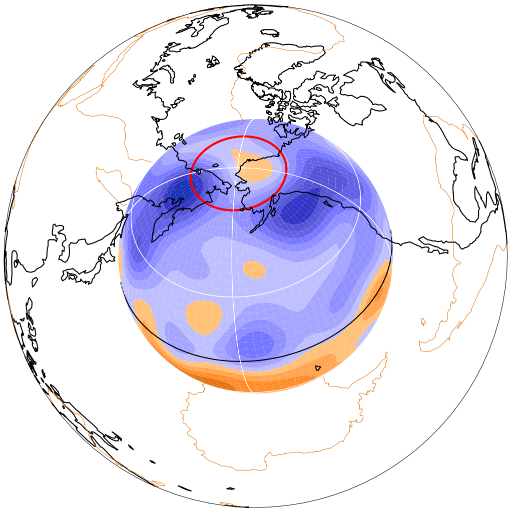
LEOPACK



Information on Key Subroutines

Steven J. Gibbons, Oslo

Original document: November 21^{st}, 2001. Updated: November 20^{th}, 2022.

1 Information on key subroutines

1.1 SUBROUTINE FDCMBD

Finite **D**ifference **C**oefficient **M**atrix **B**uil**D**.

Calling sequence:-

```
FDCMBD( NR, NBN, NLMN, NRMN, NLMC, NRMC, NCFM,
        NFDCM, NDRVS, NDRVM, IWORK, XARR, FDCM,
        COEFM, WORK1, WORK2 )
```

Purpose:-

If XARR is a double precision array of length NR such that XARR(j)$= x_j$, then **FDCMBD** builds a double precision array, FDCM, of dimension (NFDCM, NR, NDRVM) such that for a given node number, j, the n^{th} derivative of $f(x)$ is given by

$$f^{(n)}(x_j) = \sum_{i=\text{LN}}^{\text{RM}} \text{FDCM}(k, j, n) f(x_i)$$

where

```
LN = MAX( NLMC, j - NBN )
RN = MIN( NRMC, j + NBN ),
```

NBN is the number of nodes on each side for central difference formulae,

```
k = i - j + NBN + 1,
```

and NLMC and NRMC are respectively the left most and right most nodes which may be used to obtain a finite difference formula. The FDCM(k, j, n) are calculated for all j with NLMN$\leq j \leq$ NRMN. Elements corresponding to other j are not referred to.

The leading dimension of FDCM must be atleast (2*NBN+1) as must the dimension NCFM.

The maximum value of n calculated is specified by the parameter NDRVS which must be equal to or less than the value NDRVM. The value of NDRVS is limited by the bandwidth as specified by NBN. Let

```
NLCS = NLMN - NLMC
NRCS = NRMC - NRMN
```

and let

```
I = MIN( NLCS, NRCS) + NBN.
```

NDRVS must then be no greater than I.

The maximum value of n calculated is specified by the parameter NDRVS which must be equal to or less than the value NDRVM. The value of NDRVS is limited by the bandwidth as specified by NBN. Let

```
NLCS = NLMN - NLMC
NRCS = NRMC - NRMN
```

and let

```
I = MIN( NLCS, NRCS) + NBN.
```

NDRVS must then be no greater than I.

Arguments:-

NR	Int. scal.	Number of radial grid nodes.
NBN	Int. scal.	Number of bounding nodes.
NLMN	Int. scal.	Left-most (row) node. See above.
NRMN	Int. scal.	Right-most (row) node. See above.
NLMC	Int. scal.	Left-most (column) node. See above.
NRMC	Int. scal.	Right-most (column) node. See above.
NCFM	Int. scal.	Leading dimension of COEFM. Must be atleast (2*NBN+1).
NFDCM	Int. scal.	Leading dimension of FDCM. Must be atleast (2*NBN+1).
NDVRS	Int. scal.	Highest derivative to be calculated. Must not exceed NDVRM and see above.
NDVRM	Int. scal.	Maximum number of derivatives which can be stored in memory.
IWORK	Int. arr.	Dimension (NCFM). Work array.
XARR	D.p. arr.	Values of x. Dimension(NR).
FDCM	D.p. arr.	Finite difference coefficients. Dimension(NFDCM, NR, NDRVM). See above.
COEFM	D.p. arr.	Dimension (NCFM, NCFM). Work array.
WORK1	D.p. arr.	Dimension (NCFM). Work array.
WORK2	D.p. arr.	Dimension (NCFM). Work array.

