Title
Modeling and simulation of material removal with particulate flows

Authors
Arbelaez, D
Zohdi, T I
Dornfeld, David

Publication Date
2008-03-26

Peer reviewed

Comput Mech (2008) 42:749–759
DOI 10.1007/s00466-008-0273-3

ORIGINAL PAPER

Modeling and simulation of material removal with particulate flows

D. Arbelaez · T. I. Zohdi · D. A. Dornfeld

Received: 30 October 2007 / Accepted: 3 March 2008 / Published online: 26 March 2008

Abstract In this work a multibody collision model, amenable to large-scale computation, is developed to simulate material removal with particulate flows. This model is developed by computing momentum exchange to account for different force interactions: (1) particle–particle interaction, (2) particle–fluid interaction, and (3) particle–surface interaction. For the particle-fluid interaction, a velocity field for the fluid is assumed to be known, and the drag force on the particles is computed from this field. In the particle–surface interaction, the Boussinesq solution for a point load on an elastic half-space is used along with the von-Mises yield criterion to determine the amount of material removed. Employing this model, inverse problems are then constructed where combinations of the abrasive particle size, the particle size distribution, the flow velocity, etc., are sought to maximize the efficiency of the process. A genetic algorithm is used to treat this inverse problem, and numerical examples are given to illustrate the overall approach.

Keywords Granular flow · Material removal · CMP

1 Introduction

Recently, several modern applications, primarily driven by micro- and nanotechnology, have emerged where successful computational modeling of particulate flows is critical. In this work, we develop a computational strategy for material removal by particulate flows. This includes industrial applications such as sand blasting, abrasive water jet machining and chemical mechanical planarization (CMP). Zohdi [20–23] has dealt with the computational aspects of granular flows including agglomeration in thermo-chemically reacting media, charge-induced clustering, and particle-fluid systems. In this work, a similar computational technique is used and the previous models are extended to include material removal from a surface. Several researchers have performed computational studies on material removal by a single particle. For example, Aquaro and Fontani [1], Aquaro [2], and Junkar et al. [13]. For the present work, the goal is to track the trajectories of the particles and the geometry of the surface over time in an efficient manner. Therefore, a simple material removal model is developed without resorting to computational techniques.

While a general framework of material removal by particulate matter is developed here, a specific example on the mechanical aspects of CMP will be presented. CMP, which involves using abrasive particles embedded in a chemically-reacting fluid to ablate rough small-scale surfaces flat, has become essential for the success of many micro- and nanotechnologies. During the CMP process a wafer is rotated about its axis while it is pressed against a polishing pad. A slurry, composed of nano-sized abrasive particles and chemicals, is deposited on the pad and the wafer is polished by the combined action of the slurry particles, slurry chemicals, and the polishing pad. Perhaps the most widely used empirical material removal equation is Preston's Equation for glass polishing [17],

$$\mathrm{MRR} = K_e P V \tag{1}$$

where MRR is the material removal rate, P is the applied pressure, V is the relative velocity between the wafer and the polishing pad, and K_e is an experimentally determined coefficient. The main problem with Preston's equation is that it depends on the phenomenological parameter K_e. Cook

D. Arbelaez (✉) · T. I. Zohdi · D. A. Dornfeld
Department of Mechanical Engineering,
University of California, Berkeley, CA 94720-1740, USA
e-mail: darbelae@me.berkeley.edu;
darbelae@newton.berkeley.edu

[6] addressed this issue by making the assumption that a spherical particle under uniform load penetrates the surface and moves along at a velocity V removing glass volume of dimensions proportional to the penetration depth. Using this assumption, along with a Hertzian contact model, Cook derived the equation MRR $= \frac{PV}{2E_w}$, where E_w is the Young's modulus of the surface being planarized. This model agrees with Preston's equation and relates the parameter $K_e = \frac{1}{2E_w}$ to the properties of the wafer material.

The above model implies that the abrasives are embedded into the pad and indented into the wafer surface. Beside these kinds of two-body based models, models which assume that the abrasive particles float in the slurry and impact the wafer surface from time to time have also been developed, for example, the model proposed by Su [18]. This is the approach taken in this work where the focus is on material removal by flowing abrasive particles, which corresponds to the hydrodynamical contact mode in CMP. Only some of the classical models for CMP were discussed here, for comprehensive reviews of CMP the reader is referred to Luo and Dornfeld [16] and Li [15]. The general approach taken in this work is to use simple models, and parameters, for each component of the overall model, that are justified alone. The response of the entire system, which could be somewhat complicated, is an outcome of the sub-models.

2 Mathematical formulation

Particulate material removal processes usually involve numerous interactions between the individual components. In this model the following interactions are considered: particle–particle, particle–fluid, and particle–surface interactions. Once these interactions are modeled the position of each particle is determined from the equations of motion

$$m_i \ddot{\mathbf{r}}_i = \mathbf{\Psi}_i^{p-p} + \mathbf{\Psi}_i^{p-f} + \mathbf{\Psi}_i^{p-s} \tag{2}$$

where m_i is the mass of particle i, $\mathbf{r}_i$ is the position of particle i, and $\mathbf{\Psi}_i^{p-p}$, $\mathbf{\Psi}_i^{p-f}$, and $\mathbf{\Psi}_i^{p-s}$ are the forces on particle i due to the particle–particle, particle–fluid, and particle–surface interactions respectively. In the following sections the models for each individual interaction will be introduced.

2.1 Particle–particle interaction

The contact interaction between the grains is computed through a balance of linear momentum in the normal direction. For grain i, the balance of linear momentum in the direction from the center of particle i to the center of particle j ($\mathbf{n}_{ij}$) is given by

$$m_i v_{in}(t) - \delta t \overline{I}_n + \int_t^{t+\delta t} E_{in} dt = m_i v_{in}(t+\delta t) \tag{3}$$

where t corresponds to the time immediately before the particles come into contact, δt is the impact duration time, $t+\delta t$ corresponds to the time immediately after the collision occurs, $\delta t \overline{I}_n = \int_t^{t+\delta t} I_n dt$, and $E_{in} = \mathbf{E}_i \cdot \mathbf{n}_{ij}$, where $\mathbf{E}_i$ is the sum of external forces acting on particle i. In this case $\overline{I}_n \geq 0$ is interpreted to be the average normal impulsive force. Similarly, for grain j, the balance of linear momentum in the normal contact direction gives

$$m_j v_{jn}(t) + \delta t \overline{I}_n + \int_t^{t+\delta t} E_{jn} dt = m_j v_{jn}(t+\delta t), \tag{4}$$

where $E_{jn} = \mathbf{E}_j \cdot \mathbf{n}_{ij}$ and $\mathbf{E}_j$ is the sum of external forces acting on particle j. In order to solve for the velocities after impact an estimate for the coefficient of restitution, e, must be provided. The coefficient of restitution is defined as

$$e = \frac{\int_{t+\delta t_1}^{t+\delta t} I_n dt}{\int_t^{t+\delta t_1} I_n dt}. \tag{5}$$

It is important to realize that, in reality, the phenomenological parameter e depends on the severity of the impact velocity. For extensive experimental data, see Goldsmith [10].

