

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1

```
0001      !Matrix utility routines.
0002      !Generally (but not always)
0003      !Not complete
0004      !Antony Lewis May 2003-20
0005      !http://cosmologist.info/
0006
0007
0008      module MatrixUtils
0009      use AMLutils
0010      implicit none
0011
0012      logical, parameter :: Mat
0013      #ifdef MATRIX_SINGLE
0014      integer, parameter :: dm
0015      #else
0016      integer, parameter :: dm
0017      #endif
0018      !Precision of matrix operations
0019      !If changing also need to
0020      integer, parameter :: Mat
0021      integer, parameter :: mat
0022
0023      real Matrix_StartTime
0024
0025      Type TMatrixType
0026      real(dm), dimension(:)
0027      end Type TMatrixType
0028
0029      complex(dm), parameter ::
0030      real(dm), parameter :: RO
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1

```
0001      !Matrix utility routines.
0002      !Generally (but not always)
0003      !Not complete
0004      !Antony Lewis May 2003-20
0005      !http://cosmologist.info/
0006
0007
0008      module MatrixUtils
0009      use AMLutils
0010      implicit none
0011
0012      logical, parameter :: Mat
0013      #ifdef MATRIX_SINGLE
0014      integer, parameter :: dm
0015      #else
0016      integer, parameter :: dm
0017      #endif
0018      !Precision of matrix operations
0019      !If changing also need to
0020      integer, parameter :: Mat
0021      integer, parameter :: mat
0022
0023      real Matrix_StartTime
0024
0025      Type TMatrixType
0026      real(dm), dimension(:)
0027      end Type TMatrixType
0028
0029      complex(dm), parameter ::
0030      real(dm), parameter :: RO
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 31

```
0031      real, parameter :: SOne =
0032
0033      contains
0034
0035
0036      function GetMatrixTime()
0037      real GetMatrixTime
0038      real atime
0039
0040      call cpu_time(atime)
0041
0042      GetMatrixTime = atime
0043
0044
0045      end function GetMatrixTim
0046
0047      subroutine Matrix_start(
0048      character(LEN=*), intent(
0049
0050      if (Matrix_runmsgs) then
0051          Matrix_StartTime = Ge
0052          Write(*,*) 'Matrix_' /
0053      end if
0054      end subroutine Matrix_st
0055
0056      subroutine Matrix_end(Na
0057      character(LEN=*), intent(
0058
0059      if (Matrix_runmsgs) then
0060          Write(*,*) 'Matrix_' /
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 31

```
0031      real, parameter :: SOne =
0032
0033      contains
0034
0035
0036      function GetMatrixTime()
0037      real GetMatrixTime
0038      real atime
0039
0040      call cpu_time(atime)
0041
0042      GetMatrixTime = atime
0043
0044
0045      end function GetMatrixTim
0046
0047      subroutine Matrix_start(
0048      character(LEN=*), intent(
0049
0050      if (Matrix_runmsgs) then
0051          Matrix_StartTime = Ge
0052          Write(*,*) 'Matrix_' /
0053      end if
0054      end subroutine Matrix_st
0055
0056      subroutine Matrix_end(Na
0057      character(LEN=*), intent(
0058
0059      if (Matrix_runmsgs) then
0060          Write(*,*) 'Matrix_' /
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 61

```
0061      end if
0062      end subroutine Matrix_en
0063
0064      subroutine Matrix_WriteFi
0065      integer, intent(in) :: au
0066      integer, intent(in) :: n
0067      real(dm) :: vec(n)
0068      character(LEN=50) fmt
0069
0070      fmt = trim(numcat('(',n))
0071      write (aunit, fmt) vec(1:
0072
0073      end subroutine Matrix_Wri
0074
0075      subroutine Matrix_Write(a
0076      character(LEN=*), intent(
0077      character(LEN=*), intent(
0078      real(dm), intent(in) :: m
0079      logical, intent(in), opti
0080      integer i,k
0081      character(LEN=50) fmt
0082      integer shp(2)
0083      logical WriteTab
0084      integer file_unit
0085
0086      shp = shape(mat)
0087      WriteTab = shp(2)<=50
0088      if (present(forcetable))
0089          if (forcetable) Write
0090      end if
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 61

```
0061      end if
0062      end subroutine Matrix_en
0063
0064      subroutine Matrix_WriteFi
0065      integer, intent(in) :: au
0066      integer, intent(in) :: n
0067      real(dm) :: vec(n)
0068      character(LEN=50) fmt
0069
0070      fmt = trim(numcat('(',n))
0071      write (aunit, fmt) vec(1:
0072
0073      end subroutine Matrix_Wri
0074
0075      subroutine Matrix_Write(a
0076      character(LEN=*), intent(
0077      character(LEN=*), intent(
0078      real(dm), intent(in) :: m
0079      logical, intent(in), opti
0080      integer i,k
0081      character(LEN=50) fmt
0082      integer shp(2)
0083      logical WriteTab
0084      integer file_unit
0085
0086      shp = shape(mat)
0087      WriteTab = shp(2)<=50
0088      if (present(forcetable))
0089          if (forcetable) Write
0090      end if
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 91

```
0091      file_unit = new_file_unit
0092      call CreateTxtFile(aname,
0093      if (present(commentline))
0094          write(file_unit,'(a)'
0095      end if
0096      fmt = trim(numcat('(',shp
0097      do i=1, shp(1)
0098          if (.not. WriteTab) t
0099              do k=1, shp(2)
0100                  write (file_u
0101              end do
0102          else
0103              write (file_unit,
0104          end if
0105      end do
0106
0107      call CloseFile(file_unit)
0108
0109      end subroutine Matrix_Wri
0110
0111      subroutine Matrix_Write_d
0112      character(LEN=*), intent(
0113      double precision, intent(
0114      logical, intent(in), opti
0115      integer i,k
0116      character(LEN=50) fmt
0117      integer shp(2)
0118      logical WriteTab
0119      integer file_unit
0120
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 91

```
0091      file_unit = new_file_unit
0092      call CreateTxtFile(aname,
0093      if (present(commentline))
0094          write(file_unit,'(a)'
0095      end if
0096      fmt = trim(numcat('(',shp
0097      do i=1, shp(1)
0098          if (.not. WriteTab) t
0099              do k=1, shp(2)
0100                  write (file_u
0101              end do
0102          else
0103              write (file_unit,
0104          end if
0105      end do
0106
0107      call CloseFile(file_unit)
0108
0109      end subroutine Matrix_Wri
0110
0111      subroutine Matrix_Write_d
0112      character(LEN=*), intent(
0113      double precision, intent(
0114      logical, intent(in), opti
0115      integer i,k
0116      character(LEN=50) fmt
0117      integer shp(2)
0118      logical WriteTab
0119      integer file_unit
0120
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 121

```
0121      shp = shape(mat)
0122      WriteTab = shp(2)<=50
0123      if (present(forcetable))
0124          if (forcetable) Write
0125      end if
0126      file_unit = new_file_unit
0127      call CreateTxtFile(aname,
0128      fmt = trim(numcat('(',shp
0129      do i=1, shp(1)
0130          if (.not. WriteTab) t
0131              do k=1, shp(2)
0132                  write (file_u
0133              end do
0134          else
0135              write (file_unit,
0136          end if
0137      end do
0138
0139      call CloseFile(file_unit)
0140
0141      end subroutine Matrix_Wri
0142
0143
0144      subroutine Matrix_Write_B
0145      character(LEN=*), intent(
0146      real(dm), intent(in) :: m
0147      integer file_unit
0148
0149      file_unit = new_file_unit
0150      call CreateFile(aname, fi
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 121

```
0121      shp = shape(mat)
0122      WriteTab = shp(2)<=50
0123      if (present(forcetable))
0124          if (forcetable) Write
0125      end if
0126      file_unit = new_file_unit
0127      call CreateTxtFile(aname,
0128      fmt = trim(numcat('(',shp
0129      do i=1, shp(1)
0130          if (.not. WriteTab) t
0131              do k=1, shp(2)
0132                  write (file_u
0133              end do
0134          else
0135              write (file_unit,
0136          end if
0137      end do
0138
0139      call CloseFile(file_unit)
0140
0141      end subroutine Matrix_Wri
0142
0143
0144      subroutine Matrix_Write_B
0145      character(LEN=*), intent(
0146      real(dm), intent(in) :: m
0147      integer file_unit
0148
0149      file_unit = new_file_unit
0150      call CreateFile(aname, fi
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 151

```
0151      write (file_unit) mat
0152      call CloseFile(file_unit)
0153
0154      end subroutine Matrix_Wri
0155
0156
0157      subroutine MatrixSym_Writ
0158      character(LEN=*), intent(
0159      real(dm), intent(in) :: m
0160      integer i
0161      integer shp(2)
0162      integer file_unit
0163
0164      shp = shape(mat)
0165      if (shp(1) /= shp(2)) cal
0166      if (shp(1) == 0) return
0167
0168      file_unit = new_file_unit
0169      call CreateFile(aname, fi
0170      do i=1,shp(1)
0171          write (file_unit) mat
0172      end do
0173      call CloseFile(file_unit)
0174
0175      end subroutine MatrixSym_
0176
0177      subroutine MatrixSym_Writ
0178      character(LEN=*), intent(
0179      real(dm), intent(in) :: m
0180      integer i,      file_unit
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 151

```
0151      write (file_unit) mat
0152      call CloseFile(file_unit)
0153
0154      end subroutine Matrix_Wri
0155
0156
0157      subroutine MatrixSym_Writ
0158      character(LEN=*), intent(
0159      real(dm), intent(in) :: m
0160      integer i
0161      integer shp(2)
0162      integer file_unit
0163
0164      shp = shape(mat)
0165      if (shp(1) /= shp(2)) cal
0166      if (shp(1) == 0) return
0167
0168      file_unit = new_file_unit
0169      call CreateFile(aname, fi
0170      do i=1,shp(1)
0171          write (file_unit) mat
0172      end do
0173      call CloseFile(file_unit)
0174
0175      end subroutine MatrixSym_
0176
0177      subroutine MatrixSym_Writ
0178      character(LEN=*), intent(
0179      real(dm), intent(in) :: m
0180      integer i,      file_unit
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 181

```
0181      integer shp(2)
0182
0183      shp = shape(mat)
0184      if (shp(1) /= shp(2)) call
0185      if (shp(1) == 0) return
0186
0187      file_unit = new_file_unit
0188      call CreateFile(aname, fi
0189      do i=1,shp(1)
0190          write (file_unit) rea
0191      end do
0192      call CloseFile(file_unit)
0193
0194      end subroutine MatrixSym_
0195
0196
0197
0198      subroutine Matrix_WriteVe
0199      character(LEN=*), intent(
0200      real(dm), intent(in) :: v
0201      integer i, file_unit
0202
0203      file_unit = new_file_unit
0204      call CreateTxtFile(aname,
0205      do i=1, size(vec)
0206          write (file_unit, '(1
0207      end do
0208      call CloseFile(file_unit)
0209
0210      end subroutine Matrix_Wri
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 181

```
0181      integer shp(2)
0182
0183      shp = shape(mat)
0184      if (shp(1) /= shp(2)) call
0185      if (shp(1) == 0) return
0186
0187      file_unit = new_file_unit
0188      call CreateFile(aname, fi
0189      do i=1,shp(1)
0190          write (file_unit) rea
0191      end do
0192      call CloseFile(file_unit)
0193
0194      end subroutine MatrixSym_
0195
0196
0197
0198      subroutine Matrix_WriteVe
0199      character(LEN=*), intent(
0200      real(dm), intent(in) :: v
0201      integer i, file_unit
0202
0203      file_unit = new_file_unit
0204      call CreateTxtFile(aname,
0205      do i=1, size(vec)
0206          write (file_unit, '(1
0207      end do
0208      call CloseFile(file_unit)
0209
0210      end subroutine Matrix_Wri
```

```
0211
0212
0213      subroutine Matrix_Read_Bi
0214      character(LEN=*), intent(
0215      real(dm), intent(out) ::
0216      integer file_unit
0217
0218      file_unit = new_file_unit
0219      call OpenFile(aname, file
0220      read (file_unit) mat
0221      call CloseFile(file_unit)
0222
0223      end subroutine Matrix_Rea
0224
0225
0226      subroutine MatrixSym_Read
0227      character(LEN=*), intent(
0228      real(dm), intent(out) ::
0229      integer i,      file_unit
0230      integer shp(2)
0231
0232      shp = shape(mat)
0233      if (shp(1) /= shp(2)) cal
0234      if (shp(1) == 0) return
0235
0236      file_unit = new_file_unit
0237      call OpenFile(aname, file
0238      do i=1,shp(1)
0239          read (file_unit) mat(
0240          mat(i,i:shp(1)) = mat
```

```
0211
0212
0213      subroutine Matrix_Read_Bi
0214      character(LEN=*), intent(
0215      real(dm), intent(out) ::
0216      integer file_unit
0217
0218      file_unit = new_file_unit
0219      call OpenFile(aname, file
0220      read (file_unit) mat
0221      call CloseFile(file_unit)
0222
0223      end subroutine Matrix_Rea
0224
0225
0226      subroutine MatrixSym_Read
0227      character(LEN=*), intent(
0228      real(dm), intent(out) ::
0229      integer i,      file_unit
0230      integer shp(2)
0231
0232      shp = shape(mat)
0233      if (shp(1) /= shp(2)) cal
0234      if (shp(1) == 0) return
0235
0236      file_unit = new_file_unit
0237      call OpenFile(aname, file
0238      do i=1,shp(1)
0239          read (file_unit) mat(
0240          mat(i,i:shp(1)) = mat
```



```
0241      end do
0242      call CloseFile(file_unit)
0243
0244      end subroutine MatrixSym_
0245
0246      subroutine MatrixSym_Read
0247      character(LEN=*), intent(
0248      real, intent(out) :: mat(
0249      integer i,      file_unit
0250      integer shp(2)
0251
0252      shp = shape(mat)
0253      if (shp(1) /= shp(2)) cal
0254      if (shp(1) == 0) return
0255
0256      file_unit = new_file_unit
0257      call OpenFile(aname, file
0258      do i=1,shp(1)
0259          read (file_unit) mat(
0260          mat(i,i:shp(1)) = mat
0261      end do
0262      call CloseFile(file_unit)
0263
0264      end subroutine MatrixSym_
0265
0266
0267
0268
0269
0270      subroutine Matrix_Read(an
```

```
0241      end do
0242      call CloseFile(file_unit)
0243
0244      end subroutine MatrixSym_
0245
0246      subroutine MatrixSym_Read
0247      character(LEN=*), intent(
0248      real, intent(out) :: mat(
0249      integer i,      file_unit
0250      integer shp(2)
0251
0252      shp = shape(mat)
0253      if (shp(1) /= shp(2)) cal
0254      if (shp(1) == 0) return
0255
0256      file_unit = new_file_unit
0257      call OpenFile(aname, file
0258      do i=1,shp(1)
0259          read (file_unit) mat(
0260          mat(i,i:shp(1)) = mat
0261      end do
0262      call CloseFile(file_unit)
0263
0264      end subroutine MatrixSym_
0265
0266
0267
0268
0269
0270      subroutine Matrix_Read(an
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 271

```
0271      character(LEN=*), intent(  
0272      real(dm), intent(out) ::  
0273      integer j,k,      file_unit  
0274      integer shp(2)  
0275      real(dm) tmp  
0276  
0277      shp = shape(mat)  
0278  
0279      file_unit = new_file_unit  
0280      call OpenTxtFile(aname, f  
0281  
0282      do j=1,shp(1)  
0283          read (file_unit,*, en  
0284      end do  
0285      goto 120  
0286  
0287      100 rewind(file_unit)  !Try o  
0288      do j=1,shp(1)  
0289          do k=1,shp(2)  
0290              read (file_unit,*  
0291          end do  
0292      end do  
0293  
0294      120 read (file_unit,*, err =  
0295      goto 200  
0296  
0297      150 call CloseFile(file_unit)  
0298      return  
0299  
0300      200 call MpiStop('Matrix_Read
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 271

```
0271      character(LEN=*), intent(  
0272      real(dm), intent(out) ::  
0273      integer j,k,      file_unit  
0274      integer shp(2)  
0275      real(dm) tmp  
0276  
0277      shp = shape(mat)  
0278  
0279      file_unit = new_file_unit  
0280      call OpenTxtFile(aname, f  
0281  
0282      do j=1,shp(1)  
0283          read (file_unit,*, en  
0284      end do  
0285      goto 120  
0286  
0287      100 rewind(file_unit)  !Try o  
0288      do j=1,shp(1)  
0289          do k=1,shp(2)  
0290              read (file_unit,*  
0291          end do  
0292      end do  
0293  
0294      120 read (file_unit,*, err =  
0295      goto 200  
0296  
0297      150 call CloseFile(file_unit)  
0298      return  
0299  
0300      200 call MpiStop('Matrix_Read
```

```
0301
0302
0303      end subroutine Matrix_Rea
0304
0305      subroutine Matrix_ReadSin
0306      character(LEN=*), intent(
0307      real, intent(out) :: mat(
0308      integer j,k,      file_unit
0309      integer shp(2)
0310      real tmp
0311
0312      shp = shape(mat)
0313
0314      file_unit = new_file_unit
0315      call OpenTxtFile(aname, f
0316
0317      do j=1,shp(1)
0318          read (file_unit,*, en
0319      end do
0320      goto 120
0321
0322      100 rewind(file_unit)  !Try o
0323      do j=1,shp(1)
0324          do k=1,shp(2)
0325              read (file_unit,*
0326          end do
0327      end do
0328
0329      120 read (file_unit,*, err =
0330      goto 200
```

