2-16

GPGPUを用いた線形システムの過渡解析に関する研究

日大生産工（学部）　○木村　弘
日大生産工　黒岩　孝
日大生産工　松原　三人

1. はじめに

最近、3D-CGなどの画像処理に特化した画像処理演算装置（Graphics Processing Unit：以下GPU）を、より汎用的な演算処理装置として用いる、GPGPU（General Purpose computing on GPUs）という手法が注目されている[1],[2]。GPGPUは、そのメモリの管理や並列演算などの特徴を活かし、例えば行列の高速演算に適している事が知られているため[3]、様々な工学的問題を解くのに有用であると考えられる。

そこで本研究では、GPGPUを用いた線形システムの解析について検討を行う。具体的には、複数のローパスフィルタにより構成した1入力1出力の線形システムについて、GPGPUを用いて過渡応答を計算した場合の有用性について検討を行う。

2. 解析法

図1に、線形システムとして解析する、RC2段のローパスフィルタを示す。先ず、入出力信号を電圧とし、回路方程式より、図1に示した線形システムの数学モデルを以下の状態方程式で表す[4]。

$$\begin{cases}\dot{\mathbf{x}}(t)=\mathbf{A}\mathbf{x}(t)+\mathbf{b}u(t)\\ y(t)=\mathbf{c}\mathbf{x}(t)\end{cases} \quad \cdots\cdots (1)$$

ただし、状態量は$\mathbf{x}(t)=[v_1(t), v_2(t)]^{\mathrm{T}}$とし、2次の行列$\mathbf{A}$及びベクトル$\mathbf{b}$の各要素は、

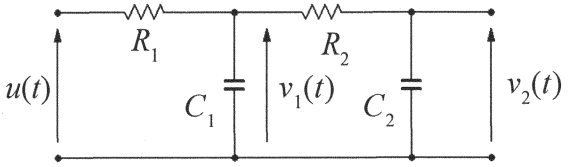

図1　RC2段のローパスフィルタ

R_1, R_2並びにC_1, C_2で表される。また、ベクトル$\mathbf{c}$は、$\mathbf{c}=[0,1]^{\mathrm{T}}$とする。$u(t)$を単位ステップ関数とした場合、(1)式のLaplace変換により、$\mathbf{x}(t)$の厳密解は次式で与えられる。

$$\mathbf{x}(t)=\begin{bmatrix}u(t)-e^{-t}\\ 1-(1+t)e^{-t}\end{bmatrix} \quad \cdots\cdots (2)$$

次に、行列指数関数$e^{\mathbf{A}t}$を用い、(1)式を次式の様に離散化する。

$$\begin{cases}\mathbf{x}[k+1]=\mathbf{A}_d\mathbf{x}[k]+\mathbf{b}_d u[k]\\ y[k]=\mathbf{c}_d\mathbf{x}[k]\end{cases} \quad \cdots\cdots (3)$$

ここで、$\mathbf{A}_d$及び$\mathbf{b}_d$は次式より求める。

$$\mathbf{A}_d=e^{\mathbf{A}\mathrm{T}}=\sum_{i=0}^{\mathrm{N}}\frac{\mathbf{A}^i\mathrm{T}^i}{i!} \quad \cdots\cdots (4)$$

$$\mathbf{b}_d=\int_0^T e^{\mathbf{A}\tau}\mathbf{b}_d\,\tau=\left(\sum_{\mathrm{i}=0}^{\mathrm{N}}\frac{1}{(\mathrm{i}+1)!}\mathbf{A}^i\mathrm{T}^{\mathrm{i}+1}\right)\mathbf{b} \quad \cdots\cdots (5)$$

ただしTは標本化周期であり、Nは打ち切り項数を表す。(3)式に対して初期値$u[0]$及び$\mathbf{x}[0]$を与える事により、離散的な状態量$\mathbf{x}[k]=\mathbf{x}(kT)$ $(k=0,1,2,\cdots)$が求まる。

Study on the transient analysis of linear system
by using General-Purpose computation on Graphics Processing Units

