Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 00 (2013) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Science, ICCS 2013

Applying high-performance computing to petascale explosive simulations

Joseph R. Peterson[a], Charles A. Wight[b], Martin Berzins[c,*]

[a]*Department of Chemistry, University of Illinois at Urbana-Champaign, 505 S. Mathews Avenue., Urbana, IL 61802, USA*
[b]*President's Office, Weber State University, 1001 University Circle, Rm. 302, Ogden, UT 84408, USA*
[c]*Scientific Computing and Imaging Institute, University of Utah, 72 S. Campus Drive Rm. 3750, Salt Lake City, UT 84112, USA*

Abstract

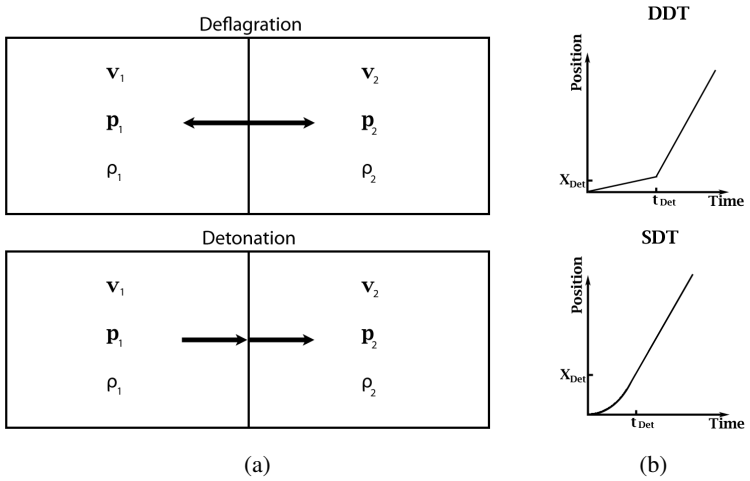
Hazardous scenarios involving explosives are difficult to experimentally study and simulation is often the only viable approach to study highly reactive phenomena. Explosive simulations are computationally expensive, requiring supercomputing resources for continued scientific discovery in the field. Here an idealized mesoscale simulation of explosive grains under mechanical insult by a high-speed projectile with reaction represented by a novel kinetic model is designed to test the scalability of the Uintah software on petascale supercomputers. Good scalability is found up to 49K processors. Timing breakdown of computational tasks are determined with relocation of Lagrangian particles and interpolation of those particles to the grid identified as the most expensive operation and ideal for optimization. Potential optimization strategies are identified. Realistic model simulations rather than toy model simulations are found to better represent scalability of a science code on a supercomputer. Estimations for total supercomputer hours necessary to complete the kinetic model validation study are reported.

Keywords: Energetic Material Hazards; Uintah; MPM; ICE; MPMICE; Scalable Parallelism

1. Introduction

The problem of interest in this study is that of the accidental detonation of explosives arising from a transport accident in Utah in 2006 [1]. Research into explosives and explosions reaches back as far as the early 20th century, with many empirical theories based on experimental evidence resulting. These empirical models work very well at the two extremes of reaction, namely combustion (also known as deflagration) and detonation. Combustion can be understood as a relatively slow, dynamic equilibrium oxidation process of a material; for example, the burning of a log in a fire. Detonation, on the other hand is a very fast, dynamic equilibrium reaction that occurs at or beyond the speed of sound characteristic of the unperturbed material in front of the detonation front. Figure 1a demonstrates the fundamental difference for these processes at dynamic equilibrium which arise from the fact that one is thermal transport limited (very slow), while the other is pressure-driven (very fast). At dynamic equilibrium, a self-sustaining reaction wave passes through and consumes the material neither excessively speeding up or slowing down. For a detonation, this wave propagates at the detonation velocity. The detonation wave is sustained by mass advection from products into the undisturbed reactants, causing pressurization and reaction. A self-sustained combustion wave is propagated via thermal diffusion from the flame to the surface, which vaporizes the explosive and breaks chemical bonds, allowing energy release by formation of gas products in the flame region.

*Corresponding author. Tel.: +1-801-585-1545 ; fax: +1-801-581-5843.
E-mail address: mb@sci.utah.edu.



(a) (b)

Fig. 1: a) A schematic of deflagration and detonation. Products (1) are on the left, and reactants (2) are on the right. For both scenarios the velocities, $v_1 > v_2$, and pressures, $p_1 > p_2$. Densities and direction of product flow differ. In deflagration, the products move away from the reaction plane and $\rho_1 < \rho_2$, while in detonation, the products move into the reaction plane and $\rho_1 > \rho_2$. b) A schematic of DDT (top) and SDT (bottom) phenomena in position-time space. The shock front accelerates in the SDT case and becomes supersonic at the transition-to-detonation point. For DDT, the shock front progresses at constant speed and a reaction front formed in its wake catches up to the shock front initiating a supersonic detonation when it collides with the lead front.

