OpenExtrap 1.1

Jan W. Thorbecke

January 27, 2022

E-mail: janth@xs4all.nl
http://www.xs4all.nl/∼janth/

Contents

1 Introduction

The OpenExtrap software contains programs which makes use of a recursive extrapolation algorithm in the space-frequency domain. The total package consists of 5 related programs. Each program gets its parameters from the command-line according to the SU convention. Not all parameters have to be specified. For most parameters the default values should be sufficient. All parameters are explained in separate sections, discussing the individual programs.
The extrapolation in the space-frequency domain is performed with optimized spatial convolution operators. Default are the Weighted Least Squares operators (WLSQ) which are described in Thorbecke (2004). Note that the optimized extrapolation operators are the columns(W^+) and rows(W^-) of the W matrices in the $W^-R^+W^+$ model.
The programs in the package all have the same structure. This structure is also reflected in the filenames of the different functions. In the initial stage of the program the data to be extrapolated (wave field measurements) and the gridded medium files (velocity and density) are read in . From the velocity file the minimum and maximum velocity is determined and together with the desired minimum and maximum frequency an operator table is calculated in advance. The operator table is stored in memory as an static array (see the **tablecalc...c** functions).
The **xw...c** functions perform the extrapolation and calculate the specific results. In these **xw...c** functions the data is transformed to the space-frequency domain and for every lateral position, for all depth steps and for all frequencies, a convolution is carried out with the optimized convolution operator. Note that a new convolution operator is read from the table if the wavenumber k changes, so if the velocity or frequency changes. After the extrapolation has finisched the calculated result is transformed to the space-time domain and the function returns to the main program.
In the sub-directory *main* the following programs can be found:

- **extrap** - extrapolation through a gridded subsurface model.
- **cfpmod** - modeling one-way travel times for CFP operator generation.
- **migr** - shot record migration using optimized extrapolation operators.
- **migr_mpi** - parallel shot record migration using optimized extrapolation operators.
- **opercalc** - calculates extrapolation operators for a given frequency.
- **onewvsp** - VSP generation with one-way extrapolation operators.

In the sub-directory *lib* there are two types of functions defined; functions which are related to the calculation of different optimized convolution operators, and functions which are related to the calculation of the spatial convolution.

operator	GaussWindow, KaiserWindow, forwExtr, invExtr, kxwfilter, optRemez, remez, shortoper, spline3, tablecalc_opt, trunc1D, weightfunct, toeplitz, findBestOper
convolution	xwMigrOpl, xwBeam, xwCFP, xwExtr, xwSnap, xwVSP, extrapEdge, xwZoMigr, xwExtrG, kwZoMigr, kwExtr, kwMigr
from SU	atopkge, docpkge, getpars
misc	calc_vz, getreccfp, getrecvsp, minmax, srcarray, verbosepkg, getFileInfo, getModelInfo, readData, writeData, wallclock_time

The spatial convolution (for every frequency) is implemented in an efficient way by making use of the symmetry in the convolution operator. On a vector machine this code should work very well. It may be usefull to use these subroutines in other programs where an extrapolation is needed.
Further details on the individual programs are documented in the next sections. All programs are selfdocumented.
The demo directory contains a gridded velocity file `syncline_cp.su` shown and in Figure 1 and a Ricker wavelet shown in Figure 2.

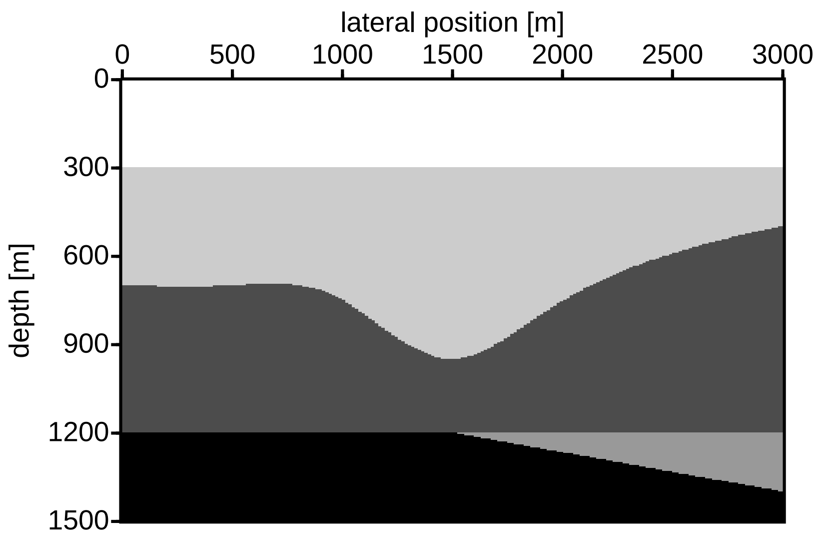


Figure 1: Syncline model which is used in the demo directory to illustrate the working of the programs.

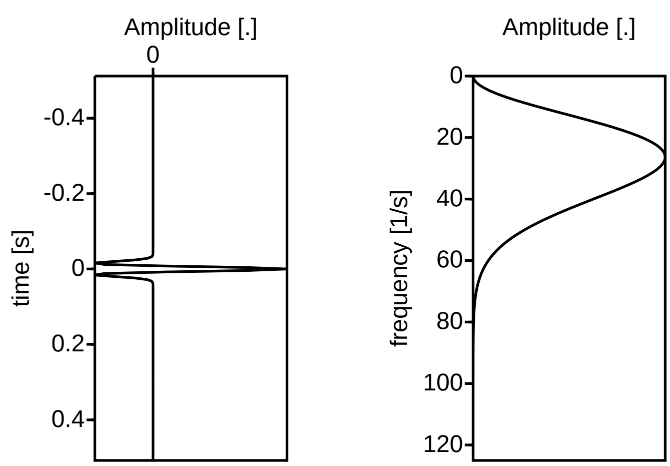


Figure 2: Ricker wavelet(left) and amplitude spectrum(right) which is used in the modeling experiments of the demo directoty.

2 migr / migr_mpi

2.1 General

Shot record migration scheme based on optimized x-w wave field extrapolation operators. There is also a parallel version of `migr` called `migr_mpi`. The parallelisation is based on MPI and the number of shot gathers to be migrated is divided over the number of available working processors. For reading the shot records the master processors reads all the shots and sent the data to the working processors. If a working processor has finished its migration the image (and eventually extrapolated source and receiver wavefields) are communicated back to the master processor, who will write the data to output file(s).

