

1. はじめに

本書は、Mapleを東京工業大学学術国際情報センターのTSUBAMEで利用する方法について説明しています。

また、TSUBAMEを利用するにあたっては、「TSUBAME利用の手引き」もご覧下さい。サーバの利用環境や注意事項などが詳細に記述されていますので、よく読んでください。

サイバネットシステム株式会社では Mapleに関するWebページを公開しています。次のアドレスを参照してください。

<http://www.cybernet.co.jp/maple/>

1.1. 利用できるバージョン

TSUBAME3で利用可能な最新バージョンについてはTSUBAME計算サービスWebサイトの [アプリケーション](#) ページをご確認下さい。

研究に支障がない限り、バグ修正の入っている最新版をご利用下さい。

1.1.1. バージョンの切り替え

module コマンドでmodule ファイルを読み込むことでバージョンの切り替えが可能です。

[TSUBAME3.0利用の手引き: 利用環境の切換え方法](#)で切り替えが可能です。

1.2. 概要

1980年にカナダ・ウォータールー大学で生まれた数式処理システム『Maple』は、数式計算機能をコアテクノロジーとして持った統合技術計算・技術文書作成環境です。電卓代わりの計算から連立方程式や微分方程式の求解、微積分計算、フーリエ変換・ラプラス変換などの基本的な数式計算・数値計算を、直観的で手軽なユーザーインターフェイスで可能にし、これまでの手計算などでは生じやすかった計算ミスの軽減を可能にします。

1.3. マニュアル

[各種マニュアル \(cybernet.co.jp\)](http://cybernet.co.jp)

[コマンドリファレンス \(cybernet.co.jp\)](http://cybernet.co.jp)