```
0301
0302
0303      end subroutine Matrix_Rea
0304
0305      subroutine Matrix_ReadSin
0306      character(LEN=*), intent(
0307      real, intent(out) :: mat(
0308      integer j,k,      file_unit
0309      integer shp(2)
0310      real tmp
0311
0312      shp = shape(mat)
0313
0314      file_unit = new_file_unit
0315      call OpenTxtFile(aname, f
0316
0317      do j=1,shp(1)
0318          read (file_unit,*, en
0319      end do
0320      goto 120
0321
0322      100 rewind(file_unit)  !Try o
0323      do j=1,shp(1)
0324          do k=1,shp(2)
0325              read (file_unit,*
0326          end do
0327      end do
0328
0329      120 read (file_unit,*, err =
0330      goto 200
```

```
0331
0332      150 call CloseFile(file_unit)
0333           return
0334
0335      200 call MpiStop('Matrix_Read
0336
0337
0338           end subroutine Matrix_Rea
0339
0340
0341           function Matrix_Diag(M, n
0342 integer, intent(in) :: n
0343 real(dm), intent(in) :: M
0344 real(dm) Matrix_Diag(n)
0345 integer i
0346
0347       do i=1,n
0348
0349 Matrix_Diag(i) = M(i,i)
0350
0351       end do
0352
0353       end function Matrix_Diag
0354
0355       function ILAENV_wrap(i,S1
0356 integer ILAENV_wrap
0357 integer, intent(in) :: i,
0358 character(LEN=*), intent(
0359 integer, external :: ILAE
0360
```

```
0331
0332      150 call CloseFile(file_unit)
0333           return
0334
0335      200 call MpiStop('Matrix_Read
0336
0337
0338           end subroutine Matrix_Rea
0339
0340
0341           function Matrix_Diag(M, n
0342 integer, intent(in) :: n
0343 real(dm), intent(in) :: M
0344 real(dm) Matrix_Diag(n)
0345 integer i
0346
0347       do i=1,n
0348
0349 Matrix_Diag(i) = M(i,
0350
0351       end do
0352
0353       end function Matrix_Diag
0354
0355       function ILAENV_wrap(i,S1
0356 integer ILAENV_wrap
0357 integer, intent(in) :: i,
0358 character(LEN=*), intent(
0359 integer, external :: ILAE
0360
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 361

```
0361      !If you don't have ILAENV
0362      !that is a guess at the b
0363      #ifdef MATRIX_SINGLE
0364          ILAENV_wrap = 16
0365      #else
0366          ILAENV_wrap = ILAENV(i,S
0367      #endif
0368      !!!IFC
0369      end function ILAENV_wrap
0370
0371
0372      subroutine Matrix_Diagona
0373      !Does  $m = U \text{ diag } U^T$ , ret
0374      integer, intent(in) :: n
0375      real(dm), intent(inout)::
0376      real(dm), intent(out) ::
0377      integer ierr, tmpsize
0378      real(dm), allocatable, di
0379
0380      call Matrix_Start('Diagon
0381      #ifdef MATRIX_SINGLE
0382          tmpsize = max( (ILAENV_w
0383          allocate(tmp(tmpsize));
0384          call SSYEV('V','U',n,m,n,
0385      #else
0386          tmpsize = max( (ILAENV_w
0387          allocate(tmp(tmpsize));
0388          call DSYEV('V','U',n,m,n,
0389      #endif
0390      if (ierr /= 0) call MpiSt
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 361

```
0361      !If you don't have ILAENV
0362      !that is a guess at the b
0363      #ifdef MATRIX_SINGLE
0364          ILAENV_wrap = 16
0365      #else
0366          ILAENV_wrap = ILAENV(i,S
0367      #endif
0368      !!!IFC
0369      end function ILAENV_wrap
0370
0371
0372      subroutine Matrix_Diagona
0373      !Does  $m = U \text{ diag } U^T$ , ret
0374      integer, intent(in) :: n
0375      real(dm), intent(inout)::
0376      real(dm), intent(out) ::
0377      integer ierr, tmpsize
0378      real(dm), allocatable, di
0379
0380      call Matrix_Start('Diagon
0381      #ifdef MATRIX_SINGLE
0382          tmpsize = max( (ILAENV_w
0383          allocate(tmp(tmpsize));
0384          call SSYEV('V','U',n,m,n,
0385      #else
0386          tmpsize = max( (ILAENV_w
0387          allocate(tmp(tmpsize));
0388          call DSYEV('V','U',n,m,n,
0389      #endif
0390      if (ierr /= 0) call MpiSt
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 391

```
0391      deallocate(tmp)
0392      call Matrix_End('Diagonal
0393
0394      end subroutine Matrix_Dia
0395
0396      subroutine Matrix_Diagona
0397      !Complex version. Does m
0398      integer, intent(in) :: n
0399      real(dm), intent(inout)::
0400      real(dm), intent(out) ::
0401      integer ierr, tmpsize ,is
0402      real(dm), allocatable, di
0403      integer, allocatable,dime
0404
0405      call Matrix_Start('Diagon
0406
0407      if (matrix_method == Mat_
0408          !Divide and conquer
0409      tmpsize = 1 + 6*N + 2
0410      isize = 3+5*N
0411      allocate(tmp(tmpsize))
0412      allocate(iwork( isize))
0413      #ifdef MATRIX_SINGLE
0414          call SSYEVD('V','U',n
0415      #else
0416          call DSYEVD('V','U',n
0417      #endif
0418          deallocate(iwork)
0419          deallocate(tmp)
0420      else
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 391

```
0391      deallocate(tmp)
0392      call Matrix_End('Diagonal
0393
0394      end subroutine Matrix_Dia
0395
0396      subroutine Matrix_Diagona
0397      !Complex version. Does m
0398      integer, intent(in) :: n
0399      real(dm), intent(inout)::
0400      real(dm), intent(out) ::
0401      integer ierr, tmpsize ,is
0402      real(dm), allocatable, di
0403      integer, allocatable,dime
0404
0405      call Matrix_Start('Diagon
0406
0407      if (matrix_method == Mat_
0408          !Divide and conquer
0409      tmpsize = 1 + 6*N + 2
0410      isize = 3+5*N
0411      allocate(tmp(tmpsize))
0412      allocate(iwork( isize))
0413      #ifdef MATRIX_SINGLE
0414          call SSYEVD('V','U',n
0415      #else
0416          call DSYEVD('V','U',n
0417      #endif
0418          deallocate(iwork)
0419          deallocate(tmp)
0420      else
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 421

```
0421      call Matrix_Diagonalize(M, diag, pow)
0422  end if
0423
0424  if (ierr /= 0) call MpiStop
0425
0426  call Matrix_End('Diagonalize')
0427
0428  end subroutine Matrix_Diagonalize
0429
0430
0431
0432  subroutine Matrix_Root(M,
0433    !Does M**pow for symmetrization
0434    !Not optimized for large n
0435    integer, intent(in) :: n
0436    real(dm), intent(inout) :: M
0437    real(dm) :: Tmp(n,n)
0438    real(dm), intent(in) :: pow
0439
0440    real(dm) :: diag(n)
0441    integer i
0442
0443    call Matrix_Diagonalize(M, diag, pow)
0444    Tmp = M
0445    diag = diag**pow
0446    do i = 1, n
0447      M(:,i) = M(:,i)*diag(i)
0448    end do
0449    M = matmul(M, transpose(Tmp))
0450
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 421

```
0421      call Matrix_Diagonalize(M, diag, pow)
0422  end if
0423
0424  if (ierr /= 0) call MpiStop
0425
0426  call Matrix_End('Diagonalize')
0427
0428  end subroutine Matrix_Diagonalize
0429
0430
0431
0432  subroutine Matrix_Root(M,
0433    !Does M**pow for symmetrization
0434    !Not optimized for large n
0435    integer, intent(in) :: n
0436    real(dm), intent(inout) :: M
0437    real(dm) :: Tmp(n,n)
0438    real(dm), intent(in) :: pow
0439
0440    real(dm) :: diag(n)
0441    integer i
0442
0443    call Matrix_Diagonalize(M, diag, pow)
0444    Tmp = M
0445    diag = diag**pow
0446    do i = 1, n
0447      M(:,i) = M(:,i)*diag(i)
0448    end do
0449    M = matmul(M, transpose(Tmp))
0450
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 451

```
0451      end subroutine Matrix_Roo
0452
0453
0454      subroutine Matrix_Diagona
0455      !Real version. Does m = U
0456      !Assumes up to nfound val
0457      integer, intent(in) :: n
0458      real(dm), intent(inout)::
0459      real(dm), intent(out) ::
0460      real(dm), intent(in) :: e
0461      integer, intent(inout) ::
0462      integer ierr, worksize, L
0463      real(dm), allocatable, di
0464      real(dm), allocatable, di
0465      integer, allocatable,dime
0466      real(dm) wsize(1)
0467      real(dm)  atol
0468      integer ISize(1)
0469
0470      atol = 1d-9
0471      call Matrix_Start('Matrix
0472      allocate(tmp(n,nfound))
0473      allocate(Supp(n))
0474      !Query
0475      WorkSize = -1
0476      LIWork = -1
0477      #ifdef MATRIX_SINGLE
0478      call SSYEVR('V','V','U',n
0479      Supp,WSize,WorkSize,ISize
0480      #else
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 451

```
0451      end subroutine Matrix_Roo
0452
0453
0454      subroutine Matrix_Diagona
0455      !Real version. Does m = U
0456      !Assumes up to nfound val
0457      integer, intent(in) :: n
0458      real(dm), intent(inout)::
0459      real(dm), intent(out) ::
0460      real(dm), intent(in) :: e
0461      integer, intent(inout) ::
0462      integer ierr, worksize, L
0463      real(dm), allocatable, di
0464      real(dm), allocatable, di
0465      integer, allocatable,dime
0466      real(dm) wsize(1)
0467      real(dm)  atol
0468      integer ISize(1)
0469
0470      atol = 1d-9
0471      call Matrix_Start('Matrix
0472      allocate(tmp(n,nfound))
0473      allocate(Supp(n))
0474      !Query
0475      WorkSize = -1
0476      LIWork = -1
0477      #ifdef MATRIX_SINGLE
0478      call SSYEVR('V','V','U',n
0479      Supp,WSize,WorkSize,I
0480      #else
```


/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 481

```
0481      call DSYEVR('V','V','U',n
0482      Supp, WSize, WorkSize, ISize
0483      #endif
0484      WorkSize = Real(WSize(1))
0485      LIWork = ISize(1)
0486      allocate(Work(WorkSize), I
0487      #ifdef MATRIX_SINGLE
0488      call SSYEVR('V','V','U',n
0489      Supp, Work, WorkSize, IWork,
0490      #else
0491      call DSYEVR('V','V','U',n
0492      Supp, Work, WorkSize, IWork,
0493      #endif
0494      deallocate(Supp, Work, IWork)
0495      if (ierr /= 0) call MpiSt
0496      M(1:n, 1:nfound) = tmp(1:n
0497      deallocate(tmp)
0498      call Matrix_End('Matrix_D
0499
0500      end subroutine Matrix_Dia
0501
0502
0503      subroutine Matrix_CDiagon
0504      !Complex version. Does m
0505      !Assumes up to nfound val
0506      integer, intent(in) :: n
0507      complex(dm), intent(inout)
0508      real(dm), intent(out) ::
0509      real(dm), intent(in) :: e
0510      integer, intent(inout) ::
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 481

```
0481      call DSYEVR('V','V','U',n
0482      Supp, WSize, WorkSize, I
0483      #endif
0484      WorkSize = Real(WSize(1))
0485      LIWork = ISize(1)
0486      allocate(Work(WorkSize), I
0487      #ifdef MATRIX_SINGLE
0488      call SSYEVR('V','V','U',n
0489      Supp, Work, WorkSize, IWork
0490      #else
0491      call DSYEVR('V','V','U',n
0492      Supp, Work, WorkSize, IWork
0493      #endif
0494      deallocate(Supp, Work, IWork)
0495      if (ierr /= 0) call MpiSt
0496      M(1:n, 1:nfound) = tmp(1:n
0497      deallocate(tmp)
0498      call Matrix_End('Matrix_D
0499
0500      end subroutine Matrix_Dia
0501
0502
0503      subroutine Matrix_CDiagon
0504      !Complex version. Does m
0505      !Assumes up to nfound val
0506      integer, intent(in) :: n
0507      complex(dm), intent(inout)
0508      real(dm), intent(out) ::
0509      real(dm), intent(in) :: e
0510      integer, intent(inout) ::
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 511

```
0511      integer ierr, worksize, L
0512      real(dm), allocatable, di
0513      complex(dm), allocatable,
0514      complex(dm), allocatable,
0515      integer, allocatable, dime
0516      complex(dm) wsize(1)
0517      real(dm) Rsize(1), atol
0518      integer ISize(1)
0519
0520      atol = 1d-9
0521      call Matrix_Start('Matrix
0522      allocate(tmp(n,nfound))
0523      allocate(Supp(n))
0524      !Query
0525      WorkSize = -1
0526      LRWork = -1
0527      LIWork = -1
0528      #ifdef MATRIX_SINGLE
0529          call CHEEVR('V','V','U',n
0530          Supp,WSize,WorkSize,RSize
0531      #else
0532          call ZHEEVR('V','V','U',n
0533          Supp,WSize,WorkSize,RSize
0534      #endif
0535      WorkSize = Real(WSize(1))
0536      LRWork = RSize(1)
0537      LIWork = ISize(1)
0538      allocate(Work(WorkSize),R
0539      #ifdef MATRIX_SINGLE
0540      call CHEEVR('V','V','U',n
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 511

```
0511      integer ierr, worksize, L
0512      real(dm), allocatable, di
0513      complex(dm), allocatable,
0514      complex(dm), allocatable,
0515      integer, allocatable, dime
0516      complex(dm) wsize(1)
0517      real(dm) Rsize(1), atol
0518      integer ISize(1)
0519
0520      atol = 1d-9
0521      call Matrix_Start('Matrix
0522      allocate(tmp(n,nfound))
0523      allocate(Supp(n))
0524      !Query
0525      WorkSize = -1
0526      LRWork = -1
0527      LIWork = -1
0528      #ifdef MATRIX_SINGLE
0529          call CHEEVR('V','V','U',n
0530          Supp,WSize,WorkSize,R
0531      #else
0532          call ZHEEVR('V','V','U',n
0533          Supp,WSize,WorkSize,R
0534      #endif
0535      WorkSize = Real(WSize(1))
0536      LRWork = RSize(1)
0537      LIWork = ISize(1)
0538      allocate(Work(WorkSize),R
0539      #ifdef MATRIX_SINGLE
0540      call CHEEVR('V','V','U',n
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 541

```
0541      Supp,Work,WorkSize,RWork,
0542      #else
0543          call ZHEEVR('V','V','U',n
0544      Supp,Work,WorkSize,RWork,
0545      #endif
0546      deallocate(Supp,Work,RWork)
0547      if (ierr /= 0) call MpiSt
0548      M(1:n,1:nfound) = tmp(1:n
0549      deallocate(tmp)
0550      call Matrix_End('Matrix_C
0551
0552
0553      end subroutine
0554
0555
0556      subroutine Matrix_CDiagon
0557      !Complex version. Does m
0558      integer, intent(in) :: n
0559      complex(dm), intent(inout
0560      real(dm), intent(out) ::
0561      integer ierr, tmpsize, is
0562      real(dm), allocatable, di
0563      complex(dm), allocatable,
0564      integer, allocatable, dime
0565
0566      call Matrix_Start('CDiagon
0567
0568      if (matrix_method == Mat_
0569          !Divide and conquer
0570          tmpsize = 2*N + N**2
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 541

```
0541      Supp,Work,WorkSize,RW
0542      #else
0543          call ZHEEVR('V','V','U',n
0544      Supp,Work,WorkSize,RW
0545      #endif
0546      deallocate(Supp,Work,RWor
0547      if (ierr /= 0) call MpiSt
0548      M(1:n,1:nfound) = tmp(1:n
0549      deallocate(tmp)
0550      call Matrix_End('Matrix_C
0551
0552
0553      end subroutine
0554
0555
0556      subroutine Matrix_CDiagon
0557      !Complex version. Does m
0558      integer, intent(in) :: n
0559      complex(dm), intent(inout
0560      real(dm), intent(out) ::
0561      integer ierr, tmpsize, is
0562      real(dm), allocatable, di
0563      complex(dm), allocatable,
0564      integer, allocatable, dime
0565
0566      call Matrix_Start('CDiagon
0567
0568      if (matrix_method == Mat_
0569          !Divide and conquer
0570          tmpsize = 2*N + N**2
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 571

