Product Requirements Document (PRD)

[Problem Description](#_m14ebq69irye)

[Scope](#_h095bb6e23m6)

[Use Cases](#_a494vmyn4i0c)

[Purpose and Vision (Background)](#_a1rehjp14t6a)

[Stakeholders](#_ni1nsrov0hlq)

[Preliminary Context](#_8fsnkm7d5o2g)

[Assumptions](#_o6z1frxke2z)

[Constraints](#_q1it1bwhygi3)

[Dependencies](#_hn9mh6b6m132)

[Market Assessment and Competition Analysis](#_kztlc8doxc1v)

[Requirements](#_agmoxh3nwlca)

[User Stories and Features (Functional Requirements)](#_26r2y8q9o6bs)

[Non-Functional Requirements](#_phuqv9aimzm2)

[Data Requirements](#_89injk5wussz)

[Integration Requirements](#_ti0b4uln9ysy)

[User Interaction and Design](#_jq7dpkh2hu4j)

[Milestones and Timeline](#_u3bp37hkrgxj)

[Goals and Success Metrics](#_ybblbzrnua5i)

[Open Questions](#_9p4b6jpz1kep)

[Out of Scope](#_c0bknaj5akzz)

# **Problem Description**

Current models for simulating the evolution of material defects caused by radiation damage on the microscopic scale take far too long to run and are bottlenecked by the speed of a single CPU core, limiting their usefulness.

## **Scope**

The scope of this project will be limited to simulating a particular model of radiation damage called cluster dynamics taking advantage of GPU parallelism. The most basic success would be the demonstration of a cluster dynamics simulation which can recreate existing results but can take advantage of GPU resources to compute them faster than they can be computed on a CPU.

Stretch goals include altering this model to incorporate more advanced considerations, such as spatial resolution.

## **Use Cases**

The project partner will benefit directly from this project because it will allow them to conduct simulations on a scale which was previously impractical to simulate.

Being able to run larger simulations can be used to help predict the behavior of materials in a high-radiation environment over the course of decades. In addition, by assuring the simulation is faithful to an underlying cluster dynamics model, any results from the simulation that disagree with experimental evidence can serve to critique the underlying model.

# **Stakeholders**

* Project Partner

The primary decision-maker because they will use this project directly. They will receive updates at least once per week, or more if the engineering team has questions about the product requirements. They require insight from the engineering team to know what is possible with the computational resources we have access to.

* Engineering Team

Will receive updates multiple times per day from other team members, especially as their work affects each other. They will make most of the low-level decisions about the computational aspect of the project. They require insight from the project partner to understand the materials science and physics aspects of the project, along with what would best serve them in a final product.

# **Preliminary Context**

## **Assumptions**

The program we develop should be able to run on Linux and utilize the resources of a GPU in a headless environment.

We will have access to the DGX at OSU for computing resources.

We have about nine months with our engineering team to bring the project to a place that is usable.

We will use the agile development process to easily adapt to changes in requirements. We will periodically address the requirements and how they may have changed, and adjust our weekly tasks accordingly.

## **Constraints**

We have limited knowledge going into this project on materials science. We will need to rely on the knowledge and research papers provided by our project partner.

We will need to utilize the GPU to speed up these simulations because there’s likely no way to get a significant speedup without relying heavily on parallelism.

We have a low number of engineering hours per week because each of us has other obligations. We can probably realistically expect at most 10 hours of engineering time per week out of each engineer.

## **Dependencies**

Any progress on creating the simulation first relies on the engineering team getting up to speed on the model of radiation damage we’ll be using.

Given that CUDA is a dependency for the GPU acceleration it is a likely place for a performance/developmental bottleneck to occur.

# **Market Assessment and Competition Analysis**

There are many existing models for nuclear degradation, all of which serve different specific purposes. These models include Monte Carlo neutronics calculations, Metropolis Monte Carlo, molecular dynamics, binary collision approximation, and more. The scope of this project is to explore cluster dynamics, which most likely already has existing models as well, although we have not been able to find any open-source GPU-accelerated ones. Our goal is to improve upon existing theoretical models by speeding them up using CUDA.

# **Requirements**

## **Features (Functional Requirements)**

The cluster dynamics simulation must be accurate to the model presented in a paper called “*Irradiation damage in 304 and 316 stainless steels: experimental investigation and modeling*” by C. Pokor et al.

In addition, the model will be augmented with some corrections made by the project partner, along with a different model of dislocation network evolution provided in a paper called “*Heterogeneous Dislocation Formation And Solute Redistribution Near Grain Boundaries In Austenitic Stainless Steel Under Electron Irradiation*” by N. Sakaguchi et al.

Must Have:

* A cluster dynamics simulation which includes the following:
  + Tracking of the concentrations of clusters of various sizes according to the paper by Pokor.
  + Generation of clusters in the initial collision cascade according to the paper by Pokor.
  + Evolution of clusters by exchange of single vacancies and interstitials according to the paper by Pokor.
  + Evolution of the dislocation network, including the effect of cluster growth on the dislocation network, according to the paper by Sakaguchi.
* The ability to export the raw data of the results of the simulation.
* GPU acceleration of the simulation.

