www.cadanda.com Computer-Aided Design AND APPLICATIONS™

Numerical Verification of CNC Machine Simulations

Gilad Israeli[1], Stephen Mann[2], Sanjeev Bedi[3] and Ajayinder Singh Jawanda[4]

[1]University of Waterloo, gilad.israeli@gmail.com
[2]University of Waterloo, smann@uwaterloo.ca
[3]University of Waterloo, sanjeev.bedi@uwaterloo.ca
[4]Thapar University, Patiala, asjawanda@thapar.edu

ABSTRACT

A CNC machine simulator is a complex piece of software. A machine model, a stock model, and a tool model must be designed, and algorithms for intersecting the partially machined stock with the surface swept by the tool must be implemented. Verification that the software is working correctly is critical if the simulator is used for industrial purposes. In this paper, we describe a method to test that the intersection of the tool and the surface swept by a 5-axis machine is correctly generated by a simulator, where the stock is represented as a height field. Further, these tests assist in determining the appropriate values for many of the internal parameters of the simulator.

Keywords:CNC simulation, grazing curve, swept surface, error evaluation.
DOI: 10.3722/cadaps.2011.507-518

1 INTRODUCTION

Numerically controlled (NC) machining was developed in the 1940s and 1950s in response to the requirements of jet powered aviation for precision parts with smaller tolerances for error [5]. Over time thecontrol units for NC machines evolved from a simple mechanical or electronic controller to a microcomputercontroller. This resulted in a better user interface, the ability to do limited tool path verification, as wellas some awareness of the physical properties of the tool. However even today's Computer Numerically Controlled (CNC) machines are only focussed on controlling their own motions. The machines are not aware ofthe final part, and only execute the tool path given to them. Thus, the machine will accept the tool pathblindly, even if it will cause the machine to damage the resulting part or the machine itself.

Another problem in CNC machining is that the parts milled by the machine are not exactly the same as the mathematical representation of the part. This is because the tool is not infinitely small and cannot pass directly over all areas of the part. Instead the field of CNC machining is interested in ensuring that this error is kept below some user specified threshold. CNC machining is also interested in computing the maximum error in a resulting part; however measuring this on a real part is a difficult, tedious and time consuming task requiring special equipment.

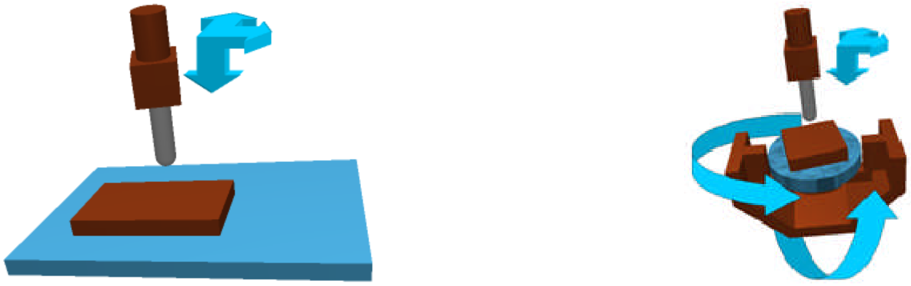
One solution is to design a simulator that creates a mathematical model of the milled part. This would allow us to predict what the machine will do when executing a tool path and prevent it from damaging itself or the part. Computing machining error from the results of the simulation is much easier and cheaper than doing the same on a real part. In particular, we can use the results of the simulation to compare the machined surface to the design surface; we can check for scallop height; we can test for gouging and undercutting; and we can check for interference between the tool and other parts of the machine. However, with the possible exception of the last of these three, there is an implicit assumption that the simulation creates a model of the surface that is an accurate representation of the machined surface.

A CNC simulator is a complex piece of software. It must model the CNC machine, the stock, and the tool, and it must intersect the surface swept by the tool with the stock to obtain a representation of the machined surface. The focus of this paper is to design some tests to gain confidence that the intersection between the stock and the swept surface is implemented correctly. Our test is appropriate for simulators that use height fields to represent the machined stock.

In the next section, we review some relevant ideas and literature. In Section 3 we will review NC machining simulators, with a focus on the details that are required for accurate simulation. Section 4 is the core of our paper, where we describe the theory behind our tests, state the test themselves, and apply them to our own simulator to verify that its intersections tests were working correctly. Further, in running the tests, we uncover the relationship between some of the internal parameters of the simulator.

2 BACKGROUND

Although there are many different possible machine configurations, two common configurations are shown in Fig. 1. The most common configuration in industrial applications is the three axis machine (Fig. 1(a)). Here the stock moves along the x, y, and$z$ translation axes in relation to the tool. The five axis machine is a more flexible machine type, adding two rotation axes to the three axes configuration; Fig. 1(b)., gives an example, although there is great variation in how the rotational axes are configured on the machine. These rotations allow for a greater number of tool orientations, which can be used to reduce machining time. The tests we describe in this paper apply to both 3-axis and 5-axis machines.



