A Search for High-Energy Neutrino Emission from Gamma-Ray Bursts

By

Nathan Whitehorn

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Physics)

at the
UNIVERSITY OF WISCONSIN–MADISON
2012

Date of final oral examination: April 16, 2012

The dissertation is approved by the following members of the Final Oral Committee:

Francis Halzen, Professor, Physics
Sebastian Heinz, Associate Professor, Astronomy
Matthew Herndon, Associate Professor, Physics
Albrecht Karle, Professor, Physics
Stefan Westerhoff, Associate Professor, Physics

A SEARCH FOR HIGH-ENERGY NEUTRINO EMISSION FROM GAMMA-RAY BURSTS

Nathan Whitehorn

Under the supervision of Professor Francis Halzen
At the University of Wisconsin-Madison

A century after their discovery, the origin of cosmic rays remains one of the most enduring mysteries in physics. They can have energies that exceed 10^{20} eV, a hundred million times as energetic as the most powerful Earth-bound particle accelerators and must therefore be produced in the universe's most violent environments. Direct observation of their origins, however, has proven difficult due to deflection of charged cosmic ray particles in galactic and intergalactic magnetic fields, obscuring their true origins. Astronomy using electrically neutral particles, such as photons and neutrinos, does not, however, share this difficulty.

This work presents a search for neutrino emission from one of the primary candidates for the sources of the highest-energy cosmic rays, Gamma-Ray Bursts, using the recently-completed IceCube neutrino telescope located at the South Pole. The null result obtained from this search contradicts well-established predictions for the neutrino flux from Gamma-Ray Bursts if they are the cosmic ray sources, forcing a reevaluation of these theoretical models.

In the course of this work, a variety of new analysis techniques and reconstructions were developed. These are described in sections 3.4.1, 4.3, 4.4, 5.1, and 5.3.

Francis Halzen

ACKNOWLEDGMENTS

This work would not have been possible without the help of a large number of people. First, I would like to thank Francis Halzen, for suggesting this extremely rewarding topic and his time, energy, and understanding in helping me see it through to completion. I would also like to thank Kevin Meagher, Pete Redl, Erik Blaufuss, Markus Ahlers, and Juanan Aguilar for their assistance, collaboration, and encouragement at all points.

Particular thanks are also due to Jakob van Santen, Sven LaFebre, and Eike Middell for their help and perseverance with the incredibly daunting Photospline project. I am also grateful to everyone in Madison for their friendliness and encouragement throughout.

Also instrumental was Ed Blucher, who allowed me an opportunity to experience neutrino physics, and to my parents, Mary Beth Saffo and Erik Whitehorn: to my father for first piquing my interest in neutrinos with a newspaper clipping when I was in 6th grade and to my mother for giving me her lifelong love of science and learning.

And for everything I thank Elizabeth, who makes all things worthwhile.

DISCARD THIS PAGE

TABLE OF CONTENTS

Page

LIST OF TABLES . . . iv

LIST OF FIGURES . . . v

1 Introduction . . . 1

2 GRBs . . . 5

2.1 Observations . . . 5
2.2 Cosmic Ray Acceleration . . . 9
2.3 Neutrino Production . . . 10
2.4 Model Variants . . . 15

3 IceCube Detector . . . 17

3.1 Detection Principle . . . 17
3.2 Digital Optical Modules . . . 18
3.3 Data Acquisition and Triggering . . . 22
3.4 Pulse extraction . . . 24
3.4.1 Wavedeform . . . 24
3.5 Optical Properties of the Ice . . . 28

4 Reconstruction . . . 32

4.1 Angular reconstruction . . . 32
4.1.1 Linefit . . . 32
4.1.2 Likelihood fits . . . 34
4.2 Energy reconstruction . . . 35
4.3 Photospline . . . 36
4.4 Millipede . . . 38

5 Analysis . . . 41

5.1 Time Windows: what does simultaneous mean? . . . 42
5.2 Backgrounds . . . 43
5.3 Event Selection . . . 44
5.4 Performance . . . 50

6 Results . . . 55

LIST OF REFERENCES . . . 62

Page

APPENDIX Photospline 67

APPENDIX Burst Catalogs 101

DISCARD THIS PAGE

LIST OF TABLES

Table Page

3.1 Triggers used during the IceCube-59 physics run 24

4.1 Angular resolution of Millipede 40

5.1 Predicted event rates for various theoretical models in IC40+IC59 41

DISCARD THIS PAGE

LIST OF FIGURES

Figure Page

1.1 Cosmic Ray Spectrum . . . 2

1.2 The IceCube detector . . . 4

2.1 GRB Redshifts . . . 6

2.2 Distribution of burst lengths . . . 7

2.3 Light curve of GRB091123 . . . 8

2.4 First-Order Fermi Acceleration . . . 11

2.5 Hillas Plot . . . 12

2.6 Waxman-Bahcall spectrum . . . 14

3.1 Example Neutrino Events . . . 19

3.2 Muon range in ice . . . 20

3.3 Angle between neutrino and muon . . . 21

3.4 Digital Optical Module . . . 22

3.5 Digital Optical Module Mainboard Block Diagram . . . 23

3.6 DOM digitizer shaping functions . . . 25

3.7 Ringing in Unfolding Problems . . . 26

3.8 Examples of waveform unfolding with Wavedeform . . . 27

3.9 Wavedeform performance . . . 29

3.10 Optical properties of the South Pole ice . . . 30

3.11 Photograph of the South Pole ice . . . 31

4.1 Muon energy loss in copper . . . 33

4.2 Pandel Approximation . . . 36

Figure Page

4.3 Effects of binning on expected photon densities . . . 37
4.4 Millipede dE/dx reconstruction . . . 40
5.1 Schematic of Expanding Time Windows . . . 42
5.2 Effect of expanding the time window . . . 43
5.3 Coincident Muons . . . 45
5.4 Example of spline fit in event classification . . . 47
5.5 Overview of performance of event selection . . . 48
5.6 Overview of performance of event selection in IC59 . . . 49
5.7 Comparison of IceCube effective area in this approach and in other IceCube analyses . . . 51
5.8 Effective area of IceCube to Northern and Southern Sources . . . 52
5.9 Neutrino-Nucleon Cross-Sections . . . 53
6.1 IC40 final events . . . 56
6.2 IC59 final events . . . 56
6.3 Limits on E^{-2} fluxes . . . 57
6.4 Limits on Waxman-Bahcall type fluxes . . . 58
6.5 Allowed Proton Flux in Ahlers *et al.* model . . . 60

Appendix
Figure

A.1 B-spline functions . . . 69
A.2 DeBoor recurrence relation . . . 70
A.3 Tensor product spline . . . 71
A.4 Illustration of B-spline pseudointerpolation . . . 79
A.5 Example of ringing in spline fits . . . 95
A.6 B- and T-spline bases of various orders. . . . 96
A.7 Performance of BLOCK3 algorithm . . . 97
A.8 Resource consumption of Photospline fits . . . 98

Figure Page

A.9 Sampling photon time delays from spline tables 99

A.10 Example of spline convolution 100

Chapter 1

Introduction

The origin of cosmic rays is one of the most enduring mysteries of modern physics. Cosmic rays were first discovered in 1912, with the balloon flights of Viktor Hess. Using an electroscope, he discovered that the ionizing radiation observed at the surface increased with altitude, proving that it had an extraterrestrial origin [1]. The last century of research has answered many questions about the properties and origins of this radiation: the cosmic rays are now known to consist overwhelmingly of charged hadrons; the lowest energy particles, at energies below 10^9 eV, are predominantly protons and have their origin in the outer layers of the sun. The cosmic ray spectrum, however, extends to much higher energies than can be explained by the sun alone (Fig. 1.1), and the origin of these particles remains a mystery. The very highest energy observed particles have energies of more than 10^{20} eV, a hundred million times higher than can be produced in the most powerful accelerators. Although these high energies imply the particles are accelerated in the most energetic and violent astrophysical environments, their sources remain unknown.

Direct measurements of the origin of these particles are challenging. Since they are dominantly charged particles, their paths are deflected by galactic and extragalactic magnetic fields, effectively scrambling their arrival directions. Only at the very highest energies ($> 10^{20}$ eV), where particle momentum is high and the bending effect of magnetic fields low, is it possible to effectively determine particle origins, and even then only if they are from close enough ($\lesssim 100$ Mpc) that deflections are minimized. The rate of such particles is however low, limiting statistics and requiring enormous detectors (e.g. the Pierre Auger Observatory). Such high energy protons also cannot travel far: interactions with the cosmic microwave background [2, 3] cause them to lose energy in flight.

Indirect detection methods using neutral particles solve this problem, while introducing others. Gamma-ray astronomy using space-borne (e.g. SWIFT, Fermi) and ground-based (e.g. HESS, MAGIC, Veritas, Milagro) telescopes is a well-understood and mature technique with high efficiencies, but gamma-ray signatures of cosmic ray acceleration are ambiguous and high-energy gamma rays are easily stopped by interactions

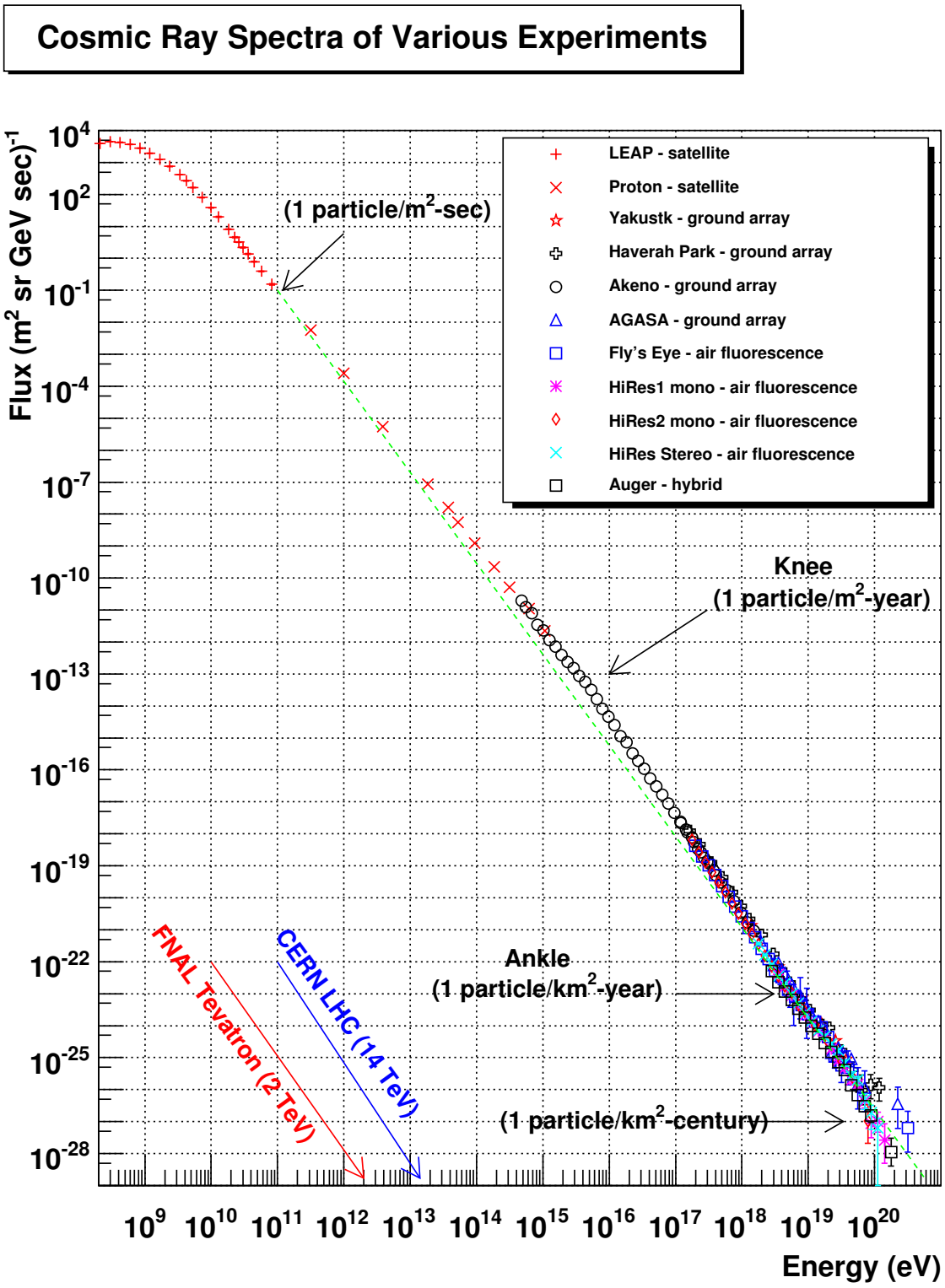


Figure 1.1 Spectrum of cosmic rays observed in several experiments from 10^8 to 10^{21} eV. Particles above the ankle are believed to be produced in extragalactic sources and are the main focus of this work. Figure from William Hanlon, University of Utah.

with interstellar dust and background light. The clearest signature of hadronic acceleration is the production of neutral pions in interactions between protons being accelerated and their surrounding environment, for example:

$$p + \gamma \rightarrow p + \pi^0 \rightarrow p + \gamma + \gamma \tag{1.1}$$

This can be detected in principle by observing the characteristic spectral features of the $\pi^0 \rightarrow 2\gamma$ decay. Distinguishing this from gamma-ray emission from inverse Compton scattering and synchrotron emission by high energy electrons is difficult, however, complicating any attempt to identify an object as a cosmic ray source. Moreover, if this process occurs in dense material, the high-energy gamma rays will interact before escaping and cascade to lower energies, further obscuring the signal.

A more promising avenue is indirect detection using neutrinos. Like photons, they are electrically neutral and so travel in straight lines. Unlike photons, they are produced predominantly in hadronic interactions, making observation of high energy neutrinos a clear signature of acceleration of high-energy hadrons. Neutrinos would be produced in a similar manner to pionic gamma rays, for example:

$$p + \gamma \rightarrow n + \pi^+ \rightarrow n + \mu^+ + \nu_\mu \tag{1.2}$$

Neutrinos introduce their own problems, however. Their low interaction rates, which help them escape even from dense environments, make them extremely difficult to detect. Given the expected low fluxes of these high-energy neutrinos, their detection requires enormous detectors, on the scale of cubic kilometers.

The IceCube detector [4] (Fig. 1.2), located at the geographic South Pole, was designed to address these issues. Made of 5160 photomultipliers, it uses over 1 km^3 of glacial ice as a detection medium, making it sensitive to even relatively low fluxes.

Among the most promising targets for observation with IceCube are Gamma-Ray Bursts (Chap. 2). These have long been proposed as an origin of the very highest energy cosmic rays [5, 6], and are themselves something of a mystery. Shining for only a few seconds, they can have luminosities exceeding the combined light output of all other sources in the universe and can be seen at extraordinary distances. The processes that cause them are not known at present (though at least some GRBs are associated with supernovae [7]) and the observation of neutrinos would go a great way toward enhancing our understanding. The connection between cosmic rays and neutrinos from GRBs is also particularly direct (Sec. 2.3), making even a null observation a potentially powerful statement on the GRB/cosmic-ray hypothesis. Further, the brief duration of the burst significantly reduces background and makes analysis straightforward (Chap. 5).

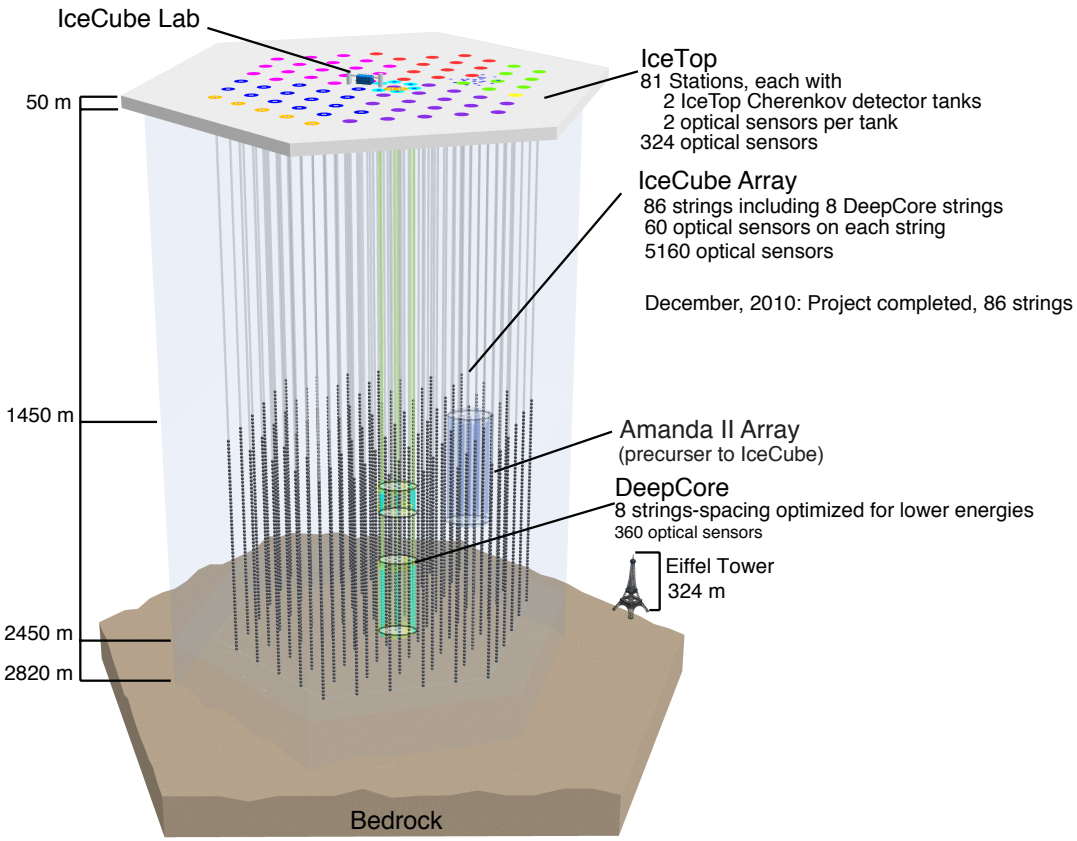


Figure 1.2 The IceCube detector

Chapter 2

GRBs

Gamma-ray bursts (GRBs), first discovered in the 1960s by arms control monitoring satellites [8], are among the most energetic objects in the universe, with apparent gamma-ray emission in excess of 10^{50} erg (approximately what the Sun will emit over its entire lifetime). The aggregate energy flux from these objects is similar to the energy density in high energy cosmic rays, which has led to suggestions that the highest energy cosmic rays may originate in GRBs [5, 6].

2.1 Observations

Gamma-ray bursts are extremely difficult objects to observe: high-energy gamma rays require space-borne observatories and their short durations (seconds to minutes) mean multiwavelength observations require a high degree of coordination and fast-moving telescopes, which has only recently been obtained. As an example, one of the fundamental questions that requires multiwavelength data is the distance to a burst. The first GRB redshift measurement was taken only in 1997 [9]; before this point inferences on the distance to GRBs were based only on the isotropy of their distribution on the sky. Even now, due to the difficulty of these observations and the nearly featureless spectra from many bursts, most GRBs continue to have unmeasured redshifts (Fig. 2.1).

The measured durations of the gamma emission from GRBs appear to be bimodal (Fig. 2.2). The so-called *long* GRBs typically have emission durations of around 30 seconds, and relatively soft spectra. *Short* GRBs typically have emission lasting for only ~ 1 second, with a harder spectrum and usually fewer spectral features. These are thought to have different origins. At least some long bursts appear to be associated with supernovae [7, 10], and short bursts have been proposed as resulting from mergers of compact objects (e.g. two neutron stars, or a neutron star accreting onto a black hole) [11].

At the large distances inferred from the redshift distribution (Fig. 2.1), the inferred gamma-ray luminosity (E_{iso}) is enormous: typically 10^{52} erg or higher (a few percent of a solar mass). Moreover, the energy required for the burst may be even larger than this if, as in supernovae, a substantial amount of energy is emitted by non-electromagnetic means. One way to reduce the required energy is to suggest that the GRB emission

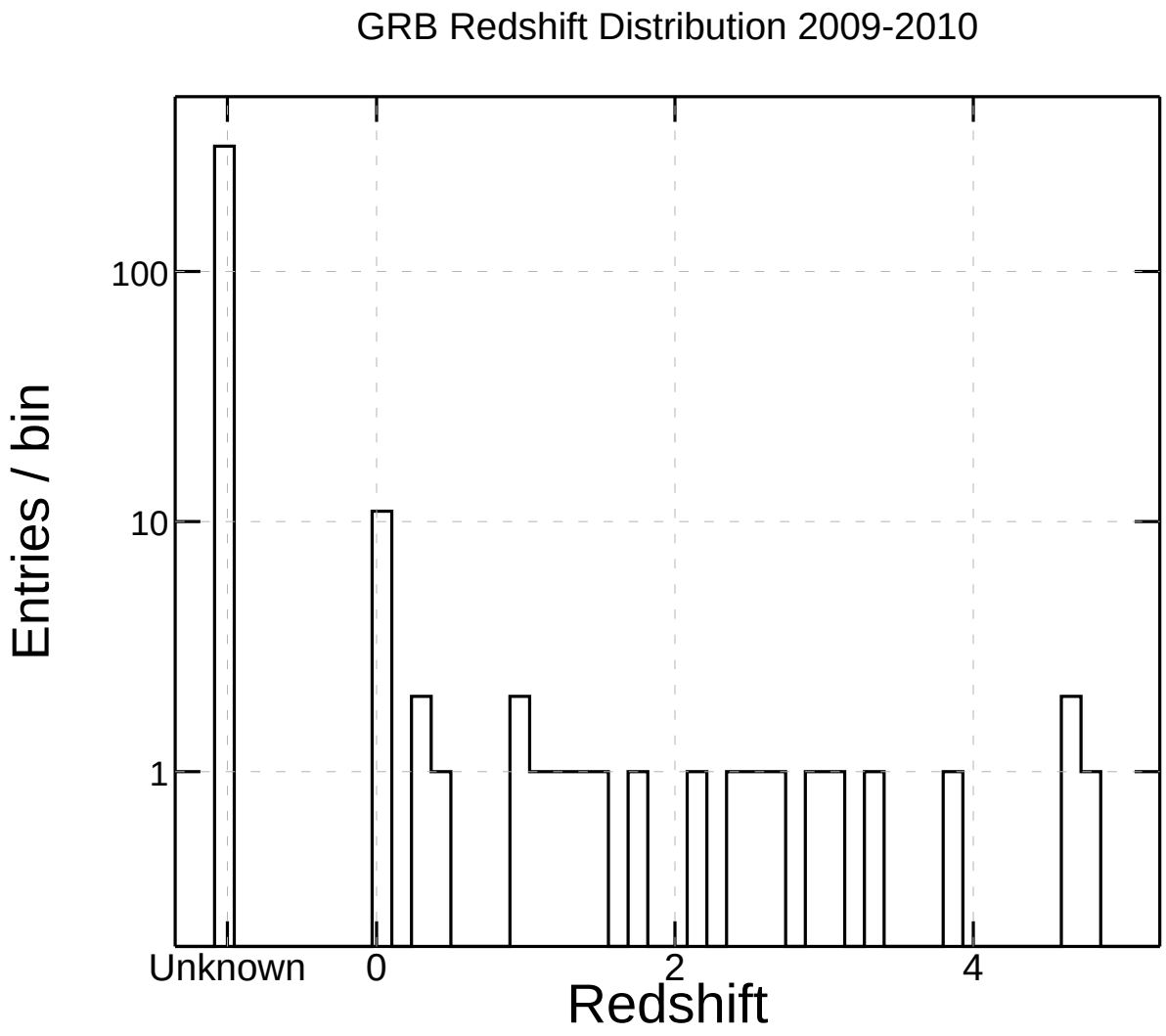


Figure 2.1 Redshifts of GRBs appearing in the IceCube GRB catalog in 2009-2010. A large majority are unknown, due to the fast response requirements and the challenging spectra. Both problems are larger for short GRBs.

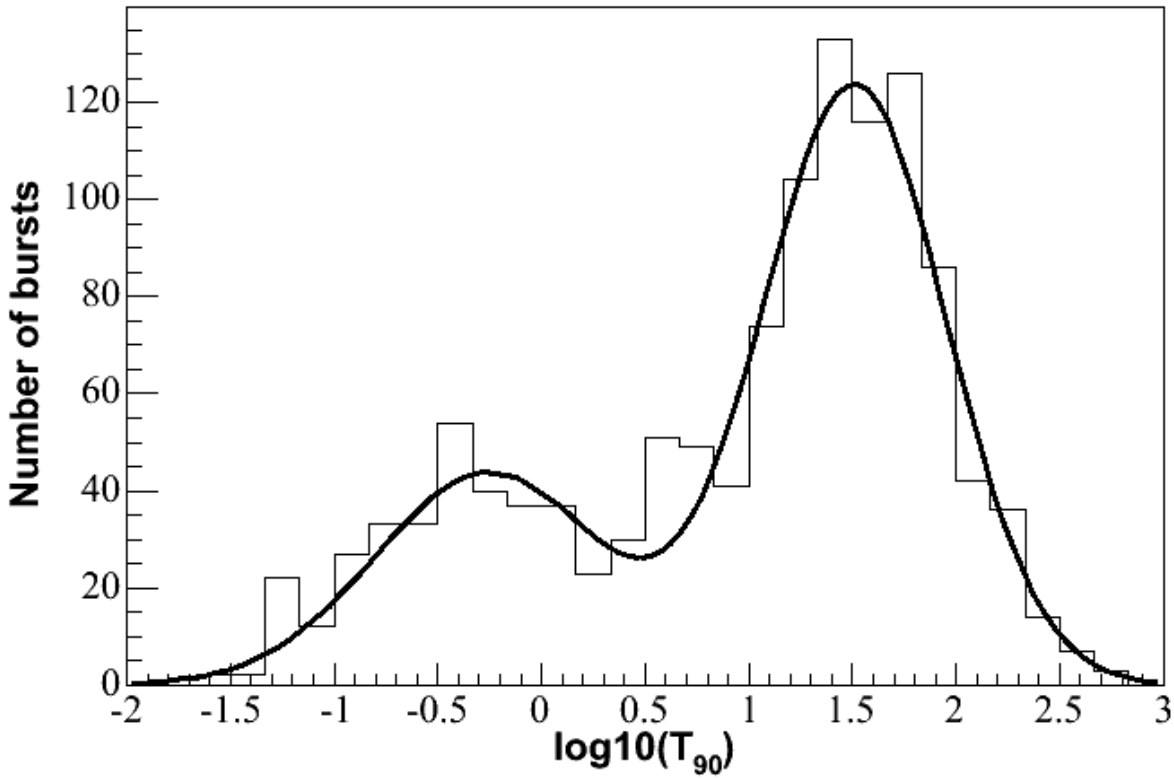


Figure 2.2 Distribution of burst lengths for GRBs in the BATSE catalog [12]. The two different populations are believed to have different sources: short bursts from mergers of compact objects [11] and long bursts from a process related to supernovae [7, 10].

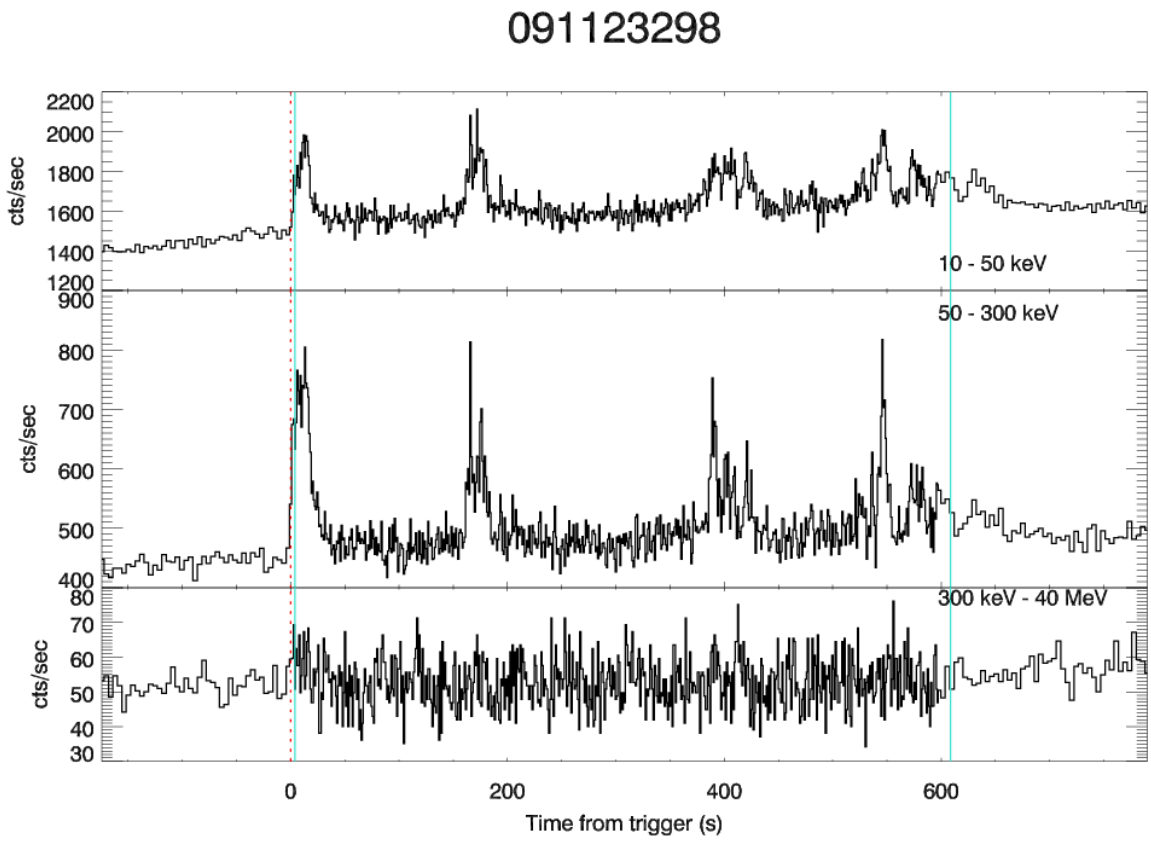


Figure 2.3 Photon flux from GRB091123 as a function of time in several energy bands, taken by Fermi/GBM. An initial rise can be seen of 20 seconds duration, followed by later emission bursts.

is collimated into a jet pointed at Earth [13] so that the observed high power density is not typical of the burst as a whole. This jet is believed to be highly relativistic, with a relativistic beaming angle similar to its width and a bulk Lorentz factor (Γ) between 100 and 1000 [14]. A highly relativistic outflow has significant implications for the particle physics occurring inside it: a high boost factor substantially reduces the energy budget for interactions within it compared to the energies of particles as observed at Earth (in an outflow with $\Gamma = 500$, a proton with a lab frame energy of 500 GeV is at rest).