The balance of momentum for both grains can be divided into a compression phase, occurring in the time interval $(t, t+\delta t_1)$, and a recovery phase, occurring in the time interval $(t+\delta t_1, t+\delta t)$. For the compression phase the balance of momentum for grains i and j in the normal direction is

$$\begin{aligned} m_i v_{in}(t) - \int_t^{t+\delta t_1} I_n dt + \int_t^{t+\delta t_1} E_{in} dt &= m_i v_{cn} \\ m_j v_{jn}(t) + \int_t^{t+\delta t_1} I_n dt + \int_t^{t+\delta t_1} E_{jn} dt &= m_j v_{cn} \end{aligned}, \tag{6}$$

where v_{cn} is a common velocity for both grains, since their relative velocity in the normal direction vanishes at time $t+\delta t_1$. Similarly, for the recovery phase, the balance of momentum for grains i and j in the normal direction is

$$\begin{aligned} m_i v_{cn} - \int_{t+\delta t_1}^{t+\delta t} I_n dt + \int_{t+\delta t_1}^{t+\delta t} E_{in} dt &= m_i v_{in}(t+\delta t) \\ m_j v_{cn} + \int_{t+\delta t_1}^{t+\delta t} I_n dt + \int_{t+\delta t_1}^{t+\delta t} E_{jn} dt &= m_j v_{jn}(t+\delta t) \end{aligned}. \tag{7}$$

Using Eqs. 5, 6, and 7, the coefficient of restitution can be written as

$$e = \frac{v_{jn}(t+\delta t) - v_{in}(t+\delta t) + \Delta_{ij}(t+\delta t_1, t+\delta t)}{v_{in}(t) - v_{jn}(t) + \Delta_{ij}(t, t+\delta t_1)}, \tag{8}$$

where $\Delta_{ij}(t+\delta t_1, t+\delta t) = \frac{1}{m_i}\int_{t+\delta t_1}^{t+\delta t} E_{in}dt - \frac{1}{m_j}\int_{t+\delta t_1}^{t+\delta t} E_{jn}dt$ and $\Delta_{ij}(t, t+\delta t_1) = \frac{1}{m_i}\int_{t}^{t+\delta t_1} E_{in}dt - \frac{1}{m_j}\int_{t}^{t+\delta t_1} E_{jn}dt$. Using Eqs. 3, 4, and 8 the normal impulse is given by

$$\overline{I}_n\delta t = \frac{(1+e)m_i m_j}{m_i+m_j}\left[v_{in}(t) - v_{jn}(t) + \frac{1}{m_j}\int_{t}^{t+\delta t_1} E_{in}dt - \frac{1}{m_i}\int_{t}^{t+\delta t_1} E_{jn}dt\right]. \tag{9}$$

As an approximation, Eq. 3 can be discretized in time, with time step Δt, giving the following equation

$$m_i v_{in}(t) - \overline{I}_n\delta t + \overline{E}_{in}\delta t \approx m_i v_{in}(t+\Delta t) \tag{10}$$

where $\delta t << \Delta t$ and $\overline{E}_{in}\delta t = \int_t^{t+\delta t} E_{in}dt$. Equation 2 can now be rewritten as

$$m_i\frac{v_{in}(t+\Delta t) - v_{in}(t)}{\Delta t} \approx -\frac{\overline{I}_n\delta t}{\Delta t} + \frac{\overline{E}_{in}\delta t}{\Delta t} \tag{11}$$

noting that $m_i\frac{v_{in}(t+\Delta t)-v_{in}(t)}{\Delta t}$ is a difference formula for $m_i\dot{v}_{in}$, then the normal contact contribution of particle–particle interaction force can be defined as

$$\boldsymbol{\Psi}_i^c = -\frac{\overline{I}_n\delta t}{\Delta t}\mathbf{n}_{ij}. \tag{12}$$

The particle–particle interactions also include a contribution due to frictional interactions between particles during collisions. To incorporate the friction force the relative tangential velocities between particles i and j are computed as follows

$$\mathbf{v}_{jt}(t) - \mathbf{v}_{it}(t) = (\mathbf{v}_j(t) - \mathbf{v}_i(t)) - ((\mathbf{v}_j(t) - \mathbf{v}_i(t))\cdot\mathbf{n}_{ij})\mathbf{n}_{ij}. \tag{13}$$

First it is assumed that particle i and j stick on contact. For this case the particles have the same velocity, v_{ct}, after the collision. The balance of linear momentum for particle i can then be written as

$$m_i v_{it}(t) - \delta t\overline{I}_f + \delta t\overline{E}_{it} = m_i v_{ct}, \tag{14}$$

where $\delta t\overline{I}_f = \int_t^{t+\delta t} I_f dt$ and $\delta t\overline{E}_{it} = \int_t^{t+\delta t}\mathbf{E}_i\cdot\mathbf{t}dt$, so that $\overline{I}_f$ is interpreted to be the average frictional force for the collision and $\overline{E}_{it}$ is the average external force in the tangential direction ($\mathbf{t} = \frac{\mathbf{v}_{jt}-\mathbf{v}_{it}}{\|\mathbf{v}_{jt}-\mathbf{v}_{it}\|}$) over the impact period. Likewise, the balance of linear momentum for particle j can then be written as

$$m_j v_{jt}(t) + \delta t\overline{I}_f + \delta t\overline{E}_{jt} = m_j v_{ct}. \tag{15}$$

Equations 14 and 15 involve two unknowns $\overline{I}_f$ and v_{ct}, the quantity of interest $\overline{I}_f$ is given by

$$\overline{I}_f = \frac{\left(\frac{\overline{E}_{it}}{m_i} - \frac{\overline{E}_{jt}}{m_j}\right)\delta t + v_{it}(t) - v_{jt}(t)}{\left(\frac{1}{m_i}+\frac{1}{m_j}\right)\delta t}. \tag{16}$$

In Eq. 16 $\overline{I}_f$ is computed under the assumption that no slip occurs. The validity of this assumption is now checked, if

$$|\overline{I}_f| > \mu_s|\overline{I}_n|, \tag{17}$$

where μ_s is the static coefficient of friction, then slip occurs and the magnitude of the force due to friction is given by $\mu_d\|\boldsymbol{\Psi}_i^c\|$, where μ_d is the dynamic coefficient of friction. In summary, the friction force in the particle–particle interactions is given by

$$\boldsymbol{\Psi}_i^f = \begin{cases} \frac{|\overline{I}_f|\delta t}{\Delta t}\mathbf{t} & \text{if } |\overline{I}_f| \le \mu_s|\overline{I}_n| \\ \mu_d\|\boldsymbol{\Psi}_i^c\|\mathbf{t} & \text{if } |\overline{I}_f| > \mu_s|\overline{I}_n| \end{cases} \tag{18}$$

and the particle–particle interaction force is given by

$$\boldsymbol{\Psi}_i^{p-p} = \boldsymbol{\Psi}_i^c + \boldsymbol{\Psi}_i^f. \tag{19}$$

