Incompressible Smoothed Particle Galerkin (ISPG) Method for an Efficient Simulation of Surface Tension and Wall Adhesion Effects in the 3D Reflow Process

Xiaofei Pan[1], C. T. Wu[1], Wei Hu[1]

[1] *Computational and Multiscale Mechanics Group*

Livermore Software Technology Corporation, 7374 Las Positas Road, Livermore, CA 94551, USA

Abstract

A new numerical method in LS-DYNA, the incompressible smoothed particle Galerkin (ISPG) method, is developed for the simulation of shape evaluation of solder joints during the electronic reflow process. The ISPG method is a combination of several unique developments aim to suppress key numerical instabilities observed in the simulation of incompressible free surface fluid flow using Lagrangian particle methods. In ISPG method, a momentum-consistent velocity smoothing algorithm is utilized to offer the desired numerical stability associated with the fluid particle integration scheme. To stabilize the solution in pressure equations, a 2nd order generalized rotational incremental pressure-correction scheme is developed for the incompressible flows. To simulate the shape evolution of solder joints during the reflow process, a numerical procedure considering the surface tension and wall adhesion effects is introduced. Several numerical examples are studied to demonstrate the accuracy and the efficiency of the new method.

Keywords: Particle; ISPG; Momentum-consistent; Incompressible; Reflow

---

1. Introduction

Solder joints are used to provide both mechanical and electrical connection in electronic packaging for IC industry. The solder geometry such as standoff height, contact angle, and so on, has the significant influence on the quality of solder joints such as their fatigue life. Thus, the prediction of shape evolution of solder joints in the reflow process is very important in electronic applications. In the past, analytical approaches, such as the truncated sphere method [1], the force-balanced solution [2-5], and the energy-based method [1], were usually used to predict the final shapes of the solder joints. However, the applications of these approaches have following major limitations: 1, their simulations of the shape evolution of the solder joints have no physical meanings. 2, they cannot simulate the reflowing with complex boundaries. For example, the Surface Evolver, a program developed by Brakke [5], has been successfully applied to predict the shapes of solder joints in the electronic packaging process, but it fails to simulate the process when molten solders flow out of the solder

pad. 3, when a ball grid array (BGA) deforms as the temperature changes in the reflowing, it affects the deformation of the solders. Under this situation, it will be too difficult for those analytical approaches to simulate such reflow processes. Thus, there is a great demand from IC industry to develop a physics-based numerical method to predict the shape evolution of solder joints during the electronic reflow process.

In general, the soldering process lasts about several minutes. On the other hand, the size of the solder is very small, which causes big challenges for the physics-based numerical simulation in terms of computational cost. In the past decade, some researchers have tried to use the Eulerian methods to simulate the shape evolution of the solder joints. Chou et. al. [6] developed the SOLA (SOLution Algorithm) scheme to solve the velocity and pressure fields in the fluid, in which the construction of the interface and the transportation of volume fractions of liquid in the cells are coupled with PLIC (Piecewise Linear Interface Calculation) and VOF (Volume of Fluid) technologies. This method has been successfully applied to simulate the overflow phenomena in the reflowing process. Nevertheless, this method has similar difficulties of Eulerian methods in tracking the complex interfaces and coupling with solids.

Recently, the simulations of free surface flow based on Lagrangian frameworks have become increasingly popular. As the transport effect represented by the advective rate of change in the fluid flow can be fully bypassed in a Lagrangian system, numerical diffusion traditionally observed in Eulerian methods is minimized in Lagrangian particle methods. Additionally, since Lagrangian particles move with the fluid flow, free surfaces or interfaces can be easily tracked by identifying boundary particles. In comparison to Eulerian methods such as Finite Difference Method (FDM) [7] and Finite Volume Method (FVM) [8], Lagrangian particle methods can solve the fluid flow problems involving more complex geometry and physics with less numerical difficulties. Among Lagrangian particle methods, the Smoothed Particle Hydrodynamics (SPH) method [9] developed by Gingold et al. in late 1970s for astrophysical problems has been considered the earliest Lagrangian particle method. Since SPH is a collocation method instead of a Galerkin method, the consistent and rigorous numerical treatment of fluid formulation and boundary conditions remains to be developed. Because of that, its application in high accuracy fluid flow problems such as the reflow process is not as popular as the Eulerian method.

Lagrangian particle methods are also popular in modeling solid problems [10][11]. Recently, the smoothed particle Galerkin (SPG) method [12][13] was developed to suppress the tensile instability and spurious zero-energy mode in standard Lagrangian particle methods for solid mechanic applications. Different from conventional penalty approaches [14] which utilized a direct smoothing of velocity fields and stabilization terms in the weak formulation, a momentum-consistent stabilization algorithm was developed for SPG to generate the smoothed velocity field for stabilization and yet to preserve the conservation of linear momentum for the fully discretized equations. Since no additional stabilization term is needed, the new algorithm demands only one integration point per particle in the computation, and thus is computational efficient. Those nice properties of SPG prompt the development of new formulation for the high accuracy and efficient simulation of fluid flow problems.

In this paper, a new SPG formulation for the simulation of incompressible fluid flows with free surfaces is presented. It is shown that the momentum-consistent stabilization algorithm in the SPG method can provide the desired numerical stability for the particle integration scheme in the flow simulation. Additionally, the moving-least square approximation with the 1st order basis is used for the spatial discretization, and the 2nd order rotational incremental pressure-correction scheme is employed to impose the incompressible condition. Those numerical schemes enhance the consistent modeling of spatial discretization and incompressible condition and guarantee the conservation of fluid volume though the reflow simulation. To further simulate the shape evolution of solder joints during reflow process, the surface tension based on the surface curvature and the treatment of wall adhesion with contact angles are also considered in our new algorithm. Two numerical examples are studied to validate the accuracy and stability of the proposed method.