A potential complication arising from strongly beamed sources is computation of the total burst rate. About 500 GRBs/year are beamed toward Earth [12], but the global rate is higher by a factor $4\pi/\Omega$, where Ω is the solid angle subtended by the beam. For the purposes of determining the total energy budget in GRBs, however, the beaming turns out not to matter. The actual burst luminosity E is related to the apparent isotropic luminosity E_{iso} by the same factor:

$$E = (\Omega/4\pi)\, E_{iso} \tag{2.1}$$

The total GRB energy budget is then the product of the per-burst energy by the number of bursts N in the universe in some unit of time:

$$\begin{aligned} E_{tot} &= NE \\ &= \left(N_{\oplus}(4\pi/\Omega)\right)\left((\Omega/4\pi)E_{iso}\right) \\ &= N_{\oplus}\cancel{(4\pi/\Omega)}\cancel{(\Omega/4\pi)}E_{iso} \\ &= N_{\oplus}E_{iso} \end{aligned} \tag{2.2}$$

where $N_{\oplus}$ is the number of GRBs beamed at the Earth (500/year). The same argument applies to any quantity (e.g. protons, neutrinos) so long as it is beamed together with the gamma-ray emission.

Many transient gamma-ray observatories belong to the Gamma-Ray Coordinates Network (GCN, [15]) operated by NASA. GCN provides coordination among observatories conducting multi-wavelength observations by sending fast messages indicating triggers by orbiting observatories both to other orbiting observatories and to ground-based telescopes, including IceCube. Messages are divided into two classes: machine readable *notices*, which provide very fast but potentially unreliable information and later prose *circulars*, which provide follow-ups from both the triggering instrument and other observers. These rapid messages both enable quick follow-up and the establishment of catalogs for later use. For the analysis described here, the catalog was built primarily from GCN circulars; GCN currently reports about 300 GRBs/year.

2.2 Cosmic Ray Acceleration

Cosmic rays are believed to be accelerated in the fireball as it expands under radiation pressure. The usual model is that this occurs in internal shocks in the GRB by the Fermi acceleration [16] process. As the

fireball expands, relativistic plasma shells with different velocities collide, potentially producing the rapid variation observed in gamma emission (e.g. Fig. 2.3). As these shells collide, particles within the shock can be accelerated by repeated collisions inside this compressive flow (first order Fermi acceleration, Fig. 2.4), potentially reaching energies in excess of 10^{20} eV [5, 6] and sourcing the high-energy tail of the cosmic ray spectrum.

Accelerating particles to these energies is difficult, however. The short duration of the burst (seconds to minutes) and small size imply that the particles must be accelerated to energies exceeding 10^{18} eV extremely quickly, within Γt (at maximum a few hours). By comparison, accelerating protons to 3 TeV in the LHC takes almost half an hour [17]. This requires high magnetic fields (Fig. 2.5) to contain the particles during acceleration as well as rapid collisions with associated high magnetic field turbulence. This picture, while difficult to arrange, is not however inconsistent with observations. GRBs with time variability as rapid as milliseconds have been observed [18], indicating very rapid collisions of plasma shells. Further, the high-energy gamma emission is consistent with synchrotron radiation, indicating both the presence of high-energy non-thermal electrons and strong magnetic fields. The power-law spectra observed are also characteristic of electrons whose energy was gained in shock acceleration processes.

One complication of particular interest from the perspective of neutrino searches is that the magnetic fields required to satisfy the Hillas condition (that the particles remain within the source during the acceleration) also mean that any accelerated protons remain *trapped* within the burst as it rapidly cools [20]. Because these particles are magnetically coupled to an adiabatically expanding system, they lose energy as the fireball rapidly fades, making it very difficult for them to escape at high energies into the inter-stellar medium (the *ejection problem*). It is possible to circumvent this, however, by arranging the conversion of protons to neutrons, decoupling them from the burst magnetic field [20, 21]. This has strong implications for the neutrino flux (Sec. 2.3), and is the scenario in which neutrino observations are best able to test the GRB/cosmic ray hypothesis.

2.3 Neutrino Production

The first prediction of neutrino production in relation to cosmic ray acceleration in GRBs is due to Waxman and Bahcall [22], and all of the predictions tested here are ultimately derived from that one. The core observation is that the coexistance of very high-energy protons and the photons later observed as the GRB inevitably results in the production of neutrinos, for example by the process $p + \gamma \rightarrow \Delta^+ \rightarrow \pi^+ + n \rightarrow e^+ + \nu_\mu + \bar{\nu}_\mu + \nu_e + n$. These high-energy neutrinos would be an unambiguous signature of the presence of very high energy protons within the burst.

It is generally assumed that, based on the hard non-thermal spectrum, the photon spectrum observed at Earth is representative of the spectrum within the burst. The energy of the protons is fixed by assuming

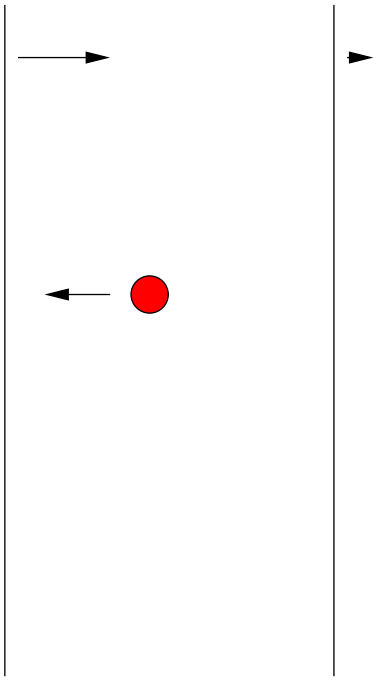


Figure 2.4 Cartoon of first-order Fermi acceleration [16]. The faster-moving flow on the left is overtaking the slower-moving one on the right, producing a *shock* at the collision point. Viewed from the frame of this shock, this constitutes a *compressive flow* – gas appears to be moving in from both sides. A test particle (red) in this region can therefore repeatedly gain energy from collisions with both sides. Reflection from the comoving magnetic fields on either side of the shock transfers momentum to the test particle, like a tennis ball hit back and forth by two (temporarily) inward moving rackets, continuously increasing its energy.

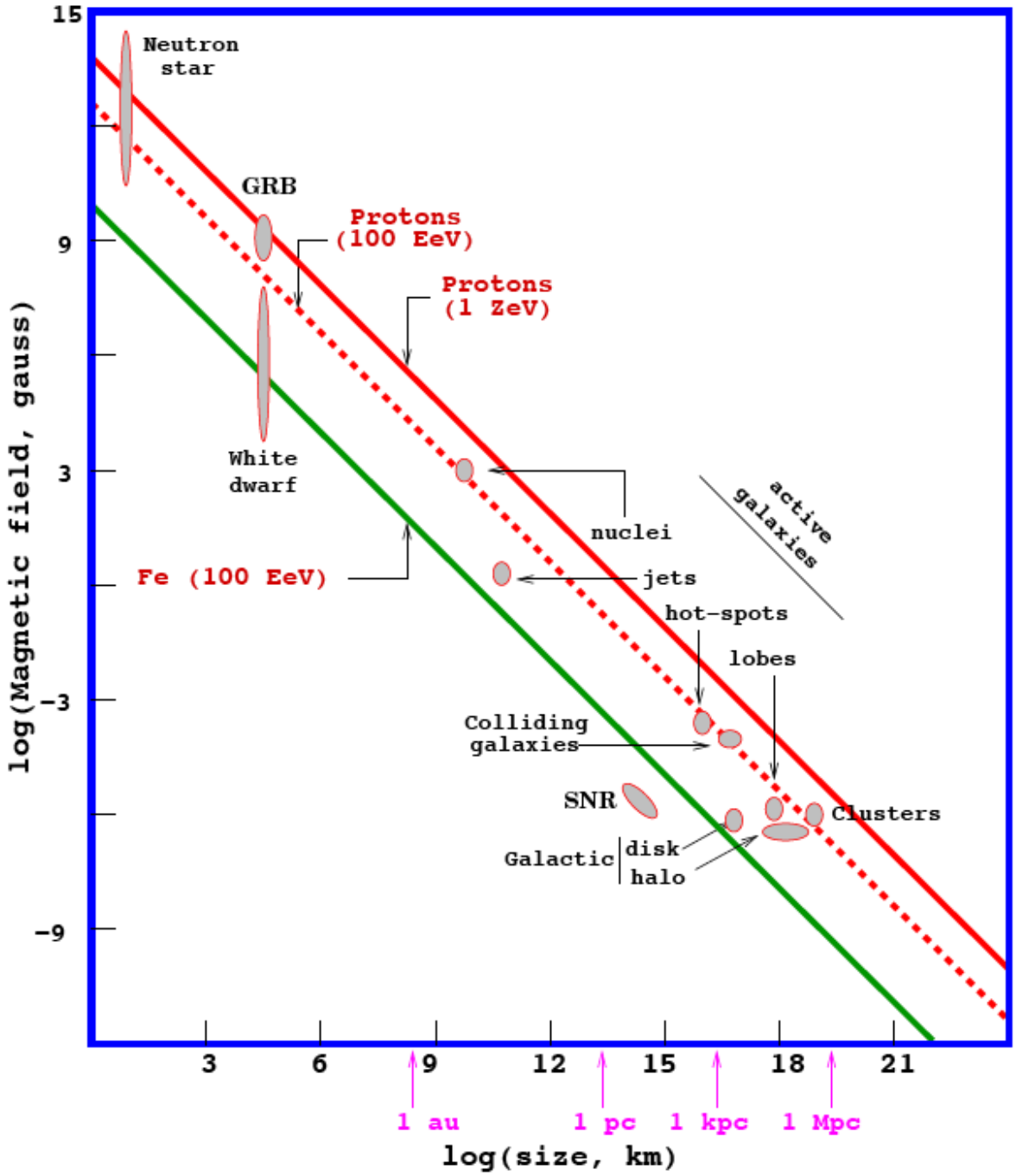


Figure 2.5 Cosmic Ray acceleration sites need to have strong enough magnetic fields to keep particles confined within the source. The required magnetic field (vertical axis) varies with the source size (horizontal axis). Only objects above and to the right of the lines meet the requirements to accelerate particles to the indicated energies. Plot from [19].

that the protons present within the burst trace the cosmic ray spectrum (Waxman and Bahcall assume 100% proton ejection efficiency). The only remaining ingredient in the energy scale of the problem is the bulk Lorentz factor of the outflows in which the acceleration is assumed to occur. If this is very high, the energies of the particles within the frame of the outflow may be much lower than would be indicated by terrestrial observations, inhibiting particle interactions. For a fixed photon energy (observed at Earth), the proton energy (at Earth) corresponding to the first Δ resonance (1232) is:

$$E_p = \gamma^2 \frac{m_\Delta^2 - m_p^2}{4E_\gamma} \tag{2.3}$$

For a 1 MeV photon and $\gamma = 300$, this corresponds to $E_p = 1.5 \times 10^{16}$ eV. For an average pion-production inelasticity of 0.2 and energy split equally between the four final state leptons (e^+, ν_μ, $\bar{\nu}_\mu$, ν_e), this implies a neutrino energy of $\frac{1}{20}E_p$, or 7×10^{14} eV. There is also an additional correction from the GRB redshift, which changes both the energy of the neutrino at Earth and the in-burst energies corresponding to the observed photon energies in much the same way as Γ: in the frame of the GRB outflow, photons and neutrinos have higher energies by a factor of Γ than they appear at Earth. The neutrino energy corresponding to the first Δ resonance (and so the minimum energy of the neutrino) is then:

$$E_\nu = \frac{1}{4} x_{p\to\pi} \frac{\gamma^2}{(1+z)^2} \frac{m_\Delta^2 - m_p^2}{4E_\gamma} \approx 8\text{GeV} \times \frac{\gamma^2}{(1+z)^2} \frac{1\text{MeV}}{E_\gamma} \tag{2.4}$$

The characteristic shape of the Waxman-Bahcall spectrum (Fig. 2.6) is formed by integration of this threshold over the photon spectrum in the burst. Below the threshold, the flux decreases steeply, and only photons of increasingly high energy can participate. Above the threshold, the neutrino spectrum traces the proton spectrum (usually assumed to be E^{-2}), as nearly all photons can participate in the interaction. An additional suppression occurs at high energies, where the lifetime of the intermediate pions and muons becomes long enough that significant amounts of their energy are lost to synchrotron radiation before decay to neutrinos.

The rate of this process is related to the product of the densities of the protons and photons within the burst. The physical length scale is given by the observed variability timescales (t_{var}) of the gamma-ray light curve (Fig. 2.3, note that these also scale into the burst frame with γ and z). The total luminosity of the burst in neutrinos is then related to the following quantities [22]:

$$\phi_\nu \propto L_\gamma \epsilon_p \Gamma^{-4} t_{var}^{-1} \tag{2.5}$$

where L_γ is the burst luminosity and ϵ_p is the fraction of the burst luminosity in protons. The aggregate neutrino flux from all GRBs can then also be calculated assuming some "typical" spectral parameters and

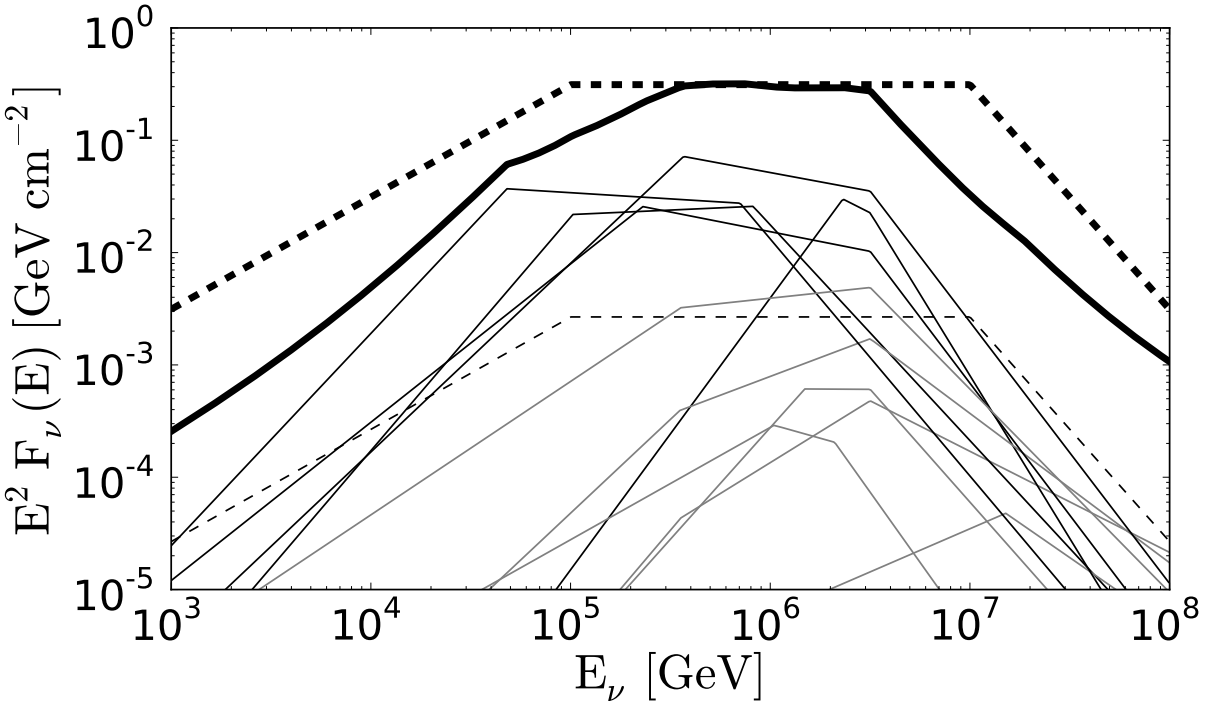


Figure 2.6 Expected neutrino spectrum from GRBs during the IceCube-40 run according to the Waxman-Bahcall [22] and Guetta *et al.* [23] models, which calculates spectra from individual bursts. The flux predictions normalized to gamma-ray spectra [23, 24, 25] are shown in solid lines; the cosmic ray normalized Waxman-Bahcall flux [22, 26] is also shown for reference (dashed lines). Thin lines are the fluences for individual bursts, and thick lines are summed for all the bursts in the sample. The predicted neutrino flux, when normalized to the gamma rays [23, 24], is proportional to the ratio of energy in protons to that in electrons, which are presumed responsible for the gamma-ray emission (ϵ_p/ϵ_e, here the standard 10). The flux shown is slightly modified [24] from the original calculation [23]. Waxman-Bahcall [22] predicts an aggregate flux from the total GRB population; the value here was obtained by scaling the prediction by the global GRB rate (here 667 bursts/year [27]). The first break in the neutrino spectrum is related to the break in the photon spectrum measured by the satellites, and the threshold for photopion production (Eq. 2.4), while the second break corresponds to the onset of synchrotron losses of muons and pions. Not all of the parameters used in the neutrino spectrum calculation are measurable from every burst. In such cases, benchmark values [27] were used for the unmeasured parameters.

setting $\sum L_\gamma \epsilon_p$ to the aggregate power in high-energy cosmic rays sourced in GRBs – this is the Waxman-Bahcall flux.

2.4 Model Variants

There are two primary variants of this scenario. The first is the Waxman-Bahcall type, where the cosmic rays are assumed to be present at all times within the burst, and to exit with 100% efficiency once the burst is concluded. The second concerns itself with how the protons eventually escape the burst into the ISM.

The first class contains two major predictions: Waxman-Bahcall [22] and Guetta *et al.* [23]. The Waxman-Bahcall prediction predicts a global aggregate flux of GRB neutrinos based on typical GRB parameters and a total proton power approximately equal to the cosmic-ray density. Guetta *et al.* uses a very similar physical picture to predict neutrino fluences from individual GRBs, based on the measured properties of the bursts and a proton power set to a fixed fraction ($1/f_e$) of the total burst luminosity. This fraction is usually picked to make the Guetta *et al.* and Waxman-Bahcall fluxes approximately agree for typical burst luminosities; this choice causes the burst proton power in both scenarios to approximately match the total power in high energy cosmic rays. Since these predictions describe the same physics, they should asymptotically converge to the same flux when averaged over many bursts. Strong variation in the properties of individual GRBs, however, results in as much as a factor of 2 variation in the predicted neutrino flux even for samples as large as several hundred bursts [23, 28].

Both of the models in the Waxman-Bahcall class suffer from large uncertainties. They depend critically on parameters that are either generally not measured (e.g. Γ) or are ill-defined (t_{var}). Both entirely neglect all neutrino production channels other than the Δ resonance process (e.g. Kaon production and multi-pion processes) as well as continued acceleration of the non-thermal pions and kaons, both of which tend to increase the flux relative to the predictions. Finally, both use many approximations in place of detailed numerical calculations. There have been some subsequent studies of the relationships between the unmeasured parameters [29] and recent better treatments of the particle physics [30], but significant theoretical work, especially in an integrated treatment of neutrino acceleration with cosmic ray acceleration and escape, remains to be done.

The second class of models, first proposed by Rachen and Mészáros [20], provide a much more direct connection between the neutrino and cosmic ray fluxes with many fewer ambiguities in the flux prediction. These predictions [20, 21] are based on the observation that, as the fireball rapidly expands, it will adiabatically cool along with any high-energy protons inside. This results in any accelerated protons being magnetically confined within the burst and unable to escape at high energies to form the highest energy cosmic rays. To source the cosmic ray spectrum, the protons then need to somehow become decoupled from the magnetic fields in the fireball, for example by becoming neutral particles.

Conversion of protons to neutrons occurs via the same kinds of processes that produce neutrinos: in particular, the predominant channel for conversion envisaged in these models is the same $p + \gamma \rightarrow \pi^+ + n$ process in Waxman-Bahcall. Since both the cosmic rays (the neutrons) and neutrinos (π^+) are the products of the same process, we need not worry about the rate of $p\gamma$ interactions themselves for the purposes of relating the neutrino and cosmic ray fluxes: every proton that enters the cosmic ray spectrum is accompanied by exactly 3 neutrinos, each with (on average) 1/12 the energy of the neutron.

The only free parameter in the neutrino flux for neutron escape models is the shape of the spectrum of escaping cosmic rays, which is set in large part by the jet bulk Lorentz factor Γ. In all the models discussed here (including non-neutron models), the proton spectrum from GRBs is assumed to be a power law of which the cosmic ray spectrum above the ankle is the tail. Neutrinos then come from protons with energies far below 10^{18} eV (Eq. 2.3). In neutron-origin models, this threshold, set by Γ, then also sets the shape of the proton spectrum, since protons escape the burst via the same process. This means, among other things, that Γ is not freely variable within the context of neutron models: if set too high ($\gtrsim 3000$), the proton flux from GRBs will begin at too-high energies, above the ankle at 4×10^{18} eV.

Chapter 3

IceCube Detector

The IceCube detector is constructed of 5160 10-inch photomultipliers embedded in the polar icecap and housed with their readout electronics in glass pressure spheres. These record Cerenkov light produced by in-ice particles, allowing the reconstruction of particle directions, energies, and types. These *optical modules* are located on 86 vertical strings, each containing 60 modules located between 1450 and 2450 meters depth. On 80 of the strings, the modules are placed 17 meters apart and the strings are located, on average, 125 meters from one another. On the 6 strings comprising the low-energy Deep Core subdetector, the modules are much more closely spaced (7 meters), and the strings much closer together. This instrumentation density is optimized for detection of neutrinos at energies of $\gtrsim$ 100 GeV.

3.1 Detection Principle

Neutrinos can only be detected by their interactions with matter. There are typically two of these that matter for neutrino detectors, where l is any type of lepton:

$$\begin{aligned} p/n + \nu_l &\rightarrow p/n + \nu_l \\ p/n + \bar{\nu}_l &\rightarrow p/n + \bar{\nu}_l \\ n + \nu_l &\rightarrow p + l^- \\ p + \bar{\nu}_l &\rightarrow n + l^+ \end{aligned} \tag{3.1}$$

The first part of Eq. 3.1 describes the *neutral current* interaction, which takes place by exchange of a Z boson, and the second the *charged current* interaction, which involves the exchange of a W. These have very different signatures in a detector. A neutral current neutrino interaction transfers energy and momentum to the nucleon, but otherwise leaves it unchanged. At very low energies, this energy can be observed by various calorimetric methods familiar from dark matter detectors. At higher energies, the energy deposition will cause the ejection of a quark from the nucleus, triggering a charged particle shower. Charged current interactions change nucleons and produce a charged lepton. Both of these can be detected. The change in nucleon type will result in a chemical change of the material, and may result in a later radioactive decay.

Large energy depositions can also produce a shower, as in the neutral-current case, and the charged lepton can be detected by a variety of means.

IceCube detects particles observing photons emitted by the Cerenkov process, the light emitted by relativistic charged particles as they travel through a refractive medium (ice). The requirement for relativistic charged particles means that IceCube is sensitive both to showers from all high-energy interactions and to the charged leptons produced in charged-current interactions.

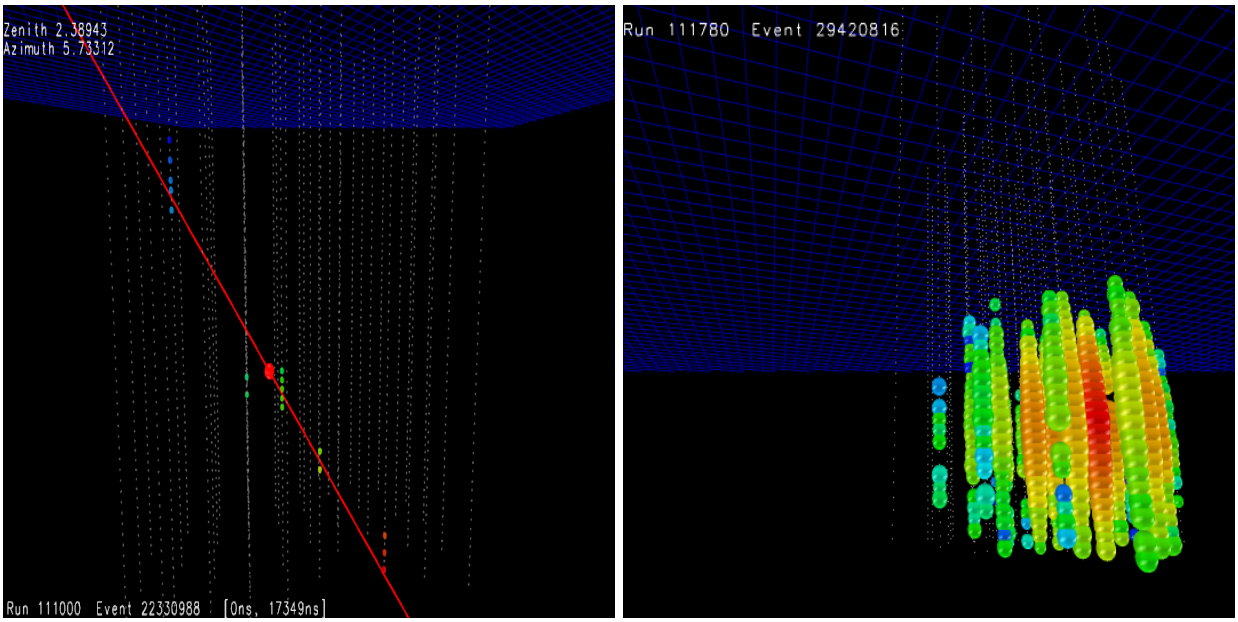
The sensitivity of the detector to particular interaction channels is, however, variable. Typical TeV-scale showers are ≈ 10 meters long, which compares unfavorably to the instrumentation density (at best order 10 m between DOMs on the same string). Electrons and positrons produced in ν_e and $\bar{\nu}_e$ interactions also shower very quickly, producing a similar detector signature (Fig. 3.1(b)). The light produced from such events, on the scale of the detector instrumentation, is then nearly isotropic. This makes reconstruction of the direction of the original neutrino very difficult and makes neutral-current and ν_e charged-current interactions generally unsuitable for use in searching for astrophysical point sources.

The much better channel for this kind of study is detection of ν_μ charged-current interactions. Because of their low rate of energy loss (Fig. 4.1), TeV-scale muons can travel for kilometers through ice (Fig. 3.2). This produces a light track that can be easily reconstructed with high precision (Fig. 3.1(a), Sect. 4.1) and the interaction is sufficiently forward that the muon and neutrino are nearly collinear (Fig. 3.3). In addition, having muon ranges larger than the physical detector size allows detection of muons originating from outside the detector volume and thereby increases the effective sensitive volume of IceCube.

3.2 Digital Optical Modules

The central element of the IceCube detector is the Digital Optical Module (DOM, Fig. 3.4). Each contains a 10-inch Hamamatsu photomultiplier, as well as a power supply, readout electronics, and calibration devices, and is housed in a glass pressure vessel in order to withstand the 300 atm. pressure in the deep ice (and up to 650 during freeze-in).

The data acquisition electronics in the DOM (Fig. 3.5) provide sampled waveforms from the PMT using two types of digitizers: the ATWD and the FADC [4]. The FADC is a commercial pipelined fADC unit clocked at 20 MHz and sampling at both edges of the clock gate, proving a 40 MHz sampling rate. This is relatively slow in comparison to the ns-scale PMT pulses and desired precision of the readout (the 25 ns sampling interval corresponds to a 1.5 m uncertainty in position). The second digitizer type, the ATWD (Analog Transient Waveform Digitizer), uses an analog capacitor storage bank, switched at 300 MHz, and a much slower digitizer. When triggered, the ATWD can sample the waveform with a 3.3 ns sampling interval, but, because of the slower ADC, cannot do this continuously. Once the 128-bin ATWD fills (422 ns), it requires several microseconds to digitize and clear. This situation is ameliorated by the presence of a second



(a) Muon neutrino event, ≈ 1 TeV

(b) Shower event, ≈ 50 TeV

Figure 3.1 Muon neutrino and shower (either neutral current or ν_e charged current) events detected during the 40-string IceCube run. The muon event makes a long track in the detector volume that allows its direction of origin to be easily determined. The shower is much more isotropic, making directional reconstruction difficult.

