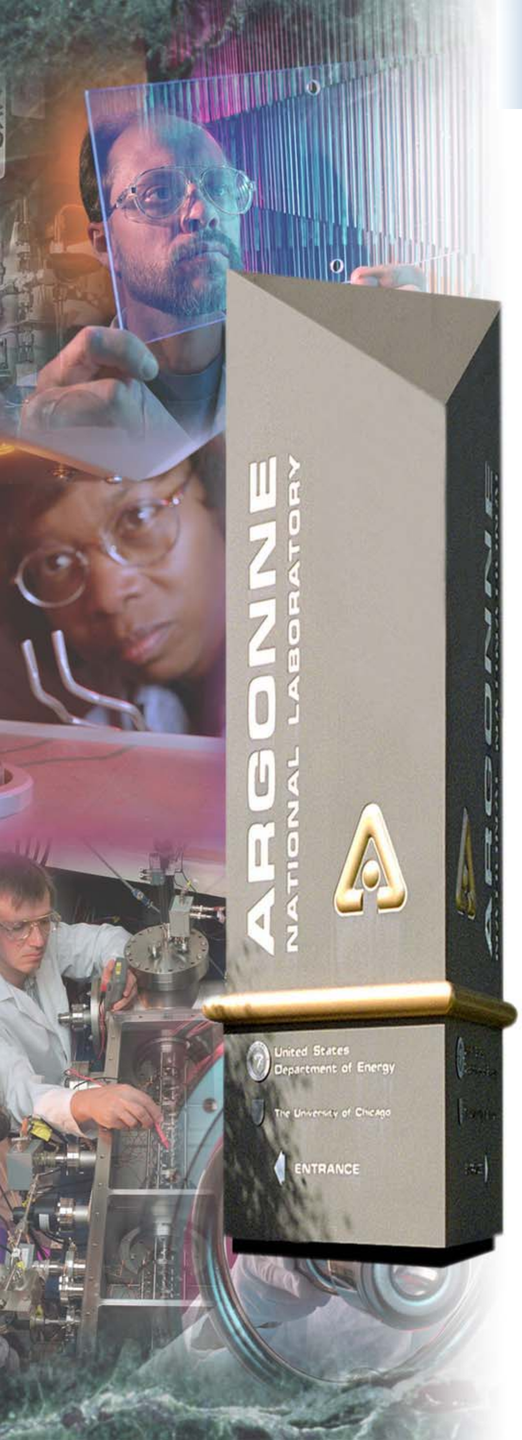
ARGONNE NATIONAL LABORATORY



The Glass Furnace Model Technology Transfer (GFM-TT) Program

Presenters: Steven Lottes and Michael Petrick

THE UNIVERSITY OF CHICAGO

Office of Science
U.S. DEPARTMENT OF ENERGY

Argonne National Laboratory is managed by
The University of Chicago for the U.S. Department of Energy

ARGONNE NATIONAL LABORATORY

Program Overview and Status

GFM-Technical Transfer Objective

- Disseminate and promote widespread use of the GFM Code throughout the Glass Industry

The GFM-TT Objective is Being Achieved By

- Widely disseminating information on GFM Code to Industry
  - Collaborating with GMIC
- Making GFM code readily available to interested users at no cost during free trial period
- Providing technical support for the GFM Code users during the free trial period
- Promoting development and implementation of long term technical support mechanisms for the GFM Code user

Chronology of Factors/Issues Impacting Technology Transfer Progress

- May 2003 – S. L. Chang, principal code architect/developer dies suddenly
  - B. Golchert assumes his responsibilities
- March 2004 – Formal technology transfer program initiated
  - Substantive number of no cost trial licenses requested and processed
- May. 2004 – GFM Development Program ends before beta testing of code completed
  - Industry program participants primarily responsible for beta testing
    - *Substantive preprocessor robustness issues surfaced*
- March 2005 – B. Golchert leaves ANL
  - Steve Lottes, a seasoned expert code developer assumes responsibility for completing program
    - *During subsequent six month period identifies other robustness issues that needed to be addressed*

GFM Technology Transfer – Accomplishments to Date

- Dissemination of information on GFM Code to industry
  - Brochure prepared and mailed to over 90 companies
- Eighteen (18) no cost trial licenses issued to twelve (12) different companies
  - Staff from eight companies have received various levels of training on how to use the code
- A website has been created and made operational to facilitate GFM technology transfer
- Robustness of GUI program greatly improved
- Remaining code robustness problems vigorously being pursued

Initial Steps Taken to Assume Tech Transfer and Support Responsibilities for the Glass Furnace Model (GFM)

- Sent out announcement of change of GFM support contact to all 18 trial license holders asking for feedback & notification of any problems
- At the same time began a review and assessment of GFM status & capabilities
  - Installation procedure
  - Graphical user interface (GUI) program
  - Combustion and Radiation Code
  - Melt Code
  - Coupling between components
- Identified four areas where enhancements were needed or would make the GFM software much more attractive to potential users

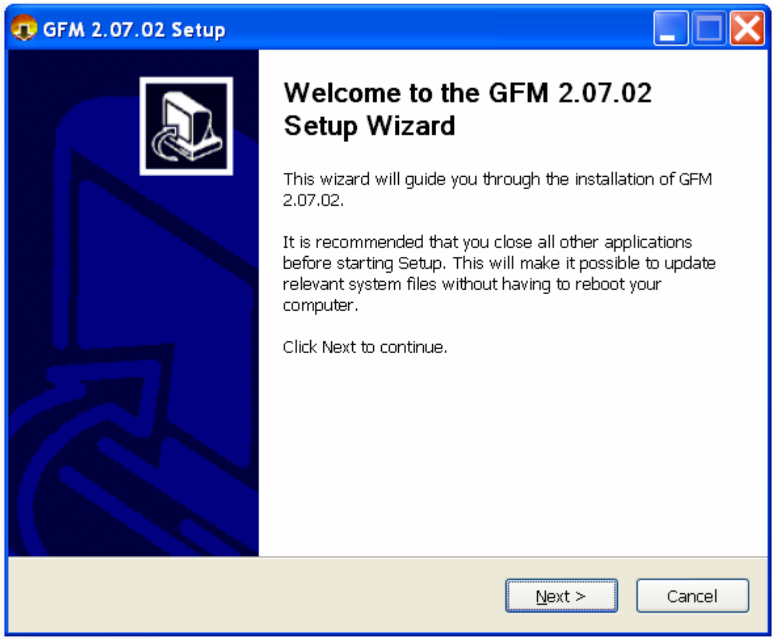
Technology Transfer Issues Addressed on the Development Side

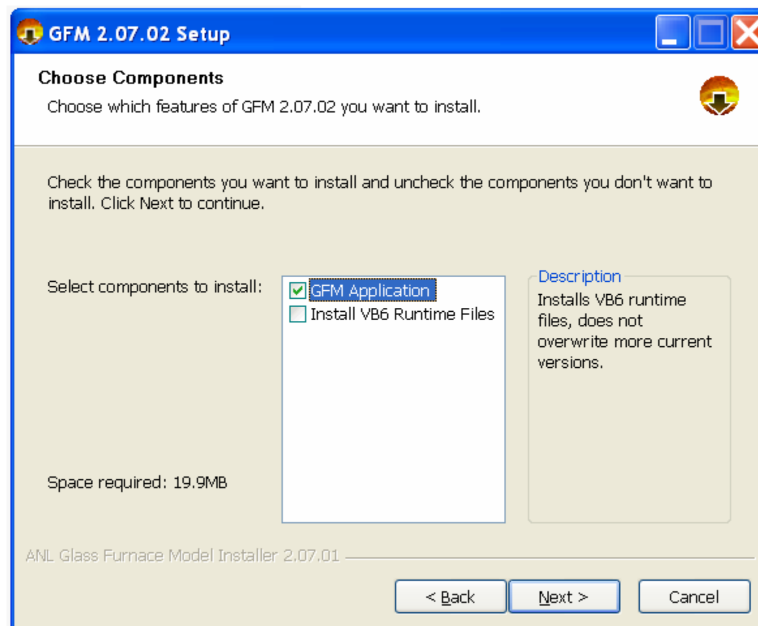
- Ease of installation
  - Modern install program
  - Include everything needed for operation (e.g. VB runtime files)
- Ease of use
  - Task oriented simulation setup and run procedures with more automation
  - User input error trapping & restructuring to prevent errors
  - Recognition of possible loss of data conditions with warnings
- Major new automated running capabilities
  - Automatic cycling for both non-regenerative and regenerative furnaces between combustion and melt spaces
- Robustness in melt, combustion, and radiation computations
  - Many issues addressed in melt and combustion codes
  - Definition: Ability of the program to generate physically reasonable results on problems that are significantly different from the ones used for validation.

