Development of parallel 3D RKPM meshless bulk forming simulation system

Wang H. *, Li Guangyao, Han X., Zhong Zhi Hua

The State Key Laboratory of Advanced Technology for Vehicle Design and Manufacture, College of Mechanical and Automotive Engineering, Hunan University, China

Received 21 November 2005; received in revised form 25 July 2006; accepted 9 August 2006
Available online 22 September 2006

Abstract

A parallel computational implementation of modern meshless system is presented for explicit for 3D bulk forming simulation problems. The system is implemented by reproducing kernel particle method. Aspects of a coarse grain parallel paradigm—domain decompose method—are detailed for a Lagrangian formulation using model partitioning. Integration cells are uniquely assigned on each process element and particles are overlap in boundary zones. Partitioning scheme multilevel recursive spectrum bisection approach is applied. The parallel contact search algorithm is also presented. Explicit message passing interface statements are used for all communication among partitions on different processors. The parallel 3D system is developed and implemented into 3D bulk metal forming problems, and the simulation results demonstrated the efficiency of the developed parallel reproducing kernel particle method system.

Keywords: Parallel; Repeating kernel particle method; Bulk forming; Domain decomposition method; Partition; Message passing interface; Speedup

1. Introduction

The non-linear finite element formulations for non-linearity of geometric and material have been well developed and a lot of significant works have been completed in bulk forming analysis. Nevertheless, FEMs are still ineffective in dealing with some extreme material distortions owing to severe mesh distortion. There are difficulties in solving problems involving large deflections and moving discontinuities. Typical problems include extremely large deformations in manufacturing processes, the propagation of interface between solids and liquids in casting, the propagation of cracks with arbitrary and complex paths in failure, and the tracking of the growth of phase boundaries and microcracking in advanced materials development. Some of FEMs require remeshing models in large deformation conditions; it still requires considerable computational efforts. Meshless methods eliminate mesh distortion for both large shape design changes and large deformation non-linear analysis.

A variety of meshless modeling methods have recently emerged [1]. RKPM was proposed by Liu et al. [1–5,39] to improve the accuracy of the SPH [27–29,31,32] method for finite domain problems. These methods have advantages over traditional FEMs for their ability to handle large deformation problems without mesh distortion, and for their solution accuracy due to the large domain of influence covered by particles/nodes. In this method, the kernel function is modified by introducing a correction function to meet the reproducing conditions. The resulting modified kernel function exactly reproduces polynomials to a specific order and thereby fulfills the completeness requirement. The shape functions developed from the method were later

Abbreviations: RKPM, reproducing kernel particle method; EFG, element free Galerkin method; PIM, point interpolation method; SPH, smoothed particle hydrodynamics; FEM, finite element method; RSB, recursive spectrum bisection; MRSB, multilevel recursive spectrum bisection; DDM, domain decomposition method; MPI, message passing interface; EBC, essential boundary condition; SYMMLQ, sparse symmetric equations; PE, process element. * Corresponding author. Tel.: +86 731 4329 266; fax: +86 731 8821 445. E-mail address: wanghuenergy@hotmail.com (H. Wang).
proven to be equivalent to moving least-squares kernel interpolates if linear basis functions were used [1–3]. Liu et al. also introduced wavelets [5] as the kernel functions and successfully applied RKPM to multiple scale analysis.

More recently, some effort has also been devoted to the solution of bulk metal forming processes by means of meshless approaches. Referenced papers encompass the following operations: ring compression [6,7], upsetting [6–13] and extrusion [6,7,10,14,15].

Meshless methods encounter the major technical barrier-low computation efficiency. Because meshless methods are based on the high order interpolation, the computation time of meshless requires much longer than FEMs. By taking a general view to the content of these papers it appears that major results are generally limited to 2D problems and simple 3D problems. Therefore, meshless methods are not wildly applied in practice. Parallelization of meshless codes is the way to settle expense of meshless computations. The parallel meshless methods to decompose the domain into several subdomains to solve are called DDM. DDMs are useful in two contexts. First, the division of problems into smaller problems through usually artificial subdivisions of the domain is a means for introducing parallelism into a problem. In this manner, problems that are intractable on serial computers can be solved on parallel computers. Second, many problems involve more than one mathematical model, each posed on a different domain, so that domain decomposition occurs naturally. DDMs are wildly used for explicit FEM problems [42,43,45]. It is natural that DDMs [44] can advance the meshless computation efficiency easily.

