LC3 MCMC technical manual

1 Introduction

This document is not an official publication just a technical manual, and so it not follows the publications strict laws. It contains quotes from our publications, and it can not be referred to.

To determine the supernova explosion parameters, we need to model the observed data. This generally done with hydrodynamical codes. However these hydrodynamical simulations are time-consuming. Because of this, a simple analytical method may also be used to get approximate results. Such simple LC models have been developed as early as the 1980's by [Arnett & Fu, 1989] to explain and fit bolometric light curves of supernovae. Later they were further improved by the inclusion of numerical computations of certain aspects, thus becoming semi-analytic. These models assume spherical symmetry and do not adequately model the initial transient behavior at early stages ($t<20$ days), which obviously makes them inferior compared to detailed hydrodynamical models; nevertheless, they are useful to derive estimates or constraints of basic parameters like the explosion energy, ejected mass and initial radius, without the high computational resources demanded by the latter. Applications of the code so far indicate that it can model IIP LC particularly well.

Although these estimates are considered only preliminary, these give first-order approximations when the observational information is limited, and their fast run speed allows to add a powerful parameter optimization method; such as Markov Chain Monte Carlo (MCMC) ([Metropolis et al., 1953]; [Hastings et al., 1970]; [Gilks et al., 1996]) algorithm, to search for the best fits to the bolometric light-curve. The MCMC method is a well-established technique for constraining parameters from observed data, and especially suited for the case when the parameter space has a high dimensionality. This method has the nice ergodic property[1], which allows the various integrals over the parameter space – mean value, standard deviation, and percentiles in particular – to be computed as simple sums over the chain elements. Because of the ergodic nature of this method, the uncertainties and the strong correlations between the physical parameters ([Nagy et al., 2014]) can also be mapped.

MCMC is a well-established technique for constraining parameters from observed data, and especially suited for cases when the parameter space has a high dimensionality ([Gilks et al., 1996]). The sampling maps the whole parameter space based on the joint posterior probability distribution of all the parameters. It has the nice ergodic property which allows the various integrals over the parameter space (mean value, standard deviation, and percentiles in particular) to be computed as simple sums over the chain elements. Therefore it makes possible to provide not only best estimates of the parameters, but also confidence intervals for them (leading to uncertainty determinations); furthermore, correlations between the various parameters can also be revealed.

Please see [Jager et al., 2020] [Nagy, Vinko, 2016], [Nagy et al., 2014]) for more information like the physx behind it.

2 Installation

This program uses only standard c++ libraries and code, so you don't need any further downloads, it needs only a c (c++) compiler. g++ *name* -o *output* for Linux.

[1]Ergodic: given enough time, the method traverses every portion of the parameter space.

3 Input and running

Current there 2 necessary input: the parameter file, and the light curve file.

The light curve file "*bol-gorbe.txt*" contains the input bolometric light-curve in the form of:
JD; mag; mag error
Currently only this form is accepted but it can be altered in section "void $meres()\{\}$". JD must be in days, however the zero-point is not fixed, and can be set in the parameter file. The time don't need to be in order, there is an arrange by time. The mag must be in bolometric magnitude (or absolute magnitude if bolometric not available). Making the bolometric light-curve is beyond this doc, so please see [Lyman et al., 2014]. The mag err must be given, if there is no error available, fill it with 1. Of course it must be given in magnitudes.

The parameter file "*parametersMC.inp*" contains the parameters for the fitting.
Comments after the number is possible, so by default the parameter file contains a label after a value. **The order is fixed, don't change.** Out-commented rows (#) won't be read, but the slot is still needed, so you have to make a new row with valid value. This is because the program recognize the parameters by their orders. If the input parameter files are not given, a new default is generated. Use this as a base.

The MAIN parameters for the fit:
- Zero-point of the JD [day] (*tkezd*; JD of shock breakout)
- Min time; start day of the fitting, days before this is not fitted (*tmin*; this is in days after the explosion; can be set to 0)
- Ionization temperature [K] (5500 for hydrogen); if negative number given, this can also be fitted ($Tion$)
- Exponential density profile exponent (a; default 0)
- Power-low density profile exponent (s; default 0); if the Exponential density profile exponent is 0, then this is used; if negative number given, this can also be fitted
- Thomson scattering opacity [cm^2/g] (*kappa*, κ; 0.2 or 0.3 recommended); if negative number given, this can also be fitted
- Initial magnetar rotational energy [erg] (default 0), this is fixed right now (Ep)
- Characteristic time scale of magnetar spin-down [day] (default 0), this is fixed right now (tp)
- Max time; Final epoch [day] (*tmax*)
- Number of MC loops (between 100000 and 1000000 recommended) (NMC)
- Time of tail; Time where the plateau ends [day] (*ttail*)

You can also change the prior/acceptable region here in this file, at the technical parameters. If not given, if the max values smaller then the min values, 0 or negative values given the default values used (s and Tion are exception where you can add 0 value). The default values can be found at section MCMC. Please note giving bad values (too small or too bug values, or for example giving bigger values then 0.4 for kappa) are not checked, so use it carefully. Beta, Delta and szoraskuszob values are technical values. Beta and Delta is the Monte Carlo parameter, and changing it sets the random steps density (See MCMC section for more). Altering it significantly cause bad fitting. Altering the the prior/acceptable region is not recommended, but you can do it if you seem it right, because reducing the prior/acceptable region, the good fitted values are found sooner, but may also lock out good fits.

The format for priori and technical parameters for the fit:

- Empty line!
- MIN Initial Radius [cm]
- MAX Initial Radius [cm]
- MIN Ejected mass [M_{sol}]
- MAX Ejected mass [M_{sol}]
- MIN Kinetic energy [10^{51} erg = foe]
- MAX Kinetic energy [10^{51} erg = foe]
- MIN Thermal energy [10^{51} erg = foe]
- MAX Thermal energy [10^{51} erg = foe]
- MIN Opacity [cm^2/g]
- MAX Opacity [cm^2/g]
- MIN Power-low density profile
- MAX Power-low density profile
- MIN Ionization temperature [K]
- MAX Ionization temperature [K]
- MIN Initial nickel mass [M_{sol}]
- MAX Initial nickel mass [M_{sol}]
- MIN Gamma-leak (Ag) [day^2]
- MAX Gamma-leak (Ag) [day^2]
- Expansion velocity priori mean (v) [km/s]
- Expansion velocity priori sigma (v)(Gaussian) [km/s]
- Determinate the nickel mass Ag function? [binary] (if yes: $fitnickelmass = 1$)
- Beta (default 0.1)
- Delta (default 30000)
- Minimum ionization zone [0.1:1]
- szoraskuszob

- Empty line!
- Whole or No Tail fitting used? [binary] (for plotting only, more from this later!)
- Plotting grid number. (for plotting only)
- Burn in step number. (for plotting only, loops before this number are not considered)

Giving the good values for the times is important. Bad values can cause bad fitting, or errors.
- *tmin*: some supernova has a double peak and the first peak (caused by an outer layer) which don't fitted (or must be fitted separately). If it doesn't present set it to 0.
- *ttail*: The date after the drop of the plateau, and the first day of the tail; nebular phase. This parameter is very important, but it is more likely a technical parameter, **later it will be described**. If there are missing points at the end of plateau, just give a date between the last plateau, and the first tail point.
- *tmax*: the maximum epoch till runs the program. After a while there may some other effects beside the Ni-Co-Fe decay, which are not fitted, so those must be let out (or it will give wrong Ni mass).
Example for this can be seen on Fig 1.

