|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 1</pre> |      | /lplopa/Compare/camb_des/Matrix_u<br>'90, Top line: 1 |
|------|--|------|---|
| 0001 | !Matrix utility routines.  | 0001 | !Matrix utility routines.                             |
| 0002 | !Generally (but not alway  | 0002 | !Generally (but not alway                             |
| 0003 | !Not complete  | 0003 | !Not complete   |
| 0004 | !Antony Lewis May 2003-20  | 0004 | !Antony Lewis May 2003-20                             |
| 0005 | !http://cosmologist.info/  | 0005 | !http://cosmologist.info/                             |
| 0006 |  | 0006 |   |
| 0007 |  | 0007 |   |
| 8000 | module MatrixUtils   | 8000 | module MatrixUtils                                    |
| 0009 | use AMLutils   | 0009 | use AMLutils  |
| 0010 | implicit none  | 0010 | implicit none   |
| 0011 | _  | 0011 | _   |
| 0012 | logical, parameter :: Mat  | 0012 | logical, parameter :: Mat                             |
| 0013 | #ifdef MATRIX SINGLE   | 0013 | #ifdef MATRIX SINGLE                                  |
| 0014 | integer, $\overline{p}$ arameter :: dm                           | 0014 | integer, parameter :: dm                              |
| 0015 | #else  | 0015 | #else   |
| 0016 | integer, parameter :: dm   | 0016 | integer, parameter :: dm                              |
| 0017 | #endif   | 0017 | #endif  |
| 0018 | !Precision of matrix oper  | 0018 | !Precision of matrix oper                             |
| 0019 | !If changing also need to  | 0019 | !If changing also need to                             |
| 0020 | integer, parameter :: Mat  | 0020 | integer, parameter :: Mat                             |
| 0021 | integer, parameter :: mat  | 0021 | integer, parameter :: mat                             |
| 0022 |  | 0022 |   |
| 0023 | real Matrix_StartTime  | 0023 | real Matrix_StartTime                                 |
| 0024 | _  | 0024 |   |
| 0025 | Type TMatrixType   | 0025 | Type TMatrixType                                      |
| 0026 | <pre>real(dm), dimension(:</pre>                                 | 0026 | real(dm), dimension(:                                 |
| 0027 | end Type TMatrixType   | 0027 | end Type TMatrixType                                  |
| 0028 |  | 0028 |   |
| 0029 | <pre>complex(dm), parameter ::</pre>                             | 0029 | <pre>complex(dm), parameter ::</pre>                  |
| 0030 | real(dm), parameter :: RO  | 0030 | real(dm), parameter :: RO                             |

|        | <pre>/lplopa/Compare/camb_simdata/Matr</pre> | /Users | <pre>/lplopa/Compare/camb_des/Matrix_u</pre> |
|--------|--|--------|--|
| ix_uti | ls.F90, Top line: 31                         | tils.F | 90, Top line: 31                             |
| 0031   | real, parameter :: SOne =                    | 0031   | real, parameter :: SOne =                    |
| 0032   | · <del>-</del>                               | 0032   | · <del>-</del>                               |
| 0033   | contains                                     | 0033   | contains                                     |
| 0034   |  | 0034   |  |
| 0035   |  | 0035   |  |
| 0036   | <pre>function GetMatrixTime()</pre>          | 0036   | <pre>function GetMatrixTime()</pre>          |
| 0037   | real GetMatrixTime                           | 0037   | real GetMatrixTime                           |
| 0038   | real atime                                   | 0038   | real atime                                   |
| 0039   |  | 0039   |  |
| 0040   | call cpu time(atime)                         | 0040   | call cpu_time(atime)                         |
| 0041   | ` ,  | 0041   | ` ´  |
| 0042   | GetMatrixTime = atime                        | 0042   | <pre>GetMatrixTime = atime</pre>             |
| 0043   |  | 0043   |  |
| 0044   |  | 0044   |  |
| 0045   | end function GetMatrixTim                    | 0045   | end function GetMatrixTim                    |
| 0046   |  | 0046   |  |
| 0047   | subroutine Matrix start(                     | 0047   | subroutine Matrix_start(                     |
| 0048   | character(LEN=*), $\overline{i}$ ntent(      | 0048   | character(LEN=*), $\overline{i}$ ntent(      |
| 0049   |  | 0049   |  |
| 0050   | if (Matrix runmsgs) then                     | 0050   | if (Matrix_runmsgs) then                     |
| 0051   | Matrix StartTime = Ge                        | 0051   | Matrix StartTime = Ge                        |
| 0052   | $Write(\overline{*},*)$ 'Matrix_'/           | 0052   | Write( $\overline{*}$ ,*) 'Matrix '/         |
| 0053   | end if                                       | 0053   | end if                                       |
| 0054   | end subroutine Matrix st                     | 0054   | end subroutine Matrix st                     |
| 0055   | _  | 0055   | _  |
| 0056   | subroutine Matrix end(Na                     | 0056   | subroutine Matrix end(Na                     |
| 0057   | character(LEN=*), $\overline{i}$ ntent(      | 0057   | character(LEN=*), $\overline{i}$ ntent(      |
| 0058   |  | 0058   |  |
| 0059   | if (Matrix runmsgs) then                     | 0059   | if (Matrix runmsgs) then                     |
| 0060   | $\mathbf{Write}(\mathbf{x},*)$ 'Matrix_'/    | 0060   | $\mathbf{Write}(\mathbf{x},*)$ 'Matrix_'/    |
|        |  |        |  |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 61</pre> |      | <pre>/lplopa/Compare/camb_des/Matrix_u 90, Top line: 61</pre> |
|------|---|------|---|
| 0061 | end if  | 0061 | end if  |
| 0062 | end subroutine Matrix en  | 0062 | end subroutine Matrix en                                      |
| 0063 |   | 0063 |   |
| 0064 | subroutine Matrix WriteFi   | 0064 | subroutine Matrix WriteFi                                     |
| 0065 | integer, intent( $i\overline{n}$ ) :: au                          | 0065 | integer, intent( $\overline{in}$ ) :: au                      |
| 0066 | integer, intent(in) :: n  | 0066 | integer, intent(in) :: n                                      |
| 0067 | real(dm) :: vec(n)  | 0067 | real(dm) :: vec(n)  |
| 0068 | charàctér(LEN=50) fmt   | 0068 | charàctér(LEN=50) fmt   |
| 0069 |   | 0069 | · · · · · · · · · · · · · · · · · · ·                         |
| 0070 | <b>fmt</b> = trim(numcat('(',n))                                  | 0070 | <pre>fmt = trim(numcat('(',n))</pre>                          |
| 0071 | write (aunit, fmt) vec(1:   | 0071 | write (aunit, fmt) vec(1:                                     |
| 0072 |   | 0072 |   |
| 0073 | end subroutine Matrix Wri   | 0073 | end subroutine Matrix Wri                                     |
| 0074 | <del>-</del>  | 0074 | _   |
| 0075 | subroutine Matrix Write(a   | 0075 | subroutine Matrix Write(a                                     |
| 0076 | character(LEN=*), intent(   | 0076 | character(LEN=*), intent(                                     |
| 0077 | <pre>character(LEN=*), intent(</pre>                              | 0077 | <pre>character(LEN=*), intent(</pre>                          |
| 0078 | real(dm), intent(in) :: m   | 0078 | real(dm), intent(in) :: m                                     |
| 0079 | logical, intent(in), opti   | 0079 | logical, intent(in), opti                                     |
| 0800 | integer i,k   | 0800 | integer i,k   |
| 0081 | character(LEN=50) fmt   | 0081 | character(LEN=50) fmt   |
| 0082 | integer shp(2)  | 0082 | integer shp(2)  |
| 0083 | logical WriteTab  | 0083 | logical WriteTab  |
| 0084 | integer file unit   | 0084 | integer file_unit   |
| 0085 |   | 0085 | _   |
| 0086 | <pre>shp = shape(mat)</pre>                                       | 0086 | <pre>shp = shape(mat)</pre>                                   |
| 0087 | WriteTab = $shp(2) <= 50$   | 0087 | WriteTab = shp(2) <= 50                                       |
| 8800 | <pre>if (present(forcetable))</pre>                               | 8800 | <pre>if (present(forcetable))</pre>                           |
| 0089 | if (forcetable) Write   | 0089 | if (forcetable) Write   |
| 0090 | end if  | 0090 | end if  |

|  |      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 91</pre> |      | s/lplopa/Compare/camb_des/Matrix_u<br>'90, Top line: 91 |
|--|------|---|------|---|
| Call CreateTxtFile(aname, 0092   if (present(commentline)) 0093   if (present(commentline)) 0093   if (present(commentline)) 0094   write(file_unit,'(a)' 0095   end if 0096   fmt = trim(numcat('(', shp 0096   fmt = trim(numcat('(', shp 0096   fmt = trim(numcat('(', shp 0097   do i=1, shp(1)  | 0091 | file unit = new file unit   | 0091 | file unit = new file unit                               |
| 0093   |      | <u> </u>  |      |   |
| 0094         write(file_unit,'(a)' 0094         write(file_unit,'(a)' 0095           0095         end if 0095         end if 0095           0096         fmt = trim(numcat('(',shp 0096         fmt = trim(numcat('(',shp 0097           0097         do i=1, shp(1) 0097         do i=1, shp(2) 0099           0099         do k=1, shp(2) 0099         do k=1, shp(2)           0100         write (file_u 0100         write (file_u end 000           0101         end do 0101         end do 0102           0103         write (file_unit, 0103         write (file_unit, 0103           0104         end if 0104         write (file_unit, 0103           0105         end do 0105         end do 0105           0106         0106         end do 0105           0107         call CloseFile(file_unit) 0107         call CloseFile(file_unit) 0107           0108         end subroutine Matrix_Wri do 0109         end subroutine Matrix_Wri do 0110           0110         subroutine Matrix_Write do 0111         subroutine Matrix_Write do 0111           0112         character(LEN=*), intent( 0112         character(LEN=*), intent( 0113           0114         logical, intent(in), opti integer i,k         integer i,k           0115         character(LEN=50) fmt integer shp(2)         integer shp(2)   | 0093 |   | 0093 | i ·   |
| 0095         end if<br>fmt = trim(numcat('(',shp 0096<br>do i=1, shp(1) 0097<br>if (.not. WriteTab) t 0098<br>do k=1, shp(2) 0099<br>do k=1, shp(2) 0099<br>write (file_u 0100<br>end do 0101<br>end do 0102<br>else 0102<br>0103<br>0104<br>0105<br>end do 0105<br>end do 0105<br>0106<br>0107<br>0108<br>0109<br>0109<br>0100         if (.not. WriteTab) t if (.not. WriteTab) t do k=1, shp(2)<br>write (file_u end do 0101<br>end do 0102<br>else 0102<br>write (file_unit, 0103<br>end if 0104<br>end if 0104<br>end do 0105<br>end do 0106<br>0106<br>0107<br>0108<br>0109<br>0100         write (file_unit, end if end do 0104<br>end do 0105<br>end do 0106<br>0107<br>0108<br>0109<br>end subroutine Matrix_Wri 0109<br>end subroutine Matrix_Wri 0109<br>end subroutine Matrix_Wri 0109<br>end subroutine Matrix_Wri 0110<br>0111<br>subroutine Matrix_Write_d 0111<br>character(LEN=*), intent( 0112<br>double precision, intent( 0113<br>double precision, intent( 0113<br>double precision, intent( 0114<br>logical, intent(in), opti 0114<br>logical, intent(in), opti integer i,k<br>character(LEN=50) fmt 0116<br>character(LEN=50) fmt integer shp(2)<br>0118 logical WriteTab 0118<br>0119 integer file_unit         end if<br>fmt = trim(numcat('(',shp do i=1, shp(1))<br>if (.not. WriteTab) tif (.not. WriteTab) if mate = trim(numcat('(',shp do i=1, shp(1))<br>if (.not. WriteTab) tif end integer i,k character(LEN=50) fmt integer shp(2)  | 0094 | ` <del>-</del>  |      |   |
| 0097         do i=1, shp(1)         0097         do i=1, shp(1)         1 if (.not. WriteTab) t 0098         if (.not. WriteTab) t if (.not. WriteTab) tif (.not. WriteTab) tif (.not. WriteTab) tif (.not. WriteTab) tif (.not. WriteTab) if (.not. WriteTab) if (.not. WriteTab) if (.not. WriteTab) if (.not. WriteTab) integer i, write (.not. W   | 0095 |   | 0095 | •   |
| 0097         do i=1, shp(1)         0097         do i=1, shp(1)         if (.not. WriteTab) t 0098         if (.not. WriteTab) t if (.not. WriteTab) t do k=1, shp(2)         0099         do k=1, shp(1)         if (.not. WriteTab) t do k=1, shp(2)         do k=1, shp(2)         write(file_u         0100         write (file_u         0100         write (file_u         write (file_u         write (file_u         end do         end do         else         write (file_unit, end if         end do         end if         e   | 0096 | <pre>fmt = trim(numcat('(',shp</pre>                              | 0096 | <pre>fmt = trim(numcat('(',shp</pre>                    |
| 0099         do k=1, shp(2)' 0099         do k=1, shp(2)' write (file_u           0100         write (file_u         0100         write (file_u           0101         end do         0101         end do           0102         else         0102         write (file_unit, 0103         write (file_unit, 0103           0104         end if         0104         end if         end if           0105         end do         0105         end do         0106           0107         call CloseFile(file_unit)         0107         call CloseFile(file_unit)         0107           0108         end subroutine Matrix_Wri         0109         end subroutine Matrix_Wri           0110         0110         subroutine Matrix_Wri         0110           0111         subroutine Matrix_Write_d         0111         subroutine Matrix_Write_d           0112         character(LEN=*), intent(         0112         character(LEN=*), intent(           0113         double precision, intent(         0113         double precision, intent(           0114         logical, intent(in), opti         integer i,k         character(LEN=50) fmt           0116         character(LEN=50) fmt         integer shp(2)         integer shp(2)           0118         integer file_unit </td <td>0097</td> <td>•</td> <td>0097</td> <td>· · · · · · · · · · · · · · · · · · ·</td>   | 0097 | •   | 0097 | · · · · · · · · · · · · · · · · · · ·                   |
| 0099         do k=1, shp(2)         0099         do k=1, shp(2)           0100         write (file_u         0100         write (file_u           0101         end do         0101         end do           0102         else         0102         else           0103         write (file_unit, 0103         write (file_unit, end if           0104         end if         0104         end if           0105         end do         0105         end do           0106         0106         call CloseFile(file_unit)         0107           0108         call CloseFile(file_unit)         0108           0109         end subroutine Matrix_Wri 0109         end subroutine Matrix_Wri           0110         0110         subroutine Matrix_Wri         0109           0111         subroutine Matrix_Wri         0110         character(LEN=*), intent(         0112           0113         double precision, intent(         0113         double precision, intent(         0113         double precision, intent(           0114         logical, intent(in), opti 0114         logical, intent(in), opti integer i,k         integer i,k         character(LEN=50) fmt           0116         character(LEN=50) fmt         0117         integer shp(2)         integer sh   | 0098 | <pre>if (.not. WriteTab) t</pre>                                  | 0098 | <pre>if (.not. WriteTab) t</pre>                        |
| 0101         end do         0101         end do         0102         else         0102         0103         end do         0102         else         0103         write (file_unit, 0103         write (file_unit, end if         0104         end if         0104         end if         end if         end if         end do         0105         end do         0106         end do         0106         0106         0106         0106         0107         call CloseFile(file_unit) 0107         call CloseFile(file_unit) 0108         call CloseFile(file_unit) 0108         end subroutine Matrix_Wri         0108         end subroutine Matrix_Wri         0109         end subroutine Matrix_Wri         0109         end subroutine Matrix_Wri         0110         subroutine Matrix_Wri         0110         0110         subroutine Matrix_Wri         0110         character(LEN=*), intent(         0112         character(LEN=*), intent(         0112         character(LEN=*), intent(         0113         double precision, intent(         0113         double precision, intent(         10114         logical, intent(in), opti integer i,k         0115         integer i,k         0115         integer shp(2)         0117         integer shp(2)         0117         integer shp(2)         0118         logical WriteTab         0119         integer file_unit         0119         integer file_unit   | 0099 | ·   | 0099 | ,   |
| olice of the control  | 0100 | · · · · · · · · · · · · · · · · · · ·                             | 0100 | 1 = 1, 1  |
| 0103         write (file_unit, 0103 end if 0104 end if 0105 end do 0105 end do 0106 0106 0106 0107 call CloseFile(file_unit) 0107 0108 end subroutine Matrix_Wri 0109 0110 0110 0110 0110 0110 0110 character(LEN=*), intent( 0112 character(LEN=*), intent( 0113 double precision, intent( 0113 double precision, intent( 0114 logical, intent(in), opti 0114 integer i,k 0115 character(LEN=50) fmt 0116 character(LEN=50) fmt 0117 integer shp(2) 0117 integer file_unit 0119 integer file_unit 0119 integer file_unit         write (file_unit, end if end if end do 016           0105         end subroutine Matrix_Write_0         call CloseFile(file_unit)           0108         end subroutine Matrix_Write_0           0110         subroutine Matrix_Write_d           0111         subroutine Matrix_Write_d           0112         character(LEN=*), intent( 0112 character(LEN=*), intent( 0113 double precision, intent( 1000 character(LEN=*), intent( 1000 character(L  | 0101 | end do  | 0101 | end do  |
| one of the proof o | 0102 | else  | 0102 | else  |
| on one of the control | 0103 | write (file unit,   | 0103 | write (file unit,                                       |
| O106 O107 O108 O109 O110 O110 O111 O111 O112 O112 O113 O108 O109 O114 O115 O115 O116 O116 O116 O117 O117 O118 O117 O118 O118 O118 O118   | 0104 | _   | 0104 | end if  |
| 0107<br>0108call CloseFile(file_unit)0107<br>0108call CloseFile(file_unit)0109<br>0110end subroutine Matrix_Wri<br>01100109<br>0110end subroutine Matrix_Wri<br>01100111<br>0112<br>0113<br>0114<br>0115<br>0115<br>0116subroutine Matrix_Write_d<br>0110oll1<br>0112<br>0113<br>0114<br>0115<br>0115<br>0116<br>0116<br>0117<br>0118<br>0118<br>0119subroutine Matrix_Write_d<br>0111<br>0112<br>0112<br>0113<br>0114<br>0115<br>0115<br>0116<br>0117<br>0118<br>0119subroutine Matrix_Write_d<br>character(LEN=*), intent(<br>0112<br>0113<br>0114<br>0115<br>0115<br>0116<br>0116<br>0117<br>0118<br>0119double precision, intent(<br>10114<br>0115<br>0115<br>0116<br>0116<br>0117<br>0118<br>0119   | 0105 | end do  | 0105 | end do  |
| 0108 0109 0110 0111 0111 0112 0112 0113 0114 0114 0115 0116 0116 0116 0116 0117 0117 0117 0118 0118 0119 010 010 010 0110 0110 01  | 0106 |   | 0106 |   |
| 0108 0109 0110 0111 0111 0112 0112 0113 0114 0114 0115 0116 0116 0116 0116 0117 0117 0117 0118 0118 0119 010 010 010 0110 0110 01  | 0107 | call CloseFile(file unit)   | 0107 | call CloseFile(file unit)                               |
| O110 O111 subroutine Matrix_Write_d O111 subroutine Matrix_Write_d O112 character(LEN=*), intent( O112 character(LEN=*), intent( O113 double precision, intent( O113 double precision, intent( O114 logical, intent(in), opti O114 logical, intent(in), opti O115 integer i,k O115 integer i,k O116 character(LEN=50) fmt O116 character(LEN=50) fmt O117 integer shp(2) O117 integer shp(2) O118 logical WriteTab O118 logical WriteTab O119 integer file_unit O119 integer file_unit   | 0108 | ` <b>-</b> /  | 0108 | ` _ ′   |
| 0111subroutine Matrix_Write_d0111subroutine Matrix_Write_d0112character(LEN=*), intent( 0112character(LEN=*), intent(0113double precision, intent( 0113double precision, intent(0114logical, intent(in), opti 0114logical, intent(in), opti integer i,k0115character(LEN=50) fmt 0116character(LEN=50) fmt integer shp(2)0117integer shp(2)0117integer shp(2)0118logical WriteTab integer file_unit0119integer file_unit   | 0109 | end subroutine Matrix Wri   | 0109 | end subroutine Matrix Wri                               |
| character(LEN=*), intent( 0112 character(LEN=*), intent( 0113 double precision, intent( 0113 logical, intent(in), opti 0114 logical, intent(in), opti 0115 integer i,k 0115 character(LEN=50) fmt 0116 character(LEN=50) fmt 0116 integer shp(2) 0117 integer shp(2) 0118 logical WriteTab 0118 integer file_unit 0119 character(LEN=*), intent(( 0112 double precision, intent( logical, intent(in), opti integer i,k character(LEN=50) fmt integer shp(2) integer file_unit  | 0110 | <del>-</del>  | 0110 | _   |
| 0113double precision, intent( 0113double precision, intent( 101140114logical, intent(in), opti 0114logical, intent(in), opti 101150115integer i,k0115integer i,k0116character(LEN=50) fmt 10116character(LEN=50) fmt 101160117integer shp(2) 1117integer shp(2) 11180118logical WriteTab 1118logical WriteTab 11190119integer file_unit 1119   | 0111 | subroutine Matrix Write d   | 0111 | subroutine Matrix Write d                               |
| logical, intent(in), opti 0114 integer i,k 0115 character(LEN=50) fmt 0116 integer shp(2) 0117 logical WriteTab 0118 integer file_unit 0119 logical, intent(in), opti 0114 integer i,k integer i,k character(LEN=50) fmt integer shp(2) logical WriteTab logical WriteTab integer file_unit  | 0112 | character(LEN=*), intent(   | 0112 | character(LEN=*), intent(                               |
| integer i,k 0115 integer i,k 0116 character(LEN=50) fmt 0116 character(LEN=50) fmt 0117 integer shp(2) 10gical WriteTab 10gic | 0113 | double precision, intent(   | 0113 | double precision, intent(                               |
| integer i,k 0115 integer i,k 0116 character(LEN=50) fmt 0116 character(LEN=50) fmt 0117 integer shp(2) 10gical WriteTab 10gic | 0114 | logical, intent(in), opti   | 0114 | logical, intent(in), opti                               |
| 0117integer shp(2)0117integer shp(2)0118logical WriteTab0118logical WriteTab0119integer file_unit0119integer file_unit   | 0115 |   | 0115 | integer i,k   |
| 0118 logical WriteTab 0118 logical WriteTab 0119 integer file_unit 0119 integer file_unit  | 0116 | character(LEN=50) fmt   | 0116 | character(LEN=50) fmt                                   |
| 0119 integer file_unit 0119 integer file_unit  | 0117 | integer shp(2)  | 0117 | integer shp(2)  |
| 0119 integer file_unit 0119 integer file_unit  | 0118 | logical WriteTab  | 0118 | logical WriteTab  |
| 0120   | 0119 | integer file_unit   | 0119 |   |
|  | 0120 | - <b>-</b>  | 0120 | _   |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 121</pre> |      | s/lplopa/Compare/camb_des/Matrix_u<br>'90, Top line: 121 |
|------|--|------|--|
| 0121 |  | 0121 |  |
|      | shp = shape(mat)   |      | shp = shape(mat)   |
| 0122 | WriteTab = shp(2)<=50  | 0122 | WriteTab = shp(2)<=50                                    |
| 0123 | <pre>if (present(forcetable)) if (forcetable)</pre>                | 0123 | <pre>if (present(forcetable)) if (forcetable)</pre>      |
| 0124 | if (forcetable) Write  | 0124 | if (forcetable) Write                                    |
| 0125 | end if   | 0125 | end if   |
| 0126 | file_unit = new_file_unit  | 0126 | file_unit = new_file_unit                                |
| 0127 | call CreateTxtFile(aname,  | 0127 | call CreateTxtFile(aname,                                |
| 0128 | <pre>fmt = trim(numcat('(',shp</pre>                               | 0128 | <pre>fmt = trim(numcat('(',shp</pre>                     |
| 0129 | do $i=1$ , $shp(1)$  | 0129 | do $i=1$ , $shp(1)$                                      |
| 0130 | if (.not. WriteTab) t  | 0130 | if (.not. WriteTab) t                                    |
| 0131 | do $k=1$ , $shp(2)$  | 0131 | do $k=1$ , $shp(2)$                                      |
| 0132 | write (file_u  | 0132 | write (file_u  |
| 0133 | end do   | 0133 | end do   |
| 0134 | else   | 0134 | else   |
| 0135 | write (file unit,  | 0135 | write (file unit,  |
| 0136 | end if   | 0136 | end if   |
| 0137 | end do   | 0137 | end do   |
| 0138 |  | 0138 |  |
| 0139 | call CloseFile(file unit)  | 0139 | call CloseFile(file unit)                                |
| 0140 | \ = /  | 0140 | \ /  |
| 0141 | end subroutine Matrix Wri  | 0141 | end subroutine Matrix Wri                                |
| 0142 | <del>-</del>   | 0142 | _  |
| 0143 |  | 0143 |  |
| 0144 | subroutine Matrix Write B  | 0144 | subroutine Matrix Write B                                |
| 0145 | character(LEN=*), intent(  |      | character(LEN=*), intent(                                |
| 0146 | real(dm), intent(in) :: m  |      | real(dm), intent(in) :: m                                |
| 0147 | integer file unit  | 0147 | integer file unit  |
| 0148 |  | 0148 |  |
| 0149 | file unit = new file unit  | 0149 | file unit = new file unit                                |
| 0150 | call CreateFile(aname, fi  |      | call CreateFile(aname, fi                                |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 151</pre> |      | <pre>/lplopa/Compare/camb_des/Matrix_u 90, Top line: 151</pre> |
|------|--|------|--|
| 0151 | write (file unit) mat  | 0151 | write (file unit) mat  |
| 0152 | call CloseFile(file unit)  | 0152 | call CloseFile(file_unit)                                      |
| 0153 | , , , , , , , , , , , , , , , , , , ,                              | 0153 | , , , , , , , , , , , , , , , , , , ,                          |
| 0154 | end subroutine Matrix Wri  | 0154 | end subroutine Matrix Wri                                      |
| 0155 | <del>-</del>   | 0155 | <b>–</b>   |
| 0156 |  | 0156 |  |
| 0157 | subroutine MatrixSym Writ  | 0157 | subroutine MatrixSym Writ                                      |
| 0158 | character(LEN=*), $\overline{intent}$                              | 0158 | character(LEN=*), $\overline{intent}$                          |
| 0159 | real(dm), intent(in) :: m  | 0159 | real(dm), intent(in) :: m                                      |
| 0160 | integer i  | 0160 | integeríi  |
| 0161 | integer shp(2)   | 0161 | integer shp(2)   |
| 0162 | integer file unit  | 0162 | integer file unit  |
| 0163 | <u> </u>   | 0163 | _  |
| 0164 | <pre>shp = shape(mat)</pre>  | 0164 | <pre>shp = shape(mat)</pre>                                    |
| 0165 | if(shp(1)) = shp(2) cal  | 0165 | if(shp(1)) = shp(2) cal  |
| 0166 | if (shp(1) == 0) return  | 0166 | if (shp(1) == 0) return  |
| 0167 | , _ , ,  | 0167 | · - · · ·  |
| 0168 | file unit = new file unit  | 0168 | file unit = new file unit                                      |
| 0169 | call CreateFile(aname, fi  | 0169 | call CreateFile(aname, fi                                      |
| 0170 | do $i=1,shp(1)$  | 0170 | do $i=1, shp(1)$   |
| 0171 | write (file unit) mat  | 0171 | write (file_unit) mat  |
| 0172 | end do   | 0172 | end do   |
| 0173 | call CloseFile(file_unit)  | 0173 | call CloseFile(file_unit)                                      |
| 0174 |  | 0174 |  |
| 0175 | <pre>end subroutine MatrixSym_</pre>                               | 0175 | <pre>end subroutine MatrixSym_</pre>                           |
| 0176 | <del>-</del>   | 0176 | <del>-</del>   |
| 0177 | <pre>subroutine MatrixSym_Writ</pre>                               | 0177 | <pre>subroutine MatrixSym_Writ</pre>                           |
| 0178 | $character(LEN=*)$ , $\overline{intent}$                           | 0178 | $character(LEN=*)$ , $\overline{intent}$                       |
| 0179 | real(dm), intent(in) :: m  | 0179 | real(dm), intent(in) :: m                                      |
| 0180 | integer i, file_unit   | 0180 | integer i, file_unit   |