2.2 Particle–fluid interaction

The interaction force between an individual particle and the surrounding fluid is due to the drag induced by the flow around a particle. The characteristics of the flow can be determined using computational fluid dynamics (e.g., using finite difference, finite volume, or finite element methods). However, in this simple simulation the velocity field of the slurry is assumed to be known, and the drag force on the particles is calculated from this field. In order to determine the force that the fluid exerts on the particles an effective drag coefficient, C_D, is used. This coefficient is generally determined from experiments for an object of a given shape and for varying flow conditions. The drag coefficient is defined as

$$C_D = \frac{\mathcal{D}}{\frac{1}{2}\rho\|\mathbf{u}-\mathbf{v}_i\|^2 A} \tag{20}$$

where $\mathcal{D}$ is the magnitude of the drag force on the object, ρ is the density of the fluid, $\mathbf{u}$ is the velocity of the flow at the center of particle i, $\mathbf{v}_i$ is the velocity of particle i, and A is the projected area of the object in the plane perpendicular to the flow. The drag coefficient is a function of the Reynolds number $Re = \frac{\rho U d}{\mu}$, where μ is the viscosity of the fluid and, for a spherical object, d is the diameter. One possible way to represent the drag coefficient for a sphere is with a piecewise

definition, as a function of the Reynolds number (Chow [5]):

$$C_D = \begin{cases} \frac{24}{Re} & \text{if } 0 < Re \leq 1 \\ \frac{24}{Re^{0.646}} & \text{if } 1 < Re \leq 400 \\ 0.5 & \text{if } 400 < Re \leq 3 \times 105 \\ 0.000366 Re^{0.4275} & \text{if } 3 \times 105 < Re \leq 2 \times 106 \\ 0.18 & \text{if } 2 \times 106 < Re < \infty \end{cases} . \tag{21}$$

The particle–fluid interaction for particle i is then given by

$$\boldsymbol{\Psi}_i^{p-f} = \mathcal{D} \frac{\mathbf{u} - \mathbf{v}_i}{\|\mathbf{u} - \mathbf{v}_i\|}. \tag{22}$$

2.3 Particle–surface interaction

The particle–surface interaction includes the particle surface impact force along with material removal from the surface. The material that is removed from the surface is assumed to be carried away by the particle, so that during the impact the mass of the particle increases. This is the material removal mechanism that is considered in this derivation; however, it is also possible that the material is carried away by the fluid after the particle has removed it from the surface. Since the mass of the particle changes this must be taken into account in the linear momentum balance. The rate of change of linear momentum for particle i is given by

$$\dot{\mathbf{L}}_i = m_i \dot{\mathbf{v}}_i + \dot{m}_i \mathbf{v}_i = \boldsymbol{\Psi}_i^{\text{tot}} \tag{23}$$

where $\boldsymbol{\Psi}_i^{\text{tot}}$ is the total external force acting on the particle which includes the force from the impact as well as any other external forces such as the drag force. Integrating Eq. 23 from time t to $t + \delta t$ and taking the inner product with the normal to the wafer surface, $\mathbf{n}$, the following momentum balance for particle i in the normal impact direction is derived,

$$\begin{aligned} &m_i(t) v_{in}(t) + \delta t \overline{I}_n + \delta t \overline{E}_{in} \\ &\quad = \int_t^{t+\delta t} \dot{m}_i v_{in} dt + m_i(t + \delta t) v_{in}(t + \delta t) \end{aligned} \tag{24}$$

where $\delta t \overline{E}_{in} = \int_t^{t+\delta t} \mathbf{E}_i \cdot \mathbf{n}$ and as before $\delta t \overline{I}_n = \int_t^{t+\delta t} I_n dt$. Notice that the mass of particle i is now a function of time. Making the assumption that all of the mass is picked up at the time $t + \delta t_1$, which corresponds to the time when the normal velocity of particle i is zero, then the rate of change of mass can be described by the function $\dot{m}_i = \Delta m \delta(t - (t + \delta t_1))$ where Δm is the total increase in mass and $\delta(\cdot)$ is the dirac delta function, this is illustrated in Fig. 1. The term in Eq. (24) involving $\dot{m}_i$ is given by,

$$\begin{aligned} \int_t^{t+\delta t} \dot{m}_i v_{in} dt &= \int_t^{t+\delta t} \Delta m \delta(t - (t + \delta t_1)) v_{in} dt \\ &= \Delta m v_{in}(t + \delta t_1) = 0 \end{aligned} \tag{25}$$

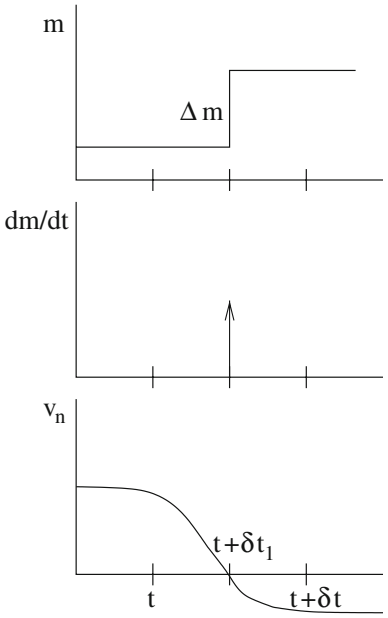

Fig. 1 The material is assumed to be removed at the instant $t + \delta t_1$

where the right hand side is true because $v_{in}(t + \delta t_1) = 0$ since the surface is not moving. Equation 24 can now be rewritten as

$$m_i(t) v_{in}(t) + \delta t \overline{I}_n + \delta t \overline{E}_{in} = m_i(t + \delta t) v_{in}(t + \delta t). \tag{26}$$

The balance of linear momentum can now be split into the compression phase which occurs in the time interval $(t, t + \delta t_1)$ and the recovery phase which occurs in the time interval $(t + \delta t_1, t + \delta t)$:

$$\begin{aligned} m_i(t) v_{in}(t) + \int_t^{t+\delta t_1} I_n dt + \int_t^{t+\delta t_1} E_{in} dt = 0 \\ \int_{t+\delta t_1}^{t+\delta t} I_n dt + \int_{t+\delta t_1}^{t+\delta t} E_{in} dt = m_i(t + \delta t) v_{in}(t + \delta t), \end{aligned} \tag{27}$$

where $m_i(t + \delta t) = m_i(t) + \Delta m$. Using the definition given in Eq. 5, the coefficient of restitution is given by

$$e = \frac{(m_i(t) + \Delta m) v_{in}(t + \delta t) - \int_{t+\delta t_1}^{t+\delta t} E_{in} dt}{-m_i(t) v_{in}(t) - \int_t^{t+\delta t_1} E_{in} dt}. \tag{28}$$