The present treatment will focus on the RKPM for explicit dynamic analysis of bulk metal forming problems, but the procedures can directly apply to some other meshless methods (e.g. EFG [1,30], PIM [16]) without difficulties. Several distinct advantages of RKPM are its ability to accurately model extremely large deformations without mesh distortion problems and its ease to adaptive modeling by simply changing particle definitions for desired refinement regions.

In this paper, an overview of a Lagrangian RKPM for explicit dynamic analysis of bulk metal forming processes by means of meshless approaches was presented to demonstrate the efficient parallel 3D RKPM system.

2. Weak form and discretization of RKPM for contact problems

The reproducing kernel approximation [1–5] of a function $u(X)$ in a domain $\Omega$ is expressed

$$u'(X) = \int_{\Omega} \Phi_d(X - Y) u(Y) d\Omega,$$  

where $u'(X)$ is the reproduced of function $u(X)$, and $\Phi_d$ is the window or kernel function with compact support.

In the following $X$ represents the material coordinates, $x$ is the spatial coordinate and $u$ is the displacement of the particles and $t$ denotes the time.

Discretizing the domain $\Omega$ by a set of particles $\{X_1, X_2, \ldots, X_{NP}\}$, where $NP$ is the total number of particles, the integral is approximated by the following summation:

$$u(X, t) = \sum_{i=1}^{NP} N_i(X) u_i(t),$$

where $N_i(X)$ is the Lagrangian shape function defined by

$$N_i(X) = C(X; X - X_i) \Phi_d(X - X_i) \Delta V_i.$$  

$C(X; X - X_i)$ is the correction function and $\Delta V_i$ is the volume of particle $i$. The more details on the construction of the shape function of RKPM can be found in Refs. [1–5].

For an approximation with the virtual work principle, the essential boundary conditions (EBCs) must be satisfied by the interpolation functions or accommodated by augmenting the variational statement with constraints. A major difference between RKPM and other methods (e.g. FEMs) is the manner in which EBCs can be enforced directly. The non-local interpolation condition of equation poses an additional computational challenge. Whereas EBCs for finite elements are imposed locally at particles (because they possess the Kronecker delta property), The EBCs enforcement with RKPM are non-local over a patch of particles/nodes. In some cases, the EBCs can be adequately approximated by local specification at the particles (assuming a Kronecker delta property). This approximation can be accurate, by St. Venant’s Principle, when the primary regions of interest are away from the EBCs. In general, however, a coupled set of equations is usually solved, even for explicit analyses. Previous efforts used Lagrange multipliers to constrain the variational statement [17] or a set of simultaneous equations is directly solved. In either case, significant computations were generally necessary to enforce the EBCs. These procedures also are not well suited for parallel processing, since they must generally be made over multiple processors.

Kent [18] proposes an alternate approach that may also require significant computational effort, as it is algebraically equivalent to other existing equations solving methods. By treating the imposition as a transformation of the interpolation functions, however, this form is better for parallel processing. Describing the EBC equations by

$$g_b = \sum_{j=1}^{NP} N_j(X_b) u_j,$$  

where $g_b$ is the boundary force vector, $N_j(X_b)$ is the shape function at boundary point $b$, and $u_j$ is the nodal displacement.
where \( g_B \) is the specified EBC at \( X_B \) or can be an integral relation along \( X_B \). It is written in matrix form for all conditions

\[
G^T g = g,
\]

(5)

where by Gram–Schmidt or Householder orthogonalization

\[
G^T G = I, \quad JG = 0, \quad J^T J = 1.
\]

(6)

The generalized variables are represented by

\[ d = J\dot{u} + Gg. \]

(7)

And we can obtain

\[
u(X) = N^T(X)J\dot{u} + N^T(X)Gg,
\]

(8)

\[
\frac{\partial u(X)}{\partial X} = \frac{\partial N^T(X)}{\partial X} J\dot{u} + \frac{\partial N^T(X)}{\partial X} Gg.
\]

(9)

Note that the constraint conditions are recovered. That is

\[
G^T d = G^T (J\dot{u} + Gg) = g.
\]

(10)

Since the new transformed interpolation functions of equation satisfy the EBCs, they can be directly applied in the virtual work statement. The alternate displacement vector \( \tilde{d} \), is now determined with the EBCs directly specified in \( g \). The orthogonalization in Eq. (6) may require a significant number of computer operations. These computations and the creation of the alternate interpolation functions of Eqs. (8) and (9), however, are done only once in the preprocessing phase. Transient displacement conditions are then accommodated by only changing the values in \( g \).

