

カップリングライブラリ Jcupの概要と適用事例

2012/12/12

地球流体データ解析・数値計算ワークショップ
名古屋大学

Takashi Arakawa

Research Organization for Information Science and
Technology



Outline of the presentation

- ▶ Jcupの概要
 - ▶ 開発の背景
 - ▶ 特徴
- ▶ 適用事例
 - ▶ MIROC AGCM + MIROC Tri-polar COCO
 - ▶ NICAM + Tri-polar COCO + IO
- ▶ 今後の展開
 - ▶ 統合カップラー
 - ▶ MIROC ESM ← 渡辺さん



- ▶ Jcupの概要
 - ▶ 開発の背景
 - ▶ ScupとJcup
 - ▶ 特徴
 - ▶ 実行形態
 - ▶ 格子の扱い方
 - ▶ 補間計算
 - ▶ インターフェース
 - ▶ 処理の流れ
 - ▶ テスト計算
 - ▶ 現状

Scup と Jcup

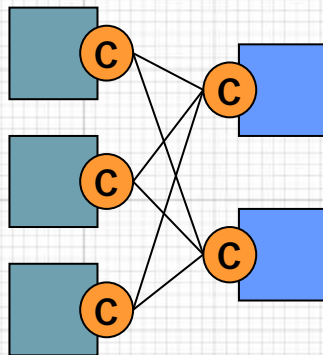
- ▶ Scupは気象研の吉村さんにより2007年に開発
 - ▶ メインターゲットはMRI/JMA ESM
- ▶ Jcupは吉村さん協力の下、荒川により開発
 - ▶ 開発当初のメインターゲットはMIROC ESM
 - ▶ MRI/JMA ESMとMIROC ESMに対して共通の機能を提供するという構想に基づき、Scupから多くの設計思想を引き継ぐ
 - ▶ インターフェースは非常に似ている
 - ▶ ただし、内部コードは別物



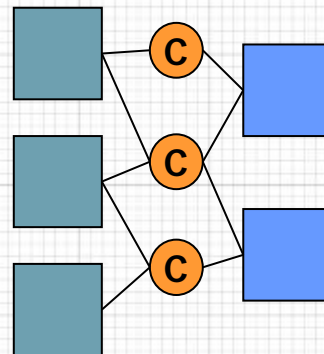
Jcupの概要

▶ 実行形態

- ▶ 各コンポーネントに付属するライブラリ
 - ▶ High efficiency on MPP Machine



Execution pattern of Jcup



Execution pattern of OASIS

- ▶ SPMD, MPMDが混在した結合をサポート
- ▶ 極めて広範囲な適用可能性

- ▶ カップラーの適用範囲は格子形状と補間アルゴリズムで規定される

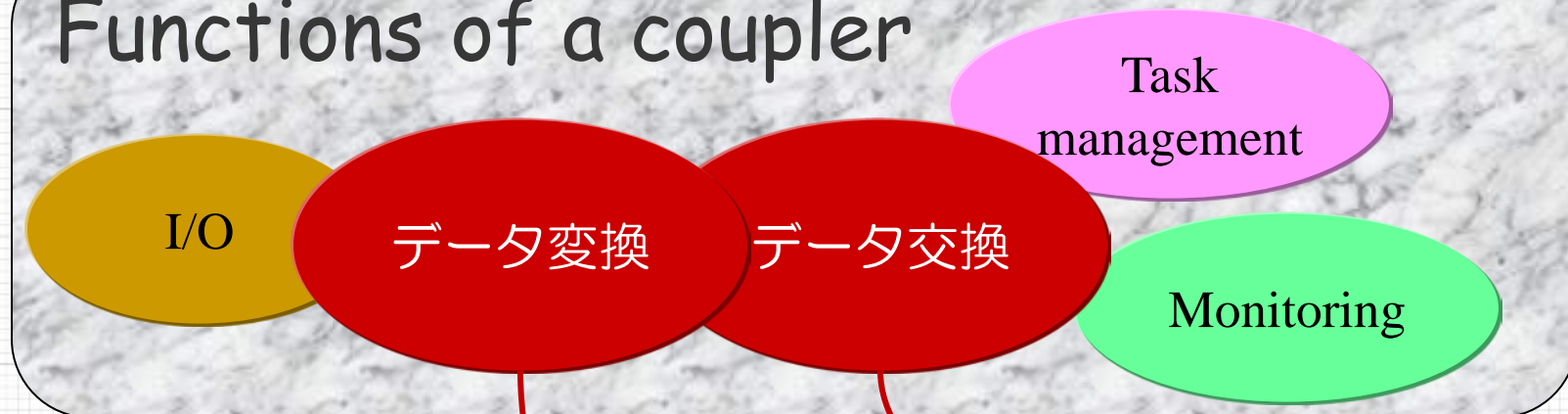


- ▶ 格子形状に依存しない
- ▶ 補間コードを利用者が実装

} ただし、使うにはそれなりの手間

補間コードを利用者に開放した背景

Functions of a coupler



カップラーの主要機能を2分類

値を変えない
値を変える

Black boxの中で値が変わってしまうのは気持ち悪い

値を変える操作（データ変換）を利用者に開放

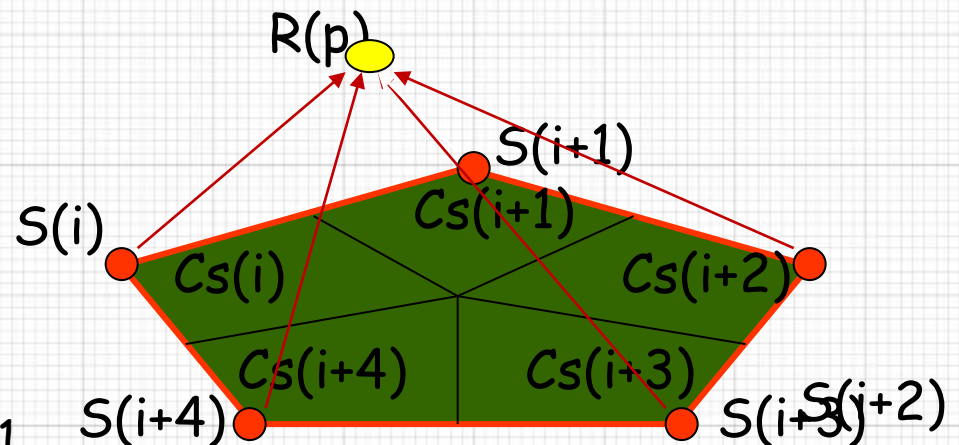
Portability

Flexibility

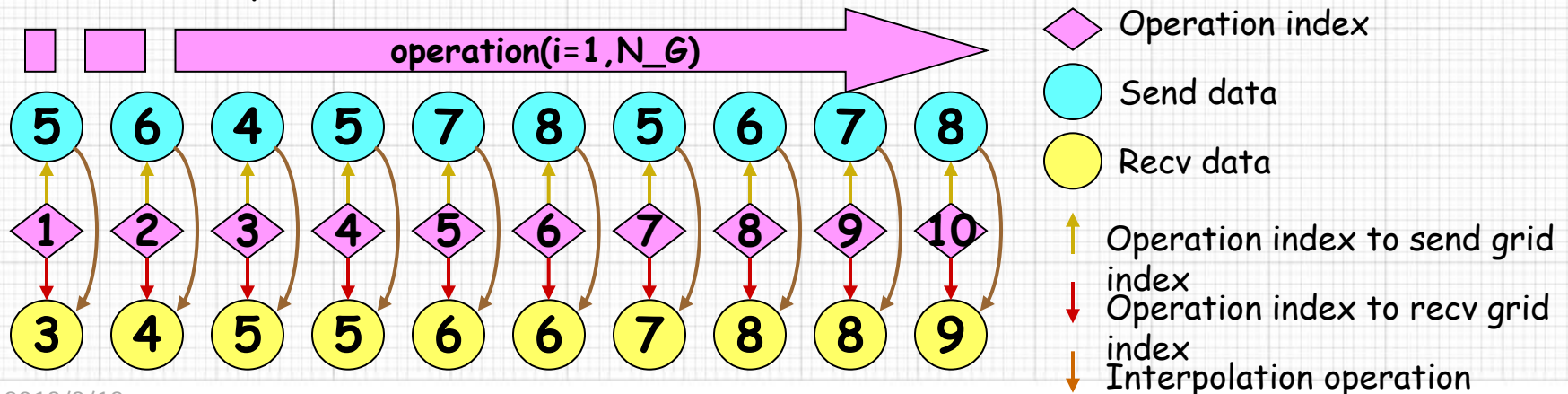


格子形状に依存しない設計

Interpolation = finite operation of a set of send data and coefficients

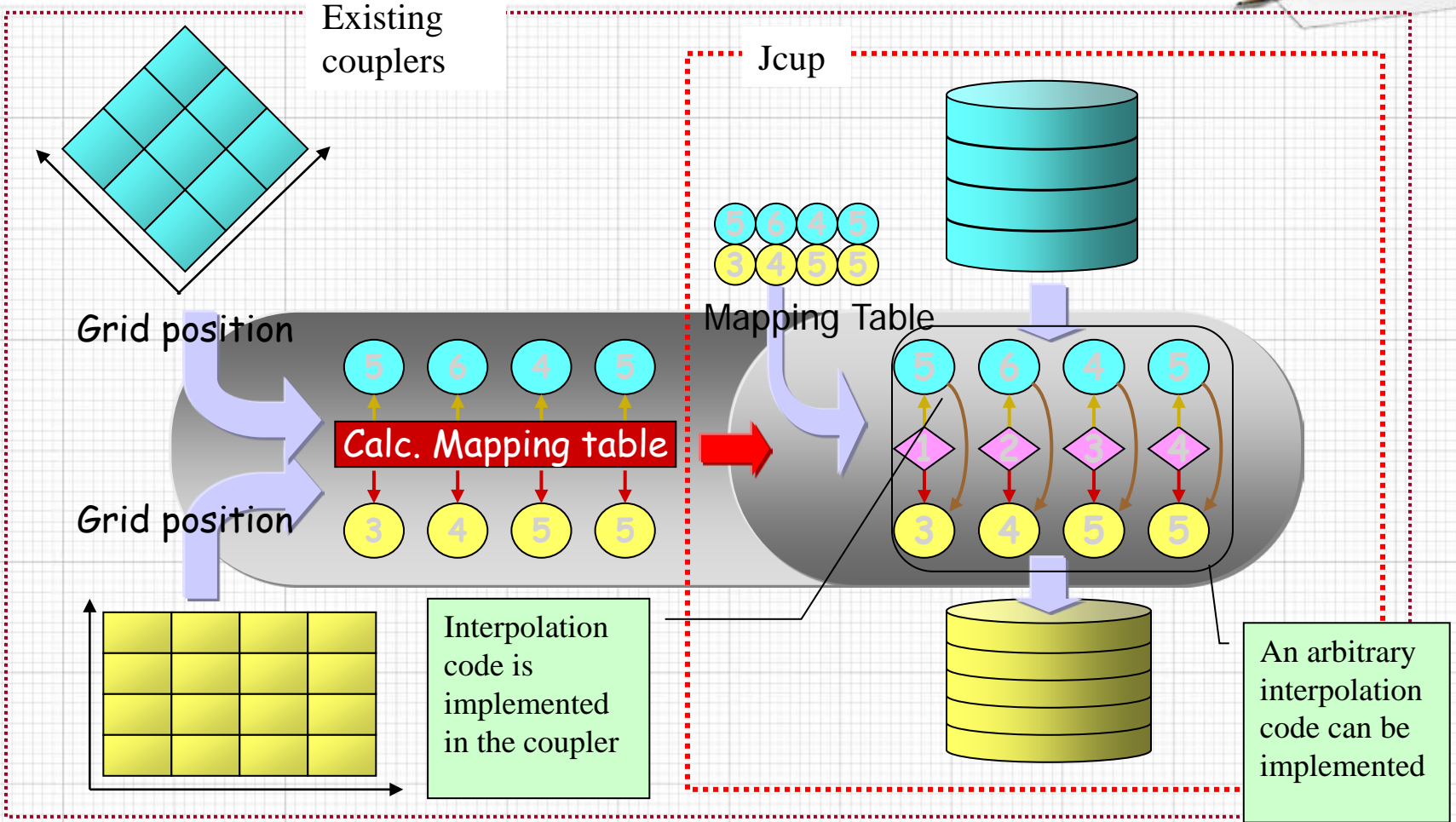


Interpolation can be represented as 1 dimensional operation





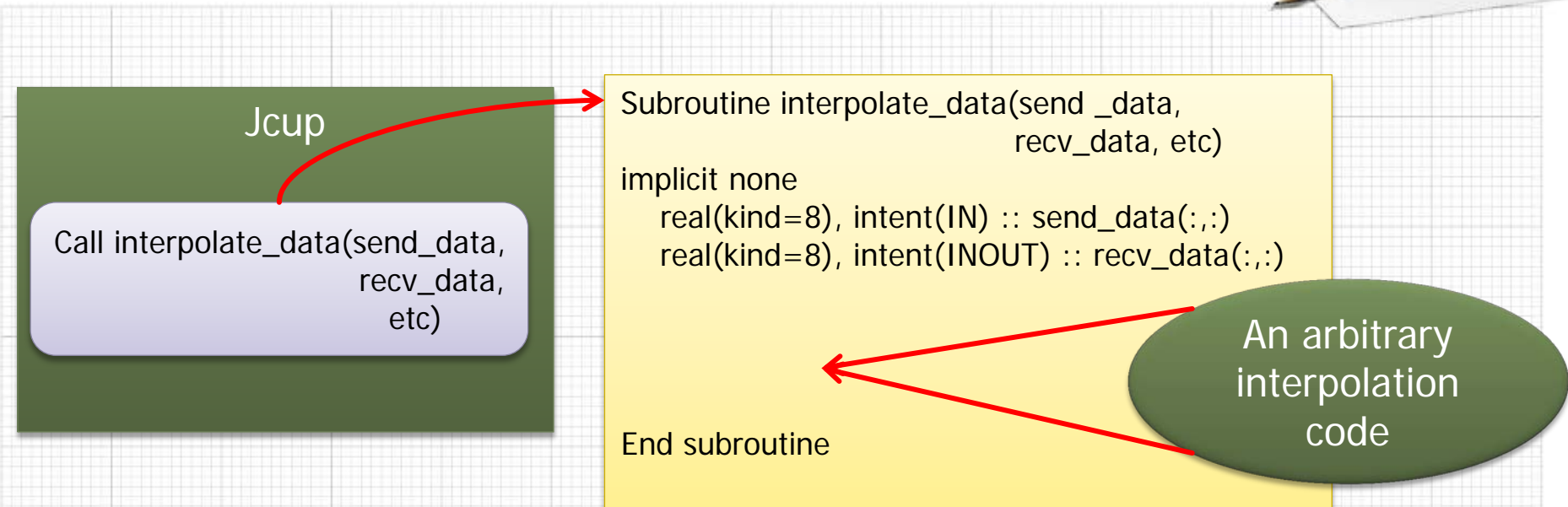
Feature of Jcup



Jcup has very wide applicability because it is free from grid structure



Implementation of Interpolation Code



Jcup内部からコールされる補間用サブルーチン
 interpolate_dataはインターフェースのみが定義され
 ている。
 利用者は各自の補間計算コードを自由に実装でき
 る・実装しなければならない。