2.2 Parameters

Via the command-line or in a parameter file: `par=<parameter_file>`.

```
MIGR - pre-stack depth migration (x-w).

migr file_shot= file_vel= [optional parameters]

Required parameters:

  file_shot= ............... input data to be migrated
  file_vel= ................ gridded velocity file for receiver field
  file_vels=file_vel ....... gridded velocity file for source field

Optional parameters:

  conjg=0 .................. 1: take complex conjugate of input data
  key=sx ................... input data sorting key for receiver field
  nxmax=512 ................ maximum number of traces in input file
  ntmax=1024 ............... maximum number of samples/trace in input file
MIGRATION
  imc=1 .................... image condition (*)
  ndepth=all ............... number of depth steps
  zrcv=oz .................. receiver depth level
  ixa=tan(alpha)*ndepth*dz . number of traces after acquisition aperture
  ixb=ixa .................. number of traces before acquisition aperture
  ntap=0 ................... number of taper points at boundaries
  eps_a=0.0 ................ absolute stabilization factor for imc=[1,2]
  eps_r=0.001 .............. relative stabilization factor for imc=[1,2]
  domain=1 ................. 1: x-w lateral variant convolution, 0: kx-w
  zomigr=0 ................. 1: zero-offset migration (=> velocity *= 0.5)
           ................. 2: zero-offset migration (=> velocity *= 1.0)
SOURCE DEFINITION
  file_src=<file_name> ..... (areal)wavelet used
  key_src=fldr ............. input data sorting key for source field
  fmin=0 ................... minimum frequency
  fmax=70 .................. maximum frequency
  conjgs=0 ................. 1: take complex conjugate of source wavefield
  selev=0 .................. 0: ignore headers for source/receiver depth
EXTRAPOLATION OPERATOR DEFINITION
  select=10 ................ type of x-w operator (*)
  opl=25 ................... length of the convolution operator (odd)
  alpha=65 ................. maximum angle of interest
  perc=0.15 ................ smoothness of filter edge
  weight=5e-5 .............. weight factor in WLSQ operator calculation
  beta=3 ................... 2 < beta < 10; factor for KAISER window
  fine=10 .................. fine sampling in operator table
  filter=1 ................. apply kx-w filter to desired operator
```

```
  limit=1.0002.............. maximum amplitude in best operators
  opl_min=15 ............... minimum length of convolution operator
 OUTPUT DEFINITION
  file_image= .............. output file with migrated result
  writeafter=10 ............ writes image/shots after # processed shots
  file_ishot=NULL .......... output file for migrated shot-records
  writeshots=0 ............. 1; writes migrated shot record
  writeinc=1 ............... trace increment of file_ishots
  verbose=0 ................ =1: shows various parameters and results
  sx_file= ................. file with extrapolated source field
  rx_file= ................. file with extrapolated receivers
  depthex= ................. depth to save extrapolated fields (m)

  Options for select:
        - 0 = Truncated operator
        - 1 = Gaussian tapered operator
        - 2 = Kaiser tapered operator
        - 3 = Smoothed Phase operator
        - 4 = Weighted Least Squares operator
        - 5 = Remez exchange operator
        - 8 = Smooth Weighted Least Squares operator (careful if dz<0.5*dx)
        - 9 = Optimum Smooth Weighted Least Squares operator
        - 10= Optimum Weighted Least Squares operator (Default)
  Imaging condition:
        - 0 = correlation
        - 1 = stabilized inversion
        - 2 = stabilized Least Squares
        - 4 = 2*data.r for zero-offset migration only
        - 5 = smoothed imaging P265 EAGE 2006: A. Guitton

 The shot and receiver positions in the model are determined by
 the hdr values gx and sx. The data from file_shot is extrapolated
 backward, the data from file_src is extrapolated forward.
 If file_src is not set a spike is taken, if file_src=pipe read from stdin
 Note that with the conjg and conjgs options the extrapolation
 direction can be changed.

 Copyright 1997, 2008 Jan Thorbecke, (janth@xs4all.nl)
```

2.3 General parameter description

The shots to be migrated (`file_shot`) and the gridded subsurface files (`file_vel` and `file_vels`) should have the same lateral extend being defined by the `gx` headers. The position of the receivers of the data in the subsurface grid is done by means of the `gx` header value of the velocity model corresponding to the `gx` header value in the data to be extrapolated. The distance between the traces in the velocity model should be smaller or equal to the distance between the receivers/shots. The program assigns the `gx` value of the receivers to the nearest grid point in the velocity model. The number of depth steps is controlled with the parameter `ndepth=`. To avoid reflections at the edges of the model the parameter `ntap` can be set. `ntap` indicates the number of points at the edges for which a spatial taper is designed according to: $\exp\left(-(0.4 * (ntap - ix)/ntap)^2\right)$. Choosing `ntap` equal to half of the operator length is an optimum value.
The `sx` or `fldr` headers in `file_shot` determine a single shor record. As long as `sx` or `fldr` header value remain constant the program considers these traces belonging to a single shot. If one of these values changes this trace is then considered to belong to the next shot. Be careful for example with common-offset data. In this kind of data set the `sx` or `fldr` header value usually change with each trace, hence each trace is considered to be one shot by the program.

To overcome this problem set **both** `sx` or `fldr` header values to a constant value
Topography is taken into account by using a velocity model which has zero velocities above the defined topography. In that case the position of the source and receivers is lowered into the velocity model until a non-zero velocity is found. From that depth the extrapolation of that point is started.
The parameter `file_src` describes the source wavelet. If `file_src` is not defined a band-limited spike is assumed. If `file_src` contains only one trace it is assumed that this trace is the wavelet used for all shots to be migrated. If `file_src` contains more than one trace it is interpreted as an areal shot record and areal short-record migration is carried out.
The parameters `ixa` and `ixb` determine the number of traces to be included in the calculation of the image gather. When they are not set the source and receiver fields are extrapolated only on the lateral grid-points that are within the receiver array. By setting the parameter `ixa=number-of-gridpoints` (e.g 200) then the extrapolated filed can extrapolate 200 grid points before and after the receiver array. This gives a larger imaging aperture. `ixa` defines the number of traces to include in the calculation with a lateral position greater than the lateral extend (max value of `sx,gx`) of the shot gather (`a` from after). `ixb` defines the number of traces to include in the calculation with a lateral shot position smaller than the lateral extend (max value of `sx,gx`) of the shot gather (`b` from before).
For a more detailed discussion on the different parameters which are related to the extrapolation operator optimization the reader is referred to the description of the program **opercalc**. The WLSQ operators are described in Thorbecke et al. (2004).
A rule of thumb for determinig the grid distances in a gridded model for extrapolating wavefields with the one-way extrapolation operators: The spatial extrapolation operators work best when `dx=receiver-distance` and $dz \approx 0.5dx$. Making dz smaller than dx helps to support higher propagation angles and usually gives a more efficient operator. When $dx = dz$ you need more x-points to cover the same angle range as. Using $dz = 0.5 * dx$ requires 2 times more dz steps, but used as more stable operator. In most cases the choice of $dz \approx 0.5 * dx$ turns out the be an efficient one. Making dz much smaller than ($\approx 0.1 * dx$) does not improve the higher propagation angle that much anymore while the number of dz steps to propagate the same depth range is increasing a lot.