| /Users/lplopa/Compare/camb_simdata/Matr   |   | /Users/lp1opa/Compare/camb_des/Matrix_u |                                      |  |
|---|---|---|--------------------------------------|--|
| <pre>ix_utils.F90, !</pre>  | Fop line: $18\overline{1}$                | tils.F90, Top line: 181                 |                                      |  |
| 0181 in   | nteger shp(2)                             | 0181                                    | integer shp(2)                       |  |
| 0182  |   | 0182                                    |                                      |  |
| 0183 sl   | hp = shape(mat)                           | 0183                                    | <pre>shp = shape(mat)</pre>          |  |
|   | f(shp(1)) = shp(2) cal                    | 0184                                    | if $(shp(1)) = shp(2)$ cal           |  |
|   | f (shp(1) == 0) return                    | 0185                                    | if $(shp(1) == 0)$ return            |  |
| 0186  |   | 0186                                    |                                      |  |
| 0187 f:   | ile unit = new file unit                  | 0187                                    | file unit = new file unit            |  |
|   | all CreateFile (aname, fi                 | 0188                                    | call CreateFile (aname, fi           |  |
| (AAAA)  | o i=1,shp(1)                              | 0189                                    | do i=1,shp(1)                        |  |
| 0190  | write (file_unit) rea                     | 0190                                    | write (file unit) rea                |  |
| 0191 <b></b> eı   | nd do                                     | 0191                                    | end do                               |  |
| 0192 ca   | all CloseFile(file unit)                  | 0192                                    | <pre>call CloseFile(file unit)</pre> |  |
| 0193  | ` = /                                     | 0193                                    | ` = '                                |  |
| 0194 eı   | nd subroutine MatrixSym                   | 0194                                    | end subroutine MatrixSym             |  |
| 0195  |   | 0195                                    |                                      |  |
| 0196  |   | 0196                                    |                                      |  |
| 0197  |   | 0197                                    |                                      |  |
| 0198 sı   | ubroutine Matrix WriteVe                  | 0198                                    | subroutine Matrix WriteVe            |  |
| 0199 cl   | haracter(LEN=*), intent(                  | 0199                                    | <pre>character(LEN=*), intent(</pre> |  |
|   | eal(dm), intent(in) :: v                  | 0200                                    | real(dm), intent(in) :: v            |  |
| recent limited and the second | ntegeríi, file unit                       | 0201                                    | integer'i, file unit                 |  |
| 0202  | <b>-</b> -                                | 0202                                    | <b>-</b>                             |  |
| 0203 f:   | ile unit = new file unit                  | 0203                                    | file unit = new file unit            |  |
| 0204 ca   | $all$ CreateTxtF $\overline{i}$ le(aname, | 0204                                    | call CreateTxtFile(aname,            |  |
|   | o i=1, size(vec)                          | 0205                                    | do i=1, size(vec)                    |  |
| 0206  | write (file únit, '(1                     | 0206                                    | write (file unit, '(1                |  |
| 0207 eı   | nd do                                     | 0207                                    | end do                               |  |
| 0208 ca   | all CloseFile(file unit)                  | 0208                                    | <pre>call CloseFile(file unit)</pre> |  |
| 0209  | ` _ /                                     | 0209                                    | ` = '                                |  |
| 0210 ei   | nd subroutine Matrix_Wri                  | 0210                                    | end subroutine Matrix_Wri            |  |

|      | /lplopa/Compare/camb_simdata/Matr<br>ls.F90, Top line: 211 |      | s/lplopa/Compare/camb_des/Matrix_u<br>'90, Top line: 211 |
|------|--|------|--|
| 0211 |  | 0211 |  |
| 0212 |  | 0212 |  |
| 0213 | subroutine Matrix Read Bi                                  | 0213 | subroutine Matrix Read Bi                                |
| 0214 | character(LEN=*), intent(                                  | 0214 | character(LEN=*), intent(                                |
| 0215 | real(dm), intent(out) ::                                   | 0215 | real(dm), intent(out) ::                                 |
| 0216 | integer file unit  | 0216 | integer file unit  |
| 0217 | _  | 0217 |  |
| 0218 | file_unit = new_file_unit                                  | 0218 | file_unit = new_file_unit                                |
| 0219 | call OpenFile(aname, file                                  | 0219 | call OpenFile(aname, file                                |
| 0220 | read (file unit) mat                                       | 0220 | read (file unit) mat                                     |
| 0221 | call CloseFile(file_unit)                                  | 0221 | call CloseFile(file_unit)                                |
| 0222 |  | 0222 |  |
| 0223 | <pre>end subroutine Matrix_Rea</pre>                       | 0223 | <pre>end subroutine Matrix_Rea</pre>                     |
| 0224 | <del>-</del>   | 0224 |  |
| 0225 |  | 0225 |  |
| 0226 | <pre>subroutine MatrixSym_Read</pre>                       | 0226 | subroutine MatrixSym_Read                                |
| 0227 | character(LEN=*), intent(                                  | 0227 | character(LEN=*), intent(                                |
| 0228 | real(dm), intent(out) ::                                   | 0228 | real(dm), intent(out) ::                                 |
| 0229 | integer i, file_unit                                       | 0229 | integer i, file_unit                                     |
| 0230 | integer shp(2)   | 0230 | integer shp(2)   |
| 0231 | _ , ,  | 0231 |  |
| 0232 | <pre>shp = shape(mat)</pre>                                | 0232 | <pre>shp = shape(mat)</pre>                              |
| 0233 | if $(shp(1) /= shp(2))$ cal                                | 0233 | if $(shp(1) /= shp(2))$ cal                              |
| 0234 | if (shp(1) == 0) return                                    | 0234 | if $(shp(1) == 0)$ return                                |
| 0235 | · - · · ·  | 0235 |  |
| 0236 | file_unit = new_file_unit                                  | 0236 | file_unit = new_file_unit                                |
| 0237 | call OpenFile(aname, file                                  | 0237 | call OpenFile(aname, file                                |
| 0238 | do $i=1$ , shp(1)  | 0238 | do $i=1$ , shp(1)  |
| 0239 | read (file_unit) mat(                                      | 0239 | read (file_unit) mat(                                    |
| 0240 | $mat(i,i:sh\overline{p}(1)) = mat$                         |      | $mat(i,i:sh\overline{p}(1)) = mat$                       |

|      | /lplopa/Compare/camb_simdata/Matr<br>ls.F90, Top line: 241 |      | s/lplopa/Compare/camb_des/Matrix_u<br>190, Top line: 241 |
|------|--|------|--|
| 0241 | end do   | 0241 | end do   |
| 0242 | call CloseFile(file unit)                                  | 0242 | call CloseFile(file unit)                                |
| 0243 | call closerific(file_anic)                                 | 0243 | carr croscrire(rrie_amrc)                                |
| 0244 | end subroutine MatrixSym                                   | 0244 | end subroutine MatrixSym                                 |
| 0245 |  | 0245 |  |
| 0246 | subroutine MatrixSym Read                                  | 0246 | subroutine MatrixSym Read                                |
| 0247 | character(LEN=*), intent(                                  | 0247 | character(LEN=*), intent(                                |
| 0248 | real, intent(out) :: mat(                                  | 0248 | real, intent(out) :: mat(                                |
| 0249 | integer i, file unit                                       | 0249 | integer i, file unit                                     |
| 0250 | integer shp(2)   | 0250 | integer shp(2)   |
| 0251 |  | 0251 |  |
| 0252 | <pre>shp = shape(mat)</pre>                                | 0252 | <pre>shp = shape(mat)</pre>                              |
| 0253 | if'(shp(1)) = shp(2) cal                                   |      | if $(shp(1)) = shp(2)$ cal                               |
| 0254 | if $(shp(1) == 0)$ return                                  | 0254 | if (shp(1) == 0) return                                  |
| 0255 |  | 0255 |  |
| 0256 | file unit = new file unit                                  | 0256 | file unit = new file unit                                |
| 0257 | call OpenFile(aname, file                                  | 0257 | call OpenFile(aname, file                                |
| 0258 | do i=1,shp(1)  | 0258 | do i=1,shp(1)  |
| 0259 | read (file unit) mat(                                      | 0259 | read (file unit) mat(                                    |
| 0260 | $mat(i,i:sh\overline{p}(1)) = mat$                         | 0260 | $mat(i,i:sh\overline{p}(1)) = mat$                       |
| 0261 | end do   | 0261 | end do   |
| 0262 | call CloseFile(file_unit)                                  | 0262 | call CloseFile(file_unit)                                |
| 0263 |  | 0263 |  |
| 0264 | <pre>end subroutine MatrixSym_</pre>                       | 0264 | end subroutine MatrixSym_                                |
| 0265 |  | 0265 |  |
| 0266 |  | 0266 |  |
| 0267 |  | 0267 |  |
| 0268 |  | 0268 |  |
| 0269 |  | 0269 |  |
| 0270 | subroutine Matrix_Read(an                                  | 0270 | subroutine Matrix_Read(an                                |

|        |        | /Users/lp1opa/Compare/camb_des/Matrix_u |        |        |                                       |
|--------|--------|---|--------|--------|---------------------------------------|
| ix_uti | ls.F90 | , Top line: 271                         | tils.F | 90, To | p line: 271                           |
| 0271   |        | <pre>character(LEN=*), intent(</pre>    | 0271   |        | <pre>character(LEN=*), intent(</pre>  |
| 0272   |        | real(dm), intent(out) ::                | 0272   |        | real(dm), intent(out) ::              |
| 0273   |        | integer j,k, file unit                  | 0273   |        | integer j,k, file unit                |
| 0274   |        | integer shp(2)                          | 0274   |        | integer shp(2)                        |
| 0275   |        | real(dm) tmp                            | 0275   |        | real(dm) tmp                          |
| 0276   |        |   | 0276   |        | · · · · · · · · · · · · · · · · · · · |
| 0277   |        | <pre>shp = shape(mat)</pre>             | 0277   |        | <pre>shp = shape(mat)</pre>           |
| 0278   |        |   | 0278   |        |                                       |
| 0279   |        | file_unit = new_file_unit               | 0279   |        | file_unit = new_file_unit             |
| 0280   |        | call OpenTxtFile(aname, f               | 0280   |        | call OpenTxtFile(aname, f             |
| 0281   |        |   | 0281   |        |                                       |
| 0282   |        | do j=1,shp(1)                           | 0282   |        | do j=1,shp(1)                         |
| 0283   |        | read (file_unit,*, en                   | 0283   |        | read (file_unit,*, en                 |
| 0284   |        | end do                                  | 0284   |        | end do                                |
| 0285   |        | goto 120                                | 0285   |        | goto 120                              |
| 0286   |        |   | 0286   |        |                                       |
| 0287   | 100    | rewind(file_unit) !Try o                | 0287   | 100    | <pre>rewind(file_unit) !Try o</pre>   |
| 0288   |        | do j=1,shp(1)                           | 0288   |        | do $j=1, shp(1)$                      |
| 0289   |        | do $k=1$ , $shp(2)$                     | 0289   |        | do $k=1,shp(2)$                       |
| 0290   |        | read (file_unit,*                       | 0290   |        | read (file_unit,*                     |
| 0291   |        | end do                                  | 0291   |        | end do                                |
| 0292   |        | end do                                  | 0292   |        | end do                                |
| 0293   | 4.00   |   | 0293   |        |                                       |
| 0294   | 120    | <pre>read (file_unit,*, err =</pre>     | 0294   | 120    | <pre>read (file_unit,*, err =</pre>   |
| 0295   |        | goto 200                                | 0295   |        | goto 200                              |
| 0296   | 4-0    |   | 0296   | 4 - 4  |                                       |
| 0297   | 150    | call CloseFile(file_unit)               | 0297   | 150    | call CloseFile(file_unit)             |
| 0298   |        | return                                  | 0298   |        | return                                |
| 0299   | 000    |   | 0299   | 000    |                                       |
| 0300   | 200    | call MpiStop('Matrix_Read               | 0300   | 200    | call MpiStop('Matrix_Read             |

| O304   O305   Subroutine Matrix_ReadSin O305   Character(LEN=*), intent( O306   Character(LEN=*), intent( O307   creal, intent(out) :: mat( O307   integer j,k, file_unit O308   integer shp(2)   O309   creal tmp   O310   O311   O312   Shp = shape(mat)   O312   Shp = shape(mat)   O313   O314   Call OpenTxtFile(aname, f O315   O316   O317   Call OpenTxtFile(aname, f O315   O316   O317   Call OpenTxtFile(aname, f O318   Call OpenTxtFile(aname, f O318   Call OpenTxtFile(aname, f O319   O320   O320   O321   O322   O323   O324   Odo j=1, shp(1)   O324   Odo j=1, shp(2)   Odo de=1, shp(2)   Odo d   |        |        |                                      | /Users/lp1opa/Compare/camb_des/Matrix_u |     |                                      |
|--|--------|--------|--------------------------------------|---|-----|--------------------------------------|
| 0302 0303 0304 0304 0305 0305 0306 0306 0307 0307 0308 0308 0308 0308 0308 0309 0309 0309  | ix_uti | ls.F90 | , Top line: 301                      | tils.F90, Top line: 301                 |     |                                      |
| 0303       end subroutine Matrix_Rea       0304       end subroutine Matrix_Rea         0304       subroutine Matrix_ReadSin       0305       subroutine Matrix_ReadSin         0306       character(LEN=*), intent(       0306       subroutine Matrix_ReadSin         0307       real, intent(out) :: mat(       0307       real, intent(out) :: mat(       real, intent(out) :: mat(       integer j,k, file_unit       integer j,k, file_unit<   | 0301   |        |                                      | 0301                                    |     |                                      |
| 0304   0305   0304   0305   0305   0306   0306   0306   0307   0307   0307   0307   0308   0308   0308   0308   0308   0309   0309   0310   0311   0311   0311   0312   0313   0314   0315   0316   0316   0316   0317   0318   0317   0318   0317   0318   0318   0319   0310   0310   0318   0310   0310   0318   0317   0318   0316   0316   0316   0317   0318   0319   0320   0320   0321   0322   0322   0322   0323   0324   0326  | 0302   |        |                                      | 0302                                    |     |                                      |
| 0304   0305   0304   0305   0305   0306   0306   0306   0306   0307   0307   0307   0308   0307   0308   0309   0310   0310   0311   0312   0311   0312   0313   0314   0315   0316   0316   0316   0316   0317   0318   0317   0318   0317   0318   0317   0318   0310   0320   0320   0321   0322   0322   0322   0323   0324   0326  | 0303   |        | end subroutine Matrix Rea            | 0303                                    |     | end subroutine Matrix Rea            |
| <pre>character(LEN=*), intent( 0306     real, intent(out) :: mat( 0307     integer j,k, file_unit 0308     integer shp(2)</pre>  | 0304   |        | _                                    | 0304                                    |     | _                                    |
| 0307   | 0305   |        | subroutine Matrix ReadSin            | 0305                                    |     | subroutine Matrix ReadSin            |
| <pre>integer j,k, file_unit 0308</pre>   | 0306   |        | <pre>character(LEN=*), intent(</pre> | 0306                                    |     | <pre>character(LEN=*), intent(</pre> |
| 0309   | 0307   |        | <pre>real, intent(out) :: mat(</pre> | 0307                                    |     | <pre>real, intent(out) :: mat(</pre> |
| 0310 real tmp 0310 0311 0312 shp = shape(mat) 0312 shp = shape(mat) 0313 file_unit = new_file_unit 0314 call_OpenTxtFile(aname, f 0315 0316 0317 do j=1,shp(1) 0317 read (file_unit,*, en 0318 end do goto 120 0321 0322 do j=1,shp(1) 0322 do j=1,shp(1) 0323 do k=1,shp(1) 0324 read (file_unit,* 0325 end do 0326 end do  | 0308   |        | integer j,k, file_unit               | 0308                                    |     | <pre>integer j,k, file_unit</pre>    |
| 0311 0312 0313 0314 0315 0316 0317 0318 0319 0320 0320 0320 0321 0322 0322 0323 0324 0324 0325 0326 0326 0326 0321 0326 0321 0324 0325 0326 0321 0326 0321 0324 0325 0326 0321 0326 0321 0324 0325 0326 0321 0326 0321 0324 0325 0326 0321 0326 0321 0324 0325 0326 0326 0321 0327 0328 0328 0328 0328 0329 0320 0320 0320 0321 0322 0323 0324 0325 0326 0326 0326 0327 0328 0328 0328 0328 0328 0329 0320 0321 0320 0321 0322 0323 0324 0325 0326 0326 0326 0327 0328 0328 0328 0328 0328 0328 0328 0328  | 0309   |        | integer shp(2)                       | 0309                                    |     | integer shp(2)                       |
| 0312       shp = shape(mat)       0312       shp = shape(mat)         0313       file_unit = new_file_unit       0314       file_unit = new_file_unit       file_unit = new_file_unit         0315       call OpenTxtFile(aname, f       0315       call OpenTxtFile(aname, f         0316       0316       do j=1,shp(1)       do j=1,shp(1)         0318       read (file_unit,*, en       o318       read (file_unit,*, en         0319       end do       goto 120       goto 120         0321       o321       o321       o321         0322       to rewind(file_unit)       !Try o 0322       od j=1,shp(1)         0324       do k=1,shp(2)       do k=1,shp(2)         0325       read (file_unit,*       o325         0326       end do       o326  | 0310   |        | real tmp                             | 0310                                    |     | real tmp                             |
| 0313 0314 0315 0316 0317 0318 0319 0320 0320 0321 0322 0321 0322 0323 0324 0324 0326 0326 0326 0326 0327 0328 0328 0328 0329 0320 0321 0320 0321 0322 0323 0324 0325 0326 0326 0326 0327 0328 0328 0328 0329 0320 0321 0320 0321 0322 0323 0324 0325 0326 0326 0326 0327 0328 0328 0328 0329 0329 0320 0321 0320 0321 0322 0323 0324 0325 0326 0326 0326 0327 0328 0328 0328 0328 0329 0329 0329 0329 0329 0329 0329 0329  | 0311   |        | _                                    | 0311                                    |     | _                                    |
| <pre>file_unit = new_file_unit 0314 call OpenTxtFile(aname, f 0315 0316 0317 0318 0319 0320 0321 0322 0321 0322 0323 0324 0325 0326</pre> file_unit = new_file_unit 0314 call OpenTxtFile(aname, f 0315 call OpenTxtFile(aname, f 0316 do j=1,shp(1)   | 0312   |        | <pre>shp = shape(mat)</pre>          | 0312                                    |     | <pre>shp = shape(mat)</pre>          |
| 0315       call OpenTxtFile(aname, f 0315 0316       call OpenTxtFile(aname, f 0315 0316         0316       0317       do j=1,shp(1) 0317 read (file_unit,*, en 0318 end do goto 120 0320 0321       do j=1,shp(1) read (file_unit) !Try o 0322 do j=1,shp(1) 0323 do j=1,shp(1) 0324 read (file_unit,* o325 end do 0326       100 rewind(file_unit) !Try o 0322 do k=1,shp(2) read (file_unit,* o325 end do 0326       100 rewind(file_unit,* o325 end do end do 0326   | 0313   |        | ,                                    | 0313                                    |     | , ,                                  |
| 0316<br>0317<br>0318<br>0318<br>0319<br>0319<br>0320<br>0320<br>0321<br>0322<br>0322<br>0323<br>0324<br>0325<br>0326<br>0316<br>0317<br>0318<br>end (file_unit,*, en end do goto 120<br>0320<br>0321<br>0322<br>0323<br>0324<br>0325<br>0326<br>0326<br>0326<br>0326<br>0327<br>0328<br>0328<br>0329<br>0329<br>0320<br>0321<br>0321<br>0323<br>0324<br>0325<br>0326<br>0326<br>0326<br>0327<br>0328<br>0328<br>0329<br>0329<br>0329<br>0320<br>0321<br>0321<br>0323<br>0324<br>0325<br>0326<br>0326<br>0326<br>0327<br>0328<br>0328<br>0329<br>0329<br>0329<br>0320<br>0320<br>0321<br>0323<br>0324<br>0325<br>0326<br>0326<br>0326<br>0326<br>0326<br>0327<br>0328<br>0328<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329<br>0329 | 0314   |        | file_unit = new_file_unit            | 0314                                    |     | file_unit = new_file_unit            |
| 0317       do j=1,shp(1)       0317       do j=1,shp(1)       read (file_unit,*, en 0318       read (file_unit,*, en end do goto 120       read (file_unit,*, en end do goto 120         0321       0321       0321       0321       100 rewind(file_unit) !Try o 0322 do j=1,shp(1)       100 rewind(file_unit) !Try o do j=1,shp(1)       100 rewind(file_unit) !Try o do j=1,shp(1)       0323       0324       0324       read (file_unit,* o325 end do       read (file_unit,* o325 end do       read (file_unit,* end do   | 0315   |        | call OpenTxtFile(aname, f            | 0315                                    |     | <pre>call OpenTxtFile(aname, f</pre> |
| 0318       read (file_unit,*, en 0318       read (file_unit,*, en 0318         0319       end do goto 120       0320         0321       0321       0321         0322       100 rewind(file_unit) !Try o 0322       100 rewind(file_unit) !Try o 0323         0323       do j=1,shp(1) 0324       0324 do k=1,shp(2) cread (file_unit,* 0325 end do 0326         0325       read (file_unit,* 0325 end do 0326       read (file_unit,* end do creat (file_unit) !Try o do k=1,shp(2) creat (file_unit,* end do creat (fi  | 0316   |        |                                      | 0316                                    |     | _ ,                                  |
| 0319       end do       0319       end do         0320       goto 120       0320       goto 120         0321       0321       100 rewind(file_unit) !Try o 0322       100 rewind(file_unit) !Try o 0323       do j=1,shp(1)         0323       do j=1,shp(1)       0323       do j=1,shp(1)         0324       read (file_unit,* 0325       read (file_unit,* 0325         0326       end do       end do  | 0317   |        | do $j=1,shp(1)$                      | 0317                                    |     | do $j=1, shp(1)$                     |
| 0320       goto 120       0320       goto 120         0321       0321       0321       100 rewind(file_unit) !Try o 0322       100 rewind(file_unit) !Try o 0323       do j=1,shp(1) do k=1,shp(2)       do j=1,shp(1) do k=1,shp(2)       do k=1,shp(2) read (file_unit,*       read (file_unit,*         0325       o326       end do       end do       end do  | 0318   |        | read (file_unit,*, en                | 0318                                    |     | read (file_unit,*, en                |
| 0321<br>0322   | 0319   |        | end do                               | 0319                                    |     | end do                               |
| 0322       100 rewind(file_unit) !Try o 0322       100 rewind(file_unit) !Try o 0323       100 rewind(file_unit) !Try o do j=1,shp(1)         0323       do j=1,shp(1)       0324       do k=1,shp(2)       do k=1,shp(2)       read (file_unit,*         0325       read (file_unit,*       0325       read (file_unit,*       end do   | 0320   |        | goto 120                             | 0320                                    |     | goto 120                             |
| 0323       do j=1,shp(1)       0323       do j=1,shp(1)         0324       do k=1,shp(2)       do k=1,shp(2)         0325       read (file_unit,* o325       read (file_unit,* end do         0326       end do  | 0321   |        |                                      | 0321                                    |     |                                      |
| 0324       do k=1,shp(2)       do k=1,shp(2)         0325       read (file_unit,* 0325       read (file_unit,* end do         0326       end do  | 0322   | 100    | <pre>rewind(file_unit) !Try o</pre>  | 0322                                    | 100 | <pre>rewind(file_unit) !Try o</pre>  |
| 0325 read (file_unit,* 0325 read (file_unit,* 0326 end do  | 0323   |        | do j=1,shp(1)                        | 0323                                    |     | do $j=1, shp(1)$                     |
| end do 0326 end do   | 0324   |        |                                      |   |     |                                      |
|  | 0325   |        | read (file_unit,*                    | 0325                                    |     | read (file_unit,*                    |
| 0327 end do 0327 end do  | 0326   |        | end do                               | 0326                                    |     | end do                               |
|  | 0327   |        | end do                               | 0327                                    |     | end do                               |
| 0328   | 0328   |        |                                      | 0328                                    |     |                                      |
| 0329   120 read (file_unit,*, err = 0329   120 read (file_unit,*, err =  |        | 120    | read (file_unit,*, err =             | 0329                                    | 120 | read (file_unit,*, err =             |
| 0330 goto 200 0330 goto 200  | 0330   |        | goto 200                             | 0330                                    |     | goto 200                             |

| /Users/lplopa/Compare/camb_simdata/Matr |        |                                      |         |        |                                      |
|---|--------|--------------------------------------|---------|--------|--------------------------------------|
| <u>ix_uti</u>                           | ls.F90 | , Top line: 331                      | tils.F9 | 90, To | p line: 331                          |
| 0331                                    |        |                                      | 0331    |        |                                      |
| 0332                                    | 150    | <pre>call CloseFile(file_unit)</pre> | 0332    | 150    | <pre>call CloseFile(file_unit)</pre> |
| 0333                                    |        | return                               | 0333    |        | return                               |
| 0334                                    |        |                                      | 0334    |        |                                      |
| 0335                                    | 200    | <pre>call MpiStop('Matrix_Read</pre> | 0335    | 200    | <pre>call MpiStop('Matrix_Read</pre> |
| 0336                                    |        |                                      | 0336    |        |                                      |
| 0337                                    |        |                                      | 0337    |        |                                      |
| 0338                                    |        | <pre>end subroutine Matrix_Rea</pre> | 0338    |        | <pre>end subroutine Matrix_Rea</pre> |
| 0339                                    |        |                                      | 0339    |        |                                      |
| 0340                                    |        |                                      | 0340    |        |                                      |
| 0341                                    |        | <pre>function Matrix_Diag(M, n</pre> | 0341    |        | <pre>function Matrix_Diag(M, n</pre> |
| 0342                                    |        | <pre>integer, intent(in) :: n</pre>  | 0342    |        | <pre>integer, intent(in) :: n</pre>  |
| 0343                                    |        | real(dm), intent(in) :: M            | 0343    |        | <pre>real(dm), intent(in) :: M</pre> |
| 0344                                    |        | real(dm) Matrix_Diag(n)              | 0344    |        | real(dm) Matrix_Diag(n)              |
| 0345                                    |        | integer i                            | 0345    |        | integer i                            |
| 0346                                    |        |                                      | 0346    |        |                                      |
| 0347                                    |        | do i=1,n                             | 0347    |        | do i=1,n                             |
| 0348                                    |        |                                      | 0348    |        |                                      |
| 0349                                    |        | $Matrix_Diag(i) = M(i,i)$            | 0349    |        | $Matrix_Diag(i) = M(i,$              |
| 0350                                    |        |                                      | 0350    |        |                                      |
| 0351                                    |        | end do                               | 0351    |        | end do                               |
| 0352                                    |        |                                      | 0352    |        |                                      |
| 0353                                    |        | end function Matrix_Diag             | 0353    |        | end function Matrix_Diag             |
| 0354                                    |        |                                      | 0354    |        |                                      |
| 0355                                    |        | <u> </u>                             | 0355    |        | <pre>function ILAENV_wrap(i,S1</pre> |
| 0356                                    |        | integer ILAENV_wrap                  | 0356    |        | integer ILAENV_wrap                  |
| 0357                                    |        | <pre>integer, intent(in) :: i,</pre> | i       |        | <pre>integer, intent(in) :: i,</pre> |
| 0358                                    |        | <pre>character(LEN=*), intent(</pre> |         |        | <pre>character(LEN=*), intent(</pre> |
| 0359                                    |        | integer, external :: ILAE            |         |        | integer, external :: ILAE            |
| 0360                                    |        |                                      | 0360    |        |                                      |

```
/Users/lplopa/Compare/camb des/Matrix u
/Users/lplopa/Compare/camb simdata/Matr
ix utils. F90, Top line: 36\overline{1}
                                         tils.F90, Top line: 361
0361
                                         0361
             !If you don't have ILAENV
                                                       !If you don't have ILAENV
             !that is a quess at the b
                                         0362
                                                       !that is a quess at the b
0362
                                         0363
0363
        #ifdef MATRIX SINGLE
                                                  #ifdef MATRIX SINGLE
0364
                                         0364
                                                       ILAENV wrap = 16
             ILAENV wrap = 16
0365
        #else
                                         0365
                                                  #else
                                         0366
0366
             ILAENV wrap = ILAENV(i,S)
                                                       ILAENV wrap = ILAENV(i,S
                                         0367
0367
        #endif
                                                  #endif
                                         0368
0368
             !!!IFC
                                                       !!!IFC
0369
                                         0369
             end
                  function ILAENV wrap
                                                       end
                                                            function ILAENV wrap
0370
                                         0370
0371
                                         0371
0372
                                         0372
             subroutine Matrix Diagona
                                                       subroutine Matrix Diagona
0373
             !Does m = U \operatorname{diag} U^T, ret
                                         0373
                                                       !Does m = U \operatorname{diag} U^T, ret
                                         0374
0374
             integer, intent(in) :: n
                                                       integer, intent(in) :: n
0375
             real(dm), intent(inout)::
                                         0375
                                                       real(dm), intent(inout)::
0376
             real(dm), intent(out) ::
                                         0376
                                                       real(dm), intent(out) ::
0377
             integer ierr, tmpsize
                                         0377
                                                       integer ierr, tmpsize
             real(dm), allocatable, di
0378
                                         0378
                                                       real(dm), allocatable, di
0379
                                         0379
0380
             call Matrix Start('Diagon
                                         0380
                                                       call Matrix Start('Diagon
0381
                                         0381
        #ifdef MATRIX SINGLE
                                                  #ifdef MATRIX SINGLE
0382
             tmpsize = max( (ILAENV w
                                         0382
                                                       tmpsize = max( (ILAENV w
0383
             allocate(tmp(tmpsize));
                                         0383
                                                       allocate(tmp(tmpsize));
0384
                                         0384
             call SSYEV('V','U',n,m,n,
                                                       call SSYEV('V','U',n,m,n,
                                                  #else
0385
        #else
                                         0385
0386
             tmpsize = max( (ILAENV w
                                         0386
                                                       tmpsize = max((ILAENV w)
0387
             allocate(tmp(tmpsize));
                                         0387
                                                       allocate(tmp(tmpsize));
0388
             call DSYEV('V','U',n,m,n,
                                         0388
                                                       call DSYEV('V','U',n,m,n,
0389
        #endif
                                         0389
                                                  #endif
0390
             if (ierr /= 0) call MpiSt
                                         0390
                                                       if (ierr /= 0) call MpiSt
```

