



---

---

## NCVC 解説書

---

---

NCVC Ver0.14.42 用  
2005 年 10 月 初版

# 目次

<b>1</b>	<b>はじめに</b>	<b>1</b>
<b>2</b>	<b>基本編</b>	<b>2</b>
2.1	CAD での作図	2
2.2	CAD データの読み込み	3
2.3	加工条件の設定	4
2.4	G コードの生成	5
2.5	加工条件の設定 2	7
2.6	G コードの加工シミュレーション	8
2.6.1	加工時間の表示	8
2.6.2	トレース	8
2.7	穴加工	9
<b>3</b>	<b>中級編</b>	<b>11</b>
3.1	移動レイヤ	11
3.1.1	加工開始位置指示レイヤ	11
3.1.2	強制移動指示レイヤ	13
3.2	G コード (文字) の埋め込み	15
3.3	深彫切削 (Z 軸ステップ切削)	16
3.4	複数レイヤ処理 (2.5D 切削)	18
3.4.1	レイヤごとの Z 座標指定	18
3.4.2	レイヤごとの切削条件	22
3.4.3	個別出力について	23
<b>4</b>	<b>応用編</b>	<b>25</b>
4.1	方向制御	25
4.2	形状切削	26
<b>5</b>	<b>パワーユーザ編</b>	<b>28</b>
5.1	スクリプト作成のすすめ	28
5.1.1	まえがき	28
5.1.2	例題	28
5.1.3	実行環境	29
5.1.4	あとがき	29

## 1. はじめに

---

NCVC (NC Viewer and Converter) は CAM ソフトです。主に CAD 情報から NC 工作機械を動かすための G コードを生成するアプリケーションです。NCVC に CAD (作図) 機能は付いていません。作図は別途 CAD ソフトで行って下さい。CAD ソフトは Jw\_cad for Windows (以降 Jw\_cad と略記) を推奨しますが、以下の条件に当てはまる CAD なら NCVC の入力源としてそのまま使えます。

- R12 形式の DXF が出力可能なもの
- DXF にレイヤ情報 (レイヤ名) が出力できるもの

ほとんどの CAD が当てはまると思います。普段使い慣れた CAD ソフトをご使用下さい。逆に言うと NCVC を使うためにわざわざ新たな CAD の操作方法を覚える必要が無いということです。以降本解説書での CAD 操作は Jw\_cad をベースに解説します。

残念ながら CAD を使ったことが無いという方、本解説書では CAD の使用を前提にしています。上記条件を満たしていればドロー系ソフトでもかまいませんが、正確な作図が要求されます。まずは CAD での作図方法を習得して下さい。

もう 1 つ、NCVC には G コードのシミュレーション機能がありますが、冒頭で述べたとおり NCVC は G コードの生成を主な目的としています。全ての G コードには対応していませんので、サポートされる G コードは付録の対応 G コード一覧を参照して下さい。また、シミュレーション結果と工作機械の動作が必ず一致するとも限りません。実際の加工にはくれぐれもご注意下さい。

NCVC (NC Viewer and Converter) は眞柄賢一の著作物です。Jw\_cad for Windows は Jiro Shimizu & Yoshifumi Tanaka 両氏の著作物です。その他本解説書に記載された製品名・会社名などは、各社の商標または登録商標です。各権利を侵害する行為は堅くお断りします。本解説書に掲載されている操作等は各自の責任で行って下さい。著者は一切責任を持ちません。

この解説書は一太郎で書いた初代 NCVC 解説書を  $\text{\TeX}$  で書き直したものです。古い表記がありますが、適宜置き換えてください。関連書籍を出版しているので、原則そちらを参照してください。

<https://shop.ohmsha.co.jp/shopdetail/000000005207/>

少なくとも書籍が絶版になるまでは、内容が更新されることはありません。

## 2. 基本編

### 2.1 CAD での作図

まずは基本的な加工を行うための基本的な作図方法を解説します。図 2.1 のような図形を書きましょう。切削対象（ワーク）を示す矩形と、その矩形左下に円を 1 つ。「NCVC」という文字は、線をつなぎ合わせたデータです。

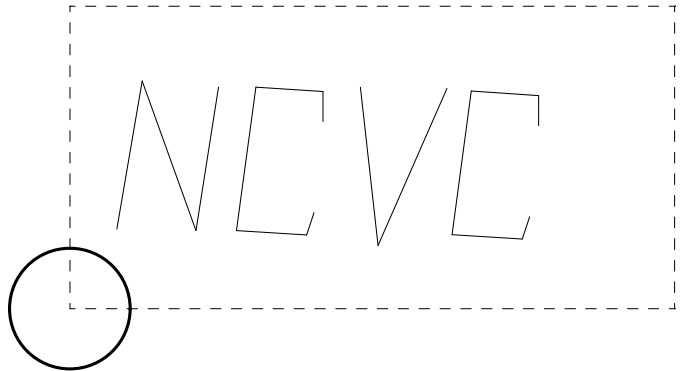


図 2.1 サンプル図形

NCVC は CAD での作図情報を全て読み込むのではなく、特定のレイヤ情報を元に作図データを読み込みます。CAD での作図において必要とされる補助線や寸法線等が加工データには必要なく、これらを選別するための仕様です。

その選別方法は『必要なレイヤに名前を付ける』こと。図 2.2 は図 2.1 のレイヤ情報ですが、0 番レイヤに「ORIGIN」という名前、1 番レイヤに「CAM\_LINE」という名前を付けています。それぞれ機械原点と切削軌跡を示し、この 2 つのレイヤは必須です<sup>\*1</sup>。機械原点レイヤには工作機械の XY 原点を示す円を 1 つだけ作図。大きさは任意ですが、円の中心が XY の原点となります。切削軌跡 CAM\_LINE レイヤには刃物のパス、すなわち削りたい図形を書きます。他、ワーク矩形を示す補助線等は別のレイヤに書きます。レイヤに名前を付ける方法は、それぞれの CAD 操作に準拠して下さい。なお、全てのデータにおいて線種、線色は関係ありません。

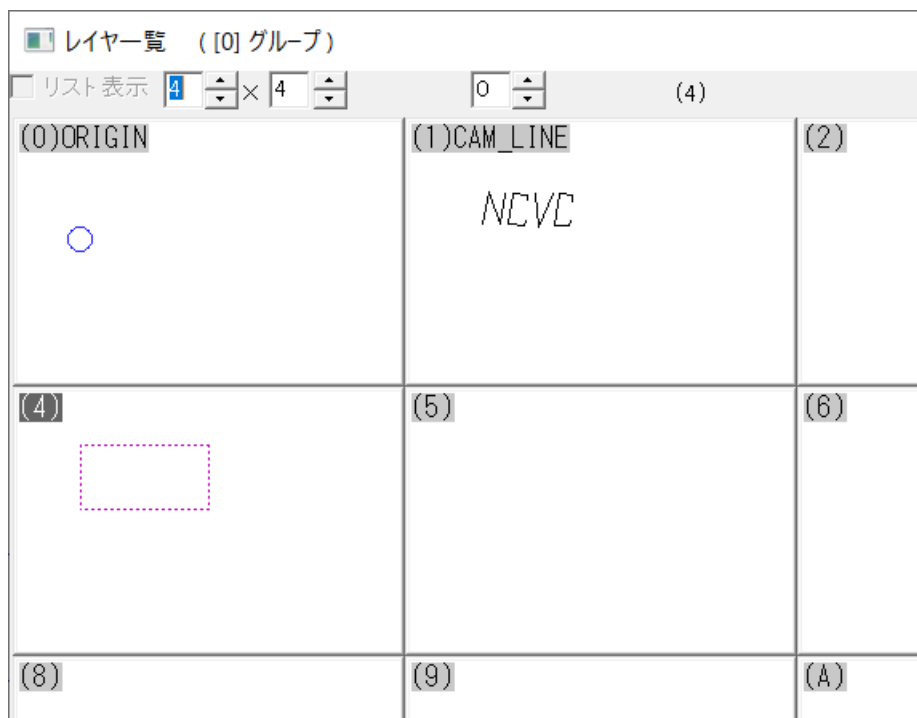


図 2.2 レイヤー一覧

<sup>\*1</sup>実は機械原点レイヤは必須ではありません。詳細は【穴加工】の節で解説しています。

作図が終われば CAD データを DXF 形式で保存します\*<sup>2</sup>。NCVC に CAD データを読み込ませるため DXF 形式で保存する必要がありますが、多くの場合、DXF 形式で保存するとその CAD 独自のデータが失われるため、使用している CAD 独自の形式でも保存しておきましょう。

## 2.2 CAD データの読み込み

NCVC で DXF 形式の CAD データを読み込みます。が、その前に確認。NCVC の **オプション** >> **DXF 関連の設定** をクリックし、NCVC が読み込むレイヤ名を設定して下さい。デフォルトで先ほど設定した値になっていると思います。基本編では「従来互換」のみ解説しますので、図 2.3 の通り設定して下さい。この値は任意です。CAD 側の設定と合わせて下さい。無事読み込めると原点を示す十字（大きさは原点円の直径）と切削対象のパスが表示されます。原点レイヤと切削レイヤ以外に作図した情報、例えば、図 2.2 の 4 番レイヤに書いたワークを表す矩形は読み込まれません（図 2.4）。CAD での線種・線色は無視され、NCVC の設定に基づき表示されます。詳細はリファレンスの表示属性を参照してください。

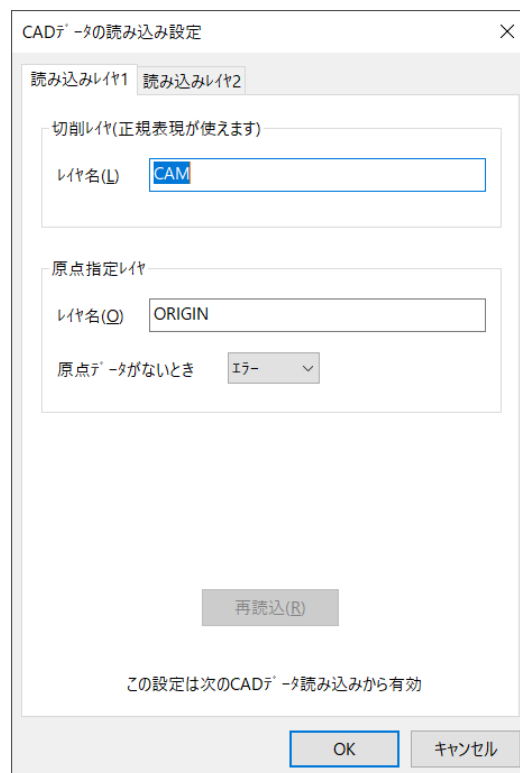


図 2.3 読み込みレイヤ設定

\*<sup>2</sup>Jw\_cad の場合、DXF 形式で保存する必要はありません。NCVC は JWW 形式を直接読み込むことが可能です。詳細は【パワーユーザ編】の【アドイン作成のすすめ】を参照して下さい。



図 2.4 CAD データの読み込み

## 2.3 加工条件の設定

いよいよ CAD データから G コードを生成するわけですが、ご覧の通り読み込んだ CAD データは 2 次元です。工作機械の Z 軸方向の移動はどうやって制御するのでしょうか？答えは「加工条件」の中にあります。[オプション] > [切削パラメータの設定] をクリックし、条件ファイル (nci ファイル) を選択します。標準で用意されている [Init.nci] の設定を変更しましょう。

条件ファイルを選択すると図 2.5 のダイアログが表示されます。ここで重要なのが切削原点 (G92) の Z 値と R 点、切り込みパラメータの 3 つです。

図 2.6 は工作機械を正面から見た図、上下に Z 軸、左右に X 軸です。ワークをセットしたあと、ワーク平面を基準に Z センサー等で Z 軸の位置決めを行います。これを切削原点 (G92) の Z 値とします。Z センサーの厚みが 100mm なら 100 と入力です。センサーでの調整後、好みの位置に移動させてもかまいません。無論そのときは移動した座標値を入力して下さい。

次に切り込みですが、イメージ通り、ワークに何ミリ切り込むかという設定です。最後に R 点ですが、これは次の切削領域、この例で言うと “N” を削って “C” に移動するときの Z 値を指定します。Z 軸の初期位置 (原点) で移動してもかまわないのですが、初期位置は高く設定する傾向があるため、効率よく移動できる下限値と考えて下さい。この設定ではワーク平面上空 1mm の所で刃物が次の切削領域へ高速移動します。

ワーク平面を基準に値を選びましたが、Z センサー調整後の位置を基準、すなわち、ワーク上空 10mm の位置を Z 軸の原点 (G92Z をゼロ) としたとき、この例では R 点が -9mm、切り込みは -12mm となります。意味は同じですから各自の好みや考えやすい方で指示して下さい。

他、主軸回転数や送り速度など、ワーク材質に合わせて設定します。



図 2.5 加工条件の設定

## 2.4 G コードの生成

加工条件の設定ができればあとは NCVC の仕事。  
 ファイル》NCデータへの変換》単一条件（従来互換）をクリック。  
 出力ファイル名（自動設定）と条件ファイルを指定（図 2.7）し、OKをクリックすれば...