2.4 Examples

Generating a pulse repsonse through the medium of Figure 1.

```
migr file_shot=ricker_shift.su file_vel=syncline_cp.su zomigr=1 file_image=migr8.su verbose=1 ixa=301 select=8
suximage < migr8.su
migr file_shot=ricker_shift.su file_vel=syncline_cp.su zomigr=1 file_image=migr10.su verbose=1 ixa=301 select=10
suximage < migr10.su
```

2.5 To do

3D extension is available on request.

2.6 References

Thorbecke, J., Wapenaar, K., and Swinnen, G., 2004, Design of one way wavefield extrapolation operators, using smooth functions in WLSQ optimization.: **Geophysics**, pages 1037–1045.

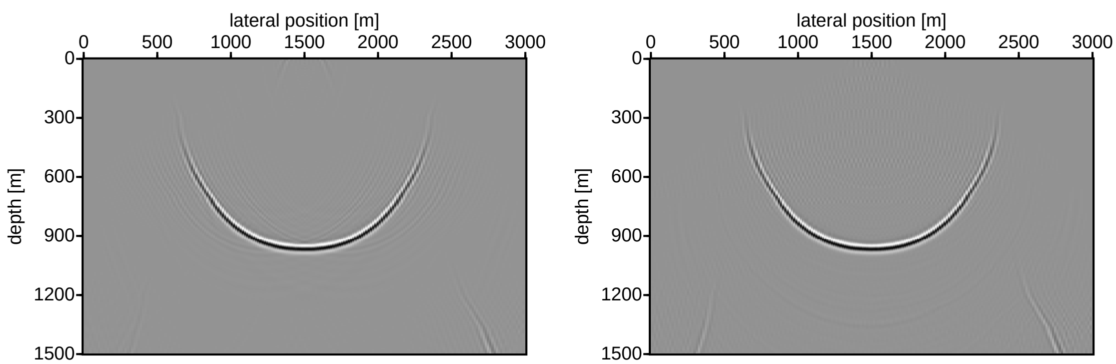


Figure 3: Pulse responses of the migration program for two different operators. Left show the smooth WSLQ operator and right shows the best selected WLSQ operator.

3 extrap

3.1 General

Extrap extrapolates (forward or inverse) an input wave-field through a gridded subsurface model. The output is the extrapolated wave-field at the desired depth, which is controlled by the parameter `zrcv=`. As a special option snapshots `snap=1` and/or beams `beam=1` can be calculated. The snapshots option selects at every depth step the samples at the desired snapshot times. The snapshot option can be used to see how a certain wavefield propagates through a medium. The beam option gives insight how the energy of the wavefield propagates through the medium.
For most velocity distributions the extrapolation works fine and gives good results. However for large depth steps (defined through the velocity model) and laterally very strong variations the program may give inferior result due to the assumption that within the length of the convolution operator the velocity is assumed to be homogeneous (velocity is taken at the mid-point of the operator).

3.2 Parameters

Via the command-line or in a parameter file: `par=<parameter_file>`.

```
extrap - forward or inverse extrapolation (x-w)

extrap file_in= file_vel= file_out= [optional parameters]

Required parameters:

  file_in= ................. Input file to be extrapolated
  file_vel= ................ gridded velocity file
  file_out= ................ output file with extrapolated result

Optional parameters:

  fmin=0 ................... minimum frequency
  fmax=70 .................. maximum frequency
  mode=1 ................... type of extrapolation (1=forward, -1=inverse)
  conjg=0 .................. take complex conjugate of input data
  nxmax=512 ................ maximum number of traces in input file
  ntmax=1024 ............... maximum number of samples/trace in input file
  zstart=0 ................. depth to start extrapolation
RECEIVER POSITIONS
  xrcv1=ox ................. x-position of the receiver (m)
  xrcv2=ox+(nx-1)*dx ....... x-position of last receiver
  dxrcv=dx ................. step in receiver x-direction
  zrcv1=oz+(nz-1)*dz ....... z-position of the receiver (m)
  zrcv2=zrcv1 .............. z-position of last receiver
  dzrcv=0 .................. step in receiver z-direction
  xrcv= .................... x-position's of receivers (array)
  zrcv=(nz-1)*dz ........... z-position of the receivers (last depth level)
  lint=1 ................... linear interpolate between the rcv points
  file_int= ................ input file describing the interfaces (makemod)
  boundary=1 ............... boundary to place the receivers(overrules zrcv)
EXTRAPOLATION OPERATOR DEFINITION
  domain=0 ................. 0: x-w, 1: kx-w operator
  select=4 ................. type of x-w operator
  opl=25 ................... length of the convolution operator (odd)
  alpha=65 ................. maximum angle of interest
  perc=0.15 ................ smoothness of filter edge
  weight=5e-5 .............. weight factor in WLSQ operator calculation
```

```
  fine=10 .................. fine sampling in operator table
  filter=1 ................. apply kx-w filter to desired operator
  ntap=0 ................... number of taper points at boundaries
  limit=1.0002.............. maximum amplitude in best operators
  opl_min=15 ............... minimum length of convolution operator
SNAPSHOTS DEFINITION (if snap=1)
  tsnap1=-nt*dt/2........... first snapshot time (s)
  tsnap2=nt*dt/2 ........... last snapshot time (s)
  dtsnap=25*dt ............. snapshot time interval (s)
  reverse=0 ................ extrapolate from deepest level back to surface
OUTPUT
  snap=0 ................... snapshots
  beam=0 ................... beams
  verbose=0 ................ silent option; >0 display info

  Options for select:
        - 0 = Truncated operator
        - 1 = Gaussian tapered operator
        - 2 = Kaiser tapered operator
        - 3 = Smoothed Phase operator
        - 4 = Weighted Least Squares operator
        - 5 = Remez exchange operator
        - 8 = Smooth Weighted Least Squares operator
        - 9 = Optimum Smooth Weighted Least Squares operator
        - 10= Optimum Weighted Least Squares operator

 Copyright 1997, 2008 Jan Thorbecke, (janth@xs4all.nl)
```