Subroutines and functions called:-

GFDCFD.

1.2 SUBROUTINE GFDCFD

General **F**inite **D**ifference **C**oefficient **F**in**D**.

Calling sequence:-

```
GFDCFD ( XO, XARR, NNDS, COEFM, NCFM, IPCM, WORK)
```

Purpose:-

If the double precision array `XARR` contains `NNDS` distinct values of x (`XARR( j ) =` x_j) then, for a given value x_0, **GFDCFD** returns the coefficients c^n_{j,x_0} such that if f is a function of x, then

$$f^n(x_0) \approx \sum_{j=1}^{\mathrm{NNDS}} \left(c^n_{j,x_0} f(x_j) \right) .$$

Here, $f^n(x_0)$ denotes the $n^{\rm th}$ derivative of f with respect to x, evaluated at x_0. The coefficient c^n_{j,x_0} is returned in the array element `COEFM( n + 1, J )`. The $n = 0$ coefficients are returned in the first row of the matrix and, although these are not very useful if x_0 corresponds to one of the x_j, they allow a useful way of interpolating a function f to $x = x_0$, if x_0 is distinct from all the x_j.

The higest derivative obtainable from `NNDS` distinct values of x is `NNDS-1` although this estimate is likely to be highly inaccurate. For arbitrarily spaced x, there is no easy way of quantifying the size of the error.

The routine checks to ensure that the `XARR( j )` are all distinct, otherwise it would be attempted to solve a singular matrix.

Arguments:-

`XO`	D.p. scal.	Value of x_0 where deriv.s are required.
`XARR`	D.p. arr.	Dimension (NNDS). See above.
`NNDS`	Int. scal.	Number of grid nodes.
`COEFM`	D.p. arr.	Dimension (NCFM, NCFM). See above.
`NCFM`	Int. scal.	Leading dimension of matrix COEFM. Must be greater than or equal to `NNDS`.
`IPCM`	Int. arr.	Dimension (NCFM). Work array.
`WORK`	D.p. arr.	Dimension (NCFM). Work array.

Subroutines and functions called:-

`MATOP` and the **LAPACK** routines `DGETRF` and `DGETRI`.

1.3 SUBROUTINE HMFRD

Har**M**onic **F**ile **R**ea**D**.

Calling sequence:-

```
HMFRD( NH, NHMAX, MHT, MHL, MHM, MHP, NCUDS, NDCS,
       MHIBC, MHOBC, LARR, LU, FNAME )
```

Purpose:-

Reads the indices of the spherical harmonic set (along with the boundary conditions) from a file specified by the filename `FNAME`.

The first line of the new file will contain only `NH`, the number of spherical harmonics. The next `NH` lines will all consist of the 5 integer values

```
MT   ML   MM    IIBF  IOBF
```

`MT` contains one of the following values which describes the function :-

```
MT = 1 for a poloidal velocity harmonic.
MT = 2 for a toroidal velocity harmonic.
MT = 3 for a temperature harmonic.
MT = 4 for a poloidal magnetic field harmonic.
MT = 5 for a toroidal magnetic field harmonic.
```

`ML` contains the spherical harmonic degree, l, and `MM` will contain m, the spherical harmonic order, for $\cos m\phi$ dependence and $-m$ for $\sin m\phi$ dependence.

`IIBF` and `IOBF` describe the boundary condition required for that radial function at the inner and outer boundaries respectively. (See **SVFDCF**, section (1.7), for details.) They are the values of `MHIBC` and `MHOBC` which correspond to the value of `MHP(ih)`.

The arrays `LARR`, `MHIBC` and `MHOBC` required by **SVFDCF** are calculated in situ in the program. On input, `NCUDS` is the number of currently used differencing schemes (possibly none) and on output, this number will be modified to include those schemes demanded by the new set of spherical harmonics. All difference schemes which are not yet needed are labelled with `LARR( is ) = -1` as required by **SVFDCF**.

