Available for licensing

# CartaBlanca

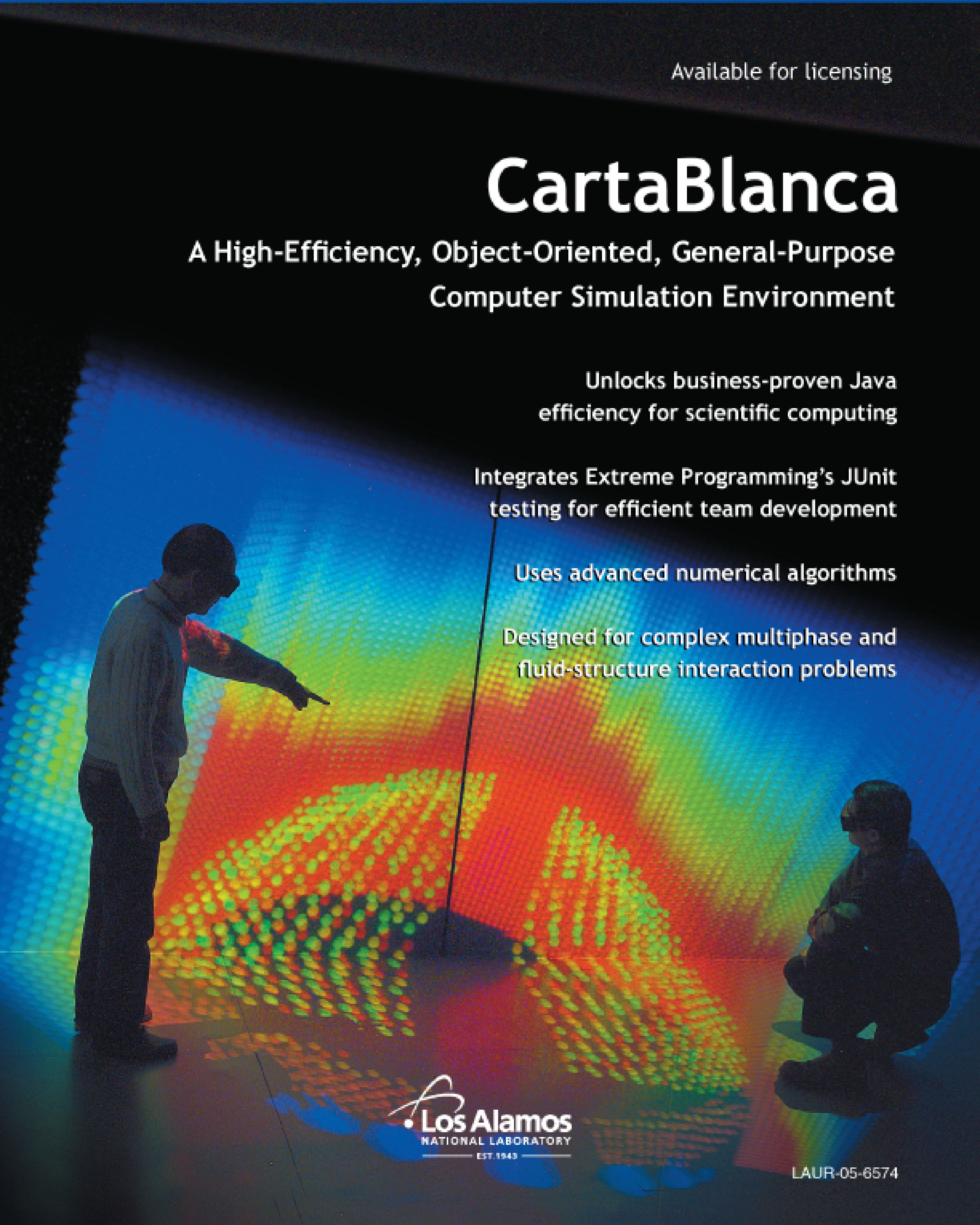
## A High-Efficiency, Object-Oriented, General-Purpose Computer Simulation Environment

Unlocks business-proven Java efficiency for scientific computing

Integrates Extreme Programming's JUnit testing for efficient team development

Uses advanced numerical algorithms

Designed for complex multiphase and fluid-structure interaction problems



Los Alamos
NATIONAL LABORATORY
EST. 1943

LAUR-05-6574

# CartaBlanca: A High-Efficiency, Object-Oriented, General-Purpose Computer Simulation Environment

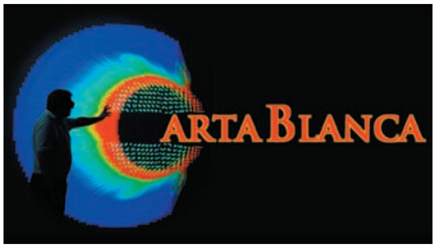
Technology Transfer Contacts:
Eric Canuteson
Charlie Gibson
Laura Barber
*email: tmt-2@lanl.gov*

Technical Contact
Brian VanderHeyden

*ABOUT THE COVER*

*CartaBlanca, the first Java-based simulation software package, can be used to simulate explosions of different kinds. Shown is a simulation of an exploding cylindrical blast container; these containers are used to dispose of an improvised explosive device (IED). The simulation shows the motion of the broken IED hemispherical case, blast wave through the gas (air), and deflection and deformation of the cylinder. The colors show the different local particle pressures. These results are being visualized at Los Alamos National Laboratory's RAVE (Reconfigurable Advanced Visualization Environment) facility, where scientists can see the data in a three-dimensional mode that provides greater insight into the details of the simulation.*

Los Alamos
NATIONAL LABORATORY
EST. 1943




# Executive Summary

## CartaBlanca: A High-Efficiency, Object-Oriented, General-Purpose Computer Simulation Environment

### Features

CartaBlanca brings the tremendous efficiency of the Java programming language to the world of scientific computing. The first of its kind, CartaBlanca is a state-of-the-art, object-oriented simulation software package poised to offer next-generation modeling and simulation capabilities to scientists in a wide-ranging number of disciplines. Written in the "developer friendly" Java language, it enables computer code developers to simulate complex nonlinear effects such as airflow through a turbo booster, blast effects on buildings, or heat transfer along a semiconductor, to name but a few of its many applications. Because CartaBlanca is a Java-based software package, the code is much easier to use, manipulate, and modify than are codes based on such programming languages as FORTRAN or C++. CartaBlanca takes advantage of the improved execution speed offered by the HotSpot™ compiler, which allows performance on par with FORTRAN and C++. CartaBlanca opens up the field of physical modeling to a much broader set of programmers because Java is one of the most common and efficient business software languages. CartaBlanca is modular and allows for rapid software application or simulation code prototyping; strong, extensive compiler checking; plug-and-play module insertion for modeling physical systems; solutions with consistent results; and integrated unit and regression testing.

### Applications

- Aerospace engineering
- Animation and special effects
- Computational fluid dynamics
- Fluid/solid interactions
- Automotive design
- Weapon/target interactions
- Pharmaceutical processing
- Homeland defense

### Benefits

- Provides accurate, physics-based computer simulations in Java
- Provides faster and lower-cost development
- Allows for easily modified and integrated code
- Runs on most hardware platforms without modification, from single PCs and Macs to parallel-processing supercomputers
- Increases software developer productivity
- Allows state-of-the-art simulations for complex reactive flows

## Overview

Modern scientific and engineering simulation projects have become large-scale and complex endeavors. Very often, simulation projects involve the modification of existing software to produce new capabilities. These projects typically require constant and ongoing attention from a variety of code developers. In fact, on some scientific projects, code developers outnumber the actual users for a significant portion of the lifetime of the project. As such, it is useful for scientific and engineering code developers to use modern productive software languages to produce a "developer-friendly" simulation code. It is just as important for such code to be developer-friendly as it is to be user-friendly. This approach is different from the mainstream approach where user-friendly interfaces are wrapped around less-developer-friendly legacy code written in FORTRAN or C++. Developers need to collaborate with each other and simultaneously use the codes to develop large-scale simulations. CartaBlanca is built for just such a purpose.

CartaBlanca is an object-oriented, nonlinear simulation and prototyping software package whose primary function is to assist code developers in solving a wide range of hydrodynamics and fluid/structure interaction problems. Because CartaBlanca is written entirely in Java, it provides scientists and engineers with developer-friendly software to use in producing large-scale computational models. CartaBlanca allows users to solve a variety of nonlinear physics problems, including multiphase flows, interfacial flows, solidifying flows, and complex material responses. CartaBlanca makes use of the powerful, state-of-the-art Jacobian-free Newton-Krylov method to solve nonlinear equations in a flexible unstructured grid finite-volume scheme. CartaBlanca couples the particle-in-cell (PIC) method—a technique used to model discrete objects—with its multiphase flow treatment to model how fluids interact with solid materials that can undergo deformation, damage, and failure.

