

Esforço necessário para a mudança do código visando retirar os parâmetros INPES e JNPES do parmata.

Conteúdo do arquivo parmata:

```
C-----  
C*** SET PRIMARY GRID DIMENSIONS AND PRESSURE OUTPUT LEVELS  
PARAMETER ( IM=199 ,JM=249 ,LM=38 ,LSM=25 )  
C-----  
C*** SET THE NUMBER OF PEs IN THE I-DIRECTION AND J-DIRECTION  
PARAMETER( INPES=10 ,JNPES=50 )  
C-----  
C  
PARAMETER( IGSTL=-5 ,IGSTR=5 )  
PARAMETER( JGSTL=-5 ,JGSTR=5 )  
PARAMETER( ITAIL=IM-( INPES*( IM/INPES ) ) )  
PARAMETER( JTAIL=JM-( JNPES*( JM/JNPES ) ) )  
C  
PARAMETER( IDIM1=IGSTL ,IDIM2=IM/INPES+IGSTR+1 )  
PARAMETER( JDIM1=JGSTL ,JDIM2=JM/JNPES+JGSTR+1 )  
C  
C-----
```

A retirada dos parâmetros INPES e JNPES do arquivo parmata implicará numa mudança significativa do código devido, principalmente, a forma de implementação dos vetores de modo estático, ou seja, os vetores são criados já no tamanho que devem ser usados referente aos parâmetros informados. Uma forma para solucionar esse problema é criar módulos em FORTRAN 90 que sejam responsáveis pelo controle de alocação dos vetores de forma dinâmica. A cada número diferente nos parâmetros o código irá alocar um vetor com o tamanho requerido.

Foi constatado que esses vetores estão espalhados por todo o código, incluindo arquivos de parâmetros (*parameters*) e arquivos com blocos *commons*. Para poder entender a complexidade da mudança, foi criado uma seqüência de dependências dos parâmetros a partir do arquivo parmata.

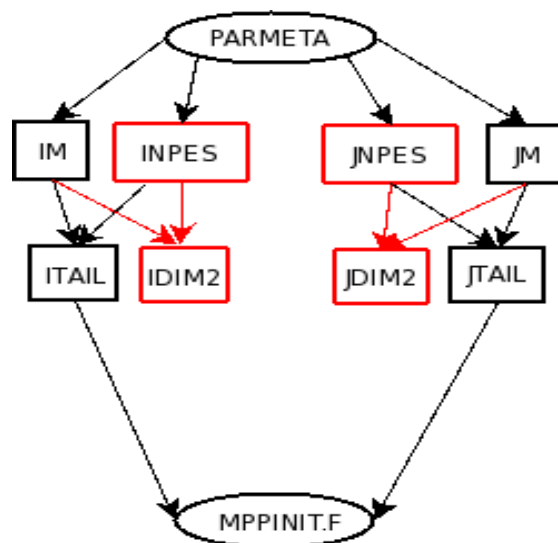


Figura 1: Parâmetros dependentes de INPES e JNPES no parmata

Na Figura 1, percebe-se que outros parâmetros são carregados com os dois iniciais informados. Os parâmetros ITAIL e JTAIL apenas são usados no arquivo MPPINIT.F. Já os indicados em vermelho, são referenciados por muitos arquivos e serão informados a parte.

A Tabela 1 mostra uma listagem realizada nos arquivos fontes que usam o parmata. Nesta listagem inicial ainda não foi considerado quais vetores dependem dos parâmetros iniciais.

