

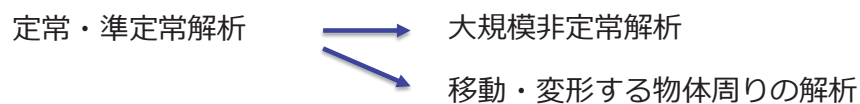
(FDC/ANSS合同企画6)「航空教育支援フォーラム：JAXA提供研究成果の最新動向」

最新動向紹介：FaSTAR-Move

宇宙航空研究開発機構 航空技術部門
数値解析技術研究ユニット
石田 崇

Introduction

計算速度の向上 (FaSTAR) / 計算機性能の向上に伴い (JSS2)
ユーザーのニーズも移行



大規模非定常解析への取り組み

FaSTARの前処理・後処理を並列化, 処理能力の向上, ストレージやI/Oの負荷削減といった整備が進んでいる. 橋本敦 *et al.*, JSASS-2016-2008-F/A, 2016.

移動・変形する物体周りの解析への取り組み

現行のFaSTARでは, 未対応であったため, 2015年度より移動・変形する物体周りの解析モジュールを有するソルバ『FaSTAR-Move ver.1』の開発に着手

青山剛史 *et al.*, JSASS-2016-5214, 2016.

FaSTAR-Moveの概要

FaSTAR-Move

移動・変形する物体周りの解析に対応した
移動・変形物体解析モジュール (ver.1)
開発中

スライディング格子に対応し、
エンジンのファン・圧縮機・タービンなどの翼列解析、
熱連成解析を可能とするエンジン解析モジュール (ver.2)
(2018年度より着手予定)

移動・変形物体解析モジュール

⇒ 移動重合格子法を採用

複数の格子の重ね合わせで、物体周りの格子を作成するため、
複雑な形状や一部分が移動・変形する物体に対して効果的な手法

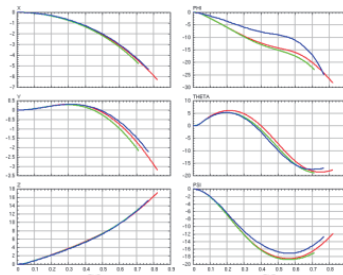
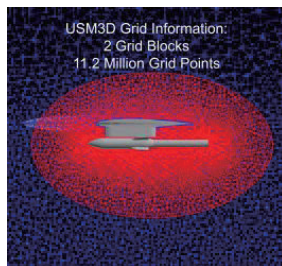
FaSTAR-Moveの開発目的/仕様設定

目的

移動・変形する物体周りの解析に対応することで、
FaSTAR (FaSTAR-Move)の実用化を促進し、実用的な技術基盤を目指す
⇒航空機開発の高速化を図り、競合機に対する市場競争力を強化する。

仕様設定

- 移動・変形する物体周りの解析に対応
- 解析現場にて現実的な時間内で解析
- 精度は空力/運動連成解析の場合、風洞試験結果との差が20 %以内



AEDCの検証問題
(搭載物分離解析)
 $M = 0.95$
青：風洞試験結果
赤：CFD(Beggar)
緑：CFD(USM3D)

W. Aziz *et al.*, Production Use of Overset Grid Technology
in the US Air Force Aircraft/Store Compatibility Enterprise, October 2012.

仕様設定に対する課題と対策

一番の課題

- 現実的な計算時間内で解析可能であること

⇒重合格子法では、複数の格子の重なりを判定し、計算点/非計算点や補間点の設定をするホール・カット処理が必要で、その処理の際に行う、重なる格子の空間探索に要する時間が問題



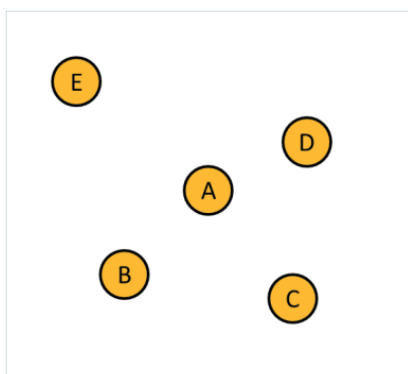
対策

- 空間を効率よく探索するアルゴリズムの Alternating Digital Tree (ADT) の導入
- MPI の実装 (ハイブリッド並列 (プロセス並列 + スレッド並列))

