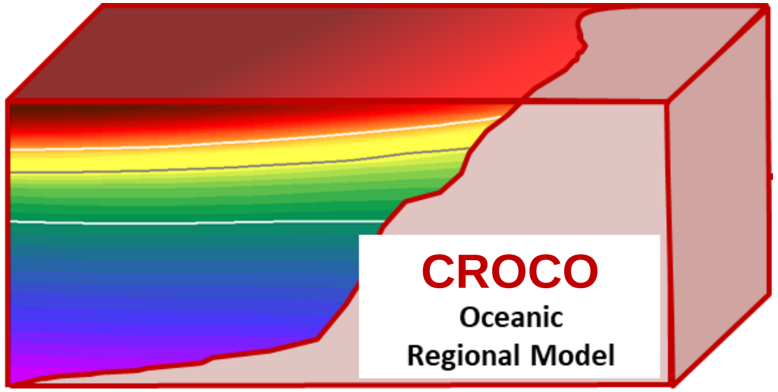
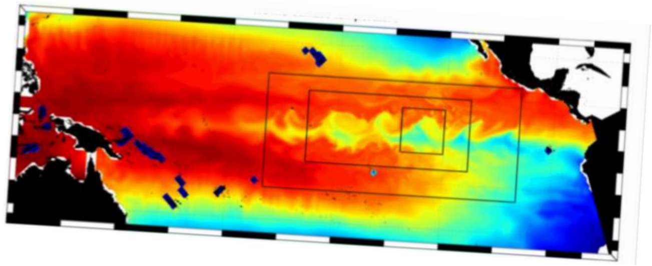
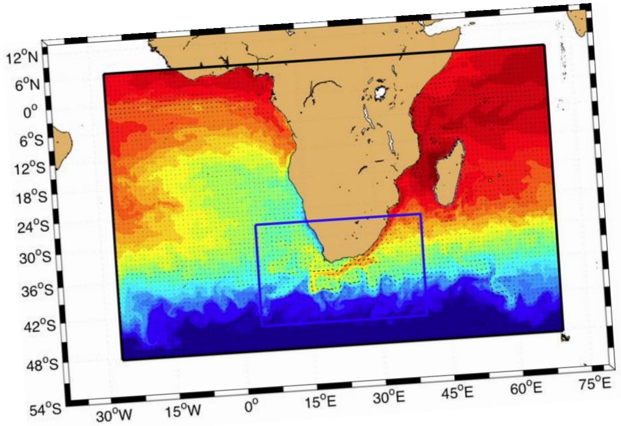


Introduction To Ocean modeling Using CROCO And The CROCO_TOOLS



http://sillig.free.fr



Forcings

Forcings: Generalities

Momentum conservation

$$\frac{\partial u}{\partial t} + \vec{u}.\nabla u - fv = -\frac{1}{\rho_0}\frac{\partial P}{\partial x} + \nabla_h\left(K_{Mh}.\nabla_h u\right) + \frac{\partial}{\partial z}\left(K_{Mv}\frac{\partial u}{\partial z}\right)$$

$$\frac{\partial v}{\partial t} + \vec{u}.\nabla v + fu = -\frac{1}{\rho_0}\frac{\partial P}{\partial y} + \nabla_h\left(K_{Mh}.\nabla_h v\right) + \frac{\partial}{\partial z}\left(K_{Mv}\frac{\partial v}{\partial z}\right)$$

advection *Coriolis* *Pressure gradient* *Horizontal diffusion* *Vertical diffusion*

Hydrostatic

$$0 = -\frac{\partial P}{\partial z} - \rho g$$

Continuity

$$0 = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}$$

Tracer conservation

$$\frac{\partial T}{\partial t} + \vec{u}.\nabla T = \nabla_h\left(K_{Th}.\nabla_h T\right) + \frac{\partial}{\partial z}\left(K_{Tv}\frac{\partial T}{\partial z}\right)$$

$$\frac{\partial S}{\partial t} + \vec{u}.\nabla S = \nabla_h\left(K_{Sh}.\nabla_h S\right) + \frac{\partial}{\partial z}\left(K_{Sv}\frac{\partial S}{\partial z}\right)$$

Equation of state

$$\rho = \rho(S, T, p)$$

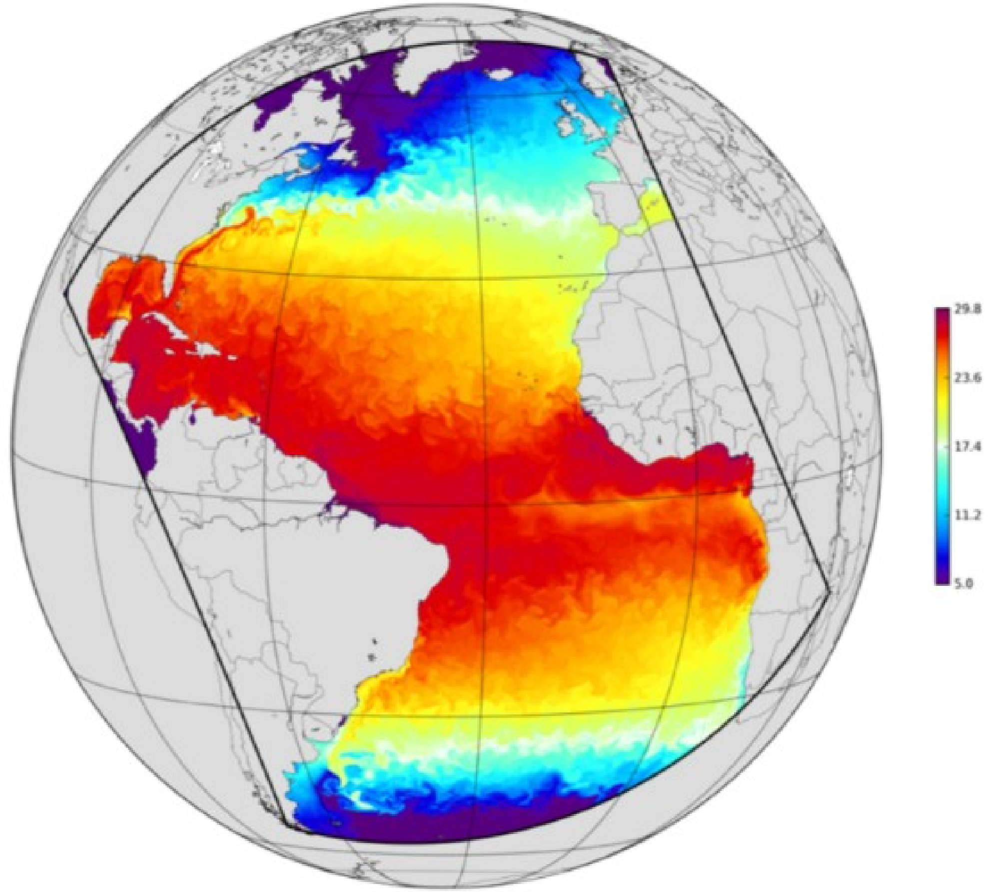
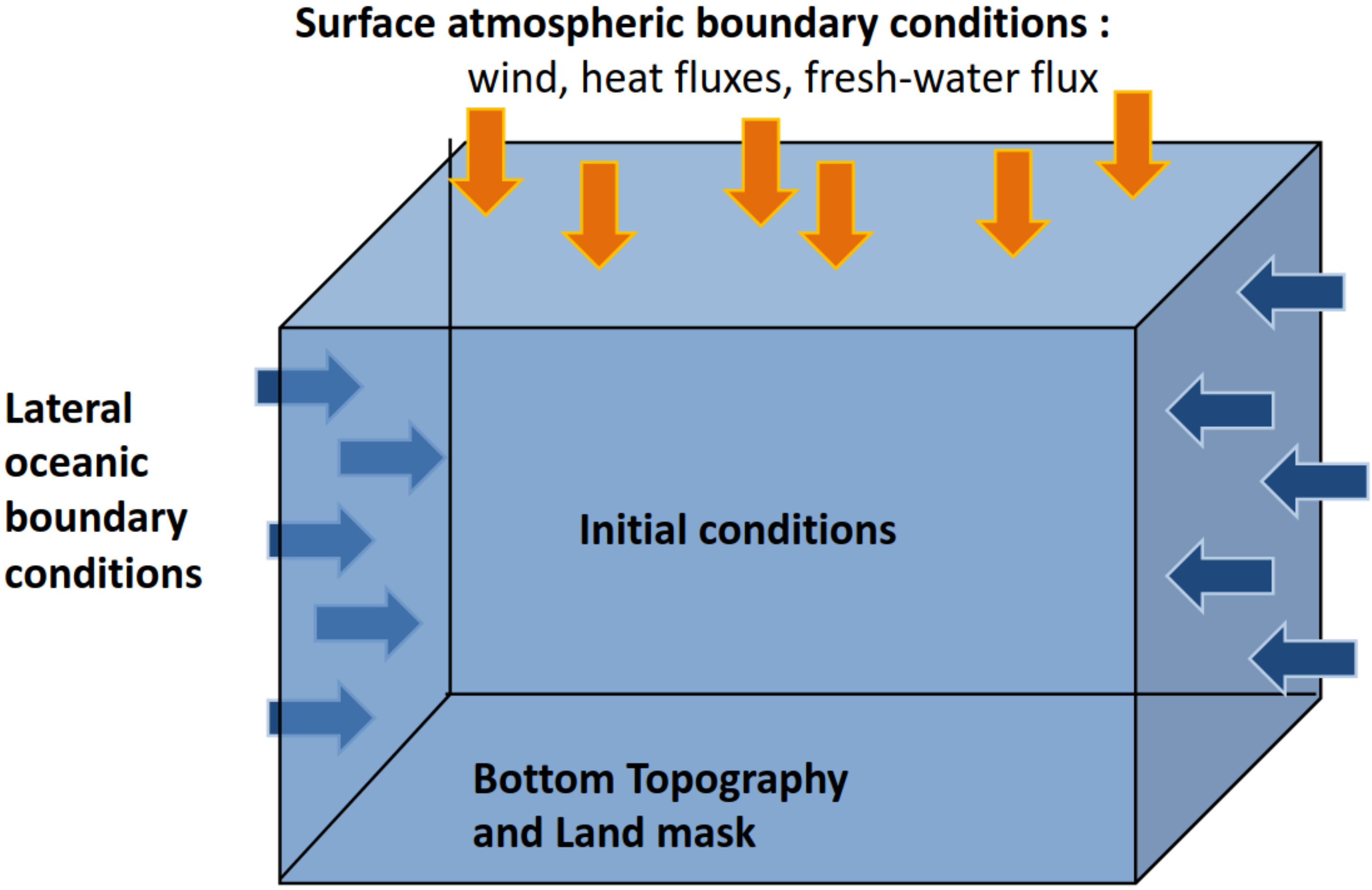


Figure 4: Snapshot of SST for the Atlantic simulation domain from a 6 km CROCO simulation.

