MSSG-Aのモデル開発

（独）海洋研究開発機構(JAMSTEC)
地球シミュレータセンター(ESC)
マルチスケールモデリング研究グループ(MSSG)
馬場 雄也

モデルワークショップ -日本における気象・気候モデルの技術的現状と今後の展開-
名古屋大学工学研究科、2012/12/12

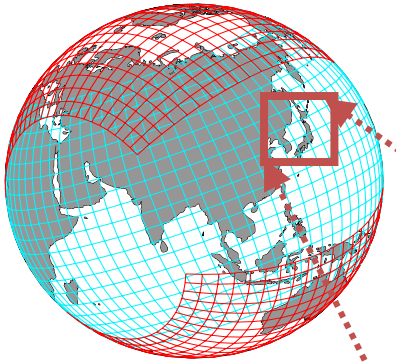
発表内容

1.MSSGの概要
2.力学コア・モデル要素概要
3.計算性能
4.モデル・モデル要素の構造
5.実装方法、コーディング上の工夫
6.ライブラリ開発と現行モデルの親和性
7.ライブラリ開発への提言

MSSG-A

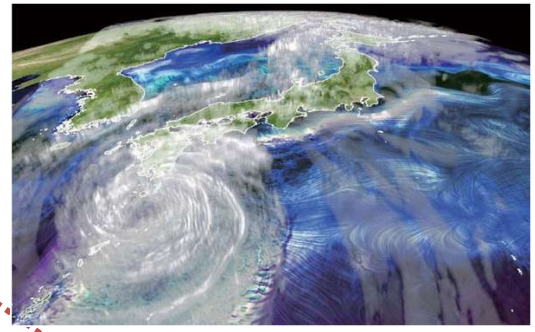
“Multi-Scale Simulator for the Geo-environment”

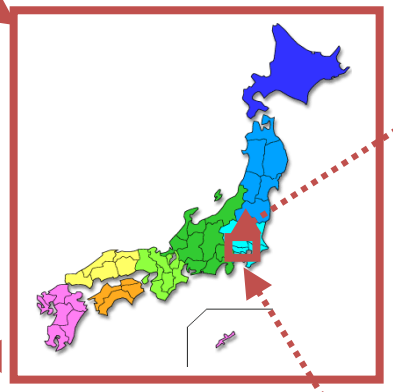
“Yin-Yang grid”



Global scale

Resolution: 1～100 km

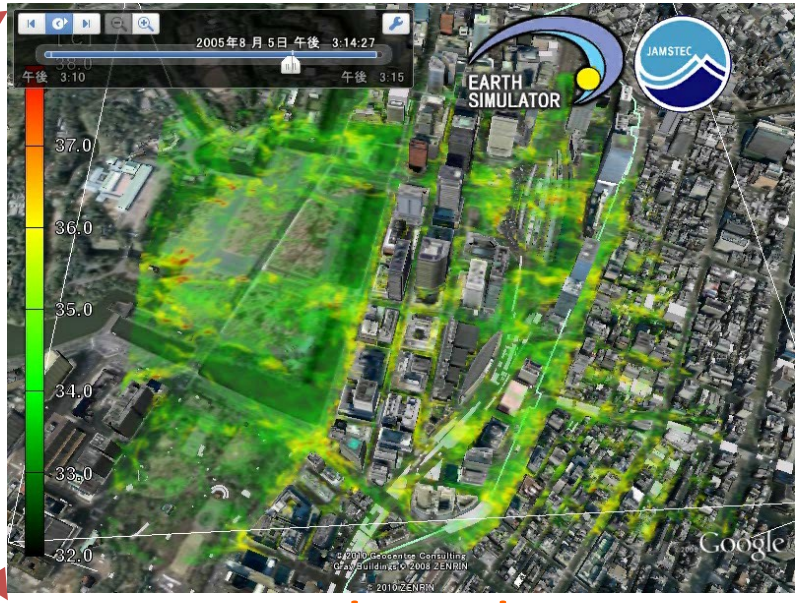




Mesoscale

Resolution:100m～5km

Tokyo, Maruno-uchi area



Urban scale

1m～100m (homogeneous grid)

- ✓ Dynamical core: Baba et al. (2010) MWR
- ✓ Quick solver: Baba & Takahashi (2011) JMSJ
- ✓ Advection scheme for CRM: Baba &Takahashi (2012) QJRMS

MSSG-A開発の経緯

メインフレーム開発期間

- ✓2003年- 既存モデルの調査、大気海洋力学コア・プロトタイプの開発、物理過程の移植、並列化手法の確立、CIP-CSLR法の実装
- ✓2004年- 適応型動的格子法の実装・拡張、陸面・海氷モデルの実装
- ✓2005年- 海洋自由表面スキームの実装、非静力学への拡張、最適化・ベクトル化

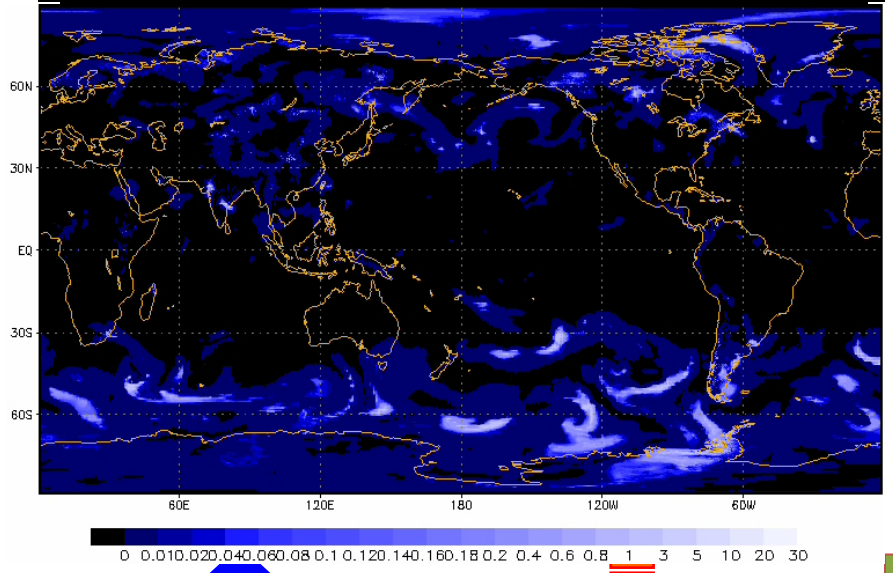
最適化・検証期間

- ✓2006年- 全球1.9-km実験、季節進行実験、台風制御実験、物理過程のデバッグ
- ✓2007年- 長期積分実験、季節進行実験、海洋全球実験、練馬豪雨実験、計算性能最適化、都市スケールシミュレーション対応
- ✓2008年- CAM3放射コード実装、COARE3.0スキーム実装、MJO実験、ES2移植・最適化

参考：NEC報告書（2003）～(2008)