```
0571      rworksize = 1 + 4*N
0572      isize = (2 + 5*N)*4
0573      allocate(tmp(tmpsize))
0574      allocate(iwork(isize))
0575      #ifdef MATRIX_SINGLE
0576          call CHEEVD('V','U',n
0577      #else
0578          call ZHEEVD('V','U',n
0579      #endif
0580          deallocate(iwork)
0581
0582      else
0583
0584          rworksize = max(1, 3*n-2
0585      #ifdef MATRIX_SINGLE
0586          tmpsize = max( (ILAENV_wr
0587          allocate(tmp(tmpsize),rwo
0588          call CHEEV('V','U',n,m,n,
0589      #else
0590          tmpsize = max( (ILAENV_wr
0591          allocate(tmp(tmpsize),rwo
0592          call ZHEEV('V','U',n,m,n,
0593      #endif
0594      end if
0595
0596      if (ierr /= 0) call Mpist
0597      deallocate(tmp,rwork)
0598
0599      call Matrix_End('CDiagona
0600
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 571

```
0571      rworksize = 1 + 4*N
0572      isize = (2 + 5*N)*4
0573      allocate(tmp(tmpsize))
0574      allocate(iwork(isize))
0575      #ifdef MATRIX_SINGLE
0576          call CHEEVD('V','U',n
0577      #else
0578          call ZHEEVD('V','U',n
0579      #endif
0580          deallocate(iwork)
0581
0582      else
0583
0584          rworksize = max(1, 3
0585      #ifdef MATRIX_SINGLE
0586          tmpsize = max( (ILAEN
0587          allocate(tmp(tmpsize))
0588          call CHEEV('V','U',n,
0589      #else
0590          tmpsize = max( (ILAEN
0591          allocate(tmp(tmpsize))
0592          call ZHEEV('V','U',n,
0593      #endif
0594      end if
0595
0596      if (ierr /= 0) call Mpist
0597      deallocate(tmp,rwork)
0598
0599      call Matrix_End('CDiagona
0600
```

/Users/lplop/Compare/camb_simdata/Matrix_utils.F90, Top line: 601

```
0601      end subroutine Matrix_CDi
0602
0603      function Matrix_CTrace(M)
0604      complex(dm), intent(in) :
0605      complex(dm) tmp,Matrix_CT
0606      integer i
0607
0608      if (size(M,dim=1) /= size
0609      tmp =0
0610      do i=1,size(M,dim=1)
0611          tmp = tmp + M(i,i)
0612      end do
0613      Matrix_CTrace = tmp
0614
0615      end function Matrix_CTrac
0616
0617      function Matrix_Trace(M)
0618      real(dm), intent(in) :: M
0619      real(dm) tmp,Matrix_Trace
0620      integer i
0621
0622      if (size(M,dim=1) /= size
0623      tmp =0
0624      do i=1,size(M,dim=1)
0625          tmp = tmp + M(i,i)
0626      end do
0627      Matrix_Trace = tmp
0628
0629      end function Matrix_Trace
0630
```

/Users/lplop/Compare/camb_des/Matrix_utils.F90, Top line: 601

```
0601      end subroutine Matrix_CDi
0602
0603      function Matrix_CTrace(M)
0604      complex(dm), intent(in) :
0605      complex(dm) tmp,Matrix_CT
0606      integer i
0607
0608      if (size(M,dim=1) /= size
0609      tmp =0
0610      do i=1,size(M,dim=1)
0611          tmp = tmp + M(i,i)
0612      end do
0613      Matrix_CTrace = tmp
0614
0615      end function Matrix_CTrac
0616
0617      function Matrix_Trace(M)
0618      real(dm), intent(in) :: M
0619      real(dm) tmp,Matrix_Trace
0620      integer i
0621
0622      if (size(M,dim=1) /= size
0623      tmp =0
0624      do i=1,size(M,dim=1)
0625          tmp = tmp + M(i,i)
0626      end do
0627      Matrix_Trace = tmp
0628
0629      end function Matrix_Trace
0630
```

```
0631
0632     function MatrixSym_LogDet
0633     real(dm), intent(in) :: m
0634     real(dm) logDet
0635     real(dm) Tmp(size(mat,dim
0636     integer i
0637
0638     if (size(mat,dim=1) /= si
0639     Tmp = mat
0640     call Matrix_Cholesky(tmp)
0641     logDet = 0
0642     do i=1, size(mat,dim=1)
0643         logDet = logDet + lo
0644     end do
0645     logDet = 2._dm*logDet
0646
0647     end function MatrixSym_Lo
0648
0649
0650     subroutine Matrix_CRotate
0651     !Gets U^dag Mat U
0652     integer, intent(in) :: m
0653     complex(dm), intent(in) :
0654     complex(dm) Out(:, :)
0655     complex(dm), dimension(:,
0656     integer n
0657     logical, intent(in), opti
0658     logical :: triang
0659
0660     call Matrix_Start('CRotat
```

```
0631
0632     function MatrixSym_LogDet
0633     real(dm), intent(in) :: m
0634     real(dm) logDet
0635     real(dm) Tmp(size(mat,dim
0636     integer i
0637
0638     if (size(mat,dim=1) /= si
0639     Tmp = mat
0640     call Matrix_Cholesky(tmp)
0641     logDet = 0
0642     do i=1, size(mat,dim=1)
0643         logDet = logDet + lo
0644     end do
0645     logDet = 2._dm*logDet
0646
0647     end function MatrixSym_Lo
0648
0649
0650     subroutine Matrix_CRotate
0651     !Gets U^dag Mat U
0652     integer, intent(in) :: m
0653     complex(dm), intent(in) :
0654     complex(dm) Out(:, :)
0655     complex(dm), dimension(:,
0656     integer n
0657     logical, intent(in), opti
0658     logical :: triang
0659
0660     call Matrix_Start('CRotat
```

```
0661
0662     if (present(triangular))
0663         triang=triangular
0664     else
0665         triang=.false.
0666     end if
0667
0668     n = Size(Mat,DIM=1)
0669     if (n /= Size(Mat,DIM=2))
0670     if (n /= Size(U,DIM=1)) c
0671     if (Size(Out,DIM=1) < m .
0672 call MpiStop('Matrix_CRot
0673
0674     if (matrix_method == Mat_
0675         Out = matmul(matmul(t
0676     else
0677 #ifdef MATRIX_SINGLE
0678     if (triang) then
0679         if (m/=n) call Mp
0680         call CHEMM('L','U
0681         call CTRMM('Left'
0682     else
0683         allocate(C(n,m))
0684         call CHEMM('L','U
0685         call CGEMM('C','N
0686         deallocate(C)
0687     end if
0688 #else
0689     if (triang) then
0690         if (m/=n) call Mp
```

```
0661
0662     if (present(triangular))
0663         triang=triangular
0664     else
0665         triang=.false.
0666     end if
0667
0668     n = Size(Mat,DIM=1)
0669     if (n /= Size(Mat,DIM=2))
0670     if (n /= Size(U,DIM=1)) c
0671     if (Size(Out,DIM=1) < m .
0672 call MpiStop('Matrix_
0673
0674     if (matrix_method == Mat_
0675         Out = matmul(matmul(t
0676     else
0677 #ifdef MATRIX_SINGLE
0678     if (triang) then
0679         if (m/=n) call Mp
0680         call CHEMM('L','U
0681         call CTRMM('Left'
0682     else
0683         allocate(C(n,m))
0684         call CHEMM('L','U
0685         call CGEMM('C','N
0686         deallocate(C)
0687     end if
0688 #else
0689     if (triang) then
0690         if (m/=n) call Mp
```

/Users/lplopa/Compare/camb_simdata/Matrix_utils.F90, Top line: 691

```
0691          call ZHEMM('L','U
0692          call ZTRMM('Left'
0693      else
0694          allocate(C(n,m))
0695          call ZHEMM('L','U
0696          call ZGEMM('C','N
0697          deallocate(C)
0698      end if
0699 #endif
0700 end if
0701 call Matrix_End('CRotates
0702
0703
0704 end subroutine Matrix_CRo
0705
0706 subroutine Matrix_Rotates
0707 !Gets U^T Mat U
0708 !If triangular U = Upper
0709 integer, intent(in) :: m
0710 real(dm), intent(in) :: M
0711 real(dm) Out(:, :)
0712 real(dm), dimension(:, :),
0713 logical, intent(in), opti
0714 logical triang
0715 integer n
0716
0717 call Matrix_Start('Rotate
0718
0719 if (present(triangular))
0720     triang=triangular
```

/Users/lplopa/Compare/camb_des/Matrix_utils.F90, Top line: 691

```
0691          call ZHEMM('L','U
0692          call ZTRMM('Left'
0693      else
0694          allocate(C(n,m))
0695          call ZHEMM('L','U
0696          call ZGEMM('C','N
0697          deallocate(C)
0698      end if
0699 #endif
0700 end if
0701 call Matrix_End('CRotates
0702
0703
0704 end subroutine Matrix_CRo
0705
0706 subroutine Matrix_Rotates
0707 !Gets U^T Mat U
0708 !If triangular U = Upper
0709 integer, intent(in) :: m
0710 real(dm), intent(in) :: M
0711 real(dm) Out(:, :)
0712 real(dm), dimension(:, :),
0713 logical, intent(in), opti
0714 logical triang
0715 integer n
0716
0717 call Matrix_Start('Rotate
0718
0719 if (present(triangular))
0720     triang=triangular
```


/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 721

```
0721     else
0722         triang=.false.
0723     end if
0724
0725     n = Size(Mat,DIM=1)
0726     if (n /= Size(Mat,DIM=2))
0727     if (n /= Size(U,DIM=1)) c
0728     if (Size(Out,DIM=1) < m .
0729 call MpiStop('Matrix_Rota
0730
0731     if (matrix_method == Mat_
0732         Out = matmul(matmul(t
0733     else
0734 #ifdef MATRIX_SINGLE
0735         if (triang) then
0736             if (m/=n) call Mp
0737             call SSYMM('L','U
0738             call STRMM('Left'
0739         else
0740             allocate(C(n,m))
0741             call SSYMM('L','U
0742             call SGEMM('T','N
0743             deallocate(C)
0744         end if
0745 #else
0746         if (triang) then
0747             if (m/=n) call Mp
0748             call DSYMM('L','U
0749             call DTRMM('Left'
0750         else
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 721

```
0721     else
0722         triang=.false.
0723     end if
0724
0725     n = Size(Mat,DIM=1)
0726     if (n /= Size(Mat,DIM=2))
0727     if (n /= Size(U,DIM=1)) c
0728     if (Size(Out,DIM=1) < m .
0729 call MpiStop('Matrix_
0730
0731     if (matrix_method == Mat_
0732         Out = matmul(matmul(t
0733     else
0734 #ifdef MATRIX_SINGLE
0735         if (triang) then
0736             if (m/=n) call Mp
0737             call SSYMM('L','U
0738             call STRMM('Left'
0739         else
0740             allocate(C(n,m))
0741             call SSYMM('L','U
0742             call SGEMM('T','N
0743             deallocate(C)
0744         end if
0745 #else
0746         if (triang) then
0747             if (m/=n) call Mp
0748             call DSYMM('L','U
0749             call DTRMM('Left'
0750         else
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 751

```
0751         allocate(C(n,m))
0752         call DSYMM('L','U
0753         call DGEMM('T','N
0754         deallocate(C)
0755     end if
0756 #endif
0757     end if
0758     call Matrix_End('RotateSy
0759
0760
0761 end subroutine Matrix_Rot
0762
0763
0764 subroutine Matrix_RotateA
0765 !Gets U^T Mat U
0766 !Where Mat = -Mat^T
0767 integer, intent(in) :: m
0768 real(dm), intent(in) :: M
0769 real(dm) Out(:, :)
0770 real(dm), dimension(:, :),
0771 integer i, j, n
0772
0773 call Matrix_Start('Rotate
0774
0775 n = Size(Mat, DIM=1)
0776 if (n /= Size(Mat, DIM=2))
0777 if (n /= Size(U, DIM=1)) c
0778 if (Size(Out, DIM=1) < m .
0779 call MpiStop('Matrix_Rota
0780
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 751

```
0751         allocate(C(n,m))
0752         call DSYMM('L','U
0753         call DGEMM('T','N
0754         deallocate(C)
0755     end if
0756 #endif
0757     end if
0758     call Matrix_End('RotateSy
0759
0760
0761 end subroutine Matrix_Rot
0762
0763
0764 subroutine Matrix_RotateA
0765 !Gets U^T Mat U
0766 !Where Mat = -Mat^T
0767 integer, intent(in) :: m
0768 real(dm), intent(in) :: M
0769 real(dm) Out(:, :)
0770 real(dm), dimension(:, :),
0771 integer i, j, n
0772
0773 call Matrix_Start('Rotate
0774
0775 n = Size(Mat, DIM=1)
0776 if (n /= Size(Mat, DIM=2))
0777 if (n /= Size(U, DIM=1)) c
0778 if (Size(Out, DIM=1) < m .
0779 call MpiStop('Matrix_
0780
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 781

```
0781      if (matrix_method == Mat_
0782          Out = matmul(matmul(t
0783      else
0784          allocate(C(n,m))
0785          C = U(1:n,1:m)
0786      #ifdef MATRIX_SINGLE
0787          call STRMM('Left','Lo
0788          call SGEMM('T','N',m,
0789      #else
0790          call DTRMM('Left','Lo
0791          call DGEMM('T','N',m,
0792      #endif
0793          deallocate(C)
0794      end if
0795
0796      do i=1, m
0797          do j=1,i
0798              Out(j,i) = Out(j,
0799              out(i,j) = -Out(j
0800          end do
0801      end do
0802
0803      call Matrix_End('RotateAn
0804
0805      end subroutine Matrix_Rot
0806
0807      subroutine Matrix_CMult_S
0808      complex(dm), intent(in) :
0809      complex(dm) Out(:, :)
0810      complex(dm), intent(in),
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 781

```
0781      if (matrix_method == Mat_
0782          Out = matmul(matmul(t
0783      else
0784          allocate(C(n,m))
0785          C = U(1:n,1:m)
0786      #ifdef MATRIX_SINGLE
0787          call STRMM('Left','Lo
0788          call SGEMM('T','N',m,
0789      #else
0790          call DTRMM('Left','Lo
0791          call DGEMM('T','N',m,
0792      #endif
0793          deallocate(C)
0794      end if
0795
0796      do i=1, m
0797          do j=1,i
0798              Out(j,i) = Out(j,
0799              out(i,j) = -Out(j
0800          end do
0801      end do
0802
0803      call Matrix_End('RotateAn
0804
0805      end subroutine Matrix_Rot
0806
0807      subroutine Matrix_CMult_S
0808      complex(dm), intent(in) :
0809      complex(dm) Out(:, :)
0810      complex(dm), intent(in),
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 811

```
0811 complex(dm) mult, beta
0812 integer n,m
0813
0814 call Matrix_Start('CMult_
0815
0816 m = Size(Mat,DIM=1)
0817 n = Size(U,DIM=2)
0818 if (n /= Size(Mat,DIM=2)
0819 call MpiStop('Matrix_CMul
0820 if (present(a)) then
0821     mult = a
0822 else
0823     mult = COne
0824 end if
0825 if (present(b)) then
0826     beta = b
0827 else
0828     beta = CZero
0829 end if
0830 if (matrix_method == Mat_
0831     if (beta /= CZero) th
0832         out = a*MatMul(Ma
0833     else
0834         out = MatMul(Mat,
0835         if (mult /= COne)
0836     end if
0837 else
0838 #ifdef MATRIX_SINGLE
0839     call CHEMM('R','U',m,
0840 #else
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 811

```
0811 complex(dm) mult, beta
0812 integer n,m
0813
0814 call Matrix_Start('CMult_
0815
0816 m = Size(Mat,DIM=1)
0817 n = Size(U,DIM=2)
0818 if (n /= Size(Mat,DIM=2)
0819 call MpiStop('Matrix_
0820 if (present(a)) then
0821     mult = a
0822 else
0823     mult = COne
0824 end if
0825 if (present(b)) then
0826     beta = b
0827 else
0828     beta = CZero
0829 end if
0830 if (matrix_method == Mat_
0831     if (beta /= CZero) th
0832         out = a*MatMul(Ma
0833     else
0834         out = MatMul(Mat,
0835         if (mult /= COne)
0836     end if
0837 else
0838 #ifdef MATRIX_SINGLE
0839     call CHEMM('R','U',m,
0840 #else
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 841

```
0841      call ZHEMM('R','U',m,  
0842      #endif  
0843      end if  
0844  
0845      call Matrix_End('CMult_Sy  
0846  
0847      end subroutine Matrix_CMu  
0848  
0849  
0850      subroutine Matrix_CMult_S  
0851      complex(dm), intent(in) :  
0852      complex(dm) Out(:, :)  
0853      complex(dm), intent(in),  
0854      complex(dm) mult, beta  
0855      integer n,m  
0856  
0857      call Matrix_Start('CMult_  
0858  
0859      m = Size(Mat,DIM=1)  
0860      n = Size(U,DIM=2)  
0861      if (m /= Size(U,DIM=1)) .o  
0862      call MpiStop('Matrix_CMul  
0863      if (present(a)) then  
0864          mult = a  
0865      else  
0866          mult = COne  
0867      end if  
0868      if (present(b)) then  
0869          beta = b  
0870      else
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 841