Courtesy of J. Gula

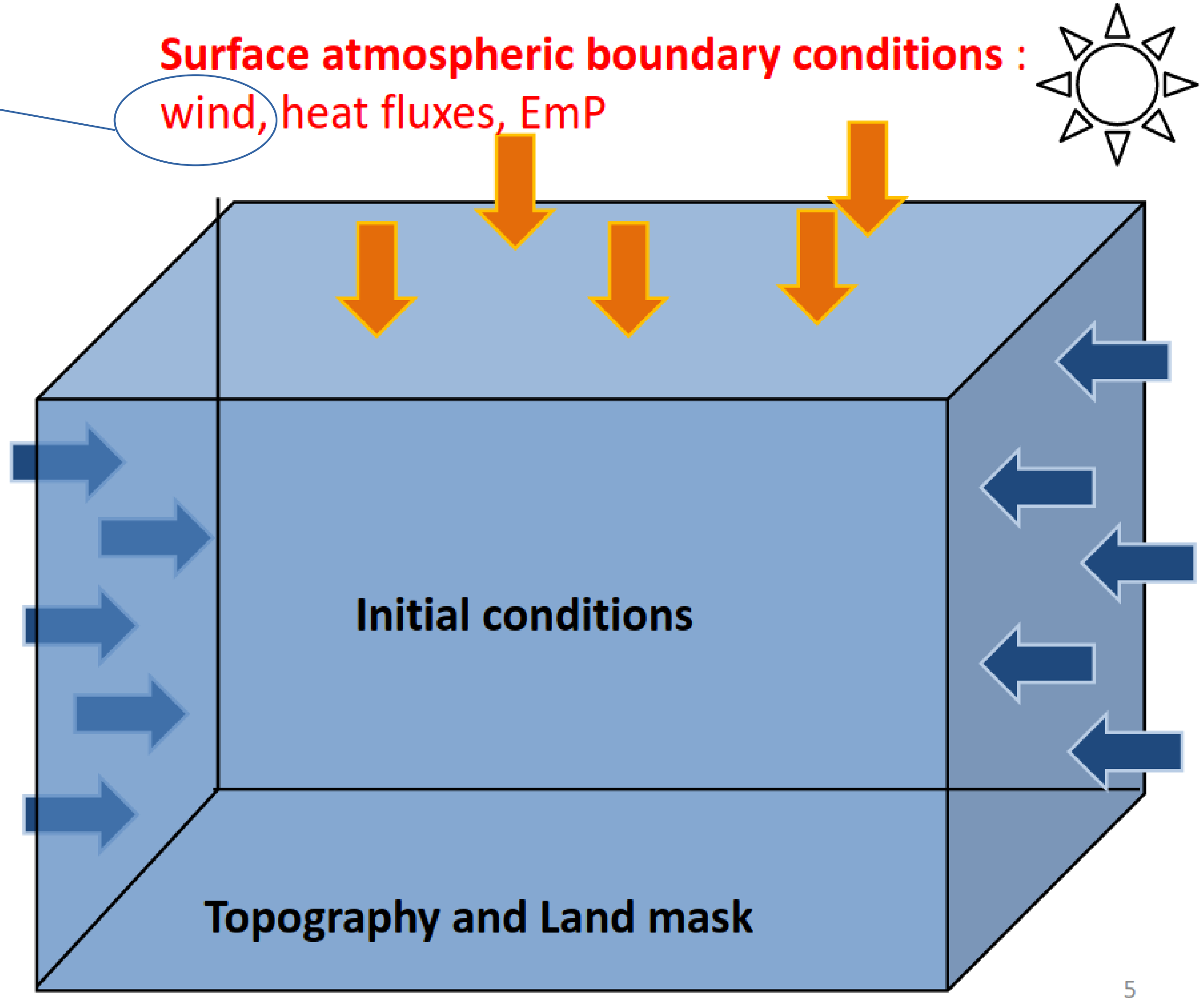
Forcings: Generalities



Atmospheric Forcings

Surface wind stress

- Approaching sea surface, the geostrophic balance is broken, even for large scales.
- The major reason is the influences of the winds blowing over the sea surface, which causes the transfer of momentum (and energy) into the ocean through turbulent processes.
- The surface momentum flux into ocean is called the surface wind stress (τ), which is the tangential force (in the direction of the wind) exerting on the ocean per unit area (Unit: Newton per square meter)



Wind stress Calculation

- Direct measurement of wind stress is difficult.
- Wind stress is mostly derived from meteorological observations near the sea surface using the bulk formula with empirical parameters.

The **transfer of momentum** between the atmosphere and the Ocean is given by the **stress** $\vec{\tau}$

$$\vec{\tau} = \rho_a C_d (\vec{U}_{rel} \cdot |\vec{U}_{rel}|)$$

Relative motion between the two fluids, then:

$$\vec{U}_{rel} = \vec{U}_a - \vec{U}_o$$

Note that C_d is highly non-linear with $\vec{U}_{rel}$

Atmospheric Forcings

Drag Coefficient C_d

- C_d is dimensionless, ranging from 0.001 to 0.0025 (A median value is about 0.0013). Its magnitude mainly depends on local wind stress and local stability.

• C_d Dependence on stability (air-sea temperature difference).

More important for light wind situation

For mid-latitude, the stability effect is usually small but in tropical and subtropical regions, it should be included.

• C_d Dependence on wind speed.

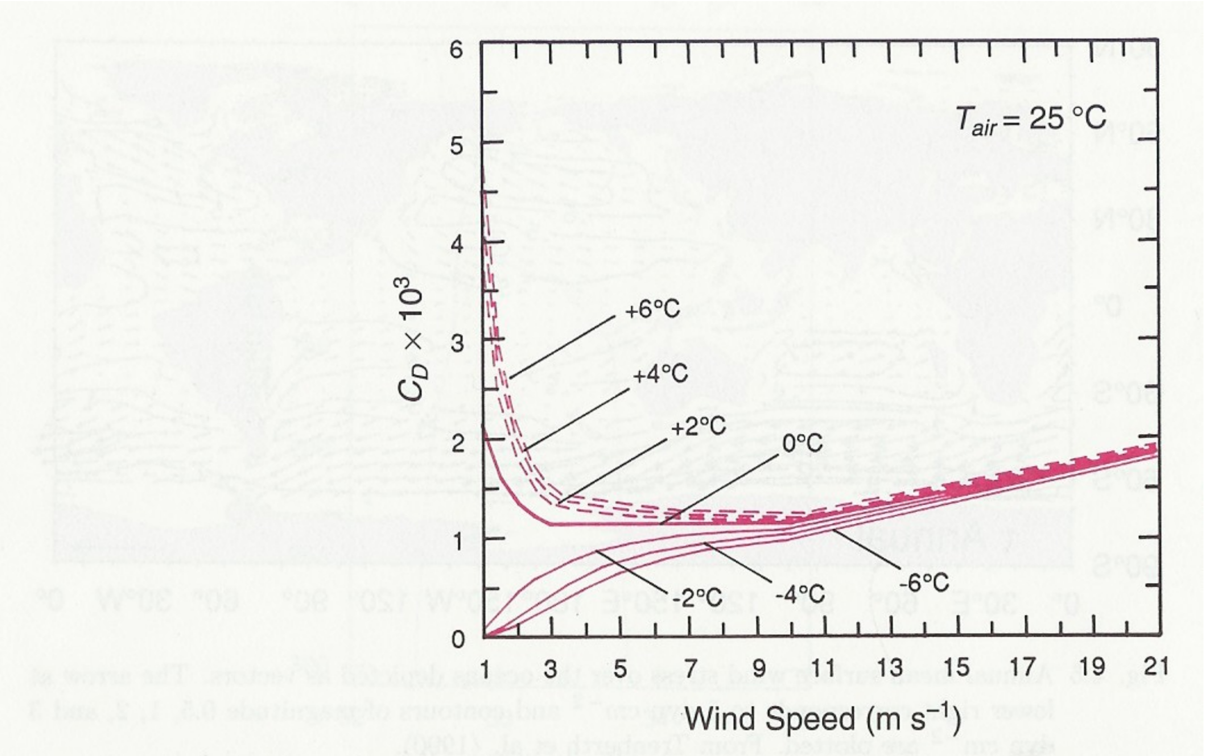


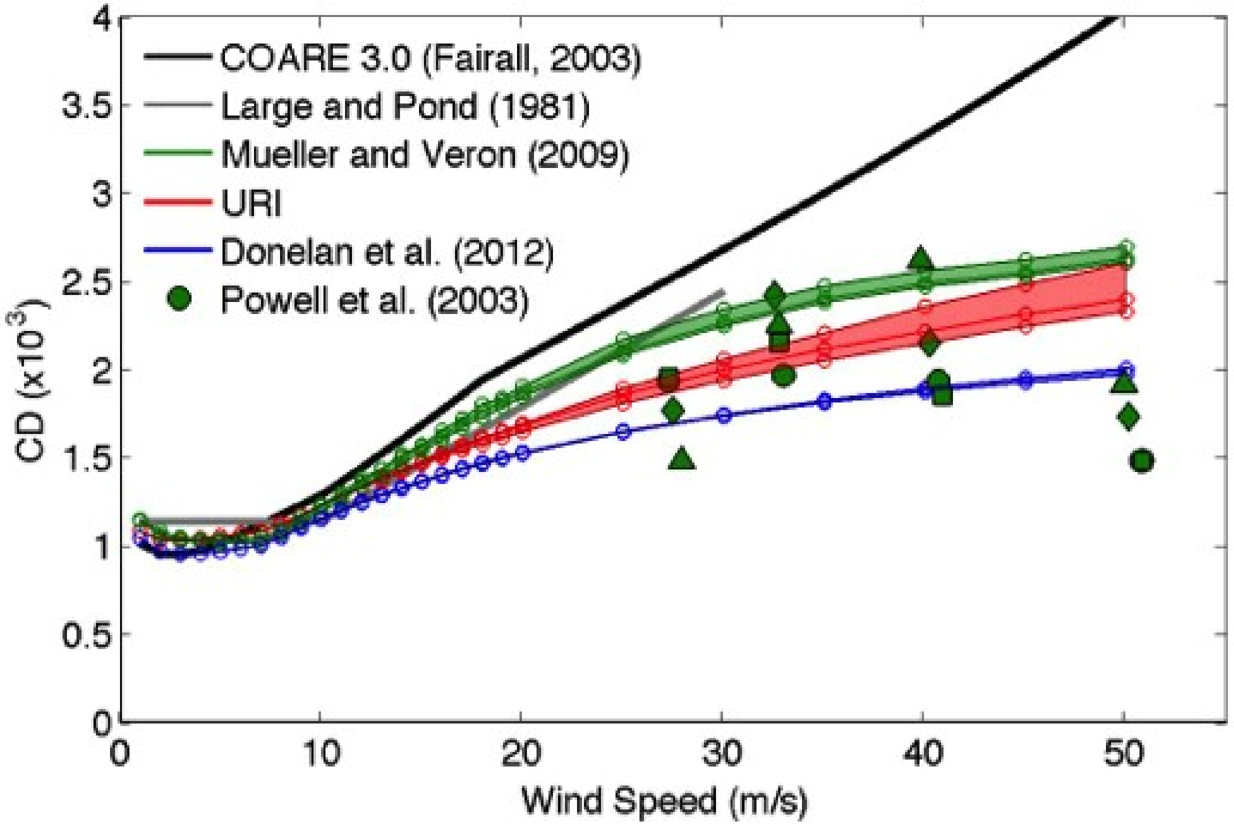
Fig. 4.6 The wind stress is computed from the wind speed, V, according to the formula $\tau = \rho_a C_D V \mathbf{v}$ and it is in the direction of the wind. Shown is the drag coefficient ($\times 10^3$), C_D , as a function of wind speed and atmospheric stability, as measured by the air–sea temperature differences, based on Large and Pond (1981), as given by Trenberth et al. (1989). Values are for an air temperature of 25°C and dashed lines indicate air less than sea temperatures (unstable).