モデルA

`jcup_set_new_comp(comp_name)`
コンポーネント名の設定

`jcup_initialize(my_task_name, init_flag, log_level, log_stderr)`
カップラーの初期化

`jcup_init_time(start_time)`
初期時刻の設定

`jcup_def_grid(grid_index, comp_name, grid_name)`
自領域の担当格子番号を設定

`jcup_end_grid_def()`
格子番号設定完了

`jcup_def_varp(varp, comp_name, data_name, grid_name, num_of_data)`
送信データ情報設定

`jcup_def_varg(varg, comp_name, data_name, grid_name, num_of_data, send_comp_name, send_data_name, recv_mode, interval, time_lag, mapping_tag, exchange_tag)`
受信データ情報設定

`jcup_end_var_def()`
送受信データ情報設定完了

`jcup_set_mapping_table(my_comp_name, send_comp_name, send_grid_name, recv_comp_name, recv_grid_name, mapping_tag, send_grid, recv_grid)`
補間計算グリッドインデックス情報設定(全領域)

モデルB

`jcup_set_new_comp(comp_name)`
コンポーネント名の設定

`jcup_initialize(my_task_name, init_flag, log_level, log_stderr)`
カップラーの初期化

`jcup_init_time(start_time)`
初期時刻の設定

`jcup_def_grid(grid_index, comp_name, grid_name)`
自領域の担当格子番号を設定

`jcup_end_grid_def()`
格子番号設定完了

`jcup_def_varp(varp, comp_name, data_name, grid_name, num_of_data)`
送信データ情報設定

`jcup_def_varg(varg, comp_name, data_name, grid_name, num_of_data, send_comp_name, send_data_name, recv_mode, interval, time_lag, mapping_tag, exchange_tag)`
受信データ情報設定

`jcup_end_var_def()`
送受信データ情報設定完了

`jcup_set_mapping_table(my_comp_name, send_comp_name, send_grid_name, recv_comp_name, recv_grid_name, mapping_tag)`
補間計算のテーブルを設定



複数回コール可

一回のみコール

モデルA

時間積分ループ開始

```

jcup_set_time(comp_name, time,
delta_t)

```

現在のモデル時刻と Δt を与える
データを送信

データバッファから不要データを
削除

データを受信

補間計算(call interpolate_data)
受信データバッファにデータ蓄積

```

jcup_get_data(varg, data)
受信データバッファからデータ取得

```

モデルの計算

```

jcup_put_data(varp, data)
送信データバッファにデータ蓄積

```

時間積分ループ終了

```

jcup_coupling_end(time)
結合の終了

```

モデルB

時間積分ループ開始

```

jcup_set_time(comp_name, time,
delta_t)

