Abstract. Direct computer programming of architectural geometry and of CNC tool pathways can control the fabrication of form and the related treatment of material. When the entire form creation and tool path process is taken on as a design problem, there is potentially a closer link between formal design intentions and their physical realization. This paper describes several case studies that engage computer programming as a first stage in an iterative design process coupled with more explicit control over CNC tool paths. It indirectly critiques the design exploration of geometry where there is only user command control over a CAD system and where the specification of CNC pathways is also less explicit. Examples of different strategies are compared in the same educational context.

Keywords. CNC, geometrical modeling, design, computer programming

Introduction

Traditional design studio model making methods have begun to migrate to the world of computer aided design and CNC, and a brave new world of curvilinear shapes and provocative spatial compositions have gained notice in both the profession and the educational preparation of architects. Yet, this is a shift in means without necessarily a complete re-engagement or discriminating shift in the conceptual framework for design. With some exceptions, the architect has perhaps transferred a visceral sensibility towards traditional modeling media into a similar approach with respect to complex geometries generated on the computer. For example, a CAD NURB surface can be treated as but a sculptural modeling form to be manipulated in a not mathematically explicit way. The resulting geometry can be post-ration­alyzed in terms of the processes needed to effect CNC fabrication. Alternatively, a more explicit set of controls can be maintained consistently throughout the design process.

Methods of Generating Tool Paths

In the examples reported here, students were first taught to explicitly write the G & M codes that generate CNC tool paths. Having gained some basic understanding, they were then free to use either automated G & M code creating software or to continue to write the G & M codes directly. The CNC equipment included a three-axis milling machine, a three axis-router, and a laser cutter. Three different design methods evolved from this setting.

Method 1: using standard CAM software. In a conventional approach, a design project is developed through the process of building a geometrical model with CAD user commands. The model is in turn translated through commercial CAM software into G & M codes. This less explicit and automated approach accommodates a great range of surface geometries. The student explores a number of tool path alternatives by trial and error until arriving at one that most closely approximates the design intentions. Through experience, the student learns to exercise greater...
control over the final product. In Figure 1a, for example, a variable landscape form is developed by a means of a user command CAD approach with final output from a laser cutter. By contrast, in Figure 1b, a second student developed a computer program to test a greater variation in the geometry of this form. The two student projects informed each other. The first project was less pure geometrically but more considered in terms of its fit to site and design purpose. The algorithmic process of the other project was less vested in the particular tectonics of construction. Such methods are more tightly integrated in the additional examples presented below.

Method 2: programming geometry in schematic design. In this case, a computer program generates the architectural forms that helped to explore design ideas. The program harnessed the CAD software (Microstation) as a graphics engine. Mathematical precision over the creation of the initial geometry led to exercising stronger control over how the CNC software would interpret it. In Figures 3a and 3b, some waves are programmatically explored according to location of initial water drops and time after their impact. The geometry is the basis of the CNC generated forms in Figures 4a and 4b used for a prospective building wall system.

Method 3: programming geometry in closer anticipation of tool paths. In the third case, the entire process was more completely subjected to...
direct programmatic control. The architectural
gonetry was programmed and re-programmed
more consciously of the CAM software produced
CNC pathways. This design tests domical sur-
faces intended as landscape forms. This
approach maximized the CNC control in design
projects explored by the students.

Comprehensive method: programming geom-
etry and CNC pathways. A more complete
approach was tested by means of programming
directly the CNC pathways as a part of the total
design scheme. This method evolved later and
will be explored more intensively in the next aca-
demic year. The further programming approach
allows greater control over how the forms are
shaped and how the materials are fabricated. In
Figures 5a and 5b, a hyperbolic parabolic and
several dome like surfaces are generated and
their CNC tool paths illustrated. A corresponding
G & M code file is also simultaneously created.
The red lines represent the rough and the smooth
tool paths. The blue lines represent the “step
over” non-cutting tool paths. Efficiency through
path reduction, co-linearity checking and other
strategies are factored into the computer program
developed in consideration of the desired tool
path treatment of fabricated surfaces.

Without this last example’s level of explicit
control, designers can still explore through trial
and error the CNC processes needed to obtain a

Figures 4a and 4b: Casting
of molds and poured forms
controlled programatically
with more explicit considera-
tion of CNC pathways 5
desired quality of material fabrication (method 1 above). Or, in some cases, the principal designer works out the forms in the manner of traditional media, and a team of more technically capable enablers work to translate and shift these forms painstakingly taught. Today three-dimensional computer visualization technologies may in part supplant the need to use descriptive geometry for illustrating complex forms. However, descriptive geometry is more than a means to visualization, but as the earlier authors of a book on that subject argued, it taught about thinking in three-dimensions (Kenison and Bradley, 1923).

Teaching computer graphics programming and related concepts of mathematical order can stimulate three-dimensional thinking about form. This approach has had currency within some schools of design but has not caught on widely. Still, programming doesn’t necessarily capture every nuance in actual built form, and so a hybrid approach of working by hand and abstractly by computer algorithm may have some advantages in a continuum of design activity. That is, the process of working by hand may still provide some sense of design reasoning that the abstract vehicle of a computer graphics program may not fully engage. Room must be left too for the on-site fabricator to have direct hands-on influence over form as a part of a larger building process. Jim Glymph, Associate Partner at Frank Gehry’s office, refers to such a full participatory process as a “team sport” (Glymph).

Coupling a computer programming approach with CNC has the potential for creating architectural forms characterized by greater depth of geometrical reasoning and greater control over the actual fabrication aesthetic. This aesthetic was possible even in pre-computer design media when descriptive geometry played itself out in the design process right up to the making of forms. The value of knowing architectural geometry intimately is emphasized in the treatises of sixteenth century scholars devoted to the subject, such as Serlio (Hart & Hicks, 1996). In programming computer geometry today, these theories find a new fit, re-discovered application and potential advancement.

In a previous generation, an architecture student used descriptive geometry to visualize their three-dimension designs on paper. Descriptive geometry was widely and at times perhaps

![Figure 5a](image1.png)

![Figure 5b](image2.png)

*Figures 5a and 5b: Hyperbolic parabolic and other dome shapes & tool paths with simple, ridge defined and dome geometry. Tool paths algorithmically fit to geometry and size of CNC material. 6*
References

Books I-V of 'Tutte l'opere d'architettura et prospetiva' by Sebastiano Serlio; translated from the Italian with an introduction and commentary by Vaughan Hart and Peter Hicks. New Haven [Conn.]: Yale University Press, c1996.

Footnotes

1 Two courses were taught concurrently in spring 2003 at UVA. Eric Field took the leading role in teaching a course in CNC fabrication. Earl Mark took at leading role in teaching a course in computer programming coupled with CNC. Advising roles were tightly integrated and a number of the students worked cooperatively on projects.
2 Project by graduate students Gennifer McGill (manual CAD) and Tim Slater (program) in loose collaboration.
3 Program ported by Kash Mehta after an SGI demo algorithm published by Erik Larsen on the Internet, with some later enhancements by Earl Mark.
4 Project by graduate student Stephanie Giles.
5 Programmed by graduate student Kraig Schmidt after initial software by Earl Mark.
6 Developed by Earl Mark. Raw stock material outline is shown in green rectangles.