Alternating Digital Tree (ADT)

- 格子生成や重合格子で良く使われている手法
- 二分木探索 (binary tree) のデータ構造
 - ✓ 要素数が n で理想的な完全二分木が構築できた場合、計算量は $O(\log_2(n))$ (線形探索の場合 $O(n)$)

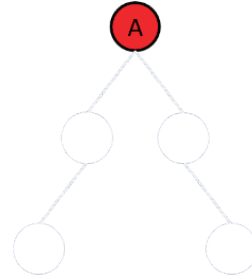
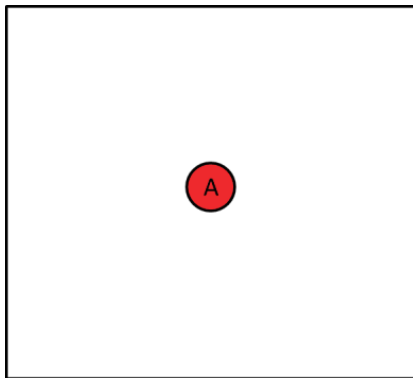
node の ADT の構築方法



Alternating Digital Tree (ADT)

- 格子生成や重合格子で良く使われている手法
- 二分木探索 (binary tree) のデータ構造
 - ✓ 要素数が n で理想的な完全二分木が構築できた場合, 計算量は $O(\log_2(n))$ (線形探索の場合 $O(n)$)

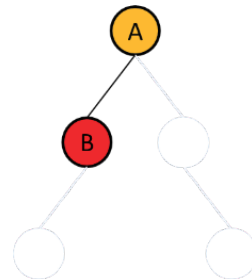
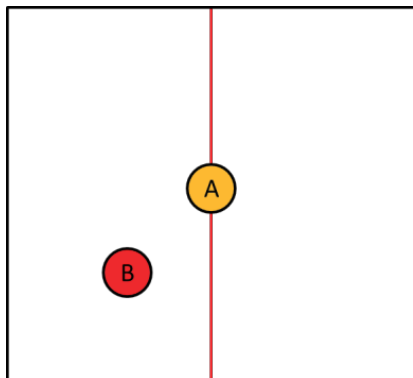
nodeのADTの構築方法



Alternating Digital Tree (ADT)

- 格子生成や重合格子で良く使われている手法
- 二分木探索 (binary tree) のデータ構造
 - ✓ 要素数が n で理想的な完全二分木が構築できた場合, 計算量は $O(\log_2(n))$ (線形探索の場合 $O(n)$)

nodeのADTの構築方法



Alternating Digital Tree (ADT)

- 格子生成や重合格子で良く使われている手法
- 二分木探索 (binary tree) のデータ構造
 - ✓ 要素数が n で理想的な完全二分木が構築できた場合, 計算量は $O(\log_2(n))$ (線形探索の場合 $O(n)$)

nodeのADTの構築方法

Alternating Digital Tree (ADT)

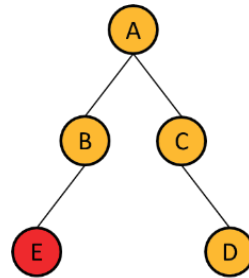
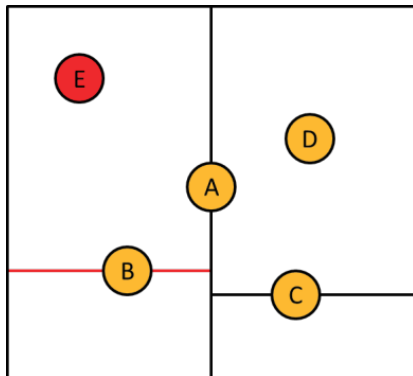
- 格子生成や重合格子で良く使われている手法
- 二分木探索 (binary tree) のデータ構造
 - ✓ 要素数が n で理想的な完全二分木が構築できた場合, 計算量は $O(\log_2(n))$ (線形探索の場合 $O(n)$)

nodeのADTの構築方法

Alternating Digital Tree (ADT)

