Make your own XRISM responses
\[ C(k) = \int_{0}^{\infty} \sum_{i} A_i(E) R_i(k, E) S_i(E) \, dE \, dT + B_i(k) \]

\[ C(k) = \int_{0}^{\infty} \sum_{i} A_i(E) R_i(k, E) S_i(E) \, dE \, dT + B_i(k) \]

✓ **Spec** = observed *spectrum*
✓ **Mod** = *spectral model*
✓ **RMF** = *response matrix*
✓ **ARF** = *effective area*
Reminder

\[ C(k) = \int_{0}^{\infty} \sum_{i} A_i(E) R_i(k, E) S_i(E) \, dE \, dT + B_i(k) \]

✓ Spec = observed spectrum
✓ Mod = spectral model
✓ RMF = response matrix
✓ ARF = effective area

\[ \text{Flux} \]

\[ E \]
\[
C(k) = \int_0^\infty \sum_i A_i(E) R_i(k, E) S_i(E) \, dE \, dT + B_i(k)
\]

\(\text{Spec} = \{ \text{RMF} \ast \text{Mod} \}\)

✓ Spec = observed spectrum
✓ Mod = spectral model
✓ RMF = response matrix
✓ ARF = effective area
\[ C(k) = \int_0^\infty \sum_i A_i(E) R_i(k, E) S_i(E) \, dE \, dT + B_i(k) \]

\[ \text{Spec} = \text{ARF} \ast \{ \text{RMF} \ast \text{Mod} \} \]

- ✓ Spec = observed spectrum
- ✓ Mod = spectral model
- ✓ RMF = response matrix
- ✓ ARF = effective area
Reminder

Resolve

RMFs
- rsl_Hp_5eV.rmf
- rsl_Mp_6eV.rmf
- rsl_Lp_18eV.rmf

ARFs (GV closed)
- rsl_standard_GVclosed.arf
- rsl_pointsource_fwBe_GVclosed.arf
- rsl_pointsource_fwND_GVclosed.arf
- rsl_pointsource_off_GVclosed.arf
- rsl_extflat_GVclosed.arf
- rsl_extbeta_GVclosed.arf

Xtend

RMFs
- xtd_standard.rmf

ARFs
- xtd_standard.arf
- xtd_extflat.arf
“I want an RMF for Hp events at pixel 27 exactly.”

“I want an RMF for Hp events at pixel 27 with electron loss continuum.”

“I want an ARF for a point source located exactly on the detector’s upper right corner.”

“I want an ARF for a beta extended source with other parameters than provided.”

“I want an ARF for a circular source with 1 arcmin radius and 2 arcmin off-axis.”
The solution...

Make RMFs (and ARFs) from real data! 😊

...But we are only at Cycle 1. No data is available yet!

Fair point… (Then let’s dig into the XRISM software and CALDB.)

**Warning**: we encourage you to make your own responses ONLY if you have a good reason to do so!

(Remember, canned responses are provided too…)}
Generating Resolve responses

What you need:

✓ **rmfparam file**: File containing basic RMF parameters. (Available in CALDB)

✓ **rsl_1att_b7optaxis.expo**: Dummy exposure map file for making Resolve non-observation ARFs. OPEN filter, gate valve OPEN. (Provided separately)

✓ **rsl_35pix_det.reg**: Resolve region file for all pixels, in DET coordinates, used to make non-observation ARFs. (Provided separately)

✓ **rsl_onaxiscfile_0p3to18kev.fits**: Energy grid file needed for raytracing to make canned ARFs. (Provided separately)
Generating Resolve RMFs

Anatomy of the rmfparam file
(e.g. /path/to/CALDB/data/xrism/resolve/bcf/response/xa_rsl_rmfparam_20190101v005.fits.gz)

Keyword:
- KW only
- Hp
- Mp
- Lp
- Ms
- Exp. Tail tau
- Exp. Tail frac.
- Escape peaks (51 extensions)
- Si
- Ka
- ELC
- MTRX
- ELC out.
- en. grid
- PIXEL INDEX