MSSG-A

Horizontal:1.9km, Vertical:32layers, 14 days integration

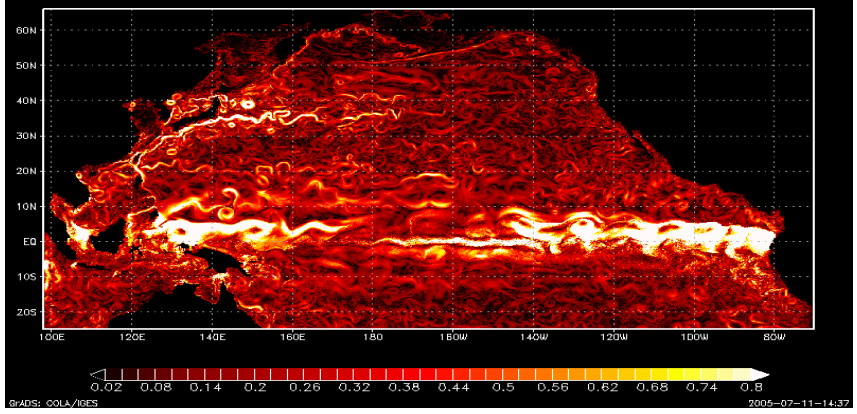


SST

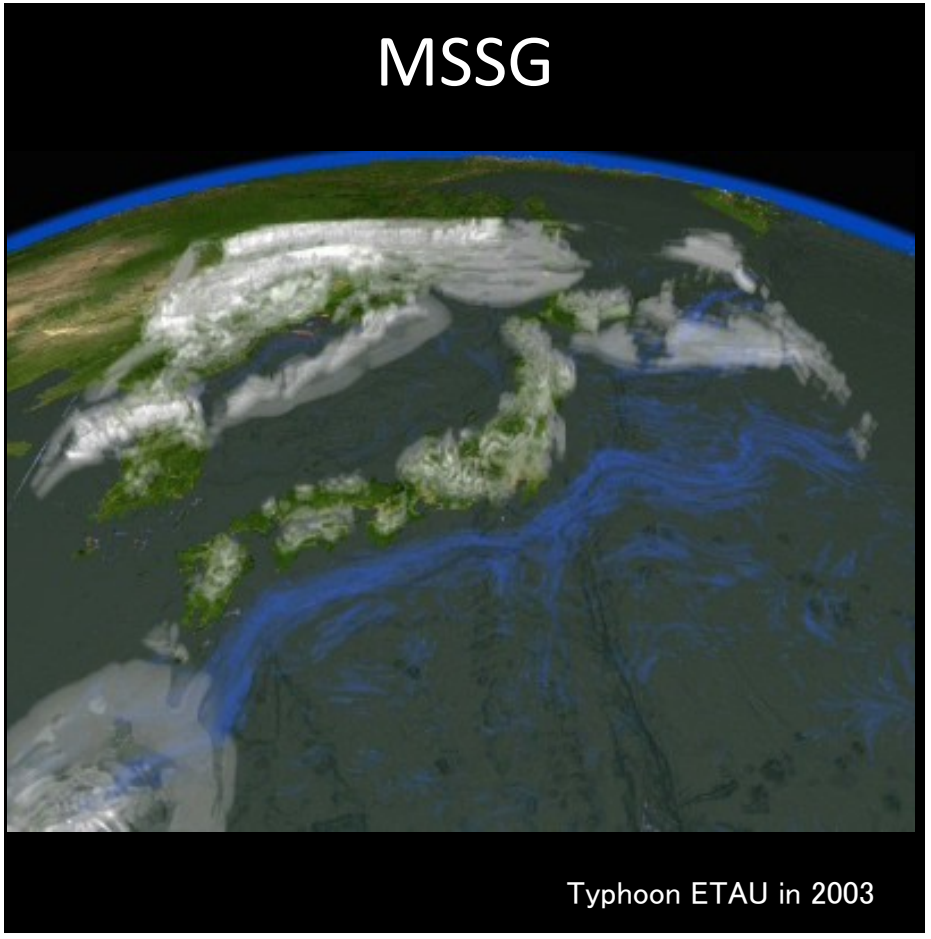
Heat, Flesh water, Momentum fluxes

MSSG-O

Horizontal: 2.78 km, vertical: 40 layer, 15 years integration



MSSG

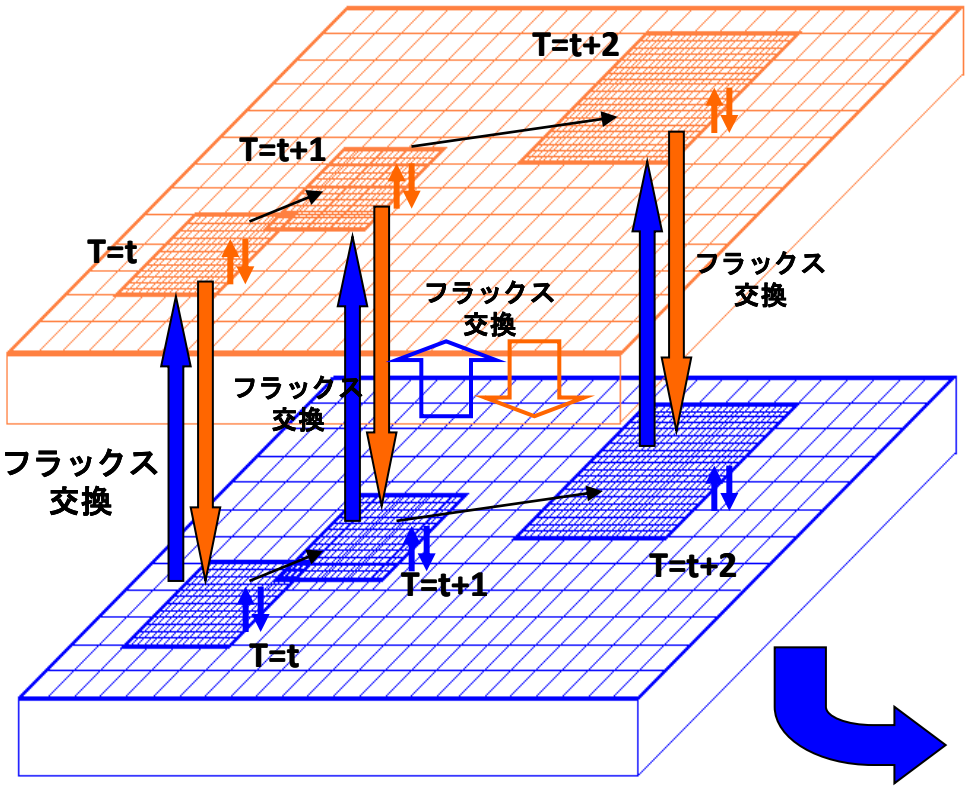


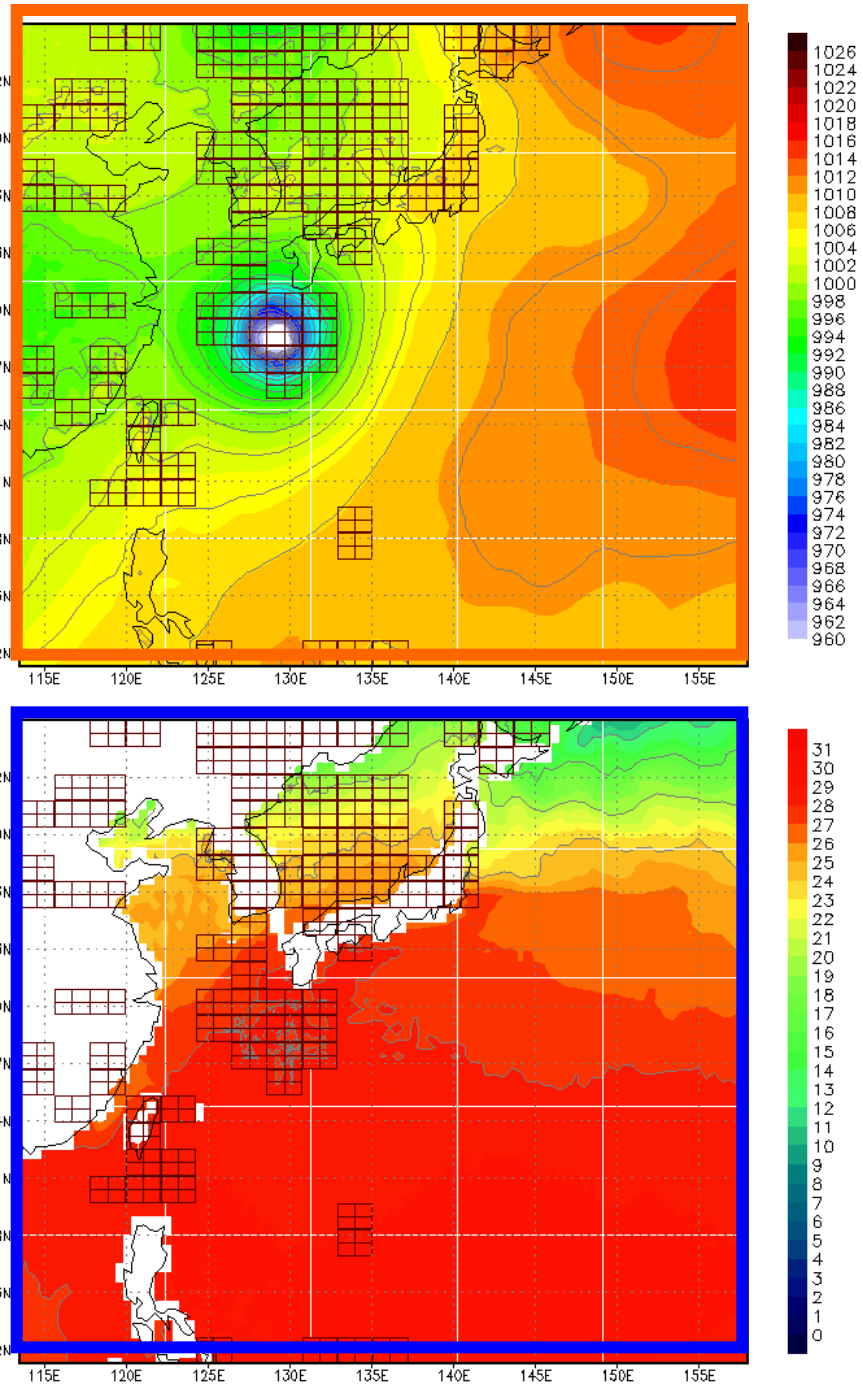
Typhoon ETAU in 2003

Sea Ice

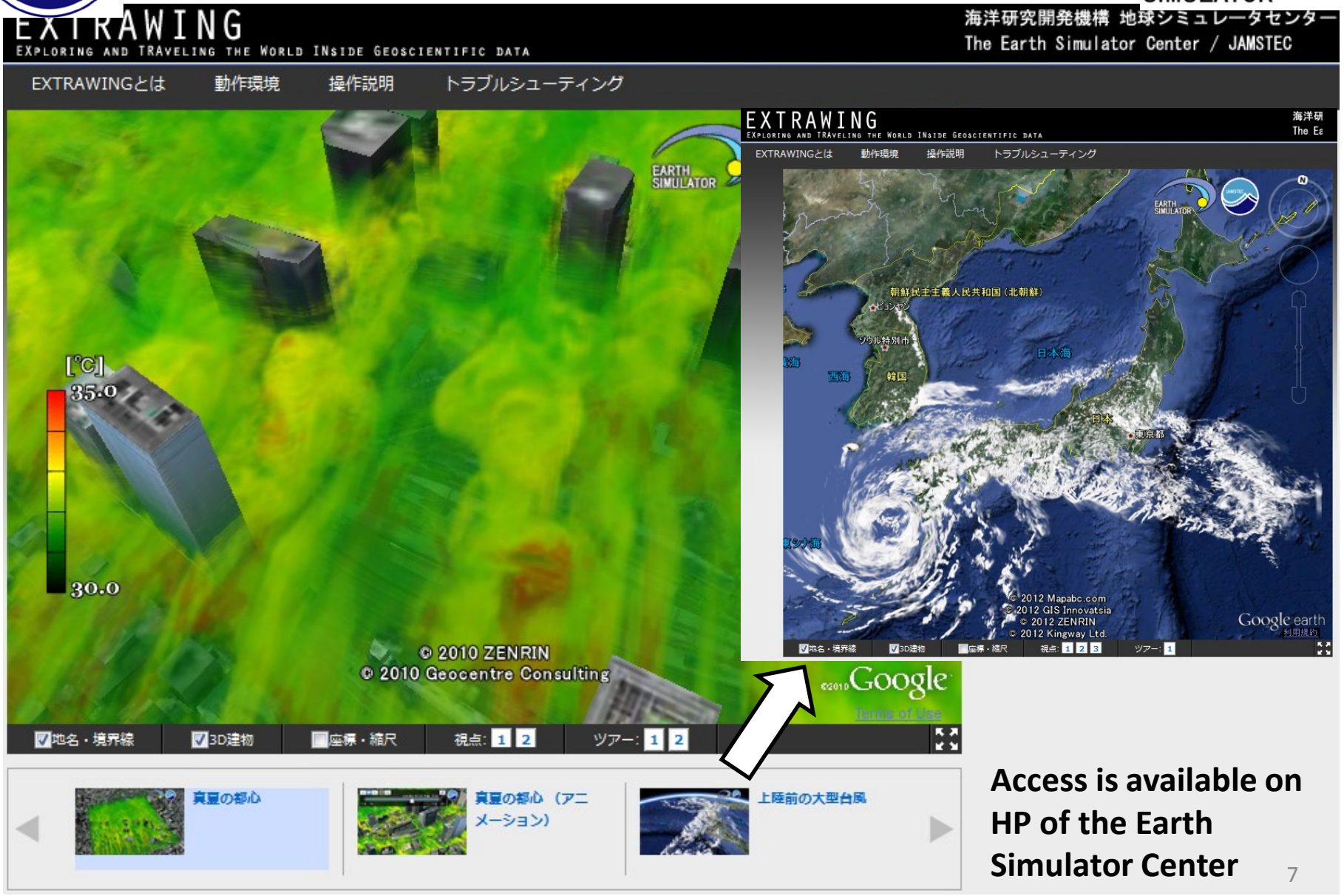
Okhotsk-sea (1/12deg.)

大気，海洋それぞれへ実装
動的適応型格子
(Adaptive Mesh Refinement: AMR)