2. Preliminary

2.1 Governing equations

In the Lagrangian description, the Navier-Stokes equation which is comprised of the pressure term, the viscosity term and the gravity term can be written as

$$\frac{D\boldsymbol{v}}{Dt} = -\frac{1}{\rho}\nabla p + \frac{\vartheta}{\rho}\nabla^2\boldsymbol{v} + \boldsymbol{g} \tag{1}$$

where $\boldsymbol{v}$, ρ, p are fluid velocity, density and pressure, respectively. ϑ is the dynamic viscosity, t denotes the time, and $\boldsymbol{g}$ is the gravity. The incompressible condition is enforced using the continuity equation by:

$$\nabla \cdot \boldsymbol{v} = 0 \tag{2}$$

2.2 Projection scheme for incompressible flows

In our study, the generalized the rotational incremental pressure-correction scheme proposed by Guermond et al. [15] is used. Firstly, the first derivative of velocity with respect of time is approximated by using the qth-order backward difference formula denoted by $\frac{1}{\Delta t}(\beta_q \boldsymbol{v}^{n+1} - \sum_{j=0}^{q}\beta_j \boldsymbol{v}^{n-j})$, where, $\beta_j (j = 0, \cdots, q)$ are the coefficients, and in particular one has

$$\begin{cases} \beta_0 = -1.0, \beta_1 = 1.0, & \text{if } q = 1 \\ \beta_0 = 0.5, \beta_1 = -2.0, \beta_2 = 1.5, & \text{if } q = 2 \end{cases} \tag{3}$$

Then, the pressure-correction schemes can be generalized into the following form:

$$\begin{cases} \dfrac{1}{\Delta t}\left(\beta_q \boldsymbol{v}^{n+1^*} - \sum_{j=0}^{q-1} \beta_j \boldsymbol{v}^n\right) = -\dfrac{1}{\rho}\nabla p^* + \dfrac{\vartheta}{\rho}\nabla^2 \boldsymbol{v}^{n+1^*} + \boldsymbol{g}, \boldsymbol{v}^{n+1^*}|_{\Gamma_v} = \overline{\boldsymbol{v}} \\ \dfrac{\beta_q}{\Delta t}\left(\boldsymbol{v}^{n+1} - \boldsymbol{v}^{n+1^*}\right) = -\dfrac{1}{\rho}\nabla\left(p^{n+1} - p^* + \chi\vartheta\nabla\cdot\boldsymbol{v}^{n+1^*}\right) \end{cases} \tag{4}$$

where p^* is a generalized pressure prediction, e.g.

$$p^* = \begin{cases} 0 & (r=0) \\ p^n & (r=1) \\ 2p^n - p^{n-1} & (r=2) \\ \cdots & (r=3) \end{cases} \tag{5}$$

and $\chi = 0$ or 1 is a user defined parameter.

As remarked by Guermond et al. [15], if one chooses $r = q$ in the generalized rotational incremental pressure-correction scheme, the consistency error in H^1-norm and the pressure in L^2-norm are both of the same order. In our study, $r = q = 1$ is chosen, which is the 2nd order generalized rotational incremental pressure-correction scheme. Subsequently, in the first sub step of the 2nd order GRIPC scheme, we have

$$\frac{1}{\Delta t}\left(\boldsymbol{v}^{n+1^*} - \boldsymbol{v}^n\right) = -\frac{1}{\rho}\nabla p^n + \frac{\vartheta}{\rho}\nabla^2 \boldsymbol{v}^{n+1^*} + \boldsymbol{g}, \boldsymbol{v}^{n+1^*}|_{\Gamma_v} = \overline{\boldsymbol{v}} \tag{6}$$

In the second sub step of the 2nd GRIPC scheme, the velocities are corrected through the following equation

$$\frac{1}{\Delta t}\left(\boldsymbol{v}^{n+1} - \boldsymbol{v}^{n+1^*}\right) + \frac{1}{\rho}\nabla\phi^{n+1} = 0, \boldsymbol{v}^{n+1}|_{\Gamma_v} = \overline{\boldsymbol{v}} \tag{7}$$

where the variable ϕ is defined as

$$\phi^{n+1} = p^{n+1} - p^n + \vartheta\nabla\cdot\boldsymbol{v}^{n+1^*} \tag{8}$$

The corrected velocity in Eq. (7) must satisfy the divergence-free condition, which yields the following Poisson equation,

$$\nabla\cdot\boldsymbol{v}^{n+1^*} = \frac{\Delta t}{\rho}\nabla^2\phi^{n+1} \tag{9}$$

From Eq. (7), it's observed that the Neumann boundary condition $\nabla\phi^{n+1}\cdot\boldsymbol{n}|_{\Gamma_v} = 0$. The Dirichlet boundary condition at the free surface can be induced from Eq. (8), that is

$$\left(\phi^{n+1} + p^n - \vartheta\nabla\cdot\boldsymbol{v}^{n+1^*}\right)|_{\Gamma_p} = p^{\mathrm{fs}} \tag{10}$$

where Γ_p is the free surface boundary, and p^{fs} is the pressure at the free surfaces.

