# 引言

## 研究背景及意义

## CFD程序的GPU加速研究现状

## 本文研究内容

# 算法介绍

## CFD数值方法

## FGMRES+ILU

SU2程序中组装出雅可比矩阵之后，可以使用很多带预处理的迭代方法进行求解，在其提供的参照case中普遍使用了其中FGMRES+ILU。

Multicolor

## AAMG

## 本章小节

# 程序实现

本文是基于SU2调用开源库AMGX实现SU2的GPU并行加速，具体在实施过程中要解决两个问题，一是：针对SU2的软件架构和C++类设计，要合理得调用AMGX，在保证正确实现程序功能的前提下能够尽量少的改动程序原有架构。二是：AMGX设计之初主要考虑的是代数多重网格方法和预处理的迭代方法，直接应用到SU2中尚有诸多问题，一些模块需要重新实现。

本章首先介绍SU2如何实现调用AMGX，之后介绍把AMGX应用到AMGX中的ILU算法、FGMRES算法和代数多重网格的具体实现。

## SU2调用AMGX

SU2选用GPU加速需要单独在配置文件中作出说明，由于AMGX库有自己的配置文件，因此需要在SU2的配置文件中说明是否使用GPU加速并指定AMGX的配置文件位置，SU2增加配置选项需要按照原有代码风格进行增加。

配置文件设置右边配代码add string 和 bool（图）

SU2使用带预处理迭代方法求解线性方程组，在SU2的CSolver的各个子类中定义了要求解的特定偏微分方程组。例如可以为可压或不可压的欧拉方程组实例化CEulerSolver，为不同的湍流模型实例化CTurbSolver等等。



图 .1 CSolver的子类和类成员变量

在父类CSolver中包含类成员变量CSysMatrix、CSysVector和CSysSolve（图 3.1）。CSysMatrix存储雅可比矩阵，同时也存储了一些针对雅可比矩阵的预处理方法，比如Jacobi，ILU和LU-SGS等等。CSysSolve求解CSysMatrix（雅可比矩阵）和CSysVector（残差和解向量）组成的线性方程组，求解方法是Krylov子空间迭代法，例如广义极小残差法（GMRES）稳定的双共轭梯度法（BCGSTAB）。

基类CSolver派生出CEulerSolver，CEulerSolver派生出CNSSolver，在基类CSolver中定义AMGX相关变量，在构造函数中进行相关变量的初始化。由于调用子类的构造函数会调用父类的无参构造函数，SU2X初始化AMGX求解器时为了避免重复初始化的情况发生，选择在CEulerSolver和CNSSolver的含参构造函数中进行AMGX求解器的初始化，同时在CEulerSolver和CNSSolver的析构函数中进行AMGX求解器的后处理。

图片（左边变量定义 右边初始化）

## ILU算法实现

### 二维问题算法实现

### 三维问题算法实现

## FGMRES算法及其实现

## 代数多重网格

## 本章小节

# 程序测试

本文首先介绍测试算例，然后对分别二维和三维算例进行测试和分析，这里主要侧重于测试GPU版本的相应算法相较CPU版本的加速效果，最后进行对比分析。

## 测试介绍

### 二维算例

算例选择SU2官网提供的二维层流平板算例，计算选用守恒Roe格式空间离散，欧拉隐式时间推进，计算来流静温300K，来流静压100000N/m2，出口静压97250 N/m2，来流马赫数为0.2Ma，雷诺数为1301233.166（参考长度为平板长度0.3048m）。计算求解RANS控制方程，选用SA湍流模式。计算边界分为入口段，出口段，远场段，对称段和平板段（粘性壁面），具体见下图。