Credits: T. Yaqoob
Anatomy of the rmfparam file
(e.g. /path/to/CALDB/data/xrism/resolve/bcf/response/xa_rsl_rmfparam_20190101v005.fits.gz)

Credits: T. Yaqoob
Generating Resolve RMFs

Anatomy of the rmfparam file
(e.g. /path/to/CALDB/data/xrism/resolve/bcf/response/xa_rsl_rmfparam_20190101v005.fits.gz)

Build 7 CalDB “rmfparam”
- Inner 4 pixels
- Outermost Perimeter pixels
- Remaining pixels

Credits: T. Yaqoob
Generating Resolve RMFs

I want an RMF for Hp events at pixel 27 exactly.

```
$ fhelp rslrmf
```

Set to NONE to select a pixel value directly

```
$ rslrmf infile=NONE outfile="rsl_Hp_pix27" pixel=27
   resol=Hp rmfparamfile="CALDB"
```

Either Hp (= GAUSFWHM1), Mp (= GAUSFWHM2), or Lp (= GAUSFWHM3)

If set to CALDB, the RMF will take info from the latest rmfparamfile in the CALDB directory

I want an RMF with a resolution of 10 eV exactly.

```
$ rslrmf infile=NONE outfile="rsl_10eV" pixel=27
   resol=Hp rmfparamfile="my_edited_rmfparam_file.fits"
```

Modified rmfparam file with all values of all pixel columns (or, at the very least, pixel 27) in GAUSFWHM1 are set to 10 (using ftcalc, python,...)
I want an RMF for Hp events at pixel 27 with electron loss continuum.

$ \texttt{rslrmf} \text{ \texttt{in]}le=NONE \text{ \texttt{ou}}utfile="rslHp\_pix27\_XL" \text{ \texttt{pix]}el=27 \text{ \texttt{resol}}=Hp \text{ \texttt{rmfparamfile}}="\text{CALDB}" \text{ \texttt{whichrmf}}=X \text{ \texttt{splitrmf}}=yes \text{ \texttt{elcbinfac}}=32$

- Coarser bin on the ELC part of the matrix
- Creates an Xtra Large RMF that includes ELC
- Splits the RMF into two files (necessary to avoid a >2 GB size)
- Gaussian core ...
  - + exponential tail ...
  - + escape peaks + Si Kα ...
  - + electron loss continuum
- \texttt{whichrmf}=S / M / L / X
✓ These RMFs do **NOT** account for *branching ratios*! (i.e. they assume only the values given at energies and pixels from the rmfparam file)

✓ Can **NOT** be representative of the *entire detector* (including all pixels) because the sum of Gaussians is **NOT** a Gaussian!

Then how do I do to get a super realistic RMF?

Make RMFs (and ARFs) from real data! 😊
Generating Resolve ARFs

Input “source type” options
- point source
- flat circle
- β-model

Input image (e.g. Chandra, sim., etc.)

Image mode

1) xrtraytrace

xaarfgen (name might still change)

2) xaxmaarfgen

Output ARF

Credits: T. Yaqoob
Generating Resolve ARFs

Input “source type” options
- point source
- flat circle
- \(\beta\)-model

Input image (e.g. Chandra, sim., etc.)

Image mode

**Generate photons for raytracing**

**Attitude histogram**

**Expand photon list according to attitude histogram**

**Run xrt\text{raytrace}**
(coarse energy grid for *indirect mode*)

**Region selection**

Generate fine-grid ARF by fitting EA ratios & including weighted efficiency and filter functions.

Output ARF

Credits: T. Yaqoob
Generating Resolve ARFs

I want an ARF for a point source located exactly on the detector’s upper right corner.

