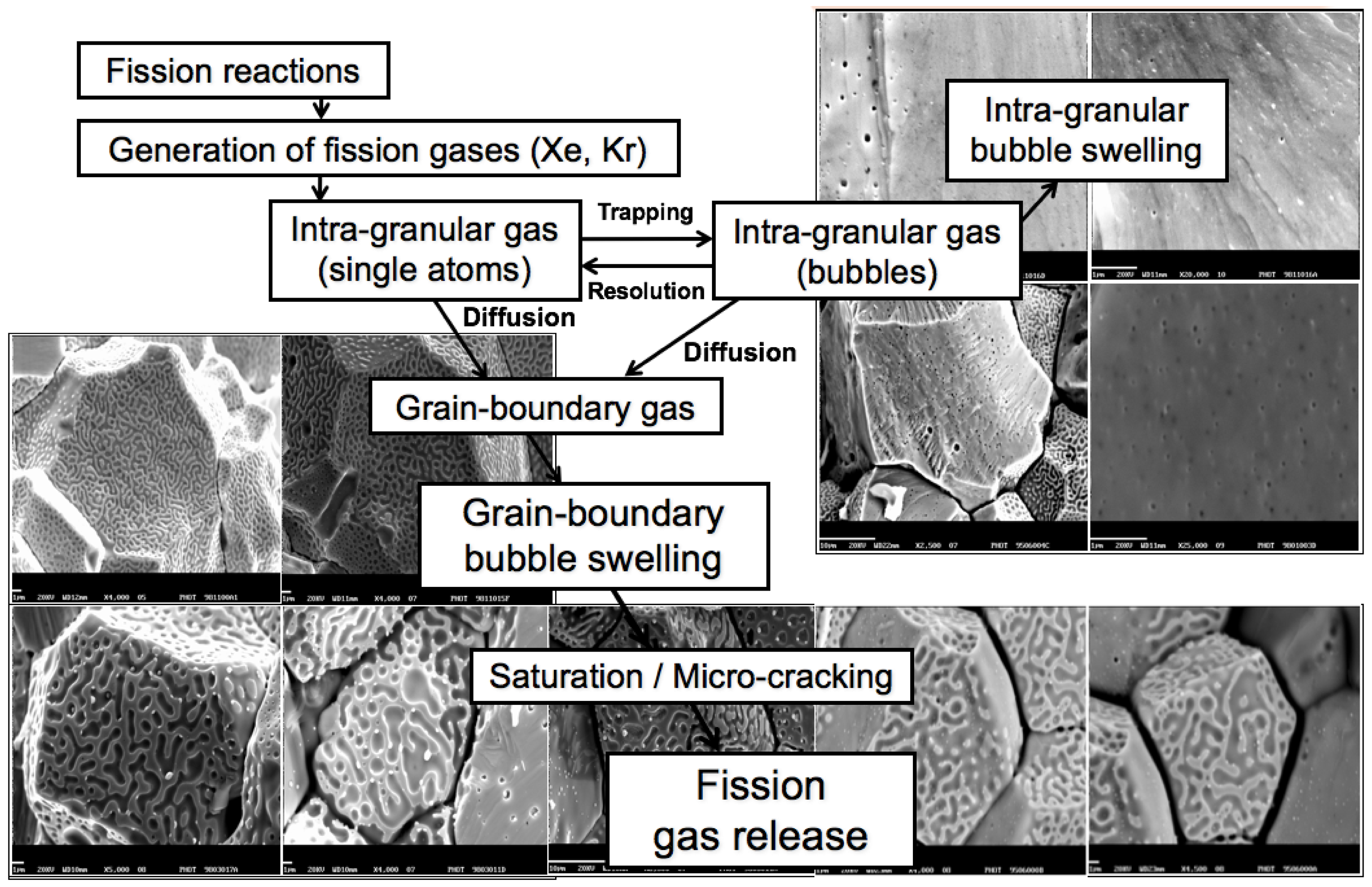
Xenon Gas Bubble Re-solution Model and Xolotl Code Development