While empirical models work well for dynamic equilibrium, they cannot capture non-equilibrium processes, or perform reliably outside their range of calibration. However, accidents involving explosives, some which resulted in loss of human life, are generally non-equilibrium [2]. Two non-equilibrium processes have been identified. The first is a deflagration-to-detonation transition (DDT), where weak impact or strong combustion causes a pressure wave in a porous, or cracked media, that results in a flame front penetrating the void space, which accelerates and catches the pressure front, causing a detonation to begin at that point [3]. The second is a shock-to-detonation transition (SDT), where an explosive is impacted by a projectile at several hundred to several thousand meters per second, and the shock wave builds due to reaction, eventually causing a detonation. A schematic of each process can be seen in Figure 1b.

These non-equilibrium processes are well characterized and accurately modeled in systems of gases (e.g. hydrogen and oxygen mixture) as well as homogenous explosives. However, heterogeneous explosive prove to be more complex. Heterogeneous explosives can be mixtures of two different types of explosives, or a mixture of explosive and plastic bonding agent that holds the explosive together. Also, a bed of packed solid grains of material with intermittent pore space filled with gas can be considered heterogeneous due to of the factor of 1000 difference in density of the solid and gas.

A digitized microstructure of a plastic bonded explosive can be seen in Figure 2 and demonstrates the variety of grain shape and void size. This figure lends insight into why DDT and SDT are difficult to understand in heterogeneous explosives. The vastly different shapes and sizes prevent bulk-scale models, which average over large volumes, from capturing the physics relevant to initiation and propagation of the reaction. For example, if the explosive sample is impacted from the top, two edges of grains may be pushed together and an energy localization forms (due to friction and plastic work). This process can be thought of like a match head striking on a igniter strip where both materials are made of small grains. Sometimes enough energy is localized to raise the temperature above the ignition threshold forming “hot-spots” which react releasing heat. Modeling on a larger length scale than the grain cannot resolve these hot-spots.

In order to form models that accurately capture relevant physics on the bulk-scale, sub-grid scale models must be formulated. These require fully resolved simulations to be statistically analyzed for trends in hot-spot number density based on a specific extent and quality of heterogeneity. Simulations like these are considered “mesoscale”

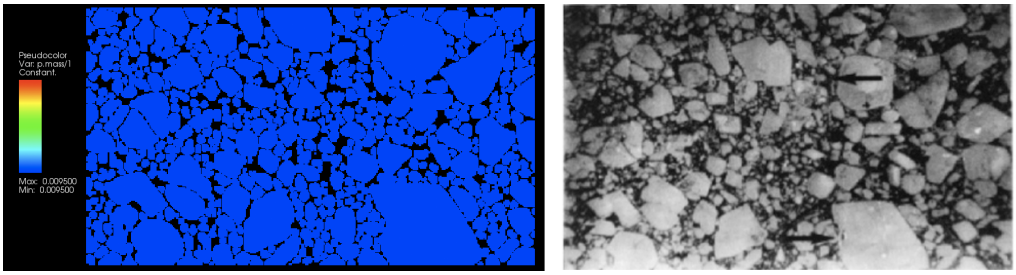


Fig. 2: A digitized version of an HMX micrograph suitable for simulation (left) and a actual photograph of a microstructure (right). The largest grains are 200 micrometers across. A simulation with this type of heterogeneity is the "Target" simulation.

meaning the scale between nano- and bulk-scale and are generally comprised of constituents with length scales on the order of a few to a few hundreds of micrometers. Once mesoscale effects are understood, statistically based, bulk-scale models can be developed and tested. Ultimately, these bulk-scale models may achieve a level of predictiveness.

One example of a heterogeneous explosive of interest is known as HMX (octahydro-1,3,5,7-tetranitro-1,3,5,7-tetrazocine) which has grains on the order of 5 to 200 micrometers. In order to computationally resolve the energy localization the simulation domain must be decomposed into sub-micormeter elements, resulting in simulations with tens of millions (2D) to billions (3D) of elements. Such large domains constitute huge numbers of calculations with huge memory requirements and can only be simulated with supercomputers. Hence, mesoscale explosive simulations are an ideal candidate for testing and ultimately compelling development of a massively parallel science code.

The following sections will describe the development of one such mesoscale simulation including performance results obtained on the Kraken[1] and Updraft[2] supercomputers. Section 2 describes the computational and physical models as well as the simulation configurations used. Results and their discussion are presented in Sections 3 and 4. Concluding remarks and future work, are presented in Section 5.

2. Method

2.1. Uintah Computational Framework and MPMICE

Uintah is a component-based framework that separates physical science or engineering model development from the computer science design topics such as memory management, parallelization, scheduling and load balancing. Uintah[3] was created at the University of Utah and originally developed for simulations of fires and explosions. It is a software framework into which science codes may be developed. The code has a fluid-structure interaction component where liquids interact with solids, contains a low Reynolds number combustion component and now a molecular dynamics component implementation is in the works. What is most interesting is the fact that several of the components in Uintah, including the ARCHES fire code and the ICE fluid code, have shown linear scalability on simple test problems up to 256K cores on Department of Energy and National Science Foundation computers [4, 5].

An asynchronous, task-based approach is used to overlap computation and communication [6, 5]. Each model task registers a set of required and computed variables and a directed acyclic dependency graph is created that represents the task workflow for a time step. The computational domain is decomposed into patches of cells which are assigned to different processors. An execution of each task in the graph is performed on the patches

[1] Kraken is a Cray XT5 based NSF supercomputer containing 18,816 2.6 GHz six-core AMD Opteron processors with 147 TB of memory and a Cray SeaStar2+ interconnect.