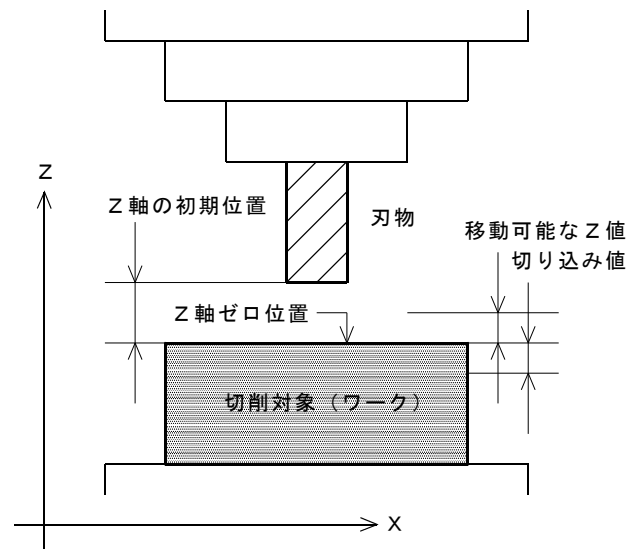


図 2.6 Z 軸における各パラメータの関係

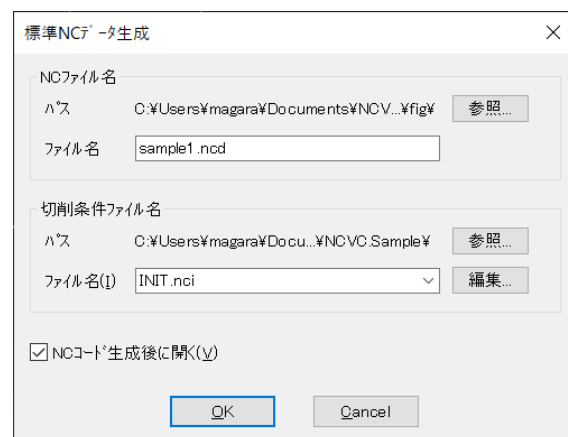


図 2.7 G コードの出力と条件ファイルの指示

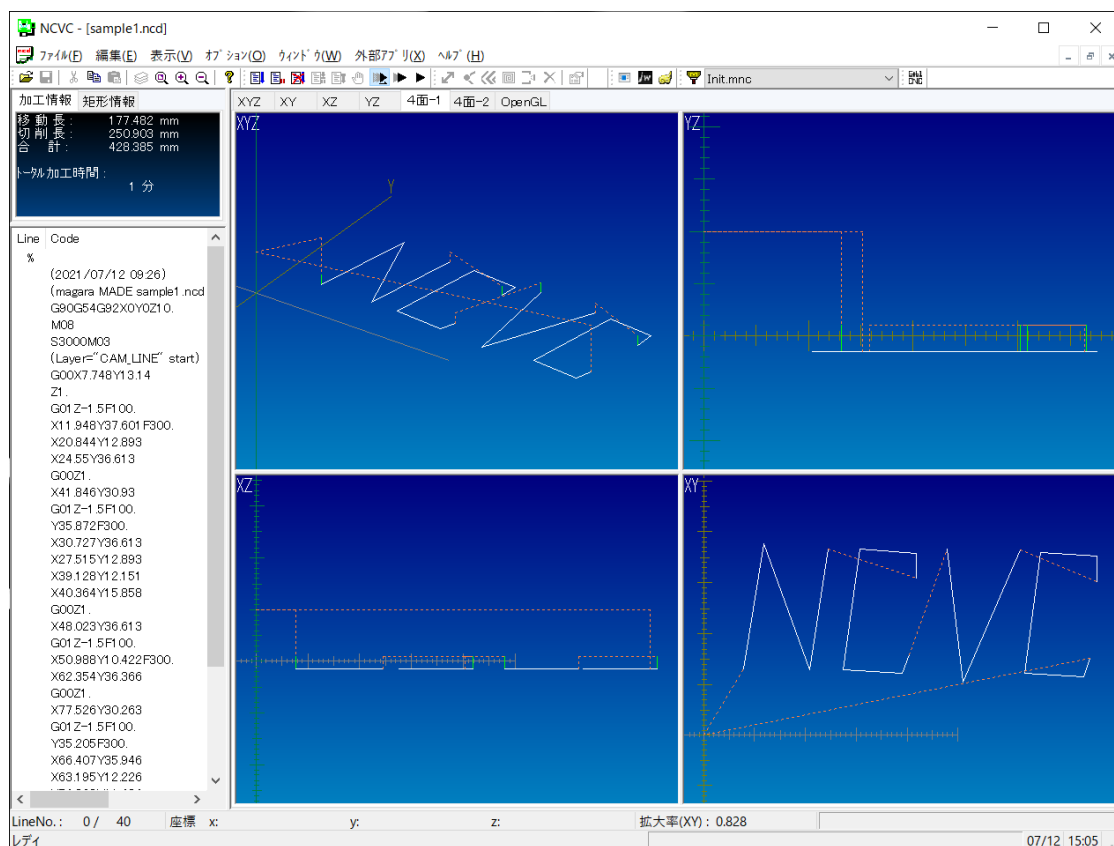


図 2.8 Gコードシミュレーション画面

おめでとうございます！見事 G コードが生成できました。図 2.7 で「NC 生成後に開く」にチェックが入っていると、即座に結果を確認することが出来ます。図 2.8 に G コードのシミュレーション結果を示します。

#### ここまでの【まとめ】

##### (1) CAD での操作

- 工作機械の XY 原点を示す円を原点レイヤに作図
- 刃物の軌跡を切削レイヤに作図
- 原点レイヤと切削レイヤに名前を付ける
- 線種・線色は無視され、NCVC の表示属性により表示される

##### (2) NCVC での操作

- CAD データを読み込むために、読み込みレイヤの設定を行う
- Z 軸の原点や切り込み量は加工条件で設定する



## 2.5 加工条件の設定 2

ところで図 2.8 の左、G コードのリストに注目すると、G54 ワーク座標系選択や M コードなど、CAD での作図データ以外のコードが生成されています。これらの設定は加工条件（図 2.5）のヘッダー・フッターの両カスタムファイルによるものです。以下のリスト、標準で用意されているカスタムファイルで、左がヘッダー、右がフッターです。それぞれ生成開始時と終了時に参照され、生成する G コードに併合されます。使用できる置換キーワードなど詳細は第 5 章の【リファレンス】で解説します。

ソースコード 2.1 Header.txt

```
%
({MakeDate} {MakeTime})
({MakeUser} MADE {MakeNCD} FROM {MakeDXF} AND {MakeCondition})
{G90orG91}G54{G92_Initial}
M8
{Spindle}M3
```

ソースコード 2.2 Footer.txt

```
M9
M5
{GOXY_Initial}
M30
%
```

テキストファイルなのでメモ帳などで編集できます。例えばワイヤー加工機のワイヤー設定、レーザー加工機の実出力設定など、ヘッダー・フッターファイルで指定しなければならない設定は多々あると思います。実際の運用では、加工条件ファイルを含め、対象となる工作機械ごとに設定ファイルを用意する方が良いでしょう。本解説書では標準ヘッダーにチップコンベア起動の“M68”を挿入しています。他、そのまま使っても問題無いと思いますが、見づらいうならコメント行（ヘッダーの 2～3 行目）は削除してもかまいません。

生成時に図 2.9 のようなエラーメッセージが出る場合があります。特に標準のインストールパス以外にインストールされた場合、両カスタムファイルが探せませんので、加工条件にて正しいパスを設定して下さい。

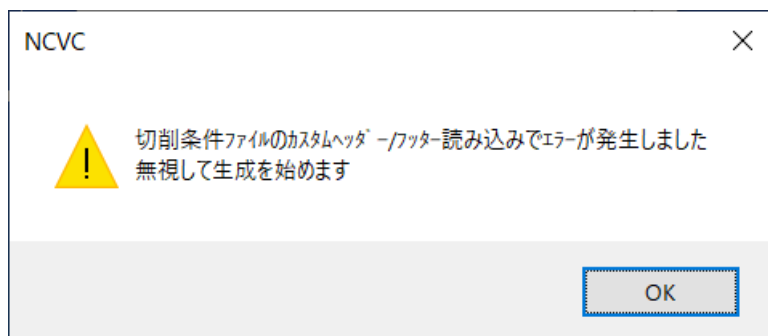


図 2.9 カスタムファイルのエラーメッセージ

## 2.6 Gコードの加工シミュレーション

### 2.6.1 加工時間の表示

図 2.8 の左上に注目して下さい。[トータル加工時間]が表示されていますが、初期設定の段階では表示されません。G01～G03（直線補間や円弧補間）等の加工時間はFパラメータとその移動距離から算出できますが、G00 早送りの移動速度は工作機械固有のパラメータです。これを NCVC に設定することで、加工時間が算出されます。

「オプション」>「工作機械の設定」をクリックし機械情報ファイル（mnc ファイル）を選択すると図 2.10 のダイアログが表示されますので、[位置決め (G0) 移動速度] に工作機械の早送り移動速度を設定して下さい。ここが空白の場合、加工時間は表示されません。

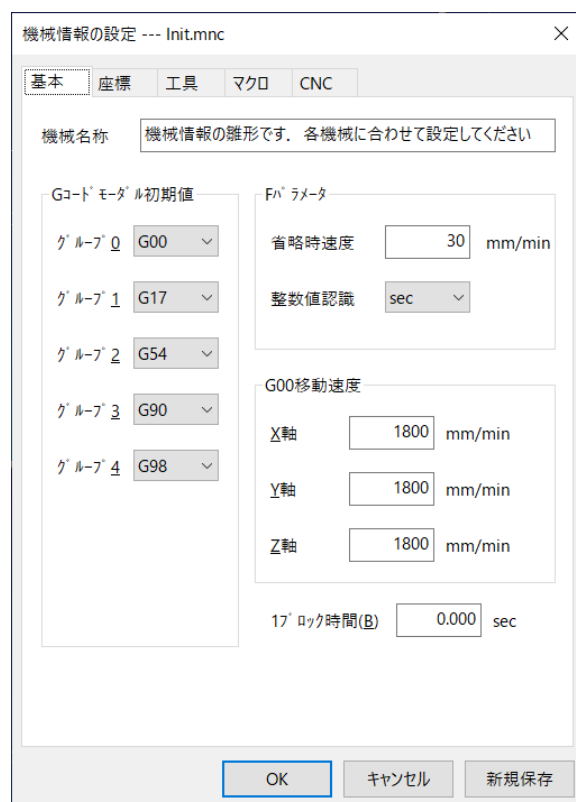


図 2.10 機械情報の設定

この設定も工作機械ごとに用意しましょう。現在選択されている機械情報ファイルはツールバーに表示されず（図 2.11）、加工時間は機械情報ファイルを切り換えるたびに再計算されます。



