SANDIA REPORT

SAND2015-8355
Unlimited Release
Printed September, 2015

Verification and Large Deformation Analysis Using the Reproducing Kernel Particle Method

Frank Beckwith

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2015-8355
Unlimited Release
Printed September, 2015

Verification and Large Deformation Analysis Using the Reproducing Kernel Particle Method

Frank N. Beckwith
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0840

Abstract

The reproducing kernel particle method (RKPM) is a meshless method used to solve general boundary value problems using the principle of virtual work. RKPM corrects the kernel approximation by introducing reproducing conditions which force the method to be complete to arbitrary order polynomials selected by the user. Effort in recent years has led to the implementation of RKPM within the *Sierra/SM* physics software framework. The purpose of this report is to investigate convergence of RKPM for verification and validation purposes as well as to demonstrate the large deformation capability of RKPM in problems where the finite element method is known to experience difficulty. Results from analyses using RKPM are compared against finite element analysis. A host of issues associated with RKPM are identified and a number of potential improvements are discussed for future work.

Acknowledgment

Much of the work presented in this document would not have been possible this summer without the contribution from multiple individuals. I would like to thank Eliot Fang for affording me the opportunity to come to Sandia and work on problems using the Sierra/SM application. Joseph Bishop provided a great wealth of insight regarding simulations and further steps to take RKPM. David Littlewood worked diligently with me and helped me with some of the RKPM input decks within Sierra/SM to jumpstart my understanding of the Sierra framework. Jim Cox gave me invaluable advise and best practices when performing the convergence study on the wave propagation example. The Solid Mechanics and Shock Physics departments in Org. 1550 was supportive, especially when sitting through presentations as I attempted to explain the RKPM formulation in a way that made sense. The team members of Org. 1542 were pleasant to sit with and also to engage in numerous, enlightening discussions. Last, but not least, my research group at UCSD including Professor J.S. Chen, Mike Hillman, Edouard Yreux, Jacob Koester, and Ivana Escobar also helped prepare me for my internship here at Sandia.

Contents

List of Figures

List of Tables

Preface

Frank Beckwith is currently a doctoral student at the University of California, San Diego and is expected to graduate in June, 2018. His interests are in computational mechanics with emphasis on meshless methods such as the reproducing kernel particle method under the supervision of Professor J.S. Chen. He also has experience in other numerical methods such as finite difference and spectral transforms and is familiar with structural dynamics. He completed a Master of Science in Structural Engineering in 2014, also at UCSD, where he worked on analyzing the seismic performance of reinforced concrete type bridges.

The purpose of this internship was to run verification problems using the reproducing kernel particle method currently implemented in *Sierra/SM* and identify current issues and bugs. Additional tasks were to create and run demonstration examples for comparison with standard finite element analysis.

Nomenclature

BC Boundary condition

BVP Boundary value problem

FEA Finite element analysis

MLS Moving least squares

RDA Rapid design to analysis

RK Reproducing kernel

RKPM Reproducing kernel particle method

SCNI Stabilized conforming nodal integration

SNNI Stabilized non-conforming nodal integration

SPH Smoothed particle hydrodynamics

VCSNNI Variationally consistent stabilized non-conforming nodal integration

Chapter 1

Introduction

The reproducing kernel particle method (RKPM) is a meshless method used to solve general boundary value problems (BVP) using the principle of virtual work. The method uses a corrected kernel function with compact support constructed such that partition of unity and first-order consistency within the approximation space are recovered. RKPM corrects the kernel approximation by introducing reproducing conditions which force the method to be complete to arbitrary order polynomials selected by the user. The method can be viewed as a correction to the kernel used in smoothed particle hydrodynamics (SPH), another meshless method which in general does not have the aforementioned properties. This lends convergence to the RKPM method and provides RKPM the ability to model rigid body motions for a body.

RKPM has a host of advantages over convential methods such as finite element analysis (FEA). RKPM is approximated using a discrete kernel convolution with a corrected kernel. This causes the reproducing kernel (RK) shape functions to have continuity equal to the continuity present in the kernel [3]. The continuity of the RK approximation is independent of the basis order of approximation. This allows the user to model certain classes of problems using RKPM which demand a certain order of continuity whereas FEA is limited to C^0 continuous functions by design. In addition, the RKPM method does not require elements and element connectivity. This obviates the issue of mesh entanglement and unsatisfactory elements associated with FEA. This also enables rapid design to analysis (RDA). However, RKPM involves inverting a moment matrix at each evaluation point of the shape function, making the evaluation of shape functions slower than FEA. Although currently RKPM is implemented in a Lagrangian formulation, the analysis is still considerably slow compared to FEA even though the shape function evaluation is performed only once. In addition, RKPM discards surface information, making it difficult to implement Neumman boundary conditions (BC) and contact type problems.

Effort in recent years has led to the implementation of RKPM within the *Sierra/SM* physics software framework. Previous work by Giffin [7] examined the convergence of RKPM. Additional functionality has since been added. The purpose of this report is to reinvestigate convergence of RKPM for verification and validation purposes as well as to demonstrate the large deformation capability of RKPM in problems where FEA is known to experience difficulty. Results from analyses using RKPM are compared against finite element analysis. A host of issues associated with RKPM are identified and a number of potential improvements

are discussed for future work.

This report is organized as follows. Chapter 2 briefly discusses the implementation of RKPM within *Sierra/SM*, including an overview of the RK shape function formulation and general guidelines for including RKPM in a *Sierra/SM* input deck. Chapter 3 discusses a number of example problems run using RKPM for verification and validation. A large deformation example is simulated with a foam crush problem and is discussed in Chapter 4. Future goals for the coming fiscal year are proposed for RKPM in Chapter 5 and includes a number of capabilities necessary to meet future milestones. Chapter 6 elaborates on a number of issues currently unresolved in RKPM analysis. Intermediate measures which currently exist are discussed and preferred solutions are proposed.

Chapter 2

RKPM Implementation

As mentioned previously in Section 1, RKPM does not utilize a mesh or mesh connectivity in determining its shape functions. This creates significant complexity in formulating the shape function, and is reviewed here. After the formulation is explained, a description for including RKPM within an *Sierra/SM* analysis is given, with some general guidelines to ensure the method runs correctly.

Overview of RKPM

Given a set of NP points in the closed domain, $\overline{\Omega} \subset \mathbb{R}^d$, the approximation function is given by

$$u^h(\boldsymbol{x}) = \sum_{I=1}^{NP} \psi_I(\boldsymbol{x}) u_I \tag{2.1}$$

where ψ_I represents the RK shape function at node I, x_I is the physical coordinate of that node, and u_I is the scalar coefficient which multiplies with the shape function. It should be noted that u_I is *not* the nodal displacement as in FEA as the RK shape functions do not possess the kronecker delta property in general. As such, it is termed the "generalized coordinate" in RKPM [3].

The shape function is given by the following equation

$$\psi_I(\boldsymbol{x}) = C(\boldsymbol{x}; \boldsymbol{x} - \boldsymbol{x}_I)\phi_a(\boldsymbol{x} - \boldsymbol{x}_I) \tag{2.2}$$

Here, $C(\boldsymbol{x}; \boldsymbol{x} - \boldsymbol{x}_I)$ represents a correction function, which is responsible for recovering the polynomial completeness that is not present by simply using the kernel itself as in SPH. It is typically composed of a set of polynomial basis functions and their coefficients. It takes the form below.

$$C(\boldsymbol{x}; \boldsymbol{x} - \boldsymbol{x}_I) = \sum_{i+j+k=0}^{n} (x_1 - x_{1I})^i (x_2 - x_{2I})^j (x_3 - x_{3I})^k b_{ijk} \tag{2.3}$$

$$\equiv \boldsymbol{H}^T(\boldsymbol{x} - \boldsymbol{x}_I)\boldsymbol{b}(\boldsymbol{x}) \tag{2.4}$$

$$\boldsymbol{H}^T(\boldsymbol{x} - \boldsymbol{x}_I) = \begin{bmatrix} 1 & x_1 - x_{1I} & \dots & x_3 - x_{3I} & (x_1 - x_{1I})^2 & \dots & (x_3 - x_{3I})^n \end{bmatrix} \tag{2.5}$$

Substituting the above into Equation 2.2 produces

$$\psi_I(\boldsymbol{x}) = \boldsymbol{H}^T(\boldsymbol{x} - \boldsymbol{x}_I)\boldsymbol{b}(\boldsymbol{x})\phi_a(\boldsymbol{x} - \boldsymbol{x}_I) \tag{2.6}$$

The above equations follow a moving least squares (MLS) methodology. In fact, by only using polynomial basis functions in $\boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}_I)$ above, then the RKPM method is equivalent to MLS approximation. However, the basis vector may be "enriched" with more functions besides simple polynomials [9], which is beyond the scope of this document.

All that is left is to determine the coefficient vector, $\boldsymbol{b}(\boldsymbol{x})$. To do this, we assert that the shape functions have polynomial completeness, given below.

$$\sum_{I=1}^{NP} \psi_I(\boldsymbol{x}_I) x_{1I}^i x_{2I}^j x_{3I}^k = x_1^i x_2^j x_3^k, \quad 0 \leq i + j + k \leq n \tag{2.7}$$

The polynomial completeness may equivalently be written as [3]

$$\sum_{I=1}^{NP} \psi_I(\boldsymbol{x}_I)(x_1 - x_{1I})^i (x_2 - x_{2I})^j (x_3 - x_{3I})^k = \delta_{0i}\delta_{0j}\delta_{0k} \tag{2.8}$$

$$\sum_{I=1}^{NP} \psi_I(\boldsymbol{x}_I)\boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}_I) = \boldsymbol{H}(\boldsymbol{0}) \tag{2.9}$$

The above is termed the reproducing condition for the approximation function and reproduces the polynomial completeness in the basis space. Taking Equation 2.6 and substituting into the above produces

$$\sum_{I=1}^{NP} \boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}_I)\boldsymbol{H}^T(\boldsymbol{x} - \boldsymbol{x}_I)\boldsymbol{b}(\boldsymbol{x})\phi_a(\boldsymbol{x} - \boldsymbol{x}_I) = \boldsymbol{H}(\boldsymbol{0}) \tag{2.10}$$

$$\boldsymbol{M}(\boldsymbol{x})\boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{H}(\boldsymbol{0}) \tag{2.11}$$

where

$$\boldsymbol{M}(\boldsymbol{x}) = \sum_{I=1}^{NP} \boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}_I)\boldsymbol{H}^T(\boldsymbol{x} - \boldsymbol{x}_I)\phi_a(\boldsymbol{x} - \boldsymbol{x}_I) \tag{2.12}$$

This gives us an expression to solve for the coefficient vector, $\boldsymbol{b}(\boldsymbol{x})$. Inverting this expression and substituting back into Equation 2.6 gives

$$\psi_I(\boldsymbol{x}) = \boldsymbol{H}^T(\boldsymbol{0})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}_I)\phi_a(\boldsymbol{x} - \boldsymbol{x}_I) \tag{2.13}$$

After having finally arrived at the formulation of the shape function, it may be used directly to approximate the solution space and employed in the principle of virtual work, just as with the FEA. As such, the definitions for the deformation gradient, $\boldsymbol{F}$, rate of deformation, $\boldsymbol{D}$, and velocity gradient, $\boldsymbol{L}$, are all similar to FEA, where only the derivatives of the RK shape functions are required. Once these values are determined, they can be passed as inputs to the *LAME* material library present in *Sierra/SM* just as one would with FEA. Although

this works in theory, there are some materials which require additional parameters that are not currently handled in RKPM. However, the large majority of material models in the production code work with RKPM.

