

G1200101

手続き的な方法による工業製品設計*

金田 泰

Industrial Product Design by Procedural Methods

Yasusi Kanada

Dasyn.com
Nakano-ku Yayoi-cho 4-2-16, Tokyo, 164-0013 Japan

As well as in computer programming, both declarative and procedural methods should be available in industrial product design. However, design for additive manufacturing is mostly based on declarative CAD as well as other areas of product design. This presentation reviews a conventional procedural method for CNC and proposes a procedural design method for 3D printing.

Key Words : Declarative method, Procedural method, Generative design, Additive manufacturing, AM, Computer-aided design, CAD.

1. 緒 言

かつては G コード⁽⁷⁾や次章でのべる APT のような手続き的な言語による制御プログラムを機械加工の設計者が記述していたが、現在ではほとんど常に computer-aided design (CAD) によって静的で宣言的なモデルが記述されるようになっている。G コードはもともと切削加工(現在の用語では subtractive manufacturing)の制御のために開発され、当初は設計者がそのプログラムを記述していた。しかし、現在では設計者は CAD によって宣言的な(静的な)モデルを記述し、それをコンピュータが手続き的な G コードに変換する。

切削加工においては手続き的な設計が宣言的な設計によってすっかりとってかわられたが、付加加工 (additive manufacturing) においては手続き的な設計にも利点があるとかんがえられる。切削加工の手続きは制約が多くて複雑であり、設計には手続き的な記述が適さなかったとかんがえられる。また、G コードのような言語は低水準で抽象化機能がなかったゆえにプログラミングが困難だった。しかし、3D 印刷のような付加加工は切削加工より直観的なので、設計者が抽象化された手続き的記述をすることが場合によっては宣言的な方法とくらべて利点があるとかんがえられる。この報告では手続き的な方法による機械加工と製品設計・製造について考察し、3D 印刷への適用例を記述する。

2. 手続き的な切削加工制御の歴史

コンピュータ数値制御 (CNC) による切削加工の技術とそのための言語 APT はすでに 1940~50 年代に開発されている。数値制御 (NC) の技術は 1942 年に John T. Parsons によって発明されたが、CNC の技術はそれにもとづいて 1950 年代にマサチューセッツ工科大学 (MIT) において開発された⁽⁷⁾。この制御のため、MIT においては APT というプログラミング言語⁽¹⁾⁽⁷⁾⁽²⁾が開発され、切削加工のプログラムを記述するのに使用された。APT は基本的にはアセンブラ風の言語であり手続き的だった。Parsons は NC プログラムの記録にパンチカードを使用した。MIT で紙テープが使用され、各種のサブルーティンが作成されるとともに、APT にはマクロや入れ子定義 (nested definitions) のような原初的な手続き抽象化機構がとりいれられた⁽⁷⁾。1970 年代には APT とデータ抽象化やオブジェクト指向との関係も議論された⁽⁷⁾。

しかし、その後、CAD 技術が開発され、APT のような手続き的な方法によって設計者が指示することはほと

* 正員, Dasyn.com (〒164-0013 東京都中野区弥生町 4-2-16)
E-mail: yasusi@kanadas.com

んどなくなった。設計者はCADによって静的つまり宣言的なモデルを記述し、それをコンピュータが手続き的なプログラムに変換するようになった。そうなったのは、切削加工には手続き的記述が適さない複雑さがあるうえ、GコードやAPTは低水準で抽象化機能がよいため(APTに関してはさまざまな改良がくわえられたが、互換性の問題もあるためそれには限界があり)設計者がプログラムするのは困難だからだとかんがえられる。

3. 手続き的な3D設計・印刷法

手続き的な3D設計・印刷の方法について考察し、それを部分的に実現する方法を開発した。この章ではまず方法論を提案し、実現したプログラミング言語とそれを使用した3D設計・印刷の方法をかたんに紹介する。

3・1 方法論

アートとはちがって、工業製品をつくるには標準化された部品からのくみだてや設計ツール(CAD)の開発をかんがえる必要がある。「工業製品設計」ということばは「インダストリアル・デザイン」ということばに似ているが、後者はアートであり、前者のほうがモジュラリティに配慮しなければならないところがちがう。

