FASTDART
A FAST, ACCURATE AND FRIENDLY VERSION OF DART CODE

Rest, Jeffrey[1] Taboada, Horacio[2]
[1]Argonne National Laboratory, Illinois, USA
[2]Comisión Nacional de Energía Atómica, República Argentina

ABSTRACT

A new enhanced, visual version of DART code is presented. DART is a mechanistic model based code, developed for the performance calculation and assessment of aluminum dispersion fuel. Major issues of this new version are the development of a new, time saving calculation routine, able to be run on PC, a friendly visual input interface and a plotting facility. This version, available for silicide and U-Mo fuels, adds to the classical accuracy of DART models for fuel performance prediction, a faster execution and visual interfaces. It is part of a collaboration agreement between ANL and CNEA in the area of Low Enriched Uranium Advanced Fuels, held by the Implementation Arrangement for Technical Exchange and Cooperation in the Area of Peaceful Uses of Nuclear Energy.

Introduction:

In a former paper, a design and development plan for a new DART calculation kernel for implementation within a parallel processing architecture, was presented by the authors. The conversion of DART code into a parallel architecture version was an ideal situation to improve calculation power. There is an increasing tendency in IT technology to appeal to multithreading techniques and parallel processing.

However, the use of this kind of hardware and techniques is not widely generalized yet, to be useful for the MTR international community of fuel researchers and developers. So, it was decided to revisit original DART programming in order to achieve a twofold goal: lower CPU time-consuming version, able to keep operative all different models included in former versions. These models were previously checked and validated by several independent comparisons against experimental data.