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 391</pre> |      | <pre>/lplopa/Compare/camb_des/Matrix_u 90, Top line: 391</pre> |
|------|--|------|--|
| 0391 | deallocate(tmp)  | 0391 | deallocate(tmp)  |
| 0392 | call Matrix End('Diagonal  | 0392 | call Matrix_End('Diagonal                                      |
| 0393 | ourr madrin_bidgomar   | 0393 | dari nadrin_ina( bragonar                                      |
| 0394 | end subroutine Matrix Dia  | 0394 | end subroutine Matrix Dia                                      |
| 0395 |  | 0395 |  |
| 0396 | subroutine Matrix Diagona  | 0396 | subroutine Matrix Diagona                                      |
| 0397 | !Complex version. Does m   | 0397 | !Complex version. Does m                                       |
| 0398 | integer, intent(in) :: n   | 0398 | integer, intent(in) :: n                                       |
| 0399 | real(dm), intent(inout)::  | 0399 | real(dm), intent(inout)::                                      |
| 0400 | real(dm), intent(out) ::   | 0400 | real(dm), intent(out) ::                                       |
| 0401 | integer ierr, tmpsize ,is  | 0401 | integer ierr, tmpsize ,is                                      |
| 0402 | real(dm), allocatable, di  | 0402 | real(dm), allocatable, di                                      |
| 0403 | integer, allocatable, dime   | 0403 | integer, allocatable, dime                                     |
| 0404 |  | 0404 |  |
| 0405 | call Matrix Start('Diagon  | 0405 | call Matrix_Start('Diagon                                      |
| 0406 |  | 0406 |  |
| 0407 | if $(matrix method == Mat$   | 0407 | <pre>if (matrix method == Mat</pre>                            |
| 0408 | !Divide and conquer  | 0408 | !Divide and conquer  |
| 0409 | tmpsize = 1 + 6*N + 2  | 0409 | tmpsize = 1 + 6*N + 2  |
| 0410 | isize = 3+5*N  | 0410 | isize = 3+5*N  |
| 0411 | allocate(tmp(tmpsize)  | 0411 | allocate(tmp(tmpsize)  |
| 0412 | allocate(iwork(isize)  | 0412 | allocate(iwork(isize)  |
| 0413 | #ifdef MATRIX SINGLE `   | 0413 | #ifdef MATRIX SINGLE `   |
| 0414 | $call \overline{S}SYEVD('V','U',n)$                                | 0414 | $call \overline{S}SYEVD('V','U',n)$                            |
| 0415 | #else  | 0415 | #else  |
| 0416 | call DSYEVD('V','U',n  | 0416 | call DSYEVD('V','U',n  |
| 0417 | #endif   | 0417 | #endif   |
| 0418 | deallocate(iwork)  | 0418 | deallocate(iwork)  |
| 0419 | deallocate(tmp) ´  | 0419 | deallocate(tmp) ´  |
| 0420 | else   | 0420 | else   |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 421</pre> |      | s/lplopa/Compare/camb_des/Matrix_u<br>'90, Top line: 421 |
|------|--|------|--|
| 0421 | call Matrix Diagonali  | 0421 | call Matrix Diagonali                                    |
| 0422 | end if   | 0422 | end if   |
| 0423 |  | 0423 |  |
| 0424 | if (ierr /= 0) call MpiSt  | 0424 | if (ierr /= 0) call MpiSt                                |
| 0425 | (,,,   | 0425 |  |
| 0426 | call Matrix End('Diagonal  | 0426 | call Matrix End('Diagonal                                |
| 0427 |  | 0427 |  |
| 0428 | end subroutine Matrix Dia  | 0428 | end subroutine Matrix Dia                                |
| 0429 | <del>-</del>   | 0429 | _  |
| 0430 |  | 0430 |  |
| 0431 |  | 0431 |  |
| 0432 | <pre>subroutine Matrix Root(M,</pre>                               | 0432 | <pre>subroutine Matrix Root(M,</pre>                     |
| 0433 | !Does M**pow for symmetri  | 0433 | !Does M**pow for symmetri                                |
| 0434 | !Not optimized for large   | 0434 | !Not optimized for large                                 |
| 0435 | <pre>integer, intent(in) :: n</pre>                                | 0435 | <pre>integer, intent(in) :: n</pre>                      |
| 0436 | real(dm), intent(inout)::  | 0436 | real(dm), intent(inout)::                                |
| 0437 | real(dm) :: Tmp(n,n)   | 0437 | real(dm) :: Tmp(n,n)                                     |
| 0438 | real(dm), intent(in) :: p  | 0438 | real(dm), intent(in) :: p                                |
| 0439 |  | 0439 |  |
| 0440 | real(dm) :: diag(n)  | 0440 | real(dm) :: diag(n)                                      |
| 0441 | integer i  | 0441 | integer i  |
| 0442 |  | 0442 |  |
| 0443 | call Matrix_Diagonalize(M  | 0443 | call Matrix_Diagonalize(M                                |
| 0444 | Tmp = M  | 0444 | Tmp = M  |
| 0445 | diag = diag**pow   | 0445 | <pre>diag = diag**pow</pre>                              |
| 0446 | do i = 1, n  | 0446 | do i = 1, n  |
| 0447 | M(:,i) = M(:,i)*diag(  | 0447 | M(:,i) = M(:,i)*diag(                                    |
| 0448 | end do   | 0448 | end do   |
| 0449 | <pre>M = matmul(M,transpose(Tm</pre>                               | 0449 | <pre>M = matmul(M,transpose(Tm</pre>                     |
| 0450 |  | 0450 |  |

| /Users/lp1opa/Compare/camb_simdata/Matr ix utils.F90, Top line: 451 |   |      | <pre>/lplopa/Compare/camb_des/Matrix_u 90, Top line: 451</pre> |
|---|---|------|--|
| 0451  | end subroutine Matrix Roo               | 0451 | end subroutine Matrix Roo                                      |
| 0452  |   | 0452 |  |
| 0453  |   | 0453 |  |
| 0454  | subroutine Matrix Diagona               | 0454 | subroutine Matrix Diagona                                      |
| 0455  | !Real version. Does $m = U$             | 0455 | !Real version. Does $m = U$                                    |
| 0456  | !Assumes up to nfound val               | 0456 | !Assumes up to nfound val                                      |
| 0457  | <pre>integer, intent(in) :: n</pre>     | 0457 | <pre>integer, intent(in) :: n</pre>                            |
| 0458  | real(dm), intent(inout)::               | 0458 | real(dm), intent(inout)::                                      |
| 0459  | real(dm), intent(out) ::                | 0459 | real(dm), intent(out) ::                                       |
| 0460  | real(dm), intent(in):: e                | 0460 | real(dm), intent(in):: e                                       |
| 0461  | <pre>integer, intent(inout) ::</pre>    | 0461 | <pre>integer, intent(inout) ::</pre>                           |
| 0462  | integer ierr, worksize, L               | 0462 | integer ierr, worksize, L                                      |
| 0463  | real(dm), allocatable, di               | 0463 | real(dm), allocatable, di                                      |
| 0464  | real(dm), allocatable, di               | 0464 | real(dm), allocatable, di                                      |
| 0465  | integer, allocatable, dime              | 0465 | integer, allocatable, dime                                     |
| 0466  | real(dm) wsize(1)                       | 0466 | real(dm) wsize(1)  |
| 0467  | real(dm) atol (                         | 0467 | real(dm) atol  |
| 0468  | integer ISize(1)                        | 0468 | integer ISize(1)   |
| 0469  | • ,                                     | 0469 |  |
| 0470  | atol = 1d-9                             | 0470 | atol = 1d-9  |
| 0471  | call Matrix Start('Matrix               | 0471 | call Matrix Start('Matrix                                      |
| 0472  | allocate(tmp(n, nfound))                | 0472 | allocate(tmp(n,nfound))  |
| 0473  | allocate(Supp(n))                       | 0473 | allocate(Supp(n))  |
| 0474  | !Query                                  | 0474 | !Query   |
| 0475  | WorkSize = -1                           | 0475 | WorkSize = -1  |
| 0476  | LIWork = -1                             | 0476 | LIWork = -1  |
| 0477  | #ifdef MATRIX SINGLE                    | 0477 | #ifdef MATRIX SINGLE   |
| 0478  | call $SSYE\overline{VR}('V','V','U',n)$ | 0478 | call $SSYEVR('V','V','U',n)$                                   |
| 0479  | Supp, WSize, WorkSize, ISize            | 0479 | Supp, WSize, WorkSize, I                                       |
| 0480  | #else                                   | 0480 | #else  |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 481</pre> |      | <pre>/lplopa/Compare/camb_des/Matrix_u 90, Top line: 481</pre> |
|------|--|------|--|
| 0481 | call DSYEVR('V','V','U',n  | 0481 | call DSYEVR('V','V','U',n                                      |
| 0482 | Supp, WSize, WorkSize, ISize                                       | 0482 | Supp, WSize, WorkSize, I                                       |
| 0483 | #endif   | 0483 | #endif   |
| 0484 | <pre>WorkSize = Real(WSize(1))</pre>                               | 0484 | <pre>WorkSize = Real(WSize(1))</pre>                           |
| 0485 | LIWork = ISize(1)  | 0485 | LIWork = ISize(1)  |
| 0486 | allocate(Work(WorkSize),I  | 0486 | allocate(Work(WorkSize),I                                      |
| 0487 | #ifdef MATRIX SINGLE   | 0487 | #ifdef MATRIX SINGLE   |
| 0488 | call $SSYE\overline{VR}('V','V','U',n)$                            | 0488 | call SSYE $\overline{V}$ R('V','V','U',n                       |
| 0489 | Supp, Work, WorkSize, IWork,                                       | 0489 | Supp, Work, WorkSize, IW                                       |
| 0490 | #else  | 0490 | #else  |
| 0491 | call DSYEVR('V','V','U',n  | 0491 | call DSYEVR('V','V','U',n                                      |
| 0492 | Supp, Work, WorkSize, IWork,                                       | 0492 | Supp, Work, WorkSize, IW                                       |
| 0493 | #endif   | 0493 | #endif   |
| 0494 | deallocate(Supp,Work,IWor  | 0494 | deallocate(Supp,Work,IWor                                      |
| 0495 | if (ierr /= 0) call MpiSt  | 0495 | if (ierr /= 0) call MpiSt                                      |
| 0496 | M(1:n,1:nfound) = tmp(1:n  | 0496 | M(1:n,1:nfound) = tmp(1:n                                      |
| 0497 | deallocate(tmp)  | 0497 | deallocate(tmp)  |
| 0498 | call Matrix_End('Matrix_D  | 0498 | call Matrix_End('Matrix_D                                      |
| 0499 | <u> </u>   | 0499 | <del>_</del> _ · · · _   |
| 0500 | <pre>end subroutine Matrix_Dia</pre>                               | 0500 | <pre>end subroutine Matrix_Dia</pre>                           |
| 0501 |  | 0501 | , <del></del>  |
| 0502 |  | 0502 |  |
| 0503 | <pre>subroutine Matrix_CDiagon</pre>                               | 0503 | subroutine Matrix_CDiagon                                      |
| 0504 | !Complex version. Does m   | 0504 | !Complex version. Does m                                       |
| 0505 | !Assumes up to nfound val  | 0505 | !Assumes up to nfound val                                      |
| 0506 | integer, intent(in) :: n   | 0506 | integer, intent(in) :: n                                       |
| 0507 | <pre>complex(dm), intent(inout</pre>                               | 0507 | <pre>complex(dm), intent(inout</pre>                           |
| 0508 | real(dm), intent(out) ::   | 0508 | real(dm), intent(out) ::                                       |
| 0509 | real(dm), intent(in) :: e  | 0509 | real(dm), intent(in) :: e                                      |
| 0510 | <pre>integer, intent(inout) ::</pre>                               | 0510 | <pre>integer, intent(inout) ::</pre>                           |

```
/Users/lplopa/Compare/camb simdata/Matr
                                         /Users/lplopa/Compare/camb des/Matrix u
ix utils.F90, Top line: 51\overline{1}
                                         tils.F90, Top line: 511
0511
                                         0511
                                                      integer ierr, worksize, L
             integer ierr, worksize, L
             real(dm), allocatable, di
                                         0512
                                                      real(dm), allocatable, di
0512
             complex(dm), allocatable,
                                         0513
                                                      complex(dm), allocatable,
0513
0514
             complex(dm), allocatable,
                                         0514
                                                      complex(dm), allocatable,
0515
             integer, allocatable, dime
                                         0515
                                                      integer, allocatable, dime
0516
                                         0516
             complex(dm) wsize(1)
                                                      complex(dm) wsize(1)
0517
             real(dm) Rsize(1), atol
                                         0517
                                                      real(dm) Rsize(1), atol
0518
             integer ISize(1)
                                         0518
                                                      integer ISize(1)
0519
                                         0519
0520
             atol = 1d-9
                                         0520
                                                      atol = 1d-9
0521
             call Matrix Start('Matrix
                                         0521
                                                      call Matrix Start('Matrix
0522
                                         0522
             allocate(tmp(n,nfound))
                                                      allocate(tmp(n, nfound))
0523
             allocate(Supp(n))
                                         0523
                                                      allocate(Supp(n))
0524
                                         0524
             ! Query
                                                      ! Query
0525
             WorkSize = -1
                                         0525
                                                      WorkSize = -1
0526
                                         0526
             LRWork = -1
                                                      LRWork = -1
0527
             LIWork = -1
                                         0527
                                                      LIWork = -1
0528
        #ifdef MATRIX SINGLE
                                         0528
                                                  #ifdef MATRIX SINGLE
             call CHEEVR('V','V','U',n 0529
0529
                                                      call CHEEVR('V','V','U',n
0530
             Supp, WSize, WorkSize, RSize
                                         0530
                                                           Supp, WSize, WorkSize, R
0531
                                         0531
                                                  #else
        #else
0532
             call ZHEEVR('V','V','U',n
                                         0532
                                                      call ZHEEVR('V','V','U',n
0533
             Supp, WSize, WorkSize, RSize
                                         0533
                                                           Supp, WSize, WorkSize, R
0534
        #endif
                                         0534
                                                  #endif
0535
             WorkSize = Real(WSize(1))
                                         0535
                                                      WorkSize = Real(WSize(1))
0536
             LRWork = RSize(1)
                                         0536
                                                      LRWork = RSize(1)
0537
             LIWork = ISize(1)
                                         0537
                                                      LIWork = ISize(1)
0538
             allocate(Work(WorkSize),R 0538
                                                      allocate(Work(WorkSize),R
                                                  #ifdef MATRIX SINGLE
0539
                                         0539
        #ifdef MATRIX SINGLE
0540
             call CHEEVR('V','V','U',n 0540
                                                      call CHEEVR('V','V','U',n
```

|      | /lplopa/Compare/camb_simdata/Matr<br>ls.F90, Top line: 541 |      | /lplopa/Compare/camb_des/Matrix_u<br>90, Top line: 541 |
|------|--|------|--|
|      |  |      |  |
| 0541 | Supp, Work, WorkSize, RWork,                               | 0541 | Supp, Work, WorkSize, RW                               |
| 0542 | #else  | 0542 | #else  |
| 0543 | call ZHEEVR('V','V','U',n                                  | 0543 | call ZHEEVR('V','V','U',n                              |
| 0544 | Supp, Work, WorkSize, RWork,                               | 0544 | Supp, Work, WorkSize, RW                               |
| 0545 | #endif   | 0545 | #endif   |
| 0546 | deallocate(Supp, Work, RWor                                | 0546 | deallocate(Supp,Work,RWor                              |
| 0547 | if (ierr /= 0) call MpiSt                                  | 0547 | if (ierr /= 0) call MpiSt                              |
| 0548 | M(1:n,1:nfound) = tmp(1:n                                  | 0548 | M(1:n,1:nfound) = tmp(1:n                              |
| 0549 | deallocate(tmp)  | 0549 | deallocate(tmp)  |
| 0550 | call Matrix_End('Matrix_C                                  | 0550 | call Matrix_End('Matrix_C                              |
| 0551 | _  | 0551 |  |
| 0552 |  | 0552 |  |
| 0553 | end subroutine   | 0553 | end subroutine   |
| 0554 |  | 0554 |  |
| 0555 |  | 0555 |  |
| 0556 | subroutine Matrix CDiagon                                  | 0556 | subroutine Matrix CDiagon                              |
| 0557 | !Complex version. Does m                                   | 0557 | !Complex version. Does m                               |
| 0558 | integer, intent(in) :: n                                   | 0558 | <pre>integer, intent(in) :: n</pre>                    |
| 0559 | complex(dm), intent(inout                                  | 0559 | complex(dm), intent(inout                              |
| 0560 | real(dm), intent(out) ::                                   | 0560 | real(dm), intent(out) ::                               |
| 0561 | integer ierr, tmpsize ,is                                  | 0561 | integer ierr, tmpsize ,is                              |
| 0562 | real(dm), allocatable, di                                  |      | real(dm), allocatable, di                              |
| 0563 | complex(dm), allocatable,                                  | 0563 | complex(dm), allocatable,                              |
| 0564 | integer, allocatable, dime                                 | 0564 | integer, allocatable, dime                             |
| 0565 |  | 0565 |  |
| 0566 | call Matrix Start('CDiago                                  | 0566 | call Matrix Start('CDiago                              |
| 0567 |  | 0567 |  |
| 0568 | <pre>if (matrix method == Mat</pre>                        | 0568 | if (matrix method == Mat                               |
| 0569 | $^{\cdot}$ !Divide and conquer                             | 0569 | $^{\cdot}$ !Divide and conquer                         |
| 0570 | tmpsize = 2*N + N**2                                       | 0570 | tmpsize = 2*N + N**2                                   |
|      |  |      |  |

```
/Users/lplopa/Compare/camb simdata/Matr
                                         /Users/lplopa/Compare/camb des/Matrix u
ix utils.F90, Top line: 57\overline{1}
                                         tils.F90, Top line: 571
                                         0571
0571
                 rworksize = 1 + 4*N
                                                          rworksize = 1 + 4*N
                                         0572
0572
                 isize = (2 + 5*N)*4
                                                           isize = (2 + 5*N)*4
                                         0573
0573
                 allocate(tmp(tmpsize)
                                                           allocate(tmp(tmpsize)
0574
                 allocate(iwork(isize)
                                         0574
                                                           allocate(iwork(isize)
        #ifdef MATRIX SINGLE
                                         0575
                                                  #ifdef MATRIX SINGLE
0575
0576
                 call CHEEVD('V','U',n 0576
                                                          call CHEEVD('V','U',n
0577
        #else
                                         0577
                                                  #else
                 call ZHEEVD('V','U',n
                                                          call ZHEEVD('V','U',n
0578
                                         0578
0579
        #endif
                                         0579
                                                  #endif
0580
                                         0580
                 deallocate(iwork)
                                                          deallocate(iwork)
0581
                                         0581
0582
                                         0582
             else
                                                      else
0583
                                         0583
0584
                                         0584
             rworksize = max(1, 3*n-2)
                                                          rworksize = max(1, 3)
        #ifdef MATRIX SINGLE
                                         0585
                                                  #ifdef MATRIX SINGLE
0585
0586
                                         0586
                                                          tmpsize = max( (ILAEN
             tmpsize = max( (ILAENV wr
0587
             allocate(tmp(tmpsize),rwo
                                         0587
                                                           allocate(tmp(tmpsize)
                                         0588
0588
             call CHEEV('V','U',n,m,n,
                                                          call CHEEV('V','U',n,
0589
                                         0589
                                                  #else
        #else
                                                          tmpsize = max((ILAEN))
0590
             tmpsize = max( (ILAENV wr
                                         0590
                                                          allocate(tmp(tmpsize)
0591
                                         0591
             allocate(tmp(tmpsize),rwo
             call ZHEEV('V','U',n,m,n,
0592
                                         0592
                                                          call ZHEEV('V','U',n,
        #endif
                                                  #endif
0593
                                         0593
0594
             end if
                                         0594
                                                      end if
0595
                                         0595
0596
                                         0596
             if (ierr /= 0) call MpiSt
                                                      if (ierr /= 0) call MpiSt
0597
             deallocate(tmp,rwork)
                                         0597
                                                      deallocate(tmp,rwork)
0598
                                         0598
0599
             call Matrix End('CDiagona
                                         0599
                                                      call Matrix End('CDiagona
0600
                                         0600
```

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 601</pre> |      | /lplopa/Compare/camb_des/Matrix_u<br>'90, Top line: 601 |
|------|--|------|---|
| 0601 | end subroutine Matrix CDi  | 0601 | end subroutine Matrix CDi                               |
| 0602 | <del>-</del>   | 0602 | <del>-</del>  |
| 0603 | <pre>function Matrix CTrace(M)</pre>                               | 0603 | function Matrix CTrace(M)                               |
| 0604 | complex(dm), intent(in):   | 0604 | complex(dm), intent(in):                                |
| 0605 | complex(dm) tmp, Matrix CT   | 0605 | complex(dm) tmp, Matrix CT                              |
| 0606 | integer i  | 0606 | integer i   |
| 0607 |  | 0607 |   |
| 0608 | if (size(M,dim=1) /= size  | 0608 | <pre>if (size(M,dim=1) /= size</pre>                    |
| 0609 | tmp =0   | 0609 | tmp =0  |
| 0610 | do i=1,size(M,dim=1)   | 0610 | <pre>do i=1,size(M,dim=1)</pre>                         |
| 0611 | tmp = tmp + M(i,i)   | 0611 | tmp = tmp + M(i,i)                                      |
| 0612 | end do   | 0612 | end do  |
| 0613 | Matrix_CTrace = tmp  | 0613 | Matrix_CTrace = tmp                                     |
| 0614 | <del>-</del>   | 0614 | _   |
| 0615 | <pre>end function Matrix_CTrac</pre>                               | 0615 | end function Matrix_CTrac                               |
| 0616 |  | 0616 | _   |
| 0617 | <pre>function Matrix_Trace(M)</pre>                                | 0617 | <pre>function Matrix_Trace(M)</pre>                     |
| 0618 | real(dm), intent(in) :: M  | 0618 | real(dm), intent(in) :: M                               |
| 0619 | real(dm) tmp,Matrix_Trace  | 0619 | real(dm) tmp,Matrix_Trace                               |
| 0620 | integer i  | 0620 | integer i   |
| 0621 |  | 0621 |   |
| 0622 | if (size(M,dim=1) /= size  | 0622 | if (size(M,dim=1) /= size                               |
| 0623 | tmp = 0  | 0623 | tmp =0  |
| 0624 | <pre>do i=1,size(M,dim=1)</pre>                                    | 0624 | do i=1,size(M,dim=1)                                    |
| 0625 | tmp = tmp + M(i,i)   | 0625 | tmp = tmp + M(i,i)                                      |
| 0626 | end do   | 0626 | end do  |
| 0627 | $\mathtt{Matrix\_Trace} = \mathtt{tmp}$                            | 0627 | Matrix_Trace = tmp                                      |
| 0628 |  | 0628 |   |
| 0629 | end function Matrix_Trace  | 0629 | end function Matrix_Trace                               |
| 0630 |  | 0630 |   |

|      | /lplopa/Compare/camb_simdata/Matr<br>ls.F90, Top line: 631 |      | /lplopa/Compare/camb_des/Matrix_u<br>90, Top line: 631 |
|------|--|------|--|
| 0631 |  | 0631 |  |
| 0632 | function MatrixSym LogDet                                  | 0632 | function MatrixSym LogDet                              |
| 0633 | real(dm), intent(in) :: m                                  | 0633 | real(dm), intent( $\overline{in}$ ) :: m               |
| 0634 | real(dm) logDet `  | 0634 | real(dm) logDet  |
| 0635 | real(dm) Tmp(size(mat,dim                                  | 0635 | real(dm) Tmp(size(mat,dim                              |
| 0636 | integer i  | 0636 | integer i  |
| 0637 |  | 0637 |  |
| 0638 | if (size(mat,dim=1) /= si                                  | 0638 | if (size(mat,dim=1) /= si                              |
| 0639 | Tmp = mat  | 0639 | Tmp = mat  |
| 0640 | <pre>call Matrix Cholesky(tmp)</pre>                       | 0640 | call Matrix Cholesky(tmp)                              |
| 0641 | logDet =0  | 0641 | logDet =0  |
| 0642 | <pre>do i=1, size(mat,dim=1)</pre>                         | 0642 | <pre>do i=1, size(mat,dim=1)</pre>                     |
| 0643 | <pre>logDet = logDet + lo</pre>                            | 0643 | logDet = logDet + lo                                   |
| 0644 | end do   | 0644 | end do   |
| 0645 | logDet = 2. dm*logDet                                      | 0645 | logDet = 2. dm*logDet                                  |
| 0646 | <u> </u>   | 0646 | _  |
| 0647 | <pre>end function MatrixSym_Lo</pre>                       | 0647 | end function MatrixSym_Lo                              |
| 0648 | <del>-</del>   | 0648 |  |
| 0649 |  | 0649 |  |
| 0650 | <pre>subroutine Matrix_CRotate</pre>                       | 0650 | subroutine Matrix_CRotate                              |
| 0651 | !Gets U^dag Mat U  | 0651 | !Gets U^dag Mat U                                      |
| 0652 | integer, intent(in) ::m                                    | 0652 | <pre>integer, intent(in) ::m</pre>                     |
| 0653 | <pre>complex(dm), intent(in) :</pre>                       | 0653 | <pre>complex(dm), intent(in) :</pre>                   |
| 0654 | <pre>complex(dm) Out(:,:)</pre>                            | 0654 | <pre>complex(dm) Out(:,:)</pre>                        |
| 0655 | <pre>complex(dm), dimension(:,</pre>                       | 0655 | <pre>complex(dm), dimension(:,</pre>                   |
| 0656 | integer n  | 0656 | integer n  |
| 0657 | <pre>logical, intent(in), opti</pre>                       | 0657 | logical, intent(in), opti                              |
| 0658 | logical :: triang  | 0658 | logical :: triang                                      |
| 0659 | <del>-</del>   | 0659 |  |
| 0660 | call Matrix_Start('CRotat                                  | 0660 | call Matrix_Start('CRotat                              |

```
/Users/lplopa/Compare/camb des/Matrix u
/Users/lplopa/Compare/camb simdata/Matr
                                          tils.F90, Top line: 661
ix utils.F90, Top line: 66\overline{1}
                                          0661
0661
                                          0662
0662
             if (present(triangular))
                                                       if (present(triangular))
0663
                 triang=triangular
                                          0663
                                                           triang=triangular
0664
                                          0664
                                                       else
             else
0665
                 triang=.false.
                                          0665
                                                           triang=.false.
0666
             end if
                                          0666
                                                       end if
0667
                                          0667
0668
             n = Size(Mat,DIM=1)
                                          0668
                                                       n = Size(Mat, DIM=1)
0669
             if (n /= Size(Mat,DIM=2))
                                          0669
                                                       if (n /= Size(Mat,DIM=2))
                                                       if (n /= Size(U,DIM=1)) c
0670
             if (n /= Size(U,DIM=1)) c
                                          0670
0671
                                          0671
                                                       if (Size(Out,DIM=1) < m.
             if (Size(Out,DIM=1) < m .</pre>
0672
                                          0672
                                                           call MpiStop('Matrix
             call MpiStop('Matrix CRot
                                          0673
0673
0674
                                          0674
                                                       if (matrix method == Mat
             if (matrix method == Mat
                                                           Out = matmul(matmul(t
0675
                 Out = matmul(matmul(t
                                          0675
0676
             else
                                          0676
                                                       else
0677
         #ifdef MATRIX SINGLE
                                          0677
                                                   #ifdef MATRIX SINGLE
0678
                 if (triang) then
                                          0678
                                                           if (triang) then
0679
                      if (m/=n) call Mp
                                          0679
                                                                if (m/=n) call Mp
                                                                call CHEMM('L','U
0680
                      call CHEMM('L','U
                                          0680
0681
                                          0681
                      call CTRMM('Left'
                                                                call CTRMM('Left'
0682
                 else
                                          0682
                                                           else
                                          0683
0683
                      allocate(C(n,m))
                                                                allocate(C(n,m))
0684
                                          0684
                      call CHEMM('L','U
                                                                call CHEMM('L','U
0685
                      call CGEMM('C','N
                                          0685
                                                                call CGEMM('C','N
0686
                                          0686
                      deallocate(C)
                                                                deallocate(C)
0687
                 end if
                                          0687
                                                           end if
0688
         #else
                                          0688
                                                   #else
0689
                                          0689
                 if (triang) then
                                                            if (triang) then
0690
                      if (m/=n) call Mp
                                         0690
                                                                if (m/=n) call Mp
```

