Advanced Materials Research *Online: 2008-04-08*
ISSN: 1662-8985, Vol. 43, pp 89-96
doi:10.4028/www.scientific.net/AMR.43.89
© 2008 The Author(s). Published by Trans Tech Publications Ltd, Switzerland.

Dynamics and Temperature Simulation in Multi-Axis Milling

Eduard Ungemach[1, a], Tobias Surmann[1, b], Andreas Zabel[1, c]

[1] Department of Machining Technology, University of Dortmund, Germany

[a]ungemach@isf.de, [b]surmann@isf.de, [c]zabel@isf.de

Keywords: Milling simulation, multi-axis milling, lightweight structures, temperature simulation, dynamic simulation, surface quality

Abstract.
Lightweight extrusion profiles with reinforcement elements are promising news in the domain of lightweight construction. The machining of them suffers from several problems: Aside from the question of choosing a suitable tool, feed rate, and milling strategy, an excessive rise in temperature could lead to stress and even a distortion due to the differing thermal expansion of the reinforcement material and the surrounding matrix material. A simulation of the milling process could, in addition to force and collision calculations, recognize this case before manufacturing.
For certain milling applications like seal surfaces, a certain roughness of the manufactured surface is necessary. In many other cases, a smooth surface of very high quality is desirable. Available simulation systems usually completely lack the simulation of dynamic effects, which have a great effect on the final surface quality, and therefore are not able to predict the resulting surface quality.
In this paper simulation methods are presented that are capable of simulating the dynamic behavior of the tool in the milling process and the resulting flank and ground surface structures.
Additionally, a fast temperature simulation for heterogeneous workpieces with reinforcement elements, which is based on the finite difference method and cellular automata, is introduced.

Introduction

Rising energy cost and the growing awareness of environment protection lead to a demand for weight reduction in fields like vehicle production and aerospace. At the same time, additional safety and comfort elements increase the weight. Lightweight construction elements are an important countermeasure while offering additional flexibility. This paper shows advantages in two different areas of simulation-based machining.
To provide properties that are comparable to conventionally built products, the use of reinforcement elements, like steel wires, is a matter of current research e.g. in the “Sonderforschungsbereich Transregio 10” which mainly deals with aluminum extrusion elements with reinforcements. After the production of the reinforced extrusion elements, it is usually necessary to machine them e. g. to add holes for rivets. Previous publications already describe the use of simulation software to ensure that occurring forces are below a certain limit [1, 2] and to guarantee that no collisions occur in the real production [3, 4, 5]. Reinforced extrusion elements are prone to distortion and stress from heat, which can be the result of too much energy added by the machining process. The reason for this sensitivity is that reinforced extrusion elements are built from different materials having differing thermal expansion coefficients. In [3] a milling simulation software with a temperature simulation that is suitable only for homogeneous materials is described. In this paper, methods are shown that are fast enough to be used in a realtime simulation and that are capable of simulating heterogeneous materials. The finite element analysis (FEA) method, which is a common method for the calculation of temperature distributions, offers a high accuracy, but is usually too slow to evaluate complete NC programs. The explicit finite difference method approximates a differential by a difference quotient and is therefore faster while introducing a discretization error.

The second area dealt with in this paper is the field of dynamics in milling. For certain applications a defined surface quality is necessary. A certain surface roughness for joint elements to generate a reliable form fit connection and for seal surfaces so that the adhesive sticks optimally. In other cases the surface should be as smooth as possible. Even in the ideal milling process, there is a relevant surface structure which is generated by the interrupted cut. Additionally, the resulting roughness is strongly influenced by dynamic effects like vibrations or chatter. Chatter can also be the reason for excessive wear of the tool. The conventional procedure to optimize the surface properties is a loop that consists of changing process parameters and remanufacturing until the results are satisfying. Stability charts are used only rarely because their compilation is very time consuming and they depend on both the used tool and the used machine.

In this paper a simulation system is introduced that is able to predict the dynamic trajectory of the milling tool and thus is able to reproduce the generated surface structure of the flank and ground. It simulates the self-excited chatter of the tool on the based on a damped harmonic oscillator. The input parameters are conventional NC files and several geometric and modal parameters describing the tool. For the calculation of the oscillator behavior, knowledge of the forces on the tool is necessary. The simulation is a time domain milling simulation working in discrete time steps. The utilized force model is based on the Kienzle equation. For each small time step, the generated chip form is calculated. Therewith, the forces on the tool can be deduced, which have influence on the tool deflection in the next iteration. With this simulation software it is possible to automatically create stability charts for variable radial immersions and cutting depths. The geometric structure of the flank and ground can be constructed and visualized.