Although the definition of materials is straightforward with RKPM, the imposition of essential, or Dirichlet BCs is not. This is due to the loss of the kronecker delta property in the RK shape function, i.e.

$$\psi_I(\boldsymbol{x}_J) \neq \delta_{IJ} \tag{2.14}$$

Because of this, in order to determine the physical displacement or other quantity at the nodes, a full summation must be performed over the shape functions whose support covers that node.

$$\sum_{I=1}^{NP_x} \psi_I(\boldsymbol{x}_I)\boldsymbol{u}_I = \hat{\boldsymbol{u}}_I \tag{2.15}$$

where above $\hat{\boldsymbol{u}}_I$ is the true nodal displacement in the RK approximation and NP_x is the number of nodes whose support covers the evaluation point. For this reason, $\boldsymbol{u}_I$ is termed the "generalized coordinate." To impose displacement boundary conditions, the system of equations must be modified either by using a transformed method where the boundary conditions are applied via Lagrange multipliers manipulated in order to simplify the system or by recovering the kronecker delta property at the boundary through the use of the boundary singular kernel method [4].

An example of the RK shape functions in one dimension and a demonstration of quadratic completeness is illustrated in Figure 2.1. The shape functions are generated using quadratic basis with a C^2 continuous spline for a kernel. A non-uniform node spacing was employed to demonstrate the flexibility of the method under non-uniform discretizations. One can see the shape functions do not exhibit the kronecker delta property given by Equation 2.14.

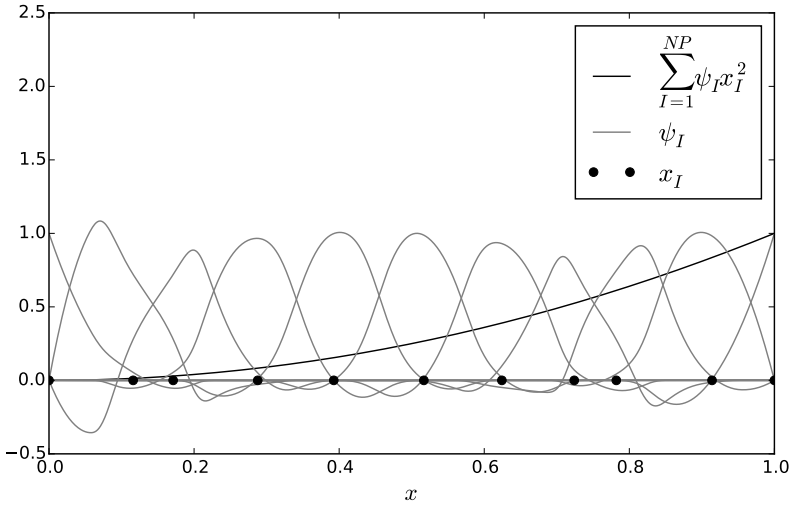


Figure 2.1: Shape function illustration

An additional and major complication involves the integration of the domain required in the variational approach. Because the element connectivity is effectively discarded, Gauss-Legendre quadrature is no longer straightforward. In addition, the use of Gauss-Legendre quadrature will introduce error into the integration because of the irregular nodal domains and because the MLS procedure produces shape functions that are rational instead of polynomial expressions. Use of Gauss-Legendre quadrature will cause RKPM to fail the patch test, even with a large number of integration points [5]. To remedy this issue, nodal integration procedures are employed including stabilized nodal conforming integration (SCNI), stabilized non-conforming nodal integration (SNNI) and variationally consistent stabilized non-conforming nodal integration (VCSNNI) [2, 5]. These methods perform a strain smoothing by constructing integration cells with so-called integration constraints imposed on them. The explanation of these methods is outside the scope of this document.

RKPM Input within *Sierra/SM*

Converting a *Sierra/SM* input deck to an RKPM analysis is a rather straightforward process. Only two additional blocks need to be included. The first defines the section and RKPM paramaters for use in the analysis. It is defined in the "Sierra" scope of the input deck.

```
BEGIN RKPM SECTION <string> section_name
    SUPPORT SIZE = <real> normalized_support
    BASIS ORDER = <int>order
    KERNEL TYPE = <int>kernel_type
    KERNEL SHAPE = <string>SPHERE | BOX
    INTEGRATION METHOD = <string>SCNI | SNNI | VCSNNI
    STABILIZATION COEFFICIENT = <real> coefficient
    FORMULATION = <string> LAGRANGIAN| SEMI-LAGRANGIAN
END RKPM SECTION <string> section_name
```

The options above must be defined with care in order for the RKPM analysis to run without errors.

- Support Size: The support size is a scale factor which multiplies with the distance to the nearest neighbor of an RK node. This multiplied distance becomes the support size for the RK node in the discretized domain, $\overline{\Omega}$, and may have a different support length in each cardinal direction. For linear basis order, a scale factor of 1.6 is determines the number of neighbors used to evaluate the shape function. The scale factor must increase by 1 for each increase in the basis order. For example, if quadratic basis is used instead, the scale factor should be increased to 2.6.

- Basis Order: The basis order determines the order of completeness of the basis functions. This may be chosen to an arbitrary order by the user. However, if the basis

order is chosen higher than 1, then the SCNI integration scheme cannot be used and the VC-SNNI integration scheme must be used instead for convergence [2]. Therefore, linear basis functions are recommended, which also have the advantage of increasing the speed of analysis.

- Kernel Type: The reproducing conditions described in Chapter 2 are automatically enforced within the RKPM code, but the user is free to choose the type of kernel used in the approximation. Up to 6 kernels exist; an input of 0 uses a C^0 linear B-spline whereas an input of 5 uses a C^4 quintic B-spline for a kernel.

- Kernel Shape: Kernels currently only have rectangular support in *Sierra/SM*. Ellipsoidal support is planned for future implementation. This parameter defaults to "BOX," regardless of user input.

- Integration Method: As mentioned before, this determines the nodal integration method used to evaluate the internal forces in the RKPM formulation. Three options exist: SCNI, SNNI, and VCSNNI. Currently, the RKPM method is implemented with a Lagrangian formulation. As a result, RKPM only uses the referential coordinate in its calculations, and the SCNI method of integration is recommended here.

- Stabilization Coefficient: This parameter populates the integration cell used above with additional evaluation points used for stabilization. This value ranges between 0 and 1, with 0 meaning that evaluation points are added but no stabilization occurs, adding time to the total RKPM analysis. As such, if no stabilization is required, this option should be omitted. The use of stabilization in the RKPM section of *Sierra/SM* does not always work and needs to be inspected for potential bugs.

- Formulation: Currently, the RKPM method is only implemented in a Lagrangian formulation. As such, the analysis defaults to "Lagrangian," regardless of user input. Semi-Lagrangian (which is similar to Updated Lagrangian) is planned for the next fiscal year.

Once the section has been defined, the particles need to be created. This can be accomplished using the built-in function for converting particles from an FEA block. The following block is included in the "Presto" scope of the input deck.

```
BEGIN CONVERSION TO PARTICLES AT INITIALIZATION
    BLOCK = <string list>block_names
    SECTION = <string>particle_section
END [CONVERSION TO PARTICLES AT INITIALIZATION]
```

Alternatively, one may use the `spheregen.py` script and can be invoked in the terminal using

```
spheregen.py --nodes_as_attr myfile.g
```

One final option that requires clarification is the definition of contact with RKPM. Currently, the contact surfaces required for the dash contact are implemented by lofting icosahedra (a

faceted sphere with 20 faces) around each RK particle. Each icosahedron is sized such that the volume of the lofted sphere is the same as the volume of the integration cell of the RK node. This produces a non-contiguous contact surface for the RKPM block and gaps may form should the material experience significant deformation. As such, defining contact between two RKPM blocks is not recommended as the two blocks may "slip" through one another.

In addition, *all* of the RK nodes become lofted when contact is activated. No option exists to convert only the nodes located on the contact surface. This adds significant time to the analysis, as the method will check for contact on interior nodes where it is not necessary. This also complicates self-contact because the lofted spheres may overlap. Should this be the case (as it almost always is), then an instability will result. This is not recommended, as the self-contact may not even work due to the "slip-through" mentioned above. In addition, in the rare case that none of the icosahedra overlap with each other, then in problems of compression the interior nodes will potentially contact one another and the material will cease to deform because the icosahedra themselves are not compressible.

Contact in RKPM is most easily modeled when the contact definition is with a FEA block, which does have a continuous contact surface. For contact to work in this scenario, the RKPM block must be sized such that the resulting icosahedra produced from lofting do not intersect the contact surface in the FEA analysis. If this occurs, then the dash contact algorithm will attempt to remove this interpenetration, which will cause an instability on the very first time step. Sizing the block during the preprocessing stage must then be done with care and requires some user judgement, as it is not possible to determine the size of the spheres before running the analysis. (A manual sizing option exists for lofted objects in *Sierra/SM* but on trying this in RKPM, it did not appear to work correctly.)

Chapter 3

Verification and Validation

A number of problems were analyzed using the RKPM implementation. These include wave propagation in a bar, taylor bar impact, and a traction patch test. Details and results of each analysis are given in the sections that follow.