Modern Installer Written for GFM

New install uses a standard installer with option to install VB runtime files.

Provides start menu item, desktop icon, populated directory structure, uninstall program, etc.



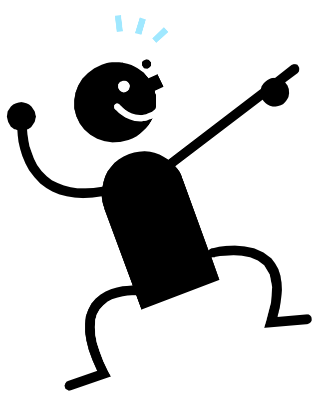


Old install procedure was manual and did not provide VB runtime files needed on some systems.

ARGONNE NATIONAL LABORATORY

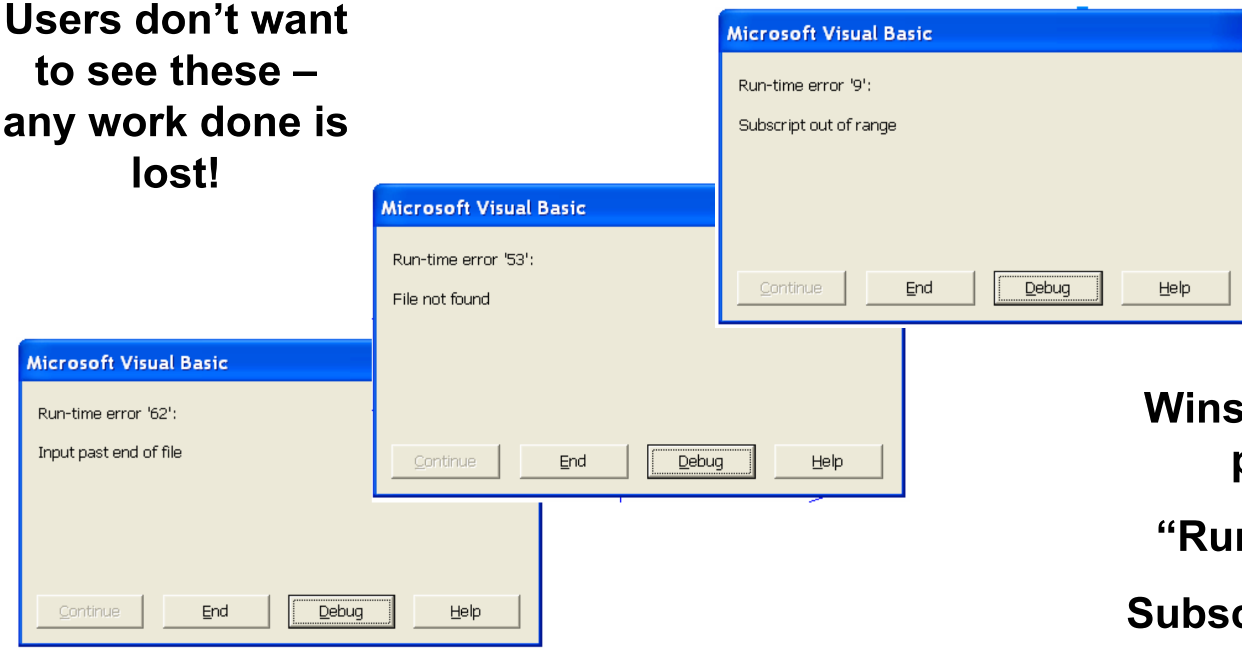
The old user interface program did not protect itself against incorrect use and input errors

The user interface obviously had to be upgraded



Runtime Errors Are Easily Generated in the Old GUI Program when Users Make Mistakes

Users don't want to see these – any work done is lost!



Wins award for most pesky error:

"Runtime error # 9"

Subscript out of range

What do you do now?

Email or Call Tech Support at ANL!

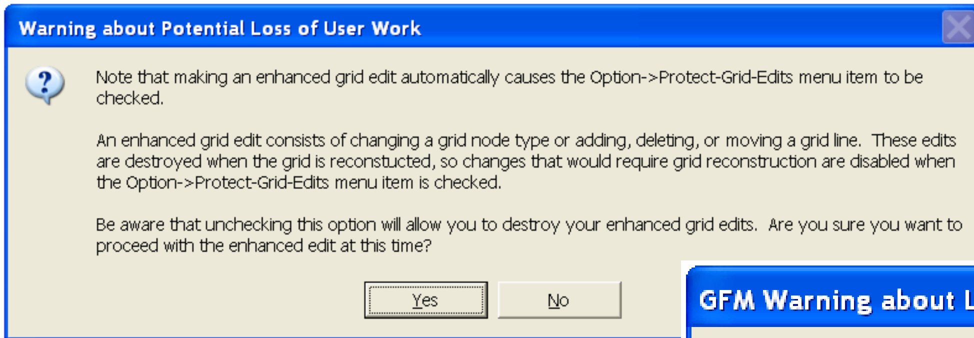
Strategy Used to Make the GUI More Usable

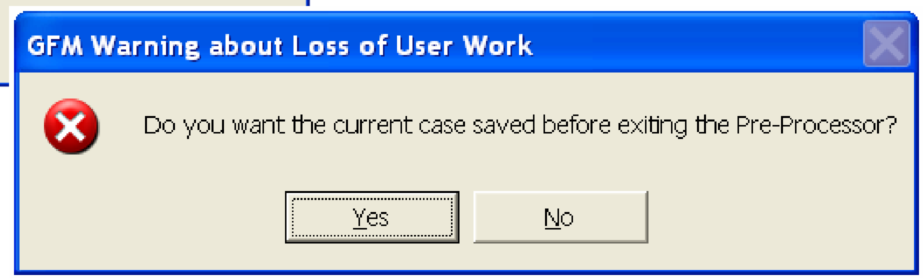
- Defined & implemented program state variables that allow correct system responses to user actions (you can't save a file when none are open)
  - State variables indicate where the program is in the task sequence
    - *E.g. (just added a burner, running the melt code, etc.)*
  - Program actions triggered by mouse & keyboard input are now limited to valid actions

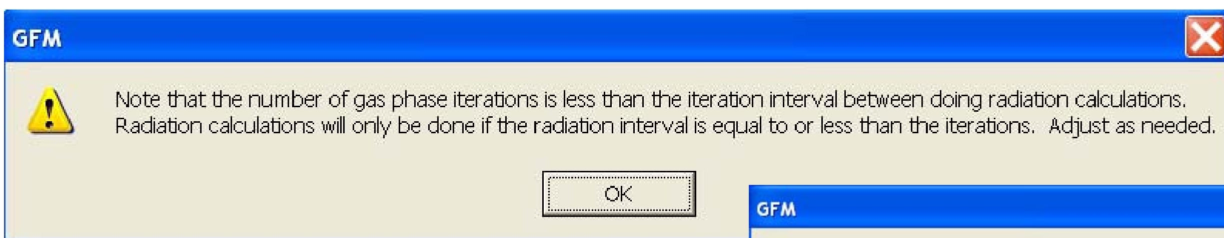
- Menu system redesigned to eliminate errors
  - Separated into tasks to setup, run, and review a simulation
  - More meaningful menu item labels
  - Menu choices are now grayed out or invisible when inappropriate

- A brief work flow document has been prepared that explains how to use the new GUI
  - The menu system is now much more intuitive and relies less on users following instructions in the user manual.