2.2 Surface tension

The surface tension causes a pressure jump $\sigma_\kappa^{\mathrm{fs}}$ at the interface between the liquid solder and air proportional to the mean curvature of the interface

$$\sigma_\kappa^{\mathrm{fs}} = \gamma\kappa \tag{11}$$

where γ is the surface tension coefficient, and $\kappa = -\nabla \cdot \boldsymbol{n}$ is the mean curvature, and $\boldsymbol{n}$ is the inward norm direction of the liquid solder surface. Subsequently, the pressure term applied at the essential pressure boundary condition can be calculated as

$$p^{\mathrm{fs}} = p^{\mathrm{air}} + \sigma_\kappa^{\mathrm{fs}} \tag{12}$$

2.3 Wall adhesion

When molten solder is in contact with the substrate, the solder liquid interface forms a contact angle θ with the wall boundary. If the angle θ is equal to static contact angle θ_{eq}, a state of static equilibrium is reached. If not, then a nonzero wall adhesion force tends to pull the interface to the equilibrium position. In our algorithm, the wall adhesion boundary condition can be expressed by the unit free surface normal $\boldsymbol{n}$ at the interface point $\boldsymbol{x}_{\mathrm{w}}$ shown in Figure 1

$$\boldsymbol{n}(\boldsymbol{x}_{\mathrm{w}}) = \boldsymbol{n}_{\mathrm{w}} \cos(\theta_{\mathrm{eq}}) + \boldsymbol{t}_{\mathrm{w}} \sin(\theta_{\mathrm{eq}}) \tag{13}$$

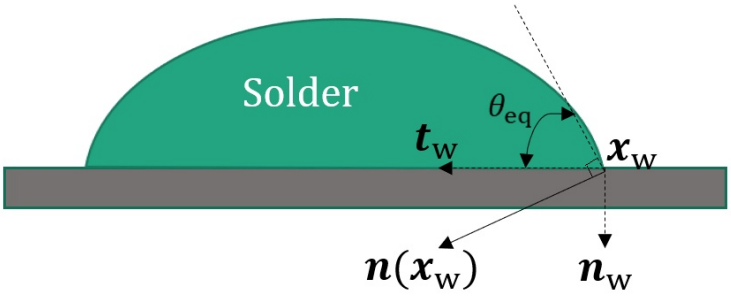


Figure 1. Illustration of the wall adhesion.

3. Incompressible momentum-consistently smoothed Lagrangian particle formulation

Using the 2nd generalized rotational incremental pressure-correction scheme, the weak from of incompressible flow problem can be formulated within the Galerkin framework using the Lagrangian approach

and the particle approximation.

3.1 Particle approximation function

With a particle distribution denoted by an index set $Z_I = \{\boldsymbol{x}_I\}_{I=1}^{N^P} \in R^3$, the function $\boldsymbol{u}(\boldsymbol{x})$ can be approximated using the meshfree or particle approximation

$$\boldsymbol{u}^h(\boldsymbol{x}) = \sum_{I \in Z_I} \Psi_I^a(\boldsymbol{x})\boldsymbol{u}_I \qquad \forall \boldsymbol{x} \in \Omega \tag{14}$$

where N^P is the total number of particles in the discretization. $\Psi_I^a(\boldsymbol{x}), I = 1, \cdots, N^P$ can be interpreted as the shape functions of the approximation for the field $\boldsymbol{u}^h$ where the superscript "a" denotes the support size of $\Psi_I^a(\boldsymbol{x})$. The moving least-square interpolation [16] with the first-order basis is used in this work.

3.2 Particle discretization in the velocity-prediction step

The weak form of Eq. (6) using Eq. (14) and the direct nodal integration (DNI) scheme [17] gives

$$\begin{aligned}
&\frac{1}{\Delta t}\sum_{K \in Z_I} \hat{m}_K \Psi_I^a(\boldsymbol{x}_K) \sum_{J \in Z_I} \Psi_J^a(\boldsymbol{x}_K)\boldsymbol{v}_{Ji}^{n+1^*} - \frac{1}{\Delta t}\sum_{K \in Z_I} \hat{m}_K \Psi_I^a(\boldsymbol{x}_K)\hat{\boldsymbol{v}}_{Ki}^n \\
&= -\frac{1}{\rho}\sum_{K \in Z_I} \hat{m}_K \Psi_I^a(\boldsymbol{x}_K) \sum_{J \in Z_I} \Psi_{J,i}^a(\boldsymbol{x}_K) p_J^n \\
&+ \frac{\vartheta}{\rho}\sum_{K \in Z_I} \hat{m}_K \Psi_I^a(\boldsymbol{x}_K) \sum_{J \in Z_I} \sum_{j=1}^{\text{ndim}} \Psi_{J,jj}^a(\boldsymbol{x}_K)\boldsymbol{v}_{Ji}^{n+1^*} + \sum_{K \in Z_I} \hat{m}_K \Psi_I^a(\boldsymbol{x}_K) g_i
\end{aligned} \tag{15}$$

In Eq. (15) $i, j = 1,2,3$ for 3D $(\text{ndim} = 3)$ and $1,2$ for 2D$(\text{ndim} = 2)$. Eq. (15) can be expressed by the matrix form given by

$$\left(\frac{1}{\Delta t}\boldsymbol{M}^{n+1} + \boldsymbol{K}_i^{n+1}\right)\boldsymbol{V}_i^{n+1^*} = \frac{1}{\Delta t}\boldsymbol{P}_i^n + \boldsymbol{F}_i^{p,n} + \boldsymbol{F}_i^{g,n+1} \tag{16}$$

where

$$\boldsymbol{M}_{IJ}^{n+1} = \sum_{K \in Z_I} \hat{m}_K \Psi_I^a(\boldsymbol{x}_K)\Psi_J^a(\boldsymbol{x}_K) \tag{17}$$

$$\boldsymbol{P}_{Ii}^n = \sum_{K \in Z_I} \hat{m}_K \Psi_I^a(\boldsymbol{x}_K)\hat{\boldsymbol{v}}_{Ki}^n \tag{18}$$

$$\boldsymbol{K}_{IJ}^{n+1} = -\frac{\vartheta}{\rho}\sum_{K\in Z_I}\widehat{m}_K\Psi_I^a(\boldsymbol{x}_K)\sum_{j=1}^{\text{ndim}}\Psi_{J,jj}^a(\boldsymbol{x}_K)(\boldsymbol{x}_K) \tag{19}$$

$$\boldsymbol{F}_{iI}^{p,n+1} = -\frac{1}{\rho}\sum_{K\in Z_I}\widehat{m}_K\Psi_I^a(\boldsymbol{x}_K)\sum_{J\in Z_I}\Psi_{J,i}^a(\boldsymbol{x}_K)p_J^n \tag{20}$$

$$\boldsymbol{F}_{iI}^{g,n+1} = \sum_{K\in Z_I}\widehat{m}_K\Psi_I^a(\boldsymbol{x}_K)g_i \tag{21}$$

The predicted velocity is adjusted at the solid boundaries to ensure the velocity boundary condition in Eq. (6) is satisfied, which is can be found in [18].