[2] Updraft is a Sun Microsystem based University of Utah supercomputer containing 512 2.8 GHz quad-core Intel Xeons with 4096 GB of memory and a Qlogic Infiniband DDR interconnect.

[3] `http://www.uintah.utah.edu`

owned by a particular compute device (node) and then halo data, that which is needed on other processing nodes, is communicated across the network.

The component used for this study is called MPMICE, an Eulerian-Lagrangian fluid structure code [7, 8] which is a marriage of the implicit-continuous-Eulerian (ICE) method [9] and the Lagrangian-based Material Point Method (MPM) [10]. Fluids are represented by ICE and solids by MPM. MPM materials are interpolated to the grid at each time step, effectively turning them into ICE materials and the coupled mass, momentum and energy conservation evolution Equations (1-3) are solved on a structured grid where p is the pressure, ρ is the density, E is the internal energy and $\vec{u}$ is the velocity vector. Then the properties are interpolated back to the MPM materials. MPM is able to retain history of the material and thus represent stresses, shears, damage and other mechanical properties necessary for solids. MPMICE is well suited for studying explosive processes and coupled fluid-structure interaction problems.

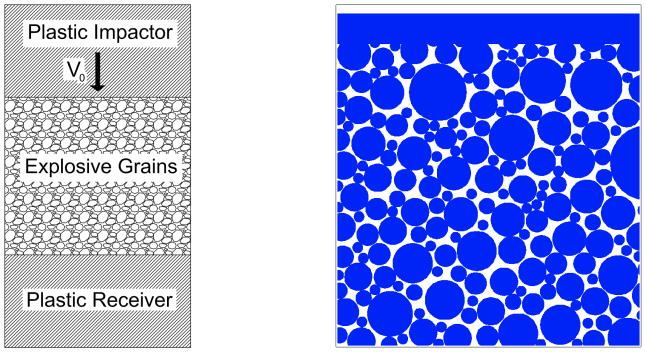
$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \tag{1}$$

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\vec{u} \otimes (\rho \vec{u})) + \nabla p = 0 \tag{2}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot (\vec{u}\,(E + p)) = 0 \tag{3}$$

2.2. Models

A schematic of the "target" simulation can be seen in Figure 3a. Two simulation setups will be used to validate the material models. In the first setup, a solid crystal of explosive is impacted with varied velocity and sample thickness. In the second setup, a randomly generated bed of cylinders based on an experimentally determined bimodal distribution of explosive crystal grain sizes is impacted at various velocities [11]. The second setup is representative of the target simulation, with idealization of grain shape. Cylindrical grains were packed until 65% of the area was the explosive HMX. Space between grains was filled with gas. The initial velocity of the impactor was chosen to match the experiments [11, 12, 13, 14]. The stress and velocity profiles at the bottom and top of the impacted crystal or bed can be compared with experimental results. The two setups will hereon be referred to as the "Single crystal" and "Granular bed" experiments.



(a) (b)

Fig. 3: a) A schematic of the granular bed simulation. An impactor flies from above with a supplied velocity v_0 and impacts a granular bed imparting a stress wave that propagates through the bed, ultimately arriving at the plastic witness plate. Measurements are taken at the two plastic/explosive interfaces. b) An idealized geometry of an impact simulation containing a bimodal distribution of cylinders. The impactor is shown as a box at the top.

Material models from literature were used for HMX. An elastic-plastic constitutive model described by Steinberg, Cochran and Guinan (SCG) [15] was used to represent the stress response of the HMX with parameters from Benson and Conley [16]. A temperature-dependent specific heat model fit by Menikoff to molecular dynamics

calculations was used [17] along with a temperature-dependent melting model [18]. A Mie-Grüneisen equation of state (EOS) was used to represent the material response of the Kel-F 800 impactor and the TPX receiver plate with parameters from literature [19]. The form of this EOS can be seen in Equation (4), where p is the pressure, ρ and ρ_0 are the current and initial density, Γ is the Grüneisen parameter and E is the specific energy.

$$p = \frac{\rho_0 C_0^2 \left[\eta - 0.5\Gamma_0 \left(\eta - 1\right)\right]}{\left[\eta - S_\alpha \left(\eta - 1\right)\right]^2} + \Gamma_0 E \qquad E = \rho_0 C_v (T - T_0) \qquad \eta = \frac{\rho}{\rho_0} \tag{4}$$

The void gas and product gas were represented by an ideal gas EOS, seen in Equation (5), with parameters for dry air. These gases differ by the specific heat C_v, initial density ρ, and adiabatic index γ, parameters. Product gas parameters were fit to experimental compositions above the reacting surface.

$$p = \rho(1 - \gamma)C_v T \tag{5}$$

A reaction rate model for HMX from Ward, Son and Brewster (WSB) [20] was used for its accurate representation of burn rate temperature and pressure dependence [20, 2, 3]. The mass flux, or amount of mass reacted through a given burning surface area, is represented by Equation (6) where definition of the constants can be found in the relevant literature. An adiabatic induction time model was used to accurately time ignition of individual Lagrangian MPM particles of HMX, with the form found in Equation (7). This model used Arrhenius parameters from Menikoff [21], where A is the scalar frequency of reaction, and E_a is the activation energy for the explosive and Q is the heat released by reaction.