```
0841      call ZHEMM('R','U',m,  
0842      #endif  
0843      end if  
0844  
0845      call Matrix_End('CMult_Sy  
0846  
0847      end subroutine Matrix_CMu  
0848  
0849  
0850      subroutine Matrix_CMult_S  
0851      complex(dm), intent(in) :  
0852      complex(dm) Out(:, :)  
0853      complex(dm), intent(in),  
0854      complex(dm) mult, beta  
0855      integer n,m  
0856  
0857      call Matrix_Start('CMult_  
0858  
0859      m = Size(Mat,DIM=1)  
0860      n = Size(U,DIM=2)  
0861      if (m /= Size(U,DIM=1)) .o  
0862      call MpiStop('Matrix_  
0863      if (present(a)) then  
0864          mult = a  
0865      else  
0866          mult = COne  
0867      end if  
0868      if (present(b)) then  
0869          beta = b  
0870      else
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 871

```
0871      beta = CZero
0872    end if
0873    if (matrix_method == Mat_
0874      if (beta /= CZero) th
0875        out = a*MatMul(Ma
0876      else
0877        out = MatMul(Mat,
0878          if (mult /= COne)
0879        end if
0880      else
0881    #ifdef MATRIX_SINGLE
0882      call CHEMM('L','U',m,
0883    #else
0884      call ZHEMM('L','U',m,
0885    #endif
0886    end if
0887
0888    call Matrix_End('CMult_Sy
0889
0890    end subroutine Matrix_CMu
0891
0892
0893    subroutine Matrix_CMult(M
0894      ! Out = a*Mat U + b*out
0895      complex(dm), intent(in) :
0896      complex(dm) Out(:, :)
0897      complex(dm), intent(in),
0898      complex(dm) mult, beta
0899      integer m,n,k
0900
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 871

```
0871      beta = CZero
0872    end if
0873    if (matrix_method == Mat_
0874      if (beta /= CZero) th
0875        out = a*MatMul(Ma
0876      else
0877        out = MatMul(Mat,
0878          if (mult /= COne)
0879        end if
0880      else
0881    #ifdef MATRIX_SINGLE
0882      call CHEMM('L','U',m,
0883    #else
0884      call ZHEMM('L','U',m,
0885    #endif
0886    end if
0887
0888    call Matrix_End('CMult_Sy
0889
0890    end subroutine Matrix_CMu
0891
0892
0893    subroutine Matrix_CMult(M
0894      ! Out = a*Mat U + b*out
0895      complex(dm), intent(in) :
0896      complex(dm) Out(:, :)
0897      complex(dm), intent(in),
0898      complex(dm) mult, beta
0899      integer m,n,k
0900
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 901

```
0901      call Matrix_Start('CMult'
0902
0903      m = Size(Mat,DIM=1)
0904      n = Size(U,DIM=2)
0905      k = Size(Mat,DIM=2)
0906      if (k /= Size(U,DIM=1)) c
0907      if (present(a)) then
0908          mult = a
0909      else
0910          mult = COne
0911      end if
0912      if (present(b)) then
0913          beta = b
0914      else
0915          beta = CZero
0916      end if
0917
0918      if (matrix_method == Mat_
0919          if (beta /= CZero) th
0920          out = a*MatMul(Ma
0921      else
0922          out = MatMul(Mat,
0923          if (mult /= COne)
0924      end if
0925      else
0926      #ifdef MATRIX_SINGLE
0927          call CGEMM('N','N',m,
0928      #else
0929          call ZGEMM('N','N',m,
0930      #endif
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 901

```
0901      call Matrix_Start('CMult'
0902
0903      m = Size(Mat,DIM=1)
0904      n = Size(U,DIM=2)
0905      k = Size(Mat,DIM=2)
0906      if (k /= Size(U,DIM=1)) c
0907      if (present(a)) then
0908          mult = a
0909      else
0910          mult = COne
0911      end if
0912      if (present(b)) then
0913          beta = b
0914      else
0915          beta = CZero
0916      end if
0917
0918      if (matrix_method == Mat_
0919          if (beta /= CZero) th
0920          out = a*MatMul(Ma
0921      else
0922          out = MatMul(Mat,
0923          if (mult /= COne)
0924      end if
0925      else
0926      #ifdef MATRIX_SINGLE
0927          call CGEMM('N','N',m,
0928      #else
0929          call ZGEMM('N','N',m,
0930      #endif
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 931

```
0931      end if
0932      call Matrix_End('CMult')
0933
0934
0935      end subroutine Matrix_CMu
0936
0937
0938      subroutine Matrix_MultSq_
0939      !U = a*Mat*U
0940      real(dm), intent(in) :: M
0941      real(dm), intent(inout) :
0942      real(dm), intent(in), opt
0943      real(dm) aa
0944      integer m,n
0945      real(dm), dimension(:, :),
0946
0947
0948      m = Size(Mat,DIM=1)
0949      n = Size(Mat,DIM=2)
0950      if (m /= n) call MpiStop(
0951      m = Size(U,DIM=1)
0952      n = Size(U,DIM=2)
0953      if (m /= n) call MpiStop(
0954
0955      allocate(tmp(n,n))
0956      if (present(a)) then
0957          aa=a
0958      else
0959          aa=ROne
0960      end if
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 931

```
0931      end if
0932      call Matrix_End('CMult')
0933
0934
0935      end subroutine Matrix_CMu
0936
0937
0938      subroutine Matrix_MultSq_
0939      !U = a*Mat*U
0940      real(dm), intent(in) :: M
0941      real(dm), intent(inout) :
0942      real(dm), intent(in), opt
0943      real(dm) aa
0944      integer m,n
0945      real(dm), dimension(:, :),
0946
0947
0948      m = Size(Mat,DIM=1)
0949      n = Size(Mat,DIM=2)
0950      if (m /= n) call MpiStop(
0951      m = Size(U,DIM=1)
0952      n = Size(U,DIM=2)
0953      if (m /= n) call MpiStop(
0954
0955      allocate(tmp(n,n))
0956      if (present(a)) then
0957          aa=a
0958      else
0959          aa=ROne
0960      end if
```



```
0961
0962      call Matrix_Mult(Mat,U,tm
0963      U = tmp
0964      deallocate(tmp)
0965
0966      end  subroutine Matrix_Mu
0967
0968      subroutine Matrix_MultTri
0969      ! Mat -> L Mat or Mat L w
0970      real(dm), intent(inout) :
0971      real(dm), intent(in) :: L
0972      character(LEN=*), intent(
0973      integer m,n
0974
0975      call Matrix_Start('Matrix
0976
0977      m = Size(Mat,DIM=1)
0978      n = Size(Mat,DIM=2)
0979
0980      if (side(1:1)=='L') then
0981          if (Size(L,DIM=2) /=
0982      else
0983          if (Size(L,DIM=1) /=
0984      end if
0985      #ifdef MATRIX_SINGLE
0986      call STRMM(side,'Lower','
0987      #else
0988      call DTRMM(side,'Lower','
0989      #endif
0990      call Matrix_End('Matrix_M
```

```
0961
0962      call Matrix_Mult(Mat,U,tm
0963      U = tmp
0964      deallocate(tmp)
0965
0966      end  subroutine Matrix_Mu
0967
0968      subroutine Matrix_MultTri
0969      ! Mat -> L Mat or Mat L w
0970      real(dm), intent(inout) :
0971      real(dm), intent(in) :: L
0972      character(LEN=*), intent(
0973      integer m,n
0974
0975      call Matrix_Start('Matrix
0976
0977      m = Size(Mat,DIM=1)
0978      n = Size(Mat,DIM=2)
0979
0980      if (side(1:1)=='L') then
0981          if (Size(L,DIM=2) /=
0982      else
0983          if (Size(L,DIM=1) /=
0984      end if
0985      #ifdef MATRIX_SINGLE
0986      call STRMM(side,'Lower','
0987      #else
0988      call DTRMM(side,'Lower','
0989      #endif
0990      call Matrix_End('Matrix_M
```

```
0991
0992     end subroutine Matrix_Mul
0993
0994
0995
0996     subroutine Matrix_Mult(Ma
0997     ! Out = a*Mat U + b*out
0998     real(dm), intent(in) :: M
0999     real(dm) :: Out(:, :)
1000     real(dm), intent(in), opt
1001     real(dm) mult, beta
1002     integer m,n,k
1003
1004     call Matrix_Start('Mult')
1005
1006
1007     m = Size(Mat,DIM=1)
1008     n = Size(U,DIM=2)
1009     k = Size(Mat,DIM=2)
1010     if (k /= Size(U,DIM=1)) c
1011
1012
1013     if (present(a)) then
1014         mult = a
1015     else
1016         mult = ROne
1017     end if
1018     if (present(b)) then
1019         beta = b
1020     else
```

```
0991
0992     end subroutine Matrix_Mul
0993
0994
0995
0996     subroutine Matrix_Mult(Ma
0997     ! Out = a*Mat U + b*out
0998     real(dm), intent(in) :: M
0999     real(dm) :: Out(:, :)
1000     real(dm), intent(in), opt
1001     real(dm) mult, beta
1002     integer m,n,k
1003
1004     call Matrix_Start('Mult')
1005
1006
1007     m = Size(Mat,DIM=1)
1008     n = Size(U,DIM=2)
1009     k = Size(Mat,DIM=2)
1010     if (k /= Size(U,DIM=1)) c
1011
1012
1013     if (present(a)) then
1014         mult = a
1015     else
1016         mult = ROne
1017     end if
1018     if (present(b)) then
1019         beta = b
1020     else
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1021

```
1021      beta = RZero
1022      end if
1023
1024      if (matrix_method == Mat_
1025          if (beta /= RZero) th
1026              out = a*MatMul(Ma
1027          else
1028              out = MatMul(Mat,
1029              if (mult /= ROne)
1030          end if
1031      else
1032      #ifdef MATRIX_SINGLE
1033          call SGEMM('N','N',m,
1034      #else
1035          call DGEMM('N','N',m,
1036      #endif
1037      end if
1038      call Matrix_End('Mult')
1039
1040
1041      end subroutine Matrix_Mul
1042
1043      subroutine Matrix_Mult_Sy
1044      real(dm), intent(in) :: M
1045      real(dm) Out(:, :)
1046      real(dm), intent(in), opt
1047      real(dm) mult, beta
1048      integer n,m
1049
1050      call Matrix_Start('Mult_S
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1021

```
1021      beta = RZero
1022      end if
1023
1024      if (matrix_method == Mat_
1025          if (beta /= RZero) th
1026              out = a*MatMul(Ma
1027          else
1028              out = MatMul(Mat,
1029              if (mult /= ROne)
1030          end if
1031      else
1032      #ifdef MATRIX_SINGLE
1033          call SGEMM('N','N',m,
1034      #else
1035          call DGEMM('N','N',m,
1036      #endif
1037      end if
1038      call Matrix_End('Mult')
1039
1040
1041      end subroutine Matrix_Mul
1042
1043      subroutine Matrix_Mult_Sy
1044      real(dm), intent(in) :: M
1045      real(dm) Out(:, :)
1046      real(dm), intent(in), opt
1047      real(dm) mult, beta
1048      integer n,m
1049
1050      call Matrix_Start('Mult_S
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1051

```
1051
1052     m = Size(Mat,DIM=1)
1053     n = Size(U,DIM=2)
1054     if (m /= Size(U,DIM=1)) .o
1055 call MpiStop('Matrix_Mult
1056 if (present(a)) then
1057     mult = a
1058 else
1059     mult = ROne
1060 end if
1061 if (present(b)) then
1062     beta = b
1063 else
1064     beta = RZero
1065 end if
1066 if (matrix_method == Mat_
1067     if (beta /= RZero) th
1068         out = a*MatMul(Ma
1069     else
1070         out = MatMul(Mat,
1071         if (mult /= COne)
1072     end if
1073 else
1074 #ifdef MATRIX_SINGLE
1075     call SSYMM('L','U',m,
1076 #else
1077     call DSYMM('L','U',m,
1078 #endif
1079     end if
1080
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1051

```
1051
1052     m = Size(Mat,DIM=1)
1053     n = Size(U,DIM=2)
1054     if (m /= Size(U,DIM=1)) .o
1055 call MpiStop('Matrix_
1056 if (present(a)) then
1057     mult = a
1058 else
1059     mult = ROne
1060 end if
1061 if (present(b)) then
1062     beta = b
1063 else
1064     beta = RZero
1065 end if
1066 if (matrix_method == Mat_
1067     if (beta /= RZero) th
1068         out = a*MatMul(Ma
1069     else
1070         out = MatMul(Mat,
1071         if (mult /= COne)
1072     end if
1073 else
1074 #ifdef MATRIX_SINGLE
1075     call SSYMM('L','U',m,
1076 #else
1077     call DSYMM('L','U',m,
1078 #endif
1079     end if
1080
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1081

```
1081      call Matrix_End('Mult_Sym
1082
1083      end subroutine Matrix_Mul
1084
1085
1086      subroutine Matrix_Mult_Sy
1087      ! Out = a*Mat U + b*out
1088      real(dm), intent(in) :: M
1089      real(dm) Out(:, :)
1090      real(dm), intent(in), opt
1091      real(dm) mult, beta
1092      integer n,m
1093
1094      call Matrix_Start('Mult_S
1095
1096      m = Size(Mat,DIM=1)
1097      n = Size(U,DIM=2)
1098      if (n /= Size(Mat,DIM=2)
1099      call MpiStop('Matrix_Mult
1100      if (present(a)) then
1101          mult = a
1102      else
1103          mult = ROne
1104      end if
1105      if (present(b)) then
1106          beta = b
1107      else
1108          beta = RZero
1109      end if
1110      if (matrix_method == Mat_
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1081

```
1081      call Matrix_End('Mult_Sym
1082
1083      end subroutine Matrix_Mul
1084
1085
1086      subroutine Matrix_Mult_Sy
1087      ! Out = a*Mat U + b*out
1088      real(dm), intent(in) :: M
1089      real(dm) Out(:, :)
1090      real(dm), intent(in), opt
1091      real(dm) mult, beta
1092      integer n,m
1093
1094      call Matrix_Start('Mult_S
1095
1096      m = Size(Mat,DIM=1)
1097      n = Size(U,DIM=2)
1098      if (n /= Size(Mat,DIM=2)
1099      call MpiStop('Matrix_
1100      if (present(a)) then
1101          mult = a
1102      else
1103          mult = ROne
1104      end if
1105      if (present(b)) then
1106          beta = b
1107      else
1108          beta = RZero
1109      end if
1110      if (matrix_method == Mat_
```

/Users/lplop/Compare/camb_simdata/Matrix_utils.F90, Top line: 1111

```
1111         if (beta /= RZero) th
1112             out = a*MatMul(Ma
1113         else
1114             out = MatMul(Mat,
1115             if (mult /= ROne)
1116         end if
1117     else
1118 #ifdef MATRIX_SINGLE
1119         call SSYMM('R','U',m,
1120 #else
1121         call DSYMM('R','U',m,
1122 #endif
1123     end if
1124
1125     call Matrix_End('Mult_Sym
1126
1127     end subroutine Matrix_Mul
1128
1129
1130     subroutine Matrix_CMultGe
1131     !      out(1:m,1:n) = MatM
1132     integer, intent(in) :: m,
1133     complex(dm), intent(in) :
1134     complex(dm) Out(:, :)
1135
1136     call Matrix_Start('CMultG
1137
1138     if (SIZE(Out,DIM=1) < m .o
1139
1140     if (matrix_method == Mat_
```

/Users/lplop/Compare/camb_des/Matrix_utils.F90, Top line: 1111

```
1111         if (beta /= RZero) th
1112             out = a*MatMul(Ma
1113         else
1114             out = MatMul(Mat,
1115             if (mult /= ROne)
1116         end if
1117     else
1118 #ifdef MATRIX_SINGLE
1119         call SSYMM('R','U',m,
1120 #else
1121         call DSYMM('R','U',m,
1122 #endif
1123     end if
1124
1125     call Matrix_End('Mult_Sym
1126
1127     end subroutine Matrix_Mul
1128
1129
1130     subroutine Matrix_CMultGe
1131     !      out(1:m,1:n) = MatM
1132     integer, intent(in) :: m,
1133     complex(dm), intent(in) :
1134     complex(dm) Out(:, :)
1135
1136     call Matrix_Start('CMultG
1137
1138     if (SIZE(Out,DIM=1) < m .o
1139
1140     if (matrix_method == Mat_
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1141

```
1141      out(1:m,1:n) = MatMul
1142      else
1143      #ifdef MATRIX_SINGLE
1144          call CGEMM('N','N',m,
1145      #else
1146          call ZGEMM('N','N',m,
1147      #endif
1148          end if
1149          call Matrix_End('CMultGen
1150
1151
1152      end subroutine Matrix_CMu
1153
1154
1155
1156      subroutine Matrix_MultGen
1157      !      out(1:m,1:n) = MatM
1158      integer, intent(in) :: m,
1159      real(dm), intent(in) :: M
1160      real(dm) Out(:, :)
1161
1162      call Matrix_Start('MultGe
1163
1164      if (SIZE(Out,DIM=1) < m .o
1165
1166      if (matrix_method == Mat_
1167          out(1:m,1:n) = MatMul
1168      else
1169      #ifdef MATRIX_SINGLE
1170          call SGEMM('N','N',m,
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1141