Arguments:-

NH	Int. scal.	Number of spherical harmonics.
NHMAX	Int. scal.	Maximum number of spherical harmonics.
MHT	Int. arr.	Dim (NHMAX). MHT(ih) = MT for harmonic number ih.
MHL	Int. arr.	Dim (NHMAX). MHL(ih) = ML for harmonic number ih.
MHM	Int. arr.	Dim (NHMAX). MHM(ih) = MM for harmonic number ih.
MHP	Int. arr.	Dim (NHMAX). MHP(ih) contains the index of the arrays MHIBC and MHOBC which correspond to harmonic ih.
NCUDS	Int. scal.	Number of currently used finite difference schemes.
NDCS	Int. scal.	Maximum distinct finite difference schemes. See **SVFDCF**.
MHIBC	Int. arr.	Dimension (NDCS). MHIBC(is) contains a flag indicating how the inner boundary must be treated in scheme is. See **SVFDCF**.
MHOBC	Int. arr.	Dimension (NDCS). MHOBC(is) contains a flag indicating how the outer boundary must be treated in scheme is. See **SVFDCF**.
LARR	Int. arr.	Dimension (NDCS). Auxilliary array for **SVFDCF**.
LU	Int. scal.	Logical file unit number.
FNAME	Char(*)	Filename.

Subroutines and functions called:-

FOPEN and FCLOSE.

1.4 SUBROUTINE HMFWT

Har**M**onic **F**ile **W**ri**T**e.

Calling sequence:-

```
HMFWT( NH, MHT, MHL, MHM, MHP, NDCS, MHIBC, MHOBC, LU, FNAME )
```

Purpose:-

Wries the indices of the spherical harmonic set (along with the boundary conditions) to a file specified by the filename `FNAME`.

The first line of the new file will contain only `NH`, the number of spherical harmonics. The next `NH` lines will all consist of the 5 integer values

```
MT   ML   MM    IIBF  IOBF
```

`MT` contains one of the following values which describes the function :-

```
MT = 1 for a poloidal velocity harmonic.
MT = 2 for a toroidal velocity harmonic.
MT = 3 for a temperature harmonic.
MT = 4 for a poloidal magnetic field harmonic.
MT = 5 for a toroidal magnetic field harmonic.
```

`ML` contains the spherical harmonic degree, l, and `MM` will contain m, the spherical harmonic order, for $\cos m\phi$ dependence and $-m$ for $\sin m\phi$ dependence.

`IIBF` and `IOBF` describe the boundary condition required for that radial function at the inner and outer boundaries respectively. (See **SVFDCF**, section (1.7), for details.) They are the values of `MHIBC` and `MHOBC` which correspond to the value of `MHP(ih)`.

Arguments:-

`NH`	Int. scal.	Number of spherical harmonics.
`MHT`	Int. arr.	Dim (*) with length atleast NH. `MHT(ih) = MT` for harmonic number `ih`.
`MHL`	Int. arr.	Dim (*) with length atleast NH. `MHL(ih) = MT` for harmonic number `ih`.
`MHM`	Int. arr.	Dim (*) with length atleast NH. `MHM(ih) = MT` for harmonic number `ih`.
`MHP`	Int. arr.	Dim (*) with length atleast NH. `MHP( ih )` contains the index of the arrays `MHIBC` and `MHOBC` which correspond to harmonic `ih`.
`NDCS`	Int. scal.	Maximum distinct finite difference schemes. See **SVFDCF**.
`MHIBC`	Int. arr.	Dimension `( NDCS )`. `MHIBC( is )` contains a flag indicating how the inner boundary must be treated in scheme `is`. See **SVFDCF**.
`MHOBC`	Int. arr.	Dimension `( NDCS )`. `MHOBC( is )` contains a flag indicating how the outer boundary must be treated in scheme `is`. See **SVFDCF**.
`LU`	Int. scal.	Logical file unit number.
`FNAME`	Char(*)	Filename.

Subroutines and functions called:-

`FOPEN` and `FCLOSE`.

1.5 SUBROUTINE LDGNMF

Linear **D**ependence of **G**rid **N**ode **M**atrix **F**orm.

Calling sequence:-

```
LDGNMF( NR, NNDS, NALF, NARF, L, IIBC, IOBC, NCFM,
        XARR, AMAT, WMAT, WORK1, WORK2, IPCM )
```

Purpose:-

Let f_j denote the function f evaluated at x_j, the value stored in `XARR( j )`. Let $f_j^{(m)}$ denote the m^{th} derivative with respect to x evaluated at x_j.

