

PROGRESS ON DART CODE OPTIMIZATION

Taboada, Horacio¹ Rest, Jeffrey² Solís, Diego¹

¹Comisión Nacional de Energía Atómica, República Argentina

²Argonne National Laboratory, Illinois, USA

ABSTRACT

This work consists about the progress made on the design and development of a new optimized version of DART code (DART-P), a mechanistic computer model for the performance calculation and assessment of aluminum dispersion fuel. It is part of a collaboration agreement between CNEA and ANL in the area of Low Enriched Uranium Advanced Fuels. It is held by the Implementation Arrangement for Technical Exchange and Cooperation in the Area of Peaceful Uses of Nuclear Energy, signed on October 16, 1997 between US DOE and the National Atomic Energy Commission of the Argentine Republic. DART optimization is a biannual program; it is operative since February 8, 1999 and has the following goals:

1. Design and develop a new DART calculation kernel for implementation within a parallel processing architecture
2. Design and develop new user-friendly I/O routines to be resident on Personal Computer (PC)/WorkStation (WS) platform.
 - 2.1. The new input interface will be designed and developed by means of a Visual interface, able to guide the user in the construction of the problem to be analyzed with the aid of a new database (described in item 3, below). The new I/O interface will include input data check controls in order to avoid corrupted input data.
 - 2.2. The new output interface will be designed and developed by means of graphical tools, able to translate numeric data output into "on line" graphic information.
3. Design and develop a new irradiated materials database, to be resident on PC/WS platform, so as to facilitate the analysis of the behavior of different fuel and meat compositions with DART-P. Currently, a different version of DART is used for oxide, silicide, and advanced alloy fuels.
4. Develop rigorous general inspection algorithms in order to provide valuable DART-P benchmarks.
5. Design and develop new models, such as superplasticity, elastoplastic feedback, improved models for the calculation of fuel deformation and the evolution of the fuel microstructure for inclusion into DART-P
6. Develop a DART-P capability for the analysis of dispersion fuel behavior during transient and/or accident conditions. The work will include the development of a DART-P restart-from-dump capability.

Up to date, issues 2 and 3 were designed and developed and they are under evaluation. These new facilities allow the user to have a friendlier guide to design the problem to study and to display graphically the outcome. The material database allowed the code unification, with the subsequent code support simplification.

Introduction:

DART (Dispersion Analysis Research Tool) is the unique mechanistic model for the assessment of dispersion fuel behavior for oxide, silicide, and other new dispersant phases (vhd alloys).

However, due to its evolution as an R&D tool, it was not developed as a user-friendly code. Besides, each simulation done to study the effect of changing a particular parameter or operating condition, demands the iteration of a process consisting in input preparation, DART run, extraction of calculated quantities from program output file and plotting. Each step of the calculation process has an extension of a couple of minutes to many hours, depending on complexity of the problem. For an analysis covering a multitude of parameters and/or operating conditions, this is a very long and tedious process.

The conversion of DART into a parallel architecture version was an ideal situation to improve calculation power, due to the increasing tendency in IT technology to appeal to multithreading techniques and parallel processing.

As a part of SISTERLAB agreement, ANL and CNEA have proposed several topics for mutual collaboration. One of them is related with modeling. It consists of a full revision of DART models and version codes and the inclusion of new models in the framework of the development of a unique parallel architecture version for DART code.

Several aims are pursued, namely to enhance DART input/output structure in order to increase its availability and usefulness in the international community and to afford the opportunity to develop an interface whereby the user can monitor the evolution of various calculated quantities "in situ."

In addition, it will provide the possibility for the user to change values of various parameters and/or operating conditions during the course of a run. The user/code dialog will become highly optimized and the analysis procedure will be more efficient. It is intended to parallelize a variety of calculations performed as a function of operating conditions and fuel morphology, like

- Evolution of the fission-gas bubble size distribution and meat thermal conductivity.
- Fuel-meat matrix interaction
- Evolution of fuel microstructure
- Stress/strain analysis,

and other issues. These processes, if parallelized, will provide a much more efficient calculation. It was thought that a project like this also would allow the merging of all different versions of DART into a single code. The project include the aim of developing new models such as superplasticity, elastoplastic feedback, improved models for the calculation of fuel deformation and fuel microstructure evolution. As a whole this project implies a rigorous inspection and overhaul of DART bringing the user and developer of the international community a very valuable benchmark and also it will form the basis of a code for the analysis of dispersion fuel during transient (and/or accident) conditions.

Works done up to date on different goals:

1. Friendly interface:

A friendly input-output interface was created. To design different windows for user-aid data input, a visual language (Visual Basic[®]) was used. This way, data was collected into 5 groups:

- 1.1. I/O files definition: including complete path (drive, directory, and file) where will be written the outcome data.
- 1.2. Data concerning geometric kind and dimension of fuel assembly to be modeled
 - 1.2.1. Plate, rod or tube
 - 1.2.2. Meat and cladding height, width and thickness
- 1.3. Data concerning material kind of fuel
 - 1.3.1. Oxide, silicide or vhd
 - 1.3.2. Particle size, grain size, fuel volume fraction, cladding volume fraction, external –as fabricated- and internal porosity
- 1.4. Data concerning simulation condition
 - 1.4.1. Fission gas to be considered (Xe, Ar or Kr)
 - 1.4.2. Bubble class to calculate its evolution (bulk, dislocation, grain face and edge)
 - 1.4.3. Number of grain radial partition into equal volume shells
 - 1.4.4. Whole simulation time
 - 1.4.5. External pressure
 - 1.4.6. For the each calculation stage, a matrix array containing step numbers, linear power, border and centerline temperatures, internal pressure and printout options was made
 - 1.4.7. Calculation of time step
- 1.5. Command to start DART execution

2. Unified DART version (oxide, silicide, and advanced alloy fuel)

A revision of different DART versions was made. It was possible to merge these versions into a single program code, around GRASS subroutine. This subroutine calculates fission gas bubble distributions according to increasing bubble size and bulk, dislocation, face and edge classes. It constitutes the core of DART program. Evolutionary processes like recrystallization, U-Al reaction, amorphization, etc; usual issues of DART versions are programmed in different subroutines. The new unified version has respected the calling logic of each version. Material and model constants of DART versions are resident on respective arrays. For the sake of code unification, it was built a common array made of shared (model and cladding) data, and a separate matrix made of specific material properties for each kind of fuel.

Progress made on the first point (friendly input interface) allowed the replacement of original data reading structure. It contributed to clarify the code logic.

Once the code was purified, it was compiled as a dynamic library link (DLL) subroutine, to allow the visual interface calling to start calculation process. It was designed a thread (Win32 application) to create, open, dump data and close auxiliary files. These files contain evaluated (partial) outcome data.

3. Chart representation of outcome data

Auxiliary files mentioned above are used to allow the user to have a plot chart representation of calculated data during runtime. A visual interface calling a plotting thread was designed to perform this issue, using some facilities given by the visual language support. Up to date, this stage is under optimization.

Conclusion:

Some progress was made on DART optimization, according with original purpose biannual project. A new visual input interface, a dynamic library link and a plotting chart representation facility were designed and built. Next development goals will be runtime optimization by means of paralellization techniques and the development and implementation of new models to describe superplasticity, elastoplastic feedback, and dispersion fuel behavior during transient.