Wave Propagation in a Bar

In this problem, a shock wave is induced in an elastic bar. The bar measures $2\,\text{cm} \times 0.02\,\text{cm} \times 0.02\,\text{cm}$ and is fixed on the left side. An initial velocity of $1000\,\text{cm/s}$ is imposed on the entire body. The bar uses an elastic material with a Young's modulus of $E = 300\,\text{GPa}$, a Poisson's ratio of $\nu = 0$, and a density of $\rho = 7.8\,\text{g/cm}^3$. A zero Poisson's ratio is used here to limit the wave propagation to the axial direction only, effectively making the analysis a 1D wave propagation problem.

Due to the fixed boundary condition on the left side of the bar, a compression wave with a vertical front is introduced and travels the length of the bar. Once the wave reaches the free end, it rebounds and the bar begins to travel in the opposite direction and the stress behind the wave goes to zero. After reaching the wall, the wave again rebounds, this time creating a tension wave, and so forth. The exact displacement is given by a Fourier series representation as

$$u_1(x,t) = \sum_{n=1}^{\infty} \left(A_n \sin\left(\omega_n t\right) \sin\left(\frac{(2n-1)\pi}{2L} x \right) \right) \tag{3.1}$$

$$A_n = \frac{8 v_0 L}{(2n-1)^2 \pi^2} \sqrt{\frac{\rho}{E}} \tag{3.2}$$

$$\omega_n = \frac{(2n-1)\pi}{2L} \sqrt{\frac{E}{\rho}} \tag{3.3}$$

This problem was discretized and run using RKPM analysis. Linear basis functions were employed and a C^4 continous spline was used for the kernel. Three separate discretizations were considered. The coarse discretization used 164 nodes, the medium discretization used 1449 nodes, and the fine discretization used 31 409 nodes. The displacement at the center of the bar for the RKPM approximation as well as the exact displacement is shown in Figure 3.1.

The displacements can be seen to converge to the exact solution. At the coarse discretization, large oscillations propagate through the solution, whereas for the fine discretizations these oscillations are almost completely removed.

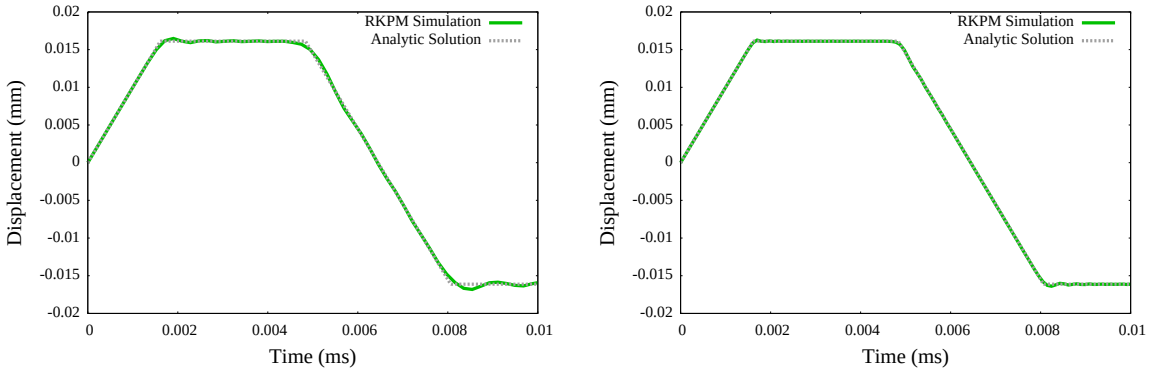
A convergence study in the energy norm was carried out. As described above, a material point is characterized by two states: one in which velocity is present and stress is not and vice versa. The kinetic energy present in the system may be determined by the length of actively moving material. Conversely, the internal energy is just the opposite, characterized by the length of strained material instead. These expressions for the exact energy in the system are thus

$$KE = \frac{1}{2}\rho x_{wave} A v_0^2 \tag{3.4}$$

$$IE = \frac{1}{2}\rho \left(L_{bar} - x_{wave}\right) A v_0^2 \tag{3.5}$$

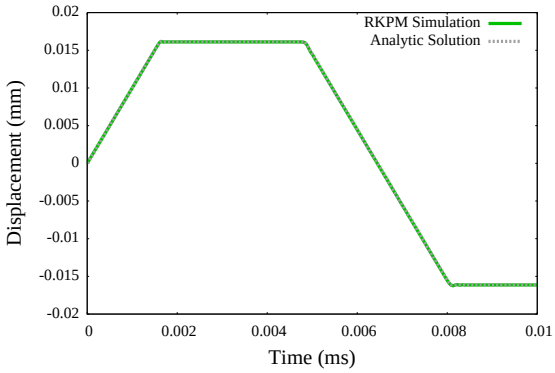
$$x_{wave} = ct \tag{3.6}$$

$$c = \sqrt{E/\rho} \tag{3.7}$$



(a) Coarse discretization

(b) Medium discretization



(c) Fine discretization

Figure 3.1: Wave propagation measured at center of bar

The energy in the approximate solution requires some additional work. Because the SCNI method was used, the strains and stresses are rendered constant within each integration cell [5]. Thus, to measure the internal energy of the approximate solution was as simple as taking the scalar product between the stress and strain tensors and multiplying them by the volume of the integration cell.

$$IE^h = \frac{1}{2}\sum_{I=1}^{NP} \boldsymbol{E}_I^h : \boldsymbol{S}_I^h V_I \tag{3.8}$$

where in the above, $\boldsymbol{E}_I^h$ is the Green-Lagrange strain tensor, $\boldsymbol{S}_I^h$ is the second Piola-Kirchhoff stress tensor, and V_I is the volume of the integration cell. These two expressions may be used to determine the energy of the error between the solutions.

$$a(\boldsymbol{e}, \boldsymbol{e}) = a(\boldsymbol{u}, \boldsymbol{u}) - a(\boldsymbol{u}^h, \boldsymbol{u}^h) \tag{3.9}$$

where

$$a(\boldsymbol{u}, \boldsymbol{u}) = 2IE \tag{3.10}$$

$$a(\boldsymbol{u}^h, \boldsymbol{u}^h) = 2IE^h \tag{3.11}$$

$$\boldsymbol{e} = \boldsymbol{u} - \boldsymbol{u}^h \tag{3.12}$$

Equation 3.9 is a result of the Galerkin orthogonality property in the principle of virtual work, where $a(\boldsymbol{e}, \boldsymbol{u}^h) = 0$, and is known as the Pythagorean corollary [10]. The energy norm can then be given as

$$\|\boldsymbol{u} - \boldsymbol{u}^h\|_E = \sqrt{a(\boldsymbol{e}, \boldsymbol{e})} \tag{3.13}$$

Using this strategy, the convergence study was carried out. The bar measuring $2\,\text{cm} \times 0.02\,\text{cm} \times 0.02\,\text{cm}$ was discretized and run using RKPM with linear basis and a C^4 continous spline kernel function. Seven discretizations were considered for the analysis where the input file was refined in the axial direction only. Given the Poission's ratio, $\nu = 0$, no refinement in the transverse directions were necessary. The characteristic length was halved for each level of refinement beginning with $h_0 = 0.08\,\text{cm}$, and the critical time step was quartered beginning with $\Delta t_0 = 2.0 \times 10^{-8}\,\text{s}$. A user-time step was supplied and refined faster than the characteristic length in order to isolate the error to the spatial discretization. The energy of the solutions were evaluated when the wave front traveled half of the length of the bar.

Figure 3.2 shows the convergence plot in the energy norm. A convergence rate of $\mathcal{O}(h^{0.5})$ was observed, which is half of the expected convergence rate for the energy norm. This may be due to two possibilities. First, the energies between the two solutions were not integrated the same way, which may introduce inconsistencies in the error and lead to a lower order of convergence. Second, the problem itself is ill-posed. A vertical shock front exists but the approximation method cannot represent a vertical front with a C^4 continuous approximation function. As such, the problem is not guaranteed to converge at the expected rate. However, the system does appear to be converging at a consistent rate.

In addition to a convergence study, a scaling study was carried out. In this study, the wave propagation was repeated but with a total of 360 192 RK nodes. This is approximately

the maximum number of particles before running out of memory on a single compute node on the cluster. The problem was repeated with 1, 2, 4, 8, 16, 32, 64, and 128 nodes. The results of the scaling study are illustrated in Figure 3.3, with an approximate slope of 0.87 in log-log space. An ideal rate of 1 is desired, but communication between processors likely leads to a slightly lower rate. Runs with more processors than 128 begin to experience diminishing returns. This suggests a single processor can accomodate roughly 2500-3000 nodes as a general guideline for selecting the number of processors.

Taylor Bar Impact

To verify the contact algorithm, a Taylor bar impact problem was investigated. This problem attempts to replicate the experiments given in [15] where various steel and aluminum rods were launched at high velocities and impacted against a rigid wall. The experiment measured the final height of the specimens and compared them to the initial height.

The experiments described above were simulated using the RKPM method. Two separate analyses were performed: one where contact was simulated by fixing the vertical direction of the impacting face of the bar; a second where contact surfaces were defined between the RKPM block and a rigid wall.

The Taylor bar problem using RKPM simulated a bar made of steel with a radius of 0.381 cm and a height of 1.27 cm. An initial velocity of 282 cm/s was used for the analysis. The Johnson-Cook material model was employed to model the material response. The Johnson-Cook material model has a yield stress that is strain dependent and temperature dependent. The yield stress is given by [13]

$$\sigma_{eq} = (\sigma_y + h\bar{\varepsilon}^n)\,(1 + C\ln\dot{\varepsilon}^*)\,(1 - T^{*m}) \tag{3.14}$$

$$\varepsilon^* = \frac{\dot{\varepsilon}}{\dot{\varepsilon}_0} \tag{3.15}$$

$$T^* = \frac{T - T_0}{T_m - T_0} \tag{3.16}$$

where the variables and their values used in the Taylor bar simulation is outlined below in Table 3.1. Temperature is updated using adiabatic heating, where a percentage of the strain energy due to plastic deformation is converted to heat. By default, the percentage converted is 95 %.

Illustrations for the taylor bar impact are given in Figure 3.4. Figure 3.4a shows the taylor bar modeled with a fixed bounday condition in the axial direction and Figure 3.4b shows the taylor bar with a contact condition specificied between the RKPM and FEA bodies. Although the model with a fixed boundary condition is sufficient, the model utilizing contact is more realistic of the test conditions. Here, the bar can be seen to flare at the impact end, which was observed in the test specimens [15].