Step 1: Run `xrtraytrace`

```
$ xrtraytrace telescop="XRISM" instrume="RESOLVE"
   energy="rsl_onaxiscfile_0p3to18kev.fits[1]" numphoton=600000
   fastmode=yes offaxis=2.317 roll=49.744 source="POINT"
   outphistfile="rsl_pointsource_uppercorner_phist.fits"
   outeafile="rsl_pointsource_uppercorner_ea.fits"
   outpsffile="rsl_pointsource_uppercorner_psf.fits"
   logfile="rsl_pointsource_uppercorner_log.log"
   mirrorfile="CALDB" obstructfile="CALDB" frontreffile="CALDB"
   backreffile="CALDB" pcolreffile="CALDB" scatterfile="CALDB"
   transmode="ALL" scattermode="ALL" psfpars="1 100 0.25"
   resplaneonly=yes phisttype=BRIEF
```

Number of photons injected in the ray-tracing simulation

The most important! (See next slide)

Output files (photon events list, effective area, PSF, log)

Energy grid (can also be entered manually)
Generating Resolve ARFs

I want an ARF for a point source located exactly on the detector’s upper right corner.

Step 1: Run `xrtraytrace`

- `offaxis`: Offset from the TELESCOPE optical axis (NOT the detector!)

- `roll`: Rotation angle from DETX (NOT the roll angle!)
Generating Resolve ARFs

I want an ARF for a point source located exactly on the detector’s upper right corner.

Step 1: Run xrtraytrace

✓ offaxis: offset from the TELESCOPE optical axis (NOT the detector!)

✓ roll: rotation angle from DETX (NOT the roll angle!)
Generating Resolve ARFs

I want an ARF for a point source located exactly on the detector’s upper right corner.

Step 1: Run \texttt{xrtraytrace}

- **offaxis**: offset from the \texttt{TELESCOPE optical axis} (NOT the detector!)
- **roll**: rotation angle from \texttt{DETX} (NOT the roll angle!)

At detector aimpoint:
\texttt{offaxis}=0.245 \quad \texttt{roll}=96.577
(could change with in-flight calibration updates...)
Generating Resolve ARFs

I want an ARF for a point source located exactly on the detector’s upper right corner.

Step 1: Run `xrtraytrace`

```
$ fhelp xrtraytrace
```

**Warning**: can take a long time! (Depending on the energy grid…)

Step 2: If necessary, change the FILTER and GATEVALV keywords in the (dummy) exposure map file

```
$ fthedit "rsl_latt_b7optaxis.expo[1]" FILTER add OPEN 
  comment="Filter state"

$ fthedit "rsl_latt_b7optaxis.expo[1]" GATEVALV add CLOSED 
  comment="Gatevalve state"
```
Generating Resolve ARFs

I want an ARF for a point source located exactly on the detector’s upper right corner.

Step 3: Run **xaxmaarfgen** (with the simulated event list as input file)

```
$ xaxmaarfgen  
  telescop="XRISM" instrume="RESOLVE"  
  emapfile="rsl_latt_b7optaxis.expo"  
  rmffile="rsl_Hp_5eV.rmf"  
  onaxiscfi="rsl_onaxiscfi_0p3to18kev.fits[1]"  
  outfile="rsl_pointsourcen_uppercorner.arf"  
  regionfile="rsl_35pix_det.reg"  
  xrt evtfile="rsl_pointsourcen_uppercorner_phist.fits"  
  qefi le="CALDB" contamifile="CALDB"  
  gatevalvefile="CALDB" onaxisffile="CALDB"  
```
Generating Resolve ARFs

I want an ARF for a beta extended source with other parameters than provided.