図 2.11 機械情報ツールバー

### 2.6.2 トレース

図 2.12 はトレースのツールバーです。とりあえずトレース速度を中速（右から 2 番目）にしてトレース実行（一番左）を押してみましょう。どういう手順で加工されていくかを見ることができます。トレースのスピード調整はリファレンスの表示属性を参照してください。その他の操作については解説するまでもなく使えると思います。



図 2.12 トレースツールバー

## 2.7 穴加工

基本編の最後として、固定サイクルでの穴加工データの生成を解説します。作図レイヤは【2.2 CAD データの読み込み】と同じで、切削レイヤに穴加工情報を作図します。

図 2.13 は矩形の四隅と中央に穴加工を行うための作図データです。穴加工を行いたい場所を中心に、四隅には半径 3mm の円、中央には半径 5mm の円を作図しています。今回は XY 原点を示す原点レイヤは作図していません。

工作機械の XY 原点がワークの中央に位置し、CAD データ上でも最大占有領域の中央が原点で良い場合、NCVC が XY 原点を補間する機能があります。

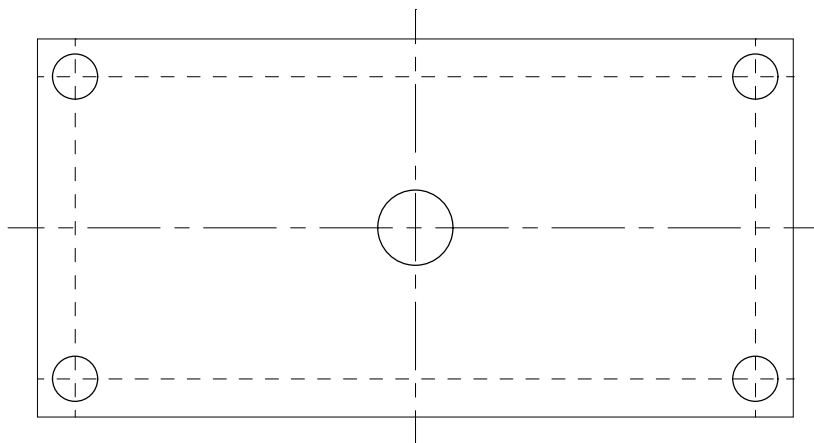


図 2.13 穴加工サンプル図形

図 2.3 (p.3) の読み込みレイヤ設定と違うのは、[原点データがないとき]の選択肢。これをエラー以外に設定しておけば、NCVC が XY 原点を補間します (図 2.14)。

ただし、補間は CAD データの最大占有領域から算出されますので、意図しない位置に原点が補間される場合があります。特別な場合を除き【2.1 CAD での作図】の通り原点情報を作図することを推奨します。

穴加工に関する加工条件は【2.3 加工条件の設定】で解説した条件ファイルの [穴加工] タブにて行います (図 2.15)。主軸回転数や切削送りなど、穴加工独自に指定可能です。ここで注目すべき所は [拡張設定グループ] の各種項目。[円データも穴加工データと見なす] にチェックを入れ、対象となる半径を入力することで、その中心座標に穴加工データを生成します。従来、穴加工の入力源は点データでしたが、イメージがつかみにくい、点データを DXF に吐けない CAD がある等の理由から、円データでの穴加工が拡張設定となっています。

加工条件が設定できればあとは【2.4 Gコードの生成】通りです。図 2.16 にシミュレーション結果を示します。加工条件の [グルーピング順序] を [降順] にすることで半径の大きい中央から固定サイクル命令が生成されます。また、[大きさごとにコメントを埋め込む] にチェックを入れることで、それぞれの半径ごとに NCVC がコメントを埋め込みますので、手作業での修正、例えば工具交換命令を埋め込む等、編集の目安となります。



図 2.14 読み込みレイヤ設定 2



図 2.15 穴加工の加工条件設定



図 2.16 穴加工のシミュレーション画面

## 3. 中級編

### 3.1 移動レイヤ

NCVC には、原点レイヤ・切削レイヤの他にあと 3 つのレイヤ情報を読み込む機能があります。ここではそのうちの 2 つ、加工開始位置指示レイヤと強制移動指示レイヤを解説します。

#### 3.1.1 加工開始位置指示レイヤ

図 3.1 のような加工を考えます。原点はワーク矩形左下で円を内側から螺旋状に切削したいのですが、これだけでは思ったような G コードを生成できません。

NCVC は次の切削データを検索するとき、現在位置に最も近い座標を検索します。したがって、原点から一番近い座標である外側の座標から G コードの生成を始めます。

これを回避するため、NCVC では「加工開始位置指示レイヤ」を用意しています。CAD での作図で原点・切削の両レイヤとは別のレイヤを用意し、円を 1 つ作図して下さい。レイヤ名も設定です。作図方法は原点指示と同じ、円の中心が加工開始座標となります。

NCVC での設定は【2.2 CAD データの読み込み】と同じです。「読み込みレイヤ 2」のタブをクリックし、NCVC が読み込むレイヤ名を設定して下さい（図 3.2）。「読み込みレイヤ 2」の設定は必須ではありませんが、CAD 側で意図的に作図しない限り NCVC は読み込みませんので、常にこの設定にしても問題ありません。レイヤ名は、原点・切削の両レイヤと同様に任意です。CAD 側の設定と合わせて下さい。加工開始指示を指示したシミュレーション結果を図 3.3 に示します。

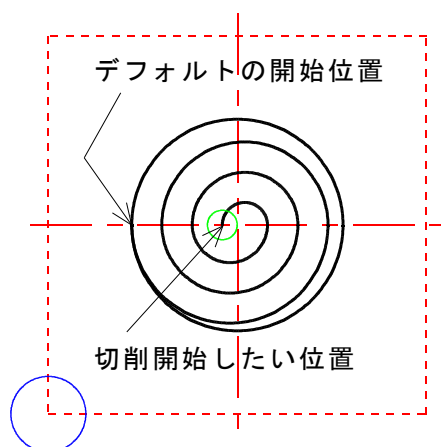


図 3.1 サンプル図形



図 3.2 読み込みレイヤ設定



図 3.3 加工開始位置を追加した G コードシミュレーション画面

加工開始位置指示レイヤにはもう 1 つ機能があります。例えば図 3.4 のようなワーク形状の場合、原点からの最短移動ではワークに干渉してしまいます。この場合、図 3.5 のように干渉しないような移動軌跡を加工開始位置指示レイヤに作図することで、原点からの移動動作を制御することができます。Z 軸の初期座標を上げることで回避できる場合もあります。用途に応じてご使用下さい。

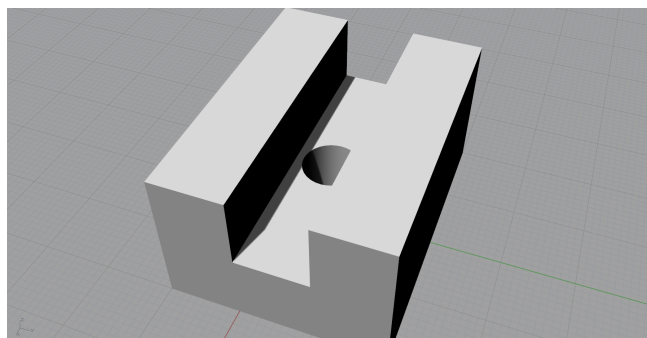


図 3.4 サンプルイメージ図

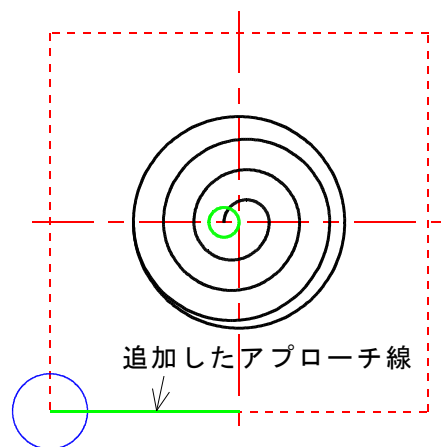


図 3.5 アプローチ線

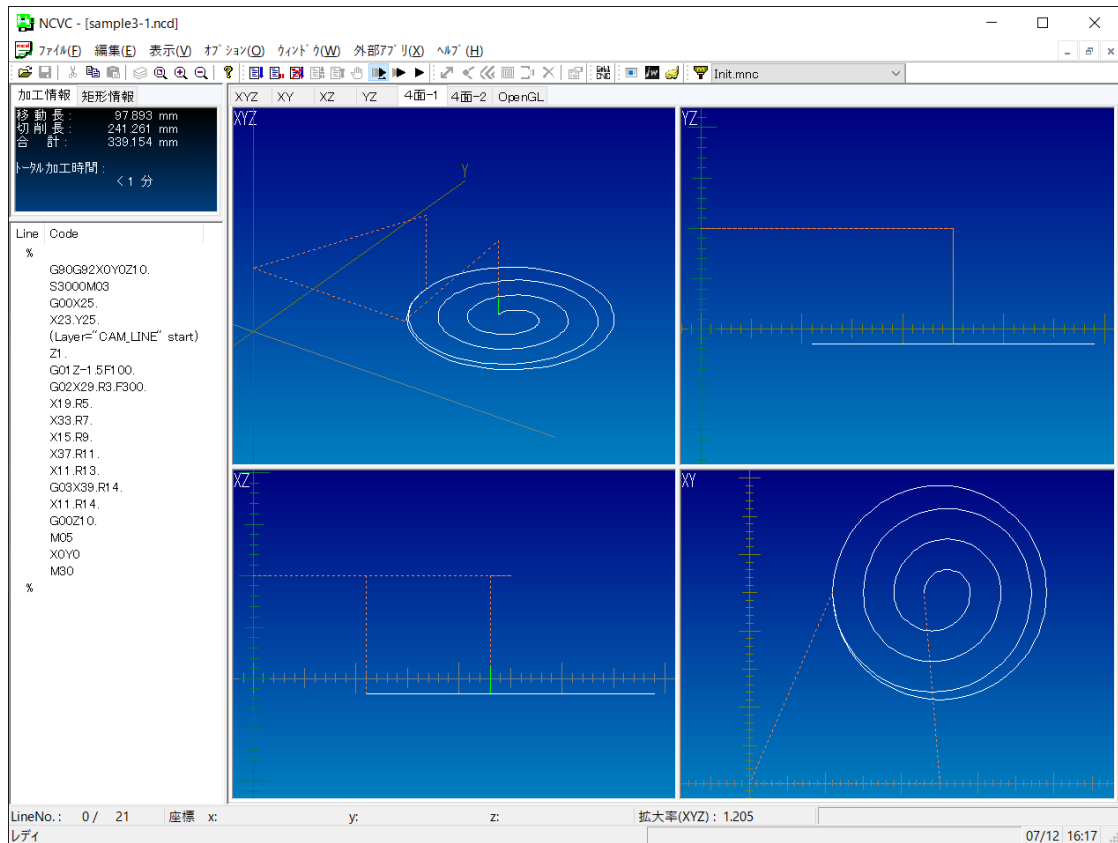


図 3.6 アプローチ線を追加した G コードシミュレーション画面

### 3.1.2 強制移動指示レイヤ

図 3.6 のシミュレーション結果を見ても解る通り、原点へ戻るときもワークと干渉してしまう可能性があります。加工開始位置指示レイヤは、最初の加工位置やアプローチを指示するものですが、強制移動指示レイヤは、次の切削データの検索途中で NCVC に移動を指示する情報となります。つまり、次の切削領域への移動、Z 軸の上下が必要などのときに強制移動指示レイヤが参照されます。

強制移動指示レイヤは線データのみを認識します。図 3.2 で設定した移動レイヤに作図して下さい。また、必ず切削データと接続されていなければなりません。図 3.7 は切削が終わったあとの移動軌跡を作図したものです。