```
1141      out(1:m,1:n) = MatMul
1142      else
1143      #ifdef MATRIX_SINGLE
1144          call CGEMM('N','N',m,
1145      #else
1146          call ZGEMM('N','N',m,
1147      #endif
1148          end if
1149          call Matrix_End('CMultGen
1150
1151
1152      end subroutine Matrix_CMu
1153
1154
1155
1156      subroutine Matrix_MultGen
1157      !      out(1:m,1:n) = MatM
1158      integer, intent(in) :: m,
1159      real(dm), intent(in) :: M
1160      real(dm) Out(:, :)
1161
1162      call Matrix_Start('MultGe
1163
1164      if (SIZE(Out,DIM=1) < m .o
1165
1166      if (matrix_method == Mat_
1167          out(1:m,1:n) = MatMul
1168      else
1169      #ifdef MATRIX_SINGLE
1170          call SGEMM('N','N',m,
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1171

```
1171      #else
1172              call DGEMM('N','N',m,
1173      #endif
1174              end if
1175              call Matrix_End('MultGen'
1176
1177
1178      end subroutine Matrix_Mul
1179
1180
1181      subroutine Matrix_CMult_N
1182      ! Out = a*Mat U^dag + b*o
1183      complex(dm), intent(in) :
1184      complex(dm) Out(:, :)
1185      complex(dm), intent(in),
1186      complex(dm) mult, beta
1187      integer m,n,k
1188
1189      m = Size(Mat,DIM=1)
1190      n = Size(U,DIM=1)
1191      k = Size(Mat,DIM=2)
1192      if (k /= Size(U,DIM=2)) c
1193      call Matrix_start('CMult_
1194      if (present(a)) then
1195          mult = a
1196      else
1197          mult = COne
1198      end if
1199      if (present(b)) then
1200          beta = b
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1171

```
1171      #else
1172              call DGEMM('N','N',m,
1173      #endif
1174              end if
1175              call Matrix_End('MultGen'
1176
1177
1178      end subroutine Matrix_Mul
1179
1180
1181      subroutine Matrix_CMult_N
1182      ! Out = a*Mat U^dag + b*o
1183      complex(dm), intent(in) :
1184      complex(dm) Out(:, :)
1185      complex(dm), intent(in),
1186      complex(dm) mult, beta
1187      integer m,n,k
1188
1189      m = Size(Mat,DIM=1)
1190      n = Size(U,DIM=1)
1191      k = Size(Mat,DIM=2)
1192      if (k /= Size(U,DIM=2)) c
1193      call Matrix_start('CMult_
1194      if (present(a)) then
1195          mult = a
1196      else
1197          mult = COne
1198      end if
1199      if (present(b)) then
1200          beta = b
```


/Users/lplop/Compare/camb_simdata/Matrix_utils.F90, Top line: 1201

```
1201      else
1202          beta = CZero
1203      end if
1204
1205      if (matrix_method == Mat_
1206          if (beta /= CZero) th
1207          Out = beta*Out +
1208          else
1209          Out = matmul(Mat,
1210          if (mult/= COne)
1211          end if
1212      else
1213      #ifdef MATRIX_SINGLE
1214          call CGEMM('N','C',m,
1215      #else
1216          call ZGEMM('N','C',m,
1217      #endif
1218      end if
1219      call Matrix_End('CMult_NT
1220
1221
1222      end subroutine Matrix_CMu
1223
1224      subroutine Matrix_Mult_NT
1225      ! Out = a*Mat U^T + b*out
1226      real(dm), intent(in) :: M
1227      real(dm) Out(:, :)
1228      real(dm), intent(in), opt
1229      real(dm) mult, beta
1230      integer m,n,k
```

/Users/lplop/Compare/camb_des/Matrix_utils.F90, Top line: 1201

```
1201      else
1202          beta = CZero
1203      end if
1204
1205      if (matrix_method == Mat_
1206          if (beta /= CZero) th
1207          Out = beta*Out +
1208          else
1209          Out = matmul(Mat,
1210          if (mult/= COne)
1211          end if
1212      else
1213      #ifdef MATRIX_SINGLE
1214          call CGEMM('N','C',m,
1215      #else
1216          call ZGEMM('N','C',m,
1217      #endif
1218      end if
1219      call Matrix_End('CMult_NT
1220
1221
1222      end subroutine Matrix_CMu
1223
1224      subroutine Matrix_Mult_NT
1225      ! Out = a*Mat U^T + b*out
1226      real(dm), intent(in) :: M
1227      real(dm) Out(:, :)
1228      real(dm), intent(in), opt
1229      real(dm) mult, beta
1230      integer m,n,k
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1231

```
1231
1232      m = Size(Mat,DIM=1)
1233      n = Size(U,DIM=1)
1234      k = Size(Mat,DIM=2)
1235      if (k /= Size(U,DIM=2)) c
1236      call Matrix_start('Mult_N
1237      if (present(a)) then
1238          mult = a
1239      else
1240          mult = ROne
1241      end if
1242      if (present(b)) then
1243          beta = b
1244      else
1245          beta = RZero
1246      end if
1247
1248      if (matrix_method == Mat_
1249          if (beta /= RZero) th
1250          Out = beta*Out +
1251      else
1252          Out = matmul(Mat,
1253          if (mult/= ROne)
1254      end if
1255      else
1256      #ifdef MATRIX_SINGLE
1257          call SGEMM('N','T',m,
1258      #else
1259          call DGEMM('N','T',m,
1260      #endif
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1231

```
1231
1232      m = Size(Mat,DIM=1)
1233      n = Size(U,DIM=1)
1234      k = Size(Mat,DIM=2)
1235      if (k /= Size(U,DIM=2)) c
1236      call Matrix_start('Mult_N
1237      if (present(a)) then
1238          mult = a
1239      else
1240          mult = ROne
1241      end if
1242      if (present(b)) then
1243          beta = b
1244      else
1245          beta = RZero
1246      end if
1247
1248      if (matrix_method == Mat_
1249          if (beta /= RZero) th
1250          Out = beta*Out +
1251      else
1252          Out = matmul(Mat,
1253          if (mult/= ROne)
1254      end if
1255      else
1256      #ifdef MATRIX_SINGLE
1257          call SGEMM('N','T',m,
1258      #else
1259          call DGEMM('N','T',m,
1260      #endif
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1261

```
1261      end if
1262      call Matrix_End('Mult_NT'
1263
1264
1265      end subroutine Matrix_Mul
1266
1267
1268      subroutine Matrix_CMult_T
1269      ! Out = a*Mat^dag U + b*O
1270      complex(dm), intent(in) :
1271      complex(dm) Out(:, :)
1272      complex(dm), intent(in),
1273      complex(dm) mult, beta
1274      integer m,n,k
1275
1276      m = Size(Mat,DIM=2)
1277      n = Size(U,DIM=2)
1278      k = Size(Mat,DIM=1)
1279      if (k /= Size(U,DIM=1)) c
1280
1281      call Matrix_Start('CMult_
1282      if (present(a)) then
1283          mult = a
1284      else
1285          mult = COne
1286      end if
1287      if (present(b)) then
1288          beta = b
1289      else
1290          beta = CZero
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1261

```
1261      end if
1262      call Matrix_End('Mult_NT'
1263
1264
1265      end subroutine Matrix_Mul
1266
1267
1268      subroutine Matrix_CMult_T
1269      ! Out = a*Mat^dag U + b*O
1270      complex(dm), intent(in) :
1271      complex(dm) Out(:, :)
1272      complex(dm), intent(in),
1273      complex(dm) mult, beta
1274      integer m,n,k
1275
1276      m = Size(Mat,DIM=2)
1277      n = Size(U,DIM=2)
1278      k = Size(Mat,DIM=1)
1279      if (k /= Size(U,DIM=1)) c
1280
1281      call Matrix_Start('CMult_
1282      if (present(a)) then
1283          mult = a
1284      else
1285          mult = COne
1286      end if
1287      if (present(b)) then
1288          beta = b
1289      else
1290          beta = CZero
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1291

```
1291      end if
1292      if (matrix_method == Mat_
1293          if (beta /= CZero) th
1294              out = mult*MatMul
1295          else
1296              out = MatMul(conj
1297                  if (mult /= COne)
1298              end if
1299          else
1300      #ifdef MATRIX_SINGLE
1301          call CGEMM('C','N',m,
1302      #else
1303          call ZGEMM('C','N',m,
1304      #endif
1305      end if
1306      call Matrix_End('CMult_TN
1307
1308      end subroutine Matrix_CMu
1309
1310      subroutine Matrix_Mult_TN
1311      ! Out = a*Mat^dag U + b*O
1312      real(dm), intent(in) :: M
1313      real(dm) Out(:, :)
1314      real(dm), intent(in), opt
1315      real(dm) mult, beta
1316      integer m,n,k
1317
1318      m = Size(Mat,DIM=2)
1319      n = Size(U,DIM=2)
1320      k = Size(Mat,DIM=1)
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1291

```
1291      end if
1292      if (matrix_method == Mat_
1293          if (beta /= CZero) th
1294              out = mult*MatMul
1295          else
1296              out = MatMul(conj
1297                  if (mult /= COne)
1298              end if
1299          else
1300      #ifdef MATRIX_SINGLE
1301          call CGEMM('C','N',m,
1302      #else
1303          call ZGEMM('C','N',m,
1304      #endif
1305      end if
1306      call Matrix_End('CMult_TN
1307
1308      end subroutine Matrix_CMu
1309
1310      subroutine Matrix_Mult_TN
1311      ! Out = a*Mat^dag U + b*O
1312      real(dm), intent(in) :: M
1313      real(dm) Out(:, :)
1314      real(dm), intent(in), opt
1315      real(dm) mult, beta
1316      integer m,n,k
1317
1318      m = Size(Mat,DIM=2)
1319      n = Size(U,DIM=2)
1320      k = Size(Mat,DIM=1)
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1321

```
1321      if (k /= Size(U,DIM=1)) c
1322
1323      call Matrix_Start('CMult_
1324      if (present(a)) then
1325          mult = a
1326      else
1327          mult = ROne
1328      end if
1329      if (present(b)) then
1330          beta = b
1331      else
1332          beta = RZero
1333      end if
1334      if (matrix_method == Mat_
1335          if (beta /= CZero) th
1336          out = mult*MatMul
1337      else
1338          out = MatMul(tran
1339          if (mult /= COne)
1340      end if
1341      else
1342      #ifdef MATRIX_SINGLE
1343          call SGEMM('T','N',m,
1344      #else
1345          call DGEMM('T','N',m,
1346      #endif
1347      end if
1348      call Matrix_End('Mult_TN'
1349
1350      end subroutine Matrix_Mul
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1321

```
1321      if (k /= Size(U,DIM=1)) c
1322
1323      call Matrix_Start('CMult_
1324      if (present(a)) then
1325          mult = a
1326      else
1327          mult = ROne
1328      end if
1329      if (present(b)) then
1330          beta = b
1331      else
1332          beta = RZero
1333      end if
1334      if (matrix_method == Mat_
1335          if (beta /= CZero) th
1336          out = mult*MatMul
1337      else
1338          out = MatMul(tran
1339          if (mult /= COne)
1340      end if
1341      else
1342      #ifdef MATRIX_SINGLE
1343          call SGEMM('T','N',m,
1344      #else
1345          call DGEMM('T','N',m,
1346      #endif
1347      end if
1348      call Matrix_End('Mult_TN'
1349
1350      end subroutine Matrix_Mul
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1351

```
1351
1352      subroutine Matrix_Cholesk
1353      !Note upper triangular is
1354      real(dm), intent(inout)::
1355      integer n, info
1356      integer, optional :: err
1357      logical, intent(in), opti
1358      integer i
1359
1360      n=Size(M,DIM=1)
1361      if (Size(M,DIM=2)/=n) cal
1362
1363      #ifdef MATRIX_SINGLE
1364      call spotrf ('L', n, M, n
1365      #else
1366      call dpotrf ('L', n, M, n
1367      #endif
1368
1369      if (present(err)) then
1370        err = info
1371      else
1372        if (info/=0) &
1373        call MpiStop('Matrix_
1374      end if
1375
1376      if (info==0 .and. present
1377        do i=1,n
1378          M(1:i-1,i)=0
1379        end do
1380      end if
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1351

```
1351
1352      subroutine Matrix_Cholesk
1353      !Note upper triangular is
1354      real(dm), intent(inout)::
1355      integer n, info
1356      integer, optional :: err
1357      logical, intent(in), opti
1358      integer i
1359
1360      n=Size(M,DIM=1)
1361      if (Size(M,DIM=2)/=n) cal
1362
1363      #ifdef MATRIX_SINGLE
1364      call spotrf ('L', n, M, n
1365      #else
1366      call dpotrf ('L', n, M, n
1367      #endif
1368
1369      if (present(err)) then
1370        err = info
1371      else
1372        if (info/=0) &
1373        call MpiStop('Mat
1374      end if
1375
1376      if (info==0 .and. present
1377        do i=1,n
1378          M(1:i-1,i)=0
1379        end do
1380      end if
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1381

```
1381
1382     end subroutine Matrix_Cho
1383
1384     subroutine Matrix_CCholes
1385         !M = L L^dag
1386         complex(dm), intent(inout)
1387         integer n, info
1388
1389         n=Size(M,DIM=1)
1390         if (Size(M,DIM=2)/=n) cal
1391
1392         #ifdef MATRIX_SINGLE
1393             call cpotrf ('L', n, M, n
1394         #else
1395             call zpotrf ('L', n, M, n
1396         #endif
1397
1398         if (info/=0) call MpiStop
1399
1400     end subroutine Matrix_CCh
1401
1402
1403     subroutine Matrix_Cholesk
1404         !M = L L^T and return L^{
1405         real(dm), intent(inout)::
1406         integer n, info
1407         integer i,j
1408         logical, intent(in), opti
1409         integer, intent(out), opt
1410         logical trans
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1381

```
1381
1382     end subroutine Matrix_Cho
1383
1384     subroutine Matrix_CCholes
1385         !M = L L^dag
1386         complex(dm), intent(inout)
1387         integer n, info
1388
1389         n=Size(M,DIM=1)
1390         if (Size(M,DIM=2)/=n) cal
1391
1392         #ifdef MATRIX_SINGLE
1393             call cpotrf ('L', n, M, n
1394         #else
1395             call zpotrf ('L', n, M, n
1396         #endif
1397
1398         if (info/=0) call MpiStop
1399
1400     end subroutine Matrix_CCh
1401
1402
1403     subroutine Matrix_Cholesk
1404         !M = L L^T and return L^{
1405         real(dm), intent(inout)::
1406         integer n, info
1407         integer i,j
1408         logical, intent(in), opti
1409         integer, intent(out), opt
1410         logical trans
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1411

```
1411
1412      call Matrix_Cholesky(M, i
1413      if (info==0) then
1414          n=size(M,dim=1)
1415      #ifdef MATRIX_SINGLE
1416          call STRTRI( 'L', 'N'
1417      #else
1418          call DTRTRI( 'L', 'N'
1419      #endif
1420      end if
1421      if (present(error)) error
1422      if (info/=0) then
1423          if (present(error)) r
1424          call MpiStop('Matrix_
1425      end if
1426
1427      if (present(transpose)) t
1428          trans = transpose
1429      else
1430          trans = .false.
1431      end if
1432
1433      if (trans) then
1434
1435      do i=1,n
1436          do j=1,i-1
1437              M(j,i) = M(i,j)
1438              M(i,j) = 0
1439          end do
1440      end do
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1411

```
1411
1412      call Matrix_Cholesky(M, i
1413      if (info==0) then
1414          n=size(M,dim=1)
1415      #ifdef MATRIX_SINGLE
1416          call STRTRI( 'L', 'N'
1417      #else
1418          call DTRTRI( 'L', 'N'
1419      #endif
1420      end if
1421      if (present(error)) error
1422      if (info/=0) then
1423          if (present(error)) r
1424          call MpiStop('Matrix_
1425      end if
1426
1427      if (present(transpose)) t
1428          trans = transpose
1429      else
1430          trans = .false.
1431      end if
1432
1433      if (trans) then
1434
1435      do i=1,n
1436          do j=1,i-1
1437              M(j,i) = M(i,
1438              M(i,j) = 0
1439          end do
1440      end do
```



```
1441
1442     else
1443
1444     do i=1,n
1445         do j=1,i-1
1446             M(j,i) = 0
1447         end do
1448     end do
1449
1450 end if
1451 end subroutine Matrix_Cho
1452
1453 subroutine Matrix_CCholes
1454 !M = L L^dag and return
1455 complex(dm), intent(inout)
1456 integer n, info
1457 integer i,j
1458 logical, intent(in), optional :: trans
1459
1460 call Matrix_CCholesky(M)
1461 n=size(M,dim=1)
1462
1463 #ifdef MATRIX_SINGLE
1464     call CTRTRI( 'L', 'N', n,
1465 #else
1466     call ZTRTRI( 'L', 'N', n,
1467 #endif
1468
1469 if (info/=0) call MpiStop
```

```
1441
1442     else
1443
1444     do i=1,n
1445         do j=1,i-1
1446             M(j,i) = 0
1447         end do
1448     end do
1449
1450 end if
1451 end subroutine Matrix_Cho
1452
1453 subroutine Matrix_CCholes
1454 !M = L L^dag and return
1455 complex(dm), intent(inout)
1456 integer n, info
1457 integer i,j
1458 logical, intent(in), optional :: trans
1459
1460 call Matrix_CCholesky(M)
1461 n=size(M,dim=1)
1462
1463 #ifdef MATRIX_SINGLE
1464     call CTRTRI( 'L', 'N', n,
1465 #else
1466     call ZTRTRI( 'L', 'N', n,
1467 #endif
1468
1469 if (info/=0) call MpiStop
```

/Users/lp1opa/Compare/camb_simdata/Matr
ix_utils.F90, Top line: 1471

```
1471
1472     if (present(dagger)) then
1473         trans = dagger
1474     else
1475         trans = .false.
1476     end if
1477
1478     if (trans) then
1479
1480         do i=1,n
1481             do j=1,i-1
1482                 M(j,i) = conjg(M(
1483                 M(i,j) = 0
1484             end do
1485         end do
1486
1487     else
1488
1489         do i=1,n
1490             do j=1,i-1
1491                 M(j,i) = 0
1492             end do
1493         end do
1494
1495     end if
1496 end subroutine Matrix_CCh
1497
1498
1499 subroutine Matrix_inverse
1500 !Inverse of symmetric mat
```

/Users/lp1opa/Compare/camb_des/Matrix_u
tills.F90, Top line: 1471