For the other main parameter in summary:
Ionization temperature: SN with high plateau drop like type II-P the 5500 K value (hydrogen) should be used. Other SN which has lower plateau drop like type II-L then it should be fitted (negative value).
Density profiles: 0 recommended. Can also be fitted (negative value), but the 0 value is the best

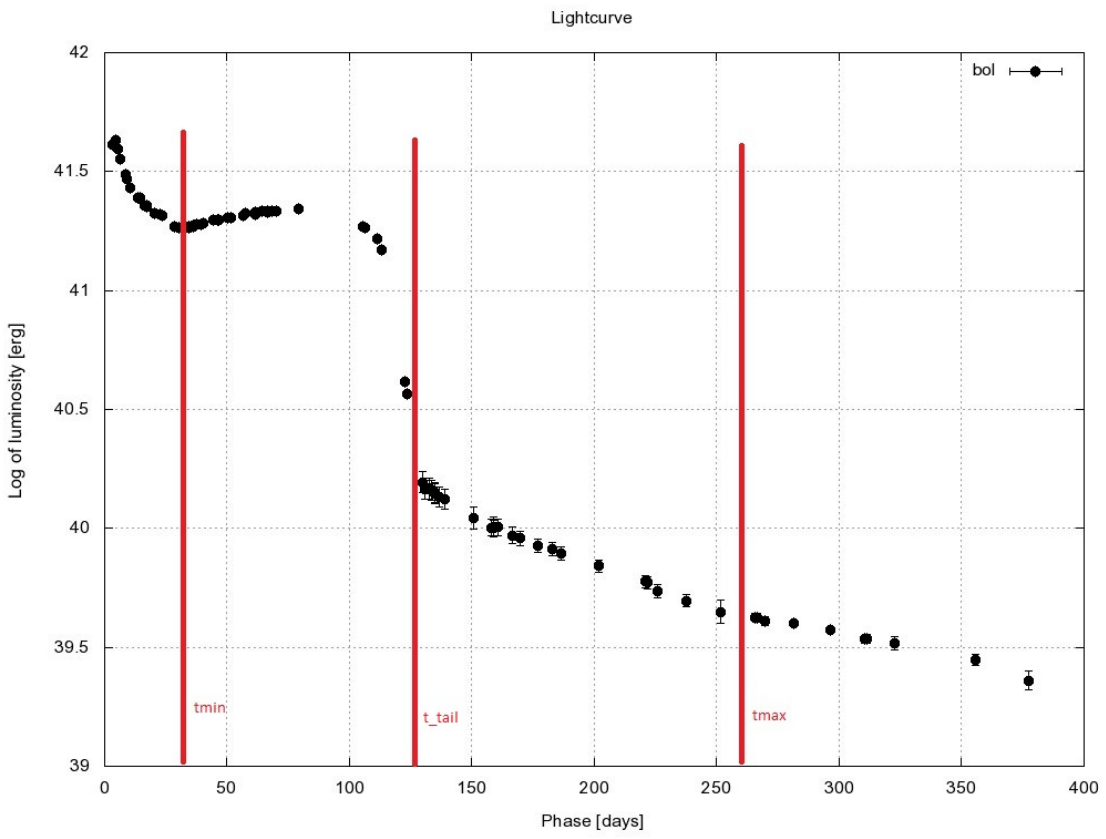

Figure 1: Example for the times to give to the parameter file. Example light curve SN 2005cs; [Pastorello et al., 2009]

match with the literature (see [Jager et al., 2020] for details).
Opacity: This parameter is tricky and important. The mean values for this is 0.2 and 0.3, but other values also possible. **So the fitting should be done with multiple kappa values or sampled it as well.** (To fit it, set negative value.)

Magnetar: 0 recommended for now! In Version 1 it is fixed. In Version 2 it is a fitted value too. It is incompatible with the NiAg function fitting (it has effect on the tail too!)! Using NiAg function fitting with non zero magnetar values will set the magnetar values to zero.

The technical parameters:
Obviously the MIN and Max values gives the minimum and maximum values for the sampling, a.k.a. the priori. The distribution for this is uniform, so any value between minimum and maximum are equally possible, but any value beside that are not. The only exception is the velocity priori which has a Gaussian distribution.

More about the velocity priori:
If we have spectroscopy then we can set this velocity prior. The expansion velocity theoretically the outermost layer speed. This speed is approximately the photosperic velocity at the end of the plateu for the core. For the outer shell it is the photosperic velocity at the beginning of the light-curve. If the observation is not adequate, by correctly including the radial velocity as an a priori probability

distribution, the parameters can be become quite well constrained. If set to 0, then every velocity value is accepted.

To determinate the nickel mass, 2 method are used: First it can be sampled as well as the other parameters in the MCMC process ($fitnickelmass = 0$). Or their is an another method where it determinated separately from the tail (which part is mostly independent, more from this in the Important technical section)($fitnickelmass = 1$). This later method ($fitnickelmass = 1$) is recommended. **In this case the program must be runned TWICE!!** This is because at the first time, the program determinates the nickel mass (MNi) and gamma-leaking (A_g) from the tail. This takes about a 5 minute. It written out to *rand-niag.out* and *NiAgBest.txt*. When it finished, it must be runned again, when it will do the prime fitting, which last for about some hours! Note: if the nickel mass (MNi) and gamma-leaking (A_g) determined once, it saved to *NiAgBest.txt*. **If *NiAgBest.txt* exist and valid (validity is not checked!) then this nickel fitting skipped.**

The main parameter sampling is done between $tmin$ and $ttail$ (explanation later) this is χ^2_T (no tail), and the Ni tail fitting between $ttail + 5$ and $tmax$! χ^2_W (whole) between $tmin$ and $tmax$ also calculated. This gives two type of χ^2.
There are also SNe types which don't have tails, or at least it cannot be determinated easily. In this case you can simply give $ttail = tmax$! However if $ttail = tmax$ then the nickel mass MUST be sampled: fit nickel mass = 0. $fitnickelmass = 1$ still possible: you give a $ttail < tmax$ date when it is guaranteed to be a tail point before $tmax$, after the NiAg fitting done, you can set $ttail = tmax$.

If $fitnickelmass = 0$ and the Nickel mass is sampled, then bigger $tail$ must be added (or even $ttail = tmax$). In this case the Ni and A_g pair sampling does not happen, so the tail must also be used for the MCMC to get a the Ni mass values. This is useful for SNe where the tail may not exist, or hard to determinate. We can also see that the $ttail$ is more likely a technical parameter, and giving it a good value is important for good fits.

The last parameters in the input file are used for the other program which plots the parameter space, and calculates the confidence intervals. Leaving them blank make no effect to toe fitting.

4 Output

The output is the *rand.out* and *rand2.out* file (and *rand-niag.out* for the Ni-Ag fitting).
rand.out contains the accepted fits, and *rand2.out* contais those which are not accepted.
Fitted values; calculated values; other values which can also be fitted; χ^2

ID, $R0$ [cm], M [M_{sol}], Ek [erg], Eth [erg]; v [cm/s], $T0$ [K], MNi [M_{sol}], A_g [d^2]; $kappa$ [g/cm^2], s, $Tion$ [K]; $\sqrt{\chi^2_{norm,W}}$, $\sqrt{\chi^2_{norm,T}}$ [mag]; χ^2_W, χ^2_T; (whole and without tail)

$$\chi^2 = \Sigma_i^N (\frac{M_i - D_i}{\sigma_i})^2 \tag{1}$$

$$\chi^2_{norm} = \frac{\chi^2}{\Sigma_i^N \sigma_i^{-2}} \tag{2}$$

There are two type of $\sqrt{\chi^2_{norm}}$ and χ^2! The first type $\sqrt{\chi^2_{norm,W}}/\chi^2_W$ (whole) shows the whole (between $tmin$ and $tmax$), while the second $\sqrt{\chi^2_{norm,T}}/\chi^2_T$ only shows before the tail (until the plateu, between $tmin$ and $ttail$).

In the code: double $szoras = \sqrt{\chi^2_{norm,T}}$ (between $tmin$ and $ttail$), double $szorasf = \sqrt{\chi^2_{norm,W}}$ between $tmin$ and $tmax$, double $khinegyzet = \chi^2_T$ (between $tmin$ and $ttail$), double $khinegyzetf = \chi^2_W$ between $tmin$ and $tmax$, and double $tszora/tkhinegyzet$ is the previous accepted $\sqrt{\chi^2_{norm,T}}/\chi^2_T$ value (between $tmin$ and $ttail$).

$R0$; R_0 is the initial radius before the explosion
M; M_{ej} is the ejected mass (the star's initial mass is bigger then this)
Ek; E_{kin} is the kinetic energy
Eth; E_{th}; E is the thermal energy
v is the expansion velocity (maximum velocity)
$T0$; T_0 Temperature from thermal energy. **Note this is not equal with gamma-leaking. This parameter used in the code.**
MNi; M_{Ni} is the synthesized nickel mass
A_g; is the effective gamma-ray trapping
$kappa$; κ is the opacity
s is the Power-low density profile exponent
$Tion$; T_{ion} is the ionization temperature

In this doc the first form used, because this form used in the code, and this refers to the name of the used variant. However the second form used in the literature. (Exception: we use κ instead of kappa, and Eth instead of E.) The output file contains description of the columns at the beginning, and also contains the main parameters.