Hiromu KIMURA , Takashi KUROIWA and Mitsuhito MATSUBARA

3. 結果

以下では、R_1=R_2=1[Ω]及びC_1=C_2=1[F]として解析を行う。ここでGPGPUは、nvidia社のGeForce GTX295(クロック周波数2[GHz]・メモリ1.8[GB])とし、開発環境にはCUDAを用いた[5]。また、GPGPUと比較するため、Intel社のCPUであるCore i7 975(クロック周波数3.33[GHz]・メモリ12[GB])を用いた。

図2は、T=0.1[sec]とした場合の過渡応答を示す。同図中の実線が厳密解を表し、プロットが状態量$\mathbf{x}[k]$を示す。ただし、打ち切り項数はN=100とした。いずれの場合も、厳密解と良い一致が見られる。

図3は、T=0.1及び0.01とした場合のGPGPU並びにCPUの計算時間を示す。同図より、いずれの場合も指数関数的に計算時間が増加しており、顕著な違いは見られなかった。

一方、表1は、GPGPU並びにCPUを用いて、単位行列の乗算を100回行った結果である。行列の次数が大きくなると、CPUの計算時間がGPGPUと桁違いに増える事が分かる。よって、分布定数系の等価回路の様に、ローパスフィルタの段数が多い場合は、GPGPUの効果がより顕著になると思われるが、これに関する検討は今後の課題である。

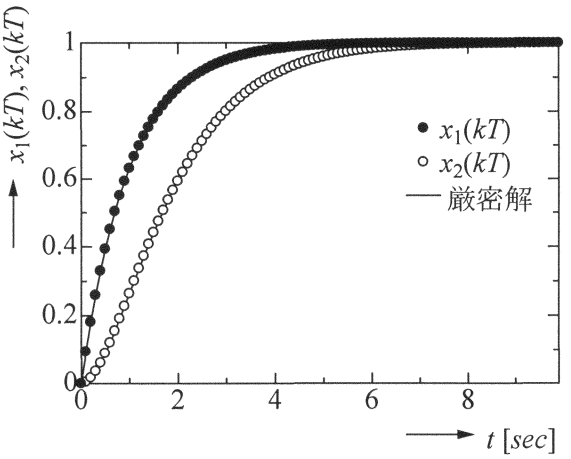

図2　状態量の過渡応答

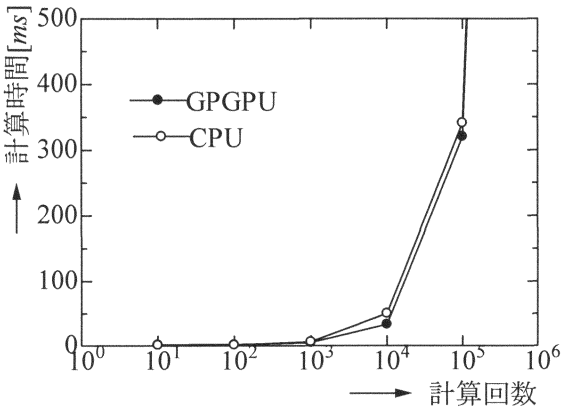

図3　計算時間の比較(過渡応答)

表1　計算時間の比較(行列の乗算)

次数	GPGPU[ms]	CPU[ms]
10	0.023	0.098
100	0.024	9.999
1000	0.034	557.030
10000	0.080	55103.15

4. まとめ

GPGPUを用いて線形システムの過渡応答を解析した結果、厳密解と良い一致が見られる事がわかった。また、システムの次数が大きい場合には、CPUに比べ、計算時間がかなり短くなるという知見が得られた。

参考文献

[1] 田村他:"GPGPUによる並列処理", アスキードットテクノロジーズ,Vol.12,pp.24-85(2009)

[2] 浅原他:"GPGPUプログラミング", アスキードットテクノロジーズ,Vol.8,pp.22-81(2010)

[3] D.B.Kirk et al.:Programming Massively Parallel Processors, Morgan Kaufman Publ.(2010)

[4] 美多勉:ディジタル制御理論, 昭晃堂(1984)

[5] 小山田他:CUDA高速GPUプログラミング入門, 秀和システム(2010)