```

現在のモデル時刻と Δt を与える
データを受信

補間計算(call interpolate_data)
受信データバッファにデータ蓄積

データを送信

データバッファからデータ削除

```

jcup_get_data(varg, data)
受信データバッファからデータ取得

```

モデルの計算

```

jcup_put_data(varp, data)
送信データバッファにデータ蓄積

```

時間積分ループ終了

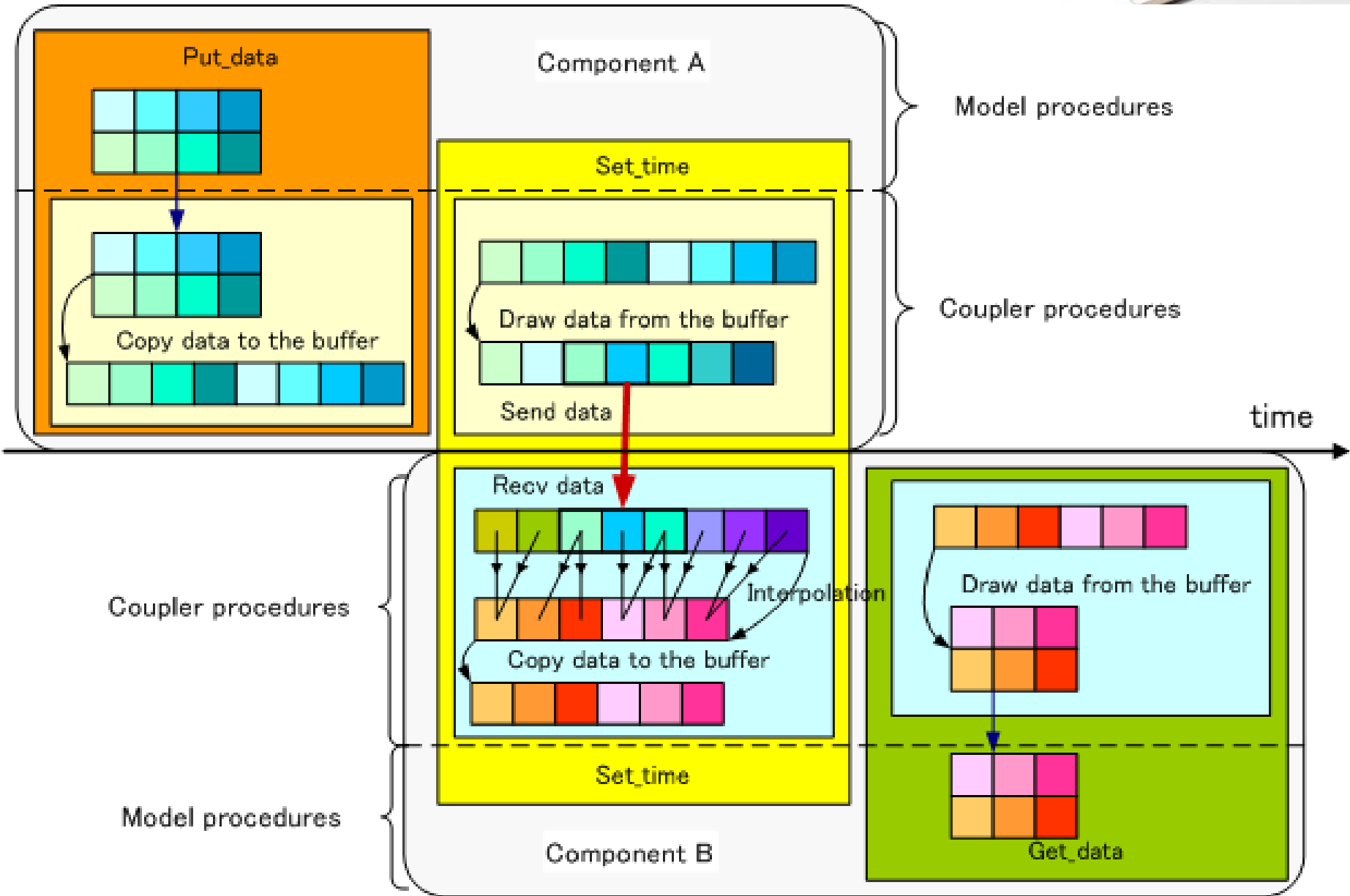
```

jcup_coupling_end(time)
結合の終了

```

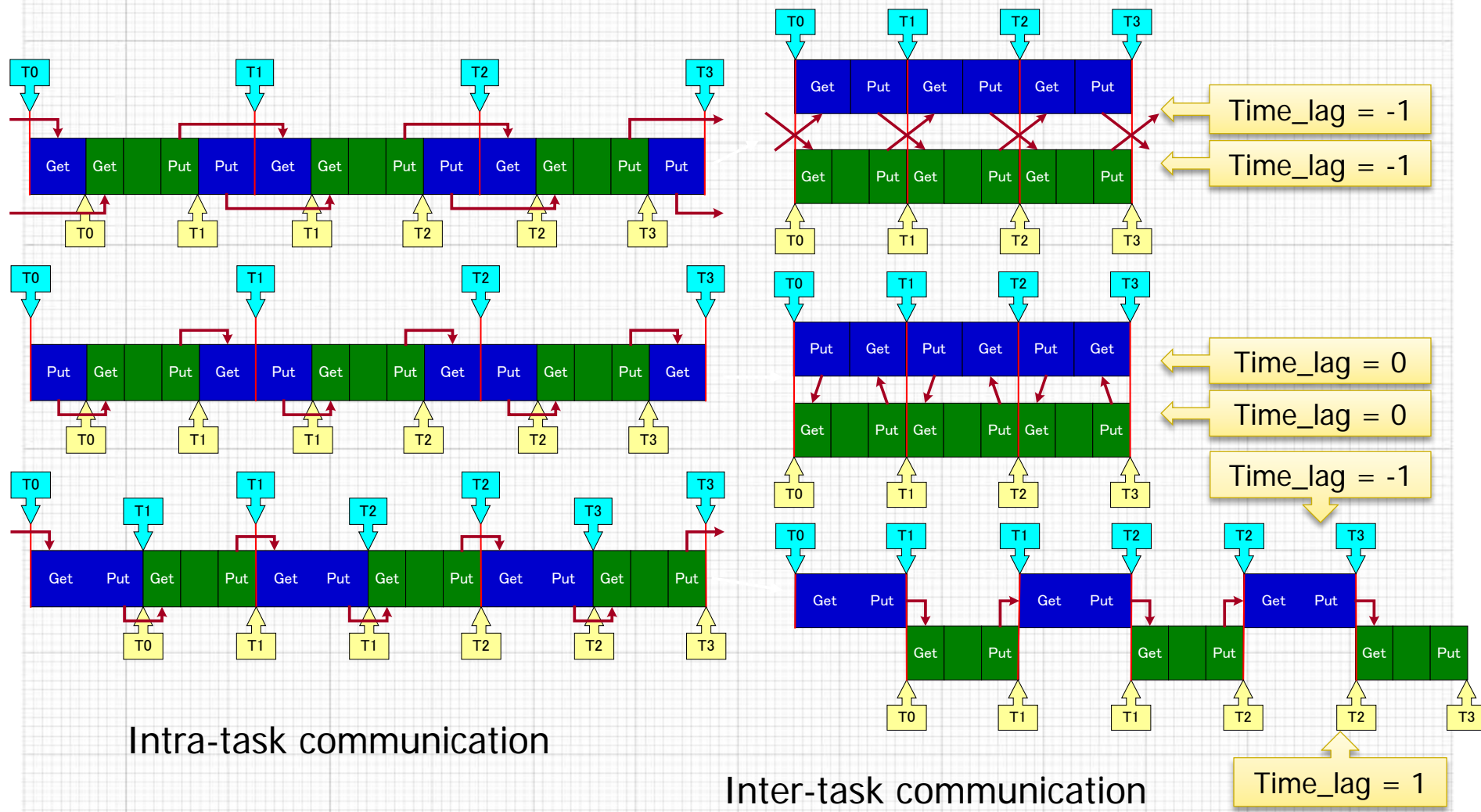


Data flow





Data exchange control



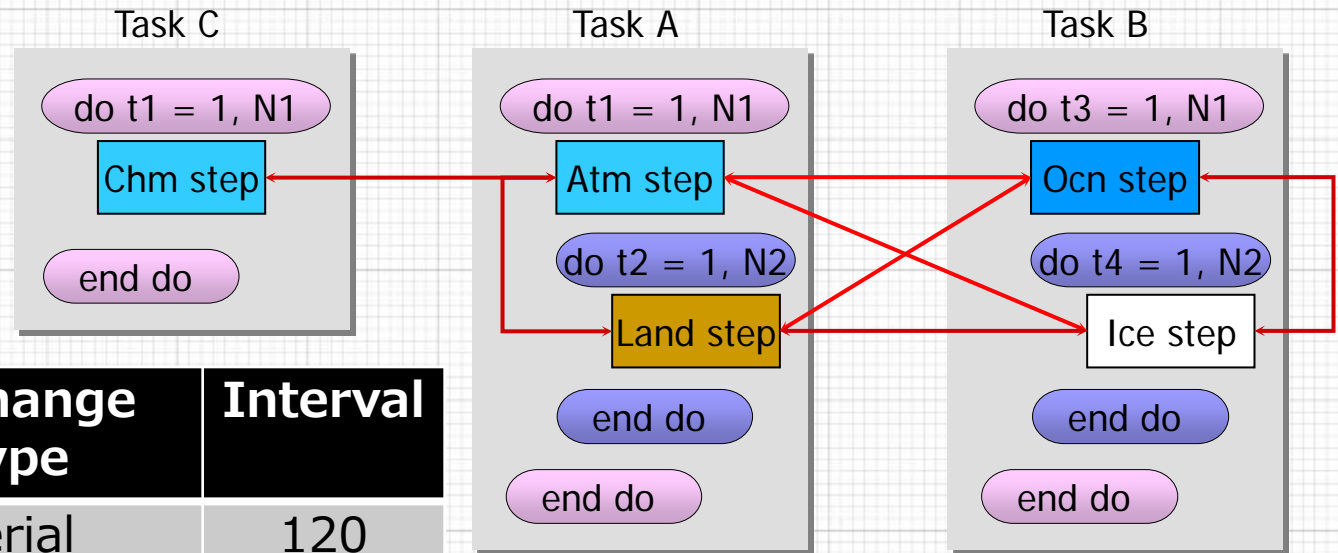
Intra-task communication

Inter-task communication



Coupling test on multi-component

► Test by Toy Models

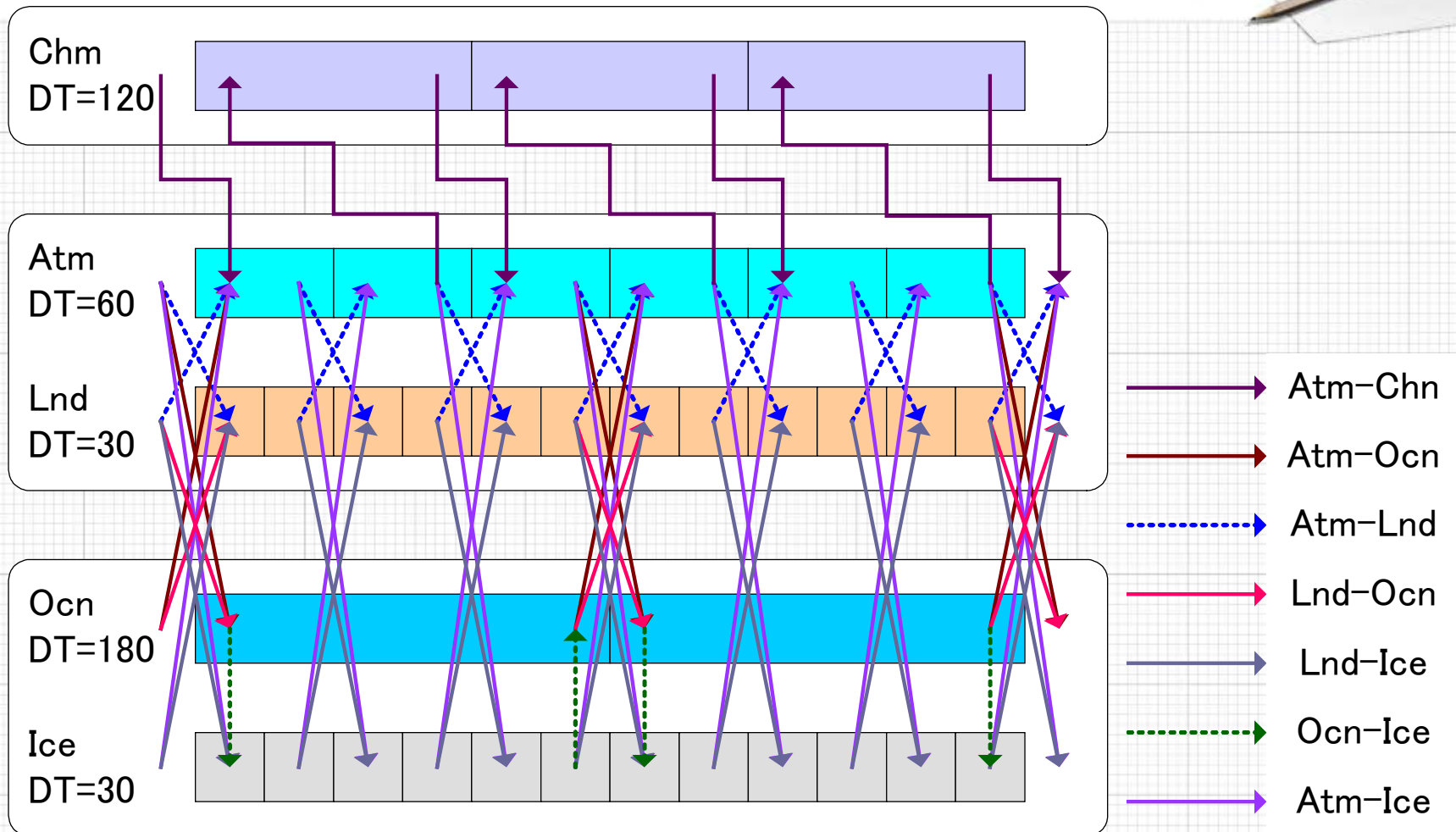


Component s	Exchange type	Interval
Atm \longleftrightarrow Chm	serial	120
Atm \longleftrightarrow Ocn	parallel	180
Atm \longleftrightarrow Ice	parallel	60
<i>Atm\longleftrightarrowLand</i>	<i>parallel</i>	<i>60</i>
Land \longleftrightarrow Ice	parallel	60
Ocn \longleftrightarrow Land	parallel	180
<i>Ocn\longleftrightarrowIce</i>	<i>immediate</i>	<i>180</i>

Task	Component	ΔT
A	Atm	60
	Land	30
B	Ocn	180
	Ice	30
C	Chm	120

Italic font: Intra-task exchange

Exchange diagram of test coupling



Very complex coupling has been successfully completed !

Jcup applied to MIROC5.0 atmosphere ocean coupling.

Jcupの現状



- ▶ Ver1.0をWebで公開中
 - ▶ ライセンスは修正BSD
 - ▶ 2012/2/1: rev.5
 - ▶ <https://sites.google.com/a/rist.jp/jcup>
- ▶ 未公開版の現行バージョンは2.01
 - ▶ 設定ファイルを不要にした
 - ▶ 更に補間用サブルーチンのインターフェースを変更した上で公開する予定



▶ 適用事例その1

▶ MIROC AGCM + MIROC Tri-polar COCO

▶ 概要

▶ 性能測定結果

MIROC大気海洋結合



- ▶ MIROCの大気モデルと海洋モデル（COCO）はオリジナルの結合プログラムで結合されていた
 - ▶ Gather→ISend→IRecv→Interpolate→Scatter
- ▶ これをJcupで置き換え
 - ▶ オリジナルの補間計算コードを流用する事により、バイナリ互換を保ちながら置換
 - ▶ Local to Localの効率的なデータ交換・補間計算

バイナリ互換性の確保



- ▶ バイナリレベルの一致
 - ▶ 高い確度でバグがないことを保証（絶対ではない）

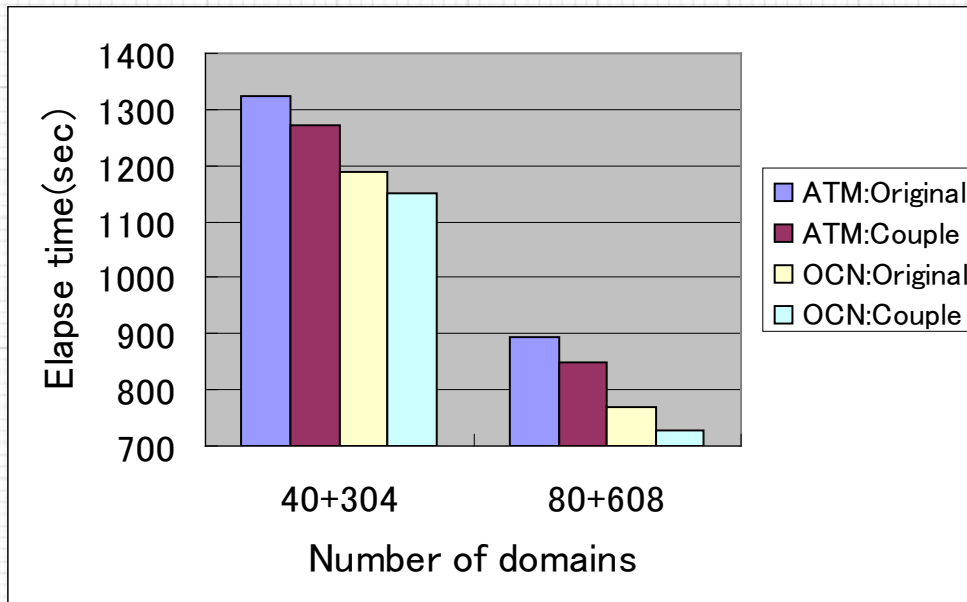
$$\begin{aligned}
 & \text{AOCN2} = \text{SOCN}(\text{LA}, \text{N}) * \text{BFOyyG}(\text{LO}, \text{L}+1) \\
 & \& \quad \quad \quad / (\text{BFSOCNG}(\text{LA}) * \text{BFOxxG}(\text{LA}, \text{L}+1)) \\
 & \text{BFOxxG}(\text{LA}, \text{L}+1+4*\text{NIC}) = \text{BFOxxG}(\text{LA}, \text{L}+1+4*\text{NIC}) \\
 & \& \quad \quad \quad + (\text{BFOyyG}(\text{LO}, \text{L}+1+4*\text{NIC}) + \text{kelvin}) * \text{AOCN2} \quad ! \text{tsi}
 \end{aligned}$$

$$\begin{aligned}
 \text{aocn} &= \\
 & \text{socn}(i) * \text{bfao}(\text{send_point}, d) / (\text{bfsocn}(\text{recv_coef}) * \text{bfaa}(\text{recv_point}, d)) \\
 \text{recv_data}(\text{recv_point}, d) &= \\
 & \text{recv_data}(\text{recv_point}, d) + (\text{send_data}(\text{send_point}, d) + \text{kelvin}) * \text{aocn}
 \end{aligned}$$

Resolution : T106, 1280x916

Domain decomposition : 40+304, 80+608

Machine : ES1



# of PE	Original		Coupler	
	ATM to OCN	OCN to ATM	ATM to OCN	OCN to ATM
40+304	49.49	23.34	0.683	0.067
80+608	49.53	23.37	0.347	0.041

補間計算にかかる時間は無視できるレベルに減少
 → Jcupの有効性を確認



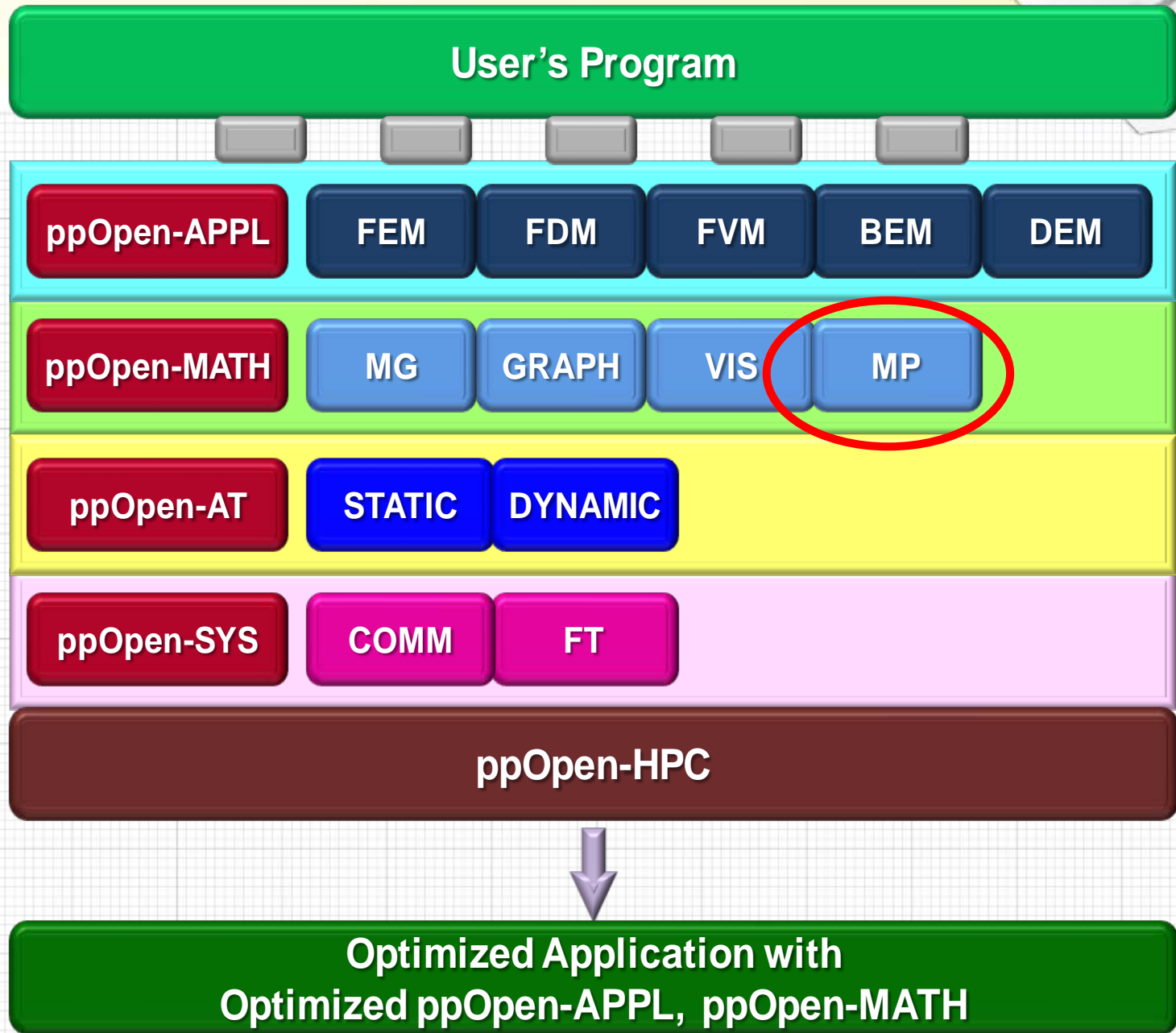
▶ 適用事例その2

▶ NICAM + Tri-polar COCO + IO

- ▶ 計画の概要：CREST課題
- ▶ Grid mapping tool開発
- ▶ NICAM + Tri-polar COCOの結合
- ▶ NICAM + IO componentの結合
- ▶ 全体の構造