There know main parameter correlations: M, Ek, κ and the $R0$, Eth. $Ek(M)$ and $Eth(R0)$ should plotted, as a function of this, we should plot the scatters (3D plot or color contour plot). For example like on Fig 5, 6 and 8.

What is the difference between χ^2 and $\sqrt{\chi^2_{norm}}$? The χ^2 is used for further statistical work (determination the likelihood), while $\sqrt{\chi^2_{norm}}$ is the mean difference between the observed data and the model. It is given in magnitude. This is very useful to see manually if the fit is good enough. For example if $\sqrt{\chi^2_{norm}}$=0.01 mag it is quite good, but if it is 1 mag then the fit is bad (average observed data error is about some 0.01 mag). The output is quite big, 100MB, which may cause some problems. It should be arranged. We have an included program for this.

The output is written out per 100 lines to fasten the run time. Until that it is stored in the RAM. If the program is interrupted in any mean, the program attempts to write out the last line. So feel free to interrupt the program any time. However blackouts may cause glitched outputs. **The work can be continued!** Running the program again continues the work exactly there where previously stopped. However glitched last lines can be a problem. Because of this the program prints out to the standard output the last line. It should be checked if the line is not deficient. If so then stop the program, delete it in the output, and rerun. Please note that every rerun will use the actual parameter file. If it has been altered since the last run, then the fitting won't be consistent!

At running: If the $\sqrt{\chi^2_{norm}}$ is smaller then 0.5 mag, then there will be + in standard output. When the there are changing +, the program runs well. The standard output also contains the acceptance ratio: how many of the fits accepted in % (see MCMC section). The average calculated from the last 100 fit. The program calculates the luminosity in $\log_{10}$ of L in $erg/s/cm^2$. The conversion made at void meres(). The output is converted back to bolometric mag.

5 Important technical

If $fitnickelmass = 1$, we fit the tail first do determinate the best Ni and A_g pair (Fig 2), or in other words, the best Ni mass (MNi) values for given A_g. This function is in the *NiAgBest.txt* (calculated at the end of the file). In this case switch int $niag = 1$. Normal run: $niag = 0$. This is possible because the tail only depends on two of these. Fitting the light-curve manually is also start with this, we try to determinate nickel mass, however the nickel mass may change with different A_g values. The first fitting handle this. We know A_g depends on M and Ek, so with given M and Ek we can defines the A_g values. Then for this A_g we know the best Mni. This done between $ttail$ and $tmax$! We can also fit the MNi with MCMC; $fitnickelmass = 0$, but it gives back the same values, and reduce the effectiveness of the fitting by adding a plus parameter.

To do the fitting, we need: χ^2; goodness of the fitting. This value calculated with least square method. For the main fit, this calculated between $tmin$ and $ttail$ (no tail)! This does not mean that, the tail is not fitted, the χ^2 (whole) between $tmin$ and $tmax$ is also measured, but doesn't this used for the MCMC method. Sometime there are more, or more accurate data for the tail, which gives bigger weight for the tail, and this sometime dominates the fitting, and gives bad fits where the tail is fitted well, but the plateau is not. Example for this can be seen on Fig 3 and 4. We could lower the weight of the tail, but this is not simple, and not exact. **Fitting only the plateau ensures that the plateau is always well fitted, and because we know the best A_g MNi pair, the tail will also be quite good. But more importantly, because the ergodic nature of the MCMC, every good fit for the plateau will be found, including those which also fits the tail well, if it exist.** Because of this there two χ^2 output: the one which doesn't include the tail (χ^2_T, no tail), and the one which includes (χ^2_W, whole).

Fits with lowest χ^2_T and χ^2_W gives the best fits, which fits the plateau and the tail too. However this is not always exist. Well at least with some given parameters. Example for this can be seen on Fig 5 and 6. For the best tail fit there is always a maximum in the goodness of fit for A_g and MNi. Using this values gives always the best fits for the tail. However the A_g value is calculated: ([Clocchiatti, Wheeler, 1997]) (T_0 is not the initial temperature, it refers the notation used in [Clocchiatti, Wheeler, 1997])

$$A_g = T_0^2 = \frac{\kappa_\gamma \cdot M}{4 \cdot \Pi \cdot f \cdot v^2} \tag{3}$$

where f is geometric factor, g_1 is an integral (see [Nagy, Vinko, 2016] for more information), and

$$v^2 = \frac{2 \cdot E_{kin} \cdot f}{g_1 \cdot M} \tag{4}$$

If we arrange it, we will get:

$$E_{kin} = \frac{g_1 \cdot \kappa_\gamma}{8 \cdot \Pi \cdot f^2} \cdot \frac{M^2}{A_g} \tag{5}$$

So there is a main correlation between Ek and M which depends on the A_g. But the best fits from the plateau fitting not always in this region, sometime differs. When it occur the two cannot be fitted perfectly, there is no intersection, Fig 5 and 6. The resolution for this, is the change of other parameters, mainly the κ; opacity. κ correlates with the M. The most likely values are 0.2 and 0.3, however we can't rule out even the 0.4 and 0.05 values (In some types heavier elements dominate, so values lower then 0.2 are possible). Changing the opacity gives another region for the best values, and this may resolves the problem. You can see this on Fig 7 which is a good fit for both. Parameter space for this case is on Fig 8. However we used too low κ for this SN type which

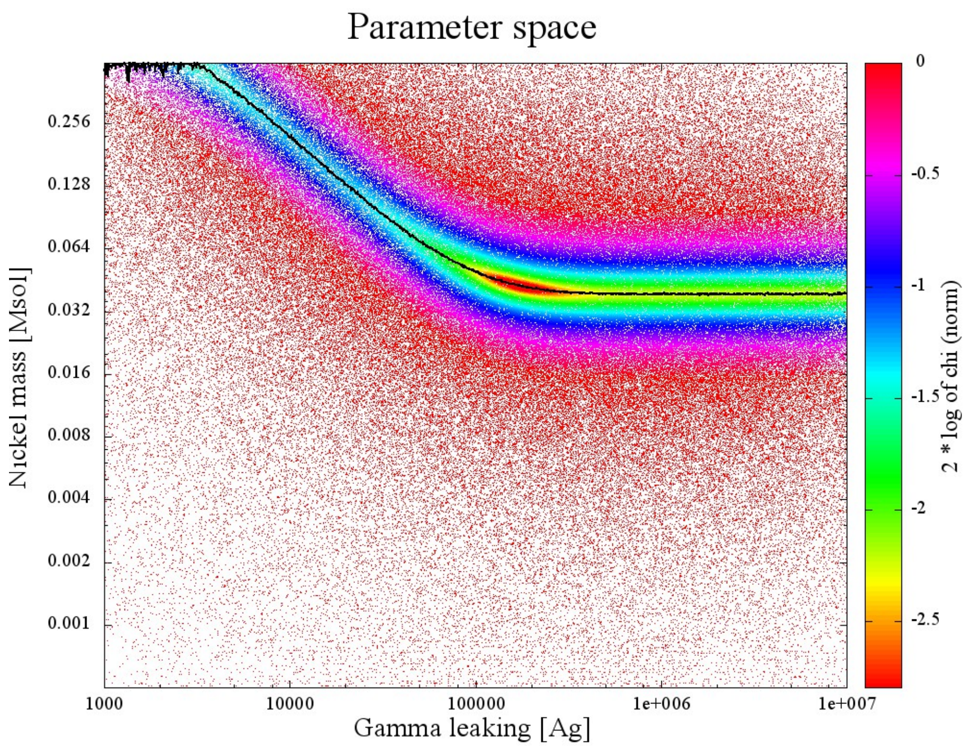

Figure 2: Ni(Ag) function for SN 2004et ([Sahu et al., 2006]). The black line is the fit. File: *NiAgBest.txt*

is not possible (type II implies hydrogen, so κ>0.2). So it is always recommended to use multiple values for the κ, or even sample it as well. **If the two output χ^2 is both good, the fitting for the whole light-curve is good too, so always examine both**. Sometime may not possible to fit them simultaneously, and there is no solutions for then in this code (hydrodynamical codes seems to resolve this).

6 Optimalisation

There are numerous optimization to make the runtime smaller per loop. The most important is the ionization radius (xi) searching which uses simplex algorithm to determinate where is the current radius within that the remnant is ionized (at xi, $T = Tion$). section double *temp*(){}. Integrals now use Simpson's rule instead of Trapezoid rule.