Sophie Blondel[1], Philip C. Roth[2], David E. Bernoldt[2], David Andersson[3], Brian D. Wirth[1,2]
[1]University of Tennessee, Knoxville, TN
[2]Oak Ridge National Laboratory, Oak Ridge, TN
[3]Los Alamos National Laboratory, Los Alamos, NM

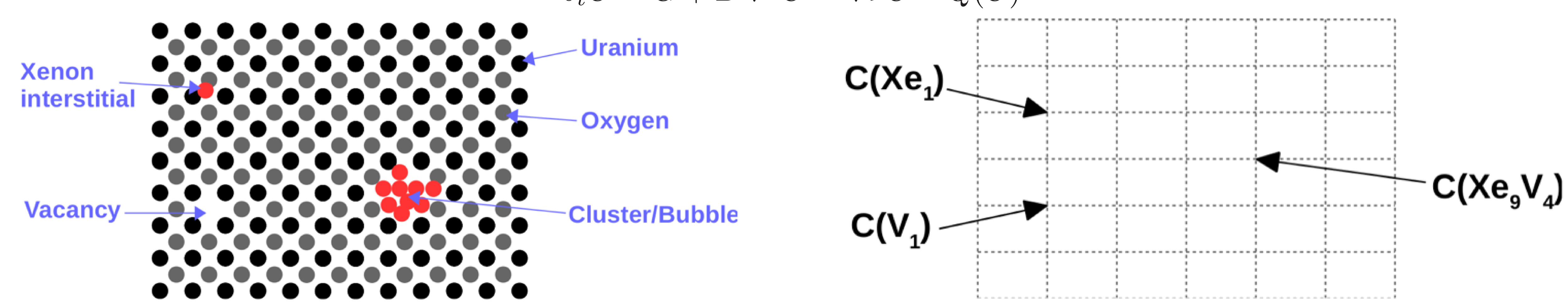
The Long Standing Fission Gas Problem



Introduction to Xolotl (https://github.com/ORNL-Fusion/xolotl)

Xolotl, named after the Aztec god of death and lightning, predicts the evolution of gas in a solid by solving the cluster dynamics formulated Advection-Diffusion-Reaction (ADR) equations with a volumetric source term, and is being used by both the NE (fission gas) and FES (Plasma Surface Interactions) SciDAC projects

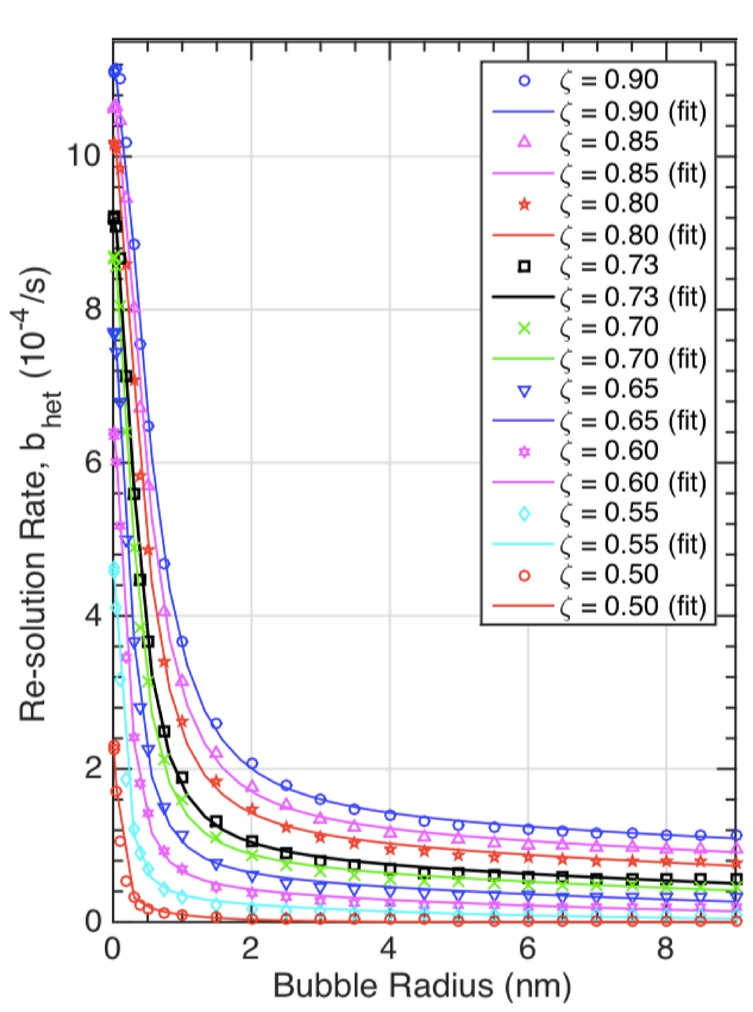
$$\delta_t \bar{C} = \bar{G} + D\nabla^2 \bar{C} - \nabla \bar{\nu} C - \bar{Q}(\bar{C})$$



