

Computer-aided manufacturing and focused ion beam technology enable machining of complex micro- and nano-structures

Frank Niessen  and Mitchell J B Nancarrow

Electron Microscopy Centre, University of Wollongong, Wollongong, New South Wales 2500, Australia

E-mail: contact@fniessen.com

Received 18 April 2019, revised 15 June 2019

Accepted for publication 16 July 2019

Published 8 August 2019



Abstract

We present a novel framework for the fabrication of geometrically complex structures at the micro- and nano-scale which relies on the synergy of integrated computer-aided design and manufacturing systems (CAD/CAM) and focused ion beam (FIB) technology in a scanning electron microscope. Here we utilise industry standard G-code syntax, for the first time, to FIB machining by designing geometries with CAD, defining machining strategies and exporting G-codes with CAM and generating a coordinate list-based beam path by using a custom-built interpreter program. This allows the fabrication of complex structures from CAD models using syntax which is readily understood in the general fabrication industry. The use of G-code allows optimization of the beam path towards a reduction of beam blanking operations and tracing of contours, leading to minimized re-deposition of material. We give a detailed description of the method, use an application example to demonstrate advantages and prospects of the approach and provide the free and open-source interpreter program *CAM2FIB* for application of this method. We contrast and compare various existing available milling strategies and demonstrate the versatility of G-code based programming.

Keywords: focused ion beam, computer-aided manufacturing, electron microscopy, patterning, nano-fabrication, sample preparation

(Some figures may appear in colour only in the online journal)

1. Introduction

Manufacturing of geometrically complex structures at the micro- and nano-scale is an important driver for new developments in science. Some examples are the design of functional surfaces [1], high aspect ratio honeycomb structures for use as drug-delivery patches [2], perforation of graphene membranes for ultrafiltration [3] and the design of nanoelectromechanical systems [4]. In comparison to different nanolithography techniques, machining with a focused ion beam (FIB) in a scanning electron microscope (SEM) stands out by the simple procedure, low preparation time and compatibility with a high range of materials [5]. Up to now FIB machining has been conventionally performed via a rastered approximation of primitive two-dimensional (2D) projection patterns. Patterning of complex geometries with projections is

tedious, difficult to reproduce, limited in spatial resolution and allows little variability of the machining strategy, i.e. the choice in beam path and dwell time. The machining strategy has significant impact on the appearance and dimensional accuracy of the machined structure [6].

Nowadays, most instruments allow patterning of more complex geometries according to patterns that are defined in monochromatic bitmap files. The pixels of such image are scanned horizontally, line by line, where each pixel intensity in the range of 0–255 serves as a scaling factor for the beam dwell time. The more refined definition of the geometry and local dwell time of patterning of bitmaps has been shown to produce complex and continuous geometries successfully [7–9]. A major drawback of bitmap-based patterning however is the limitation to a horizontal, line by line, scanning strategy, which leads to an excessive amount of beam blanking

operations when machining discontinuous geometries. This causes inaccurate beam placement at start and end points for very finely detailed structures. Further drawbacks of the set beam path are no option of tracing contours to avoid re-deposition of material and the limitation to single layer patterns [10].

The latest generation of FIB/SEM instruments allows patterning along arbitrary beam paths, defined by stream files. Stream files are *ASCII*-type files consisting of a list of beam positions, dwell times and flags for beam blanking. During execution, the coordinates and dwell times in a stream file are sent line by line to the digital-analogue-converter (DAC), which controls the positioning of the FIB by varying the voltage applied to the scanning optics [10]. Compared to bitmap-based machining, this approach allows full freedom in defining the beam path and dwell times at the cost of a more complicated procedure for pattern generation. Stream file-based patterning has been demonstrated for manufacturing of simple cone and hole geometries [8, 11] and geometrically more complex 2D patterns [12]. In these cases stream files seem to have been created manually or by mathematical formulae.

An obvious step forward in this respect was the derivation of geometries from CAD models, demonstrated in [9, 10, 13, 14] to allow easier, more advanced and more reproducible design of micro- and nano-components. Velkova *et al* [9] and Lalev *et al* [10] created stream files based on coordinates in stereolithography (*STL*) format files, which can commonly be exported from components of a CAD model. While this approach constitutes an improvement compared to bitmap-based machining, it is limited to the same horizontal scanning strategy and does thus not employ the potential of stream files of customizing the beam path. This shortcoming was addressed in recent work by De Felicis *et al* [13], who imported coordinates from a mesh of a 3D component of a CAD model, calculated dwell times by a node down-sampling approach and applied a custom scanning strategy aimed at minimizing beam blanking operations and tracing the contours of the final geometry. They demonstrated that the derivation of geometries from CAD and optimization of the FIB machining strategy are a promising path towards machining complex geometries at high spatial resolution. As the beam path was obtained from a custom-built, non-released, algorithm with a fixed scanning strategy, it however lacks applicability and adaptability to a large variety of geometries.

For several decades, computer-aided manufacturing (CAM) has been applied to derive information from a CAD model to create instructions, so called G-codes, for the control of the path of an automated cutting tool. G-code is the predominant industry standard for numerical control (NC) of several milling, turning and cutting techniques [15]. Recently, G-code extensions to overcome conventional layer-based techniques in additive manufacturing have been developed [16–18]. Currently CAD and CAM are available as integrated software solutions, in which one can directly apply and combine several machining strategies to fabricate a designed component. Strategies vary from simple parallel machining to

complex strategies such as pocket or adaptive machining. Since CAD/CAM for CNC machining is a huge market, complex machining strategies have been refined and optimized over decades and new machining strategies are still being developed [19–21].

These clear incentives led us to develop a framework for the application of CAD/CAM-generated machining strategies to the field of FIB machining. We demonstrate a unique workflow consisting of CAD of structures for FIB machining, assignment of machining strategies to regions of the CAD model, export and interpretation of G-codes as well as coordinate interpolation and scaling for the final export to stream files. The method only depends on a CAD/CAM application and a freely available interpreter program, which was developed in *MATLAB* [22]. Since the scanning strategy, i.e. the definition of the beam path, is defined in the CAD/CAM software, the workflow benefits from an easy to use graphical user interface and a large degree of versatility. The interpreter program requires the input of a G-code file, which is exported from the CAD/CAM application. Since G-code is standardized [23], the interpreter program can be combined with any kind of CAD/CAM application.

Thus, we apply a well-known syntax in the form of G-code whereby variables such as feed rate/dwell time, tool diameter/beam width and tool up and down/beam blanking are now usable in an innately understandable FIB context. This ties succinctly the generation of complex parts through standard CAD packages as well as the interpretation of these geometries into a viable beam steering program. We also demonstrate the possibility of using these methods in an additive manufacturing context.