$$\dot{m} = \sqrt{\frac{A_c R T_s^2 k_c \rho_c e^{-E_c/RT_s}}{E_c \left(C_p(T_s - T_0) - Q_c/2\right)}} \tag{6}$$

$$t_{adb} = \left(\frac{T^2 C_v R}{E_a Q}\right) \frac{e^{-E_a/RT}}{A} \tag{7}$$

Single crystal experiments were performed with 0.1 *mm* per side, cubic cells with 16 material particles per cell. Granular bed simulations were performed with 5 μm per side, cubic cells with 8 material particles per cell. In literature [22], each grain in mesoscale granular simulations is suggested to be at least five Eulerian zones across. Here there are about 2.5 Lagrangian particles across the smallest grain of 6.2 μm, for a total of about 5 Lagrangian points constituting the smallest grain, which is roughly equivalent.

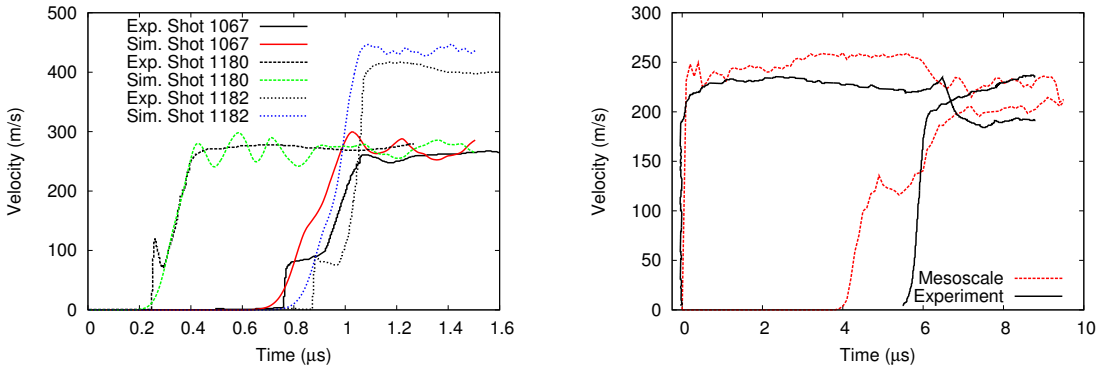
2.3. Timing and Scaling

The optimal number of patches in two dimensions (2D) and three dimensions (3D) were determined on Updraft. The idealized geometry shown in Figure 3b was used for determining the optimal side length for a patch in 2D or 3D. This ideal geometry is designed to replicate the ratio of cells filled to unfilled cells for a real explosive microstructure like that seen in Figure 2. Patch optimization simulations were run on 4 or 8 processors for 2D and 3D respectively. Simulation results were averaged over three runs. Scaling simulations were run on Updraft from 32 to 1024 cores. Weak scaling steps were run with factor of 2 increments in both simulation size and number of cores for each simulation and factor of 2 increments for fixed problem size strong scaling. Simulations on the Kraken supercomputer were run from 12 to 49152 cores. Weak scaling simulations were run with a factor of 4 increase in size and number of cores and factor of 4 increments for fixed problem size strong scaling. Additionally, task timings were output on both computers at 192 cores and 1032 cores by setting the Uintah debugging environment variable `SCI_DEBUG=ExecTimes:+`. Timings are used to identify tasks ideal for optimization.

3. Results

3.1. Model Validation

Single crystal experiments show good agreement in timing and magnitude of the velocity of the interface as the wave passes the HMX/receiver interface. Comparison with experiments are seen in Figure 4a. The SCG



(a) (b)

Fig. 4: a) Comparison of the material model behaviors with single crystal experiments. Good agreement is seen in magnitude and time scale of features. Note, however, that the initial elastic wave that is represented in the experiments by a spike, is missing form the simulations. b) A comparison of laterally average simulated velocity compared with experiments for Shot 912, a granular bed packed to about 65% theoretical maximum density. The left and right pairs of traces depict the measurement at the top and bottom of the bed.

model does not model the initial elastic wave, and thus lacks the sharp spike seen at the impact point in the experiments. One approach to to reduce the nonphysical oscillations at peak pressure as well as capture the initial elastic precursor spike would be to use the viscoplastic used by Menikoff and Stewart to study HMX [17].

A more pertinent comparison is the granular compaction, as it includes more heterogeneous behavior than the single crystal, and is a better measure of the overall ability of the model system to represent the target simulation scenario. The granular compaction simulation results shown in Figure 4b compare favorably with experimental velocity traces. The stress wave propagate through the bed too quickly, however the magnitude of response at the top (left) and bottom (right) of the bed are in good agreement.