- ▶ 「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」
 - ▶ 研究総括：米澤明憲(東大情報理工学研究科教授)
- ▶ 「自動チューニング機構を有するアプリケーション開発・実行環境」
 - ▶ 中島研吾(東大情報基盤センター教授)
 - ▶ 佐藤 正樹 東京大学 大気海洋研究所
 - ▶ 古村 孝志 東京大学 情報学環
 - ▶ 奥田 洋司 東京大学 人工物工学研究センター
 - ▶ 岩下 武史 京都大学 学術情報メディアセンター
 - ▶ 阪口 秀 海洋研究開発機構 地球内部ダイナミクス領域 チームリーダー



ppOpen-MATH/MP



- ▶ 複数のアプリケーションを結合し、大規模データ転送、変換を実施する弱連成のためのカップリングツールppOpen-MATH/MPの研究開発を実施する。ppOpen-MATH/MPは以下のような機能を有する：
 - ▶ ppOpen-APPLのサポートする様々な離散化手法を全てサポートし、二次元、三次元カップリング機能を提供する。異なった離散化手法によるモデル間のカップリングも可能とする。時間積分手法に依りて複数のプロセスをパイプライン処理する機能を提供する。
 - ▶ 計算プロセスだけでなく、シミュレーションとppOpen-VISの並列実行も可能とする。
 - ▶ 複数のアプリケーションに対して、利用可能な計算資源を最適に配分する機能（M×N配分機能）を開発する。その際、ppOpen-MATH/GRAPHを利用する。

研究体制

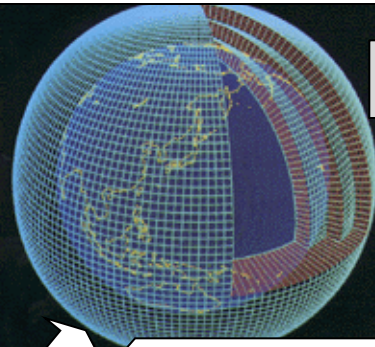
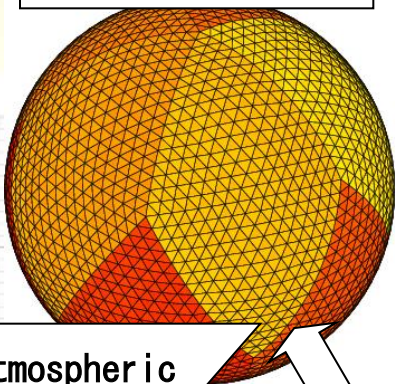
- ▶ 東京大学大気海洋研究所
 - ▶ 佐藤正樹, 羽角博康, 三浦裕亮
 - ▶ 研究員A→私
- ▶ 理研計算機構 富田浩文
- ▶ 海洋研究開発機構 河宮未知生
- ▶ 高度情報科学技術研究機構 荒川 隆

Icosahedral grid

Latitude-Longitude grid

MIROC-A

NICAM-Agrid
NICAM-ZMgrid
HICAM-ZMgrid



Atmospheric model 1

Atmospheric model 2

Joint-Shell

- Grid transformation
- Multi-ensemble
- IO coupler
- Pre- and post-process
- Fault tolerance
- CPU driver

J-cup

Shell

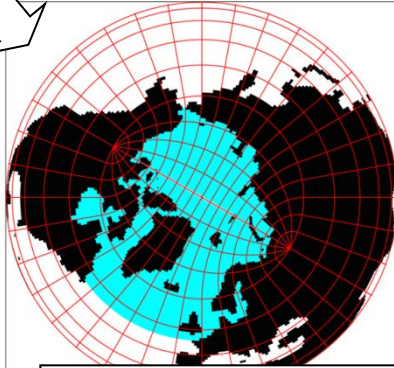
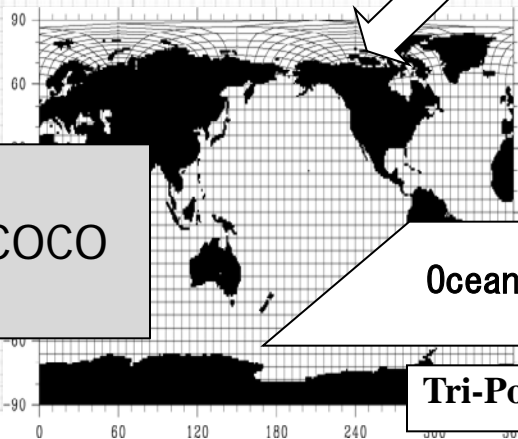
Post-Peta Flops computers
-System software
-Architecture

COCO
regional-COCO
Kinako

Ocean model

Tri-Polar grid

Ocean regional model
Nonhyrosatic model



Grid Mapping Tool



▶ 補間計算時の格子点对应関係 (mapping table) を計算

▶ 対応格子探査アルゴリズムは補間方法に依存

▶ 最近傍近似

- ▶ Kd-tree

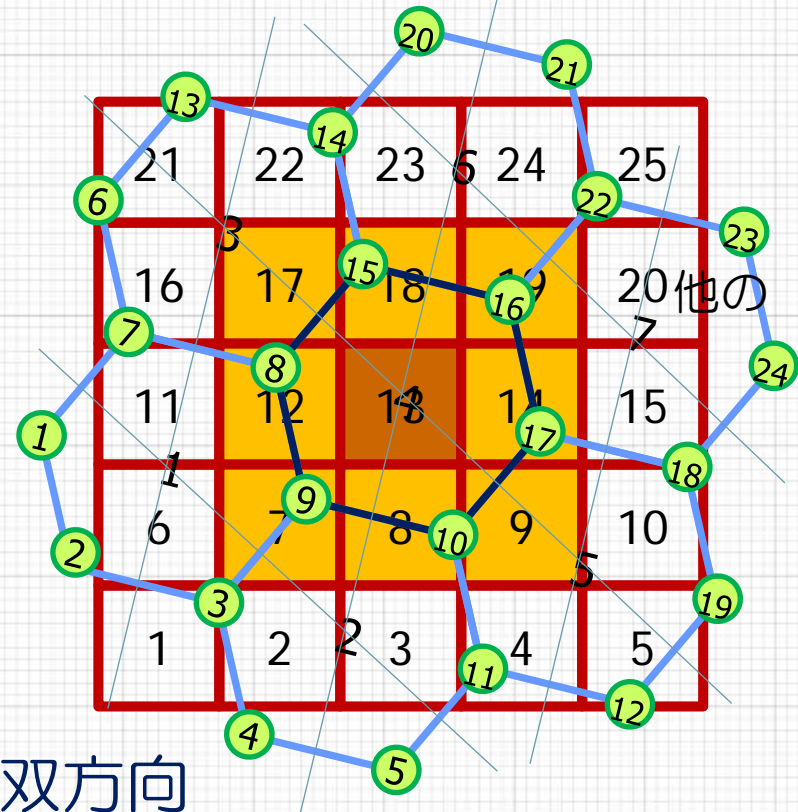
▶ Control volume

- ▶ Polygon search

▶ Polygon search

- ▶ 多角形の重なりを判定

- ▶ 重なりは、頂点が相手のどの多角形に内包されるかで一次判定 ← 双方向

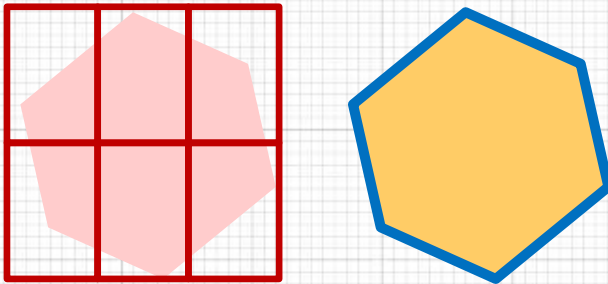


Grid Mapping Tool



▶ 補間係数

- ▶ First order approximation (by Jones 1999)
 - ▶ Area weighting



$$\overline{F}_k = \sum_{n=1}^N \overline{f}_n w_{\ln k} \quad (1)$$

$$w_{\ln k} = \frac{1}{A_k} \int_{A_{nk}} dA \quad (2)$$

テスト計算



- ▶ 送信モデルに下式の理論値を与える

$$F = 2 + \cos^2(4\varphi)\cos(2\lambda)$$

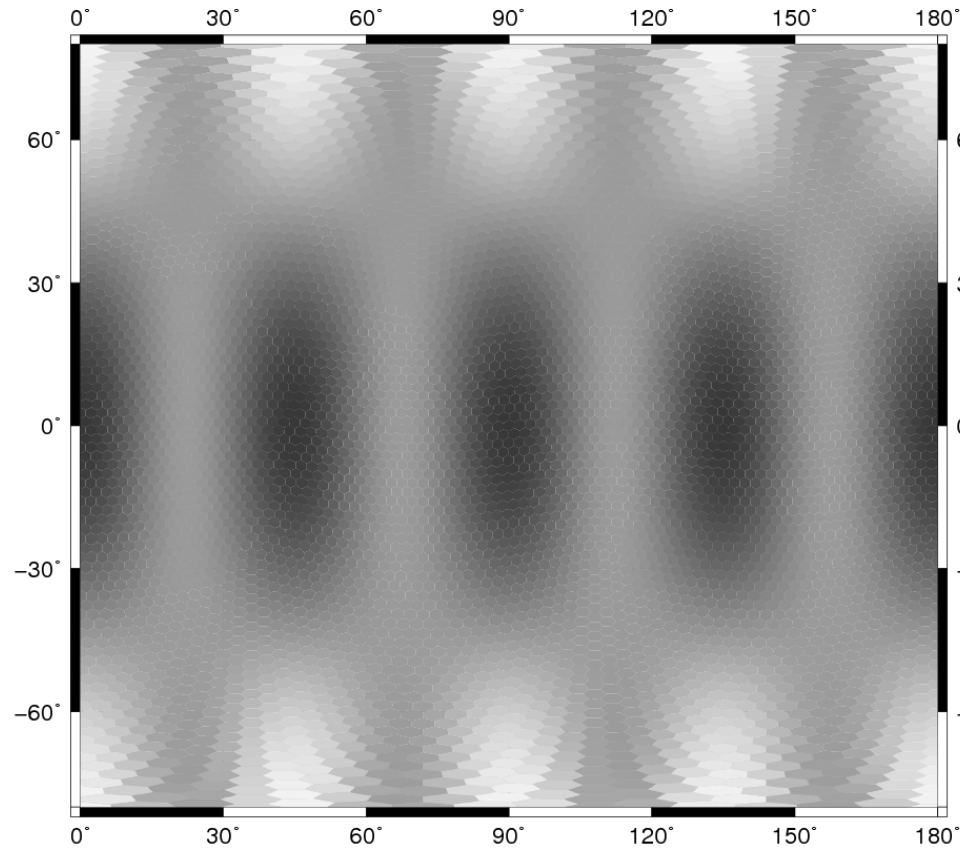
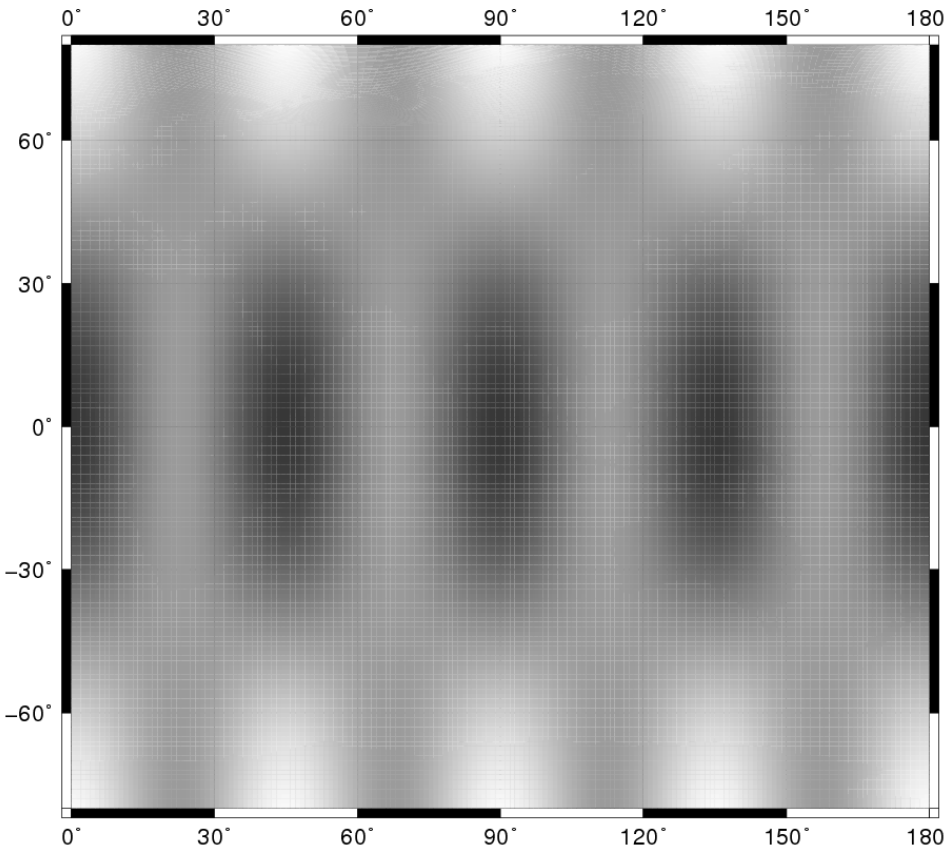
φ :latitude, λ :longitude