Warning Messages Improve Usability by Helping the User Avoid Mistakes

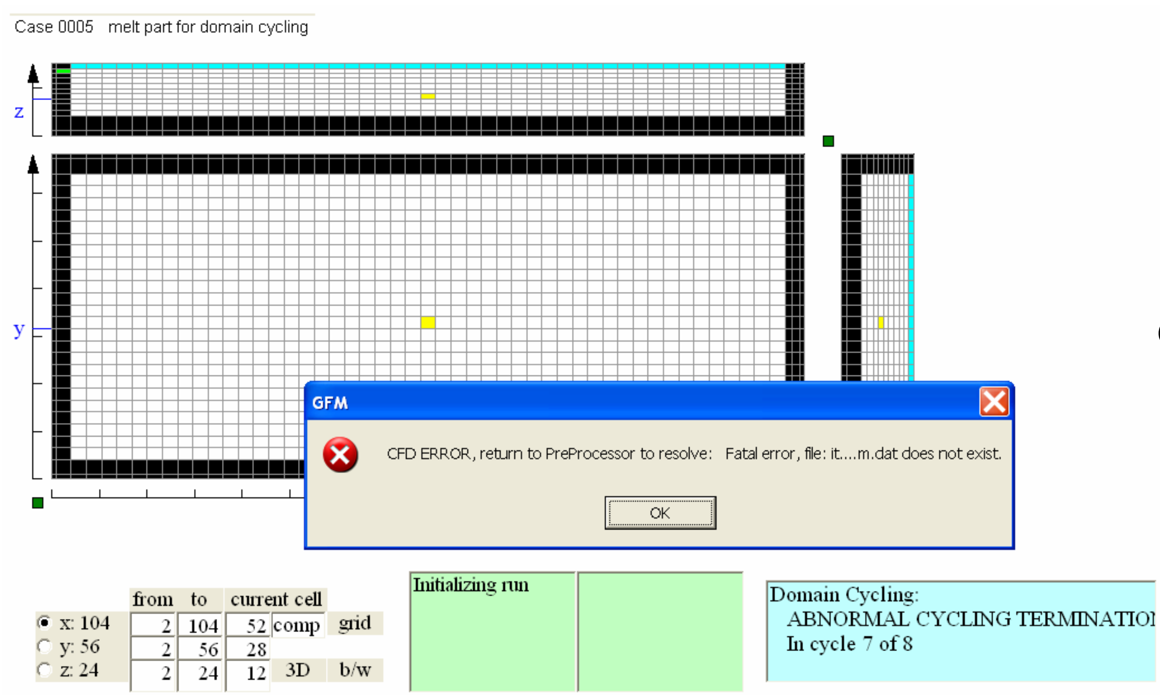






Well informed users apply software more effectively & avoid problems that cost time.

Errors Encountered in Running the CFD Codes Are Now Communicated Back to User Interface

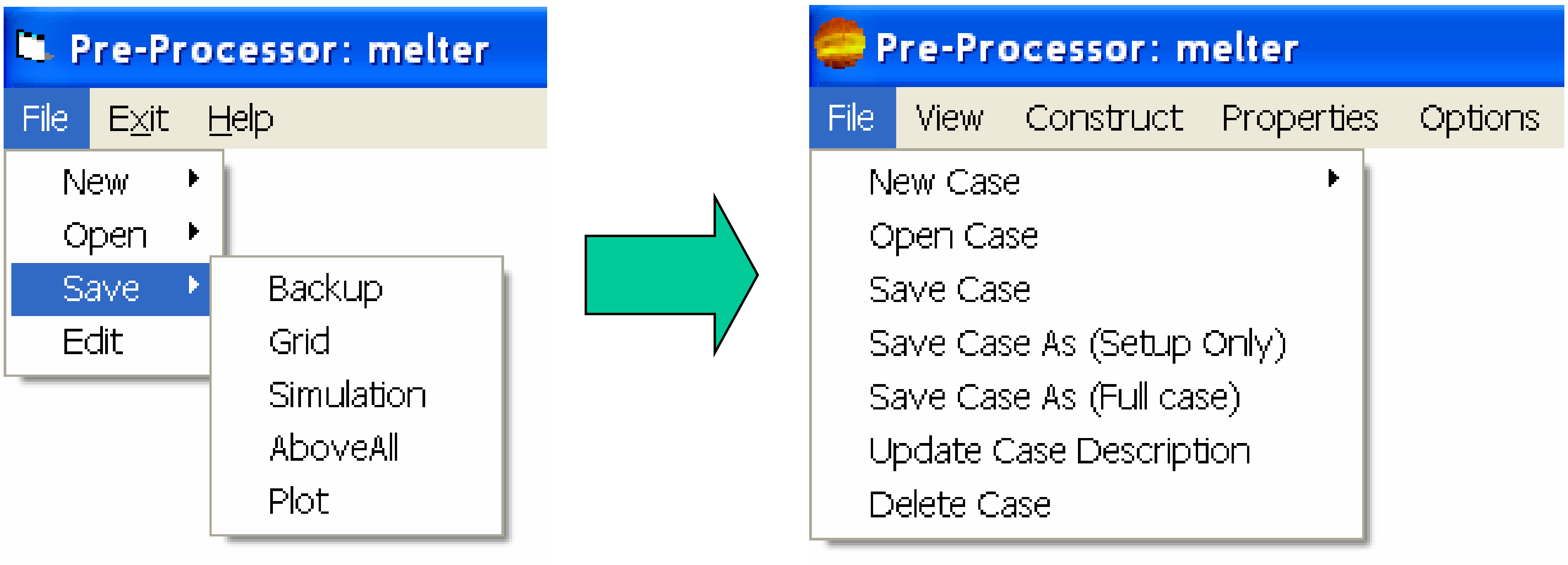


If the command window CFD task disappears from the task bar before normal end---the CFD run died. If a known cause can be recognized it is passed back to the GUI and the reason for abnormal termination is displayed in a message.

File Management has been Automated

All files for a simulation are now handled together as a "Case."

Users no longer need to know about files or their inter-dependencies.