The gradient of displacement is

\[
\nabla_X u = \sum_{I=1}^{NP} \nabla_X N_I(X) u_I.
\]

(11)

The principle of the virtual work is used to formulate the general contact problems. The virtual work done by a stress field is

\[
\delta W^{\text{int}} = \int_{D_0} \nabla_X \delta u : P d\Omega.
\]

(12)

The virtual work done by the external loads, the inertial force and the contact traction through the virtual displacement field are denoted by \( \delta W^{\text{ext}}, \delta W^{\text{kin}} \) and \( \delta W^c \), and are calculated as

\[
\delta W^{\text{ext}} = \int_{D_0} \delta u \cdot b d\Omega + \int_{\Gamma_t} \delta \tilde{u} \cdot \tilde{t} d\Gamma,
\]

(13)

\[
\delta W^{\text{kin}} = \int_{D_0} \rho_0 \delta u \cdot \ddot{u} d\Omega,
\]

(14)

\[
\delta W^c = \int_{\Gamma_c} (q_N \delta \gamma_N + q_T \delta \gamma_T) d\Gamma,
\]

(15)

where \( P \) is the nominal stress tensor, \( \rho_0 \) is the density in the reference configuration, \( \tilde{t} \) is the traction force, \( b \) is the body force, \( \gamma_N \) and \( \gamma_T \) denote the relative displacement in normal and tangent direction at the contact interface \( \Gamma_c \), \( q_N \) and \( q_T \) present normal contacting force and frictional force, respectively.

The principle of virtual work can be written as

\[
\delta W^{\text{int}} - \delta W^{\text{ext}} - \delta W^{\text{kin}} - \delta W^c = 0.
\]

(16)

Substituting the particle approximations (2) and (11) into (16) the discretizing equation is obtained as

\[
m_I \ddot{u}_I = f_I^{\text{ext}} + f_I^{\text{cont}} - f_I^{\text{int}},
\]

(17)

\[
f_I^{\text{ext}} = \int_{\bar{D}_0} \phi_i b d\Omega + \int_{\Gamma_t} \phi_i \tilde{t} d\Gamma,
\]

(18)

\[
f_I^{\text{int}} = \int_{\Gamma_t} \nabla_X \phi_i \cdot P d\Omega.
\]

(19)

where \( m_I \) denotes the mass of particle \( I \), \( f^{\text{cont}} \) is the contact force whose details are described in Refs. [29,30].

3. Central difference algorithms

For explicit time integration methods, central difference algorithm [30] is popularly being used and it can be obtained by writing down the following central difference expressions for velocity \( v_n \) and acceleration \( a_n \) and vectors

\[
v_n = (d_{n+1} - d_{n-1})/(2\Delta t),
\]

(20)

\[
a_n = (d_{n+1} - 2d_n + d_{n-1})/(\Delta t^2).
\]

(21)

Substituting these expressions in the dynamic equilibrium equations, and rearranging terms it can be written as

\[
d_{n+1} = (M + C2\Delta t)^{-1}[\Delta t^2(p_{n-1} - f_{n-1}) + 2Md_n - (M - C\Delta t2)d_{n+1}],
\]

(22)

where \( p \) is the internal force vector. This is a recurrence relation in \( d_n \). If both \( M \) and \( C \) are diagonal matrices of mass and damp, the solution for \( d_{n+1} \) does not require factorization. Since the central difference algorithm does not require assembly of global matrices and can be handled at degrees of freedom level, the parallel implementation is pretty straightforward. The interprocessor communication requirement per time step is very small when compared to the implicit algorithms. So the DDM parallel method based on MPI is the best choice for explicit algorithms.

4. The explicit parallel algorithm for RKPM development

As FEM explicit parallel algorithms, separate pre-analysis software is created to partition any general unstructured RKPM models. Similar to explicit FEMs [33–35], in this section, DDM is implemented for parallel 3D RKPM system. We mainly focus on the partition methods and data structures for complex communication among subdomains.

4.1. Partition

4.1.1. Partition scheme for meshless methods

For effective parallel computing, it is critical to balance the computational load among processors while minimizing...
interprocessor communication. If tasks are not distributed in a balanced way, system may end up waiting for one task to complete while other tasks are idle. Therefore, separate pre-analysis software is created to partition any general unstructured RKPM models. Similar to explicit finite element parallel algorithms, it requires partitioning RKPM model first. Unlike the FEM, The shape function of meshless is interpolated in dynamics domain according to dilation parameter \( a \). The integration point of meshless is not limited inside of the support domain. Program need to keep all integration and interpolation information in each process elements (PE) entirely.