- A network of clusters represents the material (interstitial and vacancy) and the gas atoms and clusters.
- The solver (PETSc) is in charge of the time evolution of the concentrations.
- The re-solution of clusters due to thermal spikes is modeled through $Xe_i \rightarrow Xe_{i-1} + Xe_1$ with the associated rate described next.

Re-Solution Model

Extensive molecular dynamics simulation study was performed to obtain the Xe re-solution rates due to thermal spike as a function Xe bubble radius, for several values of the ratio between the thermal spike energy and the total electronic stopping power, and for a fission rate density of 10^{-8} /nm3 /s.

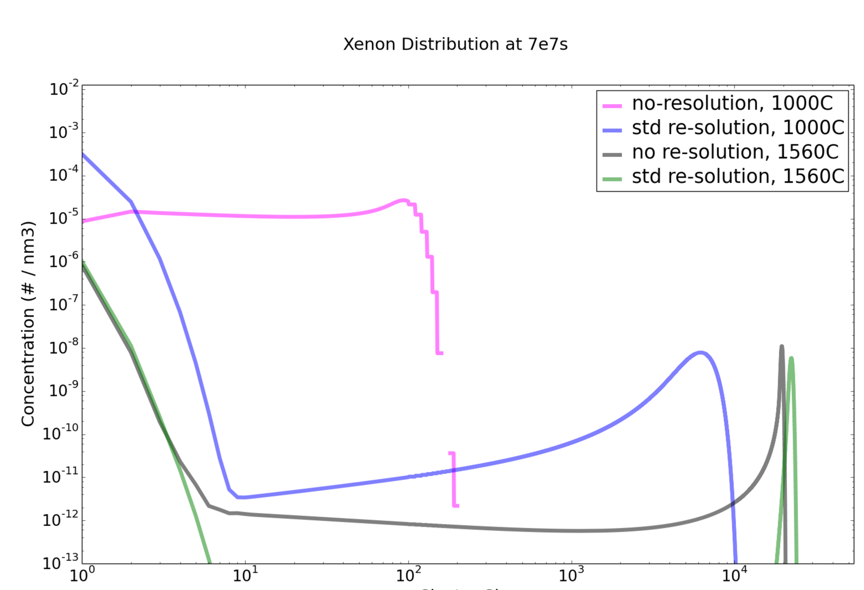


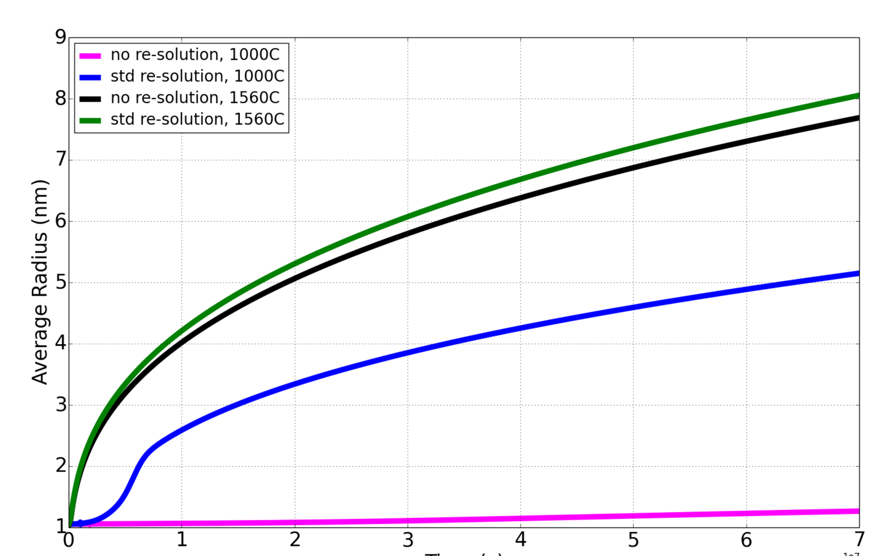
Setyawan *et al.*, J. Appl. Phys. 124, 075107 (2018)

