

ZERO-ORDER REGULARIZED INVERSION SOLAR SOFTWARE

Anna Maria Massone
CNR-INFM LAMIA
Via Dodecaneso 33, I-16146 Genova, Italy

Michele Piana
Dipartimento di Informatica, Università di Verona,
Ca' Vignal 2, Strada le Grazie 15, I-37134 Verona, Italy

Acknowledgments

The implementation of this software has been performed by A. M. Massone and M.Piana. However this is the result of a joint collaboration together with the following persons:

- J. C. Brown, Department of Physics and Astronomy, University of Glasgow
- A. G. Emslie, Department of Physics, Oklahoma State University
- E. P. Kontar, Department of Physics and Astronomy, University of Glasgow

We also want to thank J. Kašparová (Academy of Sciences, Czech Republic), A. Veronig (University of Graz, Austria) and A. Warmuth (Astrophysical Institut Potsdam, Germany) for testing the first version of the software.

The financial support for this research has been provided by the British Royal Society, the Italian Istituto Nazionale di Alta Matematica (Gruppo Nazionale di Calcolo Scientifico), the Italian MIUR and the International Space Science Institute (ISSI). These Institutions are therefore kindly acknowledged.

Plan of the ‘README’

This document is divided into two parts. In the first part we will briefly describe the theoretical framework of the regularization theory for linear inverse problems which is at the basis of the algorithm implemented in the software. In the second part we will describe in more details the architecture and performances of the software.

PART I: INVERSE PROBLEM AND METHOD

1. Introduction

This software implements an inversion method for the solution of the general bremsstrahlung problem (Brown 1971; Brown et al. 2003)

$$I(\epsilon) = \frac{1}{4\pi R^2} \bar{n} V \int_{\epsilon}^{\infty} \bar{F}(E) Q(\epsilon, E) dE \quad . \quad (1)$$

In this equation ϵ is the photon energy, $I(\epsilon)$ is the emitted hard X-ray photon spectrum (photons $\text{cm}^{-2} \text{s}^{-1} \text{keV}^{-1}$); E is the electron energy, $\bar{F}(E)$ is the density-weighted mean electron spectrum in the source; $Q(\epsilon, E)$ is the bremsstrahlung cross-section; R is the distance to the observer, V is the source volume and \bar{n} is the mean target density. The software provides reconstructions of $\bar{F}(E)$ given measurements of $I(\epsilon)$ for different analytic forms of $Q(\epsilon, E)$.

Equation (1) is a simplified model for the bremsstrahlung emission process in which a solid-angle-averaged cross section depending only on E and ϵ (isotropic assumption) and a mean electron flux differential only in E are used. In general, the correct cross section for bremsstrahlung photon emission depends also on the emission angle (θ) between the incoming electron and the emitted photon directions (Gluckstern & Hull 1953; Koch & Motz 1959) while the electron flux is differential in E and in the solid angle Ω describing the incoming electron direction. It follows that a more realistic model is given by:

$$I(\epsilon) = \frac{1}{4\pi R^2} \bar{n} V \int_{\epsilon}^{\infty} \int_{\Omega} Q(\epsilon, E; \theta) \bar{F}(E, \Omega') d\Omega' dE, \quad (2)$$

where Ω' ranges from 0 to Ω . The software also addresses the inversion of Eq. (2) by assuming $\bar{F}(E, \Omega')$ separable in E and Ω' and recovering $\bar{F}(E)$ by choosing an appropriate form for the angular dependance (Massone et al. 2004).

2. Zero-order regularization method

The first formulation of Tikhonov regularization method is in (Tikhonov 1963). For an exhaustive description of regularization theory for linear inverse problems, see (Bertero 1989). Applications of this approach to the bremsstrahlung problem are in (Craig & Brown 1986; Piana 1994; Massone et al. 2003; Piana et al. 2003; Kontar et al. 2004).