- 格子生成や重合格子で良く使われている手法
- 二分木探索 (binary tree) のデータ構造
 - ✓ 要素数が n で理想的な完全二分木が構築できた場合, 計算量は $O(\log_2(n))$ (線形探索の場合 $O(n)$)

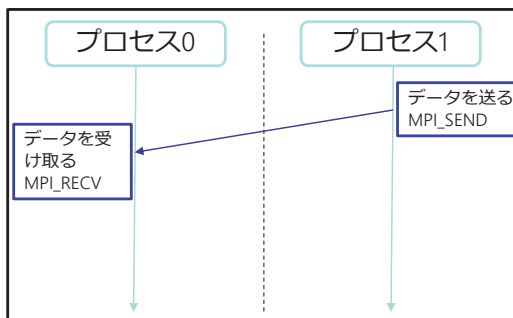
nodeのADTの構築方法



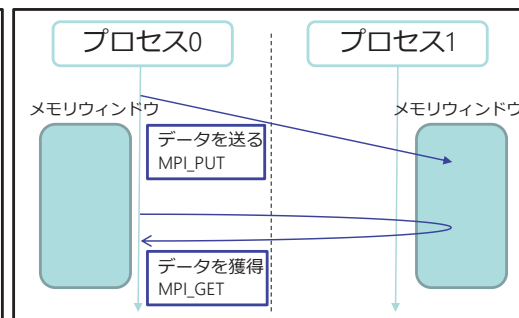
MPIの実装について

- Pre-processで格子情報を領域分割したものを使用
- ホール・カット処理のMPIは, 片方向通信を実装
 - ⇒どのプロセスが何のデータを持っているか不明なため, 双方向通信の実装は難しい.

双方向通信 (1対1通信) のイメージ図



片方向通信のイメージ図



- 相手の状態に関係なく, 他プロセスのデータにアクセスする通信方法
- MPI_PUT/GETでデータの送受信

ホール・カット処理のアルゴリズムとADT

2次元円柱重合格子のホール・カット処理手順

- ① 壁面内部にある格子をblankに設定
⇒最近傍の壁面セル探索にADT使用
境界面に対して, Hypercubeを定義.
- ② Hypercube内で, 格子の重なりを判定.
重なる格子がある場合, 壁距離
の大きい方をblankに設定する.
⇒重なる格子を探索する際にADT使用
- ③ 計算セルに隣接したblankセルを
補間点に設定する.
- ④ それぞれの格子に対し, 設定された
補間点が重なりが最小限となるまで
内側に入れていく.
⇒ADT使用
- ⑤ 補間点と計算点が全て重合出来たら
終了

格子の重なり判定 / 格子間の補間

格子の重なり判定

格子1のセル中心と最近傍の
格子2を探索, その境界面の
法線ベクトルと位置ベクトル
の内積を取り, 重なり判定を行う.

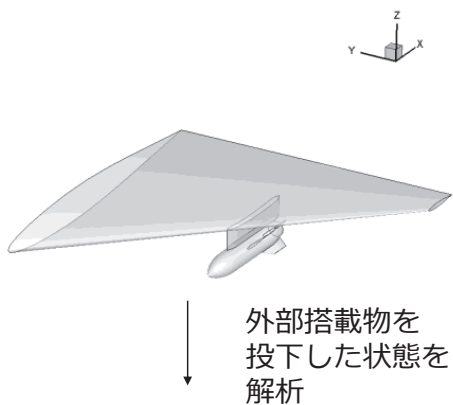
格子間の補間

逆距離加重補間を適用
(Inverse Distance Weighted
Interpolation : IDW)