3.3 General parameter description

The data to be extrapolated (`file_in`) and the gridded subsurface (`file_vel`) should at least have the same number of traces. If the number of traces of the subsurface is smaller an error message is the result. If the number of traces is bigger then the user should position the first receiver of the data in the subsurface grid. This is done by setting the `gx` header value of the velocty model corresponding to the `gx` header value in the data to be extrapolated. The distance between the traces in both gathers should be equal. The number of depth steps is controlled with the parameter `zrcv=` and the extrapolation direction is controlled with `mode=`. For inverse extrapolation the complex conjugate of the forward extrapolation operator is taken. The parameter `reverse` extrapolated the data from the deepest level in the velocity model to the lowest level.

Topography is taken into account by using a velocity model which has zero velocities above the defined topography. In that case the position of the source and receivers is lowered into the velocity model until a non-zero velocity is found. From that depth the extrapolation of that point is started.

To avoid reflections at the edges of the model the parameter `ntap` can be set. `ntap` indicates the number of points at the edges for which a spatial taper is designed according to: $\exp\left(-(0.4 * (ntap - ix)/ntap)^2\right)$. Choosing `ntap` equal to half of the operator length is an optimimum value.

The parameter `snap=1` gives at the defined snapshot times (with `tsnap1, tsnap2` and `dtsnap`) the extrapolated wave-field. With this option the propagation of the wavefield through the model can be monitored.

The parameter `beam=1` gives the energy of the extrapolated wave-field at all calculated depth steps. The energy is calculated in the frequency domain for all depth steps according to $E(x,d) = \frac{1}{nfreq}\sum_\omega \|data(x,\omega)\|^{\frac{1}{2}}$. With this option the propagation of the energy through the model can

be monitored.
The receiver spread is defined by coordinates in x (`xrcv, dxrcv`) and z (`zrcv dzrcv`). The parameters `xrcv` and `zrcv` are defined as arrays which interpretation depends on the parameter `lint`. For example if `xrcv=0,3000`, `zrcv=0,0`, `dxrcv=15`, `dzrcv=0` and `lint=1` then between the points (0,0) and (3000,0) the defined receiver positions are calculated by a linear interpolation between the two points with dx=15, which results in an receiver array of 201 receivers ranging from (0,0) to (3000,0). However, if `lint=0` the receivers are only defined at the points (0,0) and (3000,0). One can also use the paremeters `xrcv1, xrcv2` and `zrcv1, zrcv2` to define receiver arrays.
For a more detailed discussion on the different parameters which are related to the extrapolation operator optimization the reader is referred to the description of the program **opercalc**. The WLSQ operators are described in Thorbecke et al. (2004).

3.4 Examples

In the following example a Green's function in a medium is calculated which gives the data we want to extrapolate. By choosing in **extrap** the options `beam=1` and `conjg=1`, the calculated output shows how the energy is focused to depth position 1000 (the source depth of the input file) and gets defocused again for deeper depth positions. The options `snap=1` and `conjg=1` show snapshots of the wavefield propagating through the model.

```
cfpmod file_vel=syncline_cp.su xsrc1=1500 zsrc1=1200 ntap=30 file_src=ricker.su file_out=green.su

extrap file_in=green.su file_vel=syncline_cp.su verbose=1 beam=1 conjg=1 | suximage

extrap file_in=green.su file_vel=syncline_cp.su verbose=1 snap=1 conjg=1 \
tsnap1=-0.512 dtsnap=0.128 | suximage
```

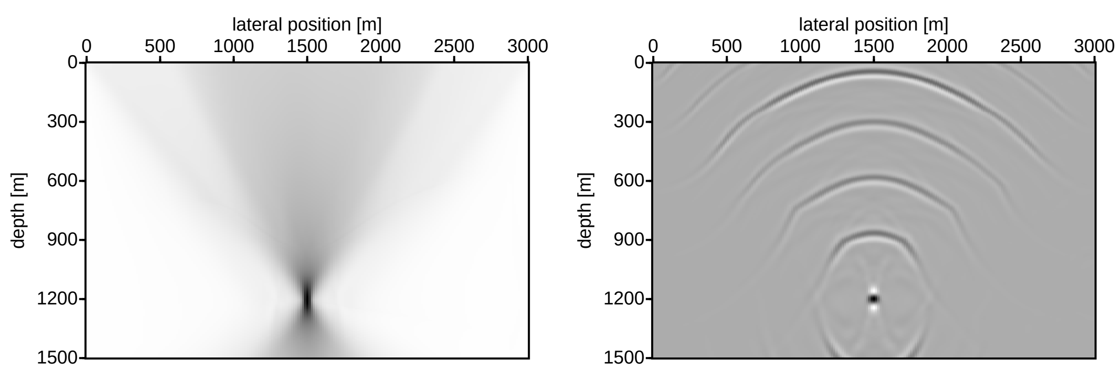


Figure 4: Inverse extrapolated results of Green's function placed in the middle of the model of Figure 1 at z=1200 m. The left picture shows the energy beam of the extrapolated wavefield and the right picture show snapshots of the converging wavefield.