There are two strategies for partition process, partition based on particles and partition based on integration cells. The integration point partition is used in this work due to two advantages:

1. When particles based partition strategy is used, particle overlap domains are generated. And then, we should assign integration cells to each subdomain according to particles. As we know, unlike FEMs, particles may not belong to only one integration cell, thus, overlap integration cell domains should be generated. We have to design data structures for both particles and integration cells, respectively, therefore, the complexity and storage size are both enlarged a lot. On the other hand, if the integration cells based partition strategy is implemented, integration cells should be first partitioned and no cell overlap subdomain should be generated. We only assign particles to relevant subdomains according to integration cells.

2. We can directly use FE partition methods for integration cells instead of reconstructing connected relationship with discrete particles of global domain.

### 4.1.2. Multilevel approach for integration cells

Like FEMs, integration cells partition problems can be easily converted to graph partition problems \([41,46]\). So we discuss graph partition approaches to present the procedure for integration cells of RKPM method.

The RSB \([19]\) algorithm is one of a class of recursive bisection methods for partitioning unstructured problems. The most straightforward implementation of RSB, which uses the Lanczos algorithm \([20]\) to find the Fiedler vectors, is unacceptably slow for many applications. The MRSB \([21]\) is a refinement of the algorithm that is much faster, typically by an order of magnitude or more, and has been instrumental in the acceptance of RSB. The basic idea behind MRSB is to speed up the Fiedler-vector computation by constructing a series of successively smaller contracted graphs that maintain the global structure of the original graph. An algorithm based on the multilevel approach normally comprises three phases.

1. **Coarsening phase**: the original graph \( G \) is reduced into a series of successively coarser graphs.

   There are several ways to coarsen a graph. In the maximal independent set of a graph is chosen as the vertices of the coarse graph. An independent set of the graph \( G \) is a set of its vertices, with no two of the vertices in the set connected by an edge of \( G \). An independent set is a maximal independent set if the addition of an extra vertex makes it no longer independent.

   The most popular method for generating the multilevel of coarse graphs is based on edge collapsing. Selected edges are collapsed so that two vertices connected by one of these edges form a multi-node. Each vertex of the resulting coarse graph has a weight associated with it, which indicates the number of original vertices it contains. Each edge of the coarse graph also has a weight associated with it that indicates the number of original edges it represents. This edge collapsing approach has the distinctive advantage that the edge-cut (generalized here as the sum of the edge weights cut by the partitioning) on the coarsest graph equals the edge-cut of the finest (original) graph, if the partitioning on the coarsest graph are inherited by the finer graphs without further refinement.

   The edges are usually selected using the idea of matching. A matching of a graph is a set of edges, such that no two of them share the same vertex. The maximal match is a matching of the largest size. In a heavy edge matching \([22]\), vertices are visited randomly. For each vertex, the heaviest unmatched edge from the vertex is selected. Heavy edge matching has the advantage that the resulting coarse graph has a relatively small total edge weight, and therefore the partitioning of it is more likely to give a small edge-cut. As an alternative to edge collapsing based on maximal matching, a more aggressive coarsening approach has been employed \([23]\), where the greedy algorithm is used to form small clusters of, say, 10 vertices for collapsing. The coarsening process is applied until the graph has less than a preset minimum number of vertices.

2. **Partitioning phase**: partition the coarsest graph \( G_l \) into \( n \) parts.

   As the coarsest graph is of very small size, any partitioning algorithms will be able to partition it rapidly. The partition is executed in a manner such that the summation of the vertex weights of each subgraph should be roughly the same. When edge collapsing approach is used for graph coarsening, the edge-cut of the partitioning is the same as that for the finest graph, should the partitioning be inherited all the way to the finest graph without refinement. The choice of partitioning algorithms at this coarsest level is not very important \([22,24]\) to the overall quality of the partition because refinement will be carried out at the subsequent finer levels of the graph.

3. **Uncoarsening and refinement phase**: the partitioning of \( G_{l-1} \) is interpolated to \( G_{l-1} \) and refined, the process repeated and terminated at \( G \).

   During the uncoarsening and refinement phase, the partitioning of a graph \( G_l \) is inherited by the finer graph \( G_{l-1} \).