複雑なモデルでFaSTAR-Moveの 移動重合機能を検証

外部搭載物分離解析

外部搭載物分離解析



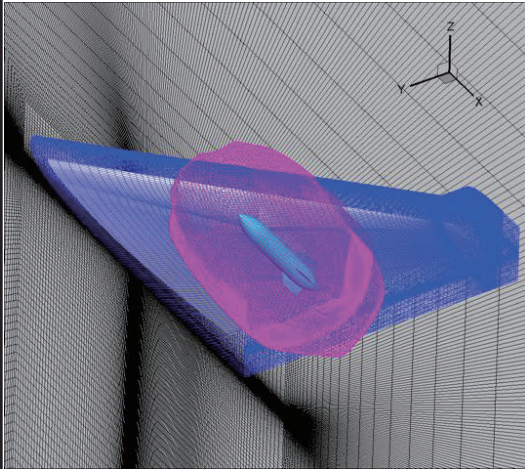
- JAXA Supercomputer System Generation2 (JSS2)で計算

解析条件

マッハ数	M = 0.95	
母機姿勢角	$\alpha = \beta = 0$ [deg]	
高度	26,000 [ft]	
搭載物質量特性	Mass = 907.184 [kg] $I_{xx} = 27.12$ [kg*m ²] $I_{yy} = I_{zz} = 488.1$ [kg*m ²]	
搭載物射出条件	射出位置 : 前方先端より36.54%位置 : 後方先端より51.57%位置 射出力 : 前方 : 10675.7 [N] : 後方 : 42702.9[N] ストローク : 0.10 [m]	
射出力以外の外力	重力 1 [G] = 9.80665 [m/sec ²]	
基準面積	0.20247 [m ²]	搭載物断面積
基準点	0.508 [m]	搭載物断面直径
モーメント基準点	先端より 41.85%	搭載物重心位置

解析手法

外部搭載物分離格子



主翼 + パイロン格子	262万セル
外部搭載物格子	316万セル
背景格子	1272万セル
合計	1850万セル

解析手法

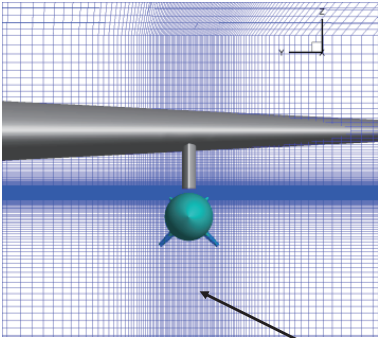
支配方程式	3次元圧縮性NS方程式
空間離散化	有限体積法 セル中心法
数値流束関数	HLLIW
勾配計算法	GLSQ
勾配制限リミタ	Hishida (van Lee型)
時間積分法	LU-SGS
乱流モデル	SA-noft2-R

合計1850万セルの外部搭載物分離の重合格子で移動重合機能及び現実的な時間で解析が可能かを検証。


目標は3日以内に全解析を終了

重合格子の格子生成における注意点

- 背景格子 正面図

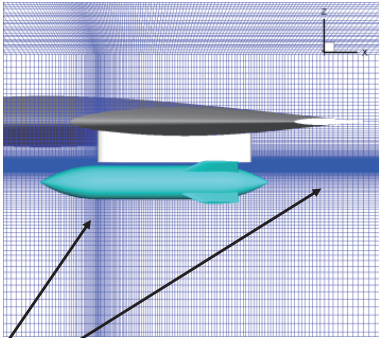


- 搭載物格子



格子サイズを細かく

側面図



格子サイズを細かく

重合格子では、重なり合う格子のサイズ比は、同等が好ましい。

また、上図のように物体間隔が狭い場合、その部分に十分な格子数が必要！

運動方程式プログラム

- 運動解析は、オイラー角変換を用い、6自由度を有する剛体の運動方程式を解く。

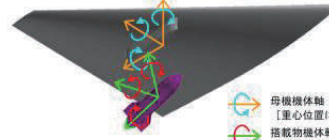
吉本稔 *et al.*, 第42回飛行機シンポジウム講演集, pp256-263, 2004.