Simulation of the Dynamic Milling Process

In the past several types of simulation of the milling process have been developed. Depending on their focus, different concepts have been used. Generally, the milling process is simulated by discretizing the movement of the tool and removing its occupied volume. An obvious choice for the simulation of the workpiece is the utilization of a voxel-based model [6]. In a voxel model, space is discretized into small, uniform cubes whereas each cube is either filled with material or empty. This model is easy to implement and ready-to-use libraries are available. The drawback of this modeling technique is that the amount of voxels depends on the resolution by $O(n^3)$. This leads to both high memory demands and runtime if a higher resolution is used.

In the average case, the dexel format [2] demands only $O(n^2)$ memory while offering floating point accuracy in one selected direction. In this model the workpiece consists of parallel line segments that are arranged on a regular grid. Each end of a line segment corresponds to a point where the material is entered or left. To support the modeling of undercuts, each of the line segments must be replaced by a list of line segments.

In [2] an improved model is also described: by aligning three dexel boards with the main Cartesian axes, an equally high accuracy in each of the three directions can be achieved. Especially steep flanges benefit from this improvement.

These three models suffer more or less from discretization errors. The quantity of the error can be reduced by increasing the resolution. The modeling of surface structures resulting from dynamic effects demands a very high accuracy, which usually cannot be fulfilled with either voxel- or dexel-based models.

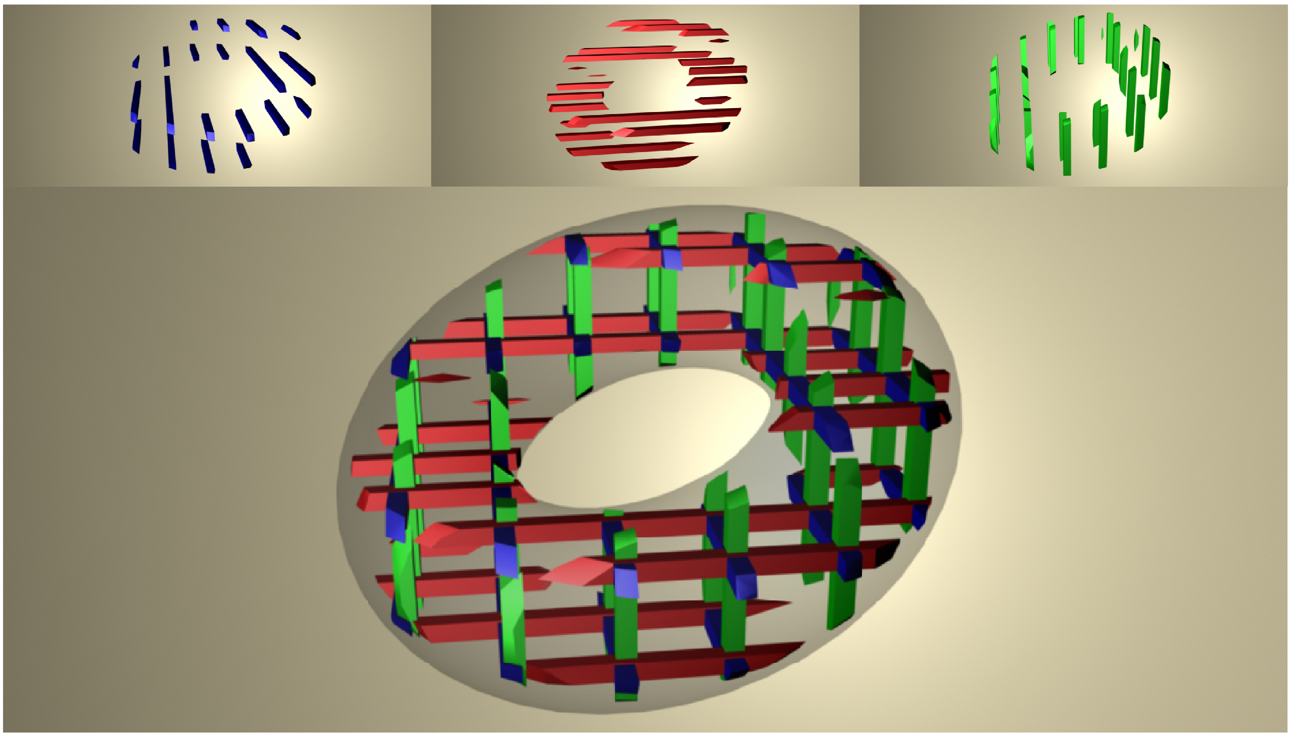


Figure 1: Torus modelled by a triple dexel model: X board (top left), Y board (top middle), Z board (top right) and their combination.