The method applies to the discretized form

$$\mathbf{I} = A\bar{\mathbf{F}} \quad (3)$$

of Eq. (1), where, in general, \mathbf{I} is an N_{ph} -vector representing the photon spectrum $I(\epsilon)$, A is an $N_{ph} \times N_e$ matrix representing the cross-section $Q(\epsilon, E)$ and $\overline{\mathbf{F}}$ is an N_e -vector representing the mean electron spectrum $\overline{F}(E)$ (we assume $N_{ph} \leq N_e$). The basic goal of this approach is to provide a reliable estimate of $\overline{\mathbf{F}}$ without a priori choosing any parametrized form for $\overline{F}(E)$. This task is accomplished by means of a regularization algorithm (zero-order regularization method) which maintains much of the fidelity in the recovered mean electron spectrum while notably reducing the noise amplification through the imposition of a degree of smoothness in the restored solution. In particular, the method seeks for the solution of the minimization problem

$$\|A\overline{\mathbf{F}} - \mathbf{I}\|^2 + \lambda\|\overline{\mathbf{F}}\|^2 = \text{minimum} \quad , \quad (4)$$

where the regularization parameter λ is a positive real number tuning the trade-off between the fidelity of the data fit and the smoothness of the recovered $\overline{\mathbf{F}}$.

The algorithm is characterized by the following steps:

1. Computation of the solution of Eq. (4): the formula

$$\overline{\mathbf{F}}_\lambda = \sum_{k=1}^{N_{ph}} \frac{\sigma_k}{\sigma_k^2 + \lambda} (\mathbf{I}, \mathbf{v}_k) \mathbf{u}_k \quad (5)$$

is applied. The set of triples $\{\sigma_k, \mathbf{u}_k, \mathbf{v}_k\}_{k=1}^{N_{ph}}$ are obtained by computing the Singular Value Decomposition (SVD) (Golub & van Loan 1993) of the matrix A , i.e. by solving the shifted eigenvalue problem

$$A\mathbf{u}_k = \sigma_k\mathbf{v}_k \quad A^T\mathbf{v}_k = \sigma_k\mathbf{u}_k \quad (6)$$

where A^T is the transpose matrix of A ; the real positive numbers σ_k are the singular values of A and satisfy $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{N_{ph}}$; the vectors \mathbf{u}_k and \mathbf{v}_k are the singular vectors of A (for an SVD analysis of the bremsstrahlung problem, see (Piana & Brown 1998)).

2. Optimal choice of the regularization parameter: for each $k = 1, \dots, N_{ph}$, the software considers the function

$$S_k^\lambda = \frac{1}{k} \sum_{i=1}^k r_i^\lambda \quad , \quad (7)$$

where r_i^λ is the i -th normalized regularized residual corresponding to $\overline{\mathbf{F}}_\lambda$. If the residuals were randomly distributed, the normalized cumulative residuals would exhibit a random walk with standard deviation $\sigma = 1/\sqrt{k}$. In Eq. (7) the normalized regularized residuals r_i^λ are increasingly correlated for increasing values of the regularization parameter λ . Therefore the optimal λ is given by the biggest value of λ for which S_k^λ is within $\pm 3/\sqrt{k}$ for all k .

3. Error propagation and resolution

An estimate of the stability of the regularized solution in correspondence with perturbation of the data is determined by computing the confidence strip of the regularized solution (Piana 1994), i.e. by superimposing regularized $\overline{\mathbf{F}}_\lambda$ corresponding to several random modifications of the original photon spectrum.

In order to assess the spectral resolution, the software determines the last significant singular component in the expansion (5) by applying the condition (Miller 1970)

$$\sigma_k \geq \sqrt{\lambda} \quad . \quad (8)$$

If $\sigma_{\overline{k}}$ is the last singular value satisfying (8), the resolution limit is given by the distance between successive zeros of the last significant singular vector $u_{\overline{k}}$.

PART II: INVERSION SOFTWARE

1. Main features

The inversion package is made of the following IDL files:

- `hsi_regularized_inversion.pro`
- `hsi_reg_invert.pro`
- `hsi_regularisation_genova.pro`
- `hsi_reg_ge_spectralindex.pro`
- `hsi_reg_ge_rescaling.pro`
- `hsi_reg_ge_cross_section.pro`
- `hsi_reg_ge_kramers.pro`
- `hsi_reg_ge_bethe.pro`
- `hsi_reg_ge_bh_elwert.pro`
- `hsi_reg_ge_cs_haug.pro`
- `hsi_reg_ge_cross3bn.pro`
- `hsi_reg_ge_ee_bremcross.pro`
- `hsi_reg_ge_directivity.pro`
- `hsi_reg_ge_angle_cross.pro`
- `hsi_reg_ge_bremss_cross.pro`
- `hsi_reg_ge_regularization.pro`
- `hsi_reg_ge_confidence.pro`
- `hsi_reg_ge_inverse_snr.pro`

Furthermore we include the graphical routine ‘`reg_ge_plot.pro`’ and the routine ‘`reg_ge_inversion.pro`’ which provides an example of how to run the code in a user-friendly manner.