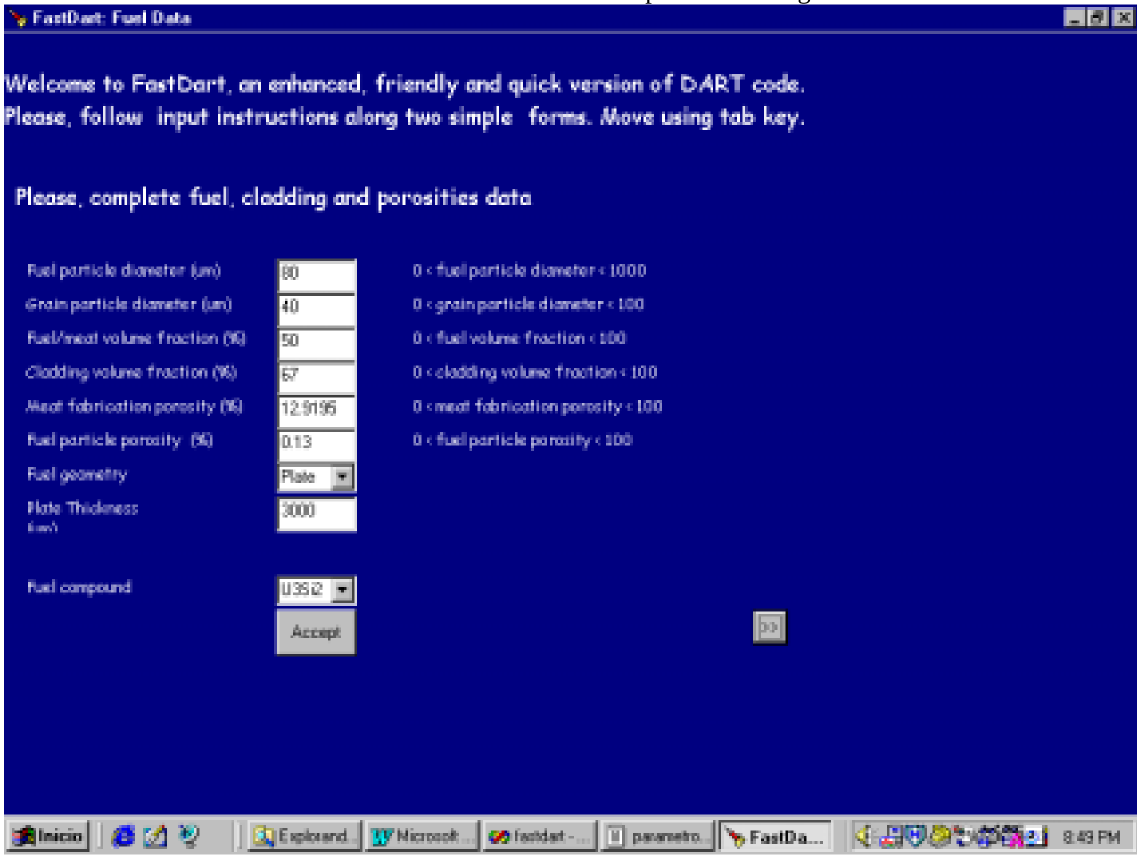
1. **Faster calculation kernel**

The main effort to reduce time calculation was done to evaluate the contribution of the coalescence of certain classes of bubbles in the balance equation of others. It was noted that some contributions of this kind were meaningless, so these coupling mechanisms between different classes balance were neglected. This way, including some minor modification of certain parameters of DART code, enabled the fulfillment of a faster PC version. This version was tested against former code outputs for several conditions, and the discrepancies found were irrelevant. A very detailed case, consisting on 20 concentric radial shells to depict the behavior of fuel particle, along a power history of 100 steps (burnup about 70% U^{235}), took about 2 minutes CPU to be ran on a Pentium III, 600 MHz. It can be said that, maintaining a radial mesh of 20 shells for fuel particle modeling and selecting 25 steps (reasonably enough detail), for history power depiction, it will take half a minute to run it. This time is compatible with what a user would expect from a code to be interactive.

2. Visual input interface

Taking account of the previous experience for visual input interface made for DART a simplification was made. Two input data screens and a validation one were enough to enter the main variables needed to define an irradiation simulation.

In the first form, input for fuel particle diameter, grain particle diameter, fuel volume fraction, cladding volume fraction, meat fabrication porosity, fuel particle porosity, fuel geometry selection and plate thickness, rod diameter or tube thickness (depending on selection) are required. If rod geometry is selected, an input box will appear to enter gap gas pressure. Finally the user must select the fuel compound. Each input box is protected from entering non-numerical, negative or out of range data


FastDart First screen: a detail of data boxes for variable input and its range limits

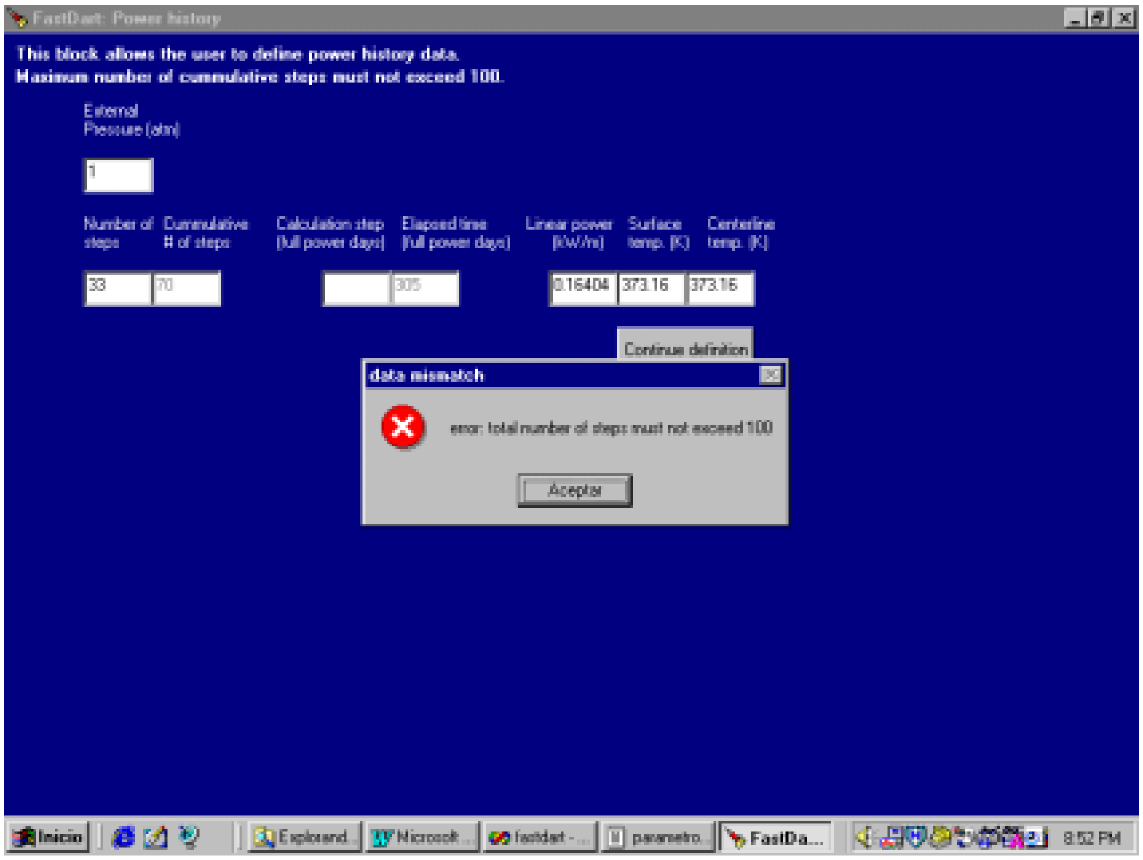
Welcome to FastDart, an enhanced, friendly and quick version of DART code.
Please, follow input instructions along two simple forms. Move using tab key.