In Xolotl we use the re-solution rate (10^{-4}/s):

$$b = (a_1 e^{-b_1 x} + \left(\frac{y(0) - a_1}{1 + cx^2}\right) e^{-b_2 x^2}) \times 10^{-4}$$

where x is the bubble radius in nm and $y(0)$ is the asymptotic value at $x = 0$.





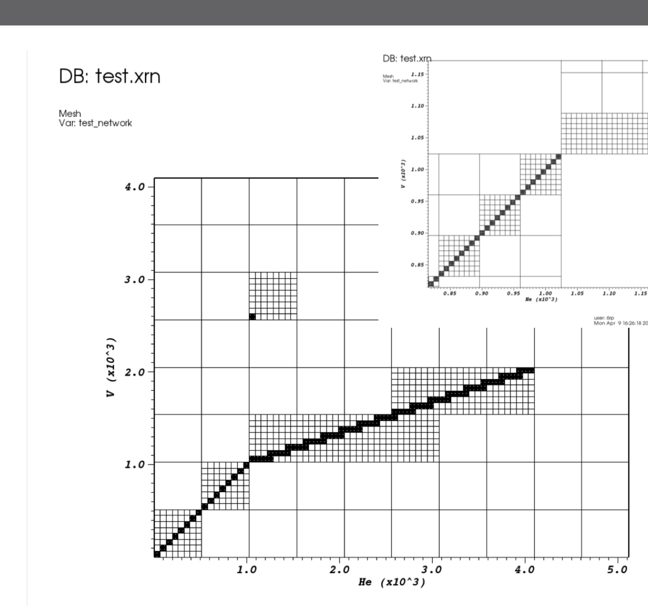
Average radius is evaluated taking into account Xe bubbles larger than 2nm diameter (containing 100 Xe atoms).

⇒ **The effect of re-solution is much larger at lower temperature.**

Next Steps

Current and future work:

- Evaluate the effect of the Xe density in combination with re-solution on the average radius
- Include the diffusion model developed from DFT simulations
- Extend the reaction network to explicitely model fuel vacancies
- XRN: XDSpace-based implementation of Xolotl reaction network