Using Eqs. 26 and 28, the velocity of particle i in the normal direction after impact is

$$\begin{aligned} v_{in}(t + \delta t) = \frac{1}{m_i(t) + \Delta m} \Bigg[&\int_{t+\delta t_1}^{t+\delta t} E_{in} dt \\ &- e \left(m_i(t) v_{in}(t) + \int_t^{t+\delta t_1} E_{in} dt \right) \Bigg]. \end{aligned} \tag{29}$$

With $v_{in}(t+\delta t)$ known the average normal impulsive force is computed as

$$\overline{I}_n = \frac{(m_i(t)+\Delta m)v_{in}(t+\delta t) - m_i(t)v_{in}(t)}{\delta t} - \overline{E}_{in}. \quad (30)$$

Note that setting $\Delta m = 0$ corresponds to the mechanism where the fluid carries away the material. Similar to the interaction force for the particle–particle contact given in the previous section, the particle–surface interaction force is given by

$$\boldsymbol{\Psi}_i^{p-s} = \frac{\delta t \overline{I}_n}{\Delta t}\mathbf{n}. \quad (31)$$

In order to determine this interaction force the amount of material removed from the wafer, Δm, must be determined. A method for computing the amount of material removed is given in the following section.

2.4 Material removal

There have been numerous studies on spalling and abrasion. For classical examples of particle abrasion models see Finnie [8] and Bitter [3,4]. Figure 2 indicates the stress distribution under a typical particle. In [11], it is shown experimentally that an area similar to the envelope represented by the dashed line in Fig. 2 is removed from contact stresses in ceramic materials. In this section, a model based on this experimental observation is derived. First, the linear elasticity result for a point load on a half-space (Boussinesq solution see [12]) is used. Knowing the stress state in the material near the point of impact, the depth of indentation into the wafer surface can be estimated. This is done by using the von-Mises yield criterion to determine the region where the yield stress of the material is exceeded. The stress components in the material near the point load are given by

$$\begin{aligned}
\sigma_r &= \frac{P}{2\pi}\left((1-2\nu)\left(\frac{1}{r^2}-\frac{z}{\gamma r^2}\right)-\frac{3zr^2}{\gamma^5}\right)\\
\sigma_\theta &= -\frac{P}{2\pi}(1-2\nu)\left(\frac{1}{r^2}-\frac{z}{\gamma r^2}-\frac{z}{\gamma^3}\right)\\
\sigma_z &= -\frac{3P}{2\pi}\frac{z^3}{\gamma^5}\\
\tau_{rz} &= -\frac{3P}{2\pi}\frac{rz^2}{\gamma^5}
\end{aligned} \quad (32)$$

where $\gamma = \sqrt{r^2+z^2}$ and $\tau_{r\theta}, \tau_{z\theta} = 0$. The coordinates (r,θ,z) are shown in Fig. 2 where the point load P is applied at the origin. In this model P is taken to be the average normal impulse force, $\bar{I}_n$, which is determined from Eqs. 29 and 30 for the particle–surface interaction force. In order to determine the region where plastic deformation occurs, the von-Mises yield criterion is used. The criterion is given by

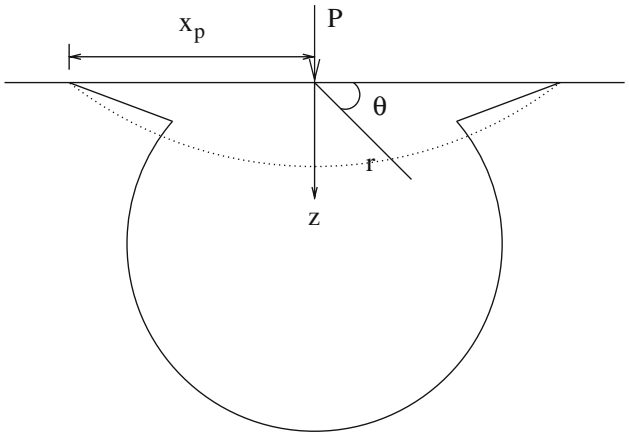

Fig. 2 Region where yield stress is exceeded for a point load on a half space

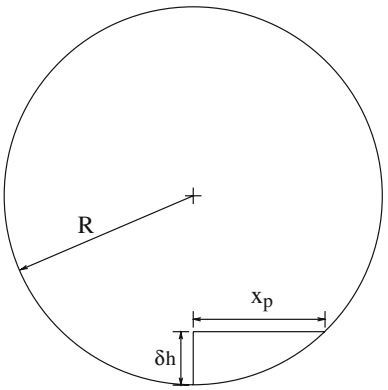

Fig. 3 Spherical cap of material assumed to be removed from the wafer surface

$$\begin{aligned}
2\bar{\sigma}^2 &= (\sigma_r-\sigma_\theta)^2+(\sigma_\theta-\sigma_z)^2+(\sigma_z-\sigma_r)^2\\
&\quad +6(\tau_{r\theta}^2+\tau_{\theta z}^2+\tau_{rz}^2),
\end{aligned} \quad (33)$$

and it states that the material yields when $\bar{\sigma} \geq \sigma_y$, where σ_y is the yield stress of a material as determined from a tension test. Figure 2 shows a representative failure envelope (region inside the solid line) that is seen when the Von Mises yield criterion is used. It is not expected that all of the material inside the envelope is removed in the collision, in this model it is assumed that the amount of material removed is the region inside the dotted line in Fig. 2. In order to determine the size of this region the length x_p is derived by setting $z=0$ in Eq. 32 and substituting into Eq. 33, giving

$$x_p = \left[\frac{\sqrt{3}P(1-2\nu)}{2\pi\sigma_y}\right]^{1/2}. \quad (34)$$

The region in the dotted line in Fig. 2 is a spherical cap from a sphere with radius R, this is shown in Fig. 3. It is now assumed that the radius of the spherical cap is the same

as the radius the particle that collided with the wafer surface ($R = d_i/2$). With this assumption in place the indentation depth δh is given by

$$\delta h = R - \sqrt{R^2 - x_p^2}. \tag{35}$$

The volume contained in the spherical cap is $V_{\text{cap}} = \frac{1}{6}\pi\delta h(3x_p^2+\delta h^2)$, given this quantity the amount of material removed from the wafer surface is

$$\Delta m = \frac{1}{6}\pi\rho_s\delta h(3x_p^2 + \delta h^2) \tag{36}$$

where ρ_s is the density of the surface material material.