Atmospheric Forcings

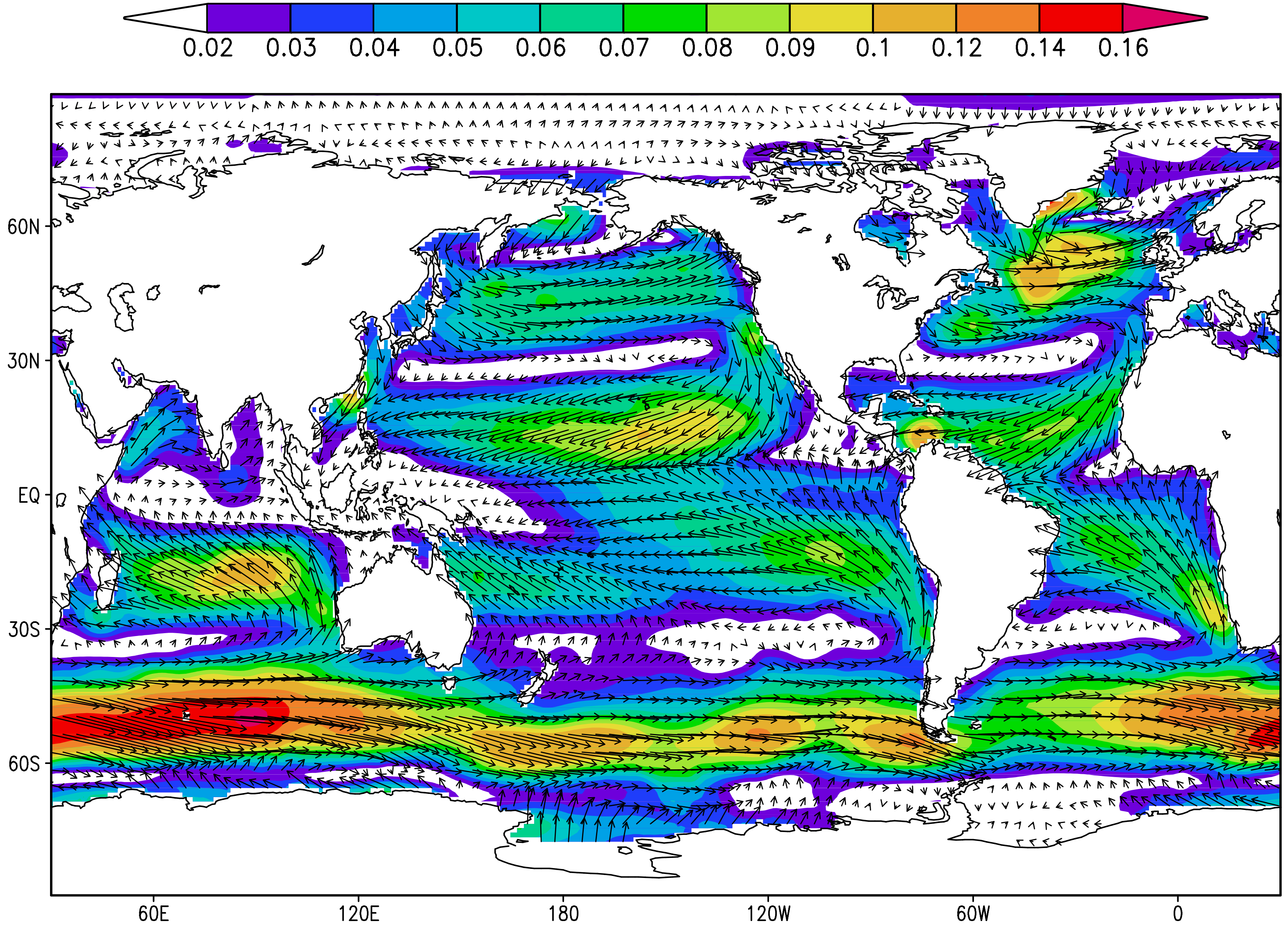
C_d dependence on wind speed in neutral condition

Large uncertainty between estimates (especially in low wind speed).

Lack data in high wind

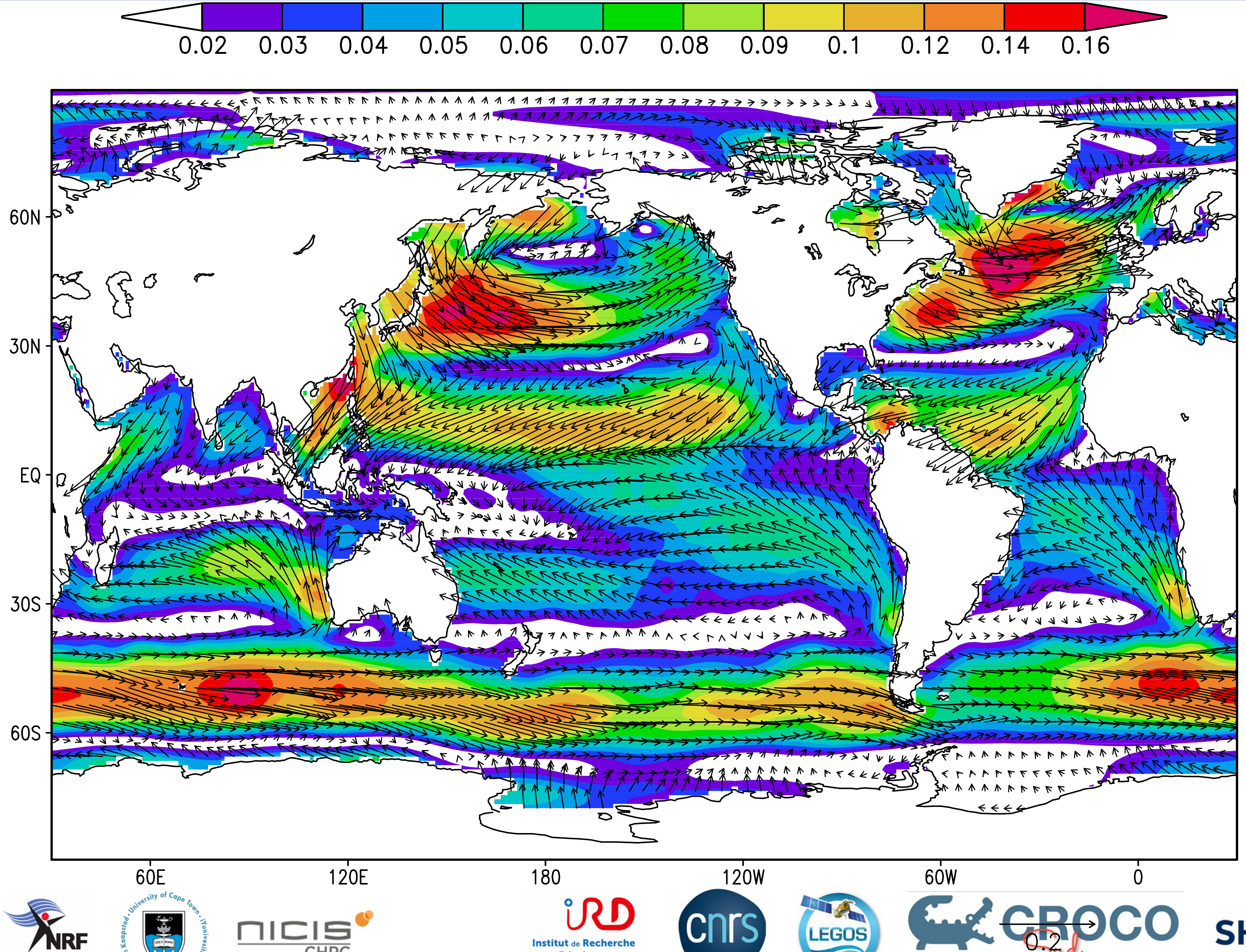


Atmospheric forcings



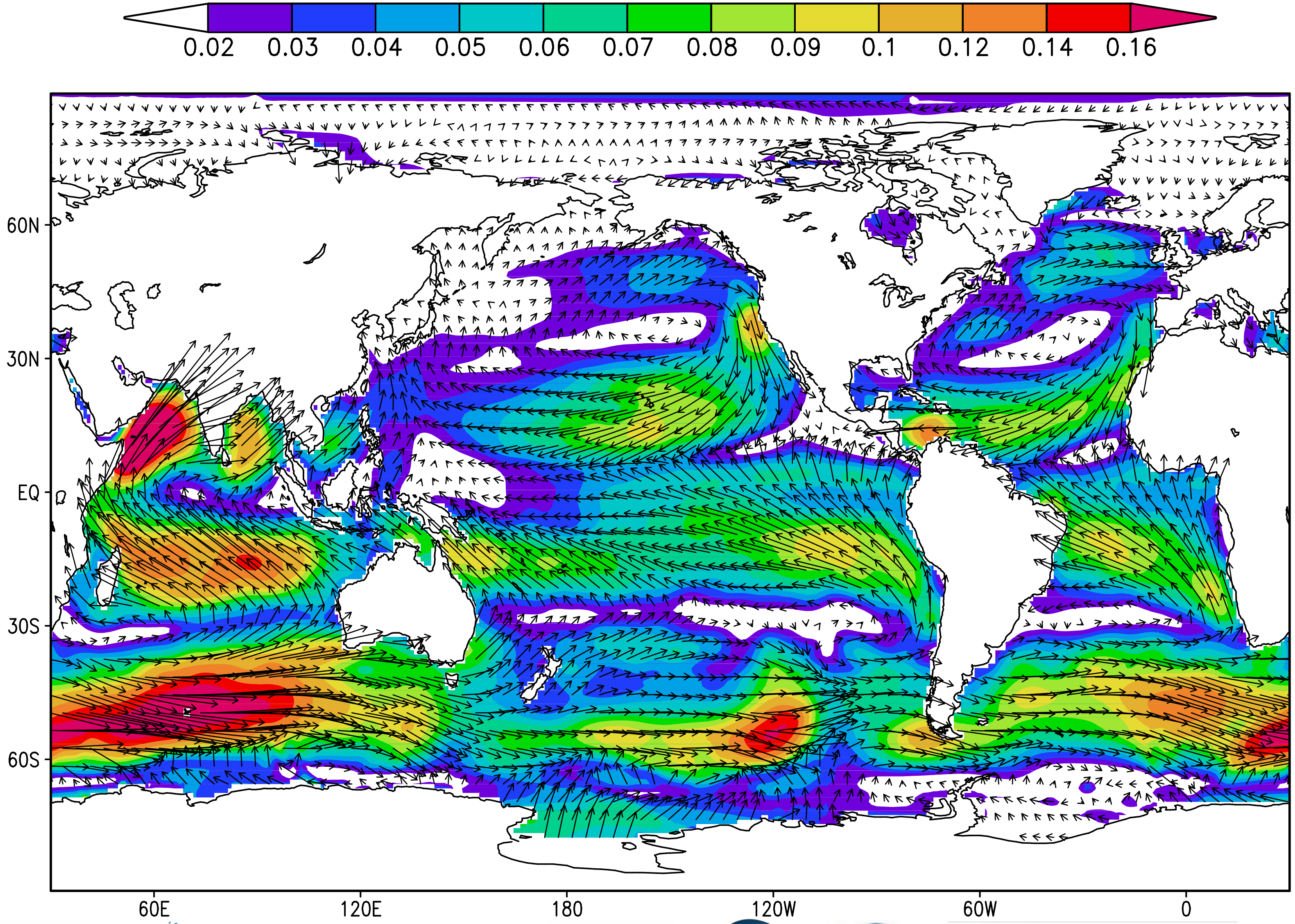
Annual Mean
surface wind stress

Atmospheric forcings



December-January-February mean wind stress

Atmospheric forcings



June-July-August mean wind stress

Atmospheric forcings

- **Two possibilities** :

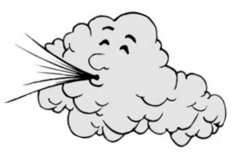
Directly read Taux, Tauy
in the forcing files in (N.m^{-2})

- **matlab: make_forcing**
Uses stress data from COADS
Look at the data !