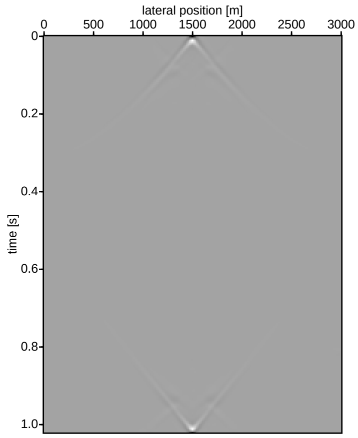
For forward extrapolation of the data through the model the following command can be given (to let the extrapolation stop at a certain depth level use the parameter `zrcv=`):

```
extrap file_in=green.su file_vel=syncline_cp.su verbose=1 mode=-1 zrcv=1200 | suximage
```

Now with cfpmod from surface to depth and then with extrap from depth back to surface:

```
cfpmod file_vel=syncline_cp.su xsrc1=1500 zsrc1=0 zrcv=1200 ntap=30 file_src=ricker.su file_out=deep.su
```

```
extrap file_in=deep.su file_vel=syncline_cp.su verbose=1 zrcv=0 zstart=1200 reverse=1 mode=-1 verbose=1 | suximag
```



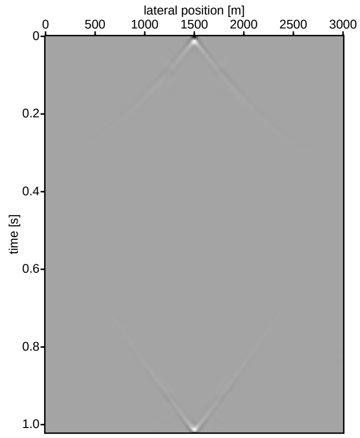


Figure 5: Inverse extrapolated wavefields of Green's function placed in the middle of the model of Figure 1 at z=1200 m. (left) the right picture show the result of a source at the surface and receivers at z=1200 m.

3.5 To do

Extension of the operator optimization algorithms to 3D media, this extension is already available in a non-official release version and can be obtained on request.

3.6 References

Thorbecke, J., Wapenaar, K., and Swinnen, G., 2004, Design of one way wavefield extrapolation operators, using smooth functions in WLSQ optimization.: **Geophysics**, pages 1037–1045.

4 cfpmod

4.1 General

Based on the same algorithm as used in the program **extrap**, **cfpmod** calculates pulse responses of sources which are defined in the subsurface. This program is therefore very usefull for the generation of CFP operators.

4.2 Parameters

Via the command-line or in a parameter file: `par=<parameter_file>`.

```
cfpmod - modeling one-way travel times in x-w domain

cfpmod file_vel= xsrc1= zsrc1= [optional parameters]

Required parameters:

  file_vel= ................ gridded velocity file
  xsrc1= ................... x-position of the source (m)
  zsrc1= ................... z-position of the source (m)

Optional parameters:

  file_out= ................ output file with traveltimes
  file_int= ................ input file describing the interfaces (makemod)
  mode=1 ................... type of extrapolation (1=forward, -1=inverse)
  ntap=0 ................... number of taper points at boundaries
  n2max=512 ................ maximum number of traces in input file
  n1max=1024 ............... maximum number of samples/trace in input file
SOURCE POSITIONS
  xsrc2=xsrc1 .............. x-position of last source
  dxsrc=0 .................. step in source x-direction
  zsrc2=zsrc1 .............. z-position of last source
  dzsrc=0 .................. step in source z-direction
  boundary=0 ............... boundary to place the sources (overrules zsrc)
RECEIVER POSITIONS
  xrcv1=ox ................. x-position of the receiver (m)
  xrcv2=ox+(nx-1)*dx ....... x-position of last receiver
  dxrcv=dx ................. step in receiver x-direction
  zrcv1=oz ................. z-position of the receiver (m)
  zrcv2=zrcv1 .............. z-position of last receiver
  dzrcv=0 .................. step in receiver z-direction
  xrcv= .................... x-position's of receivers (array)
  dxspr=0 .................. step of receiver spread in x-direction
  zrcv=0 ................... z-position of the receivers (first depth level)
  lint=1 ................... linear interpolate between the rcv points
SAMPLING AND SOURCE DEFINITION
  file_src=<file_name> ..... wavelet in time used (overrules dt)
  file_amp=<file_name> ..... wavelet in lateral direction
  wnx=1 .................... number of lateral wavelet samples
  dt=0.004 ................. stepsize in time-direction
  nt=256 ................... number of time samples
  fmin=0 ................... minimum frequency
  fmax=70 .................. maximum frequency
  add=0 .................... 1: adds all defined sources
PLANE WAVE AREAL SHOT RECORD DEFINITION (only calculated if Na != 0)
  amin=-65 ................. minimum angle of plane wave illumination
  amax=-amin ............... maximum angle of plane wave illumination
  Na=0 ..................... number of plane waves between amin and amax
```

```
Note that the plane waves cannot be added together by using add=1
EXTRAPOLATION OPERATOR DEFINITION
  select=4 ................ type of x-w operator
  opl=25 .................. length of the convolution operator (odd)
  alpha=65 ................ maximum angle of interest
  perc=0.15 ............... smoothness of filter edge
  weight=5e-5 ............. weight factor in WLSQ operator calculation
  fine=10 ................. fine sampling in operator table
  filter=1 ................ apply kx-w filter to desired operator
  limit=1.0002............. maximum amplitude in best operators
  opl_min=15 .............. minimum length of convolution operator
OUTPUT
  beam=0 .................. 1 beams, 2 add all beams for all defined shots
  verbose=0 ............... silent option; >0 display info

  Options for select:
        - 0 = Truncated operator
        - 1 = Gaussian tapered operator
        - 2 = Kaiser tapered operator
        - 3 = Smoothed Phase operator
        - 4 = Weighted Least Squares operator
        - 5 = Remez exchange operator
        - 8 = Smooth Weighted Least Squares operator
        - 9 = Optimum Smooth Weighted Least Squares operator
        - 10= Optimum Weighted Least Squares operator

 Copyright 1997, 2008 Jan Thorbecke, (janth@xs4all.nl)
```

4.3 General parameter description

The gridded subsurface (`file_vel`) which is needed in the caclulation can be made with for example the DELPHI program **makemod**. `file_int` is defined by the same program.
The source position(s) are defined by coordinates in x(`xsrc1, xsrc2, dxsrc`) and z (`zsrc1, zsrc2, dzsrc`). If `file_int` is defined then it is also possible to use the parameter `boundary` instead of the z parameters. The interface file (`file_int`) is used to place the source at the defined boundary (at the defined x position). Every source postition defines one output gather. However if `add=1` the defined source postions are combined to a planar source and only one output gather is calculated.
The receiver spread is defined by coordinates in x (`xrcv, dxrcv`) and z (`zrcv dzrcv`). The parameters `xrcv` and `zrcv` are defined as arrays which interpretation depends on the parameter `lint`. For example if `xrcv=0,3000`, `zrcv=0,0`, `dxrcv=15`, `dzrcv=0` and `lint=1` then between the points (0,0) and (3000,0) the defined receiver positions are calculated by a linear interpolation between the two points with dx=15, which results in an receiver array of 201 receivers ranging from (0,0) to (3000,0). However, if `lint=0` the receivers are only defined at the points (0,0) and (3000,0). One can also use the paremeters `xrcv1, xrcv2` and `zrcv1, zrcv2` to define receiver arrays.
Once the spread for the first shot position is defined, the parameter `dxspr` define the movement of the spread for the next shot. Choosing `dxspr=0` defines a fixed spread modeling.
The parameter `file_src` defines a source wavelet, however, if this parameter is not defined a wavelet with a flat spectrum and zero phase is used in the program. The sources can be extended in the lateral direction by using the parameter `wnx` or `file_amp`. If there are more source positions defined (with `zsrc1` $\neq$ `zsrc2` and `dzsrc > 0`) the output consists of a number of shot gathers. The source position in the gridded model is chosen at the nearest grid position.
Plane waves are defined by using the parameters `Na`, `amin` and `amax`. Which define the number of plane waves and the minimum and maximum angle respectively.

