Parameter Identification of a Friction Model in Metal Cutting Simulations with GPU-Accelerated Meshfree Methods

Author(s):
Afrasiabi, Mohamadreza; Klippel, Hagen; Röthlin, Matthias; Wegener, Konrad

Publication date:
2021

Permanent link:
https://doi.org/10.3929/ethz-b-000522444

Originally published in:
https://doi.org/10.23967/wccm-eccomas.2020.273

Funding acknowledgement:
149436 - GPU-Enhanced Metal Cutting Simulation using Advanced Meshfree Methods (SNF)

14th World Congress on Computational Mechanics (WCCM)
ECCOMAS Congress 2020
19–24 July 2020, Paris, France
F. Chinesta, R. Abgrall, O. Allix and M. Kaliske (Eds)

PARAMETER IDENTIFICATION OF A FRICTION MODEL IN METAL CUTTING SIMULATIONS WITH GPU-ACCELERATED MESHFREE METHODS

M. Afrasiabi[1,2], H. Klippel[1], M. Roethlin[3], and K. Wegener[1]

[1] Institute of Machine Tools & Manufacturing, ETH Zürich, 8092 ZH, Switzerland
https://www.iwf.mavt.ethz.ch/people/staff/index
[2] Chair of Structural Mechanics & Monitoring, ETH Zürich, 8093 ZH, Switzerland
http://www.chatzi.ibk.ethz.ch/people.html
[3] Federal Office of Meteorology & Climatology, MeteoSwiss, 8058 ZH-Airport, Switzerland
E-mail: afrasiabi@ethz.ch

Key words: Metal cutting; Friction; Parameter identification; GPU computing; SPH.

Abstract. A modular computational framework is presented for the identification of friction parameters in metal machining applications. Numerical simulation of such processes using mesh-based techniques (usually) necessitates the cumbersome re-meshing procedures and is hard to parallelize. Therefore, the present framework synthesizes the advantages of mesh-free methods with GPU parallel computing, offering an efficient tool for the optimization procedures in thermo-mechanical modeling of cutting problems. The proposed approach employs an inverse method to determine the unknown coefficients of a temperature-dependent friction model in high-speed metal cutting. Good agreement between our numerical results and experimental data is found, providing both quantitative and qualitative assessments.

1 INTRODUCTION

Today's manufacturing techniques demand precise and efficient modeling of the metal cutting process. One of the most challenging tasks in the development of numerical cutting models is the formulation of contact loads, where friction remains the most dominant phenomenon. In metal cutting operations, the simplest approach for modeling of friction is to consider Coulomb's law as

$$|\underline{f}^{\text{fric}}| = \mu |\underline{f}^{\text{cont}}| \tag{1}$$

with a constant coefficient μ over the whole chip-tool interface (see, the secondary shear zone of Fig. 1 shown in orange). Despite being a good approximation in many problems, the friction model of form (1) with a constant μ at severe contact conditions seems far from realistic [1]. The presence of such conditions in metal cutting is inescapable.

Friction models more sophisticated than the linear Coulomb law (1) exist in the literature, e.g., [3]. Another extension of Coulomb's law is a temperature-dependent model by Moufki et al. [4]. This is a particularly interesting one since the friction conditions at the tool-chip interface are greatly influenced by temperature. The present work considers a temperature-dependent coefficient $\mu(T)$ of friction similar to the thermal softening of the material constitutive model and follows the same approach offered by [4].

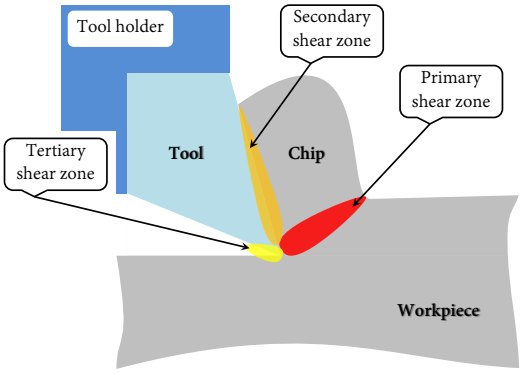


Figure 1: Sketch of a cutting geometry and the main shear zones.

That is,

$$|\underline{f}^{\text{fric}}| = \underbrace{\mu_0 \left[1 - \left(\frac{T - T_r}{T_m - T_r}\right)^q\right]}_{\mu(T)} |\underline{f}^{\text{cont}}| \tag{2}$$

where T is the contact temperature, T_r the reference or room temperature, T_m the melting point, and μ_0 and q are the unknown constants. Although the form expressed in (2) is clearly an enhancement to (1), it should be mentioned that the complexity and diversity of contact conditions in cutting processes are such that a single friction model cannot be applied to all material and cutting conditions. Historical development of friction models in metal machining is reviewed by Childs [5], where he also investigates the nature of the friction contact between chip and tool.