Then a finite difference scheme may be formulated such that

$$\begin{pmatrix} f_j^{(0)} \\ f_j^{(1)} \\ \vdots \\ f_j^{(n-1)} \end{pmatrix} = \begin{pmatrix} c_{k+1}^{(0)} & c_{k+2}^{(0)} & \cdots & c_{k+n}^{(0)} \\ c_{k+1}^{(1)} & c_{k+2}^{(1)} & \cdots & c_{k+n}^{(1)} \\ \vdots & \vdots & \cdots & \vdots \\ c_{k+1}^{(n-1)} & c_{k+2}^{(n-1)} & \cdots & c_{k+n}^{(n-1)} \end{pmatrix} \begin{pmatrix} f_{k+1} \\ f_{k+2} \\ \vdots \\ f_{k+n} \end{pmatrix}.$$

The matrix $\boldsymbol{C}$ above, containing the elements $c_{k+i}^{(m)}$, may be formed by a call to **GFDCFD** with the calling sequence

```
CALL GFDCFD ( XJ, XA, NNDS, COEFM, NCFM, IPCM, WORK)
```

where `XJ` gives the value of x_j, `XA( i )` gives the value of x_{k+i}, `NNDS` is n and `COEFM(` $m+1$`, i )` gives the value of $c_{k+i}^{(m)}$.

However, if boundary conditions must be satisfied at x_1 and $x_{\rm NR}$ (`XARR( 1 )` and `XARR( NR )` respectively) then the values f_{k+1}, f_{k+2}, $\cdots$, f_{k+n} may not be linearly independent. In other words, the equation above becomes

$$\begin{pmatrix} f_j^{(0)} \\ f_j^{(1)} \\ \vdots \\ f_j^{(n-1)} \end{pmatrix} = (\boldsymbol{C})\,(\boldsymbol{A}) \begin{pmatrix} f_{k+1} \\ f_{k+2} \\ \vdots \\ f_{k+n} \end{pmatrix},$$

where the matrix $\boldsymbol{A}$ defines the dependency between the different f_{k+i}. If the boundary condition does not effect the nodes $k{+}1$, $k{+}2$ to $k{+}n$ (to the accuracy of the finite differencing scheme) then $\boldsymbol{A}$ is clearly the identity matrix. Otherwise, $\boldsymbol{A}$ is a matrix of Rank $< n$, and the purpose of **LDGNMF** is to calculate this matrix. The element f_{k+i} is expressed in terms of the f_s (with $s \in \{k+1, k+2, \cdots, k+n\}$) by

$$f_{k+i} = \sum_{s=1}^{n} d_{i,s} f_{k+s}.$$

The integer values `NALF` and `NARF` are respectively the number of nodes at the left and the right for which the value is a linear combination of the values at the other nodes. Both `NALF` and `NARF` **MUST** take one of the values `0`, `1` or `2`. (This is because there is no case in the standard MHD equations where we need to apply more than 2 boundary conditions at a particular boundary.) If both `NALF` and `NARF` are zero then $\boldsymbol{A}$ is returned as $\boldsymbol{I}_n$. If either `NALF` or `NARF` is non-zero then the other must be zero! (Otherwise the finite difference scheme is not going to work).

If `NALF = 1` then f_{k+1} is a linear sum of $\{f_{k+2}, \cdots, f_{k+n}\}$.

If `NALF = 2` then f_{k+1} and f_{k+2} are both linear sums of $\{f_{k+3}, \cdots, f_{k+n}\}$.

If `NARF = 1` then f_{k+n} is a linear sum of $\{f_{k+1}, \cdots, f_{k+n-1}\}$.
If `NARF = 2` then f_{k+n} and f_{k+n-1} are both linear sums of $\{f_{k+1}, \cdots, f_{k+n-2}\}$.

Now the actual form of the boundary condition can be one of many different cases. This is determined by the flags `IIBC` and `IOBC`, which specify the inner and outer boundary condition respectively.

The following table gives the possibilities for `IIBC` :-

`IIBC`	Inner boundary condition.
1	None imposed
2	Function must vanish at boundary.
3	First derivative must vanish at boundary.
4	Both function and first derivative must vanish.
5	Both function and second derivative must vanish.
6	$rdf/dr - f(r) = 0$
7	$rdf/dr - lf(r) = 0$ with $l =$`L`.