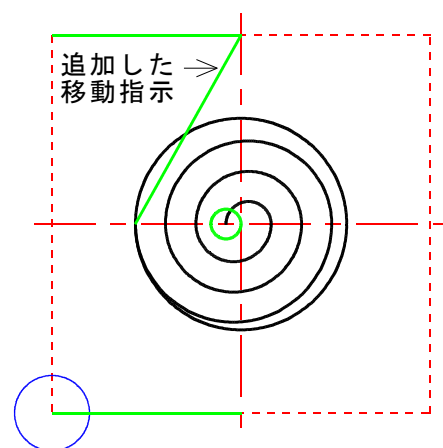


図 3.7 強制移動指示レイヤを追加

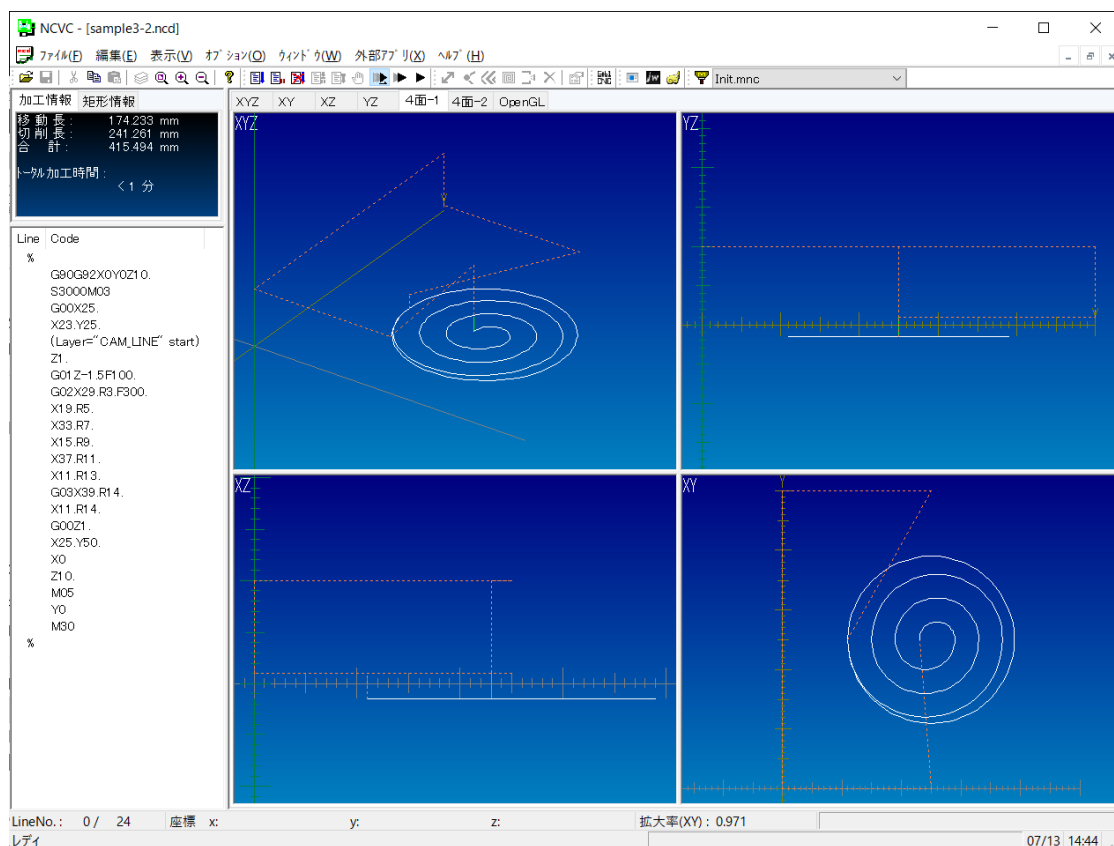


図 3.8 強制移動データを追加した G コードシミュレーション画面

図 3.8 のシミュレーション結果から、強制移動指示レイヤが R 点で移動していることが解ります。この設定は加工条件の「レイヤ」タブにあります。今回の例では「イニシャル点復帰」が正解ですが、強制移動指示レイヤは大抵の場合、次の切削領域への移動制御に使われるため、通常は「R 点復帰」で問題無いと思います。

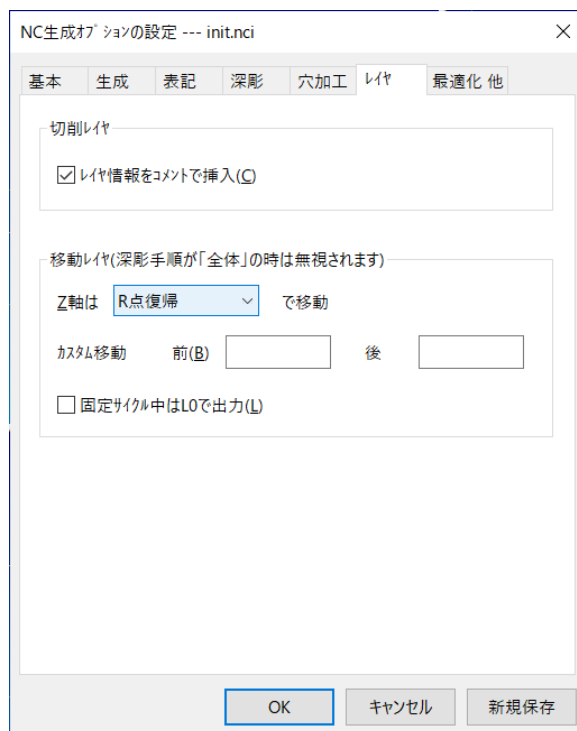


図 3.9 強制移動指示レイヤの Z 値設定



## ここまでの【まとめ】

### (1) 加工開始位置指示レイヤ

- 加工開始位置を円で示す，またはアプローチ線を作図
- 円や線は切削データと繋がって無くても良い

### (2) 強制移動指示レイヤ

- 切削データが途切れ，次の切削領域へ移動するとき，参照される
- 切削データと繋がっていないなければならない
- 強制移動指示レイヤのZ値は加工条件の設定による

## 3.2 Gコード（文字）の埋め込み

NCVCで生成されるGコードは，基本的に位置決めと直線・円弧補間のG00～G03，固定サイクルG81～の一部だけです．ここでは工作機械の特殊コード（例えばATCによるツール交換コード）やカスタムマクロ呼び出しコードなど，生成されるGコードに任意の文字列を埋め込む方法を解説します．

作図方法は至極簡単．図3.10に示すように，埋め込みたいオブジェクト（線や円弧）の端点に文字を作図するだけです．文字データは原点レイヤ以外，すなわち，切削レイヤと2つの移動レイヤ，それから【3.1 移動レイヤ】の節で解説した最後のレイヤである「コメント文字列挿入レイヤ」（p.11 図3.2）に作図することができます．

埋め込みタイミングは，文字専用レイヤである「コメント文字列挿入レイヤ」が先に参照され，次に各レイヤタイプに属する文字データが参照されます．「コメント文字列挿入レイヤ」に書かれた文字はその名の通りGコードに対するコメントと見なされ，カッコで括られます．その他のレイヤに書かれた文字はそのまま出力されます．なお，文字データは1行で作図する必要がありますが，複数行の任意Gコードを埋め込みたい場合は改行位置で“\n”と入力して下さい．

図3.10の作図から生成されるGコードは以下の通りです．反転部分が文字情報から拾われたデータです．作図は簡単ですが，文字を埋め込むレイヤと場所（タイミング）には若干知恵を絞る必要があります．

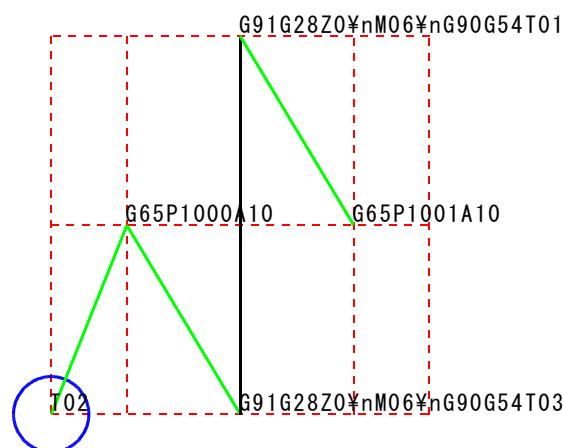


図 3.10 文字埋め込みサンプル

```
%
G90G54G92X0Y0Z10.
M8
M68
S3000M3
T02
G00X10.Y25.
G65P1000A10
X25.Y0
G91G28Z0
M06
G90G54T03
Z1.
G01Z-2.F100
```

右へ続く ➤

```
Y50.F300
G91G28Z0
M06
G90G54T01
G00Z1.
X40.Y25.
G65P1001A10
Z10.
M9
M5
X0Y0
M30
%
```

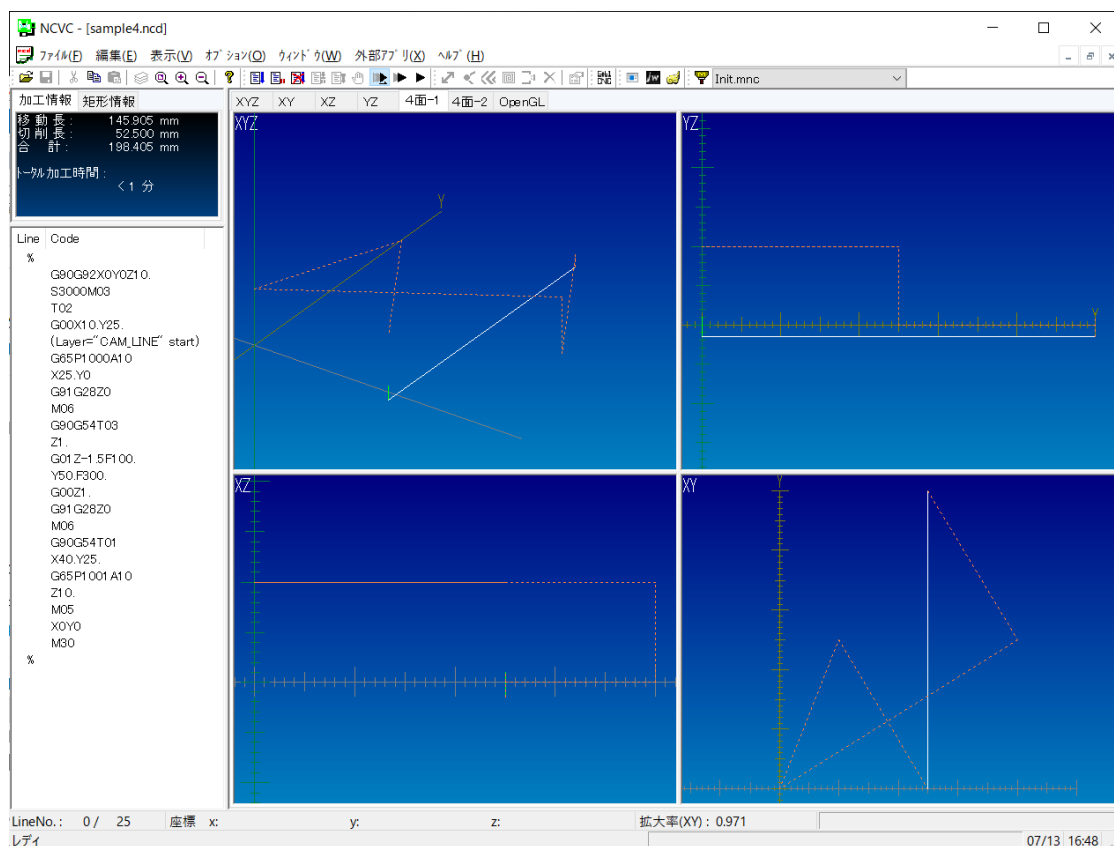


図 3.11 Gコードシミュレーション画面

### 3.3 深彫切削（Z軸ステップ切削）

ここでは「深彫切削」を解説します。細い刃物での切削で1回の切り込み量に制限がある場合等、Z軸を段階的に下げて切削するコードを生成することができます。

深彫切削は加工条件で指示します。サンプルデータは【2.1 CADでの作図】を使いましょう。同じく【2.3 加工条件の設定】の要領で条件ファイルを開き、「深彫」のタブをクリックして下さい（図 3.13）。