(a) Three axes.

(b) Five axes.

Fig. 1: Machine Types.

While machining, the tool spins much faster than the rate of motion of the tool relative to the stock. Therefore, when analysing tool motion it is common to treat the tool as if it were a *surface of revolution*, such as a sphere, cylinder, or torus. In this paper when we refer to the tool, we mean the surface of revolution of the tool.

Tool paths are a set of positions of the machine's axes, which are known as *machine coordinates*. The machine linearly interpolates these coordinates to move between these positions while machining the part.Tool paths are generally specified as a set of commands known as *g-codes*. Besides machine coordinates, g-codes also contain commands for setting tool properties or changing CNC machine settings [5]. While the syntax of the format is relatively consistent, the semantics of the codes can vary greatly between different manufactures of machines.

2.1 Previous Work

The seminal work on real time simulation and display of CNC machines was published by Van Hook [3]. He used a z-map to represent the stock, which is similar though less flexible than the approach we will take.

The simulation presented in this paper makes heavy use of swept surfaces, introduced in CNC machining by Blackmore et al.[1]. Initially swept surfaces were computed by solving differential equations, such as the Sweep-Envelop Differential Equation (SEDE) method used by Wang et al. [12]. This paper works from a different approach, using the work of Roth et al.[10] and Mann and Bedi[7]. Roth et al. computed the swept surface of a toroidal tool using grazing points. These points were computed by partitioning the tool with planes perpendicular to its axis of revolution into a finite number of cross sections. The direction of motion of the insert and the insert's normal vector was used to compute the grazing point on the surface of the insert. The direction of motion of the insert was computed using the current and next position of the insert. Later, Mann and Bedi[7] generalized this idea to work with any surface of revolution.Our own previous related work [6],[7],[8],[10] investigated the useof grazing curves and error metrics in CNC simulations, as well as various hardware techniques for CNC simulations; this paper investigates methods to verify the correctness of the simulator.

A variety of work has been done on computing errors in simulations of swept surface. For example, Tutunea-Fatan and Feng[11] compute and compare the errors of a piecewise linear tool motion to tool motion based on machine kinematics. However, we are unaware of the analysis of the error in the simulated NC machine surface using height fields as a means of verifying the correctness of a CNC simulator.

3 NC MACHINING SIMULATION

This section describes the ideas behind NC machining simulations. The simulator used in our tests was*ToolSim*, a simulator written at the University of Waterloo. Much of the discussion in this section will befocused on ToolSim, and why we made the design decisions we made. However, most of this discussionshould apply to any NC machining simulator that uses some form of a height field.

The architecture of ToolSim can be grouped into three major components: the stock, the machine, andthe tool path. Together these components are used to compute the parts resulting from milling. The toolpath provides the machine positions used to create the part. The machine is responsible for computing theposition and motion of the tool relative to the stock. Finally the stock is responsible for computing theintersection between the machine tool and the stock, as well as storing the result.

There are a number of ways to represent the stock. The straightforward way is with a three dimensionalmatrix, with each entry representing a discreet volume. However there are many challenges to successfullyimplementing this representation. One problem is that the memory cost of this representation grows quicklywith stock density and size. Other problems are that both the intersection of geometric shapes with volumesand the rendering of volumes are nontrivial.

Fortunately, most parts created by CNC machining (even by 5-axis machines) can be represented by avertical height field on a regular grid. This height field is then tessellated into a surface of triangles to be rendered (Fig. 2).*Stock density* is defined as the amount of height numbers per unit area of the grid. Thedensity can be varied to achieve either faster or more accurate results.

For CNC machining simulations, one needs to know the toolpath to simulate the machining. Abstractly,once we have the path, we compute the surface swept by the tool, and merge multiple passes of the toolto compute the machined surface. However, in real machines, the path is specified with g-codes, and thetool position is only specified at discrete locations. The question arises as to what path the tool travels inbetween these tool positions.

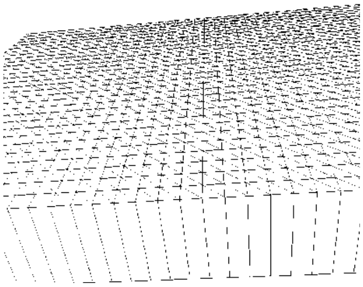
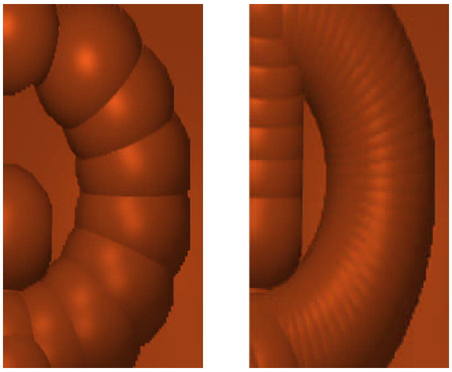


Fig. 2: Stock representation with a vertical height field.