Should Have:

* Visualization of results.
* Input parameters that are accurate to a real material in a real reactor.

Could Have:

* Interactive / real time visualization of results as they are generated.

Will Not Have:

* Cloud deployment. This will require a level of maintenance after development that we don’t expect to be available, so we have opted instead to make the program available as an executable so the user can make whatever arrangements they are comfortable with.

## **Non-Functional Requirements**

* Code should be well-documented, following coding standards and best practices.
* The product should be able to scale up for use with large computing resources, such as GPU clusters.
* There should be an increase in the performance in a single process implementation versus parallelized code.

## **Data Requirements**

The data for this project requires us to provide the ability to export it in a raw format such as a csv file, but also to present it visually.

The csv file will make accessible many popular data visualization tools, for example, Matlab.

## **Integration Requirements**

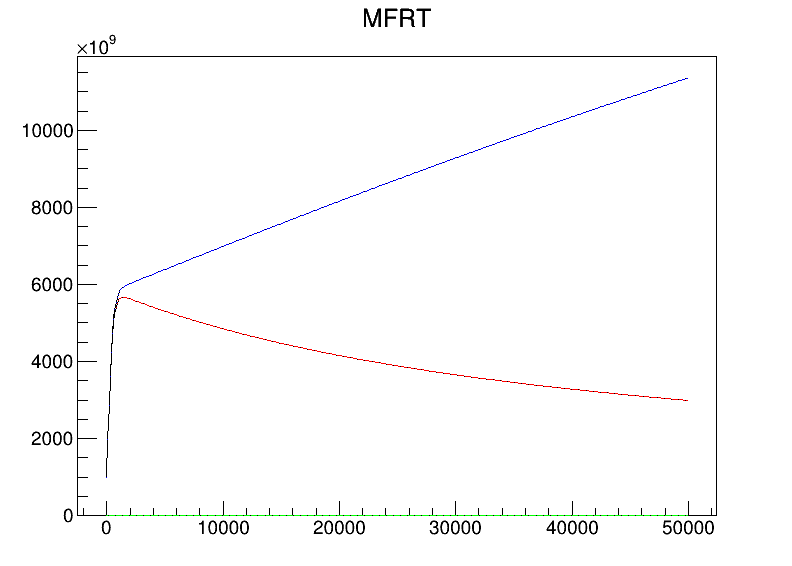
The program needs to be able to interface with a GPU using CUDA.

The program data must also be in an accessible format so the User Interface can properly present the data.

We will use continuous integration through GitHub Actions to ensure that all versions of the program will build when integrated into the main branch. Further tests can be used to ensure that the program runs accurately.

## **User Interaction and Design**

The program should be interactable by the user via a user interface. It should clearly and accurately represent the data output by the model.

****

A small demonstration of the UI for our earliest prototype. An upgraded version of this will include axis labels and lines representing concentrations of multiple cluster types.

Bugs that are found while development is still active can be discussed through the project Discord server.

# **Milestones and Timeline**

The general outline of significant milestones for version 0 of this project can be described as follows:

1. Create an accurate model simulation of Nuclear Degradation due to radiation (Week 9).
2. Create a UI for user interaction (Week 7)
3. Integrate UI with existing simulation (Week 9)
4. Utilize GPU parallelization to increase simulation speed into a relevant timescale (Week 10).
5. Finalize robust documentation and ensure the product usage is straightforward. Allow pull requests and make the engineers’ emails available so that users can suggest changes after the product is launched (Post-launch)

The most likely place for development bottlenecks to occur is the model creation as other tasks such as UI integration, documentation, and GPU parallelization cannot occur until at least some version of the model exists.

# **Goals and Success Metrics**

| **Goal** | **Metric** | **Baseline** | **Target** | **Tracking Method** |
| --- | --- | --- | --- | --- |
| Speedup cluster dynamics simulation | Percent speedup over baseline | Simulations in the timescale of picoseconds | Simulates in the timescale of years | A built-in delta time to the program. |
| Create an easy-to-understand UI for the target demographic | Approval from the project partner that UI is useable as a member of the target demographic | UI displays output data but is difficult to use | Users can use the program without great difficulty | Verbal/Written feedback on UI |

# **Open Questions**

How much of a speedup can we expect from GPU parallelization of the chosen model?

Aside from spatial resolution, what additional physical considerations might we want future cluster dynamics models to take into account?

# **Out of Scope**

Cloud deployment may be out of scope. Although cloud resources may be useful, we don’t expect a level of maintenance to be available once this team of developers has left the project in June which is able to deal with any unexpected issues that might arise in a cloud-deployed application. For this reason, we’re limiting ourselves to building something which can be operated by the end user end-to-end.