The following table gives the possibilities for `IOBC` :-

`IOBC`	Outer boundary condition.
1	None imposed
2	Function must vanish at boundary.
3	First derivative must vanish at boundary.
4	Both function and first derivative must vanish.
5	Both function and second derivative must vanish.
6	$rdf/dr - f(r) = 0$
7	$rdf/dr + (l+1)f(r) = 0$ with $l =$`L`.

Arguments:-

NR	Int. scal.	Total number of radial grid nodes.
NNDS	Int. scal.	Number of nodes which are used to take derivative (n above).
NALF	Int. scal.	Number of nodes at the left which are a linear combination of the others.
NARF	Int. scal.	Number of nodes at the right which are a linear combination of the others.
L	Int. scal.	Spherical harmonic degree, l. Only relevant for magnetic field b.c.s
IIBC	Int. scal.	Inner boundary condition flag. See above.
IOBC	Int. scal.	Outer boundary condition flag. See above.
NCFM	Int. scal.	Leading dimension of matrix AMAT.
XARR	D.p. arr.	Dimension (NR). See above.
AMAT	D.p. arr.	Dimension (NCFM,NCFM). Returns matrix $\boldsymbol{A}$ above.
WMAT	D.p. arr.	Dimension (NCFM,NCFM). Work array.
WORK1	D.p. arr.	Dimension (NCFM). Work array for **GFDCFD**.
WORK2	D.p. arr.	Dimension (NCFM). Work array for **GFDCFD**.
IPCM	Int. arr.	Dimension (NCFM). Work array for **GFDCFD**.

Subroutines and functions called:-

MATOP, GFDCFD.

1.6 SUBROUTINE SCHNLA

Calling sequence:-

```
SCHNLA ( PA, DPA, GAUX, LH, NTHPTS )
```

Purpose:-

To evaluate an array of all the associated Legendre Polynomials, P_l^m, up to and including degree l=LH at NTHPTS different values of θ. The derivatives with respect to θ are also calculated.

GAUX is a double precision array of length NTHPTS with GAUX(i) containing the element $\cos\theta_i$.

PA is an array with dimensions **((LH + 1)*(LH + 2)/2 , NTHPTS)** and $P_l^m(\cos\theta_i)$ is stored in the element PA(K, I) where the value of K is given by the function KLM(L, M): i.e. KLM = L*(L+1)/2+M+1.

DPA is arranged identically except that its elements contain $dP_l^m(\cos\theta_i)/d\theta$.

P_l^m satisfies the normalisation

$$\int_{-1}^{1} \left[P_l^m(x)\right]^2 dx = \frac{2(2-\delta_{m0})}{2l+1}.$$

Arguments:-

PA	D.p. arr. Dim. **(N1 , NTHPTS)** with **N1 = (LH + 1)*(LH + 2)/2**	See above.
DPA	D.p. arr. Dim. **(N1 , NTHPTS)** with **N1 = (LH + 1)*(LH + 2)/2**	See above.
GAUX	D.p. arr. Dim. **(NTHPTS)**	See above
LH	Int. scal.	Maximum degree, l
NTHPTS	Int. scal.	Number of θ points

Subroutines and functions called:-

PMM, PMM1 and PLM; DPMM, DPMM1 and DPLM; KLM.

1.7 SUBROUTINE SVFDCF

Solution **V**ector **F**inite **D**ifference **C**oefficient **F**ind.

Calling sequence:-

```
SVFDCF( NR, NDCS, NBN, NLMR, NRMR, MHIBC, MHOBC,
        LARR, NCFM, NFDCM, NDRVS, NDRVM, XARR,
        IWORK, SVFDC, COEFM1, COEFM2, WORK1, WORK2 )
```

Purpose:-

Let f be a function of x with NR discrete values x_j being stored in XARR(j). **SVFDCF** builds a double precision matrix SVFDC of dimension (NFDCM, NR, NDRVM+1, NDCS) such that for a given node number, j, the $n^{\rm th}$ derivative of $f(x)$ is given by

$$f^{(n)}(x_j) = \sum_{i=\mathrm{LN}}^{\mathrm{RM}} \mathrm{SVFDC}(k, j, n+1, S) f(x_i)$$

where

```
LN = MAX(  1, j - NBN )
RN = MIN( NR, j + NBN ),
```

NBN is the number of nodes on each side for central difference formulae,

```
k = i - j + NBN + 1,
```

and S is an integer value describing one of several different finite-differencing schemes which may all be stored in `SVFDC`. The properties of the scheme S (`is`) is described by two integer parameters, `MHIBC( is )` and `MHOBC( is )`. Their values are given in the following two tables.

