

TCP/IP を用いた並列分散環境の利用に関する研究 -オフセット面生成への応用-

函館高専 ○中寺 雄介 近藤 司

要旨

複数の PC をインターネット回線に接続し、それぞれの PC に Windows 用通信ライブラリ「Winsock」を組み込んだソフトウェアを起動することにより簡便に並列分散環境を構築することができる。本研究では、工具経路生成に必要なオフセット面生成アルゴリズム「逆オフセット法」の並列分散性に着目し、2台 PC で構築した分散環境下で実行した。その有効性と問題点を述べる。

1. 緒論

曲面を含む立体形状の NC 加工において、種々の工具形状に対して工具干渉処理を必要精度で解決する方法としてオフセット面生成アルゴリズム「逆オフセット法」がある。この逆オフセット法では、生成精度の向上とオフセット面データ量および加工時間の関係には指数関数的に増加する課題がある。

本研究では「逆オフセットアルゴリズム」に存在する処理の並列性・分散可能性に着目し、その高速化処理を試みる。そのため Windows 用通信ライブラリ「Winsock」を用いた並列分散環境を構築し、実験した。

2. TCP/IP

TCP/IP とはネットワーク上でコンピュータ(PC)同士が通信を行うためのプロトコルを整理したモデルであり、現在最もよく利用されているモデルである²⁾。それを用いたプログラミングでは、ソケット (Windows では Winsock) というものを使って、それに読み書きすることで相手の PC とデータ通信を行うことができる。本研究では VisualC++6.0 を用いて2台の PC をサーバとクライアントとして接続型同期通信をおこなうことで通信を行った。

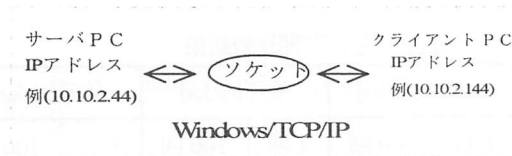


図1 TCP/IP を用いたソケット通信

3. 逆オフセット法の演算処理および並列性

図2に逆オフセット法におけるオフセット面生成方法およびその処理の並列性を示した。逆オフセット法では、加工面は点群として定義されており細くなるほど加工面の生成精度は向上する。要求面を構成する点のオフセット面が工具軸(Z軸)に対する逆向き工具形状であることを利用し、加工面のオフセット面はその点のオフセット面の包絡面により生成する。オフセット面の記憶領域としてXY平面において加工面を包含する領域に対して格子空間を導入する。この格子ピッチが小さいほどオフセット面の生成精度が向上する。格子にはオフセット面のZ座標値を記憶するようにし、はじめに最低値を記憶するように初期化する。加工面の個々の点に対して、逆向き工具の基準点(工

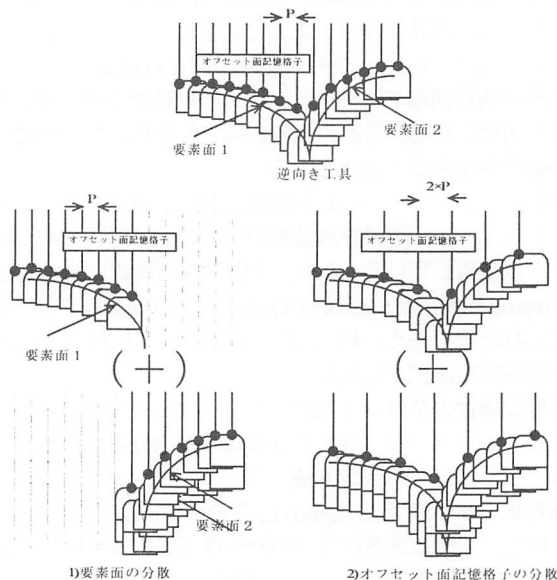


図2 逆オフセット操作の分散処理

具中心点)を配置する。次に逆向き工具形状と格子点との交点を計算する。すべての点に対して交点計算を行い、そのZ方向の最高値を更新し記憶する(包絡演算)。工具移動に関して、工具移動点(CLデータ)は空間格子データを用いた補間演算により求めることができる。このように逆オフセット法では最終的にすべての加工点に対して逆向き工具形状と格子との交点計算を終了すればよい。加工点に関する計算順序および空間格子に関する計算順序を考慮する必要はない。ここに逆オフセット法の並列処理性がある。すなわち、複数の計算環境がある場合、要求形状を分散して処理することが可能であり、また、格子空間も細かさを調整することで分散することが可能となる。

4. 並列分散処理法

逆オフセット法はその性質上、形状の最小構成要素が点であるという考え方に基づいている。したがってその計算において各点毎の計算となるため加工面のデータを順番に計算してオフセット面(点群)を算出する必要がない。本研究ではその性質を利用して二台のPCによる並列分散処理を試みた。図3に概要を示す。加工面を構成する点群を二つに分割し、二台のPCへ分配する。各PCにおいて同一格子空間を設定し、逆オフセット計算を行う。その際、

