Parallel code and corresponding data structure

Model→MPL(earlier MPE)→MPI

Model:

- Setup parallel env. : su0yoma→sumpini

  sumpini→mpl_init

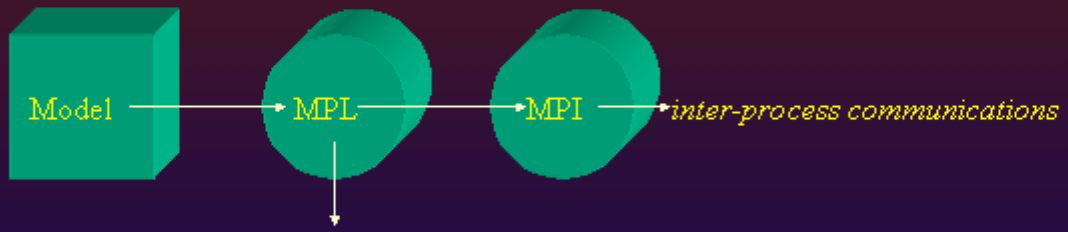
  |→su0dminit→*myproc: **myproc=mpl_myrank()***

  |→mpl_buffer_method→BUFFER

  ATTACH

Parallel code

MPL: an interface to the interface



New features compared to MPE:
Modules
Overloading

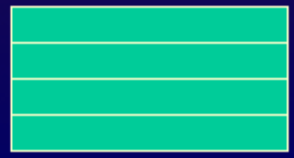
Parallel code

each procesor does the same

myproc

processors still execute the same code,
but *myproc* is different→e.g.:

ista=(myproc-1)*ndgl/nproca+1
iend=myproc*ndgl/nproca



Parallel code

SETUP phase: All „ista,iend" -type setting →setting up computational domain of each processor in different parts of time stepping

→Each processor works on its computational domain

→Information exchange between processors by *mpi_send,mpi_receive*

→Fortran programmer has a direct control on parrallelization:

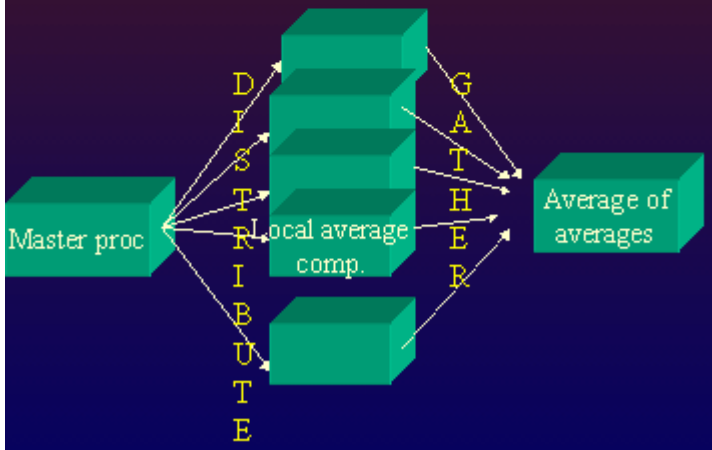
(design of domain splitting, what to exchange,etc.)←→in macro-tasking less control by programmer

Parallel code

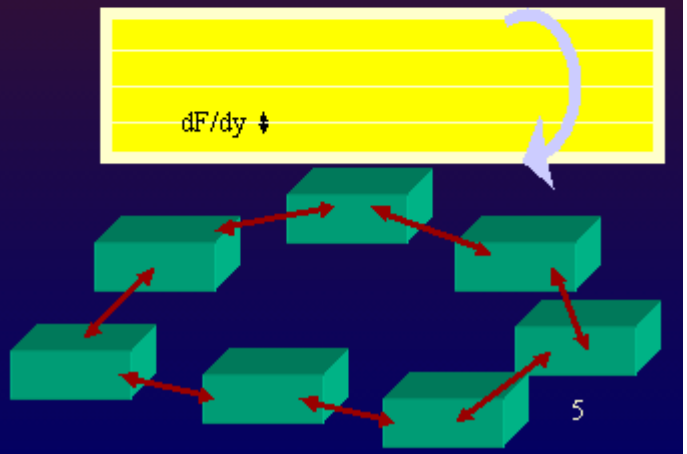
Design should depend on nature of problem:

E.g.:

Average of a huge array



Advection eq. by finite differences in time and space, periodic field



Parallel code

Typical problem: not enough buffer space allocated (env. variable for limit + *MBX_SIZE*)

To activate buffering put *MP_TYPE=2*

(without buffering one may hang forever in send/receive)

Parallel code

Other important tunable parameters:

ncombflen: size of commun. buffer (chunks)

lsplit, nprgpew, nprgpns ...