1.1. Run of the software

The calling sequence is :

```
hsi_regularized_inversion, inputfile, outputfile,$
  photon_bin_position=photon_bin_position,$
  el_energy_max_factor=el_energy_max_factor,$
  electron_bin=electron_bin,$
  lambda_multiplier=lambda_multiplier,$
  crosssection=crosssection,$
  viewingangle=viewingangle,$
  spreadangle=spreadangle,$
  Z=Z,$
  confidencestrip=confidencestrip
```

In order to run the code the user must specify input and output filenames. Furthermore the setup of the software is performed by fixing some keywords. It is not necessary to fix the keywords. If a keyword is not set, its value is automatically set to the default value.

1.2. Input file

The input file must be an ASCII file organized in 4 columns of single precision data as follows:

First column : lower energy of photon bin.
Second column : upper energy of photon bin.
Third column : X-ray flux (photons/cm²/s/keV at detector).
Fourth column : 1- σ uncertainty in flux value.

Note that the input photon energy array should have no data gaps, and it must be uniformly spaced (on either a linear or logarithmic scale). The input spectrum must contain at least 10 points. We strongly recommend that only photon energies above 10 keV be used, since the code has no capability to deal with spectral lines.

To avoid generation of (unphysical) negative flux values, all the photon fluxes must be positive and at least equal to the specified 1- σ uncertainty. The code uses only data points up to (but not including) the energy channel which first violates this criterion. The other data points are automatically neglected.

1.3. Output file

The software provides one output file containing three arrays and one string, stored in structure form in the *datastruct* structure. In the following a brief description of the four structure-type variables is given.

Tag 1 *Name* : STRIP
 Type : double Array[$N_e, N + 2$]

where N_e is the number of electron energy points and N the number of regularized solutions for the creation of the confidence strip. The array contains the electron energies and each realization of the solution $\bar{\mathbf{F}}_\lambda$ in the following format:

datastruct.strip[*],0] = a vector of N_e elements containing the electron energies
 datastruct.strip[*],1] = a vector of N_e elements containing the original
 (unperturbed) solution $\bar{\mathbf{F}}_\lambda$ at the specified energies
 datastruct.strip[*],2:N+1] = array containing N different electron fluxes $\bar{\mathbf{F}}_\lambda$, at the
 specified energies, produced by randomly perturbing the
 data in inputfile

Tag 2 *Name* : RESIDUALS
 Type : double Array[$N_{ph}, 6$]

where N_{ph} is the number of photon energy points. The array contains each realization of the original (unperturbed) solution $\bar{\mathbf{F}}_\lambda$. The data is in six columns, with each row arranged as follows:

datastruct.residuals[*],0] = Photon energy used (defined by photon energy array and
 optional keyword *photon_bin_position* value - see below)
 datastruct.residuals[*],1] = Photon flux value (from the input file)
 datastruct.residuals[*],2] = Uncertainty in photon flux value (from the input file)
 datastruct.residuals[*],3] = Photon flux corresponding to the recovered $\bar{\mathbf{F}}_\lambda$ array
 datastruct.residuals[*],4] = Residual photon flux (actual - recovered)
 datastruct.residuals[*],5] = Cumulative residual, defined as in Eq. (7)

Note that Eq. (7) can be used as a measure of residual *clustering*; for an acceptable fit to the data, $|S_k^\lambda|$ should be less than n/\sqrt{k} , where n is the number of allowed standard deviations ($n = 3$ is used in the code to drive the solution to an acceptable solution).

Tag 3 *Name* : RESOLUTION
 Type : double Array[M , 2]

where M is the number of resolution intervals. The array contains two columns representing sampled energy resolution values as follows:

datastruct.resolution[*],0] = resolution bar center ($\sqrt{E_{low} * E_{high}}$)
 datastruct.resolution[*],1] = resolution bar width ($E_{high} - E_{low}$)

If two \bar{F}_λ values are distant by more than the energy resolution, then they can be realistically treated as independent; if they fall within the energy resolution they should be treated as dependent to some degree. The number of resolution intervals (M) is determined by the code (it cannot be specified by the user), and so the number of rows in this array (typically 10-20) is determined internally.

Tag 4 *Name* : INFORMATION
 Type : string *outputfile.info*

The string is the pathname of an additional text file containing various technical pieces of information, all textually labeled.