One approach is to assume piecewise linear motion of the tool between tool positions. While sufficientfor simulating 3-axis machining, piecewise linear interpolation will introduce errors in 5-axis machining, asit ignores the kinematics of the machine. By modeling the machine kinematics, we can reduce the error inthe simulation[8].

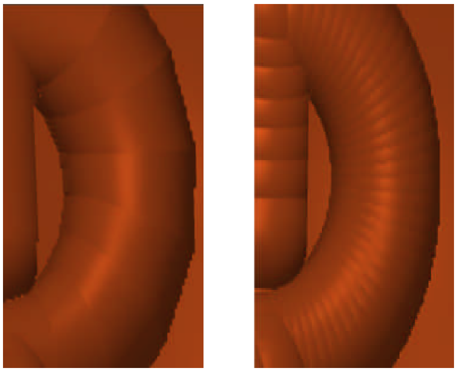
ToolSim uses a hierarchical model to represent the machine. This representation is used to render themachine and compute the position of the tool relative to the stock. By accurately modeling the movingcomponents of the machine, we are able to compute precise tool motions. For more details on this type ofmachine model, see[4],[8].

The main purpose of ToolSim is computing the results of machining. The simplest way to achieve this is to intersect the tool with the stock at all tool positions and in-between steps. This is referred to as *stamping*. However because stamping only takes into account the discrete positions of the tool and not the continuous motion of the tool, this will result in high simulation error (Fig. 3(a)). The error can be reduced by increasing the number of in-between steps in the tool path, but this requires a large number of stamps which slows down the computation speed Fig. 3(b). A better solution is to use swept surfaces.



(a) Large step size. (b) Small step size.

Fig. 3: Intersection using stamping.



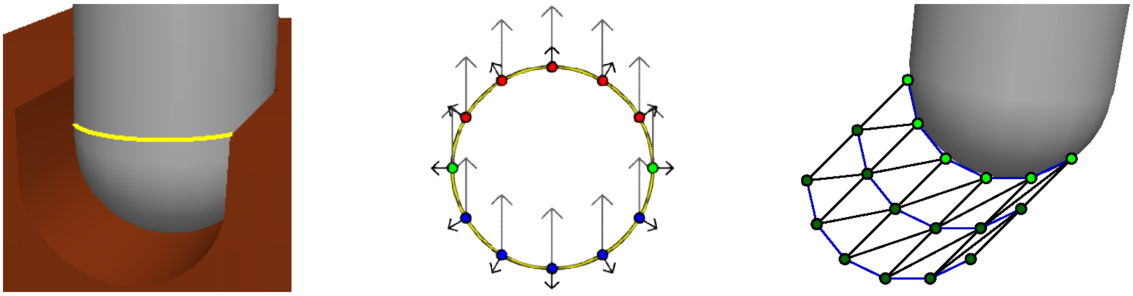
(a) Large step size. (b) Small step size.

Fig. 4: Intersection using swept surfaces.

The*swept surface* of a tool is the surface of the volume of space that the tool moves through. Intersectionof the stock with a swept surface, instead of just using stamping, gives more accurate results without slowingdown the computation speed.

A swept surface generally has a complex shape. Consequently the representation we used for the sweptsurface is a piecewise polygonal surface. To construct this approximation we first note that the tool isalways in contact with the swept surface. This region of contact forms a curve on the surface of the tool. Byconnecting the points on these curves together we can create the approximation to the swept surface (Fig.5(c)).This surface representation gives a reasonable approximation to the swept surface. However if the path hasany angular motion, the step size needs to be small to avoid high simulation errors, (compare Fig.4(b) to Fig.4(a)).

Grazing curves are the curves that form the contact regions between the tool and swept surface. If weimagine the tool moving through the swept surface then the place where the tool touches the swept surfacewill always be on the sides of the tool, relative to the direction of motion. This gives us a way to computethe grazing curves. More formally, the points in a grazing curve will be the points on the surface of the toolwhere the surface normal is perpendicular to the direction of motion of the tool[1]. This is illustratedin Fig. 5. To simplify the illustration we will only look at a cross section of the tool. The yellow circle inFig. 5(b) is the cross section of the tool, shown in yellow, in Fig.5(a). In Fig. 5(b) the larger grey arrowsrepresent the direction of motion and the smaller black arrows are the normals to the tool surface at thesepoints.The red points are points along the surface of the tool that will cut the stock, while the bluepoints are points that do not cut the stock. Only the points where the normal is perpendicular tothe direction of motion, marked in green, will remain on the stock after the tool has moved through thattool position. After computing the location of the green points for other cross sections they are combinedto create a piecewise linear approximation of the grazing curve, (the blue lines in Fig.5(c)).