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 691</pre> |      | <pre>/lplopa/Compare/camb_des/Matrix_u 90, Top line: 691</pre> |
|------|--|------|--|
| 0691 | call ZHEMM('L','U  | 0691 | call ZHEMM('L','U  |
| 0692 | call ZTRMM('Left'  | 0692 | call ZTRMM('Left'  |
| 0693 | else   | 0693 | else   |
| 0694 | allocate(C(n,m))   | 0694 | allocate(C(n,m))   |
| 0695 | call ZHEMM('L','U  | 0695 | call ZHEMM('L','U  |
| 0696 | call ZGEMM('C','N  | 0696 | call ZGEMM('C','N  |
| 0697 | deallocate(C)  | 0697 | deallocate(C)  |
| 0698 | end if   | 0698 | end if   |
| 0699 | #endif   | 0699 | #endif   |
| 0700 | end if   | 0700 | end if   |
| 0701 | call Matrix End('CRotateS  | 0701 | call Matrix End('CRotateS                                      |
| 0702 | _ `  | 0702 | _ `  |
| 0703 |  | 0703 |  |
| 0704 | end subroutine Matrix CRo  | 0704 | end subroutine Matrix CRo                                      |
| 0705 | <del>_</del>   | 0705 | _  |
| 0706 | subroutine Matrix RotateS  | 0706 | subroutine Matrix RotateS                                      |
| 0707 | !Gets U^T Mat U  | 0707 | !Gets U^T Mat U  |
| 0708 | !If triangular U = Upper   | 0708 | !If triangular U = Upper                                       |
| 0709 | <pre>integer, intent(in) ::m</pre>                                 | 0709 | <pre>integer, intent(in) ::m</pre>                             |
| 0710 | real(dm), intent(in) :: M  | 0710 | real(dm), intent(in) :: M                                      |
| 0711 | real(dm) Out(:,:)  | 0711 | real(dm) Out(:,:)  |
| 0712 | <pre>real(dm), dimension(:,:),</pre>                               | 0712 | <pre>real(dm), dimension(:,:),</pre>                           |
| 0713 | logical, intent(in), opti  | 0713 | logical, intent(in), opti                                      |
| 0714 | logical triang   | 0714 | logical triang   |
| 0715 | integer n  | 0715 | integer n  |
| 0716 |  | 0716 |  |
| 0717 | call Matrix_Start('Rotate  | 0717 | call Matrix_Start('Rotate                                      |
| 0718 |  | 0718 |  |
| 0719 | <pre>if (present(triangular))</pre>                                | 0719 | <pre>if (present(triangular))</pre>                            |
| 0720 | triang=triangular  | 0720 | triang=triangular  |

```
/Users/lplopa/Compare/camb simdata/Matr
                                          /Users/lplopa/Compare/camb des/Matrix u
                                         tils.F90, Top line: 721
ix utils.F90, Top line: 721
                                          0721
0721
             else
                                                       else
0722
                 triang=.false.
                                          0722
                                                           triang=.false.
0723
             end if
                                          0723
                                                       end if
0724
                                          0724
                                          0725
0725
             n = Size(Mat,DIM=1)
                                                       n = Size(Mat, DIM=1)
             if (n /= Size(Mat,DIM=2))
                                                       if (n /= Size(Mat,DIM=2))
0726
                                         0726
             if (n /= Size(U,DIM=1)) c
0727
                                         0727
                                                       if (n /= Size(U,DIM=1)) c
0728
             if (Size(Out,DIM=1) < m .</pre>
                                          0728
                                                       if (Size(Out,DIM=1) < m .</pre>
0729
             call MpiStop('Matrix Rota
                                         0729
                                                           call MpiStop('Matrix
0730
                                          0730
0731
                                          0731
             if (matrix method == Mat
                                                       if (matrix method == Mat
0732
                                         0732
                 Out = matmul(matmul(t
                                                           Out = matmul(matmul(t
0733
                                          0733
             else
                                                       else
0734
                                          0734
                                                  #ifdef MATRIX SINGLE
         #ifdef MATRIX SINGLE
0735
                 if (triang) then
                                         0735
                                                           if (triang) then
0736
                      if (m/=n) call Mp
                                         0736
                                                                if (m/=n) call Mp
                     call SSYMM('L','U
                                                               call SSYMM('L','U
0737
                                         0737
0738
                      call STRMM('Left'
                                          0738
                                                               call STRMM('Left'
0739
                                          0739
                 else
                                                           else
0740
                     allocate(C(n,m))
                                          0740
                                                               allocate(C(n,m))
                     call SSYMM('L','U
                                         0741
0741
                                                               call SSYMM('L','U
                                                               call SGEMM('T','N
0742
                     call SGEMM('T','N
                                         0742
0743
                                          0743
                      deallocate(C)
                                                               deallocate(C)
0744
                 end if
                                          0744
                                                           end if
0745
        #else
                                          0745
                                                  #else
0746
                 if (triang) then
                                          0746
                                                           if (triang) then
0747
                      if (m/=n) call Mp 0747
                                                                if (m/=n) call Mp
                     call DSYMM('L','U
                                                               call DSYMM('L','U
0748
                                         0748
0749
                      call DTRMM('Left'
                                          0749
                                                               call DTRMM('Left'
0750
                                          0750
                                                           else
                 else
```

|      | /lplopa/Compare/camb_simdata/Matr<br>ls.F90, Top line: 751 |      | <pre>/lplopa/Compare/camb_des/Matrix_u 90, Top line: 751</pre> |
|------|--|------|--|
| 0751 | allocate(C(n,m))   | 0751 | allocate(C(n,m))   |
| 0752 | call DSYMM('L', 'U   | 0752 | call DSYMM('Ĺ',´U  |
| 0753 | call DGEMM('T','N  | 0753 | call DGEMM('T','N  |
| 0754 | deallocate(C)  | 0754 | deallocate(C)  |
| 0755 | end if   | 0755 | end if   |
| 0756 | #endif   | 0756 | #endif   |
| 0757 | end if   | 0757 | end if   |
| 0758 | call Matrix End('RotateSy                                  | 0758 | call Matrix End('RotateSy                                      |
| 0759 | <b>–</b> `   | 0759 | _ ` ` -  |
| 0760 |  | 0760 |  |
| 0761 | end subroutine Matrix Rot                                  | 0761 | end subroutine Matrix Rot                                      |
| 0762 |  | 0762 | _  |
| 0763 |  | 0763 |  |
| 0764 | subroutine Matrix RotateA                                  | 0764 | subroutine Matrix RotateA                                      |
| 0765 | !Gets U^T Mat U —  | 0765 | !Gets U^T Mat U  |
| 0766 | !Where Mat = -Mat^T  | 0766 | !Where Mat = -Mat^T  |
| 0767 | <pre>integer, intent(in) ::m</pre>                         | 0767 | <pre>integer, intent(in) ::m</pre>                             |
| 0768 | real(dm), intent(in) :: M                                  | 0768 | real(dm), intent(in) :: M                                      |
| 0769 | real(dm) Out(:,:)  | 0769 | real(dm) Out(:,:)  |
| 0770 | real(dm), dimension(:,:),                                  | 0770 | <pre>real(dm), dimension(:,:),</pre>                           |
| 0771 | integer i, j, n  | 0771 | integer i,j,n  |
| 0772 |  | 0772 |  |
| 0773 | call Matrix Start('Rotate                                  | 0773 | call Matrix Start('Rotate                                      |
| 0774 | _ `  | 0774 | _ `  |
| 0775 | n = Size(Mat,DIM=1)  | 0775 | n = Size(Mat,DIM=1)  |
| 0776 | if $(n /= Size(Mat,DIM=2))$                                | 0776 | <pre>if (n /= Size(Mat,DIM=2))</pre>                           |
| 0777 | if (n /= Size(U,DIM=1)) c                                  | 0777 | if (n /= Size(U,DIM=1)) c                                      |
| 0778 | if (Size(Out, DIM=1) < m .                                 | 0778 | if (Size(Out, DIM=1) < m .                                     |
| 0779 | call MpiStop('Matrix Rota                                  | 0779 | call MpiStop('Matrix   |
| 0780 | <b>- -</b> ` <b>-</b>                                      | 0780 |  |

```
/Users/lplopa/Compare/camb des/Matrix u
/Users/lplopa/Compare/camb simdata/Matr
                                            tils.F90, Top line: 781
ix utils.F90, Top line: 781
0781
              if (matrix method == Mat
                                            0781
                                                          if (matrix method == Mat
                  Out = \overline{\text{matmul}}(\text{matmul}(\overline{\text{t}}))
                                            0782
                                                              Out = \overline{\text{matmul}}(\text{matmul}(\overline{\text{t}}))
0782
0783
              else
                                            0783
                                                          else
0784
                                            0784
                  allocate(C(n,m))
                                                               allocate(C(n,m))
                                                              C = U(1:n,1:m)
0785
                  C = U(1:n,1:m)
                                            0785
         #ifdef MATRIX SINGLE
0786
                                            0786
                                                     #ifdef MATRIX SINGLE
0787
                  call STRMM('Left','Lo
                                            0787
                                                               call STRMM('Left','Lo
0788
                  call SGEMM('T','N',m,
                                            0788
                                                               call SGEMM('T','N',m,
0789
         #else
                                            0789
                                                     #else
0790
                                            0790
                  call DTRMM('Left','Lo
                                                               call DTRMM('Left','Lo
0791
                  call DGEMM('T','N',m,
                                            0791
                                                               call DGEMM('T','N',m,
0792
         #endif
                                            0792
                                                     #endif
0793
                                            0793
                  deallocate(C)
                                                               deallocate(C)
              end if
0794
                                            0794
                                                          end if
0795
                                            0795
0796
                                            0796
                                                          do i=1, m
              do i=1, m
0797
                  do j=1,i
                                            0797
                                                               do j=1,i
0798
                       Out(j,i) = Out(j,i)
                                            0798
                                                                   Out(j,i) = Out(j,i)
0799
                                            0799
                                                                   out(i,i) = -Out(i
                       out(i,j) = -Out(j)
0800
                  end do
                                            0800
                                                               end do
0801
                                            0801
                                                          end do
              end do
0802
                                            0802
0803
                                            0803
              call Matrix End('RotateAn
                                                          call Matrix End('RotateAn
0804
                                            0804
0805
              end subroutine Matrix Rot
                                            0805
                                                          end subroutine Matrix Rot
0806
                                            0806
0807
              subroutine Matrix CMult S
                                            0807
                                                          subroutine Matrix CMult S
                                            0808
                                                          complex(dm), intent(in) :
8080
              complex(dm), intent(in) :
0809
              complex(dm) Out(:,:)
                                            0809
                                                          complex(dm) Out(:,:)
              complex(dm), intent(in),
                                                          complex(dm), intent(in),
0810
                                            0810
```

```
/Users/lplopa/Compare/camb simdata/Matr
                                         /Users/lplopa/Compare/camb des/Matrix u
                                         tils.F90, Top line: 811
ix utils.F90, Top line: 81\overline{1}
0811
                                         0811
             complex(dm)
                                                      complex(dm)
                          mult, beta
                                                                    mult, beta
0812
                                         0812
             integer n,m
                                                       integer n,m
0813
                                         0813
0814
                                         0814
             call Matrix Start('CMult
                                                       call Matrix Start('CMult
0815
                                         0815
0816
                                         0816
            m = Size(Mat,DIM=1)
                                                      m = Size(Mat,DIM=1)
0817
             n = Size(U,DIM=2)
                                         0817
                                                      n = Size(U,DIM=2)
0818
             if (n /= Size(Mat,DIM=2)
                                         0818
                                                       if (n /= Size(Mat,DIM=2)
0819
             call MpiStop('Matrix CMul
                                         0819
                                                           call MpiStop('Matrix
0820
             if (present(a)) then
                                         0820
                                                       if (present(a)) then
0821
                 mult = a
                                         0821
                                                           mult = a
0822
                                         0822
             else
                                                       else
0823
                                         0823
                 mult = COne
                                                           mult = COne
0824
             end if
                                                       end if
                                         0824
0825
             if (present(b)) then
                                         0825
                                                       if (present(b)) then
0826
                                         0826
                 beta = b
                                                           beta = b
0827
                                         0827
             else
                                                       else
0828
                 beta = CZero
                                         0828
                                                           beta = CZero
0829
                                         0829
                                                       end if
             end if
0830
             if (matrix method == Mat
                                         0830
                                                       if (matrix method == Mat
0831
                                         0831
                                                           if (beta /= CZero) th
                 if (beta /= CZero) th
0832
                     out = a*MatMul(Ma
                                         0832
                                                               out = a*MatMul(Ma
0833
                                         0833
                                                           else
                 else
0834
                                         0834
                     out = MatMul(Mat,
                                                               out = MatMul(Mat,
0835
                      if (mult /= COne)
                                         0835
                                                               if (mult /= COne)
0836
                                         0836
                 end if
                                                           end if
0837
                                         0837
             else
                                                      else
        #ifdef MATRIX SINGLE
                                                  #ifdef MATRIX SINGLE
0838
                                         0838
0839
                                         0839
                                                           call CHEMM('R','U',m,
                 call CHEMM('R','U',m,
0840
                                         0840
                                                  #else
        #else
```

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 841</pre> |      | /lplopa/Compare/camb_des/Matrix_u<br>90, Top line: 841 |
|------|--|------|--|
| 0841 |  | 0841 | call ZHEMM('R','U',m,                                  |
| 0842 | call ZHEMM('R','U',m,<br>#endif                                    | 0842 | #endif   |
| 0842 |  | 0842 |  |
| 0844 | end if   | 0844 | end if   |
|      | call Waterin End//CWult Co.  |      | coll Matrix End/(CM) t Co.                             |
| 0845 | call Matrix_End('CMult_Sy  | 0845 | call Matrix_End('CMult_Sy                              |
| 0846 |  | 0846 |  |
| 0847 | end subroutine Matrix_CMu  | 0847 | end subroutine Matrix_CMu                              |
| 0848 |  | 0848 |  |
| 0849 |  | 0849 |  |
| 0850 | subroutine Matrix_CMult_S  | 0850 | subroutine Matrix_CMult_S                              |
| 0851 | <pre>complex(dm), intent(in) :</pre>                               | 0851 | complex(dm), intent(in):                               |
| 0852 | <pre>complex(dm) Out(:,:)</pre>                                    | 0852 | <pre>complex(dm) Out(:,:)</pre>                        |
| 0853 | <pre>complex(dm), intent(in),</pre>                                | 0853 | <pre>complex(dm), intent(in),</pre>                    |
| 0854 | <pre>complex(dm) mult, beta</pre>                                  | 0854 | complex(dm) mult, beta                                 |
| 0855 | integer n,m  | 0855 | integer n,m  |
| 0856 |  | 0856 |  |
| 0857 | call Matrix Start('CMult   | 0857 | call Matrix Start('CMult                               |
| 0858 | _ ` _  | 0858 |  |
| 0859 | m = Size(Mat,DIM=1)  | 0859 | m = Size(Mat,DIM=1)                                    |
| 0860 | n = Size(U,DIM=2)  | 0860 | n = Size(U,DIM=2)                                      |
| 0861 | if $(m /= Size(U,DIM=1) .o$  | 0861 | if $(m /= Size(U,DIM=1) .o$                            |
| 0862 | call MpiStop('Matrix ĆMul  | 0862 | call MpiStop('Matrix_                                  |
| 0863 | if (present(a)) then   | 0863 | if (present(a)) then                                   |
| 0864 | $\mathbf{\hat{m}ult} = \mathbf{\hat{a}}$                           | 0864 | mult = a   |
| 0865 | else   | 0865 | else   |
| 0866 | mult = COne  | 0866 | mult = COne  |
| 0867 | end if   | 0867 | end if   |
| 0868 | if (present(b)) then   | 0868 | if (present(b)) then                                   |
| 0869 | beta = b   | 0869 | beta = b   |
| 0870 | else   | 0870 | else   |

```
/Users/lplopa/Compare/camb des/Matrix u
/Users/lplopa/Compare/camb simdata/Matr
ix utils.F90, Top line: 87\overline{1}
                                         tils.F90, Top line: 871
                                         0871
0871
                 beta = CZero
                                                           beta = CZero
             end if
                                         0872
                                                      end if
0872
                                         0873
0873
             if (matrix method == Mat
                                                       if (matrix method == Mat
0874
                 if (beta /= CZero) th
                                         0874
                                                           if (beta /= CZero) th
0875
                     out = a*MatMul(Ma
                                         0875
                                                               out = a*MatMul(Ma
0876
                                         0876
                 else
                                                           else
                                         0877
0877
                     out = MatMul(Mat,
                                                               out = MatMul(Mat,
0878
                      if (mult /= COne)
                                         0878
                                                               if (mult /= COne)
0879
                 end if
                                         0879
                                                           end if
                                         0880
0880
             else
                                                      else
0881
        #ifdef MATRIX SINGLE
                                         0881
                                                  #ifdef MATRIX SINGLE
0882
                                         0882
                                                           call CHEMM('L','U',m,
                 call CHEMM('L','U',m,
0883
                                         0883
                                                  #else
        #else
0884
                                         0884
                 call ZHEMM('L','U',m,
                                                           call ZHEMM('L','U',m,
                                                  #endif
0885
        #endif
                                         0885
             end if
                                         0886
0886
                                                      end if
0887
                                         0887
                                         0888
0888
             call Matrix End('CMult Sy
                                                      call Matrix End('CMult Sy
0889
                                         0889
0890
             end subroutine Matrix CMu
                                         0890
                                                      end subroutine Matrix CMu
0891
                                         0891
0892
                                         0892
0893
                                         0893
             subroutine Matrix CMult(M
                                                       subroutine Matrix CMult(M
0894
             ! Out = a*Mat U + b*out
                                         0894
                                                       ! Out = a*Mat U + b*out
0895
             complex(dm), intent(in) :
                                         0895
                                                      complex(dm), intent(in) :
0896
                                         0896
             complex(dm) Out(:,:)
                                                      complex(dm) Out(:,:)
0897
             complex(dm), intent(in),
                                         0897
                                                      complex(dm), intent(in),
                                                                    mult, beta
0898
             complex(dm) mult, beta
                                         0898
                                                      complex(dm)
0899
                                         0899
             integer m,n,k
                                                       integer m,n,k
0900
                                         0900
```

```
/Users/lplopa/Compare/camb des/Matrix u
/Users/lplopa/Compare/camb simdata/Matr
                                          tils.F90, Top line: 901
ix utils.F90, Top line: 901
                                          0901
0901
             call Matrix Start('CMult'
                                                       call Matrix Start('CMult'
                                          0902
0902
0903
             m = Size(Mat,DIM=1)
                                          0903
                                                       m = Size(Mat,DIM=1)
0904
             n = Size(U,DIM=2)
                                          0904
                                                       n = Size(U,DIM=2)
             k = Size(Mat, DIM=2)
                                                       k = Size(Mat, DIM=2)
0905
                                          0905
                                                       if (k /= Size(U,DIM=1)) c
0906
             if (k /= Size(U,DIM=1))
                                       c 0906
0907
             if (present(a)) then
                                          0907
                                                       if (present(a)) then
0908
                 mult = a
                                          0908
                                                           mult = a
0909
                                          0909
             else
                                                       else
0910
                                          0910
                 mult = COne
                                                           mult = COne
0911
             end if
                                          0911
                                                       end if
0912
                                          0912
             if (present(b)) then
                                                       if (present(b)) then
0913
                                          0913
                 beta = b
                                                           beta = b
0914
                                          0914
             else
                                                       else
0915
                                          0915
                 beta = CZero
                                                           beta = CZero
                                          0916
0916
             end if
                                                       end if
0917
                                          0917
0918
             if (matrix method == Mat
                                          0918
                                                       if (matrix method == Mat
                                                           if (be\overline{t}a /= CZero) th
0919
                 if (beta /= CZero) th
                                         0919
0920
                      out = a*MatMul(Ma
                                         0920
                                                                out = a*MatMul(Ma
0921
                                          0921
                 else
                                                           else
0922
                                          0922
                      out = MatMul(Mat,
                                                                out = MatMul(Mat,
                                                                if (mult /= COne)
0923
                      if (mult /= COne)
                                          0923
0924
                                          0924
                                                           end if
                 end if
0925
             else
                                          0925
                                                       else
0926
                                          0926
         #ifdef MATRIX SINGLE
                                                   #ifdef MATRIX SINGLE
0927
                 call CGEMM('N','N',m,
                                          0927
                                                           call CGEMM('N','N',m,
0928
         #else
                                          0928
                                                   #else
                 call ZGEMM('N','N',m,
0929
                                          0929
                                                           call ZGEMM('N','N',m,
0930
         #endif
                                          0930
                                                   #endif
```

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 931</pre> |      | <pre>/lplopa/Compare/camb_des/Matrix_u 90, Top line: 931</pre> |
|------|--|------|--|
| 0931 | end if   | 0931 | end if   |
| 0932 | call Matrix End('CMult')   | 0932 | call Matrix End('CMult')                                       |
| 0933 | , , , , , , , , , , , , , , , , , , ,                              | 0933 | ,  |
| 0934 |  | 0934 |  |
| 0935 | end subroutine Matrix CMu  | 0935 | end subroutine Matrix CMu                                      |
| 0936 | <del>-</del>   | 0936 | <del>-</del>   |
| 0937 |  | 0937 |  |
| 0938 | subroutine Matrix MultSq   | 0938 | subroutine Matrix MultSq                                       |
| 0939 | !U = a*Mat*U   | 0939 | !U = a*Mat*U   |
| 0940 | real(dm), intent(in) :: M  | 0940 | real(dm), intent(in) :: M                                      |
| 0941 | <pre>real(dm), intent(inout) :</pre>                               | 0941 | real(dm), intent(inout) :                                      |
| 0942 | real(dm), intent(in), opt  | 0942 | real(dm), intent(in), opt                                      |
| 0943 | real(dm) aa  | 0943 | real(dm) aa  |
| 0944 | integer m,n  | 0944 | integer m,n  |
| 0945 | <pre>real(dm), dimension(:,:),</pre>                               | 0945 | <pre>real(dm), dimension(:,:),</pre>                           |
| 0946 |  | 0946 |  |
| 0947 |  | 0947 |  |
| 0948 | <pre>m = Size(Mat,DIM=1)</pre>                                     | 0948 | <pre>m = Size(Mat,DIM=1)</pre>                                 |
| 0949 | <pre>n = Size(Mat,DIM=2)</pre>                                     | 0949 | n = Size(Mat,DIM=2)  |
| 0950 | if (m /= n) call MpiStop(  | 0950 | if (m /= n) call MpiStop(                                      |
| 0951 | m = Size(U,DIM=1)  | 0951 | m = Size(U,DIM=1)  |
| 0952 | n = Size(U,DIM=2)  | 0952 | n = Size(U,DIM=2)  |
| 0953 | if (m /= n) call MpiStop(  | 0953 | if (m /= n) call MpiStop(                                      |
| 0954 |  | 0954 |  |
| 0955 | allocate(tmp(n,n))   | 0955 | allocate(tmp(n,n))   |
| 0956 | <pre>if (present(a)) then</pre>                                    | 0956 | <pre>if (present(a)) then</pre>                                |
| 0957 | aa=a   | 0957 | aa=a   |
| 0958 | else   | 0958 | else   |
| 0959 | aa=ROne  | 0959 | aa=ROne  |
| 0960 | end if   | 0960 | end if   |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 961</pre> |      | /lplopa/Compare/camb_des/Matrix_u<br>90, Top line: 961 |
|------|--|------|--|
| 0961 |  | 0961 |  |
| 0962 | call Matrix Mult(Mat,U,tm  | 0962 | call Matrix Mult(Mat,U,tm                              |
| 0963 | U = tmp  | 0963 | U = tmp  |
| 0964 | deallocate(tmp)  | 0964 | deallocate(tmp)  |
| 0965 |  | 0965 |  |
| 0966 | end subroutine Matrix Mu   | 0966 | end subroutine Matrix Mu                               |
| 0967 |  | 0967 |  |
| 0968 | subroutine Matrix MultTri  | 0968 | subroutine Matrix MultTri                              |
| 0969 | ! Mat -> L Mat or Mat L w  | 0969 | ! Mat -> L Mat or Mat L w                              |
| 0970 | <pre>real(dm), intent(inout) :</pre>                               | 0970 | <pre>real(dm), intent(inout) :</pre>                   |
| 0971 | real(dm), intent(in) :: L  | 0971 | real(dm), intent(in) :: L                              |
| 0972 | character(LEN=*), intent(  | 0972 | character(LEN=*), intent(                              |
| 0973 | integer m,n  | 0973 | integer m,n  |
| 0974 |  | 0974 |  |
| 0975 | call Matrix_Start('Matrix  | 0975 | call Matrix Start('Matrix                              |
| 0976 | _ ` `  | 0976 | _ ` `  |
| 0977 | m = Size(Mat,DIM=1)  | 0977 | m = Size(Mat,DIM=1)                                    |
| 0978 | <pre>n = Size(Mat,DIM=2)</pre>                                     | 0978 | n = Size(Mat,DIM=2)                                    |
| 0979 | •  | 0979 |  |
| 0980 | if $(side(1:1)=='L')$ then   | 0980 | if $(side(1:1)=='L')$ then                             |
| 0981 | if (Size(L,DIM=2) /= :   | 0981 | <pre>if (Size(L,DIM=2) /= :</pre>                      |
| 0982 | else   | 0982 | else   |
| 0983 | if (Size(L,DIM=1) /=   | 0983 | if (Size(L,DIM=1) /=                                   |
| 0984 | end if   | 0984 | end if   |
| 0985 | $\#$ if def MATRIX_SINGLE  | 0985 | #ifdef MATRIX_SINGLE                                   |
| 0986 | call STRMM(side,'Lower','  | 0986 | call STRMM(side, 'Lower', '                            |
| 0987 | #else  | 0987 | #else  |
| 0988 | call DTRMM(side, 'Lower', '  | 0988 | call DTRMM(side, 'Lower', '                            |
| 0989 | #endif   | 0989 | #endif   |
| 0990 | call Matrix_End('Matrix_M  | 0990 | call Matrix_End('Matrix_M                              |

| /Users/lplopa/Compare/camb_simdata/Matr ix utils.F90, Top line: 991 |                                 | /Users/lplopa/Compare/camb_des/Matrix_u tils.F90, Top line: 991 |                                      |
|---|---------------------------------|---|--------------------------------------|
| 0991  |                                 | 0991  | 70, 10p line. 331                    |
| 0991  | and subrouting Matrix Mul       | 0991  | and subrouting Matrix Mul            |
| 0992  | end subroutine Matrix_Mul       | 0992  | <pre>end subroutine Matrix_Mul</pre> |
| 0994  |                                 | 0994  |                                      |
| 0995  |                                 | 0995  |                                      |
| 0996  | gubrouting Matrix Mult (Ma      | 0996  | subsouting Matrix Mult/Ma            |
|   | subroutine Matrix_Mult(Ma       |   | subroutine Matrix_Mult(Ma            |
| 0997  | ! Out = a*Mat U + b*out         | 0997  | ! Out = $a*Mat U + b*out$            |
| 0998  | real(dm), intent(in) :: M       | 0998  | real(dm), intent(in) :: M            |
| 0999  | real(dm) :: Out(:,:)            | 0999  | real(dm) :: Out(:,:)                 |
| 1000  | real(dm), intent(in), opt       | 1000  | real(dm), intent(in), opt            |
| 1001  | real(dm) mult, beta             | 1001  | real(dm) mult, beta                  |
| 1002  | integer m,n,k                   | 1002  | integer m,n,k                        |
| 1003  | moli Matada Otaat (126-14.1)    | 1003  | and 11 Makesian Object (120-141)     |
| 1004  | call Matrix_Start('Mult')       | 1004  | <pre>call Matrix_Start('Mult')</pre> |
| 1005  |                                 | 1005  |                                      |
| 1006  |                                 | 1006  | a' - (24-1 PT) 1)                    |
| 1007  | m = Size(Mat,DIM=1)             | 1007  | m = Size(Mat,DIM=1)                  |
| 1008  | n = Size(U,DIM=2)               | 1008  | n = Size(U,DIM=2)                    |
| 1009  | k = Size(Mat, DIM=2)            | 1009  | k = Size(Mat, DIM=2)                 |
| 1010  | if $(k /= Size(U,DIM=1))$ c     | 1010  | if $(k /= Size(U,DIM=1))$ c          |
| 1011  |                                 | 1011  |                                      |
| 1012  |                                 | 1012  |                                      |
| 1013  | if (present(a)) then            | 1013  | <pre>if (present(a)) then</pre>      |
| 1014  | mult = a                        | 1014  | mult = a                             |
| 1015  | else                            | 1015  | else                                 |
| 1016  | mult = ROne                     | 1016  | mult = ROne                          |
| 1017  | end if                          | 1017  | end if                               |
| 1018  | <pre>if (present(b)) then</pre> | 1018  | <pre>if (present(b)) then</pre>      |
| 1019  | beta = b                        | 1019  | beta = b                             |
| 1020  | else                            | 1020  | else                                 |

