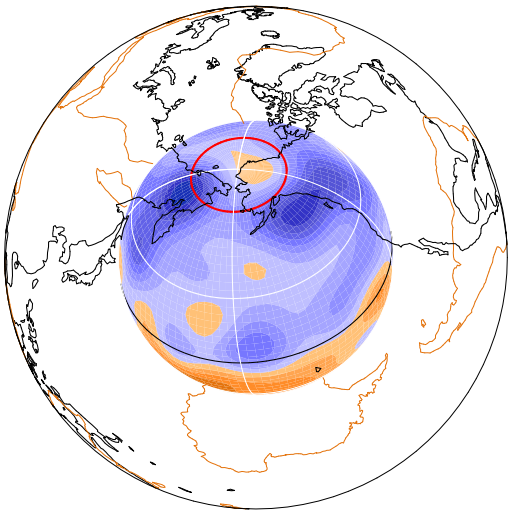
LEOPACK



itfvf

Inhomogeneous **T**emperature **F**unction **V**ector **F**orm

Steven J. Gibbons, Oslo

Original document: November 21^{st}, 2001. Updated: October 23^{rd}, 2022.

1 itfvf

Inhomogeneous **T**emperature **F**unction **V**ector **F**orm

All of the time-stepping codes assume that the part of the temperature being solved for has uniform thermal boundaries. This means that at each of the inner and outer boundaries, the temperature (Θ) or the heat-flux ($-\partial\Theta/\partial r$) is zero. Inner and outer boundaries may have either like or unlike boundary conditions. In order to impose a lateral variation in the temperature or heat-flux at either boundary, we therefore need to form a supplementary temperature function. This is referred to as T_a.

`itfvf` forms a function $T_\mathrm{a}(r,\theta,\phi)$ in the standard format (i.e. with a `.ints`, `.vecs` and `.xarr` file). T_a has an essentially arbitrary form apart from the conditions that it must satisfy either

$$T_\mathrm{a}(r_\mathrm{i},\theta,\phi) = g_\mathrm{i}(\theta,\phi) \tag{1}$$

or

$$\left.\frac{\partial T_\mathrm{a}(r,\theta,\phi)}{\partial r}\right|_{r=r_\mathrm{i}} = g_\mathrm{i}(\theta,\phi) \tag{2}$$

at the inner boundary, and either

$$T_\mathrm{a}(r_\mathrm{o},\theta,\phi) = g_\mathrm{o}(\theta,\phi) \tag{3}$$

or

$$\left.\frac{\partial T_\mathrm{a}(r,\theta,\phi)}{\partial r}\right|_{r=r_\mathrm{o}} = g_\mathrm{o}(\theta,\phi) \tag{4}$$

at the outer boundary: depending upon whether a constant temperature or constant heat-flux boundary condition is to be satisfied.

The functions g_i and g_o are both expanded as series of spherical harmonics:

$$g_\mathrm{i}(\theta,\phi) = \frac{\sqrt{\varepsilon_\mathrm{i}}}{N_\mathrm{i}} \sum_{l=0}^{l=L} \sum_{m=0}^{m=l} \left[g_{\mathrm{i},l}^{mc} \cos m\phi + g_{\mathrm{i},l}^{ms} \sin m\phi \right] P_l^m(\cos\theta) \tag{5}$$

and

$$g_\mathrm{o}(\theta,\phi) = \frac{\sqrt{\varepsilon_\mathrm{o}}}{N_\mathrm{o}} \sum_{l=0}^{l=L} \sum_{m=0}^{m=l} \left[g_{\mathrm{o},l}^{mc} \cos m\phi + g_{\mathrm{o},l}^{ms} \sin m\phi \right] P_l^m(\cos\theta), \tag{6}$$

where the associated Legendre function $P_l^m(\cos\theta)$ satisfies the Schmidt quasi-normalisation condition

$$\int_0^\pi \left[P_l^m(\cos\theta)\right]^2 \sin\theta d\theta = \frac{2(2-\delta_{m0})}{2l+1}. \tag{7}$$

The user specifies the non-zero coefficients $g^{mc}_{\mathrm{i},l}$, $g^{ms}_{\mathrm{i},l}$, $g^{mc}_{\mathrm{o},l}$ and $g^{ms}_{\mathrm{o},l}$, and **itfvf** calculates normalisation factors, N_i and N_o, such that

$$\int_0^{2\pi}\int_0^{\pi}\left[g_\mathrm{i}(\theta,\phi)\right]^2\sin\theta d\theta d\phi = \varepsilon_\mathrm{i} \tag{8}$$

and

$$\int_0^{2\pi}\int_0^{\pi}\left[g_\mathrm{o}(\theta,\phi)\right]^2\sin\theta d\theta d\phi = \varepsilon_\mathrm{o}. \tag{9}$$

Therefore, if a user specified that the only non-zero coefficients in the series (5) and (6) were to be $g^{2c}_{\mathrm{o},2} = 3.0$ and $g^{2s}_{\mathrm{o},3} = -3.0$ with $\varepsilon_\mathrm{o} = 1.0$, the result would be exactly the same had the user specified the coefficients $g^{2c}_{\mathrm{o},2} = 0.012$ and $g^{2s}_{\mathrm{o},3} = -0.012$ with $\varepsilon_\mathrm{o} = 1.0$. This gives the user a global control on the normalisation of the "strength" of the functions g_i and g_o without having to worry about the actually coefficients themselves, other than their relative sizes. I personally tend to set ε_o to unity to ensure a simple normalisation and then alter the strength of the boundary heating using the `SCAL` parameter in the inhomogeneous thermal boundary time-step programs. I have yet to impose a lateral thermal variation at the inner boundary, but designed **itfvf.f** with the possibility in mind.