ポイントは2点。まず「基本切り込み」の値ですが、ここでは“1回目の切り込み量”と解釈して下さい。これは基本タブから参照されている値です。このタブで値の変更はできません。

次に深彫切削グループ一連の設定。「最終切り込み」で最終的に必要な深さ、「切り込みステップ」にて刃物が一度に切り込む量を指定します。とりあえずこの設定情報でGコードを生成してみると、図 3.13 のようになります。

その他の加工条件で代表的なもの。「手順」は連続線グループを先に彫り進める（一筆）か、同一Z値で全体を彫り進める（全体）かの設定。「方向」は次のZ値を切削するときにそのまま戻る（往復）か、一旦最初の加工座標まで戻る（一方）かの設定です。いずれもシミュレーション画面で明確に出ますので、状況に応じて設定して下さい。残りの設定は【リファレンス】で解説します。



図 3.12 深彫の設定

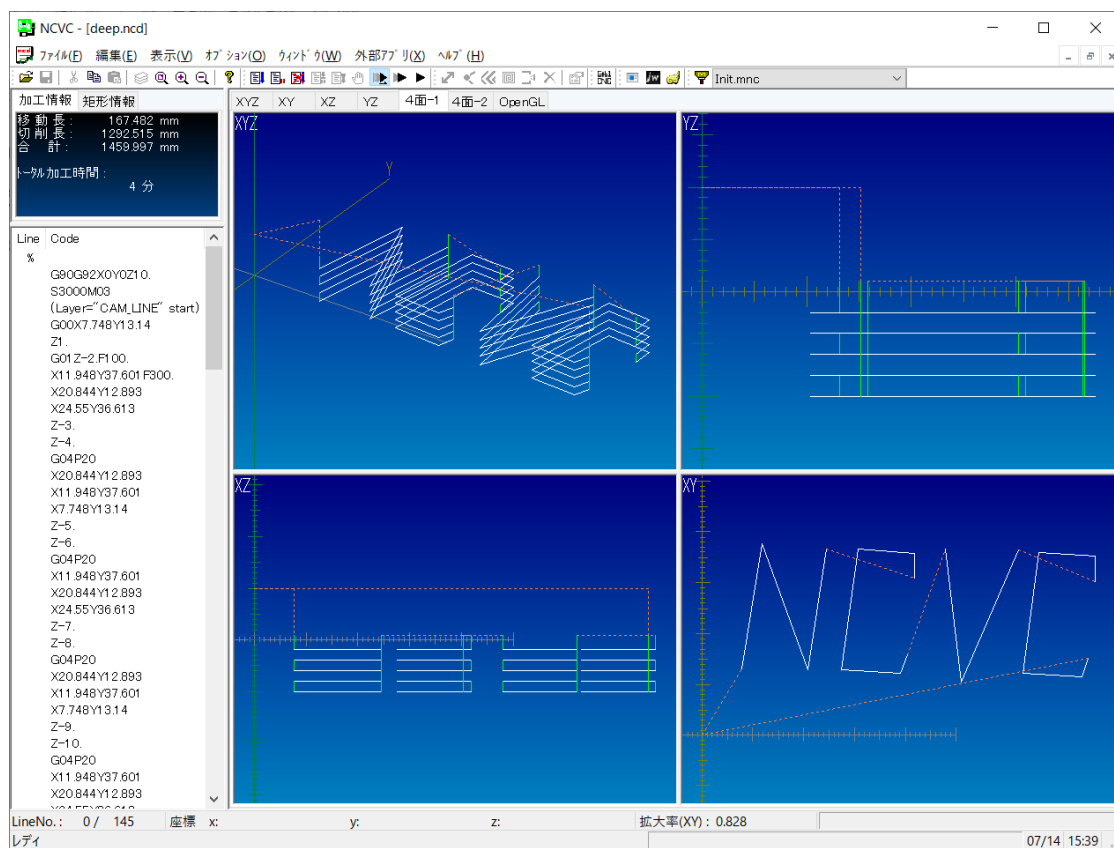


図 3.13 深彫のシミュレーション結果

### 3.4 複数レイヤ処理 (2.5D 切削)

これまで 2 次元 CAD で作図したデータと加工条件に示す Z 値から 2 次元の加工データを生成してきました。1 つの切削レイヤに 1 つの Z 値 (加工条件) を対応させていたわけです。ここでは、複数の切削レイヤを処理しそれぞれに Z 値を割り当てることで、簡易 2.5D の切削データを生成する方法を解説します。

#### 3.4.1 レイヤごとの Z 座標指定

図 3.14 は一見して解りづらい図面ですが、上が XY 平面図、下が側面図と考えて下さい。側面図の半円が切削したい曲線ですが、これを XY 平面におこすため、一定間隔 (この例では 1mm) で補助線を引き、さらに半円との交点で XY 平面用に縦の補助線を引いています。XY 平面図では側面図からの補助線と工具径を考慮しながら切削パスを作図しています。ただし、それぞれの切削パスがどの Z 座標かを示すため、図 3.15 のように複数<sup>\*1</sup> の切削レイヤに作図しています。高さ 30mm を 1mm 間隔で分割。徐々に外側へ広がっている様子が解ると思います。

作図が終われば、各レイヤに名前を割り当てましょう。特に切削レイヤには、それぞれが識別できるよう連番を振ります。

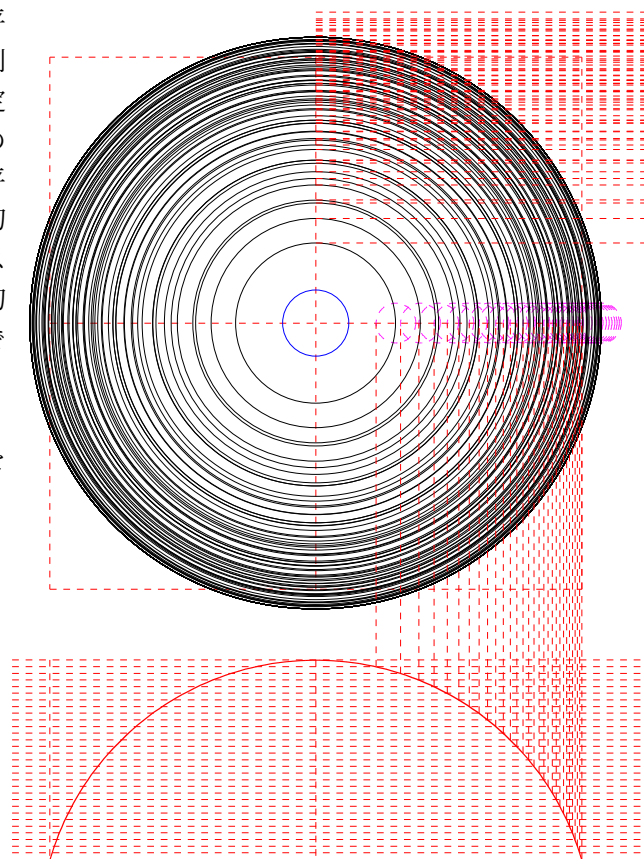


図 3.14 複数レイヤのサンプル図形

<sup>\*1</sup>Jw\_cad は 1 レイヤグループ 16 個のレイヤが割り当てられているので 2 つのレイヤグループを使用。30 個の切削レイヤと原点・補助線レイヤで計 32 個のレイヤを使用しています。

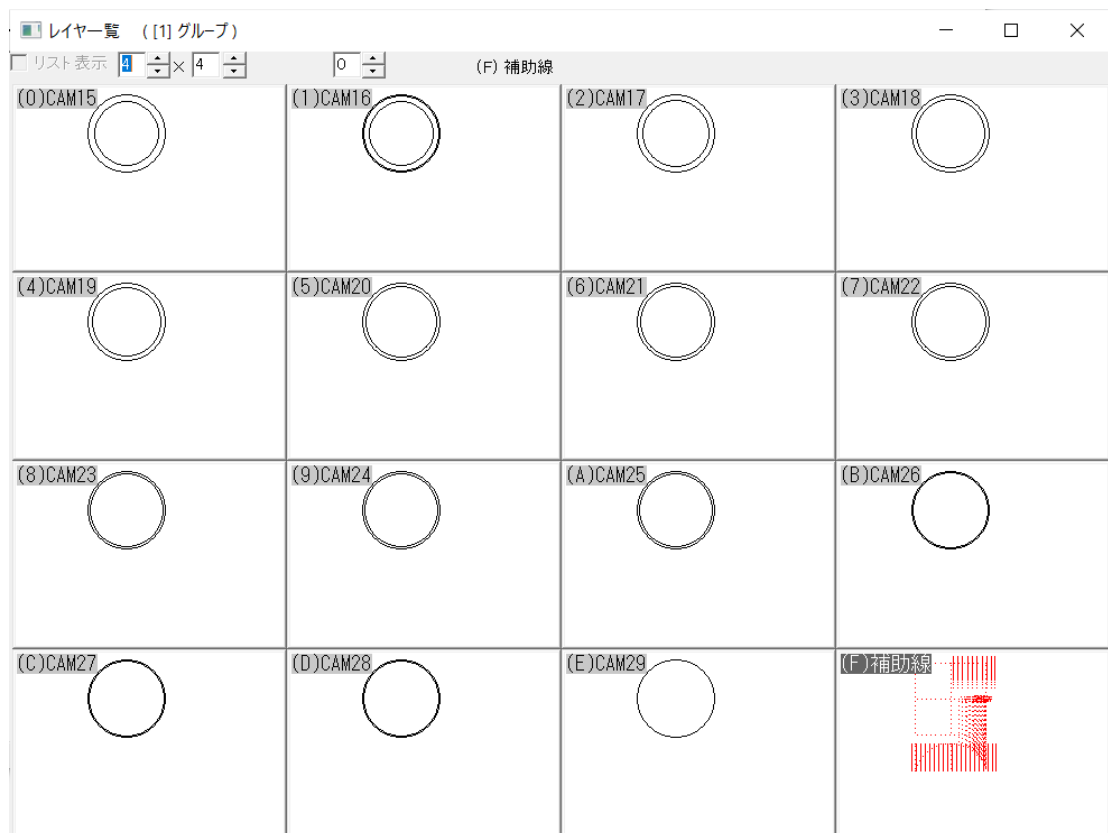
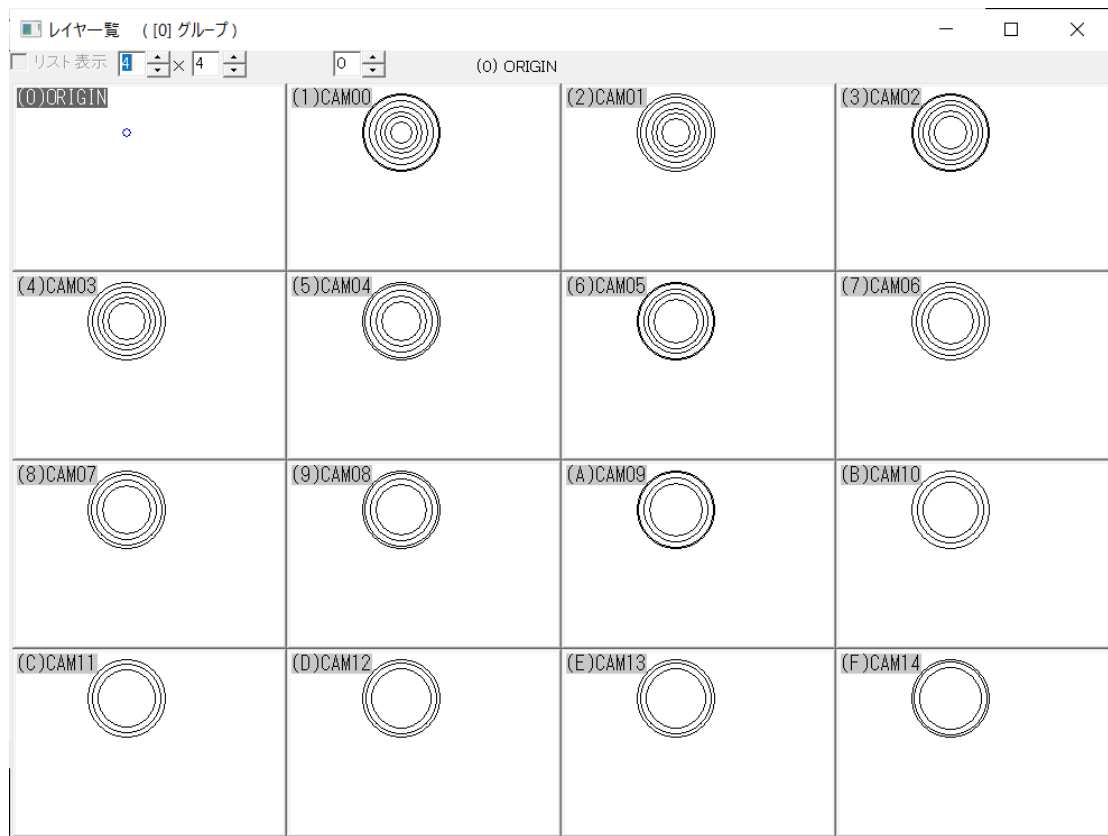