手続き的な設計においても部品化のためのしくみが必要だが、それはプログラミングにおいても同様であり、そのためのしくみとしてサブルーティン、関数のような手続き抽象化のためのしくみが用意されている。したがって、基本的には図1に示すようにライブラリ化された部品をサブルーティンとして用意して、それをくみあわせることによって構造化された部品をつくり、さらにそれをくみあわせていけばよい。ソフトウェアにおいて部品を手続き的にくみあわせる基本的な方法は**接続(concatenation)**、**選択(selection)**、**反復(repetition)**の3種類であり、手続き的設計においてもそれが基本だとかんがえられる。ただし、手続き的なCADにおいてはもうすこしくふうが必要であろう。この話題はこの発表の範囲をこえているが、3・3においてすこしだけふれる。

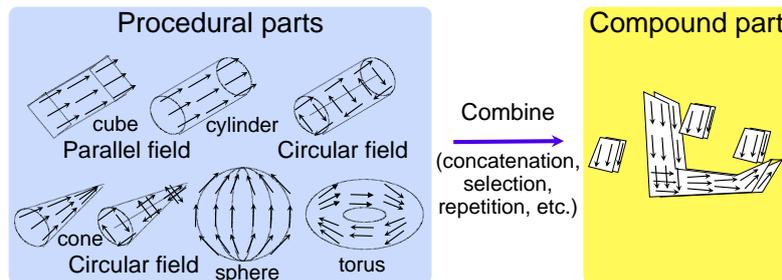


図1 部品のくみあわせによる手続き的な設計

3・2 部品の表現と生成

設計にオブジェクト指向言語(またはそれにもとづくGUIやCAD)を使用するなら、部品はオブジェクトとし、標準的な部品を生成する手続きはメソッド(関数)として定義しておけばよい。ここではそのひとつの例をあげる。FDM方式のようにフィラメントをかさねて造形する3D印刷においては、部品をつぎのように、ふとさのある糸(string) S_i のならび (S_1, S_2, \dots, S_n) として表現することができる⁽⁴⁾。

$$S_i = (P_{i_start}, P_{i_end}, c_i, v_i)$$

ここで P_{i_start} は糸の始点であり P_{i_end} はその終点である(両者は直線によってむすぶことを仮定する)。 c_i は糸の断面積(これはフィラメントの密度に関するパラメタによっておきかえられる)、 v_i は印刷速度(秒速)つまりヘッドの移動のはやさである。 v_i は概念的には不要だが、実装(糸の印刷)のためには便利なパラメタである。

部品のクラス名はTraceとしているが、まずつぎのようにして空の部品をつくる。

```
part = draw3dp.Trace(crossSection, x0, y0, z0)
```

crossSection はフィラメント断面積の初期値、(x0, y0, z0) はヘッドの初期位置である。

もっとも単純な部品として円や円弧があるが、円はライブラリがふくむつぎの関数によって生成される。

```
part.circle(radius, x1, y1, z1)
```

(x1, y1, z1) を中心とする半径 radius の円が部品 part に追加される。糸は直線をつないだものであるから、円は直線によって近似される。

また、つぎの関数をよびだせば1重の平面らせんがえがける(図2(a)参照)。hpitch がらせんのピッチである。

part.spiral(radius, hpitch, x0, y0, z0)

より複雑な部品生成関数として、つぎのようなものを用意している。

part.helix(radius, height, vpitch, x0, y0, z0)

helix は1重のヘリックス(垂直方向のらせん)つまり底面がない円筒をえがく(図2(b)参照)。層なしにかつ層のつぎめもなしに、任意の高さのうすい円筒をえがくことができる。

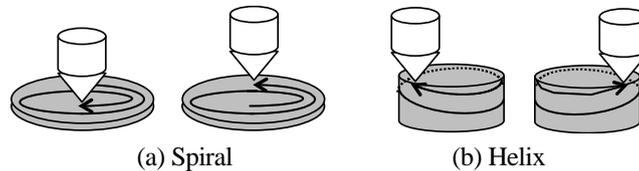


図2 平面らせん (spiral) とヘリックス (helix)