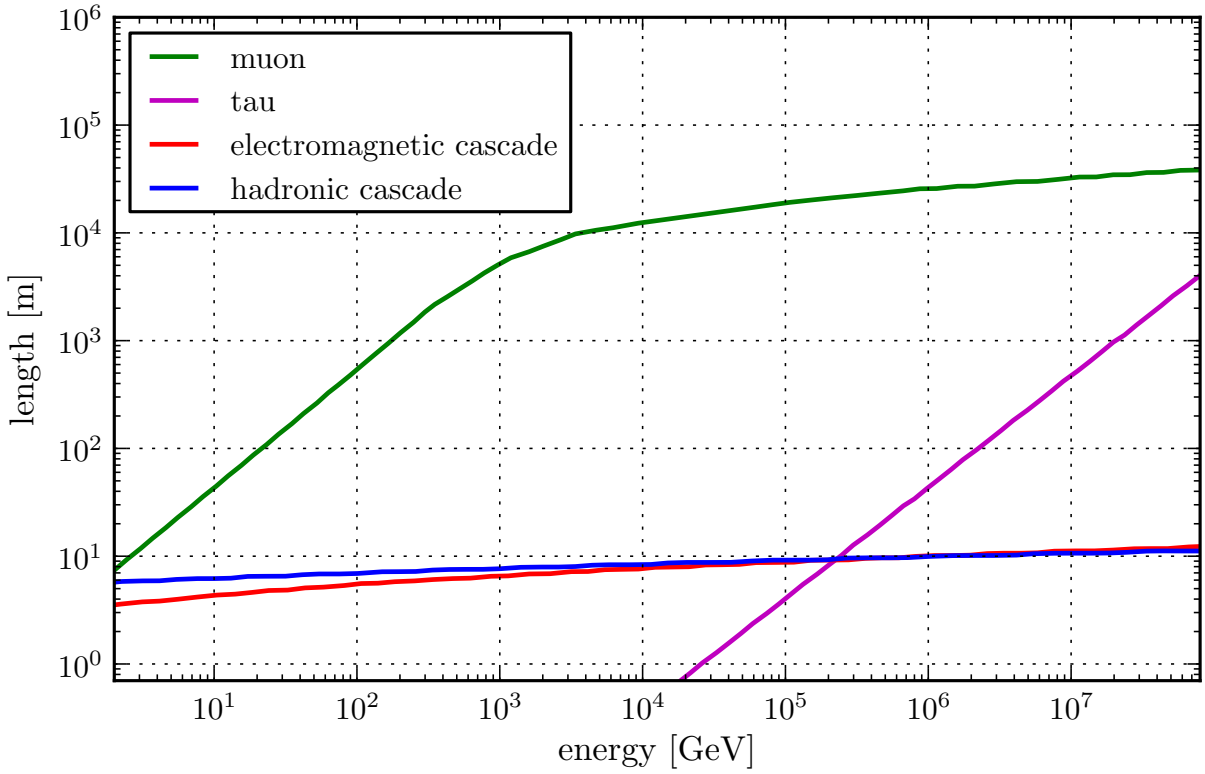


Figure 3.2 Range of events in ice as a function of the energy of the primary neutrino. Above a few hundred GeV, most muons have tracks longer than the size of the detector. Below ~ 1 TeV, where the muon energy loss rate is approximately independent of energy (Fig. 4.1), the path length is a linear function of energy. Above this, where $dE/dX \propto E$, the muon path length rises approximately logarithmically with energy. Cascades (e.g. from neutral current and electron neutrino events) have lengths that depend only weakly on energy, as both the density of particles in the shower and the loss rate of those particles rises with energy. Taus deposit tracks, like muons, but have much shorter lifetimes and energy loss rates than muons do. Figure from [31].

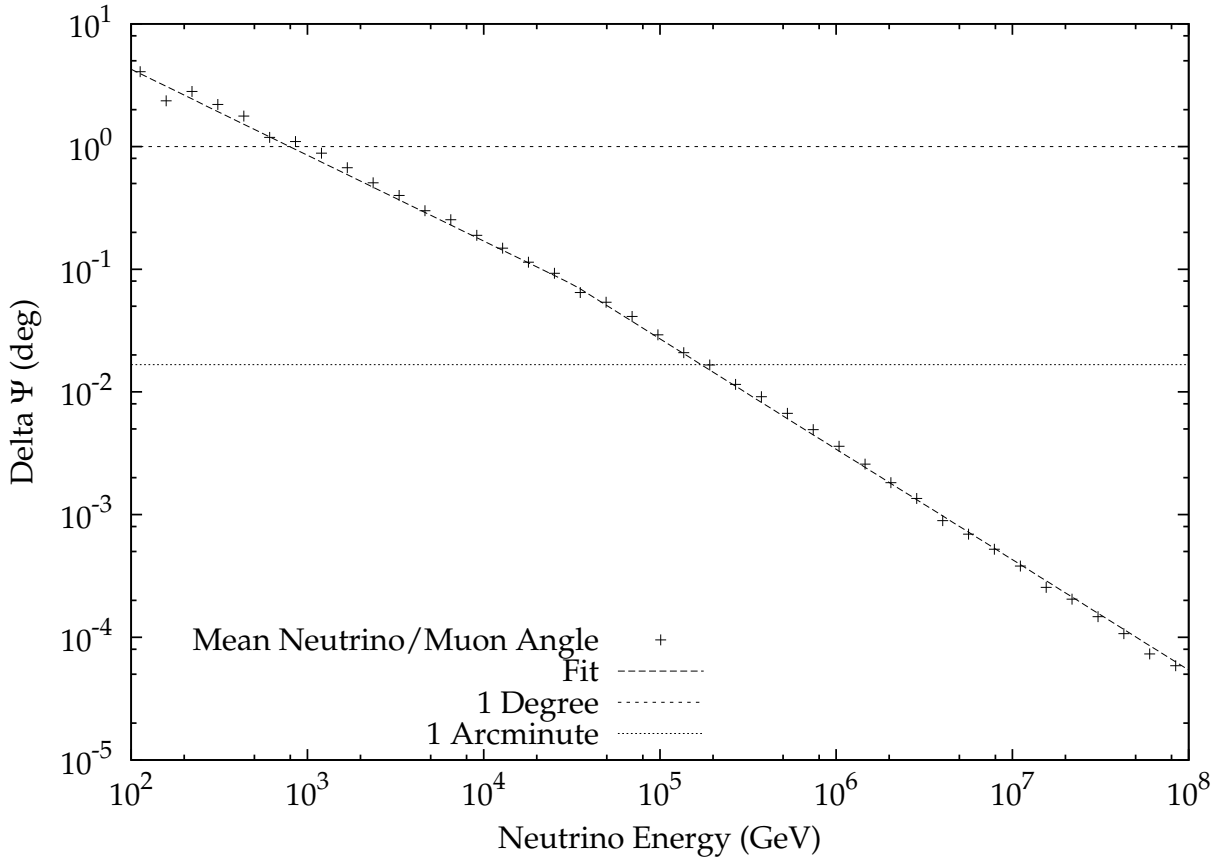


Figure 3.3 Angle between the neutrino and produced muon, as a function of energy. As the interaction becomes more and more forward at high energy, the angle between the incoming neutrino and produced muon is reduced. Above about ~ 1 TeV, the mean angle between the muon and the neutrino is generally less than the IceCube angular resolution on the muon. The shown fit is a broken power-law: $\Delta\Psi = 0.85\,\mathrm{deg} \times \left\{(E_\nu/\mathrm{TeV})^{-0.7}, E < 33\,\mathrm{TeV}; 33^{-0.7} \cdot (E_\nu/\mathrm{TeV})^{-0.9}, E \geq 33\,\mathrm{TeV}\right\}$

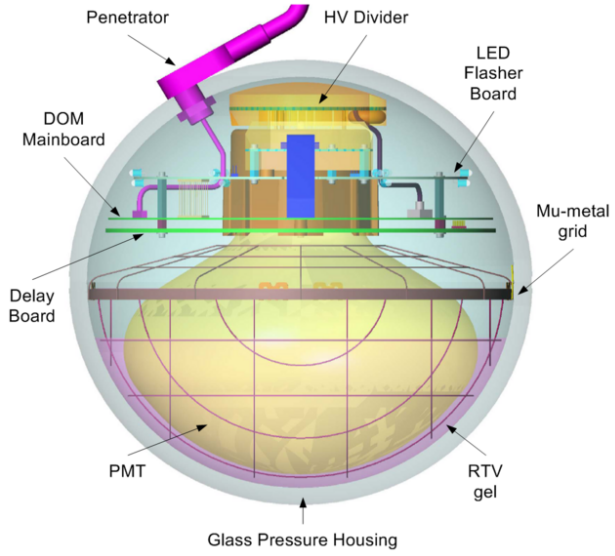


Figure 3.4 Digital Optical Module

ATWD chip, which is used in a ping-pong mode. Each ATWD chip contains three channels, each of which read out the waveform with a different gain, providing a much higher dynamic range.

Each DOM triggers independently, based on its own state and the state of its neighbors, determined by *local coincidence* signals that can be sent to neighboring DOMs on the same string. When the DOM discriminator (Fig. 3.5) detects a pulse from the PMT, it will trigger and begin digitizing the PMT waveform – the delay line before the digitizer allows it to read the triggering pulse as well – and send a local coincidence signal to its neighbors. If one of the neighboring DOMs has also triggered within 1 μs, the full recorded waveforms will be sent to the surface (Hard Local Coincidence, or HLC, mode). If not, 3 FADC samples (75 ns) corresponding to the highest amplitude part of the waveform will be sent instead (Soft Local Coincidence, or SLC).

The DOMs also include calibration devices (the LED Flasher Board) that allows measurement of absolute DOM sensitivities and the optical properties of the ice. Each DOM contains 12 LEDs, six pointed horizontally and six slightly upward, all of which can be independently flashed and then measured by other DOMs in the detector.

3.3 Data Acquisition and Triggering

Trigger records and readouts from each optical module are transmitted autonomously to the surface by each DOM. There they are collected by the global trigger, which builds DAQ events from the stream of

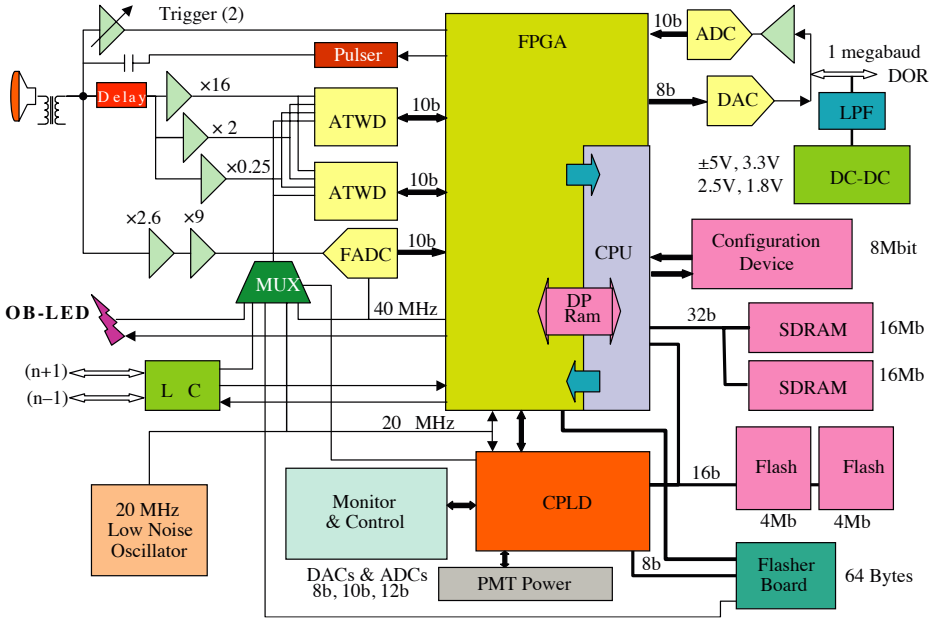


Figure 3.5 Digital Optical Module Mainboard Block Diagram

Name	Trigger condition	Used for
SMT8	More than 8 HLC launches within 5 μs	Muons and Showers
Min Bias	Random samples of data	Calibration
String	5 hits from a set of 7 contiguous DOMs on one string	Muons

Table 3.1 Triggers used during the IceCube-59 physics run. Events are extendable during trigger conditions and are made of data from 10 μs before the beginning of the first trigger to 10 μs after. The trigger conditions are designed to be fairly weak – actual event reconstruction and particle identification is performed by later online filtering.

individual DOM launches. Because of the relatively low data rate in IceCube, the trigger conditions tend to be simple (Table 3.1), with later event rate reduction performed in online and offline filtering.

3.4 Pulse extraction

For almost all particle reconstruction algorithms (Chapter 4), working directly with PMT waveforms and digitizer traces is both complex and cumbersome. There are differences between the various digitizer channels, which need to be cross-calibrated, problems correctly interpreting photon pileup, etc. A much simpler approach is to run a first reconstruction that attempts to reconstruct the photon arrival times at the photocathode, and make the results of this the input to the particle reconstructions.

There are several ways to do this, one developed in the course of this work (Section 3.4.1). The basic assumption of all of them is that the photomultipliers, amplifiers, and ADCs are linear, which appears to be true for most amplitudes of physics interest [32].

The simplest approach is simply to sum an entire PMT waveform to get a total charge and try to find the first crossing time for some chosen small threshold. For widely separated photons, this can be generalized by dividing the waveform into multiple over-threshold regions and reporting a crossing time, charge, and time-over-threshold for each. This approach, however, suffers from a number of difficulties: it handles pile-up extremely poorly (important for high energy events) and also cannot properly treat either the low-gain ATWD channels or the ATWD $\rightarrow$ FADC switchover region. It also can miss substantial amounts of charge for PMT pulses not fully contained within the readout. This frequently occurs for example in the 3-bin soft-local-coincidence waveforms, where the readout is shorter than the pulse length after transmission through the FADC amplifiers.

3.4.1 Wavedeform

A more advanced method to reconstruct waveforms is to treat this as an unfolding problem. For a fully linear electronics chain, any given ADC readout will be a linear combination of shifted copies of the single

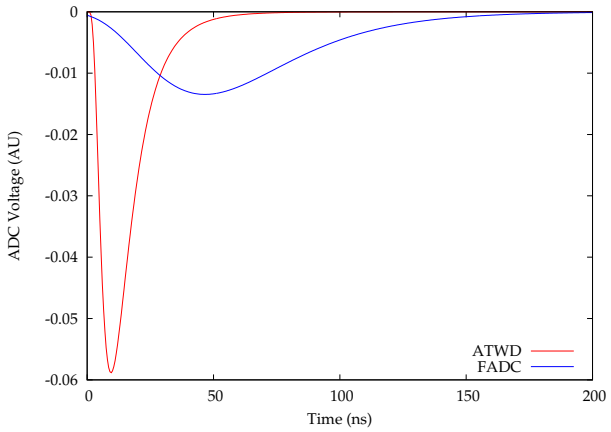


Figure 3.6 Shaping functions for the highest-gain ATWD channel and the FADC. The amplifier chains in front of the digitizers (ATWD and FADC, Fig. 3.5) apply a shaping function to any input pulse, here 2 ns wide. To prevent missed charge in the digitizers (the ATWD samples at 300 MHz and the FADC at 40 MHz), these narrow input pulses are widened to extend across several samples. Because of the different sampling rates, the FADC shaping is much wider. Differences in the amplifier chains in front of the three ATWD channels also introduce minor differences between the shaping function for each, which are not shown here.

photoelectron (SPE) response function, with a width and shape largely determined by the shaping functions of the amplifiers (Fig. 3.6). Given a basis of these shifted SPE templates, the waveform (y) can be written as the product of this basis matrix (B) and the effective charge deposition at the PMT photocathode at each of a number of closely spaced time points (α):

$$\begin{pmatrix} B_1(x_1) & B_2(x_1) & \cdots & B_n(x_1) \\ B_1(x_2) & B_2(x_2) & \cdots & B_n(x_2) \\ \vdots & & \ddots & \vdots \\ B_1(x_m) & B_2(x_m) & \cdots & B_n(x_m) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \tag{3.2}$$

Inversion of this linear system then allows the reconstruction of the charge distribution at the PMT photocathode (α), which is the quantity of interest in later reconstructions. This linear formulation very naturally handles pileup and can be easily extended to handle the multiple waveforms measured by various in-DOM digitizers by the addition of rows to the basis matrix B with the template functions for those additional channels.

One standard difficulty with this sort of approach is the danger of overfitting and ringing (Fig. 3.7). When the data are not representable as an exact combination of the basis functions (as is often the case due, for example, to statistical fluctuations), an unconstrained linear least-squares fit will try to produce the observed features by interference between basis functions, assigning alternate large negative and positive

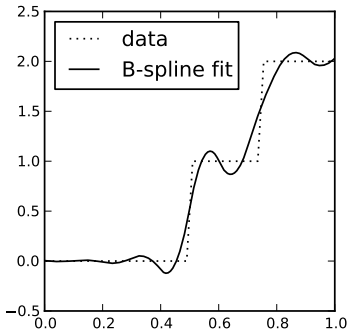


Figure 3.7 Illustration of ringing with unconstrained linear least-squares fits. Because the step function (dotted line) is not representable by the basis functions being used (B-Splines in this case), the slope of the fit has been steepened by assigning alternating large negative and positive values to the basis functions, producing the ringing visible here. Not only is the ringing itself undesired, but negative coefficients are unphysical when interpreted as photon counts or energies, as in pulse unfolding.

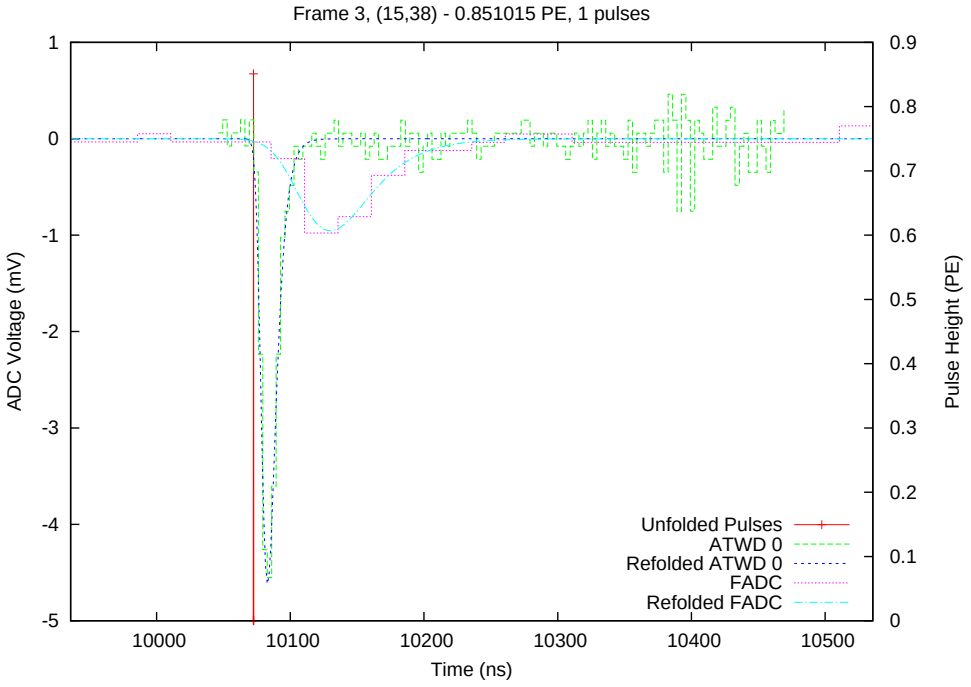
values to the basis functions. This can be beneficial when simply trying to describe the data, as in a Fourier transform, but makes it impossible to interpret the basis function weights α as physical quantities, in this case the charge deposited at the PMT photocathode.

In many cases, the solution to this problem lies in the use of regularization [33], which involves attaching a penalty in the fit to rapid variation in the coefficients. This usually takes the form of penalizing large values of the second derivative of the fit functions, thereby preferring straight lines to the oscillating functions seen as a result of ringing. In the more general case, the regularization takes the form of a Bayesian prior [33, Rejoinder], and will penalize deviations away from any fixed functional form.

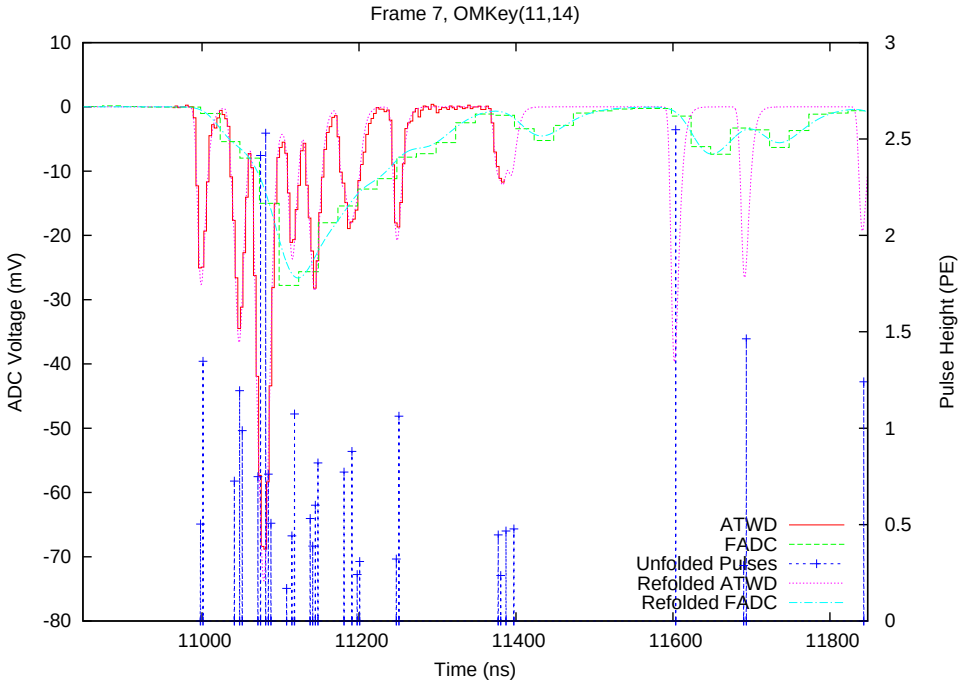
Such an approach, however, is not suitable for application to waveform unfolding, as the underlying process is stochastic. Rapid variation of the fit coefficients is expected – corresponding to the arrival of individual photons at the photocathode – and there is no expected fixed functional form. Therefore, any kind of penalty would distort and bias the result.

An alternate approach is to restrict ringing by stipulating that all the coefficients α must be *non-negative*. Without the possibility of negative coefficients, it is also impossible to use large positive coefficients to produce unrepresentable features by interference. This entirely prevents both the ringing problem and the problem of unphysical negative numbers of photoelectrons. The class of linear algorithms that provide this feature are called *Non-Negative Least Squares* (NNLS), with the most common being the Lawson-Hanson algorithm [34], which is used in Wavedeform. The algorithm is described in more detail in Sec. A.4.1.

A related problem to ringing is the stopping condition of the fit. Even without ringing, small statistical fluctuations can always be better described using a linear combination of many basis functions (overfitting). The desired stopping condition, then, is one that provides a good fit with maximum sparsity. While sparsity



(a) Unfolding a simple waveform containing one photon



(b) Unfolding a more complicated waveform

Figure 3.8 Examples of waveform unfolding for both simple and complex waveforms using Wavedeform. The lines marked *Refolded* are predictions of what the various digitizers would read given the reconstructed PMT hits. For a perfect reconstruction, and with no noise in the data, these lines would exactly match.

maximizing NNLS algorithms are NP hard [35], a useful approximation is to exploit the structure of the Lawson-Hanson NNLS algorithm. At each stage of its outer loop, NNLS adds exactly one basis function to a set allowed to be non-zero. This basis function is chosen as one where the gradient of the residual function (the expected change in the fit error from freeing that coefficient) is maximal, and the algorithm ordinarily terminates when it has exhausted the set of basis functions that would improve the fit by taking positive values. By terminating this algorithm early, when freeing no single basis function would increase fit quality by more than some threshold, only the principle components of the data can be extracted.

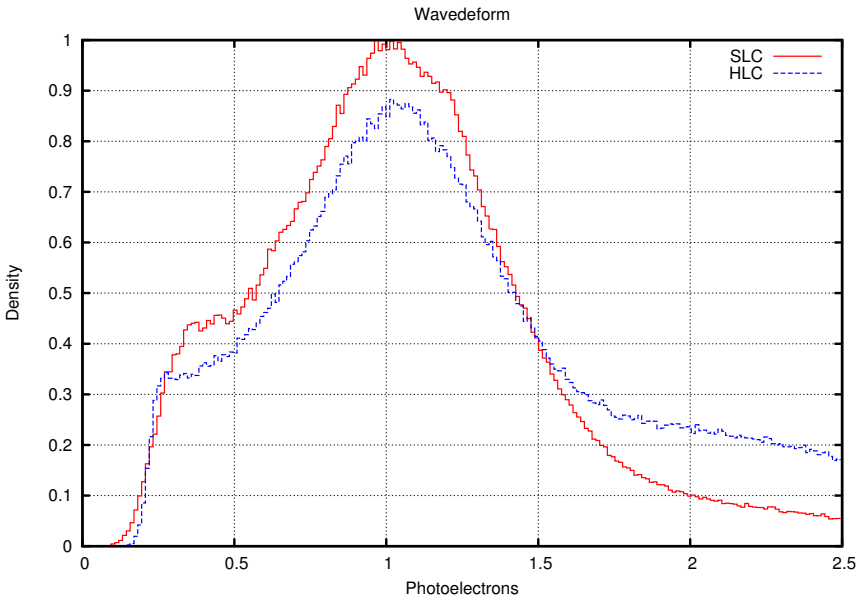
As can be seen in Fig. 3.8, the NNLS-based algorithm works well. For simple waveforms (Fig. 3.8(a)), the fit to all active digitizers is good, and only a single basis function attains a positive value. For more complex waveforms (Fig. 3.8(b)), pileup in the waveforms is handled well, as are inter-digitizer transitions. The available timing resolution is very good, as well: at maximum fit quality, time resolution as high as 250 ps can be achieved, compared to the ATWD digitizer sampling time of 3.3 ns. Charge resolution is also substantially improved compared to the previous waveform reconstruction algorithms (Fig. 3.9).

3.5 Optical Properties of the Ice

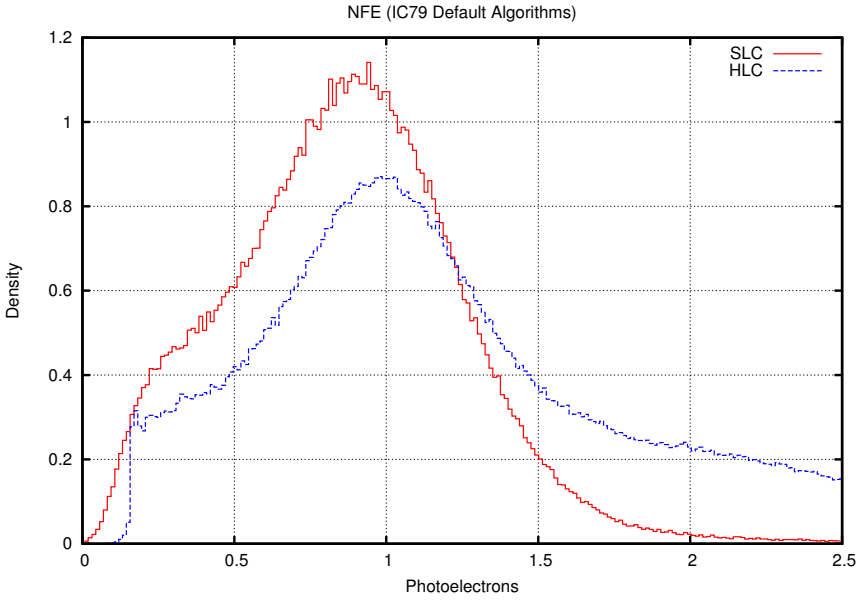
The ice in the south polar ice sheet used as the IceCube detection medium is extremely pure, with better optical properties than any laboratory ice [36] and scattering and absorption lengths on order 100 meters (Figs. 3.10, 3.11). This enables high-quality event reconstruction with IceCube's relatively low density of optical sensors.

The ice, however, presents challenges different from most physics experiments: as a naturally occurring material, it doesn't obey any manufacturing standards of purity or homogeneity. In particular, its properties must be measured *in situ*. The ice model used in IceCube simulations and reconstructions has been derived [36, 37] using the calibration LEDs present on the DOMs (Sec. 3.2) and analogous devices present in the previous AMANDA detector, as well as detailed measurements from a dust-logging device used to measure the optical properties of the hole walls during string deployment [38], and measurements from ice cores elsewhere in the south polar ice sheet. These measurements have been extensively tested using muon data in the detector, showing generally good agreement – but nonetheless leaving the modeling of the ice as the largest systematic uncertainty for IceCube measurements.

The ice exhibits significant depth structure (Fig. 3.10). At relatively shallow depths (< 1.4 km), air bubbles sharply reduce the scattering lengths. At greater depths, the high pressures cause dissolved gasses to form calibrates [39], which have very similar refraction indices to ice and so do not form effective scattering centers. At various depths, there are also *dust layers*. Very thin dust layers (~ 1 mm) are deposited by volcanic eruptions, but are thin enough that they do not significantly affect optical transport and are not visible in Fig. 3.10 (only the dust logger can resolve these). The larger visible dust layers are believed to

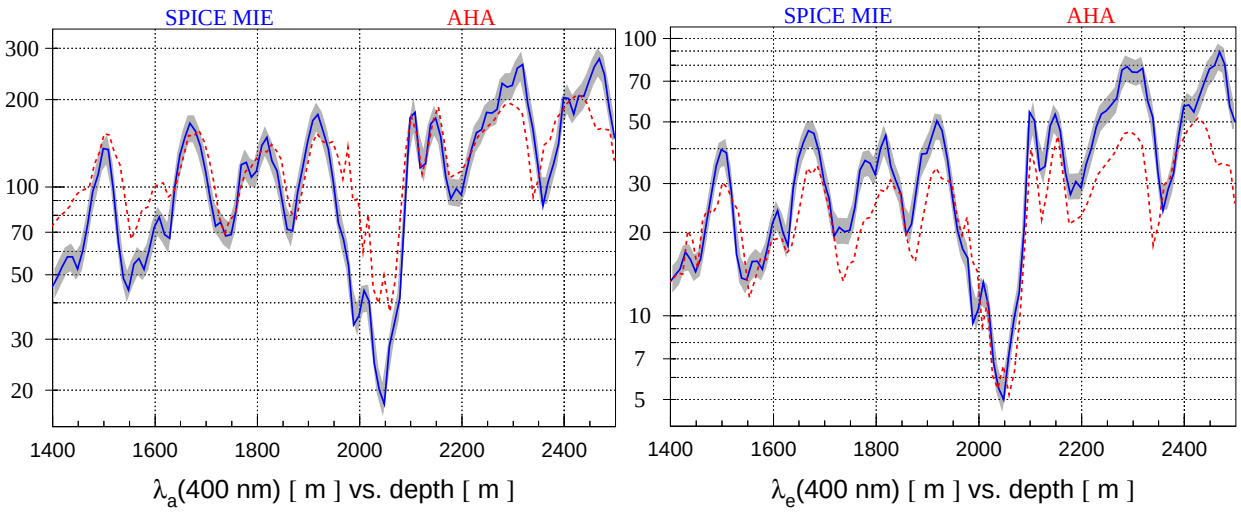


(a) Charge distributions using Wavedeform, showing good agreement between the HLC and SLC readouts, both of which peak at 1.



(b) Charge distributions using previous pulse extraction software (NFE), showing distortions from clipping in SLC waveforms as well as broader HLC distributions.