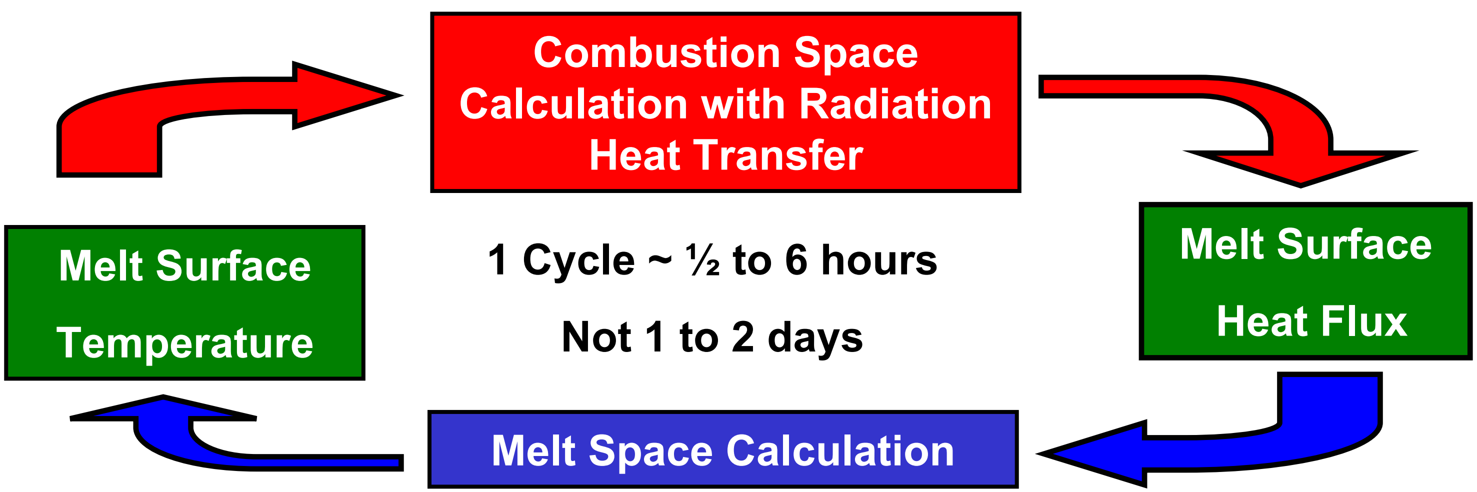
Geometry data, material properties, simulation parameters, & all other data that defines the glass furnace problem to be solved are automatically saved in the appropriate files. Changes that affect multiple files are handled properly.

Ease of Use –Summary of Major GUI Program Improvements

- Moved from **File** based to **Case** based setup, control, results review, and data management.
  - Relieves user of burden of needing to know details of file content and dependencies between files
  - GUI program does all file management for user (automatically loads and saves files as needed, etc.)
- Extensive error trapping and recognition of loss of data conditions implemented in the preprocessor GUI
  - All known conditions that caused system errors eliminated or trapped with cause of problem message issued to user
  - Actions that could cause loss of user data results in user warning message before proceeding
- Several hundred fixes and enhancements
  - Allow user to set some model parameters
  - Allow more user control over “difficult” cases
    - *Option to scale heat flux to melt to amount needed*
  - Many others

Major New Capability: Automatic Cycling Between Spaces

- Automatic cycling between combustion and melt spaces for both plain and regenerative furnaces
- 8 or more cycles may be needed – difficult to do manually
- Auto cycling reduces time for simulation from *~2 weeks to ~2 days*
- Auto cycling tightly couples combustion and melt spaces
  - *Radiation heat flux is cycled into the melt computation as soon as it is computed*



Furnace Schematic for First "Successful" Automatic Cycling Simulation

Combustion Space

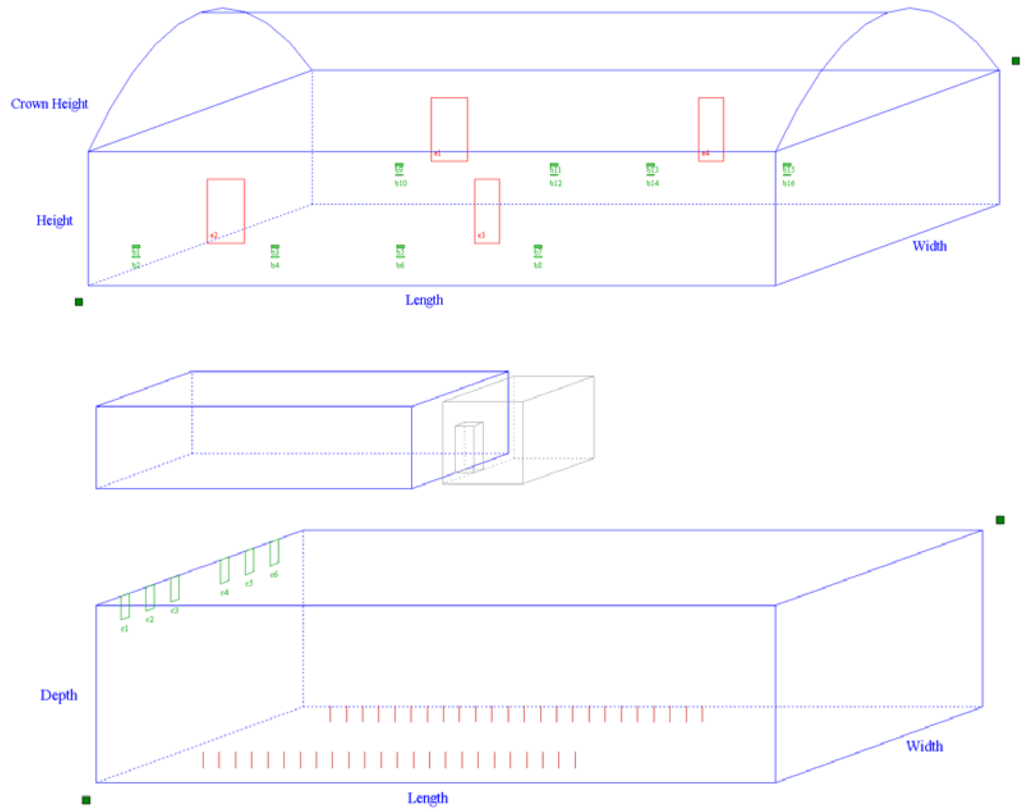
8 burners (green)

4 exhausts (red)