| /Users/lplopa/Compare/camb_simdata/Matr       |  | /Users/lp1opa/Compare/camb_des/Matrix_u |  |
|---|--|---|--|
| $ix\_utils.F90$ , Top line: $10\overline{2}1$ |  | tils.F90, Top line: 1021                |  |
| 1021  | beta = RZero   | 1021                                    | beta = RZero   |
| 1022  | end if   | 1022                                    | end if   |
| 1023  |  | 1023                                    |  |
| 1024  | if (matrix method == Mat   | 1024                                    | if (matrix method == Mat                               |
| 1025  | if (be $\overline{t}a$ /= RZero) $t\overline{h}$                 | 1025                                    | if (be $\overline{t}a$ /= RZero) $t\overline{h}$       |
| 1026  | out = a*MatMul(Ma)   | 1026                                    | out = a*MatMul(Ma                                      |
| 1027  | else   | 1027                                    | else   |
| 1028  | out = MatMul(Mat,  | 1028                                    | out = MatMul(Mat,                                      |
| 1029  | if (mult /= ROne)  | 1029                                    | if (mult /= ROne)                                      |
| 1030  | end if   | 1030                                    | end if `   |
| 1031  | else   | 1031                                    | else   |
| 1032  | #ifdef MATRIX SINGLE   | 1032                                    | #ifdef MATRIX SINGLE                                   |
| 1033  | $call \overline{S}GEMM('N','N',m,$                               | 1033                                    | $call \overline{SGEMM}('N','N',m,$                     |
| 1034  | #else  | 1034                                    | #else  |
| 1035  | call DGEMM('N','N',m,  | 1035                                    | call DGEMM('N','N',m,                                  |
| 1036  | #endif   | 1036                                    | #endif   |
| 1037  | end if   | 1037                                    | end if   |
| 1038  | call Matrix End('Mult')  | 1038                                    | call Matrix End('Mult')                                |
| 1039  | _ ` '  | 1039                                    | _ `  |
| 1040  |  | 1040                                    |  |
| 1041  | end subroutine Matrix Mul  | 1041                                    | end subroutine Matrix Mul                              |
| 1042  | <b>–</b>   | 1042                                    | <del>_</del>   |
| 1043  | subroutine Matrix Mult Sy  | 1043                                    | subroutine Matrix Mult Sy                              |
| 1044  | $real(dm)$ , $intent(\overline{in}) : \overline{:} \overline{M}$ | 1044                                    | real(dm), intent( $\overline{i}$ n) : $\overline{i}$ M |
| 1045  | real(dm) Out(:,:)  | 1045                                    | real(dm) Out(:,:)                                      |
| 1046  | real(dm), intent(in), opt  | 1046                                    | real(dm), intent(in), opt                              |
| 1047  | real(dm) mult, beta  | 1047                                    | real(dm) mult, beta                                    |
| 1048  | integer n, m   | 1048                                    | integer n, m   |
| 1049  | _  | 1049                                    | _  |
| 1050  | call Matrix_Start('Mult_S  | 1050                                    | call Matrix_Start('Mult_S                              |

```
/Users/lplopa/Compare/camb des/Matrix u
/Users/lplopa/Compare/camb simdata/Matr
                                          tils.F90, Top line: 1051
ix utils.F90, Top line: 10\overline{51}
                                          1051
1051
1052
             m = Size(Mat,DIM=1)
                                          1052
                                                       m = Size(Mat,DIM=1)
1053
             n = Size(U,DIM=2)
                                                       n = Size(U,DIM=2)
                                          1053
1054
                                                       if (m /= Size(U,DIM=1) .o
             if (m /= Size(U,DIM=1) .o
                                          1054
1055
             call MpiStop('Matrix Mult
                                          1055
                                                           call MpiStop('Matrix |
                                          1056
1056
             if (present(a)) then
                                                       if (present(a)) then
1057
                 mult = a
                                          1057
                                                           mult = a
1058
                                          1058
                                                       else
             else
1059
                                          1059
                 mult = ROne
                                                           mult = ROne
1060
             end if
                                          1060
                                                       end if
1061
                                          1061
             if (present(b)) then
                                                       if (present(b)) then
1062
                                          1062
                 beta = b
                                                           beta = b
1063
                                          1063
             else
                                                       else
                                          1064
1064
                 beta = RZero
                                                           beta = RZero
             end if
1065
                                          1065
                                                       end if
1066
             if (matrix method == Mat
                                          1066
                                                       if (matrix method == Mat
1067
                 if (beta /= RZero) th
                                          1067
                                                            if (beta /= RZero) th
1068
                      out = a*MatMul(Ma
                                          1068
                                                                out = a*MatMul(Ma
1069
                                          1069
                 else
                                                           else
1070
                                          1070
                      out = MatMul(Mat,
                                                                out = MatMul(Mat,
1071
                      if (mult /= COne)
                                          1071
                                                                if (mult /= COne)
                                                           end if
1072
                 end if
                                          1072
                                          1073
1073
             else
                                                       else
1074
                                          1074
                                                   #ifdef MATRIX SINGLE
         #ifdef MATRIX SINGLE
1075
                 call SSYMM('L','U',m,
                                          1075
                                                           call SSYMM('L','U',m,
1076
                                          1076
         #else
                                                   #else
1077
                                          1077
                 call DSYMM('L','U',m,
                                                           call DSYMM('L','U',m,
1078
         #endif
                                          1078
                                                   #endif
                                                       end if
1079
             end if
                                          1079
1080
                                          1080
```

|      |                                 | /Users/lplopa/Compare/camb_des/Matrix_u |                                 |
|------|---------------------------------|---|---------------------------------|
|      | ls.F90, Top line: 1081          |   | 90, Top line: 1081              |
| 1081 | call Matrix_End('Mult_Sym       | 1081                                    | call Matrix_End('Mult_Sym       |
| 1082 |                                 | 1082                                    |                                 |
| 1083 | end subroutine Matrix_Mul       | 1083                                    | end subroutine Matrix_Mul       |
| 1084 |                                 | 1084                                    |                                 |
| 1085 |                                 | 1085                                    |                                 |
| 1086 | subroutine Matrix_Mult_Sy       | 1086                                    | subroutine Matrix_Mult_Sy       |
| 1087 | ! Out = $a*Mat U + b*out$       | 1087                                    | ! Out = $a*Mat U + b*out$       |
| 1088 | real(dm), intent(in) :: M       | 1088                                    | real(dm), intent(in) :: M       |
| 1089 | real(dm) Out(:,:)               | 1089                                    | real(dm) Out(:,:)               |
| 1090 | real(dm), intent(in), opt       | 1090                                    | real(dm), intent(in), opt       |
| 1091 | real(dm) mult, beta             | 1091                                    | real(dm) mult, beta             |
| 1092 | integer n,m                     | 1092                                    | integer n,m                     |
| 1093 |                                 | 1093                                    |                                 |
| 1094 | call Matrix Start('Mult S       | 1094                                    | call Matrix Start('Mult_S       |
| 1095 | _ ` _                           | 1095                                    |                                 |
| 1096 | m = Size(Mat,DIM=1)             | 1096                                    | m = Size(Mat,DIM=1)             |
| 1097 | n = Size(U,DIM=2)               | 1097                                    | n = Size(U,DIM=2)               |
| 1098 | if (n /= Size(Mat,DIM=2)        | 1098                                    | if (n /= Size(Mat,DIM=2)        |
| 1099 | call MpiStop( Matrix Mult       | 1099                                    | call MpiStop('Matrix_           |
| 1100 | if (present(a)) then            | 1100                                    | if (present(a)) then            |
| 1101 | mult = a                        | 1101                                    | mult = a                        |
| 1102 | else                            | 1102                                    | else                            |
| 1103 | mult = ROne                     | 1103                                    | mult = ROne                     |
| 1104 | end if                          | 1104                                    | end if                          |
| 1105 | <pre>if (present(b)) then</pre> | 1105                                    | <pre>if (present(b)) then</pre> |
| 1106 | beta = b                        | 1106                                    | beta = b '                      |
| 1107 | else                            | 1107                                    | else                            |
| 1108 | beta = RZero                    | 1108                                    | beta = RZero                    |
| 1109 | end if                          | 1109                                    | end if                          |
| 1110 | if (matrix method == Mat        |   | if (matrix method == Mat        |
|      |                                 |   |                                 |

| /Users | /Users/lplopa/Compare/camb_simdata/Matr /Users       |        | <pre>sers/lplopa/Compare/camb_des/Matrix_u</pre>     |  |
|--------|--|--------|--|--|
| ix_uti | ls.F90, Top line: $11\overline{1}1$                  | tils.F | 90, Top line: 1111                                   |  |
| 1111   | if (beta /= RZero) th                                | 1111   | if (beta /= RZero) th                                |  |
| 1112   | out = a*MatMul(Ma                                    | 1112   | out = a*MatMul(Ma                                    |  |
| 1113   | else   | 1113   | else   |  |
| 1114   | out = MatMul(Mat,                                    | 1114   | <pre>out = MatMul(Mat,</pre>                         |  |
| 1115   | if (mult /= ROne)                                    | 1115   | if (mult /= ROne)                                    |  |
| 1116   | end if   | 1116   | end if `   |  |
| 1117   | else   | 1117   | else   |  |
| 1118   | #ifdef MATRIX SINGLE                                 | 1118   | #ifdef MATRIX SINGLE                                 |  |
| 1119   | $call \overline{S}SYMM('R','U',m,$                   | 1119   | $call \overline{S}SYMM('R','U',m,$                   |  |
| 1120   | #else  | 1120   | #else  |  |
| 1121   | call DSYMM('R','U',m,                                | 1121   | call DSYMM('R','U',m,                                |  |
| 1122   | #endif   | 1122   | #endif   |  |
| 1123   | end if   | 1123   | end if   |  |
| 1124   |  | 1124   |  |  |
| 1125   | call Matrix End('Mult Sym                            | 1125   | call Matrix End('Mult Sym                            |  |
| 1126   | _ `  | 1126   | _ `  |  |
| 1127   | end subroutine Matrix Mul                            | 1127   | end subroutine Matrix Mul                            |  |
| 1128   | _  | 1128   | _  |  |
| 1129   |  | 1129   |  |  |
| 1130   | subroutine Matrix CMultGe                            | 1130   | subroutine Matrix CMultGe                            |  |
| 1131   | ! out(1:m,1:n) = MatM                                | 1131   | ! out(1:m,1:n) = MatM                                |  |
| 1132   | <pre>integer, intent(in) :: m,</pre>                 | 1132   | <pre>integer, intent(in) :: m,</pre>                 |  |
| 1133   | <pre>complex(dm), intent(in) :</pre>                 | 1133   | <pre>complex(dm), intent(in) :</pre>                 |  |
| 1134   | complex(dm) Out(:,:)                                 | 1134   | complex(dm) Out(:,:)                                 |  |
| 1135   |  | 1135   |  |  |
| 1136   | call Matrix Start('CMultG                            | 1136   | call Matrix Start('CMultG                            |  |
| 1137   | _ `  | 1137   | _ `  |  |
| 1138   | <pre>if (SIZE(Out,DIM=1) <m .o<="" pre=""></m></pre> | 1138   | <pre>if (SIZE(Out,DIM=1) <m .o<="" pre=""></m></pre> |  |
| 1139   | , , , , , , , , , , , , , , , , , , ,                | 1139   |  |  |
| 1140   | <pre>if (matrix_method == Mat_</pre>                 | 1140   | <pre>if (matrix_method == Mat_</pre>                 |  |

| /Users/lplopa/Compare/camb_simdata/Matr |  | /Users/lp1opa/Compare/camb_des/Matrix_u |  |  |  |
|---|--|---|--|--|--|
| ix_utils.F90, Top line: 1141            |  |   | tils.F90, Top line: 1141                             |  |  |
| 1141                                    | out(1:m,1:n) = MatMul                                | 1141                                    | <pre>out(1:m,1:n) = MatMul</pre>                     |  |  |
| 1142                                    | else   | 1142                                    | else   |  |  |
| 1143                                    | #ifdef MATRIX SINGLE                                 | 1143                                    | #ifdef MATRIX SINGLE                                 |  |  |
| 1144                                    | $call \overline{CGEMM}('N','N',m,$                   | 1144                                    | $call \overline{CGEMM}('N','N',m,$                   |  |  |
| 1145                                    | #else  | 1145                                    | #else  |  |  |
| 1146                                    | call ZGEMM('N','N',m,                                | 1146                                    | call ZGEMM('N','N',m,                                |  |  |
| 1147                                    | #endif   | 1147                                    | #endif   |  |  |
| 1148                                    | end if   | 1148                                    | end if   |  |  |
| 1149                                    | call Matrix End('CMultGen                            | 1149                                    | call Matrix End('CMultGen                            |  |  |
| 1150                                    | _ `  | 1150                                    | _ `  |  |  |
| 1151                                    |  | 1151                                    |  |  |  |
| 1152                                    | end subroutine Matrix CMu                            | 1152                                    | end subroutine Matrix CMu                            |  |  |
| 1153                                    | _  | 1153                                    | _  |  |  |
| 1154                                    |  | 1154                                    |  |  |  |
| 1155                                    |  | 1155                                    |  |  |  |
| 1156                                    | subroutine Matrix MultGen                            | 1156                                    | subroutine Matrix MultGen                            |  |  |
| 1157                                    | ! out(1:m,1:n) = MatM                                | 1157                                    | ! out(1:m,1:n) = MatM                                |  |  |
| 1158                                    | <pre>integer, intent(in) :: m,</pre>                 | 1158                                    | <pre>integer, intent(in) :: m,</pre>                 |  |  |
| 1159                                    | real(dm), intent(in) :: M                            | 1159                                    | real(dm), intent(in) :: M                            |  |  |
| 1160                                    | real(dm) Out(:,:)                                    | 1160                                    | real(dm) Out(:,:)                                    |  |  |
| 1161                                    |  | 1161                                    |  |  |  |
| 1162                                    | call Matrix Start('MultGe                            | 1162                                    | call Matrix Start('MultGe                            |  |  |
| 1163                                    | _ `  | 1163                                    | _ `  |  |  |
| 1164                                    | <pre>if (SIZE(Out,DIM=1) <m .o<="" pre=""></m></pre> | 1164                                    | <pre>if (SIZE(Out,DIM=1) <m .o<="" pre=""></m></pre> |  |  |
| 1165                                    |  | 1165                                    |  |  |  |
| 1166                                    | <pre>if (matrix method == Mat</pre>                  | 1166                                    | <pre>if (matrix method == Mat_</pre>                 |  |  |
| 1167                                    | $out(1:\overline{m},1:n) = MatMu\overline{l}$        | 1167                                    | $out(1:\overline{m},1:n) = MatMu\overline{l}$        |  |  |
| 1168                                    | else   | 1168                                    | else   |  |  |
| 1169                                    | <b>#ifdef MATRIX_SINGLE</b>                          | 1169                                    | #ifdef MATRIX_SINGLE                                 |  |  |
| 1170                                    | call $\overline{S}GEMM('N','N',m,$                   | 1170                                    | call $\overline{S}GEMM('N','N',m,$                   |  |  |

| /Users/lplopa/Compare/camb_simdata/Matr ix utils.F90, Top line: 1171 |                                      | /Users/lplopa/Compare/camb_des/Matrix_u tils.F90, Top line: 1171 |                                      |
|--|--------------------------------------|--|--------------------------------------|
|  |                                      |  |                                      |
| 1171   | #else                                | 1171   | #else                                |
| 1172   | call DGEMM('N','N',m,                | 1172   | call DGEMM('N','N',m,                |
| 1173   | #endif                               | 1173   | #endif                               |
| 1174   | end if                               | 1174   | end if                               |
| 1175   | call Matrix_End('MultGen'            | 1175   | call Matrix_End('MultGen'            |
| 1176   |                                      | 1176   |                                      |
| 1177   |                                      | 1177   |                                      |
| 1178   | <pre>end subroutine Matrix_Mul</pre> | 1178   | end subroutine Matrix_Mul            |
| 1179   |                                      | 1179   |                                      |
| 1180   |                                      | 1180   |                                      |
| 1181   | subroutine Matrix_CMult_N            | 1181   | subroutine Matrix_CMult_N            |
| 1182   | ! Out = $a*Mat U^dag + b*o$          | 1182   | ! Out = $a*Mat U^dag + b*o$          |
| 1183   | <pre>complex(dm), intent(in) :</pre> | 1183   | <pre>complex(dm), intent(in) :</pre> |
| 1184   | <pre>complex(dm) Out(:,:)</pre>      | 1184   | <pre>complex(dm) Out(:,:)</pre>      |
| 1185   | <pre>complex(dm), intent(in),</pre>  | 1185   | <pre>complex(dm), intent(in),</pre>  |
| 1186   | <pre>complex(dm) mult, beta</pre>    | 1186   | <pre>complex(dm) mult, beta</pre>    |
| 1187   | integer m,n,k                        | 1187   | integer m,n,k                        |
| 1188   |                                      | 1188   |                                      |
| 1189   | m = Size(Mat,DIM=1)                  | 1189   | m = Size(Mat,DIM=1)                  |
| 1190   | n = Size(U,DIM=1)                    | 1190   | n = Size(U,DIM=1)                    |
| 1191   | k = Size(Mat,DIM=2)                  | 1191   | k = Size(Mat,DIM=2)                  |
| 1192   | if $(k /= Size(U,DIM=2))$ c          | 1192   | if $(k /= Size(U,DIM=2))$ c          |
| 1193   | call Matrix start('CMúlt             | 1193   | call Matrix start('CMult_            |
| 1194   | if $(present(a))$ then               | 1194   | if $(present(a))$ then               |
| 1195   | mult = a                             | 1195   | mult = a                             |
| 1196   | else                                 | 1196   | else                                 |
| 1197   | mult = COne                          | 1197   | mult = COne                          |
| 1198   | end if                               | 1198   | end if                               |
| 1199   | if (present(b)) then                 | 1199   | if (present(b)) then                 |
| 1200   | beta = b                             | 1200   | beta = b                             |

| <pre>/Users/lp1opa/Compare/camb_simdata/Matr / ix_utils.F90, Top line: 1201 t</pre> |                                     | /Users/lplopa/Compare/camb_des/Matrix_u tils.F90, Top line: 1201 |  |
|---|-------------------------------------|--|--|
| 1201  | else                                | 1201   | else   |
| 1202  | beta = CZero                        | 1202   | beta = CZero                                     |
| 1203  | end if                              | 1203   | end if   |
| 1204  |                                     | 1204   |  |
| 1205  | <pre>if (matrix method == Mat</pre> | 1205   | if (matrix method == Mat                         |
| 1206  | if (beta /= CZero) $t\overline{h}$  | 1206   | if (be $\overline{t}a$ /= CZero) $\overline{th}$ |
| 1207  | Out = beta*Out +                    | 1207   | Out = beta*Out +                                 |
| 1208  | else                                | 1208   | else   |
| 1209  | Out = matmul(Mat,                   | 1209   | Out = matmul(Mat,                                |
| 1210  | if (mult/= Cone)                    | 1210   | if (mult/= CÒne)                                 |
| 1211  | end if '                            | 1211   | end if `   |
| 1212  | else                                | 1212   | else   |
| 1213  | #ifdef MATRIX SINGLE                | 1213   | #ifdef MATRIX SINGLE                             |
| 1214  | call $\overline{CGEMM}('N','C',m,$  | 1214   | call $\overline{C}GEMM('N','C',m,$               |
| 1215  | #else                               | 1215   | #else  |
| 1216  | call ZGEMM('N','C',m,               | 1216   | call ZGEMM('N','C',m,                            |
| 1217  | #endif                              | 1217   | #endif   |
| 1218  | end if                              | 1218   | end if   |
| 1219  | call Matrix End('CMult NT           | 1219   | call Matrix End('CMult NT                        |
| 1220  | _ ` _                               | 1220   | = \ =  |
| 1221  |                                     | 1221   |  |
| 1222  | end subroutine Matrix CMu           | 1222   | end subroutine Matrix CMu                        |
| 1223  | <del>-</del>                        | 1223   | <b>–</b>   |
| 1224  | subroutine Matrix Mult NT           | 1224   | subroutine Matrix Mult NT                        |
| 1225  | ! Out = $a*Mat U^T + b*Out$         | 1225   | ! Out = a*Mat $U^T + b*Out$                      |
| 1226  | real(dm), intent(in) :: M           | 1226   | real(dm), intent(in) :: M                        |
| 1227  | real(dm) Out(:,:)                   | 1227   | real(dm) Out(:,:)                                |
| 1228  | real(dm), intent(in), opt           | 00000  | real(dm), intent(in), opt                        |
| 1229  | real(dm) mult, beta                 | 1229   |  |
| 1230  | integer'm,n,k                       | 1230   | integer'm,n,k                                    |

```
/Users/lplopa/Compare/camb des/Matrix u
/Users/lplopa/Compare/camb simdata/Matr
                                          tils.F90, Top line: 1231
ix utils.F90, Top line: 12\overline{31}
1231
                                          1231
1232
             m = Size(Mat,DIM=1)
                                          1232
                                                       m = Size(Mat,DIM=1)
1233
             n = Size(U,DIM=1)
                                                       n = Size(U,DIM=1)
                                          1233
1234
                                          1234
                                                       k = Size(Mat,DIM=2)
             k = Size(Mat, DIM=2)
                                          1235
1235
             if (k /= Size(U,DIM=2)) c
                                                       if (k /= Size(U,DIM=2)) c
                                                       call Matrix start('Mult_N
1236
                                          1236
             call Matrix start('Mult N
1237
                                          1237
             if (present(a)) then
                                                       if (present(a)) then
1238
                 mult = a
                                          1238
                                                           mult = a
1239
                                          1239
                                                       else
             else
1240
                                          1240
                 mult = ROne
                                                           mult = ROne
1241
             end if
                                          1241
                                                       end if
1242
                                          1242
             if (present(b)) then
                                                       if (present(b)) then
1243
                                          1243
                 beta = b
                                                           beta = b
1244
                                          1244
             else
                                                       else
1245
                                          1245
                 beta = RZero
                                                           beta = RZero
1246
                                                       end if
             end if
                                          1246
1247
                                          1247
1248
             if (matrix method == Mat
                                          1248
                                                       if (matrix method == Mat
                 if (beta /= RZero) th
                                          1249
                                                            if (beta /= RZero) th
1249
                                                                Out = beta*Out +
1250
                      Out = beta*Out +
                                          1250
1251
                                          1251
                 else
                                                           else
1252
                                          1252
                      Out = matmul(Mat,
                                                                Out = matmul(Mat,
                      if (mult/= ROne)
                                                                if (mult/= ROne)
1253
                                          1253
1254
                 end if
                                          1254
                                                           end if
1255
             else
                                          1255
                                                       else
1256
                                                   #ifdef MATRIX SINGLE
         #ifdef MATRIX SINGLE
                                          1256
1257
                 call SGEMM('N','T',m,
                                          1257
                                                           call SGEMM('N', 'T', m,
1258
         #else
                                          1258
                                                   #else
1259
                                          1259
                 call DGEMM('N','T',m,
                                                           call DGEMM('N', 'T', m,
1260
         #endif
                                          1260
                                                   #endif
```

| /Users/lp1opa/Compare/camb_simdata/Matr |                                      |        |                                      |  |
|---|--------------------------------------|--------|--------------------------------------|--|
| ix_uti                                  | ls.F90, Top line: 1261               | tils.F | 90, Top line: 1261                   |  |
| 1261                                    | end if                               | 1261   | end if                               |  |
| 1262                                    | call Matrix End('Mult NT'            | 1262   | call Matrix End('Mult NT'            |  |
| 1263                                    | _ ` _                                | 1263   |                                      |  |
| 1264                                    |                                      | 1264   |                                      |  |
| 1265                                    | end subroutine Matrix Mul            | 1265   | end subroutine Matrix Mul            |  |
| 1266                                    | _                                    | 1266   |                                      |  |
| 1267                                    |                                      | 1267   |                                      |  |
| 1268                                    | subroutine Matrix CMult T            | 1268   | subroutine Matrix CMult T            |  |
| 1269                                    | ! Out = $a*Mat^dag^U + b*O$          | 1269   | ! Out = $a*Mat^dag^U + b*O$          |  |
| 1270                                    | <pre>complex(dm), intent(in) :</pre> | 1270   | <pre>complex(dm), intent(in) :</pre> |  |
| 1271                                    | <pre>complex(dm) Out(:,:)</pre>      | 1271   | complex(dm) Out(:,:)                 |  |
| 1272                                    | <pre>complex(dm), intent(in),</pre>  | 1272   | <pre>complex(dm), intent(in),</pre>  |  |
| 1273                                    | <pre>complex(dm) mult, beta</pre>    | 1273   | complex(dm) mult, beta               |  |
| 1274                                    | integer m,n,k                        | 1274   | integer m,n,k                        |  |
| 1275                                    |                                      | 1275   |                                      |  |
| 1276                                    | m = Size(Mat,DIM=2)                  | 1276   | m = Size(Mat,DIM=2)                  |  |
| 1277                                    | n = Size(U,DIM=2)                    | 1277   | n = Size(U,DIM=2)                    |  |
| 1278                                    | k = Size(Mat,DIM=1)                  | 1278   | k = Size(Mat,DIM=1)                  |  |
| 1279                                    | if $(k /= Size(U,DIM=1)) c$          | 1279   | if $(k /= Size(U,DIM=1))$ c          |  |
| 1280                                    |                                      | 1280   |                                      |  |
| 1281                                    | call Matrix Start('CMult             | 1281   | call Matrix Start('CMult_            |  |
| 1282                                    | <pre>if (present(a)) then</pre>      | 1282   | if (present(a)) then                 |  |
| 1283                                    | mult = a                             | 1283   | mult = a                             |  |
| 1284                                    | else                                 | 1284   | else                                 |  |
| 1285                                    | mult = COne                          | 1285   | mult = COne                          |  |
| 1286                                    | end if                               | 1286   | end if                               |  |
| 1287                                    | <pre>if (present(b)) then</pre>      | 1287   | <pre>if (present(b)) then</pre>      |  |
| 1288                                    | beta = b                             | 1288   | beta = b                             |  |
| 1289                                    | else                                 | 1289   | else                                 |  |
| 1290                                    | beta = CZero                         | 1290   | beta = CZero                         |  |

```
/Users/lplopa/Compare/camb des/Matrix u
/Users/lplopa/Compare/camb simdata/Matr
                                         tils.F90, Top line: 1291
ix utils.F90, Top line: 12\overline{9}1
1291
             end if
                                         1291
                                                       end if
                                         1292
1292
             if (matrix method == Mat
                                                       if (matrix method == Mat
1293
                 if (beta /= CZero) th
                                         1293
                                                           if (beta /= CZero) th
1294
                     out = mult*MatMul
                                         1294
                                                               out = mult*MatMul
                                         1295
1295
                 else
                                                           else
                                         1296
1296
                     out = MatMul(conj
                                                               out = MatMul(conj
                     if (mult /= COne)
                                                               if (mult /= COne)
1297
                                         1297
1298
                 end if
                                                           end if
                                         1298
1299
             else
                                         1299
                                                      else
1300
        #ifdef MATRIX SINGLE
                                         1300
                                                  #ifdef MATRIX SINGLE
1301
                 call CGEMM('C','N',m,
                                         1301
                                                           call CGEMM('C','N',m,
1302
                                         1302
        #else
                                                  #else
1303
                                         1303
                 call ZGEMM('C','N',m,
                                                           call ZGEMM('C','N',m,
1304
        #endif
                                         1304
                                                  #endif
1305
                                         1305
             end if
                                                      end if
1306
                                         1306
             call Matrix End('CMult TN
                                                       call Matrix End('CMult TN
1307
                                         1307
             end subroutine Matrix CMu
                                         1308
1308
                                                       end subroutine Matrix CMu
1309
                                         1309
1310
             subroutine Matrix Mult TN
                                         1310
                                                       subroutine Matrix Mult TN
1311
                                         1311
             ! Out = a*Mat^daq U + b*O
                                                       ! Out = a*Mat^daq U + b*O
1312
             real(dm), intent(in) :: M
                                         1312
                                                       real(dm), intent(in) :: M
1313
                                         1313
             real(dm) Out(:,:)
                                                       real(dm) Out(:,:)
1314
                                         1314
                                                       real(dm), intent(in), opt
             real(dm), intent(in), opt
             real(dm) mult, beta
1315
                                         1315
                                                       real(dm) mult, beta
                                         1316
1316
             integer m,n,k
                                                       integer m,n,k
1317
                                         1317
1318
                                         1318
            m = Size(Mat, DIM=2)
                                                      m = Size(Mat, DIM=2)
1319
             n = Size(U,DIM=2)
                                         1319
                                                      n = Size(U,DIM=2)
1320
             k = Size(Mat, DIM=1)
                                         1320
                                                       k = Size(Mat, DIM=1)
```

```
/Users/lplopa/Compare/camb des/Matrix u
/Users/lplopa/Compare/camb simdata/Matr
ix utils.F90, Top line: 13\overline{21}
                                          tils.F90, Top line: 1321
             if (k /= Size(U,DIM=1)) c 1321
                                                       if (k /= Size(U,DIM=1)) c
1321
1322
                                          1322
1323
                                         1323
             call Matrix Start('CMult
                                                       call Matrix Start('CMult
1324
             if (present(a)) then
                                          1324
                                                       if (present(a)) then
                 mult = a
1325
                                          1325
                                                           mult = a
1326
             else
                                          1326
                                                       else
1327
                                          1327
                 mult = ROne
                                                           mult = ROne
1328
             end if
                                          1328
                                                       end if
1329
                                          1329
                                                       if (present(b)) then
             if (present(b)) then
1330
                                          1330
                 beta = b
                                                           beta = b
1331
                                          1331
             else
                                                       else
1332
                                          1332
                 beta = RZero
                                                           beta = RZero
1333
             end if
                                          1333
                                                       end if
1334
                                         1334
             if (matrix method == Mat
                                                       if (matrix method == Mat
1335
                 if (beta /= CZero) th
                                         1335
                                                           if (beta /= CZero) th
1336
                      out = mult*MatMul
                                          1336
                                                                out = mult*MatMul
1337
                 else
                                          1337
                                                           else
1338
                                         1338
                      out = MatMul(tran
                                                                out = MatMul(tran
1339
                      if (mult /= COne)
                                          1339
                                                                if (mult /= COne)
1340
                 end if
                                          1340
                                                           end if
1341
                                          1341
             else
                                                       else
                                                   #ifdef MATRIX SINGLE
1342
         #ifdef MATRIX SINGLE
                                          1342
1343
                                          1343
                                                           call SGEMM('T','N',m,
                 call SGEMM('T','N',m,
1344
        #else
                                          1344
                                                   #else
1345
                 call DGEMM('T','N',m,
                                          1345
                                                           call DGEMM('T','N',m,
1346
        #endif
                                          1346
                                                   #endif
                                                       end if
1347
             end if
                                          1347
                                                       call Matrix_End('Mult_TN'
1348
             call Matrix End('Mult TN'
                                          1348
1349
                                          1349
             end subroutine Matrix_Mul
                                                       end subroutine Matrix Mul
1350
                                          1350
```

```
/Users/lplopa/Compare/camb des/Matrix u
/Users/lplopa/Compare/camb simdata/Matr
                                         tils.F90, Top line: 1351
ix utils.F90, Top line: 13\overline{5}1
                                         1351
1351
                                         1352
1352
             subroutine Matrix Cholesk
                                                       subroutine Matrix Cholesk
1353
             !Note upper triangular is
                                         1353
                                                       !Note upper triangular is
1354
             real(dm), intent(inout)::
                                         1354
                                                       real(dm), intent(inout)::
             integer n, info
1355
                                         1355
                                                       integer n, info
                                                       integer, optional :: err
1356
                                         1356
             integer, optional :: err
1357
             logical, intent(in), opti
                                         1357
                                                       logical, intent(in), opti
1358
                                         1358
             integer i
                                                       integer i
1359
                                         1359
1360
                                         1360
             n=Size(M,DIM=1)
                                                      n=Size(M,DIM=1)
1361
             if (Size(M,DIM=2)/=n) cal
                                         1361
                                                       if (Size(M,DIM=2)/=n) cal
1362
                                         1362
1363
                                         1363
        #ifdef MATRIX SINGLE
                                                  #ifdef MATRIX SINGLE
1364
             call spotrf ('L', n, M, n 1364
                                                      call spotrf ('L', n, M, n
1365
        #else
                                         1365
                                                  #else
1366
             call dpotrf ('L', n, M, n 1366
                                                      call dpotrf ('L', n, M, n
1367
        #endif
                                         1367
                                                  #endif
                                         1368
1368
1369
                                         1369
             if (present(err)) then
                                                       if (present(err)) then
1370
                 err = info
                                         1370
                                                           err = info
1371
                                         1371
             else
                                                       else
1372
                 if (info/=0) &
                                         1372
                                                           if (info/=0) &
1373
                                         1373
                                                               call MpiStop('Mat
                 call MpiStop('Matrix
1374
             end if
                                         1374
                                                       end if
1375
                                         1375
1376
                                         1376
             if (info==0 .and. present
                                                       if (info==0 .and. present
1377
                 do i=1,n
                                         1377
                                                           do i=1,n
1378
                     M(1:i-1,i)=0
                                         1378
                                                               M(1:i-1,i)=0
1379
                 end do
                                         1379
                                                           end do
1380
             end if
                                         1380
                                                       end if
```