A higher accuracy can be achieved by using a Constructive Solid Geometry (CSG)-based [6] model which is a syntactical model. An object is described by a so-called CSG-tree. These Boolean trees consist of geometric primitives like spheres, cubes, etc., which form leafs of the tree. These primitives and each sub-tree can be connected by the Boolean set operators difference, union, and intersection (Fig. 2). Each (sub-)tree can be interpreted as a mathematical set and the operators form set operations. In simple implementations of milling simulations, the CSG-tree grows with every intermediate position of the tool. In this case the time necessary for each evaluation of the CSG-tree increases with the length of the previously milled path.

In this paper an enhanced CSG-based simulation of the dynamic milling process is presented. A more detailed description can be found in [7].

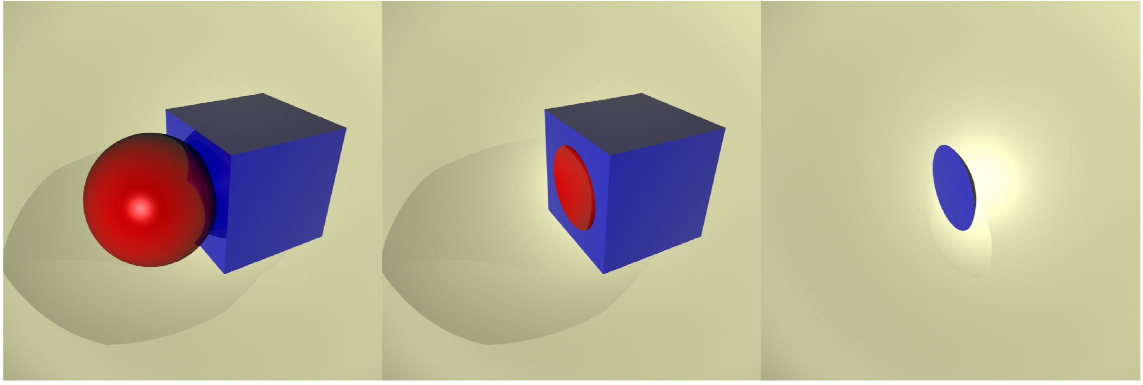


Figure 2: CSG Example: Primitives (here: sphere and cube) and the set operations: union (left), difference (cube-sphere) (middle), and intersection (right)

Oscillator Model As a simplification the tool can be interpreted as a mass point whose movement is determined by the model of a harmonic oscillator where the z-direction is assumed to be sufficiently stiff. Each of the two remaining directions of the tool movement is regarded independently (Fig. 3) and can be described by the equation

$$m\ddot{x}(t)+c\dot{x}(t)+kx(t)=F(x(t),x(t-T),t)\,. \tag{1}$$

x(t) designates the deflection, *k* is the spring rate, *c* is the damping constant, and *T* depicts the time between two tooth engagements. The force *F* on the tool depends on the current deflection *x(t)* and on the deflection at the time of the previous tooth engagement *x(t-T)*. This dependency shows the self-excited up-building character of the tool oscillations. The progression of forces on the tool can be approximated by a finite number of forces that are constant over a short span of time Δt. If *n* previous forces $F(t_i)$ each of the duration Δt are known, Eq. 1 can be solved in an approximate way by Eq. 2

$$x(t) = \sum_{i=0}^{n-1} \frac{F(t_i) \cdot \Delta t}{m \cdot \omega} \mathrm{e}^{-\gamma(t-t_i)} \sin(\omega \cdot (t - t_i)) \tag{2}$$

with ω denoting the angle velocity and γ being the damping constant.