- ▶ COCO to NICAM

- ▶ NICAM to COCO

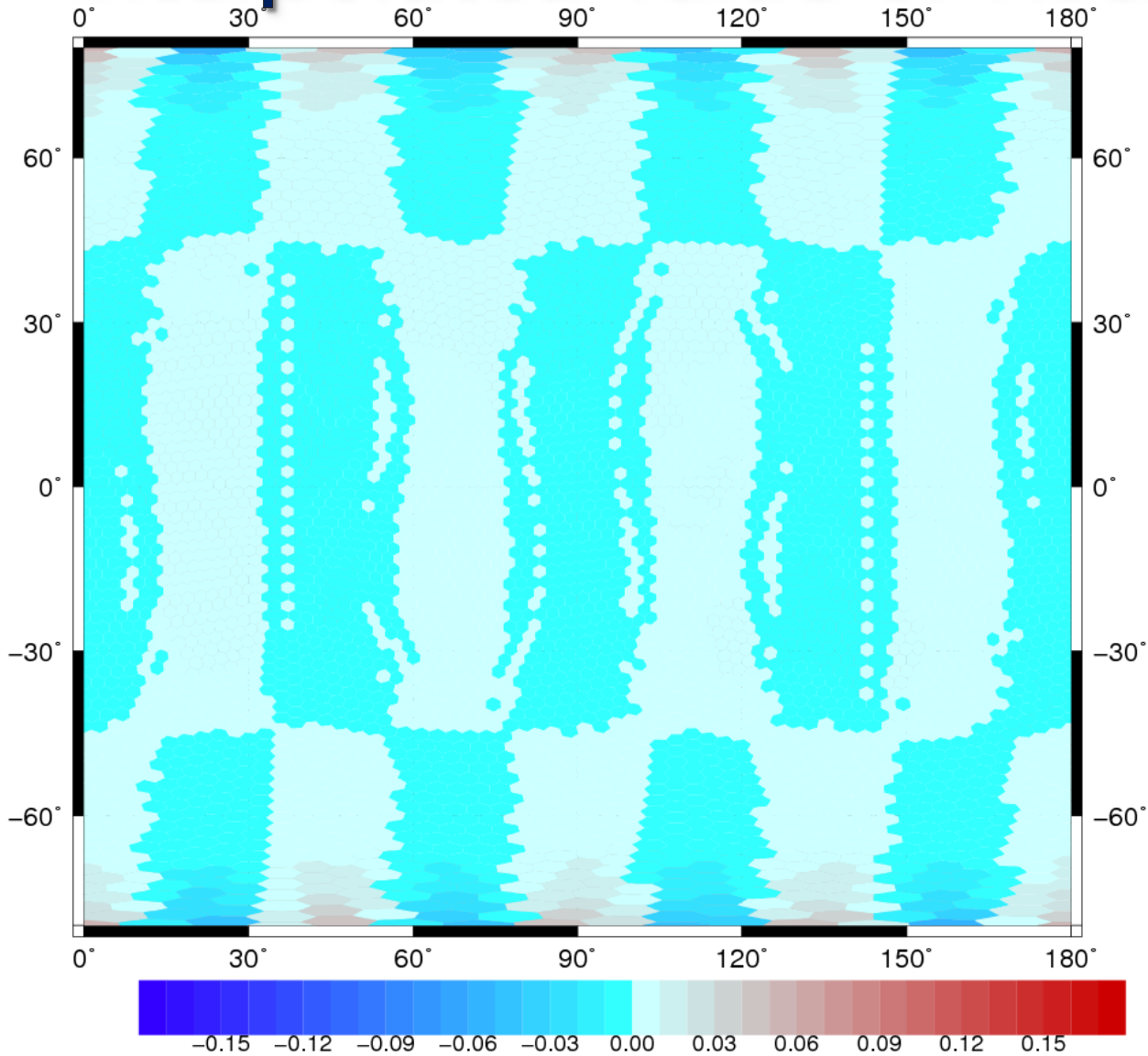
理論値と補間計算後の値を
比較

COCO to NICAM



left : COCO, right : NICAM, 80S~80N, 0E~180E

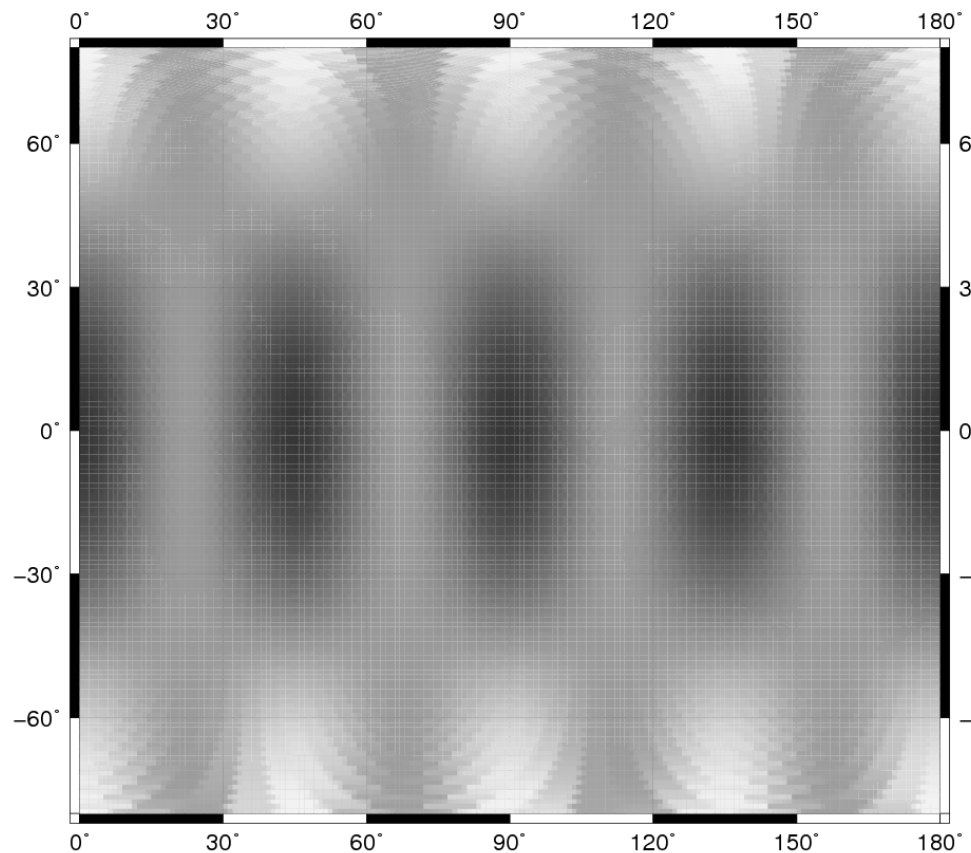
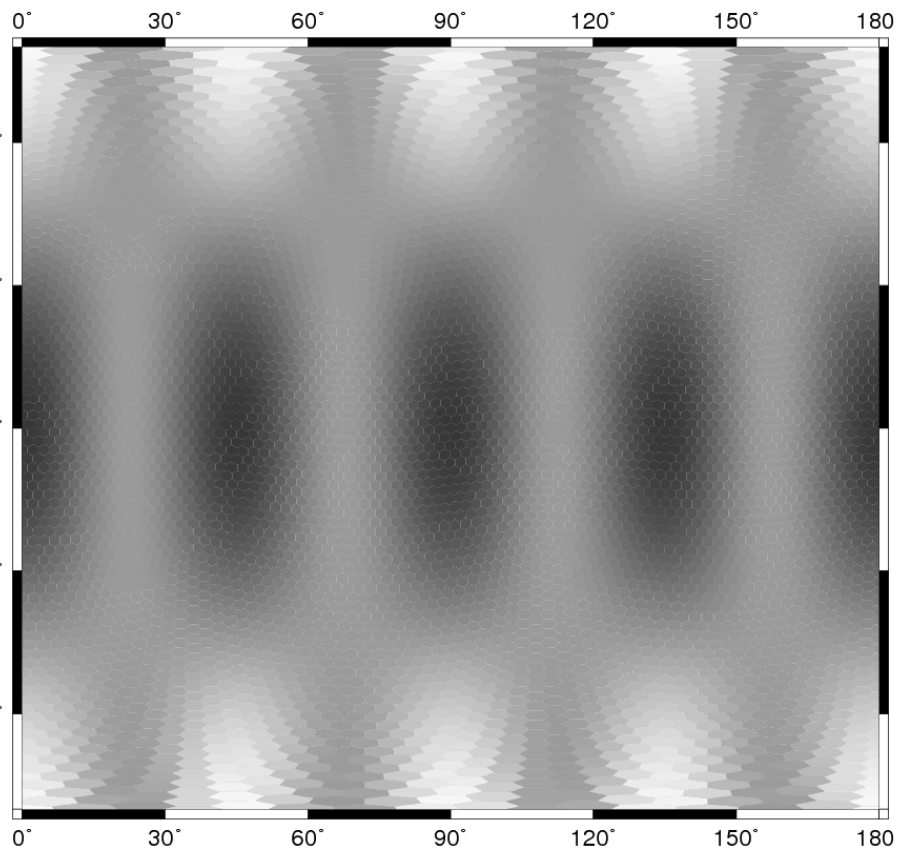
Difference between idealized and interpolated value of NICAM



From fine grid to coarse grid

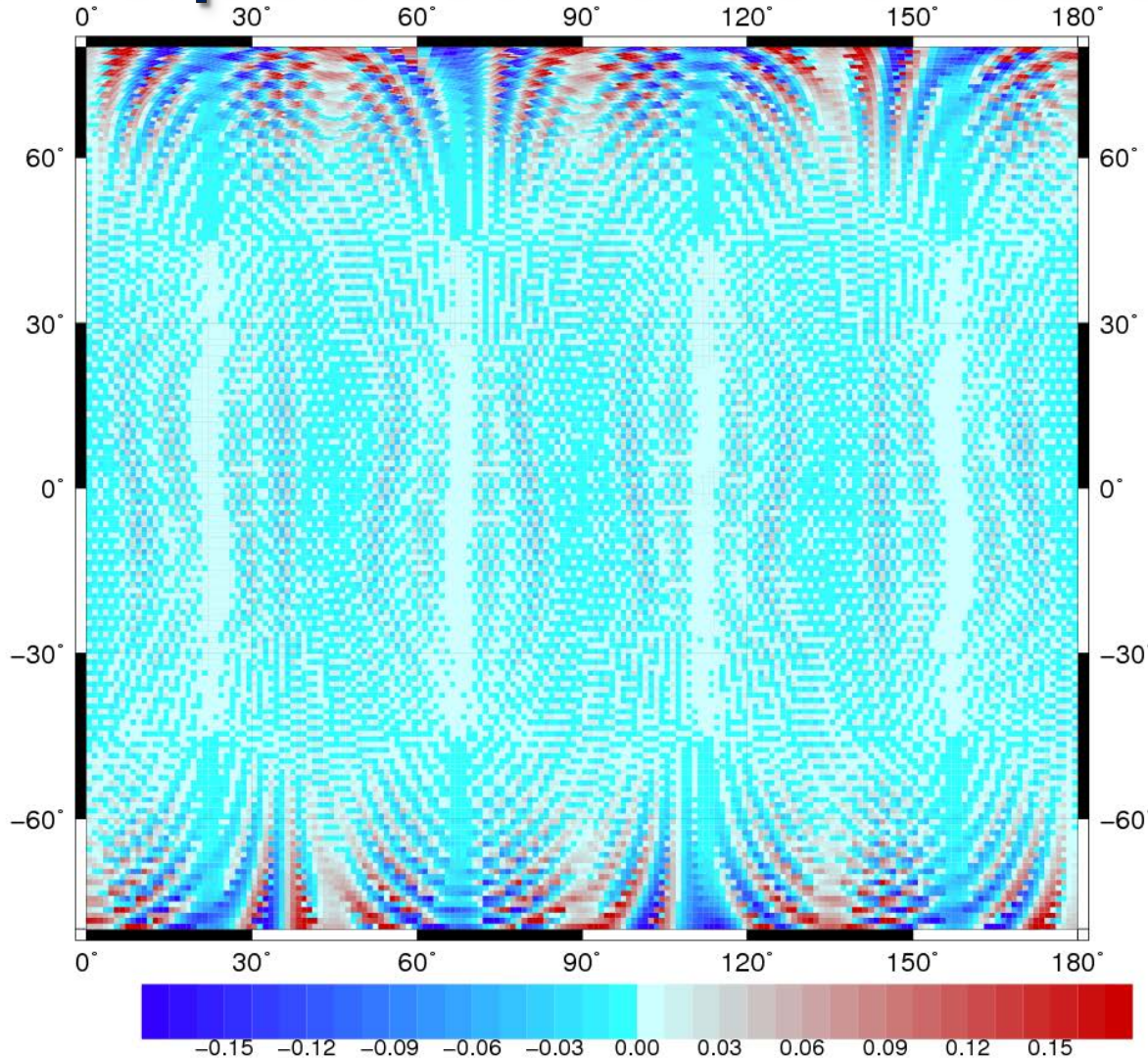
Error : about 3%

NICAM to COCO



Left : NICAM, right : COCO, 80S~80N, 0E~180E

Difference between idealized and interpolated value of COCO



From coarse grid to fine grid

Error : about 15%
* High latitude

NICAM-COCO coupling



▶ 格子数

- ▶ NICAM : GL5 (= about 220km) ノード数: 5程度
- ▶ COCO : 360x192 (= about 100km) ノード数: 8~16

