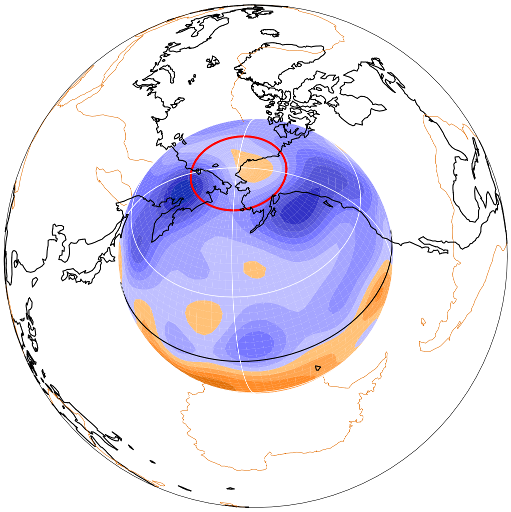
LEOPACK



msvip

Multiple **S**olution **V**ector **I**ngestion **P**rogram

Steven J. Gibbons, Oslo

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

1 msvip

Multiple **S**olution **V**ector **I**ngestion **P**rogram

Generates an arbitrary - and potentially zero solution vector in standard form (i.e. `.ints`, `.vecs` and `.xarr` files) which satisfies the requested characteristics. To this solution vector, we add scalar multiples of other solution vectors. It is essentially a simple adaption of `rsvfg`.

The primary purpose of this program was to produce single solution vectors from laterally heterogeneous heat-flux convection calculations for plotting. In the time-stepping codes, the homogeneous solution and heterogeneous temperature parts are kept in separate solution vector files: a single file is required for my plotting utilities.

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

```
make msvip
```

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

```
msvip < inputfile
```

The input file must have the following format

```
* input file for msvip
*
plotvec                                : File-name stem
 40     4      4                       : nr  iformf  nnds
 0.6666666 1.6666666           1  0.0  : ri   ro   insf   fac
 24     1     0     2     1     12     : lhv isv lhm ism minc mmax
 2           2                         : ivelbc ithebc
*
* We now add FAC2 times the following solution vector to
* FAC1 times the existing solution vector.
*
   0.0   0.3                           : FAC1  FAC2
Y22_flux_function.ints
Y22_flux_function.vecs
Y22_flux_function.xarr
*
* We now add FAC2 times the following solution vector to
* FAC1 times the existing solution vector.
*
   1.0   1.0                           : FAC1  FAC2
```

```
Pr0.6_cg200_ch1200_e0.30.ints
Pr0.6_cg200_ch1200_e0.30.vecs
Pr0.6_cg200_ch1200_e0.30.xarr
*
```

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 fixed set of arguments is as follows

- `filename stem`: First characters in output files to be generated by current run. Running `msvip` with the above input file will create the files `root.ints`, `root.vecs` and `root.xarr`.
- `nr`: number of radial grid nodes requested.
- `iformf`: Flag which chooses the order in which elements in the solution vector are stored. `iformf` can take the values `3` or `4`.
- `ri`: Radius of the inner boundary, r_i.
- `ro`: Radius of the outer boundary, r_o.
- `insf`: Radial grid node spacings flag.
  `insf = 1` forces evenly spaced grid nodes from `ESNAAS` and `insf = 2` forces Chebyshev zero spaced nodes from `ZCPAAS`
- `fac`: Scales the random initial solution. It is probably best to begin with "numerical noise" and hence a low value of `fac`.
- `lhv`: Highest spherical harmonic degree, l requested for velocity and temperature
- `isv`: Symmetry flag for velocity.
  1. Velocity is equatorially symmetric (ES)
  2. Velocity is equatorially anti-symmetric (EA)
  3. Velocity contains both ES and EA parts
- `lhv`: Highest spherical harmonic degree, l requested for magnetic field
- `isv`: Symmetry flag for velocity.
  1. Magnetic field is equatorially symmetric (ES)
  2. Magnetic field is equatorially anti-symmetric (EA)

3. Magnetic field contains both ES and EA parts

- `minc`: Increment of wavenumber.
- `mmax`: Maximum wavenumber. For example, `minc = 2` and `mmax = 6` will include the wavenumbers $\{0, 2, 4, 6\}$.
- `ivelbc`: Velocity boundary condition:
  1. Rigid (no-slip) boundary condition
  2. Stress-free (free-slip) boundary condition
- `ithebc`: Temperature boundary condition:
  1. Fixed temperature at inner and outer boundaries
  2. Fixed temperature at inner boundary and fixed heat-flux at outer boundary
  3. Fixed heat-flux at inner boundary and fixed temperature at outer boundary

After this, arguments are entered in sets of four uncommented lines. The first of these lines contains two real numbers; `FAC1` and `FAC2`. This adds `FAC2` multiplied by the solution vector described the following three lines to `FAC1` multiplied by the solution vector which is currently stored. In the example above, we first zero the new vector (`FAC1 = 0.0`) and then add `0.3` multiplied by the solution vector specified by the files

```
Y22_flux_function.ints
Y22_flux_function.vecs
Y22_flux_function.xarr
```

1.1 Subprograms required for msvip

SUBS subroutines

```
esnaas.f zcpaas.f hminda.f shvecf.f bihfrd.f svfrd.f
xarrrd.f assvar.f hmfwt.f svfwt.f xarrwt.f fopen.f
fclose.f svrint.f smitsl.f pmfabf.f fnamer.f gfdcfd.f
mnewtr.f matop.f
```

SUBS double precision function

```
pvnsrf.f tvsfrf.f pvsfrf.f pmfirf.f chorch.f chorcf.f
chorsf.f pmfsef.f
```

SUBS integer function

```
indfun.f
```

BLAS integer function

```
idamax.f
```

BLAS subroutines

```
dgemv.f dgemm.f dtrsm.f dswap.f dger.f  dscal.f
dtrmm.f dtrmv.f
```

LAPACK subroutines

```
xerbla.f dgetrf.f dgetri.f dgetrs.f dgetf2.f dlaswp.f
dtrtri.f dtrti2.f
```

LAPACK integer function

```
ilaenv.f
```

LAPACK logical function

```
lsame.f
```

1.2 Run-time limitations

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

```
 INTEGER NRMAX, NHMAX, LHMAX, NDCS, ISVMAX
 PARAMETER ( NRMAX = 300, NHMAX = 6000, LHMAX = 64,
1            NDCS  = LHMAX + 6, ISVMAX = NRMAX*NHMAX )
```

If the values are insufficient, then change them and recompile.

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

1.3 Outputs from MSVIP

If the filename stem "root" was specified in the input file, the files `root.ints`, `root.vecs` and `root.xarr` will all be created.

1.4 Sample runs of msvip

1.4.1 Example a

The files Pr0.6_cg200_ch1200_e0.30.* describe the homogeneous boundary part of a non-magnetic convection solution which is subjected to a laterally varying heat-flux at the outer boundary. The heat-flux heterogeneity is imposed by the temperature vector stored within the files Y22_flux_function.*. The input file example_a.input causes msvip to produce a single solution vector, described by the files example_aOUTPUT.*. This comprises of the full time-dependent part of the convection solution with 30% of the temperature heterogeneity solution added - the scaling which was applied in the o2ibtctsc3 program used to calculate the solution.