Because of Java's marketplace strength, a wealth of third-party off-the-shelf software components is available for use. One notable example is the JUnit testing framework that allows users to write repeatable tests and a series of typical problems. CartaBlanca uses this third-party software to allow developers to see where the "bugs" are or where their code breaks down. In fact, because CartaBlanca is Java based, many other Java-based components can be used with the software package. Examples include existing databases and computer-aided design tools.

The Java language includes a facility for spawning "threads" or processes that run simultaneously and can communicate with each other, but are controlled from the same program. CartaBlanca's software design uses this ability to enable data-parallel, shared-memory, and distributed-memory computations on a wide variety of unstructured grids with triangular, quadrilateral, tetrahedral, and hexahedral elements. This design allows CartaBlanca to handle complex geometrical shapes and mathematical domains. Java's simplicity greatly simplifies both code maintenance and modification because the same code is used for both serial and parallel calculations. Java's object inheritance allows developers to construct a hierarchy of physics-systems objects, linear and nonlinear solver objects, and material response behavior objects. This hierarchy simplifies code development while fostering software reuse. CartaBlanca's multithread nature permits easy portability to networked computers with the help of third-party distributed shared-memory systems such as JavaParty (http://www.ipd.uka.de/JavaParty).

Finally, CartaBlanca employs Java's powerful swing-graphics facility to provide an extensive, user-friendly graphical interface, or GUI, for problem specification and data input.

Details of the theoretical approaches used in the development of CartaBlanca are provided in the appendix in "CartaBlanca—A Pure-Java, Component-based Systems Simulation Tool for Coupled Nonlinear Physics on Unstructured Grids" (excerpt) and "Implementation and Performance of a Particle in Cell Code Written in Java" (excerpt). The appendix also includes letters of recommendation and examples of CartaBlanca's structure and capabilities.

*Competition*

**CFDLib**—A Los Alamos library of computer codes written in FORTRAN 77. These codes are capable of solving a wide range of computational fluid dynamics problems (http://www.lanl.gov./orgs/t/t3/codes/cfdlib.shtml). The library is available (for a fee) from the Energy Science and Technology Center (ESTSC) (phone 865-576-2606, estsc@adonis.osti.gov).

**CHAD** (Computational Hydrodynamics for Advanced Design)—Another computer code developed at Los Alamos National Laboratory. CHAD has been intensively used by the automobile industry and in defense projects. It has been written to take full advantage of parallel computers, specifically for chemically reactive flows. For information, contact Manjit Sahota (sahota@lanl.gov).

**FLUENT**—A commercial computational fluid dynamics software package developed and marketed by Fluent, Inc., of Lebanon, New Hampshire. Fluent, Inc., is the world's largest provider of computational fluid dynamics software and consulting services. The FLUENT software is written in the C++ language and employs an unstructured grid capability.

*Comparison matrix*

| Parameters | CartaBlanca | CFDLib | CHAD | FLUENT | Comments |
|---|---|---|---|---|---|
| **Written in JAVA?** | Yes | No | No | No | Java is a robust, type-safe, modern computing language. Java has been demonstrated to be a more efficient language for development when compared with conventional languages such as C++ and FORTRAN. Because of a very strong and growing marketplace presence, Java leads to a huge number of programmer tools. Java is also becoming a competitive language for both business and scientific applications. |
| **Object-Oriented?** | Yes | No | No | No | Being object-oriented makes a computer language developer-friendly and enhances programmer productivity. CFDLib, CHAD, and FLUENT use conventional procedural computer programming style and, as a result, do not benefit from the efficiency features of object-oriented Java. |
| **Built-In Thread Parallelization?** | Yes | No | No | No | CartaBlanca uses a built-in thread parallelization for shared-memory machines. Third-party programs are available to allow the execution of a Java code in a distributed memory system. In FLUENT, CHAD, and CFDLib, the use of parallel code in shared-memory or distributed-memory machines requires heavy, separate reprogramming and maintenance of both serial and parallel codes. |
| **Allows Easy Code Testing?** | Yes | No | No | No | Because of the wealth of third-party software packages available, off-the-shelf testing facilities such as JUnit can be used with CartaBlanca in compiling and testing their code. |
| **Graphic User Interface (GUI)?** | Yes | No | No | Yes | A GUI can be written easily using Java's swing library. This ability makes the program very user-friendly, eliminating the need for text-based inputs except for the mesh files. |
| **Reflection?** | Yes | No | No | No | Reflection, a feature unique to the Java language, allows information about classes and data structures to be retrieved and manipulated by name within an executing Java program. Objects can be queried as needed for program information. The object can also be used to save the information settings to the directory, and it also allows for drop-in components. |
| **Extreme Portability?** | Yes | No | No | No | A Java code is portable even without the source codes. Codes written in other languages must always be recompiled and, frequently, have the source modified in moving from one machine to another. |
| **Multiphase Flow?** | Yes | Yes | No | Yes | CartaBlanca incorporates true multiphase flow, which involves the integration of multiple momentum equations for each phase in the multiphase flow and allows slippage between the phases. One simple example of a multiphase flow would be air bubbles in water. Multiphase flow simulation is a difficult technology but one with many applications in industry and homeland defense. |
| **Uses Jacobian-free Newton-Krylov?** | Yes | No | No | No | CartaBlanca uses a Jacobian-free Newton-Krylov solver, a mathematical technique that provides robust, fully coupled solutions for nonlinear systems of equations. |
| **Allows Particle/Fluid Interactions?** | Yes | Yes | No | No | CartaBlanca implements particle/fluid interactions using the PIC method and the material point method (MPM). This is the most advanced feature available in CartaBlanca. By using the PIC method in combination with MPM, CartaBlanca, like CFDLib, can simulate the interactions of fluids and fluid blast waves with solid materials, as well as the effects of impellers in mix tanks. All of these simulations can be developed without complicated moving meshes or associated mesh-tangling problems. |

## Advantages

**CartaBlanca is written in a widely used business computer language.** CartaBlanca (Version 2.0) is a large-scale, scientific and engineering software application written in Java. CFDLib is written in FORTRAN 77, which is older technology. CHAD is written in FORTRAN 95, which is object-based but not object-oriented and requires greater effort from the developers. The inherent parallelism of the Java language leads to automatic parallel code on shared-memory machines.

**CartaBlanca minimizes development efforts for simulation projects.** Using object-oriented Java, code developers can produce a mature code more quickly, with better interaction between developers, and without sacrificing performance. Because Java is a clean, type-safe programming language, CartaBlanca can support a highly interactive and interdependent development team. Using CartaBlanca, developers can cut simulation code development time by a third.

**CartaBlanca is user-friendly.** Since Java supports GUIs, CartaBlanca is very user-friendly. Because of the GUI interface, both specifying the problem and inputting the data are much easier in CartaBlanca than in CFDLib, CHAD, or FLUENT.

**CartaBlanca's speed is comparable to the competitors' speed.** The performance speed of Java is now comparable to FORTRAN's, and because so many code developers work in Java, its performance will continue to improve. Currently, CartaBlanca achieves performance speeds equivalent to the speeds of similar C language codes. We expect CartaBlanca's performance speed to increase in the future because of the attention industry is paying to Java.

**CartaBlanca, even in compiled form, is portable without any modification.** Both CFDLib and CHAD must be frequently modified and always recompiled when transferred from one computer to another. Because CartaBlanca is written in Java, the code does not need to be modified or recompiled when it is transferred from one computer to another.

**CartaBlanca uses a Jacobian-free Newton-Krylov solver.** This solver is an improvement upon our competitors. The Jacobian-free Newton-Krylov solver enables much faster solution of the conservation equations than is possible with the relatively primitive solvers available in CFDLib and CHAD.

**CartaBlanca uses the particle-in-cell (PIC) and material point method (MPM).** CartaBlanca, like CFDLib, uses the PIC method and MPM, along with the multiphase flow formulation, to enable the simulation of complex fluid structure-interaction problems such as a blast between buildings.

*Primary applications*

**Aerospace and Defense**—CartaBlanca is used to simulate projectile/target interactions for the Army. The PIC method in CartaBlanca was used to simulate the large deformations in both the projectile and target. Large-deformation simulations are easier to produce than those using conventional Lagrangian mesh approaches that suffer from mesh tangling, or those using a pure Eulerian approach, which suffer from excessive artificial diffusion or boundary smearing.

**Chemical Process**—CartaBlanca is used to simulate and optimize centrifugal contactor-separator devices for the recovery of actinides from a waste stream. CartaBlanca's multiphase flow and unstructured grid technology is key for developing these simulations.