Figure 3.9 The charge distribution of DOM launches (mostly single photoelectrons) as measured in the 79-string IceCube detector using Wavedeform and using earlier reconstruction software (NFE). The width of this distribution arises from stochasticity in the electron cascade process in the PMT as well as uncertainties in the charge reconstruction. The DOMs trigger based on a voltage discriminator set to 0.2 PE. Correctly calibrated data should produce distributions that are always centered at 1 for both HLC and SLC readout modes (Sect. 3.3). With the exception of reduced resolution for the SLC readouts, and an increased likelihood of pileup in HLC readouts (the multi-photoelectron tail at the right side), SLC and HLC distributions should also have the same shape.



(a) Absorption Length in South Pole Ice

(b) Effective Scattering Length in South Pole Ice

Figure 3.10 Optical properties of the South Pole ice, as determined from AMANDA calibration instruments (AHA, [36]) and IceCube flashers (SPICE Mie, [37]). At the top of the ice, bubbles increase scattering, while at lower levels, high pressures compresses the bubbles into clathrates. Dust concentration traces the prevailing winds and climate over the period of snow deposition in the glacier. Figure from [37].

be the result of climatic changes [38], in which reduced precipitation in the Antarctic combined with higher atmospheric dust concentrations from desertification in South America increase the ratio of the deposition rates of atmospheric dust to snow, thereby increasing the dust concentration for a wide band in the deep ice.

Because the absorption and scattering lengths are similar (Fig. 3.10), no analytic solutions for the light amplitude or time distributions produced by particles in the detector exist. Monte Carlo, however, allows the treatment of both of these and can include the inhomogeneity of the dust layers [40]. Details like the structure of ice inside the holes (Fig. 3.11) and the details of effective simulation and reconstruction using the ice modeling (Appendix A) are also now beginning to be understood, improving the performance of the IceCube detector.

Figure 3.11 Photograph taken by a camera inside a DOM pressure vessel during the 2010-2011 deployment season, showing the clarity of the ice. The rings are spaced 1 meter apart. The green line is a calibration laser. The structure in the bottom right is ice refreezing after deployment.

Chapter 4

Reconstruction

The IceCube detector observes two things: the number and timing of photons arriving at photomultipliers in the ice. The distribution of the detected photons in time and space allows the reconstruction of particle directions. For contained events, IceCube is a calorimeter: the light emitted (and detected) by particles within the ice is proportional to the energy of those particles.

For the majority of events, however, neither the particle vertex nor decay point are within the detector (Fig. 3.2). This is one of the fundamental limits of IceCube reconstruction: non-observation of the neutrino interaction vertex means that no reconstruction can be more accurate than the difference between neutrino direction and observable muon direction, while observing only a segment of the muon track means that a direct calorimetric measurement of the muon or neutrino energy is impossible. These restrictions are, however, not as severe as they would seem. At the high energies we are concerned with, the neutrino interaction is boosted forward, making the neutrino and muon nearly collinear (Fig. 3.3). The high energies of the muons also imply that the muon energy loss rate dE/dx is proportional to the muon energy (Fig. 4.1), allowing a measurement of the mean energy loss rate to be used as a proxy for the muon energy.

4.1 Angular reconstruction

Angular reconstruction in IceCube is generally based on the arrival times of photons at the PMTs. Final reconstruction results are based on a chain of reconstructions, each providing a refinement of the previous one, starting with the simplest and fastest and moving to more complex. Typical angular resolution at the end of the chain is around 1°.

4.1.1 Linefit

The simplest first-guess algorithm is *Linefit*, which applies a simple linear regression to the times and locations of the photons observed at the PMTs (pulses) – in effect fitting a plane wave of light passing through the detector. This can be done analytically, and so is extremely fast, but neglects important information

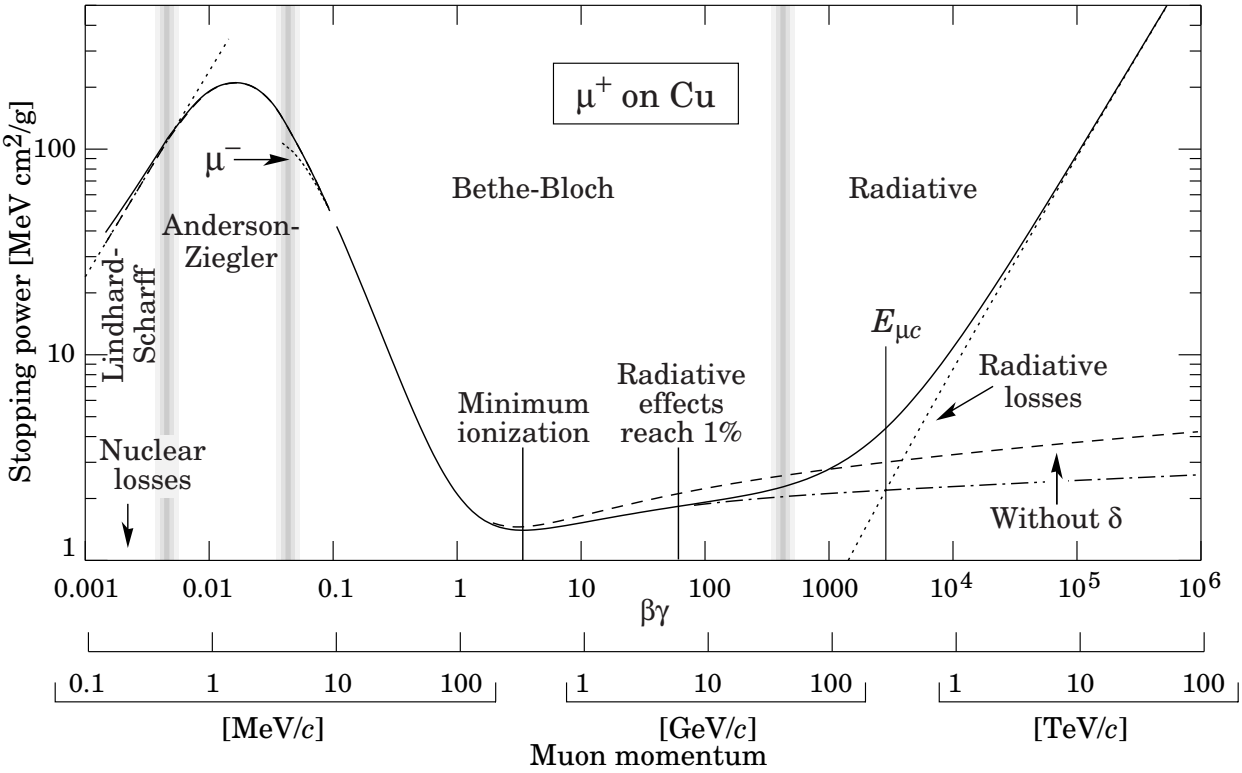


Figure 4.1 Energy loss rate of muons in copper. Above $\sim$ 1 TeV, the energy loss rate is proportional to the muon energy, allowing use of the loss rate to estimate muon energies, even for observations of only partial tracks. Figure from the Particle Data Group [41].

like the scattering of light in the ice and the emission pattern of light from the particle (i.e. the Cerenkov cone). Nonetheless, it achieves a typical angular resolution of a few degrees.

4.1.2 Likelihood fits

The more complex fits use explicit maximum likelihood methods, taking into account the light emission profile and the existence of scattering, if not always the details of the ice. As these use a numerical minimizer and require the evaluation of special functions (in particular, the incomplete gamma function), these typically take much longer computationally than Linefit, but deliver much better angular resolution (degree-scale). The use of the numerical minimizer also means that the fit can be trapped by local minima or fail to converge for other reasons.

The two primary implementations of likelihood fits are *SPE* (Single Photo-Electron) and *MPE* (Multi Photo-Electron). Each of these, despite the name, use the arrival time of the first photon in a particular PMT – which is likely to have undergone the least scattering – to estimate the track direction.

The likelihood that a photon arrives at time t at a particular optical module can be approximated by the Pandel [42] parametrization of the gamma distribution:

$$\begin{aligned} \mathcal{L} &= \tfrac{1}{\Gamma(a)b^a} x^{a-1} e^{-t/b} \\ b &= 1/\left(\tau^{-1} + c/(nx_0)\right) \\ a &= ||\vec{x}||/\lambda \end{aligned} \tag{4.1}$$

where $||\vec{x}||$ is the distance to the track, n is the index of refraction and τ, λ, and x_0 are parameters used to describe the average scattering of photons in the ice. This provides a reasonable (Fig. 4.2) approximation of the light scattering distribution.

For SPE, Eq. 4.1 is used without modification using the time of the first photon at each DOM and taking the product of this likelihood for every hit DOM in the detector. For events where more than one photon has been collected, this is not strictly statistically correct: Eq. 4.1 describes the arrival time distribution of a *random* photon, and choosing the first of N biases the result. However, the likelihood function being evaluated is computationally simple and SPE provides in general better angular resolution than line fit.

The last stop in the chain for this analysis is MPE, which attempts to correct Eq. 4.1 for the statistical problem described above. Instead of using a likelihood based on the arrival time of a random photon, it uses the cumulative distribution function, which involves an incomplete gamma function, and the total charge in each PMT, to compute the likelihood that the first of N photons arrived at a particular time. This is in principle statistically correct – although it neglects variation of scattering and energy loss (Sec. 4.4) – and MPE typically provides angular resolution on the degree or sub-degree scale.

4.2 Energy reconstruction

IceCube is in many respects a calorimeter: the amount of detected light is proportional to the energy deposited within the detector. Using tabulated descriptions of the expected light output in various parts of the detector (Secs. 3.5, 4.3), the deposited energy can be reconstructed from the observed light. For uncontained tracks, the energy deposition rate of the particles is proportional to energy (Fig. 4.1), which can be used to estimate particle energies even in these cases [43].

For a particle depositing k photons in a PMT, we can estimate the particle energy deposition E by comparing the observed number of photons to the expectation Λ for a particle with some reference energy (usually 1 GeV). The detection of light follows a Poisson distribution, and so the likelihood $\mathcal{L}$ can be maximized as follows:

$$\begin{aligned} \mathcal{L} &= \tfrac{\lambda^k}{k!} \cdot e^{-\lambda} \\ &\lambda \to E\Lambda \\ &= \tfrac{(E\Lambda)^k}{k!} \cdot e^{-E\Lambda} \\ \ln \mathcal{L} &= k \ln (E\Lambda) - E\Lambda - \ln (k!) \end{aligned} \tag{4.2}$$

Maximizing this with respect to energy, and adding the contributions from all DOMs:

$$\begin{aligned} 0 = \tfrac{\partial \ln \sum \mathcal{L}}{\partial E} &= \textstyle\sum (k\Lambda/E\Lambda - \Lambda - \cancel{\partial/\partial E \ln (k!)}) \\ &= \textstyle\sum k/E - \sum \Lambda \\ \therefore E &= \textstyle\sum k / \sum \Lambda \end{aligned} \tag{4.3}$$

The generalization allowing additional contributions (e.g. PMT noise) is to replace the substitution $\lambda = E\Lambda$ by $\lambda = E\Lambda + \nu$, where ν is the expected number of noise photons. This unfortunately prevents an analytic solution as in Eq. 4.3. The last few lines then become:

$$\begin{aligned} 0 &= \textstyle\sum (k\Lambda / (E\Lambda + \nu) - \Lambda) \\ \textstyle\sum \Lambda &= \textstyle\sum k\Lambda / (E\Lambda + \nu) \end{aligned} \tag{4.4}$$

This does not have a closed form solution for E, as Λ no longer cancels in the first term and E can therefore not be factored out. The first-order Taylor expansion is:

$$E = \textstyle\sum (k - \nu) / \sum \Lambda \tag{4.5}$$

Typical energy resolution achieved by IceCube by this method is about 30% in the logarithm of the energy for uncontained ~ 100 TeV tracks [43], which is dominated by uncertainty in the neutrino interaction vertex and so the amount of muon energy lost before entry into the detector.

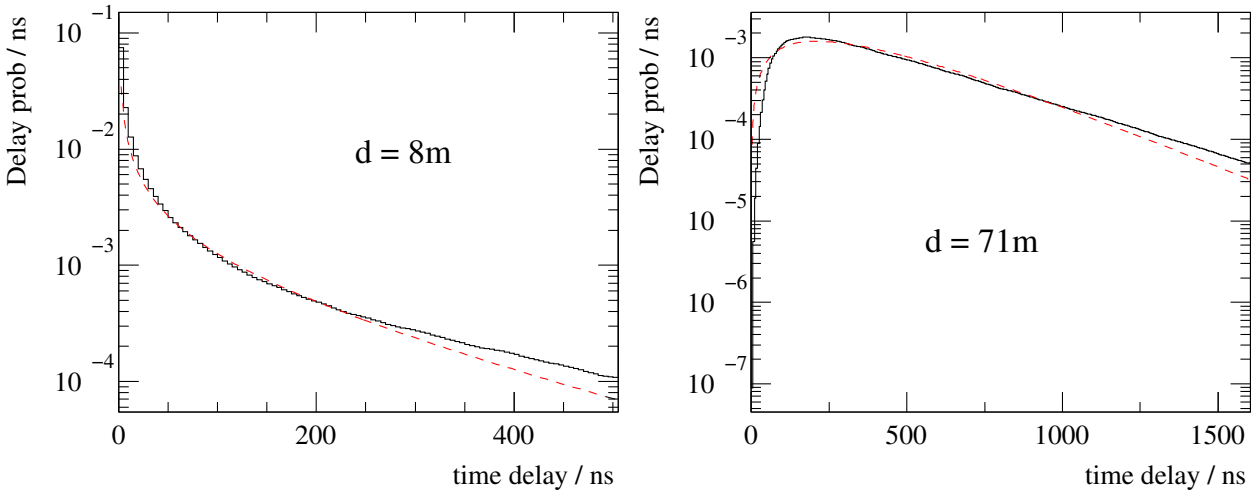


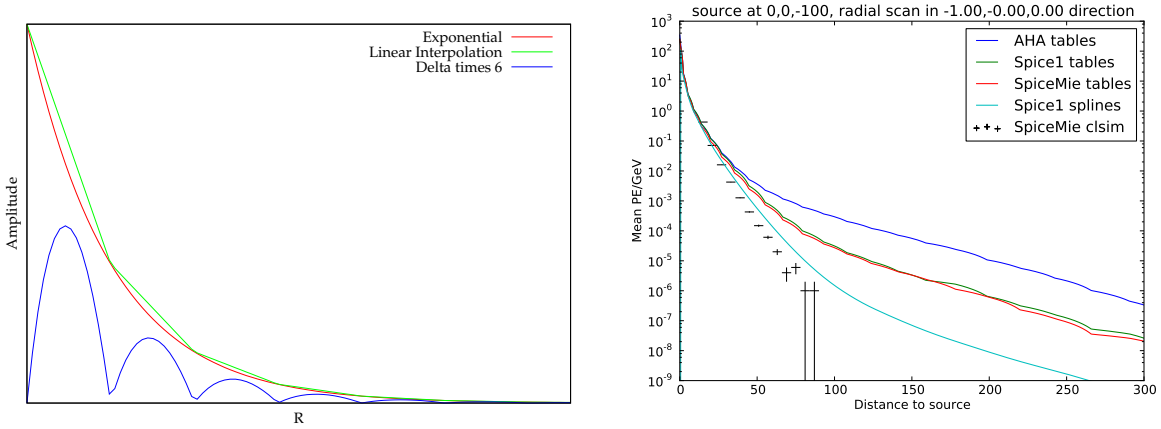
Figure 4.2 Comparison of the time residual (time compared to direct propagation) distributions for the Pandel function (Eq. 4.1, dashed curves) to the detailed simulation in (black). While it describes the general shape of the curve correctly, it typically gives incorrect results for both small and large times. Where the scattering differs substantially from the uniform ice approximation (Fig. A.9), it can differ also in the position of the peak. Figure courtesy of Mike Baker.

4.3 Photospline

While the Pandel function is a good approximation (Figs. 4.2, A.9), it is not perfect, especially when taking into account the inhomogeneity of the ice (Sec. 3.5) and better descriptions of light propagation can be expected to lead to better reconstructions. Because of the ice inhomogeneity, as well as the similarity of the scattering and absorption lengths for optical photons in ice, it is not generally possible to write in analytic form either the photon timing or amplitude distributions. They can, however, be evaluated by Monte Carlo simulation and tabulated [40].

Direct use of these tabulated distributions presents difficulties, however, as the parameter space is very large. Even exploiting the X-Y symmetry of the ice structure, the minimum coordinate system is the zenith and Z position of the source particle (θ, z) and the relative spatial coordinates of the receiver (ϕ, r, ρ), as well as the time t. To reach the intrinsic uncertainties in the detector ($\sim$ 1 meter, $\sim$ 1 degree, $\sim$ 1 ns) with the size of IceCube implies $\sim 4.5 \times 10^{17}$ table cells, which would occupy over a million terabytes of memory in single precision. Clever binning (coarser further from the source where features are expected to be washed out) can reduce this requirement dramatically, but only to a few terabytes before binning artifacts become apparent (Fig. 4.3).

At several terabytes, however, the tables are still too big to fit into memory on almost all computers (random access is required for reconstruction). Moreover, the large number of bins requires the propagation of



(a) Illustration of the effect of linearly interpolating a rapidly decreasing function, for example the photon density as a function of distance from a source. Note that the residual function is always positive – the linearly interpolated function systematically gives too much light.

(b) Comparison of coarse tabular and fine tabular (with spline fit) photon density tables. As in the left panel, the coarse tabular distribution functions are systematically higher than the fine-binned tables or direct photon tracking (CLSim). The difference between the fine-binned (spline) tables and direct tracking is due to simulation of a slightly different photon propagation model.

Figure 4.3 Effects of binning on expected photon densities. Coarsely-binned tables exhibit a number of numerical artifacts. Shown in the left panel is a toy model of one of these effects, based on the interpolation used, which causes a systematic overestimate of light estimates as well as a characteristic "scalloping" of the resulting densities. Both effects are clearly visible in the right panel.

enormous numbers of photons to provide adequate statistics. By fitting the tables with splines (Appendix A), the size of the tables can be reduced drastically while providing improved table smoothness and interpolation.

4.4 Millipede

Energy deposition in IceCube by high-energy muons is extremely stochastic, with large variations in the energy loss rate from catastrophic energy loss processes (bremsstrahlung, photonuclear interactions, pair production, etc.). This both complicates energy estimation and the construction of PDFs for angular reconstruction. In addition, not all particles of interest are through-going tracks: starting and stopping muons, taus, and ν_e interactions all produce tracks with wide variation in energy loss (very concentrated, zero for long stretches, etc.).

These problems can be solved by reconstruction of the detailed pattern of energy loss along the track. In IceCube, this is more complicated than in the usual segmented calorimeter because there are no barriers to light transport within the detector. As such, every PMT readout is a combination of light from everywhere along the track emitted at many places within the ice.

The fact that this combination is a *linear* one, however, makes the problem tractable. We have already considered energy reconstruction in the presence of multiple light sources in Sec. 4.2 when discussing the inclusion of PMT noise. Eq. 4.4 can be generalized by replacing the substitution:

$$\lambda \to E\Lambda + \nu \tag{4.6}$$

with one where the expected photon count λ is the sum of photons from multiple sources (λ_i) and the noise rate ν:

$$\lambda \to \sum E_i \Lambda_i + \nu \tag{4.7}$$

where each E_i is the energy deposition at a particular place and Λ_i is the expected light yield in a particular photomultiplier from light source i. The maximum likelihood equation 4.2 can then be rewritten in terms of vector operations:

$$\begin{aligned} \ln \mathcal{L} &= k \ln \left(E_i \Lambda^i \right) - E_i \Lambda^i - \ln (k!) \\ &= k \ln \left(\vec{E} \cdot \vec{\Lambda} \right) - \vec{E} \cdot \vec{\Lambda} - \ln (k!) \\ \sum \ln \mathcal{L} &= \textstyle\sum_j k_j \ln \left(\vec{E} \cdot \vec{\Lambda}_j \right) - \sum_j \vec{E} \cdot \vec{\Lambda}_j - \sum_j \ln (k_j!) \end{aligned} \tag{4.8}$$

Like Eq. 4.4, this has no analytic maximum for $\vec{E}$, but can be solved to first order (the approximately Gaussian error regime, applicable at high energies):

$$k_j = \vec{E} \cdot \Lambda_j + \nu \tag{4.9}$$

Introducing the matrix $\mathbf{\Lambda}$ for the predicted light yield at every point in the detector from every source position at some reference energy, this can be rewritten in terms of a matrix multiplication:

$$\vec{k} - \vec{\nu} = \mathbf{\Lambda} \cdot \vec{E} \tag{4.10}$$

This can be inverted by standard linear algebraic techniques to find the best-fit $\vec{E}$. We want, however, to incorporate additional physical constraints. In particular, negative energies are unphysical and should not be present in the solution even though a matrix inversion may often yield solutions with negative terms (Fig. 3.7). The solution is to use a non-negative least squares algorithm (Sec. A.4.1), as in Wavedeform (Sec. 3.4.1), which provides a high-quality fit to the data (Fig. 4.4) by use of a linear deposition hypothesis with possible light sources every few meters along the track. Additional physical constraints (regularity of direct Cerenkov light from the muon, preference for small energy losses, etc.) can be included by the use of regularization (Sec. A.3.3). Timing can also be included by dividing the PMT readouts for bright events into multiple time bins and interpreting $\vec{k}$ and $\mathbf{\Lambda}$ as referring to the light per time bin instead of per PMT – the math remains the same in this case.

Since the expected light distribution is heavily influenced by non-uniformities in the energy-loss rate, the best-fit energy loss distribution $\vec{E}$ can then be used to construct better PDFs for angular reconstruction. Iteratively refitting for energy and computing the likelihood 4.8 in a numerical minimizer allows angular reconstruction with knowledge of the ice structure and marginalizing over the energy loss distribution. In initial tests (Table 4.1), this seems very promising, delivering angular resolution up to 10 times better than Pandel-function based approaches at high energies. Further work is required to improve the robustness of the minimizer to low-quality seed tracks.

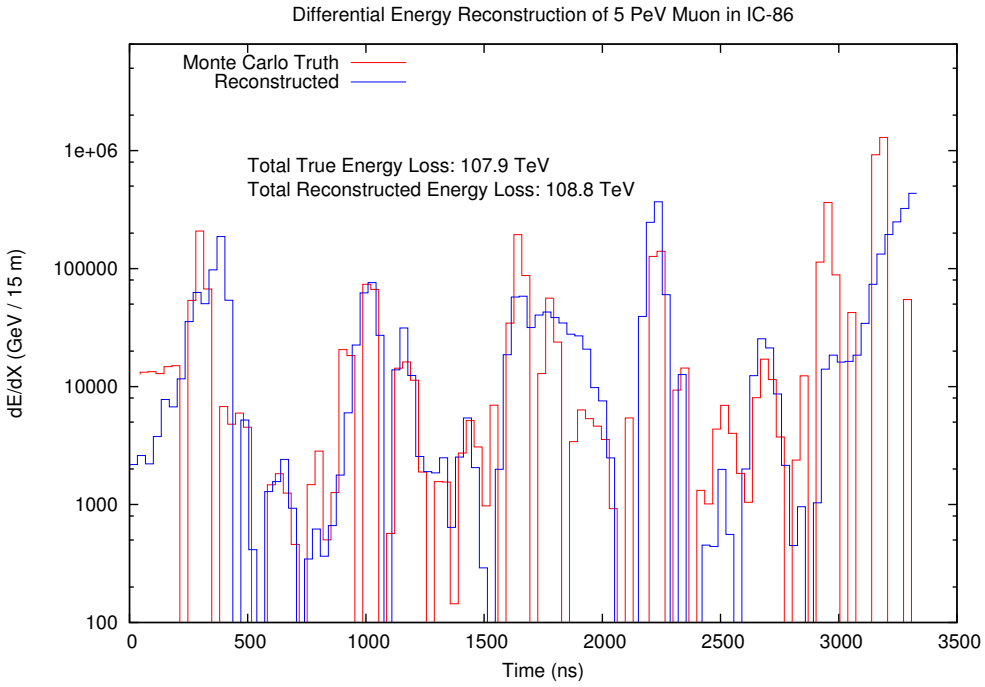


Figure 4.4 Reconstruction of the energy deposition of a 5 PeV single muon using Millipede using a 15 meter cascade spacing. The total reconstructed energy within the detector differs from the truth by less than 1%.

Reconstruction	Median $\Delta\Psi$ (degrees)	Convergence Rate	Execution Time (s)
Line Fit	2.972	100%	~ 0
MPE, linefit seed	11.35 (0.601)	60%	1.56
MPE, perturbed MC seed	0.758 (0.601)	50%	1.22
MPE, best	0.679	80%	2.78
Millipede	0.0511	100%	57.3

Table 4.1 Comparison of angular resolution ($\Delta\Psi$) and execution time for several reconstruction algorithms for a sample of PeV energy neutrinos. Millipede provides a factor of 10 improved angular resolution compared to the next-best reconstruction (MPE), at the price of much greater computational requirements. Averages were taken only over fits that converged, and numbers in parentheses indicate averages taken only over the subset of converged events where the fitter does not appear to have failed despite reporting convergence. MPE best refers to the result with the best final likelihood of the two other MPE fits. Minimization was performed with the Minuit2 SIMPLEX algorithm and the reconstructions, unless otherwise noted, were seeded with a 2° shifted version of the true track.

Chapter 5

Analysis

In principle, neutrino searches from Gamma-Ray Bursts (or any transient search) are straightforward: the signal is neutrinos coming from the right place at the right time, and off-time intervals provide high-quality sidebands that make background estimation trivial. In addition, the background rate from atmospheric neutrinos is low enough (10^{-7} s^{-1} degree^{-2}) that the analysis is nearly background-free. The definition of the signal is, however, less clear-cut than it might appear, in particular with regard to the definition of "at the right time".

The analysis presented here uses data from two years of IceCube during construction (2008-2010). The first year uses the first 40 strings of IceCube and looks only for Northern hemisphere bursts (using the Earth to exclude cosmic rays). The second, using the 59-string data, expands this search to include high-energy neutrinos in the south hemisphere, visible above the cosmic rays. Both are optimized to look for generic neutrino-GRB correlations rather than to test a particular model. Since the number of neutrinos from any particular GRB is expected to be small (on order 0.01 events), all of the GRBs are *stacked* – data relative to each burst are treated as a single sample, increasing the number of expected events to a few (Table 5.1).

Model	Normalization	Predicted ν
Ahlers et al. 2011 [21]	Cosmic Rays (neutrons)	113
Rachen et al. 1998 [20]	Cosmic Rays (neutrons)	84
Waxman 2003 [26]	Cosmic Rays	27
Guetta et al. [23]	γ spectrum	14

Table 5.1 Predicted event rates for various theoretical models in IC40+IC59. Rates are at the level used in the analysis, and are computed for the expected values of various unknown quantities. Models normalized to the γ spectrum use the measured γ fluence to estimate the total energy of the burst and assume some fixed fraction of that is present in protons, whereas cosmic ray-normalized models assume that the entire extragalactic cosmic ray spectrum has been sourced from 667 GRBs per year, of which the ones we catalog are a representative sample.

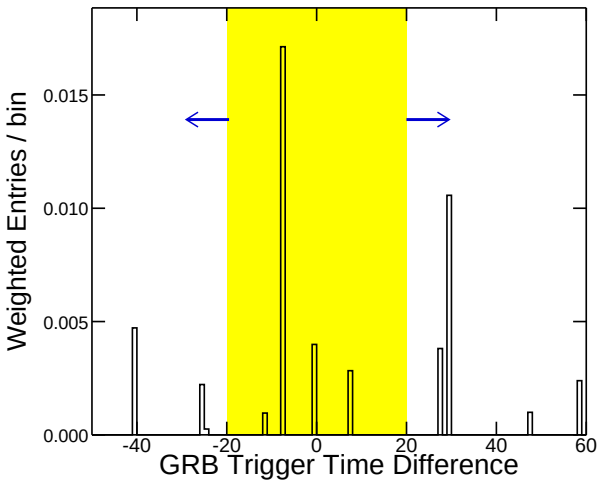


Figure 5.1 Schematic of the expanding time windows used in the analysis. For each value Δt, events within $\pm\Delta t$ of the burst (yellow region) are summed, and examined for an excess. Then the time window is widened by an equal amount in both directions and the search repeated.

5.1 Time Windows: what does simultaneous mean?

Neutrinos from GRBs are expected to arrive more or less coincident with the gamma rays. However, changes in the optical depth of the GRB fireball may result in the target gammas not reaching the Earth (e.g. the neutrinos may be produced early in the fireball, when it is still optically thick), resulting in small shifts between the neutrino and gamma time. It may also be possible to see neutrino emission from processes other than the fireball $p\gamma$ process, potentially yielding information about the GRB progenitor object. Maximum model-independence, therefore, implies searching a range of possibilities for "coincidence".

Since we are integrating the fluxes from many bursts, each potentially with different neutrino timing, and with those time offsets scaled by variable (and unknown) redshifts, the simplest approach is to model time offsets as a spread, rather than a shift, in emission time. This is incorporated into the analysis by searching within a variable window Δt around the burst time. This window is widened iteratively, and each is checked for a signal excess (Fig. 5.1). For this analysis, the window was widened 1 second at a time up to $\pm$ 1 day, where background becomes dominant. This is much longer than the typical length of the GRB gamma emission (seconds to minutes) and also reaches the limits of what can reasonably be attributed to the GRB fireball.