3.3 Solution of Poisson equation with the 2nd order GRIPC scheme

The pressure field $p^h(\boldsymbol{x})$ and the variable $\phi^h(\boldsymbol{x})$ defined in Eq. (8) are approximated by the meshfree/particle approximation, that is

$$p^h(\boldsymbol{x}) = \sum_{I\in Z_I}\Psi_I^a(\boldsymbol{x})p_I \qquad \forall\boldsymbol{x}\in\Omega \tag{22}$$

$$\phi^h(\boldsymbol{x}) = \sum_{I\in Z_I}\Psi_I^a(\boldsymbol{x})\phi_I \qquad \forall\boldsymbol{x}\in\Omega \tag{23}$$

where p_I and ϕ_I are nodal coefficients for pressure and ϕ at the particle I.

To enforce different boundary conditions, particles are divided into three groups:

1) Interior particles, for which the Poisson equation (9) is employed, that is

$$\frac{\Delta t}{\rho}\sum_{i}^{\text{ndim}}\Psi_{J,ii}^a(\boldsymbol{x_I})\,\phi_J^{n+1} = \sum_{i}^{\text{ndim}}\Psi_{J,i}^a(\boldsymbol{x}_I)\boldsymbol{v}_{Ji}^{n+1^*}.\,\text{ndim} = 2\ \text{ for }\ 2\text{D, and }\ 3\ \text{ for }\ 3\text{D} \tag{24}$$

2) Free surface particles, for which the Dirichlet boundary condition (10) is employed, that is

$$\Psi_J^a(\boldsymbol{x}_I)\phi_J^{n+1} + \Psi_J^a(\boldsymbol{x}_I)p_J^n - \vartheta\sum_{i}^{\text{ndim}}\Psi_{J,i}^a(\boldsymbol{x}_I)\boldsymbol{v}_{Ji}^{n+1^*} = \bar{p}(\boldsymbol{x}_I) \tag{25}$$

3) Neumann boundary particles, for which the Neumann boundary condition $\nabla\phi^{n+1}\cdot\boldsymbol{n}|_{\Gamma_v} = 0$ is employed, that is

$$\sum_{i}^{\text{ndim}} n_i(\boldsymbol{x}_I)\Psi_{J,i}^{a}(\boldsymbol{x}_I)\,\phi_J^{n+1} = 0 \tag{26}$$

where $\boldsymbol{n}(\boldsymbol{x}_I)$ is the outward normal direction of the fluid at the Neumann particle I. Using Eq. (24)-(26), the ϕ_J^{n+1} can be solved. Subsequently, the pressure field $p^h(\boldsymbol{x})$ at the time step $n+1$ can be obtained through Eq. (8), that is

$$p^{h,n+1}(\boldsymbol{x}) = \sum_{J\in Z_I}\Psi_J^a(\boldsymbol{x})\,\phi_J^{n+1} + \sum_{J\in Z_I}\Psi_J^a(\boldsymbol{x})\,p_J^n - \vartheta\sum_{i}^{\text{ndim}}\sum_{J\in Z_I}\Psi_{J,i}^a(\boldsymbol{x})\boldsymbol{v}_{Ji}^{n+1^*} \tag{27}$$

Their values at the particles are used as particle coefficients of pressure p^{n+1} for the computation in the next time step.

3.4 Particle discretization in the velocity-correction step with the 2nd GRIPC scheme

By employing the meshfree/particle shape function and the direct nodal integration DNI scheme, Eq. (7) becomes

$$\frac{\boldsymbol{M}^{n+1}}{\Delta t}\left(\boldsymbol{V}_i^{+1} - \boldsymbol{V}_i^{n+1^*}\right) = \boldsymbol{F}_i^{\phi,n+1}. \quad i = 1,2,3 \text{ for 3D and } 1,2 \text{ for 2D} \tag{28}$$

where

$$\boldsymbol{F}_{iI}^{\phi,n+1} = -\frac{1}{\rho}\sum_{K\in Z_I}\widehat{m}_K\Psi_I^a(\boldsymbol{x}_K)\sum_{J\in Z_I}\Psi_{J,i}^a(\boldsymbol{x}_K)\phi_J^{n+1}$$

The consistent mass matrix in Eq. (28) can be replaced by the lumped mass matrix, in which only the diagonal element is non-zero. Thus Eq. (28) becomes

$$\frac{\boldsymbol{M}^{\text{lump},n+1}}{\Delta t}\left(\boldsymbol{V}_i^{n+1} - \boldsymbol{V}_i^{n+1^*}\right) = \boldsymbol{F}_i^{\phi,n+1}. \quad i = 1,2,3 \text{ for 3D and } 1,2 \text{ for 2D} \tag{29}$$

where the particle velocity at time step $n+1$ can be explicitly computed and corrected.