**Homeland Security**—CartaBlanca is used to simulate the effects of a small nuclear blast between buildings. The calculation includes not only the effect of the blast wave through the atmosphere, but also the transmission of blast and damage in the building materials.

**Heat Transfer/Phase Change**—CartaBlanca is used to develop simulation algorithms using the Newton-Krylov solver for phase-changing flows in metals such as those encountered in industrial casting operations.

**Reactive Flows**—CartaBlanca is used to develop models for safe handling of high-explosive charges. These simulations use CartaBlanca's multiphase flow code with detonation chemistry.

*Other applications*

**Animation and Special Effects**—Physical models are used in the entertainment field for special effects depicting realistic fire, water, air, hair, and moving objects. CartaBlanca can be used to simulate the movement of hair and other special effects for computer games, computer animation, video games, and the film industry.

**Automotive Industry**—CartaBlanca can be used to simulate reactive flows in combustion chambers. The PIC method can be used to simulate moving valves and pistons in an engine and show the effects of heat transfer in each of the moving parts.

**Biomedical Industry**—CartaBlanca can be used to simulate pharmaceutical centrifugal devices used to separate liquids and mass exchange operations. The software allows engineers to optimize device designs.

**Oil and Gas Industry**—Process equipment design for oil production fields, including separators for drilling fluids, can be simulated and optimized with CartaBlanca.

**Environmental Industry**— CartaBlanca can be used to design and optimize waste cleanup operations.

**Nuclear Plant Safety Industry**—CartaBlanca can be used to accurately simulate the multiple physics involved in a nuclear plant coolant accident.

*Summary*

Accurate, physics-based computer simulations are widely used in all branches of science and engineering. When a new airplane turbine is designed, physics-based computer simulations are used to predict the performance of the finished part and to make changes before the aerospace company manufactures a prototype. In addition to being important in the technical disciplines, physical models are also used in the entertainment field, i.e., film industry, video games, and computer games. Physics-based computer simulations are the drivers behind special effects depicting realistic fire, water, air, hair, and moving objects. Simulation software for these special effects can be quite complex. Very often, simulation projects involve the modification of existing software to produce new capabilities. Furthermore, program or project goals change frequently as funding priorities evolve or as research developments lead to new branches of investigation. As such, code development is often the bottleneck on these projects. This bottleneck becomes a substantial incentive for software development packages that are developer-friendly—easy for scientific and engineering software developers to modify and extend their code.

CartaBlanca is the first scientific software package to employ the full power of Java, a language that continuously derives benefits from the extensive community of developers and supporters. CartaBlanca is a modern flexible software package for large-scale computational method and physical model development. CartaBlanca allows scientific and engineering code developers to produce and maintain less costly code that is not only user-friendly, but also developer-friendly.

Java is a programming language that addresses deficiencies in the C++ object-oriented language. Because of features such as type safety, array-bounds checking, and simplified object-oriented syntax, Java is a more efficient programming language. According to G. Phipps in “Comparing Observed Bug and Productivity Rates for Java and C++” (*Software: Practice and Experience*, 1999; 29:345–348), Java typically has two to three times fewer bugs and 30%–200% more code development productivity than C++. Since its introduction in the mid-90s, Java's superiority in terms of security, portability, and code-developer productivity has led to a rapid growth in the use of Java as the programming language of choice within the business community. So far, scientists and engineers have been reluctant to abandon old languages such as FORTRAN and C++ because of the small number of successful Java scientific or engineering applications. CartaBlanca is the platform for future scientific code developers.

CartaBlanca contains advanced nonlinear solver technology through the Jacobian-free Newton-Krylov scheme. At the same time, CartaBlanca's GUI capabilities lead to an extremely user-friendly problem specification. CartaBlanca has a novel design and uses object-oriented Java to provide a component-like computer architecture for the solvers and for the physics and material response modules.

In summary, CartaBlanca represents a paradigm shift in scientific computing and takes advantage of the wealth of capabilities built into the Java programming language. In addition, CartaBlanca incorporates advanced numerical algorithms, such as the Newton-Krylov solution method and the multiphase particle-in-cell treatment, that allow scientists and engineers to simulate complex phenomena such as fluid/structure interactions and reactive flows with phase changes.

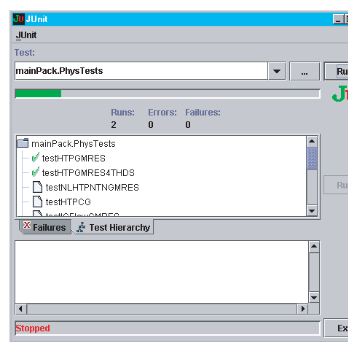
# Appendix

# CartaBlanca – Java-based Solver Environment

- Newton-Krylov solver for implicit, non-linearly consistent solutions
- Multiphase flow
- Rapid prototyping
- PIC method
- GUI interface
- Integrated unit and regression testing

# CartaBlanca

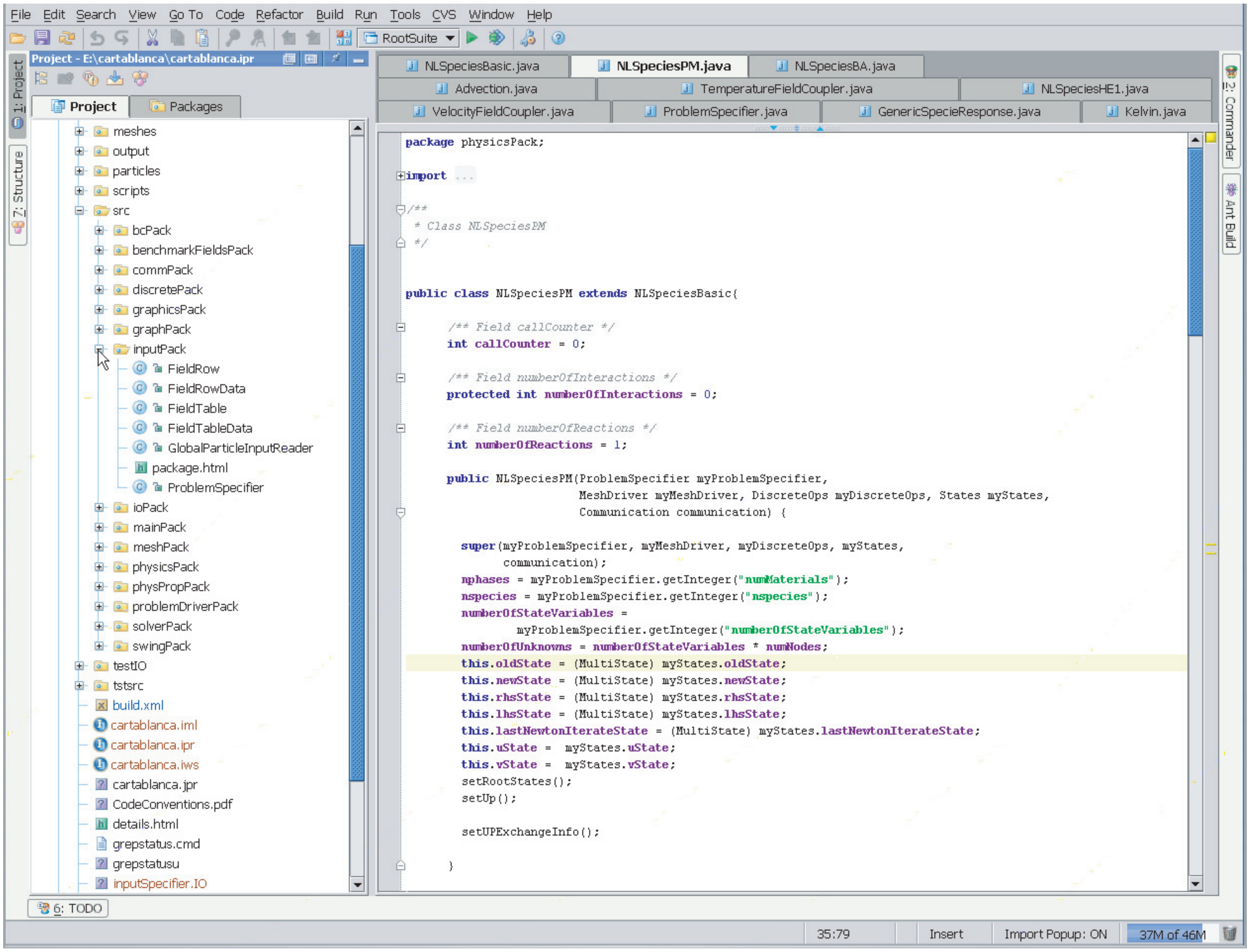
- Pure Java development environment
- Object-oriented design: developer productivity
- Phipps: Java 30-200% more productive
- Multiphase flow on 3-D unstructured grids
- JUnit for testing