As mentioned, the coefficients μ_0 and q in (2) are unknown and need to be determined for the desired system. Previous studies have shown that inverse fitting methods combined with finite element analysis (FEA) can be utilized for this purpose. The efforts by [6] and [7] are relevant to this type of methodology. On the other hand, however, mesh-free particle methods feature two characteristics which are advantageous for parameter identification:

- These methods are highly efficient for modeling of metal cutting since they are intrinsically capable of handling large deformations.
- Compared with FEA, particle-based simulations are easier to parallelize.

Therefore, a numerical framework based on the smoothed particle hydrodynamics (SPH) method [8, 9] is developed in this work to identify a new friction model in meshfree cutting simulations. SPH has been widely used in many other fields of applications, e.g., [10], [11] and [12].

A review of the technical literature reveals that SPH is rapidly gaining recognition as an adept tool in the numerical modeling of manufacturing processes. Some recent publications include the use of SPH in high-speed cutting simulations [13], the thermal simulation of a simplified laser drilling problem [14], and a multi-resolution SPH framework for orthogonal chipping by [15]. More interestingly, the concept of parallel programming on the Graphics Processing Unit (GPU) has been introduced to the current SPH cutting models. For example, a GPU-accelerated code developed by [16] and 3D single grain cutting simulations by [16]. One area where GPU-accelerated codes can be of particular interest and efficiency

is the parameter identification problems with inverse fitting methods. Except for a very recent effort by Afrasiabi et al. [2], the adaptation of SPH cutting models to this field of application seems unexplored.

In this work, SPH is adapted to accurately resolve thermal and mechanical fields in a metal cutting process which involves the large deformation, separation, heating, and chip formation of Ti6Al4V work material. The present framework synthesizes previously established SPH schemes from disparate fields for spatial discretizations while supplementing them with a GPU implementation for parallel computing. The speedup gained by this GPU parallelization enables very short evaluation cycles required for iterative simulations and inverse fitting. As a result, unknown parameters of a new friction model expressed in (2) are determined and an enhancement to the current SPH cutting models is proposed. Validation of the methodology is undertaken by presenting a comparison of the predicted forces to a cutting experiment.

2 GOVERNING EQUATIONS

In the previous section, the geometry of a metal cutting process was illustrated in Fig. 1. To describe the motion of a continuum in an updated Lagrangian frame, the field equations need to be evolved simultaneously. The mechanical and thermal equations to be solved and coupled are outlined.

- **Mechanical**

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \underline{v} \tag{3}$$

$$\frac{d\underline{v}}{dt} = \frac{1}{\rho}\nabla \cdot \underline{\underline{\sigma}} + \frac{1}{m}\underline{b} \tag{4}$$

$$\frac{d\underline{r}}{dt} = \underline{v} \tag{5}$$

where ρ is the density, m the mass, $\underline{v}$ the velocity, $\underline{\underline{\sigma}}$ the Cauchy stress tensor, $\underline{b}$ the body forces, and $\underline{r}$ the position vector.

- **Thermal**

$$\frac{dT}{dt} = \frac{k}{\rho c_p}\nabla^2 T + \frac{Q}{\rho c_p} \tag{6}$$

in which k is the heat conductivity assuming an isotropic heat conduction, c_p the specific isobaric heat capacity, and the heat source Q for the metal cutting application is considered as

$$Q = \underbrace{\chi\left(\sigma_y \dot{\varepsilon}^{pl}\right)}_{\text{plastic heat}} + \underbrace{\eta\left(\frac{\rho\,|\underline{f}^{\text{fric}}| \cdot |\underline{v}^{\text{rel}}|}{m}\right)}_{\text{frictional heat}} \tag{7}$$

where χ and η are dimensionless parameters between 0 and 1, specifying how much of the plastic and frictional work is converted to heat. In (7), the term σ_y represents the yield stress and $\dot{\varepsilon}^{pl}$ indicates the plastic strain rate, while the formulation of $|\underline{f}^{\text{fric}}|$ was given in (2). The boundary conditions of (6) are imposed by considering a constant room temperature on fixed surfaces and perfect isolation (i.e., adiabatic heating) elsewhere.