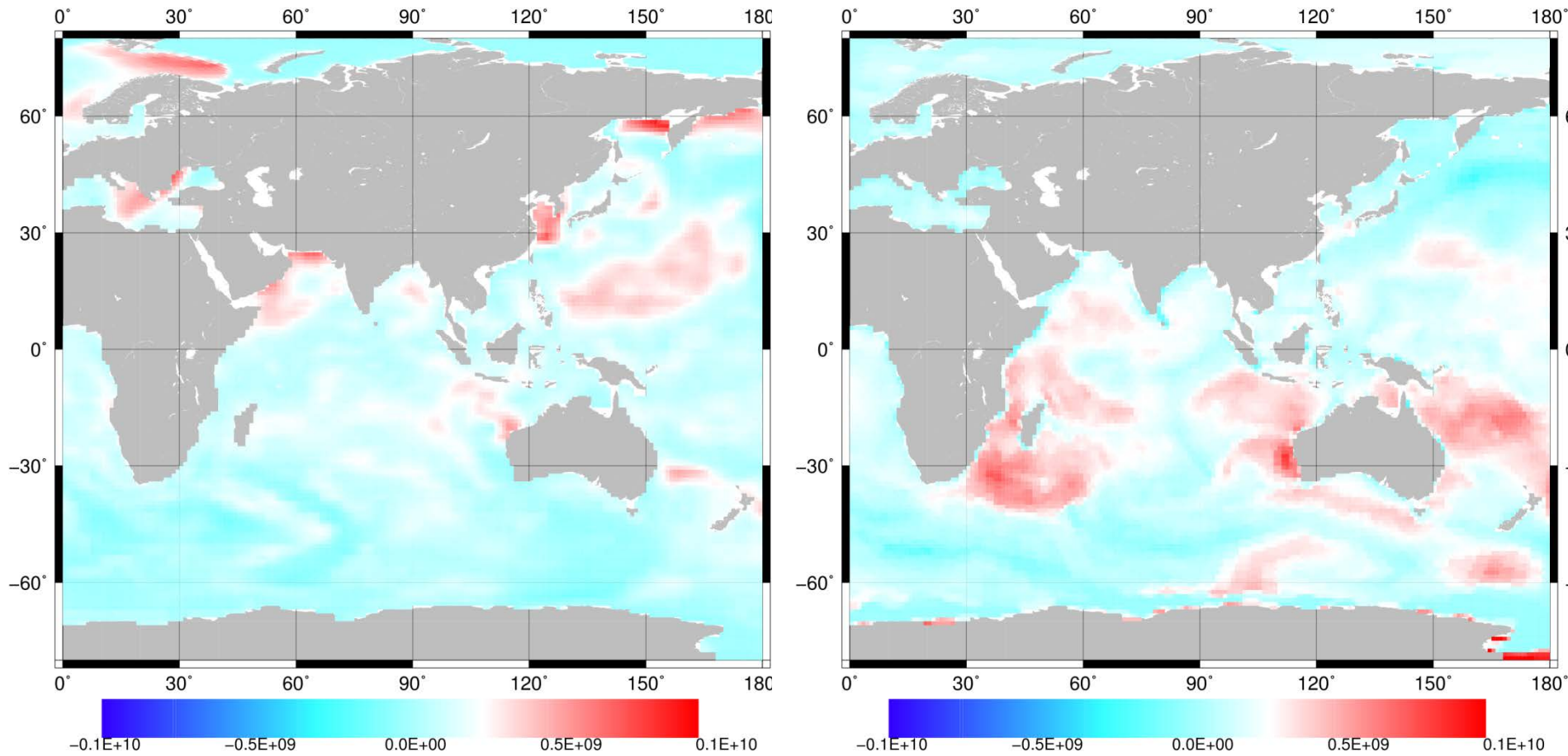
▶ 交換データ

- ▶ 基本的にMIROCの交換データ - α
 - ▶ 河川がない、海面の流れは使わない
 - ▶ 海氷のカテゴリ: シングルカテゴリ
- ▶ NICAM→COCO: 13 variables
- ▶ COCO→NICAM: 6 variables

MIROC AGCMと結合していたCOCOのコードをほとんど変更せずに結合できた

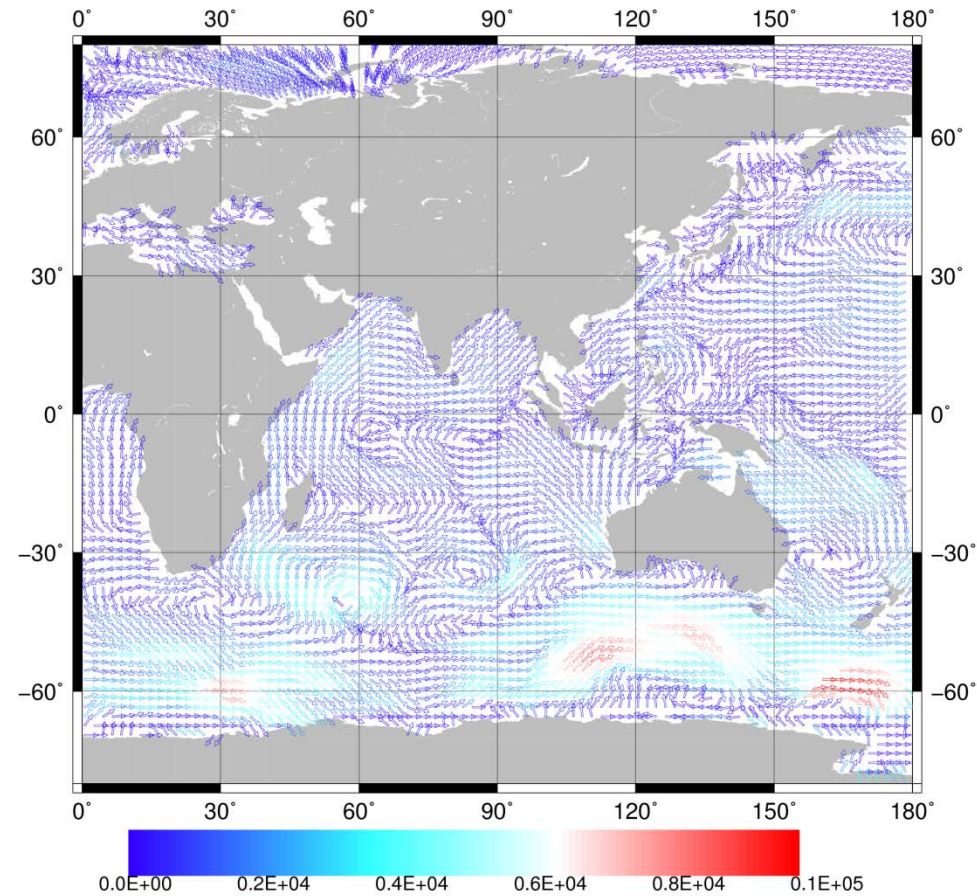
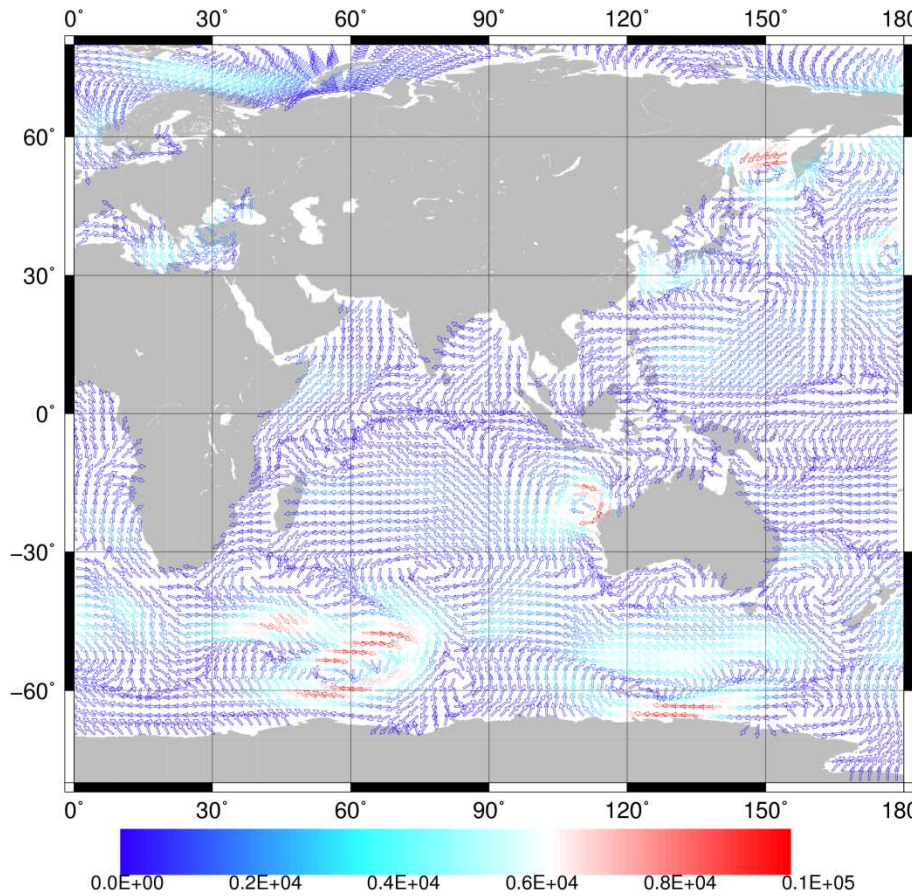
NICAM→COCO	COCO→NICAM
13 variables wind, precipitation etc.	6 variables SST, sea ice thickness etc.

Heat flux(TQAO)



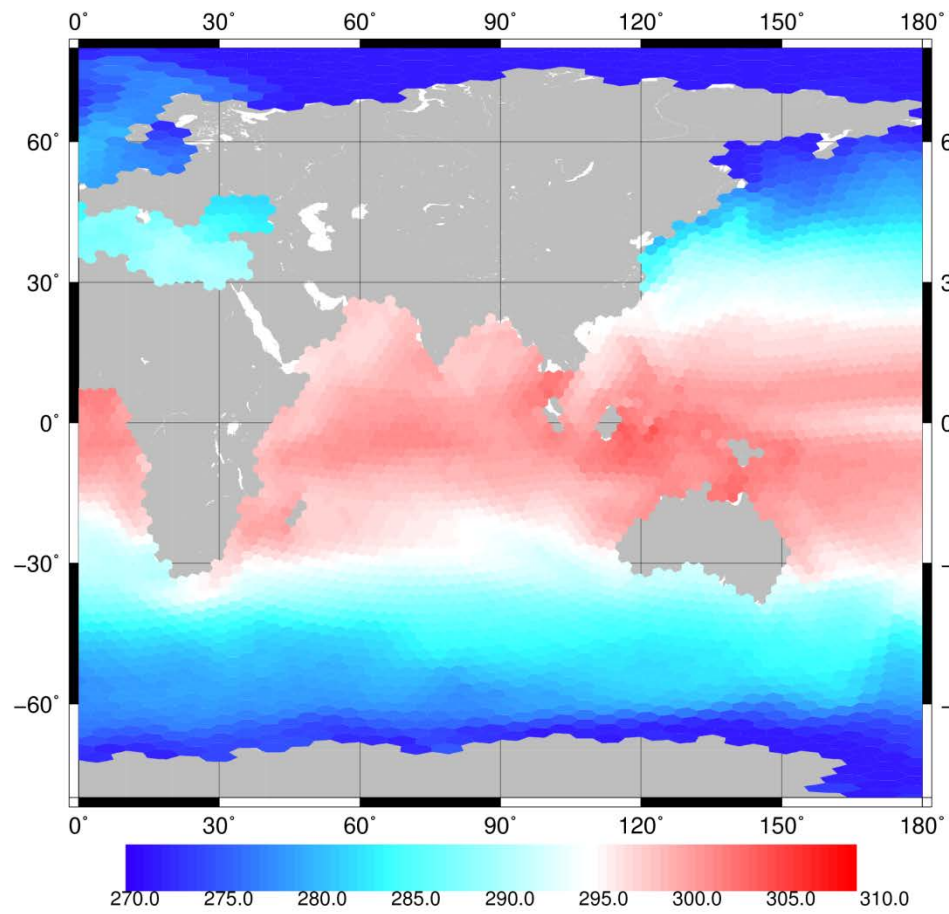
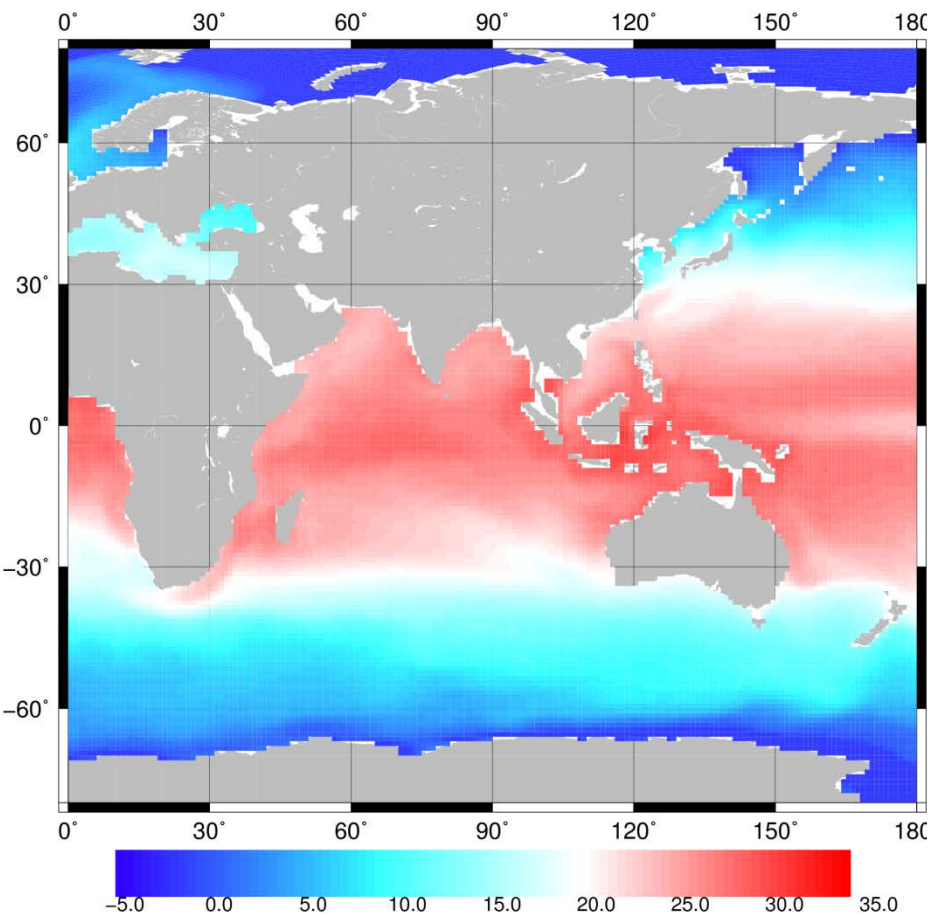
left : MIROC-A→COCO, right : NICAM→COCO