図 3.15 複数レイヤのレイヤー一覧

次にこの作図データを NCVC に読み込ませるわけですが、図 2.3 読み込みレイヤ設定では単一の切削レイヤしか処理できません。複数の切削レイヤを処理するには、図 3.16 のように正規表現<sup>\*2</sup>で認識文字列パターンを入力するか、従来互換を選択し、「レイヤ名 “CAM”」を“部分一致”で“認識”すると指定して下さい。図 3.16 の入力例は「文字列 “CAM” に続く数値 (0~9) を認識」と解釈されます。

読み込みレイヤが設定できれば、NCVC で CAD データを開きます。図 3.17 は「表示」レイヤをクリックし、複数レイヤが読み込まれていることを確認しています。



図 3.16 複数レイヤの読み込み設定

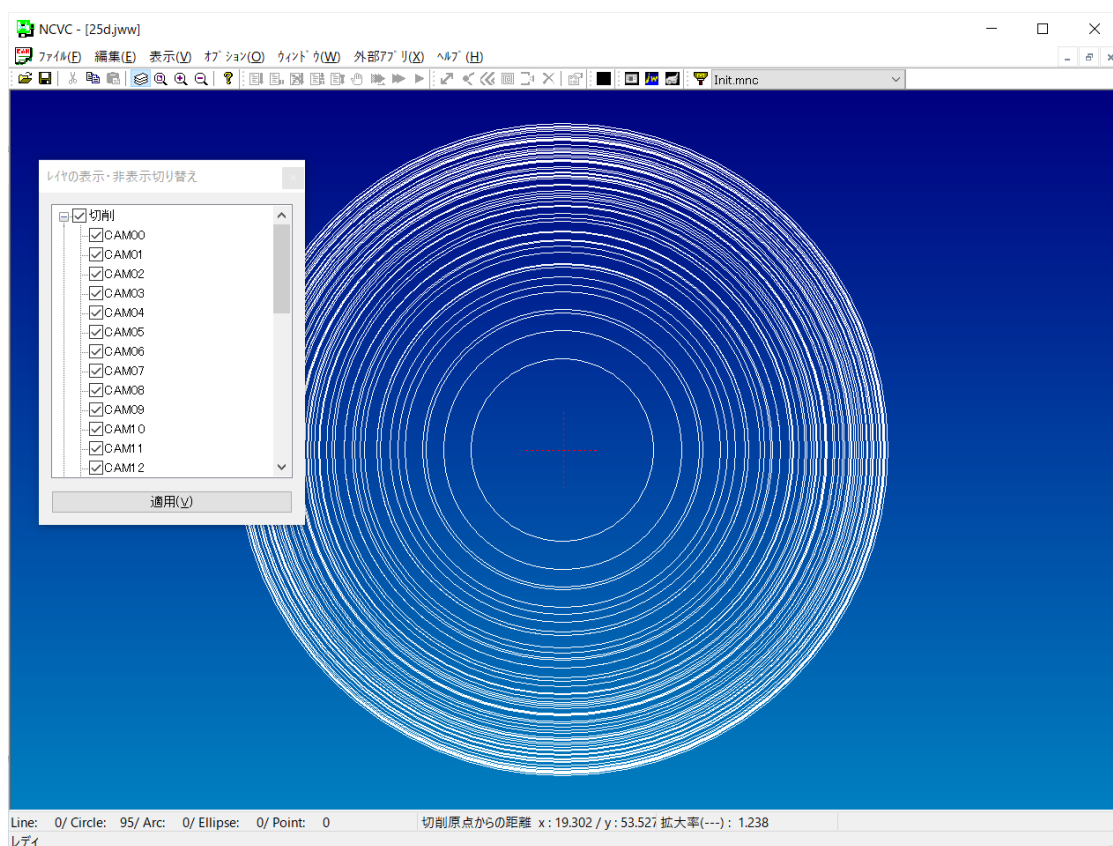


図 3.17 複数レイヤの読み込み

<sup>\*2</sup> [XX で始まる文字列] や [YY で終わる文字列] など、文字列のパターンマッチングを表現、コンピュータに指示するための表記のこと。NCVC に限った話ではない。正規表現の解説はそれだけで 1 冊の本ができるほど様々な表記法があるので、覚える必要はない。

複数の切削レイヤに Z 値を割り当てるには、**ファイル**》**NC データへの変換**》**レイヤごとの Z 座標指定**<sup>\*3</sup>をクリックし、図 3.18 のダイアログから行います。NC データの出力ファイル名と条件ファイルの指定は 2 章と同じ。[レイヤ名と Z 座標との関係] は次に説明する設定を保存するファイル (ncf ファイル) です。ここの編集ボタンをクリックすると図 3.19 のダイアログが表示されます。

レイヤ名には読み込んだ切削レイヤの一覧が列挙されます。[切り込みの自動設定] に [これ以降 “-1” mm ごとに] と入力し設定ボタンを押すと、CAM00～CAM29 まで順に-1mm づつ Z 座標を設定できます。この値は加工条件の切り込み<sup>\*4</sup>に置き換えられます。

OK を押すと図 3.18 に戻ります。図 3.19 で設定した Z 値を保存しておきたい場合は、[レイヤ名と Z 座標との関係] のファイル覧に適当な名前 (フルパス名) を入力します。

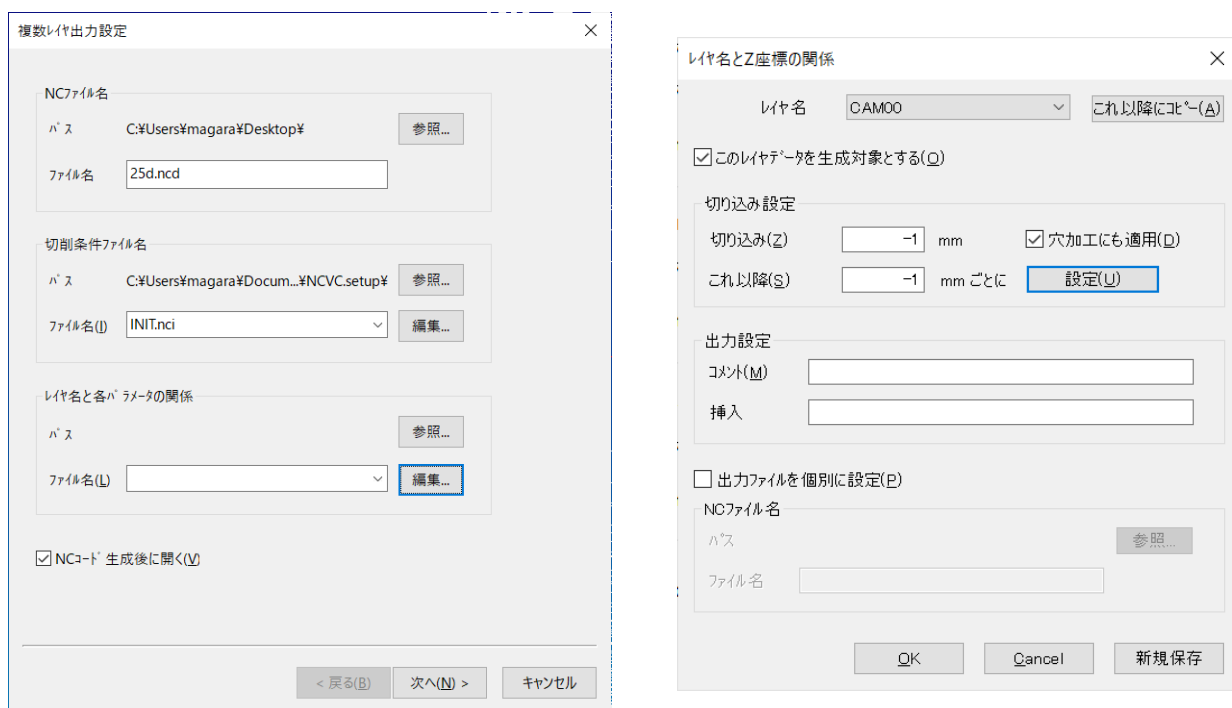


図 3.18 複数レイヤの設定

図 3.19 複数レイヤの Z 値設定

全ての設定が完了し図 3.18 で OK ボタンを押すと、図 3.20 の通り簡易 2.5D の切削データが生成できました。

<sup>\*3</sup> 複数の切削レイヤが読み込まれていないとこのメニューは選択できません

<sup>\*4</sup> 【2.3 加工条件の設定】を参照



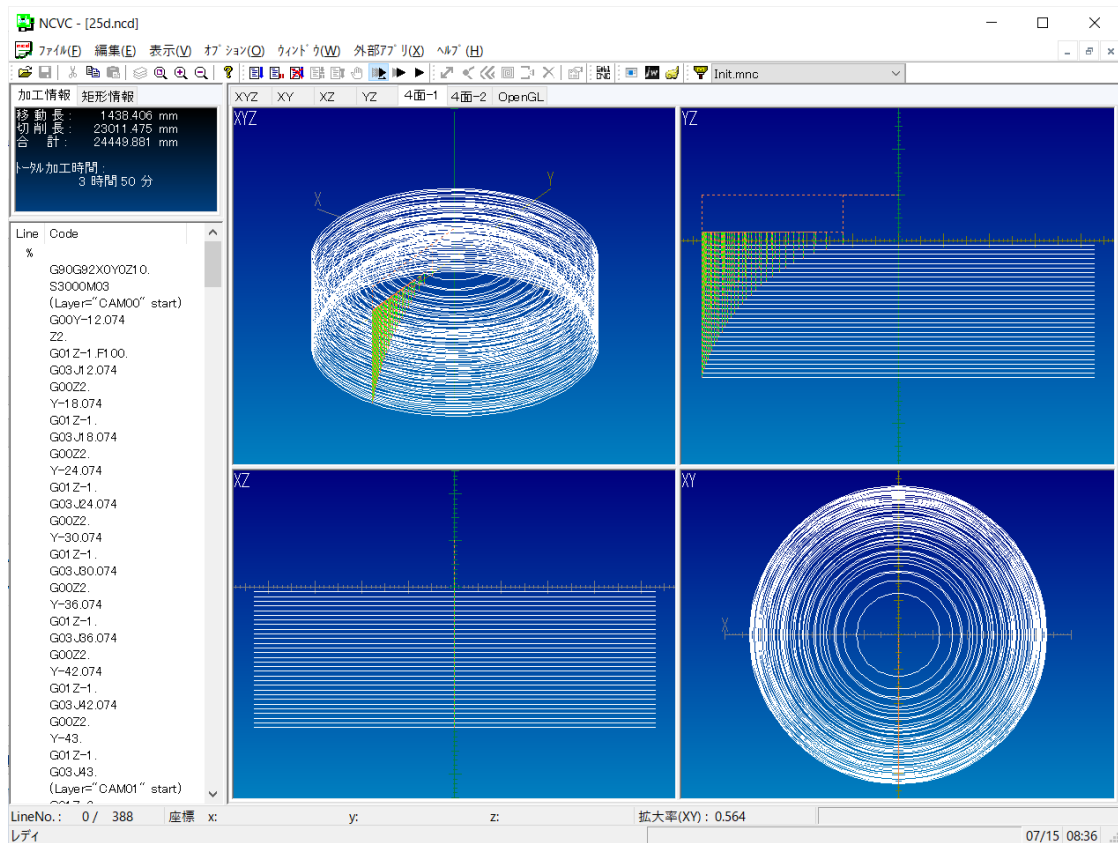


図 3.20 複数レイヤによる簡易 2.5D シミュレーション画面

### 3.4.2 レイヤごとの切削条件

【3.4.1 レイヤごとの Z 座標指定】では、その名の通り複数の切削レイヤに Z 値だけを強制的に設定しました。加工条件は 1 つだけなので、主軸回転数や送り速度等は全て同じです。これらをもレイヤごとに設定したい、例えば図 3.21 のような島加工で外側と内側の切削条件を変えたい場合、[レイヤごとの複数条件]を使います。