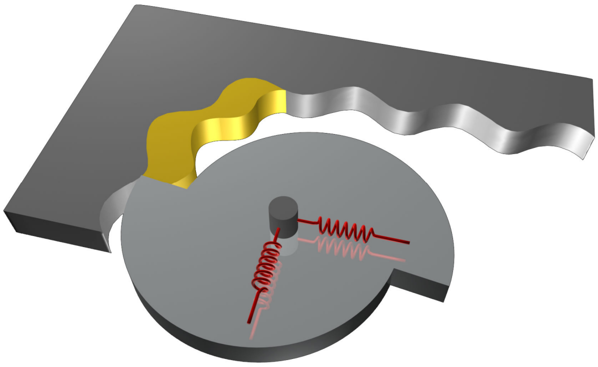


Figure 3: Regenerative effect leading to self-excited vibrations/ chatter

Force calculation For the calculation of forces on the tool, the cutting edge of the tool is split into many small cutting wedges (Fig. 4). For each wedge the forces in cutting direction, and normal and tangential to the cut are calculated based on the Kienzle equation. The overall force on the tool is the sum of forces across all wedges. The current chip thickness is an important input value for the Kienzle equation and can to be calculated by computing intersections between a straight line and the chip form. The generation of the chip form is covered in the next chapter.

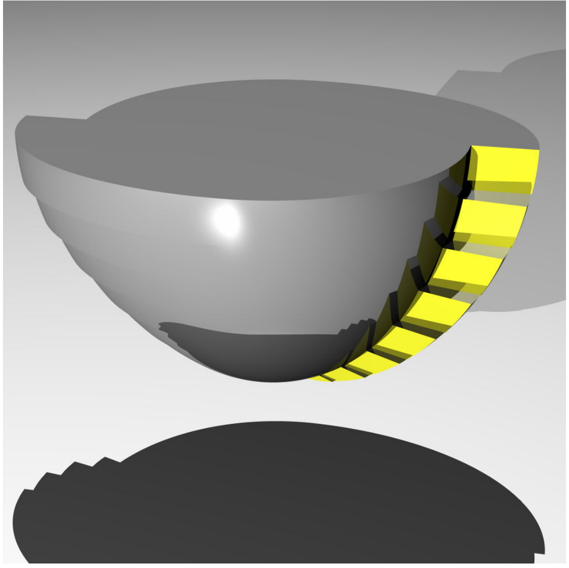


Figure 4: Model of the tool composed from several cutting wedges

Modeling of the chip form Each input NC program can be reduced to a sequence of short straight line segments. In the following the tool position is identified by a pair *(s, p)* with *s* being the index of the current line segment and *p* the distance of the tool from the beginning of segment number. *s*. The chip form *C* can then be described by Eq. 3

$$C(s,p)=\left(W_0-\bigcup_{i=0}^{s-1}V(i)\right)\cap\left(T\left(s,p\right)-T\left(s,p-f_z\right)\right) \tag{3}$$

where W_0 is the starting workpiece e.g. a cuboid, $V(i)$ is the sweep volume of the tool for segment *i,* and *T(s,p)* and $T(s,p-f_z)$ are the occupied volumes of the tool at position *(s, p)* and at the position of the previous tooth engagement, respectively. Eq. 3 can be modeled by the previous mentioned CSG method directly element by element. If the generation of the sweep volume $V(i)$ is too complex, it can be approximated by discretizing the tool movement and uniting the volumes of the tool at their distinct positions. For fast evaluation in the calculation of the chip thickness, it is not necessary to consider all sweep volumes V_i as they may be far away from the current place of interest and thus have no influence on the generated chip form. One way to avoid a lot of calculations is to sort the line segments into a data structure like octrees. Therewith, it is possible to efficiently find the line segments with an index lower than s that reside in an area around *(s, p).* These are the only segments that may influence the chip form. By this and additional optimizations, the size of the generated CSG-tree can often be reduced by about 80% without altering the chip form. The intersection of a straight line and a complete CSG-tree can be reduced to the intersection calculation between a straight line and the used primitives.