3.5 Updating the velocities and positions of the Lagrangian particles

After particle velocities are corrected, particles move with corrected particle velocities, that is

$$\boldsymbol{x}_J^{n+1} = \boldsymbol{x}_J^n + \sum_{I \in Z_I} \Psi_I^a(\boldsymbol{x}_J)\boldsymbol{v}_I^{n+1} \tag{30}$$

The unsmoothed value of particle velocities should be updated for the computation in the next time step using the following equation,

$$\hat{\boldsymbol{v}}_J^{n+1} = (1-\alpha)\left(\hat{\boldsymbol{v}}_J^n + \sum_{I \in Z_I} \Psi_I^a(\boldsymbol{x}_J)(\boldsymbol{v}_I^{n+1} - \boldsymbol{v}_I^n)\right) + \alpha \sum_{I \in Z_I} \Psi_I^a(\boldsymbol{x}_J)\boldsymbol{v}_I^{n+1} \tag{31}$$

Here α is the relaxation factor to control the numerical dissipation 错误!未找到引用源。 given by

$$\Delta E(\alpha) = -\alpha(2-\alpha)\frac{1}{2}\sum_{I \in Z_I} \hat{m}_I \left(\hat{\boldsymbol{v}}_I - \sum_{I \in Z_I} \Psi_I^a(\boldsymbol{x}_J)\boldsymbol{v}_J\right) \tag{32}$$

By varying α from 0 to 1, one can change the amount of dissipation introduced in the system. In our simulations, α is taken as 0.5.

4. LS-DYNA keyword of ISPG

ISPG has been implemented in LS-DYNA as a new 3D element formulation (#49) in the keyword *SECTION_FPD for the semi-implicit dynamic analysis. In other words, ISPG solves the pressure equation implicitly while the velocity field is computed explicitly as that in SPG. The numerical advantages of utilizing semi-implicit analysis for the fully incompressible flow problem are the high accuracy and efficiency in which the numerical diffusion is improved, and critical time steps do not drop abruptly by implicitly solving the pressure equation. Because the momentum equations are well-preserved and the fully incompressible condition is met in the Navier-Stokes equation, the ISPG method is very suitable to simulate free surface flow problems. The ISPG nodes are automatically converted from those of the user's FEM model (8-noded solid element). The input deck of *SECTION_FPD for card 2 and card 3 is described as follows:

Card 2	**DX**	**DY**	**DZ**	ISPLINE	**KERNEL**	BOX	SMSTEP	SSTYPE
	1.8	1.8	1.8	0	0			
Card 3	**MCVISC**	VMAX						
	0.1							

DX,DY,DZ	Normalized dilation parameters of the kernel function in X,Y and Z directions, the recommended range in ISPG is 1.4-1.8
KERNEL	Kernel type. KERNEL=0 Updated Lagrangian (UL) kernel. Currently, only UL kernel is supported.
MCVISC	Relaxation parameter to control the numerical dissipation used in Eq. (32).

Recommended range 0-1

The material property of the fluid in ISPG is defined with the keyword *MAT_IFPD, the card 1 is described as follows:

Card 1	MID	**RO**	**DYNVIS**	**SURFTEN**				
	1	1.8	1.8	0				

RO	Fluid density
DYNVIS	Dynamic viscosity of the fluid
SURFTEN	Surface tension coefficient

The contact between fluid and solid is defined with the keyword *DEFINE_FP_TO_SURFACE_COUPLING, the card 1 and 2 are described as follows:

Card 1	**SLAVE**	**MASTER**	**STYPE**	**MTYPE**				
	1	1	1	0				
Card 2	**SBC**	**SCA**				**SFP**		
	0	**0.5**				**0.1**		

SLAVE	Slave part ID
MASTER	Master segment set ID
STYPE	Slave type, currently only **STYPE=1 (**slave part**)** is supported.
MTYPE	Master type, currently only **MTYPE=0** (segment set) is supported
SBC	Type of boundary condition. **SBC=0**, free-slip boundary; **SBC=1**, non-slip boundary.
SCA	Static (equilibrium) contact angle
SFP	Stiffness coefficient along the normal direction of the contact interface. **SFP<1.0**

5. Numerical Example

The following two examples are solved with the Beta version of LS-DYNA® [19].

5.1 Shape prediction of solder joint with fixed contact angles

In this example, the shape evolution of a cubic solder is simulated, and its final size is compared with the analytical solution. The original size of the cubic solder is $1mm \times 1mm \times 1mm$. The density and surface tension of the solder are $8.93 \times 10^{-6} g/mm^3$ and $4.985 \times 10^{-6} kN/mm$, respectively. The solder is discretized with 11 nodes in every direction shown in Figure 2, which leads to a model having a total of 1331 particles. For the contact between solder and pad, the free-slip boundary condition is applied. Two equilibrium contact angles $\pi/6$ and $\pi/3$ are used separately for the simulation. No gravity is considered in this study, and the total simulation time is 100ms.

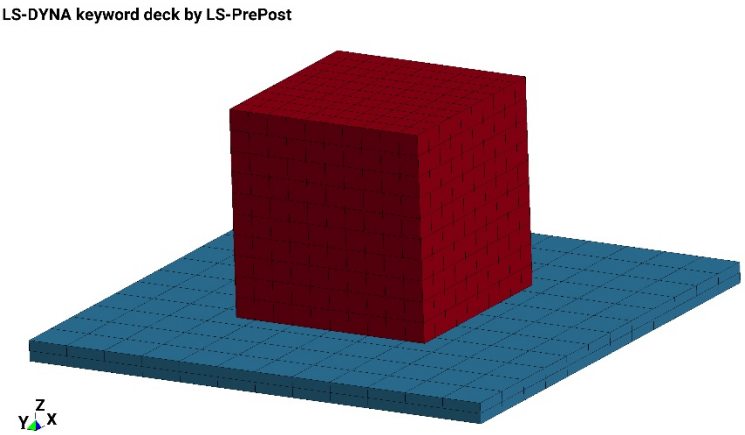


Figure 2. Discretization of the solder and pad

Figure 3 **a** and **b** give the final equilibrium configuration of the solder with the equilibrium contact angles $\pi/6$ and $\pi/3$, respectively. The comparison of final height and width is summarized in Table 1 and Table 2. The difference between the predicted size and the analytical solution is smaller than 5%, despite a very coarse resolution is used in our simulations.