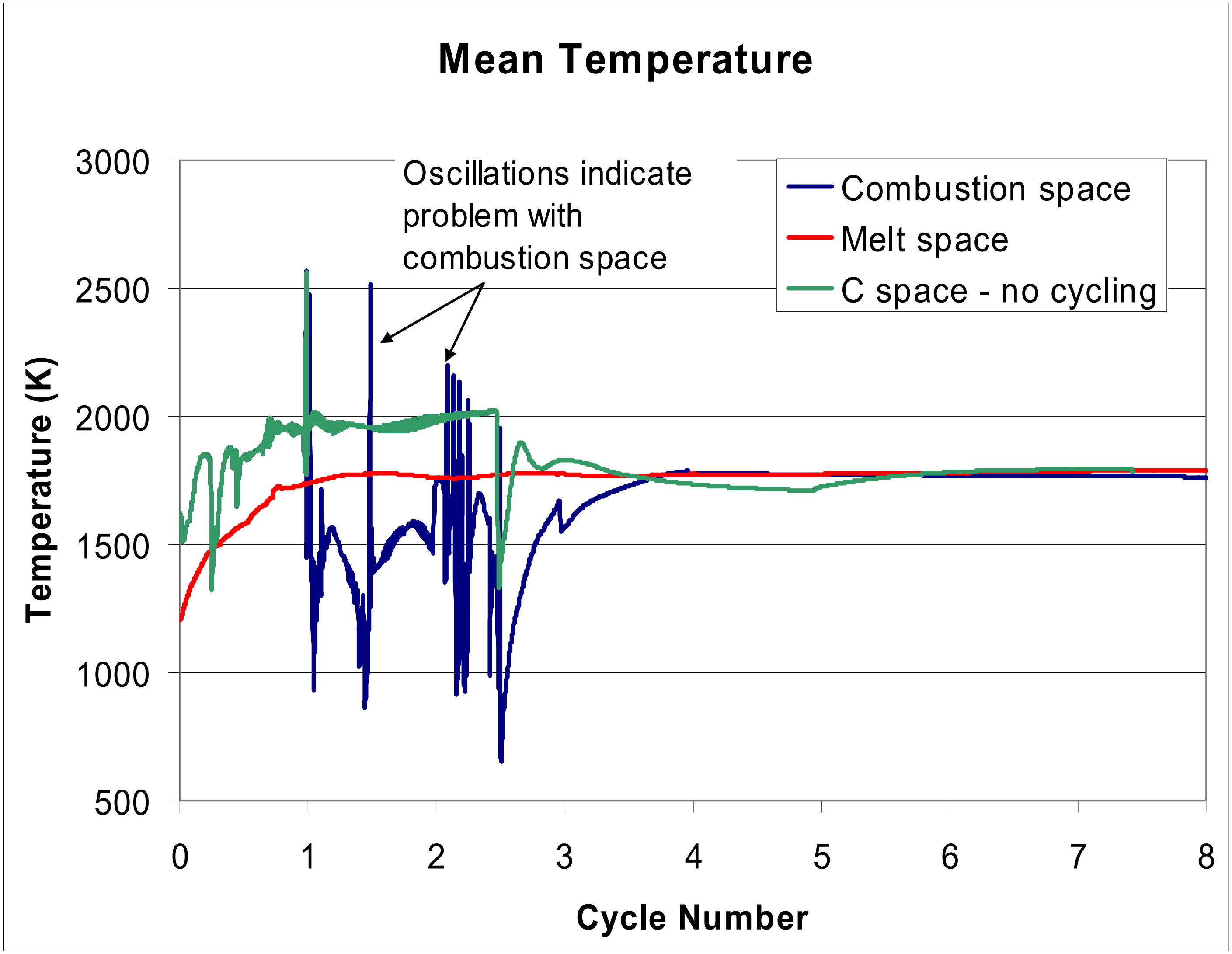
Melt Space

6 chargers (green)

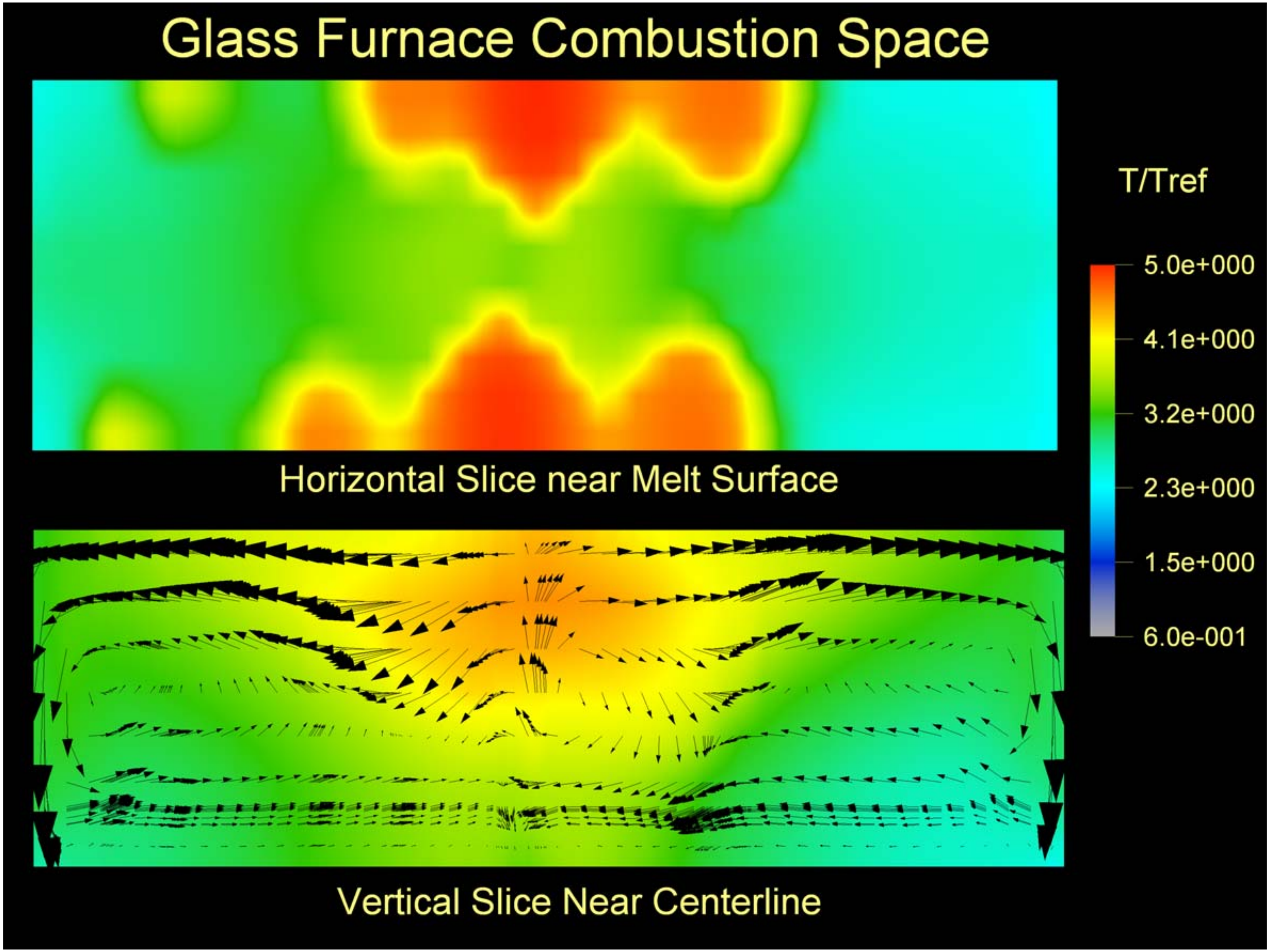
Many electric boosters (red)



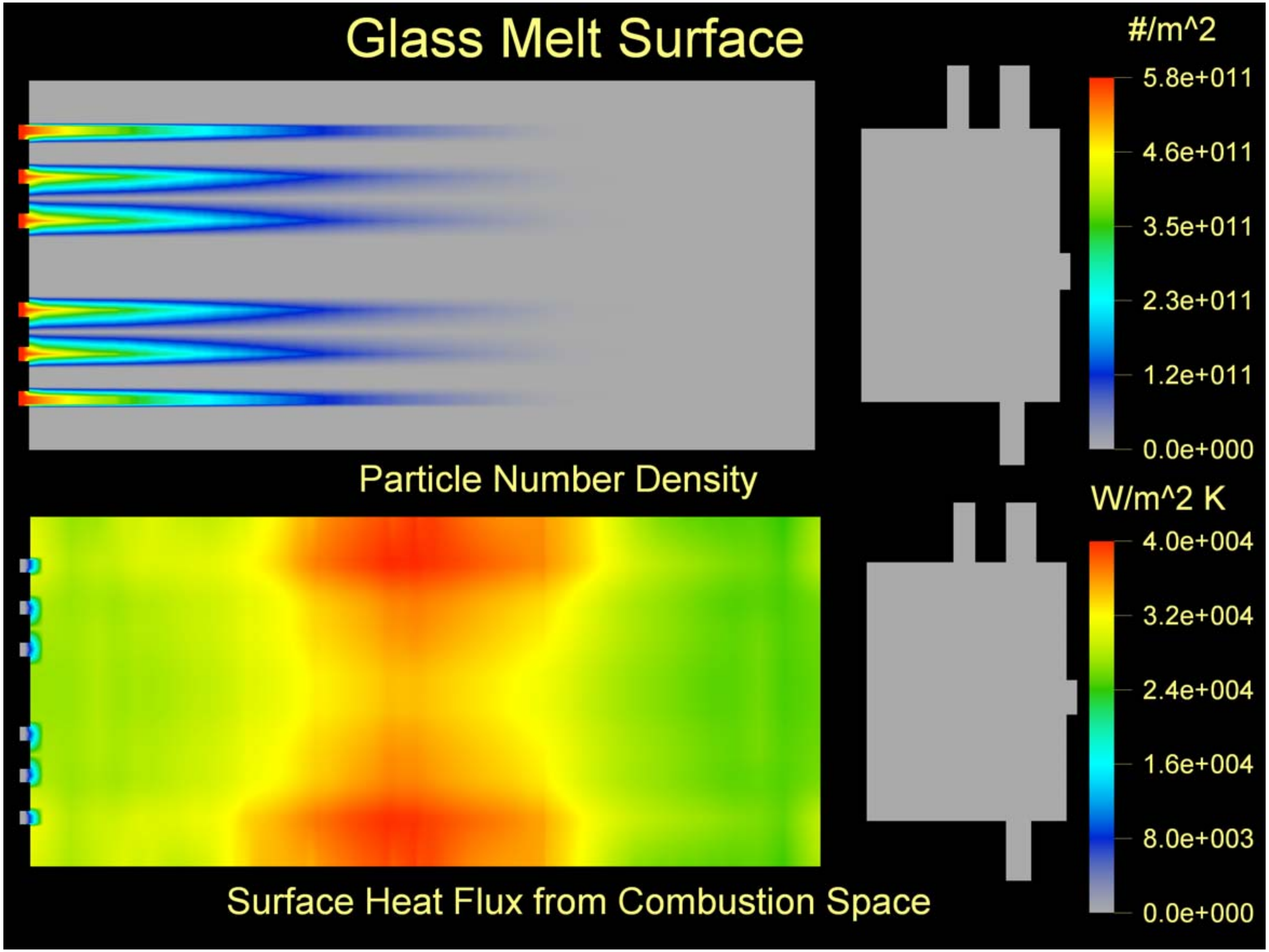
Results from First "Successful" Automatic Cycling Run Reveal Energy Solve Step Robustness Issue