2. Method

The developed method is described in a general matter and may be applied in combination with the majority of available CAD/CAM applications. The CAD/CAM part of the overall method has been validated using *Autodesk Inventor Professional* [24], *Autodesk Fusion 360* [25] and *FreeCAD* [26].

The description of the method is supported by showing the steps that were involved in creating the geometries and machining instructions for the application example in section 3. The application example consists of machining a complex shaped structure of $2.16 \times 2.28 \mu\text{m}$ according to an orthogonally projected geometry.

2.1. CAD of structures for FIB machining

2.1.1. Scaling. Considering the conventional length scales at which FIB is applied for machining structures, the dimensioning of geometries in CAD should, in theory, be done in nanometres. In practice, this proves to be inconvenient for the following reasons: (1) The standard length units of most CAD applications are either inch or millimetre. Dimensioning structures in nanometre could thus exceed the resolution limit of the application. (2) The view settings of most CAD applications are optimized towards

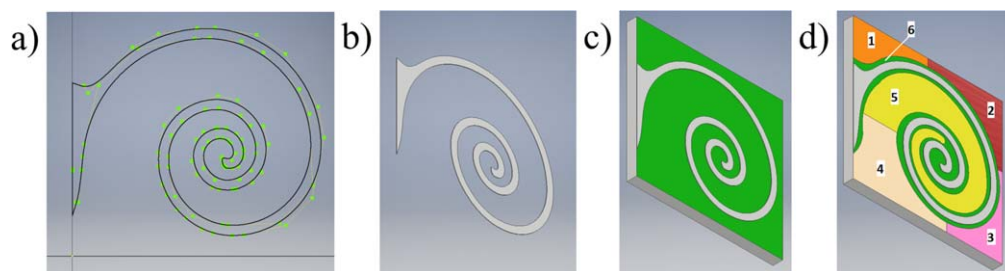


Figure 1. CAD steps towards application of CAM for FIB machining demonstrated on the geometry of a Fibonacci spiral with final dimensions of $1.83 \times 2.22 \mu\text{m}$: (a) creation of a fully constrained 2D sketch ($1.83 \times 2.22 \text{ m}$); (b) 6 mm extruded body based on the profile defined in the 2D sketch; (c) –150 mm extrusion of an attached bottom plate ($2.16 \times 2.28 \text{ m}$) required for assignment of pocket milling strategies; (d) example of defining individual boundary patches (coloured regions 1–6) for local assignment of different machining strategies.

use of these standard length units, which may render dimensioning in nanometres impractical. (3) As in the current workflow the machining strategies in the CAM module are adapted from CNC milling, the ratio between the dimensions of the designed component and a standard milling tool should resemble the ratio between the dimensions of the final structure and the beam width.

Realising that the nanometre in standard FIB machining operations scales approximately with the millimetre in standard CNC milling operations, all lengths are dimensioned in millimetre. The exported G-codes adapt the millimetre convention and the interpreter program converts all lengths from millimetre to nanometre.

2.1.2. Designing the structure. In CAD, most designs of a solid body start with a 2D sketch of its outline (figure 1(a)). Here we used a Fibonacci spiral template as a test model, as the gradual reduction in radius is advantageous to test the interpreter's ability to modulate curves, as well as the gradual, predictable narrowing of the gap between the desired features allows re-deposition of material characteristics and the performance of beam blanking operations in non-continuous structures to be evaluated. As the present implementation is focused on orthogonally projected geometries, the design of a component simply consists of extruding a thin outline to a defined extrusion length (figure 1(b)). The value of the extrusion length is not of great importance, but should for convenience be a small value compared to the length and width of the structure. In order to assign operations for machining the designed component in figures 1(b), a machining stock needs to be created. Here, a bottom plate is created by defining a second sketch of a sufficiently large rectangle and extruding the profile away from the main body (green body in figure 1(c)). The outer edges of the bottom plate are then defined as the machining stock so that machining operations can be assigned to the green surface area of the plate. This approach can be taken a step further by defining multiple areas for machining with individual machining strategies as shown in figure 1(d). This can be achieved by creating additional 2D sketches on the bottom plate and defining these areas as boundary patches, which

renders them selectable as individual areas for machining in the CAM module.

2.1.3. Derivation of CAD geometries from monochromatic bitmap files. Switching from bitmap- to stream file-based FIB machining does generally not require redesign of existing geometries. Most CAD applications support the import of geometries from vector-based image formats such as SVG or DXF into 2D sketches. Transfer of existing geometries from bitmaps into the CAD application therefore only requires the transformation of a bitmap to a vector-based image format. Several applications are available to conduct such transformation. The 2D sketch in figure 1(a) was derived from a monochromatic bitmap, which was transformed into a vector graphic object by using the *Trace bitmap* feature in the open-source vector graphics editor *Inkscape* [27]. The feature supports bitmap tracing by various criteria such as brightness cut off, edge detection or colour quantization and works especially well with bitmaps that show geometries with distinct contours.

2.2. Definition of the beam path

Once the 3D model of a structure has been designed in CAD, machining strategies can be assigned to individual regions. All here presented machining strategies are originally designed for CNC milling and were adapted to FIB machining. We therefore adapt common terms, such as *tool path* in CNC machining, to their FIB-equivalent, *beam path*. Some examples of different machining strategies are introduced in the application example in section 3. The present section gives an overview of some general conventions and parameters that need adjustment for applying CAM to FIB machining.

2.2.1. Assignment of machining strategies. Different CAD/CAM applications have different implemented machining strategies and ways of assigning these. It is commonly required to indicate the stock boundaries and the machining region for each selected machining strategy. Machining strategies typically allow the definition of an offset towards the contour of a structure without explicitly defining dedicated sub-regions. This feature is useful when finishing

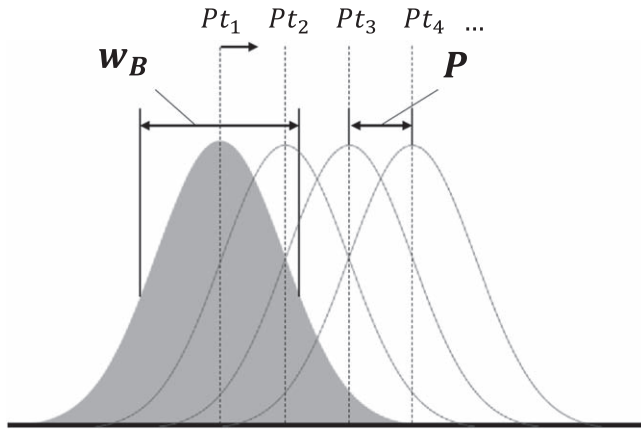


Figure 2. Illustration of the beam width, w_B , and the beam pitch, P . The bell curves depict the beam current distribution of the ion beam at consecutively scanned points Pt_1 to Pt_4 . The beam width is here shown as the full width at half the intensity maximum and the beam pitch as the intervals between the scanned points. The relative overlap, δ , in equation (1) is set by the proportion of these lengths.

of a final contour should be carried out by a dedicated contour machining strategy.