The luminosity is calculated from 3 component: L, *Lion*, *Lpoz*. *Lpoz* (positron) is fully analytical all through the process. *Lion* needs the ionization radius (xi), and the change of the ionization radius (dxi), which is calculated numerically. L is fully numerical at first. It calculated with a 4th grade Runge-Kutta method, in the section double *Frk*(){}. Here we can found the differential equation (see [Nagy et al., 2014] equation (12)).

$$\frac{d\phi(t)}{dt}\tau_{Ni} = \frac{R(t)}{R_0 \cdot x_\mathrm{i}^3} \cdot (p_1\zeta(t) - p_2 x_\mathrm{i} \cdot \phi(t) - 2\tau_{Ni} \cdot \phi(t) \cdot \frac{R_0}{R(t)}\frac{dx_\mathrm{i}}{dt}) \tag{6}$$

$\zeta(t)$ is the energy coming from the ^{56}Co and ^{56}Ni decay, and τ_{Ni} is the decay time of the nickel. τ_d is the diffusion timescale, p_1 and p_2 are constants (see [Nagy et al., 2014]).

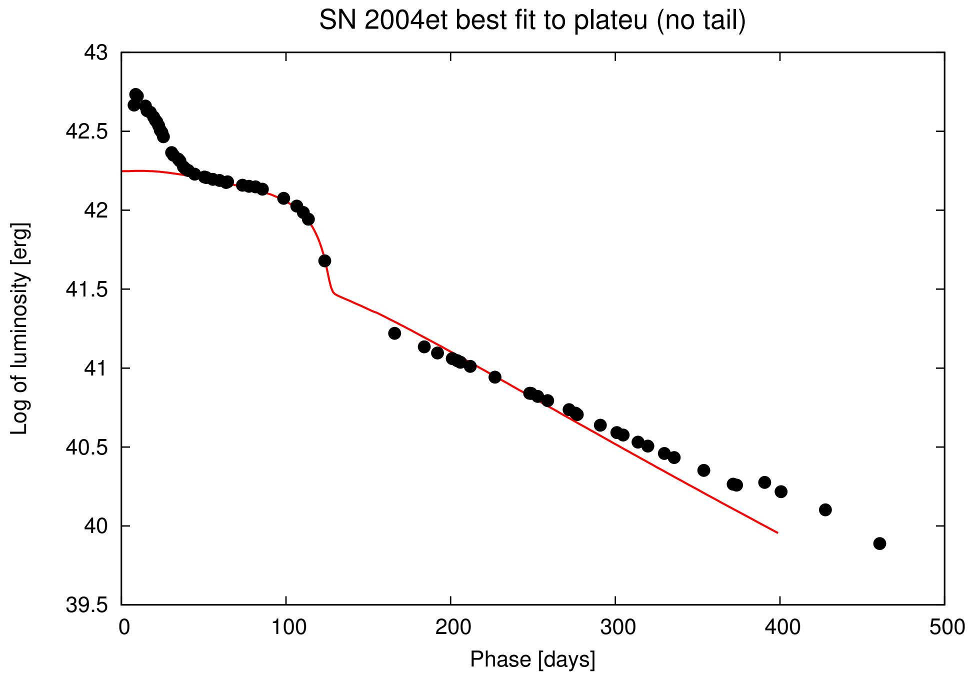

Figure 3: Best fits for SN 2004et ([Sahu et al., 2006]) using the $\sqrt{\chi^2_{norm,T}}$ (between *tmin* and *ttail*, without tail). The plateau is fitted well, however the tail not so well. $\kappa = 0.3$ cm^2/g.

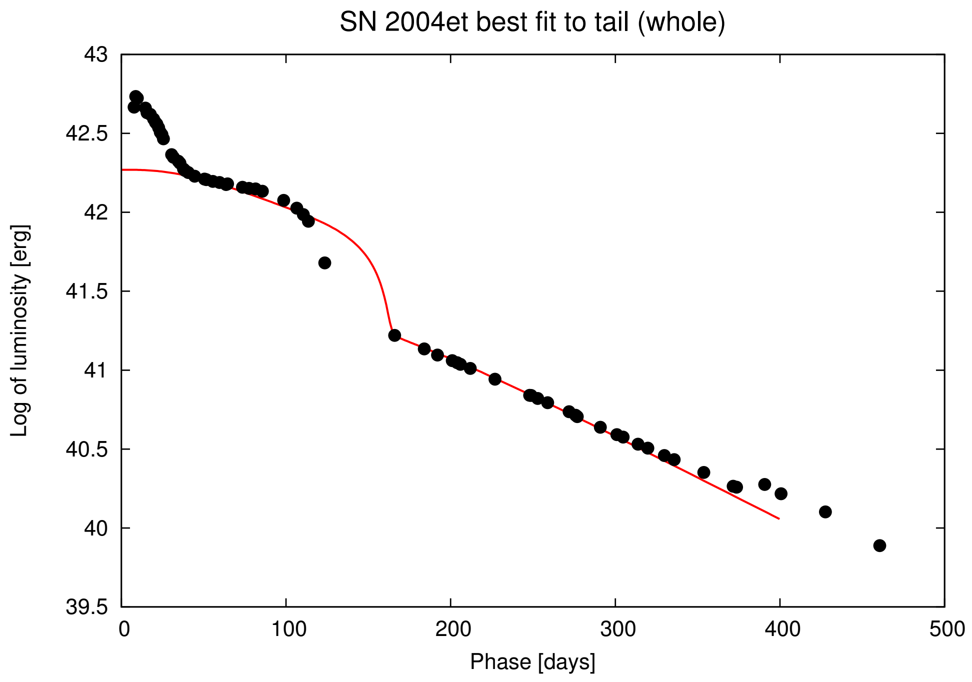

Figure 4: Best fits for SN 2004et ([Sahu et al., 2006]) using the $\sqrt{\chi^2_{norm,W}}$ (between *tmin* and *tmax*, whole). The tail is fitted well this time, but the plateau is not good. $\kappa = 0.3$ cm^2/g.

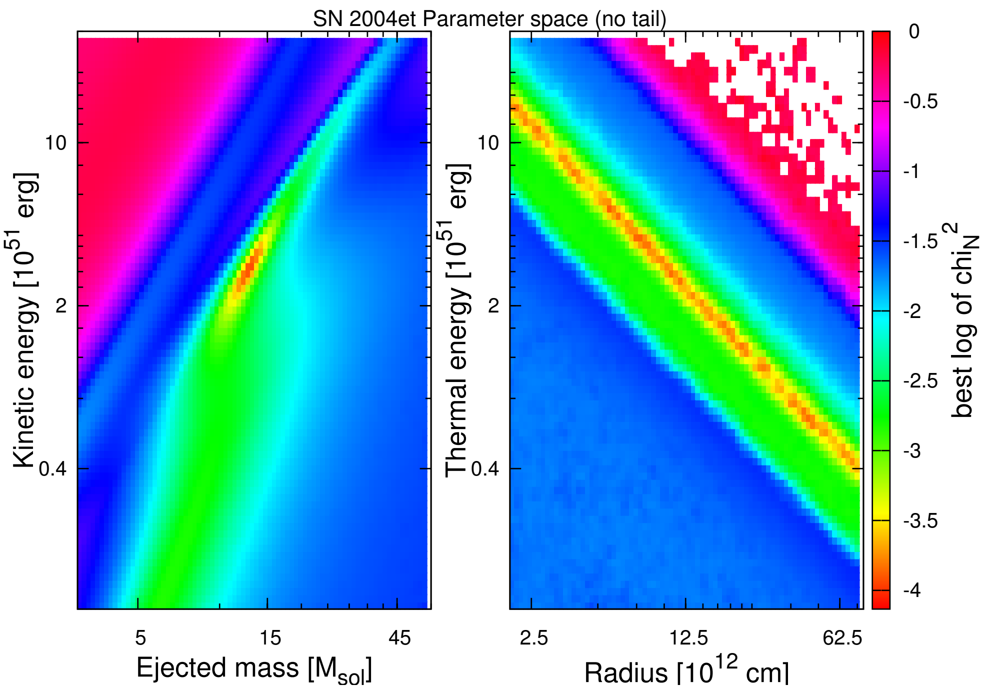

Figure 5: Parameter space for SN 2004et ([Sahu et al., 2006]) using the $\sqrt{\chi^2_{norm,T}}$ (between *tmin* and *ttail*, without tail). The case when the plateau is fitted well, however the tail not so well. $\kappa = 0.3$ cm^2/g. The color shows the $\sqrt{\chi^2_{norm,T}}$ (goodness of the fits).

$$L + L_{\mathrm{ion}} = x_{\mathrm{i}} \cdot \frac{\phi(t) E_{\mathrm{th}}(0)}{\tau_d} \cdot (1 - e^{-A_g/t^2}) + 4\pi r_{\mathrm{i}}^2 Q \cdot \rho(x_{\mathrm{i}}, t) \cdot \frac{dr_{\mathrm{i}}}{dt} \tag{7}$$

In this function xi, dxi, dF calculated every Runge-Kutte step, give back to the main program, then used again for the next Runge-Kutta step while increasing the time with a half step. So F and dxi also calculated with 4 grade Runge-Kutte, and this make the precision much bigger.