Tabela 1: Lista dos arquivos que usam o parmata

<i>BOCOH.F:</i>	<i>EXIT.F:</i>	<i>MPPINIT.F:</i>	<i>TABLE.F:</i>
<i>BOCOHF.F:</i>	<i>FILT25.F:</i>	<i>NEWFLT.F:</i>	<i>TTBLEX.F:</i>
<i>BOCOV.F:</i>	<i>FST88.F:</i>	<i>O3INT.F:</i>	<i>TURBL.F:</i>
<i>CHKOUT.F:</i>	<i>GFDLRD.F:</i>	<i>OZON2D.F:</i>	<i>TWR.F:</i>
<i>CLO89.F:</i>	<i>GOSSIP.F:</i>	<i>PDNEW.F:</i>	<i>UPDATE.F:</i>
<i>CLTEND.F:</i>	<i>GRADFS.F:</i>	<i>PDTEDT.F:</i>	<i>VADZ.F:</i>
<i>COLLECT.F:</i>	<i>GSMCOLUMN.F:</i>	<i>PDTE.F:</i>	<i>VDIFH.F:</i>
<i>CONRAD.F:</i>	<i>GSMCONST.F:</i>	<i>PGCOR.F:</i>	<i>VDIFQ.F:</i>
<i>CUCNVC.F:</i>	<i>GSMDRIVE.F:</i>	<i>PRODQ2.F:</i>	<i>VDIFV.F:</i>
<i>DDAMP.F:</i>	<i>HADZ.F:</i>	<i>QUILT.F:</i>	<i>VTADVbeforepcwln</i>
<i>DIFCOF.F:</i>	<i>HDIFF.F:</i>	<i>RADFS.F:</i>	<i>Change.F:</i>
<i>DIGFLT.F:</i>	<i>HZADV2.F:</i>	<i>RADTN.F:</i>	<i>VTADV_cld.F:</i>
<i>DIST.F:</i>	<i>HZADV.F:</i>	<i>RDTEMP.F:</i>	<i>VTADV.F:</i>
<i>DIVHOA.F:</i>	<i>HZADV_LM1.F:</i>	<i>READ_NHB.F:</i>	<i>VTADV.F.F:</i>
<i>DIVHOAS.F:</i>	<i>HZADVS.F:</i>	<i>READ_RESTRT2.F:</i>	<i>VTADVpcwln.F:</i>
<i>DIVHOAST.F:</i>	<i>IDSTRB.F:</i>	<i>READ_RESTRT.F:</i>	<i>VWR.F:</i>
<i>DSTRB.F:</i>	<i>ILOC2GLB.F:</i>	<i>SFCDIF.F:</i>	<i>WRTCOM.F:</i>
<i>E1E290.F:</i>	<i>INIT.F:</i>	<i>SLADVT.F:</i>	<i>WRTRST.F:</i>
<i>E290.F:</i>	<i>INITS.F:</i>	<i>SLP.F:</i>	<i>ZENITH.F:</i>
<i>E2SPEC.F:</i>	<i>LOC2GLB.F:</i>	<i>SLPSIG.F:</i>	<i>ZERO2.F:</i>
<i>E3V88.F:</i>	<i>LWR88.F:</i>	<i>SLPSIGSPLINE.F:</i>	<i>ZERO3.F:</i>
<i>EBU.F:</i>	<i>MIXLEN.F:</i>	<i>SPA88.F:</i>	<i>ZERO3_T.F:</i>
<i>EBU_old.F:</i>	<i>module_EXCHM.F:</i>	<i>SURFCE.F:</i>	
<i>EPS.F:</i>	<i>MPI_FIRST.F:</i>	<i>SWR93.F:</i>	

Percebe-se que em grande parte dos fontes o arquivo parmata é usado.

Os parâmetros IDIM2 e JDIM2, mostrados, respectivamente, nas Tabelas 2 e 3 são os que mais influenciam na mudança do código, pois eles indicam a maioria das dimensões dos vetores usados. É bom lembrar que eles são carregados pelos parâmetros INPES e JNPES.