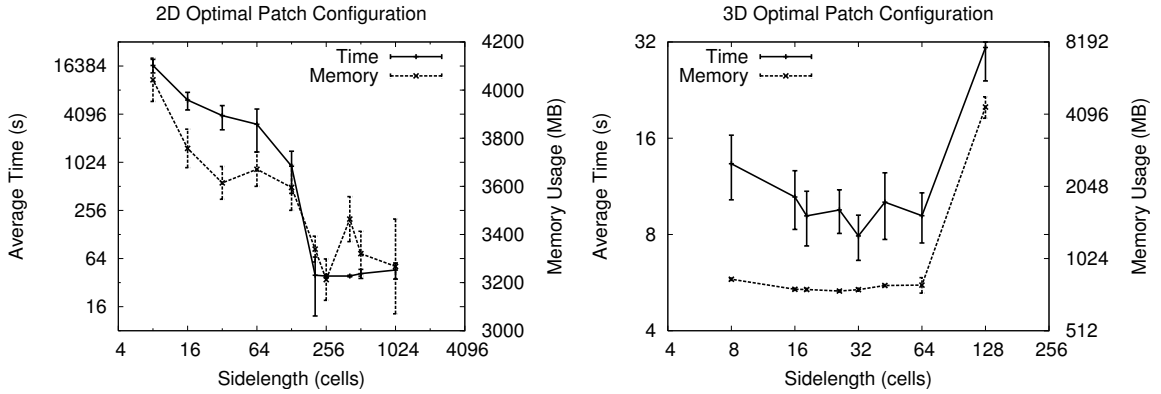
3.2. Patch Size Optimization

Running the idealized geometry for a 256^2 cell domain in 2D yielded the timing and memory usage results seen in Figures 5a and 5b. Similarly, for a 3D scenario, with a 128^3 cell domain yielded the results seen in Figures 5a and 5b. An examination of the average memory at patch side lengths greater than the optimal show little correlation, while average memory usage below the optimal patch side length shows strong correlation.

It can be seen that the optimal patch side lengths are 128 cells and 32 cells for 2D and 3D corresponding to 16384 and 32768 cells, respectively. When accounting for ghost/halo cells, data that must be transferred between compute devices, this amounts to a total of 50700 and 39304 cells for 2D and 3D, respectively, which are similar to within about 29%. However, these simulations were run on a single node and neglect any internode communication costs for halo information. On multiple nodes the 2D case will have have a considerable advantage due to smaller halo data transfers. All scaling simulations are run in 3D with 16^3 cells per patch. The reduction from optimal 3D patch size is due to the memory constraints on each node. The overall performance hit by the reduction compared to optimal patch size is only about 10%.

3.3. Model Timings

A breakdown of task time as a percentage of the total time for given time step is shown in Figures 6. Simulations were run on Updraft at 192 and 1032 cores to probe the effect that different core count has on the times. Data shown is an average over the timing on each node. Only the tasks that account for 90% of the total simulation time are shown. The results are similar, but some of the tasks switched place. However, due to the size of the error bars it is difficult to distinguish the order of the top six tasks. At larger core counts, the task `relocateParticles`, which takes the bulk of the computation time, increases in overall time relative to the rest of the tasks. At high



(a) (b)

Fig. 5: Average time and average memory use for a time step as a function of cell side length with error bars indicating one standard deviation of the mean for a) 2D and b) 3D domains.