The numerical step has been made adaptive. The most precious part is still that part when ionization radius reaching it's minimum, so at this time the step is smaller. The formula is: after $x_{\mathrm{i}} = x_{\mathrm{min}}$; at the nebular phase it is 40000 sec. Before nebular phase: 20000 sec $\cdot f(x_{\mathrm{i}})$. $f(x_{\mathrm{i}}) = 1$ if $xi > 0.4$, else $f(x_{\mathrm{i}}) = 0.01 + 11 \cdot (x_{\mathrm{i}} - 0.1)^2$. Where xi is the current ionization zone, and given in comoving scale, so it's value can only be between 0 and 1. The formula is empirical. When the ionization zone reaches its minimum at the end of the nebular phase ($x_{\mathrm{i}} = x_{\mathrm{min}}$), it becomes constant, thus $dr = 0$; so does $L_{\mathrm{i}} = 0$, and L becomes analytic (energy coming from the decay chain of nickel and cobalt).
However there is a small (about 5-10 day) transient zone, which is still computed numerically, after this, it's become analytical. **Because of this, the first tail fitting (NiAg, $fitnickelmass = 1$) start at $ttail + 5$ till $tmax$. (Normal MCMC fitting $tmin$ to $ttail$, there is no plus). Keep this in mind if there are few points in the tail.** Fig 9 shows an example for the comoving ionization zone.

The minimum ionization zone (x_{min}, $xmin$) in theory is 0.1, but was raised to 0.2 for better performance. This does not affect the light curve significantly, but it speeds up the process.

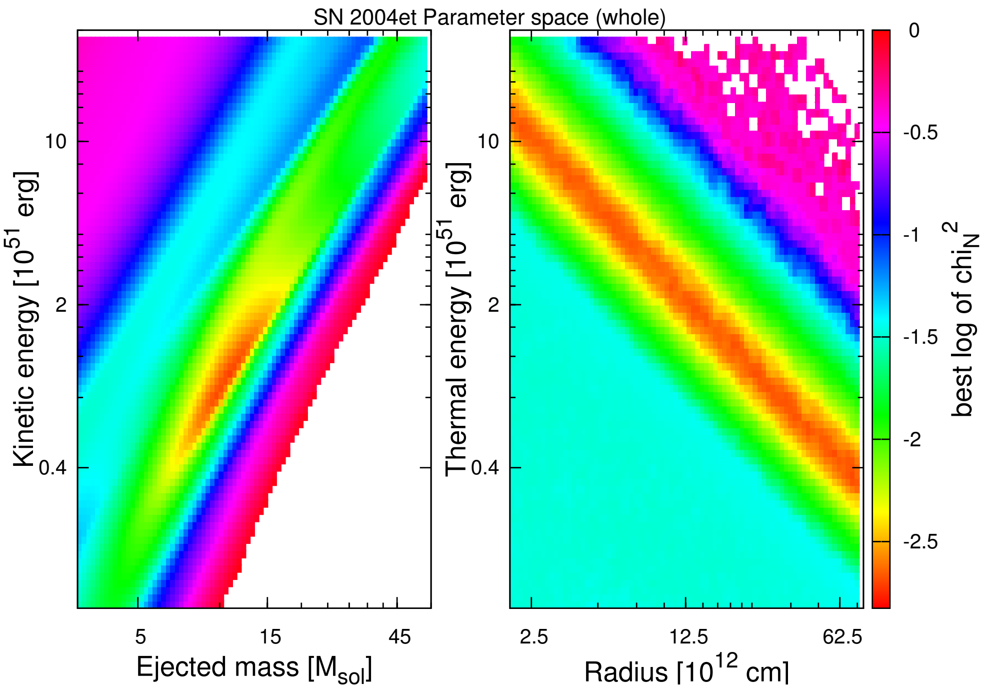

Figure 6: Parameter space for SN 2004et ([Sahu et al., 2006]) using the $\sqrt{\chi^2_{norm,W}}$ (between $tmin$ and $tmax$, whole). The case when the tail is fitted well, but the plateau is not. Note: this is different from the other. $\kappa = 0.3$ cm^2/g. The color shows $\sqrt{\chi^2_{norm,W}}$ (goodness of the fits).

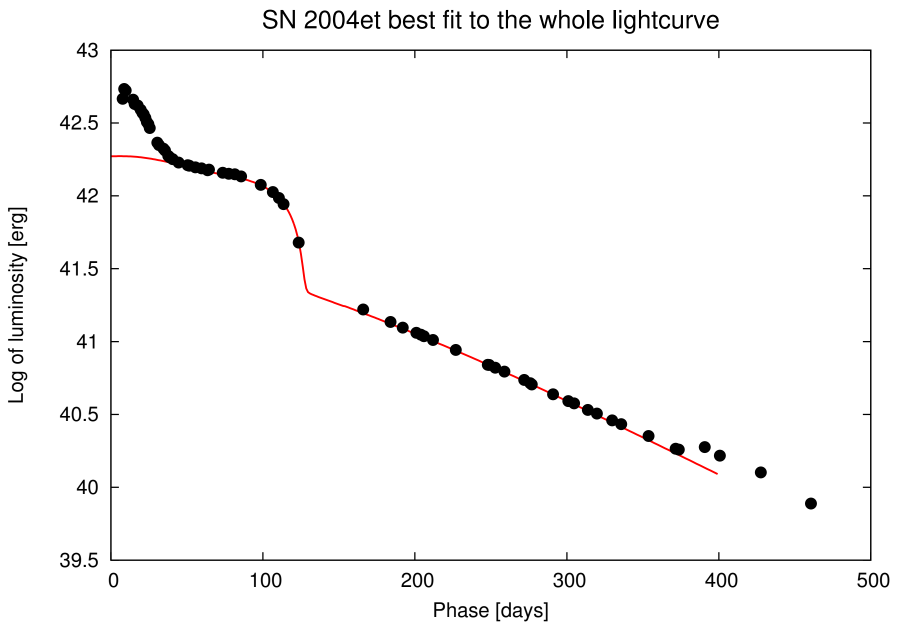

Figure 7: Best fits for SN 2004et ([Sahu et al., 2006]) using the $\sqrt{\chi^2_{norm,W}}$ (between $tmin$ and $tmax$, whole) if κ also fitted. The tail and the plateau are both fitted well this time. Best κ values about 0.1 cm^2/g. This value is too low for SN 2004et (Type II-P which implies H, but κ value is not)!

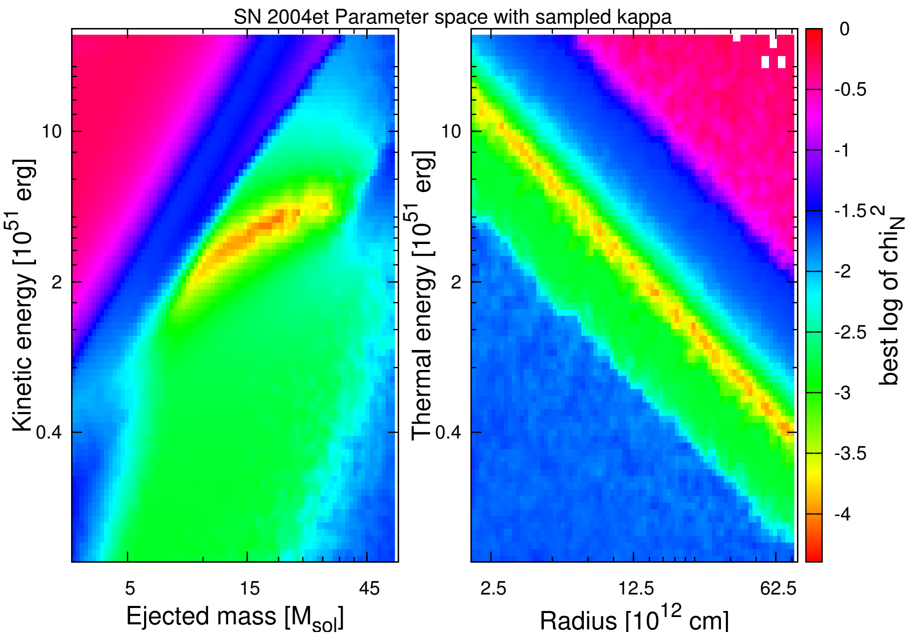

Figure 8: Parameter space for SN 2004et ([Sahu et al., 2006]) using the $\sqrt{\chi^2_{norm,W}}$ (between *tmin* and *tmax*, whole) if κ also fitted. The case when both the tail and the plateau are well. Best κ values about 0.1 cm^2/g. κ value is too low for SN 2004et (Type II-P which implies H, but κ value is not)! The color shows $\sqrt{\chi^2_{norm,W}}$ (goodness of the fits).