(a) A cross section of the tool.

(b) The grazing curve points on the cross section.

(c) Swept surface created from the grazing curve points.

Fig. 5: Construction of a swept surface.

Sometimes the change in the direction of the tool motion is discontinuous. This can occur when the machine finishes moving into one tool position and begins to move into another. At this point there will be a gap in the swept surface (Fig. 6). To overcome this the tool is intersected with the stock at the point of discontinuity. In between machine positions the machine coordinates are linearly interpolated, so the change in direction is always continuous.

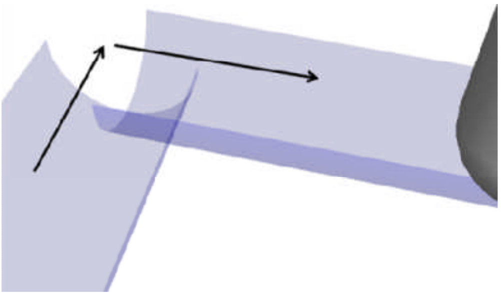


Fig. 6: Swept surfaces resulting from a discontinuous tool path.

Since the stock is represented using height fields each line segment of the height field can be treated as a ray. These rays are then intersected with the shape of the tool. Algorithms for intersecting rays with geometric objects and triangles have been thoroughly investigated for ray tracing[2]. If a ray intersects the shape, the location of the intersection is computed. If the location is lower than the value ofthe corresponding height field then the height field is updated with the new value.

4 NUMERICAL VERIFICATION OF TOOLSIM

Having implemented a simulator, it is important to know that it is constructing the correct surface. Simple visual inspection of the results can pick out the most glaring errors. However, more sophisticated tests are required to verify this moderately complex piece of software.

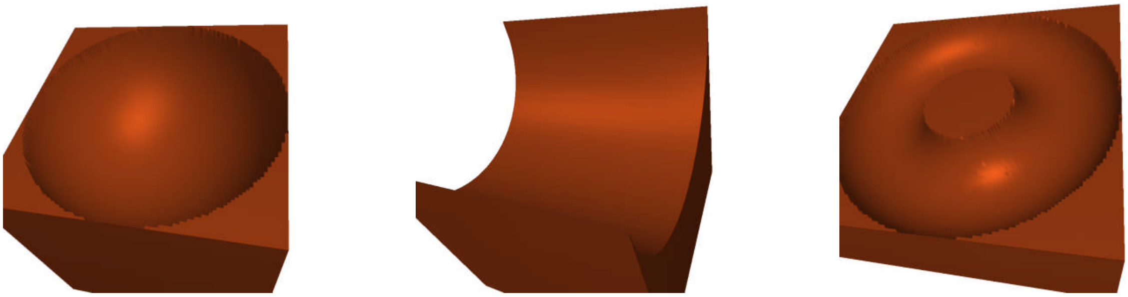
In this section, we describe the tests we used to validate the intersection computation of our simulator. The basic idea of the test is to compute the error between a machined stock and a design surface. As noted in Section 4.2 there is a theoretical bound on the resulting error. If the error matches these bounds then we have confidence that the implementation is correct.

Running a numerical test involves simulating machining of a simple surface and then comparing the machined stock to the design surface. Computing this comparison over the entire surface gives us the maximum error. If we only compare the height field with the design surface then the error should be on the order of floating point precision. The realization is that the machined stock's height field representation is not a set of points in space but a piecewise linear approximation of a surface. By comparing the surface at points in between the points on the stock height field grid we can get a sense of the error in our approximation.

4.1 Design Surfaces

The design surfaces were selected to isolate particular simulation features. We wanted surfaces to test both stamping and sweeping motions, and we wanted surfaces to test the translational and rotational motions of the machines.

We used three different design surfaces for our tests: a hemisphere, a half-cylinder and a half-torus. Aspherical tool was used to machine all these surfaces. For the hemisphere, the machine stamped the centreof the stock (Fig.7(a)). For the half-cylinder, the machine swept the tool horizontally along the x-axisthrough the stock (Fig.7(b)). For the half-torus the tool was placed on an angle and rotated around thevertical axis (Fig.7(c)).



(a) Hemisphere.

(b) Half-cylinder.

(c) Half-torus.

Fig. 7: Verification tests.

Each design surface tests different aspects of the simulation. The hemisphere tests stamping, while thehalf-cylinder and half-torus test grazing curves. The half-cylinder design surface tests grazing curves that areaxis aligned and have no rotational motion, while in the half-torus the grazing curves are not axis aligned.The tests using these surfaces also test different simulation variables, as noted in Section 4.4.

4.2 Error Metrics and Bounds

There are a number of ways to measure the error between two surfaces [6]. The simplest way is to measure the difference between the height field and the *z*-value of the surface at that point. This is simple to compute because both the stock and design surface are functional. Another advantage is that there is a theoretical error bound on this metric. This is the bound that we use to verify that ToolSim works correctly. For these reasons, we use this vertical computation of error in this paper.