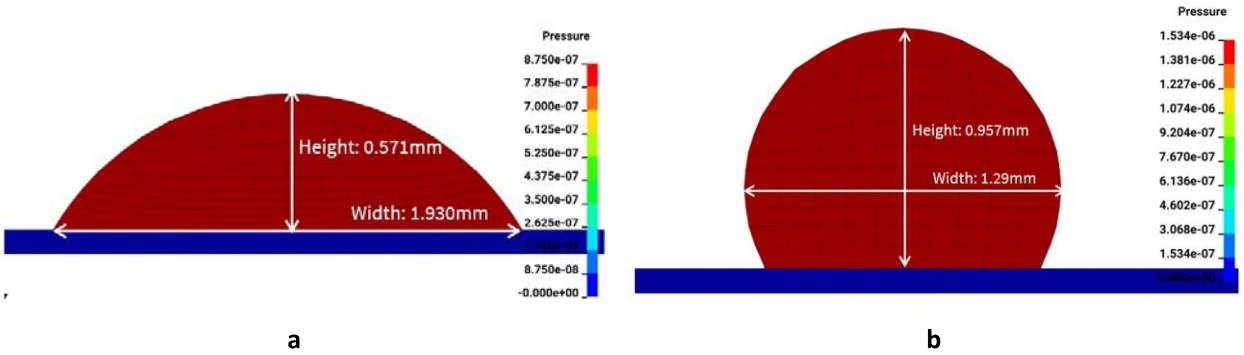


Figure 3. Mesh section view of the final equilibrium configuration of the solder. **a**, result with

equilibrium static angle $\pi/6$; **b**, result with equilibrium static angle $\pi/3$

Table 1. Final solder geometry with equilibrium contact angle $\pi/6$

	Analytical [mm]	ISPG predicted [mm]	Difference
Height	0.576	0.571	0.87%
Width	1.995	1.930	3.26%

Table 2. Final solder geometry with equilibrium contact angle $\pi/3$

	Analytical [mm]	ISPG predicted [mm]	Difference
Height	0.985	0.957	2.84%
Width	1.313	1.290	1.17%

5.2 Prediction of the standoff height and width of the solder joints on BGA

The standoff height and the maximum width of the solder joints are very important in the design process, because they affect significantly the fatigue life of solder joints. In general, the standoff height and the maximum width of solder joints are related to the change of solder volume. In this study, different volumes of the solder are simulated, and the predicted standoff height and maximum with are compared with the experimental result. The solder pad and solder mask are shown in Figure 4. The material of solder pad is Ni-Au and the mask is made of epoxy. The equilibrium contact angles of the molten solder formed on the solder pad and mask are 5° and 148°, respectively. Five different volumes of solder are used for the simulations, and they are $0.0141mm^3, 0.0564mm^3, 0.1077mm^3, 0.1218mm^3$ and $0.2298mm^3$. The solders are discretized with 1365, 2299, 4389, 3355 and 3742 particles, respectively. The density and surface tension of the solder are $8.93 \times 10^{-6} g/mm^3$, $4.985 \times 10^{-6} kN/mm$, respectively. The gravity is $9.81 \times 10^{-3} mm/ms^3$.

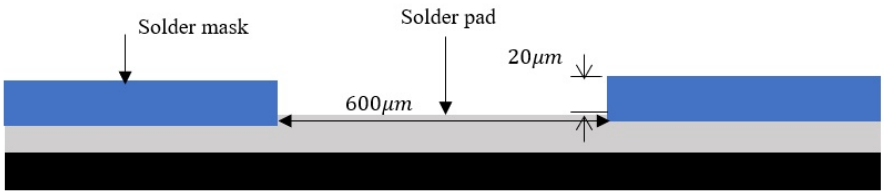
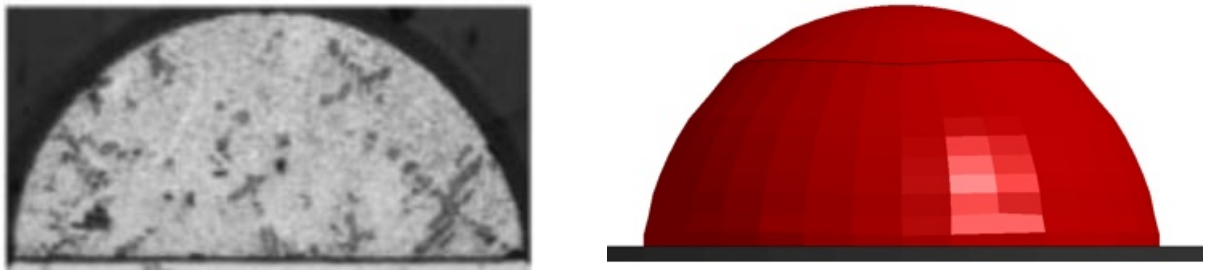


Figure 4. The cross section of BGA substrate.

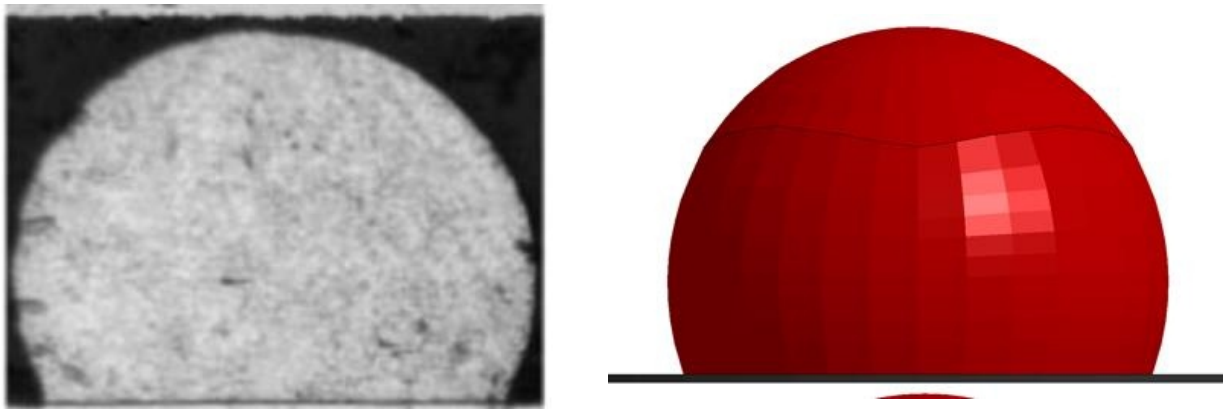
Figure 5 compares the cross sections of solder joints between experimental observations [6] and simulation

results. In general, they agree very well with each other. As shown in Figure 5, the contact angle formed in solder pads increases with the increase of solder volume.

