

RC-002

## GPUを用いた2次元定常熱伝導方程式の数値計算高速化

Fast Numerical Calculation of Steady State Heat Equation by Graphics Processing Unit

高田 直樹†  
Naoki Takada滝沢 努†  
Tsutomu Takizawa増田 信之†  
Nobuyuki Masuda伊藤 智義†  
Tomoyoshi Ito

## 1. まえがき

近年、コンピュータグラフィックス(CG)処理用のプロセッサであるGPU(Graphics Processing Unit)の性能は著しく向上している。GPUは内部にシェーディング処理を行うパイプラインを持ち並列処理が可能である。GPUの演算性能及びコストパフォーマンスはCPUに比べ格段に優れている。GPUに搭載されたシェーダはプログラム可能であり、単精度浮動小数点演算を行うことができる。GPUを用いてCG処理以外の汎用的な数値計算を高速化させる研究は、GPGPU(General Purpose on GPU)と呼ばれており最近盛んに行われている[1][2][3]。

2次元定常熱伝導方程式の数値計算を行う専用計算機のアーキテクチャが提案されている[4]。FPGAに実装し、専用計算機の妥当性が示されている。このアーキテクチャを基に、パイプライン数を増やせば計算高速化されることになる。

本研究では、GPUを用いて2次元定常熱伝導方程式の数値計算を実装し並列計算を行った。GPUとしてNVIDIA社製のGeforce8800 GTXを使用した。CG処理のテクスチャを用いて2次元空間に対する温度配列をGPUに割り当て並列計算を行った。開発環境として、DirectX 10と上位レベルシェーダ言語(HLSL)を使用した。その結果、CPUのみで計算した場合に比べ、約15倍の計算高速化が達成された。

## 2. 2次元定常熱伝導方程式の数値解法

2次元熱伝導方程式は次式となる。

$$\rho c \frac{\partial T}{\partial t} = \lambda \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + Q \quad (1)$$

ここで、 $T$ は温度、 $\lambda$ は熱伝導率、 $\rho$ は密度、 $c$ は比熱、 $Q$ は外部から与えられた熱量を示す。

定常な場合、温度の時間変化は一定となることから、式(1)は次式となる。

$$\lambda \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + Q = 0 \quad (2)$$

式(2)に中心差分を適用し、 $x, y$ 方向の空間離散間隔を $h$ とすると、2次元定常熱伝導方程式の差分式は次式となる。

$$T_{i,j}^{n+1} = \frac{1}{4} \left( T_{i+1,j}^n + T_{i-1,j}^n + T_{i,j+1}^n + T_{i,j-1}^n + \frac{Qh^2}{\lambda} \right) \quad (3)$$

式(3)において、求める温度は $T_{i,j}$ である。数値解を求めるのに次式を用いる。

$$T_{i,j}^{n+1} = \frac{1}{4} \left( T_{i+1,j}^n + T_{i-1,j}^n + T_{i,j+1}^n + T_{i,j-1}^n + \frac{Qh^2}{\lambda} \right) \quad (4)$$

式(4)の差分計算を計算領域の全離散点について行う。収束解を求めるために、各離散点において、式(4)の差分計算を繰り返す。 $n$ は、式(4)の差分計算の繰り返し回数を示す。これにより、収束した解が定常解となる。

## 3. GPUを用いた定常熱伝導方程式の数値計算

## 3.1 定常熱伝導方程式の数値計算の実装

GPUを用いた定常熱伝導方程式の数値計算フローチャートを図1に示す。CPUにおいて、計算領域の温度配列をCG処理で使用するテクスチャに割り当てる。テクスチャをGPUに転送し、 $n$ 回目の式(4)の差分計算で求められた温度配列データは、GPUを搭載したグラフィックスボード上のビデオメモリのテクスチャバッファに格納される。シェーダ・プログラムにより、テクスチャバッファに格納された温度配列を用いて $n+1$ 回目の式(4)の差分計算がGPU内で行われる。この計算結果である温度配列データをテクスチャを用いてCPUに戻す。これを繰り返し熱伝導方程式の定常解を求める。

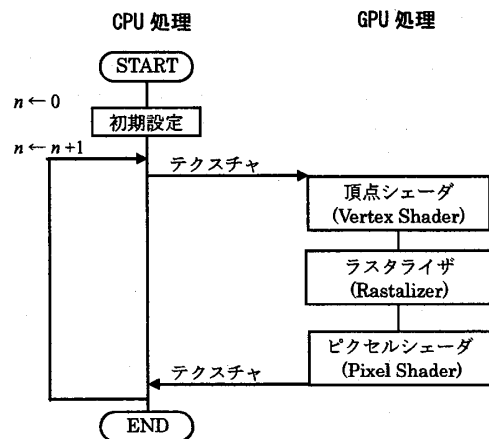


図1 GPUを用いた本数値計算のフローチャート

## 3.2 シェーダ・プログラミング

GPUに2次元定常熱伝導方程式の差分計算(式(4))を実装する手順を図2に示す。最初に、頂点シェーダにおいて、4つの頂点をスクリーンの端にくるようにセットする。こ

†湘北短期大学情報メディア学科

‡千葉大学大学院工学研究科

の頂点は、2次元の計算領域の頂点となる。次に、ラスライザの自動補間機能を用いて、四角形ポリゴンを描画する命令を与え、その頂点に囲まれたピクセルが生成される。その後、ピクセルシェーダにおいて、補間されたテクスチャ座標データを、熱伝導数値計算で用いる温度配列の座標データとして利用し、式(4)の差分計算を行う。