2. TSUBAME3での利用方法

2.1. Mapleの実行

2.1.1. インタラクティブモードでのMapleの起動/実行

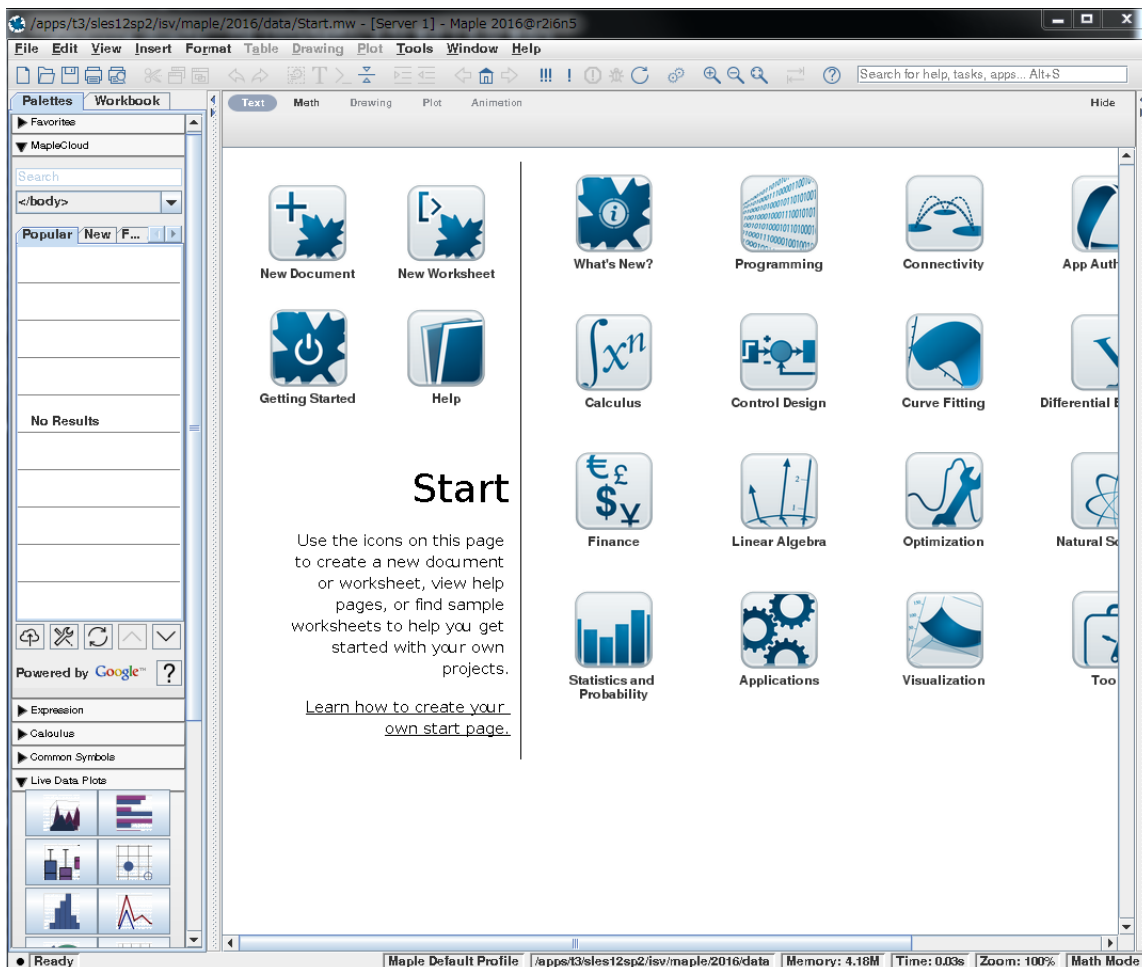
[ログイン方法](#)を参考にログインノードにログインして下さい。

CUIの場合は、[インタラクティブジョブの投入](#)を参考に計算ノードを確保し、以下のコマンドを実行して下さい。

```
$ cd <利用したいディレクトリ>
$ module load maple/2019.1
CUIの場合
$ maple
  |\^/|      Maple 2019 (X86 64 LINUX)
._|\||  |/_|. Copyright (c) Maplesoft, a division of Waterloo
Maple Inc. 2019
 \  MAPLE  / All rights reserved. Maple is a trademark of
 <---- ----> Waterloo Maple Inc.
   |      Type ? for help.
「quit」コマンドにより、終了することが出来ます。
> quit
memory used=0.9MB, alloc=8.3MB, time=0.06
```

GUIの場合は[インタラクティブノードを利用したX転送](#)を参考にノードをX転送付きで確保し、以下の方法で実施して下さい。

```
r0i0n0:~> module load maple
r0i0n0:~> xmaple
```



2.1.2. Univa Grid Engineによるバッチ投入

```
$ cd <利用したいディレクトリ>  
サンプルファイル (maple.sh) の場合  
$ qsub maple.sh
```

サンプルスクリプト (maple.sh)

```
#!/bin/bash  
#$ -cwd  
#$ -l f_node=1  
#$ -l h_rt=0:10:00  
#$ -V  
  
. /etc/profile.d/modules.sh  
module load cuda openmpi maple/2016.2  
  
maple < test.m
```

2.2. ライセンス使用状況の確認

Mapleのライセンス利用状況を以下のコマンドで確認できます。

```
$ lmutil lmstat -S maplelmg -c 27007@lice0:27007@remote:  
27007@t3ldap1
```

3. GPU による高速化

3.1. CUDA Package コマンド

Mapleでは、GPU デバイスのオン・オフ、情報参照するための CUDA Package コマンドを用意しています。次の 3 つのコマンドがあります。

Enable: GPU デバイス使用のオン・オフを行う

IsEnabled GPU デバイス使用状況を表示

Properties GPU デバイスの情報を表示

デフォルトでは、GPU デバイスの使用はオフになっています。そのため、GPU デバイスを使用できるようにするには、`CUDA:-Enable(true);` を実行する必要があります。

まず、`cuda`モジュールを読み込み、Mapleを起動します。

```
$ module load cuda
$ maple
```

`CUDA:-IsEnabled();` により、GPU デバイス認識状況を確認します。デフォルトでは、オフになっています。

```
> CUDA:-IsEnabled();

false
```

`CUDA:-Enable(true);` により、GPU デバイスを使用できるようにします。

```
> CUDA:-Enable(true);

false
```

`CUDA:-IsEnabled();` の結果が `true` となり、GPU デバイスが使用できる状態になりました。

```
> CUDA:-IsEnabled();

true
```

CUDA:-Enable(false); により、GPU デバイスを使用しない状態に戻すこともできます。

```
> CUDA:-Enable(false);  
  
true
```

CUDA:-IsEnabled(); の結果が false となりました。

```
> CUDA:-IsEnabled();  
  
false
```

GPU デバイスの情報を得るには、CUDA:-Properties(); を実行します。

```
> CUDA:-Properties();  
[table([%12, "ID" = 0, "Warp Size" = 32, %11, %10, %9, %8, %7, %6,  
"Major" = 6, %5, %4, %3, %2, "Minor" = 0,  
"Clock Rate" = 1480500, "Device Overlap" = 1,  
%1  
]), table([%12, "ID" = 1, "Warp Size" = 32, %11, %10, %9, %8,  
%7, %6, "Major" = 6, %5, %4, %3, %2, "Minor" = 0,  
"Clock Rate" = 1480500, "Device Overlap" = 1,  
%1  
]), table([%12, "ID" = 2, "Warp Size" = 32, %11, %10, %9, %8,  
%7, %6, "Major" = 6, %5, %4, %3, %2, "Minor" = 0,  
"Clock Rate" = 1480500, "Device Overlap" = 1,  
%1  
]), table([%12, "ID" = 3, "Warp Size" = 32, %11, %10, %9, %8,  
%7, %6, "Major" = 6, %5, %4, %3, %2, "Minor" = 0,  
"Clock Rate" = 1480500, "Device Overlap" = 1,  
%1  
])]  
%1 := "Total Constant Memory" = 65536  
%2 := "Texture Alignment" = 512  
%3 := "Total Global Memory" = 4294967295  
%4 := "Max Grid Size" = [2147483647, 65535, 65535]  
%5 := "Resisters Per Block" = 65536  
%6 := "MultiProcessor Count" = 56  
%7 := "Name" = "Tesla P100-SXM2-16GB"  
%8 := "Shared Memory Per Block" = 49152  
%9 := "Memory Pitch" = 2147483647  
%10 := "Kernel Exec Timeout Enabled" = false  
%11 := "Max Threads Per Block" = 1024  
%12 := "Max Threads Dimensions" = [1024, 1024, 64]
```