The graphical routine included in the software allows to produce also five postscript files representing the input photon spectrum, the regularized solution, the confidence strip with the resolution bars superimposed and two plots for the residuals.

1.4. Keywords

The setup of the software is performed by fixing the keywords of *hsi_regularized_inversion.pro*. The keywords are:

- *photon_bin_position* : fractional location within photon energy bin to use. Range is [0,1] with 0 = low end, 0.5 = mid-point, 1 = upper end, etc. The default is 0.5. NOTE: If the routine determines that logarithmic data sampling is to be used, this parameter is ignored, and the geometric mean bin energy is used by default.
- *el_energy_max_factor* : the maximum electron energy (in units of the maximum photon energy provided) in the returned electron energy spectrum array (i.e., $E_{max} = el_energy_max_factor * \epsilon_{max}$). Must be in the range [1,5]; the default is 2. The minimum sampled electron energy is always equal to the minimum sampled photon energy.

- *electron_bin* : width (keV) of the energy bins in the returned electron flux array. Values are returned at values spaced by this quantity. Default = 1.
- *lambda_multiplier* : factor by which the user wishes to increase or decrease the value of the smoothing parameter used in the algorithm. It must be non-negative and in the range [1.e-2,1.e2].

lambda_multiplier > 1 implies a smoother solution and worse residuals;

lambda_multiplier < 1 implies a more unstable solution and better residuals.

The default value, which is STRONGLY recommended, is 1. We cannot emphasize strongly enough the desirability of using the default value *lambda_multiplier*=1. The value of the smoothing parameter lambda is determined through a rigorous procedure, and using other values of lambda usually results in highly undesirable features in the solution and/or the photon residuals. Nevertheless, we understand that users may wish to vary this parameter in order to explore the effects of such a variation. Attempts to use a value of *lambda_multiplier* outside the range [0.5,2] generates an appropriately warning message.

- *crosssection* : a text string containing one of possible choices for the cross-section $Q(\epsilon, E)$:

‘Kramers’	=	$Q \sim \frac{1}{\epsilon E}$. Simple analytic form of the electron-proton bremsstrahlung cross-section. Not accurate, primarily for comparison with analytic results;
‘Bethe-Heitler’	=	$Q \sim \frac{1}{\epsilon E} \frac{\ln(1+\sqrt{1-\frac{\epsilon}{E}})}{\ln(1-\sqrt{1-\frac{\epsilon}{E}})}$. More accurate non relativistic analytic form (Koch & Motz 1959);
‘Bethe-Heitler-Elwert’	=	the Bethe-Heitler form with the Elwert collisional correction term applied (Johns & Lin 1992) (Johns & Lin 1992, a);
‘Cross3BN’	=	fully relativistic, solid-angle-averaged, cross-section (Koch & Motz 1959). Includes the Elwert collisional correction factor. Used as default.
‘Haug’	=	a functional fit to the Cross3BN relativistic cross section. Includes the Elwert collisional correction term (Haug 1997).
‘Cross3BNee’	=	Cross3BN plus the term due to electron-electron bremsstrahlung (Haug 1998).

‘Anisotropy’ = fully angle-dependent electron-proton cross-section (Gluckstern & Hull 1953; Massone et al. 2004). If this cross section is chosen, keywords *viewingangle* and *spreadangle* should also be specified.

The default is ‘Cross3BN’. Although the selection ‘Anisotropy’ is valid, we caution the user that use of this cross-section increases the computational time considerably, since it entails not only a more complicated form of cross-section, but also the need for significant trigonometrical calculation. The same caution applies to addition of the electron-electron bremsstrahlung term through use of ‘Cross3BNee’. Unless photon energies significantly above 100 keV are used, the results for ‘Cross3BN’ and ‘Cross3BNee’ are very similar.

- *viewingangle* : the angle between the mean electron propagation vector and the line of sight to the observer (in degrees between 0 and 180). Ignored unless *crosssection*=‘Anisotropy’ is chosen. Default is 180 degrees.
- *spreadangle* : half-angle of the cone within which the electron velocities are (uniformly) distributed (in degrees between 0 and 180). Ignored unless *crosssection*=‘Anisotropy’ is chosen. Default is 180 degrees. Note that *spreadangle*=180 corresponds to an isotropic distribution.
- *Z* : value of the root-mean-square atomic number of the target. Range = [1,2]. Default = 1.2
- *confidencestrip* : number of realizations of the regularized solution to be determined. Each realization is produced by randomly perturbing the data in *inputfile* using the noise levels specified therein. Used to construct a ‘set’ of solutions in order to estimate the uncertainty in the recovered solution. Default = 10, maximum = 50. Setting *confidencestrip* = 0 or 1 produces a single solution.