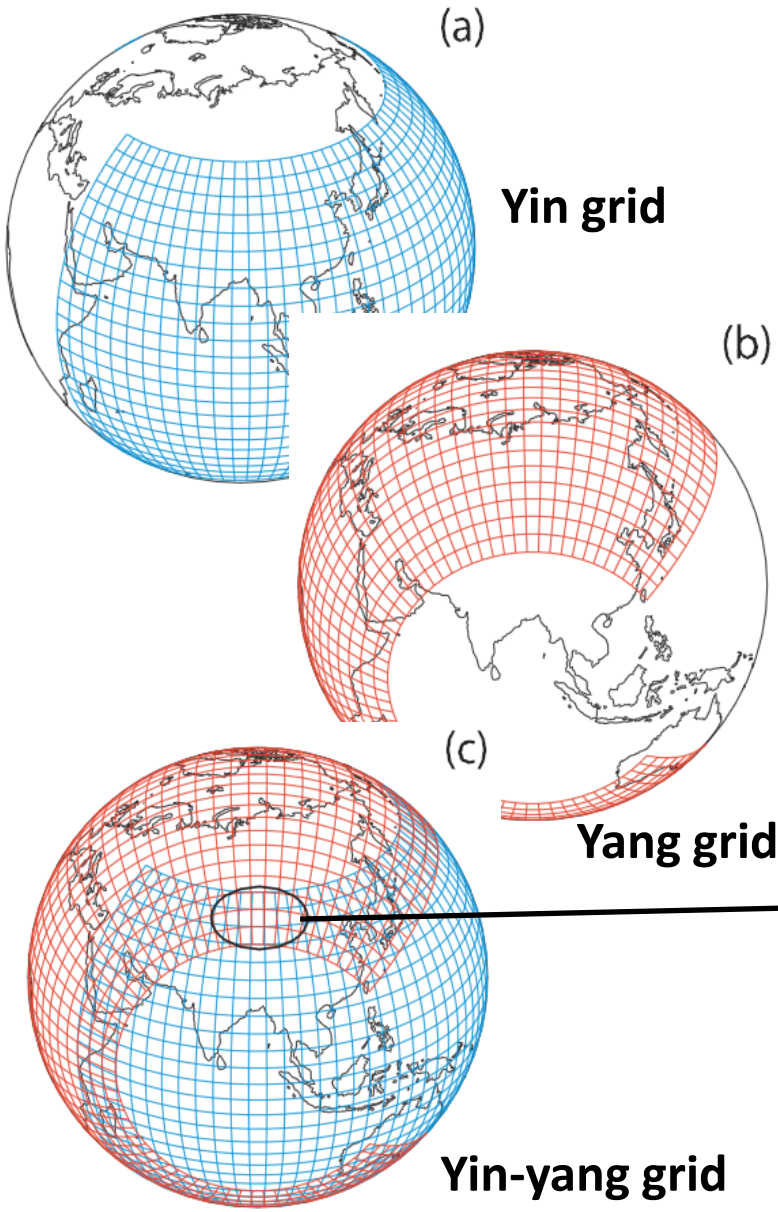


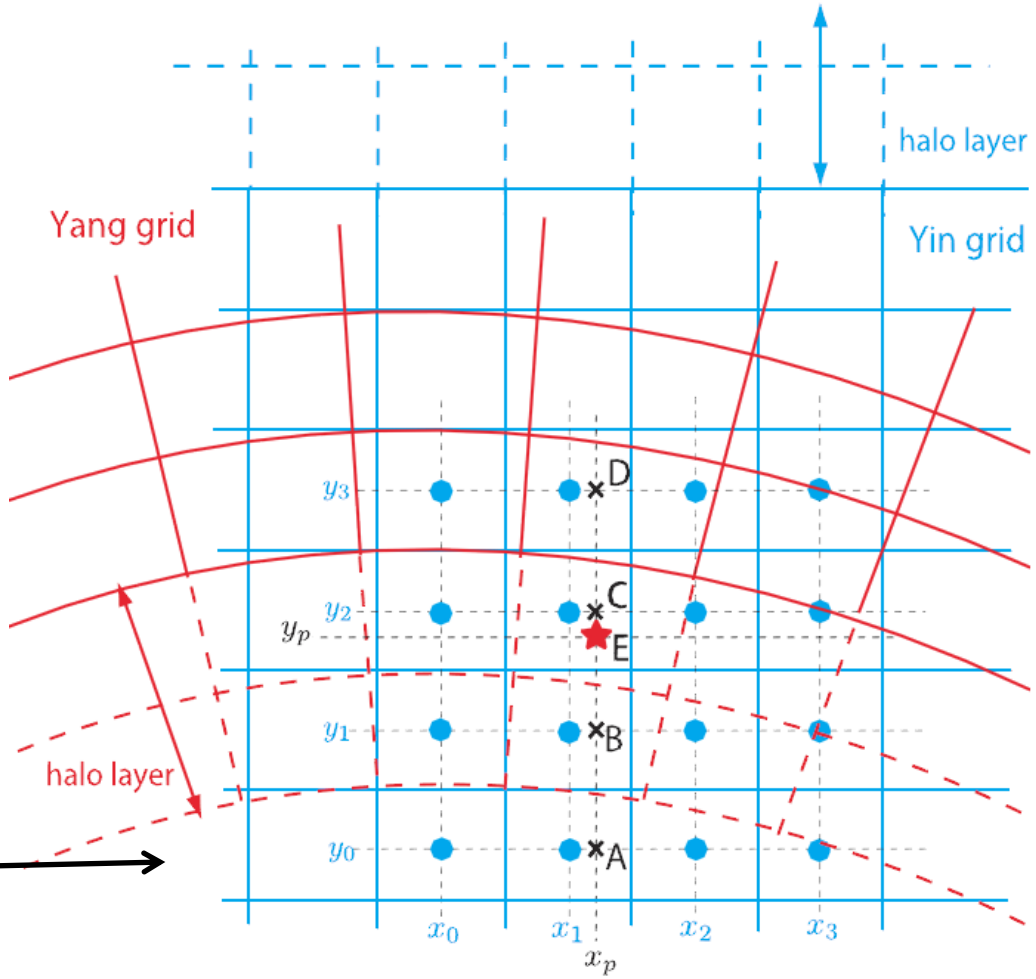
EXTRAWING



Access is available on HP of the Earth Simulator Center

The Yin-Yang grid





Schematic diagram of data interpolation at the grids' bound. Third-order Lagrange interpolation is used in the over3 overset configuration.

8

HE-VI (horizontal-explicit vertical-implicit) method

Mass conservation

$$\frac{R^{*\tau+\Delta\tau}-R^{*\tau}}{\Delta\tau}+\nabla_H\cdot\mathbf{V_H}^{\tau+\Delta\tau}+\frac{\partial}{\partial z^*}\left(\mathbf{V_H}^{\tau+\Delta\tau}\cdot\mathbf{G^z}+\frac{W^{*\tau+\Delta\tau}}{G^{1/2}}\right)=-\frac{\partial}{\partial z^*}\left(\frac{W^t}{G^{1/2}}\right)$$

Momentum conservation

$$\frac{\mathbf{V_H}^{*\tau+\Delta\tau}-\mathbf{V_H}^{*\tau}}{\Delta\tau}+\nabla_H P^{*\tau}+\frac{\partial}{\partial z^*}(\mathbf{G^z}P^{*\tau})=-\nabla_H P^t-\frac{\partial}{\partial z^*}(\mathbf{G^z}P^t)-\mathbf{A_H}^t-\mathbf{C_H}^t+\mathbf{F_{V_H}}^t$$

$$\frac{W^{*\tau+\Delta\tau}-W^{*\tau}}{\Delta\tau}+\frac{\partial}{\partial z^*}\left(\frac{P^{*\tau+\Delta\tau}}{G^{1/2}}\right)+R^{*\tau+\Delta\tau}g=-\frac{\partial}{\partial z^*}\left(\frac{P^t}{G^{1/2}}\right)-R^tg-A_z^t-C_z^t+F_W^t$$

Energy conservation

$$\frac{P^{*\tau+\Delta\tau}-P^{*\tau}}{\Delta\tau}+(\gamma-1)\,p^t\left[\nabla_H\cdot\mathbf{v}_H^{\tau+\Delta\tau}+\frac{\partial}{\partial z^*}\left(\mathbf{v_H}^{\tau+\Delta\tau}\cdot\mathbf{G^z}+\frac{w^{*\tau+\Delta\tau}}{G^{1/2}}\right)\right]$$

$$+\nabla_H\cdot\left(p^t\mathbf{v_H}^{\tau+\Delta\tau}\right)+\frac{\partial}{\partial z^*}\left(p^t\mathbf{v_H}^{\tau+\Delta\tau}\cdot\mathbf{G^z}+\frac{p^tw^{*\tau+\Delta\tau}}{G^{1/2}}\right)$$

$$=-(\gamma-1)\,p^t\frac{\partial}{\partial z^*}\left(\frac{w^t}{G^{1/2}}\right)-\frac{\partial}{\partial z^*}\left(\frac{p^tw^t}{G^{1/2}}\right)+(\gamma-1)q_{heat}^t.$$

HE-VI (horizontal-explicit vertical-implicit) method

Mass conservation

$$\frac{R^{*\tau+\Delta\tau} - R^{*\tau}}{\Delta\tau} + \frac{\partial}{\partial z^*}\left(\frac{W^{*\tau+\Delta\tau}}{G^{1/2}}\right) = S_R,$$

Momentum conservation

$$\frac{W^{*\tau+\Delta\tau} - W^{*\tau}}{\Delta\tau} + \frac{\partial}{\partial z^*}\left(\frac{P^{*\tau+\Delta\tau}}{G^{1/2}}\right) + R^{*\tau+\Delta\tau} g = S_W,$$

Energy conservation

$$\frac{P^{*\tau+\Delta\tau} - P^{*\tau}}{\Delta\tau} + \gamma p^t \frac{\partial}{\partial z^*}\left(\frac{W^{*\tau+\Delta\tau}}{\rho^t G^{1/2}}\right) + \frac{W^{*\tau+\Delta\tau}\tilde{g}}{G^{1/2}} = S_P,$$

Helmholtz equation

$$\frac{\gamma}{G^{1/2}}\delta_{z^*} p^t \delta_{z^*} \frac{W^{*\tau+\Delta\tau}}{\rho^t} + \frac{1}{G^{1/2}}\delta_{z^*}\left(W^{*\tau+\Delta\tau}\tilde{g}\right) + \frac{g}{G^{1/2}}\delta_{z^*} W^{*\tau+\Delta\tau} - \frac{W^{*\tau+\Delta\tau}}{\Delta\tau^2}$$

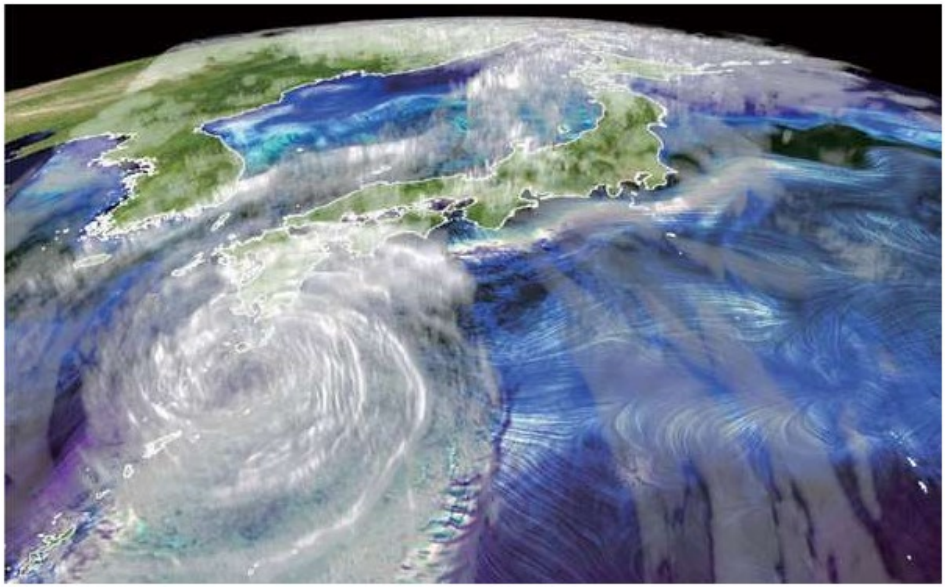
$$= -\frac{\left(W^{*\tau} + \Delta\tau S_W\right)}{\Delta\tau} + \frac{1}{\Delta\tau G^{1/2}}\delta_{z^*}\left(P^{*\tau} + \Delta\tau S_P\right) + \frac{g}{\Delta\tau}\left(R^{*\tau} + \Delta\tau S_R\right),$$

Feature of this scheme

1. Horizontal acoustic wave is explicit but implicit in vertical.
2. Vertical equations are formulated as conservative.

メソスケール・境界層スケールに応じた力学コアの使用

メソスケールシミュレーション



2003年台風10号のシミュレーション

Z*系座標
（地形に沿ったterrain-coordinate）

境界層スケールシミュレーション

皇居・丸の内地域の風況シミュレーション

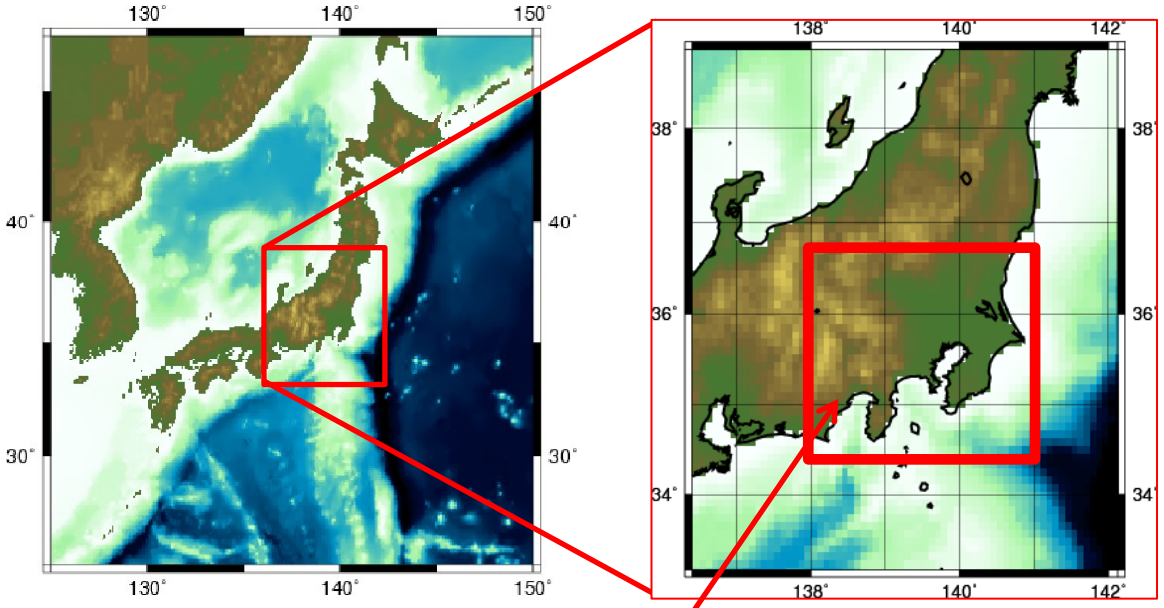
Z系座標
(絶対座標系)

✓方程式はz*系とz系で同じもので、座標変換したものを使う。
✓Z系では3D Helmholz方程式を解くのでAMGソルバーを使用。

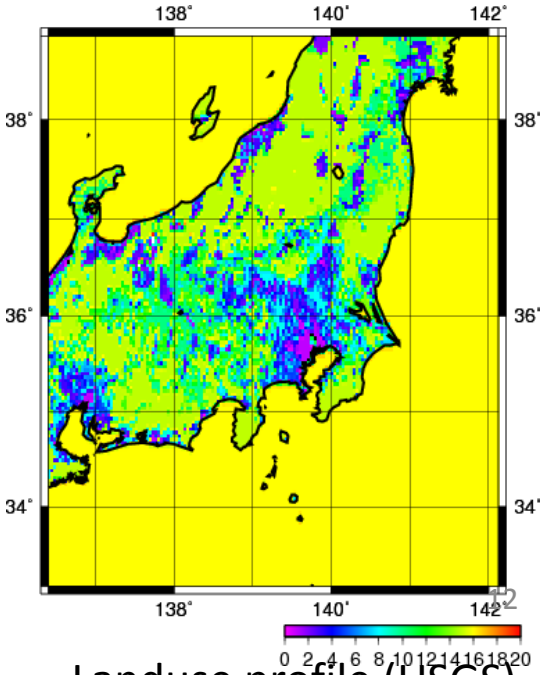
Regional climate run using 2M scheme

- ✓ Target area : Tokyo plane (600km2)
- ✓ Analyzed area : 34.5-37.5, 137.5-141
- ✓ Side boundary : MSM(every 6 hour)
- ✓ Resolution : 4km, 50 layers (20km)
- ✓ Time integration : yr:2006～2010 (5 year, from Jul. to Sep.)