In order to close the thermal-mechanical equations listed from (3) to (6), the strength values of the deforming material (i.e., Ti6Al4V here) still need to be expressed and evolved. By definition, the instantaneous value of strength at which a metallic material starts to plastically deform, i.e., flowing of the metal, is referred to as *flow stress*. Generally speaking, the flow stress σ_y is mainly dependent upon strain, strain rate, and temperature. A common choice is the Johnson-Cook law [27] that accounts for these three effects and is chosen to calculate the flow stress of Ti6Al4V in this work. It reads

$$\sigma_y^{\mathrm{JC}} = \left[A + B\,(\bar{\varepsilon}^{pl})^n\right]\left[1 + C\,\ln\left(\frac{\dot{\bar{\varepsilon}}^{pl}}{\dot{\bar{\varepsilon}}_0^{pl}}\right)\right]\left[1 - \left(\frac{T - T_r}{T_m - T_r}\right)^m\right] \tag{8}$$

where $\bar{\varepsilon}^{pl}$ is the equivalent plastic strain, and the material parameters $A = 862$ MPa, $B = 331$ MPa, $C = 0.01$, $m = 0.8$, and $n = 0.35$ are used for the simulations of this paper. These values have been used by numerous researchers in the field for a similar application, e.g., [2, 16, 27].

3 COMPUTATIONAL FRAMEWORK

To enable inverse identification of friction parameters, a batch file of the unknown constants in (2) is first created. This file contains different values of the $\mu_0 - q$ sets and is given as input to the software. When the simulation is complete, an optimization algorithm is performed at the post-processing stage to determine the optimum combination of $\mu_0 - q$ for minimum error. Since this procedure needs to be carried out iteratively, it is crucial to minimize the runtime of the solver as much as possible. For the present cutting problems, we propose a robust computational framework that can tackle this issue on two fronts. Firstly, SPH is used for spatial discretizations as a truly mesh-free method with great potential for parallelization. Secondly, the computation cycles are accelerated by implementing the code to run entirely on a GPU. The following sub-sections discuss these two remarks considerations in more detail.

3.1 SPH Formalism & Discrete Equations

SPH is an interpolation method. It defines the physical field at discrete points called particles. These Lagrangian particles move with the material velocity and carry the system variables such as mass, density, pressure, stress tensor, and temperature. Using SPH, a field function f at point/particle i is calculated by a weighted interpolation of all points/particles j inside the support domain of i as

$$\langle f_i \rangle \approx \sum_j f_j W_{ij} V_j \tag{9}$$

with $V_j = m_j/\rho_j$ and $W_{ij} = W_h(\underline{r}_i - \underline{r}_j, h)$ being a smooth kernel function, in which h denotes the radius of its support domain also known as the smoothing length of W. A list of common choices for W can be found in [18], from which the cubic B-spline is chosen for this paper. To solve the partial differential equations (PDEs) given in (3)–(6), the first and second derivatives of the physical fields also need to be discretized. According to [20, 21], the subtractive and additive forms of SPH are employed to approximate the gradient of f as

$$\langle \nabla f_i \rangle \approx \sum_j (f_j - f_i)\,\nabla W_{ij} V_j \tag{10}$$

$$\langle \nabla f_i \rangle \approx \rho_i \sum_j \left(\frac{f_i}{\rho_i^2} + \frac{f_j}{\rho_j^2}\right) \nabla W_{ij} m_j \tag{11}$$

As for the Laplace operator, an adaptation of the Brookshaw scheme [22] is considered. It benefits from a simple correction to the original SPH form and is expressed by

$$\langle \nabla^2 f_i \rangle \approx \sum_j 2 \left[\left(\frac{f_i - f_j}{|r_{ij}|} \right) \underline{e}_{ij} \cdot \nabla W_{ij} \right] V_j \tag{12}$$

where $\underline{e}_{ij} = \underline{r}_{ij}/|r_{ij}|$ is a unit vector in the inter-particle direction. A broad array of other meshfree schemes for discretizing the Laplace operator can be found in [23]. Using (12) guarantees the conservation of energy to the precision of the time-stepping algorithm, which is a key advantage of this scheme. Now, the governing PDEs can be discretized in space using the meshfree schemes presented from (10) to (12). The SPH approximations of (3), (4), and (6) are summarized in the following.