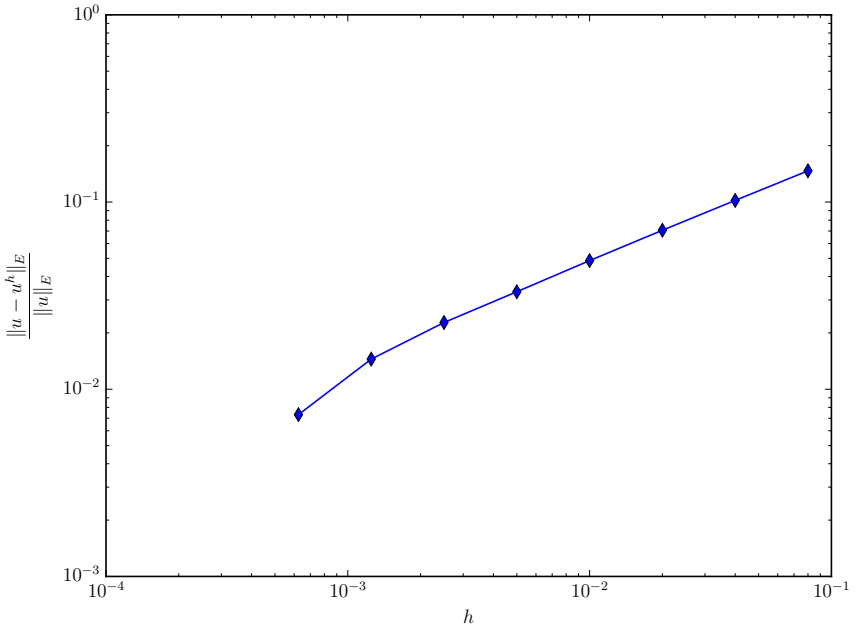


Figure 3.2: Wave propagation convergence

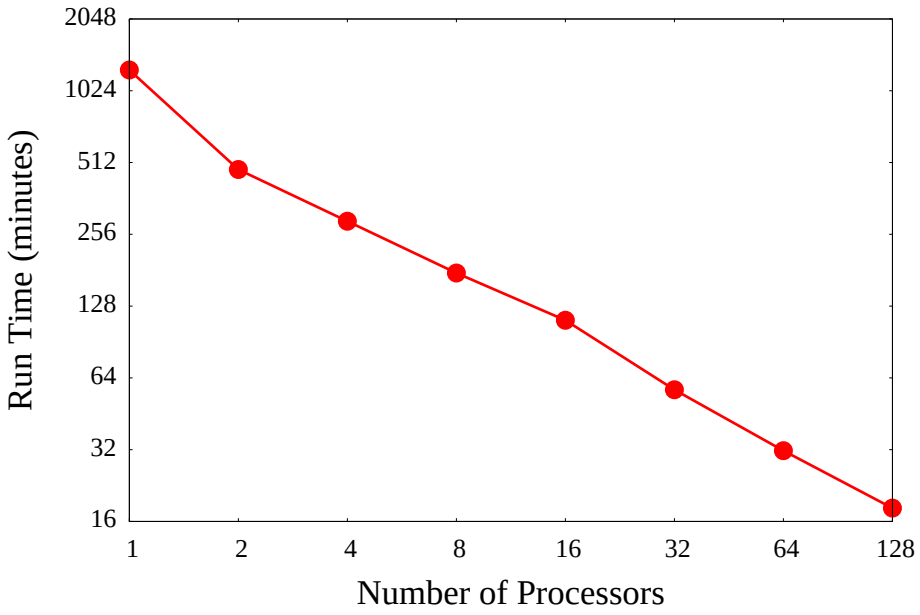


Figure 3.3: Wave propagation scaling study [12]

Table 3.1: Johnson-cook material model parameters for the taylor bar simulation [11, 12]

Parameter	Value
Young's modulus E	200 GPa
Poisson's ratio ν	0.29
Density ρ	7830 kg/m^3
Specific heat capacity C_p	477 J/kgK
Reference temperature T_0	300 K
Melting temperature T_m	1793 K
Temperature softening exponent m	1.03
Yield stress σ_y	792 MPa
Strain hardening coefficient h	510 MPa
Strain hardening exponent n	0.26
Strain rate hardening exponent C	0.014
Reference strain rate $\dot{\varepsilon}_0$	1.0 s^{-1}

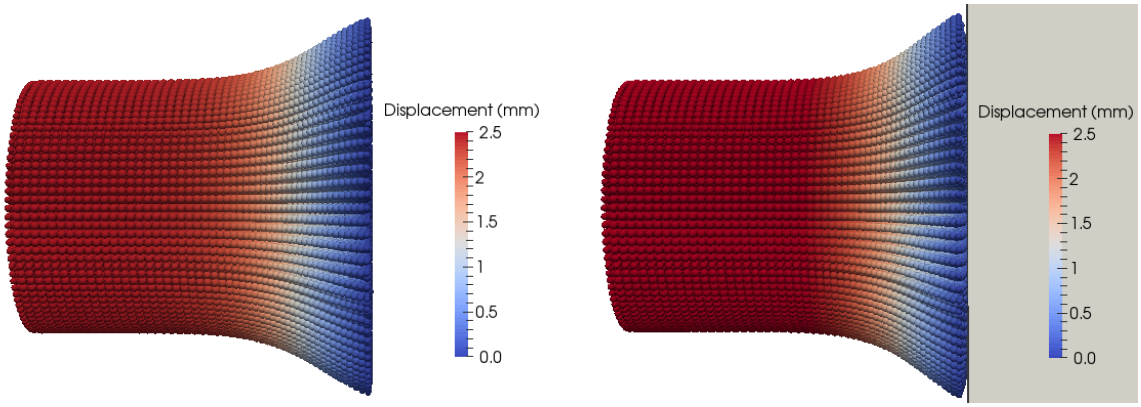
The height of the taylor bar simulation is compared against the experimental height in Figure 3.5. The grey, horizontal line is the measured height from the experiment and the other lines are the predictions given by the simulations. Three discretizations are considered: a coarse discretization with 938 nodes, a medium discretization with 26 488 nodes, and a fine discretization with 46 592 nodes [12]. The medium and fine refinements have a nearly identical response and predict the final height very well. The coarse discretization, while appearing to have less total deformation, also approximates the final height well. These results show heuristically the convergence of RKPM when applied to a nonlinear problem.

Traction Patch Test

Tractions require special attention for RKPM analysis because of the removal of the surface definition when the finite element block is converted to particles. As a result, the RKPM method does not store surface information and the imposition of Neumann type boundary conditions becomes difficult to implement. However, the elements from the original mesh are not completely discarded. The analysis simply copies the necessary information for RKPM but still retains the original finite element mesh from before.

Although RKPM does not include routines to determine external forces due to tractions, it was discovered that tractions may still be applied to the RKPM analysis. The tractions are being integrated using the finite element routines and taking the resulting nodal forces and applying them as forces to the RKPM analysis. This occurs because the original finite element mesh is never truly discarded; it is simply ignored in the majority of RKPM analysis.

With this information in hand, a patch test was performed using traction boundary conditions. A bar with dimensions $3.0 \times 1.0 \times 1.0$ was loaded with tractions in the axial direction at the right end. The prescribed boundary conditions at the left end restricted



(a) Fixed boundary condition

(b) Contact surface

Figure 3.4: Taylor bar impact simulation. Color denotes displacement in the horizontal direction. [12]

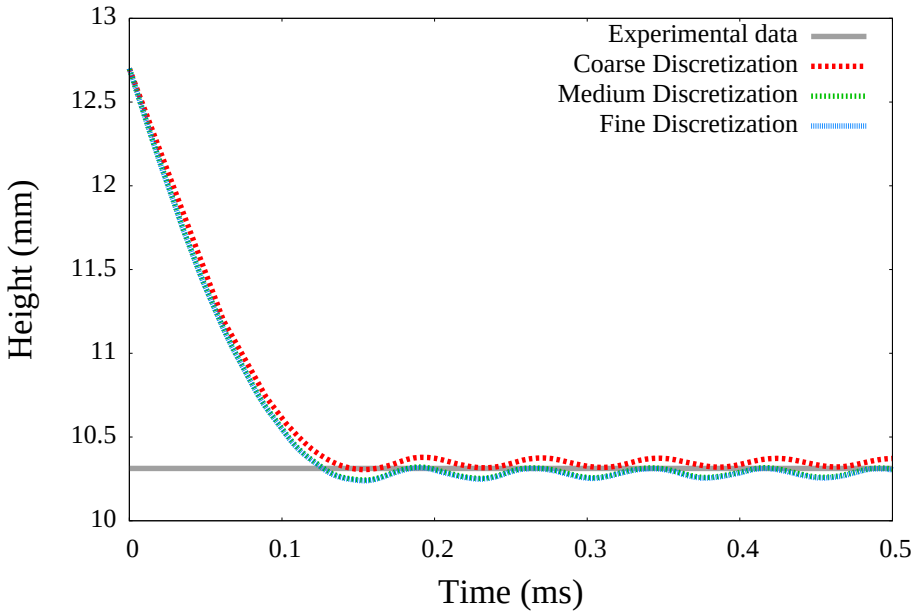


Figure 3.5: Computed final height of taylor bar [12]

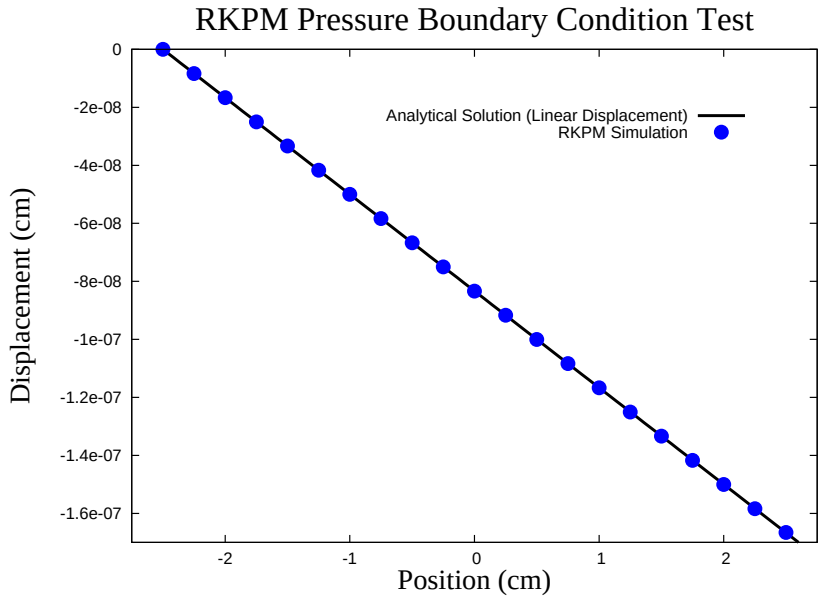


Figure 3.6: Traction patch test of RKPM simulation

movement in the x-direction only to ensure a linear displacement field was achieved in all directions. To remove transient motion from the analysis the explicit analysis employs a halver-cosine ramp function with a duration of 1.0×10^{-3} s to limit the amount of oscillation in the bar.