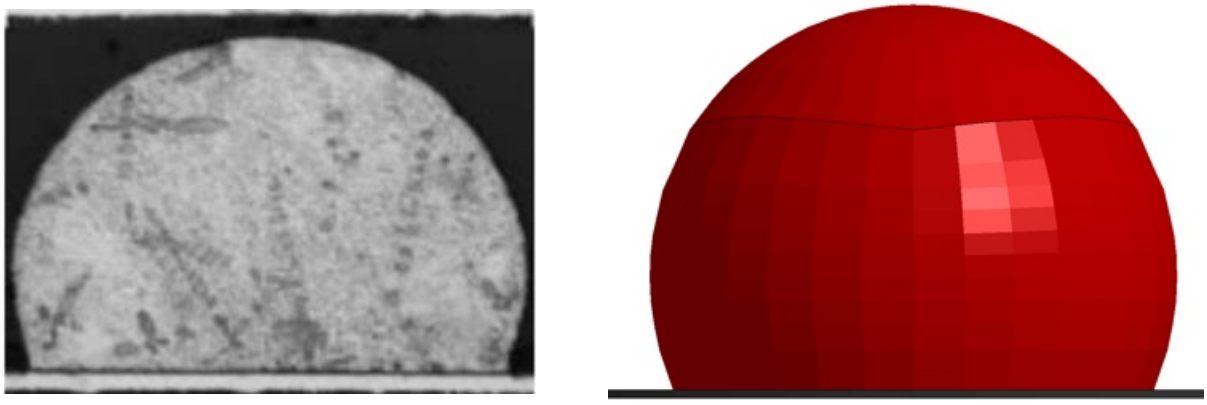
a, volume = $0.0141mm^3$



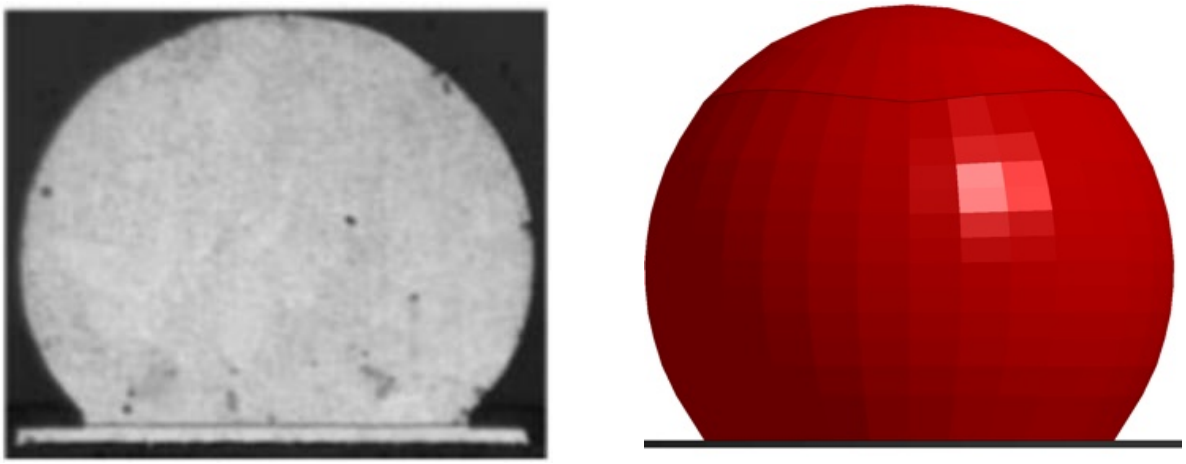
b, volume = $0.0564mm^3$



c, volume = $0.1077mm^3$



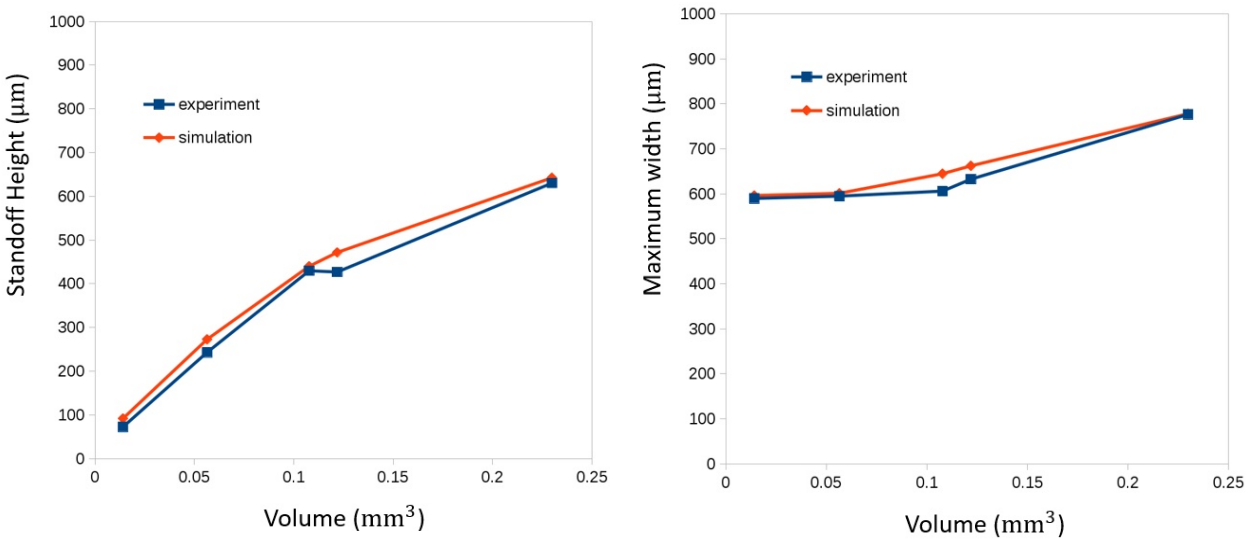
d, volume = $0.1218mm^3$



e, volume = $0.2298mm^3$

Figure 5. Comparison on the cross sections of the solder joints between the experimental observations and the simulation results. Left: experimental observations; right: simulation

Figure 6 compares the simulated standoff height and the maximum width of the solder joints between the simulated results and the experimental measurements [6], and the results match very well. `



a, standoff height vs. solder volume **b**, maximum width vs. solder volume

Figure 6. Comparison of the standoff height and maximum width of the solder joints with experimental data.

6. Conclusions

In this paper, an incompressible smoothed particle Galerkin (ISPG) is developed for the simulation of shape evolution of solder joints in the reflowing process. Shape prediction with fixed contact angles shows that there is little difference between our simulated results and the analytical solution, which verifies that our method is very accurate. The prediction of the standoff height and the maximum width of solder joins on BGA shows that our simulation results agree with the experimental observation.

It is important to note that ISPG is not limited to the simulation of reflow problems. As a matter of fact, ISPG is a very promising method in the simulation of violent free surface fluid flows. Several numerical examples including liquid sloshing with large free surface deformation and dam breaking have been studied to demonstrate the accuracy and stability of the ISPG method in solving 2D fluid flow problems [18]. The application to three-dimensional free surface and thermal fluid flow problems will be presented in the near future.

Acknowledgements

The authors wish to thank Dr. John O. Hallquist of LSTC for his support to this research.

References

[1] Chiang KN, Yuan CA (2001) An Overview of Solder Bump Shape Prediction Algorithm with Validations. IEEE TRANSACTIONS ON ADVANCED PACKAGING 24:158-162

[2] Chiang KN, Chen WL (1998) Electronic packaging reflow shapes prediction for the solder mask defined ball grid array. Journal of Electronic Packaging 120:175–178.

[3] Heinrich SM, Liedtke PE, Nigro N J, Elkouh A F, Lee P S (1993) Effect of chip and pad geometry on solder joint formation in SMT. Journal of Electronic Packaging 115:433–439.

[4] Heinrich SM, Schaefer M, Schroeder S A, Lee P S (1996) Prediction of solder joint geometries in array-type interconnects. Journal of Electronic Packaging 118:114–121.