$$\left\langle \frac{d\rho_i}{dt} \right\rangle \approx -\rho_i \sum_j (\underline{v}_j - \underline{v}_i) \cdot \nabla W_{ij} V_j \tag{13}$$

$$\left\langle \frac{d\underline{v}_i}{dt} \right\rangle \approx \sum_j \left(\frac{\underline{\underline{\sigma}}_i}{\rho_i^2} + \frac{\underline{\underline{\sigma}}_j}{\rho_j^2} + \underbrace{\Pi_{ij}\underline{\underline{I}} + \underline{\underline{\Lambda}}_{ij}}_{\text{stabilizers}} \right) \cdot \nabla W_{ij} m_j + \frac{\underline{b}_i}{m_i} \tag{14}$$

$$\left\langle \frac{dT_i}{dt} \right\rangle \approx \frac{k}{\rho c_p} \sum_j 2 \left[\left(\frac{T_i - T_j}{|r_{ij}|} \right) \underline{e}_{ij} \cdot \nabla W_{ij} \right] V_j + \frac{Q_i}{\rho c_p} \tag{15}$$

where Π and $\underline{\underline{\Lambda}}$ are the artificial viscosity term [24] and the artificial stress tensor [25] with the parameters taken similarly to [16, 2]. After solving these equations for each particle, it is left to evolve the variables and update the physical system. To do so, we move the particles with a smoothed velocity instead of the actual material velocity. That is, a modification term is added to the material velocity leading to

$$\left\langle \frac{d\underline{r}_i}{dt} \right\rangle \approx \underline{v}_i + \underbrace{\beta \sum_j \frac{m_j}{\rho_i + \rho_j} (\underline{v}_j - \underline{v}_i) W_{ij}}_{\text{X-SPH modification}} \tag{16}$$

where β is a tuning parameter taken as 0.3 according to [16, 2]. This smoothing scheme was suggested by Monaghan [26] and is referred to as the X-SPH correction. In the last step, the boundary conditions are imposed and the ordinary differential equations (ODEs) from (13) to (16) are discretized in time using an explicit second-order leapfrog stepping. The time step Δt follows the Courant–Friedrichs–Lewy criterion. Regarding the implementation procedure of this time integration scheme for SPH models, please consider [26] and [19].

3.2 Parallel Computing & GPU Implementation

High-performance and parallel computing techniques are rapidly gaining interest in academia and industry because they offer the possibility to solve large problems in science and engineering. As pointed out at the beginning of this section, the second consideration for boosting the computational efficiency in this work is the speedup gained by GPU programming. Nevertheless, the literature review in Section 1 implies that GPU-accelerated SPH codes in manufacturing simulations are currently in their nuclei stage. The inverse identification problem in this work is ideally suited to GPU-accelerated SPH codes

due to the abundance of computational cores on a single GPU. In what follows, we briefly describe some important remarks about the present GPU implementation.

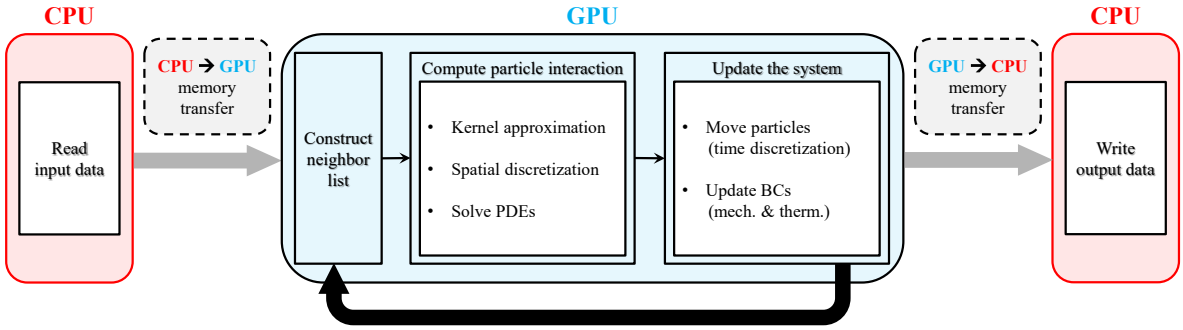


Figure 2: Flow diagram of the SPH code executed at each time step on a GPU.

Principally speaking, the GPU implementation in this work follows the procedure elaborated upon by [29]. For this task, we utilize the CUDA platform created by NVIDIA® in 2007 [30]. To illustrate the flow diagram of the SPH code implemented on a GPU, the main blocks executed at each time step are shown in Fig. 2. As can be seen in this diagram, the computer program in this approach is run on the GPU completely, using the CPU for I/O only. Parts of the program may, therefore, be slower than the CPU but at the benefit of not having to copy data. Meanwhile, the organization of neighbor particles – regarded as one of the most computationally expensive tasks in SPH codes – is handled by implementing a spatial hashing algorithm with the cell-list data structure.