1.5. Warnings

Warnings apply to two different situations:

- For all keywords: if a keyword is set to a forbidden value, the warning message is (example)

```
WARNING : Invalid value for the keyword PHOTON_BIN_POSITION
PHOTON_BIN_POSITION must be in the interval [0,1]
PHOTON_BIN_POSITION set to default (=0.5)
```

- For the keyword *lambda_multiplier*: if the keyword is set to a value outside the range [0.5,2], the warning message is

WARNING : LAMBDA_MULTIPLIER =1 is STRONGLY recommended

2. Graphical routine

The graphical routine ‘reg_ge_plot.pro’ allows the user to produce plots of the results both on the screen and in hardcopy. The calling sequence is :

reg_ge_plot, *outputfile*, **hard=hard**

where *outputfile* is the name of the output file (with data stored in structure form) provided by the software and *hard* is a keyword (default=0) whose value must be set to 1 in order to save the plots in postscript format. If *hard*=1, the outputs of the routine are:

1. a plot containing the photon spectrum (*outputfile.photons.ps*);
2. a plot containing the mean electron spectrum (*outputfile.fbar.ps*);
3. a plot containing the confidence strip with the resolution bars superimposed (*outputfile.strip.ps*);
4. a plot containing the original photon spectrum with the photon spectrum corresponding to the regularized mean electron spectrum (upper panel) and the normalized residuals (lower panel) (*outputfile.residuals.ps*);
5. a plot containing the histogram of the normalized residuals frequencies (upper panel) and the cumulative residuals (lower panel) (*outputfile.cumulative.ps*).

3. The ‘reg_ge_inversion.pro’ routine

The routine ‘reg_ge_inversion.pro’ is an example routine showing how the user can run the software by simply specifying input and output filenames.

REFERENCES

- Bertero, M., 1989, in P. W. Hawkes (ed.) *Advances in Electronics and Electron Physics*, **75** (New York: Academic).
- Brown, J. C., *Solar Phys.*, **18**, 489, 1971.
- Brown, J. C., Emslie, A. G. and Kontar, E. P., *Astrophysical Journal*, **595**, L115-L117, 2003.
- Craig, I. J. D. and Brown, J. C. 1986 *Inverse Problems in Astronomy* (London: Adam Hilger).
- Gluckstern, R. L., and Hull, M. H. *Phys. Rev.*, **90**, 1030-1035, 1953.
- Golub, G. H. and van Loan, C. F. 1993 *Matrix Computation* (Baltimore: Johns Hopkins).
- Haug, E., *Astron. Astrophys.*, **326**, 417-418, 1997.
- Haug, E., *Solar Phys.*, **178**, 341-351, 1998.
- Johns, C. M. and Lin, R. P. *Solar Phys.*, **137**, 121-140, 1992.
- Johns, C. M. and Lin, R. P. *Solar Phys.*, **142**, 219-220, 1992,a.
- Koch, H. W. and Motz, J. W. *Rev. Mod. Phys.*, **31**, 920-953, 1959.
- Kontar, E.P., Piana, M., Massone, A.M., Emslie, A.G., & Brown, J.C., *Sol. Phys.*, **225**, 293, 2004.
- Massone, A.M., Piana, M., Conway, A., and Eves, B. *Astron. Astrophys.*, **405**, 325-330, 2003.
- Massone, A.M., Emslie, A.G., Kontar, E.P, Piana, M., Prato, M. and Brown, J.C., *Astrophysical Journal*, **613**, 1233-1240, 2004.
- Miller, K., *SIAM J. Math. Anal.*, **1**, 27-32, 1970.
- Piana, M., *Astron. Astrophys.*, **288**, 949-959, 1994.
- Piana, M. and Brown, J.C., *Astron. Astrophys.*, **132**, 291-299, 1998.
- Piana, M., Massone, A. M., Kontar, E.P., Emslie, A. G., Brown, J. C. and Schwartz, R. A., *Astrophys. J.*, **595**, L127-L130, 2003.
- Tikhonov, A. N. 1963, *Dokl. Akad. Nauk. USSR*, **153**, 501-504, 1963.