Given a design surface *f*, we can compute an error bound on the simulated piecewise linear approximationof the simulated machined surface. As long as our design surface is C^2 the error bound is

$$\|f - s\| = M h^2, \tag{1}$$

where *f* is the design surface, s is the approximation function, h = max $\{h_x, h_y\}$, h_x is the maximum distancebetween adjacent sample points in the *x* direction, and h_y is similarly defined in the *y* direction. The valueof *M* is

$$M = 4.\max_{i+j=2}\left\|D_x^i D_y^j f\right\|, \;\; \forall i,j \;\; 0 \leq i,j \leq 2 \tag{2}$$

when linear interpolation is used[9]. With this error bound if the density of *s* is doubled then the errorgiven by Eqn.(1) should drop by a factor of four.

One minor complication for our tests is that the constant *M* depends on the derivatives of the design surface *f*. For our tests, as you approach the edge between the plane and the hemisphere, half-cylinder, or half-torus, the derivatives approach infinity. This results in unbounded error near the edge. To avoid this unbounded theoretical error, we excluded data along the edge of the shape. For the hemisphere we used only the area within $0.6 * radius$of the centre sample. For the half-cylinder it is the area within $0.6 * radius$of the cylinder axis and for the half-torus it is the area within $0.6 * radius_{minor}$of the circle in the middle of the torus tube. We refer to this area as the *range of interest*.

4.3 Density Variables

There are many different densities used in ToolSim and in the tests in this section. The *stock density* is the number of height points per unit area of the stock representation. The *grazing curve density* is the number of points used to construct the grazing curve. The *sampling density* is the number of points we used per unit area when interpolating the stock values to estimate the errors; a sampling density of one only samples at the stock points, where as a sampling density of two or higher would also sample between the stock points. Finally the *in-between-steps* are the number of positions between each tool position. This is related to the grazing curve density as discussed in Section 4.4.

While we are primarily interested in how the error changes as we increase the stock density, as observed in the next section, many of these other parameters have to be dense enough to see the theoretical change in error.

4.4 Results and Analysis

An initial pre-test was run to determine the appropriate parameters to use in the simulations. This test also allowed us to verify that stamping was implemented correctly. Another two tests were then run with the other two design surfaces to verify that the swept surface intersections of the simulator wereworking correctly.

For the tests done using the hemisphere as the design surface, the maximum error was computed repeatedly while sampling density and stock density were varied. For the test using the half-cylinder and half-torus, the maximum error was computed repeatedly while grazing curve density and stock density were varied to see their effect on the error. Tab. 1 contains the parameters used with each design surface type when they were sampled using the vertical error metric and closest point on triangles within the continuous area.

	hemisphere	half-cylinder	half-torus
stock width	10 mm		30 mm
stock length	10 mm		30 mm
stock depth	10 mm		
tool	sphere		
tool radius	5 mm		
sampling	linear		
sampling density	{1, 2, 4, 8}	2	
stock density	{1, 2, 4, 8, 16, 32, 64, 128}		{1, 2, 4, 8, 16, 32, 64}
grazing curve density	NA	{1, 2, 4, 8, 16, 32, 64, 128}	
angle step	NA		4 * $density_{\text{grazing curve}}$

Tab. 1: Parameters used in the tests for each design surface.

In addition to the test data, we give tables of the ratio between the columns and rows of the tests using the vertical height error. This makes it easier to see the effect of the h^2 term in Eqn. 1.

Before running the verification tests, we first determined the density at which to sample each triangle in ToolSim's piecewise linear approximation of the machined surface to estimate the error. To find the optimal sampling density, sampling density was one of the parameters we varied while running the test using the hemisphere. Tab. 2 shows that increasing the sampling density beyond two does not change the computed error. Therefore, for the sake of speed, we used a samplingdensity of two for the verification tests.

The simplest verification test used the hemisphere, which only involved stamping and not swept surfaces. Therefore the only factor affecting s in Eqn. 1 is the stock density. If we look at the ratio between the columns of Tab. 2 (illustrated by Tab. 3), then we can see that the error decreases by a factor of four when the stock density is doubled, verifying that ToolSim matched the theoretical convergence of Eqn. 1 when only stamping is used.