Momentum flux (TAO)



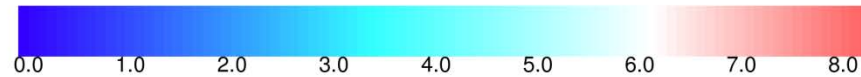
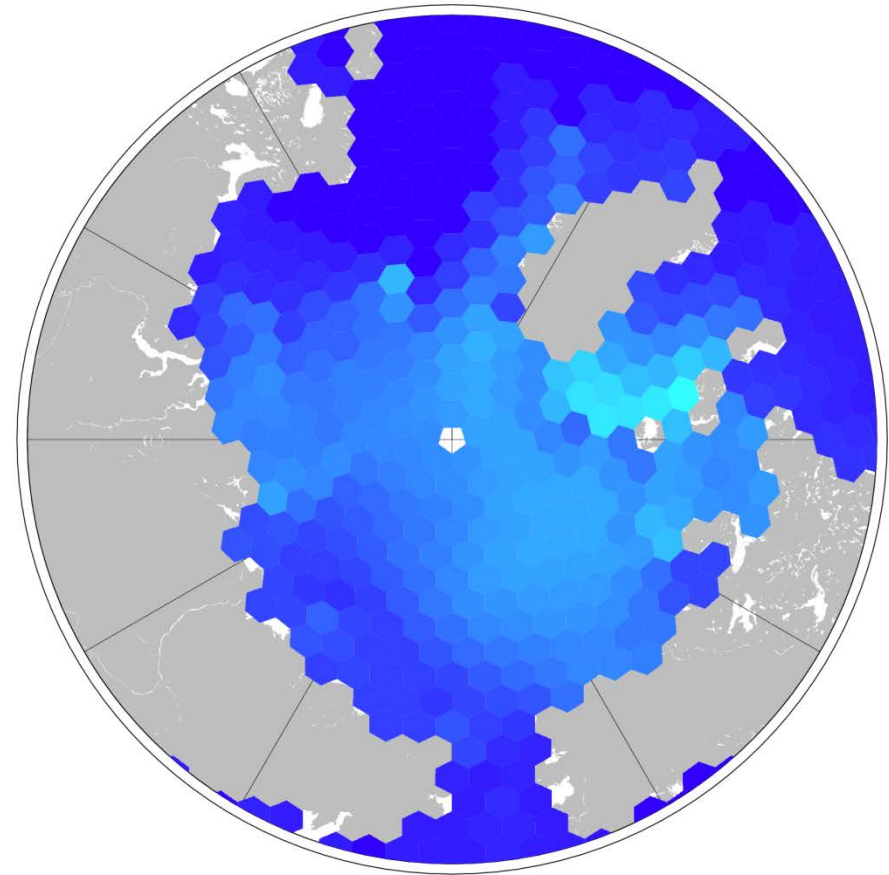
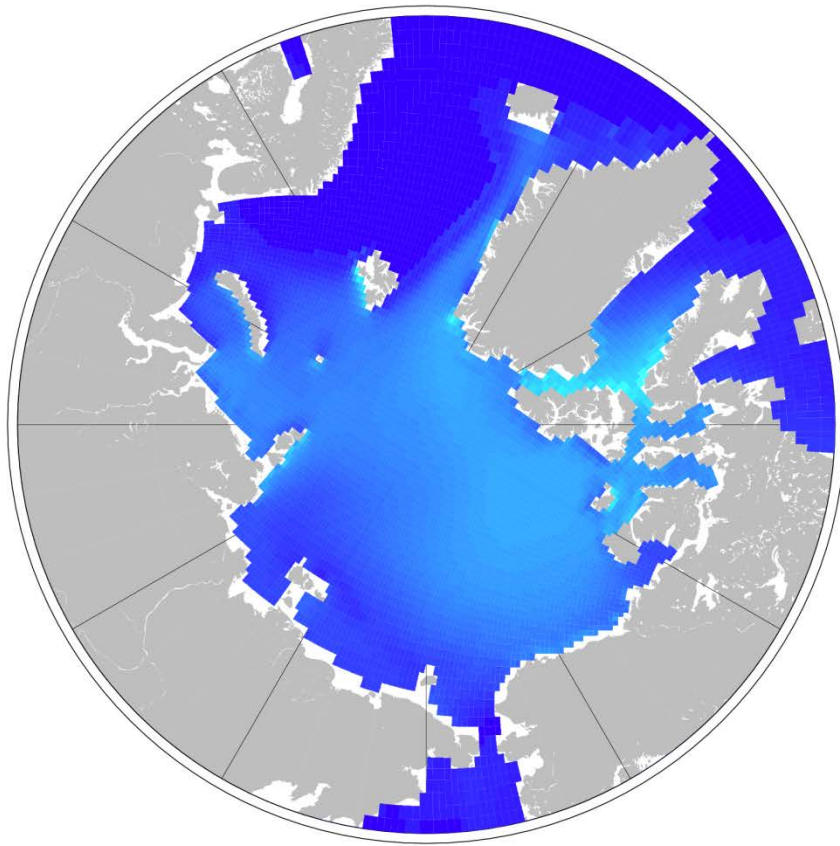
left : MIROC-A→COCO, right : NICAM→COCO

Sea surface temperature (OSST)



left : COCO, right : NICAM

Sea Ice Thickness (OHI)



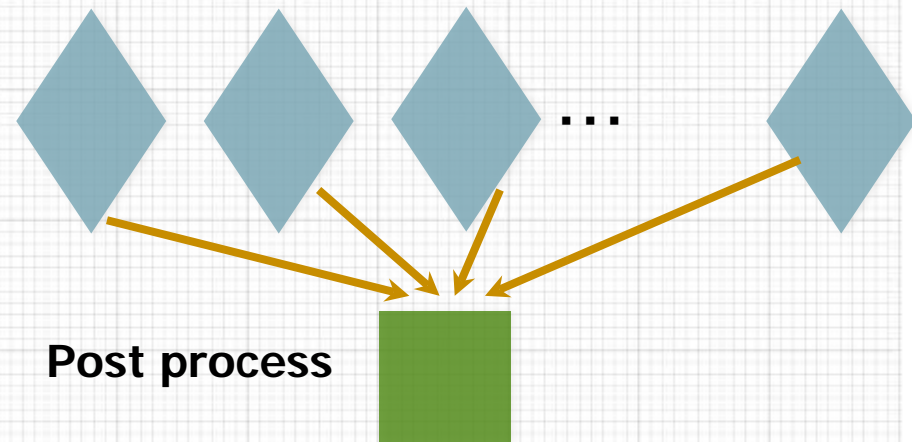
left : COCO, right : NICAM

About NICAM IO



- ▶ 二十面体格子のデータは解析に不便
 - ▶ zonal meanの計算
 - ▶ 多くの解析・作図ツールが緯度経度格子を前提
- ▶ NICAMオリジナルのIO tool: ico2ll
 - ▶ NICAMのhistory dataを緯度経度格子に変換
 - ▶ ポストプロセスとして実行
 - ▶ 並列に制限あり

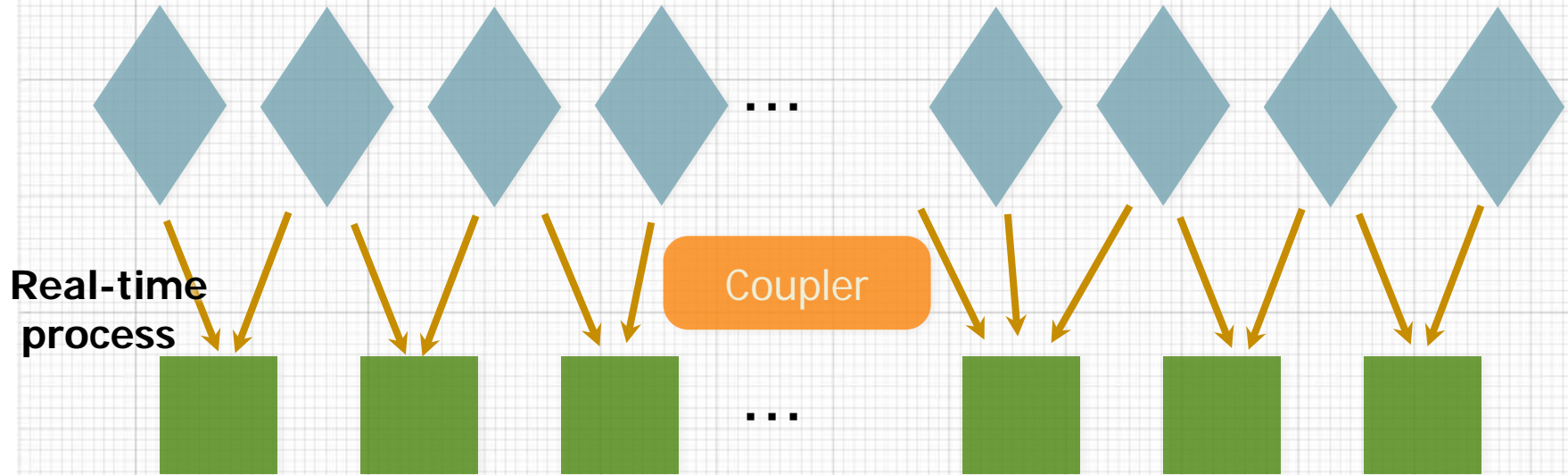
- ▶ Real-time parallel IO





NICAM nodes

grid system: Icosahedral

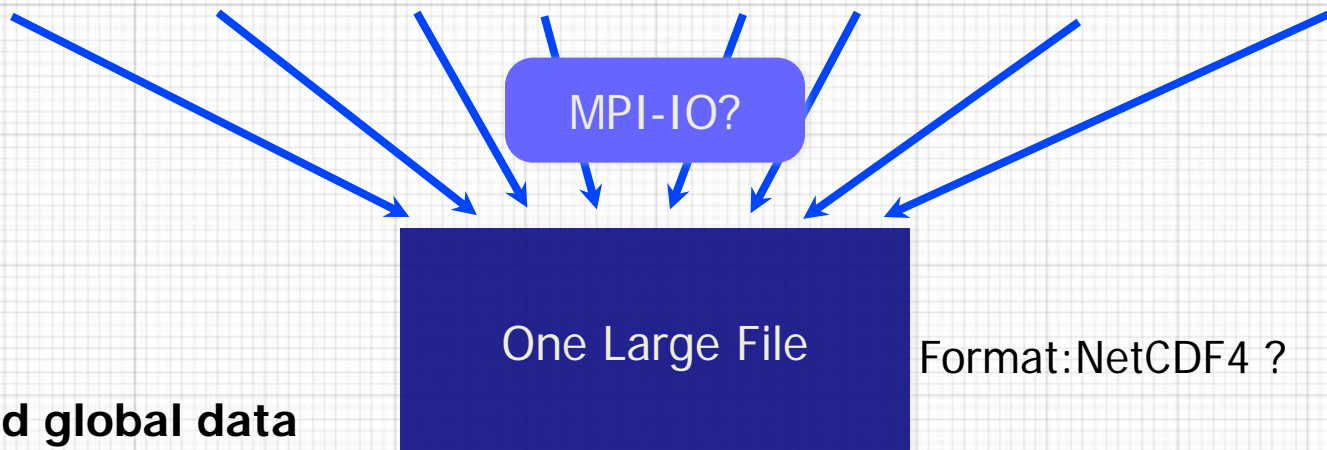


Real-time process

Coupler

I/O nodes

grid system: rectangle



One Large File

Format: NetCDF4 ?