Most CAD/CAM applications include a tool for the visual simulation of the final beam path. This is a useful feature to check the correct definition of the beam path as a final step before exporting the G-code.

2.2.2. Accounting for the beam width and specifying scanning pitch. By default, CAM applications compensate for the diameter of the tool when computing a tool path along the contours of a geometry. Similarly, the ion beam has a known width and geometry which must be defined in order to create beam paths for FIB machining. Here the beam geometry is defined by selecting an end mill tool with a diameter in millimetre that corresponds to the beam width in nanometre. The patterning point coordinates are compensated for the beam width during generation of the beam path. During interpretation of G-codes no beam compensation is applied.

In milling, the distance between the centrelines of two adjacent tool paths is known as the stepover distance. Equivalently in scanning microscopy, the distance between the centrelines of two adjacent beam paths is known as beam pitch. The relation between beam width and pitch is illustrated in figure 2. The respective relations of tool diameter, w_T , to stepover distance, d_s , and beam width, w_B , to beam pitch, P , are equivalent and linked by the relative overlap, δ :

$$\delta = 1 - \frac{d_s}{w_T} = 1 - \frac{P}{w_B}. \quad (1)$$

Controlling the stepover distance in the CAD/CAM application is therefore equivalent to controlling the beam pitch in FIB machining. In simple machining strategies, such as parallel or contour milling, the stepover distance can be set to a constant value, while more advanced machining strategies, such as pocket or adaptive milling, only allow control over the maximum stepover distance.

2.2.3. Adjustment of settings that are specific to milling. A major characteristic of milling, which is fundamentally different to FIB machining, is that the cutting movement is performed by a rotating tool. Since the direction of the cutting motion is different from the feed movement of the tool, their directional relationship finds consideration in the design of tool paths ('climb' versus 'conventional' milling strategies) [15]. This relationship is not relevant for FIB machining and needs to be either adjusted or deactivated.

To ensure a gradual engagement of the tool with the sample in milling, additional path sections such as leads, transitions and ramps are implemented into the tool path. These path sections are of no immediate relevance for FIB machining and should thus be deactivated as far as possible.

2.3. Assembly of G-codes

2.3.1. Supported G-codes and their interpretation. G-code (also referred to as RS274 language [28]) is a NC programming language and is used to control automated machine tools. While it used to be custom to program G-codes manually in an editor, CAM software nowadays generates G-code automatically via a postprocessor. Various G-code implementations are available, most of which are specific to different machine manufacturers. The command syntax consists of the capital letter 'G' followed by a number. The interpreter program, which was created for application of G-codes to FIB machining, was designed to accept the most common G-code implementations. The compatibility of the interpreter was tested with the standardized RS274D [23], WinCNC and the generic FreeCAD implementations.

While a file of G-codes may contain a large number of different commands, the geometry of the tool path is conventionally defined by a combination of only four basic G-codes:

- **G00:** Rapid positioning. This code positions a tool by driving each axis of an NC machine at its maximum speed. The movement is rapid and non-interpolated which requires less computational power, but may not result in movement along a straight line. It is thus by convention only used for non-structure forming movements.
- **G01:** Linear interpolation. This code interpolates a line between two coordinates to enable a linear movement of a tool. In contrast to G00, this command is by convention used for structure forming movements.
- **G02:** Clockwise controlled arc move. This code moves a tool clockwise along an arc, starting at the current position and ending at a given set of coordinates. The centre point of the arc is given by a centre point offset.
- **G03:** Counter-clockwise controlled arc move. This code corresponds to G02 with a counter-clockwise movement of the tool.

In milling, the G00 code is used for rapid movements in which the tool is repositioned without being engaged in the work piece. Equivalently in FIB machining, we interpret G00 as repositioning of the blanked beam. We define G01, analogous to its use in NC machining, as a structure forming

movement. Most machining strategies in CAD/CAM applications allow some control over the relative use of *G00* over *G01*. As an example, the parameter *maximum stay-down distance* indicates the maximum distance at which *G01* should be chosen over *G00* in a non-extrusion movement. In FIB machining, this parameter can be tuned to avoid excessive beam blanking for bridging particularly short distances.

2.3.2. Exporting G-codes. Exporting of G-codes requires a choice of G-code implementation and allows parameterization of some export settings. In our experience with exporting G-codes it was important to allow an extremely large maximum circular radius (arbitrarily set to $>10^6$ m), as some of the more advanced milling strategies work with extremely slight curvatures. It is possible to enforce exporting of the tool path in mm as a length unit, in case the component was designed in a different unit.

In contrast to exporting a mesh or coordinate list for the creation of stream files, G-code stores spatial information in a vector-based approach and thus results in fast processing and a much smaller file size. Some CAD/CAM applications allow smoothing of the tool paths, which replaces excessive amounts of points by fitted arcs within a given tolerance. Activation of this option is recommended as it may lead to further simplification and shortening of the G-code file.

2.4. Interpretation and interpolation of G-code

Interpretation and interpolation of G-codes for the assembly of a stream file is conducted by a custom-built and freely available interpreter program in *MATLAB*. The program, *CAM2FIB* [29], adapted basic features of G-code interpretation and interpolation from the function *G-Code Reader* by Williamson [30].

2.4.1. Structure of the interpreter program. The basic purpose of the interpreter program is to interpolate vector-based G-code files to coordinate-based stream files. More specifically, the G-code file is read in, G-codes are interpreted line by line, the related line or arc segments are interpolated and are appended to the coordinate list which makes up the beam path. The entire beam path may be interpolated a second time before optional transformations are applied. Finally, the beam path is scaled to the resolution of the DAC and exported to a stream file. The basic structure of the interpreter program is depicted in the flowchart in figure 3 and further details on the major processing steps are given in the following subsections.