- ✓ Microphysics : 1M vs F2M
- ✓ Radiation : MSTRNX
- ✓ PBL scheme : MYNN level-2.5
- ✓ Urban canopy scheme : Kusaka et al. (2001) (from WRF ver.3)



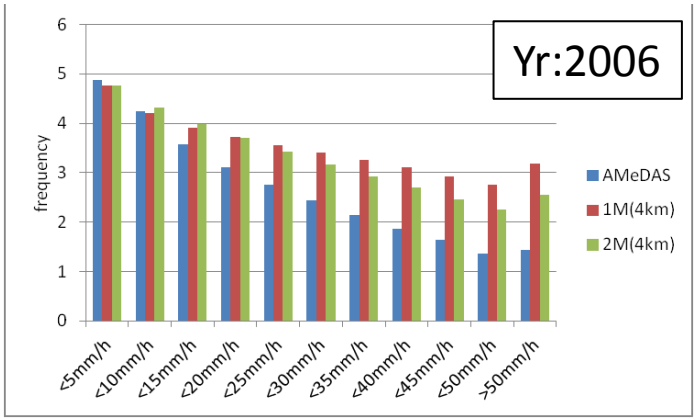
Targeted area

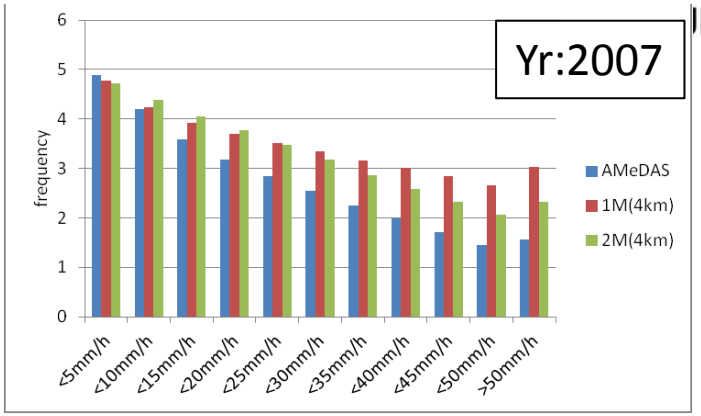


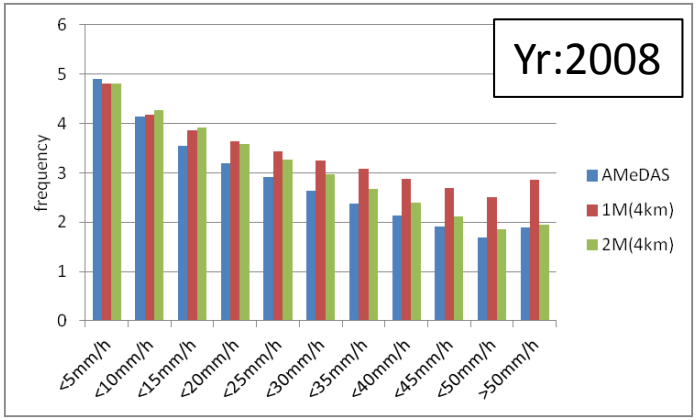
Landuse profile (USGS)

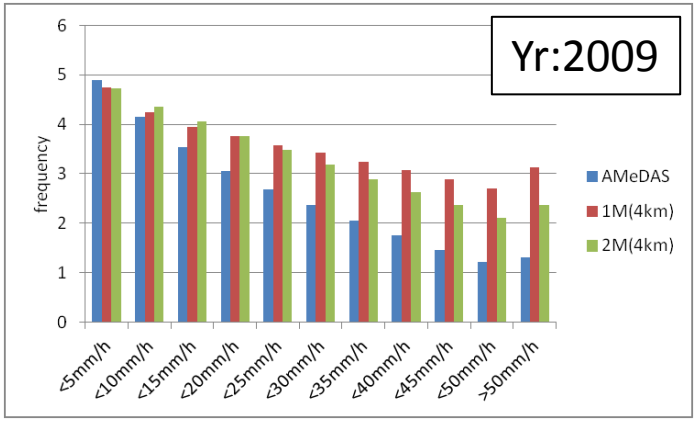
(urban area: index=1,2)

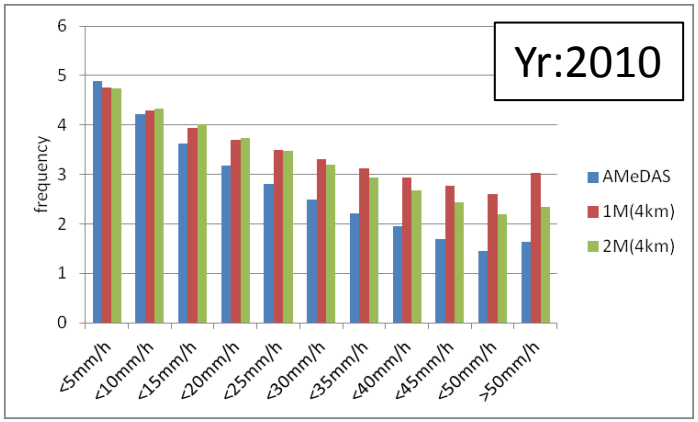
Comparison of precipitation (vs OBS)







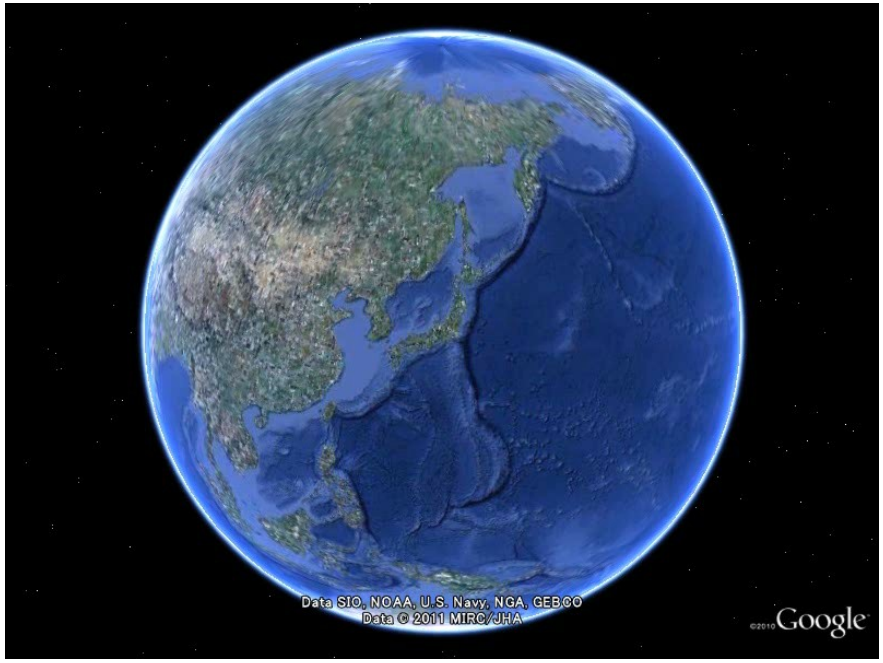


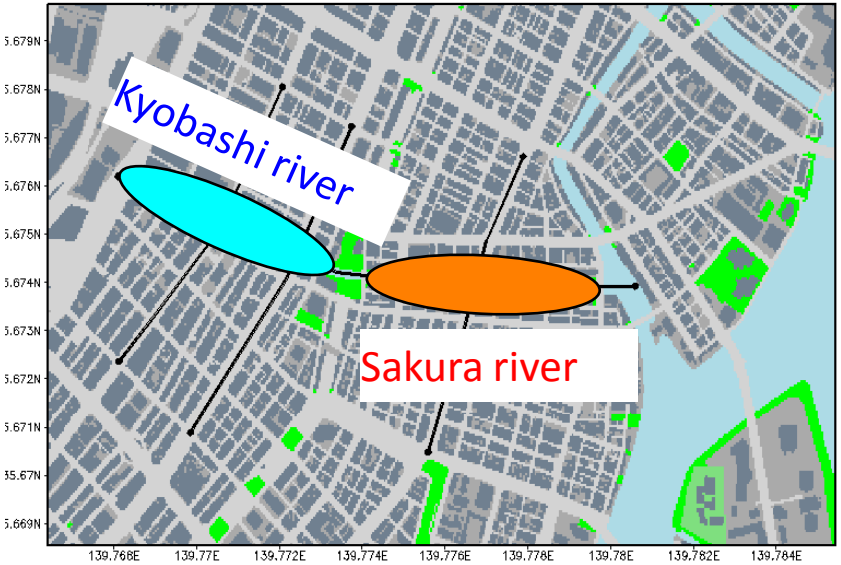


- ✓Normalized by total precip. of each case.
- ✓Reproducibility of strong precip. is not good, and this trend becomes clearer as precip. becomes stronger.
- ✓1M overestimates heavy rain, and 2M suppresses this overestimation.

Case: Kyobashi river

- Evaluate “wind street” when Kyobashi river is recovered.
  - ※ Kyobashi river is reclaimed, however, possibility of recovering the river is considered in recent realistic urban design.
- Wind street effects:
  - ✓ Bringing sea breeze to inland by advection
  - ✓ Cooling by river and grassland
  - ✓ Exchange hot air in vertical direction
- How does the river work ?
  - Is the sea breeze brought into urban ?
  - Really does temperature decreases ?
  - Does the air exchange occur ?
- How does the wind distribution change ?