A standard problem with parameter-space scans of this type is the so-called Look Elsewhere Effect, which can substantially reduce the significance of an excess due to the many opportunities to find one. Naïvely, the 86400 time windows examined would constitute 86400 distinct trials, turning, for example, a 5σ excess

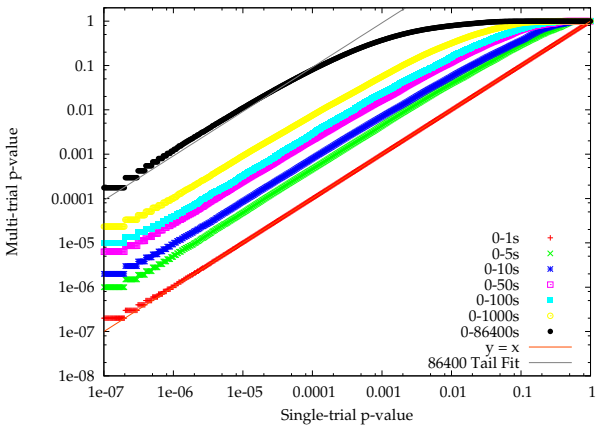


Figure 5.2 Effects of increasing the size of the signal time window Δt. The plot relates the P-values of an excess interpreted for a single time window alone (X-axis) to the probability of having seen any bin with such an excess anywhere in the data set (Y-axis). This has an asymptotically linear tail on the left, the slope of which is the number of effective trials, and converges to one on the right. For small values of Δt, adjacent 1-second spaced intervals are nearly independent, resulting in a linear increase in the number of effective trials. As the number of intervals increases, the number of effective trials becomes strongly sublinear. For example, the number of effective trials for 1000 bins is only a factor of 10 larger than for 5.

into a 2σ one. In the context of small numbers of events, this would potentially be fatal. However, the trials in question are not entirely distinct. As an example, the time window with $\Delta t = 10000$ seconds shares 99.99% of its events with $\Delta t = 10001$. As the time window expands further, the inter-trial correlation rapidly increases, yielding a (relatively) low effective trials penalty (Fig. 5.2) of only a few hundred.

5.2 Backgrounds

The primary backgrounds for GRB searches in IceCube are products of cosmic ray air showers at the Earth: muons from the showers in the Southern hemisphere and neutrinos from showers in the Northern hemisphere. In their respective hemispheres, these backgrounds are *irreducible*, in the sense that their detector signatures (a muon) are identical to the expected signal. They can be separated from GRB neutrinos by the time and direction of their arrival (neither background should be correlated with GRBs) and to a lesser extent by the energy, although this can inject some model-dependence since it involves assumptions about the energy of neutrinos from GRBs. The rates of these backgrounds are very different: 2500 Hz of cosmic ray muons in the south compared to 3 mHz of atmospheric neutrinos in the north.

Event selection in the north need not take particular effort to exclude the atmospheric neutrino background since the rate per unit solid angle is quite low. A much larger background is caused by downgoing cosmic ray muons misreconstructed as upgoing. The rate of this is low (10^{-3}), but large in comparison to

both the atmospheric neutrino rate and the rate of the expected signal. Reconstruction failures have a variety of causes; the most common is pileup. The detector is large enough that multiple muons from unrelated air showers can be present simultaneously and are contained within the same trigger (e.g. Fig 5.3). Most IceCube reconstructions reconstruct only a single track and fitting such a coincident pair to a single track will produce an essentially random direction (i.e. is upgoing half the time). While such events form the vast majority ($> 99.9\%$) of upgoing-reconstructed events, they have a fundamentally different signature than the signal and so can be removed without injecting bias into the event selection.

5.3 Event Selection

The low backgrounds for GRB searches mean that the sensitivity is almost entirely dominated by high signal acceptance, not by background rejection. Further, the total event sample (~ 200 million events per year) is small enough that cuts are not critical to make data processing tractable.

For a typical binned analysis, the counting result N and variance σ^2 are both simply the number of events after final cuts:

$$\begin{aligned} N &= N_{cuts} \\ \sigma^2 &= N_{cuts} \end{aligned} \tag{5.1}$$

This can be rewritten as a sum over all events, before cuts, in terms of the indicator function $I(\vec{x}_n)$, which is 1 if the nth event passes the cuts and 0 otherwise:

$$\begin{aligned} N = N_{cuts} &= \textstyle\sum_n I(\vec{x}_n) \\ \sigma^2 = N_{cuts} &= \textstyle\sum_n I(\vec{x}_n)^2 \end{aligned} \tag{5.2}$$

The variance in N, σ^2, can be rewritten in terms of the variance of the indicator function σ_I^2:

$$\sigma^2 = N_{total}\sigma_I^2 + \frac{\left(\sum_n I(\vec{x}_n)\right)^2}{N_{total}} = N_{total}\sigma_I^2 + \frac{N^2}{N_{total}} \tag{5.3}$$

For fixed $N = \sum_n I(\vec{x}_n)$ and N_{total}, it is therefore clear that the variation in the measurement result, σ, can be reduced by minimizing the variance of the event classifier σ_I^2. An indicator function (taking values only at the extreme ends of its range) has maximum possible variance – replacement of the indicator function by a real-valued *weighting function* with the same mean thereby reduces the measurement variance. This allows measurements at final cut level to have effectively sub-Poissonian errors, relative to an analysis with a smaller data sample due to hard cuts, improving the limit-setting ability and potential significance of the measurement (Fig. 5.7).

The natural choice for the weighting function $I(\vec{x}_n)$ would be the probability that a particular event is a signal. This has a number of advantageous properties. The range of probabilities is $0 - 1$, fulfilling the

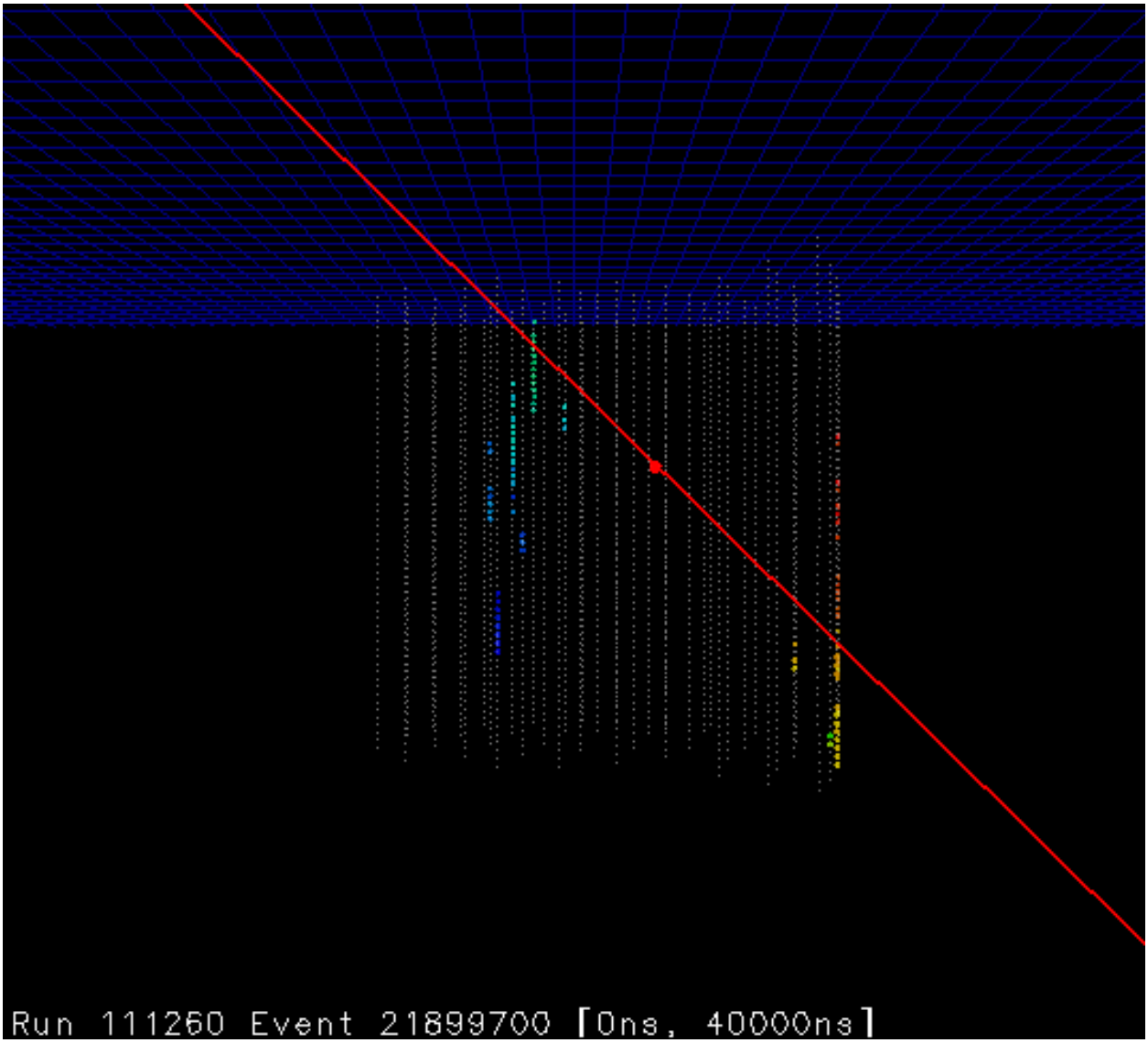


Figure 5.3 Two muon from unrelated air showers in the same trigger window. The first moves down the right side of the detector (red and yellow points) and the second down the left (green and blue). The reconstruction (red line) is the best fit single muon track, which is upgoing, mimicking a neutrino. This type of event is the primary background to northern hemisphere neutrino searches.

criterion above. The sum $\sum_n I(\vec{x}_n)$ is the expectation value of the number of signal events, and can be interpreted in a natural way. Finally, this approach is asymptotically convergent to maximum likelihood methods [44], which are otherwise impracticable for very large datasets.

The determination of this probability presents some difficulties. Standard machine-learning techniques tend to produce as output opaque classification parameters not easily mapped to probabilities and more information is required than can be usefully provided manually. The simplest method to compute such a probability is to histogram known samples of background- and signal-like events in a few interesting parameters and divide those histograms to form an odds ratio, which can be mapped onto a probability using Bayes's Theorem (where λ is the *a priori* odds that an event is a neutrino and $S(\vec{x}_n)$ and $B(\vec{x}_n)$ are the densities of the signal and background histograms, respectively, for an event's parameters:

$$\begin{aligned} O &= \frac{S(\vec{x}_n)}{B(\vec{x}_n)} \\ P_{sig} &= \frac{\lambda O}{1+\lambda O} \end{aligned} \tag{5.4}$$

While the variables in question could conceivably be the output of a machine classifier, acceptable performance was obtained using raw event parameters. Since using these avoids the introduction of the extra indirection of machine classifiers, they were used instead.

One of the difficulties in the computation of this quantity is the potential for large fluctuations in the odds ratio O due to low or absent statistics in the appropriate histogram cell. This problem can be alleviated as in Section 4.3, by fitting a multi-dimensional spline surface to the input histograms (Fig. 5.4, Appendix A).

The weighting function described here depends primarily on reconstruction-quality parameters – the number of hit PMTs near the track (NDirC), the number of hit PMTs over all (NChan), and the final likelihood of the angular reconstruction (RLogL). For a purely northern-hemisphere analysis like the IC40 one, the cosmic-ray background is entirely misreconstructed events, so reconstruction quality is strongly related to whether a particular muon was neutrino induced. Energy-related terms were avoided to avoid bias to a particular energy spectrum and are unnecessary in the face of the very small atmospheric background. For the 59-string portion of the analysis, which included both hemispheres, zenith angle was added as well to allow for the zenith-dependent cosmic ray background. These perform well in separating neutrino simulation from background events (Figs. 5.5, 5.6).

To incorporate the angular separation from the GRB, an additional term was added to the weighting function turning it into the overlapping probability density that an event is a neutrino and coincides in space with a GRB. The uncertainties of both are asymptotically two-dimensional Gaussians [45], and so the overlap integral of those distributions is:

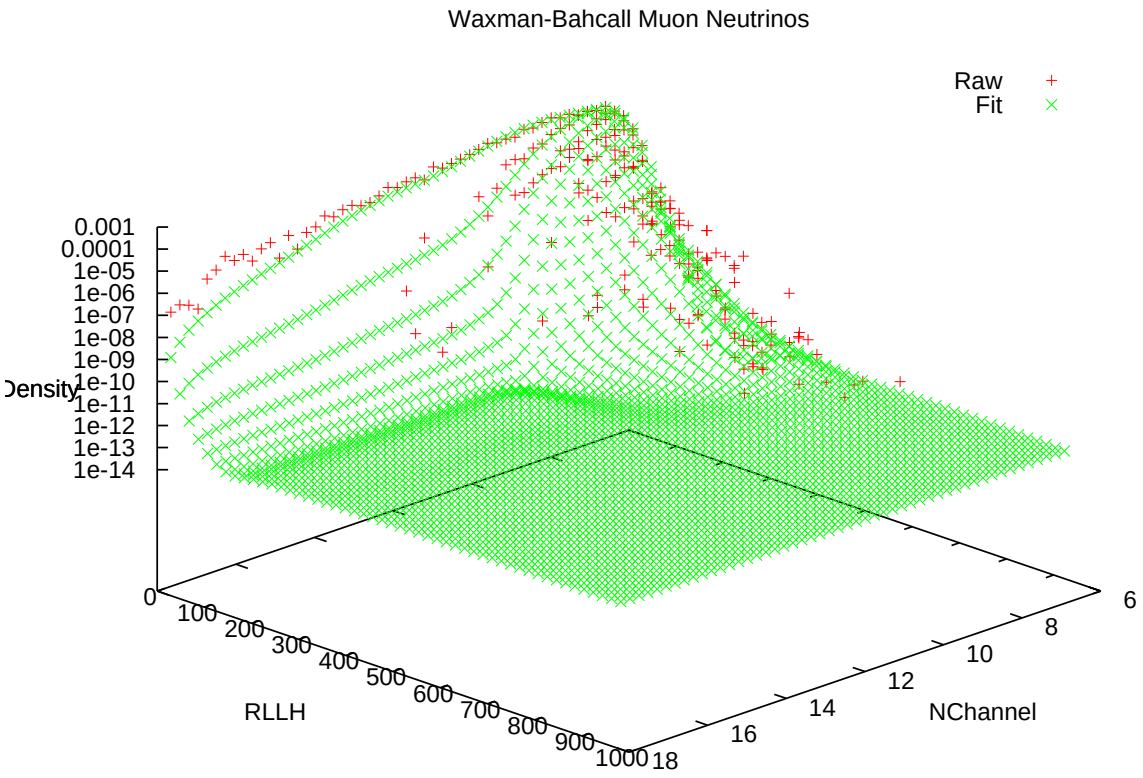


Figure 5.4 Example of spline fit in event classification. In red are shown the raw counts in the histogram cells and in green the result of the fit. An artificial floor of 10^{-13} has been established to avoid wild extrapolation. RLLH is the same quantity as RLogL: the final likelihood of the MPE reconstruction, divided by the effective number of degrees of freedom.

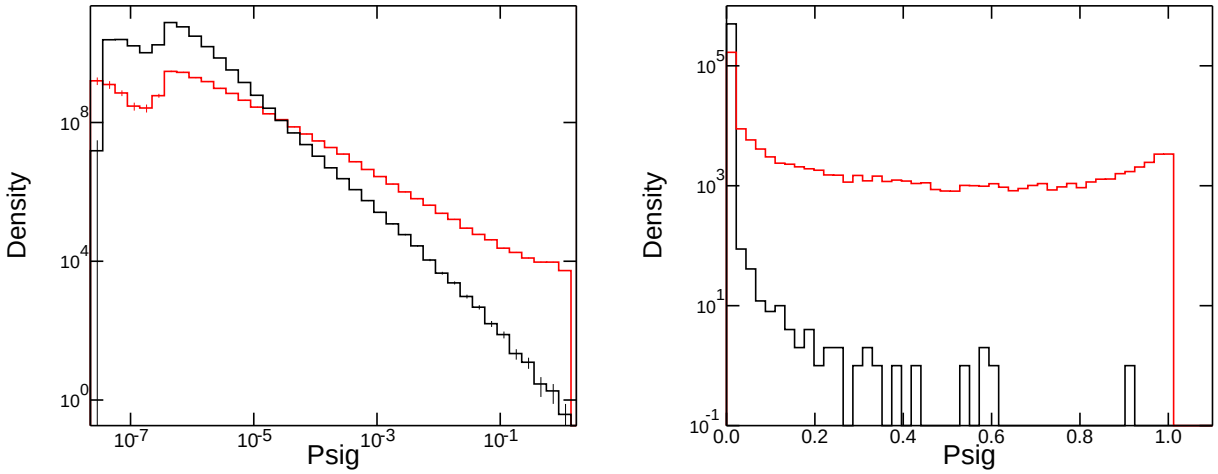


Figure 5.5 Overview of the overall performance of event selection used in the IC40 analysis. Distributions of the weighting function P_{sig} are shown for simulated neutrinos (red) and for detector data (black), which are overwhelmingly misreconstructed cosmic rays. The mean weight for simulation is much higher. The data shown are for a full day and should be contaminated with several tens of atmospheric neutrinos; these are the few high-weight data events visible in the right panel. The two panels differ in the scale of the X-axis.

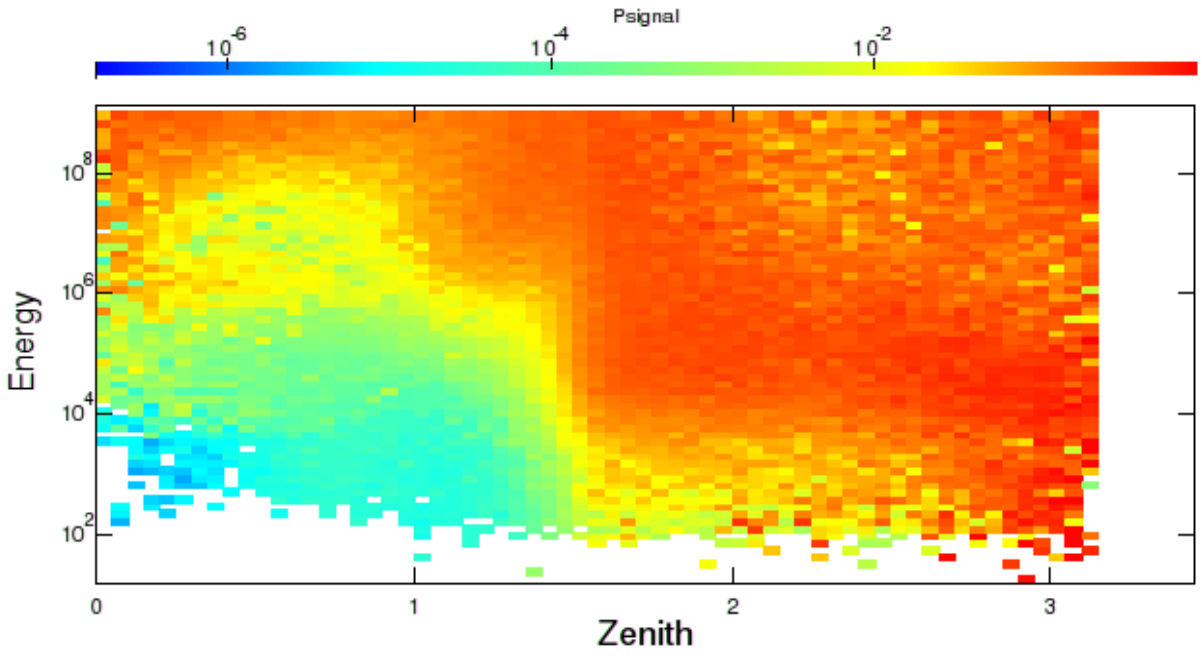


Figure 5.6 Overview of the performance of the IC59 event selection on a sample of simulated neutrino events. In the northern hemisphere, where reconstruction quality is sufficient to identify neutrino candidates, the selection is nearly energy-independent. Above the horizon, only high-energy events can be distinguished from cosmic rays. The X-axis is in units of radians, which 0 being straight down-going and π straight up-going. The Y-axis is in units of GeV, and refers to the neutrino energy.

$$
\begin{aligned}
I(\vec{x}_n, \psi_n) &= P_{sig}(\vec{x}_n) \frac{1}{2\pi(\sigma^2_{\mathrm{GRB}} + \sigma_i^2)} \exp\left(-\frac{|\vec{\psi}_{\mathrm{GRB}} - \vec{\psi}_i|^2}{2(\sigma^2_{\mathrm{GRB}} + \sigma_i^2)}\right) \\
\vec{\psi}_{\mathrm{GRB}} &= \text{GRB coordinates} \\
\vec{\psi}_{\mathrm{i}} &= \text{Reconstructed coordinates of neutrino candidate } i \\
\sigma_{\mathrm{GRB}} &= \text{Position uncertainty of GRB} \\
\sigma_{\mathrm{i}} &= \text{Position uncertainty of neutrino candidate } i
\end{aligned}
\tag{5.5}
$$

5.4 Performance

This approach provides substantially larger signal acceptance than previous techniques (Fig. 5.7), while maintaining high sensitivity (limits comparable to or better than other searches). In addition to the model-independence afforded by the variable time windows, these higher event rates allow an increased chance of detection. Use of weighting instead of straight cuts typically improves effective areas by $\sim 30\%$, with a $\sim 10\%$ increase in sensitivity (many of the extra events are low quality, so the sensitivity increase is smaller than the acceptance one).

For the 59-string portion of the analysis, neutrinos and GRBs from the southern hemisphere were included in the search. Typically, IceCube's greatest sensitivity is to neutrinos from the northern hemisphere. The bulk of the Earth can be used as a cosmic-ray muon shield, attenuating the muon flux to zero. Any observed upgoing muon can then only be result of a neutrino interaction. In the southern hemisphere, however, neutrinos are mixed in with a large (2500 Hz at detector depth) flux of muons produced in cosmic ray showers. This makes identification of neutrino events challenging. For the purposes of a search for correlations with GRBs, however, this constraint is much less severe. The constraint of spatial and time correlation reduces the background substantially (~ 0.1 s^{-1} deg^{-1}). Additionally, because neutrinos from GRBs are believed to be at high energies, above most of the cosmic ray muons, particle energy can be used to discriminate between possible neutrino candidates and muons produced in air showers (Fig. 5.6).

The southern hemisphere also provides somewhat higher effective areas than the northern one at the high energies where GRB neutrinos are expected (Fig. 5.8). At very high energies ($\gtrsim 300$ TeV), the neutrino-nucleon cross section becomes high enough that neutrino absorption in the Earth causes significant decreases in the neutrino flux arriving from the northern hemisphere. The comparatively small detector overburden in the southern hemisphere then allows higher efficiency detection of very high energy neutrino fluxes.

The largest systematic uncertainty in this search is from uncertainties in the neutrino-nucleon cross sections (Fig. 5.9) at high energies since the energy range being probed here is out of the range of laboratory measurement. The extrapolations from lower energies that we use (CTEQ 5 [47]) are believed to be accurate to better than 10%, but, as the physics in this energy is essentially unknown, it is possible that deviations from the CTEQ 5 cross sections are much larger than this value.

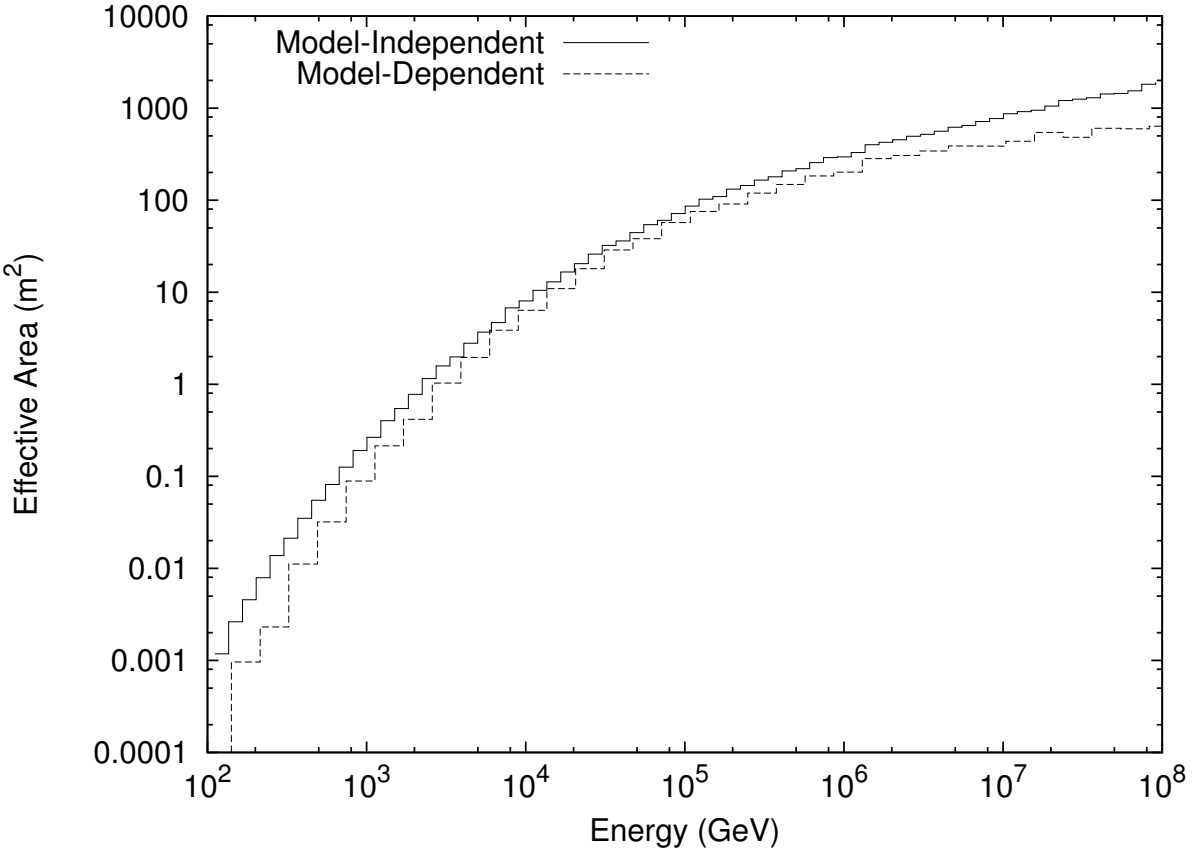


Figure 5.7 Effective area of the IceCube neutrino observatory using the event selections of the model-dependent [24, 46] and model-independent (this work) analyses, averaging over the 40- and 59-string detector configurations and zenith angles according to the distribution of bursts in the catalog (Appendix B). The effective area of the model-independent event selection is in general somewhat larger, due to using a weight scheme instead of hard cuts. The model-independent average effective area includes the southern hemisphere for the 59-string portion of the analysis. Data files for this figure can be found in the supplemental information of [46].

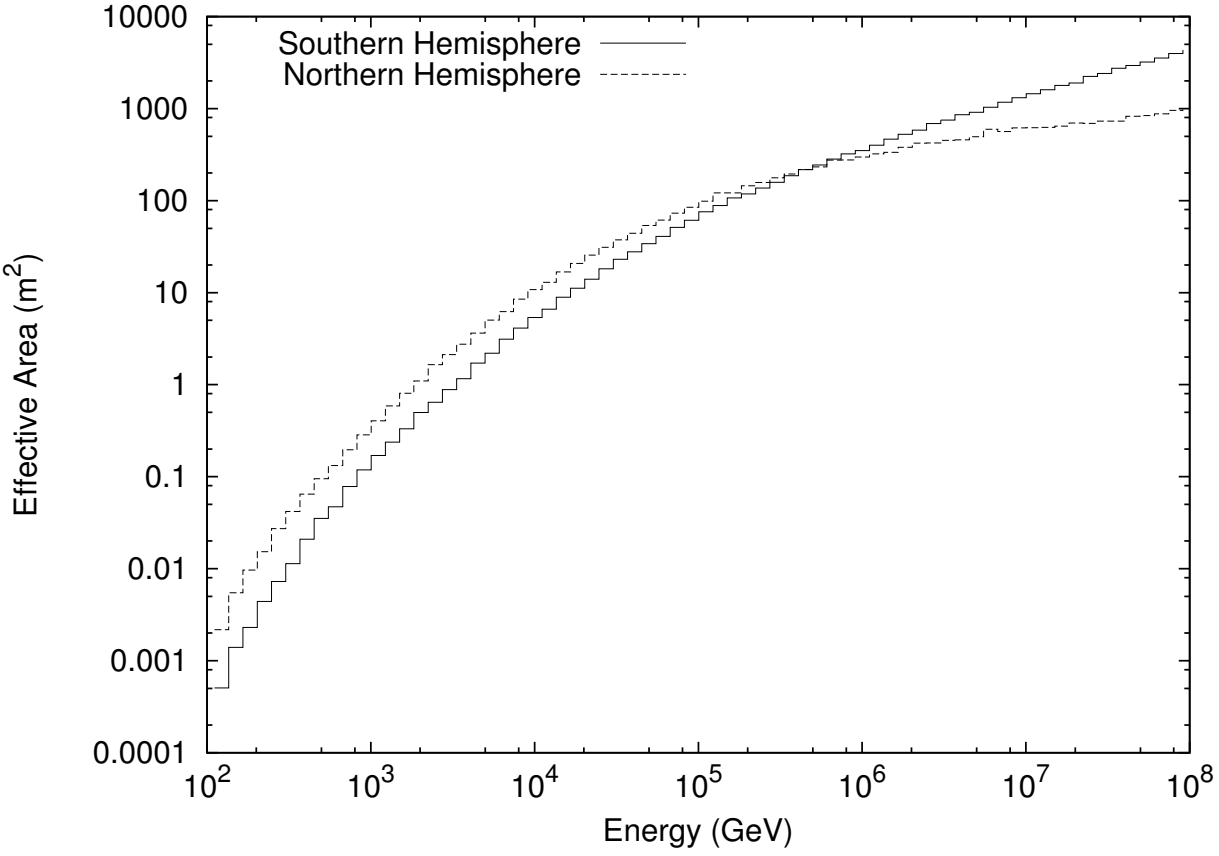


Figure 5.8 Effective area of the IceCube neutrino observatory in its 59-string configuration to fluxes of muon neutrinos from the northern and southern hemispheres, using the event selection from the model-independent analysis. At low energies, the ability to use the Earth to filter out neutrinos from cosmic ray air showers reduces backgrounds and improves the sensitivity of the detector to neutrinos from the northern sky. As the neutrino energy increases, so does the neutrino-nucleon cross-section (Fig. 5.9), increasing the neutrino interaction probability and the chances of a detection in IceCube. At very high energies, backgrounds from cosmic-ray muons are substantially reduced and the neutrino-nucleon cross-section becomes large enough that northern neutrinos will be absorbed in the Earth before reaching the detector, making the southern hemisphere the region of highest sensitivity for $E_\nu \gtrsim 1$ PeV. Note that the effective area is not a direct estimate of the sensitivity, due to variable backgrounds as a function of zenith angle and energy. In general, backgrounds are highest in the southern hemisphere and at low energies. Data files for this figure can be found in the supplemental information of [46].