Table 1: Runtime in seconds taken from [2], recorded for a 2D cutting test using the present SPH code. The serial CPU calculations are taken on Intel® Core™ i5-4690 and the GPU code runs on Tesla® P100. The floating-point format in both calculations is double-precision.

N_p	**CPU [sec]**	**GPU [sec]**	**Speedup**
1,550	52.8	41.4	1.3
6,100	534.9	66.0	8.1
24,200	5639.7	141.8	39.8
96,400	49359.5	454.4	108.6

Discussing the runtime of the solver is not within the objectives of this article. Nonetheless, it is still beneficial to throw some light on the speedup provided by GPU computing. The timing data in Table 1 are taken from [2], related to an orthogonal cutting simulation with SPH. This test case considers a 0.1 mm cut of a Ti6Al4V workpiece with a speed of 500 m/min. The CPU and GPU execution times of this model are measured, where the workload in both cases is the same. One realizes from the runtime reports

associated with $N_p = 96,400$ that the GPU code runs almost 100x faster than the serial CPU program. Note that the symbol N_p indicates the total number of SPH particles, which is also equal to the number of GPU threads here. Before closing this section, the GPU speedups are plotted versus the number of threads in Fig. 3 to gain more insights.

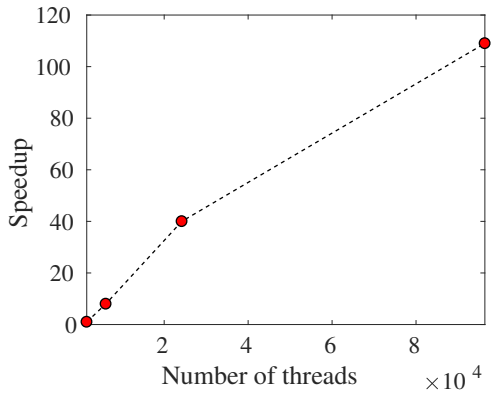


Figure 3: Plot of speedup for a 2D metal cutting test using the present GPU-accelerated code.

4 RESULTS

In this section, a cutting experiment is simulated with SPH particles to determine the unknown coefficients of (2) through an optimization process. This metal cutting experiment is conducted at the speed of $v_c = 318.5$ m/min on a workpiece made of Ti6Al4V alloy. A full list of parameters used in the experiment and simulation is provided in Table 2. The 5 Johnson-Cook constants are excluded from this table since they were already given in Section 2. The height of the workpiece (i.e., $l_y = 0.3$ mm) is discretized by 41 particles with regular spacing. As for the software, an in-house code `iwf-mfree-gpu-2d` originally published and made public by [16] is utilized. This open-source SPH solver can be downloaded from `https://github.com/mroethli/mfree_iwf-ul_cut_gpu`. The runtime data provided in Table 1 were also obtained using this software. Regarding the hardware, a Tesla® P100 by Nvidia Corporation hosts all GPU computations.

Table 2: Experimental and numerical parameters for the cutting test.

Body	Property	Symbol	Unit	Value
Tool	Clearance angle	α	deg	7.0
	Rake angle	γ	deg	0.0
	Cutting edge radius	r_c	mm	0.0028
	Speed	v_c	m min^{-1}	318.5
	Density	ρ	kg m^{-3}	15,250
	Heat conductivity	k	W m^{-1} K^{-1}	88
	Specific heat capacity	c_p	J kg^{-1} K^{-1}	292
Workpiece	Geometry	$l_x \times l_y$	mm^2	2.0×0.3
	Depth of cut	d	mm	0.1
	Length of cut	l_c	mm	1.0
	Density	ρ	kg m^{-3}	4,430
	Young's modulus	E	GPa	110.0
	Poisson ratio	ν	–	0.35
	Heat conductivity	k	W m^{-1} K^{-1}	7.3
	Specific heat capacity	c_p	J kg^{-1} K^{-1}	553.0
	Reference temperature	T_r	K	300
	Melting temperature	T_m	K	1,878
	Coeff. of plastic work into heat	χ	–	0.9
	Coeff. of frictional work into heat	η	–	1.0