2.4.2. Interpretation and interpolation of G-codes. After reading in the G-code file, it is interpreted line by line. Potential line numbers (N following by the line number) in the G-code file are ignored by the interpreter. If a tool diameter is written into the header of the G-code file, which is common in CNC milling, it is automatically identified by the interpreter program. This value in millimetre is interpreted as the beam width in nanometre and serves, together with the relative overlap, to determine the beam pitch. The beam pitch

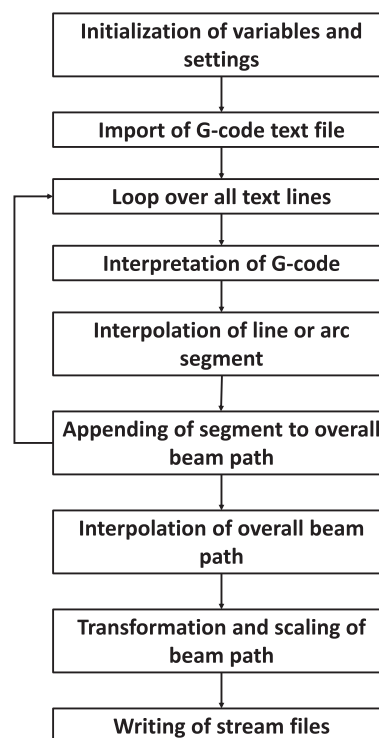


Figure 3. Flowchart describing the structure of the interpreter program to convert G-code files to stream files.

is equal to the interpolated point spacing. All G-codes other than *G00*–*G03* (see section 2.3.1) are ignored by the interpreter. Such other G-codes may indicate tool properties, coolant choices or coordinate conventions, which are not immediately relevant to the present application.

During line by line processing of the G-code file a valid G-code type is assigned to a variable which tracks the current interpolation mode. All following coordinates are interpolated according to this particular mode until a new G-code is recognized. Commands *G01*–*G03* are interpolated to beam path segments according to standard linear and arc-interpolation algorithms with the initially determined interpolated point spacing. If the interpolated length of a segment is shorter than the interpolated point spacing, only the start and end point of that segment are added to the beam path. Each interpolated segment is appended to an initially empty list of x - and y -coordinates that makes up the beam path. Coordinates of mode *G00*, i.e. repositioning of the blanked beam, do not require interpolation of intermediate points. In this case only the final position is appended to the beam path list. Beam blanking is controlled by an initially all-ones Boolean vector with the same length as the beam path list. When a *G00* coordinate is interpolated the value at this point is changed from one to zero. The interpretation of G-codes according to this procedure is repeated line by line until the end of the G-code file is reached.

2.4.3. Interpolation of the assembled beam path. After all G-code segments have been interpolated and assembled to a beam path, it is not mandatory, but recommended, to carry out an interpolation of the entire beam path. When interpolating

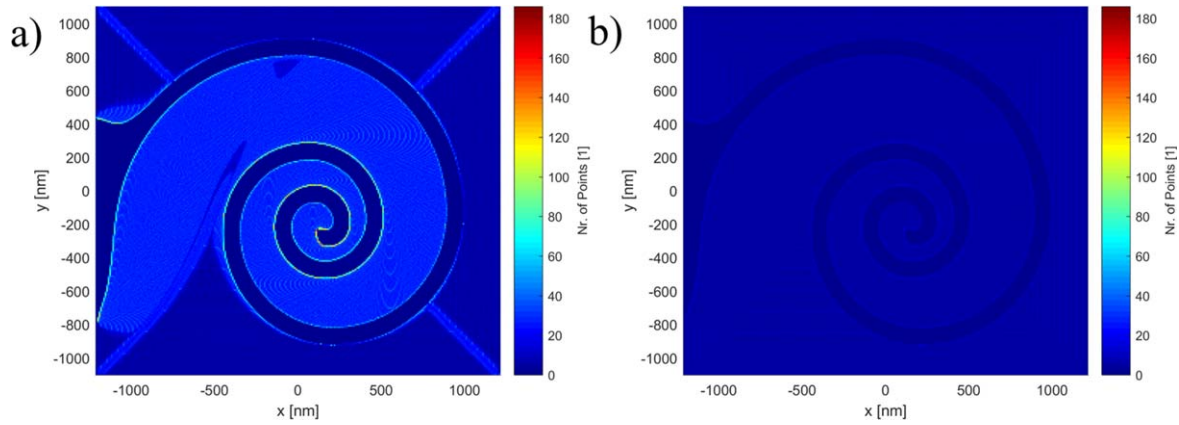


Figure 4. Point density surface plots of the adaptive machining strategy (see figure 5(c)): (a) before and (b) after interpolation of the entire beam path. The colour bar represents the density of scanning points on a grid of 267×267 bins. Before cleaning, a high density of scanning points is found at areas of changing feed direction. Interpolation of the entire beam path largely removes this unwanted effect.

the individual segments during interpretation of G-codes, small remainders of interpolations lead to local increases in the number of patterning points at the end of segments. This is most significant when changing feed direction at the contour of a structure. In these regions the postprocessor of the CAM module commonly computes a combination of short line and arc segments to ensure a smooth change in feed direction of the cutting tool. Saving start and end points of segments that are shorter than the interpolated point spacing leads to an overly fine discretization in these regions. During FIB machining this would lead to the unwanted effect of a locally increased beam dose and sputtering yield, when considering constant dwell times.

To resolve this issue, the entire beam path is interpolated with a linear interpolation algorithm according to the prescribed interpolated point spacing. If required, the point spacing can at this stage be further decreased by a binning factor. The interpreter program plots 3D contour histograms of the point density within discretized 2D bins. Comparison of this plot before and after interpolation of the beam path is useful to check the effect of this interpolation on the homogeneity of the patterning point distribution. An example of such plots to visualize the effect of interpolating the beam path is given in figure 4.

2.4.4. Scaling and transformation of the beam path. After having created a beam path defined by x - and y -coordinates of the final geometry and a Boolean vector for control of beam blanking operations, the path is scaled to the scanning resolution of the DAC for a given magnification and horizontal field of view. Subsequently, the transformations linear scaling, rotation and mirroring may be applied.

The spatial calibration of the DAC is carried out by measuring the horizontal field width, d_{HFW} , for a series of magnifications, x_M . The calibration factor a is defined as:

$$a = \frac{n_x}{m} \sum_{i=1}^m (d_{\text{HFW},i} \times x_{M,i})^{-1}, \quad (2)$$

where n_x is the number of addressable points of the DAC in the x -direction and, m , the number of measurements of

horizontal field widths and respective magnifications. The conversion from spatial coordinates (x, y) to DAC coordinates (i, j) at a given magnification is then achieved by:

$$(i, j) = a \times x_M \times (x, y). \quad (3)$$

In the case of the here used 16 bit DAC, n_x was $2^{16} = 65\,536$ points. Calibration on the horizontal field width at several magnifications in the range of 500–150 000 led to a calibration factor a of $316\,181 \pm 524 \frac{1}{m}$.

Optionally, transformations can be applied to the beam path. Rotation and mirroring allow for correction of the pattern in cases where the G-code file was created with respect to a different coordinate system in the CAD application. Linear scaling may be used to increase the pattern size and interpolation resolution by a common scaling factor.

The dwell time for all patterning points is determined by the ratio of overall patterning time and number of patterning points in the interpreter program.

3. Application example

The application example emphasizes the feasibility of the CAM/FIB approach and the effect of applying different machining strategies, i.e. beam paths, to a geometrically complex structure.