In addition if the $\sqrt{\chi^2_{norm}}$ is high before the end of the plateau, minimum ionization zone becomes $x_{\min} = 0.3$ to fasten the run time.

$$\chi^2_{norm} = \frac{\chi^2}{\Sigma_i^N \sigma_i^{-2}} \tag{8}$$

(standard deviation)

This applies when the fitting is bad, so the program does not attempt to calculate the end of the plateau with such precision, but when the fitting is good enough it has no effect. The criteria for this is:
if $\sqrt{\chi^2_{norm}}$ between *tmin* ($t_{\min}$, time of first fitted point) and $tmin + 50$ days < 0.5 mag, then $x_{\min} = 0.3$. So if the difference between *tmin* and the end of the plateau is smaller then 50 days, it has no effect, and 0.2 values applied. This also fastens the program, and it only has effect when the model and the light curve differ significantly. With these changes, there are no significant numerical errors.

7 Misc technical

The transient time; the time which is still computed numerically after the plateau is 10 day when $Ek < M^3/(4{\cdot}10^{50})$, else 0.25 day. This smooth this transient phase, because there is some numerical instabilities here.

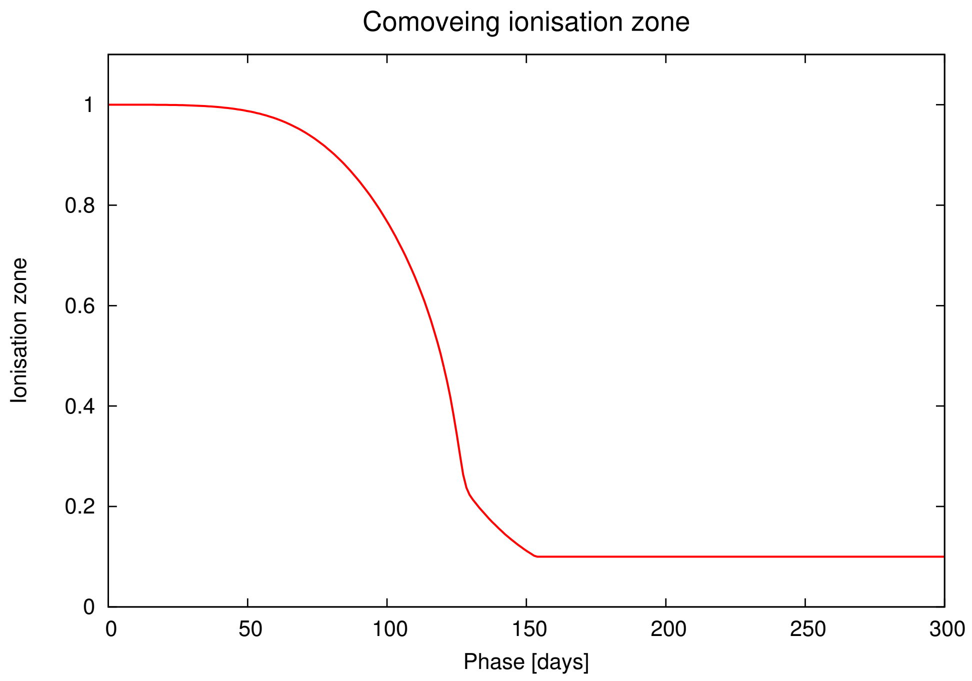

Figure 9: Example for the comoving ionization zone (best fit of SN 2005cs). In this example $xmin$ is 0.1.

8 MCMC

$R0$, M, Ek, Eth are randomly sampled. **The sample smooth in $\log_{10}$!** The ratio of maximum and minimum of the parameters are approx. 100. This means every parameter space reduced to approx. between 0 and 2. The random generator is at unsigned $rand256()\{\}$, which uses a 64 bit random generator, which makes multiple smaller random number then binary swift the random number next to each other making a bigger number. Then a random step used to jump to another random value by multiplying it with a random number (step). The distribution of the random step is Gaussian with zero mean, and a $Beta$ and $Delta$ dependent distribution;

$$distribution = Beta \cdot \sqrt{2.5 \cdot tszoras} \cdot (exp(-(n/Delta) + 1) + 1) \tag{9}$$

$Beta$, $Delta$ are the Monte Carlo parameters and they are empirical fix numbers, double $tszoras = \sqrt{\chi^2_{norm,T}}$ (not including the tail). $Beta$ default value: 0.1 (2 means the whole parameter space). n is the MCMC loop number and so $Delta$ defines loop dependent factor.

Default $Beta$ is the value where the sample's autocorrelation converge to 0 the fastest way (sample shifted with loop number). Too small step is not ergodic, too big step and the sample would jumps too often to low goodness of fit points (which will not be accepted, and the sample would stay on the same point too long). Using standard Gaussian is not enough because the sample would stack in local minimum, and would not sample the whole space.

So because of this the distribution depends on 2 parameters: $tszoras$, so better fits gives smaller steps, and worse fits ensure bigger steps, so its unlikely to stuck in a local minimum. In addition the step size also depends on the loop number. In the beginning (low loop number) the step is quite big: at start it is $e + 1$ times $Beta$. After $Delta$ loop steps it will 2 times $Beta$. Its converging to $Beta$. So in the beginning it is ensured to be ergodic, and do not stuck. This time (approx $Delta$ loops) can be considered as the burn in time. The Gaussian step is also altered.

After 2 σ it changes to an uniform distribution between 2 and 10 σ to increase the change of big steps. Note: If there are multi-modal pole very distant each other, the sample may stuck in one of them at the end. The other maximum also get some point, but fewer then it should. This can be easily spotted if the parameter space is plotted. This scenario is very unlikely for the SNe fit.

When we step to a new value, we calculate the χ^2 (χ^2_T: no tail) and more importantly the likelihood: $\mathcal{L} = \exp(-\chi^2/2)$. Better fits means bigger likelihood (perfect fit: likelihood=1). If the new random samples likelihood is better then the old, then the step is accepted, and this value will used to jump to another. However if new step is worse, then it may also be accepted: new likelihood/old likelihood > random number [0:1] (metropolis ratio).

$$\frac{\mathcal{L}_{new}}{\mathcal{L}_{old}} \geq \mathcal{U}[0:1] \tag{10}$$

If not accepted, the values still written off to the secondary output (*rand2.out*), but the old value used for the next jump and it written off again (to *rand.out*).
Note the $Ni(Ag)$ sampling uses $\sqrt{\chi^2_{norm}}$ for metropolis ratio, which samples the less likely fits too:

$$\sqrt{\frac{\chi^2_{norm,T_{old}}}{\chi^2_{norm,T_{new}}}} \geq \mathcal{U}[0:1] \tag{11}$$

When the new step is out the acceptable range, new jump generated.
Note: if bad acceptance region given in the parameter file (e.g. MIN>MAX), default values used.
Default prior or acceptable range:
(note: this values written out too if new MCMC parameter file generated.)

$R0$: $[2 \cdot 10^{12} : 80 \cdot 10^{12}]$ cm
M: $[3 : 60]$ M_{sol}
Ek: $[0.1 : 30]$ 10^{51} erg = foe
Eth: $[0.1 : 30]$ 10^{51} erg = foe
κ: $[0.05 : 0.4]$ cm^2/g
s: $[0 : 3]$
$Tion$: $[0 : 20000]$ K
Mni: $[5 \cdot 10^{-4} : 0.5]$ M_{sol}
A_g: $[1000 : 10^7]$ day^2

The first value is fixed for the Ni(Ag) and for the kappa, s, Tion.
For Ni(Ag) it is the geometric middle ($\sqrt{min \cdot max}$) for the rest it is the mean. The R0, M, Ek, Eth it is a random values between the prior/acceptable region. They are located in the void $data()\{\}$.