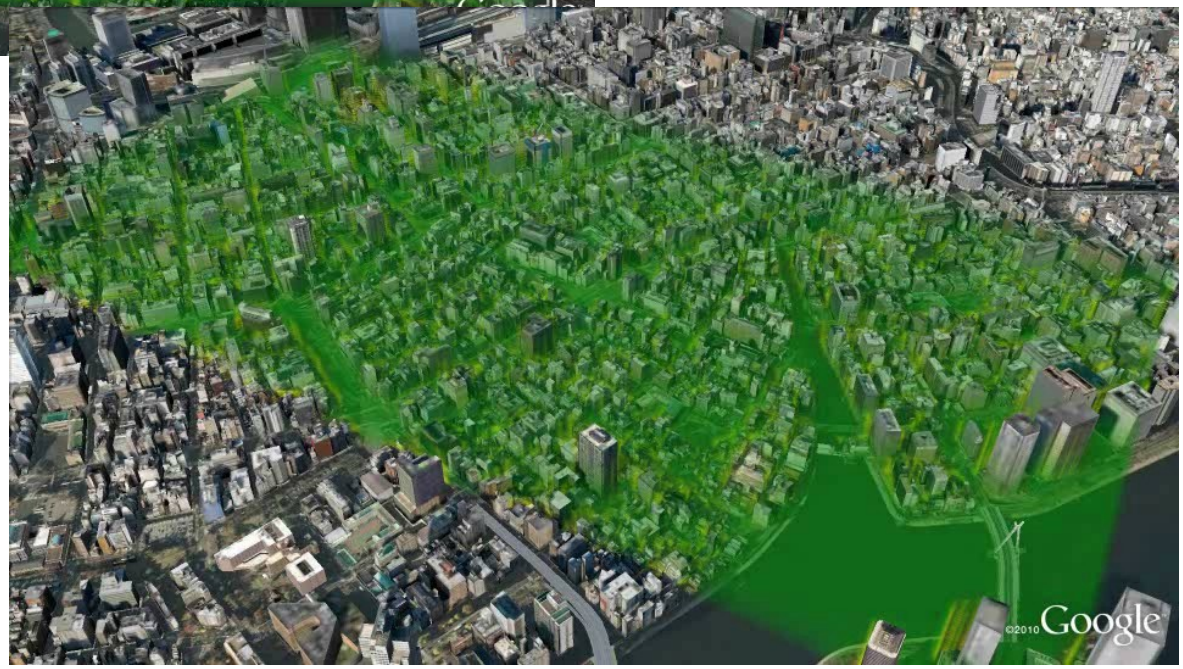
Three-dimensional temperature profile

top view

2005-08-05T15:00

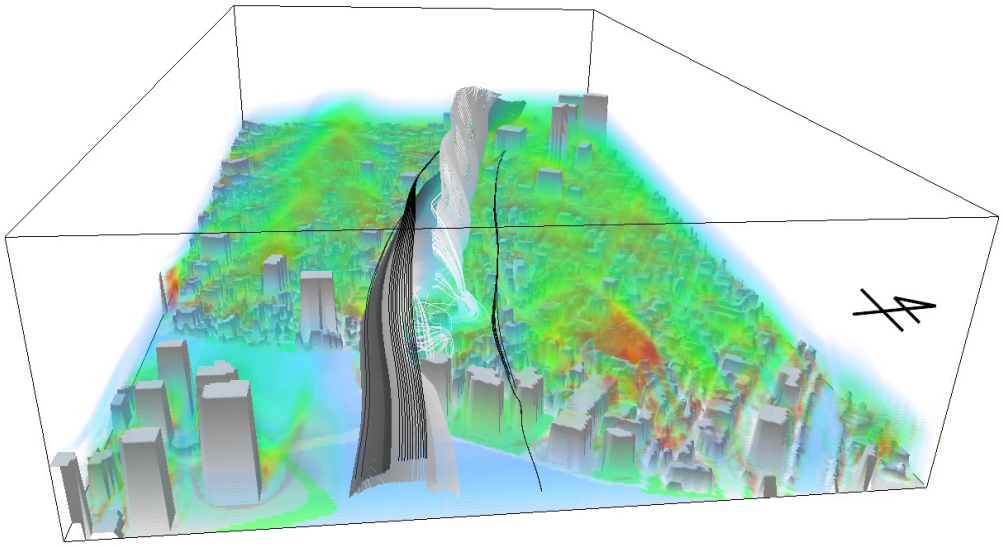
[deg C]
35
34
33
32
31

oblique view

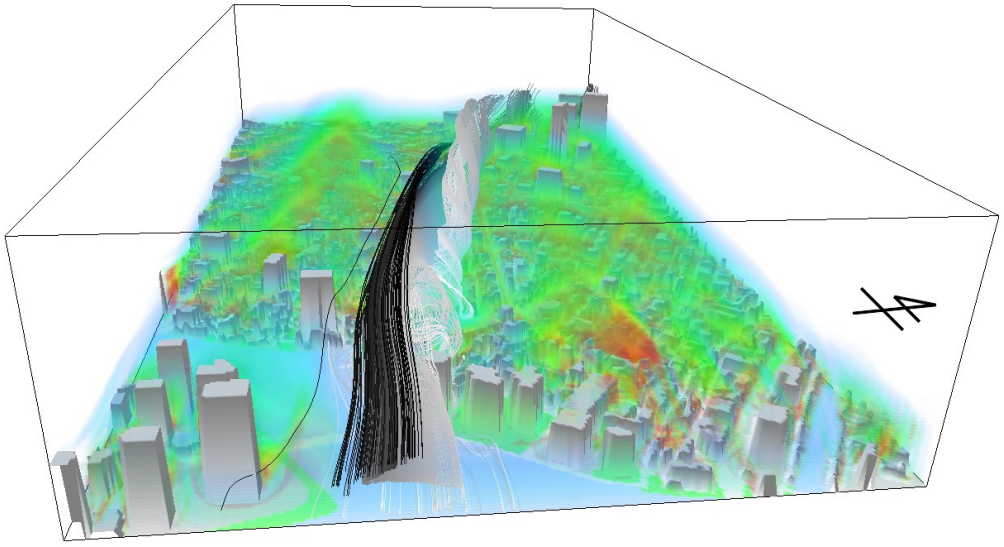


Streamline over river

Case1(No River)



Case2(With River)



- Volume rendering: temperature. Line: streamline.
- Streamlines are illustrated by different colors, from east: white, from west: black.
- In both cases, vertical vortex exists that enhance air exchange between upper and lower atmosphere.
- In case1, number of black streamline is small.

→ Streamline is blocked off between east and west.

- In case2, number of black streamline is large.

→ Air from mount of the river flows toward west side.

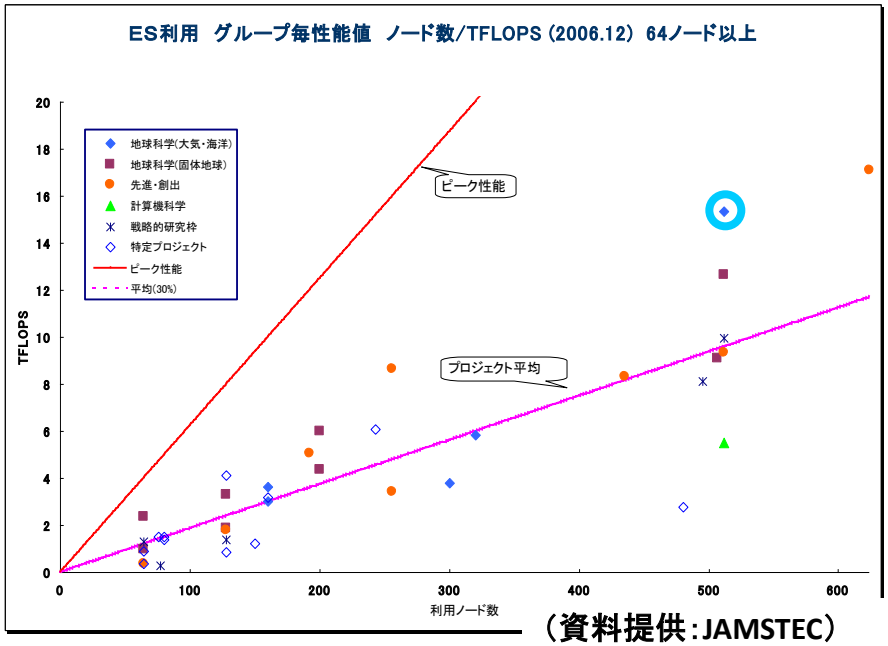
- Vertical vortex location moves northward.

モデル要素

Dynamical core	Fully compressible Euler equations (Baba et al. 2010)
Solvers	HEVI (Baba et al. 2010), HEVE, fractional step (Moureau 2007, Baba and Takahashi 2011)
Horizontal coordinate	Yin-Yang (global) and lat-long (regional) grid (C-grid)
Vertical coordinate	Terrain-following (z*coordinate), z coordinate
Advection scheme	3rd/5th-order Wicker Skamarock (Wicker and Skamarock 2002), 2nd-order WAF (Toro 1989), 3rd-order QUICK (Leonard 1979), 2nd/3rd-order ENO (Shu and Osher 1989), 3rd/5th-order WENO (Baba & Takahashi 2012), CIP-CSL2 (Yabe et al. 2001), PD and MO flux limiter (Skamarock 2006)
Cloud physics	Reisner (Reisner et al. 1998), SBM, Grabowski (1998,1999), 1M/2M (Baba 2012), KF2 (Kain and Fritsch 1992), Emanuel (Emanuel 1991,1999), large-scale condensation (Treut and Li 1991)
Boundary layer scheme & LES	MYNN level-2.5 (Nakanishi and Niino 2004, 2006, 2009), Deardorff (1980), Smagorinsky (1969)
Urban canopy scheme	UCSS (Ashie et al. 2004), Single-layer canopy scheme (Kusaka et al. 2001)
Surface flux & landsurface	Grell et al. (1992), Louis (1979), Fairall et al. (COARE3.0) (2003), bucket model, MATSIRO(under progress)
Radiation	Simple radiation, Cloudrad (MM5), CAM3, MSTRNX (Sekiguchi and Nakajima 2008), 3D radiation scheme (s2srad)

Computational performance on ES1

case	node	CPU	Mflops/CPU	vector length	vector ratio	Tflops	peak performance ratio	acceleration ratio	parallelization ratio
couple	512	4096	4166.7	229	99.30%	17.07	52.10%	461	99.9973
	384	3072	4273.8	229	99.30%	13.13	53.40%	354.6	99.9968
	256	2048	4401.9	229	99.30%	9.02	55.00%	242.6	–
atmos	512	4096	4575.2	228	99.30%	18.74	57.20%	479.1	99.9983
	384	3072	4606.1	228	99.30%	14.15	57.60%	365.2	99.9969
	256	2048	4692.4	228	99.30%	9.61	58.70%	247.5	–
ocean	498	3984	3629.3	240	99.30%	14.46	45.40%	401.3	99.994
	398	3184	3568.5	240	99.30%	11.36	44.60%	333.7	99.989
	207	1656	4234.3	240	99.30%	7.01	52.90%	188.2	–



（資料提供：JAMSTEC）

- Performance of MSSG(couple): 4.2GFLOPS/CPU
- peak performance ratio :52%
- Parallelization: 99.997%

H21年度ES利用者連絡会発表資料(MSSG最適化,NEC)

Earth simulator 2

Earth simulator 1 (since 2002-2009)

- ✓8CPU (8Gflops) & 16GB memory per node
- ✓Total 640 nodes, 5120 CPUs
- ✓Theoretical peak performance: 40TFlops

Earth simulator 2 (since 2009)

- ✓8CPU(102.4Gflops) & 128GB memory per node
- ✓Total 160 nodes, 1280 CPUs
- ✓Theoretical peak performance: 131TFlops

Computational performance on ES2

EXCLUSIVE TIME[sec]	%	MFLOPS	V.OP RATIO	AVER. V.LEN	I-CACHE MISS	O-CACHE MISS	BANK CONFLICT		PROC.NAME	
							CPU PORT	NETWORK		
19777.543	**99.7**	**18592.9**	**99.54**	**236.1**	**256.595**	**772.348**	**262.653**	**6554.572**	**(A1)**	**main loop**
4479.512	22.6	22277.2	99.51	239.2	90.681	198.871	74.955	1697.248	(A2)	N-S HEVI
2632.633	13.3	24765.7	99.50	238.7	26.207	71.759	52.430	821.099	(A2)	N-S eq.(large)
4649.974	23.4	34140.6	99.80	238.7	16.851	60.720	20.966	566.511	(A2)	tracer eq.
3996.377	20.1	7798.0	99.20	213.7	87.334	272.048	57.238	1878.027	(A2)	physics
285.278	1.4	1471.8	99.36	230.4	8.858	15.927	10.138	198.995	(A2)	boundary
596.373	3.0	584.6	99.40	224.2	5.130	8.725	13.396	445.971	(A2)	boundary (side)
235.785	1.2	6361.6	99.11	239.6	0.385	1.991	5.099	93.416	(A2)	z2ps
1073.107	5.4	305.1	81.38	172.9	1.550	3.569	1.977	218.644	(A2)	output
130.365	0.7	23363.6	99.42	238.4	0.399	1.103	22.172	65.026	(A2)	RKG
504.566	2.5	646.3	98.62	239.8	0.295	0.326	0.012	352.960	(A2)	diagno
448.958	2.3	13480.8	99.24	227.6	16.439	33.263	4.128	192.399	(A2)	subfield
734.071	3.7	1083.8	79.70	236.7	0.406	99.532	0.134	23.240	(A2)	recalc dt
0.317	0.0	2.1	90.13	236.1	0.037	0.091	0.001	0.013	(A2)	restart

- Overall (main loop) performance: 18GFLOPS/CPU (18%)
- Dynamical core (N-S HEVI, N-S eq., tracer eq.): 60%
- Physical process (physics): 20%
- Communication (boundary, boundary(side)): 5%
- Others: 15%

H21年度ES利用者連絡会発表資料(MSSG最適化,NEC)