```
1471
1472     if (present(dagger)) then
1473         trans = dagger
1474     else
1475         trans = .false.
1476     end if
1477
1478     if (trans) then
1479
1480         do i=1,n
1481             do j=1,i-1
1482                 M(j,i) = conj
1483                 M(i,j) = 0
1484             end do
1485         end do
1486
1487     else
1488
1489         do i=1,n
1490             do j=1,i-1
1491                 M(j,i) = 0
1492             end do
1493         end do
1494
1495     end if
1496 end subroutine Matrix_CCh
1497
1498
1499 subroutine Matrix_inverse
1500 !Inverse of symmetric mat
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1501

```
1501      !This should not be used
1502      real(dm), intent(inout)::
1503      integer i,j,n
1504      integer info
1505      integer, optional :: err
1506
1507      n=Size(M,DIM=1)
1508      if (Size(M,DIM=2)/=n) call
1509      call Matrix_Start('Invers
1510      if (present(err)) then
1511          call Matrix_Cholesky(
1512          if (err/=0) return
1513      else
1514          call Matrix_Cholesky(
1515      end if
1516      #ifdef MATRIX_SINGLE
1517          call spotri ('L', n, M, n
1518      #else
1519          call dpotri ('L', n, M, n
1520      #endif
1521      if (present(err)) then
1522          err = info
1523          if (err/=0) return
1524      else
1525          if (info/=0) call Mpi
1526      end if
1527      do i=1,n
1528          do j=1,i-1
1529              M(j,i) = M(i,j)
1530          end do
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1501

```
1501      !This should not be used
1502      real(dm), intent(inout)::
1503      integer i,j,n
1504      integer info
1505      integer, optional :: err
1506
1507      n=Size(M,DIM=1)
1508      if (Size(M,DIM=2)/=n) call
1509      call Matrix_Start('Invers
1510      if (present(err)) then
1511          call Matrix_Cholesky(
1512          if (err/=0) return
1513      else
1514          call Matrix_Cholesky(
1515      end if
1516      #ifdef MATRIX_SINGLE
1517          call spotri ('L', n, M, n
1518      #else
1519          call dpotri ('L', n, M, n
1520      #endif
1521      if (present(err)) then
1522          err = info
1523          if (err/=0) return
1524      else
1525          if (info/=0) call Mpi
1526      end if
1527      do i=1,n
1528          do j=1,i-1
1529              M(j,i) = M(i,j)
1530          end do
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1531

```
1531 end do
1532 call Matrix_End('Inverse'
1533
1534 end subroutine Matrix_i
1535
1536 subroutine Matrix_Inverse
1537 !Inverse of symmetric pos
1538 real(dm), intent(inout)::
1539 integer i, n
1540
1541 !      real(dm) w(Size(M,D
1542 !      real(dm), dimension
1543 !      real(dm), dimension
1544
1545
1546 n = size(M,DIM=1)
1547 do i=1, size(M,DIM=1)
1548     if (abs(M(i,i)) < 1d-
1549 end do
1550
1551 call Matrix_Inverse_Chol(
1552 !
1553 !      allocate(tmp(Size(M
1554 !
1555 !      n=Size(M,DIM=1)
1556 !      if (n<=1) return
1557 !      if (Size(M,DIM=2)/=
1558 !      call Matrix_Start('
1559 !
1560 !
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1531

```
1531 end do
1532 call Matrix_End('Inverse'
1533
1534 end subroutine Matrix_i
1535
1536 subroutine Matrix_Inverse
1537 !Inverse of symmetric pos
1538 real(dm), intent(inout)::
1539 integer i, n
1540
1541 !      real(dm) w(Size(M,D
1542 !      real(dm), dimension
1543 !      real(dm), dimension
1544
1545
1546 n = size(M,DIM=1)
1547 do i=1, size(M,DIM=1)
1548     if (abs(M(i,i)) < 1d-
1549 end do
1550
1551 call Matrix_Inverse_Chol(
1552 !
1553 !      allocate(tmp(Size(M
1554 !
1555 !      n=Size(M,DIM=1)
1556 !      if (n<=1) return
1557 !      if (Size(M,DIM=2)/=
1558 !      call Matrix_Start('
1559 !
1560 !
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1561

```
1561      !      allocate(norm(n))
1562      !      do i=1, n
1563      !          norm(i) = sqrt(a
1564      !          if (norm(i) < 1d
1565      !          call MpiStop('M
1566      !          M(i,:) = M(i,+)/
1567      !          M(:,i) = M(:,i)/
1568      !      end do
1569      !
1570      !      call Matrix_Diagona
1571      !      write (*,*), 'min/m
1572      !      if (any(w<=0)) then
1573      !          write (*,*), '
1574      !          call MpiStop('
1575      !      end if
1576      !      do i=1, n
1577      !          tmp(i,:) = M(:,i
1578      !      end do
1579      !      allocate(tmp2(Size(
1580      !      call Matrix_Mult(M,
1581      !      M = tmp2
1582      !      do i=1, n
1583      !          M(i,:) = M(i,+)/
1584      !          M(:,i) = M(:,i)/
1585      !      end do
1586      !      deallocate(tmp, tmp
1587      !      deallocate(norm)
1588      !      call Matrix_End('In
1589
1590      end subroutine Matrix_Inv
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1561

```
1561      !      allocate(norm(n))
1562      !      do i=1, n
1563      !          norm(i) = sqrt(a
1564      !          if (norm(i) < 1d
1565      !          call MpiStop('M
1566      !          M(i,:) = M(i,+)/
1567      !          M(:,i) = M(:,i)/
1568      !      end do
1569      !
1570      !      call Matrix_Diagona
1571      !      write (*,*), 'min/m
1572      !      if (any(w<=0)) then
1573      !          write (*,*), '
1574      !          call MpiStop('
1575      !      end if
1576      !      do i=1, n
1577      !          tmp(i,:) = M(:,i
1578      !      end do
1579      !      allocate(tmp2(Size(
1580      !      call Matrix_Mult(M,
1581      !      M = tmp2
1582      !      do i=1, n
1583      !          M(i,:) = M(i,+)/
1584      !          M(:,i) = M(:,i)/
1585      !      end do
1586      !      deallocate(tmp, tmp
1587      !      deallocate(norm)
1588      !      call Matrix_End('In
1589
1590      end subroutine Matrix_Inv
```

```
1591
1592     function Matrix_GaussianL
1593     !Returns -Log Likelihood
1594     !** Cov is destroyed by t
1595     real(dm), intent(inout)::
1596     real(dm), intent(in):: d(
1597     real(dm), allocatable ::
1598     real(dm) :: LogLike
1599     integer info,i,n
1600
1601     call Matrix_Start('Gaussi
1602     n = size(COV,DIM=1)
1603     if (Size(COV,DIM=2)/=n) c
1604     if (Size(d)/=n) call MpiS
1605
1606     call Matrix_Cholesky(Cov)
1607     LogLike = 0
1608     !Log Det term:
1609     do i=1, n
1610         LogLike = LogLike +
1611     end do
1612
1613     !Solve for Cov^{-1}d [cou
1614     allocate(tmp(n))
1615     tmp = d
1616     #ifdef MATRIX_SINGLE
1617         call SPOTRS('L', N, 1, Co
1618     #else
1619         call DPOTRS('L', N, 1, Co
1620     #endif
```

```
1591
1592     function Matrix_GaussianL
1593     !Returns -Log Likelihood
1594     !** Cov is destroyed by t
1595     real(dm), intent(inout)::
1596     real(dm), intent(in):: d(
1597     real(dm), allocatable ::
1598     real(dm) :: LogLike
1599     integer info,i,n
1600
1601     call Matrix_Start('Gaussi
1602     n = size(COV,DIM=1)
1603     if (Size(COV,DIM=2)/=n) c
1604     if (Size(d)/=n) call MpiS
1605
1606     call Matrix_Cholesky(Cov)
1607     LogLike = 0
1608     !Log Det term:
1609     do i=1, n
1610         LogLike = LogLike +
1611     end do
1612
1613     !Solve for Cov^{-1}d [cou
1614     allocate(tmp(n))
1615     tmp = d
1616     #ifdef MATRIX_SINGLE
1617         call SPOTRS('L', N, 1, Co
1618     #else
1619         call DPOTRS('L', N, 1, Co
1620     #endif
```

/Users/lplopa/Compare/camb_simdata/Matr
ix_utils.F90, Top line: 1621

```
1621      if (INFO/=0) call MpiStop
1622
1623      !Add together
1624      LogLike = LogLike + dot_p
1625      deallocate(tmp)
1626
1627      call Matrix_End('Gaussian
1628
1629      end function Matrix_Gauss
1630
1631      function Matrix_GaussianL
1632      !Returns -Log Likelihood
1633      !** Cov is destroyed by t
1634      double precision, intent(
1635      double precision, intent(
1636      double precision, allocat
1637      double precision :: LogLi
1638      integer info,i,n
1639
1640      call Matrix_Start('Gaussi
1641      n = size(COV,DIM=1)
1642      if (Size(COV,DIM=2)/=n) c
1643      if (Size(d)/=n) call MpiS
1644
1645      call dpotrf ('L', n, Cov,
1646      if (info/=0) call MpiStop
1647
1648      LogLike = 0
1649      !Log Det term:
1650      do i=1, n
```

/Users/lplopa/Compare/camb_des/Matrix_u
tils.F90, Top line: 1621

```
1621      if (INFO/=0) call MpiStop
1622
1623      !Add together
1624      LogLike = LogLike + dot_p
1625      deallocate(tmp)
1626
1627      call Matrix_End('Gaussian
1628
1629      end function Matrix_Gauss
1630
1631      function Matrix_GaussianL
1632      !Returns -Log Likelihood
1633      !** Cov is destroyed by t
1634      double precision, intent(
1635      double precision, intent(
1636      double precision, allocat
1637      double precision :: LogLi
1638      integer info,i,n
1639
1640      call Matrix_Start('Gaussi
1641      n = size(COV,DIM=1)
1642      if (Size(COV,DIM=2)/=n) c
1643      if (Size(d)/=n) call MpiS
1644
1645      call dpotrf ('L', n, Cov,
1646      if (info/=0) call MpiStop
1647
1648      LogLike = 0
1649      !Log Det term:
1650      do i=1, n
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1651

```
1651      LogLike = LogLike +
1652      end do
1653
1654      !Solve for Cov^{-1}d [cou
1655      allocate(tmp(n))
1656      tmp = d
1657      call DPOTRS('L', N, 1, Co
1658      if (INFO/=0) call MpiStop
1659
1660      !Add together
1661      LogLike = LogLike + dot_p
1662      deallocate(tmp)
1663
1664      call Matrix_End('Gaussian
1665
1666      end function Matrix_Gauss
1667
1668      subroutine Matrix_Inverse
1669      !This should not be used
1670      real(dm), intent(inout)::
1671      real(dm) w(Size(M,DIM=1))
1672      real(dm), dimension(:, :),
1673      integer i, n
1674
1675      n=Size(M,DIM=1)
1676      if (n<=1) return
1677      if (Size(M,DIM=2)/=n) cal
1678
1679      allocate(tmp(n,n),VT(n,n)
1680
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1651

```
1651      LogLike = LogLike +
1652      end do
1653
1654      !Solve for Cov^{-1}d [cou
1655      allocate(tmp(n))
1656      tmp = d
1657      call DPOTRS('L', N, 1, Co
1658      if (INFO/=0) call MpiStop
1659
1660      !Add together
1661      LogLike = LogLike + dot_p
1662      deallocate(tmp)
1663
1664      call Matrix_End('Gaussian
1665
1666      end function Matrix_Gauss
1667
1668      subroutine Matrix_Inverse
1669      !This should not be used
1670      real(dm), intent(inout)::
1671      real(dm) w(Size(M,DIM=1))
1672      real(dm), dimension(:, :),
1673      integer i, n
1674
1675      n=Size(M,DIM=1)
1676      if (n<=1) return
1677      if (Size(M,DIM=2)/=n) cal
1678
1679      allocate(tmp(n,n),VT(n,n)
1680
```


/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1681

```
1681 call Matrix_SVD(M,n,n,w,V
1682
1683 do i=1, n
1684     tmp(i,:) = M(:,i)/w(i)
1685 end do
1686
1687 call Matrix_Mult_TN(VT,tm
1688 ! M = matmul(transpose(V
1689
1690 deallocate(tmp,VT)
1691
1692 end subroutine Matrix_Inv
1693
1694 subroutine Matrix_SVD(Mat
1695 !Do singular value decomp
1696 !Mat = U D V^T
1697 !returns U in Mat, vector
1698 integer, intent(in) :: m,
1699 real(dm), intent(inout) :
1700 real(dm), intent(out) ::
1701
1702 integer WorkSize, ierr
1703 real(dm), allocatable, di
1704
1705
1706 WorkSize=3*n**2
1707
1708 allocate(rv1(WorkSize))
1709 call Matrix_Start('SVD')
1710 #ifdef MATRIX_SINGLE
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1681

```
1681 call Matrix_SVD(M,n,n,w,V
1682
1683 do i=1, n
1684     tmp(i,:) = M(:,i)/w(i)
1685 end do
1686
1687 call Matrix_Mult_TN(VT,tm
1688 ! M = matmul(transpose(V
1689
1690 deallocate(tmp,VT)
1691
1692 end subroutine Matrix_Inv
1693
1694 subroutine Matrix_SVD(Mat
1695 !Do singular value decomp
1696 !Mat = U D V^T
1697 !returns U in Mat, vector
1698 integer, intent(in) :: m,
1699 real(dm), intent(inout) :
1700 real(dm), intent(out) ::
1701
1702 integer WorkSize, ierr
1703 real(dm), allocatable, di
1704
1705
1706 WorkSize=3*n**2
1707
1708 allocate(rv1(WorkSize))
1709 call Matrix_Start('SVD')
1710 #ifdef MATRIX_SINGLE
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1711

```
1711      call SGESVD('O','A',m,n,  
1712      #else  
1713      call DGESVD('O','A',m,n,  
1714      #endif  
1715      if (ierr/=0) call MpiStop  
1716      call Matrix_End('SVD')  
1717  
1718      deallocate(rv1)  
1719  
1720      end subroutine Matrix_SVD  
1721  
1722      subroutine Matrix_SVD_VT(  
1723      !Do singular value decomp  
1724      !Mat = U D V^dag  
1725      !returns V^dag in Mat, ve  
1726      integer, intent(in) :: m,  
1727      real(dm), intent(inout) :  
1728      real(dm), intent(out),opt  
1729      real(dm), intent(out) ::  
1730  
1731      integer WorkSize, ierr  
1732      integer,allocatable, dime  
1733      real(dm), allocatable, di  
1734      real(dm) OptWk  
1735  
1736      if (n<=m) call MpiStop('M  
1737  
1738      call Matrix_Start('SVD_VT  
1739  
1740      if (present(U) .and. Matr
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1711

```
1711      call SGESVD('O','A',m,n,  
1712      #else  
1713      call DGESVD('O','A',m,n,  
1714      #endif  
1715      if (ierr/=0) call MpiStop  
1716      call Matrix_End('SVD')  
1717  
1718      deallocate(rv1)  
1719  
1720      end subroutine Matrix_SVD  
1721  
1722      subroutine Matrix_SVD_VT(  
1723      !Do singular value decomp  
1724      !Mat = U D V^dag  
1725      !returns V^dag in Mat, ve  
1726      integer, intent(in) :: m,  
1727      real(dm), intent(inout) :  
1728      real(dm), intent(out),opt  
1729      real(dm), intent(out) ::  
1730  
1731      integer WorkSize, ierr  
1732      integer,allocatable, dime  
1733      real(dm), allocatable, di  
1734      real(dm) OptWk  
1735  
1736      if (n<=m) call MpiStop('M  
1737  
1738      call Matrix_Start('SVD_VT  
1739  
1740      if (present(U) .and. Matr
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1741

```
1741      !Use divide and conqu
1742      allocate(IWork(8*MIN(
1743      WorkSize= -1 !3*min(M
1744      #ifdef MATRIX_SINGLE
1745      call SGESDD('O',m,n,
1746      #else
1747      call DGESDD('O',m,n,
1748      #endif
1749      WorkSize = nint(OptWk
1750      allocate(rv1(WorkSize
1751      #ifdef MATRIX_SINGLE
1752      call SGESDD('O',m,n,
1753      #else
1754      call DGESDD('O',m,n,
1755      #endif
1756      deallocate(IWork)
1757      else
1758      call MpiStop('Matrix_
1759      end if
1760
1761      if (ierr/=0) call MpiStop
1762      deallocate(rv1)
1763
1764      call Matrix_End('SVD_VT')
1765
1766      end subroutine Matrix_SVD
1767
1768
1769      subroutine Matrix_CSVD_VT
1770      !Do singular value decomp
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1741

```
1741      !Use divide and conqu
1742      allocate(IWork(8*MIN(
1743      WorkSize= -1 !3*min(M
1744      #ifdef MATRIX_SINGLE
1745      call SGESDD('O',m,n,
1746      #else
1747      call DGESDD('O',m,n,
1748      #endif
1749      WorkSize = nint(OptWk
1750      allocate(rv1(WorkSize
1751      #ifdef MATRIX_SINGLE
1752      call SGESDD('O',m,n,
1753      #else
1754      call DGESDD('O',m,n,
1755      #endif
1756      deallocate(IWork)
1757      else
1758      call MpiStop('Matrix_
1759      end if
1760
1761      if (ierr/=0) call MpiStop
1762      deallocate(rv1)
1763
1764      call Matrix_End('SVD_VT')
1765
1766      end subroutine Matrix_SVD
1767
1768
1769      subroutine Matrix_CSVD_VT
1770      !Do singular value decomp
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1771