For the multilevel recursive spectral bisection (MRSB) algorithm \([21]\), the eigenvector for the coarse graph \( G_l \) is interpolated to the graph \( G_{l-1} \) as the first approximation.
of the eigenvector for \( G_{l-1} \). From there, instead of using the Lanczos algorithm to recalculate the eigenvector, Rayleigh iterations are used to improve the approximation. The resulting positive semi-definite linear systems are solved using sparse symmetric equations (SYMMLQ) [22]. Because a good initial approximation exists, the number of matrix–vector products involved in finding the Fiedler vector using the Rayleigh quotient algorithm is normally no more than ten, as opposed to up to several hundred if the Lanczos algorithm is used. Thus the MRSB algorithm is about an order of magnitude faster than the single level Lanczos implementation of RSB.

But there are two disadvantages to the multilevel spectral bisection approach. First, this is still a bisection algorithm and the partitioning of a graph into many subdomains would involve recursive use of the algorithm. Second, even though the Rayleigh quotient procedure is substantially faster than the Lanczos algorithm, each iteration is still of the order \( O(n) \), where \( n \) is the number of vertices in the graph. However, all that may be needed on the finer graph \( G_{l-1} \) is some refinement of the subdomain boundary to reduce edge-cut and to maintain the load balance. For a large graph with a good existing partition, the size of the boundary should be only a small fraction of that of the interior. Therefore, multilevel algorithms that adopt Kernighan–Lin (K–L) [26] type algorithms during the uncoarsening and refinement phase have proved themselves to be superior to MRSB [23–25]. The boundary refinement approach is not restricted to bisection and \( n \)-way refinement to the existing partition could be used. In [25] vertices of any subdomain are allowed to move to any of the other \( n-1 \) subdomains, while in [23] boundary vertices are allowed to move to the neighboring subdomains only.

4.1.3. Comparison between FE and meshless partition results

Meshless methods use the particle values in the local support domain to interpolate the shape functions. The RKPM approach needs about 20–30 particles in a support domain to construct the shape functions in this work, and the area of overlap zone depend on the number of particles in the local support domain. Thus the partition results of RKPM meshless models will generate much wider range boundary zones than FEM models. Comparing the partition results of FEM and RKPM models, the boundary of each subdomain is dissimilitude. Fig. 1 shows that the boundary between two partitions of FE model is a line. The integration points of each element are contained the inside of each element. Fig. 2 shows that the boundary between two partitions of RKPM model is widely overlap zone. It is clearly show that the number of particles in the overlap zone of RKPM model is much larger than the number of particles in the boundary belt of FEM model.

4.2. The data communication scheme

Because quantity of particles in boundary zones of the RKPM model is much larger than the particles of the FEM model, the time consumed in communication during the computation of RKPM is much longer than FEM. Additional, the EBCs cannot be enforced directly as mentioned in Section 2. For problems with a large number of EBCs, however, the number of support particles can significantly increase for integration points affected by the modifications in Eqs. (8) and (9). The number of shared particles may drastically increase. Therefore, communication may greatly increase thereby reducing the number of effective processors in some cases. Further effort regarding the accommodation of EBCs in parallel is still needed.

For these reasons, proposed parallel 3D RKPM system requires constructing structures for efficient data communication. For purpose of transferring the information in high efficiency, the system need to store following information before exchanging data. Firstly, in order to estimate time of communication, the number of particles and integration points in each overlap boundary zone among subdomains.
must be summed. For positioning the particles needed exchanging data, the particle’s location in each subdomain requires fixing and storing. Finally, we build the location mapping table to store relations of particles in each boundary zone as shown in Fig. 3 and the procedure of exchanging data of overlap zone is shown in Fig. 4.

4.3. The parallel algorithms for contact searching process

To simulate the contact between a deformable body and several rigid bodies in 3D parallel RKPM, a direct approach based on the segments, particle to segment contact algorithm, is used for 3D parallel system. The deformable body is discretized by a set of particles. The boundary particles of the deformable body are referred as slave particles in this work. The surfaces of the rigid bodies (master) are modeled by quadrilateral (or triangular) segments. The particles to segment contact algorithm is based on three major procedures: the test pairs’ creation, the contact detection and the contact forces evaluation. More details about contact algorithm are discussed in Refs. [29,30].