3.1. Structure geometry and material

CAD/CAM was applied in conjunction with the *CAM2FIB* [29] interpreter program to create four different beam paths, following distinct machining strategies, to machine the structure designed in figure 1. The structure follows the outline of a Fibonacci spiral and was machined as an orthogonal projection on a silicon wafer with the crystallographic direction $[1\ 1\ 0]$ in parallel alignment with the surface normal. Silicon is a popular material for nano-structuring [11] and is commonly used as a reference material [13].

3.2. Experimental setup

The work was carried out on an FEI Helios NanoLab G3 CX. The FIB/SEM instrument has a patterning-imaging-acquisition system with a DAC resolution of 16 bit. The addressable range is 65 536 points in x -direction and 56 576 points in y -direction. The acceleration voltage and current of the ion beam during patterning were 30 kV and 28 pA, respectively. The width of the ion beam during patterning was derived from the microscope calibration for silicon and was 13 nm. Patterning was carried out at a horizontal field width of 21 μm .

3.3. Pattern generation

Four machining strategies were applied to the entire machining stock (green area in figure 1(c)). In the CAM module, the tool diameter was set to 13 nm, corresponding to a beam width of 13 nm. Likewise, the stepover distance was set to 4 nm to correspond to a beam pitch of 4 nm. This resulted in a relative beam overlap of 69% (see equation (1) and figure 2). G-code was exported as RS-274D format to four different G-code files¹. The linear interpolation distance in the interpreter program was set identical to the stepover distance of 4 nm. The pattern area of $2.16 \times 2.28 \mu\text{m}$ was resolved to 7148×6753 patterning points. All patterns were given an overall machining time of 100 s which was chosen over using fixed dwell times in order to prevent issues with drift for more complex (i.e. more points) paths. With, on average, 35 000 points per pattern the dwell time per point amounted to an average of 3 ms.

3.4. Application of different machining strategies

With $2.16 \times 2.28 \mu\text{m}$ in total area and down to 90 nm thin channels, machining of the Fibonacci spiral structure poses a challenging application example. The main two challenges in machining the structure are:

1. The structure is discontinuous and complex in geometry.
2. The structure contains wide sections as well as narrow channels.

The associated requirements for a suitable machining strategy to address these challenges are:

1. Tracing of the complex contour of the structure, leading to a significant reduction in beam blanking operations.
2. Finishing of narrow sections after clearing of larger sections to improve the sputtering yield and reduce re-deposition of material in these areas.

We chose four machining strategies based on their different ability to fulfil these requirements. The machining strategies are visualized in contour plots in figure 5. The horizontal and vertical axes indicate the spatial coordinates of

patterns and the colour scale depicts the patterning order of points. The white dashed lines indicate movement of the blanked beam to a different region for machining.

In the following, the applied machining strategies are introduced and discussed based on their ability to successfully machine the Fibonacci structures. The machined structures are presented in figure 6 and are overlayed with the outline of half of the pattern for visual inspection of dimensional accuracy.

3.4.1. Parallel machining with simple ordering. The most basic machining strategy is parallel machining. With the option *simple ordering* activated, the beam path consists of an array of consecutively scanned horizontal lines. This machining strategy is most similar to horizontal scanning in bitmap-based FIB machining. The beam path is shown in figure 5(a). The prescribed horizontal patterning direction and line order implies an excessive amount of beam blanking operations and no tracing of contours. Further, no distinction in the machining order of large and thin sections is made.

Figure 6(a) shows that the top down feed direction led to significant re-deposition of material on the machined structure in the opposite direction. The narrowest channels in the centre of the spiral were poorly resolved.

3.4.2. Parallel machining with optimized ordering. With the option *simple ordering* deactivated, the beam path is computed with the objective of minimizing the overall length of the beam path. In contrast to figure 5(a), the advanced line ordering leads to a separation of the beam path into sub-regions (figure 5(b)). Compared to the simply ordered beam path in figure 5(a), the advanced ordering achieves a drastic reduction in beam blanking operations from 376 to 14.

Compared to the structure which was machined with the option *simple ordering* activated (figure 6(a)), the parallel machining strategy with advanced ordering led to somewhat less re-deposition of material on the structure (figure 6(b)). Clearing of the large section in the upper-right corner of the pattern as a last step led to significant re-deposition of material in this area. Since the centre of the spiral is machined as the last section in its immediate surrounding, the fine channels of the spiral were resolved more accurately compared to the parallel machining strategy with simple ordering.

3.4.3. Adaptive machining strategy. The adaptive machining strategy was developed to compute a tool path with continuous changes in feed direction to decrease wear of milling tools. While this objective is irrelevant for machining with FIB, the adaptive machining strategy creates a smooth beam path that is adapted to the complex geometry of the Fibonacci spiral (figure 5(c)). The strategy only applies four beam blanking operations and, for the present geometry, clears larger areas before machining narrow sections.

Figure 6(c)) shows the machined structure. Compared to the structures that were machined by parallel machining

¹ All applied G-code files and generated stream files are provided in the supplementary material available online at stacks.iop.org/NANO/30/435301/mmedia.