# Why Java?

- Simple, clean object-oriented language
- Strong typing, extensive compiler checking
- Commercially robust: good developer tools
- Portable, robust
- Built-in thread facility, networking
- JavaGrande addressing HPC
- New fast JVMs, native compilers
- C/C++/FORTRAN interoperability
- Growing pool of developers, will soon be most widely used
- Built-in GUI (Swing), graphics, database access (JDBC)

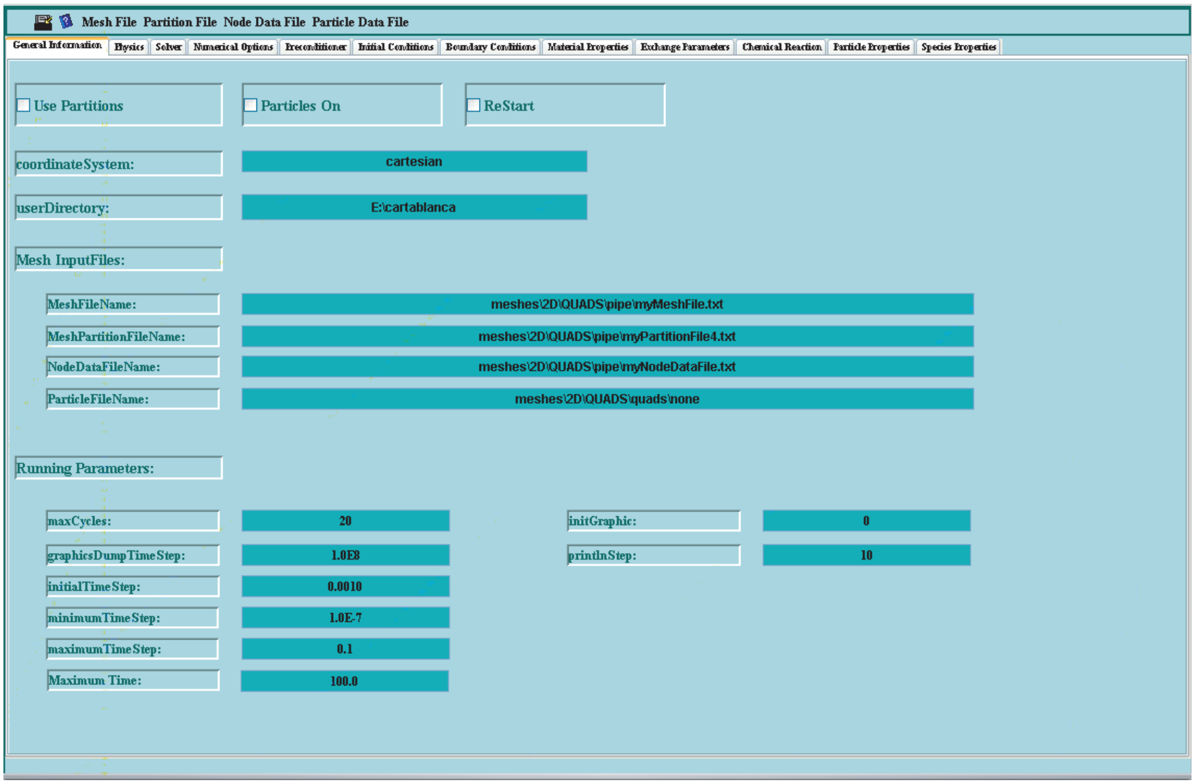
# CartaBlanca's File Structure



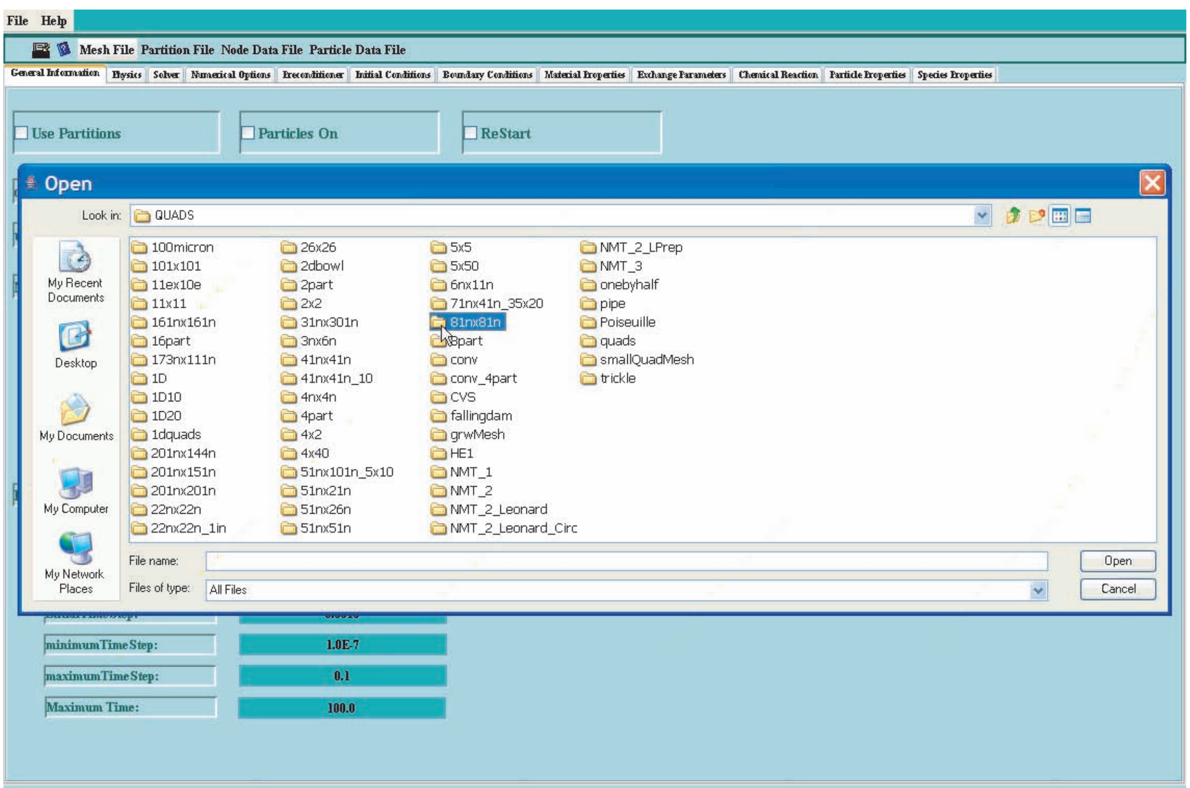

This figure shows an example of an edit session using Borland's JBuilder Java development environment to examine the CartaBlanca file structure. Because of the ubiquity of Java, the Foundation version of the Borland JBuilder environment shown here is freely available. This version is more than adequate for developing code with CartaBlanca. The same kind of capability for FORTRAN, for example, would cost several hundred dollars.

In the pane on the left, you see the directory structure of the CartaBlanca code. The user can easily interact with the objects using famliar point and click methods to drill down through the file hierarchy and to navigate in general. On the right, you see a part of the code for a species transport class in CartaBlanca. The color coding is automatic and helps the developer quickly identify the different types of variables and Java language components. JBuilder enables point and click interaction with the code in this pane. This ability is state-of-the-art and free because of Java's strong marketplace presence. This free software is part of the reason for Java's efficient development protocol.