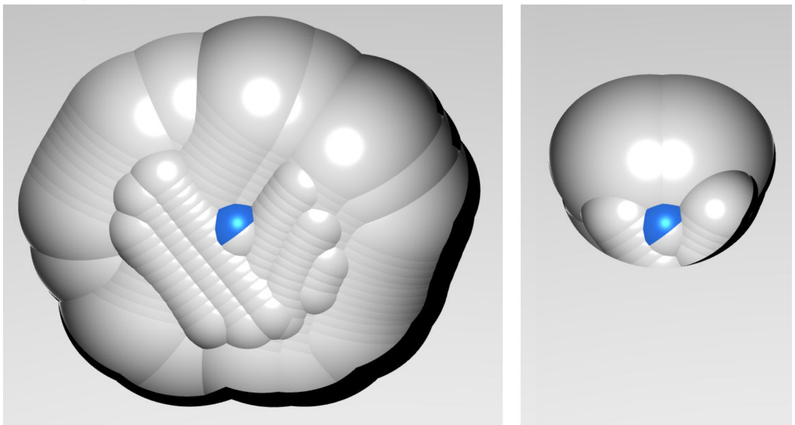
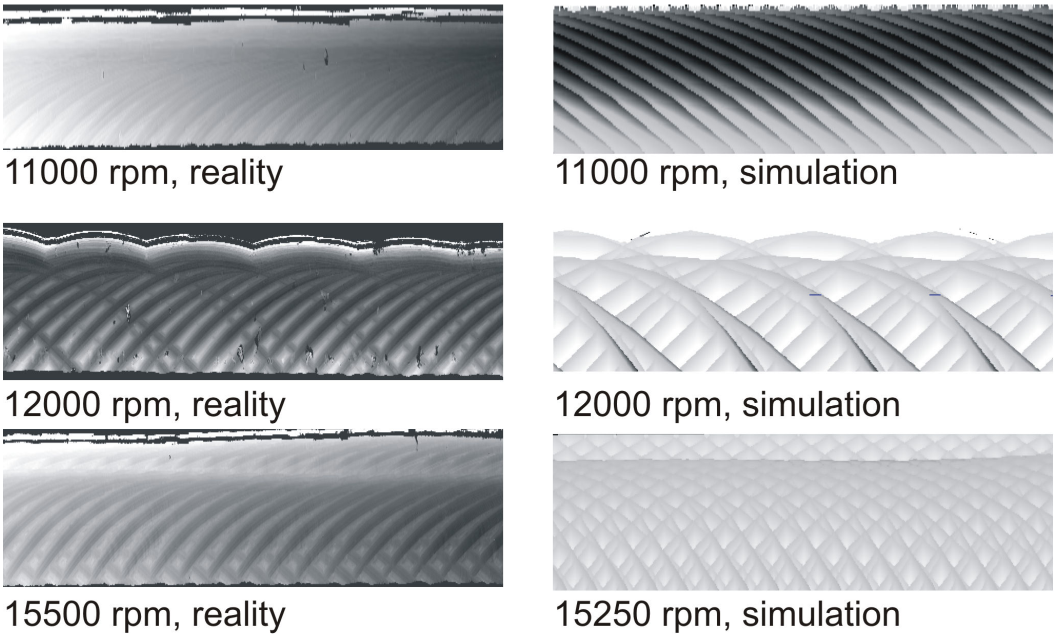


Figure 5: Workpiece model without (left) and with (right) minimization of the CSG tree. The generated chip form (blue) is identical

Modeling of tool vibrations For the simulation of the dynamic behavior, it is necessary to consider that the current chip thickness depends on the current tool deflection and on the tool deflection that has formed the current surface. This can be modeled by splitting the duration of each tool rotation into many steps. For the calculation of the chip thickness, the current deflection at a specific tool rotation angle and the tool deflection exactly one tooth engagement before are to be applied to the tool positions in the CSG-tree. By this procedure the self-excitation is modeled as the force on the tool influences the chip shape and the chip shape influences the force on the tool. Due to the fact the actual structure of the CSG-tree is not modified by this operation (only the values representing the two tool positions are altered), it can be accomplished very efficiently.

Merging the parts At this point all necessary parts for the simulation of the dynamic milling process have been described separately. The complete simulation works in the following order: First the chip form is calculated. Depending on the chip form the forces on the tool are determined and stored. With the knowledge of previous forces the deflection of the tool is computed. Based on this the deflections the next chip form can be calculated and so forfth. The surface can be modeled by superimposing the computed trajectory with the tool movement and subtracting the sweep volume of the tool from the stock material. For a correct modeling of the ground structure, it is important to consider that in case of oscillation the tool is not shifted but bent around a pivot. The resulting

CSG-tree can be visualized in photo quality by methods like ray-tracing and radiosity (Fig. 6). Complete stability charts can be generated by simulating different combinations of radial immersions and feedrates. By examining the trajectory of the tool, it is possible to draw conclusions whether chatter occurs or not.