The result is presented in Figure 3.6. As can be readily observed, the method is satisfying the patch test for the axial direction. However, slight oscillations near the traction boundary condition are still present in the transverse direction. Although the traction loads are not applied consistently with the RKPM formulation, they are still only in the axial direction and should not produce displacement in the transverse direction. Thus, these oscillations give rise for concern and require further investigation. It should be noted, however, that very few methods besides the linear hex element impose traction boundary conditions consistently with the shape functions and the results presented here are acceptable as an intermediate measure.

Chapter 4

Foam Crush Simulation

To demonstrate the large deformation capability of RKPM, a foam-crush simulation was chosen. This work was based off that by Nick Kerschen and his mentor, Dr. Chris Hammeter. In their work, an impactor strikes a foam block made of TufFoam which is contained inside of an aluminum housing. Using FEA, the acceleration experienced in the block was simulated and compared to experimental results.

Due to the compression experienced by the foam block, this problem made for an ideal candidate to verify the large deformation capability of RKPM. Towards that end, the simulation was repeated with the only major difference being the use of RK particles to simulate the foam material as opposed to FEA.

Although the only major difference between the two simulations is the use of RKPM instead of FEA to simulate the foam block, other minor changes had to be made due to complexities involved in simulating contact with RK particles. The model uses only a single RKPM block for the foam and FEA for all other blocks because contact between separate RKPM blocks is not well behaved. In addition, in the original analysis two foam blocks were modeled with a contact definition between them. For the same reason listed before, the two foam blocks were merged and only a single block was considered for RKPM. Finally, the foam block had to be shortened slightly to introduce a gap between it and the surrounding objects. This is due to contact surfaces being lofted around RK particles. If a gap was not introduced, the lofted icosahedra would interpenetrate the surfaces of the FEA blocks, leading to instability on the first time step of analysis.

The material specified for the foam block is the foam plasticity model. Material properties for the foam block are given in Table 4.1. This material defines a shear dependent and pressure dependent yield surface. The law for updating the yield stresses is given by the following equations [14]

$$\sigma_{y,s} = \sigma_s + H_s \phi^{n_s} \tag{4.1}$$

$$\sigma_{y,p} = \sigma_p + H_p \phi^{n_p} \tag{4.2}$$

The evolution of the yield stresses of the foam plasticity model rely on a density parameter, $\phi = \rho/\rho_{solid}$, rather than on the amount of plastic strain. This approximates a cellular material as it approaches the density of the solid block of the same material. Foam materials are expected to experience locking as they approach the solid limit, approximated by using values of n_s and n_p larger than 1. The larger these values, the faster these the foams lock as

they approach the maximum density. [14]

The aluminum casing was modeled using an elastic-plastic material. Perfect plasticity was assumed. Material parameters for the aluminum are given in Table 4.2. The impact block

Table 4.1: Foam plasticity material model parameters for the foam crush simulation.

Parameter	Value
Density ρ	35 pcf
Young's modulus E	66 500 psi
Poisson's ratio ν	0.30
Deviatoric yield strength σ_s	1480 psi
Deviatoric hardening H_s	11 200 psi
Deviatoric exponent n_s	3
Hydrostatic yield strength σ_h	1480 psi
Hydrostatic hardening constant H_h	16 500 psi
Hydrostatic exponent n_h	3
Associative flow parameter β	0.0

Table 4.2: Elastic plastic material model parameters for the foam crush simulation.

Parameter	Value
Density ρ	169 pcf
Young's modulus E	$10\,370 \times 10^3$ psi
Poisson's ratio ν	0.33
Yield stress σ_y	10 878 psi
Hardening coefficient H	0.0 psi
Isotropic parameter β	1.0

utilized an elastic material having a Young's modulus $E = 28\,000 \times 10^3$ psi, a Poisson's ratio of $\nu = 0.27$, and a density of $\rho = 3110$ pcf. The impact block was given an initial velocity of 1116 in/s in the downward direction. The simulations were performed to a maximum time of $t_{max} = 5.0 \times 10^{-3}$ s.

In the figures that follow, the displacement profile as well as the Von Mises stress is observed. Two discretizations are considered: a coarse discretization with 99 664 nodes, illustrated in Figure 4.1; and a fine discretization with 582 884 nodes, illustrated in Figure 4.2. The FEA models simulated for comparison purposes employ similar discretization levels.

As a show of confidence, both methods achieve relatively similar levels of displacement, with some minor differences. Both methods predict a displacement of roughly 1.30 in for the coarse refinement. At the fine refinement, the RKPM method experiences less displacement with a predicted value of 1.27 in whereas the FEA solution remains largely the same. This behavior is curious as variational approaches are expected to converge from the stiff side. Additionally, the stresses in the aluminum cannister are much higher than the foam block, as

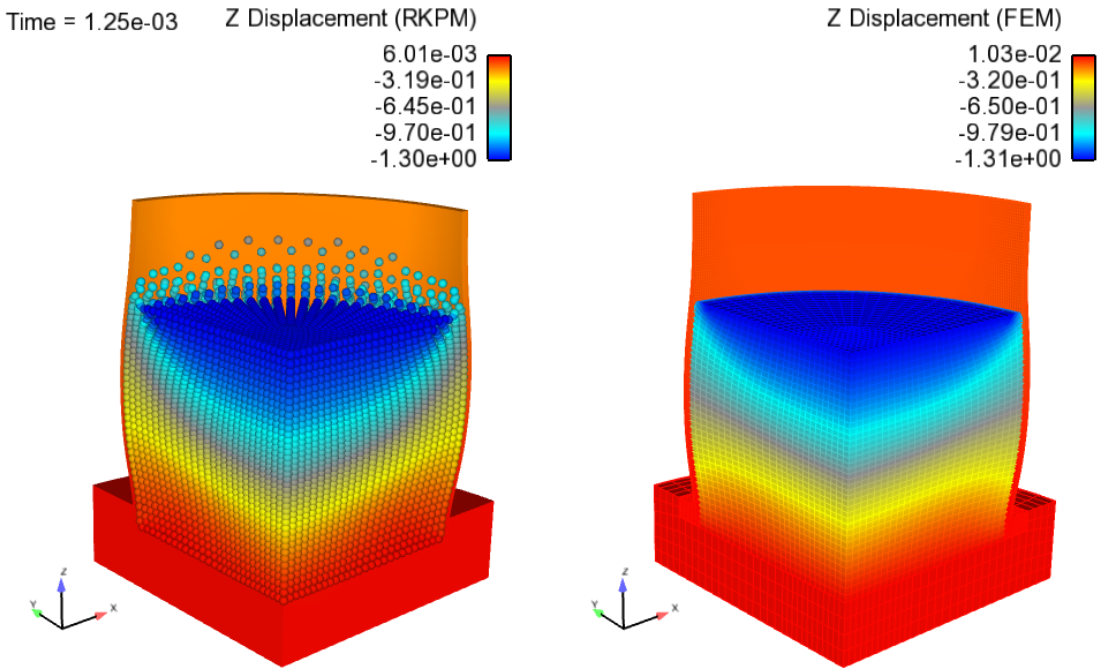
one would expect due to the large difference in material stiffnesses. Both discretizations show the same stress levels. Conversely, the foam experiences much more plastic deformation.

Although the solutions typically agree with each other, the solutions employing RKPM appear to have non-physical behavior. Figure 4.1 shows axisymmetric behavior in the FEA solution whereas the RKPM method appears to have spurious or, more bluntly, wrong results. This may indicate issues and coding errors within the code base and require closer inspection to determine the root of the error. Although less pronounced, the same trend may be observed in Figure 4.2. For the fine discretization, non-physical stress waves may be observed in the aluminum housing, indicating possible instabilities related to RKPM.

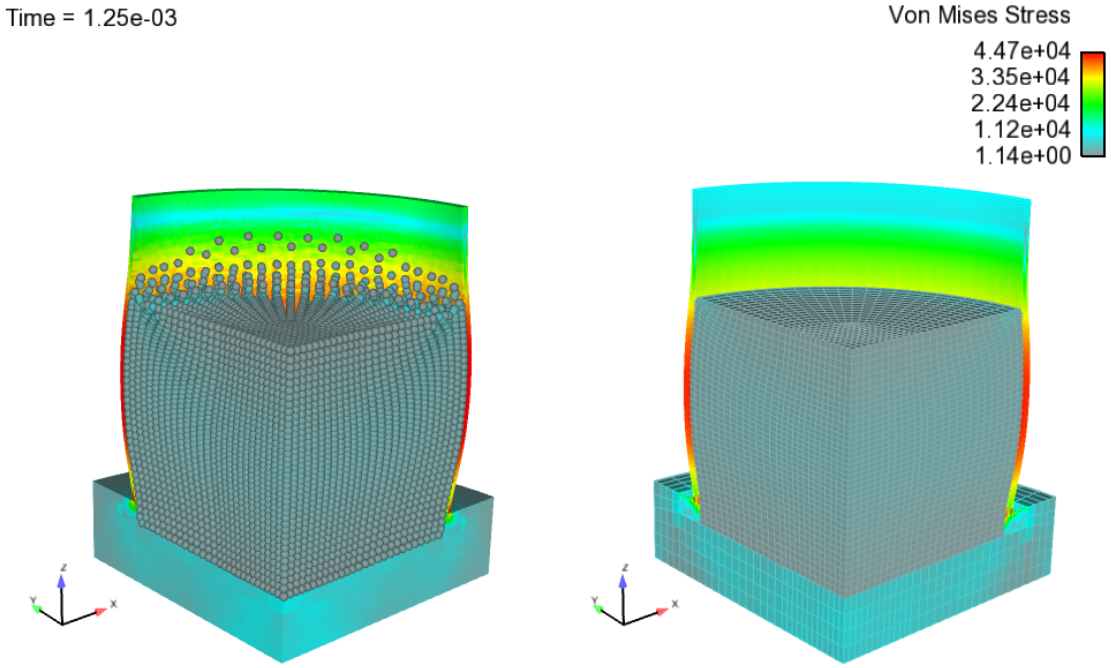
Figure 4.3 compares the deformations between the two meshes localized around the corner of the impactor. Although the FEA solution runs without issue in this example, likely because of the use of uniform-gradient hex elements, it experiences large mesh distortions which introduce error into the analysis. The RKPM analysis does not suffer from this limitation, although the spurious behavior is questionable. The FEA mesh itself may inhibit the foam material from "squeezing" out between the impactor and the aluminum cannister. For the coarse model, the interior node must slip around the corner of the impactor before the material is allowed to deform through the gap as the foam is compressed. In contrast, the RKPM method readily begins to deform between the impactor and the cannister due to the absence of a mesh.

