1 Genreal remarks

Doublecheck if "binary" or "unformatted" is correct everywhere (since I used the different programs for different other things, you might need to change some of them). In general, density files should be read in with "binary" and the xfrac files with "unformatted". Both start with the "n n n" info.

2 double to single

Since all of the following programs work with single precision as input, one has to convert the xfrac files first to single precision. You can use the small program double2single256.f to do so. input:
double2single <fin> <fout>

Here `fin` is the full path to the input file in double precision and fout the output file name.

3 fof_bubbles

use of fof_bubbles:

```
fof_bubbles  filein  fileout ( xth  HIflag  isoflag)
```

`filein` is the inputfile (ionized fraction) in single precision
`fileout` is the output ascii file (see below)
`xth` is the value at which the isosurface is drawn (0.5).
`HIflag` set to 1 means, the regions with values lower than the threshold value are bubbles, otherwise the regions with values higer than the threshold value (0).
`isoflag` (0)??? not used.

fileout:

Number	vol/boxsize	radius	collapsed fraction	mass contained withing

The last two are not used here, the density file is not even loaded.
`Number` is the serial number of the bubble.
`vol/boxsize` is the fractional volume of each bubble with respect to the volume of the box, real number.
`radius` is the radius, a spherical bubble with the same volume would have, in Mpc, real number.

4 zahn_bubbles

use of zahn_bubbles:

```
zahn_bubbles  filein1  fileout ( n  boxsize  h  xth  nscales)
```

`filein1` is an input file in single precision (ionized fraction).
`fileout` is the ascii output file (see below).
`n` is the number of cells along one dimension. (203)
`boxsize` is the box size in Mpc/h (so given 100 Mpc here, will result in 100/h Mpc) (100)
`h` is hubble parameter in units of 100 km/s /Mpc (0.7)
`xth` is the threshold value (ionized fraction) which a sphere (in average) around a cell has to have at least for the cell to be concidered to be in an ionized bubble (0.9)
`nscales` The number of different radii to concider (20). output:
`Radius` is the radius in Mpc

Radius | Num_ion | Num_neutral | inbin | Filternumber

`Num_ion` is R dP/dR, for HII bubbles with average ionization fraction larger than $x_{\rm th}$.
`Num_neutral` R dP/dR for HI bubbles with average neutral fraction larger than $x_{\rm th}$.
`inbin` is the number of cells which are the center of a sphere with Radius between R(i) and R(i+1) [R(i+1) is smaller than R(i)] that is at least 0.9 ionized.
`Filternumber` is a serial number of the radius (the `i` from above)
Beside this, there are 3 more files produced, but not needed: **sizes.ascii** radii of ionized sphere whichs center are in the middle plain.
center.dat Radius | average value of ionized fraction in ball of Radius centered around center
rhoHI.ascii x_{HII} in middle plain (from input file)

5 cc_spherical

Makes the cross-correlation between the two input files and gives out the power spectra of each of them and the cross correlation.
use: `cc_spherical input1 input2 output (n boxsize h)`
where:
`input1` is the density input file in single precision
`input2` is the ionization fraction in single precision. Order important!
`output` see below
`n` is the number of cells in one direction
`boxsize` is the physical size of the box in Mpc/h

`h` is the hubble parameter in units of km/s/Mpc

output
The first output is k The second output is the power spectrum of the (normalized) density field.
The third output is the power spectrum of the (not normalized by average) ionization fraction.
The fourth output is the cross-correlation of the two inputs.
The fifth output is the ps of the combined input as (file1-(1-file2)).
The sixth output is number of used points.

6 Minkowski

use of minkowski:

```
minkowski -x256 -y256 -z256 -b1 -l0.5 -h0.5 -m2 -s0 -iinput -ooutput
```

`-x,-y, -z` are the dimensions of the grid. If one wants to smooth the data (see below), only powers of two work. `-b` is the number of threshold bins. `-l, -h` are the lower and upper limits of the threshold. `-m` is the number of oversamplings. `-s` is the size of radius for gaussian smoothing. `-i` is the full path to the input file. Since the program is written in c, the byte-size information from the fortran unformatted files has to be cut away, that's done in the shell script. Calculate the number of bytes you want to cut away before.
The output has 9 columns:
threshold value | V0 | V1 | V2 | V3 | V0 | V1 | V2 | V3 , the first set is calculated by using Crofton's formula, the second Koenderinks; Don't expect to get correct results for the second method without smoothing.

7 shell script

I include the bash shell script I was using, in case it's useful for you. It has some comments, may be that helps...

8 redshiftlist

This calculates the volume and density weighted average ionization fractions.
input:
redshiftlist <fin1> <dens_in> <fout>
Here, `<fin1>` is expected to be the xfrac file and `<dens_in>` the density file. output:
volume averaged ionization fraction | density averaged ionization fraction