ALADIN is limited to

lsplit=.f. nprgpew=1 ... (see next slide)

ALADIN specific tuner: *TCDIS*

NOUTPUT,NSTRIN,NSTROUT ...

Parallel code

NPROCA (old notation with global meaning in each space)	*NPROCB* (old notation with global meaning in each space)
NPROC=NPROCA*NPROCB	
NPRGPNS: gridpoint space, north-south distribution	*NPRGPEW*: gridponit space, east-west distribution
NPRTRW: transform phase dimension1 *NPRTRNS*: " (north-south)	*NPRTRV*: transform phase dimension2 (vertical)
((NPRTRM: spectral space distribution)) Looks to be obsolete (??)	*NPRTRN*: spectral space distribution

For setup see *SUMPINI*

Transposition routines (e.g.: *TRGTOL,ETRMTOS…*)

Parallel code

Processor identifiers:

NPROC → ***MYPROC***

(in *TAL* mysetns)

NPROCA,NPRGPNS → ***MYSETA***
NPROCB,NPRGPEW → ***MYSETB***
NPRTRW,NPRTRNS → ***MYSETW***
NPRTRV → ***MYSETV***
NPRTRM → ***MYSETM***
NPRTRN → ***MYSETN***

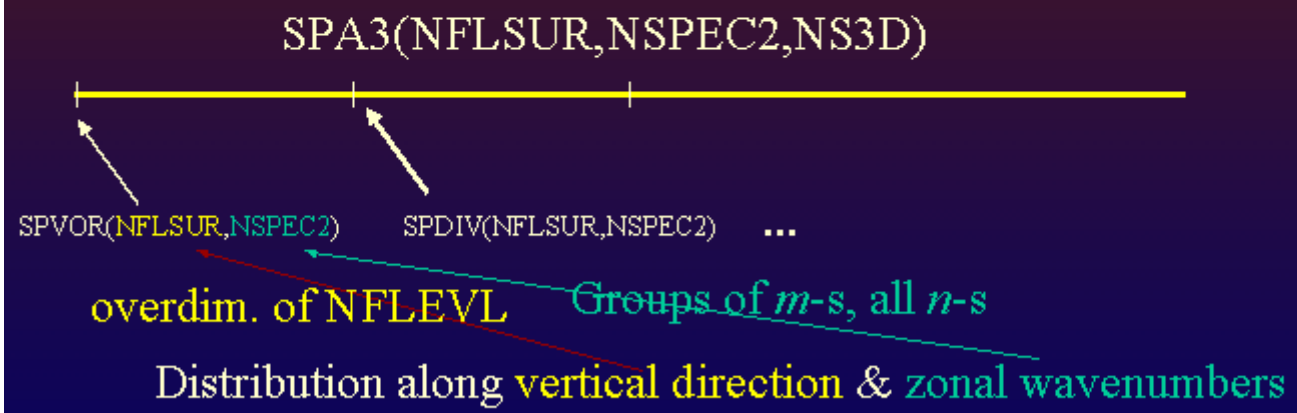
(See PE2SET): myproc: 1,2, ……..NPROC

e.g.:mysetv: 1,2,…NPRTRV,1,2,…
& mysetw: 1,1,1,2,2,2,…,NPRTRW,NPRTRW

Data structure in distributed environment

Spectral space:

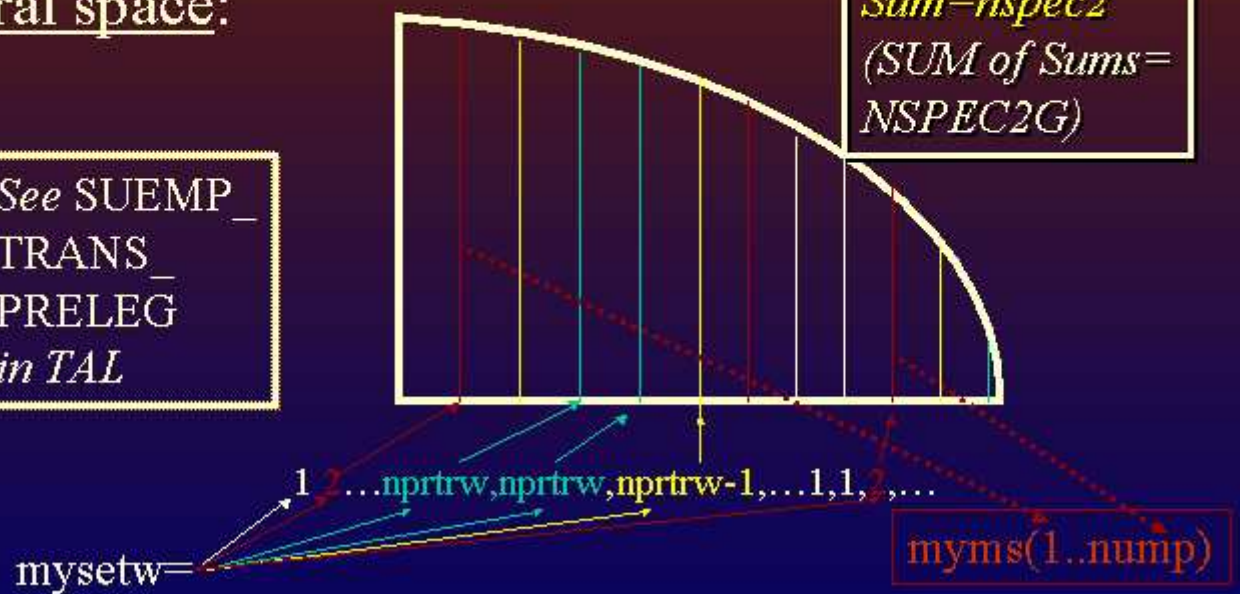
- 3D fields: (see *SUALSPA*)



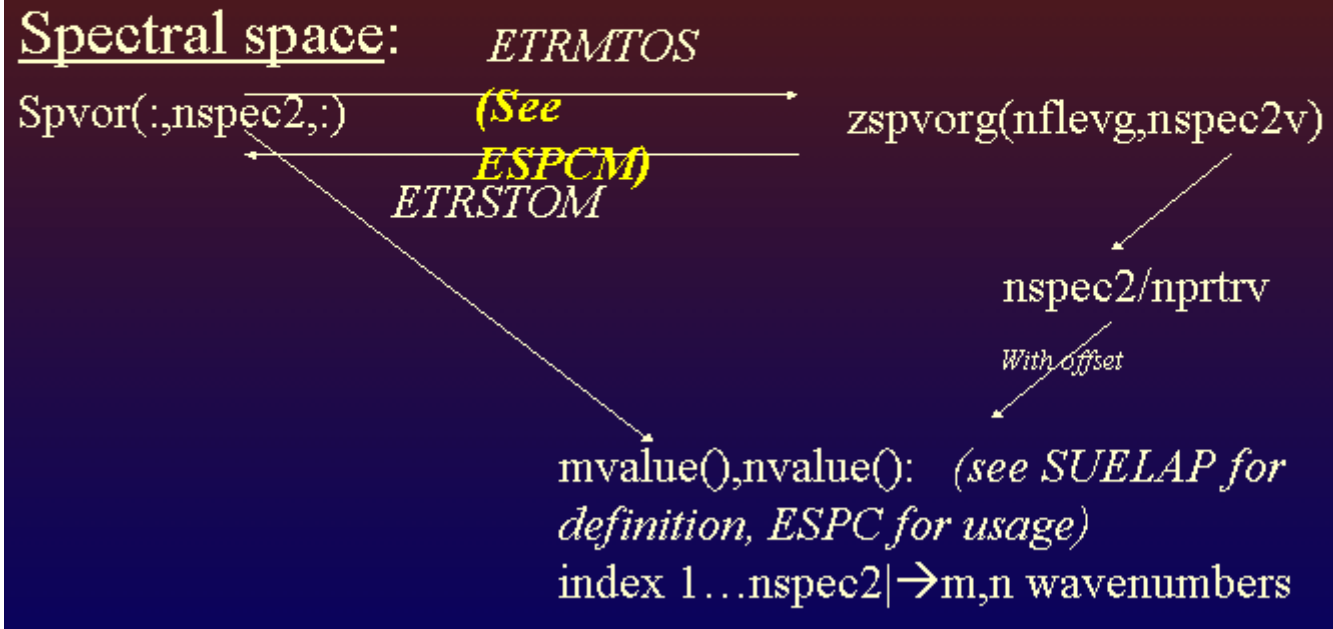
Data structure in distributed environment