位置 $\frac{dr}{dt} = E^t v_b$

速度 $\frac{dv_b}{dt} = \frac{F}{m} - \omega \times v_b$

姿勢角 $\frac{d\phi}{dt} = T^{-1}\omega$

角速度 $\frac{d\omega}{dt} = I^{-1}[G - \omega \times (I\omega)]$



母機機体軸 (搭載物の搭載時重心原点)
[重心位置に適用]
搭載物機体軸 (搭載物重心原点)
[重心速度, 角速度, 空力係数に適用]

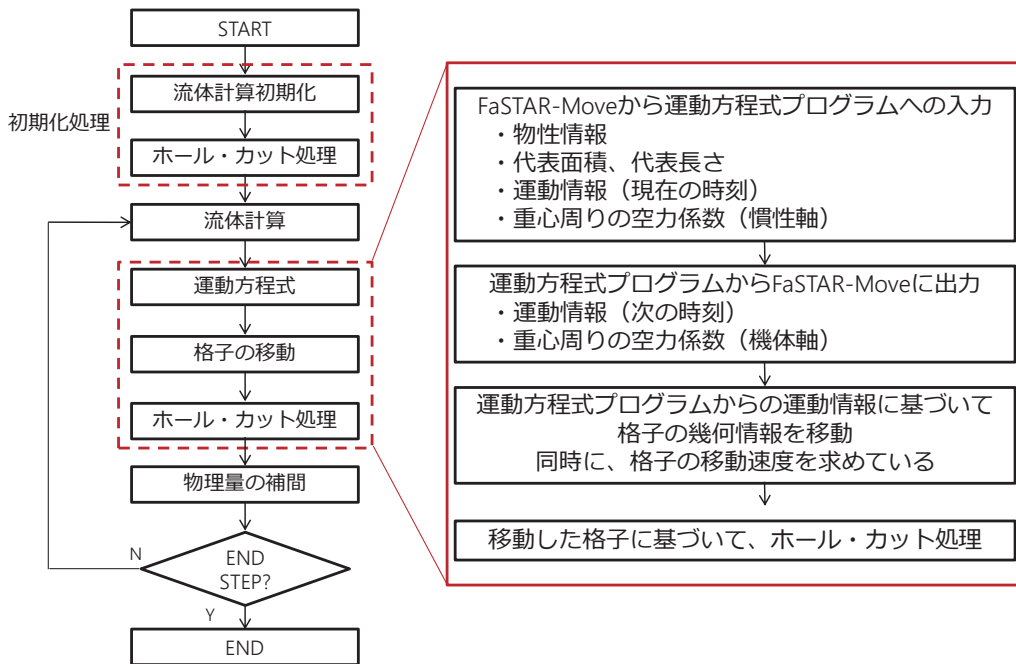


出力

- 運動物体の重心座標位置
- 運動物体の重心速度
- 運動物体の姿勢角
- 運動物体の角速度

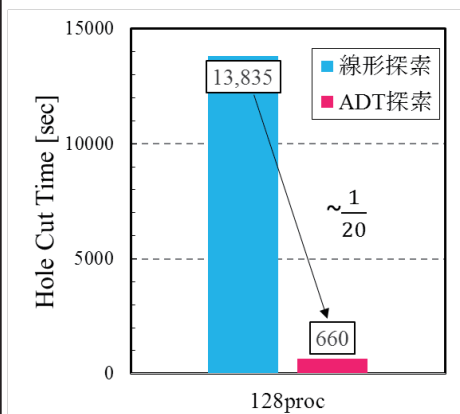
m	: 運動物体の質量
r	: 運動物体の重心位置(慣性系)
v_b	: 運動物体の速度(運動物体固定系)
ω	: 運動物体の重心周り角速度(運動物体固定系)
F	: 運動物体に働く外力(慣性系)
G	: 運動物体に働く重心周り外力モーメント(慣性系)
ϕ	: 慣性系表示を運動物体固定系表示にするオイラー角
$E(\phi)$: 慣性系表示を運動物体固定系表示にするオイラー角変換
$T(\phi)$: オイラー角の時間微分を角運動量にする変換