Tabela 2: Lista de arquivos que usam IDIM2

<i>CHKOUT.F:</i>	<i>FILT25.F:</i>	<i>LWR88.F:</i>	<i>TTBLEX.F:</i>
<i>CLO89.F:</i>	<i>FST88.F:</i>	<i>module_EXCHM.F:</i>	<i>TURBL.F:</i>
<i>CLTEND.F:</i>	<i>GSMCOLUMN.F:</i>	<i>OZON2D.F:</i>	<i>TWR.F:</i>
<i>CUCNVC.F:</i>	<i>GSMDRIVE.F:</i>	<i>PDTEDT.F:</i>	<i>VADZ.F:</i>
<i>DDAMP.F:</i>	<i>HADZ.F:</i>	<i>PDTE.F:</i>	<i>VTADVbeforepcwln</i>
<i>DIGFLT.F:</i>	<i>HDIFF.F:</i>	<i>PGCOR.F:</i>	<i>Change.F:</i>
<i>DIVHOA.F:</i>	<i>HZADV2.F:</i>	<i>RADFS.F:</i>	<i>VTADV_cld.F:</i>
<i>DIVHOAS.F:</i>	<i>HZADV.F:</i>	<i>RADTN.F:</i>	<i>VTADV.F:</i>
<i>DIVHOAST.F:</i>	<i>HZADV_LM1.F:</i>	<i>RDTEMP.F:</i>	<i>VTADV.F.F:</i>
<i>DSTRB.F:</i>	<i>HZADVS.F:</i>	<i>READ_RESTRT2.F:</i>	<i>VTADVpcwln.F:</i>
<i>E1E290.F:</i>	<i>IDSTRB.F:</i>	<i>READ_RESTRT.F:</i>	<i>ZERO2.F:</i>
<i>E290.F:</i>	<i>ILOC2GLB.F:</i>	<i>SLADVT.F:</i>	<i>ZERO3.F:</i>
<i>E2SPEC.F:</i>	<i>INIT.F:</i>	<i>SPA88.F:</i>	<i>ZERO3_T.F:</i>
<i>E3V88.F:</i>	<i>INITS.F:</i>	<i>SURFCE.F:</i>	
<i>EPS.F:</i>	<i>LOC2GLB.F:</i>	<i>SWR93.F:</i>	

Tabela 3: Lista de Arquivos que usam JDIM2

<i>CHKOUT.F:</i>	<i>HADZ.F:</i>	<i>PDTEDT.F:</i>	<i>VADZ.F:</i>
<i>CLTEND.F:</i>	<i>HDIFF.F:</i>	<i>PDTE.F:</i>	<i>VTADVbeforepcwln</i>
<i>CUCNVC.F:</i>	<i>HZADV2.F:</i>	<i>PGCOR.F:</i>	<i>Change.F:</i>
<i>DDAMP.F:</i>	<i>HZADV.F:</i>	<i>RADTN.F:</i>	<i>VTADV_cld.F:</i>
<i>DIGFLT.F:</i>	<i>HZADV_LM1.F:</i>	<i>RDTEMP.F:</i>	<i>VTADV.F:</i>
<i>DIVHOA.F:</i>	<i>HZADVS.F:</i>	<i>READ_RESTRT2.F:</i>	<i>VTADV.F.F:</i>
<i>DIVHOAS.F:</i>	<i>IDSTRB.F:</i>	<i>READ_RESTRT.F:</i>	<i>VTADVpcwln.F:</i>
<i>DIVHOAST.F:</i>	<i>ILOC2GLB.F:</i>	<i>SLADVT.F:</i>	<i>ZERO2.F:</i>
<i>DSTRB.F:</i>	<i>INIT.F:</i>	<i>SURFCE.F:</i>	<i>ZERO3.F:</i>
<i>EPS.F:</i>	<i>INITS.F:</i>	<i>TTBLEX.F:</i>	<i>ZERO3_T.F:</i>
<i>GSMCOLUMN.F:</i>	<i>LOC2GLB.F:</i>	<i>TURBL.F:</i>	
<i>GSMDRIVE.F:</i>	<i>module_EXCHM.F:</i>	<i>TWR.F:</i>	

Explorando os diversos arquivos, percebe-se que é recorrente a alocação de vetores dependentes de *IDIM1* e *IDIM2*. Em todos os casos eles são carregados de forma estática, o que sugere uma mudança significativa de todos os fontes envolvidos. Para melhor exemplificar, foi retirado um trecho de código do arquivo *TURBL.F*, Figura 2, referente a declaração de diversos vetores usados pela subrotina *TURBL*.

```

                                R E A L
& CKLQ( IDIM1:IDIM2,JDIM1:JDIM2 )
& ,CT ( IDIM1:IDIM2,JDIM1:JDIM2 )
& ,APE ( IDIM1:IDIM2,JDIM1:JDIM2,LM )
& ,AKH ( IDIM1:IDIM2,JDIM1:JDIM2,LM1 )
& ,AKM ( IDIM1:IDIM2,JDIM1:JDIM2,LM1 )
& ,ZINT( IDIM1:IDIM2,JDIM1:JDIM2,LP1 )
& ,UZ0H( IDIM1:IDIM2,JDIM1:JDIM2 )
& ,VZ0H( IDIM1:IDIM2,JDIM1:JDIM2 )
C
                                R E A L
& AKMCOL( IDIM1:IDIM2,JDIM1:JDIM2,LM1 )
& ,AKHCOL( IDIM1:IDIM2,JDIM1:JDIM2,LM1 )
& ,AKMSV ( IDIM1:IDIM2,JDIM1:JDIM2 )