Spectral space:

See SUEMP_TRANS_PRELEG *in TAL*

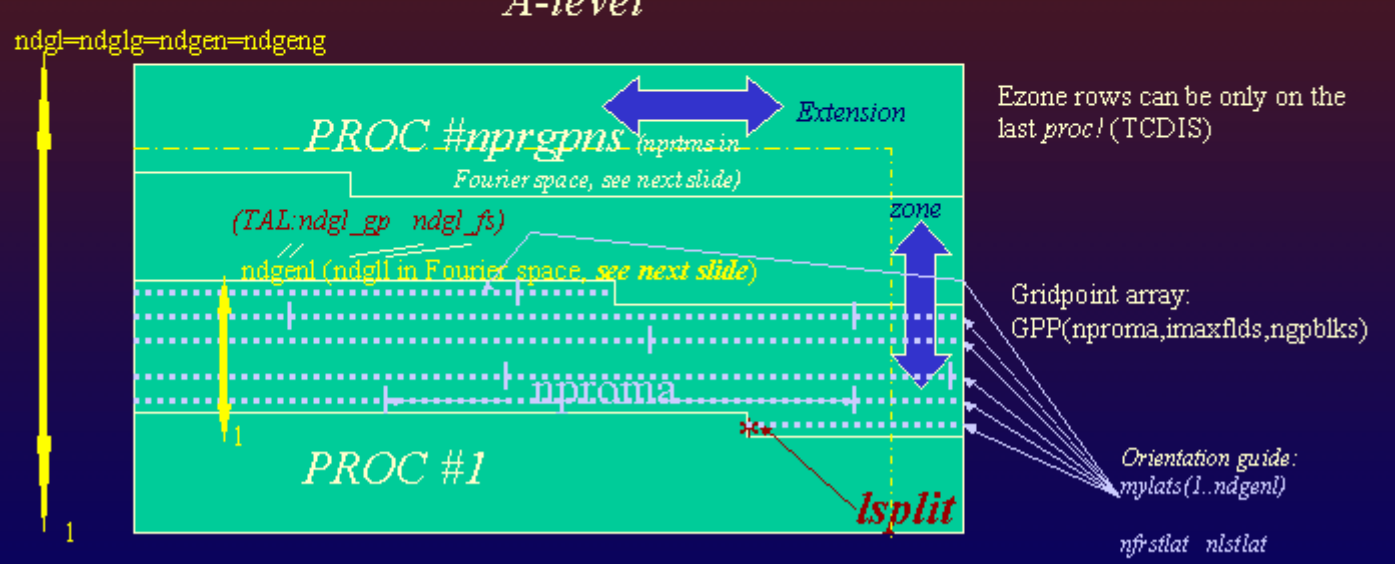


Data structure in distributed environment



Data structure in distributed environment

Gridpoint space:



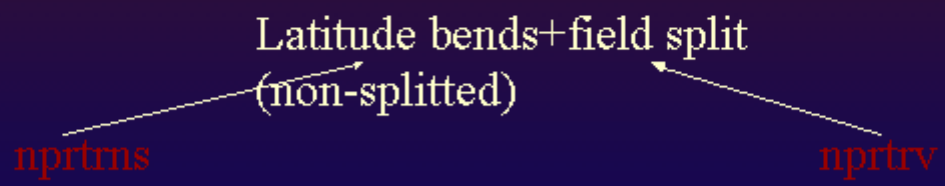
Data structure in distributed environment

Fourier space:

Direct transforms: Blocked GP arrays

(Inverse is its opposite With TRLTOG)