Please, complete fuel, cladding and porosities data

Fuel particle diameter (um)	80	0 < fuel particle diameter < 1000
Grain particle diameter (um)	40	0 < grain particle diameter < 100
Fuel/meat volume fraction (%)	50	0 < fuel volume fraction < 100
Cladding volume fraction (%)	67	0 < cladding volume fraction < 100
Meat fabrication porosity (%)	12.9195	0 < meat fabrication porosity < 100
Fuel particle porosity (%)	0.13	0 < fuel particle porosity < 100
Fuel geometry	Plate	
Plate Thickness (um)	3000	
Fuel compound	U3Si2	

The second screen is prepared for power history data definition. First, the value for reactor operating external pressure is required. The user can define the power history. For a selected number of time steps, it must be defined the duration of the time step. Two dialog boxes guide the user: the total number of time steps used (it must be almost 100) and the elapsed cumulative time. Time data is entered or shown in full power days (former time definition –sec- divided by 86400) bringing a more direct contact between simulation and a real case. For each number of time steps, common values of time steps duration, linear power, surface and centerline fuel temperature must be indicated. A very flexible way to describe the power history is achieved.

FastDart Second screen: the number of steps is greater than 100, so it must be corrected



This block allows the user to define power history data.
Maximum number of cummulative steps must not exceed 100.

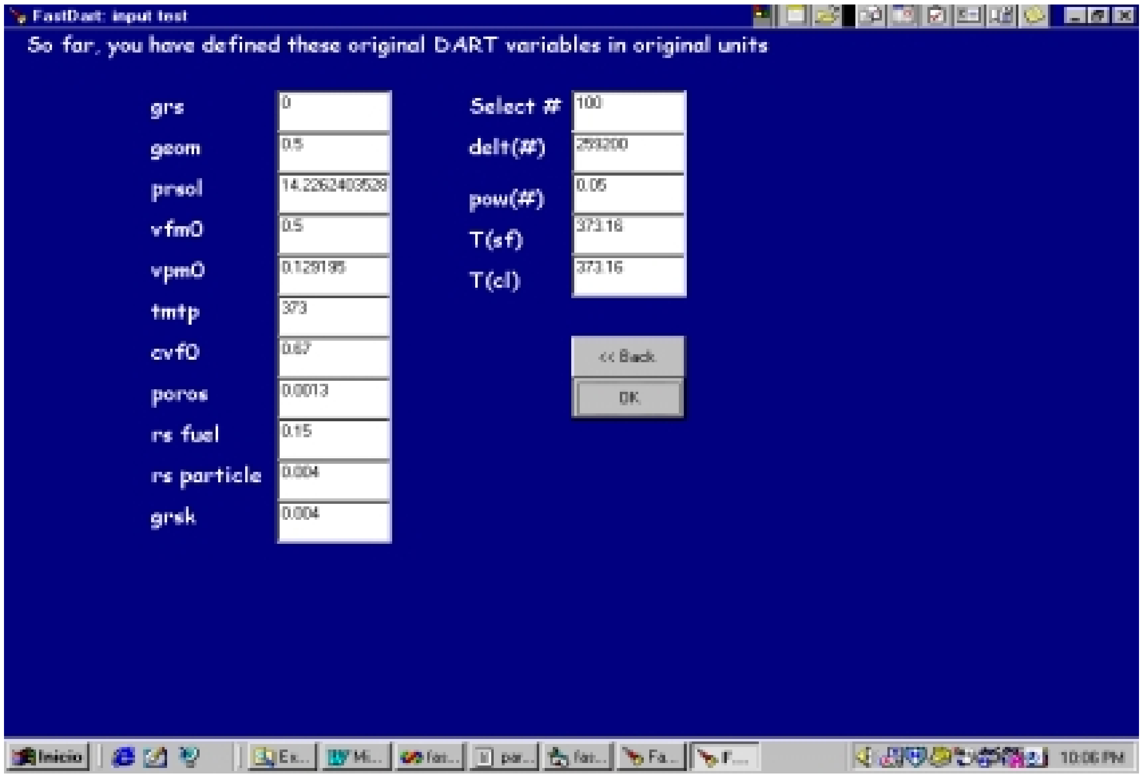
External Pressure (atm): 1

Number of steps	Cummulative # of steps	Calculation step (full power days)	Elapsed time (full power days)	Linear power (kW/m)	Surface temp. (K)	Centerline temp. (K)
33	70		305	0.16404	373.16	373.16

error: total number of steps must not exceed 100

The third screen was built to show data in terms of original variable names and units. Particularly, it enables to keep track of power history definition. Also, if any input value is wrong, the screen allows the user to go back to previous screen and modify it.