The external energy, kinetic energy, and internal energy were tracked in both the FEA and RKPM analyses. They are shown below in Figure 4.4. It can be observed that both methods produce relatively the same energies for analysis. However, what comes as a slight surprise is the determination of external energy for RKPM. At the beginning of analysis, the external energy of both FEA and RKPM are roughly equal, whereas later on the external energy for RKPM is somewhat below that of FEA. This is unexpected and a potentially indicates a problem within the energy calculation within the code.

During the RKPM analysis, significant oscillation of the critical time step was observed. Because there no longer exists a mesh for which to estimate the time step on an element basis, other methods have to be considered. Instead of considering a characteristic element length, the RKPM uses the support distance instead as well as the wave speed of the material given by the effective lamé parameters and material density. The integration method (SCNI, SNNI, VCSNNI) also affects the critical time step [1, 12]. In the foam crush analysis, significant oscillation could be observed, shown in Figure 4.5.

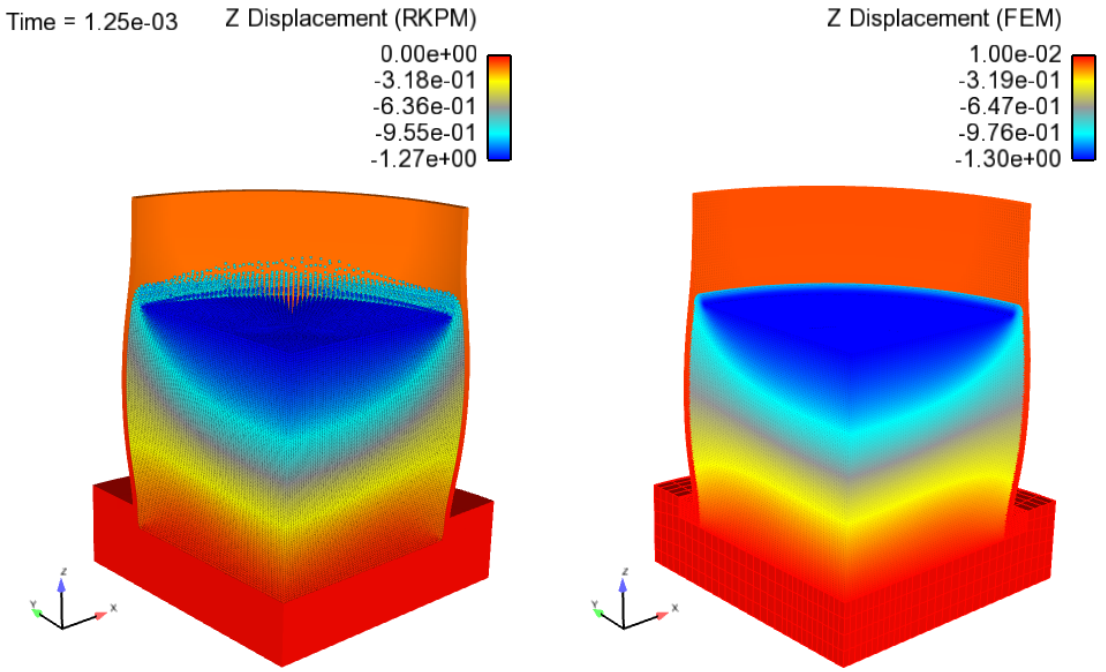


(a) RKPM and FEA displacement comparison

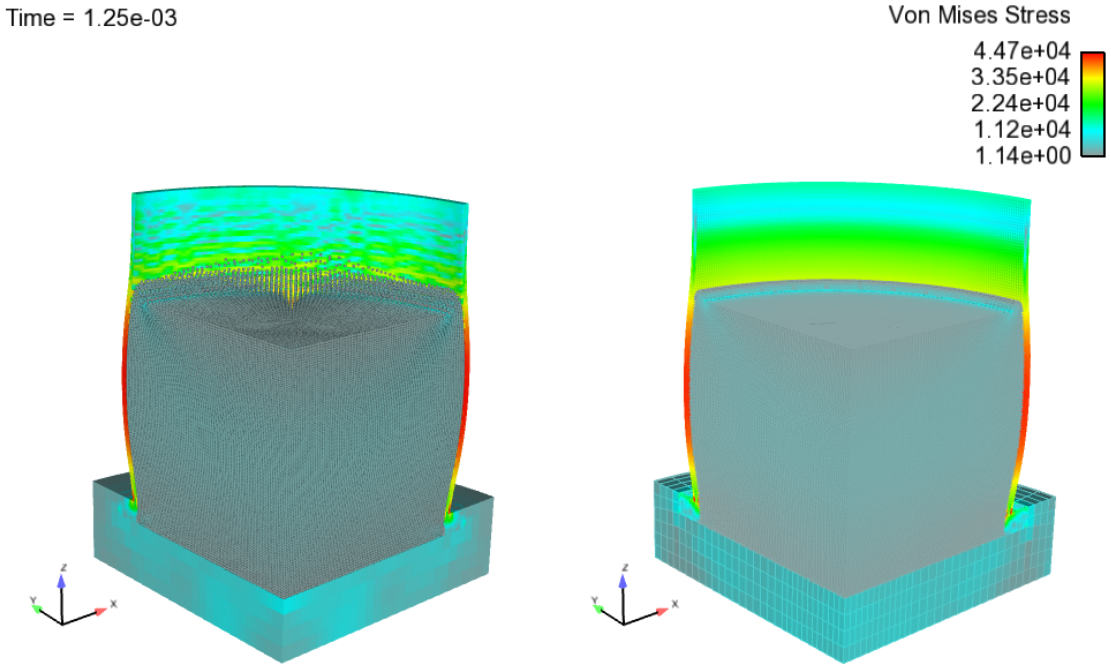


(b) RKPM and FEA stress comparison

Figure 4.1: Foam crush response, coarse discretization (99 664 nodes)



(a) RKPM and FEA displacement comparison



(b) RKPM and FEA stress comparison

Figure 4.2: Foam crush response, fine discretization (582 884 nodes)