# CartaBlanca Graphical User Interface

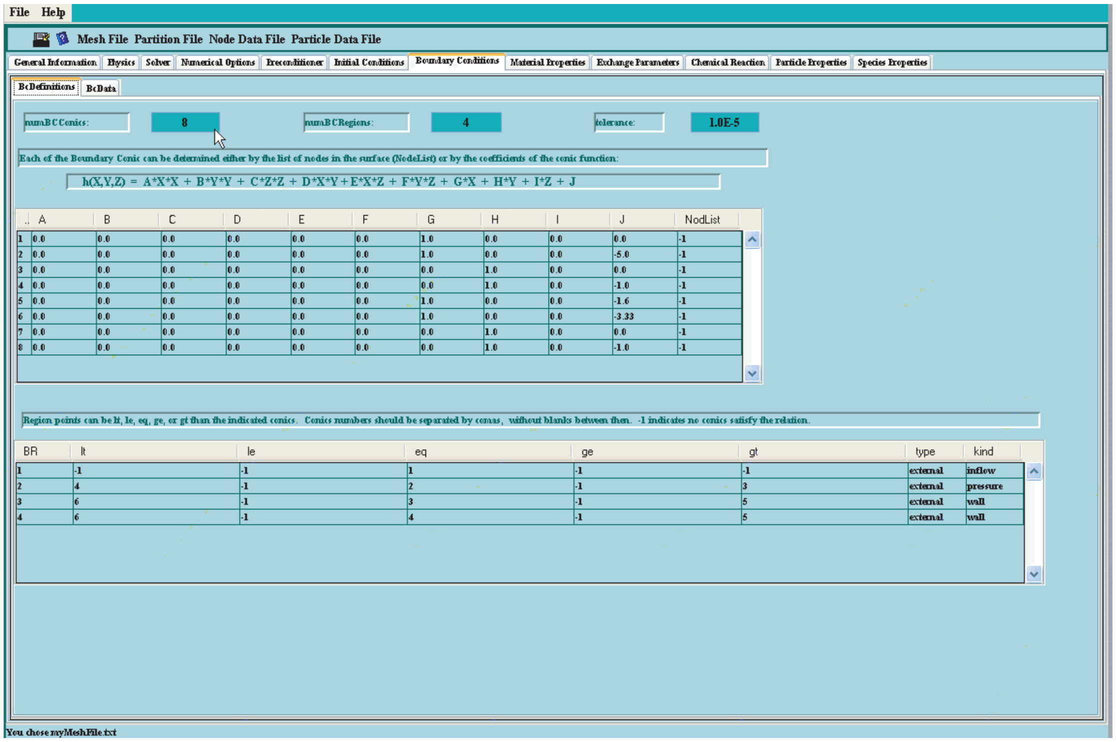



Main Page. Note multiple tabs along top to allow access to a variety of pages for specification of boundary conditions, material contants, etc.

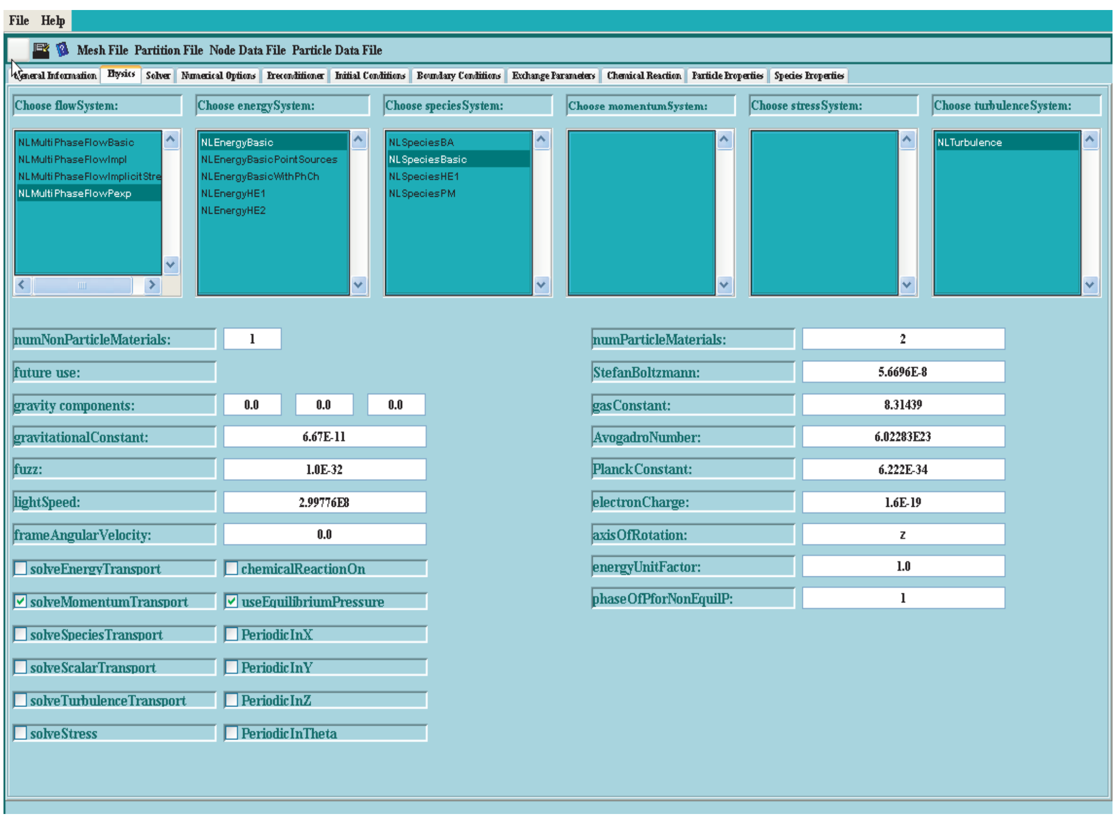



Here we see the interaction of the CartaBlanca GUI and the windows file system for point and click access to mesh files.

# CartaBlanca Graphical User Interface



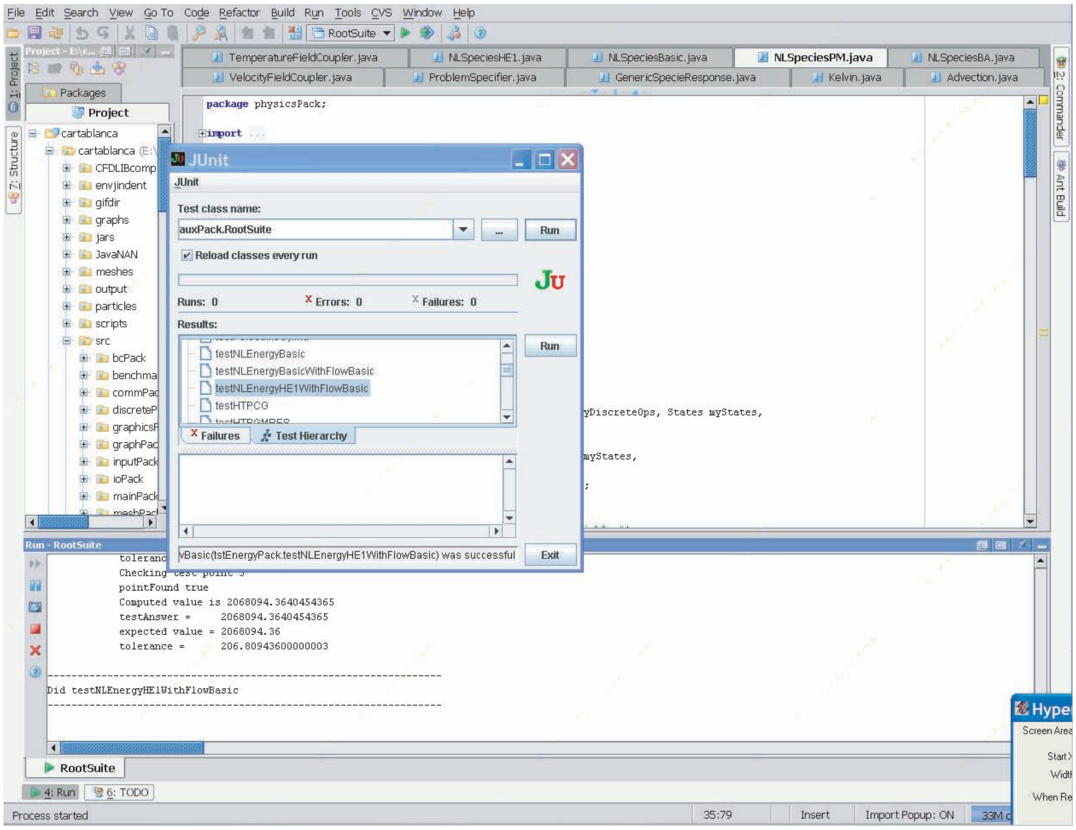

This figure is the boundary conditions page. In the top table, the user can prescribe general conic surfaces that serve as boundary sections for boundary conditions. In the second table, the user specifies which conic surfaces correspond to the boundary sections.