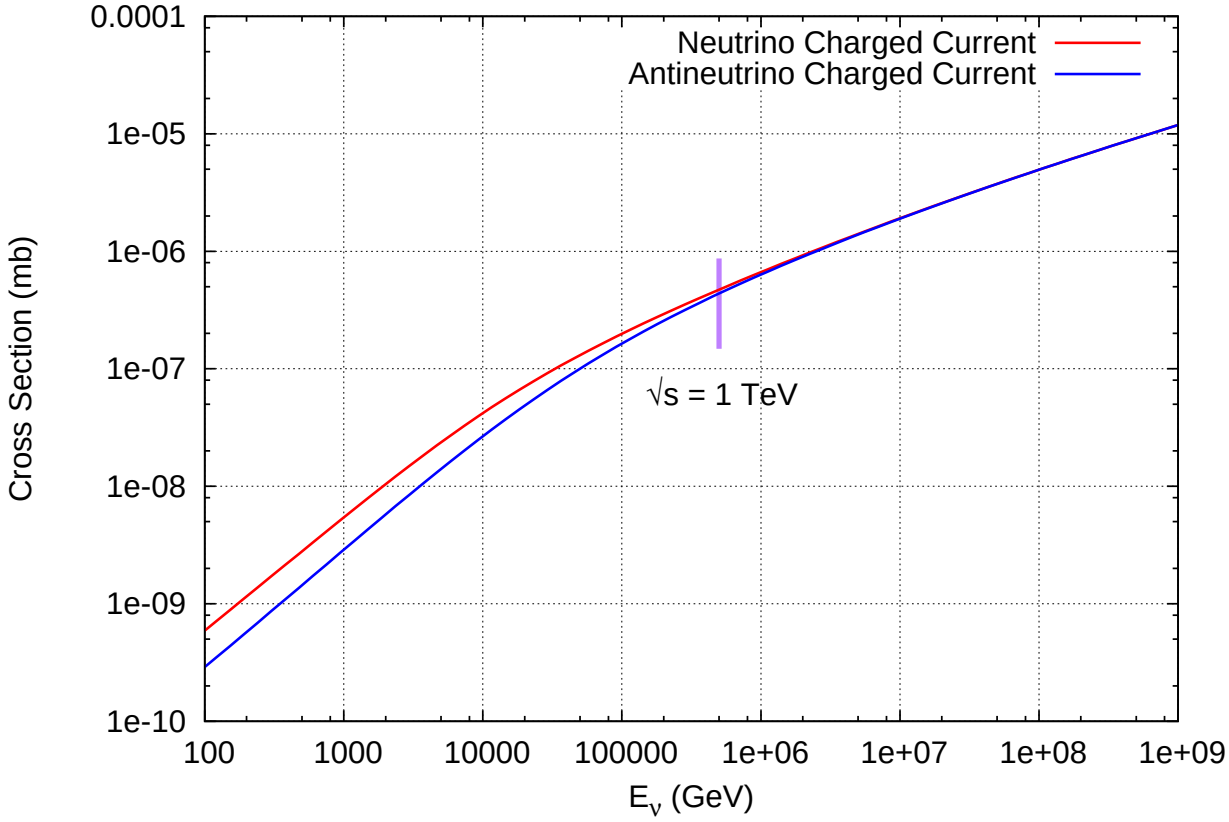


Figure 5.9 Cross sections for neutrino nucleon interactions from CTEQ5 [47]. At low energies, the interaction cross-section is dominated by interactions with the nuclear valence quarks, leading to lower cross sections for antineutrino interactions. At high energies (above $\sqrt{s} = 1$ TeV), interactions with sea quarks become dominant, equalizing the cross sections. Note that at these energies, the cross sections are extrapolated: no precision measurement of any physical process exists for $\sqrt{s} \gg 1$ TeV.

The effect of systematic uncertainties from detector effects is minimal. Ample sidebands from off-source and off-time regions remove any dependence on Monte Carlo from background estimation. Since the immediate quantity of interest in a GRB neutrino search is simply the number of events present, rather than a spectrum, systematic effects influencing the presented result would have to cause such events either not to trigger the detector or to be incorrectly reconstructed. As the energy range at which GRB neutrinos are expected ($\gtrsim$ 100 TeV) is far above the detector threshold (a few hundred GeV), changes in, for example, the PMT quantum efficiency have only minimal effects on the trigger efficiency. The low background also implies that the tolerance for systematic pointing uncertainties is very large. In aggregate, detector systematics are responsible for only 2-5% systematic uncertainties in IceCube sensitivity to these types of fluxes.

Chapter 6

Results

During the 59-string data taking period (May 21, 2009 - May 31, 2010), 190 GRBs were observed and reported via the GRB Coordinates Network [15], with 105 in the northern sky (Sec. B.2). Most of these were reported by Fermi GBM, with additional contributions from Swift-BAT and Fermi-LAT. Of those GRBs, 9 were not included in our catalog due to detector downtime associated with construction and calibration. Two additional GRBs were included from test runs before the start of the official 59-string run. 117 northern-sky GRBs (Sec. B.1) were included from the 40-string period (April 5, 2008 - May 20, 2009) to compute the final combined result. GRB positions were taken from the satellite with the smallest reported error, which is typically smaller than the IceCube resolution (0.6° for $E_\nu \gtrsim 100$ TeV). The GRB gamma-emission start (T_{start}) and stop (T_{stop}) times were taken by finding the earliest and latest time reported for gamma emission.

After unblinding, two candidate events were observed at low significance, one 30 seconds after GRB 091026A (Event 1, zenith angle 2.03°) and another 14 hours before GRB 091230A (Event 2, zenith angle 36.24°). The first was shortly after the main emission ended; the second was far outside the expected time window. Subsequent examination showed they had both triggered several tanks in the IceTop surface air shower array, and are thus very likely muons from cosmic ray air showers (Fig. 6.2). During the 40-string run, several candidate events were observed (Fig. 6.1), but these were of low significance and consistent with the expected atmospheric neutrino background. In Fig. 6.3 are shown limits from this analysis on the normalization of E^{-2} muon neutrino fluxes at Earth as a function of the size of the time window $|\Delta t|$, the difference between the neutrino arrival time and the first reported satellite trigger time. As a cross-check on both results, the limit from this analysis on the average individual burst spectra [23, 24] during the time window corresponding to the median duration of the bursts in the sample (28 seconds) was 0.24 times the predicted flux, within 10% of the model-dependent analysis conducted at the University of Maryland [46].

Assuming that the GRBs in our catalog are a representative sample of a total of 667 per year [27], we can scale the emission from our catalog to the emission of all GRBs. The resulting limits can then be compared to the expected neutrino rates from models that assume that GRBs are the main sources of ultra high energy

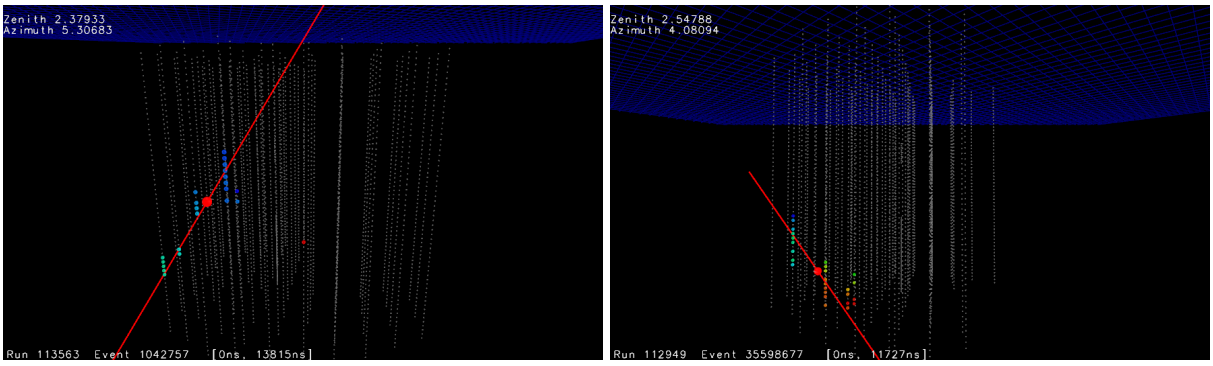


Figure 6.1 The two most significant events in the final IC40 data sample. The first occurred 2249 seconds before GRB 090417B, 1.95 degrees off-source, with a localization uncertainty of 1.61 degrees and an estimated energy of 1.3 TeV. The second occurred 3594 seconds before GRB 090219, 6.11 degrees off-source, with a localization uncertainty of 6.12 degrees and an estimated energy of 3.3 TeV. Both are much lower energy than would be expected for neutrinos from a GRB, and come substantially further in time from the burst than predicted by most models. Both are most likely atmospheric neutrinos, and are consistent with the expected background (P-value of ~ 0.4).

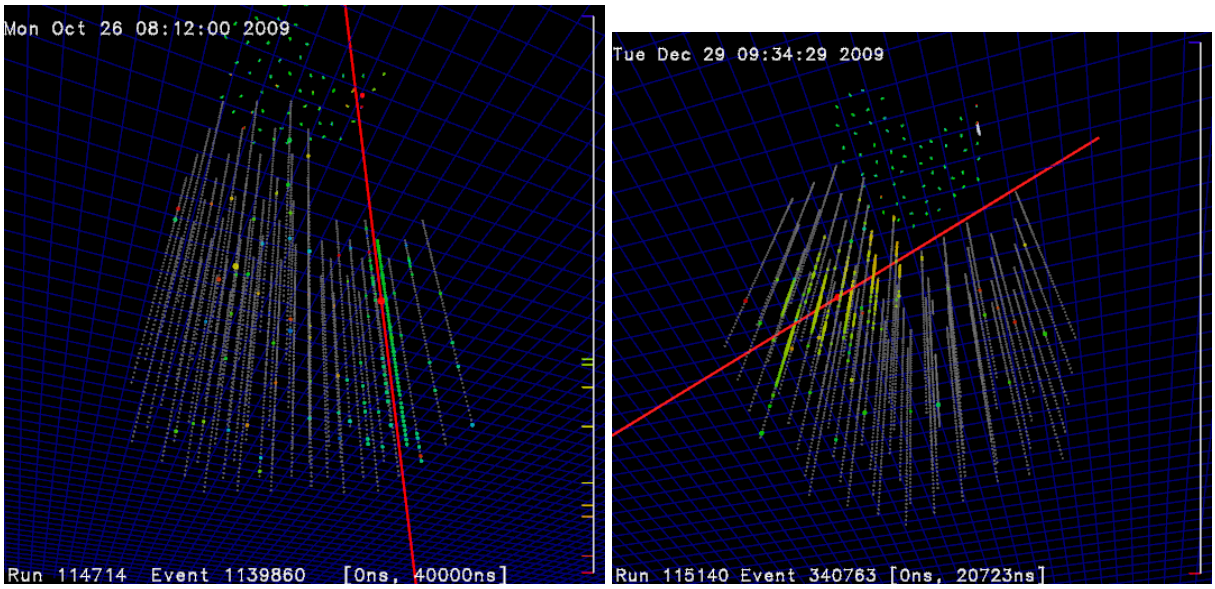


Figure 6.2 The two most significant events in the final IC59 data sample. The first occurred 30 seconds after GRB091026A, 4.5 degrees off-source, with a localization uncertainty of 10.5 degrees, with an estimated energy of 35 TeV. The second occurred 14 hours before GRB091230A, 0.2 degrees off-source, localized to 0.2 degrees, with an estimated energy of 109 TeV. The first is very clearly a cosmic ray instead of a neutrino, as it triggered the IceTop surface array. The second has one triggered IceTop tank at the correct time, strongly indicating that it is part of an air shower. It also occurs a long time from the GRB, where backgrounds are higher, and so has low significance. This type of event and the lowering of the IceTop energy threshold beginning with IceCube-79 will likely result in an integrated IceTop veto in future analyses. Both events are consistent with cosmic ray backgrounds (P-value of 0.21).

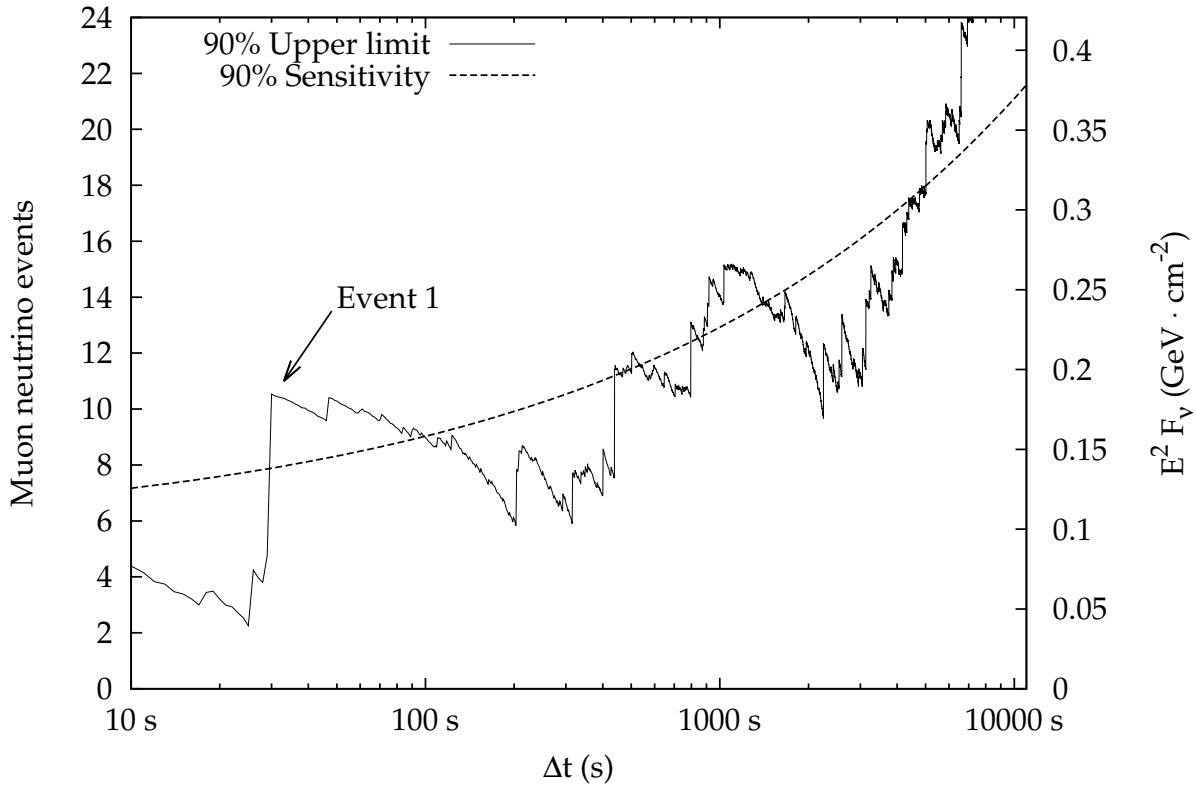


Figure 6.3 Limits on E^{-2} fluxes analysis as a function of the size of the time window $|\Delta t|$, calculated using the Feldman-Cousins method [48]. The left y-axis shows the total number of expected ν_μ events. The right axis shows the summed fluence from the bursts in our sample (F_ν). A time window of Δt implies observed events arriving between t seconds before the burst and t afterward. The variation of the upper limit with Δt reflects statistical fluctuations in the observed background rate, as well as the presence of individual events of varying quality. The event at 30 seconds (Event 1) is consistent with background and believed to be a cosmic-ray air shower.

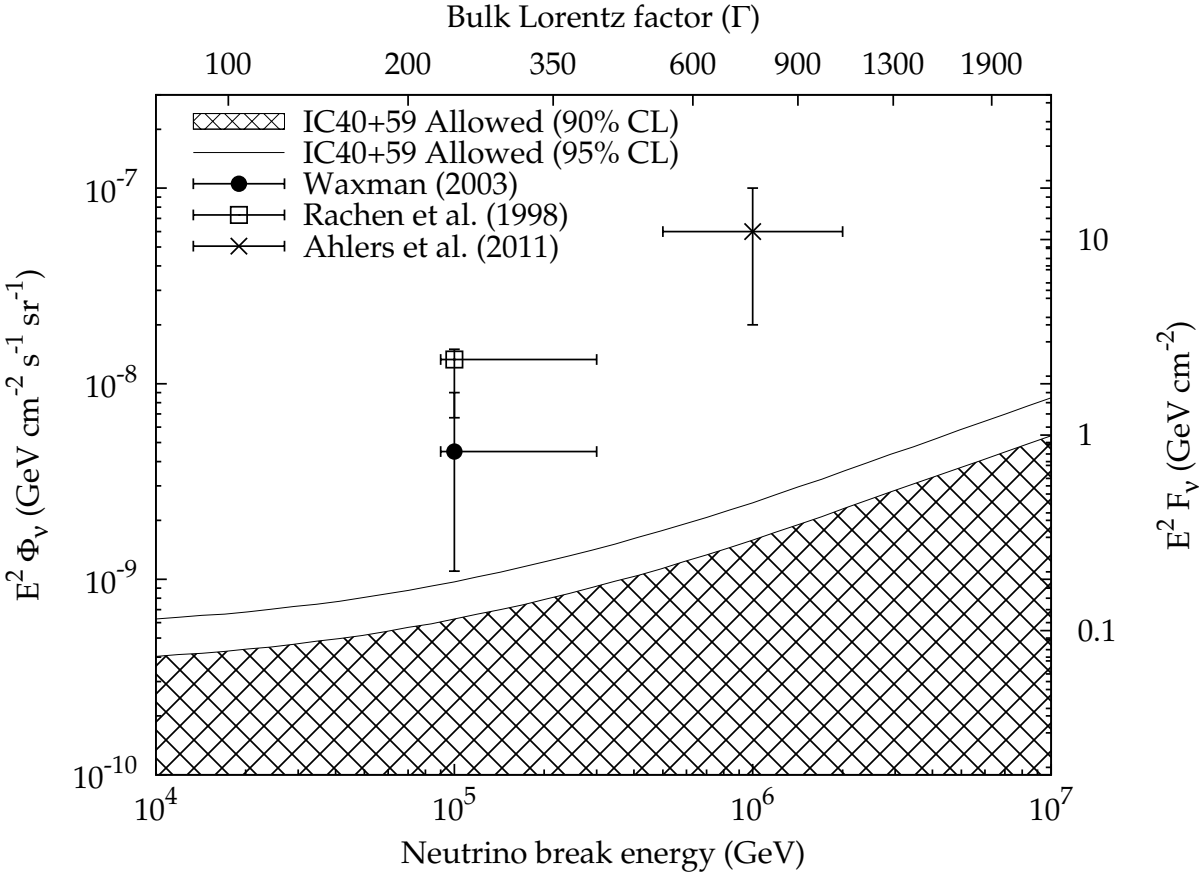


Figure 6.4 Limits from the model-independent analysis in comparison to theoretical predictions relating GRB neutrino fluxes to the cosmic ray flux. Data are taken from the time window corresponding to the median duration of the GRBs in our catalog ($|\Delta t| = 28$ seconds). Spectra are represented as broken power laws ($\phi_\nu \cdot \{E^{-1}/\epsilon_b, E < \epsilon_b; E^{-2}, E > \epsilon_b\}$). The neutrino break energy ϵ_b is related to the bulk Lorentz factor Γ ($\epsilon_b \propto \Gamma^2$). As plotted here, Γ is approximate and derived from the break energy (ϵ_b) assuming neutrino production at the Δ resonance from proton interaction with a 1 MeV photon field at a redshift of 1 [23, Eq. A7]. Vertical axes are related to the accelerated proton flux by the model-dependent constant of proportionality f_π. For models assuming a neutron-decay origin of cosmic rays (Rachen and Ahlers) f_π is independent of Γ; for others (Waxman-Bahcall) $f_\pi \propto \Gamma^{-4}$. Error bars on model predictions are approximate and were taken either from the original papers, where included [21], or from the best-available source in the literature [29] otherwise. The errors are due to uncertainties in f_π and in fits to the cosmic-ray spectrum. Waxman-Bahcall [22] and Rachen *et al.* [20] were calculated using a cosmic ray density of $1.5 - 3 \times 10^{44}$ erg Mpc^{-3} yr^{-1}, with 3×10^{44} the central value [26].

cosmic rays [22, 20, 21], with sampling biases of the same order as model uncertainties in the flux predictions [29, 28]. Limits from the model-independent analysis on fluxes of this type are shown in Fig. 6.4. The muon flux in IceCube is dominated by neutrinos with energies around the first break (ϵ_b). As such, the upper break, due to synchrotron losses of π^+, has been neglected in Fig. 6.4, as its presence or absence does not contribute significantly to the muon flux and thus does not have a significant effect on the presented limits.

These limits exclude all tested models [23, 22, 20, 21] with their standard parameters and uncertainties on those parameters (Fig. 6.4). The models are different formulations of the same fireball phenomenology, producing neutrinos at proton-photon ($p\gamma$) interactions in internal shocks. The remaining parameter spaces available to each therefore have similar characteristics: either a low density of high-energy protons, below that required to explain the cosmic rays, or a low efficiency of neutrino production.

In the fireball scenario, protons are accelerated stochastically in collisions of internal shocks in the expanding GRB. The neutrino flux is proportional to the rate of $p\gamma$ interactions, and so to the proton content of the burst by a model-dependent factor. Assuming a model-dependent proton ejection efficiency, the proton content can in turn be related to the measured flux of high-energy cosmic rays if GRBs are the cosmic ray sources. Limits on the neutrino flux for extragalactic cosmic ray normalized models are shown in Fig. 6.4; each model prediction has been normalized to a value consistent with the observed ultra high-energy cosmic ray flux.

An alternative is to reduce the neutrino production efficiency. For some models (Waxman-Bahcall-type, Sec. 2.4), the uncertainties in the neutrino efficiency are large. Recent papers [28, 30] have indicated that accounting for detailed microphysics in $p\gamma$ interactions may change the flux prediction by a factor of a few. This is still overwhelmed by the astrophysical uncertainties in these models and it is not clear at this point where the predictions will end up. The applicability of these models to the question of cosmic ray acceleration is also doubtful in the first place, since they do not treat the problems of proton acceleration or ejection; they appear to be best suited only for determination of the proton content of the outflow, with its unknown internal environment, rather than connections to the cosmic ray spectrum.

Another possibility, which applies to all models, is to increase the bulk Lorentz factor Γ of the outflows. Increasing Γ increases the proton energy threshold for pion production in the observer frame, thereby reducing the neutrino flux due to the lower proton density at higher energies. Astrophysical lower limits on Γ are established by pair production arguments [23], but the upper limit is less clear. Although it is possible that Γ may take values of up to 1000 in some unusual bursts, the average value is likely lower (usually assumed to be around 300 [23, 24]) and the non-thermal gamma-ray spectra from the bursts set a weak constraint that $\Gamma \lesssim 2000$ [18]. For all considered models, with uniform fixed proton content, very high average values of Γ are required to be compatible with our limits (Fig. 6.4).

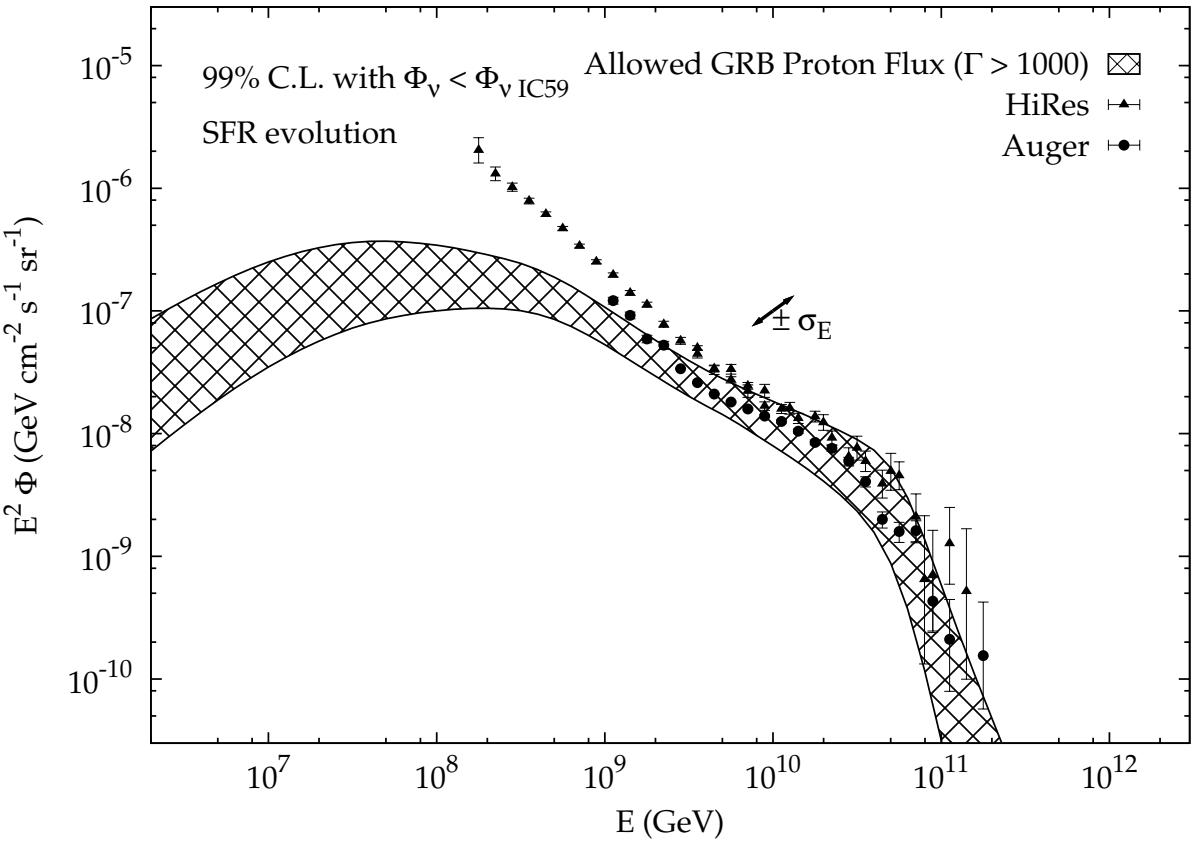


Figure 6.5 Proton fluxes from GRBs allowed by this work if GRBs are responsible for the highest energy tail of the cosmic ray spectrum and cosmic rays escape the fireball as neutrons [20, 21].

These results are strongest in the case of models where cosmic rays escape from the GRB fireball as neutrons [20, 21], which are also the ones that directly treat the problem of cosmic ray production in GRBs. In these models, the neutrons and neutrinos are created in the same $p\gamma$ interactions, directly relating the cosmic ray and neutrino fluxes and removing many uncertainties in the flux calculation. In these scenarios, Γ also sets the threshold energy for production of cosmic rays. The requirement that the extragalactic cosmic rays be produced in GRBs therefore does set a strong upper limit on Γ: increasing it beyond ~ 3000 causes the proton flux from GRBs to disagree with the measured cosmic ray flux above 4×10^{18} eV, where extragalactic cosmic rays are believed to be dominant. Limits on Γ in neutron-origin models from this analysis ($\gtrsim 2000$, Fig. 6.4) are very close to this point, and as a result all such models in which GRBs are responsible for the entire extragalactic cosmic-ray flux are now largely ruled out (Fig. 6.5).

Although the precise constraints are model dependent, the general conclusion is the same for all the versions of fireball phenomenology we have considered here: either the proton density in gamma ray burst fireballs is substantially below the level required to explain the highest energy cosmic rays or the physics in gamma ray burst shocks is significantly different from that included in current models. In either case, our current theories of cosmic ray and neutrino production in gamma ray bursts will have to be revisited.

LIST OF REFERENCES

[1] V. F. Hess. Über Beobachtungen der durchdringenden Strahlung bei sieben Freiballonfahrten. *Physikalische Zeitschrift*, 13:1084–1901, 1912.

[2] Kenneth Greisen. End to the cosmic ray spectrum? *Phys. Rev. Lett.*, 16:748–750, 1966. doi:10.1103/PhysRevLett.16.748.

[3] G. T. Zatsepin and V. A. Kuzmin. Upper limit of the spectrum of cosmic rays. *JETP Lett.*, 4:78–80, 1966.

[4] R. Abbasi et al. The IceCube data acquisition system: Signal capture, digitization, and timestamping. *Nuclear Instruments and Methods in Physics Research A*, 601:294–316, April 2009. arXiv:0810.4930, doi:10.1016/j.nima.2009.01.001.

[5] E. Waxman. Cosmological gamma-ray bursts and the highest energy cosmic rays. *Phys. Rev. Lett.*, 75:386–389, July 1995. doi:10.1103/PhysRevLett.75.386.

[6] M. Vietri. The Acceleration of Ultra–High-Energy Cosmic Rays in Gamma-Ray Bursts. *Astrophysical Journal*, 453:883, November 1995. arXiv:astro-ph/9506081, doi:10.1086/176448.

[7] T. J. Galama, P. M. Vreeswijk, J. van Paradijs, C. Kouveliotou, T. Augusteijn, H. Böhnhardt, J. P. Brewer, V. Doublier, J.-F. Gonzalez, B. Leibundgut, C. Lidman, O. R. Hainaut, F. Patat, J. Heise, J. in't Zand, K. Hurley, P. J. Groot, R. G. Strom, P. A. Mazzali, K. Iwamoto, K. Nomoto, H. Umeda, T. Nakamura, T. R. Young, T. Suzuki, T. Shigeyama, T. Koshut, M. Kippen, C. Robinson, P. de Wildt, R. A. M. J. Wijers, N. Tanvir, J. Greiner, E. Pian, E. Palazzi, F. Frontera, N. Masetti, L. Nicastro, M. Feroci, E. Costa, L. Piro, B. A. Peterson, C. Tinney, B. Boyle, R. Cannon, R. Stathakis, E. Sadler, M. C. Begam, and P. Ianna. An unusual supernova in the error box of the γ-ray burst of 25 April 1998. *Nature*, 395:670–672, October 1998. arXiv:astro-ph/9806175, doi:10.1038/27150.

[8] R. W. Klebesadel, I. B. Strong, and R. A. Olson. Observations of Gamma-Ray Bursts of Cosmic Origin. *Astrophysical Journal Letters*, 182:L85, June 1973. doi:10.1086/181225.