3 Algorithms

3.1 Temporal discretization and iterative solution

In this section an iterative solution scheme using the implicit Euler method with fixed point iterations is described. A first order integration scheme is used since the collisions (particle–particle and particle–surface) limit the accuracy of the method to first order (the overlap in a collision is of order dt). For the first order system $\dot{y} = f$, the implicit Euler scheme is given by $y^{L+1} = y^L + \Delta t f(y^{L+1})$, where the superscripts L and $L+1$ correspond to the times t and $t+\Delta t$ respectively. Using Eq. 2 and the definition of the implicit Euler method above, the velocity and position of particle i can be discretized as

$$\begin{aligned}\dot{\mathbf{r}}_i^{L+1} &= \dot{\mathbf{r}}_i^L + \Delta t\boldsymbol{\Psi}_i^{\text{tot}}(\mathbf{r}_j^{L+1})\\ \mathbf{r}_i^{L+1} &= \mathbf{r}_i^L + \Delta t\dot{\mathbf{r}}_i^{L+1}\end{aligned}. \tag{37}$$

Substituting the lower equation into the top equation, the final form is derived as

$$\mathbf{r}_i^{L+1} = \frac{\Delta t^2}{m_i}\boldsymbol{\Psi}_i^{\text{tot}}(\mathbf{r}_j^{L+1}) + \mathbf{r}_i^L + \Delta t\dot{\mathbf{r}}_i^L \tag{38}$$

To solve Eq. 38 an iterative scheme must be used. One possible solution scheme is Newton's method; however, the tangent matrix may not be easy to form due to the lack of smoothness of the trajectories of the particles. This lack of smoothness is produced by the impacts of particles with other particles or surfaces. Another option is to use a fixed point iteration which usually converges at a slower rate but is expected to be more robust. Using this choice, an algorithm to adapt the time step size in order to keep the system stable, can also be derived. In order to solve Eq. 38 with a fixed point iteration, it is written as

$$\mathbf{r}_i^{L+1,k+1} = \frac{\Delta t^2}{m_i}\boldsymbol{\Psi}_i^{\text{tot}}\left(\mathbf{r}_j^{L+1,k}\right) + \mathbf{r}_i^L + \Delta t\dot{\mathbf{r}}_i^L \tag{39}$$

where k indicates the iteration count. Zohdi [20] shows that the error at iteration k can be bounded by $\|e^k\| \leq \eta^k\|e^0\|$, where $\eta \propto EIG(\frac{\Delta t^2}{m_i}\boldsymbol{\Psi}_i^{\text{tot}}) \propto \frac{\Delta t^2}{m_i}$. Therefore, the fixed point iteration is guaranteed to converge when $\eta < 1$, implying that the method converges as long as the time step size is small enough. If convergence is slow within a time step, the time step can be reduced to increase the rate of convergence; however, it is desirable that the time step-sizes are maximized in order to reduce the computation time. Using the rate of convergence of the fixed point iteration, an adaptive time stepping scheme can be derived by approximating η as

$$\eta \approx S(\Delta t)^p \tag{40}$$

where S is a constant, using this approximation the error at iteration k is approximated as

$$\|e^k\| \approx (S(\Delta t)^p)^k\|e^0\|. \tag{41}$$

The goal is to meet an error tolerance in exactly a preset number of iterations, which leads to following equation,

$$\text{TOL} \approx (S(\Delta t_{\text{tol}})^p)^{k_d}\|e^0\|. \tag{42}$$

where TOL is the preset tolerance and k_d is the desired number of iterations for convergence. Given Eqs. 41 and 42, the time step size needed to achieve convergence in k_d iterations is approximated as

$$\Delta t_{\text{tol}} = \left(\frac{\left(\frac{\text{Tol}}{e^0}\right)^{\frac{1}{pk_d}}}{\left(\frac{e^k}{e^0}\right)^{\frac{1}{pk}}}\right)\Delta t \tag{43}$$

so that when the iteration does not converge in the desired number of iterations the new time step size can be computed from the previous time step size used. If the solution does converge in less than k_d iterations then the time step size for the next time step can be computed to achieve convergence in the desired number of iterations. When this occurs the method computes a larger time step size for the next time step. It is important that the time step size does not increase above a point where collisions are missed or the truncation error becomes too large. For this reason, a maximum time step size Δt^{lim} is chosen to ensure that the physics of the process are not lost in the numerical discretization.[1] A description of the time stepping scheme is given in the following list.

[1] In this case $\Delta t^{\text{lim}} = \frac{\tilde{d}\gamma}{V_{\text{pad}}}$ where γ is a parameter, in this case taken to be $\gamma = 0.01$.

Summary of Time Stepping Scheme:

1. Set $k = 0$
2. Compute position of particles $i = 1$ to N:
   $\mathbf{r}_i^{L+1} = \frac{\Delta t^2}{m_i} \boldsymbol{\Psi}_i^{\text{tot}} \left(\mathbf{r}_j^{L+1} \right) + \left(\mathbf{r}_i^L + \Delta t \dot{\mathbf{r}}_i^L \right)$
3. Compute the error for the k^{th} iteration:
   (a) $\epsilon_k \doteq \frac{\sum_{i=1}^{n} \| \mathbf{r}_i^{L+1,k} - \mathbf{r}_i^{L+1,k-1} \|}{\sum_{i=1}^{n} \| \mathbf{r}_i^{L+1,k} \|}$
   (b) $\Phi_k \doteq \left(\frac{\left(\frac{\text{Tol}}{e^0} \right)^{\frac{1}{pk_d}}}{\left(\frac{e^k}{e^0} \right)^{\frac{1}{pk}}} \right)$
4. If tolerance is met ($\epsilon_k \leq TOL$) and $k < k_d$:
   (a) Construct new time step: $\Delta t = \Phi_k \Delta t$
   (b) Select minimum: $\Delta t = MIN(\Delta t^{lim}, \Delta t)$
   (c) Increment time ($t = t + \Delta t$) and go to (1)
5. If tolerance is not met ($\epsilon_k > TOL$) and $k = k_d$:
   (a) Construct new time step: $\Delta t = \Phi_k \Delta t$
   (b) Restart at time t and go to (1)

3.2 Parameter optimization

In material removal processes there are a large number of input parameters which control the quality of the process. For example these include abrasive quantity, abrasive size, abrasive size distribution, and many other parameters. The computational model described in the previous chapters can be used as a design tool by combining it along with an optimization method in order to find optimal input parameters for a given desired output. This is achieved by defining a cost function which can be minimized using some type of optimization algorithm. The cost function can be chosen to optimize different output parameters for example this function can be chosen to produce a high material removal rate while minimizing the amount of waste from consumable materials. Perhaps, the simplest cost function is one which produces the flattest surface for a given amount of time, for this case the cost function, Π, is given as

$$\Pi = \frac{\sum_{i=1}^{N_{\text{node}}} (|z_i(t) - z_o|)^q}{\sum_{i=1}^{N_{\text{node}}} (|z_i(t_o) - z_o|)^q} \tag{44}$$

where $z_i(t)$ is the height of node i at time t, $z_i(t_o)$ is the initial height of node i, and z_o is the desired final height for all of the nodes. Since z_o is the same for all of the nodes, the desired final configuration is a flat surface after the a material removal process has run for a time t. The exponent q in Eq. 44 controls the effect of the distribution of the final peak heights, for example a flat surface with only one large peak may produce a higher cost than a surface with many small peaks if q is large, however, if q is small the surface with many small peaks may have higher cost than the surface with one large peak.