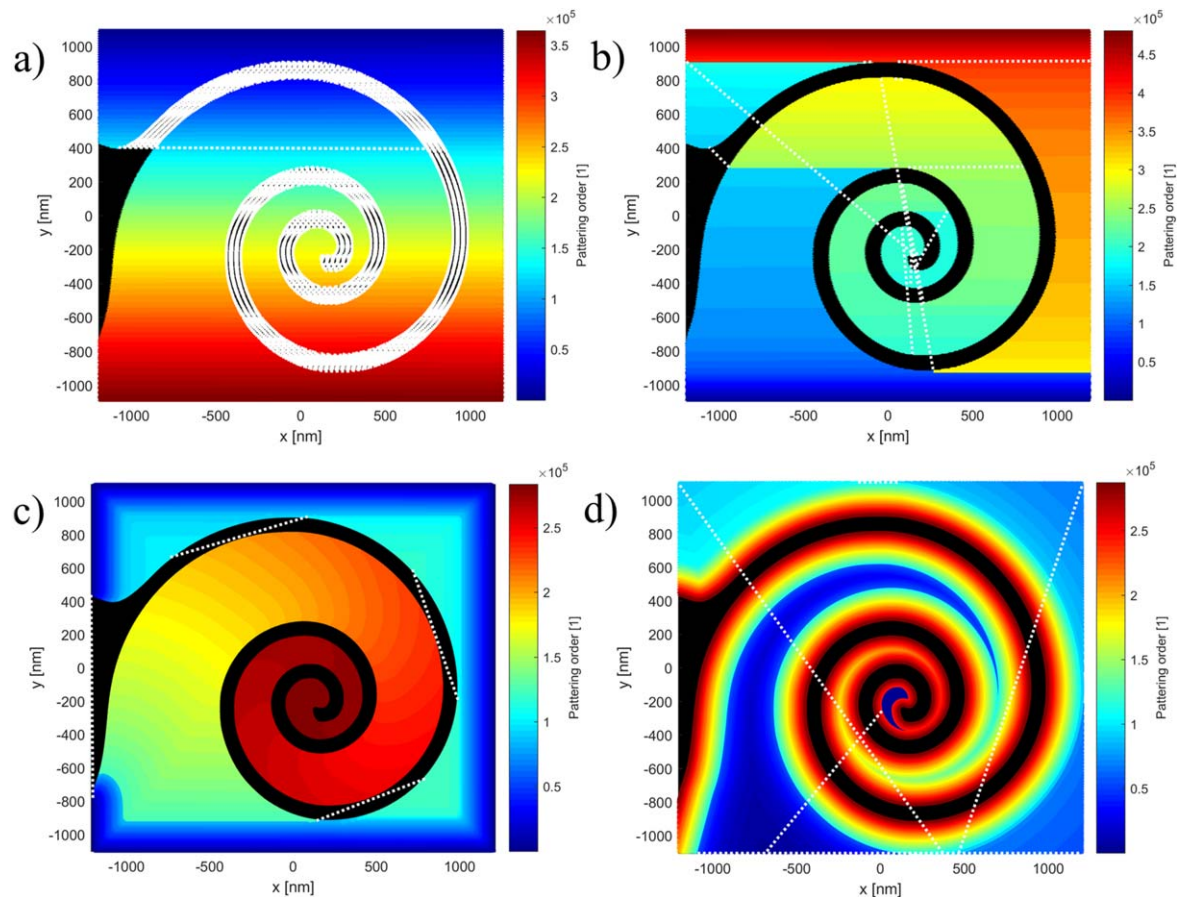


Figure 5. Visual representation of different machining strategies. The horizontal and vertical axes indicate the spatial coordinates of patterns and the colour scale shows the order of scanned points. The white broken lines indicate repositioning of the blanked beam. The machining strategies are: (a) parallel machining with simple ordering (similar to bitmap-based patterning), (b) parallel machining with optimized ordering, (c) adaptive machining and (d) pocket machining.

strategies (figures 6(a) and (b)), it is evident that clearing large sections before machining of smaller sections led to a better resolution of the outer part of the Fibonacci spiral. After about the first revolution of the spiral, the feed direction of machining led to re-deposition of material on already finished sections.

3.4.4. Pocket machining strategy. The pocket machining strategy creates beam paths that approach the contours of a structure with a parallel feed direction. The strategy starts by removing the bulk of the material and finishes the structure by tracing the contours (figure 5(d)). It uses only five beam blanking operations to remove the bulk of material in different regions and then machines the narrower sections with the final tracing of contours. This last step ensures that re-deposition of material on the structure is minimal.

The machined structure is shown in figure 6(d). Compared to the previous machining strategies, the pocket machining strategy led to a remarkably accurate machining of the Fibonacci spiral. While the unavoidable re-deposition of the material certainly occurred, the pocket machining strategy controlled the re-deposition to ensure that the contour of the structure remained intact. Further, the reduction in beam blanking and rapid changes of direction led to much more precise beam placement during machining operations.

The final structure therefore has well-defined edges and good dimensional accuracy.

4. Discussion

4.1. Qualification of CAM and G-code for FIB machining

CAM in combination with G-code as a NC language has successfully made its way to an industry standard for numerous milling, turning, cutting and additive manufacturing techniques [15–18, 31]. The success of this method is owed to the automated code generation by a postprocessor in CAM and the simplicity and versatility of the G-code command structure, allowing full and precise control of arbitrary processing parameters and multiple axes.

Conventional geometrical patterns, bitmap-based patterns and the CAD-based approach in [9, 10] represent patterning techniques that operate with a fixed parallel machining strategy, similar to the one in figure 5(a). Work by De Felicis *et al* [13] and the present work (see figure 6) demonstrate that control over material re-deposition is crucial for obtaining dimensional accuracy and that this control is obtainable by employing optimized machining strategies. The G-code command structure constitutes the ideal platform for the further

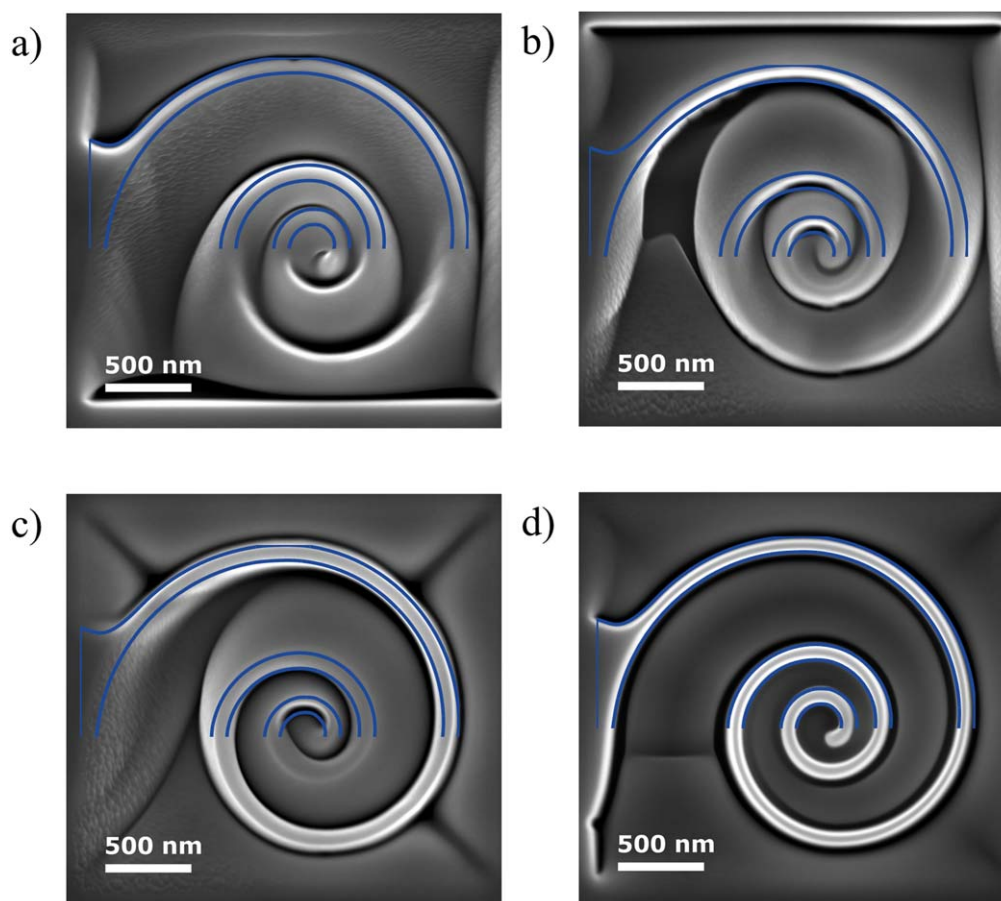


Figure 6. Structures created from patterning according to the machining strategies in figure 5 on a silicon wafer. The blue lines show the outlines of half of the pattern for visual inspection of dimensional accuracy. (a) Parallel machining strategy with simple ordering (similar to bitmap-based patterning), (b) parallel machining strategy with advanced ordering, (c) adaptive machining strategy and (d) pocket machining strategy.

development of FIB machining at complex geometries and advanced machining strategies. While offering customizability and control over the beam path, it is fully downward compatible—all lower level machining strategies and input formats, such as bitmaps and *STL* files, are easily convertible to G-code.