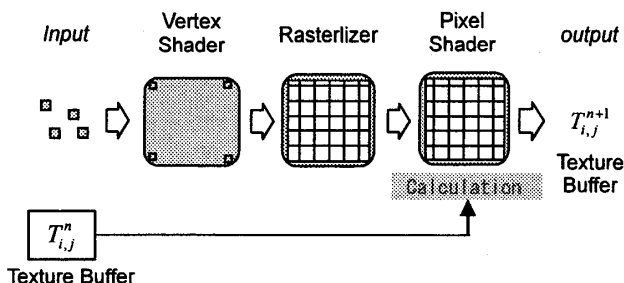


図2 GPUにおける定常熱伝導方程式の差分計算の実装

ピクセルシェーダのプログラムは、次のようになる。

```
float4 PS0 ( VS_OUTPUT0 In ) : COLOR
{
    float4 T;
    float h=1.0f/512, H=h*h;
    float4 t1 = tex2D(temp1_Samp, In.Tex0 + float2(invTexSize, 0.0f));
    float4 t2 = tex2D(temp1_Samp, In.Tex0 + float2(-invTexSize, 0.0f));
    float4 t3 = tex2D(temp1_Samp, In.Tex0 + float2(0.0f, invTexSize));
    float4 t4 = tex2D(temp1_Samp, In.Tex0 + float2(0.0f, -invTexSize));
    if (In.Tex0.x == 0.5 && In.Tex0.y == 0.5){
        T.r = 1.0f;
        T.b = 1.0f;
        T.g = 1.0f;
    }else{
        T = float4((t1.xyz + t2.xyz + t3.xyz + t4.xyz)/4 + Q*H/4, 1.0);
    }
    return T;
}
```

#### 4. 数値計算モデル

数値計算モデルを図3に示す。計算領域を  $512 \times 512$  とした。その中心の座標(256, 256)に温度一定な熱源  $Q=1.0$  を配置した。計算領域の終端の境界条件にディリクレ条件 (温度:  $T=0$ ) を与えた。式(4)を図3の全計算領域に適用し、この差分計算を繰り返すことにより、定常解を求めた。

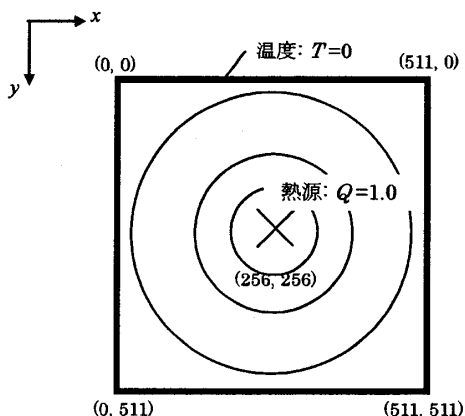


図3 数値計算モデル

#### 5. 結果

計算領域  $512 \times 512$  において、式(4)の差分計算を10万回繰り返し、その計算時間を測定した。CPU 単独での計算時間 (SSE 命令を使用) と、本手法による GPU を用いた計算時間を比較した。その結果を表1に示す。

CPU: Intel Pentium 4 3.6-GHz, Memory: 2GB
OS: Windows XP
Driver: NVIDIA 169.21
Compiler: Microsoft Visual Studio 2005 (option -O2)
GPU: NVIDIA Geforce 8800 GTX
(Stream processors:128, Shader Clock:1.35-GHz)
Graphics API: Directx10, Shading language: HLSL

表1 CPU 計算と GPU 計算の比較

CPU (秒)	GPU (秒)	速度比
204	13.5	15.1

本手法による GPU を用いた計算では、CPU 単独で計算した場合に比べ、約 15 倍計算高速化された。10 万回差分計算を繰り返したあとの CPU 単独で計算した結果と GPU を用いて計算した結果を比較したところ、GPU を用いた計算は、単精度を保持していることが確認された。

#### 6. まとめ

CG 処理を用いて、2次元熱伝導方程式の数値解法を GPU に実装した。その結果、CPU 単独の計算と比べ、約 15 倍の計算高速化を実現することができ本手法の有効性が示された。また、CPU 単独の計算結果と比較したところ、本手法は単精度を保持していることが確認された。これにより、本手法による GPU を用いた定常熱伝導方程式の数値計算は、単に高速化するだけでなく、科学技術計算として利用できることが示された。

#### 謝辞

本研究の一部は、文部科学省科学研究費補助金若手研究 (B) (課題番号 20700053) による。

#### 参考文献

- [1] N. Masuda, T. Ito, T. Tanaka, A. Shiraki and T. Sugie, "Computer generated holography using a graphics processing unit", Optics Express, Vol.14, No.2, pp.587-592, 2006.
- [2] 下馬場朋祿, 伊藤智義, 杉江崇繁, 増田信之, 阿部幸男, 白木厚司, 市橋保之, 高田直樹, "統合型シェーダを搭載した GPU によるフレネル回折積分の高速化", 信学論(D), Vol. J90-D, No. 9, pp. 2656-2658, 2007.
- [3] N. Takada, N. Masuda, T. Tanaka, Y. Abe and T. Ito, "A GPU implementation of the 2-D finite-difference time-domain code using High Level Shader Language", Appl. Comput. Electromagn. Soc. J., in print.
- [4] 佐竹信一, 廣井義明, 増田信之, 伊藤智義, "プログラミング可能な LSI による二次元定常熱伝導シミュレータ", Thermal Science & Engineering, Vol.15, No.2, pp.85-90, 2007.