```
1771      !Mat = U D V^dag
1772      !returns V^dag in Mat, ve
1773      integer, intent(in) :: m,
1774      complex(dm), intent(inout)
1775      complex(dm), intent(out),
1776      real(dm), intent(out) ::
1777
1778      integer WorkSize, ierr
1779      integer, allocatable, dime
1780      complex(dm), allocatable,
1781      real(dm), allocatable, di
1782
1783      if (n<=m) call MpiStop('M
1784
1785      call Matrix_Start('CSVD_V
1786
1787
1788      if (present(U) .and. Matr
1789          !Use divide and conqu
1790          WorkSize= 2*min(M,N)*
1791          allocate(rv1(WorkSize
1792          allocate(rwork(5*min(
1793          allocate(IWork(8*MIN(
1794      #ifdef MATRIX_SINGLE
1795          call CGESDD('O',m,n,
1796      #else
1797          call ZGESDD('O',m,n,
1798      #endif
1799          deallocate(IWork)
1800      else
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1771

```
1771      !Mat = U D V^dag
1772      !returns V^dag in Mat, ve
1773      integer, intent(in) :: m,
1774      complex(dm), intent(inout)
1775      complex(dm), intent(out),
1776      real(dm), intent(out) ::
1777
1778      integer WorkSize, ierr
1779      integer, allocatable, dime
1780      complex(dm), allocatable,
1781      real(dm), allocatable, di
1782
1783      if (n<=m) call MpiStop('M
1784
1785      call Matrix_Start('CSVD_V
1786
1787
1788      if (present(U) .and. Matr
1789          !Use divide and conqu
1790          WorkSize= 2*min(M,N)*
1791          allocate(rv1(WorkSize
1792          allocate(rwork(5*min(
1793          allocate(IWork(8*MIN(
1794      #ifdef MATRIX_SINGLE
1795          call CGESDD('O',m,n,
1796      #else
1797          call ZGESDD('O',m,n,
1798      #endif
1799          deallocate(IWork)
1800      else
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1801

```
1801
1802      allocate(rwork((max(3*min
1803      WorkSize= 3*max(m,n)**2
1804      allocate(rv1(WorkSize), S
1805      if (ierr /=0) then
1806          WorkSize= MAX(3*MIN(M
1807          allocate(rv1(WorkSize
1808      end if
1809      #ifdef MATRIX_SINGLE
1810          if (present(U)) then
1811              call CGESVD('S','O',m
1812          else
1813              call CGESVD('N','O',m
1814          end if
1815      #else
1816          if (present(U)) then
1817              call ZGESVD('S','O',m
1818          else
1819              call ZGESVD('N','O',m
1820          end if
1821      #endif
1822      end if
1823
1824      if (ierr/=0) call MpiStop
1825      deallocate(rv1)
1826
1827      call Matrix_End('SVD_VT')
1828
1829      end subroutine Matrix_CSV
1830
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1801

```
1801
1802      allocate(rwork((max(3
1803      WorkSize= 3*max(m,n)*
1804      allocate(rv1(WorkSize
1805      if (ierr /=0) then
1806          WorkSize= MAX(3*M
1807          allocate(rv1(Work
1808      end if
1809      #ifdef MATRIX_SINGLE
1810          if (present(U)) then
1811              call CGESVD('S','
1812          else
1813              call CGESVD('N','
1814          end if
1815      #else
1816          if (present(U)) then
1817              call ZGESVD('S','
1818          else
1819              call ZGESVD('N','
1820          end if
1821      #endif
1822      end if
1823
1824      if (ierr/=0) call MpiStop
1825      deallocate(rv1)
1826
1827      call Matrix_End('SVD_VT')
1828
1829      end subroutine Matrix_CSV
1830
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1831

```
1831      subroutine Matrix_CSVD_U(
1832      !Do singular value decomp
1833      !Mat = U D VT
1834      !returns U in Mat, vector
1835      integer, intent(in) :: m,
1836      complex(dm), intent(inout
1837      complex(dm), intent(out),
1838      real(dm), intent(out) ::
1839      integer WorkSize, ierr
1840      integer,allocatable, dime
1841      complex(dm), allocatable,
1842      real(dm), allocatable, di
1843
1844      call Matrix_Start('CSVD_U
1845
1846      if (m<n) call MpiStop('Ma
1847
1848      if (present(VT) .and. Mat
1849      WorkSize= 2*min(M,N)*
1850      allocate(rv1(WorkSize
1851      allocate(rwork(5*min(
1852      allocate(IWork(8*MIN(
1853      #ifdef MATRIX_SINGLE
1854      call CGESDD('O',m,n,
1855      #else
1856      call ZGESDD('O',m,n,
1857      #endif
1858      deallocate(IWork)
1859      else
1860      allocate(rwork((max(3
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1831

```
1831      subroutine Matrix_CSVD_U(
1832      !Do singular value decomp
1833      !Mat = U D VT
1834      !returns U in Mat, vector
1835      integer, intent(in) :: m,
1836      complex(dm), intent(inout
1837      complex(dm), intent(out),
1838      real(dm), intent(out) ::
1839      integer WorkSize, ierr
1840      integer,allocatable, dime
1841      complex(dm), allocatable,
1842      real(dm), allocatable, di
1843
1844      call Matrix_Start('CSVD_U
1845
1846      if (m<n) call MpiStop('Ma
1847
1848      if (present(VT) .and. Mat
1849      WorkSize= 2*min(M,N)*
1850      allocate(rv1(WorkSize
1851      allocate(rwork(5*min(
1852      allocate(IWork(8*MIN(
1853      #ifdef MATRIX_SINGLE
1854      call CGESDD('O',m,n,
1855      #else
1856      call ZGESDD('O',m,n,
1857      #endif
1858      deallocate(IWork)
1859      else
1860      allocate(rwork((max(3
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1861

```
1861
1862      WorkSize= 3*max(m,n)*
1863      allocate(rv1(WorkSize
1864      if (ierr /=0) then
1865          WorkSize= MAX(3*M
1866          allocate(rv1(Work
1867      end if
1868      #ifdef MATRIX_SINGLE
1869          if (present(VT)) then
1870              call CGESVD('O','
1871          else
1872              call CGESVD('O','
1873          end if
1874      #else
1875          if (present(VT)) then
1876              call ZGESVD('O','
1877          else
1878              call ZGESVD('O','
1879          end if
1880      #endif
1881      end if
1882      if (ierr/=0) call MpiStop
1883      call Matrix_End('CSVD_U')
1884
1885      deallocate(rv1,rwork)
1886
1887      end subroutine Matrix_CSV
1888
1889
1890      subroutine Matrix_CSVD_Al
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1861

```
1861
1862      WorkSize= 3*max(m,n)*
1863      allocate(rv1(WorkSize
1864      if (ierr /=0) then
1865          WorkSize= MAX(3*M
1866          allocate(rv1(Work
1867      end if
1868      #ifdef MATRIX_SINGLE
1869          if (present(VT)) then
1870              call CGESVD('O','
1871          else
1872              call CGESVD('O','
1873          end if
1874      #else
1875          if (present(VT)) then
1876              call ZGESVD('O','
1877          else
1878              call ZGESVD('O','
1879          end if
1880      #endif
1881      end if
1882      if (ierr/=0) call MpiStop
1883      call Matrix_End('CSVD_U')
1884
1885      deallocate(rv1,rwork)
1886
1887      end subroutine Matrix_CSV
1888
1889
1890      subroutine Matrix_CSVD_Al
```


/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1891

```
1891      !n>m
1892      !Do singular value decomp
1893      !Mat = U D V^dag
1894      !returns all nxn V^dag in
1895      integer, intent(in) :: m,
1896      complex(dm), intent(inout
1897      complex(dm), intent(out):
1898      real(dm), intent(out) ::
1899
1900      integer WorkSize, ierr
1901      complex(dm), allocatable,
1902      complex(dm), allocatable,
1903      integer, allocatable, dim
1904      real(dm), allocatable, di
1905
1906
1907      call Matrix_Start('CSVD_A
1908
1909      if (Matrix_method == Mat_
1910          !Divide and conquer d
1911          WorkSize= 2*min(M,N)*
1912          allocate(rv1(WorkSize
1913          allocate(rwork(5*min(
1914          allocate(IWork(8*MIN(
1915          allocate(U(m,m) )
1916      #ifdef MATRIX_SINGLE
1917          call CGESDD('A',m,n,
1918      #else
1919          call ZGESDD('A',m,n,
1920      #endif
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1891

```
1891      !n>m
1892      !Do singular value decomp
1893      !Mat = U D V^dag
1894      !returns all nxn V^dag in
1895      integer, intent(in) :: m,
1896      complex(dm), intent(inout
1897      complex(dm), intent(out):
1898      real(dm), intent(out) ::
1899
1900      integer WorkSize, ierr
1901      complex(dm), allocatable,
1902      complex(dm), allocatable,
1903      integer, allocatable, dim
1904      real(dm), allocatable, di
1905
1906
1907      call Matrix_Start('CSVD_A
1908
1909      if (Matrix_method == Mat_
1910          !Divide and conquer d
1911          WorkSize= 2*min(M,N)*
1912          allocate(rv1(WorkSize
1913          allocate(rwork(5*min(
1914          allocate(IWork(8*MIN(
1915          allocate(U(m,m) )
1916      #ifdef MATRIX_SINGLE
1917          call CGESDD('A',m,n,
1918      #else
1919          call ZGESDD('A',m,n,
1920      #endif
```


/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1921

```
1921      deallocate(U)
1922      deallocate(IWork)
1923
1924      else
1925          WorkSize= 2*m*n + 2*
1926          allocate(rwork(5*max(
1927          allocate(rv1(WorkSize
1928          if (ierr /=0) then
1929              WorkSize= MAX(3*M
1930              allocate(rv1(Work
1931          end if
1932      #ifdef MATRIX_SINGLE
1933          call CGESVD('N','A',m
1934      #else
1935          call ZGESVD('N','A',m
1936      #endif
1937      end if
1938
1939      if (ierr/=0) call MpiStop
1940      deallocate(rv1,rwork)
1941
1942      call Matrix_End('CSVD_All
1943
1944
1945      end subroutine Matrix_CSV
1946
1947
1948      subroutine Matrix_DiagPre
1949      ! M -> matmul(diag(D),M)
1950      real(dm), intent(inout) :
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1921

```
1921      deallocate(U)
1922      deallocate(IWork)
1923
1924      else
1925          WorkSize= 2*m*n + 2*
1926          allocate(rwork(5*max(
1927          allocate(rv1(WorkSize
1928          if (ierr /=0) then
1929              WorkSize= MAX(3*M
1930              allocate(rv1(Work
1931          end if
1932      #ifdef MATRIX_SINGLE
1933          call CGESVD('N','A',m
1934      #else
1935          call ZGESVD('N','A',m
1936      #endif
1937      end if
1938
1939      if (ierr/=0) call MpiStop
1940      deallocate(rv1,rwork)
1941
1942      call Matrix_End('CSVD_All
1943
1944
1945      end subroutine Matrix_CSV
1946
1947
1948      subroutine Matrix_DiagPre
1949      ! M -> matmul(diag(D),M)
1950      real(dm), intent(inout) :
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1951

```
1951      real(dm), intent(in) :: D
1952      integer i
1953
1954      if (Size(D) /= Size(M,DIM
1955      do i = 1, size(D)
1956          M(i,:) = M(i,:)*D(i)
1957      end do
1958
1959      end subroutine Matrix_Dia
1960
1961
1962      subroutine Matrix_SolveSy
1963      real(dm), intent(out) ::
1964      real(dm), intent(in):: M(
1965      integer IPIV(size(a)),inf
1966      real(dm), dimension(:, :),
1967      real(dm), dimension(:), a
1968      integer n, WorkSize
1969
1970      n=Size(M,DIM=1)
1971      if (n<=1) return
1972      if (Size(M,DIM=2)/=n) cal
1973      call Matrix_Start('SolveS
1974
1975
1976      WorkSize = n**2
1977      allocate(work(WorkSize))
1978      allocate(tmp(n,n))
1979      tmp = M
1980      #ifdef MATRIX_SINGLE
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1951

```
1951      real(dm), intent(in) :: D
1952      integer i
1953
1954      if (Size(D) /= Size(M,DIM
1955      do i = 1, size(D)
1956          M(i,:) = M(i,:)*D(i)
1957      end do
1958
1959      end subroutine Matrix_Dia
1960
1961
1962      subroutine Matrix_SolveSy
1963      real(dm), intent(out) ::
1964      real(dm), intent(in):: M(
1965      integer IPIV(size(a)),inf
1966      real(dm), dimension(:, :),
1967      real(dm), dimension(:), a
1968      integer n, WorkSize
1969
1970      n=Size(M,DIM=1)
1971      if (n<=1) return
1972      if (Size(M,DIM=2)/=n) cal
1973      call Matrix_Start('SolveS
1974
1975
1976      WorkSize = n**2
1977      allocate(work(WorkSize))
1978      allocate(tmp(n,n))
1979      tmp = M
1980      #ifdef MATRIX_SINGLE
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 1981

```
1981      call SSYTRF('U',n,tmp,n,I
1982      #else
1983      call DSYTRF('U',n,tmp,n,I
1984      #endif
1985      deallocate(work)
1986      if (info/=0) call MpiStop
1987      soln(1:n) = a(1:n)
1988      #ifdef MATRIX_SINGLE
1989      call SSYTRS('U',n,1,tmp,n
1990      #else
1991      call DSYTRS('U',n,1,tmp,n
1992      #endif
1993      if (info/=0) call MpiStop
1994      deallocate(tmp)
1995
1996      call Matrix_End('SolveSym
1997
1998
1999      end subroutine Matrix_Sol
2000
2001
2002      subroutine Matrix_SolveAS
2003      real(dm), intent(out) ::
2004      real(dm), intent(in):: M(
2005      integer IPIV(size(a)),inf
2006      real(dm), dimension(:, :),
2007      integer n
2008
2009      n=Size(M,DIM=1)
2010      if (n<=1) return
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 1981

```
1981      call SSYTRF('U',n,tmp,n,I
1982      #else
1983      call DSYTRF('U',n,tmp,n,I
1984      #endif
1985      deallocate(work)
1986      if (info/=0) call MpiStop
1987      soln(1:n) = a(1:n)
1988      #ifdef MATRIX_SINGLE
1989      call SSYTRS('U',n,1,tmp,n
1990      #else
1991      call DSYTRS('U',n,1,tmp,n
1992      #endif
1993      if (info/=0) call MpiStop
1994      deallocate(tmp)
1995
1996      call Matrix_End('SolveSym
1997
1998
1999      end subroutine Matrix_Sol
2000
2001
2002      subroutine Matrix_SolveAS
2003      real(dm), intent(out) ::
2004      real(dm), intent(in):: M(
2005      integer IPIV(size(a)),inf
2006      real(dm), dimension(:, :),
2007      integer n
2008
2009      n=Size(M,DIM=1)
2010      if (n<=1) return
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 2011

```
2011      if (Size(M,DIM=2)/=n) cal
2012
2013      call Matrix_Start('SolveA
2014
2015      allocate(tmp(n,n))
2016      tmp = M
2017      #ifdef MATRIX_SINGLE
2018      call SGETRF(n,n,tmp,n,IPI
2019      #else
2020      call DGETRF(n,n,tmp,n,IPI
2021      #endif
2022      if (info/=0) call MpiStop
2023      soln(1:n) = a(1:n)
2024      #ifdef MATRIX_SINGLE
2025      call SGETRS('N',n,1,tmp,n
2026      #else
2027      call DGETRS('N',n,1,tmp,n
2028      #endif
2029      if (info/=0) call MpiStop
2030      deallocate(tmp)
2031
2032      call Matrix_End('SolveASy
2033
2034      end subroutine Matrix_Sol
2035
2036      function Matrix_vecdot(ve
2037      real(dm) vec1(:),vec2(:)
2038      real(dm) Matrix_vecdot
2039      integer n
2040      #ifdef MATRIX_SINGLE
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 2011

```
2011      if (Size(M,DIM=2)/=n) cal
2012
2013      call Matrix_Start('SolveA
2014
2015      allocate(tmp(n,n))
2016      tmp = M
2017      #ifdef MATRIX_SINGLE
2018      call SGETRF(n,n,tmp,n,IPI
2019      #else
2020      call DGETRF(n,n,tmp,n,IPI
2021      #endif
2022      if (info/=0) call MpiStop
2023      soln(1:n) = a(1:n)
2024      #ifdef MATRIX_SINGLE
2025      call SGETRS('N',n,1,tmp,n
2026      #else
2027      call DGETRS('N',n,1,tmp,n
2028      #endif
2029      if (info/=0) call MpiStop
2030      deallocate(tmp)
2031
2032      call Matrix_End('SolveASy
2033
2034      end subroutine Matrix_Sol
2035
2036      function Matrix_vecdot(ve
2037      real(dm) vec1(:),vec2(:)
2038      real(dm) Matrix_vecdot
2039      integer n
2040      #ifdef MATRIX_SINGLE
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 2041