4.2. Future perspective of CAM/FIB with a G-code framework

While the present work is a proof of concept, the combination of CAM and FIB with a G-code framework bears great potential to significantly extend the functionalities of existing FIB/SEM systems. In the following, we outline some possible developments and potentials.

4.2.1. FIB/SEM as a hybrid system. Figure 7 shows three 3D-structures which were additively manufactured by electron beam Pt-deposition. A 3D model of the structure was designed in a CAD application (see inset with isometric projection in figure 7) and exported as an *STL* file. The geometry was then translated to G-code in the 3D-print software *Ultimaker Cura* [32] and converted to a stream file with the *CAM2FIB* [29] interpreter program. While no substantial effort was made to optimize the deposition process towards better dimensional accuracy, we consider

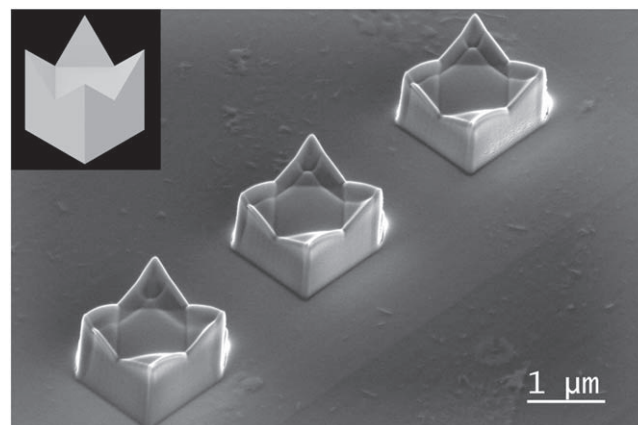


Figure 7. Proof of concept for applying the CAM/FIB workflow to additive manufacturing of 3D-structures with electron beam Pt-deposition. The structures, shown at 45° stage tilt, were manufactured by patterning with an electron beam at 0.5 kV acceleration voltage and 0.69 nA beam current.

the reproducibility and detail of the deposited 3D-structures as encouraging for future applications.

On a macroscopic manufacturing level, G-code is the standard NC language for both additive and subtractive manufacturing techniques. Combining both additive and

subtractive processing methods in one instrument and being able to integrate sub-components by insertion with a micro-manipulator, FIB/SEM instruments are by definition hybrid systems. While advanced access to such hybrid capabilities can be obtained via manufacturer extensions [33, 34], a G-code based framework would grant full control over these individual capabilities and unleash the full potential of their coordinated interaction.

4.2.2. Online interpretation of FIB/SEM-fitted G-code. In this work, G-code intended for online interpretation on CNC milling machines was interpreted offline in the *CAM2FIB* program and converted to stream files as an input to the FIB/SEM instrument. Currently there is no G-code implementation that is fitted towards FIB/SEM instruments and no option of interpreting G-code online. Implementation of an online G-code framework installed on the FIB/SEM instrument would bear great potential.

A G-code dialect tailored towards application on FIB/SEM instruments would contain the basic commands *G00* to *G03* for describing beam path geometries. Similar to the specification of an extrusion length per interpolated length segment in 3D-printing, commands for specifying the dwell time per interpolated length segment would ensure optimal beam control in FIB/SEM patterning. Advanced commands could enable switching between milling and deposition modes and specifying beam parameters. As an advantage over offline interpretation, online interpretation of G-code could be instrument-specific. In this way G-code files could be shared and used between different instruments and research groups without a need for instrument-dependent adaptation.

4.2.3. 5-axis FIB machining. Similar to 5-axis CNC milling [35], laser milling [36] and additive manufacturing [16, 18], control over the rotational axes of an SEM-stage would enable 5-axis FIB machining to create freeform surfaces and undercuts. While the positioning accuracy and stability of conventional SEM-stages may not be optimal, control over a high-precision substage would render 5-axis FIB machining feasible. The introduction of such machining technique would require seamless interplay between the beam and stage control. A G-code based framework would constitute the ideal prerequisite for any development towards such technique.

4.2.4. Customized machining strategies for FIB. The here applied machining strategies were all derived from CNC milling strategies. CAM modules generally support manufacturing processes such as multi-axis milling, turning, drilling and cutting. With a rising market for machining of micro- and nano-structures, support for FIB/SEM instruments in CAM applications could become an attractive area of development. As demonstrated by De Felicis *et al* [13], machining strategies optimized towards FIB machining would respect material re-deposition behaviour and access material databases to determine material dependent parameters like the sputter yield. Based on the dimensions of a structure, the CAM module would determine optimal beam parameters and

dwell times for machining. Similar to the tool-selection in CNC milling, different beam currents could be applied for clearing and finishing operations.

Support of FIB machining in CAD/CAM software would provide a seamless solution to online interpretation of specialised G-code on FIB/SEM instruments. As discussed on section 4.2.1, an ideal support of FIB/SEM in a CAM module would make use of the hybrid capabilities of a FIB/SEM instrument.

5. Conclusion

We presented a novel framework for manufacturing of micro- and nano-structures with FIB machining by using modern CAD/CAM technology for the generation of intelligent beam paths. In this proof of concept, beam paths were exported from the CAD/CAM application in the vector-based G-code format and converted to a coordinate-based beam path with the custom-built and freely available interpreter program *CAM2FIB*. We disclosed the framework in detail and presented the workflow from design to manufacturing of a complex structure with help of an application example. It was found that the machining strategy had a great effect on the appearance and dimensional accuracy of the machined structure and that purpose-fit machining strategies clearly outperformed conventional parallel machining strategies. Finally, the future perspective of CAM/FIB with a G-code framework was discussed. Judging from this proof of concept work, we predict that the implementation of an online G-code based framework on FIB/SEM instruments will be essential for the development of hybrid, i.e. additive and subtractive, machining as well as 5-axis FIB machining.