11000 rpm, reality

11000 rpm, simulation

12000 rpm, reality

12000 rpm, simulation

15500 rpm, reality

15250 rpm, simulation

Figure 6: Visualization of ground structures: left column reality, right column simulation

Temperature Simulation

In the course of the scientific project TR10, reinforced aluminum extrusion elements have been successfully produced by the subproject A2. Currently the reinforcements consist of steel wires- the usage of other materials is planned. For many applications the machining of the reinforced extrusion elements is necessary. Complex multi-axis machining could lead to a considerable heat-up of the workpiece. In case of reinforced materials, this is very problematic because the different thermal expansion coefficients can lead to the build-up of stress or even bending of the material easily. The utilization of a temperature simulation makes it possible to recognize problems before the real machining could damage a workpiece.

Numerical Solutions For practical temperature problems, it is usually not feasible to compute a analytical solution. A remedy to this problem is the utilization of numerical solution procedures. In the field of mechanical engineering, it is common to calculate the solution to static and to dynamic problems with the help of the finite element analysis, which is capable of providing a numerical solution in fields like temperature simulation, deforming, stress calculation, etc. One disadvantage with the FEA is its long runtime. The computing time for a temperature simulation of a few seconds of machining can take many hours to days and therefore is not suitable for the simulation of whole NC programs.

The simpler finite difference method is a faster alternative. In this method a differential equation is solved by replacing the differentials with difference quotients, which introduces an error. Therefore, this method is in comparison to the FEA-method less exact but much faster. For the temperature simulation of non-homogenous materials, it has to be considered that the spatial differentiation of the heat conduction coefficient is not constant.

In our software implementation two different ways of partitioning the workpiece were chosen. In the simpler solution the workpiece is divided into a regular grid. In the second solution the workpiece can be split adaptively into tetrahedrons. Both implementations only consider heat conduction as both convection and thermal radiation are of little relevance for the temperature distribution inside the workpiece material.

Regular Grid In the simpler of the two implementations, space is split evenly in each of the three Cartesian directions. Each resulting voxel is treated as being homogenous i.e. each voxel has only one temperature value and conductivity coefficient. The accuracy is identical over the complete volume.

Adaptive Grid Areas of the workpiece that are far away from the NC paths or that have a low density of NC paths can be expected not to suffer from a high temperature build-up. These areas can easily be identified prior to the temperature simulation. It would be an optimal utilization of computing time if areas of higher interest were examined with a high resolution while the other areas could be discretized roughly. One possibility is to determine a tetrahedralization of the volume on a point set. It is planned to choose the density of vertices according to the importance of the area e.g. a high density should be chosen where many NC paths are near and a lower density in the remaining areas. As the temperature of each tetrahedron is represented by only one value, the generated tetrahedrons should be as compact as possible. Long, thin objects should be avoided. The Delaunay triangulation is a good way to fulfill these demands [8]. In 3D a Delaunay triangulation has the property that all tetrahedrons fulfill the so-called circumsphere constraint. This means that no other vertices reside inside the circumsphere of other tetrahedrons. Thus the minimum interior angle over all tetrahedrons gets maximized and the generated objects are usually compact.

As the tetrahedrons are distributed unevenly and as only one temperature value per tetrahedron is included in the calculation, it is necessary to find an interpolation scheme to deduce the current temperature on a random position. In our implementation this problem of scattered data interpolation is solved by the use of a radial basis function. Radial basis functions are functions whose value only depends on the distance of the parameter value to the origin or a chosen point e.g. $f(x)=f(|x|)$. The interpolation function is a linear combination of the basis functions, the general form is

$$f(\vec{p}) = \sum_{i=1}^{n} \alpha_i f_i(d(\vec{p}_i, \vec{p})) + p_m(\vec{p}) \tag{4}$$

where α_i are coefficients that are to be determined, $d(\vec{p}_i, \vec{p})$ is the distance between p_i and p, and p_m is a d-variate polynomial with dimension m.

The chosen basis functions are the Hardy's multiquadrics [9], functions of the form $f(d) = (d^2 + r^2)^\mu$ with $r>0$ and $\mu \neq 0$. The α_i can be determined by setting up a linear equation system with additional constraints whereas the polynomial p_m is zero.