空力/運動連成解析の流れ



ホール・カット処理時間におけるADTの効果

ADT探索と線形探索の比較



外部搭載物分離モデルにおいて機能比較のため、線形探索法（空間全探索）とADT探索法でのホール・カットの処理時間を比較

プロセス並列数128(4CPU)で実施

⇒ **ADT探索が20倍早く探索**

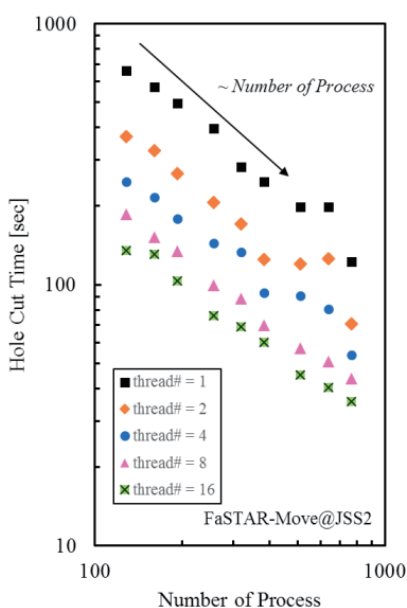
しかし、プロセス並列数128では、ホール・カット処理時間が11分かかっている。
100秒以上は現実的ではない！

そこで、より並列数を増やすとともに、スレッド並列も導入

⇒ **ハイブリッド並列の導入**

ハイブリッド並列性能の検証

ホール・カット処理時間のハイブリッド並列数依存性



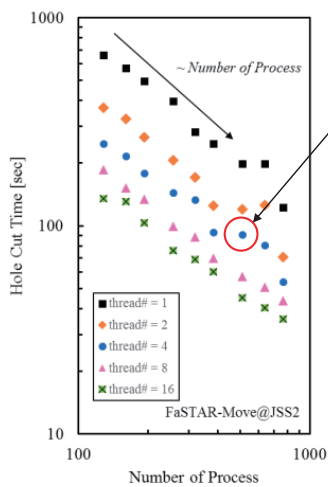
- プロセス並列数128 ~ 768, スレッド並列数1, 2, 4, 8, 16でホール・カット処理時間を計測



- プロセス並列の性能及びスレッド並列の性能が明確に出ていることを確認！
- 768proc/16threads (384CPU)において、ホール・カット処理時間約35秒を記録！

しかしながら、MPI_Put/Getを用いたプロセス並列は並列数増加につれ、1コア当たりのメモリ量が増える問題があり、今後、低メモリ化に対応する予定。

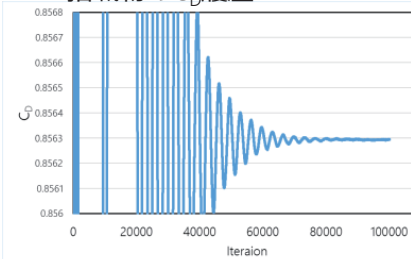
外部搭載物分離の解析時間



解析は、512proc/4threads(64CPU)で実施

- 定常計算に10万ステップ実施 (20時間)