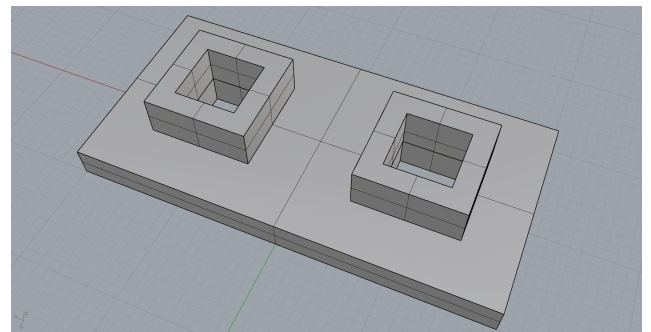


図 3.21 島加工切削例

ファイル ≫ NC データへの変換 ≫ レイヤごとの複数条件 をクリックすると図 3.22 のダイアログが表示されます。基本的には図 3.18 と同じですが、加工条件をレイヤごとに設定するため、切削条件ファイルは明細の方に移動しています。前節同様 [レイヤ名と Z 座標との関係] の編集ボタンを押すと図 3.23 のダイアログが表示されるので、各レイヤごとに条件ファイルを割り当てます。

各レイヤごとに Z 値を指定するか加工条件を指定するかの差で前節と同じです。シミュレーション画面も割愛します。



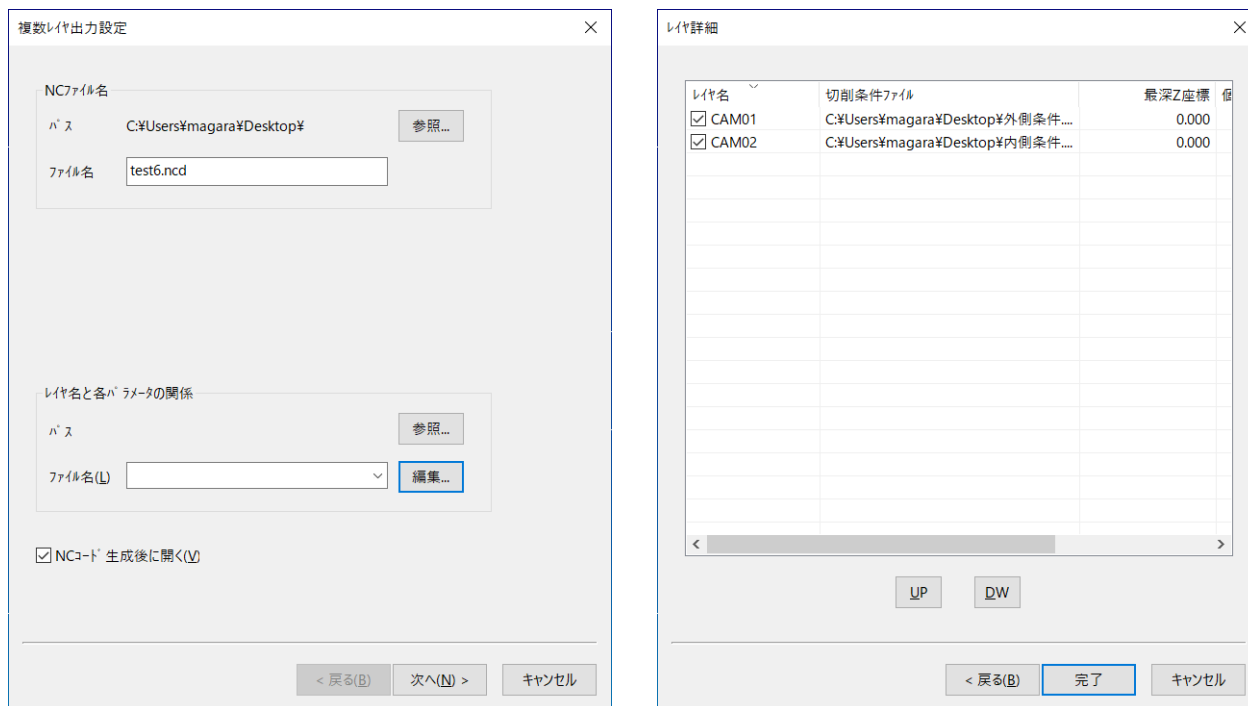


図 3.22 複数レイヤの Z 値設定

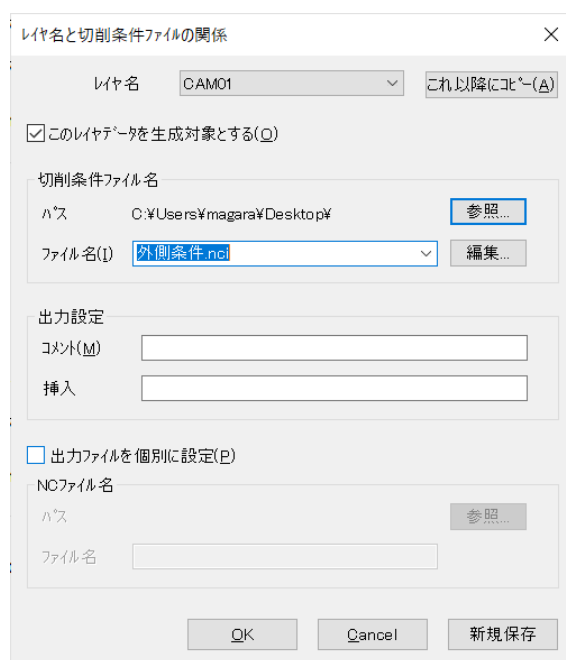


図 3.23 複数レイヤの加工条件割り当て

### 3.4.3 個別出力について

図 3.19 および図 3.23 の「出力ファイルを個別に設定」にチェックを入れると、各レイヤごとに出力するファイルを指示できます。

ここまでの【まとめ】

- 複数レイヤによる簡易 2.5D 切削であっても XZ または YZ 平面での指示はできない

## 4. 応用編

### 4.1 方向制御

結論から言うと、普通の方法では方向制御できません。例えば1つの座標に複数のオブジェクトが接続されている場合、NCVCが最初にどれを選択するかは内部で独自に決定されます。ここでは応用編の第一弾としてムリヤリ(?)方向制御を行う方法を解説します。

図4.1を見てください。中心の加工原点と左下の加工開始を示す円データ、五角形の線データがあります(それ以外は補助線)。一見して普通の作図、3.1節で解説した移動レイヤだけに見えますが、実は複数レイヤ処理との合わせ技で作図されています。図4.2は図4.1のCADデータを読み込み、レイヤ表示で切削レイヤの2つ目を非表示にした状態を表しています。

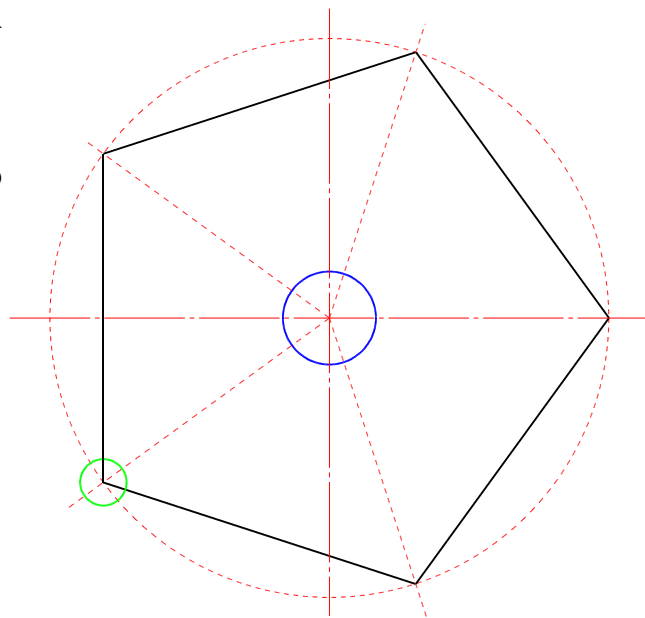


図 4.1 方向制御作図例

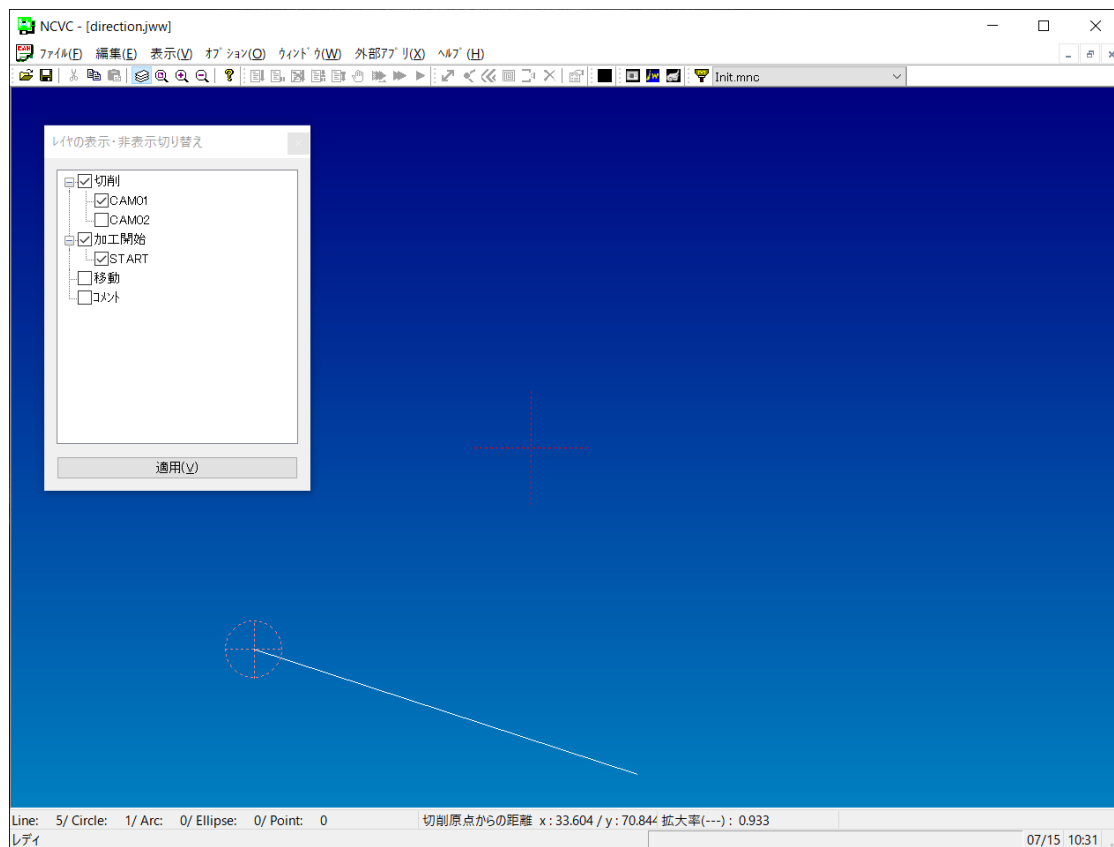


図 4.2 複数レイヤを読み込んだ状態

もうお解りですね。開始したい座標に複数のオブジェクトがあれば、それを違うレイヤに作図し、方向が明確に示せるようにすれば OK です。ただし、複数レイヤ処理を行う必要がありますので、単一条件での NC 生成では希望通り生成できません。【3.4 複数レイヤ処理】で解説した拡張 NC 生成、[レイヤごとの Z 座標指定] か [レイヤごとの切削条件] で処理する必要があります。どちらの場合でも同じ Z 値、または、同じ条件を割り当てることができます (図 4.3)。