```
2041      real(dm) sdot
2042      external sdot
2043      #else
2044      real(dm) ddot
2045      external ddot
2046      #endif
2047      n=size(vec1)
2048      if (n/=size(vec2)) call M
2049      #ifdef MATRIX_SINGLE
2050      Matrix_vecdot = sdot(n, v
2051      #else
2052      Matrix_vecdot = ddot(n, v
2053      #endif
2054      end function Matrix_vecdo
2055
2056      function Matrix_QuadForm(
2057      !Get vec^T*Mat*vec where
2058      real(dm) Matrix_QuadForm
2059      real(dm) vec(:)
2060      real(dm) Mat(:, :)
2061      real(dm), dimension(:), a
2062      integer n
2063
2064      n=size(vec)
2065      allocate(out(n))
2066      call Matrix_MulVecSymm(Ma
2067      Matrix_QuadForm = Matrix_
2068      deallocate(out)
2069
2070      end function Matrix_QuadF
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 2041

```
2041      real(dm) sdot
2042      external sdot
2043      #else
2044      real(dm) ddot
2045      external ddot
2046      #endif
2047      n=size(vec1)
2048      if (n/=size(vec2)) call M
2049      #ifdef MATRIX_SINGLE
2050      Matrix_vecdot = sdot(n, v
2051      #else
2052      Matrix_vecdot = ddot(n, v
2053      #endif
2054      end function Matrix_vecdo
2055
2056      function Matrix_QuadForm(
2057      !Get vec^T*Mat*vec where
2058      real(dm) Matrix_QuadForm
2059      real(dm) vec(:)
2060      real(dm) Mat(:, :)
2061      real(dm), dimension(:), a
2062      integer n
2063
2064      n=size(vec)
2065      allocate(out(n))
2066      call Matrix_MulVecSymm(Ma
2067      Matrix_QuadForm = Matrix_
2068      deallocate(out)
2069
2070      end function Matrix_QuadF
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 2071

```
2071
2072      subroutine Matrix_MulVec(
2073      ! Out = a*Mat*vec + b*out
2074      real(dm), intent(in) :: M
2075      real(dm) vec(:)
2076      real(dm) Out(:)
2077      real(dm), intent(in), opt
2078      real(dm) mult, beta
2079      integer m,n
2080
2081      call Matrix_Start('MulVec
2082
2083      m = Size(Mat,DIM=1)
2084      n = Size(Vec)
2085      if (Size(Mat,DIM=2) /= n)
2086      if (present(a)) then
2087          mult = a
2088      else
2089          mult = ROne
2090      end if
2091      if (present(b)) then
2092          beta = b
2093      else
2094          beta = RZero
2095      end if
2096
2097      if (matrix_method == Mat_
2098          if (beta /= RZero) th
2099          out = a*MatMul(Ma
2100      else
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 2071

```
2071
2072      subroutine Matrix_MulVec(
2073      ! Out = a*Mat*vec + b*out
2074      real(dm), intent(in) :: M
2075      real(dm) vec(:)
2076      real(dm) Out(:)
2077      real(dm), intent(in), opt
2078      real(dm) mult, beta
2079      integer m,n
2080
2081      call Matrix_Start('MulVec
2082
2083      m = Size(Mat,DIM=1)
2084      n = Size(Vec)
2085      if (Size(Mat,DIM=2) /= n)
2086      if (present(a)) then
2087          mult = a
2088      else
2089          mult = ROne
2090      end if
2091      if (present(b)) then
2092          beta = b
2093      else
2094          beta = RZero
2095      end if
2096
2097      if (matrix_method == Mat_
2098          if (beta /= RZero) th
2099          out = a*MatMul(Ma
2100      else
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 2101

```
2101          out = MatMul(Mat,  
2102          if (mult /= ROne)  
2103          end if  
2104      else  
2105      #ifdef MATRIX_SINGLE  
2106          call SGEMV('N',m,n,mu  
2107      #else  
2108          call DGEMV('N',m,n,mu  
2109      #endif  
2110      end if  
2111      call Matrix_End('MulVec')  
2112  
2113      end subroutine Matrix_Mul  
2114  
2115      subroutine Matrix_MulVecS  
2116      ! Out = a*Mat*vec + b*out  
2117      real, intent(in) :: Mat(:  
2118      real vec(:)  
2119      real Out(:)  
2120      real, intent(in), optional  
2121      real mult, beta  
2122      integer m,n  
2123  
2124      call Matrix_Start('MulVec  
2125  
2126      m = Size(Mat,DIM=1)  
2127      n = Size(Vec)  
2128      if (Size(Mat,DIM=2) /= n)  
2129      if (present(a)) then  
2130          mult = a
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 2101

```
2101          out = MatMul(Mat,  
2102          if (mult /= ROne)  
2103          end if  
2104      else  
2105      #ifdef MATRIX_SINGLE  
2106          call SGEMV('N',m,n,mu  
2107      #else  
2108          call DGEMV('N',m,n,mu  
2109      #endif  
2110      end if  
2111      call Matrix_End('MulVec')  
2112  
2113      end subroutine Matrix_Mul  
2114  
2115      subroutine Matrix_MulVecS  
2116      ! Out = a*Mat*vec + b*out  
2117      real, intent(in) :: Mat(:  
2118      real vec(:)  
2119      real Out(:)  
2120      real, intent(in), optional  
2121      real mult, beta  
2122      integer m,n  
2123  
2124      call Matrix_Start('MulVec  
2125  
2126      m = Size(Mat,DIM=1)  
2127      n = Size(Vec)  
2128      if (Size(Mat,DIM=2) /= n)  
2129      if (present(a)) then  
2130          mult = a
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 2131

```
2131      else
2132          mult = SOne
2133      end if
2134      if (present(b)) then
2135          beta = b
2136      else
2137          beta = SZero
2138      end if
2139
2140      if (matrix_method == Mat_
2141          if (beta /= SZero) th
2142          out = a*MatMul(Ma
2143          else
2144              out = MatMul(Mat,
2145              if (mult /= SOne)
2146          end if
2147      else
2148          call SGEMV('N',m,n,mu
2149      end if
2150      call Matrix_End('MulVecSi
2151
2152      end subroutine Matrix_Mul
2153
2154
2155
2156
2157      subroutine Matrix_MulVecS
2158      ! Out = a*Mat*vec + b*out
2159      real(dm), intent(in) :: M
2160      real(dm) vec(:)
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 2131

```
2131      else
2132          mult = SOne
2133      end if
2134      if (present(b)) then
2135          beta = b
2136      else
2137          beta = SZero
2138      end if
2139
2140      if (matrix_method == Mat_
2141          if (beta /= SZero) th
2142          out = a*MatMul(Ma
2143          else
2144              out = MatMul(Mat,
2145              if (mult /= SOne)
2146          end if
2147      else
2148          call SGEMV('N',m,n,mu
2149      end if
2150      call Matrix_End('MulVecSi
2151
2152      end subroutine Matrix_Mul
2153
2154
2155
2156
2157      subroutine Matrix_MulVecS
2158      ! Out = a*Mat*vec + b*out
2159      real(dm), intent(in) :: M
2160      real(dm) vec(:)
```


/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 2161

```
2161 real(dm) Out(:)
2162 real(dm), intent(in), opt
2163 real(dm) mult, beta
2164 integer m,n
2165
2166 call Matrix_Start('MulVec
2167
2168 m = Size(Mat,DIM=1)
2169 n = Size(Vec)
2170 if (m /= n) call MpiStop(
2171 if (present(a)) then
2172     mult = a
2173 else
2174     mult = ROne
2175 end if
2176 if (present(b)) then
2177     beta = b
2178 else
2179     beta = RZero
2180 end if
2181
2182 if (matrix_method == Mat_
2183     if (beta /= RZero) th
2184         out = a*MatMul(Ma
2185     else
2186         out = MatMul(Mat,
2187         if (mult /= ROne)
2188     end if
2189 else
2190 #ifdef MATRIX_SINGLE
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 2161

```
2161 real(dm) Out(:)
2162 real(dm), intent(in), opt
2163 real(dm) mult, beta
2164 integer m,n
2165
2166 call Matrix_Start('MulVec
2167
2168 m = Size(Mat,DIM=1)
2169 n = Size(Vec)
2170 if (m /= n) call MpiStop(
2171 if (present(a)) then
2172     mult = a
2173 else
2174     mult = ROne
2175 end if
2176 if (present(b)) then
2177     beta = b
2178 else
2179     beta = RZero
2180 end if
2181
2182 if (matrix_method == Mat_
2183     if (beta /= RZero) th
2184         out = a*MatMul(Ma
2185     else
2186         out = MatMul(Mat,
2187         if (mult /= ROne)
2188     end if
2189 else
2190 #ifdef MATRIX_SINGLE
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 2191

```
2191      call SSYMV('U',m,mult
2192      #else
2193      call DSYMV('U',m,mult
2194      #endif
2195      end if
2196      call Matrix_End('MulVecSy
2197
2198      end subroutine Matrix_Mul
2199
2200      subroutine Matrix_MulVecS
2201      ! Out = a*Mat*vec + b*out
2202      real, intent(in) :: Mat(:
2203      real vec(:)
2204      real Out(:)
2205      real, intent(in), optiona
2206      real mult, beta
2207      integer m,n
2208
2209      call Matrix_Start('MulVec
2210
2211      m = Size(Mat,DIM=1)
2212      n = Size(Vec)
2213      if (m /= n) call MpiStop(
2214      if (present(a)) then
2215          mult = a
2216      else
2217          mult = SOne
2218      end if
2219      if (present(b)) then
2220          beta = b
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 2191

```
2191      call SSYMV('U',m,mult
2192      #else
2193      call DSYMV('U',m,mult
2194      #endif
2195      end if
2196      call Matrix_End('MulVecSy
2197
2198      end subroutine Matrix_Mul
2199
2200      subroutine Matrix_MulVecS
2201      ! Out = a*Mat*vec + b*out
2202      real, intent(in) :: Mat(:
2203      real vec(:)
2204      real Out(:)
2205      real, intent(in), optiona
2206      real mult, beta
2207      integer m,n
2208
2209      call Matrix_Start('MulVec
2210
2211      m = Size(Mat,DIM=1)
2212      n = Size(Vec)
2213      if (m /= n) call MpiStop(
2214      if (present(a)) then
2215          mult = a
2216      else
2217          mult = SOne
2218      end if
2219      if (present(b)) then
2220          beta = b
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 2221

```
2221      else
2222          beta = SZero
2223      end if
2224
2225      if (matrix_method == Mat_
2226          if (beta /= RZero) th
2227              out = a*MatMul(Ma
2228          else
2229              out = MatMul(Mat,
2230              if (mult /= ROne)
2231          end if
2232      else
2233          call SSYMV('U',m,mult
2234      end if
2235      call Matrix_End('MulVecSy
2236
2237      end subroutine Matrix_Mul
2238
2239      function Matrix_vecdotSin
2240      real vec1(:),vec2(:)
2241      real Matrix_vecdotSingle
2242      integer n
2243      real sdot
2244      external sdot
2245
2246      n=size(vec1)
2247      if (n/=size(vec2)) call M
2248      Matrix_vecdotSingle = sdo
2249
2250      end function Matrix_vecdo
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 2221

```
2221      else
2222          beta = SZero
2223      end if
2224
2225      if (matrix_method == Mat_
2226          if (beta /= RZero) th
2227              out = a*MatMul(Ma
2228          else
2229              out = MatMul(Mat,
2230              if (mult /= ROne)
2231          end if
2232      else
2233          call SSYMV('U',m,mult
2234      end if
2235      call Matrix_End('MulVecSy
2236
2237      end subroutine Matrix_Mul
2238
2239      function Matrix_vecdotSin
2240      real vec1(:),vec2(:)
2241      real Matrix_vecdotSingle
2242      integer n
2243      real sdot
2244      external sdot
2245
2246      n=size(vec1)
2247      if (n/=size(vec2)) call M
2248      Matrix_vecdotSingle = sdo
2249
2250      end function Matrix_vecdo
```

```
2251
2252
2253      subroutine Matrix_Inverse
2254      !Invert array of matrices
2255      integer, intent(in) :: nm
2256      #ifdef __GFORTRAN__
2257          Type(TMATRIXType), target
2258      #else
2259          Type(TMATRIXType), target
2260      #endif
2261          Type(TMATRIXType), pointer
2262          integer n
2263          integer i,MpiID, MpiSize
2264          integer sz
2265      #ifdef MPI
2266          integer j, ierr, sid
2267          Type(TMATRIXType), target
2268      #endif
2269
2270      call MpiStat(MpiID, MpiSi
2271      if (MpiID==0) then
2272          n=nm
2273          sz = Size(Arr(1)%M,DI
2274      end if
2275      !      if (MpiID==0) then
2276      !          do i=1,nmat
2277      !              print *, 'inverting
2278      !              call Matrix_invers
2279      !          end do
2280      !      end if
```

```
2251
2252
2253      subroutine Matrix_Inverse
2254      !Invert array of matrices
2255      integer, intent(in) :: nm
2256      #ifdef __GFORTRAN__
2257          Type(TMATRIXType), target
2258      #else
2259          Type(TMATRIXType), target
2260      #endif
2261          Type(TMATRIXType), pointer
2262          integer n
2263          integer i,MpiID, MpiSize
2264          integer sz
2265      #ifdef MPI
2266          integer j, ierr, sid
2267          Type(TMATRIXType), target
2268      #endif
2269
2270      call MpiStat(MpiID, MpiSi
2271      if (MpiID==0) then
2272          n=nm
2273          sz = Size(Arr(1)%M,DI
2274      end if
2275      !      if (MpiID==0) then
2276      !          do i=1,nmat
2277      !              print *, 'inverting
2278      !              call Matrix_invers
2279      !          end do
2280      !      end if
```

/Users/lplop/Compare/camb_simdata/Matr
ix_utils.F90, Top line: 2281

```
2281      !      return
2282      #ifdef MPI
2283          if (MpiID==0) print *, 'M
2284      call MPI_BCAST(n,1,MPI_IN
2285      call MPI_BCAST(sz,1,MPI_I
2286      if (MpiID/=0) then
2287          allocate(tmp%M(sz,sz)
2288          AM => tmp
2289      end if
2290  #endif
2291
2292      do i= 1,n
2293          if (MpiID==0) AM => A
2294  #ifdef MPI
2295          if (mod(i,MpiSize)/=M
2296              !Do nothing
2297              if (MpiId==0) the
2298                  j=mod(i,MpiSi
2299      call MPI_SEND
2300      end if
2301      else
2302          if (MpiId/=0) the
2303              !Get from mai
2304      call MPI_RECV
2305
2306      end if
2307  #endif
2308      call Matrix_Inver
2309
2310  #ifdef MPI
```

/Users/lplop/Compare/camb_des/Matrix_u
tils.F90, Top line: 2281

```
2281      !      return
2282      #ifdef MPI
2283          if (MpiID==0) print *, 'M
2284      call MPI_BCAST(n,1,MPI_IN
2285      call MPI_BCAST(sz,1,MPI_I
2286      if (MpiID/=0) then
2287          allocate(tmp%M(sz,sz)
2288          AM => tmp
2289      end if
2290  #endif
2291
2292      do i= 1,n
2293          if (MpiID==0) AM => A
2294  #ifdef MPI
2295          if (mod(i,MpiSize)/=M
2296              !Do nothing
2297              if (MpiId==0) the
2298                  j=mod(i,MpiSi
2299      call MPI_SEND
2300      end if
2301      else
2302          if (MpiId/=0) the
2303              !Get from mai
2304      call MPI_RECV
2305
2306      end if
2307  #endif
2308      call Matrix_Inver
2309
2310  #ifdef MPI
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 2311

```
2311         if (MpiID==0) then
2312             do j = max(1,
2313                 sid = mod
2314                 call MPI_
2315             end do
2316         else
2317             call MPI_SEND
2318         end if
2319
2320     end if
2321 #endif
2322 end do
2323
2324
2325 #ifdef MPI
2326     if (MpiID==0) then
2327         do j=n - mod(n,MpiSiz
2328             sid= mod(j,MpiSiz
2329             call MPI_RECV(ARr
2330         end do
2331     else
2332         deallocate(tmp%M)
2333     end if
2334 #endif
2335     if (MpiID==0) print *, 'M
2336
2337
2338 end subroutine Matrix_Inv
2339
2340
```

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 2311

```
2311         if (MpiID==0) then
2312             do j = max(1,
2313                 sid = mod
2314                 call MPI_
2315             end do
2316         else
2317             call MPI_SEND
2318         end if
2319
2320     end if
2321 #endif
2322 end do
2323
2324
2325 #ifdef MPI
2326     if (MpiID==0) then
2327         do j=n - mod(n,MpiSiz
2328             sid= mod(j,MpiSiz
2329             call MPI_RECV(ARr
2330         end do
2331     else
2332         deallocate(tmp%M)
2333     end if
2334 #endif
2335     if (MpiID==0) print *, 'M
2336
2337
2338 end subroutine Matrix_Inv
2339
2340
```

/Users/lp1opa/Compare/camb_simdata/Matrix_utils.F90, Top line: 2341

2341	
2342	end module MatrixUtils
2343	

/Users/lp1opa/Compare/camb_des/Matrix_utils.F90, Top line: 2341

2341	
2342	end module MatrixUtils
2343	