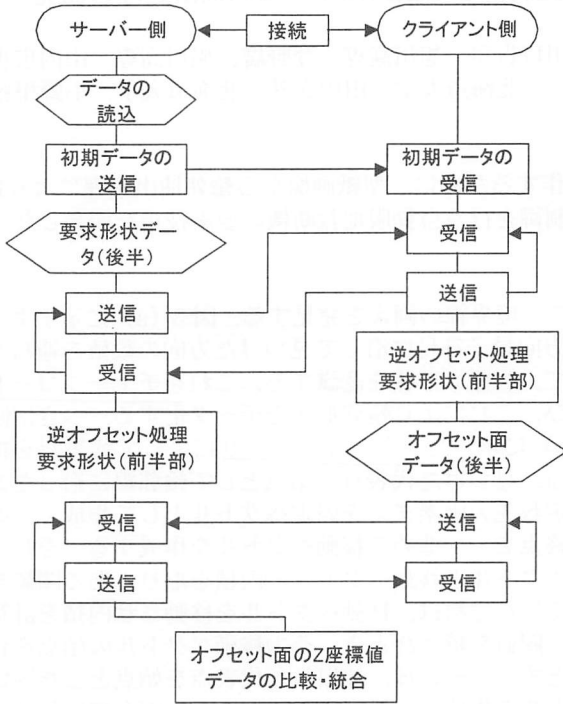


図3 フローチャート

加工面座標間を任意の数に分割し、交点計算回数を増加させて加工精度を高める。各 PC からオフセット面データを集配し統合する。これらの処理では、逆オフセット処理は基本的に負荷が均等な並列処理となり、各 PC への分配と集配の通信処理時間が単一 PC における処理に対する余時間である。

5. Winsock を用いた並列分散処理プログラム

分散処理環境は Windows 用通信ライブラリ「Winsock」を用いて二台の PC 間の通信システムを構築した。Winsock は Windows で TCP/IP の機能を利用したソフトウェアの開発するための API である。プログラムのフローチャートを図3に作成した。二台の PC を一方はサーバー側 PC、もう一方はクライアント側 PC とする。サーバー側 PC はクライアント側 PC と接続し、データの読込を行う。次に加工面（点群）の座標データを前半部と後半部に分け、クライアント側 PC に逆オフセット処理の計算条件（始点、格子点間隔、格子数、工具情報、後半部の加工面データ）を送信する。サーバー側 PC、クライアント側 PC はともに要求形状データより逆オフセット法に基づいて格子空間上で格子交点の Z 座標値を計算しオフセット面を生成する。（この際計算精度をより高めるために加工面の座標データ間を分割・計算している。）計算が終了するとクライアント側 PC はオフセット面 Z 座標値データをサーバー側 PC に送り返し、サーバー側 PC は前半部と後半部のオフセット面 Z 座標値データを統合し、最終的なオフセット面を算出する。

6. 実験結果

ある加工面のデータを用いて、一台の PC で直列に計算した場合と、二台の PC で並列分散処理した場合（図4）の計算所要時間を比較した。それぞれ座標データ間を 100、200、300 と座標間分割数を設定し、異なる計算回数でのデータを測定した。得られた結果より図5にグラフを示す。「単独計算A」は単独で計算を行った場合、「分散計算」は二台の PC に分散させて計算した場合、「単独計算B」は一

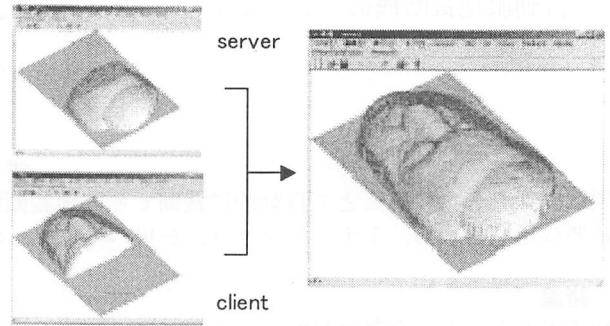


図4 実験結果

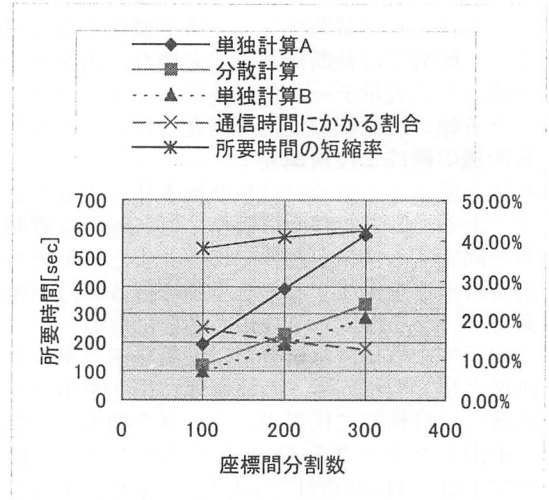


図5 所要時間と座標間分割数の関係

台の PC で加工面のデータの半分（分散計算において一台あたりの PC が計算する量）を計算した場合である。分散計算と単独計算Bは一台あたりの計算量が同じであるので計算時間は等しい。したがって、ここで見られる計算の所要時間の差は通信に要する時間であると思われる。また、グラフからわかるように座標間分割数が多くなる、つまり逆オフセット法による交点計算回数が増加することによって計算に要する時間が増加するが、データの通信量は変わらないので通信時間にかかる割合は減少してゆく。また、座標間の分割数が増えてゆくと共に所要時間の短縮率も増加することが確認できた。

7. 結論

1. Winsock を用いた並列分散処理環境を構築し、逆オフセット法の分散処理を行うことができた。
2. 精度を高めることで増加する逆オフセット法の交点計算回数を、分散処理することによって計算に要する時間を 40～43%短縮できた。
3. グラフより逆オフセット法の交点計算回数が増加すると、所要時間の減少率も増加することが確認できた。したがってデータが大量であるほどこの手法の効果は高くなる。

参考文献

- 1) 岸浪 他： 逆オフセット法に基づく形状処理技術開発、型技術、Vol.8 No.13,p29-32,1993
- 2) 糸井康隆：猫でもわかるネットワークプログラミング、ソフトバンクパブリッシング（社）