- **ROMS: CPP_KEYS=**

```
undef BULK_FLUX
```

Use of a bulk formula :



- Compute the wind stress from the *Cd* drag coefficient, model SST and wind stress

- **matlab: make_bulk**
- **Uses wind data from COADS**
- **Look at the data !**
- **ROMS: CPP_KEYS=**

```
define BULK_FLUX
```

Fairall formula is by default but please note that other bulk formulae are possible

Coastal and Regional Ocean COmmunity model

Atmospheric forcings: Heat fluxes

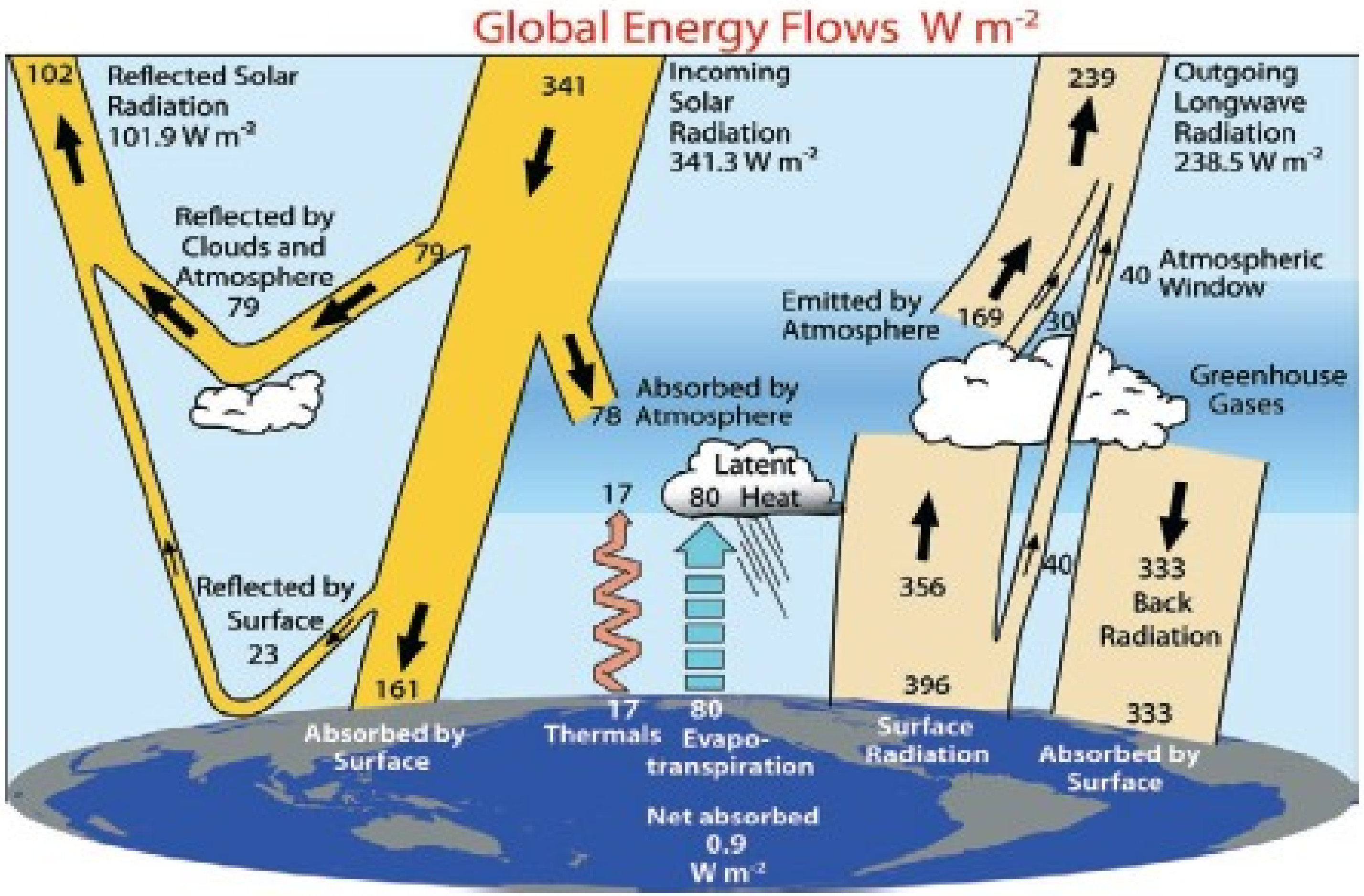
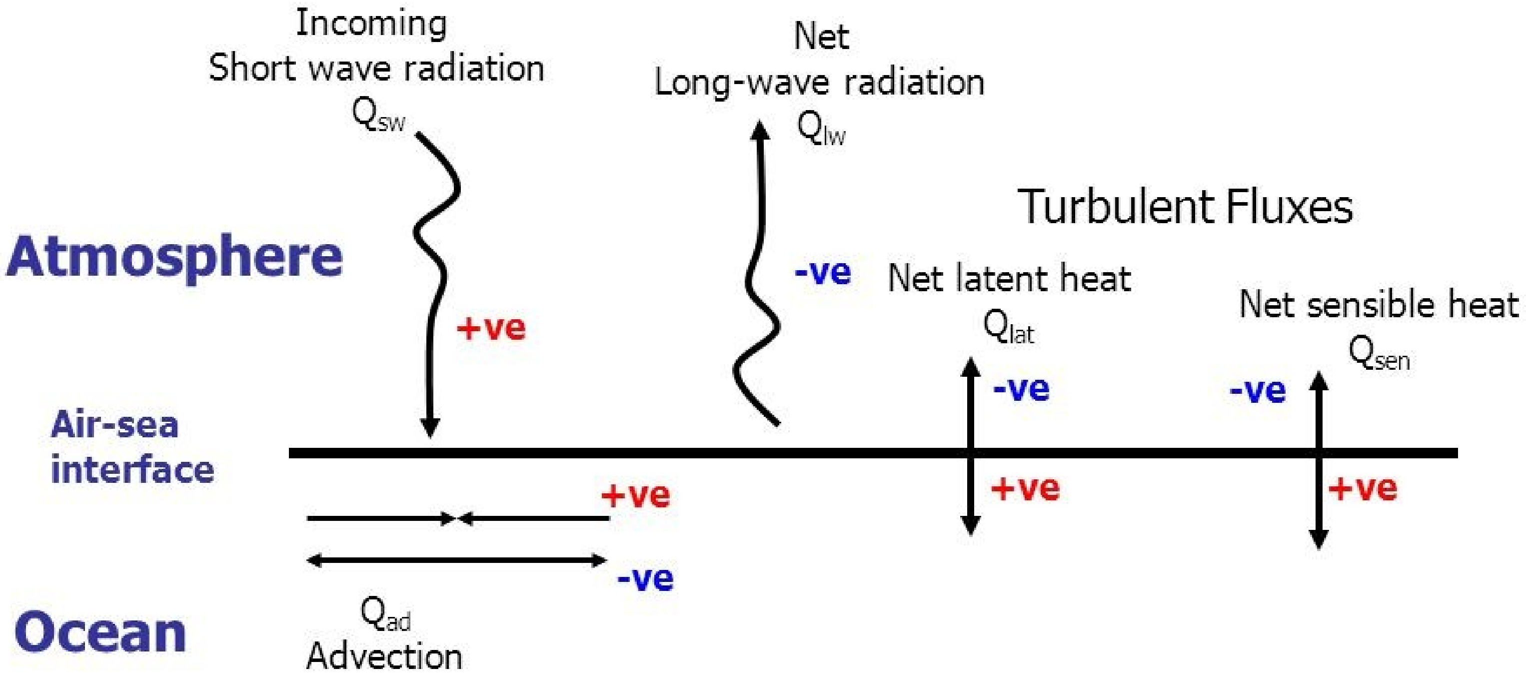


FIG. 1. The global annual mean Earth's energy budget for the Mar 2000 to May 2004 period (W m^{-2}). The broad arrows indicate the schematic flow of energy in proportion to their importance.

Atmospheric forcings: Heat fluxes

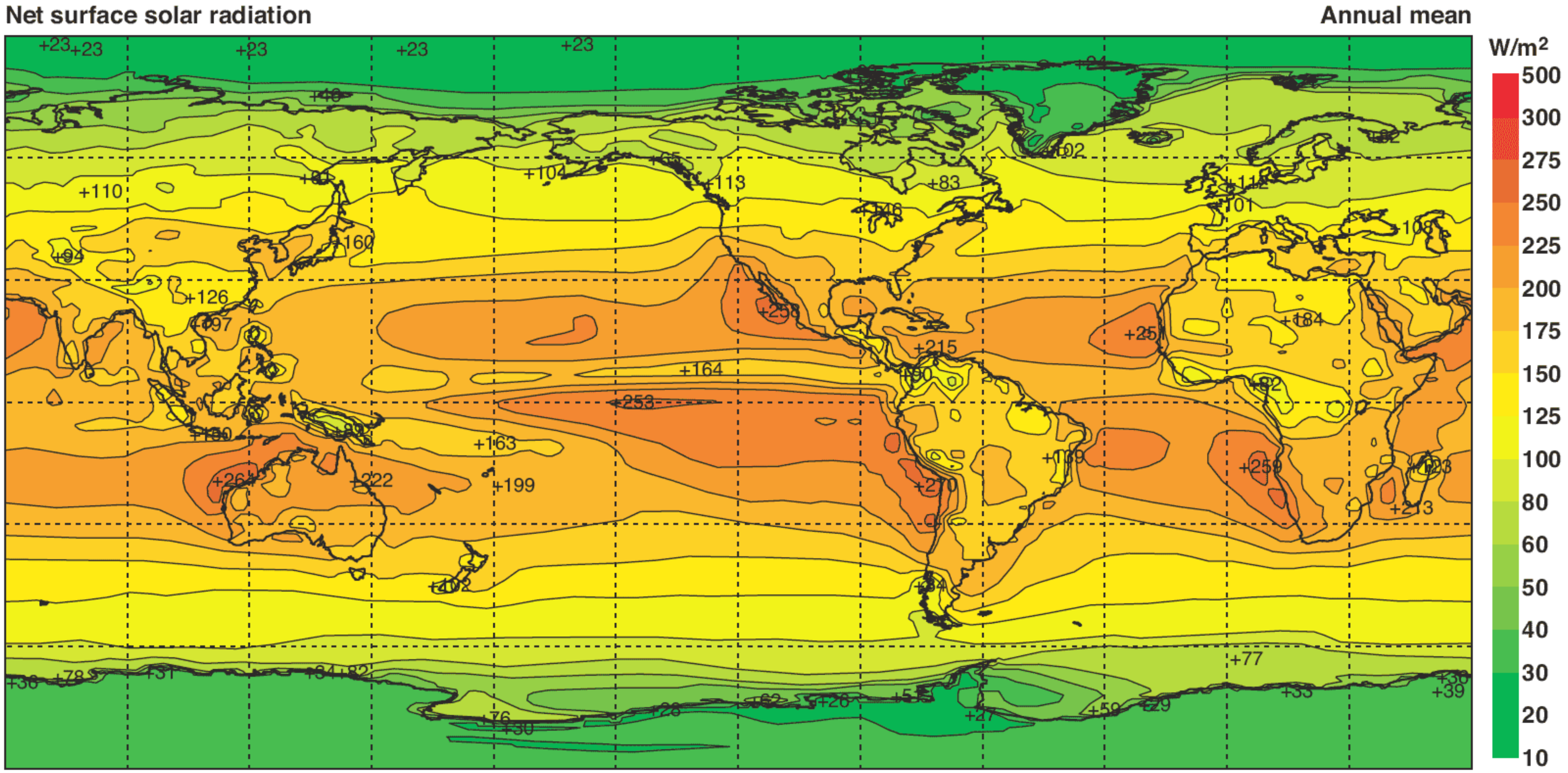
Heat budget equation

Heat fluxes into a region of ocean:



