-05	0.0481517	1.59E-95	0.0027161	7.05E-08	3.02E-112	0.00355634	2.28E-07
11	1.32E-05	0.0481517	1.26E-94	0.00315243	7.47E-08	1.95E-111	0.00421566	2.43E-07
12	1.48E-05	0.0481517	5.69E-94	0.00333966	8.07E-08	7.68E-111	0.00448486	2.65E-07
13	1.62E-05	0.0481517	1.84E-93	0.0036051	8.60E-08	2.28E-110	0.00489106	2.86E-07
14	1.75E-05	0.0481517	4.80E-93	0.00384205	9.08E-08	5.56E-110	0.00527196	3.01E-07
15	1.89E-05	0.0481517	1.46E-92	0.00405642	9.91E-08	1.57E-109	0.00556366	3.33E-07
16	2.04E-05	0.0481517	5.40E-92	0.00442783	1.06E-07	5.20E-109	0.00614726	3.60E-07
17	2.21E-05	0.0481517	2.53E-91	0.00475724	1.15E-07	2.07E-108	0.00663871	3.89E-07
18	2.40E-05	0.0481517	1.56E-90	0.00512053	1.23E-07	9.32E-108	0.00718803	4.22E-07
19	2.61E-05	0.0481517	1.30E-89	0.00551518	1.33E-07	5.15E-107	0.00778493	4.58E-07
20	2.83E-05	0.0481517	1.42E-88	0.00595209	1.44E-07	3.43E-106	0.00844349	4.97E-07

Figure 5: Image of the CSV output when opened in excel. Information printed out here includes the breakthrough curves, average adsorption, and average heats of adsorption for the Kr and Xe species in the bed.

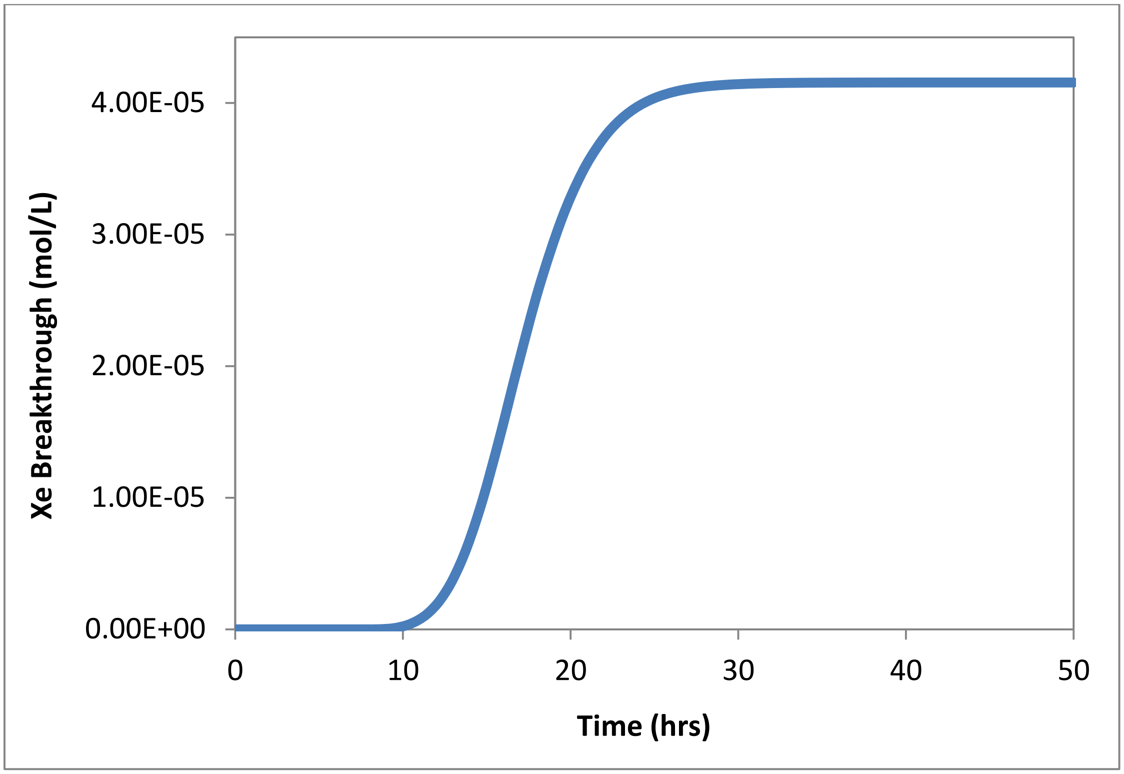


Figure 6: Breakthrough curve generated from the CSV file shown in Figure 5.