[9] D. E. Reichart. The Redshift of GRB 970508. *Astrophysical Journal Letters*, 495:L99, March 1998. arXiv:astro-ph/9712100, doi:10.1086/311222.

[10] A. I. MacFadyen and S. E. Woosley. Collapsars: Gamma-Ray Bursts and Explosions in "Failed Supernovae". *Astrophysical Journal*, 524:262–289, October 1999. arXiv:astro-ph/9810274, doi:10.1086/307790.

[11] D. Eichler, M. Livio, T. Piran, and D. N. Schramm. Nucleosynthesis, neutrino bursts and gamma-rays from coalescing neutron stars. *Nature*, 340:126–128, July 1989. doi:10.1038/340126a0.

[12] W. S. Paciesas, C. A. Meegan, G. N. Pendleton, M. S. Briggs, C. Kouveliotou, T. M. Koshut, J. P. Lestrade, M. L. McCollough, J. J. Brainerd, J. Hakkila, W. Henze, R. D. Preece, V. Connaughton, R. M. Kippen, R. S. Mallozzi, G. J. Fishman, G. A. Richardson, and M. Sahi. The Fourth BATSE Gamma-Ray Burst Catalog (Revised). *Astrophysical Journal Supplements*, 122:465–495, June 1999. arXiv:astro-ph/9903205, doi:10.1086/313224.

[13] R. Sari, T. Piran, and J. P. Halpern. Jets in Gamma-Ray Bursts. *Astrophysical Journal Letters*, 519:L17–L20, July 1999. arXiv:astro-ph/9903339, doi:10.1086/312109.

[14] E. S. Rykoff, F. Aharonian, C. W. Akerlof, M. C. B. Ashley, S. D. Barthelmy, H. A. Flewelling, N. Gehrels, E. Göğüş, T. Güver, Ü. Kiziloğlu, H. A. Krimm, T. A. McKay, M. Özel, A. Phillips, R. M. Quimby, G. Rowell, W. Rujopakarn, B. E. Schaefer, D. A. Smith, W. T. Vestrand, J. C. Wheeler, J. Wren, F. Yuan, and S. A. Yost. Looking Into the Fireball: ROTSE-III and Swift Observations of Early Gamma-ray Burst Afterglows. *Astrophysical Journal*, 702:489–505, September 2009. arXiv:0904.0261, doi:10.1088/0004-637X/702/1/489.

[15] GRB Coordinates Network. http://gcn.gsfc.nasa.gov.

[16] E. Fermi. On the origin of the cosmic radiation. *Physical Review*, 75:1169–1174, 1949.

[17] URL: http://lhc-machine-outreach.web.cern.ch/lhc-machine-outreach/lhc-machine-outreach-faq.htm.

[18] P. Mészáros. Gamma-ray bursts. *Reports on Progress in Physics*, 69:2259–2321, August 2006. arXiv:astro-ph/0605208, doi:10.1088/0034-4885/69/8/R01.

[19] A. M. Hillas. The origin of ultra-high-energy cosmic rays. *Annual Review of Astronomy and Astrophysics*, 22(1):425, 1984.

[20] J. P. Rachen and P. Mészáros. Cosmic rays and neutrinos from gamma-ray bursts. In C. A. Meegan, R. D. Preece, & T. M. Koshut, editor, *Gamma-Ray Bursts, 4th Hunstville Symposium*, volume 428 of *American Institute of Physics Conference Series*, pages 776–780, May 1998. arXiv:astro-ph/9811266, doi:10.1063/1.55402.

[21] M. Ahlers, M. C. Gonzalez-Garcia, and F. Halzen. GRBs on probation: Testing the UHE CR paradigm with IceCube. *Astroparticle Physics*, 35:87–94, September 2011. arXiv:1103.3421, doi:10.1016/j.astropartphys.2011.05.008.

[22] E. Waxman and J. Bahcall. High energy neutrinos from cosmological gamma-ray burst fireballs. *Phys. Rev. Lett.*, 78:2292–2295, March 1997. doi:10.1103/PhysRevLett.78.2292.

[23] D. Guetta, D. Hooper, J. Alvarez-Muñiz, F. Halzen, and E. Reuveni. Neutrinos from individual gamma-ray bursts in the BATSE catalog. *Astroparticle Physics*, 20:429–455, January 2004. doi:10.1016/S0927-6505(03)00211-1.

[24] R. Abbasi et al. Search for Muon Neutrinos from Gamma-ray Bursts with the IceCube Neutrino Telescope. *Astrophysical Journal*, 710:346–359, February 2010. doi:10.1088/0004-637X/710/1/346.

[25] J. K. Becker, M. Stamatikos, F. Halzen, and W. Rhode. Coincident GRB neutrino flux predictions: Implications for experimental UHE neutrino physics. *Astroparticle Physics*, 25:118–128, March 2006. arXiv:astro-ph/0511785, doi:10.1016/j.astropartphys.2005.12.006.

[26] E. Waxman. Astrophysical sources of high energy neutrinos. *Nucl. Phys. B Proc. Suppl.*, 118:353–362, April 2003. doi:10.1016/S0920-5632(03)01333-1.

[27] R. Abbasi et al. Limits on Neutrino Emission from Gamma-Ray Bursts with the 40 String IceCube Detector. *Phys. Rev. Lett.*, 106:141101, 2011. arXiv:1101.1448, doi:10.1103/PhysRevLett.106.141101.

[28] P. Baerwald, S. Hümmer, and W. Winter. Systematics in Aggregated Neutrino Fluxes and Flavor Ratios from Gamma-Ray Bursts. *ArXiv e-prints*, July 2011. arXiv:1107.5583.

[29] D. Guetta, M. Spada, and E. Waxman. On the Neutrino Flux from Gamma-Ray Bursts. *Astrophysical Journal*, 559:101–109, September 2001. doi:10.1086/322481.

[30] Svenja Hümmer, Philipp Baerwald, and Walter Winter. Neutrino Emission from Gamma-Ray Burst Fireballs, Revised. 2011. arXiv:1112.1076.

[31] Franck Bernard. *Caractérisation des performances d'un télescope sous-marin à neutrinos pour la détection du cascades contenues dans le cadre du project ANTARES*. PhD thesis, Université de la Méditerranée, 2000.

[32] R. Abbasi et al. Calibration and characterization of the IceCube photomultiplier tube. *Nuclear Instruments and Methods in Physics Research A*, 618:139–152, June 2010. arXiv:1002.2442, doi:10.1016/j.nima.2010.03.102.

[33] Brian Marx and Paul Eilers. Flexible smoothing with B-splines and penalties (with comments and rejoinder). *Statistical Science*, 11(2):89–121, 1996.

[34] Charles L. Lawson and Richard J. Hanson. *Solving least squares problems*. Prentice-Hall, Englewood Cliffs, NJ, 1974.

[35] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Constructive Approximation*, 13:57–98, 1997. doi:10.1007/BF02678430.

[36] M. Ackermann et al. Optical properties of deep glacial ice at the South Pole. *Journal of Geophysical Research (Atmospheres)*, 111:13203, July 2006. doi:10.1029/2005JD006687.

[37] R. Abbasi et al. Study of south pole ice transparency with IceCube flashers. *in prep*, 2012.

[38] N. E. Bramall, R. C. Bay, K. Woschnagg, R. A. Rohde, and P. B. Price. A deep high-resolution optical log of dust, ash, and stratigraphy in South Pole glacial ice. *Geophysical Research Letters*, 32:21815, November 2005. doi:10.1029/2005GL024236.

[39] P. B. Price. Kinetics of Conversion of Air Bubbles to Air Hydrate Crystals in Antarctic Ice. *Science*, 267:1802–1804, March 1995. arXiv:astro-ph/9501073, doi:10.1126/science.267.5205.1802.

[40] Johan Lundberg, Predrag Miočinović, Kurt Woschnagg, Thomas Burgess, Jenni Adams, Stefan Hundertmark, Paolo Desiati, and Peter Niessen. Light tracking through ice and water–scattering and absorption in heterogeneous media with photonics. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 581(3):619 – 631, 2007. arXiv:astro-ph/0702108v2, doi:10.1016/j.nima.2007.07.143.

[41] C. Amsler et al. Review of particle physics. *Physics Letters B*, 1, 2008.

[42] Dirk Pandel. Bestimmung von Wasser- und Detektorparametern und Rekonstruktion von Myonen bis 100 TeV mit dem Baikal-Neutrinoteleskop NT-72. Diploma thesis, Humboldt-Universität zu Berlin, 1996.

[43] Sean Grullon, David Boersma, Gary Hill, Kotoyo Hoshina, and K. Mase. In *Proc. of 30th ICRC, Merida, Mexico*, 2007.

[44] M. F. Morales, D. A. Williams, and T. De Young. An advanced analysis technique for transient searches in wide-field gamma-ray observatories. *Astroparticle Physics*, 20:485–497, January 2004. doi:10.1016/j.astropartphys.2003.08.003.

[45] T. Neunhöffer. *Astroparticle Physics*, 25:220–225, April 2006. doi:10.1016/j.astropartphys.2006.01.002.

[46] R. Abbasi, Y. Abdou, T. Abu-Zayyad, M. Ackermann, J. Adams, J. A. Aguilar, M. Ahlers, D. Altmann, K. Andeen, J. Auffenberg, and et al. An absence of neutrinos associated with cosmic-ray acceleration in γ-ray bursts. *Nature*, 484:351–354, April 2012. arXiv:1204.4219, doi:10.1038/nature11068.

[47] H. L. Lai, J. Huston, S. Kuhlmann, J. Morfin, F. Olness, J. F. Owens, J. Pumplin, and W. K. Tung. Global QCD analysis of parton structure of the nucleon: CTEQ5 parton distributions. *European Physical Journal C*, 12:375–392, February 2000. arXiv:hep-ph/9903282, doi:10.1007/s100529900196.

[48] G. J. Feldman and R. D. Cousins. Unified approach to the classical statistical analysis of small signals. *Physical Review D*, 57:3873–3889, April 1998. doi:10.1103/PhysRevD.57.3873.

[49] Photon Propagation Code. URL: http://www.icecube.wisc.edu/~dima/work/WISC/ppc/.

[50] Carl de Boor. *A Practical Guide to Splines*, volume 27 of *Applied Mathematical Sciences*. Springer-Verlag, 1978.

[51] Paul H. C. Eilers, Iain D. Currie, and María Durbán. Fast and compact smoothing on large multidimensional grids. *Computational Statistics and Data Analysis*, 50:61–76, 2006. URL: http://www.ma.hw.ac.uk/~iain/research/GLAM/CSDA_2006/CSDA.html.

[52] G. Beliakov. Shape preserving approximation using least squares splines. *Approximation Theory and its Applications*, 16(4):80–98, 2000.

[53] Luis F. Portugal, Joaquim J. Judice, and Luis N. Vicente. A comparison of block pivoting and interior-point algorithms for linear least squares problems with nonnegative variables. *Mathematics of Computation*, 63(208):625–643, 1994. URL: http://www.jstor.org/stable/2153286.

[54] Mikael Adlers. *Sparse Least Squares Problems with Box Constraints*. PhD thesis, Linköpings Universitet, S-581 83 Linköping, Sweden, 1998. URL: http://www.mai.liu.se/~milun/lic.ps.

[55] Timothy A. Davis and William W. Hager. Row modifications of a sparse cholesky factorization. *SIAM Journal on Matrix Analysis and Applications*, 26(3):621–639, 2005. doi:10.1137/S089547980343641X.

[56] Flexible image transport system: The astronomical image and table format. URL: http://fits.gsfc.nasa.gov/.

[57] PyFITS. URL: http://www.stsci.edu/resources/software_hardware/pyfits.

[58] CFITSIO - a FITS file subroutine library. URL: http://heasarc.gsfc.nasa.gov/fitsio/.

[59] Photospline. URL: http://code.icecube.wisc.edu/svn/projects/photospline.

[60] Radford M. Neal. Probabilistic inference using markov chain monte carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto, 1993. URL: http://www.cs.toronto.edu/~radford/review.abstract.html.

[61] Kyrre Strøm. On convolutions of B-splines. *Journal of Computational and Applied Mathematics*, 55(1):1 – 29, 1994. doi:10.1016/0377-0427(94)90182-1.

[62] Lyle Ramshaw. Blossoming: A connect-the-dots approach to splines. Technical Report 19, Digitial Equipment Corporation, 1987. URL: http://www.hpl.hp.com/techreports/Compaq-DEC/SRC-RR-19.pdf.

APPENDIX
Photospline

Nathan Whitehorn, Jakob van Santen

A.1 Introduction

Understanding the response of a detector to a physics process is the single most important aspect of analysis. For Icecube, this understanding involves first the energy loss processes of a particle in the ice, then the propagation of light to the photomultipliers, the response of the photomultipliers to light, and finally the behavior of the data acquisition. As Icecube operates using heterogeneous naturally occurring material in an energy range beyond the reach of accelerator experiments, both the propagation of particles and light in the detector are unlikely to conform to specification. Here we discuss technical aspects of Photospline, a set of software for representing our understanding of light propagation in the ice in an efficient, accurate, and numerically stable way.

We want to know the answers to two questions about light propagation: given a physics event in the detector, how many photons arrive at each photomultiplier, and when do they arrive there? Because the ice is heterogeneous and because the scattering and absorption lengths for photons in the detector are similar, there are no simple analytic answers to these questions. Far away from a light source, in the purely diffusive regime, the number of photons is approximately exponential with distance, and it is possible to neglect structure in the ice and reasonably approximate the time arrival distribution as a gamma distribution [42]. However, with the high timing resolution and heterogeneous ice of Icecube, these approaches do not deliver sufficient accuracy either for simulation or reconstruction.

Instead, we take recourse to Monte Carlo simulation of photon propagation. As of this writing, there are two approaches to this. The first is exemplified by the software PPC [49], which directly propagates photons through the detector for each physics event. This approach has a time complexity of $\mathcal{O}(n)$, which is proportional to the energy deposited in the detector. The alternative approach is currently employed by the software package Photonics [40], which pre-computes and tabulates the results of many types of light-producing events. Simulating physics events with Photonics then just involves a table lookup and generation of a Poisson random number, and so is $\mathcal{O}(1)$. Except for very small events, a table-based simulation is

then much faster than direct propagation[1]. This is especially important for reconstruction, where photon densities must be computed for each hypothesis considered and the actual Poisson mean is required, not just a sample from the distribution, which would require several orders of magnitude more propagation runs, making direct propagation impracticable.

The problem with a table-based approach, however, is that the tables can potentially be enormous and therefore difficult to work with, and that tabulation can introduce numerical artifacts. These numerical artifacts arise from averaging over a table cell during binning, from inaccurate interpolation between cells, and from statistical fluctuations as the bins are filled. Any attempt to reduce the size of the tables exacerbates the binning artifacts, and vice versa.

The technique described here is an attempt to remove ourselves from this trap. Photospline provides a post-processing technique on very large fine-binned photon tables that produces a much smaller, compressed format relatively insensitive to fluctuations in the original tables. The photon probabilities are represented as a multi-dimensional spline surface, which provides smooth and accurate interpolation while maintaining the speed of the table-based approach. For reconstruction, we also add to the essential Poisson mean of photons the ability to quickly and analytically compute derivatives and integrals of the distributions, which improves the speed and accuracy of maximization of likelihoods incorporating the tabulated photon counts.

The following describes the mathematical operations involved in fitting histograms of simulated photons as well as providing a simple user's guide to the software and a description of operations on the stored spline tables used in simulation and reconstruction.

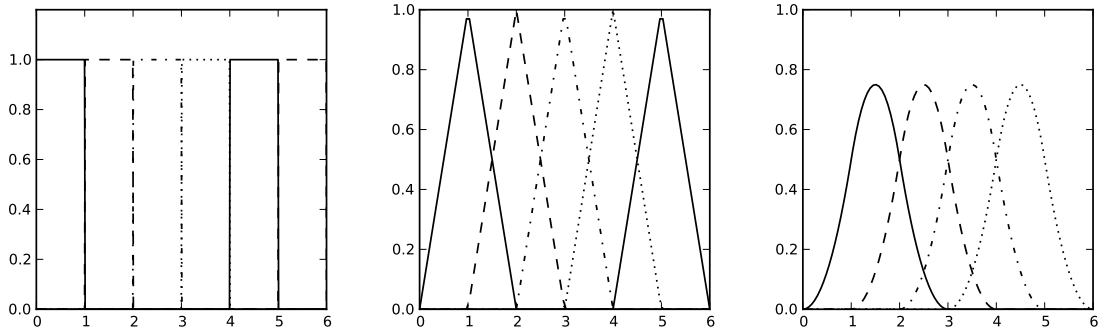
A.2 B-spline surfaces

An ideal interpolation function for the photonics tables would have several properties: it would be smooth, fast to evaluate, possible to fit deterministically in a small amount of time, extend easily to large numbers of dimensions, and have parameters that are easy to store, while minimizing introduction of bias and artifact into the fit tables. B-splines fit this description almost perfectly: a spline of order n has $n-1$ continuous derivatives, the functions can be evaluated quickly, and fitting them to a table is a linear problem, so it can be done quickly with standard techniques.

A.2.1 B-splines

The Basis Spline (B-spline) is a smooth function commonly used in interpolation problems. The simplest such function is the unit box, which is one from $\vec{k}_0$ to $\vec{k}_1$ and 0 otherwise. This is called a *B-spline of order 0*:

[1]For very small events, the large constant involved in table lookup means than direct propagation can be faster.



(a) Order-0 B-splines (b) Order-1 B-splines (c) Order-2 B-splines

Figure A.1 B-spline functions

$$B_0(x) = \begin{cases} 1 & : \quad \vec{k}_0 \leq x < \vec{k}_1 \\ 0 & : \quad \text{otherwise} \end{cases} \tag{A.1}$$

Higher orders of spline are formed by iterative self-convolution (Figure A.1), and gradually converge to a Gaussian as the order is increased. In general, the basis spline of order n is defined by a *knot vector* ($\vec{k}$) of $n+2$ points. In the region spanned by this vector, the function is non-zero, and is identically zero at all other points. This can be stably evaluated for orders greater than 0 by a recursive function known as the DeBoor algorithm[50, Page 131, Eq. 5]:

$$B_n^i(x) = \frac{(x - \vec{k}_i)B_{n-1}^i(x)}{\vec{k}_{i+n} - \vec{k}_i} + \frac{(\vec{k}_{i+n+1} - x)B_{n-1}^{i+1}(x)}{\vec{k}_{i+n+1} - \vec{k}_{i+1}} \tag{A.2}$$

It is often desirable to calculate not just the value of a specific B-spline $B_n^i(x)$, but the values of all $n+1$ non-zero B-splines at x. Applying Equation (A.2) directly is not the most efficient way to go about this, since the recurrence relations for neighboring splines involve many of the same terms, most of which will be recalculated unnecessarily. The calculation can be done much more efficiently by starting from the single nonzero 0th order B-spline at x and building up until one obtains the nth order splines as illustrated in Figure A.2 [50, Page 132, Subprogram BSPLVB]. For order 2 splines, this approach is already ~ 4 times faster than naïve application of Equation (A.2).

A.2.2 Spline Approximation

To create a smooth approximating function in one dimension, we take a linear combination of B-spline functions defined on a common knot vector $\vec{k}$, such that the nth B-spline of order m, B_m^n, is defined on the subset of $\vec{k}$ from $\vec{k}_n$ to $\vec{k}_{n+m+1}$.

0	$B^{n-1}_{i-n+2}(x)$	$B^{n-1}_{i-n+3}(x)$	.	.	$B^{n-1}_{i-1}(x)$	$B^{n-1}_{i}(x)$	0
.	.	.	.	.	.		
0	$B^{2}_{i-2}(x)$	$B^{2}_{i-1}(x)$	$B^{2}_{i}(x)$	0			
0	$B^{1}_{i-1}(x)$	$B^{1}_{i}(x)$	0				
0	$B^{0}_{i}(x)$	0					

Figure A.2 A visualization of a "bottom-up" version of the DeBoor recurrence relation for the nonzero nth order B-splines for $x \in [k_i, k_{i+1})$. Each row of the table can be generated from the one below it by application of (A.2). All the B-splines that do not appear in the table are identically equal to zero on $[k_i, k_{i+1})$, and so do not need to appear in the calculation.

Functions defined in this way have several attractive properties for us. Because they are linear combinations of smooth functions, they are likewise smooth, the amount of information required to represent the function is small, and they can be fit to data using standard linear techniques such as Singular Value, QR, or Cholesky decomposition. In addition, because all but $m+1$ of the basis functions are identically zero at any particular point (a property called *local support*), to evaluate the approximation function we need only evaluate the DeBoor algorithm $m+1$ times. This vastly improves computation time for lookups compared to functional representations like Fourier decompositions or polynomial interpolation without this property.

A.2.3 Tensor Product Surfaces

We can extend this approach easily into multiple dimensions by adopting the concept of the *tensor product spline*. Instead of a knot vector, the tensor product basis functions are defined on a rectangular n-dimensional knot grid $\overline{k}$ formed by taking the tensor product of n one-dimensional knot vectors. Each basis function is then a product of one-dimensional B-splines (Figure A.3, which are then taken in linear combination as in the one-dimensional case:

$$B(\vec{x}) = \overline{\alpha} \cdot \overline{B} = \overline{\alpha} \cdot \left(B^i_1(\vec{x}_1) \otimes B^j_2(\vec{x}_2) \otimes \dots \right) \tag{A.3}$$

Because most components of the basis tensor $\overline{B}$ are zero, as in the one-dimensional case, we recover the same local support property, albeit with more contributing basis functions due to the increased dimensionality of the problem.

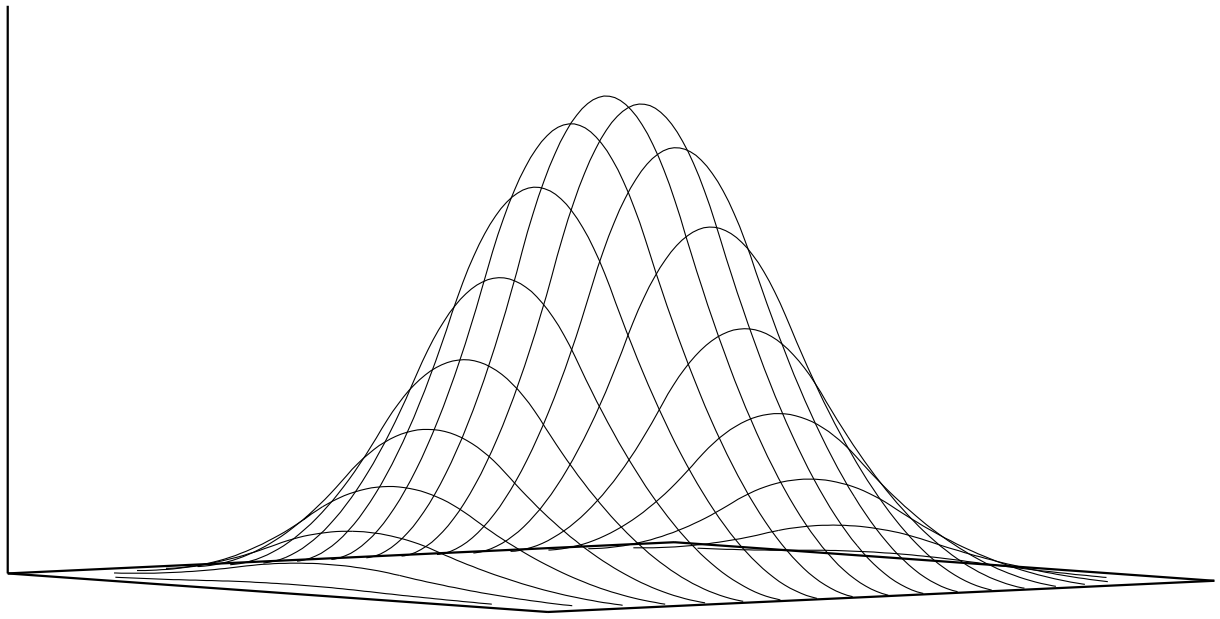


Figure A.3 A 2-dimensional tensor product spline constructed from two one dimensional basis splines

A.3 Determining B-spline coefficients

A.3.1 Ordinary Least Squares

In one dimension, fitting a spline function to data points is a simple linear procedure. The basic idea is to note that the value of the function at each data point is a linear combination of the values of each basis function at that point. In general, for a set of n basis functions, and n data points (the *exactly determined* case), we can write down a set of linear equations the solutions to which are the coefficients of an interpolating spline function:

$$\begin{aligned} \alpha_1 B_1(x_1) + \alpha_2 B_2(x_1) + \alpha_3 B_3(x_1) + \cdots + \alpha_n B_n(x_1) &= y_1 \\ \alpha_1 B_1(x_2) + \alpha_2 B_2(x_2) + \alpha_3 B_3(x_2) + \cdots + \alpha_n B_n(x_2) &= y_2 \\ &\vdots \\ \alpha_1 B_1(x_m) + \alpha_2 B_2(x_m) + \alpha_3 B_3(x_m) + \cdots + \alpha_n B_n(x_m) &= y_m \end{aligned} \tag{A.4}$$

This system of linear equations can then be written in matrix form:

$$\begin{pmatrix} B_1(x_1) & B_2(x_1) & \cdots & B_n(x_1) \\ B_1(x_2) & B_2(x_2) & \cdots & B_n(x_2) \\ \vdots & & \ddots & \vdots \\ B_1(x_m) & B_2(x_m) & \cdots & B_n(x_m) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \tag{A.5}$$

where $B_{ij} = B_i(x_j)$ is the so-called *model matrix*. In this form, the problem can be quickly solved by a variety of standard matrix factorization tools.

But what if the problem is not exactly determined? If, for example, there are more data points than coefficients, it is possible that no solution exists to Equation A.5. This is in fact the situation when using this procedure to compress data, as with the photon tables. In this case, what we would like is a way to compute the best approximation to an exact solution under the circumstances.

It is possible to construct a new problem the exact solution to which forms the global least-squares solution to the original overdetermined problem – as well as the maximum-likelihood solution under the assumption of normally distributed errors. We do this by formulating the *normal equations*. These are constructed by noticing that multiple data points are affected by changes in a single basis function. Thus, we can use a linear combination of data points as a proxy for their combined effect. In more mathematical language, we use the transpose of the model matrix B_{ij} to project the data points into the space of the coefficients. Instead of the exact linear equation

$$B\alpha = y \tag{A.6}$$

we multiply from the left by the transpose of the model matrix $B^\intercal$ to obtain the normal equation:

$$B^\intercal B\alpha = B^\intercal y \tag{A.7}$$

The solution of this equation is then the vector α such that $||B\alpha - y||$ is minimized. This can be seen clearly by writing the gradient of the least squares residual of $B\alpha - y$:

$$\begin{aligned}
\vec{\nabla}_\alpha ||B\alpha - y|| &= \nabla_\alpha \left(B\alpha - y\right)^2 \\
&= \nabla_\alpha \left((B\alpha - y)^\intercal (B\alpha - y)\right) \\
&= \nabla_\alpha \left((\alpha^\intercal B^\intercal - y^\intercal)(B\alpha - y)\right) \\
&= \nabla_\alpha \left(\alpha^\intercal B^\intercal B\alpha - \alpha^\intercal B^\intercal y - y^\intercal B\alpha\right) + \cancel{\nabla_\alpha y^\intercal y} \\
&= 2B^\intercal B\alpha - B^\intercal y - (y^\intercal B)^\intercal \\
&= 2B^\intercal B\alpha - 2B^\intercal y \\
&= 2\left(B^\intercal B\alpha - B^\intercal y\right) \\
\therefore B^\intercal B\alpha - B^\intercal y = 0 &\Leftrightarrow \min ||B\alpha - y||
\end{aligned} \tag{A.8}$$

From this, α can easily be obtained using the same procedures we used to solve Equation A.6, after adding suggestive parentheses:

$$\left(B^\intercal B\right)\alpha = \left(B^\intercal y\right) \tag{A.9}$$

In particular, if B has full column rank (that is, the columns are linearly independent, as is the case with a B-spline basis), then $B^\intercal B$ is symmetric and positive-definite (that is, all of its eigenvalues are positive). We can factorize such matrices into a product of a lower-triangular matrix L and its transpose using the *Cholesky decomposition*:

$$B^\intercal B = LL^\intercal \tag{A.10}$$

The triangular form is useful, as we use it to solve linear equations of the form $B^\intercal B\alpha = B^\intercal y$ by direct substitution. First, we re-group the terms

$$B^\intercal B\alpha = LL^\intercal\alpha = L(L^\intercal\alpha) = B^\intercal y \tag{A.11}$$

before solving for the value of $L^\intercal\alpha$ by forward substitution from the lower-triangular matrix L:

$$L(L^\intercal\alpha) = Lz = B^\intercal y \tag{A.12}$$

Given this intermediate solution, we can solve for α by backward substitution from the upper-triangular matrix $L^\intercal$:

$$L^\intercal \alpha = z \tag{A.13}$$

A.3.2 Weighted Least Squares

It may be the case that some of the data points being fit are less important than others – for instance, they may have larger errors – and so we would like the fit curve to care less about fitting these points exactly if it can do better elsewhere at their expense. We incorporate this condition into the least-squares formalism (Equation A.9) by incorporating a *weight* term.

In ordinary least squares, we minimize $||BA - y||$, which is to say the sum $\sum_i (f(x_i) - y_i)^2$, where $f(x) = \sum_j A_j B_j(x)$. The weight condition is then incorporated by instead minimizing

$$\sum_i W_i (f(x_i) - y_i)^2 \tag{A.14}$$

Terms with $W_i = 0$ then do not contribute to the fit, while terms with large W_i dominate the objective function (A.14) and will be fit very well, if necessary at the expense of the rest of the data set. If the weights W_i are chosen to be $1/\sigma$, where σ is the measured variance of the data points, then minimizing the objective function (A.14) becomes a minimization of χ^2:

$$\chi^2 = \sum_i \frac{(f(x_i) - y_i)^2}{\sigma} \tag{A.15}$$

We can integrate this into the linear-algebraic formulation of least-squares (Equation A.9) by defining a diagonal weight matrix W_{ii}.[2] This matrix is then incorporated into the normal equations (Equation A.9):

$$(B^\intercal W B)\, \alpha = (B^\intercal W y) \tag{A.16}$$

We can see that it does in fact minimize the objective function (A.14) by rearranging it slightly and applying the logic of (A.8):

$$B^\intercal (W (B\alpha - y)) = 0 \tag{A.17}$$

A.3.3 Penalized Least Squares

If there is some statistical jitter in the data points to be fit, we may want the fit curve to have some intrinsic rigidity and follow the general trend instead of following the fluctuations. The way this is accomplished is called *regularization*.