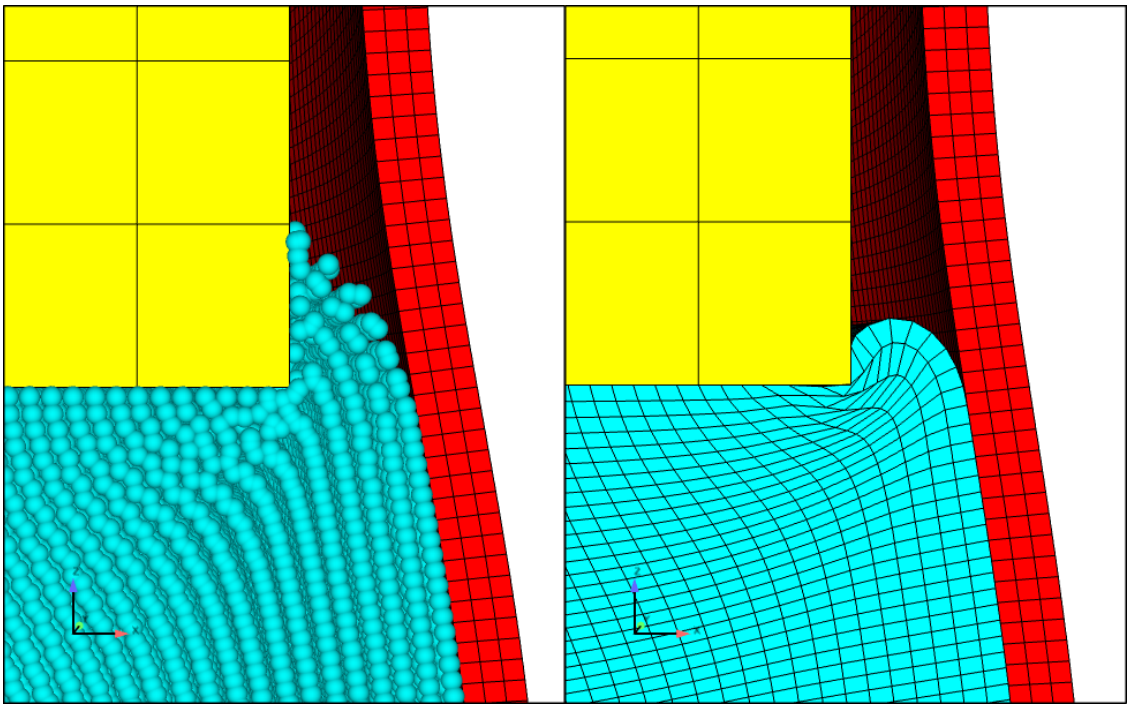


Figure 4.3: Foam crush response at corner of impactor

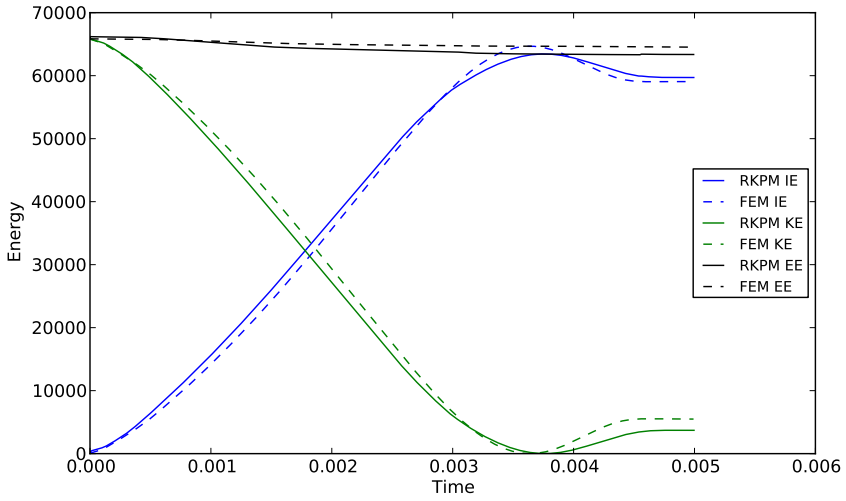


Figure 4.4: Foam crush energy comparison

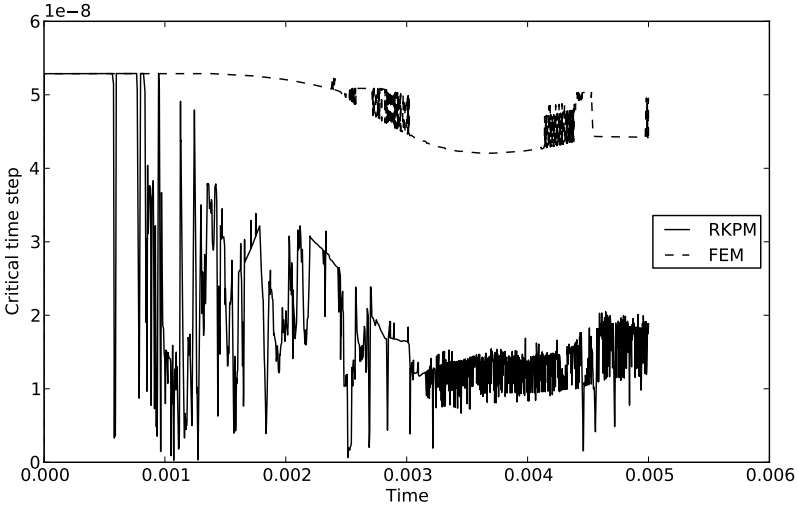


Figure 4.5: Foam crush critical time step

Chapter 5

Future Goals for RKPM

A number of different capabilities are targeted as future goals for the next fiscal year.

Currently, RKPM is implemented using a Lagrangian formulation. This formulation requires the deformation gradient, $\boldsymbol{F}$, to remain positive definite at all of the evaluation points within the domain in order to convert the Cauchy stress returned by the *LAME* material library to first Piola-Kirchhoff stress [8]. Should $\boldsymbol{F}$ lose its postive-definiteness, the method will terminate. This makes it difficulte to model phenomena with extremely large deformation such as fracture.

To make the method more versatile, the RKPM method should be updated to give the option to use the semi-Lagrangian formulation. This formulation acts by moving the nodes with the material points, but the kernels associated with the RK shape function remain in an Eulerian formulation. That is, their kernel supports do not distort with the material although the node itself does. This introduces a time-dependence into the shape functions and thus a velocity component into the governing equation of motion.

$$\int_{\Omega_x} \delta u_i \rho \ddot{u}_i d\Omega_x + \int_{\Omega_x} \delta u_{(i,j)} \sigma_{ij} d\Omega_x = \int_{\Omega_x} \delta u_i b_i d\Omega_x + \int_{\partial\Omega_x^h} \delta u_i h_i d\Gamma \tag{5.1}$$

The semi-Lagrangian approach uses the approximation as follows.

$$\psi_I(\boldsymbol{x}) = \boldsymbol{H}^T(\boldsymbol{0})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}(\boldsymbol{X}_I,t))\phi_a(\boldsymbol{x}-\boldsymbol{x}(\boldsymbol{X}_I,t)) \tag{5.2}$$

$$\boldsymbol{M}(\boldsymbol{x}) = \sum_{I=1}^{NP} \boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}(\boldsymbol{X}_I,t))\boldsymbol{H}^T(\boldsymbol{x}-\boldsymbol{x}(\boldsymbol{X}_I,t))\phi_a(\boldsymbol{x}-\boldsymbol{x}(\boldsymbol{X}_I,t)) \tag{5.3}$$

$$\boldsymbol{H}^T(\boldsymbol{x}-\boldsymbol{x}(\boldsymbol{X}_I,t)) = \begin{bmatrix} 1 & x_1 - x_1(\boldsymbol{X}_I,t) & x_2 - x_2(\boldsymbol{X}_I,t) & \dots & (x_3 - x_3(\boldsymbol{X}_I,t))^n \end{bmatrix} \tag{5.4}$$

Here, $\boldsymbol{x}$ refers to the current coordinate given by the mapping $\boldsymbol{x} = \phi(\boldsymbol{X},t)$ whereas $\boldsymbol{X}$ refers to the referential coordinate. By approximating the velocity as opposed to displacement in the solution space, we can take a single derivative on ψ_I, resulting in

$$\ddot{u}_i(\boldsymbol{x},t) = \dot{v}_i(\boldsymbol{x},t) = \sum_{I=1}^{NP} \psi_I(\boldsymbol{x})\dot{v}_{iI}(t) + \psi_I^*(\boldsymbol{x})v_{iI}(t) \tag{5.5}$$

Inserting the above into Equation 5.1, results in the following discrete system of equations.

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{N}\boldsymbol{v} = \boldsymbol{f}_I^{ext} - \boldsymbol{f}_I^{int} \tag{5.6}$$

where

$$\boldsymbol{M}_{IJ} = \int_{\Omega_x} \rho \psi_I(\boldsymbol{x}) \psi_J(\boldsymbol{x}) d\Omega \tag{5.7}$$

$$\boldsymbol{N}_{IJ} = \int_{\Omega_x} \rho \psi_I(\boldsymbol{x}) \psi_J^*(\boldsymbol{x}) d\Omega \tag{5.8}$$

$$\boldsymbol{f}_I^{int} = \int_{\Omega_x} \boldsymbol{B}_I^T \boldsymbol{\sigma} d\Omega \tag{5.9}$$

$$\boldsymbol{f}_I^{ext} = \int_{\Omega_x} \psi_I \boldsymbol{b} d\Omega + \int_{\partial\Omega_x^h} \psi_I \boldsymbol{h} d\Gamma \tag{5.10}$$

Specifics can be found in Guan [8]. The semi-Lagrangian formulation requires a robust neighbor search algorithm. This is because the kernels no longer deform with the material, thus allowing nodes to gain and lose neighbors as the material deforms. Neighbor search algorithms which perform this task already exist within the *Sierra/SM* framework and could easily be modified to work with RKPM. In addition, the SNNI and VCSNNI integration schemes are required here, since now the integration cells must be updated according to the deformation in the material. Generating conforming cells becomes prohibitively expensive to create, and SNNI or VCSNNI must be used instead.

Moving to a semi-Lagrangian framework allows for modeling of large deformation type problems. Because the semi-Lagrangian method no longer depends on the deformation gradient, $\boldsymbol{F}$, issues associated with it losing invertibility is obviated [8]. This allows for the modeling of extremely large deformation such as shear banding in soft material. An example of the semi-Lagrangian capability is illustrated in Figure 5.1.

Related to the above, soft-target impact between two bodies will greatly benefit from the semi-Lagrangian analysis. The large deformation will allow for materials with large void dilitation as well as shear banding. This topic is of great interest for particular explosives which become more sensitive to these phenomena.

Finally, with the advent of large deformation capabilities, fracture and fragmentation are more readily modeled with RKPM. By attaching a damage parameter, particles are allowed to separate from the block. This simulation of fracture is not the most realistic, as the simulation would produce a particle spray. However, it is a step in the direction towards full-fledged fracture, where an evolving fracture surface and contact definition would be required to define the interactions between pieces or "chunks" of separated material.

An overview of the various targets, the current capabilities within *Sierra/SM*, and potential advantages of using RKPM are presented in Table 5.1.

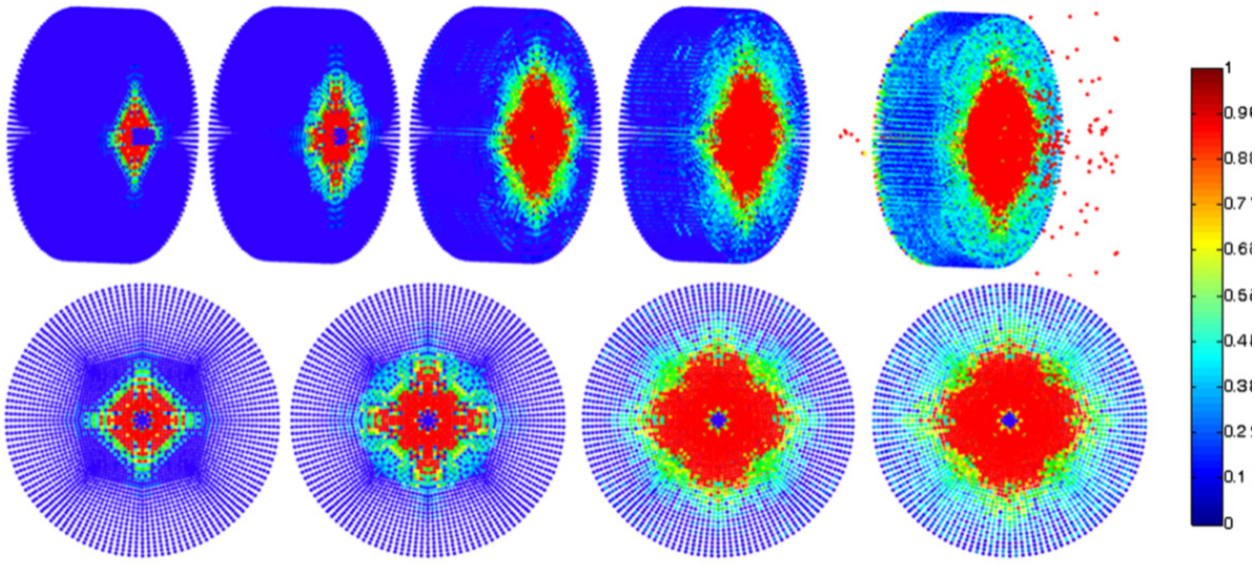


Figure 5.1: Penetration of bullet through concrete [8]

Table 5.1: Target applications of RKPM

Target Application	Current Capability	Advantages of RKPM
Large deformation	FEA remeshing/remapping	RKPM Semi-Lagrangian avoids mesh entanglement
Impact/Penetration	Cavity expansion, FEA element death, coupled CTH-Presto	RKPM Semi-Lagrangian
Fracture/Fragmentation	Element death, XFEA	RKPM Semi-Lagrangian
Rapid analysis	Meshing with Cubit-SCULPT	RKPM is semi-meshfree and can use vorocrust

Chapter 6

Current Issues

A number of issues still exist in the RKPM implementation of the *Sierra/SM* package which need to be addressed.