9 Uncertainties

Making a nice colorful plot about the parameter space is nice, and very visual, but not egzact. We need a mean values with errors. In MCMC we can do it with a simple method: calculate the mean of all points.
The fits with best *likelihood* are there where the χ^2 is minimal. The best fit is the most likely, so this is the mean value. The uncertainties can be driven from ([Hastings et al., 1970]):

$$C \cdot \int_{x_1}^{x_2} e^{-0.5 \cdot \chi^2(x)} \, dx = 0.95 \text{ or } 0.667 \tag{12}$$

$$e^{-0.5 \cdot \chi(x)^2} = \int \cdots \int e^{-0.5 \cdot \chi^2(x,y,\ldots,z)} \, dy \ldots dz = \int \cdots \int p(x, y, \ldots, z, I) \, dy \ldots dz \tag{13}$$

$$C^{-1} = \int \int \cdots \int e^{-0.5 \cdot \chi^2(x,y,\ldots,z)} \, dx \, dy \ldots dz \tag{14}$$

$$p(x, y, \ldots, z, I) = e^{-0.5 \cdot \chi^2(x,y,\ldots,z)} \tag{15}$$

Where $(x, y, \ldots, z$ represents the physical parameters (M_{ej} for example), $\int = \int_0^\infty$, and C is the normalization factor. We search for the smallest $x_2 - x_1$ solution, which gives the lower (x_1) error value and upper (x_2) error value. Value 0.95 is for 2 σ, and 0.667 is for 1 σ. N is the number of samples. Note: $e^{-0.5 \cdot \chi^2(x,y,\ldots,z)}$ is the likelihood, but because we use uniform priors, it is also equal with the posterior $p(x, y, \ldots, z, I)$.

In the MCMC the likelihood correlates with the sample number, so this simplifies the calculation, as only the sample number needs to be plotted, and the sample mean (x_0), standard deviation (σ_{xx}), measure of correlation matrix (r_{xy}) and confidence intervals (x_1, x_2) can be computed by simple summation over the chain elements:

$$N^{-1} \cdot \sum_{x_1}^{x_2} histrogram(x) = 0.95 \text{ or } 0.667 \tag{16}$$

$$histogram(x) = \sum_{i=0}^{N} 1 \ \text{ if}(x_i = x) \tag{17}$$

Reminder: x is the actual physical parameter e.g. mass. In other words: The histogram gives how many fits are in between x and x+dx. Eq. (12) = Eq. (16) in the MCMC.
The correlation matrix: (x_0 is the average mean)

$$x_0 = N^{-1} \cdot \sum_{i=0}^{N} x_i \tag{18}$$

$$\sigma_{xx}^2 = N^{-1} \cdot \sum_{i=0}^{N} (x_i - x_0)^2 \tag{19}$$

$$\sigma_{xy}^2 = N^{-1} \cdot \sum_{i=0}^{N} (x_i - x_0) \cdot (y_i - y_0) \tag{20}$$

$$r_{xy} = \frac{\sigma_{xy}^2}{\sqrt{\sigma_{xx}^2 \cdot \sigma_{yy}^2}} \tag{21}$$

Burn in: Because the first step is fully random, it takes some time (steps) to reach that region where the fitting is quite good. Because of the nature of the MCMC, only those accepted which has high likelihood. So most of the accepted parameters are good fits. But at the beginning there will be bad fits too until it find the first good fit. This is the burn in.

10 Other programs

LC3p3.cpp

This program is the light curve modeling program. It creates the light curve from the given parameters. It uses the same methods as the MCMC, so you can use it to check light-curves. However the MCMC coed does not uses this. Altering this code wont alter the other. Fig 3, 4, and 7 made with this. There are 2 main switch in the code:
int $fastrun = 1$;
int $csekk = 0$;
$fastrun = 1$ use the same values as the MCMC. If it's value is 0, then the program runs a bit slower (about 1-2 sec) but make a more precise output. There are no significant difference between the two.
$csekk$ defines the input.
Fastrun must be altered in the code. However csekk controlled with command line argument. So altering it in the code has no effect.

Input:

int $csekk = 0$;
Default running.
file = "parameters.inp"
$R0$; Initial radius [cm]
M; Ejecta mass [M_{sol}]
$Tion$; Ionization/recombination temperature [K]
Mni; Initial nickel mass [M_{sol}]
Ek; Initial kinetic energy [10^{51} erg]
Eth; Initial thermal energy [10^{51} erg]
a; Exponential density profile exponent
s; Power-low density profile exponent
κ; Thomson scattering opacity [cm^2/g]
Ep; Initial magnetar rotational energy [10^{51} erg]
tp; Characteristic time scale of magnetar spin-down [d]
A_g1; Comparison Gamma-leak [day^2]. Currently unused. The used A_g is calculated in the code!
$tmax$; Final epoch [day]

Comments (#) after the number is possible, so by default the parameter file contains a label after a value. **The order is fixed, don't change. (Same as with the MCMC input.)** If the input parameter files are not given, a new default is generated. Use this as a base.

int $csekk = 1$;
run: add any character (does not matter which character) after the compiled file, like:
LC3p3.exe 1
files = "*parametersMC.inp*" and "*par.inp*"
This mode is for to check the output of the MCMC values. Use the same "*parametersMC.inp*" as you used for the MCMC fitting. Copy the row from *rand.out* which you want to be checked to the "par.inp", then you can run it.

Output:

The output includes the input parameters. The values for the light-curve:
time [day], Lsn: full energy [erg], logLsn: log full energy [log of erg], MAG: bolometric magnitude [mag],
$Ldiff$:energy without ionization and positron ($Lsn - Lion - Lpoz$) [erg], $Lion$: energy from ionization [erg], xi: ionization radius [comoving frame], F [comoving frame]

SpaceMax.cpp

This program help manage the big output file. It reads in the output file of the MCMC (*rand.out*). This is optional, and the output can be handled with other programs, but it compatible with the output, and easy to run. You should try it at least once. It basically count the means and confidence intervals using the equations in section Uncertainties.

What is this program do:
Best fits in order.
After running this program, it can create a plot containing the light-curve, the best fit and m accepted level 0.95 fit (randomly chosen).
Calculated means, standard dev., best value, confidence intervals, and correlation matrix.
Nice colorful plot to see the parameter space.
Fig 5, 6, 8 and 10 made with this.

It also reads in the parameterMC input file for the plots to be in sync with the fitting. If there is no input file, default parameters used. **The 3 last parameter used in the 'parameterMC' are used for this program, and only this must be altered!**

This 3 parameter:
int *whole*:
Which χ^2 and $\sqrt{\chi^2_{norm}}$ type used? 1 for χ^2_W (whole, between *tmin* and *tmax*), 0 for χ^2_T (without tail; between tmin and ttail) Both whole 0 and 1 should be runned. The output won't overwrite each other, so it can be runned safety ofter other.
int *grid*:
The grid number used for the plotting and griding.
int *burnin*:
Burn in number in step numbers. The files reads in the output lines. This many lines ignored, so it won't be contained in the mean for example.

You can also set in the code (optional):
int n How many best values write out? default 250. Giving too big values (100 000-1 000 000) may cause crash.
it m How many fits be plotted? Default 50 (note: some of them may overlap).

Parameter space plotting:
To plot $Ek(M)$ and $Eth(R0)$ functions we need to marginalize it (make a histogram):

$$histogram(x, y) = \sum_{i=0}^{N} 1 \ \text{ if}(x_i = x \ \& \ y_i = y) \tag{22}$$

The $\sum$ is discrete, but $x_i = x$ is not. We need an interval; we accept points between $x, x+dx$. So we need to grid the parameter space. This is the role of the variable *grid*. $dx = (x_{max} - x_{min})/grid$. The grid is the same for all parameter. However every parameter plotted in log, so the max range does not change significantly. But because of this everything must be done in $\log_{10}$! **So in the code x_i refers to $log_{10}(x_i)$.** After this you can run the gnuplot script which gives nice plots. Example: Fig. 10 This plot will also contain the contour of the confidence intervals.