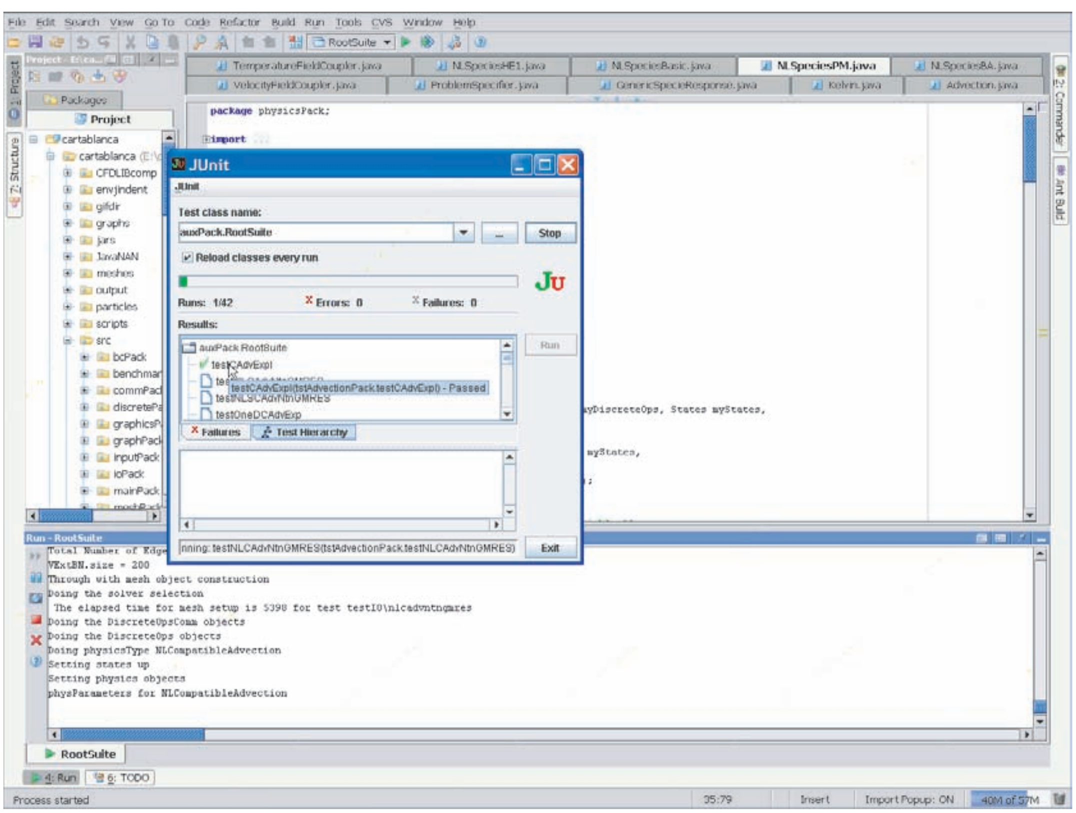

This figure is the general physics page. The bottom section contains constants and switches that control basic physics parameters. The top layer contains scroll panes that allow the user to select the type of flow, heat transfer, species transport, and other types of physcis. If a developer adds a new instance of one type, Java's reflection mechanism will automatically pullit into the menu of choice.

# CartaBlanca's Test Interfaces

## Single Test



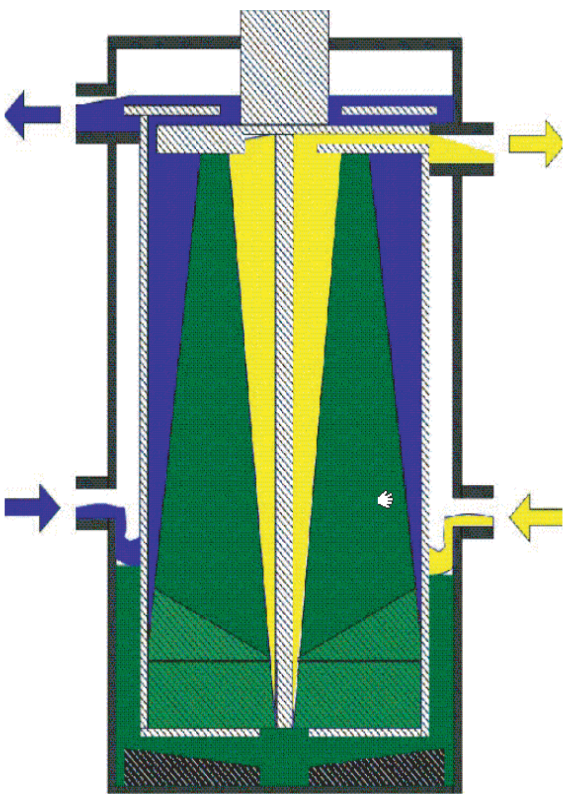

## Multiple Test



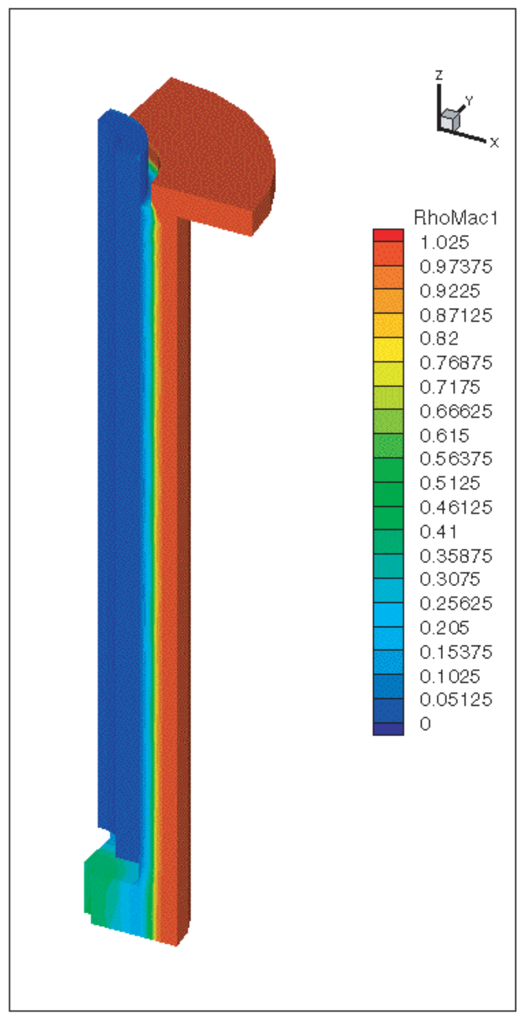

Here we see two examples showing the test suite for CartaBlanca. CartaBlanca uses the JUnit test package that drops seamlessly into CartaBlanca's framework. CartaBlanca's test suite allows the developer to make sure new modifications to the code have not broken previously installed capabilities. Nearly 50 tests are built into CartaBlanca to test many aspects of the code. These tests include solvers and advection algorithms, as well as tests of problems such as projectile target interactions. The tests are run either one by one, as shown in the single test, or automatically, as shown in the multiple test. The GUIs shown allow the developer to see the progress of the tests and to see where any failures occur. The tests are written to run quickly and to update automatically with any changes to the code. The use of JUnit along with the inherent efficiency of the Java programming language is one of the keys to the success of the CartaBlanca framework.

# Phase Separation Simulation Using Annular Centrifugal Contactors

Annual centrifugal contactors consist of two concentric cylindrical zones. The spinning rotor and the stationary housing wall form the external zone, where some liquid or organic material are being mixed. After mixing, the flow mixture enters the inner rotating cylinder through an annular opening in the bottom. In this zone, the flows are separated by high centrifugal forces, and each liquid leaves the device through exit ports on top. CartaBlanca's multiphase-flow solver is used to simulate the hydraulics of the separation zone. The lighter fluid leaves through one small opening at the center of the outside wall of the central cylindrical region at the top. The heavier leaves through another small opening at the center of the outside wall of the external cylindrical region at the top.



Schematic of Centrifugal Contactor



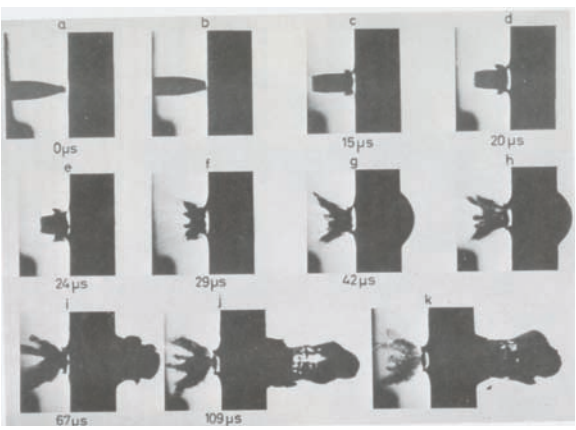

Quarter Section CartaBlanca Simulation of Contactor

# Projectile/Target Simulation

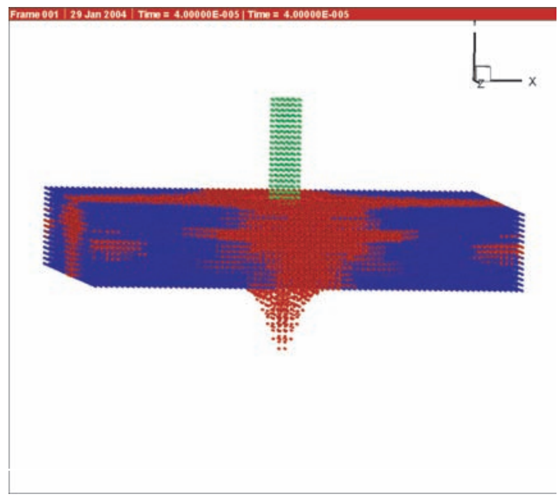
Simulations of projectile/target interactions are of interest to the armed forces. These interactions are challenging problems because of the high deformation and material damage that occurs, CartaBlanca addresses these problems by use of a combination of multiphase flow and the mesh-free particle method. It tracks the high deformation and damage without the mesh tangling experienced with conventional Lagrangian mesh code. CartaBlanca also handles the effects of gas evolution from a reactive armor problem using the same procedure.