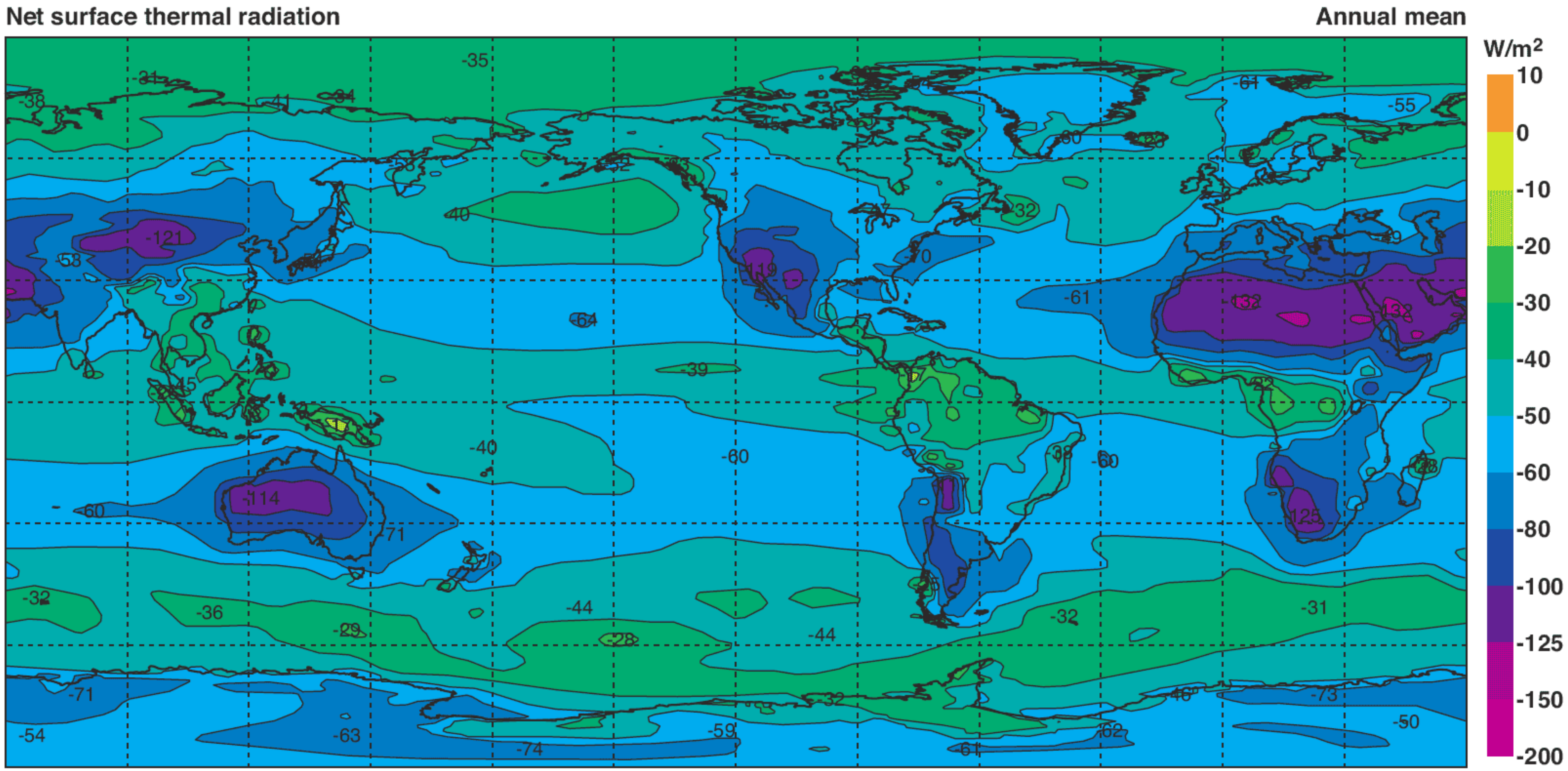
$$Q_{net} = Q_{sw} + Q_{lw} + Q_{lat} + Q_{sen} + Q_{ad}$$