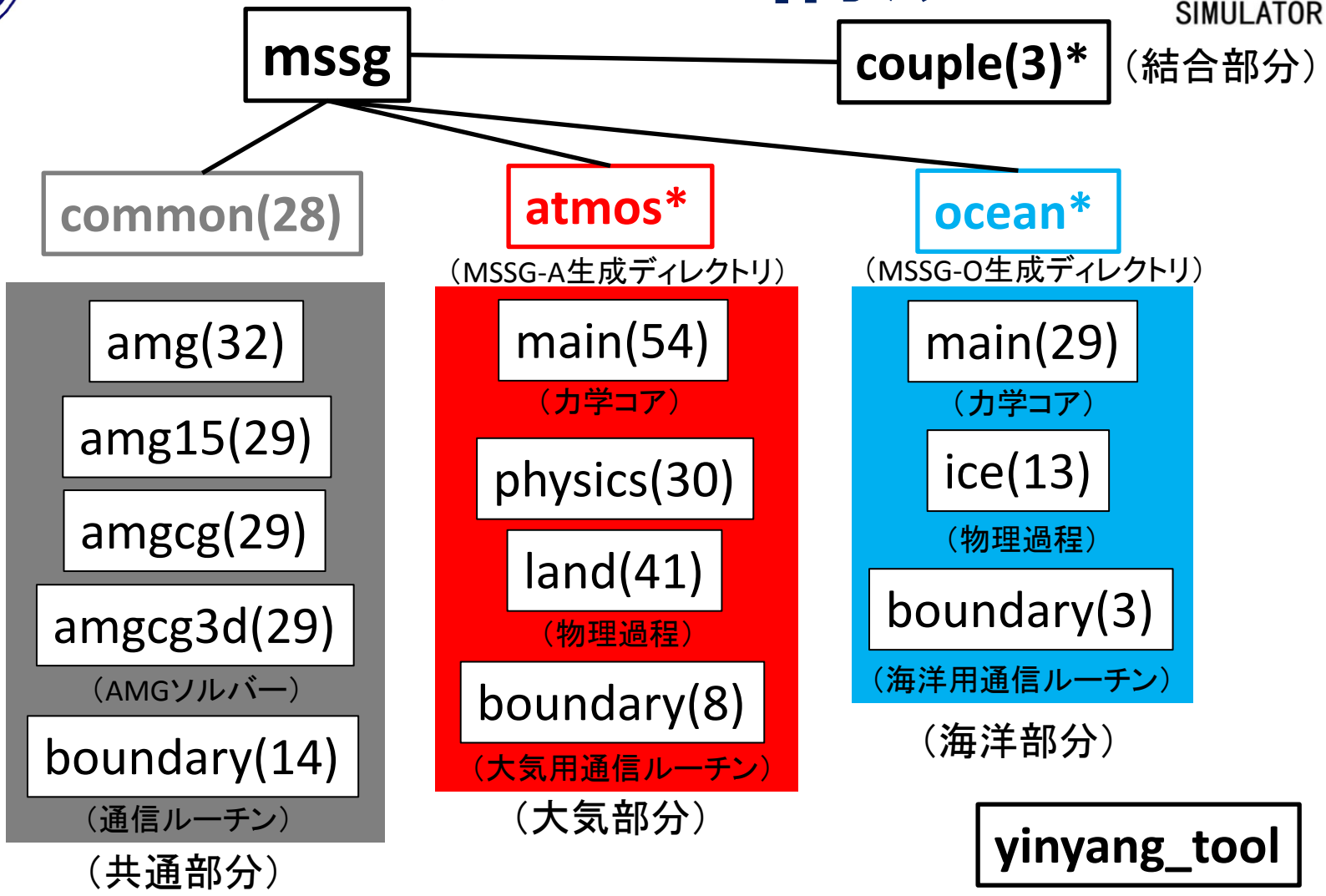
現在の開発体制

- ✓研究員３名（大気３名、海洋１名）、ポスドク２名（海洋）、テクニカル・サポート2名(大気・海洋両方)。
- ✓モデルコンポーネントの導入、デバッグは主に研究員、外注ではテクニカル・サポートメンバーが行なっている。
- ✓マンパワーは常に不足している。

開発上の工夫

- ✓各種テストケース(理想実験)を検証のため用意。
- ✓ソースコード管理にはsubversionを使用。
- ✓ソースブラウザにはredmineとwebsvnを併用。
- ✓バグ報告は主にSVN/redmine上で行なっている。
- ✓モデル説明書2種類をlatexで作成。
- ✓報告書、実験データ管理にsambaサーバを利用。
- ✓モデル説明の補助としてwiki(pukiwiki)ページを用意。

モデルの構成



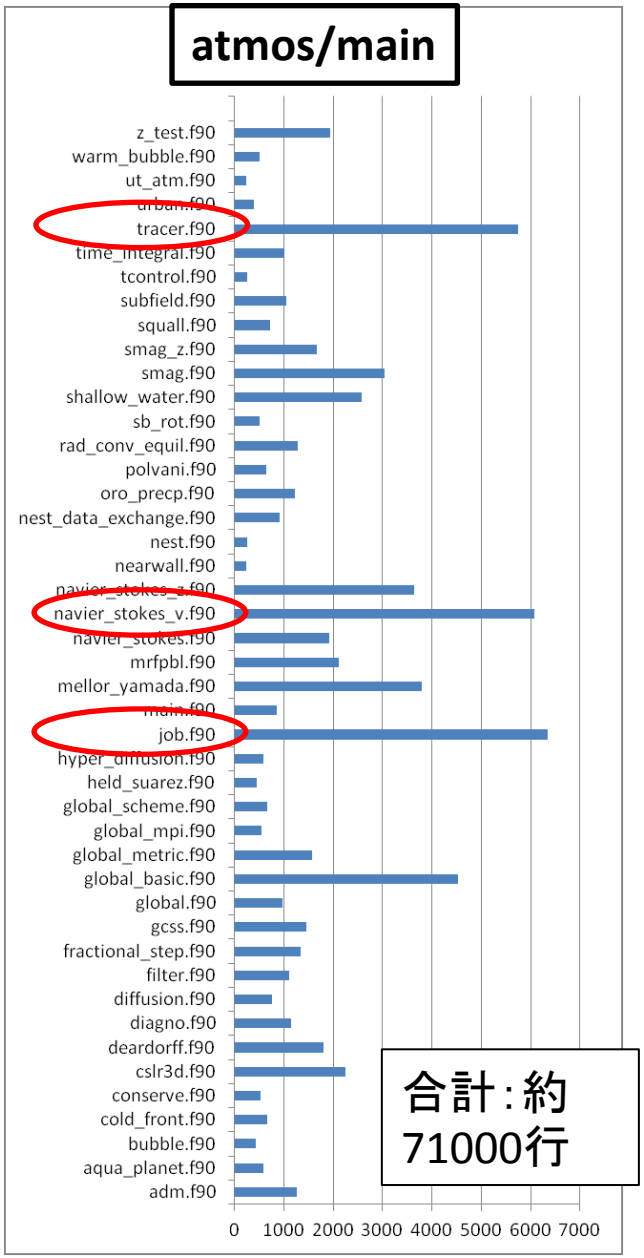
✓カッコはディレクトリに含まれるファイル数。
✓Fortranで分けられない部分はCのifdefで分岐。
✓結合のコンパイルでもプリプロセッサを使用。

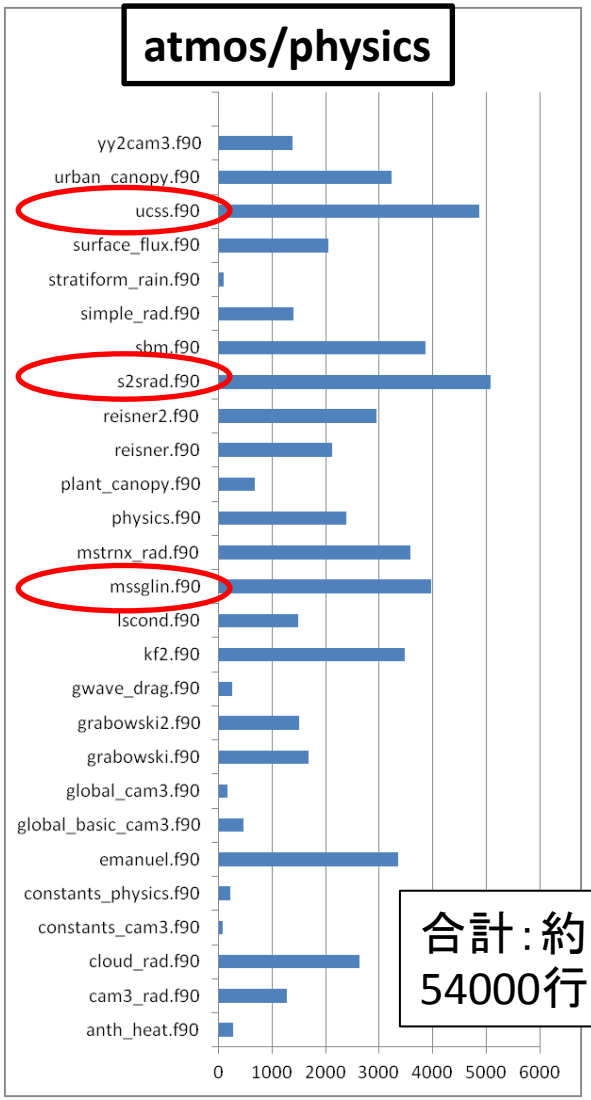
モデルの構成

トレーサ移流・拡散

力学コアの方程式

ジョブ管理



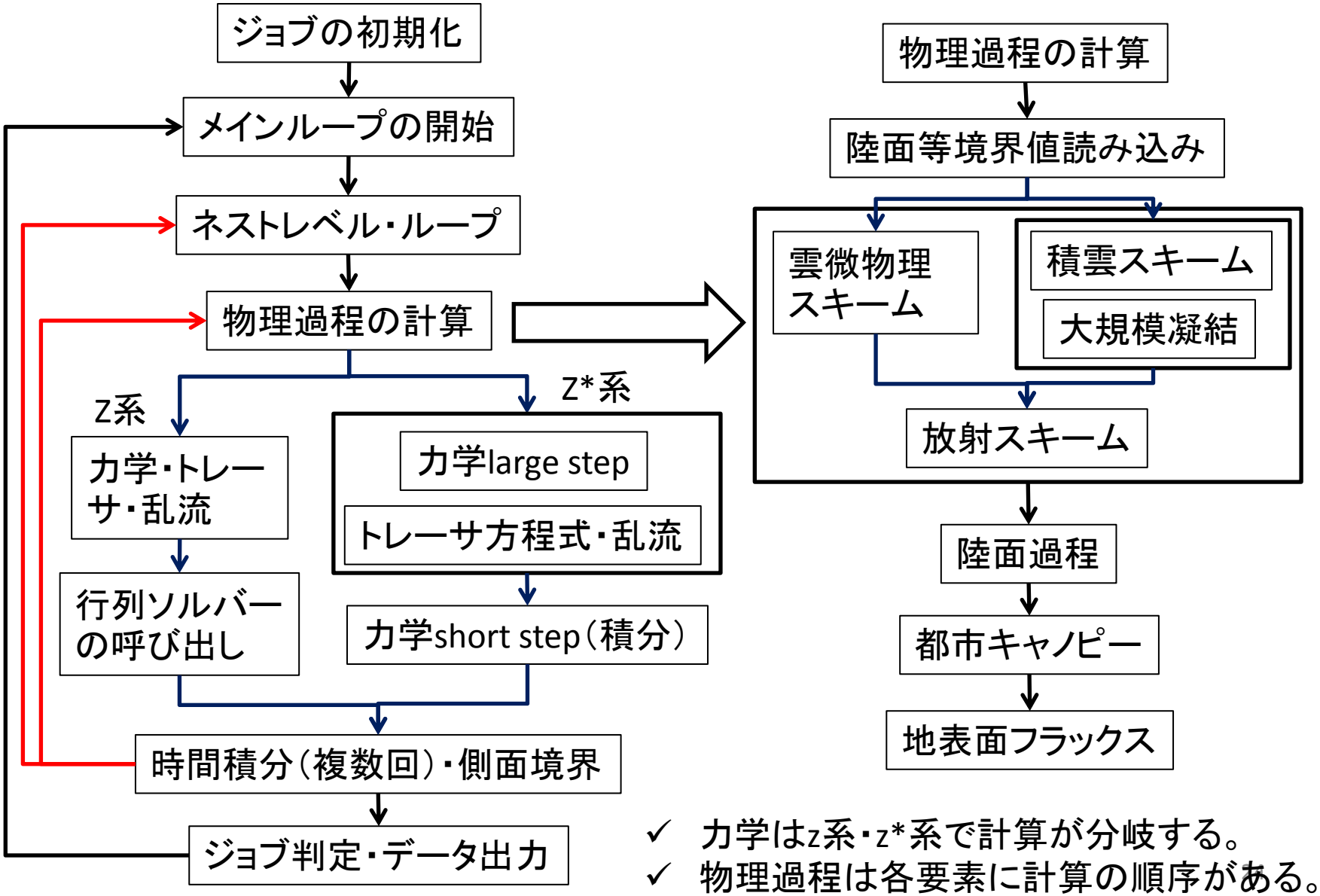


都市キャノピーモデル(2)