In order to minimize the cost function, several methods can be used such as gradient based methods; however, these methods require that the cost function be differentiable and in order to guarantee convergence to a global minimum the cost function must be globally convex. Since the cost function for these optimization problem is expected to be non-convex (multiple minima and maxima) and non-differentiable a genetic algorithm can be used to circumvent these difficulties. Genetic algorithms are insensitive to gradient fluctuations; therefore, the function is guaranteed to be minimized with every step. For reviews of these methods, see for example, Davis [7], Goldberg [9], and Lagaros et al. [14]. The genetic algorithm used here follows the approach taken by Zohdi [20]. This involves initially generating random populations of S genetic strings $\boldsymbol{\Lambda}^i = \{\Lambda_1^i, \Lambda_2^i, \Lambda_3^i, \ldots, \Lambda_N^i\}$ which contain the N parameters. The parameters chosen must lie within a chosen parameter space so that the search is limited to only a feasible region ($\Lambda^{i-} \leq \Lambda^i \leq \Lambda^{i+}$). For example the average diameter of the abrasives will have a lower and upper limit ($\overline{d}^- \leq \overline{d} \leq \overline{d}^+$). After generating the original random populations, the cost function in Eq. 44 is evaluated for each genetic string. The genetic strings are then ranked according to their "fitness", where the strings with the lowest cost are the most fit. The genetic strings are combined to create two new offspring for each pair of parent strings and the worst performing genetic strings are then eliminated from the population. This process is repeated for several generations until the cost of the best performing string no longer decreases. Note that the top performing parent strings are kept in the population in order to guarantee that the cost of the most fit genetic string decreases monotonically after each generation.

Summary of Genetic Algorithm:

1. Randomly generate population of starting (S) genetic strings,
   $\boldsymbol{\Lambda}^i = \{\Lambda_1^i, \Lambda_2^i, \Lambda_3^i, \ldots, \Lambda_N^i\}, (i = 1, S)$
2. Compute the fitness of each string $\Pi(\boldsymbol{\Lambda}^i)$
3. Rank the genetic strings according to their fitness
4. Mate the nearest pairs of the S strings to produce two offspring for each pair:
   $\boldsymbol{\lambda}^i = \Phi^I \boldsymbol{\Lambda}^i + (1 - \Phi^I) \boldsymbol{\Lambda}^{i+1}$
   $\boldsymbol{\lambda}^{i+1} = \Phi^{II} \boldsymbol{\Lambda}^i + (1 - \Phi^{II}) \boldsymbol{\Lambda}^{i+1}$,
   where ($0 \leq \Phi^I, \Phi^{II} \leq 1$) are uniformly distributed random numbers
5. Replace the bottom M strings with new randomly generated strings while keeping the top K offspring and the top $K = (S - M)/2$ parents
6. Repeat steps 2–5 for G generations

4 Representative example

In this section the results from a polishing simulation are presented. In this simulation, the particles, which are usually on the order of hundreds of nanometers in CMP, are assumed to be floating in a thin layer of fluid between a pad and the surface. The pad surface is modeled as a rigid plane that translates horizontally at a given velocity V_{pad} and moves vertically downward as the material is removed. From the relative motion between the pad and the surface, a linear velocity profile is assumed for the fluid layer. This relative velocity is on the order of one m/s in CMP.

The simulations are performed over a square region of dimension $D \times D$ with the height determined by the pad and wafer surfaces. During the simulation, if a particle escapes from the control volume, the position component is reversed and the same velocity component is retained. In the polishing process the topography of the wafer surface is never known a priori, therefore, in the simulation a randomly generated wafer surface is used. The first step of the simulation is to generate this surface, an example of the initial wafer configuration is illustrated in Fig. 4. The surface is generated by selecting a node on the surface and raising this node and three adjacent nodes to a height h_a, where the four nodes form the corners of a square, to create an asperity. CMP is commonly used following etching and deposition processes. During the etching process, material is removed forming trenches with nearly vertical walls. In many processes this is followed by a deposition step in which another material is deposited, forming asperities that are similar in shape to those used in the simulation. These asperities usually form a structured pattern; however, for this work the interest is on the effect of the process parameters on an arbitrary surface pattern. For this reason we use randomly generated asperities. To form the test surface, the asperities are repeatedly created N_a times at random positions, where N_a is the number of asperities on the surface. A useful measure in the simulations is the asperity density, $\rho_a \doteq \frac{2N_a}{N_e}$, where N_e is the total number of elements that make up the surface. This is a measure of the ratio of the projected area covered by the asperities to the total projected area. Once the surface is created, N particles are introduced using a random sequential packing algorithm. In the polishing process the abrasive size is only known as a statistical distribution, for this reason the minimum and maximum diameters are specified and the diameter of each individual particle is determined as $d_i = d_{\min} + (d_{\max} - d_{\min})\Phi^I$, where $0 \leq \Phi^I \leq 1$ is a uniformly distributed random number. The quantity $\bar{d} = \frac{d_{\min}+d_{\max}}{2}$ is defined as the average diameter and $\delta d \doteq \frac{d_{\max}-d_{\min}}{2\bar{d}}$ is a measure of the relative deviation in the distribution of the particle diameters. The initial velocity of each particle is computed as $\dot{\mathbf{r}} = \dot{\mathbf{r}}_{\min} + (\dot{\mathbf{r}}_{\max} - \dot{\mathbf{r}}_{\min})\Phi^I$ where once again $0 \leq \Phi^I \leq 1$ is determined using a random number generator. With the initial positions and velocities of the particles in place the velocity of the pad, V_{pad}, is specified and the simulation is run for a total time T.