Atmospheric forcings: Heat fluxes



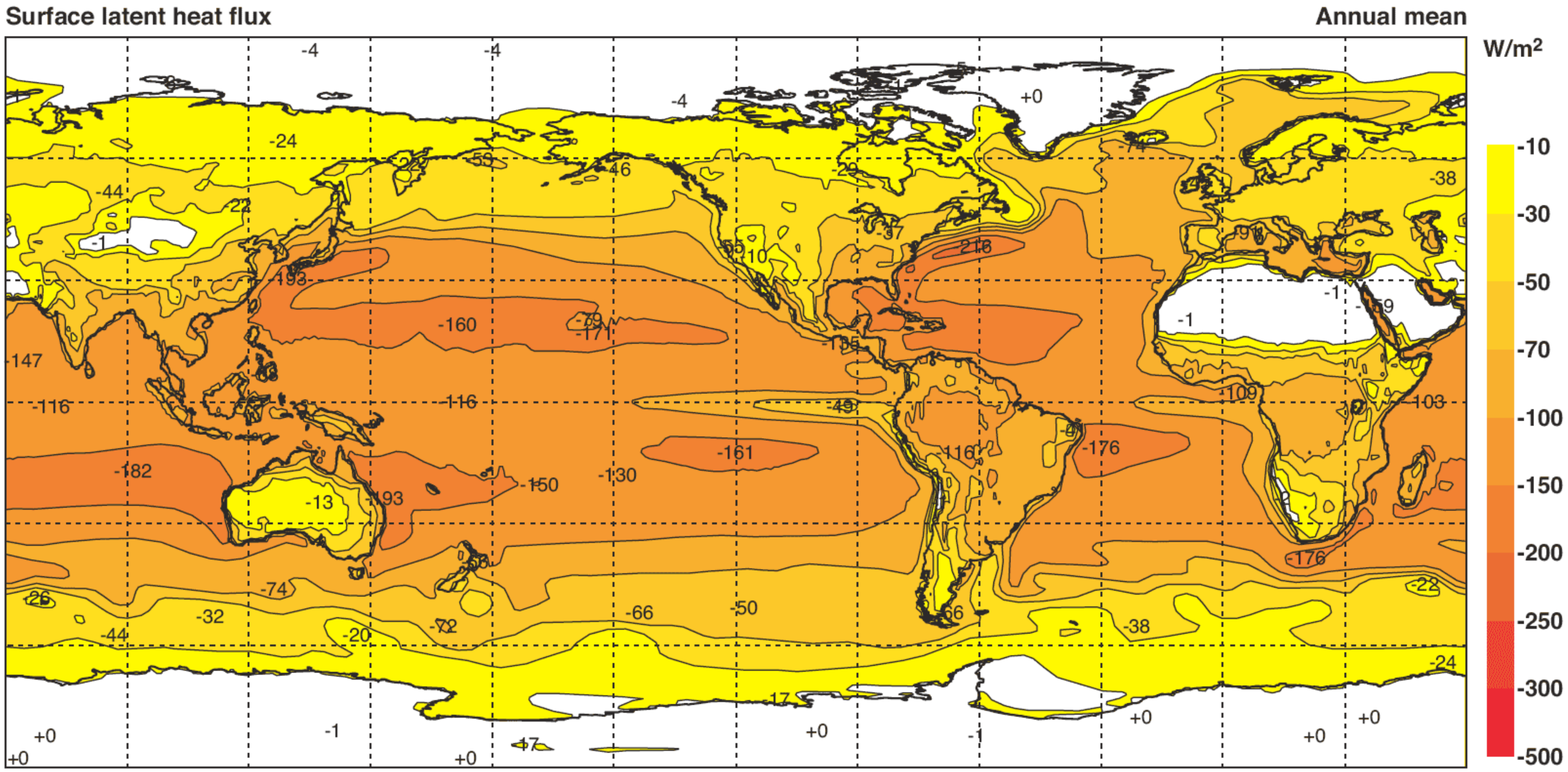
Q_{sw}

Atmospheric forcings: Heat fluxes



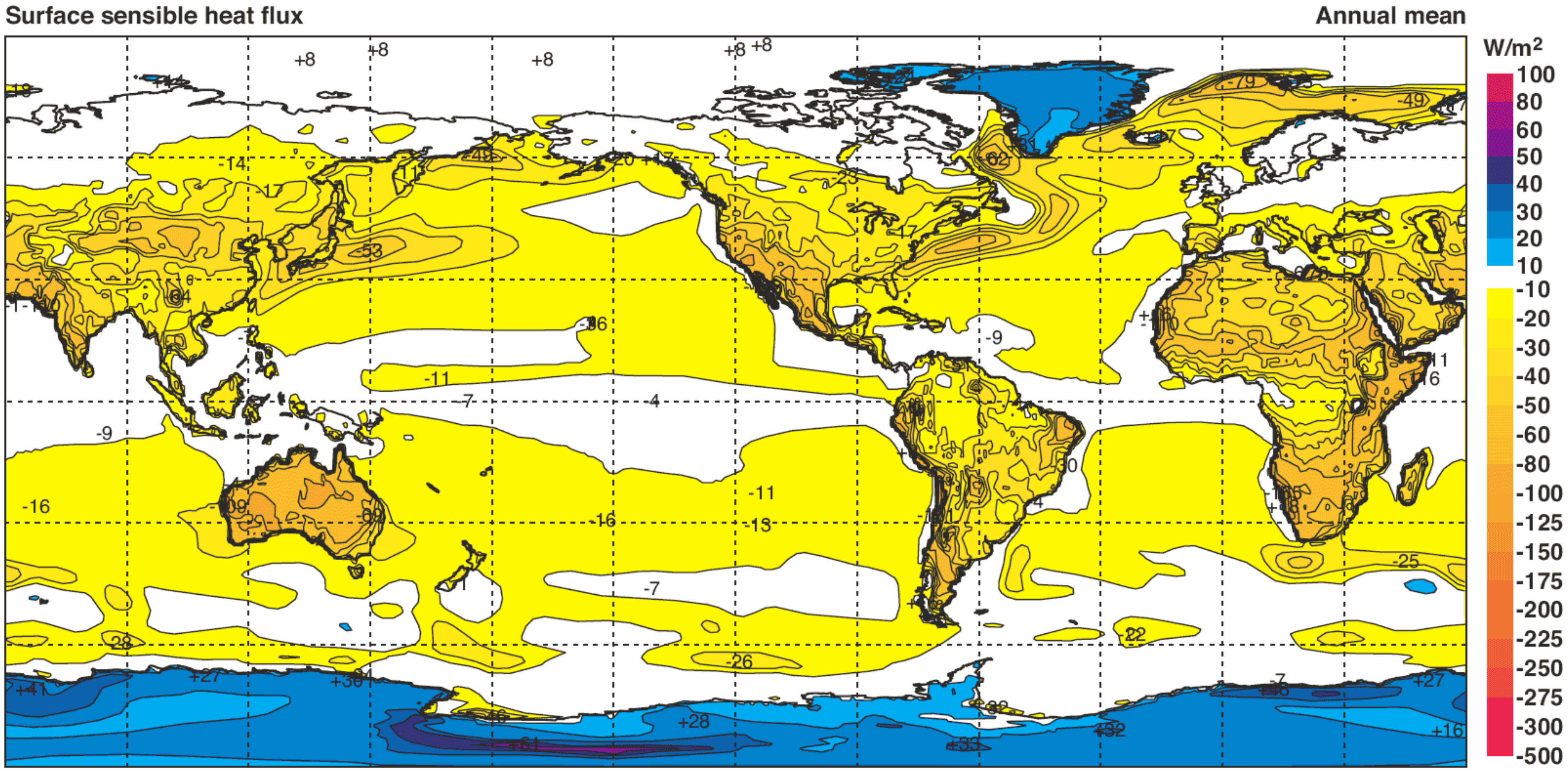
Q_{lw} = Black body radiation #T^4

Atmospheric forcings: Heat fluxes



$$Q_{lat} = \rho C_E (q_s - q_{10}) U_{10},$$

Atmospheric forcings: Heat fluxes

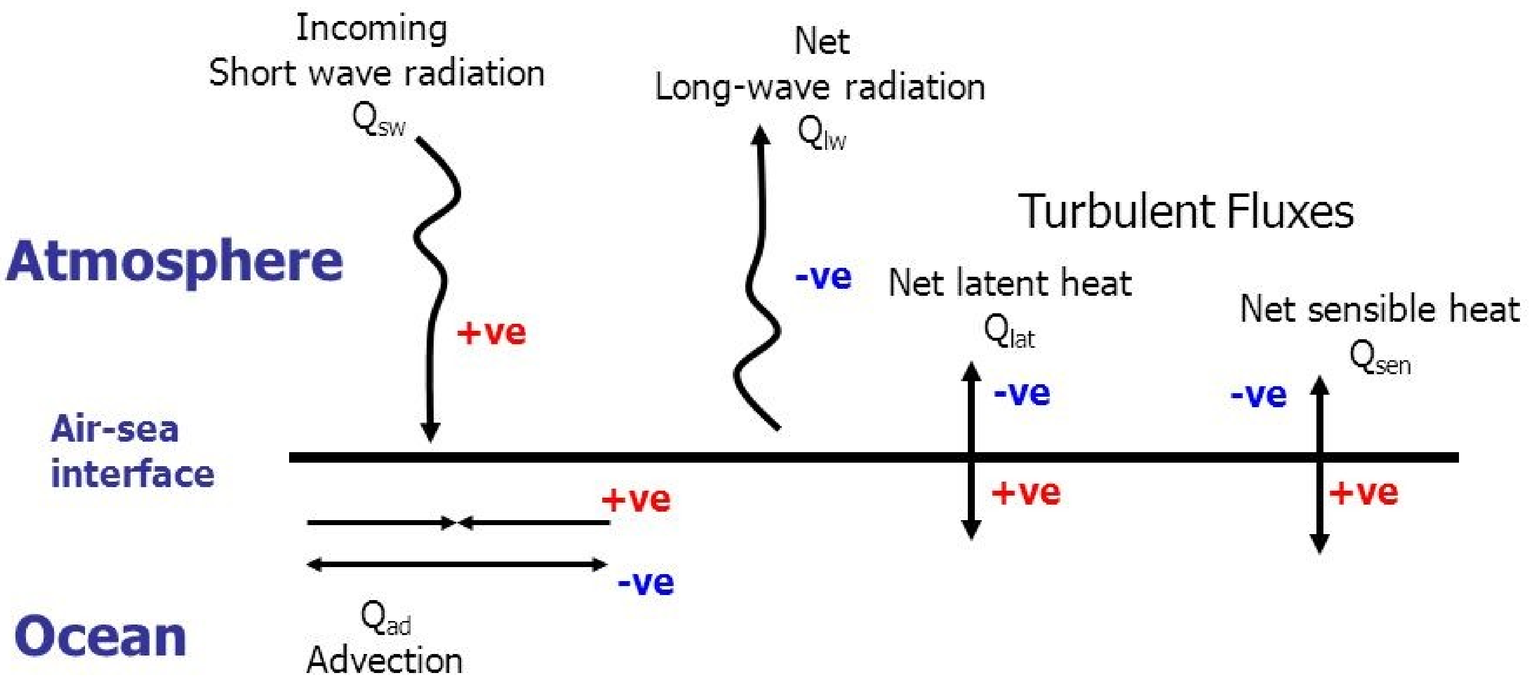


$$Q_{sen} = \rho C_H C_p (T_s - T_{10}) U_{10}$$

Atmospheric forcings: Heat fluxes

Heat budget equation

Heat fluxes into a region of ocean:



$$Q_{net} = Q_{sw} + Q_{lw} + Q_{lat} + Q_{sen} + Q_{ad}$$

Atmospheric forcings: Heat fluxes

Heat budget equation

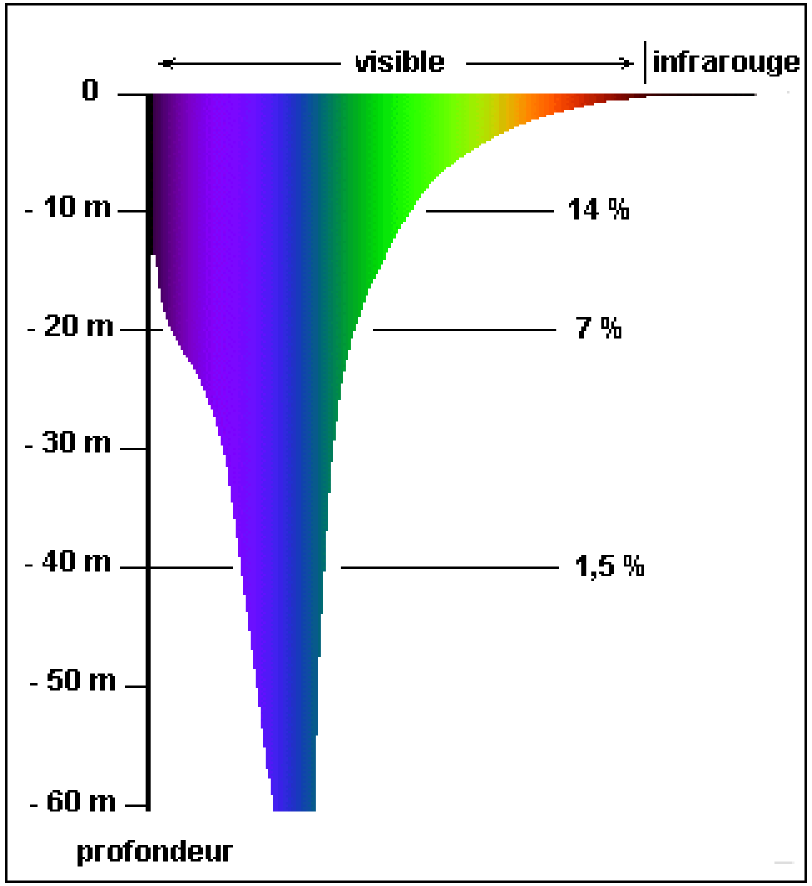
Heat fluxes into a region of ocean:

- At the surface :

$$Q_{lw}+Q_{lat}+Q_{sen}$$

- At the surface and sub-surface :

$$Q_{sw}$$



Atmospheric forcings: Heat fluxes

Heat budget equation

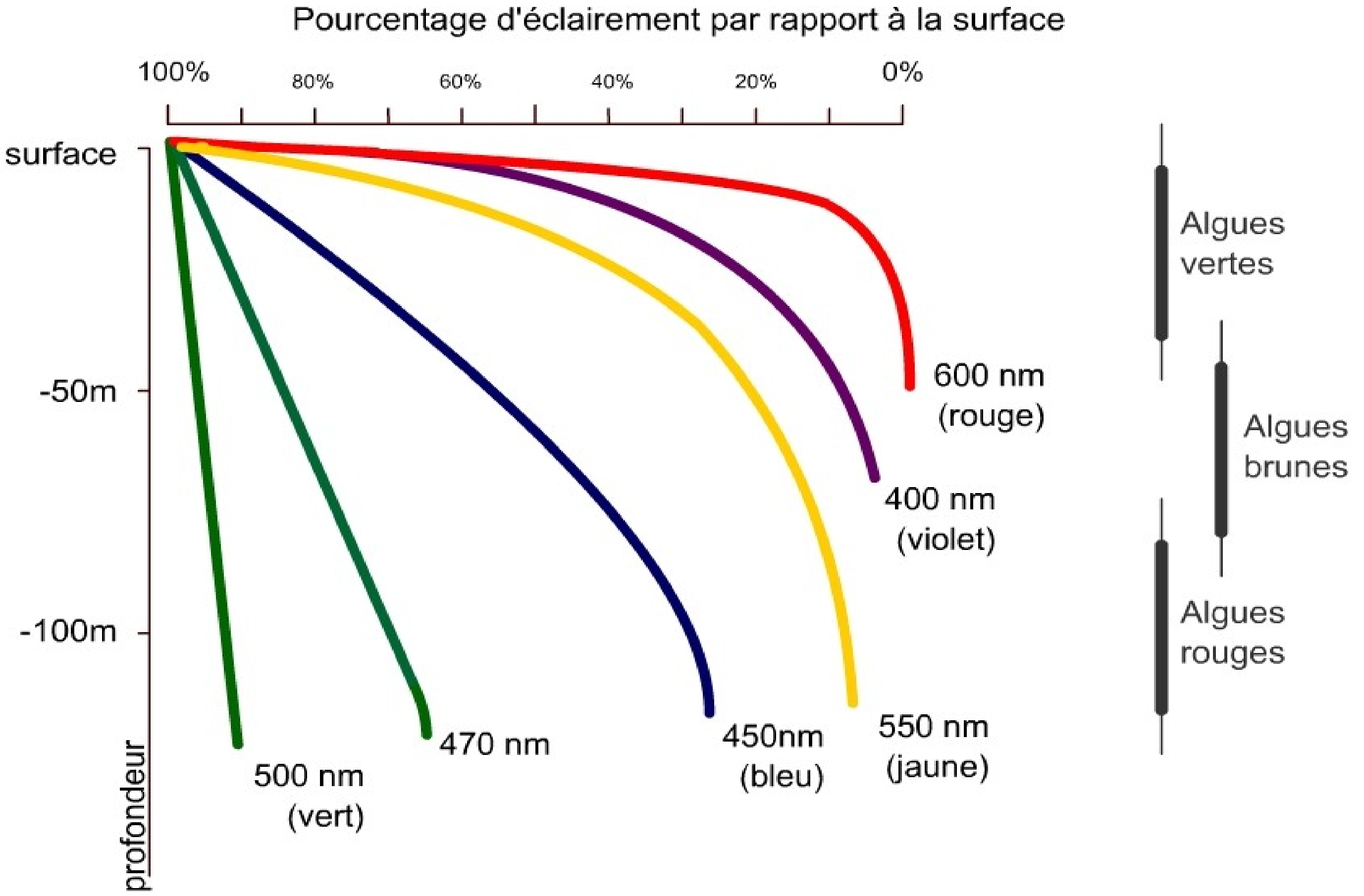
Heat fluxes into a region of ocean:

- At the surface :

$$Q_{lw}+Q_{lat}+Q_{sen}$$

- At the surface and sub-surface:

$$Q_{sw}$$



Atmospheric forcings: Heat fluxes

- **Two possibilities** :

Directly read of **Qnet** and **Qsolar**

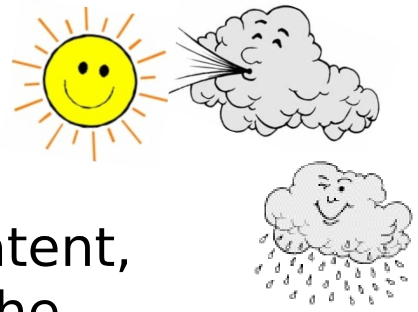
in the forcing files in (W.m^{-2})

- **matlab: make_forcing**
Uses stress data from COADS

- **ROMS: CPP_KEYS=**

undef BULK_FLUX

Use of a bulk formula :



- Compute the sensible, latent, long wave heat flux from the *CE,CH* coefficient using model SST, 10m wind speed, T2m q2m
  - Use **Qsolar** too.
  - **matlab: make_bulk**
  - **Uses wind data from COADS**

- **ROMS: CPP_KEYS=**
define BULK_FLUX
define BULK_LW:

Longwave “in” only, outgoing longwave are calculated for the model SST

Atmospheric forcings: Fresh water fluxes

- **Two possibilities** :

Directly read of EmP
in the forcing files in (cm.day^{-1})

- **matlab: make_forcing**
Uses stress data from COADS

- **ROMS: CPP_KEYS=**
`undef BULK_FLUX`

Use of a bulk formula :

$$E = \rho C_E (q_s - q_{10}) U_{10},$$

- Compute the evaporation from *CE* coefficient using model SST, 10m wind speed, q2m
- Use Precipitation too.

- **matlab: make_bulk**
- **Uses wind data from COADS**

- **ROMS: CPP_KEYS=**

`define BULK_FLUX`

Fairall formula is by default but please note that other bulk formulae are possible

Atmospheric forcings: Air-sea interactions

➢ Heat fluxes & Freshwater fluxes

- Directly read the forcing files
- Use of a bulk formulae :

  - Heat flux : compute total heat flux from latent, sensible, solar and longwave fluxes and model SST
  - Freshwater flux : compute from evap, prate and model SSS

bulk_flux.F

➢ Wind stress:

- Directly read the forcing files
- Use of a bulk formulae :

  - Compute the wind stress from the *Cd* drag coefficient, model SST and wind stress

Atmospheric forcings: Recap and keys

Surface atmospheric boundary conditions

Heat fluxes & Freshwater fluxes:

- Directly read the forcing files
- OR use of a bulk formulae to compute
  - Heat flux : compute total heat flux from latent, sensible, solar and long-wave fluxes and model SST
  - Fresh-water flux : compute from evap, prate and model SSS

Wind stress:

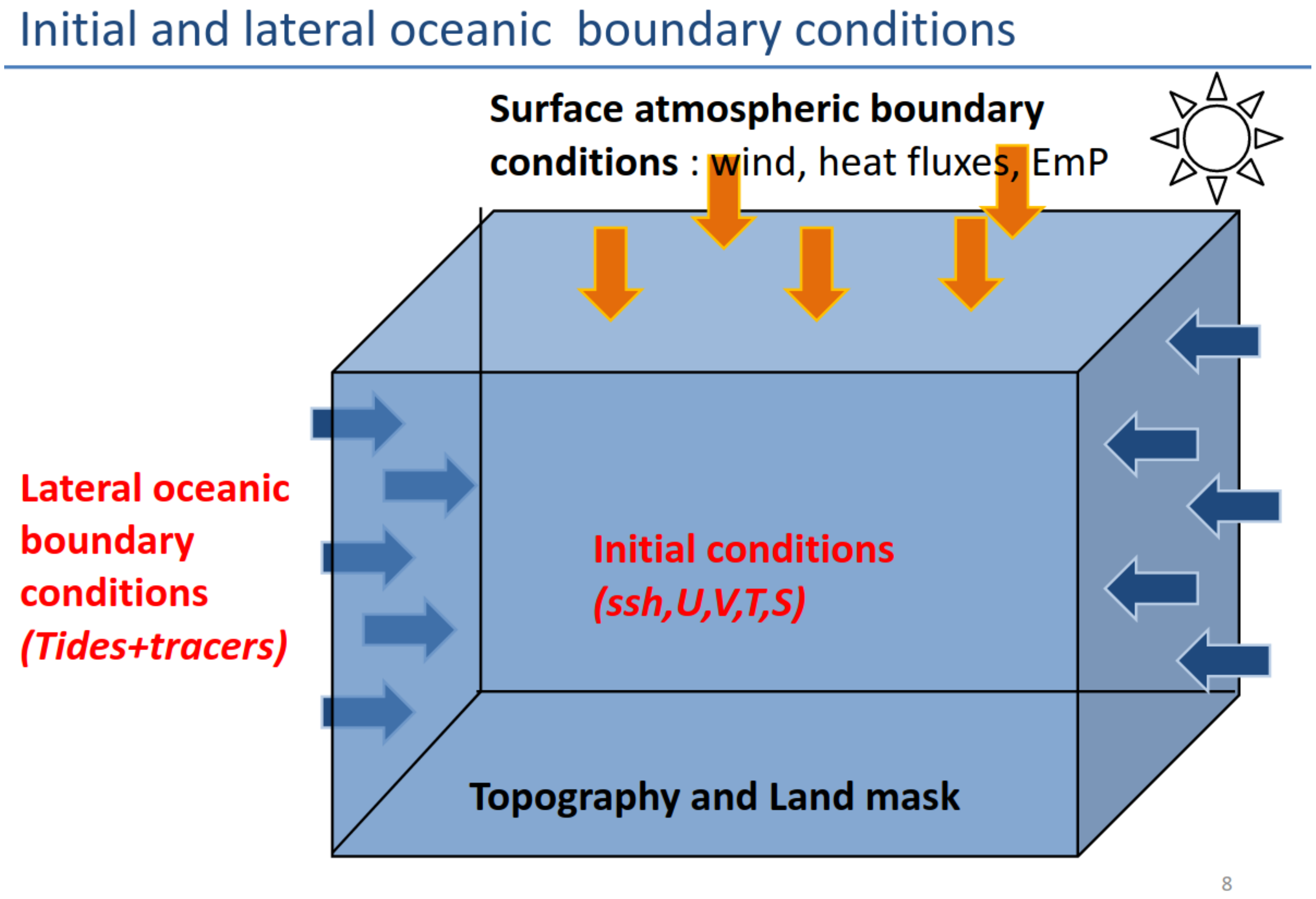
- Directly read the wind stress files
- Or use of bulk formulae to compute the wind stress from the Cd drag coefficient, model SST and wind

cppdefs.h

```
# undef  BSTRESS_FAST
                          /* Vertical Mixing */
# undef  BODYFORCE
# undef  BVF_MIXING
# define LMD_MIXING
# undef  GLS_MIXING
# ifdef LMD_MIXING
#  define LMD_SKPP
#  define LMD_BKPP
#  define LMD_RIMIX
#  define LMD_CONVEC
#  undef  LMD_DDMIX
#  define LMD_NONLOCAL
#  undef  MLCONVEC
# endif
                          /* Surface Forcing */
# undef BULK_FLUX
# ifdef BULK_FLUX
#  define BULK_FAIRALL
#  define BULK_LW
#  define BULK_EP
#  define BULK_SMFLUX
#  undef  SST_SKIN
#  undef  ANA_DIURNAL_SW
#  undef  ONLINE
#  ifdef ONLINE
#   undef  AROME
#   undef  ERA_ECMWF
#  endif
#  undef READ_PATM
#  ifdef READ_PATM
#   define OBC_PATM
#  endif
# else
#  define QCORRECTION
#  define SFLX_CORR
#  undef  SFLX_CORR_COEF
#  define ANA_DIURNAL_SW
# endif
# undef SMFLUX_CFB
# undef  SEA_ICE_NOFLUX
                          /* Wave-current interactions */
# ifdef OW_COUPLING
#  define MRL_WCI
#  define BBL
# endif
# ifdef MRL_WCI
#  ifndef OW_COUPLING
#   define WAVE_OFFLINE
```

Coastal and Regional Ocean COmmunity model

Forcings: Bonduary Conditions



Forcings: Bonduary Conditions

Open boundary conditions (OBC type)

Adaptative mixed radiations/nudging open boundary conditions (*Marchesiello et al, 2001*).

$$\frac{\partial \phi}{\partial t} + c_x \frac{\partial \phi}{\partial x} + c_y \frac{\partial \phi}{\partial y} = -\frac{1}{\tau}(\phi - \phi_{ext})$$

- Radiation, (Orlanski, 1982) for tracers
- Possibility to **use "Characteristic method" for barotropic mode :** Specially designed for tidal applications

Adaptative nudging term :

Adaptativity

- Ingoing signal (Cx >0) : strong nudging toward external data using $\tau = \tau_{in}$
- Outgoing signal (Cx >0) : weak nudging toward ext. Data $\tau = \tau_{out}$

$\tau_{out} \approx 180\,\text{days}$

$\tau_{in} \approx 1\,\text{days}$

τ_{M_in} , τ_{M_out} : momentum

τ_{T_in} , τ_{T_out} : tracer

```
# define OBC_M2CHARACT
# undef  OBC_M2ORLANSKI
# define OBC_M3ORLANSKI
# define OBC_TORLANSKI
# undef  OBC_M2SPECIFIED
# undef  OBC_M3SPECIFIED
# undef  OBC_TSPECIFIED
```

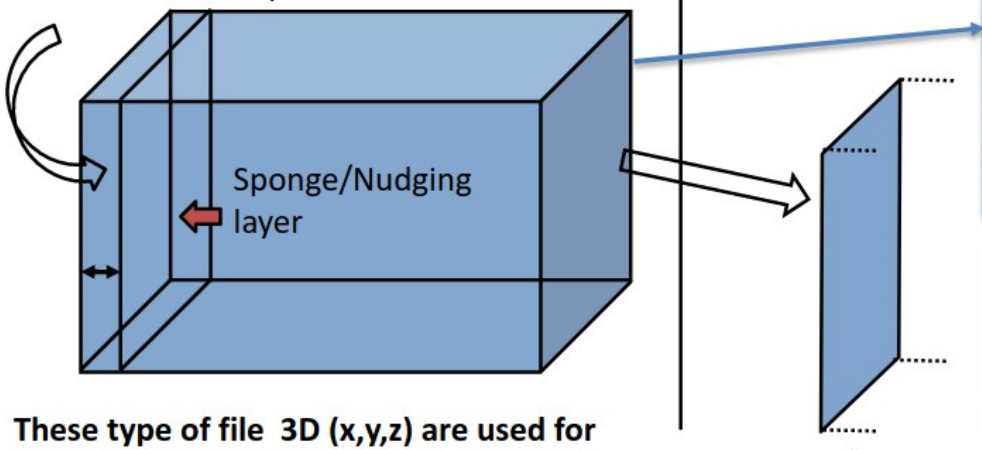
Forcings: Bonduary Conditions

3D Open boundary forcing (Clim or Bry)

Different ways to impose OBC

CLIM : **'3D+time' files (x,y,z,t)** only used at boundaries point + sponge/nudging layer : large amount of data unused.