For a more detailed discussion on the different parameters which are related to the operator optimization the reader is referred to the description of the program **opercalc**.

4.4 Examples

To run the program a gridded velocity file has to be defined. We use the gridded velocity file (synclin_cp.su, see Figure 1) which can be found in the demo directory, and a ricker wavelet (ricker.su, see Figure 2), which can also be found in the demo directory.
In this gridded subsurface file we want to place a source at depth 1000 m, at x-position 500 and model the response at the surface. The result is shown in Figure 6 and the command to generate this results is:

```
../bin/cfpmod file_vel=syncline_cp.su xsrc1=1000 zsrc1=500 ntap=30 file_src=ricker.su | suximage
```

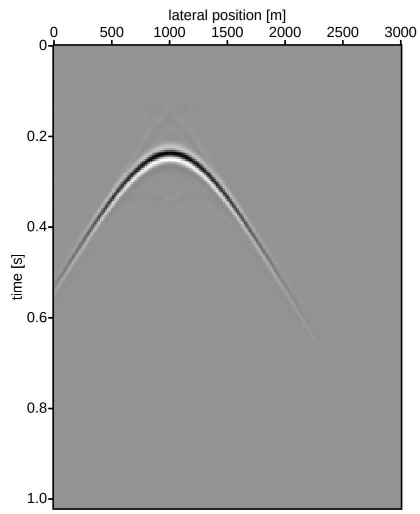


Figure 6: Response of a point source at x=1000 and z=500 m., measured at the surface z=0.

To calculate 5 plane wave responses, with 5 different angles ranging from -20 to 20, from the subsurface the following command can be used:

```
../bin/cfpmod file_vel=syncline_cp.su xsrc1=1000 zsrc1=1200 ntap=30  \
    file_src=ricker.su Na=5 amin=-20 verbose=1 | suximage
```

To model more than one source position use:

```
../bin/cfpmod file_vel=syncline_cp.su xsrc1=300 xsrc2=2700 \
    zsrc1=1200 zsrc2=1200 dxsrc=300 file_src=ricker.su \
    ntap=30 verbose=1 | suximage
```

and add those together (and do only one modeling step) do

```
../bin/cfpmod file_vel=syncline_cp.su xsrc1=300 xsrc2=2700 \
    zsrc1=1200 zsrc2=1200 dxsrc=300 file_src=ricker.su \
    ntap=30 add=1 verbose=1 | suximage
```

4.5 To do

Using the local reflectivity function (or a source distribution) instead of a pulse.
Better source position definition if the source does not lie on a point defined by the subsurface grid (which is an implementation of a local extrapolation).

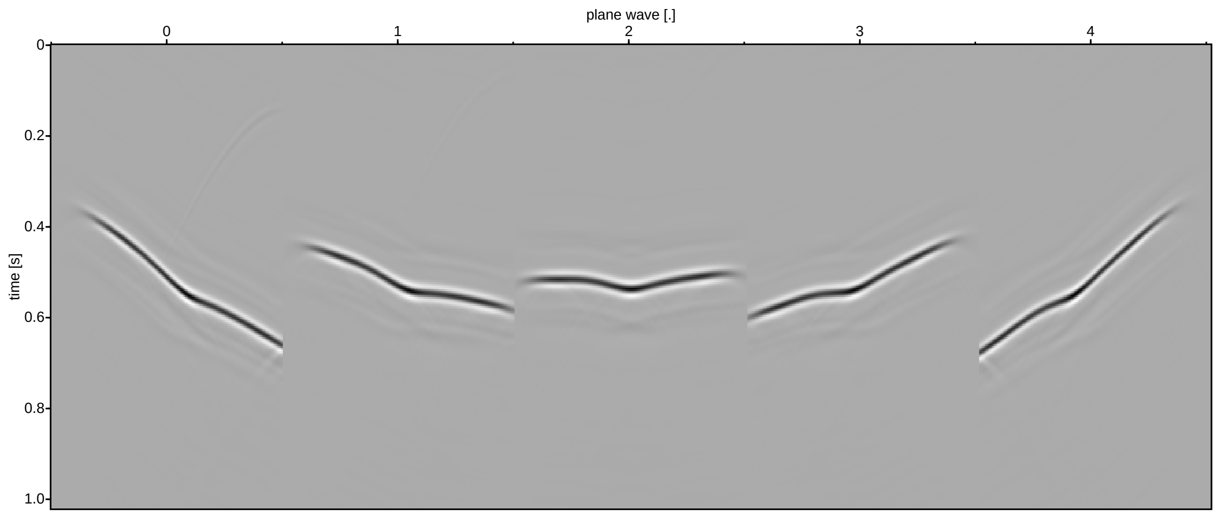


Figure 7: Response of 5 plane waves at z=1200 and x=1500 with angles ranging from -20 to 20.

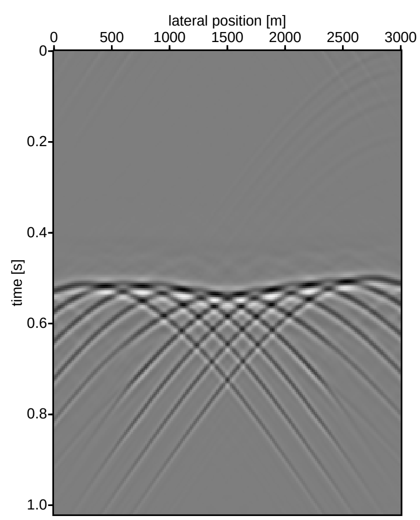


Figure 8: Response of several point sources position at x=[300,2700] and z=1200 m., measured at the surface z=0.