## 3-D Bullet Brittle Plate

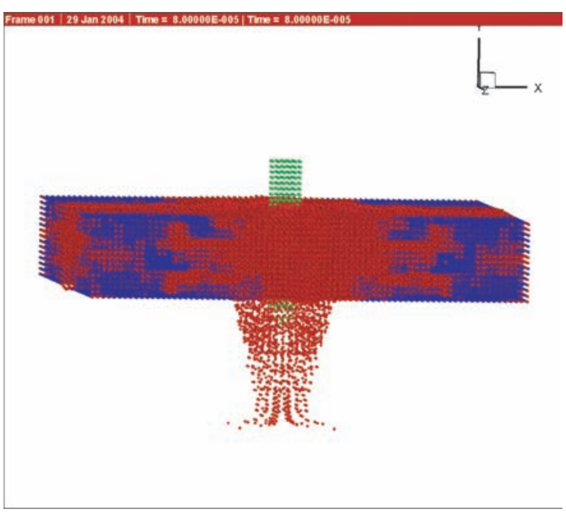
Awerbuch Experiment



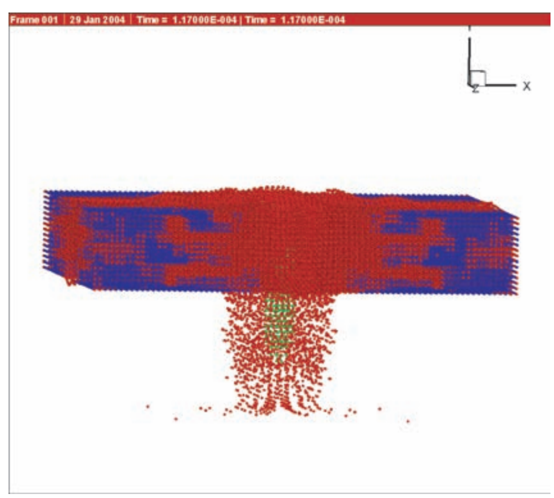

3D – Brittle – 40 microseconds




3D – Brittle – 80 microseconds




3D – Brittle – 120 microseconds

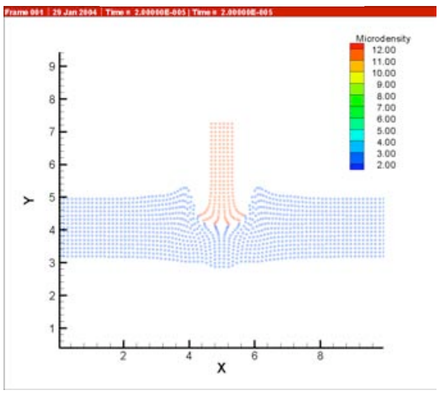


# 2-D and 3-D Bullet/Plate Interaction Simulation

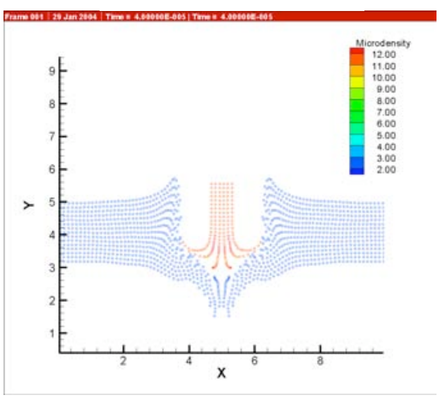
This figure set shows a simple bullet/plate interaction calculation and compares the results in two- and three-dimensional simulations. The bullet is lead and the plate is aluminum. Both are represented as particles. The 3-D simulation shows some non-axisymmetric behavior resulting from the initial plate boundary conditions.

## 2-D Bullet/Plate Simulation with Plastic Response

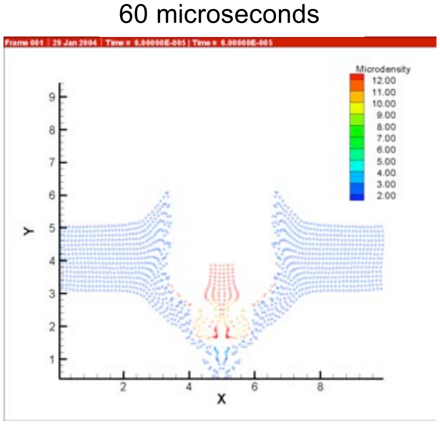
20 microseconds




40 microseconds

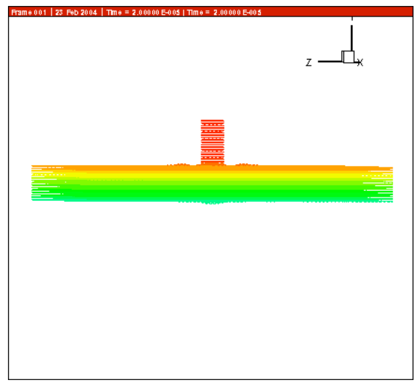



60 microseconds

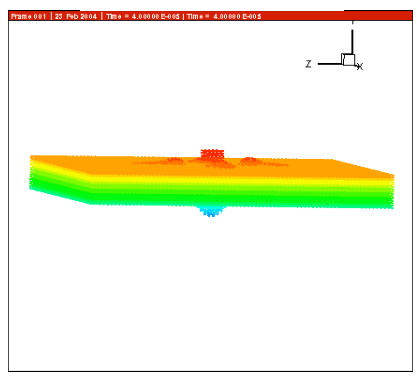



## 3-D Bullet/Plate Simulation (Serial and Parallel)

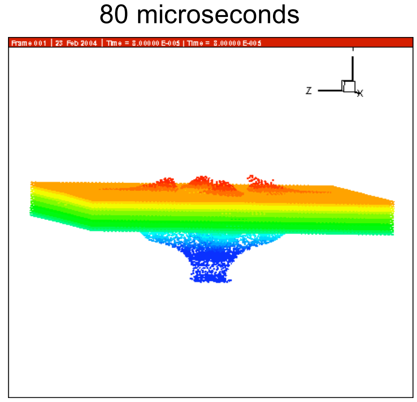
20 microseconds




40 microseconds



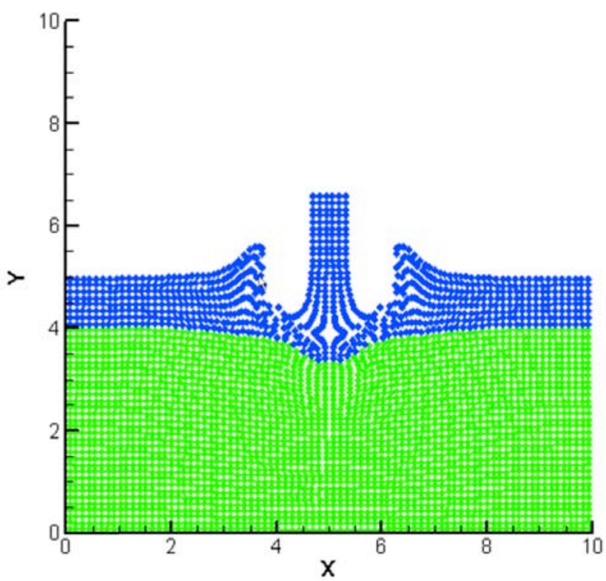

80 microseconds



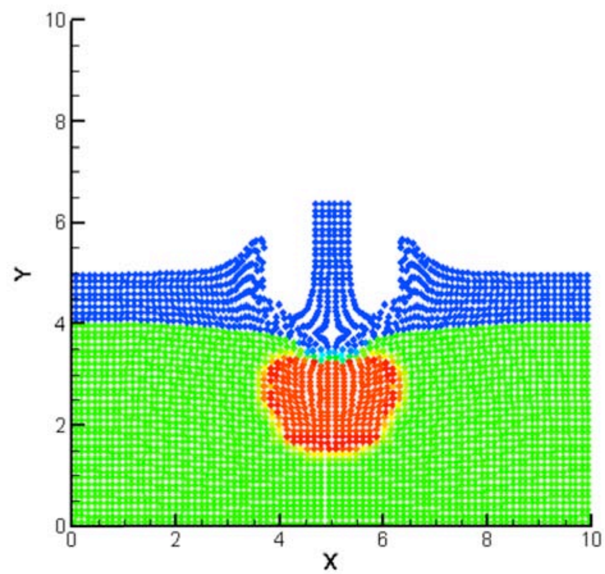
# 2-D 4-Phase Reactive Bullet Plate

Shown are the results from a 4-phase, two-dimensional calculation where a high explosive has been put at the bottom beneath the target plate. The calculation includes 1) lead bullet, 2) aluminum target, 3) gas, and 4) high explosive. The bullet penetrates the target plate and at 26 microseconds, the high explosive ignites. The orange color is the ignition front, which propagates through the material.