The outputs:
The output gives processed and reduced datafiles, gnuplot scripts which can be used plot them (best

fits and the parameter space), and a mean.log to see the mean values, and it's errors/confidence intervals. The gnu plot scrips are recommended to be checked, altered if you want a different plot. Be advised to rename them, as if you alter the file, and run the program again, it will be overwritten! (The mean.log is an exception which only append, so always check the last.)

output files: 2 type: W and T. W for whole, T for no tail (χ^2 types, see below) int *whole* defines it. Here we write the W type only.

SpaceBestW.txt (or *SpaceBestT.txt*)
SpaceBestW2.txt
SpaceALLW.txt
arrenged data files for the parameter space plot, gridded for $Ek(M)$ (SpaceBest) and $Eth(R0)$ (SpaceBest2), and everything with everything (SpaceALL)
spaceW.gp
spaceAllW.gp
the gnuplot script for parameters space. First plots only $Ek(M)$ and $Eth(R0)$, while the second plots every parameter.
SpaceMeanW.txt
1 Dimension marginalisation of the 4 main parameter

bestW.txt
data file which contains the best fits in order (χ^2 type dependent). Note: most of the overlaps if the acceptance ratio was low. Uses the same structure as *rand.out*!
brandW.txt
data file which contains the n randomly chosen level 0.95 fit. Note: may contain less then n, not a problem is $n > m$. Uses the same structure as *rand.out*!
bestbatW.bat
bestscriptW.sh
Script for windows and linux to calculate the best and the choosen light-curves with the light curve program (LC3p3). LC3p3 must be compiled for windows (and should be checked for linux too)
bestW.gp
gnuplot script to plot the calculate best fits. Run after bestbatW.bat/bestscriptW.sh done
mean.log
log for the calculated means and errors. This is the results.

gnuplot outputs:
bestW.eps
Output for bestW.gp *spaceW.eps*
spaceWl.eps
spaceWn.eps
Outputs for spaceW.gp. spaceWl: plots the likelihood, spaceWn: plots the distribution of points (how many in one grid), because of the nature of the MCMC this two are identical (but spaceWl normalised). spaceW: plots the best of the $\ln(\sqrt{\chi^2_{norm}})$ [mag] in the grid.
spaceALLW.eps
Output for spaceALLW.gp. Note: every script crates JPG too, expect spaceALLW.gp which only crates JPG or EPS.

Technical:
The outputs read in 3 times! First it reads in *rand.out* to get and store the best fits. The marginal-

ization for both 1D and 2D also done here.
Second it reads in *rand2.out* too, which contains the not accepted bads fits to extend the data. Used for spaceW.eps.
Third time reads in *rand.out* again. At this time randomly selects the 0.95 confidence level accepted fits and stores it. Note: for one line there is an n/N change to select. This means that not always n selected. If more, then any $> n$ are not chosen.
Further technical info commented in the code.

11 Download

Our code can be downloaded from here:
`https://github.com/Hydralisk24/Science/tree/master/SN-LC-MCMC`
Contact: jagerz24@gmail.com

12 Version 2

Version 1 is the tested version used in [Jager et al., 2020]. It does not include the magnetar fitting. In Version 2 the magnetar fitting is also included. In addition the product $ER0 = Eth \cdot R_0$ used insted of $R0$ Eth due to the high correlation. Important note: NiAg fitting incompatible with th magnetar!

The code updated to LC3p4.cpp, fixing an issue with $E_{magnetar} \neq 0$ cases. In addition header being used.

Because of this the input and output has been altered, and so incompatible with the old one! *szoraskuszob* removed. If magnetar values not zero then NiAg function fitting set to zero, and Ni mass MCMC sampled too.

The MCMC parameter file *parametersMC.inp* readin has been removed to *Fejlec.cpp*. This new file has been included in every file, so the readin process is uniform in every code. (Note some variable declaration also removed to this file).

References

[Arnett & Fu, 1989] Arnett, W. D., Fu, A., 1989, ApJ, 340, 396

[Clocchiatti, Wheeler, 1997] Clocchiatti, A.; Wheeler, J. C., 1997, ApJ, 491, 375C

[Jager et al., 2020] Jäger, Zoltán, Jr.; Vinkó, József; Bíró, Barna I.; Hegedüs, Tibor; Borkovits, Tamás; Jäger, Zoltán, Sr.; Nagy, Andrea P.; Molnár, László; Kriskovics, Levente 2020, MNRAS, 496, 3725J

[Lyman et al., 2014] Lyman, J. D., Bersier, D., James, P. A., 2014, MNRAS, 437, 3848L

[Gilks et al., 1996] Gilks W. R., Richardson S., and Spiegelhalter D. J., Markov Chain Monte Carlo in Practice (Chapman & Hall: London, 1996)

[Hastings et al., 1970] Hastings W. K., Biometrika, 57, 97 (1970)

[Nagy et al., 2014] Nagy, A. P., Ordasi, A., Vinkó, J., Wheeler, J. C., 2014, A&A, 571, A77

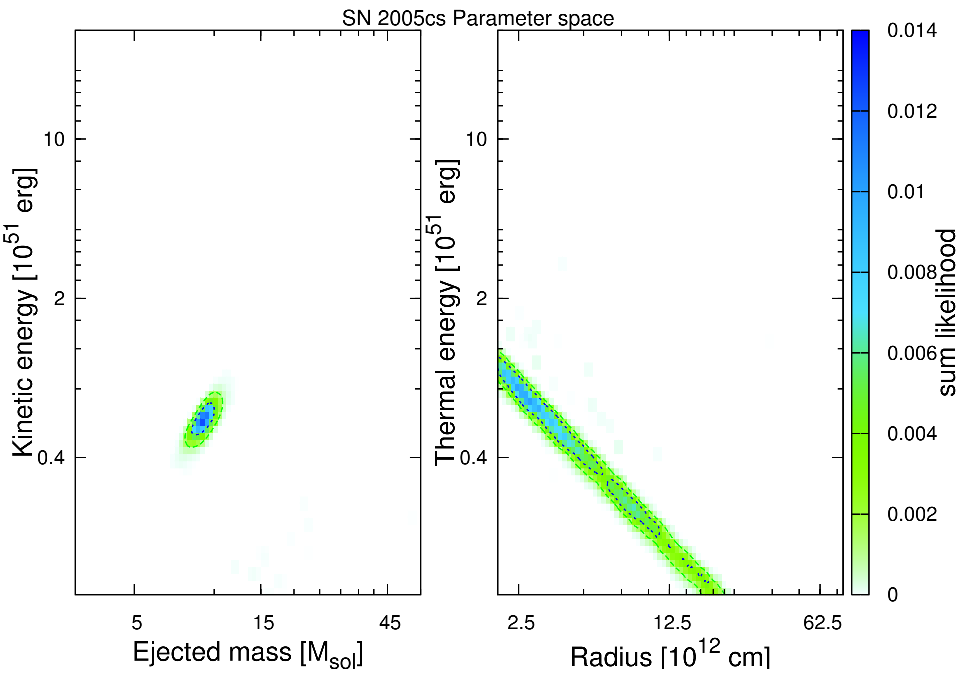

Figure 10: SN 2005cs ([Pastorello et al., 2009]) parameter space, $s = 0$, $\kappa = [0.2 : 0.3]$ cm^2/g. The colors encode the goodness of the fits; Likelihood. The contour shows the confidence interval of 67% and 95%.

[Nagy, Vinko, 2016] Nagy, A. P., Vinkó, J., 2016, A&A, 589, A53

[Metropolis et al., 1953] Metropolis N., Rosenbluth A. W., Rosenbluth M. N., Teller A. H., and Teller E., J.Chem.Phys., 21, 1087 (1953)

[Pastorello et al., 2009] Pastorello, A., Valenti, S., Zampieri, L., Navasardyan, H., Taubenberger, S., Smartt, S. J., Arkharov, A. A., Bärnbantner, O., Barwig, H., Benetti, S., Birtwhistle, P., Botticella, M. T., Cappellaro, E., Del Principe, M., di Mille, F., di Rico, G., Dolci, M., Elias-Rosa, N., Efimova, N. V., Fiedler, M., Harutyunyan, A., Höflich, P. A., Kloehr, W., Larionov, V. M., Lorenzi, V., Maund, J. R., Napoleone, N., Ragni, M., Richmond, M., Ries, C., Spiro, S., Temporin, S., Turatto, M., Wheeler, J. C., 2009, MNRAS, 394, 2266P

[Sahu et al., 2006] Sahu, D. K., Anupama, G. C., Srividya, S., Muneer, S., 2006, MNRAS, 372, 1315S