TRGTOL



*See in **TAL** SUEMP_TRANS →SUEMPLATF →SUEMPLATB && EFTDIR_CTL →TRGTOL*

Data structure in distributed environment

Gridpoint space:

B-level

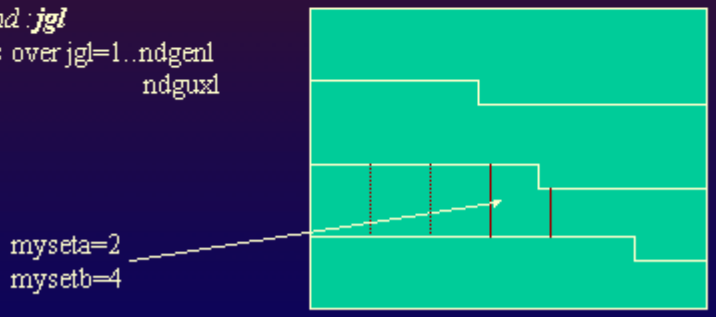
(See SUSTAONL of TAL)

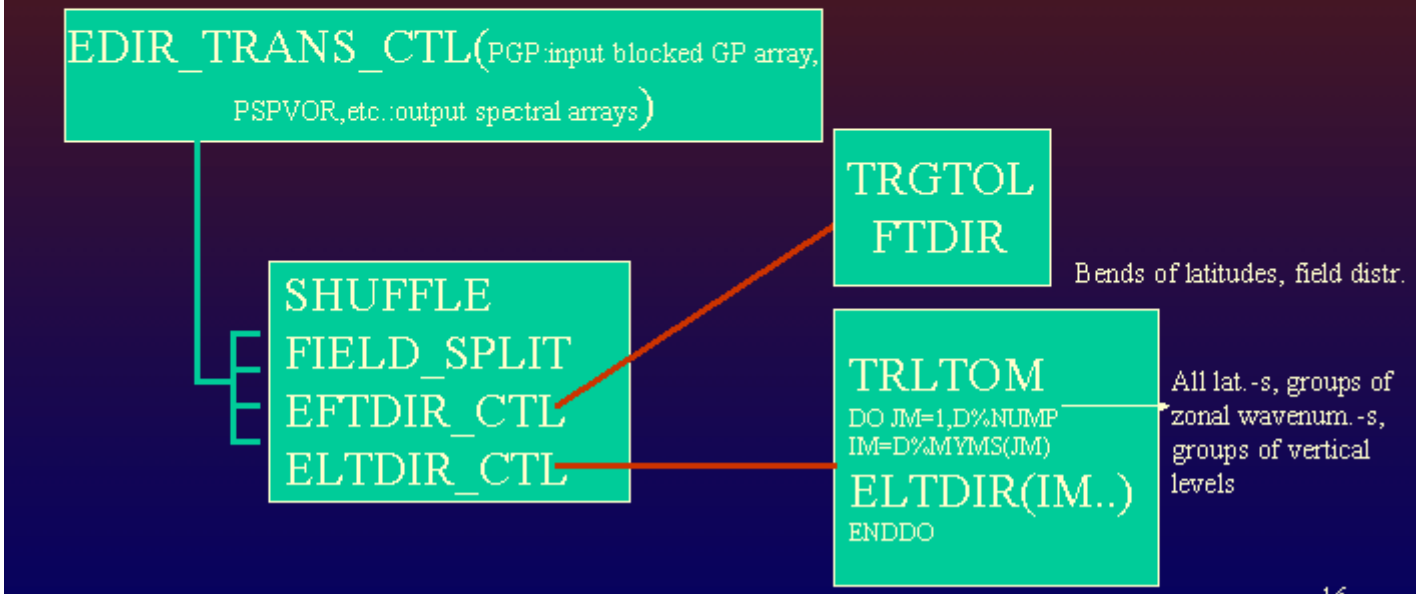
Number of longitudes: NONL(<*global latitude index*>, *mysetb*)

offset of myseta: ***nptrfloff***+*local lat. ind* : ***jgl***

Number of gridpoints: NGPTOT: Sum of NONL-s over jgl=1..ndgenl

NGPTOT_CAP : " ndguxl



Data structure in distributed environment

GP→FS→SS