A batch file including 25 variations of μ_0 and q is first created, where the initial values of these two unknowns are taken from $\mu_0 = \{0.1, 0.43, 0.77, 1.1, 1.43\}$ and $q = \{0.5, 2.0, 3.5, 5.0, 6.5\}$. Now the SPH cutting model is simulated 25 times using 25 different initial values for μ_0 and q. In the next step, we carry out a simple optimization process on predicted forces by calculating the relative error

$$\delta\diamondsuit = \frac{|\diamondsuit^{\text{exp}} - \diamondsuit^{\text{sim}}|}{\diamondsuit^{\text{exp}}} \tag{17}$$

for both cutting (i.e., $\diamondsuit = F_c$) and thrust (i.e., $\diamondsuit = F_t$) forces. The experimentally measured data for this cutting test read $F_c = 157.5$ N/mm and $F_t = 104.2$ N/mm at the stationary zone. To minimize the errors of F_c and F_t simultaneously, we propose to consider the term $\delta\hat{F} = \sqrt{(\delta F_c)^2 + (\delta F_t)^2}$ as the optimization measure. The batch simulation results provide 25 sample points to reconstruct the $\delta\hat{F}$ error function. These values are then interpolated into the entire domain of parameters to create the original sampling of the function. We chose a cubic spline interpolation in `MATLAB` using 50 additional query points in each dimension. The choice of a cubic spline was made since a linear interpolation cannot return a global minimum at off-sample points. This is an important issue as the μ_0–q set resulting in the minimum error might (and usually does) occur where no simulation data is available. Fig. 4 sums up the results of this investigation by plotting the $\delta\hat{F}$ error surface. The red point in this figure highlights the coordinates $\mu_0 = 0.81$ and $q = 5.03$ at which the minimum error occurs, determining the unknown constants of $\mu(T)$ in (2). Consequently, the friction coefficient is enhanced with a thermal dependency and obtained as

$$\mu(T) = 0.81 \left[1 - \left(\frac{T - T_r}{T_m - T_r}\right)^{5.03}\right] \tag{18}$$

This expression is plugged in (2) for calculating the friction force and yields $\delta F_c = 14\%$ and $\delta F_t = 1.5\%$ for the optimized simulation. Forces computed by SPH reach the steady-state after almost 30% of the simulation and compare very well with the experimental measurements. These conclusions are drawn from the force evolution and bar-chart diagrams plotted in Fig. 5.

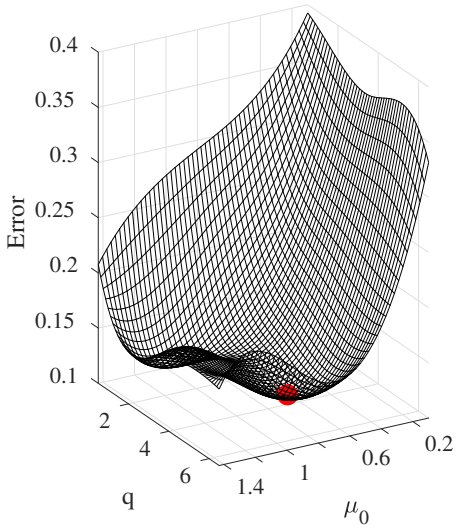


Figure 4: Plot of the $\delta\hat{F}$ error surface using 25 SPH simulations and a cubic spline interpolation. The minimum error is found by choosing $\mu_0 = 0.81$ and $q = 5.03$ at the red point coordinates.

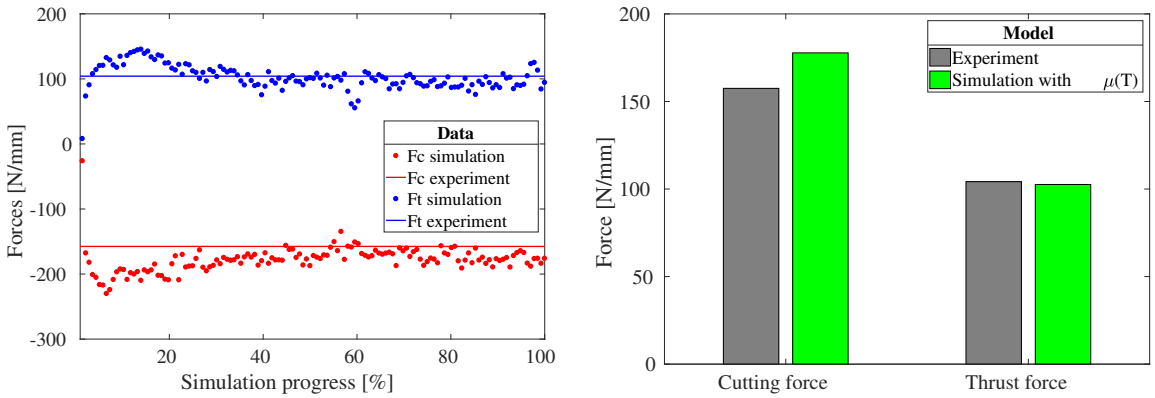
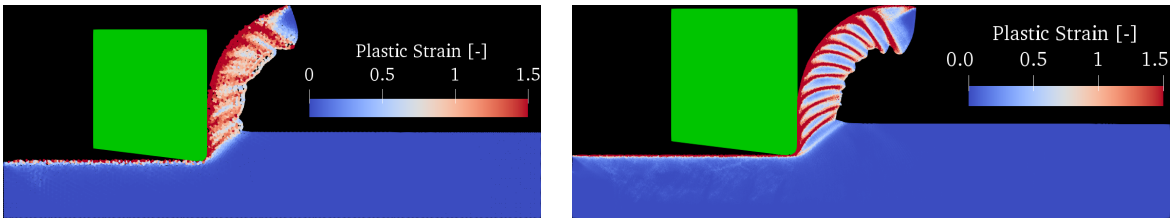