The possibilities for `MHIBC( is )` are :-

`MHIBC( is )`	Inner boundary condition.
1	None imposed
2	Function must vanish at boundary.
3	First derivative must vanish at boundary.
4	Both function and first derivative must vanish.
5	Both function and second derivative must vanish.
6	$rdf/dr - f(r) = 0$
7	$rdf/dr - lf(r) = 0$ with $l =$`L`.

The possibilities for `MHOBC( is )` are :-

`MHOBC( is )`	Outer boundary condition.
1	None imposed
2	Function must vanish at boundary.
3	First derivative must vanish at boundary.
4	Both function and first derivative must vanish.
5	Both function and second derivative must vanish.
6	$rdf/dr - f(r) = 0$
7	$rdf/dr + (l+1)f(r) = 0$ with $l =$`L`.

In schemes with `MHIBC( is ) = 1`, all nodes near the inner boundary are required for taking derivatives. In schemes with `MHIBC( is ) =` 2, 3, 6 and 7, the first grid node is essentially omitted from the solution vector and only nodes $2, 3, \cdots$ are used for taking derivatives. In schemes with `MHIBC( is ) =` 4 and 5, the first two grid nodes are omitted and only nodes $3, 4, \cdots$ are used for taking derivatives. A similar situation applies at the outer boundary.

Set `LARR( is )` to the spherical harmonic degree, l, for poloidal magnetic field (insulating boundaries) finite difference schemes and to `-1` to make **SVFDCF** overlook scheme `is` altogether (maybe to be filled using a different set of parameters).

Arguments:-

`NR`	Int. scal.	Total number of radial grid nodes.
`NDCS`	Int. scal.	Number of distinct differencing schemes to be stored in the array `SVFDC`.
`NBN`	Int. scal.	Maximum number of nodes to left or right which may be used to calculate derivatives.
`NLMR`	Int. scal.	The lowest j for which terms are to be calculated for `SVFDC(i,j,ND+1,K)`.
`NRMR`	Int. scal.	The highest j for which terms are to be calculated for `SVFDC(i,j,ND+1,K)`.
`MHIBC`	Int. arr.	Dimension `( NDCS )`. `MHIBC( is )` contains a flag indicating how the inner boundary must be treated in scheme `is`. See above list.
`MHOBC`	Int. arr.	Dimension `( NDCS )`. `MHOBC( is )` contains a flag indicating how the outer boundary must be treated in scheme `is`. See above list.
`LARR`	Int. arr.	Dimension `( NDCS )`. `LARR( is )` contains the spherical harmonic degree, l, for scheme `is`. If `LARR( is ) = -1`, scheme `is` is ignored.
`NCFM`	Int. scal.	Leading dimension of work arrays. Must be atleast $2 * NBN + 1$.
`NFDCM`	Int. scal.	Leading dimension of array `SVFDC`. Must be atleast $2 * NBN + 1$.
`NDRVS`	Int. scal.	Number of derivatives to be calculated.
`NDRVM`	Int. scal.	Maximum no. of deriv. coefficients which may be stored in `SVFDC`.
`XARR`	D.p. arr.	Dimension `(NR)`. Radial grid values.
`IWORK`	Int. arr.	Dimension `(NCFM)`. Work array.
`SVFDC`	D.p. arr.	Dimension `(NFDCM, NR, NDRVM+1, NDCS )`. Coefficients. See above
`COEFM1`	D.p. arr.	Dimension `( NCFM, NCFM )`. Work array.
`COEFM2`	D.p. arr.	Dimension `( NCFM, NCFM )`. Work array.
`WORK1`	D.p. arr.	Dimension `( NCFM )`. Work array.
`WORK2`	D.p. arr.	Dimension `( NCFM )`. Work array.

Subroutines and functions called:-

`LDGNMF`, `GFDCFD`, `EMMULT`.