Cellular Automaton The cellular automaton is a concept for abstract machines mainly used in mathematics and computer science, especially in computability theory [10]. It is used to model spatial discrete dynamic systems. Space is divided into discrete cells, usually formed as a lattice. Each cell is in one of a predefined finite number of states. Time is discretized into steps. The status of a cell at time $t+1$ depends only on its own status and on the state of the cells in a finite predefined neighborhood each at time t. Formally, a cellular automaton A is a tuple (d,S,V,f) where d (the number of dimensions of A) is a positive integer, S is a finite set of states, V is the finite neighborhood, and $f : S^V \to S$ is a local transition function. It has been proven that certain cellular automata are computationally universal. This means that their computational power is equivalent to that of a Turing machine and therefore that everything that can be calculated on a computer (and probably, according to the Church's theorem, everything that can be computed intuitively) can be calculated on a cellular automaton. With this background it is not surprising that the previously shown temperature simulations can be interpreted as a cellular automaton, but it is remarkable that the analogy is very direct and intuitive: The cells of the automaton correspond to the voxels of the regular grid and the finite neighborhood corresponds to the voxels surrounding the examined voxel. The temperature, though usually seen as a continuous value, is digitally stored with a fixed memory consumption and so the number of different temperature values is finite. Therefore, it can be interpreted as finite set of states. The calculation of a new temperature for a voxel depends on its previous temperature and the temperature of the voxels surrounding in a predefined neighborhood. The way it is implemented, the calculation directly fits into the requirements for the local transition function of the cellular automaton.

Conclusion and outlook

This paper shows advantages in two fields of milling simulations. For the simulation of dynamic effects in milling, the resulting flank and ground structures are generated and compared to machined ones, which show a very similar structure for the tool used. With the knowledge of the transfer function it should be possible to simulate the behavior of a complete production system. At the moment, oscillations of the workpiece, which may have a significant influence, are not considered. Additionally, the influence of the exact tool form should be analyzed.

For the temperature simulation, the runtime of the adaptive model should be compared to the one of the regular grid model. As the interpolation scheme in the case of the adaptive grid is very time consuming, alternative methods could be searched for. The resulting temperature distribution should be compared to experimental data.

Acknowledgement

This paper is based on investigations of the Collaborative Research Center SFB/TR10 which is kindly supported by the German Research Foundation (DFG).

References

[1] M. Stautner: Simulation und Optimierung der mehrachsigen Fräsbearbeitung, Dissertation, Vulkan Verlag, Essen, 2006, ISBN 3-8027-8732-3, 978-3-8027-8732-4

[2] P. Damm: Rechnergestützte Optimierung des 5-Achsen Simultanfräsens von Freiformflächen, Dissertation, Vulkan Verlag, Essen, 2006, ISBN 3-8027-8734-x, 978-8027-8734-8

[3] E. Ungemach, S. Odendahl, M. Stautner and J. Mehnen in: In-Process Simulation of Multi-Axis Milling in the Production of Lightweight Structures

[4] A. Zabel, M. Stautner, P. Kersting: Improvement of Machine Tool Movements for Simultaneous Five-Axes Milling; in: Proc. of the 5th CIRP Int. Seminar on Intelligent Computation in Manufacturing Engineering Ischia, Italy, pp. 159-164, ISBN: 88-95028-01-5

[5] A. Zabel, M. Stautner: Einsatzfelder der mehrachsigen Frässimulation, wt Werkstattstechnik online, 95 (2005) 1-2, S. 56-61

[6] J. D. Foley, A. Van Dam, S. K. Feiner, J. F.Hughes, Computer Graphics, Principles and Practice. Addison-Wesley Professional, 1995

[7] T. Surmann: Geometrisch-physikalische Simulation der Prozessdynamik für das fünfachsige Fräsen von Freiformflächen, Dissertation, Vulkan Verlag, Essen, 2006 ISBN 3-8027-8733-1, 978-8027-8733-1

[8] B. Delaunay: Sur la sphère vide. In: Bulletin of Academy of Sciences of the USSR 7 (1934), No. 6, pp. 793-800

[9] R. L. Hardy: Multiquadric equations of topography and other irregular surfaces. Jour. Geophys. Res. v. 76 no. 8, 1971, pp. 1905–1915

[10] J. von Neumann: The Theory of Self-reproducing Automata, A. Burks, ed., Univ. of Illinois Press, Urbana, IL. 1976