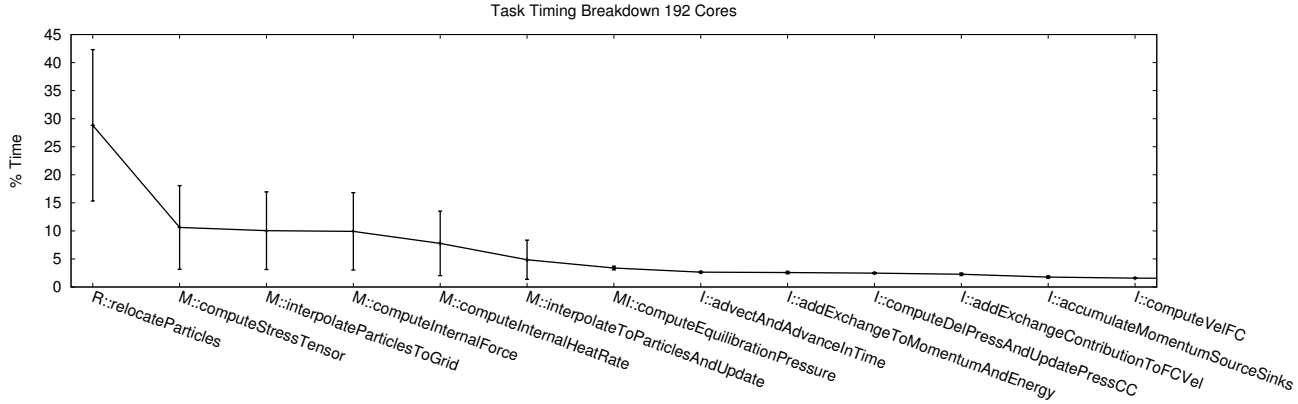


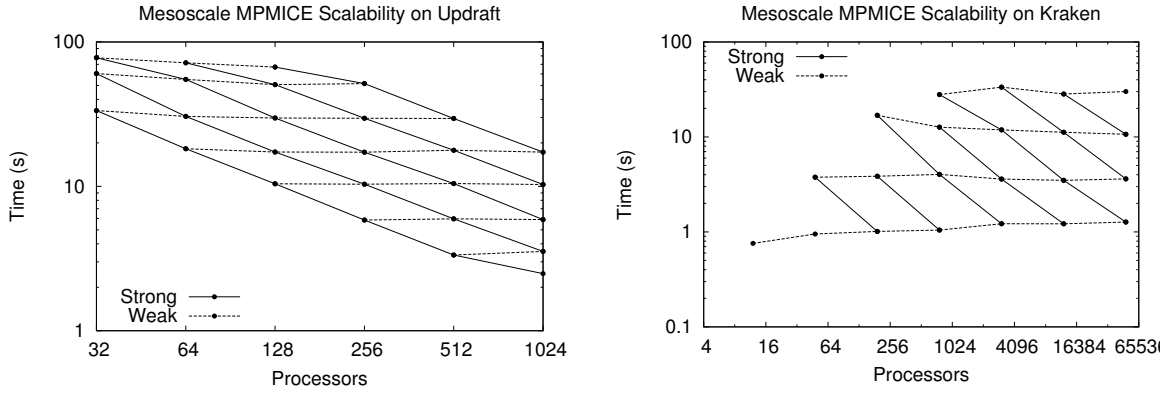
Fig. 6: Breakdown of task timings run on Updraft for 192 cores using a fundamental feature of the particle code. The 1032 core result is similar. Task prefixes are: “M” - MPM, “I” - ICE, “MI” - MPMICE and “R” - relocation.

core counts the rest of the time is distributed over the remaining tasks. The most expensive six tasks account for 67% and 77% for the 192 and 1032 core runs, respectively.

3.4. Scaling Results

Initial scaling results were performed on the local 2048 core Updraft supercomputer. Simulations were run up to half the total size of the machine. The results can be seen in Figure 7a. Strong scaling (increasing the number of cores for a fixed problem size) is ideal out to at least 1024 cores for all but the smallest problem size. Similarly, weak scaling (increasing core counts with a fixed amount of work per core) trends on the graphs are nearly flat, or even slightly downturned, which would indicate ideal or better than ideal scaling. These results provide a baseline comparison for those for the more costly simulations on the Kraken supercomputer which has considerably higher processor cost and of which we have a limited allocation of resources.

Scaling results from the Kraken supercomputer are presented in Figure 7b. Simulations were run up to 4096 12-core nodes. Kraken contains enough cores to run full 3D mesoscale simulations, making good scalability of the target simulation on this computer paramount. Less than ideal weak scaling can be seen for small simulation



(a) (b)

Fig. 7: Weak and strong scaling results from the a) Updraft and b) Kraken supercomputers where points indicate time to simulate a time step. Strong scaling depicts performance behavior of a fixed problem size as the processor count increases. Weak scaling depicts the performance behavior when a problem sized is scaled proportionally to the processor count. Points represent the times measured at that processor count.

sizes. A slight breakdown of strong scaling is seen above a factor of 16 increase in the number of processors. This effect is minimal. However, weak scaling becomes more ideal as the problem size increases similarly to what is seen on Updraft. The limited access to greater than 49K cores and high memory cost of the simulation prevents this scaling study for larger problem sizes or core counts.

4. Discussion

During patch optimization, optimal patch sizes in 2D and 3D were found to be 128^2 and 32^3 cells per patch. Both of these are factor-of-two numbers, likely due to memory alignment. In the 2D case there a sharp threshold exists between 128^2 and 71^2, which cannot be explained by memory usage. In the 3D optimization, time to solution follows memory usage, especially for the large patch size. In the case of the 128^3 cells per patch the available on-node memory is exceeded and the data spills to disk.

Task time measurements give a good idea of where effort on optimization efforts should be focused. The `relocateParticles` task takes a large portion of the computation, and should be optimized. The task is used to determine which particles crossed patch boundaries and need to be transmitted from one MPI context to another, possibly pushing data across a network interconnect. As such, it loops over all particles on a node, determines if they have crossed a cell boundary into another patch, and communicates them to the processor that needs them. This causes a global barrier, as all tasks must reach this point before all data exchange can occur. Particles are not identified with specific cells. Identifying particles with cells could speed up this process. MPMICE takes time steps related to the speed of sound in a material, essentially preventing transmission of a pressure wave across a computational cell in fewer than "x" time steps, which is a value usually around 10. Utilizing this fact, it can be shown that a particle cannot traverse more than one cell in "x" time steps. Therefore, if every fraction of "x" time steps a list of all the particles in boundary cells were enumerated and added to a list of particles to check, the `relocateParticles` task could be sped up on all other time steps just by checking those cells effectively would amortize the expense over a number of time steps. Alternately, a solution using an asynchronous implementation of this task could remove the blocking behavior, which could increase efficiency. This could be implemented by spawning an additional task to buffer send and receive communications on a node. A third alternative is to use one MPI process per multicore node so that particles will much more often only cross an internal boundary inside the patches on a multicore node [23]. This approach has resulted in good scaling up to 256K cores.