BRY: **'2D+time' file (x,z,t) only** used at boundaries point : much less data needed !! **but no nudging layer** (for the moment)



These type of file 3D (x,y,z) are used for initialization

CLIM or BRY = Datasets+ Interpolation from crocotools

cppdefs.h

```
/* Lateral Forcing
# define CLIMATOLOGY
# ifdef CLIMATOLOGY
#  define ZCLIMATOLOGY
#  define M2CLIMATOLOGY
#  define M3CLIMATOLOGY
#  define TCLIMATOLOGY

#  define ZNUDGING
#  define M2NUDGING
#  define M3NUDGING
#  define TNUDGING
#  undef  ROBUST_DIAG
# endif

# undef  FRC_BRY
# ifdef FRC_BRY
#  define Z_FRC_BRY
#  define M2_FRC_BRY
#  define M3_FRC_BRY
#  define T_FRC_BRY
# endif
```

Forcings: Bonduary Conditions

Initial and lateral oceanic boundary conditions

```
#if defined REGIONAL
/*
!====================================================================
!               REGIONAL (realistic) Configurations
!====================================================================
!
!----------------------
! BASIC OPTIONS
!----------------------
!
*/
                      /* Configuration Name */
# define BENGUELA_LR
                      /* Parallelization */
# undef  OPENMP
# undef  MPI
# undef  MPI_NOLAND
                      /* I/O server */
# undef  XIOS
                      /* Non-hydrostatic option */
# undef  NBQ
                      /* Nesting */
# undef  AGRIF
# undef  AGRIF_2WAY
                      /* OA and OW Coupling via OASIS (MPI) */
# undef  OA_COUPLING
# undef  OW_COUPLING
                      /* Wave-current interactions */
# undef  MRL_WCI
                      /* Open Boundary Conditions */
# undef  TIDES
# define OBC_EAST
# define OBC_WEST
# define OBC_NORTH
# define OBC_SOUTH
                      /* Applications */
# undef  BIOLOGY
# undef  FLOATS
# undef  STATIONS
# undef  PASSIVE_TRACER
```

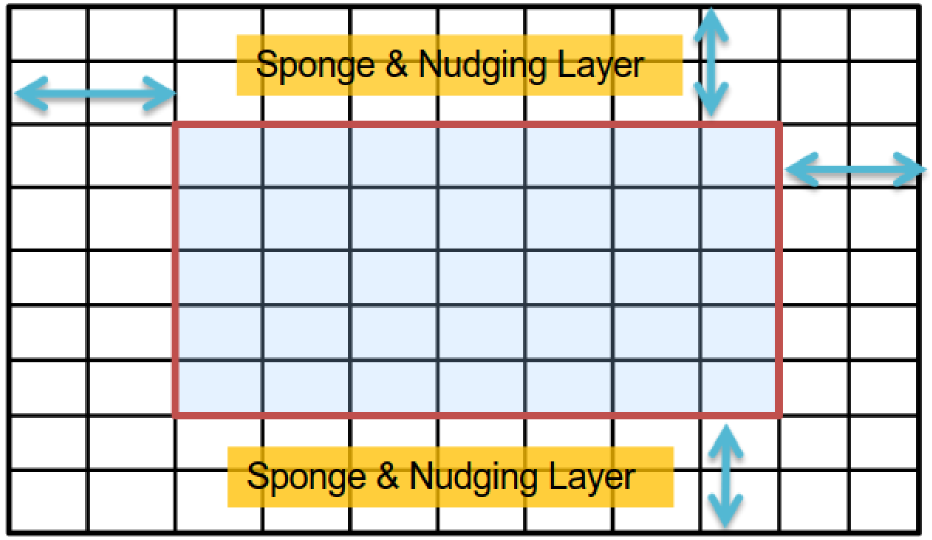
How to activate OBC ?

Use open boundary conditions

If undef then the domain is closed (see e.g. basin) or periodic.

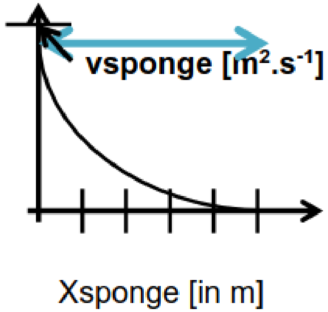
Forcings: Bonduary Conditions

Sponge/Nudging Layer

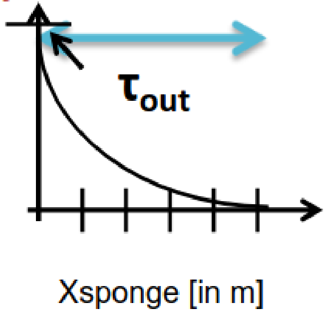


- Sponge : Additional viscosity/diffusivity
- Nudging : Add a weak nudging, $\tau = 0 \rightarrow \tau_{out}$, toward climatology, if available

$K^T h$, $K^m h$ profil across sponge layer



τ_{out} profil cross nudging layer



Activate areas of enhanced viscosity and diffusivity near lateral open boundaries.

Tutorial 04:
Create My CROCO Clim Config

OBJECTIVES

- Prepare the CROCO input files
- Compile the code
- Run the model
- Visualize the outputs

STEP 1: Logging onto the cluster

- From a terminal/konsole:

```
ssh -X login@scp.chpc.ac.za
```

- Reserve an interactive processor for pre-processing:

```
[login@login2 ~]$ qsubi1
```

- Go in your CROCO directory (lustre/croco):

```
[login@cnode022 ~]$ cd lustre/CROCO/croco-v2.0.1
```

- Go into your Run Clim:

```
[login@cnode0220 ]$ cd Run_Clim
```

STEP 2: Creating input files for Run_Clim

- Launch Matlab :

```
[login@cnode0220 Run_Clim]$ matlab -nodesktop
```

- Create your croco grid

```
>> start; make_grid;
```

- Create your surface forcing files

```
>> make_forcing; make_bulk; …
```

- Create your CROCO initial and boundary conditions

```
>> make_clim; OR make_bry; make_ini;
```

STEP 3: Compiling CROCO model

- Copy the script to compile the code in your Run_Clim directory :

```
cp /mnt/lustre/users/sillig/CROCO_TRAINING_Basic/3_Some_files/jobcomp_lengau .
```

- Edit and fix the parameter file **param.h**

```
[login@cnode0220 Run_Clim]$ nedit param.h &
```

- Edit and set the **cppdefs.h**

```
[login@cnode0220 Run_Clim]$ nedit cppdefs.h &
```

- Compile CROCO using the **jobcomp_lengau** script

```
[login@cnode0220 Run_Clim]$ ./jobcomp_lengau
```

STEP 4: Running CROCO

- Copy the job file to run the code in your Run_Clim directory :

```
cp /mnt/lustre/users/sillig/CROCO_TRAINING_Basic/3_Some_files/run_croco.pbs .
```

- Edit and fix the parameter file **croco_inter.in**

```
[login@cnode0220 Run_Clim]$ nedit croco_inter.in &
```

- Edit and fix the script **run_croco.pbs**

```
[login@cnode0220 Run_Clim]$ nedit run_croco.pbs &
```

- Launch your simulation

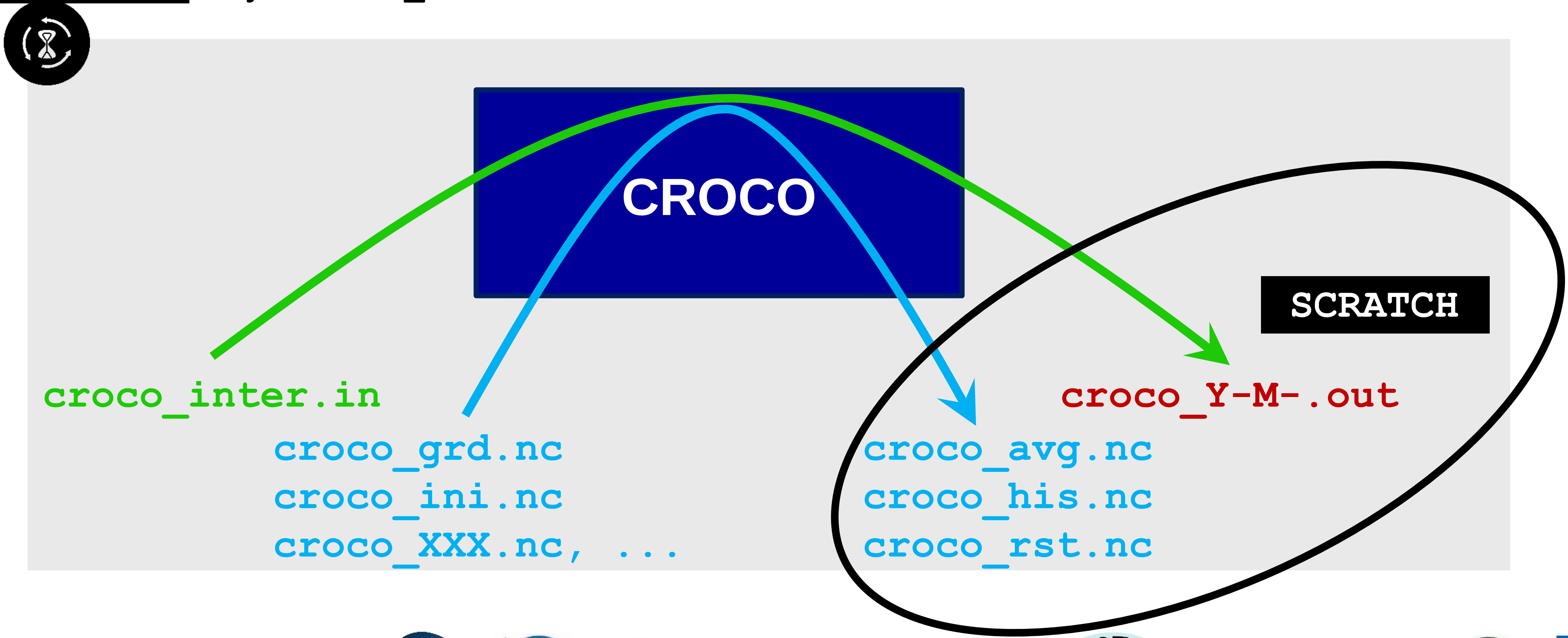
```
[login@cnode0220 Run_Clim]$ qsub run_croco.pbs
```

STEP 4: Running CROCO

Run_Clim

run_croco.pbs: → run_croco.pbs.oxxxx
run_croco.pbs.exxxx

1. Copy executable, input files, and croco_inter.in.template
2. Loop over time : adjust croco_inter.in and run CROCO:



Coastal and Regional Ocean COmmunity model

Variables and forcings

Prognostic variables:

u

v

temp

salt

zeta

Forcing variables:

sustr

svstr

srfl

stfl

precip

The Bug Sheet

- Problem associated with :
  - Diagnostic 1
    - Solution:
  - Diagnostic 2
    - Solution:

STEP 5: Visualising model outputs

- Launch Matlab and edit the following file:

```
>> edit croco_diags.m
>> croco_diags
```

- Make your first plots:

```
>> plot_diags
```

- Visualise the outputs with croco_gui

```
>> croco_gui
```

- Enjoy!!!

STEP 6: Exiting

- Exit Matlab:

```
exit
```

- Give back the compute node:

```
exit
```

- Logoff the Lengau cluster:

```
exit
```