The stand-alone source code version of the program is compiled by typing

```
make itfvf
```

within this directory. Once the executable is created, begin execution by typing

```
itfvf < inputfile
```

The inputs file must have the following format.

```
* input file for itfvf
root                                    : Output filename stem
boundary_coeffs                         : File containing coeff.s
 40  2   4   0.53846154   1.5384615  1  : nr isp ifrmf ri ro ithebc
 1.0     1.0                            : epsi  epso
```

Any line in the input file beginning with an asterisk, `*`, is ignored by the program and can thus be used to enter comments and notes.

The arguments are as follows

- `filename stem`: First characters in output files to be generated by current run. Running `itfvf` with the above input file will create the files `root.ints`, `root.vecs` and `root.xarr`

- `coefficient file name`. This file contains the specifications for the coefficients $g^{mc}_{\mathrm{i},l}$, $g^{ms}_{\mathrm{i},l}$, $g^{mc}_{\mathrm{o},l}$ and $g^{ms}_{\mathrm{o},l}$. Details regarding the form of this file are found below.

- `nr`: number of radial grid nodes to be used in the expression of our temperature function, $T_{\mathrm{a}}(r, \theta, \phi)$.

- `isp`: flag for the spacing of these grid nodes. `isp = 1` forces evenly spaced grid nodes from `ESNAAS` and `isp = 2` forces Chebyshev zero spaced nodes from `ZCPAAS` (see Figure 2).

- `ifrmf`: flag for order of elements in the resulting solution vector. `ifrmf` may take the values 3 and 4. For explanation, see Figure (1).

- `ri`: Radius of the inner boundary, r_{i}.

- `ro`: Radius of the outer boundary, r_{o}.

- `ithebc`: Flag for thermal boundary conditions at inner and outer boundaries.
  `ithebc = 1` means fixed temperature at both inner and outer boundaries. (In other words, we apply conditions 1 and 3.)
  `ithebc = 2` means fixed temperature at the inner boundary and fixed flux at the outer. (In other words, we apply conditions 1 and 4.)
  `ithebc = 3` means fixed flux at the inner boundary and fixed temperature at the outer. (In other words, we apply conditions 2 and 3.)

- `epsi`: ε_{i}.

- `epso`: ε_{o}.

Note that it is quite trivial to create a program which does not normalise the output, and uses the exact same coefficients in the input file.

1.1 Specifying coefficients for thermal heterogeneity

In the above input file, a filename is specified: `boundary_coeffs`. This file specifies all coefficients $g^{mc}_{\mathrm{i},l}$, $g^{ms}_{\mathrm{i},l}$, $g^{mc}_{\mathrm{o},l}$ and $g^{ms}_{\mathrm{o},l}$. Coefficients not listed are assumed to be zero.

This file contains one line for each coefficient. The first two characters of each line must contain either `IB` or `OB`: which specify either inner or outer boundary. These two characters must be exactly at the beginning of a line. The remaining part of the line contains, in any format, the numbers `L`, `M`, `ICS` and `COEF`.

`L` and `M` are clearly l and m: `ICS` is `1` to indicate a $\cos m\phi$ harmonic (i.e. $g^{mc}_{\mathrm{i},l}$ or $g^{mc}_{\mathrm{o},l}$) and `2` to indicate a $\sin m\phi$ harmonic (i.e. $g^{ms}_{\mathrm{i},l}$ or $g^{ms}_{\mathrm{o},l}$). `COEF` is ofcourse

just the value of the coefficient.

Examples

The file

```
OB   2   2   1      1.0
```

sets all coefficients to zero except for $g^{2c}_{\mathrm{o},2}$ which is set to unity.

The file

```
IB   2   2   1      1.0
OB   2   2   1      1.0
OB   3   2   2      1.0
```

sets $g^{2c}_{\mathrm{i},2} = 1.0$, $g^{2c}_{\mathrm{o},2} = 1.0$ and $g^{2s}_{\mathrm{o},3} = 1.0$. This fixes the contribution from $g^{2c}_{\mathrm{o},2}$ relative to that from $g^{2s}_{\mathrm{o},3}$, whereas the contribution from $g^{2c}_{\mathrm{i},2}$ is only fixed relative to the others by the parameters ε_{i} and ε_{o}.