Conditions in Combustion Domain



Conditions in Melt Domain



New Progress Monitoring Program Added to GFM Package, Visually Monitor Computation Progress with RunPlot

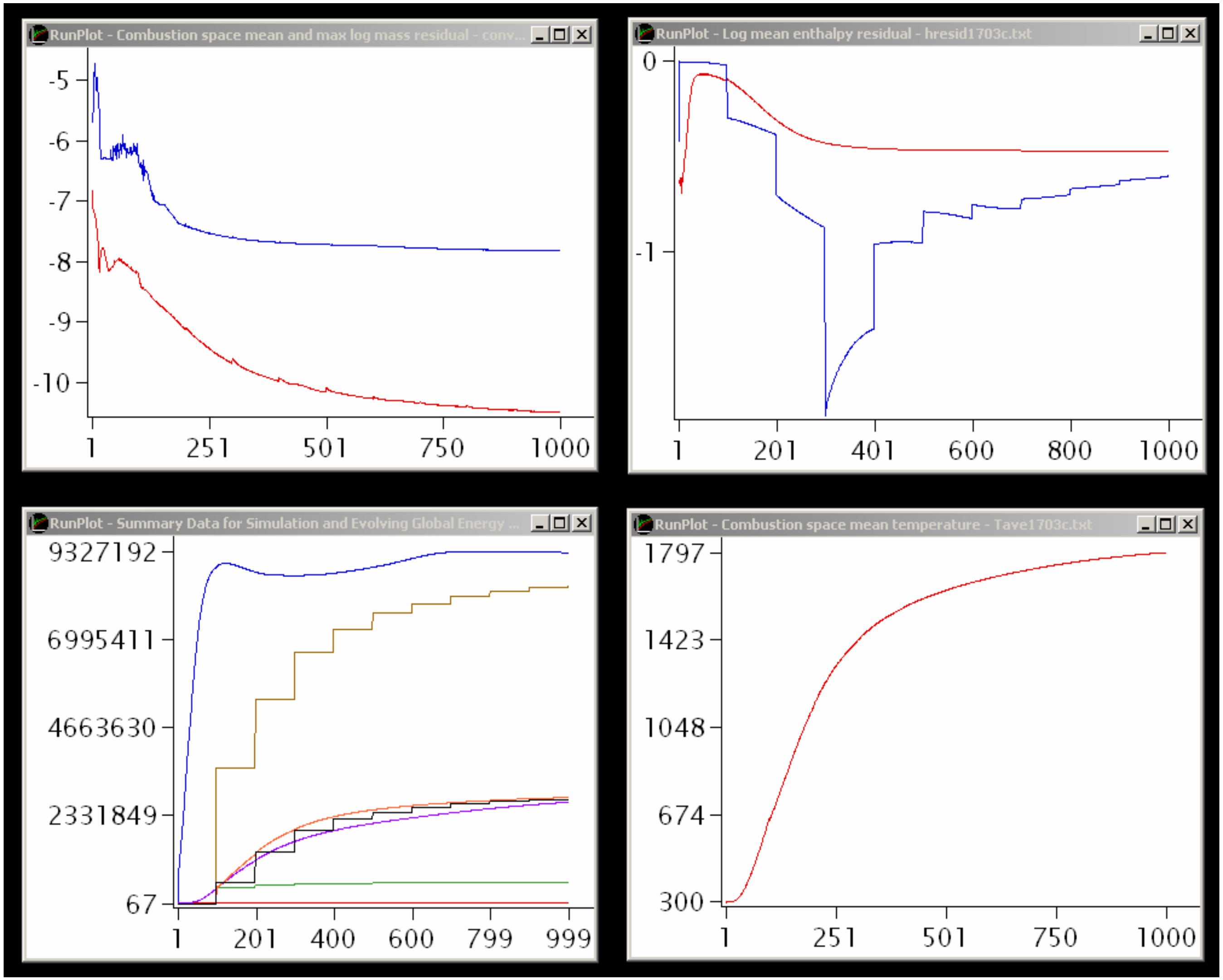
- GFM has been instrumented to output various state variables
- RunPlot will plot them in a window and update it automatically

- Easily spot problems developing in a computation
- Problems may be real problems in one of the codes or they may be problems in simulation setup (parameter values, etc.)
- Plot & monitor during computation or plot after computation
- Variables that can be plotted:
  - Average temperature
  - Mass, energy, & other residuals
  - Combustion energy release and distribution (to melt, to exhausts, and loss through walls)
  - Energy into melt and distribution to solids, liquid, wall loss, etc.

Sample Monitoring Plots – The energy residuals show a potential problem in the energy solvers