5 onewvsp

5.1 General

onewvsp is based on the same algorithm as used in **extrap**. **onewvsp** can be used to generate pseudo VSP data from seismic surface data. For the extrapolation of the data one-way extrapolation operators (these operators can be calculated with **opercalc**) are used.

5.2 Parameters

Via the command-line or in a parameter file: `par=<parameter_file>`.

```
onewvsp - One-way VSP generation

onewvsp file_in= file_vel= [optional parameters]

Required parameters:

  file_in= ................. Input file
  file_vel= ................ gridded velocity file

Optional parameters:

  file_vsp= ................ Output file of calculated VSP
  file_ex= ................. Output file with the extrapolated result
  file_over= ............... writes model file with vsp positions
  nxmax=512 ................ maximum number of traces in input file
  ntmax=1024 ............... maximum number of samples/trace in input file
  file_init= ............... filename for ProMax IO initialization
  line=1  .................. 1: black lines; 0: white lines in overlay
  verbose=0 ................ silent option; >0 display info
RECEIVER POSITIONS
  xrcv=0 ................... x-position's of receivers (array)
  zrcv=0,nz*dz ............. z-position of the receivers (array)
  dxrcv=dx ................. step in receiver x-direction
  dzrcv=dz ................. step in receiver z-direction
  lint=1 ................... linear interpolate between the rcv points
  dxspr=0 .................. step of receiver spread in x-direction
  nvsp=1 ................... number of VSP positions
EXTRAPOLATION
  mode=-1 .................. type of extrapolation (1=forward, -1=inverse)
  fmin=0 ................... minimum frequency
  fmax=70 .................. maximum frequency
  ntap=0 ................... number of taper points at boundaries
EXTRAPOLATION OPERATOR DEFINITION
  select=4 ................. type of x-w operator
  opl=25 ................... length of the convolution operator (odd)
  alpha=65 ................. maximum angle of interest
  perc=0.15 ................ smoothness of filter edge
  weight=5e-5 .............. weight factor in WLSQ operator calculation
  fine=10 .................. fine sampling in operator table
  filter=1 ................. apply kx-w filter to desired operator
  limit=1.0002.............. maximum amplitude in best operators
  opl_min=15 ............... minimum length of convolution operator

  Options for select:
        - 0 = Truncated operator
        - 1 = Gaussian tapered operator
        - 2 = Kaiser tapered operator
        - 3 = Smoothed Phase operator
        - 4 = Weighted Least Squares operator
```

```
        - 5 = Remez exchange operator
        - 8 = Smooth Weighted Least Squares operator
        - 9 = Optimum Smooth Weighted Least Squares operator
        - 10= Optimum Weighted Least Squares operator

  The weighting factor is used in the convolution operator calculation.
  This calculation is done in an optimized way.
  The default weight factor is for most cases correct. For a more stable
  operator chooce a weight factor closer to 1, if 1 is chosen no
  optimization is carried out and the convolution operator is the
  truncated Inverse Fourier Transform of the Kx-w operator.
  The non-optimized operator is the truncated IFFT of a smooth Kx-w
  operator. This operator is designed by Gerrit Blacquiere.

  Note that all coordinates are related to the velocity model.

 Copyright 1997, 2008 Jan Thorbecke, (janth@xs4all.nl)

     initial version   : 14-12-1993 (j.w.thorbecke@tudelft.nl)
         version 1.0   : 11-10-1995 (release version)
         version 2.0   : 23-06-2008 (janth@xs4all.nl) 2008
```

5.3 General parameter description

The input files are the same as for the program **extrap**. The data to be extrapolated (`file_in`) and the gridded subsurface (`file_vel`) should at least have the same number of traces. If the number of traces of the subsurface is smaller an error message is the result. If the number of traces is bigger then the user should position the first receiver of the data in the subsurface grid. In no correct hdrs (`gx` and `sx`) are available this can be done with the parameter `rpos`, which indicates the tracenumber (an integer) in the subsurface grid where the first receiver is positioned. The distance between the traces in both gathers should be equal. The extrapolation direction is controlled with `mode=`. For inverse extrapolation the complex conjugate of the forward extrapolation operator is taken.
To avoid reflections at the edges of the model the parameter `ntap` can be set. `ntap` indicates the number of points at the edges for which a spatial taper is designed according to: $\exp\left(-(0.4 * (ntap - ix)/ntap)^2\right)$. Choosing `ntap` equal to half of the operator length is an optimimum value.
The output file `file_vsp` contains the pseudo VSP records for the different VSP positions. If `file_over` is defined the VSP positions are overlayed on the gridded velocity model. Displaying the `file_over` file gives an overview of the chosen VSP positions. `file_ex` (if defined) contains the extrapolated data which is extrapolated upto the deepest receiver in the VSP array.
The receiver positions of the VSP array are defined with the parameters `xrcv`, `zrcv`, `dxrcv`, `dzrcv`, and `lint`. If `xrcv` and `zrcv` are not defined the default receiver array is calculated. This receiver array is positioned at x=0 for all depth positions in the gridded subsurface model. If `xrcv` is defined with only one value (e.g. `xrcv=1500`) then the VSP is positioned at x=xrcv for all defined depth positions. Note that `dzrcv` defines at which depth postions receivers should be placed. Choosing an array for `zrcv` (e.g. `zrcv=0,2000`) gives only positions inbetween the defined depth array. For a deviated VSP configuration both `xrcv` and `zrcv` should be defined as arrays (e.g. `xrcv=1000,1000,1500 zrcv=0,2000,3000`). The parameter `lint` set to 1 calculates (with `dxrcv` and `dzrcv` defined) inbetween the given positions the receiver array (use the `file_over` option to see how the receivers are positioned). If `lint` is set to 0 then only the receiver postions at every pair of `xrcv` and `zrcv` are chosen (in the previous example only three postions).

The parameters `nvsp` and `dxspr` give the posibility to define more than one receiver arrays, `nvsp` defines the number of arrays and `dxspr` gives the distance between the receiver arrays. The first receiver array is defined with `xrcv`, `zrcv`, `dxrcv`, `dzrcv`, and `lint` (see above). The other receiver arrays have the same structure but are calculated at x-positions which are dxspr moved (use the
`file_over` option to see how the receivers are positioned).
For a more detailed discussion on the different parameters which are related to the operator optimization the reader is referred to the description of the program **opercalc**.