一重の円筒が helix という関数によって生成されるのに対して、中心までつまった円柱を生成するには cylinder という関数を使用する。

part.cylinder(radius, height, vpitch, hpitch, x0, y0, z0)

hpitch は水平方向のピッチ, vpitch は垂直方向のピッチである。

以上のようなライブラリだけではかぎられた形状しかつけれないが、この方法を拡張していくことができるとかんがえられる。なお、ライブラリの試作品 draw3dp.py は <http://bit.ly/1EBVbSB> において公開している。

3・3 部品のくみたて

3・1 においてのべたように、製造においては部品をくみあわせるしくみが必要である。切削加工の場合は加工後にくみたてるが、付加工の場合は複数の部品を同時に製造することが多い。部品ごとに印刷するだけでなく、すべての部品がくみあわせられたかたちで印刷することも多いので、このような場合も想定する必要がある。

部品がつぎの2条件をみたしていれば、部品を逐次に印刷する(接続するまたは逐次に反復する)ことによって、くみたてられる。

- 印刷ヘッドがすでに印刷したフィラメントによって妨害されないこと。
- 印刷したフィラメントが印刷台(プリントベッド)やすでに印刷したフィラメントによって支持されること。

部品の印刷順序をどのようにきめてもこれらの条件をみたすことができないときは、部品を分割すれば条件がみたされるかどうかをしらべる⁽⁴⁾。鎖は分割しないかぎり部品(輪)の印刷順序変更によって印刷可能にはならない。そこで、前論文⁽³⁾においては単純な逐次実行だけでなく、分割して分割点を(まだ自動化されていないため)手動できめ、立体化した(鎖状にした)五輪の印刷をこころみた。

部品のくみたて手続きを関数化すれば複合部品を生成する関数となり、モジュラーな構造が実現される。

3・4 変形と変調

前節でのべたように宣言的な方法とくらべると制約があるため、部品のくみあわせだけでなく、他の方法もあわせてより多様な形状がつけられるようにしている。そのための方法として変形 (deformation)⁽⁴⁾と変調 (modulation)⁽⁶⁾を提案しているが、この報告の主題からははずれるので、これだけの記述にとどめる。

4. 手続き的な3D印刷の実験

手続き的な方法で(まだ「製品」とはいえない)オブジェクトを設計し、3D印刷する実験をおこなった。現在のところはオブジェクトを構成する部品数は1~3個程度であり、多様な形状をつくるために、くみあわせよりは変形と変調の技法をおもに使用している。ここでは2種類のくみあわせでつくったオブジェクトを紹介する。これらはそれぞれ10~20分で印刷できる。いずれもWebサイト (<http://bit.ly/1EZ4SZI> または <http://store.shopping.yahoo.co.jp/dasyn/>) を通じてサンプル品を入手することができる。

第1に、ヘリックスにうすい円柱形の底をつけてつくったカップを変形・変調してえられた多様な形状の皿や

容器を図3に示す。図3(a)はヘリックスの部分をひらたく変形してつくった皿 (<http://store.shopping.yahoo.co.jp/dasyn/1011-04.html>, <http://youtu.be/5P1vaahzW98>) だが、フィラメントを水平にちかい角度でまきつけるときもサポートは不要である。場所によって皿の角度やフィラメントの密度がことなるために光の反射に変化が生じている⁽⁴⁾。純粋な透明のPLAを使用することによってこのような輝きをえている。図3(b)はヘリックスに円をハート型に変換する変形を適用してつくった皿である (<http://store.shopping.yahoo.co.jp/dasyn/3db0f5bafe.html>, http://youtu.be/G9x14DZYN_8)。円をハート形に変形させる関数を使用した。図3(c)⁽⁴⁾は三角関数を多用した花瓶 (<http://store.shopping.yahoo.co.jp/dasyn/1025-10.html>) である。図3(d)は上記とおなじハート形をねじってつくった花瓶 (<http://store.shopping.yahoo.co.jp/dasyn/1073-01.html>) である (いずれも上記のサイトからサンプル品入手可能)。