Step 1: Run \texttt{xrtraytrace}

```
$ \texttt{xrtraytrace} \texttt{telescop="XRISM" instrume="RESOLVE"}
\texttt{energy="rsl_onaxiscfile_0p3to18kev.fits[1]" numphoton=600000}
\texttt{fastmode=yes offaxis=0.245 roll=96.577}
\texttt{source="BETAMODEL" betapars="1.26 0.53 5.7"}
\texttt{outphistfile="rsl_extbeta_phist.fits"}
\texttt{outeafile="rsl_extbeta_ea.fits"}
\texttt{outpsffile="rsl_extbeta_psf.fits"}
\texttt{logfile="rsl_extbeta_log.log" mirrorfile="CALDB"}
\texttt{obstructfile="CALDB" frontreffile="CALDB" backreffile="CALDB"}
\texttt{pcolreffile="CALDB" scatterfile="CALDB" transmode="ALL"}
\texttt{scattermode="ALL" psfpars="1 100 0.25" resplaneonly=yes}
\texttt{phisttype=BRIEF}
```

The source is now a beta model!

Center of the source is on-aimpoint

Parameters to change (see next slide)
Generating Resolve ARFs

Step 1: Run `xrtraytrace`

I want an ARF for a beta extended source with other parameters than provided.

`$ fhelp xrtraytrace`

$betapars="1.26 0.53 5.7"

\[
N(r) = C \left[1 + \left(\frac{r}{r_c}\right)^2\right]^{(1.5 - 3\beta)}
\]
Generating Resolve ARFs

I want an ARF for a beta extended source with other parameters than provided.

Step 1: Run `xrtraytrace`

```
$ fhelp xrtraytrace
```

`betapars="1.26 0.53 5.7"`

\[ N(r) = C \left[ 1 + \left( \frac{r}{r_c} \right)^2 \right]^{(1.5-3\beta)} \]
Generating Resolve ARFs

I want an ARF for a beta extended source with other parameters than provided.

Step 1: Run `xrtraytrace`.

```
betapars="1.26 0.53 5.7"
```

```
N(r) = C [1 + (r/r_c)^2]^{(1.5-3\beta)}
```

Step 2 & 3: As before.

$ fhelp xrtraytrace $
Generating Xtend responses

What you need:

✓ `xtd_1att_nobadpix_b7optaxis.expo`: Dummy exposure map file for making Xtend non-observation ARFs. OPEN filter, gate valve OPEN. (Provided separately)

✓ `xtd_det_r2p50_b7optaxis.reg`: Xtend 2.5’ radius circle region file in DET coordinates, centered on the optical axis position, used to make non-observation ARFs. (Provided separately)

✓ `xtd_onaxiscfile_0p3to18kev.fits`: Energy grid file needed for raytracing to make canned ARFs. (Provided separately)
Generating Xtend ARFs

I want an ARF for a circular source with 1 arcmin radius and 2 arcmin off-axis.

Step 1: Run `xrtraytrace`

```
$ xrtraytrace telescop="XRISM" instrume="XTEND"
energy="xtd_onaxisicfile_0p3to18kev.fits[1]" numphoton=600000
fastmode=yes offaxis=2.0 roll=0.0 source="FLATCIRCLE"
flatradius=1.0 outphistfile="xtd_extflatoff_phist.fits"
outeafile="xtd_extflatoff_ea.fits"
outpsffile="xtd_extflatoff_psf.fits"
logfile="xtd_extflatoff_log.log" mirrorfile="CALDB"
obstructfile="CALDB" frontreffile="CALDB"
backreffile="CALDB" pcolreffile="CALDB" scatterfile="CALDB"
transmode="ALL" scattermode="ALL" psfpars="1 100 0.25"
resplaneonly=yes phisttype=BRIEF
```

Now using Xtend (also in input and output files)
Generating Xtend ARFs

Step 1: Run `xrtraytrace`

$ fhelp xttraytrace

✓ In the case of Xtend, the on-aimpoint (i.e. center of the detector) almost coincides with the on-axis (i.e. center of the telescope)!

At detector aimpoint:

`offaxis=0.0 roll=0` is a good approximation

Step 2 & 3: As before
✓ Generating accurate ray-tracing events at many energies can take a long time!

✓ Try to find the best compromise between science case accuracy vs. computing cost

✓ **Remember**: this is an advanced tutorial! It is **VERY** likely that your science justification can reasonably be done with the responses already available online (provided by the GOF)