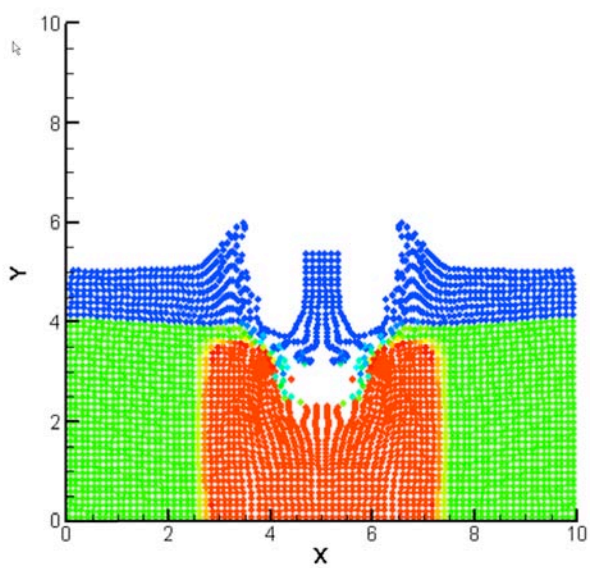
Reactive – 24 microseconds



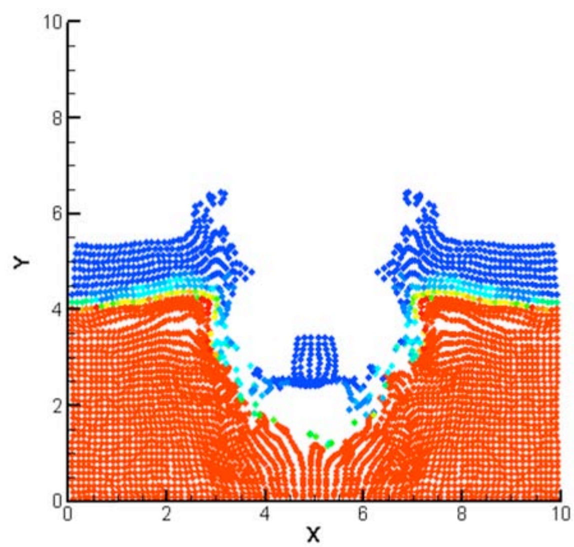

Reactive – 26 microseconds

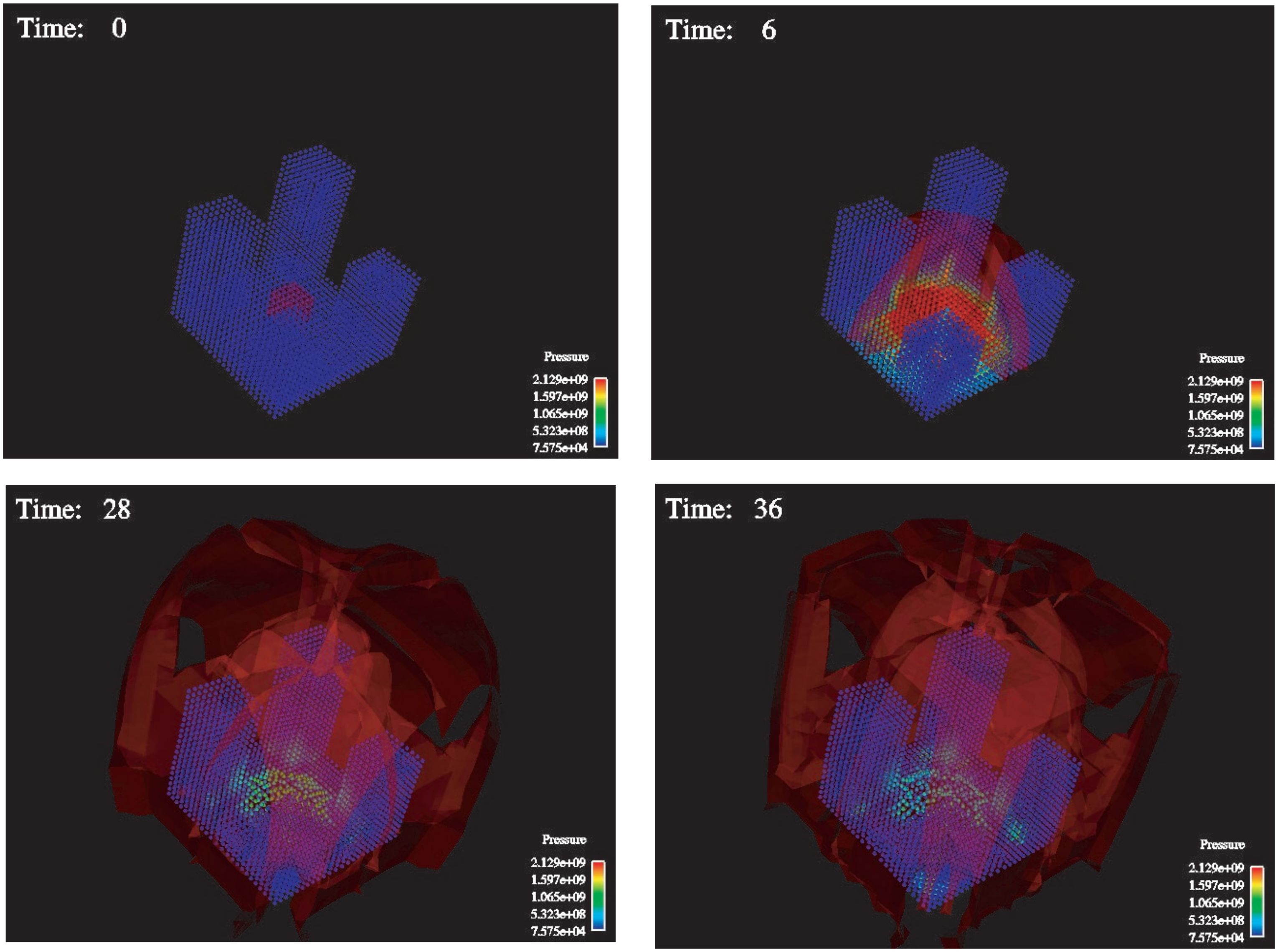


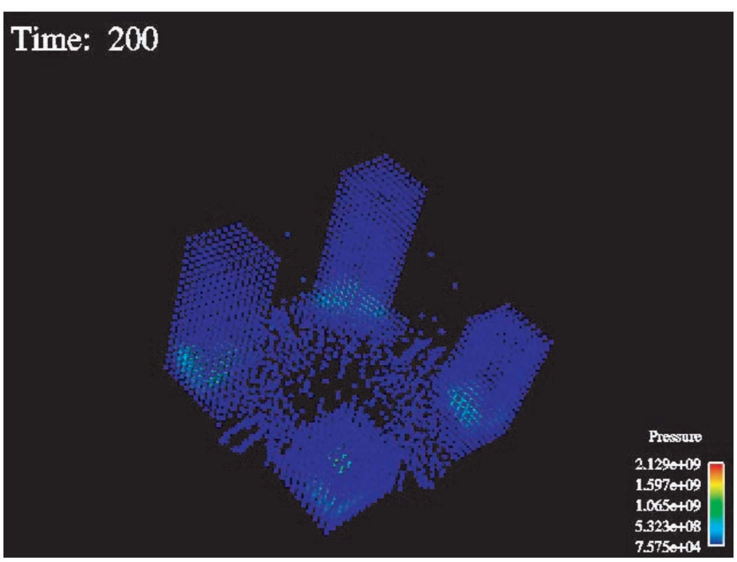

Reactive – 36 microseconds




Reactive – 56 microseconds






## 3D Multiphase-FLIP-MPM Simulation of Blast Between Buildings

- 100m cube
- 20m x 20m square buildings
- 20, 30, 40, 50 m tall
- ~3kt blast
- Elastic law with brittle fracture
- Multiphase flow

This sequence of snapshots from a CartaBlanca simulation shows the development of a blast and its effects on buildings in an urban setting. The application is analysis for homeland security. In the snapshots, the red isosurface shows the progression of the blast wave through the air. The blue particles make up the buildings and the concrete base. As the pressure wave interacts with the buildings, the material deforms and fails creating debris.