The semi-lagrangian formulation for RKPM needs to be implemented, especially to meet future milestones for RKPM. To accomodate this, the SNNI and the VCSNNI integration schemes must be verified and thoroughly tested. The SNNI and VCSNNI options are currently implemented and are used in the test repository for *Sierra/SM*. However, under some problems, the analyses failed to initialize when either SNNI or VCSNNI were specified. This may be related to memory issues and/or the use of additional stabilization. For general use from an analyst perspective, this option must be made more robust.

Contact may only be defined between an RKPM block and a FEA block. All other methods fail. This is due to the lofted particle surface used to define a surface for particle methods. This creates difficulty when attempting to define contact between two particle blocks. The lofted geometry creates a discontinuous contact surface and can allow two particle blocks to slip through each other under large deformations. This limitation also precludes self contact in RKPM analysis. In addition, all nodes of the RKPM block become lofted, even those that are not on the contact surface. Finally, the contact definition uses the generalized displacement instead of the true physical displacement.

Performance of RKPM analysis is a major detractor, both from a CPU perspective and a memory perspective. On average, the RKPM analysis will run 10-20 times as slow as a similarly defined problem using FEA. This may be attributed to the number of node neighbors, number of evaluation points associated with the integration cell of each RK node, and the critical time step. This may potentially be corrected by re-factoring the code base and identifying areas for improvement. The code will benefit both in terms of memory and performance and also from a maintainability standpoint.

The ability to define different material definitions within the same RKPM block currently does not exist. This requires a weak discontinuity where displacements are continuous but the strains across the material interface is not. Although not tested, one may potentially define two separate blocks and use a tied contact definition to simulate this. The presence of the generalized displacement coordinate may complicate this method of analysis.

Some applications have non-physical behavior when undergoing extreme plastic deforma-

tion. In Section 4, the solution exhibits asymmetric deformations in the tangential coordinate, whereas the FEA solution maintains axisymmetry. This may be attributed to stability issues which may arise when using SCNI or from flaws in the code base for RKPM.

Certain materials (especially those currently in development) do not work with RKPM. This was encountered when attempting to use the foam damage material model with RKPM and the analysis failed to initialize. This will probably require additional class member functions similar to other elements in the production code to allow RKPM to pass additional information necessary for these materials.

The time step for the RKPM analysis seems rather restrictive compared to FEA. For the problem as described in Section 4, the time step appeared to oscillate after enough deformation had occurred. This may be physical, given the softer foam block experiences a large amount of deformation and plasticity. The oscillations could then be a result of RKPM nodes sliding past each other due to the deformation. However, the total timestep oscillated between 4.4×10^{-8} s to 9.0×10^{-10} s, adding significant time to the analysis in general.

Capability Matrix

There are a number of capabilities within the RKPM analysis which either are missing or require special attention. These features are listed in Table 6.1 and are ordered in terms of highest priority according to the application they may apply to.

RKPM Improvements

With the capability matrix as outlined in Table 6.1, the effort/cost of implementing the various capabilities needs to be accounted for.

The semi-Lagrangian uses velocity as the independent variable as outlined in Section 5 [8]. As such, its implementation requires a different routine than for RKPM solved in the Lagrangian formulation. The central difference method of time integration may still be used. Much of the functionality required for semi-Lagrangian such as SNNI and VCSNNI are already implemented within the code, although they may require additional testing.

A continuous surface for the purpose of imposing contact between RKPM bodies and with bodies with other formulations is highly desirable for soft-target impact analysis. This may be achieved using a level-set method to implicitly define a surface between RKPM blocks [6]. Using this method, the lofted icosahedra and the gaps associated with them described in Section 2 would be removed. This allows for a more intuitive contact definition between RKPM blocks. Contact with FEA blocks do not offer shape functions outside the support of the element and thus using the level-set with them would require additional functionality such as ghost elements.

Table 6.1: Capability matrix for RKPM analysis

Capability	Intermediate Solution	Preferred Solution	Application
Contact	Lofted Spheres	Implicit surface definition	Soft target impact
Large Deformation	Lagrangian formulation (limited)	Semi-Lagrangian formulation	Soft target impact
Fracture	None	Semi-Lagrangian formulation	Blast & fracture applications
Multiple Material Definitions	Tied Contact	Basis Function Enrichment	Blast on Heterogeneous materials
Traction BCs	Use FEA surface and integration	Use RKPM implicit surface and integration	Blast and Pressure problems
Body Forces	Use FEA nodal mass	Use RKPM nodal mass	Basic BVPs
Lanzcos Time Control	Use nodal critical time step	Integrate RKPM with FETI	Potentially faster analysis
Restart	None	Functionality similar to FEA	Sierra/SM standard functionality

Using enrichment, RKPM may be coupled with FEA [9]. This would have the advantage of using FEA on the surface of the body and use its capabilities to apply boundary conditions and model interactions with other finite element bodies. This solution has its complexities, as *Sierra/SM* would need to identify merged blocks with separate element formulations. The RKPM block uses enriched shape functions at the interface to seamlessly merge with the FEA block.

Integration with the FETI package in *Sierra/SM* would allow the method to be run implicitly in *Adagio*. Not only that, but the Lanzcos method for determining eigenvalues could also lead to a more efficient analysis similar to the procedure described in Chapter 3 of the *Sierra/SM User Manual* [14].

Finally, an issue exists involving the modeling of concave surfaces with RKPM. This is an issue with meshless methods in general. RK nodes will associate with neighbors that are located across the surface of the body at concave geometries. In other words, a straight line traveling between the nodes would cross the surface. This introduces an interaction at that location and does not accurately model the physics there. A potential fix would involve using an existing ray-trace algorithm to identify such node pairs and modify their interactions accordingly.

A summary of the possible items to implement and their projected costs is given in Table 6.2.

Table 6.2: Cost/benefit table of potential items to implement in RKPM

Item	Specifics	Implementation Cost	Implementation Priority
Semi-Lagrangian Formulation	Requires solver to use velocity as independent variable	Low to medium	High
Performance	RKPM is CPU & memory heavy; re-factor of code likely required	Medium	High
Implicit surface	Level-set surface definition for contact; solves contact between RKPM bodies	High	High
RKPM-FEA Coupling	Simplifies complexities with contact	High	Medium
FETI integration	Benefits performance by utilizing Lanzcos	High	Low
Basis Function Enrichment	Requires redefining $\boldsymbol{H}$ vector to allow other basis functions	Medium to High	Low
RKPM nodal mass	Potentially already implemented	Low	Low
Non-convex geometry	Ray-trace algorithm	Medium to High	Low

References

[1] T. Belytschko, Y. Guo, W. K. Liu, and S. P. Xiao. A unified stability analysis of meshless particle methods. *International Journal for Numerical Methods in Engineering*, 48:1359–1400, 2000.

[2] J.-S Chen, M. Hillman, and M. Rüter. An arbitrary order variationally consistent integration for galerkin meshfree methods. *International Journal for Numerial Methods in Engineering*, 95:387–418, 2013.

[3] J.-S Chen, C. Pan, C.-T Wu, and W. K. Liu. Reproducing kernel particle methods for large deformation analysis of non-linear structures. *Computer Methods in Applied Mechanics and Engineering*, 139:195–227, 1996.

[4] J.-S Chen and H.-P Wang. New boundary condition treatments in meshfree computation of contact problems. *Computer Methods in Applied Mechanics and Engineering*, 187:441–468, 2000.

[5] J.-S Chen, C.-T Wu, S. Yoon, and Y. You. A stabilized conforming nodal integration for galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50:435–466, 2001.

[6] S.-W. Chi, C.-H. Lee, J.-S. Chen, and P.-C. Guan. A level set enhanced natural kernel contact algorithm for impact and penetration modeling. *International Journal for Numerical Methods in Engineering*, 102(3-4):839–866, April 2015.

[7] B. D Giffin. Verification tests for sierra/sms reproducing kernal particle method. Sand report, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, 2013. In review.

[8] P.C. Guan, S.W. Chi, J.-S. Chen, T.R. Slawson, and M.J. Roth. Semi-lagrangian reproducing kernel particle method for fragment-impact problems. *International Journal of Impact Engineering*, 38:1033–1047, 2011.

[9] A. Huerta and S. Fernández-Méndez. Enrichment and coupling of the finite element and meshless methods. *International Journal for Numerical Methods in Engineering*, 48(11):1615–1636, September 2000.

[10] T. J. R. Hughes. *The finite element method: Linear static and dynamic finite element analysis*. Dover Publications, Inc., 31 East 2nd Street, Mineola, N.Y. 11501, 2000.

[11] G. R. Johnson and W. H. Cook. Fracture characteristics of three metals subjected to various strains, strain rates, temperatures and pressures. *Engineering Fracture Mechanics*, 21:31–48, 1985.

[12] David Littlewood, Mike Hillman, Edouard Yreux, Joseph Bishop, Frank Beckwith, and Jiun-Shyan Chen. Implementation and verification of RKPM in the Sierra/SolidMechanics analysis code. In *Proceedings of the ASME 2015 International Mechanical Engineering Congress and Exposition (IMECE)*, Houston, Texas, 2015. Accepted for publication.

[13] W. M. Scherzinger and D. C. Hammerand. Constitutive models in LAME. SAND Report 2007-5873, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, 2007.

[14] SIERRA Solid Mechanics Team. Sierra/SolidMechanics 4.36 user's guide. SAND Report 2015-2199, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, 2015.

[15] M. L. Wilkins and M. W. Guinan. Impact of cylinders on a rigid boundary. *Journal of Applied Physics*, 44:1200–1206, 1973.

DISTRIBUTION:

1 Jiun-Shyan Chen, Department of Structural Engineering, University of California, San Diego, 9500 Gilman Drive Mail Code 0085, La Jolla, CA 92093-0085

1 MS 0840 E. Fang, Org. 1554
1 MS 0840 J. E. Bishop, Org. 1554
1 MS 0845 S. W. Attaway, Org. 1555
1 MS 0845 K. H. Pierson, Org. 1542
1 MS 0845 M. R. Tupek, Org. 1542
1 MS 1322 D. J. Littlewood, Org. 1444
1 MS 0899 Technical Library, 9536 (electronic copy)


Sandia National Laboratories