In common sense, master segments and slave segments of meshless models should be partitioned in synchronism, but such a strategy cannot be implemented due to following reasons:

1. Complexity of meshless parallel contact algorithms.

As we know, each subdomain of meshless model has boundary domains composed of quite a few shared particles. When the master segments group and the slave particles group are partitioned in the same domains, the segments belonging to this domain should be searched by particles in limited local zone. Thus, the segments of others should be ignored. The contact search under such statement probably losses contact particles in practice. In order to solve above problems, contact searching procedure need exchanging information among all subdomains. This procedure requires complex data structure to guarantee accurate results.

2. Significant computational effort.

Due to the large deformation problems, the position of each particle should be changed a lot. The initial partition results cannot fit for the contact search requirements. Therefore, it requires partitioning in each time step. If the master segments and slave particles are both partitioned, the information of master segments and slave particles need updating and rearranging. So partition process will employ significant computational effort.

For above reasons, we suggest the dynamic partition strategy for parallel contact problems. Only slave particles should be partitioned in each step during contact procedures. Although the search range should be expanded and more CPU time should be increased, dynamic partition strategy can find contact pairs completely and balance the communication time of each PE.

4.4. The parallel computation of RKPM explicit dynamic analysis implemented by MPI

Explicit MPI statements make all communication of parallel 3D RKPM system. In order to transfer data in
safety status and avoid possible deadlock and overhead associated with buffering, the non-blocking method is applied. Due to the explicit algorithms, the computations in each particle are independent. The developed system only needs to transfer the internal force of particles limited in overlapped boundary zones. Due to partition scheme proposed in Section 4.1, integration points are partitioned firstly; subdomains of integration points are all in non-overlap fashion. We only require summing all internal force of each overlap boundary zone in correct position to obtain the true value. The explicit parallel RKPM algorithm of whole process is described in Fig. 5.

5. Numerical examples

In this section, Taylor bar impact problem is benchmark test for demonstrating the validity and efficiency of the parallel 3D RKPM algorithm for large deformation. Back extrusion problem as practical example is used to test the efficiency of proposed system for bulk forming problems. Both of commercial software and the proposed parallel 3D RKPM system run on the IBM p690 with the AIX5.2 OS in this work. In all computations, the material model is elastic-plastic with Von Mises flow theory and linear isotropic hardening. Taylor bar impact problem is solved by LSDYNA3D; Back extrusion problem is solved by ANSYSY8.0, and the type 8-nodes hex solid element is used for analysis.

In this paper, we apply speedup and efficiency to estimate the performance of parallel 3D RKPM system. For a given algorithm or computation, let \( T_p \) be the time to perform the computation using \( p \) processors or arithmetic units. Note that this makes \( T_1 \) the depth of the algorithm. We define the speedup \( S_p \) with \( p \) processors as

\[
S_p = \frac{T_1}{T_p}
\]

and the efficiency \( E_p \) with \( p \) processors as

\[
E_p = \frac{S_p}{p}
\]

Because \( p \) processors can do no more than \( p \) times as much work per unit time as one processor, the efficiency is theoretically bounded above by one.

5.1. Taylor bar benchmark test

We need a benchmark test in order to verify the accuracy of 3D RKPM system and compare the efficiency of the various proposed parallelization methods presented further. Taylor bar impact problem which involves large deformation and moderate high strain rate is selected as benchmark. In the standard Taylor bar impact, a cylindrical bar impacts a rigid surface, as shown in Fig. 6a. Because of the symmetry of the problem only a quarter of the bar is discretized as shown in Fig. 6b. In this paper,
we will focus on the speedup obtained after the parallelization of the code using this benchmark test. The initial dimensions of the rod are $r_0 = 3.2$ mm and $l_0 = 32.4$ mm. The impact is assumed frictionless and the impact velocity is set to $V_i = 227$ m/s. The final configuration is obtained after 80 $\mu$s. The constitutive law is elasto-plastic with a linear isotropic hardening, material properties, given in Ref.

### Table 1
Material parameter list for Taylor bar impact

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Young's modulus $E$</td>
<td>117 GPa</td>
</tr>
<tr>
<td>Poisson's ratio $\nu$</td>
<td>0.35</td>
</tr>
<tr>
<td>Initial yield stress $\sigma$</td>
<td>400 MPa</td>
</tr>
<tr>
<td>Hardening modulus $H$</td>
<td>100 MPa</td>
</tr>
<tr>
<td>Mass density $\rho$</td>
<td>8930 kg/m³</td>
</tr>
</tbody>
</table>