デバイスを指定しての実行も可能です。GPU が4つあるので id=0, id=1, id=2, id=3の4通りの指定が可能です。

```
> CUDA:-Properties(id=0);  
table(["Max Threads Dimensions" = [1024, 1024, 64], "ID" = 0,  
"Warp Size" = 32, "Max Threads Per Block" = 1024,  
"Kernel Exec Timeout Enabled" = false, "Memory Pitch" =  
2147483647, "Shared Memory Per Block" = 49152,  
"Name" = "Tesla P100-SXM2-16GB", "MultiProcessor Count" = 56,  
"Major" = 6, "Resisters Per Block" = 65536,  
"Max Grid Size" = [2147483647, 65535, 65535], "Total Global  
Memory" = 4294967295, "Texture Alignment" = 512, "Mi  
"Clock Rate" = 1480500, "Device Overlap" = 1,  
"Total Constant Memory" = 65536  
])
```

3.2. GPU 対応している機能

LinearAlgebraの行列積演算をGPUにより加速することができます。

サイバネットシステム株式会社のMaple紹介ページ「効率性の向上 - CUDA 加速 (6.1), (6.2)」にある サンプルを用いてTSUBAME2上で計算した例を以下に示します。

<http://www.cybernet.co.jp/maple/product/maple/maple14/>

あらかじめ、下記のファイルを作成しておきます。

test.m

```
`N := 20:  
  
m1 := LinearAlgebra:-RandomMatrix ( 2000, 2000, outputoptions =  
[ datatype = float [ 4 ] ] ):  
m2 := LinearAlgebra:-RandomMatrix ( 2000, 2000, outputoptions =  
[ datatype = float [ 4 ] ] ):  
  
t := time[ real ] ():  
to N do mNoCuda := m1 . m2: end:  
tNoCuda := time[ real ] () - t;  
  
CUDA:-Enable( true );  
t := time[ real ] ():
```



```
to N do mCuda := m1 . m2: end:  
tCuda := time[ real ] ( ) - t;
```

ジョブ投入するためのスクリプトを作成します。

maple.sh

```
#!/bin/bash  
#$ -cwd  
#$ -l f_node=1  
#$ -l h_rt=0:10:00  
#$ -V  
  
. /etc/profile.d/modules.sh  
module load cuda openmpi maple  
  
maple < test.m
```

qsubコマンドによりバッチ実行します。

```
$ qsub maple.sh
```

計算終了後、「maple.sh. o[ジョブID]」ファイルができていますので、確認します。
ファイル上部のNo licenses available, queueing request...はライセンスが不足していることを示します。

ライセンス数が1本となっておりますので、ライセンスの使用状況を確認し、ジョブを一旦停止して再実行してください。ライセンスが不足していても計算資源は確保されたままとなりますので、ご注意ください。

```
$ cat maple.sh. o[ジョブID]  
No licenses available, queueing request...  
  |\^/|      Maple 2019 (X86 64 LINUX)  
  _|\|      |/_|. Copyright (c) Maplesoft, a division of Waterloo  
Maple Inc. 2019  
  \ MAPLE / All rights reserved. Maple is a trademark of  
  <____ ____> Waterloo Maple Inc.  
    |          Type ? for help.  
> N := 20:  
  
> m1 := LinearAlgebra:-RandomMatrix ( 2000, 2000, outputoptions =  
[ datatype = float [ 4 ] ] ):  
> m2 := LinearAlgebra:-RandomMatrix ( 2000, 2000, outputoptions =  
[ datatype = float [ 4 ] ] ):  
>
```

```
> t := time[ real ] ( ):
> to N do mNoCuda := m1 . m2: end:
memory used=77.1MB, alloc=84.6MB, time=4.05
memory used=168.7MB, alloc=99.9MB, time=7.99
> tNoCuda := time[ real ] ( ) - t;
                                tNoCuda := 2.528

>
> CUDA:-Enable( true );
                                false

> t := time[ real ] ( ):
> to N do mCuda := m1 . m2: end:
memory used=397.9MB, alloc=115.2MB, time=20.59
> tCuda := time[ real ] ( ) - t;
                                tCuda := 0.578

> quit
memory used=642.5MB, alloc=359.3MB, time=22.97
```

3.3. リファレンス

詳細については、Mapleのオンラインヘルプを参照してください。

<http://www.maplesoft.com/support/help/Maple/view.aspx?path=CUDA>

改訂履歴

改定日付	内容
2019/08/02	mkdocs版作成
2018/03/14	初版