Mass residuals

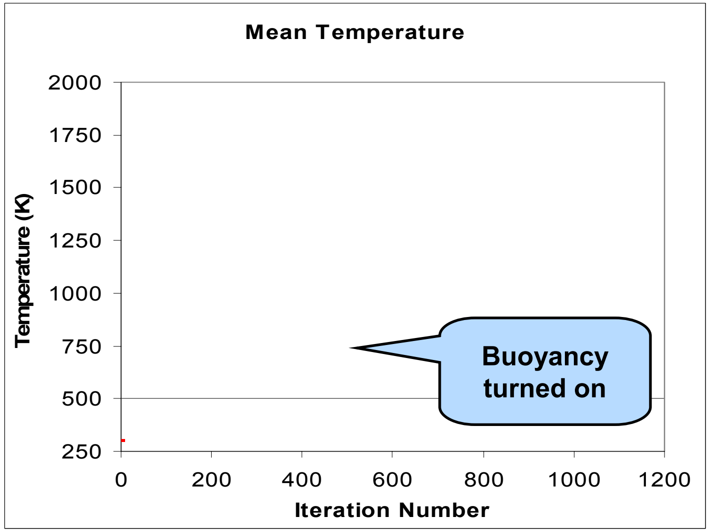
Energy residuals

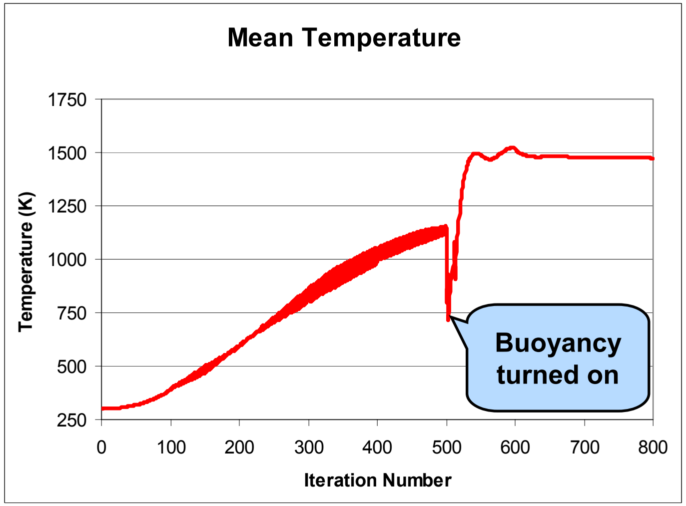


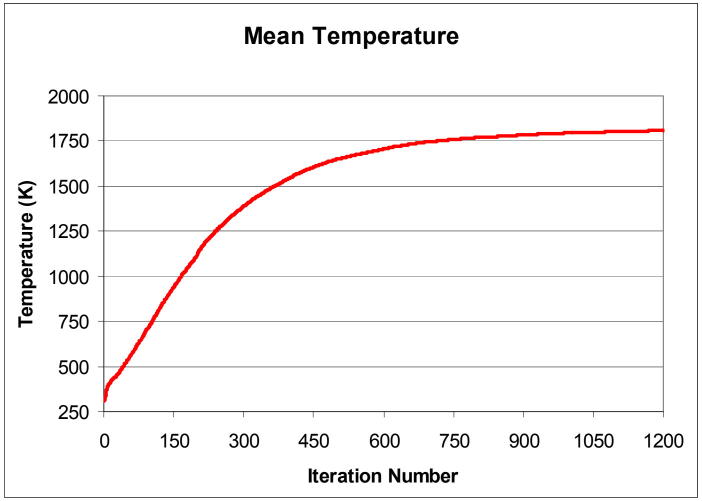
Energy release and distribution

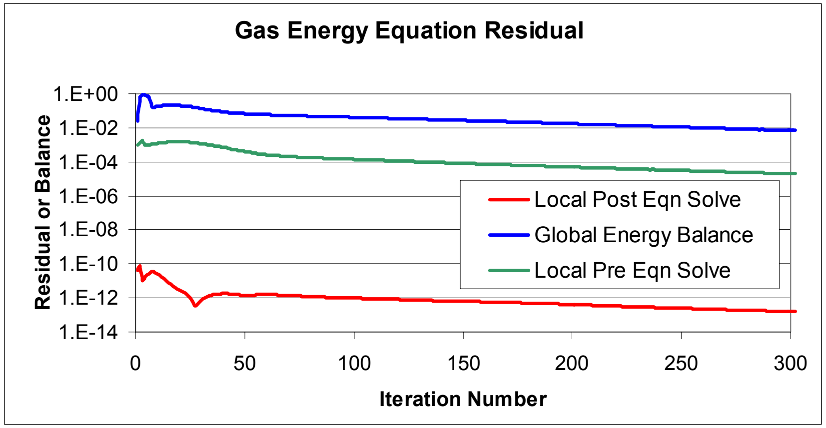
Combustion space mean temperature

Improving Convergence In Gas Phase Energy Equation, Mean Temperature Signatures & Energy Convergence

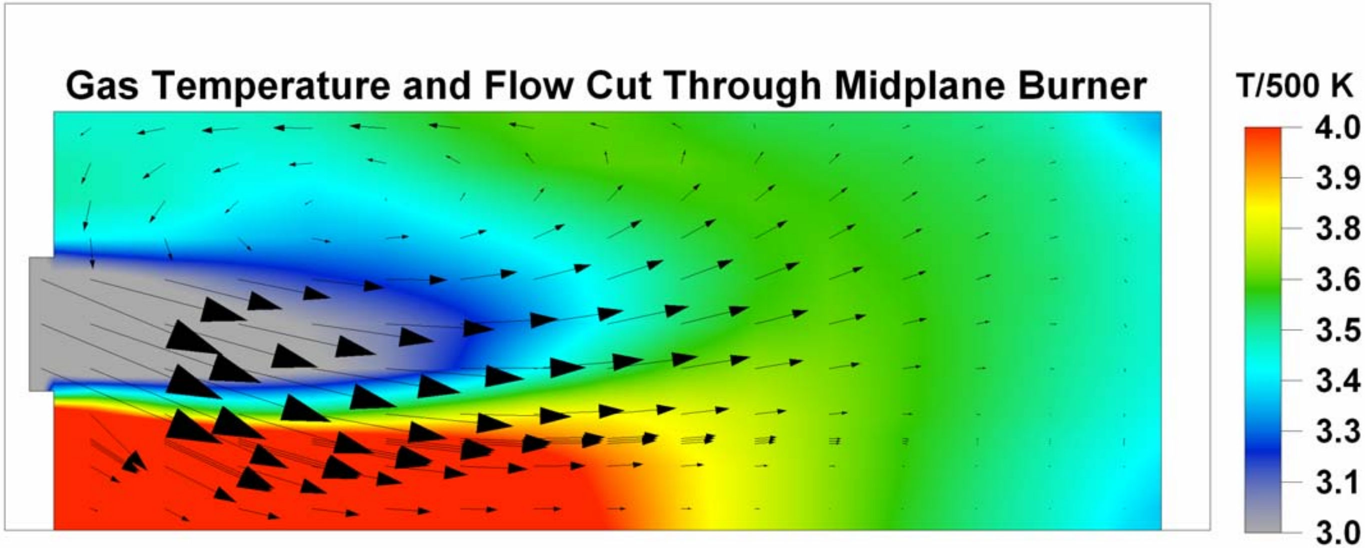


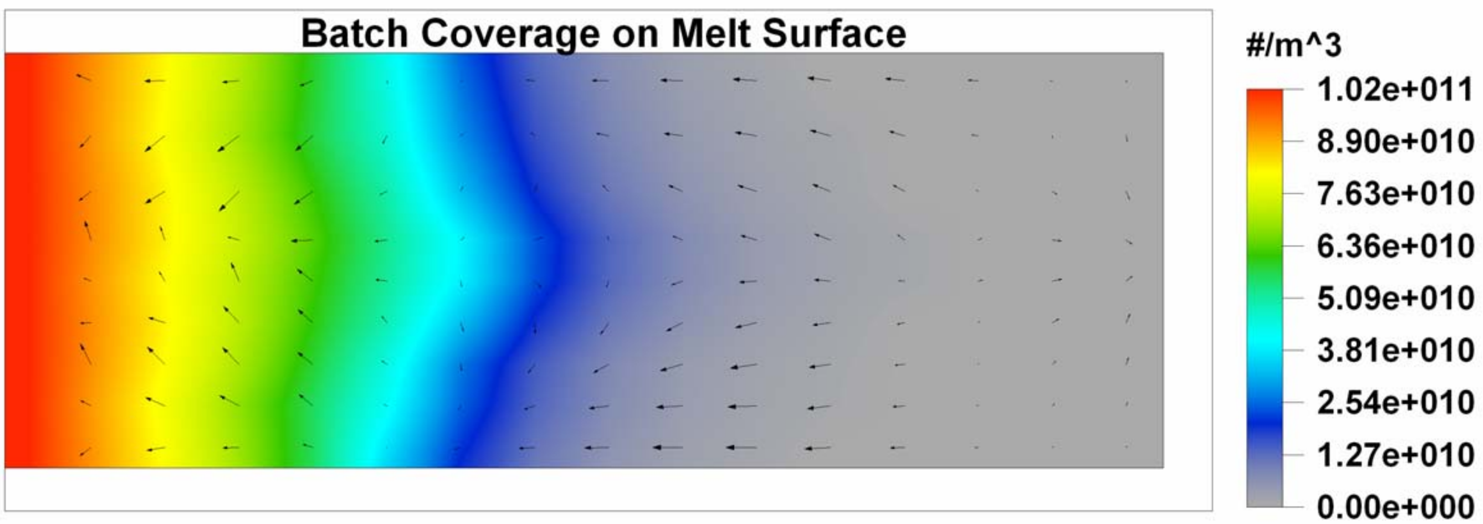






Preliminary TC21 Problem Results, Very Coarse Grid, Eight Cycles through Left & Right Burner Combustion Spaces with Averaged Heat Flux into Melt Space