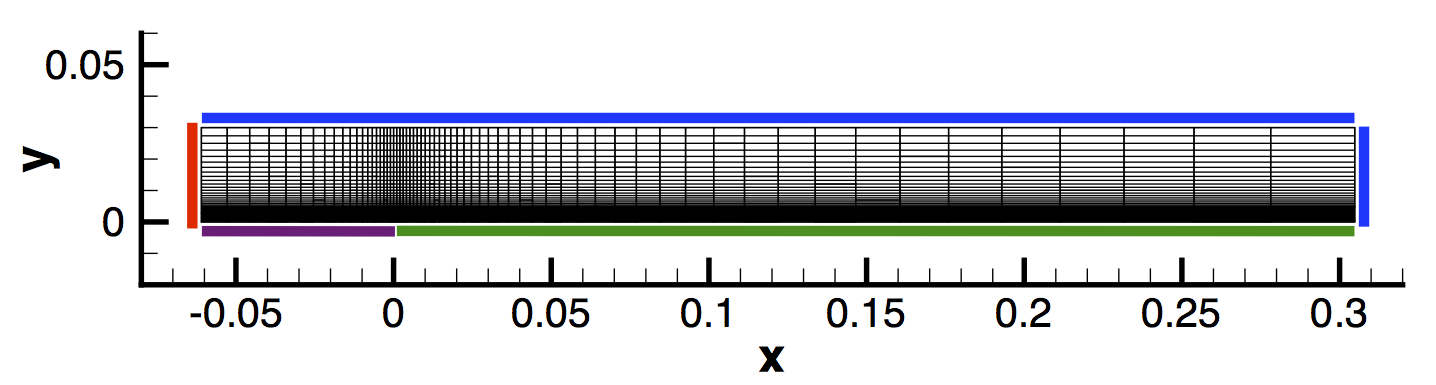


图 4.1 二维平板算例计算域和边界条件[1]

为研究GPU的加速效果，本章节选用三套不同网格对该算例进行计算，如表 4.1所示

表 4.1 二维平板算例不同计算网格

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **网格**  **编号** | **网格**  **点数** | **网格**  **单元数** | **雅可比矩阵**  **行数** | **雅可比矩阵**  **非零块数** |
| 网格1 | 60,689 | 119,308 | 60,689 | 420,681 |
| 网格2 | 979,377 | 975,936 | 979,377 | 4,890,001 |
| 网格3 | 3,462,468 | 3,787,734 | 3,462,468 | 17,962,870 |
| 网格4 | 9,999,000 | 9,988,002 | 9,999,000 | 49,973,002 |

### 三维算例

三位测试算例分别是ONERA M6，DLR-F4和HL-CRM标模。

ONERA M6计算设置为：来流温度为288.15K，马赫数为0.8395，来流攻角3.06度，雷诺数为11.72e6（参考长度为0.64607m）。

DLR-F4的计算设置为：来流温度为288.15K，马赫数为0.75，来流攻角2度，雷诺数为3e6（参考长度为0.1412m）。

HL-CRM的计算设置为：来流温度为288.15K，马赫数为0.2，来流攻角8度，雷诺数为3.26e6（参考长度为84.06m）。

边界条件设置为远场，对称面和绝热粘性壁面。

以上三个算例都是求解RANS控制方程，采用选用SA湍流模式，基于二阶 MUSCL 插值的 Roe 格式求解通量，使用欧拉隐式时间推进。

图ONERA M6(左) DLR-F4（中）和HL-CRM（右）

三个算例的网格基本情况见表 4.2，网格量按照十万、百万和千万量级依次递增。

表 4.2 三维不同计算网格

|  |  |  |
| --- | --- | --- |
| **网格** | **网格点数** | **网格单元数** |
| ONERA M6 | 108,396 | 582,752, |
| DLR-F4 | 1,220,883 | 3,158,428 |
| H | 8,302,012 | 18,011,980 |

### 测试环境

在本文所有算例的测试都是在表 4.2所示的工作站上进行的。

表 4.3 工作站配置

|  |  |  |
| --- | --- | --- |
|  | **CPU** | **GPU** |
| 型号 | Intel(R) Core(TM) i9-7900X CPU @ 3.30GHz | TITAN V |
| 数量 | 1（10核心20进程） | 2 |
| 内存/显存 | 128GB | 12GBx2 |

## 测试结果和分析

//输出50个矩阵，然后求解测试时间（各个部分的时间），占用显存情况，要有柱形图，要有表格，这里二维和三维的测试结果要放在一块进行比较分析

总耗时，显存使用，各种组分对比

二维三个算例

本章节主要侧重于测试在GPU版本的各种算法的加速效果，对比单个CPU核心和单个GPU，具体测试流程是：首先使用SU2通过单个CPU求解控制方程输出完整的雅可比矩阵，解向量初值和右端向量，然后这些向量和矩阵作为输入传递给AMGX，比较单个CPU核心和GPU的求解效率。

测试中SU2各项耗时结果由50次连续迭代做统计平均，SU2X通过求解 SU2连续迭代过程中输出的50个雅可比矩阵，解向量初值和右端向量，最终结果做统计平均。

### FGMRES+ILU（0）FGMRES+ILU（1）vs CPU

二维三个算例总耗时（两个算法）

二维三个算例各部分时间分别占比，介绍清楚各部分时间是干什么的

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 网格1 |  |  |  |  |
| 网格2 |  |  |  |  |
| 网格3 |  |  |  |  |

三维三个算例总耗时或者两个？

三维三个算例各部分时间分别占比，介绍清楚各部分时间是干什么的

### 对比分析

## 本章小节

# CPUs+GPUs测试

## 程序实现

## 测试介绍

## 结果对比分析

## 本章小节

# 总结与展望

做了什么，

本论文所有代码，测试算例的配置和网格文件以及测试结果都已经

展望