While the `relocateParticles` task takes a good portion of the overall task, four other tasks including the `computeStressTensor` taks, the `interpolateParticlesToGrid` task, the `computeInternalForce` task,

and the `computeInternalHeatRate` task take another 30-40% of the total computation time. One method that is common to all these tasks is the `findCellAndShapeDerivatives` which is called for each particle. It contains many repeated memory accesses and multiplications as the particles are interpolated to the grid points. A simple replacement of repeated operations with cached values incurs an approximately 4% reduction in computation time. To put this in context, a full 3D simulation running one 32K cores for one whole day with these code changes could save about 30K computer hours of an allocation.

5. Conclusion

A model system has been developed for the exploration of mesoscale reaction of granular crystalline explosives under impact. The numerical models have been validated for nonreactive scenarios here and elsewhere [1]. A breakdown of times required for different portions of the calculation was studied and potential areas of optimization identified. Optimal work per processor was determined in both two and three dimensions for a particle heavy simulation. Lastly, scaling studies were performed on a representative mock of the target simulation in 3D in order to estimate the scalability on two available supercomputing resources.

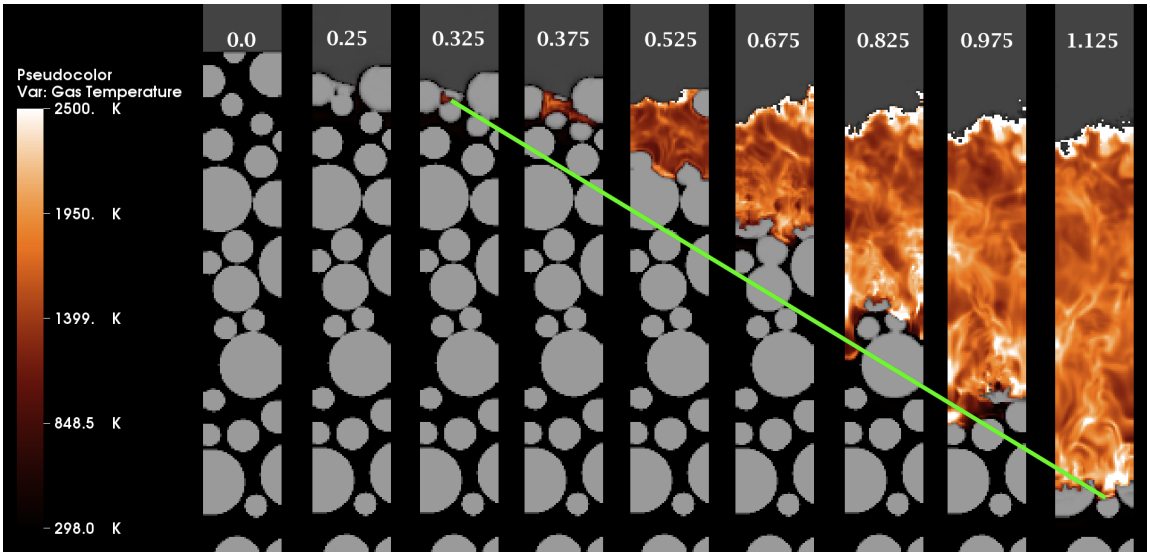


Fig. 8: A frame capture from a 2D simulation of a granular bed impacted at 696 m/s showing reaction initiation in a pore. Indicated times are in microseconds. The shock front has already passed far beyond the area shown, and thus this is a model for DDT. The green line qualitatively demonstrates that the reaction front accelerates, which hints that the simulation may be capable of capturing the DDT phenomena.

An image capture from several frames of a preliminary 2D simulation of the granular bed simulation are shown in Figure 8. The bed was impacted at 696 m/s and begins reacting. The frames demonstrate an acceleration of the reaction front as was found in the DDT experiment, which is a very encouraging result [13]. This constitutes the first simulation of its kind. If the physical model is able to represent the experimentally measured quantities, which are necessarily averages due to the length and time scales of measurement, it has the potential of being the first model that can quantitatively study the mesoscale effects that lead to transitions between reaction regimes. From mesoscale effects, bulk-scale models that had predictive powers could then be designed [1]. This provides a compelling argument for doing a full validation of the reaction model.

These scaling studies enable estimation of total supercomputer hours required to validate the model. For a 512^3 cell simulation, roughly the size needed, 32K or more cores are necessary. This assumes about 1 million time steps will be needed based on the average time per time step and a total simulation time of 20 microsecond. A total of 9 simulations are needed to compare to experiments and validate the reaction/material model combination. Total estimated processor hours for validation of the reaction simulations is 5 million. Accounting for unexpected

issues, and potentially for 3D validations of non-reactive scenarios, this number should probably be doubled to about 10 million computer hours. The sheer size of the simulation creates memory requirements that exceed the capabilities of small supercomputers like Updraft. Hence, national resources such as Kraken, Ranger, Stampede or Titan will be required and will be used in future work.