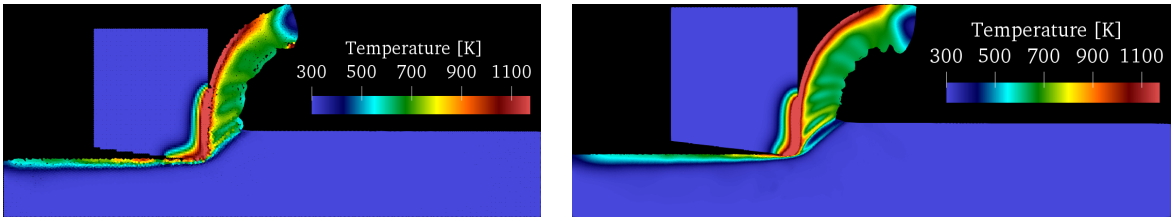
Figure 5: Measured and simulated forces in orthogonal cutting of Ti6Al4V with $v_c = 318.5$ m/min and $r_c = 28\,\mu$m.

In the last step, the findings of inverse parameter identification are inserted into a high-resolution SPH model using a total of approximately 152k particles. The plastic strain and temperature distributions are displayed in Figs. 6 and 7. These pictures permit several interesting observations. Firstly, it is understood that discretization has a significant impact on the SPH cutting models. Comparing the left and right frames in Fig. 6 reveals that a high-resolution simulation is indeed necessary to capture the shear bands located at the chip. Secondly, the temperature distribution in Fig. 7 demonstrates that the



11k particles 152k particles

Figure 6: Distribution of plastic strain in the present SPH cutting simulations on the GPU.



11k particles 152k particles

Figure 7: Distribution of temperature in the present SPH cutting simulations on the GPU.

frictional contact zone between the tool and workpiece is the hottest area during the machining processes. This shows why a temperature-dependent friction model is crucial to the thermomechanical analysis of this problem. Thirdly, it can be seen that as time passes the generated heat diffuses from the hot shear bands into the cold nearby regions in the chip.

5 CONCLUSIONS

An improvement to the SPH cutting simulations has been presented. The main contributions of this work can be summarized:

- Modeling of friction in these numerical frameworks is enhanced by proposing a temperature-dependent coefficient with two unknown constants.
- These two parameters are determined through an optimization procedure conducted on predicted fores from 25 SPH simulations.
- To make the runtime manageable, we accelerate the computations by implementing the code on a GPU. This allows one to solve the thermo-mechanical equations more efficiently.

It was shown in a preliminary study that a 100x speedup can be achieved by the present GPU implementation. The proposed simulation framework is thus suggested to be utilized for parameter identification purposes in metal cutting. Incorporating the newly identified friction model into the current SPH solver, one obtains accurate force prediction results compared to the experimental data.

ACKNOWLEDGMENTS

This work was partially funded by the Swiss National Science Foundation under Grant No. 200021-149436. MA would also like to especially thank Prof. Dr. Eleni Chatzi for her support and many helpful suggestions during this research.

REFERENCES

[1] Özel, Tugrul. The influence of friction models on finite element simulations of machining. *International journal of machine tools and manufacture* 46.5 (2006): 518-530.

[2] Afrasiabi, M., et al. GPU-Accelerated Meshfree Simulations for Parameter Identification of a Friction Model in Metal Machining. *International Journal of Mechanical Sciences* (2020): 105571.

[3] Usui, E., and T. Shirakashi. Mechanics of machining—from descriptive to predictive theory. *on the art of cutting metals-75 years later* 7.1 (1982): 13-55.