5.4 Examples

```
onewvsp file_vel= file_vsp= file_over=
```

5.5 To do

Nothing really.

6 opercalc

6.1 General

The program **opercalc** calculates extrapolation operators for a given frequency. This program uses the same algorithm as is used in the programs **extrap**, **cfpmod** and **migr**. This program can be used to check whether the defined parameters for the calculation of the operator table are correct. It also showes how the different optimization algortihms distort the spatial-frequency spectrum of the extrapolation operator.

6.2 Parameters

Via the command-line or in a parameter file: `par=<parameter_file>`.

```
opercalc - calculates extrapolation operators for a given frequency

opercalc  file_out= [optional parameters] > Kx-file and X-file

Required parameters:

  file_out= ................ base name of the output file(s)

Optional parameters:

  freq=20 .................. frequency at which the operator is calculated
  c=2000 ................... velocity of the medium
  dx=15 .................... stepsize in spatial direction
  dz=dx .................... extrapolation step
  nkx=512 .................. number of kx samples
EXTRAPOLATION OPERATOR DEFINITION
  opl=25 ................... length of the convolution operator (odd)
  alpha=65 ................. maximum angle of interest
  perc=0.15 ................ smoothness of filter edge
  amp=0.5 .................. amplitude smooth operator
  weight=5e-5 .............. weight factor in WLSQ operator calculation
  filter=1 ................. using filter in kx-w domain before WLSQ
  beta=3 ................... 2 < beta < 10; factor for KAISER window
  nbw=3 ......... ......... order of butterworth filter
OUTPUT DEFINITION
  cycle=0 .................. 1; units along kx-axis set to 1.0/nkx
  on_su_pipe=0 ............. 1: x or 2: Kx results on SU-pipe
  verbose=0 ................ >0: shows various parameters and results

The two files produced have a _x or _kx extension in the filename.
The _x-file contains the optimized convolution operators (9x).
The _kx-file contains the spatial spectrum of the operators (10x):
        - 1 = Truncated operator
        - 2 = Gaussian tapered operator
        - 3 = Kaiser tapered operator
        - 4 = Smoothed Phase operator
        - 5 = Weighted Least Squares operator
        - 6 = Remez exchange operator
        - 7 = Hankel function H_1(2)
        - 8 = Non-linear CFSQP optimization
        - 9 = Smooth WLSQ operator
        - 10= Exact operator (phase shift)

 Copyright 1997, 2008 Jan Thorbecke, (janth@xs4all.nl)

    intitial version   : 14-12-1993 (j.w.thorbecke@tudelft.nl)
         version 1.0   : 17-10-1995 (release version)
```

```
        version 2.0   : 23-06-2008 (janth@xs4all.nl) 2008
```

NOTE: This program can be run using the same parameter setup or parameter file as the other main programs of the EXTRAP directory.

6.3 File formats

The two files produced have a _x or _kx extension in the filename. The x-file contains the optimized operators (7 traces) in the spatial domain. The kx-file has 9 traces with trace number:

(1) Truncated operator
(2) Gaussian tapered operator
(3) Kaiser tapered operator
(4) Smoothed Phase operator
(5) Weighted least-squares
(6) Remez exchange operator
(7) Hankel function $H_1(2)$
(8) Non-linear CFSQP optimization
(9) smooth WLSQ operator
(10) Exact operator

6.4 General parameter description

All are set to default values. The operators are computed in the wavenumber domain for all wavenumber values defined. The operators are transformed back to the spatial domain, with or without an optimization step. The optimization is described in Thorbecke (2004). Parameter `nkx` defines the number of operator points in the wavenumber domain (double sided number). Parameter `opl` defines the number of operator points in the space domain (double sided number, odd). Parameters `dx` and and `dz` define the spatial sampling intervals, parameter `c` the velocity, `freq` the frequency. The parameter `alpha` defines the minimum and maximum angle of the extrapolation operators. The optimization using least-squares is controlled via the parameters `weight`, `alpha` and `perc`.
The parameter `alpha` defines the wavenumber window for which the optimization is carried out. The influence of the wavenumbers outside this window is taken to be less important. The importance of the wavenumbers outside the window is described by the parameter `weight`. If `weight=1` then all wavenumbers are equally important for the optimization and the 'optimized' result is in fact a truncation. For weighting values less than one the wavenumbers outside the wavenumber band of interest are less important. A very small weight factor can give rise to unstable results so in order to remain stable in the recursion a weight factor between 1.e-4 and 1e-7 is most convenient. The parameter `perc` describes over which bandwidth the wavenumber spectrum is filtered. It is defined as the fraction of the band of interest over which the filtering is carried out.
The output is arranged via the parameters `file_out`, `on_su_pipe` and `cycle`.

6.5 Examples

To display the default operators just type:

```
opercalc on_su_pipe=2 file_out=nep.su | suamp | suxgraph style=normal
```

for the spectra of the convolution operators, to display the spatial convolution operators type:

```
opercalc on_su_pipe=1 file_out=nep.su | suamp | suxgraph style=normal
```

6.6 To do

Nothing really...

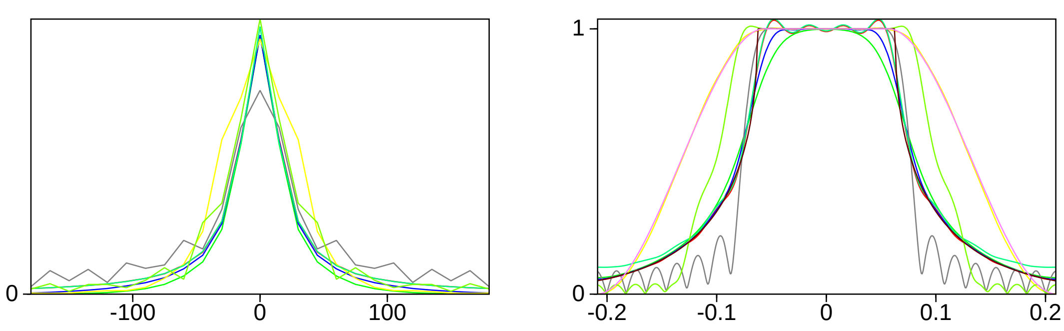


Figure 9: Wavefield extrapolation operators calculated by opercalc. Left shows the amplitude of the optimised operator in the spatial domain and right shows the amplitude in the wavenumber domain.