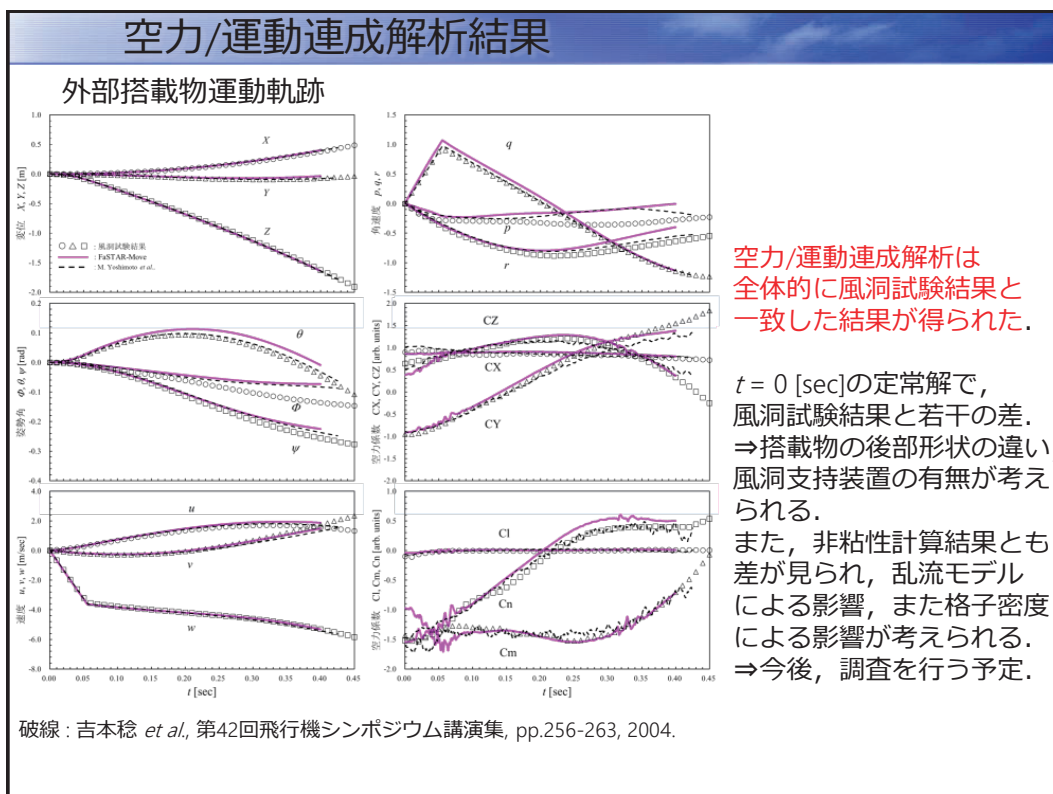
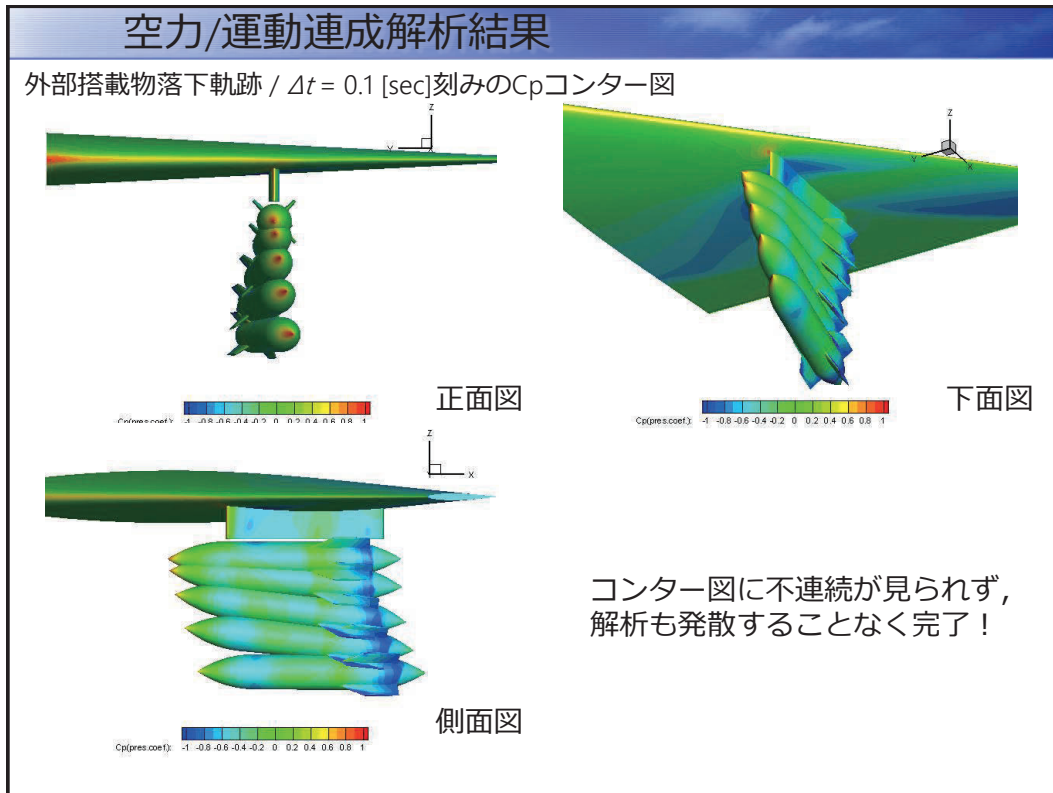
搭載物の C_D 履歴



- 空力/運動連成解析 (非定常計算) は
時間の刻み幅 $2.0e-4$ [sec],
陰解法における内部反復回数10 (CFL = 5)で
2000ステップ実施(38時間)

- 合計58時間で解析終了!
目標の3日(72時間)以内を達成!!