[4] Moufki, A., Alain Molinari, and D. Dudzinski. Modelling of orthogonal cutting with a temperature dependent friction law. *Journal of the Mechanics and Physics of Solids* 46.10 (1998): 2103-2138.

[5] Childs, T. H. C. Friction modelling in metal cutting. *Wear* 260.3 (2006): 310-318.

[6] Brocail, J., M. Watremez, and L. Dubar. Identification of a friction model for modelling of orthogonal cutting. *International Journal of Machine Tools and Manufacture* 50.9 (2010): 807-814.

[7] Puls, H., F. Klocke, and D. Lung. Experimental investigation on friction under metal cutting conditions. *Wear* 310.1-2 (2014): 63-71.

[8] Gingold, Robert A., and Joseph J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society* 181.3 (1977): 375-389.

[9] Lucy, Leon B. A numerical approach to the testing of the fission hypothesis. *The astronomical journal* 82 (1977): 1013-1024.

[10] Afrasiabi, M., and S. Mohammadi. Analysis of bubble pulsations of underwater explosions by the smoothed particle hydrodynamics method. *ECCOMAS international conference on particle based methods*, Spain. 2009.

[11] Price, Daniel J. Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics* 231.3 (2012): 759-794.

[12] Afrasiabi, M., M. Roethlin, and K. Wegener. Thermal simulation in multiphase incompressible flows using coupled meshfree and particle level set methods. *Computer Methods in Applied Mechanics and Engineering* 336 (2018): 667-694.

[13] Limido, Jerome, et al. SPH method applied to high speed cutting modelling. *International journal of mechanical sciences* 49.7 (2007): 898-908.

[14] Afrasiabi, Mohamadreza, Eleni Chatzi, and Konrad Wegener. A particle strength exchange method for metal removal in laser drilling. *Procedia CIRP* 72 (2018): 1548-1553.

[15] Afrasiabi, Mohamadreza, et al. Meshfree simulation of metal cutting: an updated Lagrangian approach with dynamic refinement. *International Journal of Mechanical Sciences* 160 (2019): 451-466.

[16] Röthlin, M., et al. Metal cutting simulations using smoothed particle hydrodynamics on the GPU. *The International Journal of Advanced Manufacturing Technology* 102.9-12 (2019): 3445-3457.

[17] Röthlin, Matthias, et al. Meshless single grain cutting simulations on the GPU. *International Journal of Mechatronics and Manufacturing Systems* 12.3-4 (2019): 272-297.

[18] Liu, Gui-Rong, and Moubin B. Liu. Smoothed particle hydrodynamics: a meshfree particle method. *World scientific* 2003.

[19] Monaghan, Joe J. Smoothed particle hydrodynamics. *Reports on progress in physics* 68.8 (2005): 1703.

[20] Monaghan, Joe J. Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics* 30.1 (1992): 543-574.

[21] Liu, M. B., and G. R. Liu. Smoothed particle hydrodynamics (SPH): an overview and recent developments. *Archives of computational methods in engineering* 17.1 (2010): 25-76.

[22] Brookshaw, Leigh. A method of calculating radiative heat diffusion in particle simulations. *Publications of the Astronomical Society of Australia* 6.2 (1985): 207-210.

[23] Afrasiabi, Mohamadreza, Matthias Roethlin, and Konrad Wegener. Contemporary Meshfree Methods for Three Dimensional Heat Conduction Problems. *Archives of Computational Methods in Engineering* (2019): 1-35.

[24] Monaghan, Joseph J., and Robert A. Gingold. Shock simulation by the particle method SPH. *Journal of computational physics* 52.2 (1983): 374-389.

[25] Gray, James P., Joseph J. Monaghan, and R. P. Swift. SPH elastic dynamics. *Computer methods in applied mechanics and engineering* 190.49-50 (2001): 6641-6662.

[26] Monaghan, J. J. On the problem of penetration in particle methods. *Journal of Computational physics* 82.1 (1989): 1-15.

[27] Johnson, Gordon R. A constitutive model and data for materials subjected to large strains, high strain rates, and high temperatures. *Proc. 7th Inf. Sympo. Ballistics* (1983): 541-547.

[28] Sima, Mohammad, and Tugrul Ozel. Modified material constitutive models for serrated chip formation simulations and experimental validation in machining of titanium alloy Ti–6Al–4V. *International Journal of Machine Tools and Manufacture* 50.11 (2010): 943-960.

[29] Crespo, Alejandro JC, et al. DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH). *Computer Physics Communications* 187 (2015): 204-216.

[30] http://docs.nvidia.com/cuda/index. Nvidia toolkit documentation v8.0. 2017.