Acknowledgments

F Niessen acknowledges the Danish Council for Independent Research (Grant: DFF-8027-00009B) for financial support. This research used the FEI NanoLab G3 CX funded by the Australian Research Council (ARC)—Linkage, Infrastructure, Equipment and Facilities (LIEF) Grant (LE160100063) located at the UOW Electron Microscopy Centre.

ORCID iDs

Frank Niessen  <https://orcid.org/0000-0001-5849-710X>

References

- [1] Liu Y, Shen J, Wang X, Zheng X, Sun J, Cooper M A, Wu D, Yang Z, Wu J and Jiang C 2018 Biomimetic Moth-eye nanofabrication: enhanced antireflection with superior self-cleaning characteristic *Sci. Rep.* **8** 1–10
- [2] Jeong G S, No D Y, Lee J, Yoon J, Chung S and Lee S H 2016 Viscoelastic lithography for fabricating self-organizing soft micro-honeycomb structures with ultra-high aspect ratios *Nat. Commun.* **7** 1–9

- [3] Choi K, Droudian A, Wyss R M, Schlichting K-P and Park H G 2018 Multifunctional wafer-scale graphene membranes for fast ultrafiltration and high permeation gas separation *Sci. Adv.* **4** 0476
- [4] Craighead H G 2000 Nanoelectromechanical systems *Science* **290** 1532–5
- [5] Reyntjens S and Puers R 2001 A review of focused ion beam applications in microsystem technology *J. Micromech. Microeng.* **11** 287–300
- [6] Wilhelmi O, Reyntjens S, Mitterbauer C, Roussel L, Stokes D J and Hubert D H W 2008 Rapid prototyping of nanostructured materials with a focused ion beam *Jpn. J. Appl. Phys.* **47** 5010–4
- [7] Gorelick S and De Marco A 2018 Fabrication of glass microlenses using focused Xe beam *Opt. Express* **26** 13647–55
- [8] Ageev O A, Kolomiitsev A S, Bykov A V, Smirnov V A and Kots I N 2015 Fabrication of advanced probes for atomic force microscopy using focused ion beam *Microelectron. Reliab.* **55** 2131–4
- [9] Velkova V, Lalev G, Hirshy H, Omar F, Scholz S, Minev E and Dimov S 2011 Process chain for serial manufacture of 3D micro- and nano-scale structures *CIRP J. Manuf. Sci. Technol.* **4** 340–6
- [10] Lalev G, Dimov S, Kettle J, Van Delft F and Minev R 2008 Data preparation for focused ion beam machining of complex three-dimensional structures *Proc. Inst. Mech. Eng. B* **222** 67–76
- [11] Hopman W C L, Ay F, Hu W, Gadgil V J, Kuipers L, Pollnau M and Ridder R M D 2007 Focused ion beam scan routine, dwell time and dose optimizations for submicrometre period planar photonic crystal components and stamps in silicon *Nanotechnology* **18** 195305
- [12] Raffa V, Vittorio O, Pensabene V, Menciassi A and Dario P 2008 FIB-nanostructured surfaces and investigation of bio/nonbio interactions at the nanoscale *IEEE Trans. Nanobiosci.* **7** 1–10
- [13] De Felicis D, Mughal M Z and Bemporad E 2017 A method to improve the quality of 2.5 dimensional micro-and nano-structures produced by focused ion beam machining *Micron* **101** 8–15
- [14] Eifert A, Mizaikoff B and Kranz C 2015 Advanced fabrication process for combined atomic force-scanning electrochemical microscopy (AFM-SECM) probes *Micron* **68** 27–35
- [15] Autodesk 2014 *Fundamentals of CNC Machining* (San Rafael, CA: Autodesk Inc.)
- [16] Tang Q, Yin S, Zhang Y and Wu J 2018 A tool vector control for laser additive manufacturing in five-axis configuration *Int. J. Adv. Manuf. Technol.* **98** 1671–84
- [17] Bourell D L, Leu M C and Rosen D W 2009 Roadmap for additive manufacturing: identifying the future of freeform processing *Solid Freeform Fabr.*
- [18] Chen Y, Zhou C and Lao J 2011 A layerless additive manufacturing process based on CNC accumulation *Rapid Prototyp. J.* **17** 218–27
- [19] Yang J and Yuen A 2017 An analytical local corner smoothing algorithm for five-axis CNC machining *Int. J. Mach. Tools Manuf.* **123** 22–35
- [20] Otsuki T, Sasahara H and Sato R 2018 Method for generating CNC programs based on block-processing time to improve speed and accuracy of machining curved shapes *Precis. Eng.* **55** 33–41
- [21] Tunc L T 2019 Smart tool path generation for 5-axis ball-end milling of sculptured surfaces using process models *Robot. Comput.—Integr. Manuf.* **56** 212–21
- [22] MATLAB 2016 MATLAB 2016b (9.1.0.441655)
- [23] International Organization for Standardization 2009 Automation systems and integration – Numerical control of machines – Program format and definitions of address words – Part 1: Data format for positioning, line motion and contouring control systems ISO 6983-1:2009 pp 1–26
- [24] Autodesk 2018 Inventor Professional 2019, Build 136
- [25] Autodesk 2018 Fusion 360 2.0.5119
- [26] Riegel J, Mayer W and van Havre Y 2018 FreeCad (Version 0.17) [Software] (<http://freecadweb.org>)
- [27] Inkscape 2018 Inkscape 0.92 [Software] (www.inkscape.org)
- [28] Frederick T, Kramer T, Proctor F and Messina E 2000 *The NIST RS274NGC Interpreter - Version 3* 6556 National Institute of Standards and Technology 1–113
- [29] Niessen F 2019 CAM2FIB [Software] (<https://github.com/frankNiessen/CAM2FIB>)
- [30] Williamson T 2018 G-Code Reader, (<https://au.mathworks.com/matlabcentral/fileexchange/67767-g-code-reader>)
- [31] Milewski J O, Lewis G K, Thoma D J, Keel G I, Nemec R B and Reinert R A 1998 Directed light fabrication of a solid metal hemisphere using 5-axis powder deposition *J. Mater. Process. Technol.* **75** 165–72
- [32] Ultimaker 2019 Ultimaker Cura 3.6.0
- [33] ThermoFisher 2018 DataSheet NanoBuilder Software - Automatic fabrication of complex nanostructures using Thermo Scientific NanoBuilder Software, 1–4
- [34] Zeiss 2017 DataSheet ZEISS Atlas 5, 1–25
- [35] Lasemi A, Xue D and Gu P 2010 Recent development in CNC machining of freeform surfaces: a state-of-the-art review, *CAD Comput. Aided Des.* **42** 641–54
- [36] Cuccolini G, Orazi L and Fortunato A 2013 5 Axes computer aided laser milling *Opt. Lasers Eng.* **51** 749–60