```

```

&,ZCOL ( IDIM1:IDIM2,JDIM1:JDIM2,LP1 )
&,UCOL ( IDIM1:IDIM2,JDIM1:JDIM2,LM )
&,VCOL ( IDIM1:IDIM2,JDIM1:JDIM2,LM )
C
                                R E A L
&,AKH_T ( LM1, IDIM1:IDIM2,JDIM1:JDIM2 )
&,AKM_T ( LM1, IDIM1:IDIM2,JDIM1:JDIM2 )
&,APECOL_T ( LM, IDIM1:IDIM2,JDIM1:JDIM2 )
&,ZCOL_T ( LP1, IDIM1:IDIM2,JDIM1:JDIM2 )
&,ZCOL_T2 ( LP1, IDIM1:IDIM2,JDIM1:JDIM2 )
&,UCOL_T ( LM, IDIM1:IDIM2,JDIM1:JDIM2 )
&,VCOL_T ( LM, IDIM1:IDIM2,JDIM1:JDIM2 )
&,TCOL_T ( LM, IDIM1:IDIM2,JDIM1:JDIM2 )
&,QCOL_T ( LM, IDIM1:IDIM2,JDIM1:JDIM2 )
&,Q2COL_T ( LM, IDIM1:IDIM2,JDIM1:JDIM2 )
C

```

Figura 2: Trecho de código do arquivo TURBL.F: Declaração dos vetores

Também existe a necessidade de modificar os blocos common para módulos devido a alocação estática. Na Figura 3 é apresentado o arquivo PRFHLD.comm.

```

COMMON /PRFHLD/
& TLMIN( IDIM1:IDIM2,JDIM1:JDIM2 ), TLMAX( IDIM1:IDIM2,JDIM1:JDIM2 )

```

Figura 3: Arquivo PRFHLD.comm

Mesmo blocos pequenos devem ser alterados e, na Figura 4, é apresentada uma possível modificação do código original em um módulo (PRFHLD.f90) aplicando a alocação dinâmica de memória. Percebe-se que o trabalho relacionado a mudança também deve seguir uma padronização ainda não definida.

```

module PRFHLD

  implicit none
  public

  real, allocatable, dimension(:,:) :: TLMIN
  real, allocatable, dimension(:,:) :: TLMAX

  contains

  subroutine Alloc(idim1, idim2, jdim1, jdim2)
    integer, intent(IN) :: idim1
    integer, intent(IN) :: idim2
    integer, intent(IN) :: jdim1
    integer, intent(IN) :: jdim2

    integer :: ierr
    allocate ( TLMIN(idim1:idim2,jdim1:jdim2) , STAT=ierr )
    if (ierr .ne. 0) then
      write (*,*) "ERROR: PRFHLD.f90: allocate TLMIN!"
      write (*,*) "Execution ends"
      stop
    end if
    allocate ( TLMAX(idim1:idim2,jdim1:jdim2) , STAT=ierr )
    if (ierr .ne. 0) then
      write (*,*) "ERROR: PRFHLD.f90: allocate TLMAX!"
      write (*,*) "Execution ends"
      stop
    end if
  end subroutine Alloc

  subroutine Dealloc()

    integer :: ierr
    deallocate ( TLMIN, STAT=ierr )
    if (ierr .ne. 0) then
      write (*,*) "ERROR: M_PRFHLD.f90: deallocate TLMIN!"
      write (*,*) "Execution ends"
      stop
    end if
    deallocate ( TLMAX, STAT=ierr )
    if (ierr .ne. 0) then
      write (*,*) "ERROR: M_PRFHLD.f90: deallocate TLMAX!"
      write (*,*) "Execution ends"
      stop
    end if
  end subroutine Dealloc

end module PRFHLD

```

Figura 4: Módulo criado para substituição do arquivo PRFHLD.comm.