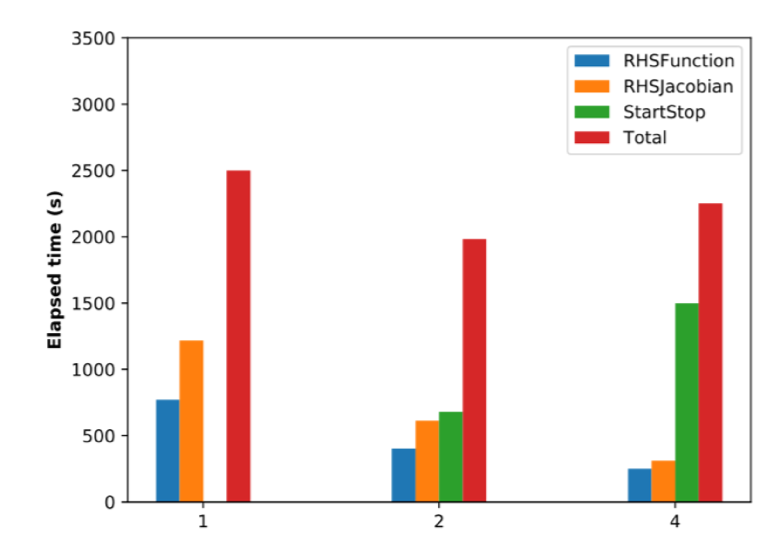


Acknowledgments:
This work was supported by the U. S. Department of Energy, Office of Fusion Energy Sciences, Office of Nuclear Energy, and Office of Advanced Scientific Computing Research through the Scientific Discovery through Advanced Computing program.

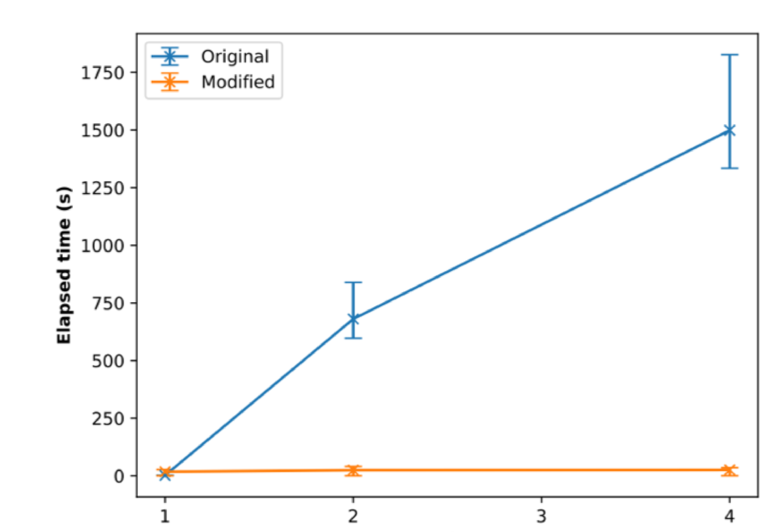
Code Development

Recent performance engineering:

- Built-in timers showed increasing cost in Xolotl's `startStop1D()` PETSc callback as number of nodes increased
- Diagnosed as I/O problem when writing HDF5 concentration file
- Changing code to use true HDF5 parallel write gave substantial performance and performance variability improvement



Elapsed time as reported by built-in Xolotl timers when running on Oak Ridge Leadership Computing Facility (OLCF) Eos Cray XC30. 32 processes per node. Data shown is max of reported timer value across all processes, averaged over at least three runs.



Elapsed time as reported by built-in Xolotl timers when running on OLCF Eos Cray XC30. 32 processes per node. Data shown is max of reported timer value across all processes, averaged over at least three runs. Bars show min/max of values. ⇒ **57.5x faster at 4 nodes.**

Adding accelerator support:

- After evaluation of alternatives (e.g., CUDA, OpenMP offload), we plan to use Kokkos as portability layer for executing performance-critical code on GPUs and CPUs
- Ongoing effort to adapt Xolotl data structures for Kokkos' View type
  - Xolotl (e.g., reaction network) currently creates many small C++ objects in deep, complicated organization, connected by pointers
  - Kokkos View is a multidimensional, rectangular array
  - XDSpace: a C++ library for decomposing a multidimensional space (e.g., Xolotl's phase space), with flexible refinement and Kokkos support
- Investigating how to incorporate Kokkos acceleration into an application that uses PETSc