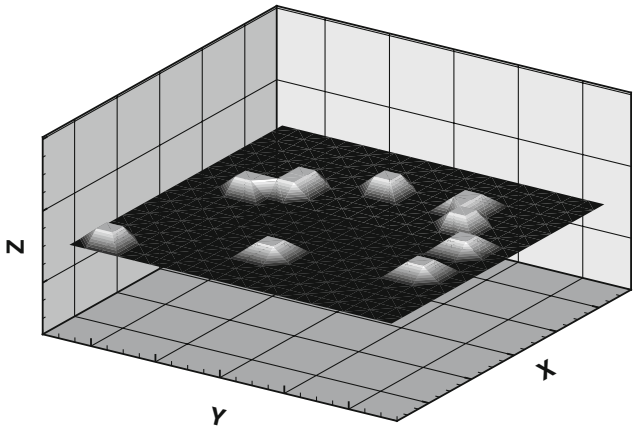

Fig. 4 Randomly generated initial configuration of the wafer surface. The surface is generated by randomly selecting four adjacent nodes that form the corners of a square and raising them to create an asperity

Figure 5 shows an example of a typical simulation where snapshots are taken at equal time intervals. For this example the following parameters were used: $N = 73$, $\bar{d} = 0.9\ \mu$m, $\delta d = 0.1$, $h_a = 0.5\ \mu$m, $V_{\text{pad}} = 1$ m/s, $D = 20\ \mu$m, $\rho_a = 0.025$, and $T = 1$ s. The vectors on each particle represent the velocity of the particle at that instant. Notice that as the surface is planarized the particles move at faster velocity since there are no longer obstacles to slow them down. In order to measure the "flatness" of the surface, Eq. 44, which was defined for use with the genetic algorithm, is used. Figure 6 shows how the "flatness" of the surface as the simulation time increases. Notice that at first the surface remains unchanged as the pad begins to move down towards the wafer surface. Once the pad is close enough the material is rapidly removed and Π seems to decrease close to linearly with time. As Π begins to approach zero the rate of change of Π with respect to time decreases quickly and Π levels off. After the surface becomes flat ($\Pi \to 0$) it is still possible for Π to increase if the surface is further indented, this corresponds to surface scratches is the polishing process. Although the particles move faster when the surface is flat, in general this does not occur because the vertical component of the particle velocity is small; therefore, the impact force when a particle contacts the surface is small.

4.1 Parameter optimization

With the polishing simulation described in the previous section it is possible to determine qualitative information as a function of the simulation parameters. However, one is sometimes interested in designing a process to achieve optimal results. When there are a large number of input parameters, trial and error experiments become very time consuming.

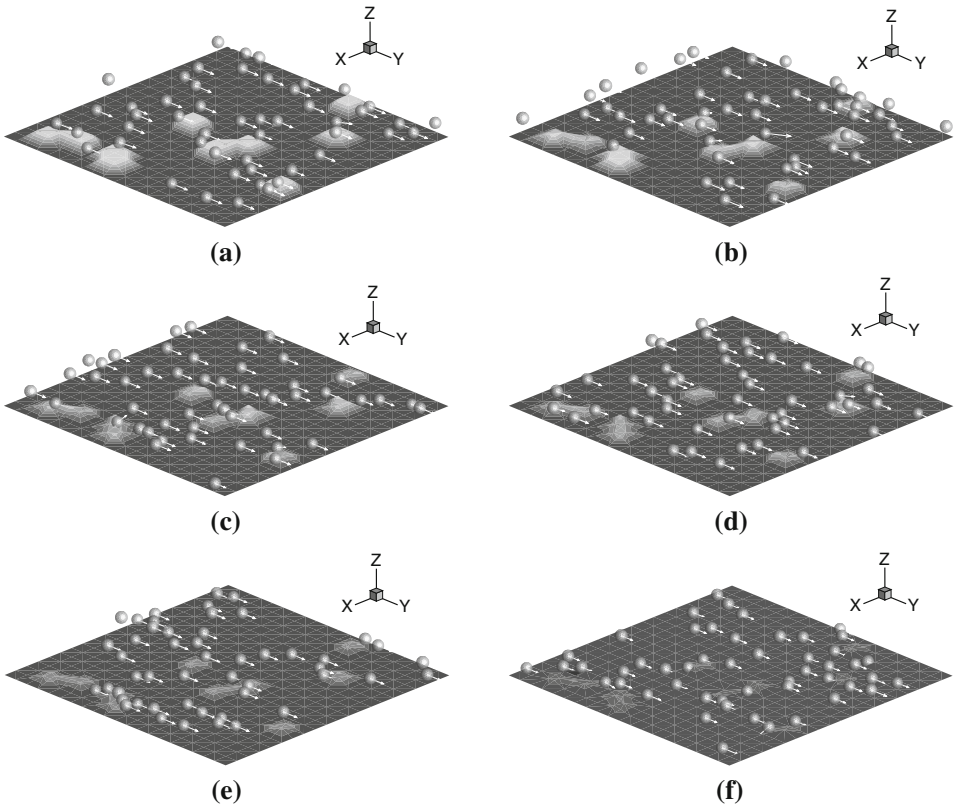

Fig. 5 Example of a representative simulation. The *grayscale* map represents the height of the surface

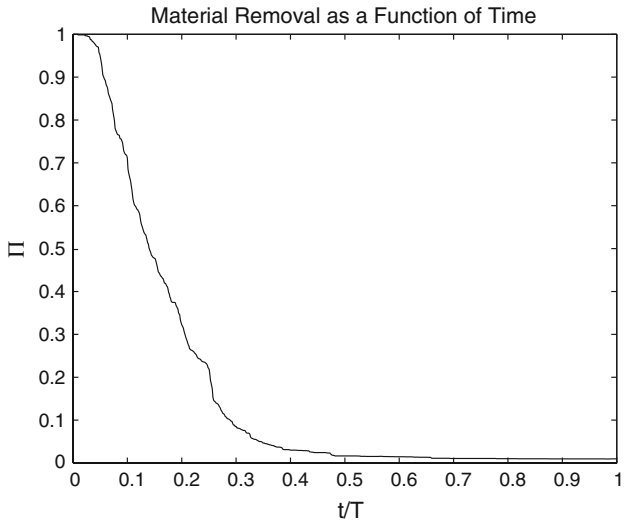

Fig. 6 The value of the cost function over time for a representative simulation

This issue can be alleviated by the use of an optimization method such as a genetic algorithm. For this example the input parameters are V_{pad}, N, $\bar{d}$, and δd, and the cost function is given by Eq. 44 where $q = 2$. Since the size of the particles is governed by a distribution of random variables as well as the initial conditions, an ensemble average must be used to determine the cost for each set of input parameters. It is usually not known a priori how many samples are needed to compute the ensemble average to an acceptable tolerance. Therefore, it is necessary to do these averages in an adaptive manner. This can be done by computing the average after the cost of each new sample is computed until the difference in the average for two consecutive samples is within the desired tolerance. However, a minimum number of samples must be set before this process is carried out since it is possible that consecutive samples have similar cost values that are not representative of the mean. Figure 7 shows the distribution of the cost values for simulations with the same parameters. For this particular example it takes approximately 15 samples to obtain a reasonable ensemble average. Notice that in plot (b) the distribution is biased to the right since the cost function can not be negative.

The cost function Π is minimized with respect to the parameters V_{pad}, N, $\bar{d}$, and δd. Figure 8 shows how Π for the top performing genetic string changes after each generation. In this example each generation consisted of 20 genetic strings. At each generation four parent strings were kept, four offspring were created, and 16 randomly chosen string were introduced. This process was carried out for ten generations. Table 1 shows the values for the top ten performing genetic strings. From these results it is concluded that the pad velocity, number of particles, and particle size play a larger role than the particle size distribution in the material removal process. This is evident since the top performing string all have similar values for V_{pad}, N,and $\bar{d}$, while δd varies over a wider range.