The results of the vertical error tests for the cylinder and torus are displayed in Tab. 4. For the cylinder and torus test when we doubled both the grazing curve and stock density we also saw a factor of four decrease in error (Tab. 5(c) and Tab. 6(c)). If we only increased one of them there was limited improvement (Tab.s 5(b), 5(a), 6(b), and 6(a)) implying a relationship between these parameters.

		stock density							
		1	2	4	8	16	32	64	128
sample density	1	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
	2	0.0939004	0.0225692	0.0058032	0.0014800	0.0003772	0.0000949	0.0000238	0.0000059
	4	0.0939004	0.0225692	0.0058032	0.0014800	0.0003772	0.0000949	0.0000238	0.0000059
	8	0.0939004	0.0225692	0.0058032	0.0014800	0.0003772	0.0000949	0.0000238	0.0000059

Tab. 2: Maximum Error (mm) - Hemisphere Test with vertical error metric and 0.6 range of interest.

		stock density$_i$/ stock density$_{i+1}$						
		d_1/d_2	d_2/d_4	d_4/d_8	d_8/d_{16}	d_{16}/d_{32}	d_{32}/d_{64}	d_{64}/d_{128}
sample density	1	undefined	undefined	undefined	undefined	undefined	undefined	undefined
	2	4.16	3.88	3.92	3.92	3.97	3.98	3.99
	4	4.16	3.88	3.92	3.92	3.97	3.98	3.99
	8	4.16	3.88	3.92	3.92	3.97	3.98	3.99

Tab. 3: Analysis of hemisphere test data - ratio between successive stock densities.

		stock density							
		1	2	4	8	16	32	64	128
grazing curve density	1	0.1187823	0.0894227	0.0894227	0.0894227	0.0894227	0.0894227	0.0894227	0.0894227
	2	0.0495521	0.0297029	0.0206271	0.0206271	0.0206271	0.0206271	0.0206271	0.0206271
	4	0.0434738	0.0132945	0.0072835	0.0050696	0.0050696	0.0050696	0.0050696	0.0050696
	8	0.0396484	0.0117982	0.0036838	0.0016805	0.0012278	0.0012278	0.0012278	0.0012282
	16	0.0389957	0.0107988	0.0029091	0.0009255	0.0004528	0.0003018	0.0003018	0.0003018
	32	0.0388863	0.0108109	0.0029092	0.0007882	0.0002412	0.0001129	0.0000746	0.0000749
	64	0.0388534	0.0107687	0.0028660	0.0007497	0.0002030	0.0000621	0.0000299	0.0000184
	128	0.0388418	0.0107640	0.0028599	0.0007404	0.0001910	0.0000516	0.0000155	0.0000073

Tab. 4(a): Maximum Error (mm) - with vertical error metric and 0.6 range of interest-Half-cylinder Test.

		stock density						
		1	2	4	8	16	32	64
grazing curve density	1	0.1363228	0.1037322	0.0995873	0.0980008	0.1006394	0.1009843	0.1009843
	2	0.0989301	0.0381199	0.0267250	0.0247300	0.0247325	0.0249510	0.0250009
	4	0.0948554	0.0248746	0.0104826	0.0067159	0.0061857	0.0062443	0.0062604
	8	0.0938597	0.0241302	0.0071245	0.0026564	0.0016853	0.0015559	0.0015559
	16	0.0935855	0.0235213	0.0061061	0.0017751	0.0007034	0.0004313	0.0003908
	32	0.0935451	0.0234599	0.0060048	0.0015778	0.0004581	0.0001829	0.0001064
	64	0.0935345	0.0234636	0.0059931	0.0015431	0.0003973	0.0001150	0.0000486
	128	0.0935345	0.0234562	0.0059931	0.0015270	0.0003852	0.0001012	0.0000305

Tab. 4(b): Maximum Error (mm) - with vertical error metric and 0.6 range of interest-Half-torus Test.

		stock density							
		1	2	4	8	16	32	64	128
grazing curve density$_i$ / grazing curve density$_{i+1}$	d_1/d_2	2.39	3.01	4.33	4.33	4.33	4.33	4.33	4.33
	d_2/d_4	1.13	2.23	2.83	4.06	4.06	4.06	4.06	4.06
	d_4/d_8	1.09	1.12	1.97	3.01	4.12	4.12	4.12	4.12
	d_8/d_{16}	1.01	1.09	1.26	1.81	2.71	4.06	4.06	4.06
	d_{16}/d_{32}	1.00	0.99	0.99	1.17	1.87	2.67	4.04	4.02
	d_{32}/d_{64}	1.00	1.00	1.01	1.05	1.18	1.81	2.49	4.05
	d_{64}/d_{128}	1.00	1.00	1.00	1.01	1.06	1.20	1.92	2.51

Tab. 5(a): Analysis of half-cylinder test data- ratio between successive grazing curve densities - (each cell $t_{i,j}$ of this table is $d_{i,j}/d_{i,j+1}$ where d is a cell from Tab. 4(a))

		stock density$_i$/ stock density$_{i+1}$						
		d_1/d_2	d_2/d_4	d_4/d_8	d_8/d_{16}	d_{16}/d_{32}	d_{32}/d_{64}	d_{64}/d_{128}
grazing curve density	1	1.32	1.00	1.00	1.00	1.00	1.00	1.00
	2	1.66	1.43	1.00	1.00	1.00	1.00	1.00
	4	3.27	1.82	1.43	1.00	1.00	1.00	1.00
	8	3.36	3.20	2.19	1.36	1.00	1.00	0.99
	16	3.61	3.71	3.14	2.04	1.50	1.00	1.00
	32	3.59	3.71	3.69	3.26	2.13	1.51	0.99
	64	3.60	3.75	3.82	3.69	3.26	2.07	1.61
	128	3.60	3.76	3.86	3.87	3.70	3.32	2.10