FastDart: Validation screen. Data are shown in terms of original names and units.


FastDart: input test

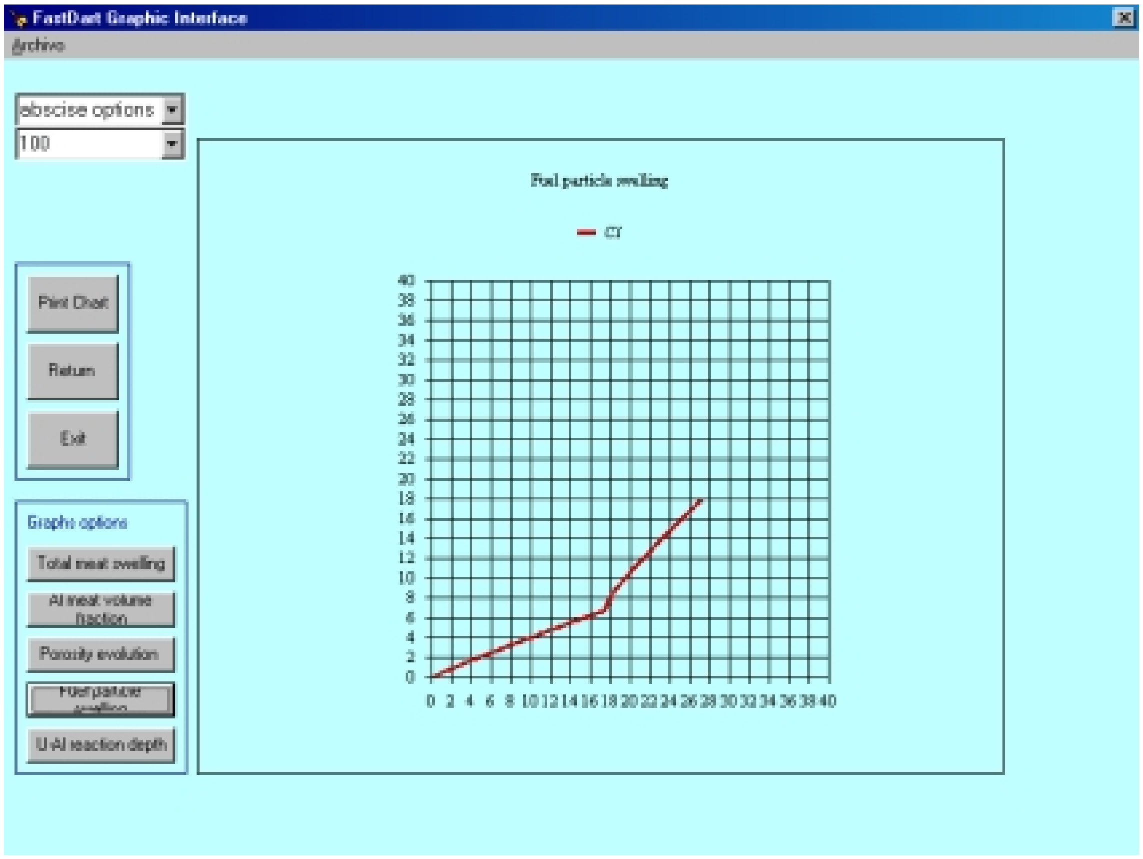
So far, you have defined these original DART variables in original units

grs	0	Select #	100
geom	0.5	delt(#)	259200
prsol	14.2262403528	pow(#)	0.05
vfm0	0.5	T(sf)	373.16
vpm0	0.129195	T(cl)	373.16
tmtp	373		
cvf0	0.67		
poros	0.0013		
rs fuel	0.15		
rs particle	0.004		
grsk	0.004		

3. **Dynamic Library Fortran routine conversion**

The new FastDart FORTRAN routine depicted in the first section was compiled as a Dynamic Library Link. This is the way to include FORTRAN calculation subroutine into Visual interfaces. The defined data through first and second screen are passed to the DLL as entry arrays. The calculation made by the DLL is returned as an export common block, ready to be used by the final screen (Chart Facility). As a result of each run, two files are generated: fastdart.txt and fastdart.csv. The first one collects all data generated during calculation for each time step and fuel particle radial partition. The second one collects the data to be plotted.

4. Plotting facility



The plotting screen allows the user to select the abscise array (fission density, full power days or burnup) and to plot total meat and fuel particle swelling, Al (meat) volume fraction, the porosity evolution and the U-Al reaction depth. Also it can be defined titles, axis labels, etc.

5. Conclusion

This new DART code version for silicide and U-Mo dispersed fuels enables the user to have a short runtime calculation code and friendly interfaces to input data and to plot DART outcome variables. The beta version will be distributed via e-mail (when available) or in floppy units to MTR fuel researchers and developers.