| /Users/lp1opa/Compare/camb_simdata/Matr /Uix utils.F90, Top line: 1381 ti |  |      | <pre>/lplopa/Compare/camb_des/Matrix_u 90, Top line: 1381</pre> |
|---|--|------|---|
| 1381  |  | 1381 |   |
| 1382  | end subroutine Matrix Cho                | 1382 | end subroutine Matrix Cho                                       |
| 1383  | cha Babioacine Maciin_ono                | 1383 | end subfoutine Mattin_one                                       |
| 1384  | subroutine Matrix CCholes                | 1384 | subroutine Matrix CCholes                                       |
| 1385  | $!M = L L^{\dagger}$                     | 1385 | $!M = L L^{\dagger} dag$  |
| 1386  | complex(dm), intent(inout                | 1386 | complex(dm), intent(inout                                       |
| 1387  | integer n, info                          | 1387 | integer n, info   |
| 1388  |  | 1388 |   |
| 1389  | n=Size(M,DIM=1)                          | 1389 | n=Size(M,DIM=1)   |
| 1390  | if (Size(M,DIM=2)/=n) cal                | 1390 | if (Size(M,DIM=2)/=n) cal                                       |
| 1391  |  | 1391 |   |
| 1392  | #ifdef MATRIX SINGLE                     | 1392 | #ifdef MATRIX SINGLE  |
| 1393  | call cpot $\overline{r}$ f ('L', n, M, n | 1393 | call cpot $\overline{r}$ f ('L', n, M, n                        |
| 1394  | #else                                    | 1394 | #else   |
| 1395  | call zpotrf ('L', n, M, n                | 1395 | call zpotrf ('L', n, M, n                                       |
| 1396  | #endif                                   | 1396 | #endif  |
| 1397  |  | 1397 |   |
| 1398  | if (info/=0) call MpiStop                | 1398 | if (info/=0) call MpiStop                                       |
| 1399  | ,  | 1399 | ,                         |
| 1400  | end subroutine Matrix CCh                | 1400 | end subroutine Matrix CCh                                       |
| 1401  | _  | 1401 | _   |
| 1402  |  | 1402 |   |
| 1403  | subroutine Matrix Cholesk                | 1403 | subroutine Matrix Cholesk                                       |
| 1404  | $!M = L L^T and return L^{$              | 1404 | $!M = L L^T and return L^{}$                                    |
| 1405  | real(dm), intent(inout)::                | 1405 | real(dm), intent(inout)::                                       |
| 1406  | integer n, info                          | 1406 | integer n, info   |
| 1407  | integer i,j                              | 1407 | integer i,j   |
| 1408  | logical, intent(in), opti                | 1408 | logical, intent(in), opti                                       |
| 1409  | integer, intent(out), opt                | 1409 | integer, intent(out), opt                                       |
| 1410  | logical trans                            | 1410 | logical trans   |

```
/Users/lplopa/Compare/camb simdata/Matr /Users/lplopa/Compare/camb des/Matrix u
                                         tils.F90, Top line: 1411
ix utils.F90, Top line: 1411
1411
                                         1411
1412
                                         1412
             call Matrix Cholesky(M, i
                                                      call Matrix Cholesky(M, i
                                                       if (info==0) then
             if (info==0) then
1413
                                         1413
1414
                 n=size(M,dim=1)
                                         1414
                                                           n=size(M,dim=1)
1415
        #ifdef MATRIX SINGLE
                                         1415
                                                  #ifdef MATRIX SINGLE
1416
                                         1416
                 call STRTRI( 'L', 'N'
                                                           call STRTRI( 'L', 'N'
1417
        #else
                                         1417
                                                  #else
                 call DTRTRI( 'L', 'N'
1418
                                         1418
                                                           call DTRTRI( 'L', 'N'
1419
        #endif
                                         1419
                                                  #endif
1420
             end if
                                         1420
                                                      end if
1421
                                         1421
                                                       if (present(error)) error
             if (present(error)) error
1422
             if (info/=0) then
                                         1422
                                                       if (info/=0) then
1423
                 if (present(error)) r 1423
                                                           if (present(error)) r
1424
                                         1424
                 call MpiStop('Matrix
                                                           call MpiStop('Matrix
             end if
1425
                                         1425
                                                       end if
1426
                                         1426
1427
                                       t 1427
             if (present(transpose))
                                                       if (present(transpose)) t
1428
                 trans = transpose
                                         1428
                                                           trans = transpose
1429
                                         1429
             else
                                                       else
1430
                 trans = .false.
                                         1430
                                                           trans = .false.
1431
                                         1431
                                                      end if
             end if
1432
                                         1432
1433
                                         1433
             if (trans) then
                                                       if (trans) then
1434
                                         1434
1435
             do i=1,n
                                         1435
                                                           do i=1,n
1436
                 do j=1, i-1
                                         1436
                                                               do j=1, i-1
1437
                     M(j,i) = M(i,j)
                                         1437
                                                                   M(j,i) = M(i,
1438
                     M(i,j) = 0
                                         1438
                                                                   M(i,j)
1439
                                         1439
                                                               end do
                 end do
1440
             end do
                                         1440
                                                           end do
```

| /Users/lp1opa/Compare/camb_simdata/Matr |                                      |        |                                      |  |
|---|--------------------------------------|--------|--------------------------------------|--|
| ix_util                                 | s.F90, Top line: 1441                | tils.F | 90, Top line: 1441                   |  |
| 1441                                    |                                      | 1441   |                                      |  |
| 1442                                    | else                                 | 1442   | else                                 |  |
| 1443                                    |                                      | 1443   |                                      |  |
| 1444                                    | do i=1,n                             | 1444   | do i=1,n                             |  |
| 1445                                    | do j=1,i-1                           | 1445   | do j=1,i-1                           |  |
| 1446                                    | M(j,i) = 0                           | 1446   | M(j,i) = 0                           |  |
| 1447                                    | end do                               | 1447   | end do                               |  |
| 1448                                    | end do                               | 1448   | end do                               |  |
| 1449                                    |                                      | 1449   |                                      |  |
| 1450                                    | end if                               | 1450   | end if                               |  |
| 1451                                    | end subroutine Matrix Cho            | 1451   | end subroutine Matrix Cho            |  |
| 1452                                    | _                                    | 1452   | <del>_</del>                         |  |
| 1453                                    | subroutine Matrix CCholes            | 1453   | subroutine Matrix CCholes            |  |
| 1454                                    | $!M = L L^{\alpha} and return$       | 1454   | $!M = L L^{\alpha} and return$       |  |
| 1455                                    | <pre>complex(dm), intent(inout</pre> | 1455   | <pre>complex(dm), intent(inout</pre> |  |
| 1456                                    | integer n, info                      | 1456   | integer n, info                      |  |
| 1457                                    | integer i,j                          | 1457   | integer i,j                          |  |
| 1458                                    | <pre>logical, intent(in), opti</pre> | 1458   | <pre>logical, intent(in), opti</pre> |  |
| 1459                                    | logical trans                        | 1459   | logical trans                        |  |
| 1460                                    |                                      | 1460   |                                      |  |
| 1461                                    | call Matrix_CCholesky(M)             | 1461   | <pre>call Matrix_CCholesky(M)</pre>  |  |
| 1462                                    | n=size(M,dim=1)                      | 1462   | n=size(M,dim=1)                      |  |
| 1463                                    |                                      | 1463   |                                      |  |
| 1464                                    | #ifdef MATRIX_SINGLE                 | 1464   | <b>#ifdef MATRIX_SINGLE</b>          |  |
| 1465                                    | call CTRTRI( 'L', 'N', n,            | 1465   | call CTRTRI( 'L', 'N', n,            |  |
| 1466                                    | #else                                | 1466   | #else                                |  |
| 1467                                    | call ZTRTRI( 'L', 'N', n,            | 1467   | call ZTRTRI( 'L', 'N', n,            |  |
| 1468                                    | #endif                               | 1468   | #endif                               |  |
| 1469                                    |                                      | 1469   |                                      |  |
| 1470                                    | if (info/=0) call MpiStop            | 1470   | if (info/=0) call MpiStop            |  |

| /Users/lplopa/Compare/camb_simdata/Matr / |                                      |              | <del>-</del>                         |
|---|--------------------------------------|--------------|--------------------------------------|
| ix_utils.F9                               | 0, Top line: 1471                    | tils.F90, To | p line: 1471                         |
| 1471                                      |                                      | 1471         |                                      |
| 1472                                      | <pre>if (present(dagger)) then</pre> | 1472         | <pre>if (present(dagger)) then</pre> |
| 1473                                      | trans = dagger                       | 1473         | trans = dagger                       |
| 1474                                      | else                                 | 1474         | else                                 |
| 1475                                      | <pre>trans = .false.</pre>           | 1475         | <pre>trans = .false.</pre>           |
| 1476                                      | end if                               | 1476         | end if                               |
| 1477                                      |                                      | 1477         |                                      |
| 1478                                      | if (trans) then                      | 1478         | if (trans) then                      |
| 1479                                      | ` '                                  | 1479         | ,                                    |
| 1480                                      | do i=1,n                             | 1480         | do i=1,n                             |
| 1481                                      | do j=1,i-1                           | 1481         | do j=1,i-1                           |
| 1482                                      | M(j,i) = conjg(M(                    | 1482         | M(j,i) = conj                        |
| 1483                                      | M(i,j) = 0                           | 1483         | M(i,j) = 0                           |
| 1484                                      | end dò                               | 1484         | end do                               |
| 1485                                      | end do                               | 1485         | end do                               |
| 1486                                      |                                      | 1486         |                                      |
| 1487                                      | else                                 | 1487         | else                                 |
| 1488                                      |                                      | 1488         |                                      |
| 1489                                      | do i=1,n                             | 1489         | do i=1,n                             |
| 1490                                      | do j=1,i-1                           | 1490         | do j=1,i-1                           |
| 1491                                      | M(j,i) = 0                           | 1491         | M(j,i) = 0                           |
| 1492                                      | end do                               | 1492         | end do                               |
| 1493                                      | end do                               | 1493         | end do                               |
| 1494                                      |                                      | 1494         |                                      |
| 1495                                      | end if                               | 1495         | end if                               |
| 1496                                      | end subroutine Matrix CCh            | 1496         | end subroutine Matrix CCh            |
| 1497                                      |                                      | 1497         | _                                    |
| 1498                                      |                                      | 1498         |                                      |
| 1499                                      | <pre>subroutine Matrix_inverse</pre> | 1499         | <pre>subroutine Matrix_inverse</pre> |
| 1500                                      | !Inverse of symmetric mat            | 1500         | !Inverse of symmetric mat            |

```
/Users/lplopa/Compare/camb des/Matrix u
/Users/lplopa/Compare/camb simdata/Matr
                                         tils.F90, Top line: 1501
ix utils.F90, Top line: 15\overline{01}
1501
                                         1501
             !This should not be used
                                                       !This should not be used
1502
             real(dm), intent(inout)::
                                         1502
                                                       real(dm), intent(inout)::
             integer i,j,n
                                         1503
1503
                                                       integer i,j,n
1504
                                         1504
             integer info
                                                       integer info
1505
             integer, optional :: err
                                         1505
                                                       integer, optional :: err
1506
                                         1506
1507
             n=Size(M,DIM=1)
                                         1507
                                                      n=Size(M,DIM=1)
                                                       if (Size(M,DIM=2)/=n) cal
1508
             if (Size(M,DIM=2)/=n) cal
                                         1508
1509
             call Matrix Start('Invers
                                         1509
                                                      call Matrix Start('Invers
1510
             if (present(err)) then
                                         1510
                                                       if (present(err)) then
1511
                 call Matrix Cholesky(
                                         1511
                                                           call Matrix Cholesky(
1512
                                         1512
                 if (err/=0) return
                                                           if (err/=0) return
1513
                                         1513
             else
                                                       else
1514
                                         1514
                 call Matrix Cholesky(
                                                           call Matrix Cholesky(
1515
             end if
                                         1515
                                                       end if
                                         1516
1516
        #ifdef MATRIX SINGLE
                                                  #ifdef MATRIX SINGLE
1517
             call spotri ('L', n, M, n 1517
                                                      call spotri ('L', n, M, n
1518
        #else
                                         1518
                                                  #else
1519
                                         1519
             call dpotri ('L', n, M, n
                                                      call dpotri ('L', n, M, n
1520
        #endif
                                         1520
                                                  #endif
1521
                                         1521
             if (present(err)) then
                                                       if (present(err)) then
1522
                 err = info
                                         1522
                                                           err = info
1523
                                         1523
                 if (err/=0) return
                                                           if (err/=0) return
1524
                                         1524
                                                       else
             else
                                         1525
                                                           if (info/=0) call Mpi
1525
                 if (info/=0) call Mpi
1526
                                         1526
             end if
                                                       end if
1527
             do i=1,n
                                         1527
                                                      do i=1,n
1528
                                         1528
                 do j=1, i-1
                                                           do j=1, i-1
1529
                                         1529
                     M(j,i) = M(i,j)
                                                               M(j,i) = M(i,j)
1530
                                         1530
                 end do
                                                           end do
```

|              | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 1531</pre> | /Users/lplopa/Compare/camb_des/Matrix_u tils.F90, Top line: 1531 |
|--------------|---|--|
| 1531 <b></b> | end do  | 1531 end do  |
| 1532         | call Matrix End('Inverse'   | 1532 call Matrix End('Inverse'                                   |
| 1533         |   | 1533   |
| 1534         | end subroutine Matrix i   | end subroutine Matrix i  |
| 1535         |   | 1535   |
| 1536         | subroutine Matrix Inverse   | 1536 subroutine Matrix Inverse                                   |
| 1537         | !Inverse of symmetric pos   | 1537 !Inverse of symmetric pos                                   |
| 1538         | real(dm), intent(inout)::   | real(dm), intent(inout)::  |
| 1539         | integer'i, n  | 1539 integer i, n  |
| 1540         |   | 1540   |
| 1541         | ! real(dm) w(Size(M,D   | 1541 ! real(dm) w(Size(M,D                                       |
| 1542         | ! real(dm), dimension   | 1542 ! real(dm), dimension                                       |
| 1543         | ! real(dm), dimension   | 1543 ! real(dm), dimension                                       |
| 1544         |   | 1544   |
| 1545         |   | 1545   |
| 1546         | n = size(M,DIM=1)   | 1546 	 n = size(M,DIM=1)   |
| 1547         | do $i=1$ , $size(M,DIM=1)$  | 1547 do i=1, size(M,DIM=1)                                       |
| 1548         | if $(abs(M(i,i)) < 1d-$   | 1548 if $(abs(M(i,i)) < 1d-$                                     |
| 1549         | end do  | 1549 end do  |
| 1550         |   | 1550   |
| 1551         | call Matrix Inverse Chol(   | 1551 call Matrix_Inverse_Chol(                                   |
| 1552         | <u> </u>  | 1552 !   |
| 1553         | ! allocate(tmp(Size(M   | 1553 ! allocate(tmp(Size(M                                       |
| 1554         | !   | 1554 !   |
| 1555         | ! n=Size(M,DIM=1)   | $1555 \qquad ! \qquad n=Size(M,DIM=1)$                           |
| 1556         | ! if (n<=1) return  | 1556 ! if (n<=1) return  |
| 1557         | ! if (Size(M,DIM=2)/=   | 1557 ! if (Size(M,DIM=2)/=                                       |
| 1558         | ! call Matrix_Start('   | 1558 ! call Matrix_Start('                                       |
| 1559         | !   | 1559   |
| 1560         | !   | 1560 !   |

|      |        | mpare/camb_simdata/Matr<br>p line: 1561 |          | .plopa/Co | <del>-</del>                     |
|------|--------|---|----------|-----------|----------------------------------|
| 1561 | 1      | allocate(norm(n))                       | 1561     | 1         | allocate(norm(n))                |
| 1562 | •<br>• | do i=1, n                               | 1562     | •         | do i=1, n                        |
| 1563 | ·<br>! | norm(i) = sqrt(a                        |          | •         | norm(i) = sqrt(a                 |
| 1564 | •<br>• | if (norm(i) < 1d                        |          | •<br>!    | if (norm(i) < 1d                 |
| 1565 | •<br>! | call MpiStop('M                         |          | •         | call MpiStop('M                  |
| 1566 | ·<br>! | M(i,:) = M(i,:) /                       |          | i         | M(i,:) = M(i,:) /                |
| 1567 | •<br>• | M(:,i) = M(:,i)/                        | 1567     | •         | M(:,i) = M(:,i) /                |
| 1568 | •<br>• | end do                                  | 1568     | •<br>!    | end do                           |
| 1569 | •<br>• | cha ao                                  | 1569     | •         | cha ao                           |
| 1570 | •<br>! | call Matrix Diagona                     | 100001   | •         | call Matrix Diagona              |
| 1571 | !      | write (*,*), 'min/m                     |          | !         | write (*,*), 'min/m              |
| 1572 | !      | if $(any(w \le 0))$ then                |          | !         | if $(any(w \le 0))$ then         |
| 1573 | !      | write (*,*), '                          |          | i         | write (*,*), '                   |
| 1574 | 1      | call MpiStop('                          |          | 1         | call MpiStop('                   |
| 1575 | 1      | end if                                  | 1575     | 1         | end if                           |
| 1576 | 1      | do i=1, n                               | 1576     | 1         | do i=1, n                        |
| 1577 | 1      | tmp(i,:) = M(:,i)                       |          | 1         | tmp(i,:) = M(:,i)                |
| 1578 | !      | end do                                  | 1578     | 1         | end do                           |
| 1579 | !      | allocate(tmp2(Size(                     | <u> </u> | !         | allocate(tmp2(Size(              |
| 1580 | !      | call Matrix Mult(M,                     | 700000   | !         | call Matrix Mult(M,              |
| 1581 | !      | M = tmp2                                | 1581     | !         | M = tmp2                         |
| 1582 | !      | do i=1, n                               | 1582     | !         | do $i=1$ , n                     |
| 1583 | !      | M(i,:) = M(i,:)/                        | 1583     | !         | M(i,:) = M(i,:)/                 |
| 1584 | !      | M(:,i) = M(:,i)/                        | 1584     | !         | M(:,i) = M(:,i)/                 |
| 1585 | !      | end do                                  | 1585     | !         | end do                           |
| 1586 | !      | <pre>deallocate(tmp, tmp</pre>          | 1586     | !         | deallocate(tmp, tmp              |
| 1587 | !      | deallocate(norm)                        | 1587     | !         | deallocate(norm)                 |
| 1588 | !      |   | 1588     | !         | call Matrix End('In              |
| 1589 |        | _ `                                     | 1589     |           | _ `                              |
| 1590 | end    | <pre>subroutine Matrix_Inv</pre>        | 1590     | end       | <pre>subroutine Matrix_Inv</pre> |

| <pre>/Users/lplopa/Compare/camb_simdata/Matr ix_utils.F90, Top line: 1591</pre> |   |      | <pre>/lplopa/Compare/camb_des/Matrix_u 90, Top line: 1591</pre> |
|---|---|------|---|
| 1591  |   | 1591 |   |
| 1592  | function Matrix GaussianL                 | 1592 | function Matrix GaussianL                                       |
| 1593  | !Returns -Log L $\overline{f i}$ kelihood | 1593 | !Returns -Log $L\overline{i}$ kelihood                          |
| 1594  | !** Cov is destroyed by t                 | 1594 | !** Cov is destroyed by t                                       |
| 1595  | real(dm), intent(inout)::                 | 1595 | real(dm), intent(inout)::                                       |
| 1596  | real(dm), intent(in):: d(                 | 1596 | real(dm), intent(in):: d(                                       |
| 1597  | real(dm), allocatable ::                  | 1597 | real(dm), allocatable ::  |
| 1598  | real(dm) :: LogLike                       | 1598 | real(dm) :: LogLike   |
| 1599  | integer info,i,n                          | 1599 | integer info,i,n  |
| 1600  | _ , ,                                     | 1600 |   |
| 1601  | call Matrix Start('Gaussi                 | 1601 | call Matrix Start('Gaussi                                       |
| 1602  | n = size(COV, DIM=1)                      | 1602 | $n = size(CO\overline{V}, DIM=1)$                               |
| 1603  | if (Size(COV,DIM=2)/=n) c                 | 1603 | if (Size(COV,DIM=2)/=n) c                                       |
| 1604  | if (Size(d)/=n) call MpiS                 | 1604 | if (Size(d)/=n) call MpiS                                       |
| 1605  |   | 1605 | •                         |
| 1606  | <pre>call Matrix Cholesky(Cov)</pre>      | 1606 | <pre>call Matrix Cholesky(Cov)</pre>                            |
| 1607  | LogLike = 0                               | 1607 | LogLike = 0   |
| 1608  | !Log Det term:                            | 1608 | !Log Det term:  |
| 1609  | do ī=1, n                                 | 1609 | do i=1, n   |
| 1610  | LogLike = LogLike +                       | 1610 | LogLike = LogLike +   |
| 1611  | end do                                    | 1611 | end do  |
| 1612  |   | 1612 |   |
| 1613  | !Solve for Cov^{-1}d [cou                 | 1613 | !Solve for Cov^{-1}d [cou                                       |
| 1614  | allocate(tmp(n))                          | 1614 | allocate(tmp(n))  |
| 1615  | tmp = d                                   | 1615 | tmp = d   |
| 1616  | #ifdef MATRIX SINGLE                      | 1616 | #ifdef MATRIX_SINGLE  |
| 1617  | call SPOTRS('L', N, 1, Co                 | 1617 | call SPOTRS('L', N, 1, Co                                       |
| 1618  | #else                                     | 1618 | #else   |
| 1619  |   | 1619 | call DPOTRS('L', N, 1, Co                                       |
| 1620  | #endif                                    | 1620 | #endif  |

| /Users | <pre>/lplopa/Compare/camb_simdata/Matr</pre> | /Users | /lplopa/Compare/camb_des/Matrix_u |
|--------|--|--------|-----------------------------------|
| ix_uti | ls.F90, Top line: $16\overline{2}1$          | tils.F | 90, Top line: 1621                |
| 1621   | if (INFO/=0) call MpiStop                    | 1621   | if (INFO/=0) call MpiStop         |
| 1622   | `  | 1622   | `                                 |
| 1623   | !Add together                                | 1623   | !Add together                     |
| 1624   | LogLike = LogLike + dot_p                    | 1624   | LogLike = LogLike + dot_p         |
| 1625   | deallocate(tmp)                              | 1625   | deallocate(tmp)                   |
| 1626   |  | 1626   |                                   |
| 1627   | call Matrix_End('Gaussian                    | 1627   | call Matrix_End('Gaussian         |
| 1628   |  | 1628   |                                   |
| 1629   | end function Matrix_Gauss                    | 1629   | end function Matrix_Gauss         |
| 1630   |  | 1630   |                                   |
| 1631   | function Matrix_GaussianL                    | 1631   | function Matrix_GaussianL         |
| 1632   | !Returns -Log Likelihood                     | 1632   | !Returns -Log Likelihood          |
| 1633   | !** Cov is destroyed by t                    | 1633   | !** Cov is destroyed by t         |
| 1634   | double precision, intent(                    | 1634   | double precision, intent(         |
| 1635   | double precision, intent(                    | 1635   | double precision, intent(         |
| 1636   | double precision, allocat                    | 1636   | double precision, allocat         |
| 1637   | double precision :: LogLi                    | 1637   | double precision :: LogLi         |
| 1638   | integer info,i,n                             | 1638   | integer info,i,n                  |
| 1639   |  | 1639   |                                   |
| 1640   | call Matrix_Start('Gaussi                    | 1640   | call Matrix_Start('Gaussi         |
| 1641   | n = size(COV, DIM=1)                         | 1641   | n = size(COV, DIM=1)              |
| 1642   | if (Size(COV,DIM=2)/=n) c                    | 1642   | if (Size(COV,DIM=2)/=n) c         |
| 1643   | if (Size(d)/=n) call MpiS                    | 1643   | if (Size(d)/=n) call MpiS         |
| 1644   |  | 1644   |                                   |
| 1645   | call dpotrf ('L', n, Cov,                    | 1645   | call dpotrf ('L', n, Cov,         |
| 1646   | if (info/=0) call MpiStop                    | 1646   | if (info/=0) call MpiStop         |
| 1647   |  | 1647   |                                   |
| 1648   | LogLike = 0                                  | 1648   | LogLike = 0                       |
| 1649   | !Log Det term:                               | 1649   | !Log Det term:                    |
| 1650   | do i=1, n                                    | 1650   | do i=1, n                         |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 1651</pre> |      | s/lplopa/Compare/camb_des/Matrix_u<br>'90, Top line: 1651 |
|------|---|------|---|
| 1651 | LogLike = LogLike +   | 1651 | LogLike = LogLike +                                       |
| 1652 | end do  | 1652 | end do  |
| 1653 |   | 1653 |   |
| 1654 | !Solve for Cov^{-1}d [cou   | 1654 | !Solve for Cov^{-1}d [cou                                 |
| 1655 | allocate(tmp(n))  | 1655 | allocate(tmp(n))  |
| 1656 | tmp = d   | 1656 | tmp = d   |
| 1657 | call DPOTRS('L', N, 1, Co   | 1657 | call DPOTRS('L', N, 1, Co                                 |
| 1658 | if (INFO/=0) call MpiStop   | 1658 | if (INFO/=0) call MpiStop                                 |
| 1659 | `   | 1659 | , , ,   |
| 1660 | !Add together   | 1660 | !Add together   |
| 1661 | LogLike = LogLike + dot_p   | 1661 | LogLike = LogLike + dot_p                                 |
| 1662 | deallocate(tmp)   | 1662 | deallocate(tmp)   |
| 1663 | , <u>-</u> ,  | 1663 | , -,  |
| 1664 | call Matrix_End('Gaussian   | 1664 | call Matrix_End('Gaussian                                 |
| 1665 | <del>-</del> ·  | 1665 | _ ,   |
| 1666 | <pre>end function Matrix_Gauss</pre>                                | 1666 | end function Matrix_Gauss                                 |
| 1667 | _   | 1667 |   |
| 1668 | subroutine Matrix_Inverse   | 1668 | subroutine Matrix_Inverse                                 |
| 1669 | !This should not be used  | 1669 | !This should not be used                                  |
| 1670 | <pre>real(dm), intent(inout)::</pre>                                | 1670 | <pre>real(dm), intent(inout)::</pre>                      |
| 1671 | real(dm) w(Size(M,DIM=1))   | 1671 | real(dm) w(Size(M,DIM=1))                                 |
| 1672 | <pre>real(dm), dimension(:,:),</pre>                                | 1672 | <pre>real(dm), dimension(:,:),</pre>                      |
| 1673 | integer i, n  | 1673 | integer i, n  |
| 1674 |   | 1674 |   |
| 1675 | n=Size(M,DIM=1)   | 1675 | n=Size(M,DIM=1)   |
| 1676 | if (n<=1) return  | 1676 | if (n<=1) return  |
| 1677 | if (Size(M,DIM=2)/=n) cal   | 1677 | if (Size(M,DIM=2)/=n) cal                                 |
| 1678 |   | 1678 |   |
| 1679 | allocate(tmp(n,n),VT(n,n)   | 1679 | allocate(tmp(n,n),VT(n,n)                                 |
| 1680 |   | 1680 |   |