解析結果



まとめ

移動・変形を伴う物体周りの解析に対応するため
移動重合格子法を用いたFaSTAR-Moveの開発を行った。

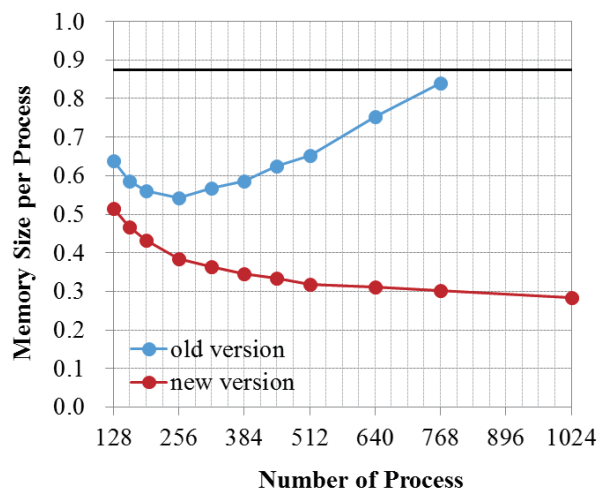
外部搭載物分離検証（合計1850万セル）において、

- 空間探索にADT法を用いることで、ホール・カット処理時間が線形探索法と比較して約1/20に短縮された。
ハイブリッド並列では、798proc/16threadsにおいて1ステップ当たりのホール・カット時間が30秒台まで短縮することを確認した。
- 空力/運動連成解析では、512proc/4threadsにおいて、計算時間はサンプリングを含む合計58時間で全解析を終了し、目標を達成した。
- 空力/運動連成解析結果は、搭載物の運動軌跡が風洞試験結果と概ね一致。
移動重合機能が正常に動作していることを確認した。

今後の予定

- ホール・カット処理の並列時におけるメモリ使用量の削減
- ホール・カット処理の更なる高速化

メモリ使用量の削減



プログラム改修実施

プロセス数の増加に伴うコア当たりのメモリ使用量の増加が改善

FaSTAR-Move Ver.1の詳細スペック

下線はデフォルトの手法
青字はFaSTAR-Move上では未検証

- **流体解析機能**
 - 支配方程式 : Euler, Full N-S
 - 離散化 : Cell-center (Faceベースデータ) のみ
 - 並び替え : Cuthill-Mackee
 - 非粘性流束 : HLLW, Roe, HLLC, AUSM+-UP, SLAU
 - 勾配評価 : Least-Square, Green-Gauss, GLSQ
 - 勾配制限関数 : Hishida, Venkatakrisnan, Barth-Jespersen, minmod
 - 高精度化 : MUSCL, U-MUSCL
 - 乱流モデル : SA, SST, EARSM, DES, LES
 - 遷移モデル : 強制遷移、自然遷移 ($\gamma-Re_{\theta t}$)
 - 領域分割 : METIS
 - 並列計算 : MPI
 - 時間積分 : LU-SGS (定常/非定常、local/global time stepping)
 - 安定化 : 初期条件, 初期CFL, 低品質格子対応、低圧力対応
- **重合格子機能**
 - 並列計算 : ホールカット処理のOpenMP/MPIハイブリッド並列対応
 - 運動連成解析 : free motion(6自由度運動方程式), prescribed motion
 - 物体数 : 複数可 (接触判定無し)
 - 格子間補間 : 距離の逆数を重み係数とした補間
- **前処理対応ソフト** : Gridgen(Pointwise), HexaGrid, MEGG3D(TAS-Mesh)
- **後処理対応ソフト** : Fieldview, Tecplot, Paraview