3次元放射モデル

1M/2M雲微物理スキーム

計算のフローチャート



- ✓ 力学はz系・z*系で計算が分岐する。
- ✓ 物理過程は各要素に計算の順序がある。

力学過程の計算

```
          divvfp = gMe%sqrtgb1_v(i,j)                                    &
             *(  gMe%r2b1(k)                                             &
                 * ( - fpvrr2_vw(i,j,k  )                                &
                     + fpvrr2_vw(i,j,k+1)  ) * gBa%sw_mask               &
               + gMe%rcpdpb1_v(j,k)                                      &
                 *(fpvpcp_t(i,j  ,k)                                     &
                  -fpvpcp_t(i,j-1,k))                                    &
               + gMe%rcpdlb1_v(j,k)                                      &
                 *(fpvl_uv (i+1,j,k)                                     &
                  -fpvl_uv (i  ,j,k)) )                                  &
               + metric_v

          divvfl = gMe%sqrtgb1_u(i,j)                                    &
             *(  gMe%r2b1(k)                                             &
                 * ( - flvrr2_uw(i,j,k  )                                &
                     + flvrr2_uw(i,j,k+1)  ) * gBa%sw_mask               &
               + gMe%rcpdpb1(j,k)                                        &
                 *(flvpcp_uv(i,j+1,k)                                    &
                  -flvpcp_uv(i,j  ,k))                                   &
               + gMe%rcpdlb1(j,k)                                        &
                 *(flvl_t   (i  ,j,k)                                    &
                  -flvl_t   (i-1,j,k)) )                                 &
               + metric_u
        else
          divvfp = gMe%sqrtgb1_v(i,j)                                    &
             *(  gMe%r2b1(k)                                             &
!sugi-v s
                 * ( fpvrr2_vw(i,j,k  ) * cdiff_z_vec(k,1)               &
                   + fpvrr2_vw(i,j,k+1) * cdiff_z_vec(k,2) ) * gBa%sw_mask &
!sugi-v e
               + gMe%rcpdpb1_v(j,k)                                      &
                 *(fpvpcp_t(i,j  ,k)                                     &
                  -fpvpcp_t(i,j-1,k))                                    &
               + gMe%rcpdlb1_v(j,k)                                      &
                 *(fpvl_uv (i+1,j,k)                                     &
                  -fpvl_uv (i  ,j,k)) )                                  &
               + metric_v

          divvfl = gMe%sqrtgb1_u(i,j)                                    &
             *(  gMe%r2b1(k)                                             &
!sugi-v s
                 * ( flvrr2_uw(i,j,k  ) * cdiff_z_vec(k,1)               &
                   + flvrr2_uw(i,j,k+1) * cdiff_z_vec(k,2) ) * gBa%sw_mask &
!sugi-v e
               + gMe%rcpdpb1(j,k)                                        &
                 *(flvpcp_uv(i,j+1,k)                                    &
                  -flvpcp_uv(i,j  ,k))                                   &
               + gMe%rcpdlb1(j,k)                                        &
                 *(flvl_t   (i  ,j,k)                                    &
                  -flvl_t   (i-1,j,k)) )                                 &
               + metric_u
        end if

        if ( gBa%advection_fr == ADV_CIP ) then
          divvfr = + metric_w
        elseif( gBa%advection_fr == ADV_ENO2ND .or. &
                gBa%advection_fr == ADV_ENO3RD ) then
          divvfr = gMe%sqrtgb1(i,j) * gBa%sw_mask                        &
             *(  gMe%r2b1_w(k)                                           &
```

navier_stokes_v.f90/the_equation_l

ほとんどのルーチンが3次元配列を引数として引き渡している。

```
  rdro  = 0.0_DP
  rdps  = 0.0_DP
  rdfl  = 0.0_DP
  rdfp  = 0.0_DP
  rdfr  = 0.0_DP

  call work_fields( nl, np, nr,                                          &
                    fl, fp, fr,                                          &
                    vl, vp, vzs,                                         &
                    cintp_z_vec,   cintp_z_vec_w,                        &
                    flvl_t,    fpvl_uv,   frvl_uw,                       &
                    fpvrr2_vw, flvrr2_uw, frvr_t,                        &
                    frvpcp_vw, fpvpcp_t,  flvpcp_uv )

  if ( gBa%test_case == SW_TEST ) then
    do k = gDkmin, gDkmax
      do j = gDjmin, gDjmax
        do i = gDimin, gDimax
          if ( abs( vl(i,j,k) ) < EPS ) vl(i,j,k) = 0.0_DP
        end do
      end do
    end do
  end if
```

```
    ! HEVI法のメイン。鉛直運動量について3重対角行列を解く。
    ! 中で密度、圧力の水平移流も計算している。
    call the_sound_fr  ( gM_nl  ,gM_np  ,gM_nr  ,                        &
                         fl_tau ,fp_tau ,fr_tau ,gN%fr  ,gN%ro  ,gN%ps  , &
                         gN%vl  ,gN%vp  ,gN%vzs ,gN%fzs ,                &
                         gN%roa ,gN%psa ,soa_w  ,gN%dfr ,gN%dro , dros, &
                         gN%dps ,dpss   ,gN%divv,                        &
                         gtilde_w, roa_tau,                              &
                         gSc%cintp_z_scl_w,                              &
                         gSc%cintp_z_vec_w,                              &
                         gSc%cdiff_z_scl_w,                              &
                         gMe%cosphi )
                                                call ft_end( 3,"(A3) N-S HEVI/sound (fr)" )
                                                call ft_bgn( 3,"(A3) N-S HEVI/sound (rops)" )
    ! 得られたWで密度と圧力を補正する。
    call the_sound_rops( gM_nl  ,gM_np  ,gM_nr  ,                        &
                         gN%fr  ,gN%ro  ,gN%ps  ,                        &
                         soa_w  ,dros   ,dpss, gtilde_w, roa_tau )
                                                call ft_end( 3,"(A3) N-S HEVI/sound (rops)" )
```

i,j,k全てのインデックスにループ依存性がある。
→3次元ループで回す。

物理過程の計算

```
subroutine equation_core_wrapper(prcpx,dQ_rei,dflp,dfpp,dfrp,optc,opti)
   real(DP), dimension(gDimin:gDimax,gDjmin:gDjmax,gM_ntrc),intent(inout) :: prcpx
   real(DP), dimension(gDimin:gDimax,gDjmin:gDjmax,gDkmin:gDkmax),intent(inout) :: dQ_rei
   real(DP), dimension(gDimin:gDimax,gDjmin:gDjmax,gDkmin:gDkmax),intent(inout) :: dflp,dfpp,dfrp
   real(DP), dimension(gDimin:gDimax,gDjmin:gDjmax),intent(inout) :: optc,opti
   integer :: j

#ifdef use_mssglin_

!cdir pardo by = 1, nobarr = (entry,exit)
!cdir nodep
   do j = 1, gM_np
     call equation_core(0, gM_nl+1, 0, gM_np+1,          &
                        gDimin, gDimax, gDjmin, gDjmax,  &
                        gAimin, gAimax, gAjmin, gAjmax,  &
                        gM_nr, gM_ntrc, j, 1, gM_nl,     &
                        gN%roa, gN%psa, gN%fl, gN%fp, gN%fr, gN%rq, &
                        gN%drop, gN%dpsp, dflp, dfpp, dfrp, gN%drqp, prcpx, &
                        gMe%sqrtgb1, 1.0_DP, &
                        dQ_rei,md%ze,optc,opti)
   end do

#endif

end subroutine equation_core_wrapper
```

Type A: mssglin.f90

3次元配列を2次元配列として処理し、緯度方向はマイクロタスクに割り付ける。最内ループは経度方向で、ベクトル化する。鉛直計算が入る積雲スキーム、雲微物理（落下）、放射スキームに有効。

```
!kg!cdir pardo for
!kg    microtask : do n = 1, ntask
!kg      ijs = ij_sta(n)
!kg      ije = ij_end(n)
!cdir pardo by=1
    microtask : do j = 1, gM_np
      call psurfc ( &
                runoff     ,                                    & ! OUT
                GG         , gN%wa_g    ,                       & ! INOUT
                TFLX_SURF  , QFLX_SURF  ,                       & ! INOUT
                invI       , d_sens     , d_evap     ,          & ! IN
                rflxsu_surf, rflxsd_surf, rflxld_surf,          & ! IN
                rflxlu_surf,                                    & ! INOUT
                           GPH         ,                        & ! IN
                GCPRC      , GLPRC      , GCSNW      , GLSNW    , & ! IN
        md%idsrf_convert, md%jdsrf_convert, gN%dt      , gN%dt    , & ! IN
!kg                ijs, ije   , md%beta_pre                      ) ! IN
                1, gM_nl  ,j , md%beta_pre                     ) ! IN

   enddo  microtask
  end subroutine bucket__main_microtask
```

Type B: land.f90

2次元配列を1次元配列にし、経度方向をマイクロタスクに割り付ける。ただし、配列は2Dとして渡す（配列外参照チェックのため）。陸面過程に有効。

モデル要素の最適化

最内ループの2段アンロール

変更前のコード

```
  do j = 1, np+1
    do k = 2, nr-1
      do i = 1, nl+1
        roa_ij  = roapd(i,  j,k)
        roa_ip1 = roapd(i+1,j,k)
        roa_ip2 = roapd(i+2,j,k)
        roa_im1 = roapd(i-1,j,k)
        roa_im2 = roapd(i-2,j,k)
        roa_im3 = roapd(i-3,j,k)
        rovl_u(i,j,k) = P060B1 * (                          &
               vl(i,j,k) *( 37.0_DP*(roa_ij +roa_im1)   &
                          - 8.0_DP*(roa_ip1+roa_im2)   &
                          +        (roa_ip2+roa_im3) ) &
         - abs(vl(i,j,k))*( 10.0_DP*(roa_ij -roa_im1)   &
                          - 5.0_DP*(roa_ip1-roa_im2)   &
                          +        (roa_ip2-roa_im3) ) )
        roa_ij  = roapd(i,  j,k)
        roa_jp1 = roapd(i,j+1,k)
        roa_jp2 = roapd(i,j+2,k)
        roa_jm1 = roapd(i,j-1,k)
        roa_jm2 = roapd(i,j-2,k)
        roa_jm3 = roapd(i,j-3,k)
        rovp_v(i,j,k) = P060B1 * (                                  &
               vp(i,j,k) *( 37.0_DP*(roa_ij *cosphi(j  )+roa_jm1*cosphi(j-1))   &
                          - 8.0_DP*(roa_jp1*cosphi(j+1)+roa_jm2*cosphi(j-2))   &
                          +        (roa_jp2*cosphi(j+2)+roa_jm3*cosphi(j-3)) ) &
         - abs(vp(i,j,k))*( 10.0_DP*(roa_ij *cosphi(j  )-roa_jm1*cosphi(j-1))   &
                          - 5.0_DP*(roa_jp1*cosphi(j+1)-roa_jm2*cosphi(j-2))   &
                          +        (roa_jp2*cosphi(j+2)-roa_jm3*cosphi(j-3)) ) )
      end do
    end do
  end do
```