### Table 2
Comparison of computation efficiency of different PE for Taylor bar problem

<table>
<thead>
<tr>
<th>Number of processors</th>
<th>Run time (s)</th>
<th>$S_p$</th>
<th>$E_p$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5323</td>
<td>1.00</td>
<td>100.00</td>
</tr>
<tr>
<td>2</td>
<td>2812</td>
<td>1.81</td>
<td>90.5</td>
</tr>
<tr>
<td>4</td>
<td>1656</td>
<td>3.21</td>
<td>80.35</td>
</tr>
<tr>
<td>8</td>
<td>830</td>
<td>6.41</td>
<td>80.16</td>
</tr>
<tr>
<td>16</td>
<td>512</td>
<td>10.39</td>
<td>64.64</td>
</tr>
<tr>
<td>32</td>
<td>220</td>
<td>24.14</td>
<td>75.5</td>
</tr>
</tbody>
</table>

Fig. 7. Taylor bar impact: time histories of the variation of the dimensions of the bar. (a) Radius and (b) height.

Fig. 8. Deformed configurations for Taylor impact at time 80 $\mu$s. Distribution for 16 CPUs, different deep color means different processes. (a) RKPM, (b) FEM and (c) Gaussian integration cells.

Fig. 9. Parallel performances on IBM690 of RKPM for Taylor bar problem: (a) speedup comparison and (b) efficiency comparison.
corresponding to an OHFC copper are reported in Table 1.

For verifying the accuracy of parallel 3D RKPM system, the commercial code LS-DYNA3D is also implemented for solving same problems. Models of meshless consist of 1369 particles for RKPM with stress point integration and 1200 elements for LSDYNA3D.

Fig. 7a shows the time history of the radius at the impact plane. It can be seen that the results by RKPM meshless code and LSDYNA3D FEM program are nearly the same. The results by FEM are also given. At $t = 80 \mu s$, the radius at the plane of impact is 7.30 mm with RKPM. The result by FEM is 7.38 mm. Many results for this radius have been collected in Ponthot [40] and the mean value is 7.11 mm.

---

Fig. 10. Schematic model for back extrusion analysis: (a) continuum domain and (b) discretized model.

Fig. 11. Deformed configurations for back extrusion process.
5.2. Back extrusion analysis

The simulation of the back extrusion process is considered. Fig. 10 shows the statement of the problem. The die is fixed while the punch is moved downward with a constant velocity of 10 m/s. The coefficient of friction is 0.12. The simulation time is 0.01 s. Because of the symmetry of the problem only half of the workpiece has been discretized with 5621 (73 x 7 x 11) particles and 16,000 Gaussian integration points are applied for integration. The material is the same as in Table 1.

Fig. 11 shows the deformed configuration at punch stroke of 0.05 m, 0.15 m. The deformed patterns are nearly the same by RKPM and FEM. Fig. 12 demonstrates a good effective strain and stress distribution agreement by RKPM and FEM. However, as simulation process contin-
ues at punch stroke of 0.25 m, the mesh distortion is more severe and difficulties occur in FEM computation as shown in bottom of Fig. 11. It can be seen that some elements in FEM solution are overlap at the punch stroke and will come to a halt. It is proved that the proposed 3D RKPM system can easy to solve element distortion problems.

In order to demonstrate the efficiency of parallel meshless program, The 1, 2, 4, 8, 16, 32 CPUs are implemented for back extrusion problem. The performance parameters speedup $S_p$, efficiency $E_f$ and computation time are listed in the following Table 3. The distribution of Gaussian integration cells of RKPM in each process during computation is shown in Fig. 13. Comparison among practical speedup and efficiency of parallel 3D RKPM system, commercial software ANSYS and ideal value are shown in Fig. 14.

5.3. Wheel forging analysis

This example aims to simulate the cold forging process of a wheel. A cylindrical slug as shown in Fig. 15a is used to obtain a wheel as shown in Fig. 15b, meshless model is presented in Fig. 15c. The following materials parameters are used for the simulation.

Because of the symmetry of the problem an axisymmetric formulation is used and only half of the domain is discretized with 9610 particles. The punch and die, assumed to be rigid and frictionless, are discretized by piecewise linear segments to fit the geometry of the wheel. The punch is moving downward with constant velocity of 10 m/s while the die is fixed. The maximum stroke of the punch at the end of the simulation is 0.02 m. The configuration at punch stroke of 0.01 m, 0.02 m and distribution of stress and strain at 0.01 m are shown in Figs. 16 and 17. It is presented that proposed parallel 3D RKPM system and ANSYS8.0 have almost same results.