|      | /lplopa/Compare/camb_simdata/Matr<br>ls.F90, Top line: 1681 |      | <pre>/lplopa/Compare/camb_des/Matrix_u 90, Top line: 1681</pre> |
|------|---|------|---|
| 1681 | call Matrix SVD(M,n,n,w,V                                   | 1681 | call Matrix SVD(M,n,n,w,V                                       |
| 1682 | _ ` ` ' ' ' '   | 1682 | _ ` ` ` ` ` ` `   |
| 1683 | do i=1, n   | 1683 | do i=1, n   |
| 1684 | tmp(i,:) = M(:,i)/w(i                                       | 1684 | tmp(i,:) = M(:,i)/w(i   |
| 1685 | end do  | 1685 | end do  |
| 1686 |   | 1686 |   |
| 1687 | call Matrix Mult_TN(VT,tm                                   | 1687 | call Matrix Mult_TN(VT,tm                                       |
| 1688 | ! $M = matmul(transpose(V))$                                | 1688 | ! $M = matmul(transpose(V))$                                    |
| 1689 | ` - `   | 1689 | · - ·   |
| 1690 | deallocate(tmp,VT)  | 1690 | deallocate(tmp,VT)  |
| 1691 | , _   | 1691 | · - · ·   |
| 1692 | end subroutine Matrix Inv                                   | 1692 | end subroutine Matrix Inv                                       |
| 1693 | <del>-</del>  | 1693 | <del>_</del>  |
| 1694 | <pre>subroutine Matrix_SVD(Mat</pre>                        | 1694 | subroutine Matrix SVD(Mat                                       |
| 1695 | !Do singular value decomp                                   | 1695 | !Do singular value decomp                                       |
| 1696 | $!Mat = U D V^T$  | 1696 | $!Mat = U D V^T$  |
| 1697 | !returns U in Mat, vector                                   | 1697 | !returns U in Mat, vector                                       |
| 1698 | <pre>integer, intent(in) :: m,</pre>                        | 1698 | <pre>integer, intent(in) :: m,</pre>                            |
| 1699 | <pre>real(dm), intent(inout) :</pre>                        | 1699 | <pre>real(dm), intent(inout) :</pre>                            |
| 1700 | real(dm), intent(out) ::                                    | 1700 | real(dm), intent(out) ::  |
| 1701 |   | 1701 |   |
| 1702 | integer WorkSize, ierr                                      | 1702 | integer WorkSize, ierr  |
| 1703 | real(dm), allocatable, di                                   | 1703 | real(dm), allocatable, di                                       |
| 1704 | ·   | 1704 |   |
| 1705 |   | 1705 |   |
| 1706 | WorkSize=3*n**2   | 1706 | WorkSize=3*n**2   |
| 1707 |   | 1707 |   |
| 1708 | allocate(rv1(WorkSize))                                     | 1708 | allocate(rv1(WorkSize))   |
| 1709 | call Matrix_Start('SVD')                                    | 1709 | call Matrix_Start('SVD')  |
| 1710 | #ifdef MATRIX_SINGLE \                                      | 1710 | #ifdef MATRIX_SINGLE `  |

|      | /lplopa/Compare/camb_simdata/Matr       | /Users/lplopa/Compare/camb_des/Matrix_u |
|------|---|---|
|      | ls.F90, Top line: 1711                  | tils.F90, Top line: 1711                |
| 1711 |   | 1711 call SGESVD('O','A',m,n,           |
| 1712 | #else                                   | 1712 #else                              |
| 1713 | call DGESVD('O','A',m,n,                | 1713 call DGESVD('O','A',m,n,           |
| 1714 | #endif                                  | 1714 #endif                             |
| 1715 | if (ierr/=0) call MpiStop               | 1715   if (ierr/=0) call MpiStop        |
| 1716 | call Matrix End('SVD')                  | 1716   call Matrix End('SVD')           |
| 1717 | _ ` ,                                   | 1717                                    |
| 1718 | deallocate(rv1)                         | 1718 deallocate(rv1)                    |
| 1719 |   | 1719                                    |
| 1720 | end subroutine Matrix SVD               | 1720 end subroutine Matrix SVD          |
| 1721 | <del>-</del>                            | 1721                                    |
| 1722 | subroutine Matrix SVD VT(               | 1722 subroutine Matrix SVD VT(          |
| 1723 | !Do singular value decomp               | 1723 !Do singular value decomp          |
| 1724 | !Mat = U D V^dag                        | 1724 !Mat = U D V^dag                   |
| 1725 | !returns V^dag in Mat, ve               | 1725 !returns V^dag in Mat, ve          |
| 1726 | integer, intent(in) :: m,               | integer, intent(in) :: m,               |
| 1727 | real(dm), intent(inout):                | real(dm), intent(inout):                |
| 1728 | real(dm), intent(out), opt              | 1728 real(dm), intent(out), opt         |
| 1729 | real(dm), intent(out)::                 | real(dm), intent(out)::                 |
| 1730 |   | 1730                                    |
| 1731 | integer WorkSize, ierr                  | 1731 integer WorkSize, ierr             |
| 1732 | integer, allocatable, dime              |   |
| 1733 | real(dm), allocatable, di               | ,                                       |
| 1734 | real(dm) OptWk                          | 1734 real(dm) OptWk                     |
| 1735 | <b>`</b>                                | 1735                                    |
| 1736 | <pre>if (n&lt;=m) call MpiStop('M</pre> | if (n<=m) call MpiStop('M               |
| 1737 | , ,                                     | 1737                                    |
| 1738 | call Matrix Start('SVD VT               | 1738 call Matrix Start('SVD VT          |
| 1739 |   | 1739                                    |
| 1740 | <pre>if (present(U) .and. Matr</pre>    |   |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 1741</pre> | _    | plopa/Compare/camb_des/Matrix_u<br>, Top line: 1741 |
|------|---|------|---|
| 1741 | !Use divide and conqu   | 1741 | !Use divide and conqu                               |
| 1742 | allocate(IWork(8*MIN(   | 1742 | allocate(IWork(8*MIN(                               |
| 1743 | WorkSize= -1 !3*min(M   | 1743 | WorkSize= -1 !3*min(M                               |
| 1744 | #ifdef MATRIX SINGLE  | 1744 | #ifdef MATRIX SINGLE                                |
| 1745 | call SGESDD('O',m,n,  | 1745 | call SGESDD('O',m,n,                                |
| 1746 | #else   | 1746 | #else \'\'\'\'                                      |
| 1747 | call DGESDD('O',m,n,  | 1747 | call DGESDD('O',m,n,                                |
| 1748 | #endif  | 1748 | #endif  |
| 1749 | WorkSize = nint(OptWk   | 1749 | WorkSize = nint(OptWk                               |
| 1750 | allocate(rv1(WorkSize   | 1750 | allocate(rv1(WorkSize                               |
| 1751 | #ifdef MATRIX SINGLE `  | 1751 | #ifdef MATRIX SINGLE                                |
| 1752 | $call \overline{SGESDD}('O',m,n,$                                   | 1752 | call $\overline{SGESDD}('O',m,n,$                   |
| 1753 | #else   | 1753 | #else   |
| 1754 | call DGESDD('O',m,n,  | 1754 | call DGESDD('O',m,n,                                |
| 1755 | #endif  | 1755 | #endif  |
| 1756 | deallocate(IWOrk)   | 1756 | deallocate(IWOrk)                                   |
| 1757 | else  | 1757 | else  |
| 1758 | call MpiStop('Matrix_   | 1758 | call MpiStop('Matrix_                               |
| 1759 | end if  | 1759 | end if  |
| 1760 |   | 1760 |   |
| 1761 | if (ierr/=0) call MpiStop   | 1761 | if (ierr/=0) call MpiStop                           |
| 1762 | deallocate(rv1)   | 1762 | deallocate(rv1)                                     |
| 1763 |   | 1763 |   |
| 1764 | call Matrix_End('SVD_VT')   | 1764 | call Matrix_End('SVD_VT')                           |
| 1765 |   | 1765 |   |
| 1766 | end subroutine Matrix_SVD   | 1766 | <pre>end subroutine Matrix_SVD</pre>                |
| 1767 |   | 1767 |   |
| 1768 |   | 1768 |   |
| 1769 | subroutine Matrix_CSVD_VT   | 1769 | subroutine Matrix_CSVD_VT                           |
| 1770 | !Do singular value decomp   | 1770 | !Do singular value decomp                           |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 1771</pre> |      | /lplopa/Compare/camb_des/Matrix_u<br>90, Top line: 1771 |
|------|---|------|---|
|      |   |      |   |
| 1771 | !Mat = U D V^dag  | 1771 | !Mat = U D V^dag  |
| 1772 | !returns V^dag in Mat, ve   | 1772 | ,   |
| 1773 | <pre>integer, intent(in) :: m,</pre>                                | 1773 | <pre>integer, intent(in) :: m,</pre>                    |
| 1774 | complex(dm), intent(inout   | 1774 | complex(dm), intent(inout                               |
| 1775 | complex(dm), intent(out),   | 1775 | complex(dm), intent(out),                               |
| 1776 | real(dm), intent(out) ::  | 1776 | real(dm), intent(out) ::                                |
| 1777 |   | 1777 |   |
| 1778 | integer WorkSize, ierr  | 1778 | integer WorkSize, ierr                                  |
| 1779 | integer, allocatable, dime  | 1779 | integer, allocatable, dime                              |
| 1780 | <pre>complex(dm), allocatable,</pre>                                | 1780 | <pre>complex(dm), allocatable,</pre>                    |
| 1781 | real(dm), allocatable, di   | 1781 | real(dm), allocatable, di                               |
| 1782 |   | 1782 |   |
| 1783 | <pre>if (n&lt;=m) call MpiStop('M</pre>                             | 1783 | <pre>if (n&lt;=m) call MpiStop('M</pre>                 |
| 1784 |   | 1784 |   |
| 1785 | call Matrix Start('CSVD_V   | 1785 | call Matrix Start('CSVD_V                               |
| 1786 | _   | 1786 |   |
| 1787 |   | 1787 |   |
| 1788 | <pre>if (present(U) .and. Matr</pre>                                | 1788 | <pre>if (present(U) .and. Matr</pre>                    |
| 1789 | !Use divide and conqu   | 1789 | !Use divide and conqu                                   |
| 1790 | WorkSize= 2*min(M,N)*   | 1790 | WorkSize= 2*min(M,N)*                                   |
| 1791 | allocate(rv1(WorkSize   | 1791 | allocate(rv1(WorkSize                                   |
| 1792 | allocate(rwork(5*min(   | 1792 | allocate(rwork(5*min(                                   |
| 1793 | allocate(IWork(8*MIN(   | 1793 | allocate(IWork(8*MIN(                                   |
| 1794 | #ifdef MATRIX SINGLE  | 1794 | #ifdef MATRIX SINGLE                                    |
| 1795 | call $\overline{CGESDD}('O',m,n,$                                   | 1795 | call CGESDD('O',m,n,                                    |
| 1796 | #else   | 1796 | #else   |
| 1797 | call ZGESDD('O',m,n,  | 1797 | call ZGESDD('O', m, n,                                  |
| 1798 | #endif  | 1798 | #endif  |
| 1799 | deallocate(IWOrk)   | 1799 | deallocate(IWOrk)                                       |
| 1800 | else  | 1800 | else  |
|      |   |      |   |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 1801</pre> |      | /lplopa/Compare/camb_des/Matrix_u<br>90, Top line: 1801 |
|------|---|------|---|
| 1801 |   | 1801 |   |
| 1802 | allocate(rwork((max(3*min   | 1802 | allocate(rwork((max(3                                   |
| 1803 | WorkSize= 3*max(m,n)**2   | 1803 | WorkSize= 3*max(m,n)*                                   |
| 1804 | allocate(rv1(WorkSize), S   | 1804 | allocate(rv1(WorkSize                                   |
| 1805 | if (ierr /=0) then  | 1805 | if (ierr /=0) then                                      |
| 1806 | WorkSize= MAX(3*MIN(M   | 1806 | WorkSize= MAX(3*M                                       |
| 1807 | allocate(rv1(WorkSize   | 1807 | allocate(rv1(Work                                       |
| 1808 | end if  | 1808 | end if  |
| 1809 | #ifdef MATRIX SINGLE  | 1809 | #ifdef MATRIX SINGLE                                    |
| 1810 | if (present(U)) then  | 1810 | if (present(U)) then                                    |
| 1811 | call CGESVD('S','O',m   | 1811 | call CGESVD('S','                                       |
| 1812 | else  | 1812 | else  |
| 1813 | call CGESVD('N','O',m   | 1813 | call CGESVD('N','                                       |
| 1814 | end if  | 1814 | end if  |
| 1815 | #else   | 1815 | #else   |
| 1816 | <pre>if (present(U)) then</pre>                                     | 1816 | <pre>if (present(U)) then</pre>                         |
| 1817 | call ZGESVD('S','O',m   | 1817 | call ZGESVD('S','                                       |
| 1818 | else  | 1818 | else  |
| 1819 | call ZGESVD('N','O',m   | 1819 | call ZGESVD('N','                                       |
| 1820 | end if  | 1820 | end if  |
| 1821 | #endif  | 1821 | #endif  |
| 1822 | end if  | 1822 | end if  |
| 1823 |   | 1823 |   |
| 1824 | if (ierr/=0) call MpiStop   | 1824 | if (ierr/=0) call MpiStop                               |
| 1825 | deallocate(rv1)   | 1825 | deallocate(rv1)   |
| 1826 |   | 1826 |   |
| 1827 | call Matrix_End('SVD_VT')   | 1827 | call Matrix_End('SVD_VT')                               |
| 1828 |   | 1828 | <del>-</del>  |
| 1829 | end subroutine Matrix_CSV   | 1829 | <pre>end subroutine Matrix_CSV</pre>                    |
| 1830 |   | 1830 |   |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 1831</pre>   |       | /lplopa/Compare/camb_des/Matrix_u<br>90, Top line: 1831 |
|------|---|-------|---|
| 1831 | subroutine Matrix CSVD U(   | 1831  | subroutine Matrix CSVD U(                               |
| 1832 | !Do singular value decomp   | 1832  | !Do singular value decomp                               |
| 1833 | !Mat = U D VT   | 1833  | !Mat = U D VT   |
| 1834 | !returns U in Mat, vector   | 1834  | !returns U in Mat, vector                               |
| 1835 | <pre>integer, intent(in) :: m,</pre>  | 1835  | <pre>integer, intent(in) :: m,</pre>                    |
| 1836 | complex(dm), intent(inout   | 1836  | complex(dm), intent(inout                               |
| 1837 | complex(dm), intent(induction),   | 1837  | complex(dm), intent(indut),                             |
| 1838 | real(dm), intent(out) ::  | 1838  | real(dm), intent(out) ::                                |
| 1839 | integer WorkSize, ierr  | 1839  | integer WorkSize, ierr                                  |
| 1840 | integer worksize, left<br>integer, allocatable, dime  | 1840  | integer, allocatable, dime                              |
| 1841 | complex(dm), allocatable,   | 1841  | complex(dm), allocatable,                               |
| 1842 | real(dm), allocatable, di   | 1842  | real(dm), allocatable, di                               |
| 1843 | rear(am), arrocatable, ar   | 1843  | rear(am), arrocaeabre, ar                               |
| 1844 | call Matrix Start('CSVD U   | 1844  | call Matrix Start('CSVD U                               |
| 1845 | cail Maclix_beare( cbvb_6   | 1845  | call Maclix_beale( cbvb_6                               |
| 1846 | if (m <n) call="" mpistop('ma<="" td=""><td>1846</td><td>if (m<n) call="" mpistop('ma<="" td=""></n)></td></n)> | 1846  | if (m <n) call="" mpistop('ma<="" td=""></n)>           |
| 1847 | ii (m ii) caii npibcop( na  | 1847  | ii (mai) daii iipiboop( ma                              |
| 1848 | <pre>if (present(VT) .and. Mat</pre>  | 1848  | <pre>if (present(VT) .and. Mat</pre>                    |
| 1849 | WorkSize= 2*min(M,N)*   | 1849  | WorkSize= 2*min(M,N)*                                   |
| 1850 | allocate(rv1(WorkSize   | 1850  | allocate(rv1(WorkSize                                   |
| 1851 | allocate(rwork(5*min(   | 1851  | allocate(rwork(5*min(                                   |
| 1852 | allocate(IWork(8*MIN(   | vvvvv | allocate(IWork(8*MIN(                                   |
| 1853 | #ifdef MATRIX SINGLE  | 1853  | #ifdef MATRIX SINGLE                                    |
| 1854 | call CGESDD('O',m,n,  |       | call CGESDD('O',m,n,                                    |
| 1855 | #else   | 1855  | #else   |
| 1856 | call ZGESDD('O', m, n,  | 1856  | call ZGESDD('O', m, n,                                  |
| 1857 | #endif  | 1857  | #endif  |
| 1858 | deallocate(IWOrk)   | 1858  | deallocate(IWOrk)                                       |
| 1859 | else  | 1859  | else  |
| 1860 | allocate(rwork((max(3   | 1860  | allocate(rwork((max(3                                   |
|      | , , , ,   |       | · · · · · ·   |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 1861</pre> |      | <pre>/lplopa/Compare/camb_des/Matrix_u 90, Top line: 1861</pre> |
|------|---|------|---|
| 1861 |   | 1861 |   |
| 1862 | WorkSize= 3*max(m,n)*   | 1862 | WorkSize= 3*max(m,n)*   |
| 1863 | allocate(rv1(WorkSize   | 1863 | allocate(rv1(WorkSize   |
| 1864 | if (ierr /=0) then  | 1864 | if (ierr /=0) then  |
| 1865 | WorkSize= MAX(3*M   | 1865 | WorkSize= MAX(3*M   |
| 1866 | allocate(rv1(Work   | 1866 | allocate(rv1(Work   |
| 1867 | end if  | 1867 | end if  |
| 1868 | #ifdef MATRIX SINGLE  | 1868 | #ifdef MATRIX SINGLE  |
| 1869 | if (present(VT)) then   | 1869 | if (present(VT)) then   |
| 1870 | call CGESVD('O','   | 1870 | call CGESVD('O','   |
| 1871 | else  | 1871 | else  |
| 1872 | call CGESVD('O','   | 1872 | call CGESVD('O','   |
| 1873 | end if  | 1873 | end if  |
| 1874 | #else   | 1874 | #else   |
| 1875 | if (present(VT)) then   | 1875 | if (present(VT)) then   |
| 1876 | call ZGESVD('O','   | 1876 | call ZGESVD('O','   |
| 1877 | else  | 1877 | else  |
| 1878 | call ZGESVD('O','   | 1878 | call ZGESVD('O','   |
| 1879 | end if  | 1879 | end if  |
| 1880 | #endif  | 1880 | #endif  |
| 1881 | end if  | 1881 | end if  |
| 1882 | if (ierr/=0) call MpiStop   | 1882 | if (ierr/=0) call MpiStop                                       |
| 1883 | call Matrix End( 'CSVD U')  | 1883 | call Matrix End( 'CSVD U')                                      |
| 1884 | /   | 1884 | _ ` _ /   |
| 1885 | deallocate(rv1,rwork)   | 1885 | deallocate(rv1,rwork)   |
| 1886 |   | 1886 | ,   |
| 1887 | end subroutine Matrix CSV   | 1887 | end subroutine Matrix CSV                                       |
| 1888 | <del>-</del>  | 1888 | <del>-</del>  |
| 1889 |   | 1889 |   |
| 1890 | subroutine Matrix_CSVD_Al   | 1890 | subroutine Matrix_CSVD_Al                                       |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 1891</pre> |      | /lplopa/Compare/camb_des/Matrix_u<br>90, Top line: 1891 |
|------|---|------|---|
| 1891 | ! n>m   | 1891 | ! n>m   |
| 1892 | !Do singular value decomp   | 1892 | !Do singular value decomp                               |
| 1893 | !Mat = U D V^daq  | 1893 | !Mat = U D V^dag  |
| 1894 | !returns all nxn V^dag in   | 1894 | !returns all nxn V^dag in                               |
| 1895 | <pre>integer, intent(in) :: m,</pre>                                | 1895 | <pre>integer, intent(in) :: m,</pre>                    |
| 1896 | complex(dm), intent(inout   | 1896 | complex(dm), intent(inout                               |
| 1897 | complex(dm), intent(out):   | 1897 | <pre>complex(dm), intent(out):</pre>                    |
| 1898 | real(dm), intent(out) ::  | 1898 | real(dm), intent(out) ::                                |
| 1899 |   | 1899 |   |
| 1900 | integer WorkSize, ierr  | 1900 | integer WorkSize, ierr                                  |
| 1901 | <pre>complex(dm), allocatable,</pre>                                | 1901 | <pre>complex(dm), allocatable,</pre>                    |
| 1902 | <pre>complex(dm), allocatable,</pre>                                | 1902 | complex(dm), allocatable,                               |
| 1903 | integer, allocatable, dim   | 1903 | integer, allocatable, dim                               |
| 1904 | real(dm), allocatable, di   | 1904 | real(dm), allocatable, di                               |
| 1905 |   | 1905 |   |
| 1906 |   | 1906 |   |
| 1907 | call Matrix_Start('CSVD_A   | 1907 | call Matrix_Start('CSVD_A                               |
| 1908 | <u> </u>  | 1908 |   |
| 1909 | <pre>if (Matrix_method == Mat_</pre>                                | 1909 | <pre>if (Matrix_method == Mat_</pre>                    |
| 1910 | !Divide and conquer d   | 1910 | !Divide and conquer d                                   |
| 1911 | WorkSize= 2*min(M,N)*   | 1911 | WorkSize= 2*min(M,N)*                                   |
| 1912 | allocate(rv1(WorkSize   | 1912 | allocate(rv1(WorkSize                                   |
| 1913 | allocate(rwork(5*min(   | 1913 | allocate(rwork(5*min(                                   |
| 1914 | allocate(IWork(8*MIN(   | 1914 | allocate(IWork(8*MIN(                                   |
| 1915 | allocate(U(m,m))  | 1915 | allocate(U(m,m))  |
| 1916 | #ifdef MATRIX_SINGLE  | 1916 | #ifdef MATRIX_SINGLE                                    |
| 1917 | call CGESDD('A',m,n,  | 1917 | call CGESDD('A',m,n,                                    |
| 1918 | #else   | 1918 | #else   |
| 1919 | call ZGESDD('A',m,n,  | 1919 | call ZGESDD('A',m,n,                                    |
| 1920 | #endif  | 1920 | #endif  |

| /Users/lplopa/Compare/camb_simdata/Matr ix utils.F90, Top line: 1921 |   |      | /lplopa/Compare/camb_des/Matrix_u<br>90, Top line: 1921 |
|--|---|------|---|
| 1921   | deallocate(U)                           | 1921 | deallocate(U)   |
| 1922   | deallocate(U) deallocate(IWork)         | 1922 | deallocate(U)   |
| 1923   | usu2255u55(2m52m)                       | 1923 | usu2256uss(2115211)                                     |
| 1924   | else                                    | 1924 | else  |
| 1925   |   | 1925 | WorkSize= 2*m*n + 2*                                    |
| 1926   | allocate(rwork(5*max(                   | 1926 | allocate(rwork(5*max(                                   |
| 1927   | allocate(rv1(WorkSize                   | 1927 | allocate(rv1(WorkSize                                   |
| 1928   | if $(ierr^{\prime}/=0)$ then            | 1928 | if (ierr`/=0) then                                      |
| 1929   | `WorkSize= MAX(3*M                      | 1929 | WorkSize MAX(3*M  |
| 1930   | allocate(rv1(Work                       | 1930 | allocate(rv1(Work                                       |
| 1931   | end if                                  | 1931 | end if  |
| 1932   | #ifdef MATRIX SINGLE                    | 1932 | #ifdef MATRIX SINGLE                                    |
| 1933   | call CGESVD('N','A',m                   | 1933 | call CGESVD('N','A',m                                   |
| 1934   | #else                                   | 1934 | #else   |
| 1935   | call ZGESVD('N','A',m                   | 1935 | call ZGESVD('N','A',m                                   |
| 1936   | #endif                                  | 1936 | #endif  |
| 1937   | end if                                  | 1937 | end if  |
| 1938   |   | 1938 |   |
| 1939   | if (ierr/=0) call MpiStop               | 1939 | if (ierr/=0) call MpiStop                               |
| 1940   | deallocate(rv1,rwork)                   | 1940 | deallocate(rv1,rwork)                                   |
| 1941   | •                                       | 1941 |   |
| 1942   | call Matrix_End('CSVD_All               | 1942 | call Matrix End('CSVD All                               |
| 1943   | _ ` _                                   | 1943 | _ ` _   |
| 1944   |   | 1944 |   |
| 1945   | end subroutine Matrix CSV               | 1945 | end subroutine Matrix CSV                               |
| 1946   | <del>-</del>                            | 1946 | _   |
| 1947   |   | 1947 |   |
| 1948   | subroutine Matrix DiagPre               | 1948 | subroutine Matrix DiagPre                               |
| 1949   | ! M -> matmul(dia $\overline{g}(D)$ ,M) | 1949 | ! M -> matmul(dia $\overline{g}(D)$ ,M)                 |
| 1950   | real(dm), intent(inout):                | 1950 | real(dm), intent(inout):                                |

|        | <pre>/lp1opa/Compare/camb_simdata/Matr</pre> | /Users | <pre>/lplopa/Compare/camb_des/Matrix_u</pre> |
|--------|--|--------|--|
| ix_uti | ls.F90, Top line: $19\overline{5}1$          | tils.F | 90, Top line: 1951                           |
| 1951   | real(dm), intent(in) :: D                    | 1951   | real(dm), intent(in) :: D                    |
| 1952   | integeríi                                    | 1952   | integer i                                    |
| 1953   |  | 1953   |  |
| 1954   | <pre>if (Size(D) /= SiZE(M,DIM</pre>         | 1954   | <pre>if (Size(D) /= SiZE(M,DIM</pre>         |
| 1955   | do i = 1, size(D)                            | 1955   | do i = 1, size(D)                            |
| 1956   | M(i,:) = M(i,:)*D(i)                         | 1956   | M(i,:) = M(i,:)*D(i)                         |
| 1957   | end do                                       | 1957   | end do                                       |
| 1958   |  | 1958   |  |
| 1959   | end subroutine Matrix Dia                    | 1959   | end subroutine Matrix Dia                    |
| 1960   | _  | 1960   | _  |
| 1961   |  | 1961   |  |
| 1962   | subroutine Matrix_SolveSy                    | 1962   | subroutine Matrix_SolveSy                    |
| 1963   | real(dm), intent(out) ::                     | 1963   | real(dm), intent(out) ::                     |
| 1964   | real(dm), intent(in):: M(                    | 1964   | real(dm), intent(in):: M(                    |
| 1965   | <pre>integer IPIV(size(a)),inf</pre>         | 1965   | <pre>integer IPIV(size(a)),inf</pre>         |
| 1966   | <pre>real(dm), dimension(:,:),</pre>         | 1966   | <pre>real(dm), dimension(:,:),</pre>         |
| 1967   | <pre>real(dm), dimension(:), a</pre>         | 1967   | real(dm), dimension(:), a                    |
| 1968   | integer n, WorkSize                          | 1968   | integer n, WorkSize                          |
| 1969   |  | 1969   |  |
| 1970   | n=Size(M,DIM=1)                              | 1970   | n=Size(M,DIM=1)                              |
| 1971   | if (n<=1) return                             | 1971   | if (n<=1) return                             |
| 1972   | if (Size(M,DIM=2)/=n) cal                    | 1972   | if (Size(M,DIM=2)/=n) cal                    |
| 1973   | call Matrix_Start('SolveS                    | 1973   | call Matrix_Start('SolveS                    |
| 1974   |  | 1974   |  |
| 1975   |  | 1975   |  |
| 1976   | WorkSize = n**2                              | 1976   | WorkSize = n**2                              |
| 1977   | allocate(work(WorkSize))                     | 1977   | allocate(work(WorkSize))                     |
| 1978   | allocate(tmp(n,n))                           | 1978   | allocate(tmp(n,n))                           |
| 1979   | tmp = M                                      | 1979   | tmp = M                                      |
| 1980   | #ifdef MATRIX_SINGLE                         | 1980   | #ifdef MATRIX_SINGLE                         |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 1981</pre> |      | s/lplopa/Compare/camb_des/Matrix_u<br>'90, Top line: 1981 |
|------|---|------|---|
| 1981 | call SSYTRF('U',n,tmp,n,I   | 1981 | call SSYTRF('U',n,tmp,n,I                                 |
| 1982 | #else   | 1982 | #else   |
| 1983 | call DSYTRF('U',n,tmp,n,I   | 1983 | call DSYTRF('U',n,tmp,n,I                                 |
| 1984 | #endif  | 1984 | #endif  |
| 1985 | deallocate(work)  | 1985 | deallocate(work)  |
| 1986 | if (info/=0) call MpiStop   | 1986 | if (info/=0) call MpiStop                                 |
| 1987 | soln(1:n) = a(1:n)  | 1987 | soln(1:n) = a(1:n)  |
| 1988 | #ifdef MATRIX SINGLE  | 1988 | #ifdef MATRIX SINGLE                                      |
| 1989 | call SSYTRS('U',n,1,tmp,n   | 1989 | call SSYTRS('U',n,1,tmp,n                                 |
| 1990 | #else   | 1990 | #else   |
| 1991 | call DSYTRS('U',n,1,tmp,n   | 1991 | call DSYTRS('U',n,1,tmp,n                                 |
| 1992 | #endif  | 1992 | #endif  |
| 1993 | if (info/=0) call MpiStop   | 1993 | if (info/=0) call MpiStop                                 |
| 1994 | deallocate(tmp)   | 1994 | deallocate(tmp)   |
| 1995 | ` - /   | 1995 | ` _ '   |
| 1996 | call Matrix End('SolveSym   | 1996 | call Matrix_End('SolveSym                                 |
| 1997 | _ ` -   | 1997 | _ ` -   |
| 1998 |   | 1998 |   |
| 1999 | end subroutine Matrix Sol   | 1999 | end subroutine Matrix Sol                                 |
| 2000 | _   | 2000 | _   |
| 2001 |   | 2001 |   |
| 2002 | subroutine Matrix_SolveAS   | 2002 | subroutine Matrix_SolveAS                                 |
| 2003 | real(dm), intent(out) ::  | 2003 | <pre>real(dm), intent(out) ::</pre>                       |
| 2004 | real(dm), intent(in):: M(   | 2004 | <pre>real(dm), intent(in):: M(</pre>                      |
| 2005 | <pre>integer IPIV(size(a)),inf</pre>                                | 2005 | <pre>integer IPIV(size(a)),inf</pre>                      |
| 2006 | <pre>real(dm), dimension(:,:),</pre>                                | 2006 | <pre>real(dm), dimension(:,:),</pre>                      |
| 2007 | integer n   | 2007 | integer n   |
| 2008 |   | 2008 |   |
| 2009 | n=Size(M,DIM=1)   | 2009 | n=Size(M,DIM=1)   |
| 2010 | if (n<=1) return  | 2010 | if (n<=1) return  |