[5] Brakke KA (1994) Surface Evolver Manual, version 1.95. Minneapolis, MN: The Geometry Center.

[6] Chou YY, Chang HJ, Kuo JH, Hwang WS (2006) The simulation of shape evolution of solder joints during reflow process and its experimental validation. Materials Transactions 47(4):1186-1192.

[7] Smith G (1985) Numerical solution of partial differential equations: finite different methods. Oxford applied Mathematics and computing science series. Clarendon Press, Oxford.

[8] Versteeg H, Malalasekera W (2007) An introduction to computational fluid dynamics: the finite volume method. Pearson Education Limited, Singapore.

[9] Gingold RA, Monaghan JJ (1977) Smoothed particle hydrodynamics – Theory and application to non-spherical stars. Monthly Notices of the Royal Astronomical Society 181:375-389.

[10] Wu CT, Wu Y, Crawford JE, Magallanes JM (2017) Three-dimensional concrete impact and penetration simulations using the smoothed particle Galerkin method. International Journal of Impact Engineering 106:1-17.

[11] Wu CT, Bui TQ, Wu Y, Luo TL, Wang M, Liao CC, Chen PY, Lai YS (2018) Numerical and experimental validation of a particle Galerkin method for metal grinding simulation. Computational Mechanics 58:391-

409.

[12] Pan X, Wu CT, Hu W, Wu Y (2019) A momentum-consistent stabilization algorithm for Lagrangian particle methods in the thermo-mechanical friction drilling analysis. Computational Mechanics 64(3):625-644.

[13] Wu CT, Wu Y, Lyu D, Pan X, Hu W (2019) The momentum-consistent smoothed particle Galerkin (MC-SPG) method for simulating the extreme thread forming in the flow drill screw-driving process. Computational Particle Mechanics. https://doi.org/10.1007/s40571-019-00235-2.

[14] Wu CT, Chi SW, Koishi M, Wu Y (2016) Strain gradient stabilization with dual stress points for the meshfree nodal integration method in inelastic analysis. International Journal for Numerical Methods in Engineering 107:3-30.

[15] Guermond JL, Minev P, Shen J (2006) An overview of projection methods for incompressible flows. Computational Methods in Applied Mechanics and Engineering 195:6011-6045.

[16] Belytschko T, Lu YY, Gu L (1994) Element-free Galerkin methods. International Journal for Numerical Methods in Engineering 37:229-256.

[17] Beissel S, Belytschko T (1996) Nodal integration of the element-free Galerkin method. Computational Methods in Applied Mechanics and Engineering 139:49-74.

[18] Pan X, Wu CT, Hu W (2019) An incompressible smoothed particle Galerkin formulation for free surface fluid flow simulations. Submitted to Journal of Computational Physics.

[19] Hallquist JO (2019) LS-DYNA® Keyword User's Manual, Livermore Software Technology Corporation, Livermore, California.