Tab. 5(b): Analysis of half-cylinder test data- ratio between successive stock densities-(each cell $t_{i,j}$ of this table is $d_{i,j}/d_{i+1,j}$ where d is a cell from Tab. 4(a))

		stock density$_i$/ stock density$_{i+1}$						
		d_1/d_2	d_2/d_4	d_4/d_8	d_8/d_{16}	d_{16}/d_{32}	d_{32}/d_{64}	d_{64}/d_{128}
$\frac{\textit{grazing curve density}_i}{\text{grazing curve density}_{i+1}}$	d_1/d_2	3.99	4.33	4.33	4.33	4.33	4.33	4.33
	d_2/d_4	3.72	4.07	4.06	4.06	4.06	4.06	4.06
	d_4/d_8	3.68	3.60	4.33	4.12	4.12	4.12	4.12
	d_8/d_{16}	3.67	4.05	3.98	3.71	4.06	4.06	4.06
	d_{16}/d_{32}	3.60	3.71	3.69	3.83	4.00	4.04	4.02
	d_{32}/d_{64}	3.61	3.77	3.88	3.88	3.88	3.77	4.04
	d_{64}/d_{128}	3.60	3.76	3.87	3.92	3.93	4.00	4.06

Tab. 5(c): Analysis of half-cylinder test data- ratio between successive stock and grazing curve densities -(each cell $t_{i,j}$ of this table is $d_{i,j}/d_{i+1,j+1}$ where d is a cell from Tab. 4(a))

		stock density						
		1	2	4	8	16	32	64
$\frac{\textit{grazing curve density}_i}{\text{grazing curve density}_{i+1}}$	d_1/d_2	1.37	2.72	3.72	3.96	4.06	4.04	4.03
	d_2/d_4	1.04	1.53	2.54	3.68	3.99	3.99	3.99
	d_4/d_8	1.01	1.03	1.47	2.52	3.67	4.01	4.02
	d_8/d_{16}	1.00	1.02	1.16	1.49	2.39	3.60	3.98
	d_{16}/d_{32}	1.00	1.00	1.01	1.12	1.53	2.35	3.67
	d_{32}/d_{64}	1.00	0.99	1.00	1.02	1.15	1.58	2.18
	d_{64}/d_{128}	1.00	1.00	1.00	1.01	1.03	1.13	1.58

Tab. 6(a): Analysis of half-torus test data- ratio between successive grazing curve densities - (each cell $t_{i,j}$ of this table is $d_{i,j}/d_{i,j+1}$ where d is a cell from Tab. 4(a))

		stock density$_i$/ stock density$_{i+1}$					
		d_1/d_2	d_2/d_4	d_4/d_8	d_8/d_{16}	d_{16}/d_{32}	d_{32}/d_{64}
grazing curve density	1	1.31	1.04	1.01	0.97	0.99	1.00
	2	2.59	1.42	1.08	0.99	0.99	0.99
	4	3.81	2.37	1.56	1.08	0.99	0.99
	8	3.88	3.38	2.68	1.57	1.08	1.00
	16	3.97	3.85	3.43	2.52	1.63	1.10
	32	3.98	3.90	3.80	3.44	2.50	1.71
	64	3.98	3.91	3.88	3.88	3.45	2.36
	128	3.98	3.91	3.92	3.96	3.80	3.30

Tab. 6(b): Analysis of half-torus test data- ratio between successive stock densities- (each cell $t_{i,j}$ of this table is $d_{i,j}/d_{i+1,j}$ where d is a cell from Tab. 4(a))

		stock density$_i$/ stock density$_{i+1}$					
		d_1/d_2	d_2/d_4	d_4/d_8	d_8/d_{16}	d_{16}/d_{32}	d_{32}/d_{64}
$\frac{grazing\ curve\ density_i}{\text{grazing curve density}_{i+1}}$	d_1/d_2	3.57	3.88	4.02	3.96	4.03	4.03
	d_2/d_4	3.97	3.63	3.97	3.99	3.96	3.98
	d_4/d_8	3.93	3.49	3.94	3.98	3.97	4.01
	d_8/d_{16}	3.99	3.95	4.01	3.77	3.90	3.98
	d_{16}/d_{32}	3.98	3.91	3.86	3.87	3.84	4.05
	d_{32}/d_{64}	3.98	3.91	3.89	3.97	3.98	3.76
	d_{64}/d_{128}	3.98	3.91	3.92	4.00	3.92	3.76

Tab. 6(c): Analysis of half-torus test data- ratio between successive stock and grazing curve densities - (each cell $t_{i,j}$ of this table is $d_{i,j}/d_{i+1,j+1}$ where d is a cell from Tab. 4(a))

Tab.s 5(a) and 6(a) indicate that increasing the grazing curve density without increasing the stockdensity produces diminishing returns. Similarly increasing the stock density without increasing the grazingcurve density produces zero returns after some point (Tab.s 5(b) and 6(b)). This suggests that the grazingcurve density should be equal to stock density.