Acknowledgements

We would like to thank Scott Bardenhagen, Ananda Barua, James Guilkey, Todd Harman, Alan Humphrey and Qingyu Meng for helpful discussion. This work was supported by the National Science Foundation under subcontract No. OCI0721659. Uintah was developed by the University of Utah's Center for the Simulation of Accidental Fires and Explosions (C-SAFE) and funded by the Department of Energy, subcontract No. B524196. We would like to thank the University of Utah Center for High Performance Computing (CHPC) along with the Texas Advanced Computing Center (TACC) and National Institute for Computational Science (NICS) for access to computing resources.

References

[1] J. Peterson, J. Beckvermit, T. Harman, M. Berzins, C. Wight, Multiscale modeling of high explosives for transportation accidents, in: Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment, Chicago, IL, USA, 2012.
[2] C. Wight, E. Eddings, Science-Based Simulation Tools for Hazard Assessment and Mitigation, Advancements in Energetic Materials and Chemical Propulsion 114 (2008) 921–937.
[3] J. Peterson, C. Wight, An eulerian-lagrangian computational model for deflagration and detonation of high explosives, Combustion and Flame 159 (7) (2012) 2491–2499.
[4] M. Berzins, J. Luitjens, Q. Meng, T. Harman, C. Wight, J. Peterson, Uintah - a scalable framework for hazard analysis, in: TeraGrid'10, Pittsburgh, PA, USA, 2010.
[5] M. Berzins, J. Schmidt, Q. Meng, A. Humphrey, Past, Present and Future Scalability of the Uintah Software, in: Blue Waters Workshop, Chicago, IL, USA, 2012.
[6] Q. Meng, J. Luitjens, M. Berzins, Dynamic Task Scheduling for the Uintah Framework, in: 3rd IEEE Workshop on Many-Task Computing on Grids and Supercomputinger (MTAGS10), 2010.
[7] T. Harman, J. Guilkey, B. Kashiwa, J. Schmidt, P. McMurtry, An eulerian-lagrangian approach for large deformation fluid-structure interaction problems, part 2: Multi-physics simulations within a modern computational framework, in: Fluid Structure Interationcs II, WIT Press, Cadiz, Spain, 2003.
[8] J. Guilkey, T. Harman, B. Kashiwa, J. Schmidt, An eulerian-lagrangian approach for simulating large deformation fluid structure interaction problems, part 1: Algorithm development, in: Fluid Structure Interationcs II, WIT Press, Cadiz, Spain, 2003.
[9] B. Kashiwa, A multi field model and method for fluid-structure interaction dynamics, Tech. Rep. LA-UR-01-1136, Los Alamos National Laboratory, Los Alamos, New Mexico (2001).
[10] D. Sulsky, S. Zhou, H. Schreyer, Applicaiton of particle-in-cell method to solid mechanics, Computer Physics Communications 87 (1995) 236–252.
[11] J. Dick, Measurements of the Shock Initiation Sensitivity of Low Density HMX, Combustion and Flame 54 (1983) 121–129.
[12] J. Dick, Stress-Time Profiles in Low Density HMX, Combustion and Flame 67 (1987) 257–262.
[13] S. Sheffield, R. Gustavsen, R. Alcon, Shock Initiation Studies of Low Density HMX Using Electromagnetic Particle Velocity and PVDF Stress Gauges, in: 10th International Detonation Symposium, Boston, MA, 1993.
[14] S. S. R.L. Gustavsen, R. Alcon, Low Pressure Shock Initiation of Porous HMX for Two Grain Size Distributions and Two Densities, in: 1995 APS Topical Conference on "Shock Compression of Condensed Matter", American Physics Society, Seattle, WA, 1995.
[15] D. Steinberg, S. Cochran, M. Guinan, A constitutive model for metals applicable at high-strain rate, Journal of Applied Physics 51 (3) (1980) 1498–1504.
[16] D. Benson, P. Conley, Eulerian finite-element simulations of experimentally acquired HMX Microstructures, Modelling and Simulation in Materials Science and Engineering 7 (1999) 333–354.
[17] T. Sewell, R. Menikoff, Complete Equation of State for Beta-HMX and Implications for Initiation, Shock Compression of Condensed Matter (157-160).
[18] R. Menikoff, T. Sewell, Constitutent Properties of HMX Needed for Meso-Scale Simulations, Combustion Theory and Modeling 6 (2002) 103–125.
[19] K. Gonthier, R. Menikoff, S. Son, B. Asay, Modeling Energy Dissipation Induced by Quasi-Static Compaction of Granular HMX, in: Schmidt/Danderkar/Forbes (Ed.), Shock Compression of Condensed Matter, 1997, pp. 289–292.
[20] M. Ward, S. Son, M. Brewster, Steady Deflagration of HMX With Simple Kinetics: A Gas Phase Chain Reaciton Model, Combustion and Flame 114 (1998) 556–568.
[21] R. Menikoff, Detonation Wave Profile in PBX-9501, in: 14th APS Topical Conference on Shock, American Physics Society, APS, 2005.
[22] J. B. S.G. Bardenhagen, D. Sulsky, Shear Deformation in Granular Materials, in: 11th International Detonation Symposium, Snowmass, CO, 1998.
[23] Q. Meng, M. Berzins, Scalable large-scale fluid-structure interaction solvers in the uintah framework via hybrid task-based parallelism algorithms, SCI Technical Report UUSCI-2012-004, SCI Institute, University of Utah (2012).