A seguir são apresentados, na tabelas 4, os arquivos fontes que deverão ser mudados para a inclusão desse novo paradigma APENAS para o arquivos do diretório *etafcst_all*:

Tabela 4: Arquivos que deverão modificados

Arquivos fontes (Total de 105):		
BOCOH.F	HDIFF.F	SFCDIF.F
BOCOHF.F	HZADV2.F	SFLX.F
BOCOV.F	HZADV.F	SGETMO.F
CHKOUT.F	HZADV_LM1.F	SGETMY.F
CLO89.F	HZADVS.F	SLADVT.F
CLTEND.F	IDSTRB.F	SLP.F
COLLECT.F	ILOC2GLB.F	SLPSIG.F
CONRAD.F	INIT.F	SLPSIGSPLINE.F
CUCNVC.F	INITS.F	SOLARD.F
DDAMP.F	LOC2GLB.F	SPA88.F
DIFCOF.F	LWR88.F	SURFCE.F
DIGFLT.F	MIXLEN.F	SWR93.F
DIST.F	module_EXCH_BUF_INTEGER.F	TABLE.F
DIVHOA.F	module_EXCH_BUF_REAL.F	timef_bog.F
DIVHOAS.F	module_EXCHM.F	TTBLEX.F
DIVHOAST.F	MPI_FIRST.F	TURBL.F
DSTRB.F	MPI_LAST.F	TWR.F
E1E290.F	MPPINIT.F	UPDATE.F
E290.F	NEWFLT.F	VADZ.F
E2SPEC.F	O3CLIM.F	VDIFH.F
E3V88.F	O3INT.F	VDIFQ.F
EBU.F	OZON2D.F	VDIFV.F
EBU_old.F	PARA_RANGE.F	VTADVbeforepcwlnChange.F
EPS.F	PDNEW.F	VTADV_cld.F
EXIT.F	PDTEDT.F	VTADV.F
FILT25.F	PDTE.F	VTADV.F
FST88.F	PGCOR.F	VTADVpcwln.F
GFDLRD.F	PRODQ2.F	VWR.F
GOSSIP.F	QUILT.F	WRTCOM.F
GRADFS.F	RADFS.F	WRTRST.F
GSMCOLUMN.F	RADTN.F	ZENITH.F
GSMCONST.F	RDTEMP.F	ZERO2.F
GSMDRIVE.F	READ_NHB.F	ZERO3.F
HADZ.F	READ_RESTRT2.F	ZERO3_T.F
HCONST.F	READ_RESTRT.F	
	SETUP_SERVERS.F	
Blocos Commons (total de 42):		
ACMCLD.comm	DYNAM.comm	PHYS.comm
ACMCLH.comm	DYNAMD.comm	PPTASM.comm
ACMPRE.comm	HCON.comm	PRFHLD.comm
ACMRDL.comm	INDX.comm	PVRBLS.comm
ACMRDS.comm	LOOPS.comm	RNDDTA.comm
ACMSFC.comm	MAPOT1.comm	SEASO3.comm
BOCO.comm	MAPOT.comm	SLOPES.comm
BUFFER.comm	MASKS.comm	SOIL.comm
C_FRACN.comm	MSLP.comm	TABCOM.comm
CLDWTR.comm	NHYDRO.comm	TEMPV.comm
CNVCLD.comm	OPTIONS.comm	VRBLS.comm
CO2DTA.comm	OUTFIL.comm	ZOEFFT.comm
CONTIN.comm	PARA.comm	
C_Tadj.comm	PHYS1.comm	
CTLBLK.comm	PHYS2.comm	

Esse passo é importante para a melhoria do modelo, mas o foco do projeto está na melhoria do desempenho em um ambiente massivamente paralelo, o que nos obriga, a princípio, deixar essa mudança como meta futura.