1.2 Subprograms required for itfvf

SUBS subroutines

```
vecop.f fopen.f fclose.f esnaas.f zcpaas.f shcanc.f
ithcar.f itfa.f hmfwt.f svfwt.f xarrwt.f fnamer.f
lmfind.f itfcf.f cntric.f
```

SUBS integer function

```
indfun.f indshc.f
```

1.3 Run-time limitations

Several parameters are set at the outset which limit the physical size of the problem.

```
 INTEGER NRMAX, NHMAX, ISVMAX, LHMAX, LHLH2M, NDCS, NITHMX
 PARAMETER ( NRMAX = 100, LHMAX = 64, LHLH2M = LHMAX*(LHMAX+2),
1            NHMAX = LHLH2M+1, ISVMAX = NHMAX*NRMAX, NDCS = 1,
2            NITHMX = NHMAX )
```

If the values are insufficient, then change them and recompile. (Note that `NDCS` is not size dependent and should not be changed.)

- `NRMAX` is the maximum permitted number of radial grid nodes.
- `LHMAX` is the highest permitted spherical harmonic degree, l.

1.4 Sample runs of itfvf

The directory

`$LEOPACK_DIR/SAMPLERUNS/ITFVF`

contains example input files, coefficient files and model output. Do not under any circumstances edit these files, as these examples should serve as a control for the correct working of the code. After compiling the program, copy the `.input` and `.coeff` files to another directory, run the code and confirm that the output agrees with that in the directory.

1.4.1 Example a

We have a spherical shell with radius ratio $r_o/r_i = 0.35$. We want a fixed heat-flux with a Y_2^{2c} variation at the outer boundary only, 40 grid nodes with Chebyshev spacing and $\varepsilon = 1$.

```
itfvf < example_a.input
```

produces the three output files `example_aOUTPUT.ints`, `example_aOUTPUT.vecs` and `example_aOUTPUT.xarr`.

$$
(a)\quad
\begin{bmatrix}
f_1(r_1)\\ f_2(r_1)\\ \vdots\\ f_j(r_1)\\ \vdots\\ f_{N_h}(r_1)\\ \cdots\\
f_1(r_2)\\ f_2(r_2)\\ \vdots\\ f_j(r_2)\\ \vdots\\ f_{N_h}(r_2)\\ \cdots\\
\vdots\\
\cdots\\
f_1(r_i)\\ f_2(r_i)\\ \vdots\\ f_j(r_i)\\ \vdots\\ f_{N_h}(r_i)\\ \cdots\\
vdots\\
\cdots\\
f_1(r_{N_r})\\ f_2(r_{N_r})\\ \vdots\\ f_j(r_{N_r})\\ \vdots\\ f_{N_h}(r_{N_r})
\end{bmatrix}
\qquad
(b)\quad
\begin{bmatrix}
f_1(r_1)\\ f_1(r_2)\\ \vdots\\ f_1(r_i)\\ \vdots\\ f_1(r_{N_r})\\ \cdots\\
f_2(r_1)\\ f_2(r_2)\\ \vdots\\ f_2(r_i)\\ \vdots\\ f_2(r_{N_r})\\ \cdots\\
\vdots\\
\cdots\\
f_j(r_1)\\ f_j(r_2)\\ \vdots\\ f_j(r_i)\\ \vdots\\ f_j(r_{N_r})\\ \cdots\\
\vdots\\
\cdots\\
f_{N_h}(r_1)\\ f_{N_h}(r_2)\\ \vdots\\ f_{N_h}(r_i)\\ \vdots\\ f_{N_h}(r_{N_r})
\end{bmatrix}
$$

Figure 1: Alternative arrangements of elements within the solution vector. (a) shows **IFORMF = 3** and (b) shows **IFORMF = 4**.

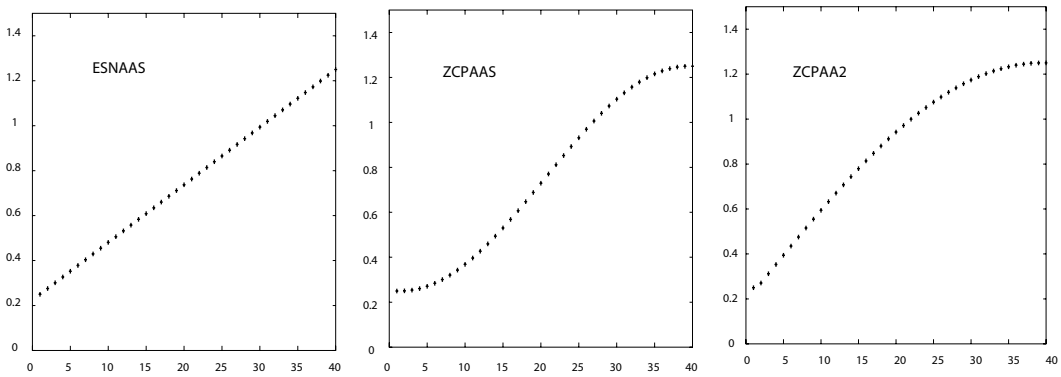


Figure 2: Distributions of radial grid nodes with $r_{\mathrm{i}} = 0.25$, $r_{\mathrm{o}} = 1.25$ and $N_r = 40$. Subroutine used to generate r values as shown.