The 1, 2, 4, 8, 16, 32 CPU are also implemented for forging problem to demonstrate efficiency of parallel 3D RKPM system. The performance parameters speedup $S_p$, efficiency $E_f$ and computation time are listed in the following Table 4. The distribution of Gaussian integration cells in each process is presented in Fig. 18. Comparison between practical speedup and efficiency of parallel 3D RKPM system and ideal value are shown in Fig. 19.

5.4. Discussions

The capabilities of proposed parallel 3D RKPM method in simulating 3D bulk forming problems with bulk forming problems have been studied. Figs. 11, 12 and 15, 16 show that both proposed parallel 3D RKPM system and the

<table>
<thead>
<tr>
<th>Number of processors</th>
<th>Run time (s)</th>
<th>$S_p$</th>
<th>$E_f$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>69783.17</td>
<td>1.00</td>
<td>100.00</td>
</tr>
<tr>
<td>2</td>
<td>35479.84</td>
<td>1.96</td>
<td>98.00</td>
</tr>
<tr>
<td>4</td>
<td>20545.67</td>
<td>3.39</td>
<td>84.75</td>
</tr>
<tr>
<td>8</td>
<td>12674.82</td>
<td>5.53</td>
<td>69.13</td>
</tr>
<tr>
<td>16</td>
<td>7765.21</td>
<td>8.98</td>
<td>56.13</td>
</tr>
<tr>
<td>32</td>
<td>3116.71</td>
<td>22.39</td>
<td>69.96</td>
</tr>
</tbody>
</table>

Fig. 13. Gaussian integration cells distributed in 16 PE at punch stroke of 0.25 m. Effective strain distribution (TOP) and effective stress distribution predicted by RKPM and FEM.

Fig. 14. Performance diagram for drawback extrusion: (a) speedup comparison and (b) efficiency comparison.

Fig. 15
Fig. 15. Schematic model for back extrusion analysis: (a) raw slug, (b) geometry of the wheel and (c) discretized model.

Fig. 16. Deformed configurations for wheel forging process.
FEM software LSDYNA3D, ANSYS8.0 have a good convergence rate in configuration and distribution of strain and stress, but the RKPM can solve the distortion of meshes easily.

From Tables 2, 3 and 5, Figs. 9, 14 and 19 of above numerical examples, we can see almost ideal speedup is achieved for both the fixed and scaled models even up to 32 processors, and this trend is anticipated to continue over 32 processors. Figs. 8c, 13 and 18 present good balancing rates during simulation process, each process has almost same amount of Gaussian integration cells. It is the key reason why proposed parallel 3D RKPM system has such good parallel computation efficiency based on MPI. Additional, although commercial FEM software (e.g. ANSYS, DEFORM) have better computation efficiency than developed 3D RKPM system, current system has better speedup.

Table 4
Material parameter list for wheel forging process

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Young's modulus $E$</td>
<td>71 GPa</td>
</tr>
<tr>
<td>Poisson’s ratio $v$</td>
<td>0.3</td>
</tr>
<tr>
<td>Mass density $\rho$</td>
<td>2700 kg/m$^3$</td>
</tr>
<tr>
<td>Stress–strain curve $\sigma$</td>
<td>$576.79(0.01658 + \rho^{0.3593})$ MPa</td>
</tr>
</tbody>
</table>

Fig. 17. Computed results for the compression of wheel forging problems at final punch stroke deformed models: effective strain distribution (TOP) and effective stress distribution predicted by RKPM and FEM.

Fig. 18. Gaussian integration cells distributed in 16 processes in the final step of wheel forging process.
processors by MRSB method and shared particles definitions are overlap, so that all support domains for all particles are defined locally on the corresponding processor. Explicit MPI message passing statements are used for all communication among subdomains on distributed processors. Taylor bar impact problem is presented for proving validity of parallel 3D RKPM. Both the simulation of Taylor bar and back extrusion analysis demonstrate the efficiency of the current parallel RKPM for bulk forming problems. However, it should be noted that the communication execution time of meshless is much than FEM due to large overlap boundary zones.

6. Conclusions

A parallel computational implementation of 3D RKPM methods is presented for the explicit dynamic analysis. A DDM parallel implementation is used with model partitioning for a Lagrangian formulation. With proposed strategy, integration cells are uniquely partitioned on separate and parallel efficiency than ANSYS. The parallel 3D RKPM system is a useful system for bulk forming problems in consideration of above factors.

References