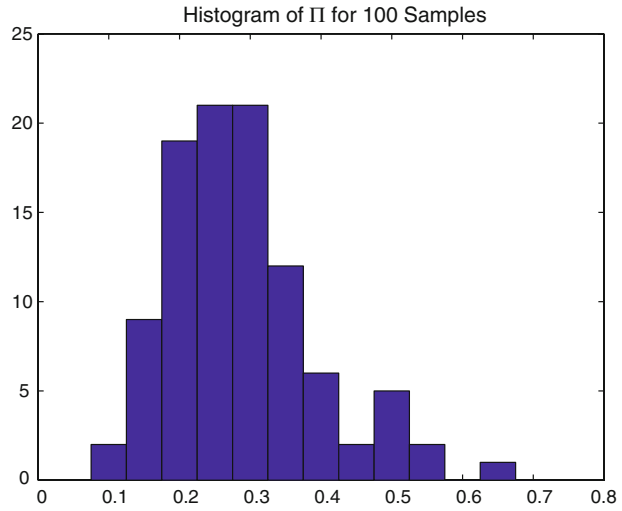

Fig. 7 Distribution of Π for samples with the same simulation parameters

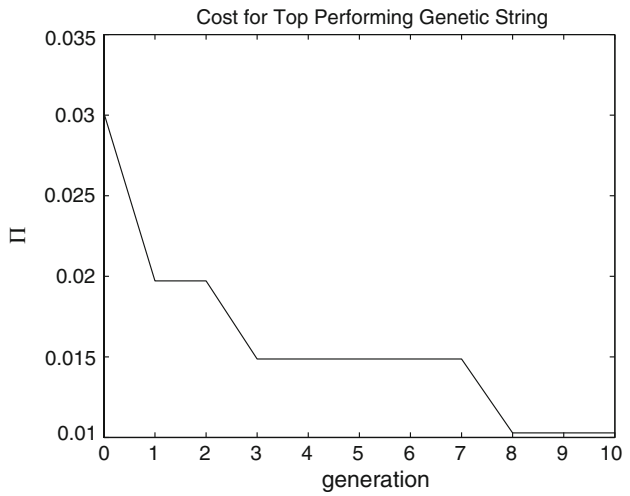

Fig. 8 Cost function value for the top performing genetic string in each generation

Table 1 Top Ten performing genetic strings

V_{pad}	N	$\bar{d}$	δd	Π
3.96	73	2.98×10^{-2}	0.16	1.03×10^{-2}
3.92	74	2.85×10^{-2}	0.18	1.49×10^{-2}
4.84	72	2.99×10^{-2}	4.99×10^{-2}	1.97×10^{-2}
4.40	73	2.99×10^{-2}	8.72×10^{-2}	1.98×10^{-2}
4.48	73	2.96×10^{-2}	9.36×10^{-2}	2.00×10^{-2}
4.45	73	2.98×10^{-2}	9.31×10^{-2}	2.02×10^{-2}
4.46	73	2.97×10^{-2}	8.82×10^{-2}	2.95×10^{-2}
4.46	73	2.93×10^{-2}	5.83×10^{-2}	3.14×10^{-2}
4.53	70	2.97×10^{-2}	1.32×10^{-4}	3.98×10^{-2}
4.50	72	2.33×10^{-2}	3.81×10^{-2}	9.00×10^{-2}

The pad velocity V_{pad} is in units of cm/s

5 Conclusions

In this work a method for simulating the material removal process was developed by employing techniques commonly used in the simulation of flowing granular materials. This new simulation technique along with a genetic algorithm was used to determine the input parameters that optimize a process in terms of a desired outcome which is specified in terms of a cost function. This method can be used to obtain useful results; however, its effectiveness relies on the ability to accurately simulate the underlying physics of the specific process. For example, the material removal process or the fluid flow. In order to improve this simulation, more accurate physical models must be developed.

In this simulation, the material removal was modeled using linear elasticity theory in order to predict the stresses in the material. The plastic zone could then be determined using the von-Mises yield criterion, and the amount of material removed was extracted based on this result. While this model is a useful estimate, a more accurate description may be obtained using computational methods (Wriggers [19]). While it would be computationally expensive to solve the contact problem each time a particle strikes the surface, it may be possible to determine a priori the amount of material removed in terms of a few parameters in the form of a computationally generated library.

References

1. Aquaro D, Fontani E (2001) Erosion of ductile and brittle materials. Meccanica 36:651–661
2. Aquaro D (2006) Erosion due to the impact of solid particles of materials resistant at high temperatures. Meccanica 41:539–551
3. Bitter JPA (1963) A study of erosion phenomena part I. Wear 6: 5–21
4. Bitter JPA (1963) A study of erosion phenomena part II. Wear 6:169–190
5. Chow CY (1980) An introduction to computational fluid dynamics. Wiley, London
6. Cook LM (1990) Chemical processes in glass polishing. J Non-Crystal Solids 120:152–171
7. Davis L (1991) Handbook of genetic algorithms. Thompson Computer Press
8. Finnie I (1960) Erosion of surfaces by solid particles. Wear 3:87–103
9. Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading
10. Goldsmith W (2001) Impact: the theory and physical behavior of colliding solids. Dover Re-issue, Toronto
11. Green DJ (1998) Introduction to the mechanical properties of ceramics. Cambridge University Press, Cambridge
12. Johnson K (1985) Contact mechanics. Cambridge University Press, Cambridge
13. Junkar M, Jurisevic B, Fajdiga M, Grah M (2006) Finite-element analysis of single particle impact in abrasive water jet machining. Int J Impact Eng 32:1095–1112

14. Lagaros N, Papadrakakis M, Kokossalakis G (2002) Structural optimization using evolutionary algorithms. Comput Struct 80:571–589
15. Li SW, Miller RO (2000) Chemical mechanical polishing in silicon processing. Semiconductors and semimetals. Academic Press, London, vol 63
16. Luo J, Dornfeld DA (2004) Integrated modeling of chemical mechanical planarization for sub-micron IC fabrication. Springer, Heidelberg
17. Preston F (1927) The theory and design of plate glass polishing machines. J Soc Glass Technol 11:214–256
18. Su YT (2000) Investigation of removal rate properties of a floating polishing process. J Electrochem Soc 147(6):2290–2296
19. Wriggers P (2002) Computational contact mechanics. Wiley, London
20. Zohdi TI (2004) Modeling and direct simulation of near-field granular flows. Int J Solids Struct 42/2:539–564
21. Zohdi TI (2004) A computational framework for agglomeration in thermo-chemically reacting granular flows. Proc R Soc 460(2052):3421–3445
22. Zohdi TI (2005) Charge-induced clustering in multifield granular flow. Int J Numer Methods Eng 62(7):870–898
23. Zohdi TI (2007) Computation of strongly coupled multifield interaction in particle–fluid systems. Comput Methods Appl Mech Eng 196:3927–3950