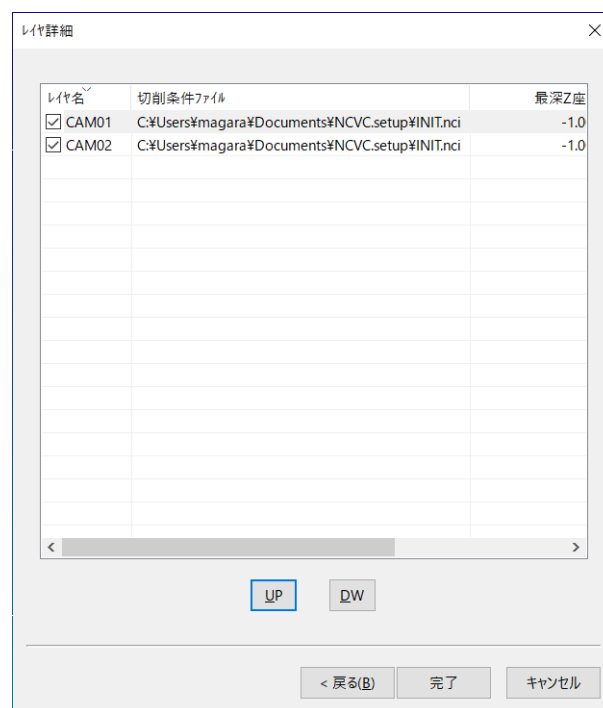


図 4.3 2つの切削レイヤに同じ加工条件を割り当て

もう1つ、裏技的というより、これぞムリヤリですが (自爆) 要するに1つの座標に2つ以上のオブジェクトを作図しなければ良いだけで、例えば図 4.4 のように矩形切削の方向を指示したい場合、影響の無い範囲内で他方の座標をずらしてやれば良いのです。前の切削領域からの最短距離なら NCVC が自動で検索します。明確に指示したい場合は【3.1 移動レイヤ】が使えます。

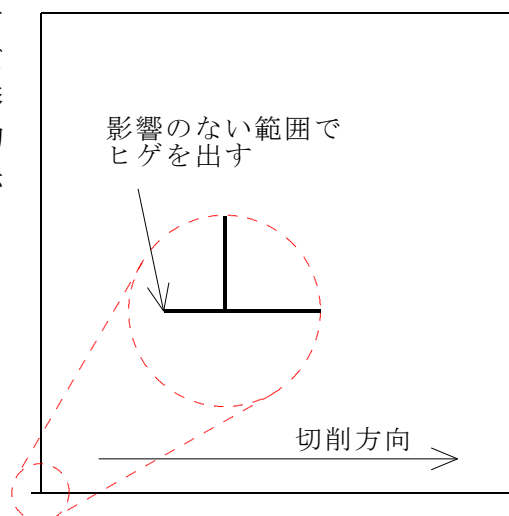


図 4.4 矩形切削例

どちらにしてもあまり格好の良い方法ではありませんね...

## 4.2 形状切削

方向制御と同じく形状切削についても NCVC では生成できません。一般的に CAD では完成された形状を作図しますが、実際の加工では工具の半径を考慮しなければなりません。特に島加工・ポケット加工については、CAD で作図した形状情報から NCVC が加工データを生成できません。NCVC は CAD のデータを工具の軌跡データと扱うのです。

もうカンの良い皆様ならお気づきだと思います。NCVC で形状切削の加工データを得るには、今度こそ CAD の力を存分に借りる必要があります。図 4.5 は【2.7 穴加工】のサンプルデータ (p.9) からポケット加工のデータを得るための作図例です。工具半径や必要な肉厚を考慮しつつ CAD の複線機能を駆使して作図しました。面倒そ

うですが慣れると意外と簡単に作図できます\*<sup>5</sup>。削りすぎにはくれぐれもご注意を（体験談）。削り残しも作図段階でチェックできませんが\*<sup>6</sup>，NCVC のシミュレーション画面もワイヤー表示なので，どのみちチェックできませんね（自爆 x 2）

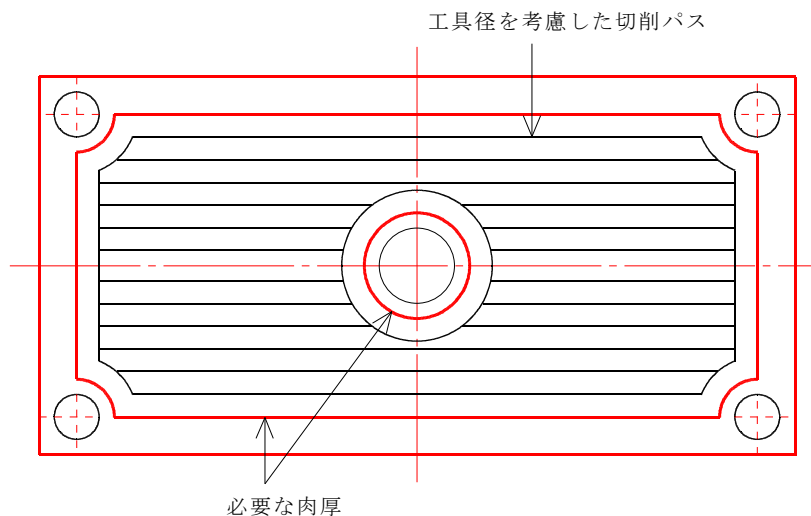


図 4.5 ポケット加工作図例

ここまでの【まとめ】

- 方向制御と形状切削は，CAD の作図次第
- スマートに指示するにはもう少し NCVC の進化を待つ必要がある

\*<sup>5</sup>ご使用の CAD の性能や習熟度次第ですが...

\*<sup>6</sup>Jw\_cad の場合，円の作図で半径を入力すれば，これから作図しようとする円が表示されます。これを工具に見立ててマウスを動かせば簡易チェッカ（？）。無論補助レイヤにて工具円を書きまくるのが確実デス。

## 5. パワーユーザ編

---

### 5.1 スクリプト作成のすすめ

#### 5.1.1 まえがき

awk(オーク), Perl(パール), Ruby(ルビー) これらの名前を聞いたことがあるでしょうか？この3つを知らなくても、Basic や C/C++なら聞いたことがあるかと思います。これら全てプログラミング言語で、先に挙げた3つの言語は特にテキストファイルの処理に優れた能力を発揮し、しかもそれらの処理を簡単に記述できるという特徴を持ったプログラミング言語です。

さて、「プログラミング言語」と聞いて尻込みしているアナタ！それは大きな誤解と難しいという先入観だと思います。冒頭でも述べた通り、テキストファイルの処理に優れた能力を発揮できる非常に良く考えられた言語で、目的にもよりますが極端な話し、たった1行でも十分実用可能なプログラムを作成することができます。3.4節で少しだけ登場した正規表現がサポートされているので強力な文字列のパターンマッチングが可能、つまり、テキストファイルの中から必要な情報を検索したり置換するといったことが簡単に記述できます。

これが NCVC の解説書とどう関係があるのか？答えは簡単、NCVC が吐く G コードがショボイから（自爆）... たしかにそれもありますが（汗）、例えば特殊なコード変換や埋め込み、サポートされていない機能の補間など、小粒だがピリリと気の利いた自分専用のツールを作ってみよう！というのがこの章の目的です。そう、G コード（NC ファイル）はテキストファイルです。プログラミング可能な一定のルールがあるなら、もうメモ帳でシコシコ手修正する必要はありません。

本解説書では、Basic や C/C++のプログラミング言語と区別するため、以後、スクリプト言語と称します。

#### 5.1.2 例題

とエラそうなことを書きましたが、ここで言語仕様等を書いてはキリが無いので、言語自体の解説は専門書にオマカセします。また、スクリプト言語といえば Perl が有名ですが、著者は awk ヲタク。例題もただの検索や置換だけならメモ帳でもできるので、ちょっと複雑な例題を用意しました。おかげで前述の「たった1行でも」というフレーズは実現できませんでした（ゴメンナサイ）。

では早速例題のテーマですが、『シーケンス番号の追加』とします。NCVC にもシーケンス番号を付加する生成オプションがありますが、コメント行などお構いなしに付加するため、この例題では次に示す行にシーケンス番号を付加しない仕様とします。

- O 番号（プログラム番号）, “%”, カッコ “(” で始まる行
- 空行

awk スクリプトで書くと以下のようなリストになります。これをメモ帳などで入力し、ファイルとして保存しておきましょう。一見して解る通り、Basic や C/C++では必要なファイルのオープンや1行ずつ読み込んでループ等の処理は必要ありません。awk で必要なのは条件パターンとその条件に対する処理（アクション）だけです。

## ソースコード 5.1 num.awk

```
BEGIN {          # 最初に実行されるブロック
    num=1000;      # シーケンス番号の初期値
    add=5;         # 増分値
}
/^0|^\(|^%| || length($0)==0 { # シーケンス番号を付加しない条件
    print $0        # 1行そのまま出力（$0は入力1行全てを表す）
    next           # 次の処理（入力）へ
}
{
    printf("N%04d", num);    # 条件パターン無しブロック（上記条件以外）
    print $0;               # N に続く4桁の数字を出力（改行無し）
    num += add;             # 1行そのまま出力
                           # 増分の加算
}
```

### 5.1.3 実行環境

もともとこれらスクリプト言語は UNIX 文化で培われたものです。UNIX にはほぼ標準で用意されていますが、Windows で実行するには、それぞれの処理系をインストールする必要があります。awk なら GNUawk(gawk) が有名ですのでインストールしておいて下さい<sup>\*7</sup>。

実行環境が整えば、Windows スタートメニューからコマンドプロンプト<sup>\*8</sup>を開き、以下のように入力します。

```
gawk -f ScriptName.awk <Input-file
```

希望通りに変換された G コードが表示されましたか？必要なら

```
gawk -f ScriptName.awk <Input-file >Output-file
```

とすることで、*Output-file* に示すファイルに書き出されます。

### 5.1.4 あとがき

これくらいの例題だけでは、自分がしたい処理を書くことは正直難しいと思います。けれども、スクリプト言語は G コードの置換処理だけでなくその他事務処理等にも応用できます。得られるものは大きいのでぜひチャレンジしてみてください。

新たに挑戦する場合は先人たちの知恵が大いに参考になると思います。NCVC の Web ページにはたくさんのスクリプトが掲載されていますのでご参考に。また、NCVC からスクリプトが簡単に実行できるラッパーもあります。Scriptorium のドキュメントも併せてご参考に。

<sup>\*7</sup>大規模なシステムではないので、展開して実行ファイル（EXE）をパスの通ったフォルダに移動させる、もしくはスクリプトと同じ場所に置くだけで良い

<sup>\*8</sup>Windows のバージョンによって違います。DOS 窓のこと