More User Control is Provided Over Solver Parameters and the Solution Procedure

- New solver parameters that can be set
  - Initial temperature field
  - Initial melt surface temperature
  - Radiation calculation interval
  - Auto cycling interval between combustion and melt spaces
  - Oxy-fuel or air soot kinetic parameters
  - Others in new documentation
- New solution procedure options
  - Automatic file moves in manual cycling between spaces
  - GUI control program automatically cycles between spaces
  - Optional scaling of calculated heat flux distribution to amount needed for specified melt exit temperature
    - *Use when testing burner position or firing rate to optimize efficiency & required fuel flow rate is not known*
    - *Use when material properties have significant uncertainty, etc.*

Summary of GFM Technology Transfer Status

- Accomplishments
  - Modern installer program developed for GFM
  - Major redesign of the user interface to enhance ease of use
  - Automatic cycling between combustion and melt spaces
  - Many robustness improvements in combustion and melt CFD codes
- Next Steps
  - Resolve remaining code robustness issues and release the next upgrade of GFM to trial licensees
  - Select and provide technical support to a small number of code licensees interested in using GFM to analyze a furnace problem
    - *Demonstrates value of code*
    - *Continues beta testing of GFM*
  - Promote Industry participation in a Core-User-Group on ANL's GFM web site to facilitate further development and support of GFM for industry use

Additional Material on Robustness – Not Part of Presentation

Robustness Issues – How Good is Good Enough?

- The answer depends on how the software is used
  - Useful results for large scale design, optimization, and trouble shooting may be obtained if flow patterns and trends are approximately correct
  - Example: positioning burners or adjusting firing rates and angles to avoid short circuiting flow of hot combustion products to exhausts
  - Example: detailed burner design is not within the scope of GFM
- Reasonably good visualization does not require high accuracy
  - A visualization of what is most likely going on inside a furnace may be enough to reveal many possibilities for improvement to someone with considerable experience in furnace design or operation
- Quantitative measures of quality and performance in simulated systems often require tight convergence and fairly high accuracy.
- Increasing robustness and accuracy is always a high priority

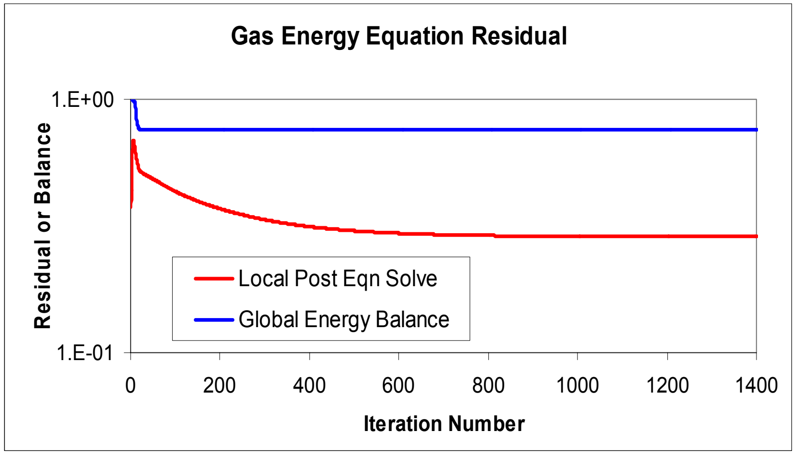
Resolving Robustness & Performance Issues, Combustion Space Energy Equation Example

- Instrumenting a CFD code to collect data on its own performance is necessary to improve robustness
  - Some of this data collection existed in GFM
  - Much more has been added
- Just as visualization of CFD results helps users “see” how well processes in a furnace are doing, ***visualization of CFD monitoring*** data helps developers and technical support staff “see” how well the computation is going, identify problems, and improve robustness
- Improving the combustion energy solver is given as an example
- Many similar successful efforts have been carried out in melt space solvers
- Evaluation and improvement of the radiation heat transfer solver is underway

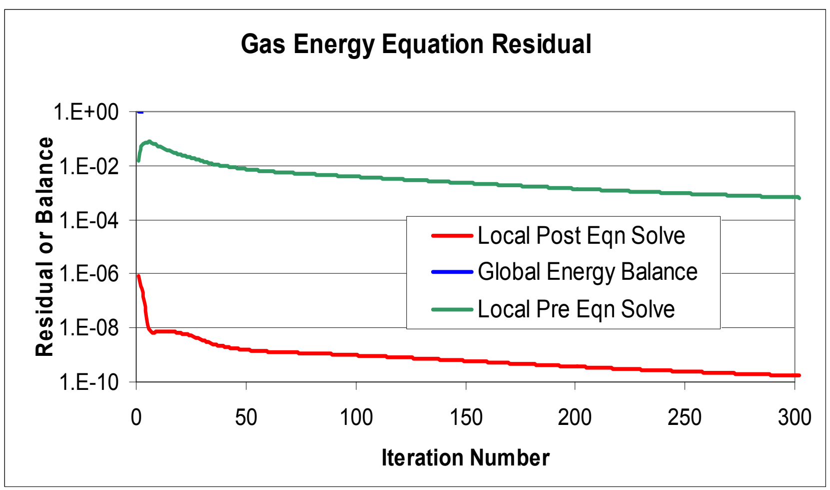
Combustion Space Energy Equation Solver, Steps Taken to Improve Performance

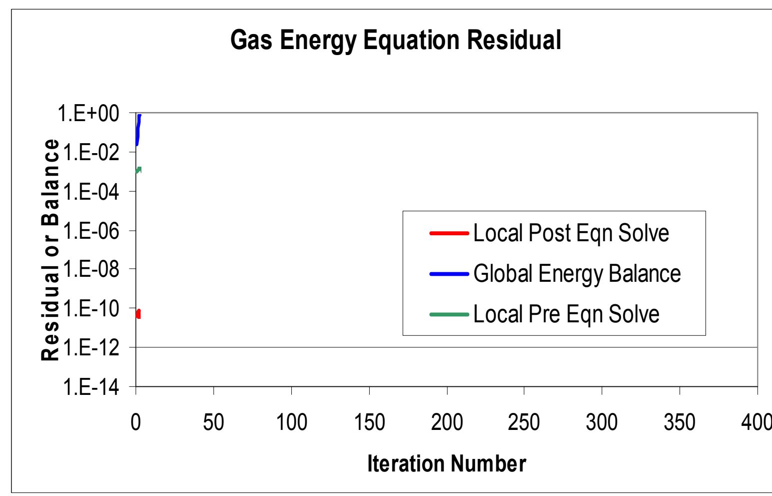
- Carefully checked non-dimensional source terms
  - Discovered radiation emission volume sink before first absorption calculation & eliminated before absorption source available
  - Discovered buoyancy turned off before iteration 500 & enabled buoyancy for entire run
- Changed initial species distribution to that of reaction product distribution rather than air (accounting for incoming excess air)
- Optimized energy equation relaxation factor
- Implemented generalized minimal residual (GMRES) algorithm in core solver to drive energy residuals down to near machine precision
- These changes produce a near best case energy equation solve step
- Some difficulties in radiation calculation still affect global balance

Combustion Space Energy Equation Solver Performance



- Worst case:
  - Global balance off by more than 70%
  - Local residuals < 1 digit of accuracy
- Improved:
  - Global decreasing, no longer flat
  - Local is good
- Improved more:
  - Global good to a couple of digits
  - Local converged to machine precision





Improving Convergence In Gas Phase Energy Equation, Mean Temperature Signatures