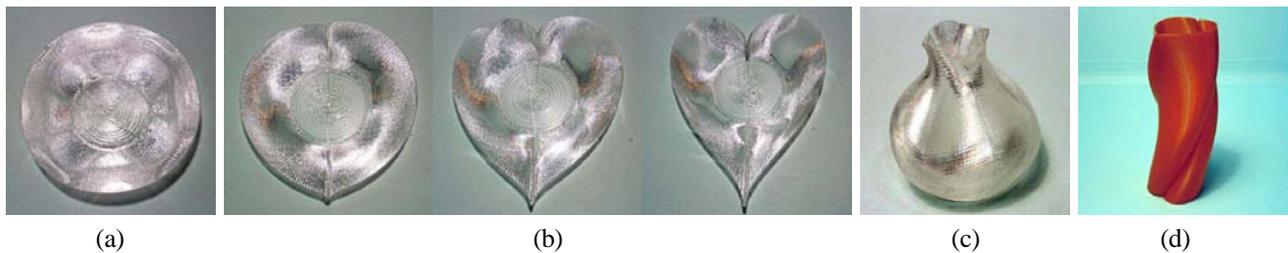


図3 皿と容器

第2に、1個のヘリックスだけからつくったオブジェクトを示す。図4の左はヘリックスを変形してつくった単純な球である。印刷時に球を1点だけでは支持できないため支持にすこしくふうが必要だが、通常の意味のサポートは使用していない⁽⁴⁾。図4の右は三角関数によって球をさらに変形してつくったオブジェクトである。図5には球を地図によって変調してつくった地球儀である (<http://store.shopping.yahoo.co.jp/dasyn/1201-03.html>, <http://youtu.be/YWx1vqig2-o>)。



図4 球と変形球



図5 変調した球 (地球儀) と碗

5. 結 語

設計者による手続き的な記述は切削加工においては成功していない。しかしそれが付加加工においては利点があるとかんがえて、その方法論を提案するとともに、実現したプログラミング言語とそれを使用して3D印刷のころみた。この方法では印刷可能な形状は限定されるが、層をなくして層のつぎめもなくし、従来の方法においては必要だった支持材料(サポート)もなくして、シームレスでより美的な印刷を実現している。

文 献

- (1) Brown, S. A., Drayton, C. E., and Mittman, B., "A Description of the APT Language", *Communications of the ACM*, Vol. 6, No. 11, pp. 649-658, 1963.
- (2) ISO, "Numerical Control of Machines -- NC Processor Input -- Basic Part Program Reference Language", ISO 4342:1985.
- (3) Kanada, Y., "Method of Designing, Partitioning, and Printing 3D Objects with Specified Printing Direction", *2014 International Symposium on Flexible Automation (ISFA)*, July 2014.
- (4) Kanada, Y., "3D Printing of Generative Art using the Assembly and Deformation of Direction-specified Parts", *Rapid Prototyping Journal*, to be published, 2015.
- (5) Kanada, Y., "Natural-Direction-Consistent 3D-Design and -Printing Methods", *International Journal of Computer, Control, Quantum and Information Engineering*, WASET, Vol. 9, No. 4, pp. 689-693, 2015.
- (6) Kanada, Y., "Creating Thin Objects with Bit-mapped Pictures / Characters by FDM Helical 3D Printing", *8th Int'l Conference on Leading Edge Manufacturing in 21st Century (LEM21)*, Oct. 2015.
- (7) Kramer, T. R., Proctor, F. M., Messina, E., "The NIST RS274NGC Interpreter - Version 3", National Institute of Standards and Technology (NIST), NISTIR 6556, August 2000.
- (8) Ross, D. T., "Origins of the APT Language for Automatically Programmed Tools", *ACM SIGPLAN Notices*, Vol. 13, No. 8, pp. 61-99, August 1978.