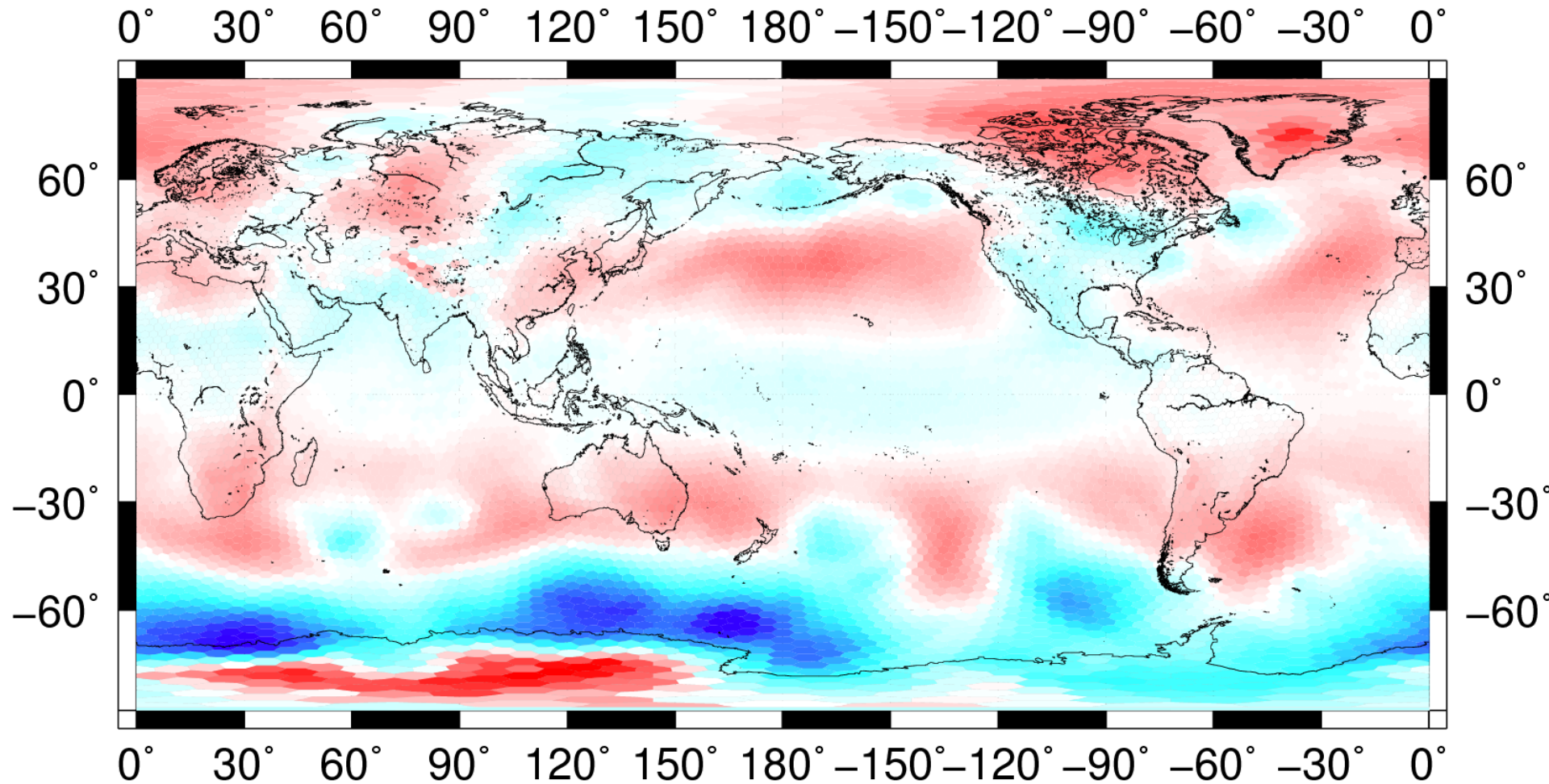
merged global data

IOコンポーネントの補間計算

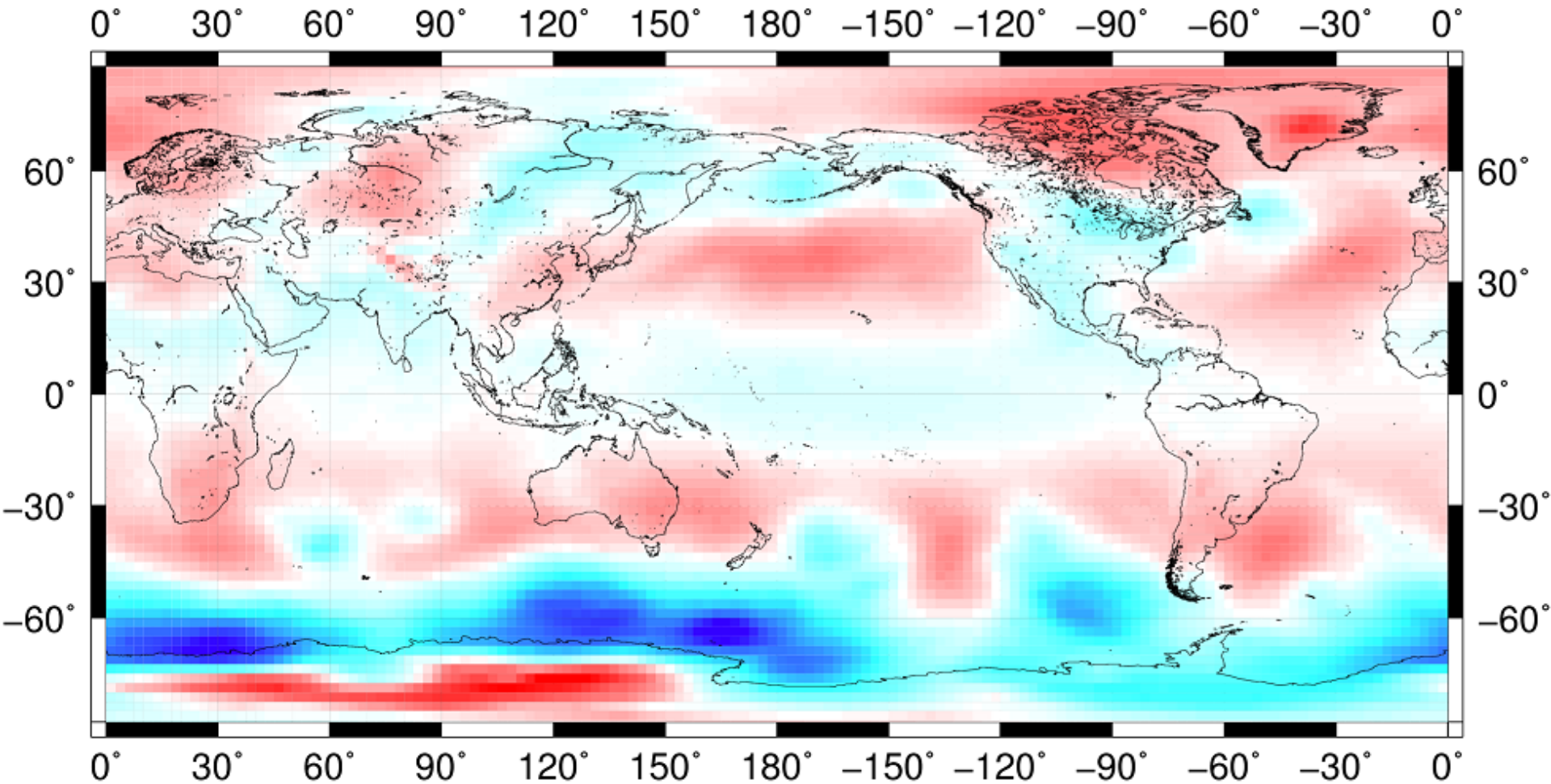
- ▶ Bi-linear interpolation
 - ▶ Ico2IIが採用している補間方法
 - ▶ mapping tableと係数は既存のico2II用を使用
- ▶ Control volume interpolation
 - ▶ NICAM-COCO結合で採用した補間方法
 - ▶ mapping tableと係数は、NICAM-COCO結合のために作成したmapping table計算プログラムを流用して計算
- ▶ 補間方法の切り替えはデータ毎

```
#===== 2D snapshot (every 1h) =====  
### liquid water path  
&nmhist item='sl_cldw',    file='ss_cldw',    output_type='SNAPSHOT', step=3, kmax=1 /  
### sea level pressure  
&nmhist item='sl_slp',    file='ss_slp',    output_type='SNAPSHOT', step=3, kmax=1, opt_intpl= 'C' /
```

SLP of NICAM

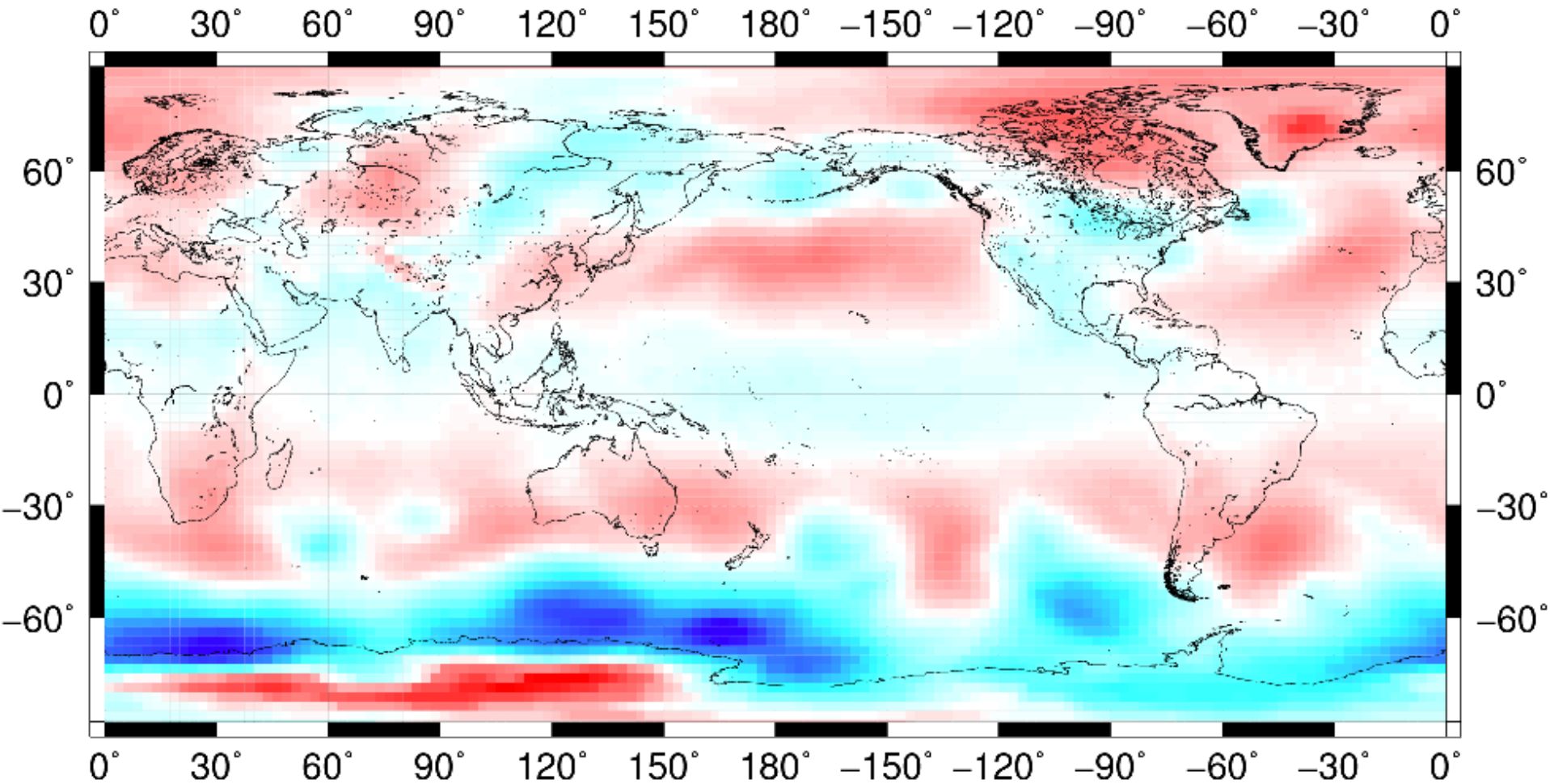


SLP of IO : Bi-linear



144x72

SLP of IO : Control volume



144x72



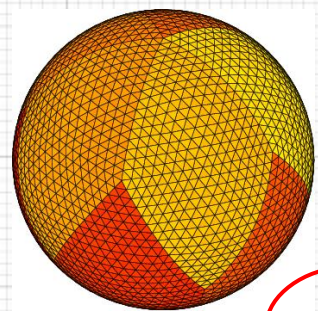
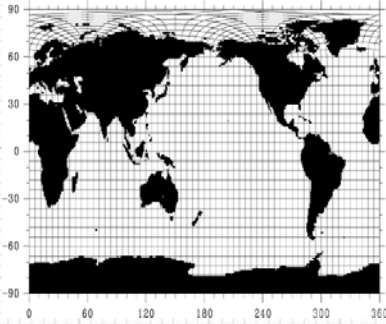
結合のコントロール

- ▶ 結合のコントロールは設定ファイルで行う
 - ▶ OCEAN_TYPE
 - ▶ LLPUT_TYPE

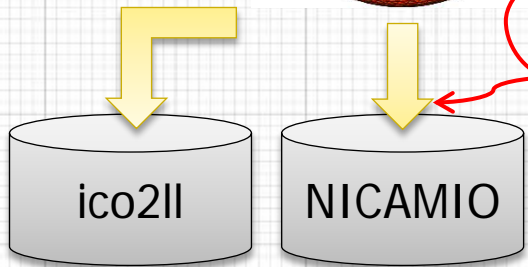
```
&RUNCONFPARAM
OCEAN_TYPE      = 'COCO',
LLOUT_TYPE     = 'MPMD'
```

Full coupling

Mixed layer
ocean model



Coupled by Jcup



```
&RUNCONFPARAM
OCEAN_TYPE     = 'mixed_latyer',
LLOUT_TYPE     = 'SPMD'
```

Stand alone without Jcup



- ▶ 今後の展開
 - ▶ 統合カップラー

統合カップラー



	Scup	Jcup	統合カップラー
(i,k,j)配列のサポート	○	○	○
鉛直内挿(鉛直座標変換)のサポート	○	△	○
座標変換用サブルーチンをユーザーが実装可能		○	○
カップラーリスタートファイルのサポート	○		○
各モデルのMPIノードへのより柔軟な割付		○	○
通信設定ファイルで変換テーブルを指定	○		○
通信設定ファイルの冗長性が低い	○		○
複数データをひとまとめに送受信		○	○
毎ステップ終わりにcoupler__sendをcallする必要なし		○	○
バンド毎のput_valueのサポート	○		coupler__sendをなくすため廃止?

開発方針

- ▶ ScupにありJcupにない機能をJcupに追加
 - ▶ Jcup内部に実装するかShellとして実装するか
- ▶ 追加する機能
 - ▶ (i,k,j) 配列のサポート
 - ▶ 鉛直内挿のサポート
 - ▶ リスタートファイルのサポート
 - ▶ 設定ファイルで変換テーブルを指定
 - ▶ 設定ファイルのフォーマット

Thank you for your attention