| /Users                       | <pre>/lplopa/Compare/camb_simdata/Matr</pre> | /Users                   | <pre>/lplopa/Compare/camb_des/Matrix_u</pre> |
|------------------------------|--|--------------------------|--|
| ix_utils.F90, Top line: 2011 |  | tils.F90, Top line: 2011 |  |
| 2011                         | if (Size(M,DIM=2)/=n) cal                    | 2011                     | if (Size(M,DIM=2)/=n) cal                    |
| 2012                         |  | 2012                     |  |
| 2013                         | call Matrix Start('SolveA                    | 2013                     | call Matrix Start('SolveA                    |
| 2014                         | _ `  | 2014                     | _ `  |
| 2015                         | <pre>allocate(tmp(n,n))</pre>                | 2015                     | allocate(tmp(n,n))                           |
| 2016                         | tmp = M                                      | 2016                     | tmp = M                                      |
| 2017                         | #ifdef MATRIX SINGLE                         | 2017                     | #ifdef MATRIX SINGLE                         |
| 2018                         | call SGETRF(n,n,tmp,n,IPI                    | 2018                     | call $SGETRF(n,n,tmp,n,IPI)$                 |
| 2019                         | #else  | 2019                     | #else  |
| 2020                         | call DGETRF(n,n,tmp,n,IPI                    | 2020                     | call DGETRF(n,n,tmp,n,IPI                    |
| 2021                         | #endif                                       | 2021                     | #endif                                       |
| 2022                         | if (info/=0) call MpiStop                    | 2022                     | if (info/=0) call MpiStop                    |
| 2023                         | soln(1:n) = a(1:n)                           | 2023                     | soln(1:n) = a(1:n)                           |
| 2024                         | #ifdef MATRIX SINGLE                         | 2024                     | #ifdef MATRIX SINGLE                         |
| 2025                         | call SGETRS('N',n,1,tmp,n                    | 2025                     | call $SGETRS('N',n,1,tmp,n)$                 |
| 2026                         | #else  | 2026                     | #else  |
| 2027                         | call DGETRS('N',n,1,tmp,n                    | 2027                     | call DGETRS('N',n,1,tmp,n                    |
| 2028                         | #endif                                       | 2028                     | #endif                                       |
| 2029                         | if (info/=0) call MpiStop                    | 2029                     | if (info/=0) call MpiStop                    |
| 2030                         | deallocate(tmp)                              | 2030                     | deallocate(tmp)                              |
| 2031                         | , = /  | 2031                     | , -,   |
| 2032                         | call Matrix End('SolveASy                    | 2032                     | call Matrix End('SolveASy                    |
| 2033                         |  | 2033                     | _ ` _  |
| 2034                         | end subroutine Matrix Sol                    | 2034                     | end subroutine Matrix Sol                    |
| 2035                         | _  | 2035                     | _  |
| 2036                         | <pre>function Matrix_vecdot(ve</pre>         | 2036                     | <pre>function Matrix_vecdot(ve</pre>         |
| 2037                         | real(dm) vec1(:), vec2(:)                    | 2037                     | real(dm) vec1(:), vec2(:)                    |
| 2038                         | real(dm) Matrix_vecdot                       | 2038                     | real(dm) Matrix_vecdot                       |
| 2039                         | integer n                                    | 2039                     | integer n                                    |
| 2040                         | #ifdef MATRIX_SINGLE                         | 2040                     | #ifdef MATRIX_SINGLE                         |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 2041</pre> |      | s/lplopa/Compare/camb_des/Matrix_u<br>'90, Top line: 2041 |
|------|---|------|---|
| 2041 | real(dm) sdot   | 2041 | real(dm) sdot   |
| 2042 | external sdot   | 2042 | external sdot   |
| 2043 | #else   | 2043 | #else   |
| 2044 | real(dm) ddot   | 2044 | real(dm) ddot   |
| 2045 | external ddot   | 2045 | external ddot   |
| 2046 | #endif  | 2046 | #endif  |
| 2047 | n=size(vec1)  | 2047 | n=size(vec1)  |
| 2048 | if (n/=size(vec2)) call M   |      | if (n/=size(vec2)) call M                                 |
| 2049 | #ifdef MATRIX SINGLE  | 2049 | #ifdef MATRIX SINGLE                                      |
| 2050 | Matrix vecdot = sdot(n, v   |      | Matrix vecdot = sdot(n, v                                 |
| 2051 | #else   | 2051 | #else   |
| 2052 | Matrix vecdot = ddot(n, v   | 2052 | Matrix vecdot = ddot(n, v                                 |
| 2053 | #endif  | 2053 | #endif  |
| 2054 | end function Matrix vecdo   | 2054 | end function Matrix vecdo                                 |
| 2055 |   | 2055 |   |
| 2056 | function Matrix QuadForm(   |      | function Matrix QuadForm(                                 |
| 2057 | !Get vec^T*Mat*vec where  | 2057 | !Get vec^T*Mat*vec where                                  |
| 2058 | real(dm) Matrix QuadForm  | 2058 | real(dm) Matrix QuadForm                                  |
| 2059 | real(dm) vec(:)   | 2059 | real(dm) vec(:)   |
| 2060 | real(dm) Mat(:,:)   | 2060 | real(dm) Mat(:,:)   |
| 2061 | real(dm), dimension(:), a   |      | real(dm), dimension(:), a                                 |
| 2062 | integer n   | 2062 | integer n   |
| 2063 |   | 2063 |   |
| 2064 | n=size(vec)   | 2064 | n=size(vec)   |
| 2065 | allocate(out(n))  | 2065 | allocate(out(n))  |
| 2066 | call Matrix MulvecSymm(Ma   | 2066 | call Matrix MulvecSymm(Ma                                 |
| 2067 | $Matrix Quad\overline{F}orm = Matrix$                               | 2067 | $Matrix Quad\overline{F}orm = Matrix$                     |
| 2068 | deallocate(out)   | 2068 | deallocate(out)   |
| 2069 |   | 2069 |   |
| 2070 | end function Matrix_QuadF   | 2070 | end function Matrix_QuadF                                 |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 2071</pre> |      | /lplopa/Compare/camb_des/Matrix_u<br>90, Top line: 2071 |
|------|---|------|---|
| 2071 |   | 2071 |   |
| 2072 | subroutine Matrix_MulVec(   |      | subroutine Matrix MulVec(                               |
| 2073 | ! Out = a*Mat*vec + b*out   |      | ! Out = a*Mat*vec + b*out                               |
| 2074 | real(dm), intent(in) :: M   |      | real(dm), intent(in) :: M                               |
| 2075 | real(dm) vec(:)   | 2075 | real(dm) vec(:)   |
| 2076 | real(dm) Out(:)   | 2076 |   |
| 2077 | real(dm), intent(in), opt   |      | real(dm), intent(in), opt                               |
| 2078 | real(dm) mult, beta   | 2078 |   |
| 2079 | integer m, n  | 2079 | integer m, n  |
| 2080 | , ,   | 2080 |   |
| 2081 | call Matrix Start('MulVec   | 2081 | call Matrix_Start('MulVec                               |
| 2082 | _ `   | 2082 | _ `   |
| 2083 | m = Size(Mat,DIM=1)   | 2083 | m = Size(Mat,DIM=1)                                     |
| 2084 | n = Size(Vec)   | 2084 | n = Size(Vec)   |
| 2085 | if $(Size(Mat,DIM=2) /= n)$   | 2085 | if (Size(Mat,DIM=2) /= n)                               |
| 2086 | if (present(a)) then  | 2086 | if (present(a)) then                                    |
| 2087 | mult = a  | 2087 | mult = à  |
| 2088 | else  | 2088 | else  |
| 2089 | mult = ROne   | 2089 | mult = ROne   |
| 2090 | end if  | 2090 | end if  |
| 2091 | <pre>if (present(b)) then</pre>                                     | 2091 | <pre>if (present(b)) then</pre>                         |
| 2092 | beta = b  | 2092 |   |
| 2093 | else  | 2093 | else  |
| 2094 | beta = RZero  | 2094 | beta = RZero  |
| 2095 | end if  | 2095 | end if  |
| 2096 |   | 2096 |   |
| 2097 | <pre>if (matrix method == Mat</pre>                                 | 2097 | <pre>if (matrix method == Mat</pre>                     |
| 2098 | if (beta /= RZero) $t\overline{h}$                                  | 2098 | if (beta /= RZero) $t\overline{h}$                      |
| 2099 | out = a*MatMul(Ma   | 2099 | out = a*MatMul(Ma                                       |
| 2100 | else  | 2100 | else  |

```
/Users/lplopa/Compare/camb des/Matrix u
/Users/lplopa/Compare/camb simdata/Matr
                                         tils.F90, Top line: 2101
ix utils.F90, Top line: 21\overline{01}
2101
                                         2101
                     out = MatMul(Mat,
                                                               out = MatMul(Mat,
                     if (mult /= ROne)
                                         2102
                                                               if (mult /= ROne)
2102
2103
                                         2103
                 end if
                                                           end if
             else
2104
                                         2104
                                                      else
        #ifdef MATRIX SINGLE
                                         2105
                                                  #ifdef MATRIX SINGLE
2105
2106
                                         2106
                                                           call SGEMV('N',m,n,mu
                 call SGEMV('N',m,n,mu
2107
        #else
                                         2107
                                                  #else
2108
                                         2108
                 call DGEMV('N',m,n,mu
                                                           call DGEMV('N',m,n,mu
2109
        #endif
                                         2109
                                                  #endif
2110
             end if
                                         2110
                                                      end if
2111
             call Matrix End('MulVec')
                                         2111
                                                      call Matrix End('MulVec')
2112
                                         2112
2113
                                         2113
             end subroutine Matrix Mul
                                                      end subroutine Matrix Mul
2114
                                         2114
             subroutine Matrix MulVecS 2115
                                                      subroutine Matrix MulVecS
2115
2116
             ! Out = a*Mat*vec + b*out
                                         2116
                                                      ! Out = a*Mat*vec + b*out
2117
                                         2117
             real, intent(in) :: Mat(:
                                                      real, intent(in) :: Mat(:
2118
             real vec(:)
                                         2118
                                                      real vec(:)
                                         2119
                                                      real Out(:)
2119
             real Out(:)
                                                      real, intent(in), optiona
2120
             real, intent(in), optiona 2120
2121
                                         2121
             real mult, beta
                                                      real mult, beta
2122
                                         2122
             integer m,n
                                                      integer m,n
2123
                                         2123
2124
                                         2124
             call Matrix Start('MulVec
                                                      call Matrix Start('MulVec
2125
                                         2125
2126
                                         2126
             m = Size(Mat, DIM=1)
                                                      m = Size(Mat, DIM=1)
             n = Size(Vec)
                                                      n = Size(Vec)
2127
                                         2127
2128
             if (Size(Mat,DIM=2) /= n)
                                         2128
                                                      if (Size(Mat,DIM=2) /= n)
2129
             if (present(a)) then
                                         2129
                                                      if (present(a)) then
                                                           mult = a
2130
                 mult = a
                                         2130
```

|      | /lplopa/Compare/camb_simdata/Matr<br>ls.F90, Top line: 2131 |      | /lplopa/Compare/camb_des/Matrix_u<br>90, Top line: 2131 |
|------|---|------|---|
| 2131 | else  | 2131 | else  |
| 2132 | mult = SOne   | 2132 | mult = SOne   |
| 2133 | end if  | 2133 | end if  |
| 2134 | if (present(b)) then  | 2134 | <pre>if (present(b)) then</pre>                         |
| 2135 | `beta = b   | 2135 | beta = b  |
| 2136 | else  | 2136 | else  |
| 2137 | beta = SZero  | 2137 | beta = SZero  |
| 2138 | end if  | 2138 | end if  |
| 2139 |   | 2139 |   |
| 2140 | <pre>if (matrix method == Mat</pre>                         | 2140 | <pre>if (matrix method == Mat</pre>                     |
| 2141 | if (beta /= SZero) $t\overline{h}$                          | 2141 | if (be $\overline{t}a$ /= SZero) $t\overline{h}$        |
| 2142 | out = a*MatMul(Ma   | 2142 | out = a*MatMul(Ma                                       |
| 2143 | else  | 2143 | else  |
| 2144 | <pre>out = MatMul(Mat,</pre>                                | 2144 | out = MatMul(Mat,                                       |
| 2145 | if (mult /= Sone)   | 2145 | if (mult /= SOne)                                       |
| 2146 | end if  | 2146 | end if `  |
| 2147 | else  | 2147 | else  |
| 2148 | call SGEMV('N',m,n,mu                                       | 2148 | call SGEMV('N',m,n,mu                                   |
| 2149 | end if  | 2149 | end if  |
| 2150 | call Matrix End('MulVecSi                                   | 2150 | call Matrix End('MulVecSi                               |
| 2151 | _ `   | 2151 | _ `   |
| 2152 | end subroutine Matrix Mul                                   | 2152 | end subroutine Matrix Mul                               |
| 2153 | <del>-</del>  | 2153 |   |
| 2154 |   | 2154 |   |
| 2155 |   | 2155 |   |
| 2156 |   | 2156 |   |
| 2157 | subroutine Matrix_MulVecS                                   | 2157 | subroutine Matrix_MulVecS                               |
| 2158 | ! Out = $a*Mat*vec$ + $b*out$                               | 2158 | ! Out = $a*Mat*vec$ + $b*out$                           |
| 2159 | real(dm), intent(in) :: M                                   | 2159 | real(dm), intent(in) :: M                               |
| 2160 | real(dm) vec(:)   | 2160 | real(dm) vec(:)   |

|      | /lplopa/Compare/camb_simdata/Matr<br>ls.F90, Top line: 2161 |              | /lplopa/Compare/camb_des/Matrix_u<br>90, Top line: 2161 |
|------|---|--------------|---|
| 2161 | real(dm) Out(:)   | 2161         | real(dm) Out(:)   |
| 2162 | real(dm), intent(in), opt                                   | 00000        | real(dm), intent(in), opt                               |
| 2163 | real(dm) mult, beta   | 2163         | real(dm) mult, beta                                     |
| 2164 | integer m,n   | 2164         | integer m,n   |
| 2165 | integer m/n   | 2165         | inceger m,n   |
| 2166 | call Matrix Start('MulVec                                   | 2166         | call Matrix Start('MulVec                               |
| 2167 | carr Macrix_Beare( Marvee                                   | 2167         | cail Matlix_Btalt( Mulvet                               |
| 2168 | <pre>m = Size(Mat,DIM=1)</pre>                              | 2168         | <pre>m = Size(Mat,DIM=1)</pre>                          |
| 2169 | n = Size(Vec)   | 2169         | n = Size(Vec)   |
| 2170 | if (m /= n) call MpiStop(                                   | 2170         | if (m /= n) call MpiStop(                               |
| 2171 | if (present(a)) then  | 2171         | if (present(a)) then                                    |
| 2172 | mult = a  | 2172         | mult = a  |
| 2173 | else  | 2172         | else  |
| 2174 | mult = ROne   | 2174         | mult = ROne   |
| 2175 | end if  | 2175         | end if  |
| 2176 | if (present(b)) then  | 2176         | if (present(b)) then                                    |
| 2177 | beta = b  | 2177         | beta = b  |
| 2178 | else  | 2178         | else  |
| 2179 | beta = RZero  | 2179         | beta = RZero  |
| 2180 | end if  | 2179         | end if  |
| 2181 | end II  | 2181         | end II  |
| 2182 | <pre>if (matrix method == Mat</pre>                         |              | <pre>if (matrix method == Mat</pre>                     |
| 2183 | if (beta /= RZero) th                                       |              | if (beta /= RZero) th                                   |
| 2184 | ,   |              | ,   |
| 2185 | <pre>out = a*MatMul(Ma else</pre>                           | 2185         | <pre>out = a*MatMul(Ma else</pre>                       |
| 2186 | _   | 2186         | _   |
| 2187 | out = MatMul(Mat,   |              | out = MatMul(Mat,                                       |
| 2188 | if (mult /= ROne)   | 2187         | if (mult /= ROne)                                       |
| 2189 | end if  | 2188<br>2189 | end if  |
|      | else<br>#:fdof mampry crncte                                |              | else  |
| 2190 | #ifdef MATRIX_SINGLE  | 2190         | #ifdef MATRIX_SINGLE                                    |

|              | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 2191</pre> |              | /lplopa/Compare/camb_des/Matrix_u<br>90, Top line: 2191 |
|--------------|---|--------------|---|
| 2191         |   |              |   |
|              | <pre>call SSYMV('U',m,mult #else</pre>                              |              | call SSYMV('U',m,mult                                   |
| 2192<br>2193 |   | 2192         | #else   |
|              | call DSYMV('U',m,mult   | 2193<br>2194 | call DSYMV('U',m,mult                                   |
| 2194         | #endif  |              | #endif  |
| 2195         | end if  | 2195         | end if  |
| 2196         | call Matrix_End('MulVecSy   |              | call Matrix_End('MulVecSy                               |
| 2197         |   | 2197         |   |
| 2198         | end subroutine Matrix_Mul   | 2198         | end subroutine Matrix_Mul                               |
| 2199         |   | 2199         |   |
| 2200         | subroutine Matrix_MulVecS   | 2200         | subroutine Matrix_MulVecS                               |
| 2201         | ! Out = $a*Mat*vec + b*out$   | 2201         | ! Out = a*Mat*vec + b*out                               |
| 2202         | real, intent(in) :: Mat(:   | 2202         | real, intent(in) :: Mat(:                               |
| 2203         | real vec(:)_  | 2203         | real vec(:)   |
| 2204         | real Out(:)   | 2204         | real Out(:)   |
| 2205         | real, intent(in), optiona   | 2205         | real, intent(in), optiona                               |
| 2206         | real mult, beta   | 2206         | real mult, beta   |
| 2207         | integer m,n   | 2207         | integer m,n   |
| 2208         |   | 2208         |   |
| 2209         | call Matrix_Start('MulVec   | 2209         | call Matrix_Start('MulVec                               |
| 2210         | <u> </u>  | 2210         |   |
| 2211         | m = Size(Mat,DIM=1)   | 2211         | m = Size(Mat,DIM=1)                                     |
| 2212         | n = Size(Vec)   | 2212         | n = Size(Vec)   |
| 2213         | if (m /= n) call MpiStop(   | 2213         | if (m /= n) call MpiStop(                               |
| 2214         | if (present(a)) then  | 2214         | if (present(a)) then                                    |
| 2215         | $\mathbf{\hat{m}ult} = \mathbf{\hat{a}}$                            | 2215         | mult = a  |
| 2216         | else  | 2216         | else  |
| 2217         | mult = SOne   | 2217         | mult = SOne   |
| 2218         | end if  | 2218         | end if  |
| 2219         | <pre>if (present(b)) then</pre>                                     | 2219         | <pre>if (present(b)) then</pre>                         |
| 2220         | beta = b  | 2220         | beta = b  |

| /Users | <pre>/lp1opa/Compare/camb_simdata/Matr</pre> | /Users | /lplopa/Compare/camb_des/Matrix_u                |
|--------|--|--------|--|
| ix_uti | 1s.F90, Top line: 2221                       | tils.F | 90, Top line: 2221                               |
| 2221   | else   | 2221   | else   |
| 2222   | beta = SZero                                 | 2222   | beta = SZero                                     |
| 2223   | end if                                       | 2223   | end if   |
| 2224   |  | 2224   |  |
| 2225   | <pre>if (matrix method == Mat</pre>          | 2225   | <pre>if (matrix method == Mat</pre>              |
| 2226   | if (beta /= RZero) $t\overline{h}$           | 2226   | if (be $\overline{t}a$ /= RZero) $t\overline{h}$ |
| 2227   | out = a*MatMul(Ma                            | 2227   | out = a*MatMul(Ma                                |
| 2228   | else   | 2228   | else   |
| 2229   | out = MatMul(Mat,                            | 2229   | out = MatMul(Mat,                                |
| 2230   | if (mult /= ROne)                            | 2230   | if (mult /= ROne)                                |
| 2231   | end if                                       | 2231   | end if `   |
| 2232   | else   | 2232   | else   |
| 2233   | call SSYMV('U',m,mult                        | 2233   | call SSYMV('U',m,mult                            |
| 2234   | end if                                       | 2234   | end if   |
| 2235   | call Matrix End('MulVecSy                    | 2235   | call Matrix End('MulVecSy                        |
| 2236   | _ ` -  | 2236   | _ ` -  |
| 2237   | end subroutine Matrix Mul                    | 2237   | end subroutine Matrix Mul                        |
| 2238   | _  | 2238   | _  |
| 2239   | function Matrix vecdotSin                    | 2239   | function Matrix vecdotSin                        |
| 2240   | real vec1(:), $vec2(:)$                      | 2240   | real vec1(:), vec2(:)                            |
| 2241   | real Matrix vecdotSingle                     | 2241   | real Matrix vecdotSingle                         |
| 2242   | integer n                                    | 2242   | integer n  |
| 2243   | real sdot                                    | 2243   | real sdot  |
| 2244   | external sdot                                | 2244   | external sdot                                    |
| 2245   |  | 2245   |  |
| 2246   | n=size(vec1)                                 | 2246   | n=size(vec1)                                     |
| 2247   | <pre>if (n/=size(vec2)) call M</pre>         | 2247   | <pre>if (n/=size(vec2)) call M</pre>             |
| 2248   | Matrix_vecdotSingle = sdo                    | 2248   | Matrix_vecdotSingle = sdo                        |
| 2249   | <u> </u>                                     | 2249   | _  |
| 2250   | end function Matrix_vecdo                    | 2250   | end function Matrix_vecdo                        |

|      | <pre>/lplopa/Compare/camb_simdata/Matr ls.F90, Top line: 2251</pre> |      | /lplopa/Compare/camb_des/Matrix_u<br>90, Top line: 2251 |
|------|---|------|---|
| 2251 |   | 2251 |   |
| 2252 |   | 2252 |   |
| 2253 | subroutine Matrix Inverse   | 2253 | subroutine Matrix Inverse                               |
| 2254 | !Invert array of matrices   | 2254 | !Invert array of matrices                               |
| 2255 | <pre>integer, intent(in) :: nm</pre>                                | 2255 | <pre>integer, intent(in) :: nm</pre>                    |
| 2256 | #ifdef GFORTRAN   | 2256 | #ifdef GFORTRAN   |
| 2257 | Type(TMatrixType), target   | 2257 | Type(TMatrixType), target                               |
| 2258 | #else 11 \ 11 //  | 2258 | #else   |
| 2259 | <pre>Type(TMatrixType), target</pre>                                | 2259 | Type(TMatrixType), target                               |
| 2260 | #endif  | 2260 | #endif  |
| 2261 | <pre>Type(TMatrixType), pointe</pre>                                | 2261 | Type(TMatrixType), pointe                               |
| 2262 | integer n   | 2262 | integer n   |
| 2263 | integer i, MpiID, MpiSize   | 2263 | integer i, MpiID, MpiSize                               |
| 2264 | integer sz  | 2264 | integer sz  |
| 2265 | #ifdef MPI  | 2265 | #ifdef MPI  |
| 2266 | integer j, ierr, sid  | 2266 | integer j, ierr, sid                                    |
| 2267 | Type(TMatrixType), target   | 2267 | Type(TMatrixType), target                               |
| 2268 | #endif  | 2268 | #endif  |
| 2269 |   | 2269 |   |
| 2270 | call MpiStat(MpiID, MpiSi   | 2270 | call MpiStat(MpiID, MpiSi                               |
| 2271 | if (MpiId==0) then  | 2271 | if (MpiId==0) then                                      |
| 2272 | n=nmat  | 2272 | n=nmat  |
| 2273 | sz_= Size(Arr(1)%M,DI   | 2273 | sz = Size(Arr(1)%M,DI                                   |
| 2274 | end if  | 2274 | end if  |
| 2275 | ! if (MpiID==0) then  | 2275 | ! if (MpiID==0) then                                    |
| 2276 | ! do i=1,nmat   | 2276 | ! do i=1,nmat   |
| 2277 | ! print *,'inverting  | 2277 | ! print *,'inverting                                    |
| 2278 | ! call Matrix_invers  | 2278 | ! call Matrix_invers                                    |
| 2279 | ! end do  | 2279 | ! end do  |
| 2280 | ! end if  | 2280 | ! end if  |

```
/Users/lplopa/Compare/camb simdata/Matr /Users/lplopa/Compare/camb des/Matrix u
                                          tils.F90, Top line: 2281
ix utils.F90, Top line: 22\overline{81}
2281
                                          2281
                  return
                                                            return
2282
         #ifdef MPI
                                                   #ifdef MPI
                                          2282
2283
             if (MpiID==0) print *, 'M 2283
                                                       if (MpiID==0) print *, 'M
2284
             call MPI BCAST(n,1,MPI IN
                                                       call MPI BCAST(n,1,MPI IN
                                          2284
                                                       call MPI_BCAST(sz,1,MPI_I
2285
             call MPI BCAST(sz,1,MPI I
                                          2285
2286
             if (MpiID/=0) then
                                          2286
                                                       if (MpiID/=0) then
                                          2287
2287
                 allocate(tmp%M(sz,sz)
                                                           allocate(tmp%M(sz,sz)
2288
                 AM => tmp
                                          2288
                                                           AM => tmp
2289
             end if
                                          2289
                                                       end if
2290
         #endif
                                          2290
                                                   #endif
2291
                                          2291
2292
                                          2292
             do i= 1,n
                                                       do i= 1,n
2293
                 if (MpiID==0) AM => A 2293
                                                           if (MpiID==0) AM => A
                                          2294
2294
         #ifdef MPI
                                                   #ifdef MPI
2295
                 if (mod(i,MpiSize)/=M 2295
                                                           if (mod(i,MpiSize)/=M
                                                                !Do nothing
2296
                                          2296
                      !Do nothing
2297
                      if (MpiId==0) the 2297
                                                                if (MpiId==0) the
2298
                                          2298
                          j=mod(i,MpiSi
                                                                    j=mod(i,MpiSi
                                          2299
2299
                          call MPI SEND
                                                                    call MPI SEND
                                                                end if
2300
                      end if
                                          2300
2301
                                          2301
                 else
                                                           else
2302
                      if (MpiId/=0) the 2302
                                                                if (MpiId/=0) the
                                          2303
2303
                          !Get from mai
                                                                    !Get from mai
2304
                                          2304
                          call MPI RECV
                                                                    call MPI RECV
2305
                                          2305
2306
                      end if
                                          2306
                                                                end if
2307
         #endif
                                          2307
                                                   #endif
2308
                      call Matrix Inver 2308
                                                                call Matrix Inver
2309
                                          2309
2310
         #ifdef MPI
                                          2310
                                                   #ifdef MPI
```

| /Users | /lplopa/Compare/camb simdata/Matr                | /Users | /lplopa/Compare/camb des/Matrix u    |
|--------|--|--------|--------------------------------------|
| ix_uti | $1s.F90$ , Top line: $23\overline{1}1$           | tils.F | 90, Top line: 2311                   |
| 2311   | if (MpiID==0) the                                | 2311   | if (MpiID==0) the                    |
| 2312   | $\hat{do} j = max(1,$                            |        | $\hat{do} j = max(1,$                |
| 2313   | sid = mod  | 2313   | sid = mod                            |
| 2314   | call MPI   | 2314   | call MPI                             |
| 2315   | end do   | 2315   | end do                               |
| 2316   | else   | 2316   | else                                 |
| 2317   | call MPI SEND                                    | 2317   | call MPI SEND                        |
| 2318   | $ \stackrel{-}{\texttt{end if}}  \stackrel{-}{}$ | 2318   | end if                               |
| 2319   |  | 2319   |                                      |
| 2320   | end if   | 2320   | end if                               |
| 2321   | #endif   | 2321   | #endif                               |
| 2322   | end do   | 2322   | end do                               |
| 2323   |  | 2323   |                                      |
| 2324   |  | 2324   |                                      |
| 2325   | #ifdef MPI                                       | 2325   | #ifdef MPI                           |
| 2326   | if (MpiID==0) then                               | 2326   | if (MpiID==0) then                   |
| 2327   | do j=n - mod(n,MpiSiz                            | 2327   | do j=n - mod(n,MpiSiz                |
| 2328   | sid= mod(j,MpiSiz                                | 2328   | sid= mod(j,MpiSiz                    |
| 2329   | call MPI_RECV(ARr                                | 2329   | call MPI_RECV(ARr                    |
| 2330   | end do   | 2330   | end do                               |
| 2331   | else   | 2331   | else                                 |
| 2332   | deallocate(tmp%M)                                | 2332   | deallocate(tmp%M)                    |
| 2333   | end if   | 2333   | end if                               |
| 2334   | #endif   | 2334   | #endif                               |
| 2335   | <pre>if (MpiID==0) print *, 'M</pre>             | 2335   | <pre>if (MpiID==0) print *, 'M</pre> |
| 2336   |  | 2336   |                                      |
| 2337   |  | 2337   |                                      |
| 2338   | end subroutine Matrix_Inv                        | 2338   | end subroutine Matrix_Inv            |
| 2339   |  | 2339   |                                      |
| 2340   |  | 2340   |                                      |

|                      | /lplopa/Compare/camb_simdata/Matr<br>ls.F90, Top line: 2341 |                      | /lplopa/Compare/camb_des/Matrix_u<br>00, Top line: 2341 |
|----------------------|---|----------------------|---|
| 2341<br>2342<br>2343 | end module MatrixUtils                                      | 2341<br>2342<br>2343 | end module MatrixUtils                                  |