In running the experiments we found that a constant number of in-between steps between tool positionsdid not result in the factors of four error reduction seen in Tab. 6(c), and the error did not improve as itdid for the other tests. We realized that this was because the distance between steps affects the simulationin the same way that the grazing curve density does, but in the direction of motion of the tool, as opposedto perpendicular to the direction of motion. By setting the number of in-between steps to be

$$4 * density_{grazing\ curves},$$

we were able to get the theoretical factor of four improvements.

Overall, this sequence of tests showed that, using the appropriate densities, our simulator producedresults that matched mathematical theory.

5 SUMMARY

This paper described theoretical error bounds of piecewise linear approximation and showed how to usethem to verify that stamping and swept surface intersections of a 5-axis CNC simulator are functioningcorrectly. Beyond just verifying that the error decreases as a factor of four as the stock density is doubled,we found that other parameters of the simulation also have to be increased to achieve this factor of fourerror reduction.

6 CONCLUSION

CNC machine simulators are a useful tool for verifyingtoolpaths without having to machine a test piece.However, the simulator itself must be verified as functioning correctly for it to be of use in practice. Thetests described in this paper are a first step in such a verification process. While the tests described in thispaper were for a particular simulator, these tests should be applicable to other simulators that represent thestock as a height field.

Other tests will need to be performed to fully verify the simulator. In particular, tests to verify that the machine has been correctly modeled. At a minimum, a comparison between the simulated surface and themachined surface need to be made. Further, when computing the error between the design surface and thesimulated machined surface, a closest point error metric should be employed. However, the tests in thispaper provide a way to verify one portion of the code, and should prove useful in locating coding errors andin determining the appropriate settings of the internal parameters of the simulator.

REFERENCES

[1] Blackmore,D.;Leu,M.C.;Wang L.P.: Applications of flows and envelopes to NC machining, Annals of the CIRP, 41(1), 1992, 493-496.DOI:10.1016/S0007-8506(07)61252-9

[2] Glassner,A. S.: editor. An Introduction to Ray Tracing, Academic Press LTD., San Diego, CA 92101, 1989.ISBN:0-12-286160-4

[3] Hook,T. V.: Real-time shaded NC milling display, In SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques, New York, NY, USA, ACM Press, 1986, 15-20.DOI:/10.1145/15922.15887

[4] Israeli,G.: Software simulation of numerically controlled machining, Master's thesis, University of Waterloo, 2006.

[5] Lee,K.: Principles of CAD/CAM/CAE, Addison Wesley, Boston, MA, USA, 1999.ISBN:0201380366

[6] Li,C.;Mann,S.;Bedi,S.: Error Measurement for Flank Milling, Computer-Aided Design, 37(14), 2005, 1459-1468. DOI:10.1016/j.cad.2005.02.014

[7] Mann,S.;Bedi,S.: Generalization of the imprint method to general surfaces of revolution for NC machining, Computer Aided Design, 34(5), 2002, 373-378.DOI:10.1016/S0010-4485(01)00103-8

[8] Mann,S.;Bedi,S.;Israeli,G.; Zhou, X.: Machine models and tool motions for simulating five-axis machining,Computer-Aided Design, 42, 2010, 231-237.DOI:10.1016/j.cad.2009.11.005

[9] Prenter,P.M.:Splines and Variational Methods, John Wiley & Sons, June 1975.ISBN 0471504025

[10] Roth D.;Bedi S.;Ismail F.; Mann S.: Surfaces swept by a toroidal cutter during 5-axis machining, Computer Aided Design, 33, January 2001,57-63.DOI:10.1016/S0010-4485(00)00063-4

[11] Tutunea-Fatan,O. R.;Feng,H.-Y.: Determination of geometry-based errors for interpolated tool paths in five-axis surface machining, Journal Manufacturing Science and Engineering, 127, 2005,60-67.DOI:10.1115/1.1831285

[12] Wang,L.;Leu,M. C.;Blackmore,D.: Generating swept solids for NC verification using the SEDE method. In SMA '97: Proceedings of the fourth ACM symposium on Solid modeling and applications, New York, USA, ACM Press, 1997, 364-375.DOI:10.1145/267734.267819