力学過程の最適化に有効。

- ｉの差分とｊの差分が1つのループに混在
- roapd(i,j,k) のみロードが共通

モデル要素の最適化

最内ループの2段アンロール

変更後のイメージ

```
      do k = 2, nr-1
        do j = 1, np+1
!cdir unroll=2
          do i = 1, nl+1
            roa_ij  = roapd(i,  j,k)
            roa_ip1 = roapd(i+1,j,k)
            roa_ip2 = roapd(i+2,j,k)
            roa_im1 = roapd(i-1,j,k)
            roa_im2 = roapd(i-2,j,k)
            roa_im3 = roapd(i-3,j,k)
            rovl_u(i,j,k) = ...
          end do
        end do

!cdir outerunroll=8
        do j = 1, np+1
          do i = 1, nl+1
            roa_ij  = roapd(i,  j,k)
            roa_jp1 = roapd(i,j+1,k)
            roa_jp2 = roapd(i,j+2,k)
            roa_jm1 = roapd(i,j-1,k)
            roa_jm2 = roapd(i,j-2,k)
            roa_jm3 = roapd(i,j-3,k)
            rovp_v(i,j,k) = ...
          end do
        end do
      end do
```

力学過程の最適化に有効。

- ループを分割
- i の差分ループは、unroll=2 を i のループに対し適用
  →ベクトルロードを削減
  （バンクアクセス単位の16Byte化により、2段アンロールまでは、メモリのスループットが落ちない）
- j の差分ループは、outerunroll=8 を j のループに対し適用
  → ベクトルロードを削減

モデル要素の最適化

計算式の変更

変更前のコード

物理過程の最適化に有効。

```
0694        do k = 2, nr-1
0695          do i = ijs, ije
0701            lambdab_r(i,k)=(rqr(i,k)*L_PIB*Norrorb(i,k))**0.25_DP
0709            lambdap2b_r(i,k) = lambdab_r(i,k)*lambdab_r(i,k)
0710            lambdap3b_r(i,k) = lambdap2b_r(i,k)*lambdab_r(i,k)
0711            lambdap4b_r(i,k) = lambdap3b_r(i,k)*lambdab_r(i,k)
0712            lambdap5b_r(i,k) = lambdap4b_r(i,k)*lambdab_r(i,k)
0713            lambdap6b_r(i,k) = lambdap5b_r(i,k)*lambdab_r(i,k)
0716            Praci(i,k)=c025PIEirNor(i,k) *qi(i,k)                    &
0717                  *(-c0267gm3     *lambdap3b_r(i,k)                  &
0718                    +c515gm4      *lambdap4b_r(i,k)                  &
0719                   -c1022500gm5   *lambdap5b_r(i,k)                  &
0720                   +c75500000gm6*lambdap6b_r(i,k))
0766          enddo
0767        enddo
0771        do k = 2, nr-1
0772          do i = ijs, ije
0861 ! Reisner(1998) (A.47)
0865            Psacr(i,k)=PIp2ErsrowNor(i,k) *aUrmUscUrUsp05Nossod      &
0866                * ( 5.0_DP * lambdap6b_r(i,k)*lambdab_s(i,k)         &
0867                   +2.0_DP * lambdap5b_r(i,k)*lambdap2b_s(i,k)       &
0868                   +0.5_DP * lambdap4b_r(i,k)*lambdap3b_s(i,k))
0869 ! Reisner(1998) (A.48)
0870            Pracs(i,k)=PIp2ErsrosNor(i,k) *aUrmUscUrUsp05Nossod      &
0871            * ( 5.0_DP * lambdap6b_s(i,k)*lambdab_r(i,k)             &
0872               +2.0_DP * (lambdap3b_s(i,k)*lambdap2b_s(i,k))*lambdap2b_r(i,k) &
0873               +0.5_DP * (lambdap3b_s(i,k)*lambdab_s(i,k))*lambdap3b_r(i,k))
0884          enddo
0885        enddo
```

lambdab_r,
lambda2b_r,
lambda3b_r等を
メモリへストア

lambdab_r,
lambda2b_r,
lambda3b_r等を
メモリからロード

H21年度ES利用者連絡会発表資料(MSSG最適化,NEC)

モデル要素の最適化

計算式の変更

物理過程の最適化に有効。

変更後のコード

```
0694        do k = 2, nr-1
0695          do i = ijs, ije
0750            lambdab_r(i,k)=(rqr(i,k)*L_PIB*Norrorb(i,k))**0.25_DP
0769            lambdap2b_r = lambdab_r(i,k)*lambdab_r(i,k)
0770            lambdap3b_r = lambdap2b_r*lambdab_r(i,k)
0785            Praci(i,k)=c025PIEirNor(i,k) *qi(i,k)                  &
0786               *lambdap3b_r*(-c0267gm3                             &
0787                             +c515gm4      *lambdab_r(i,k)        &
0788                             -c1022500gm5 *lambdap2b_r             &
0789                             +c75500000gm6*lambdap3b_r)
0858          enddo
0859        enddo
0866        do k = 2, nr-1
0867          do i = ijs, ije
0980            lambdap2b_r=lambdab_r(i,k)*lambdab_r(i,k)
0981            lambdap4b_r=lambdap2b_r*lambdap2b_r
0988 ! Reisner(1998) (A.47)
0989            Psacr(i,k)=L_PIp2ErsROW*Nor(i,k)*aUrmUscUrUsp05Nossod &
0990                * lambdap4b_r*lambdab_s(i,k)                       &
0991                * ( 5.0_DP * lambdap2b_r                           &
0992                   +2.0_DP * lambdab_r(i,k)*lambdab_s(i,k)         &
0993                   +0.5_DP * lambdap2b_s )
1003 ! Reisner(1998) (A.48)
1005            Pracs(i,k)=PIp2ErsrosNor(i,k) *aUrmUscUrUsp05Nossod   &
1009                * lambdap4b_s*lambdab_r(i,k)                       &
1010                * ( 5.0_DP * lambdap2b_s                           &
1011                   +2.0_DP * lambdab_s(i,k)*lambdab_r(i,k)         &
1012                   +0.5_DP * lambdap2b_r )
```

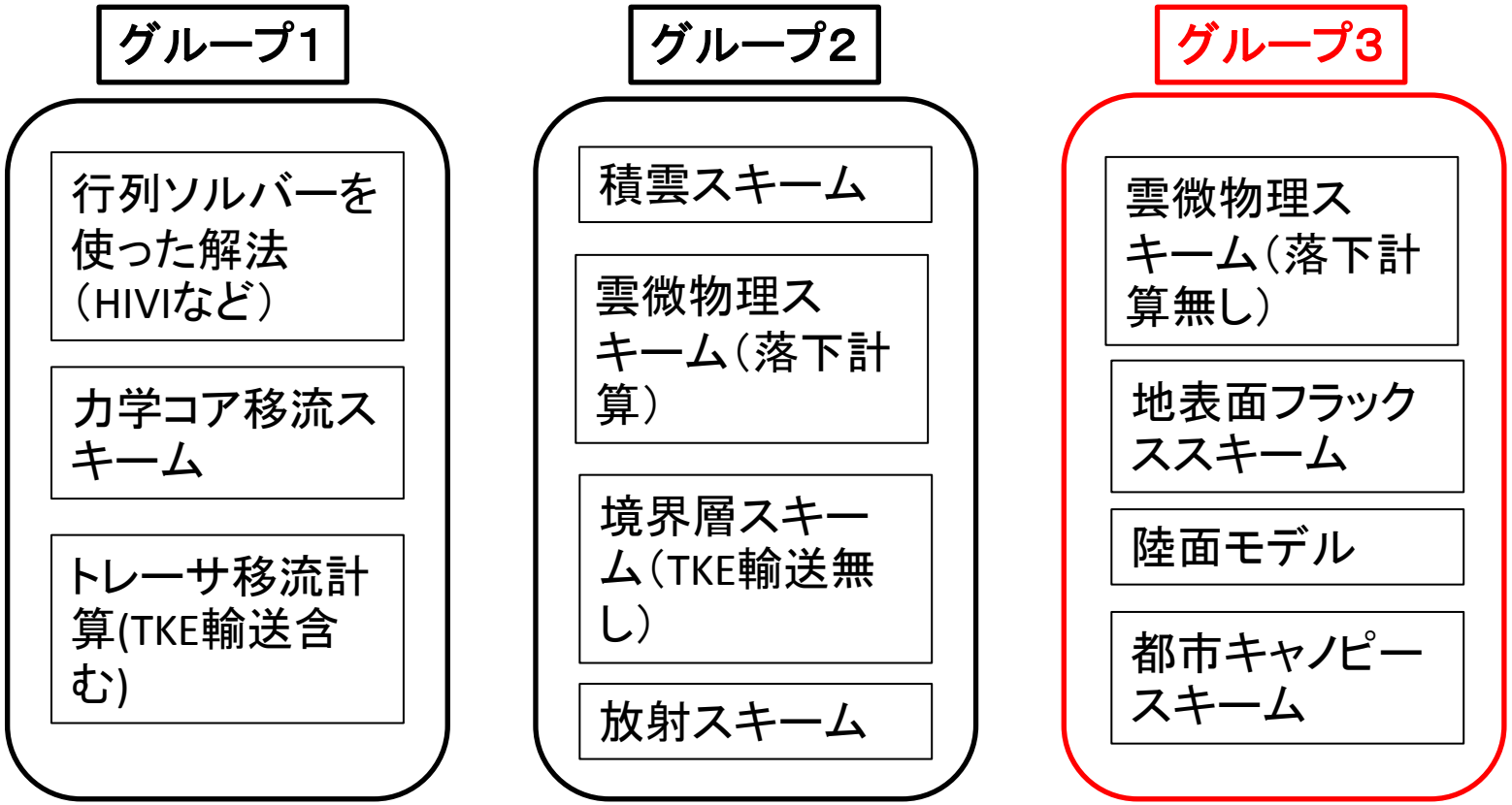
lambdab_rのみストアし、lambda2b_r, lambda3b_r等はメモリへのストアなし

lambda2b_r, lambda4b_r等は、lambdab_rから再計算

→演算は増ふえるが、メモリアクセス回数は減らせる

モデル要素の構造上の分類

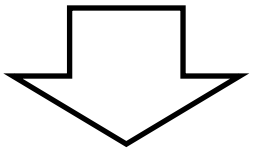
1. 力学コアから分離することが難しい。
2. 力学コアから分離できるが鉛直座標の制限を受ける。
3. 力学コアから分離できて鉛直座標の制限が無い。



✓物理過程は別グループでも計算順序の依存性がある。

開発上の問題点

- ✓**知見の分散**：モデルに関する情報の共有と知見の蓄積が不十分。→redmine，pukiwikiである程度解決できる。しかし、まとめるツールの便利さは人によって違うので、フォーマットが異なる文書が分散しがち。
- ✓**モデル確立までのタイムラグ**：先進的なモデル要素を実装しても、論文にするレベルにまでに精緻化するには膨大な検証実験が必要。実装から論文にするまでのタイムラグが大きい。
- ✓**コーディングの違い**：研究とチューニングのコーディングが異なる。研究では可読性と、式に忠実であることを重視。チューニングでは高速化と演算量の低下を重視。



これらの問題点は恐らく
共通ライブラリ構築で解決できる。

ライブラリ化の問題点

共通ライブラリが便利なのは明らか。それなのに実現されていないのには理由がある。

- ✓ **モデル構造の問題**：モデル要素には力学コアと密接に関係しているものもあり、容易に分離することができない。例：トレーサスキーム、乱流スキームなど。
- ✓ **計算順序の問題**：計算順序に依存するパラメタリゼーションが多数存在するため個別に分離できない。例：積雲スキームと大規模凝結スキーム。
- ✓ **計算機の問題**：すべてのアーキテクチャで高速に動作する物理過程モジュールを作ることは不可能。例：ベクトルとスカラーで高速に動作するモジュールの構造は大きく異なる。
- ✓ **モチベーションの問題**：成果を出すために使える計算機と、ライブラリ開発のための計算機が異なるとモチベーションが低下する（もともとモデル開発は成果を目指して行なっているため）。→評価体制とも関係する。

問題解決に向けた方策

- ✓**ライブラリ化が可能なものを限定して共通化する**：物理過程で、グループ3はライブラリ化には適切な構造をしており、ライブラリ化はまずこれらに限定する。
  →モデル構造と計算順序の問題を解決。

- ✓**スカラー、ベクトルの区別をしないコーディングを徹底する**：ライブラリがスカラーのみのコーディングではベクトル機がメインの研究者にはメリットが少ない。例えば鉛直方向を最内ループに持ってくるコーディングは禁止したキャッシュ利用方法を考える。
  →計算機の問題とモチベーションの問題を解決。