[2] This approach can be expanded to handle cross-correlations by extending W_{ii} with off-diagonal terms. In the case of normally-distributed errors, χ^2 is minimized by defining $W_{ij} = \sigma_{ij}^{-1}$, where σ_{ij} is the variance-covariance matrix.

For fitting photonics tables, we employ a standard technique called Tikonoff regularization, in which a penalty is attached to large values of the second derivative. A very noisy fit will be continuously changing curvature, whereas a smooth one will have a relatively constant curvature, and so the second derivative will in general be small. Thus the smooth fit will be preferred by the fitter.

This is accomplished in the context of least squares fitting by adding a term to the normal equations. Instead of solving Equation A.9, we solve the penalized normal equation:

$$(B^\intercal B + \lambda P^\intercal P)\,\alpha = (B^\intercal y) \tag{A.18}$$

The solution to this equation now minimizes $||(B\alpha - y)|| + \lambda ||P\alpha||$. λ is then the strength of the regularization (called the smoothing parameter in the Photospline code). For $\lambda = 0$ we recover the unpenalized normal equations (A.9). For $\lambda \to \infty$, the regularization term dominates, and we end up with a curve of the shape specified by the regularization term, which is a form of Bayesian prior (for order-2 Tikonoff regularization, this will be a straight line).

A.3.3.1 Penalized Splines

For the specific case of regularized fits of B-splines, we employ the P-spline (penalized spline) technique [33]. To employ the regularization technique outlined above, we must write a matrix P such that it will transform the vector of B-spline coefficients A into a vector whose terms are related to the second derivatives of the spline function.

B-splines have a property that makes this simple. The j-th derivative of a spline function can be written as a new spline function of order $k - j$ using the B-spline derivative recurrence relation [50, Page 139, Eq. 12]:

$$D^j \left(\sum_i \alpha_i B_{i,k} \right) = \sum_i \alpha_i^{j+1} B_{i,k-j} \tag{A.19}$$

where

$$\alpha_r^{j+1} = \begin{cases} \alpha_r & , \quad j = 0 \\ \frac{\alpha^j_{r,k} - \alpha^j_{r-1,k}}{(t_{r+k-j} - t_r)/(k-j)} & , \quad j > 0 \end{cases} \tag{A.20}$$

We can now rewrite Equation A.20 in matrix form to acquire our penalty matrix, which will transform the vector α into α^{j+1}. Since this is recursive, it is difficult to write explicitly in the general case, but some examples are provided for the case of equispaced knots, 1 unit apart:

$$\begin{aligned}
\text{Order 0}: &\quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
\text{Order 1}: &\quad \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix} \\
\text{Order 2}: &\quad \begin{pmatrix} 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \end{pmatrix}
\end{aligned} \tag{A.21}$$

A.3.3.2 Penalized Least Squares in Multiple Dimensions

In principle, the extension of least squares fitting to multiple dimensions is easy. As before, we have a set of (n-dimensional) basis functions B, to be fit at a set of n-dimensional positions to a measured set of data y. With some smoothness constraint on each dimension, we then want to minimize

$$||B\alpha - y|| + \lambda_1||P_1\alpha|| + \lambda_2||P_2\alpha|| + \cdots + \lambda_n||P_n\alpha|| \tag{A.22}$$

In analogy to Equation A.18, we can write down the corresponding system of linear equations:

$$\left(B^\intercal B + \sum_i^n \lambda_i P_i^\intercal P_i\right)\alpha = B^\intercal y \tag{A.23}$$

The difficulty here arises in that B can be enormous in the multi-dimensional case. For 25 knots on each of 4 axes, and a table with 200 million cells (typical numbers for photon tables), B will have dimensions 390625 by 200 million and require 568 TB of RAM just to store, which, as of 2010, is substantially larger than the memory capacity of any computer on Earth.

We get around this problem by noticing two important facts. The first is that B is quite sparse. Because of the local support property of B-splines, most of the basis functions have no support at most of the data points. As such, we can eliminate all of the zeroes from the matrix using sparse-matrix routines. This reduces memory consumption in the above problem by a factor of 100.

The second thing is that B is not present in isolation in Equation A.23, but instead only occurs as the combination $B^\intercal B$. Whereas B has dimensions of the number of basis functions by the number of data points, $B^\intercal B$ is a square matrix of side length the number of basis functions. For the example problem above, being able to work directly with $B^\intercal B$ instead of B would reduce memory consumption by another factor of 500.

This also means, critically, that neither the memory requirements nor the CPU requirements of this fitting algorithm would depend on the number of data points, but instead only on the number of basis functions.

The algorithm that allows us to do this is called the Generalized Linear Array Model (GLAM) technique [51], and exploits the fact that our basis functions are tensor product splines and that our data are located on a grid (though this requirement can be relaxed by weighting non-observed points required to maintain the grid structure with weights of 0). The algorithm itself is complicated, and more clearly illustrated by source code than equations. As such, readers interested in the details of this algorithm are referred to the Python GLAM code included in Photospline as well as to the original paper [51].

A.3.4 Table Stacking and Pseudointerpolation

Each run of the photon simulation we are fitting we produce light distributions for a single source configuration. For a table-based simulation, many source configurations are simulated and the distributions for an arbitrary source produced from interpolation between the tables for similar sources. For spline tables, we wish to be able to do this as well. The first solution to come to mind, to do a global fit, is impracticable. In addition, because there is no bin averaging between sources, smoothing (and thus least-squares fitting) is unnecessary, and we can stick to interpolation algorithms.

What we do then is to "stack" the individual three- or four-dimensional fit tables $(r, \theta, \phi, [t])$ for each simulated source into a single five- or six-dimensional table $(r, \theta, \phi, [t], \theta_{\mathrm{src}}, z_{\mathrm{src}})$, still of tensor product splines. The normalization of the spline basis functions is such that we can simply stack the fit coefficients of each sub-table while creating a knot vector such that the original values of the sub-tables form control points for the interpolating splines in the extra dimensions.

Given a knot vector $\vec{k}$, the maxima of the B-splines defined on that vector can be found by solving Equation A.20. Given equally spaced splines, these also are the centers of each basis function and can be found at:

$$\vec{B}_n^{max} = \vec{k} - \frac{n-1}{2} \tag{A.24}$$

By inverting Equation A.24, we can acquire a formula for a knot vector given a vector of centers $\vec{x}$:

$$\vec{k} = \vec{x} + \frac{n-1}{2} \tag{A.25}$$

We must also add points to the knot vector, since a spline function of order n made of m basis functions (and so m coefficients) must be defined by $m + n + 1$ knots. Thus we need $n + 1$ extra knots, of which the first n are placed at linearly extrapolated positions before the positions from Equation A.25, and one placed at a linearly extrapolated position after the last element.

Iterative application of Equation A.25 with stacking of coefficients will then produce a new tensor product spline surface with the sub-table locations as control points. It is worth noting that this surface does *not* exactly interpolate the original tables, as B-splines do not interpolate their control points. Instead, it forms a curve that maximally fills the convex hull of the set of control points (Figure A.4). In general, this is an excellent approximation to interpolation, but sharp extrema will be systematically underestimated by a small amount.

A.4 Enforcing shape constraints

While the B-spline surface determined by the least-squares condition is by definition the closest fit to the data, it does not necessarily share some desirable properties with the data. In particular, when the data can't be exactly represented as a B-spline surface, the fit surface may "ring," that is, make erratic sweeps around the data (see Figure A.5). This can cause the fit surface to become negative in a region where the data are strictly positive or non-monotonic where the data are strictly monotonic. The latter case is critical for our application, where we fit for the cumulative photon arrival time distribution, which must by definition be strictly monotonic. It is possible, however, to express the monotonicity constraint along a single dimension (time) without stepping too far away from the elegant simplicity of the least-squares problem.

We have implemented the method suggested by [52], in which we transform one axis of the tensor-product B-spline basis to a cumulative T-spline ("Trapezoidal" spline, see Figure A.6(b)) basis in which each basis function is the sum of the B-spline basis functions following it . Unlike the B-splines from which they are derived, the T-spline basis function have highly non-local support; e.g. the first T-spline basis function has support over the entire knot field. This makes evaluating the spline surface arduous; in the extreme case of a point in the support of the last T-spline basis function, evaluating the spline surface involves all the preceding coefficients. Luckily, we can use the T-spline basis for fitting only, transforming the T-spline coefficients into the corresponding B-spline coefficients before storing them.

The T-spline basis, however, makes it particularly easy to enforce monotonicity in one dimension. In this basis, the spline surface is monotonic if all the T-spline coefficients are positive. Determining the best set of T-spline coefficients is now a least squares problem with a simple constraint; instead of a simple least-squares problem, we have a constrained optimization problem, the so-called non-negative least squares problem. While not quite as simple as the unconstrained problem, there exist algorithms to solve the constrained problem as a series of unconstrained problems in a finite number of steps. We have implemented two of these algorithms, the Portugal/Judice/Vincente (PJV) block-pivoting algorithm [53] and Adlers BLOCK3 [54], using sparse matrix operations.

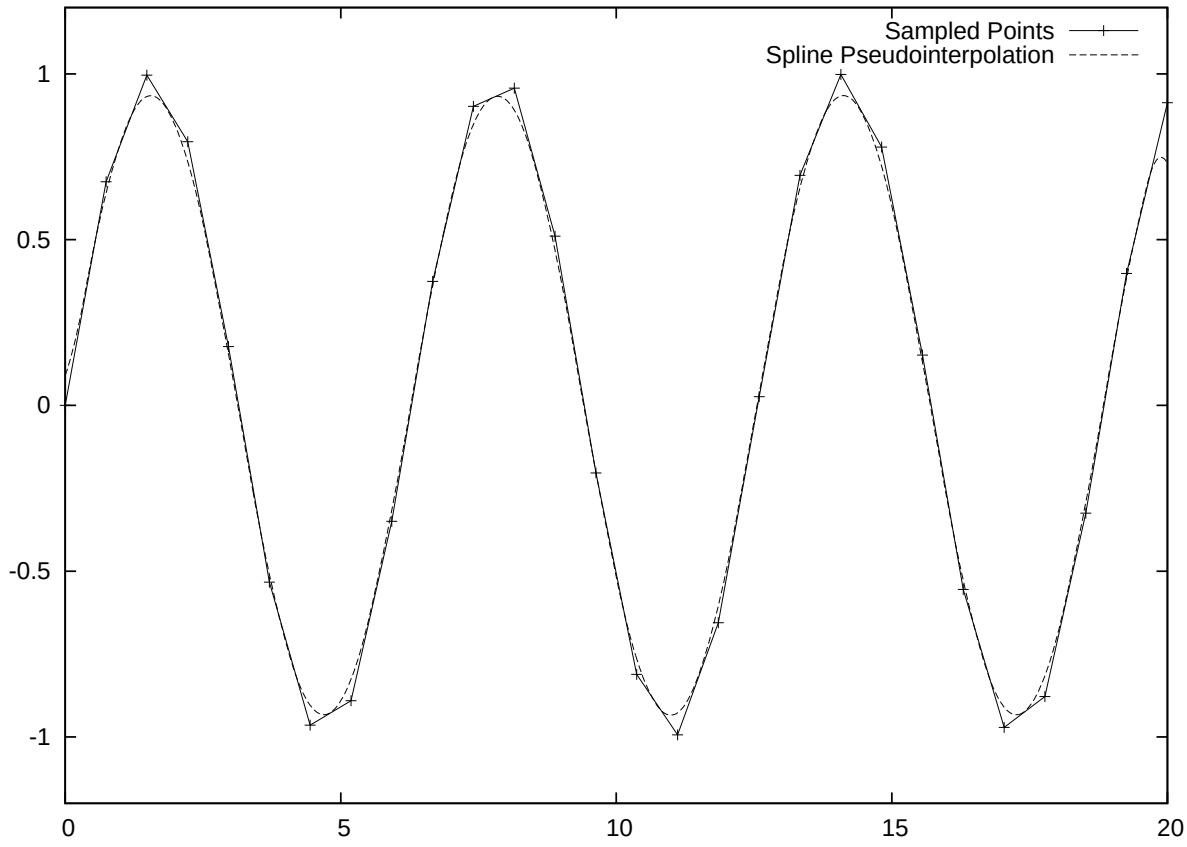


Figure A.4 Pseudointerpolation of a set of points sampled from $\sin(x)$ using the same algorithm as for table stacking. Since B-splines do not interpolate their control points, but instead maximally fill the convex hull formed by those points, the interpolation algorithm will fall slightly short of extrema, while generally maintaining accuracy.

A.4.1 Solving the non-negative least squares problem

Given $A \in \mathbb{R}^{M \times N}$ and $b \in \mathbb{R}^N$, $x \in \mathbb{R}^M$ is the solution to the non-negative least squares problem:

$$\min_x ||Ax - b||_2^2 : x \geq 0 \tag{A.26}$$

The Karush-Kuhn-Tucker optimality conditions lead to the following linear complementarity problem [53]:

$$y = A^T Ax - A^T b : y \geq 0,\ x \geq 0,\ x^T y = 0 \tag{A.27}$$

An *active set* approach is commonly used to find the solution to (A.27). The set of coefficients x is partitioned into an free set $\mathcal{F}$ and a constrained set $\mathcal{G}$[3]. The coefficients in the free set are the solution to the unconstrained least squares problem in the subspace of the free set, while those in the constrained set are clamped to zero. The problem then reduces to quickly and efficiently finding the partition of the coefficients that satisfies (A.27). The solution can be found iteratively, but unlike the minimization of a general function, it is possible to construct algorithms that are provably finite.

A.4.2 Algorithm NNLS

The first of these algorithms is NNLS by Lawson and Hanson [34], which we will sketch here to illustrate the general features of non-negative least squares solvers. The algorithm consists of two nested loops, an outer loop that updates and tests y for optimality, and an inner loop that finds a strictly positive x. At the beginning of the algorithm, $x \equiv 0$ and $y = -A^T b$. In each iteration of the outer loop, one finds the most negative entry in y. If all the entries in y are positive, then the KKT conditions are satisfied and the algorithm terminates. If not, the index of the most negative entry is freed from the constrained set, whereupon the algorithm enters the inner loop. In the inner loop, one solves for the unconstrained least squares problem in the subspace of the new free set, yielding a trial solution x_t at the global minimum in the subspace. If all the entries in x_t are positive, it is accepted as a new solution and used to update y in the next iteration of the outer loop.

If any elements of x_t are negative, however, it is not necessarily a better solution to the constrained problem than the current solution x. Simply projecting the global minimum into the feasible space [4], however, could result in *increasing* the magnitude of the residual (see Figure 5.3 in [54]), which could cause the algorithm to cycle. To avoid this, one scales the *descent vector* connecting x and x_t such that exactly one coefficient becomes 0:

[3] The constrained set is somewhat counterintuitively called the active (for *actively constrained*) set, while the free set is called the passive set.

[4] By "projecting," we simply mean setting any negative coefficients in x_t to zero.

$$\max_{\alpha} : x + \alpha(x_t - x) \geq 0,\ \alpha \in [0, 1] \tag{A.28}$$

Since the residual norm $||Ax_t - b||^2$ is the global minimum and $\alpha \in [0, 1]$, $||A(x + \alpha(x_t - x)) - b||^2 \leq ||Ax - b||^2$, the inner loop either decreases the residual norm or reduces the dimensionality of $\mathcal{F}$, and is thus finite. Similarly, the outer loop either increases the dimensionality of $\mathcal{F}$ or terminates, and thus the entire algorithm is finite.

A.4.3 Block-pivoting algorithms for sparse problems

NNLS is a *single-pivoting* algorithm, that is, it changes the free set by exactly one element every time it solves the unconstrained problem. The computational effort needed to find the constrained solution can be reduced significantly by employing *block pivoting* methods, that is, freeing or constraining multiple coefficients between solves [53]. We implemented both the PJV block-pivoting algorithm [53] and Adlers' BLOCK3 [54] using the CHOLMOD sparse matrix package [55]. While very fast in some cases, the PJV algorithm occasionally exhibits cyclic behavior. The reasons why this happens are illustrative; we will briefly sketch the PJV algorithm before describing BLOCK3, which avoids this pathology.

A.4.3.1 Algorithm PJV

The structure of PJV is slightly different from that of NNLS. It is initialized with $x \equiv 0$ and $y = -A^T b$ just like NNLS, but has only a single loop. At the beginning of the loop, one recalculates y based on the current free set, then searches for negative entries in x and y. These *infeasible coefficients* are freed if they are in y or constrained if they are in x. One then solves the unconstrained problem in the subspace of the free space before returning to the beginning of the loop. If all of the coefficients are feasible, the KKT conditions are satisfied and the algorithm terminates.

Usually the number of infeasible coefficients is reduced in every iteration, but this does not necessarily happen in every case. This makes it necessary to introduce a fallback mechanism: in each iteration, one records the number of infeasible coefficients if it is smaller than the number of infeasibles from the previous iteration. If the number of infeasibles hasn't been reduced in the last n iterations, one reverts to a single-pivoting strategy, freeing or constraining the last infeasible coefficient, until the number of infeasibles has been reduced to below the previous minimum.

While PJV can be quite fast when it manages to reduce the number of infeasibles in every iteration, unavoidable numerical errors can cause coefficients that are nearly zero to jump between the free and constrained sets in a cyclic fashion. When this happens, even the fallback procedure can cycle. In addition the properties of the intermediate solutions are poorly defined. Since the algorithm always jumps directly to the

subspace minimum orthogonally projected into the feasible space, it does not necessarily monotonically reduce the magnitude of the residual. If one simply terminates the algorithm after a certain maximum number of iterations, one may still obtain an unacceptably poor solution. For this reason, we chose to also investigate BLOCK3, which shares the monotonicity properties of NNLS while using a more efficient pivoting strategy.

A.4.3.2 Algorithm BLOCK3

BLOCK3 [54] is mostly a block-switching variant of NNLS, so its structure is quite similar[5]. In the outer loop, one frees all infeasible coefficients in y; the algorithm terminates if they are all positive. In the inner loop, one finds a trial solution to the unconstrained problem. If the trial solution is entirely positive, one returns to the outer loop.

If some coefficients are negative, though, one has to scale the descent vector to find an acceptable solution. Instead of choosing the smallest α as in NNLS, one attempts to violate as many constraints as possible while still reducing the residual. To do this, one constructs a list of α corresponding to each negative coefficient in the trial solution compares the resulting residual norm to the residual norm for the current solution in descending order until an α is found that reduces the residual norm. Since the residual norm is given by

$$||Ax - b||^2 = x^T A^T Ax - 2x^T A^T b + b^T b = x^T (A^T Ax - 2A^T b) + k \tag{A.29}$$

where k is an additive constant, the complexity of the calculation is dominated by an $\mathcal{O}(nnz)$ sparse matrix-vector multiplication, which is cheap compared to a full factorization of $A^T A$. In cases where $\alpha = 1$ reduces the residual norm, the calculation is practically free. Once a feasible x has been found, one returns to the outer loop.

Since the inner loop of BLOCK3 either reduces the dimensionality of the free set or reduces the residual norm, it is finite and incapable of cyclic behavior. Just like NNLS, BLOCK3 reduces the residual norm monotonically, and thus produces better and better intermediate solutions. This makes it possible to terminate the algorithm if it has not converged after a fixed number of iterations and still obtain an acceptable solution.

A.5 Fitting procedure and storage format

While the methods discussed above have been implemented for some special cases, we have written what we believe to be the first completely general implementation. Our software can fit an n-dimensional grid of data to a tensor-product B-spline surface of arbitrary order in each dimension. In addition, we can add

[5] Instead of starting with $x \equiv 0$, however, we initialize the solver loop with the solution to the unconstrained problem. While computing the solution in the full coefficient space is more expensive than in the subspace of a typical free set, starting with a solution that is similar to the expected final solution reduces the number of iterations significantly [54].

a regularization term to the normal equations composed of an arbitrary linear combination of penalties, where each term penalizes changes in the kth derivative of the spline surface along a certain dimension. If monotonicity is required along a single dimension, we can reformulate the problem such that monotonicity is expressed by a linear constraint on the spline coefficients. In this section, we will give an overview of the fitting procedure.

The user interface for our fitting software is a set of Python functions and classes that can be used to write scripts to fit a specific kind of data[6]. Some of these functions are in pure Python, while others are implemented in C. All of the computationally-intensive components exist in two forms: a reference implementation in Python, and a heavily-optimized implementation in C. The reference implementations are intended to be readable rather than fast, and are the easiest way to understand the what the algorithms are actually doing. The C implementations, on the other hand, are intended for production and make heavy use of multi-threaded operations on sparse matrices. Since they expose the same API, however, the Python implementation can be used to check the C implementation for correctness, which can be extraordinarily useful.

Since the interface is in Python, the data can be loaded and manipulated in arbitrarily complex ways before being passed to the fitter. Each fitting script must provide the following parameters, which completely specify the fitting problem:

- An n-dimensional array Z, giving the grid of data points.
- An n-dimensional array W, giving the weight to assign to each data point.
- A list of vectors $\{\vec{x}\}$ giving the positions of the data points along each of the n axes.
- A list of vectors $\{\vec{k}\}$ giving the positions of the B-spline knots along each of the n axes.
- The order of spline to use in each dimension.
- The smoothing parameter λ to use in each dimension.
- The order of penalty to use in each dimension.
- Optionally, the dimension in which monotonicity should be enforced.

The fitter uses $\{\vec{x}\}$ and $\{\vec{k}\}$ to calculate a tensor-product B-spline basis for the problem and formulate the penalized normal equations (Equation (A.23)) using the method of [51]. In the non-monotonic case,

[6]For example, different choices of knots are appropriate for PHOTONICS tables in cylindrical and spherical coordinates.

the normal equations can be solved directly via the Cholesky decomposition, returning the best-fit set of B-spline coefficients.

When monotonicity is to be enforced, however, a few modifications are needed. First, the B-spline basis along the monotonic axis is converted to a T-spline basis, and the normal equations are formulated with a hybrid T-/B-spline tensor-product basis. These normal equations are then passed to a non-negative least-squares solver. As of 2010, there were no efficient and robust implementations of non-negative least squares that operated directly on sparse normal equations, so we implemented our own sparse versions of PJV and BLOCK3 for this purpose[7]. Once the solver has converged, the coefficients are converted back to coefficients on a tensor-product B-spline basis.

The results of the fit are returned to Python in a data structure that specifies all the information needed to evaluate the spline surface:

- The knot vectors in each dimension.
- The order of spline in each dimension.
- The coefficient array.
- Auxiliary information, such as the coordinate system used in the table.

This information is then written to disk. Since the coefficient array is essentially an n-dimensional image, we store it in a FITS image file [56]. This format can be read and written from Python using PyFITS[8] or from C using cfitsio [58]. Within the FITS file, the coefficient array is stored in the primary Header-Data Unit (HDU), while each of the knot vectors is stored in an auxiliary HDU.

A.6 Photospline fitting guide

After having tested several different schemes for fitting PHOTONICS tables, we have settled on the general strategy of the fitting the total amplitude and normalized timing distributions separately. Because of the enormous dynamic range present in the tabulated values ($\mathcal{O}(10^{20})$ in the total amplitudes alone), it is much easier to fit the logarithm of the amplitude in each bin rather than the amplitude itself. Each three-dimensional amplitude table is fit in log space with order-2 splines. The cumulative timing distributions, on the other hand, are constrained between 0 and 1. These we fit in linear space, enforcing monotonicity along the time axis. We use order-2 splines in the spatial dimensions and order-3 splines along the time axis in order to have smooth time derivatives for sampling (see Section A.7). In typical usage the amplitude fit is

[7] Due to occasional cyclic behavior in PJV, we use BLOCK3 exclusively.

[8] PyFITS [57] is a product of the Space Telescope Science Institute, which is operated by AURA for NASA.

very fast, finishing in a few seconds, while the timing fit may take anywhere from tens of minutes to several hours, depending on the number of knots used.

We have implemented support for these sorts of tables in PHOTONICSSERVICE. Sampling from the cumulative timing distribution is discussed in Section A.7, querying amplitude information for reconstruction purposes in Section A.8. In the remainder of the section, we will describe the fitting scripts distributed with PHOTOSPLINE and the API of the fitting code.

A.6.1 Scripts

PHOTOSPLINE is distributed [59] with a few scripts for fitting spline surfaces to PHOTONICS tables and viewing the results of the fit. The fitting scripts implement schemes of knot placement, weighting, and penalty terms that we have found useful in producing accurate fits with a necessary minimum of computation time and memory footprint. The scripts can be found under photospline/private/fitter.

A.6.1.1 glam-photonics.py

This script is intended for fitting EM shower tables in cylindrical geometry[9]. The knots are space logarithmically in ρ and t in order to be able to represent rapid variations at small distances and early times while wasting few resources representing the diffusive region far from the source and at large delay times. In l, one-third of the knots are space quadratically from $-l_{\max}$ to 0, while the remainder are space logarithmically from 0 to $l_{\max}$. This helps to better represent rapid variations in amplitudes and timing distributions in the region directly in front of the source, where photons may arrive with relatively little scattering. The knots in ϕ are spaced uniformly.

A.6.1.2 glam-photonics-spherical.py

This script is similar to glam-photonics.py, except that it is intended for use with tables in spherical geometry[10]. The knots are spaced quadratically in r and logarithmically in t, while the knots in ϕ and $\cos\theta$ are spaced uniformly.

[9] In this geometry the table is binned in cylinder radius $\rho \in [0, \rho_{\max}]$, azimuth $\phi \in [0, 180]$, axial displacement $l \in [-l_{\max}, +l_{\max}]$, and delay time $t \in [0, t_{\max}]$. l gives the displacement from the light source projected along the direction the light source is pointing and ρ the displacement projected into a plane perpendicular to $\vec{l}$. ϕ is measured in degrees with respect to a vector perpendicular l, but pointing towards positive z in the IceCube coordinate system. The delay time is the difference between the actual photon arrival time and the direct propagation time $\frac{rn_g}{c}$

[10] In this geometry the table is binned in distance from the source $r \in [0, r_{\max}]$, azimuth $\phi \in [0, 180]$, cosine of polar angle $\cos\theta \in [-1, 1]$, and delay time $t \in [0, t_{\max}]$. ϕ and t have the same definition as in cylindrical geometry; θ is the opening angle between $\vec{r}$ and the direction the source is pointing. This coordinate system is more natural for point-like light sources, as amplitudes and timing distributions tend to vary more smoothly with respect to r and $\cos\theta$ than ρ or l.

Figure A.8 shows the resource consumption of glam-photonics-spherical.py running over a large set of spherically-binned EM shower tables. The timing fit (which dominates the run-time) was done with 15 "core" knots in r, 6 in ϕ, 15 in $\cos\theta$, and 25 in t.

A.6.1.3 stack-photonics.py

Each PHOTONICS table represents the light yield from a source at a particular depth in the ice, originating from a particular zenith angle; the spline surfaces fit to these tables represent the same information. In order to create a spline surface that represents the light yield from a source with an arbitrary orientation anywhere in the ice, we have to "stack" neighboring spline surfaces and interpolate between them. This script implements such a stacking procedure.

As a concrete example, consider stacking spline tables representing different source depths $\{z\}$[11]. First, an extra dimension is added to the coefficient array of each spline table and the coefficient arrays are stacked along the last dimension. Then, a knot vector is constructed for z such that each of the splines is centered on the depth corresponding to one of the individual fit tables. The resulting spline surface blends together the coefficients corresponding to different source depths using the values of the B-splines along z as weights[12]. Since it averages over neighboring depths[13], the interpolating spline in z has a tendency to smooth out sharp variations with respect to z in amplitudes and timing distributions, even if those sharp variations are the best fit to the corresponding PHOTONICS table.

A.6.1.4 glam-photoverify.py

This script iterates over 1- or 2-dimensional slices of a PHOTONICS table, plotting the tabulated values and the results of a spline fit. This can be useful for quickly checking the quality of a fit. It requires `gnuplot` and PyGnuplot.

A.6.2 Python API

The fitting API is contained entirely in a Python module called `glam`, which is implemented in Python with Numpy (dense) matrix operations. The corresponding C version, `spglam`, uses sparse matrix operations and is many orders of magnitude faster, but exposes the same API.

[11] The same procedure applies to stacking tables at the same depth, but with a different orientation. If given tables from multiple depths and orientations, stack-photonics.py will first stack the tables in zenith angle, then in depth

[12] This is similar to the way splines are commonly used in computer-aided design; in computer-graphics parlance, the coefficients for each depth are the "control points" of the spline in z.

[13] Note that this trick implicitly relies on all of the splines in z having the same normalization, and thus only works with equispaced depths.

A.6.2.1 `glam.fit`

Fit data to a tensor-product B-spline surface.

Signature:

```
fit(z, w, coords, knots, order, smooth, periods=None, penalties={2:None},
    monodim=None)
```

Parameters:

- `z`: An n-dimensional Numpy array containing the data points to be fit.
- `w`: An array of weights for each data point. This must have the same shape as `z`. If you don't want to weight data points differently, use uniform weights, e.g. `numpy.ones(z.shape)`.
- `coords`: A list of arrays, each giving the positions of the data points along the corresponding dimension.
- `knots`: A list of arrays, each giving the positions of the knots in the corresponding dimension.
- `order`: A list of integers, each giving the order of B-spline to use in the corresponding dimension.
- `smooth`: The global smoothing parameter λ. This will be used only if no dimension-specific smoothing parameters are specified in `penalties`.
- `periods`: A list of floats, each giving the periodicity of the corresponding dimension. If the periodicity is nonzero, `glam.fit` will construct a periodic B-spline basis in that dimension where the ends are identical by construction. This functionality is only implemented in `glam`; `spglam` will silently ignore the periodicity of the basis.
- `penalties`: A dictionary giving the penalty term to apply along each dimension. Each key-value pair specifies an order of penalty and the coefficient of that penalty order in each dimension. For example, to apply second-order penalties in the first three dimensions and a third-order penalty in the fourth (all with $\lambda = 1$), you would pass `{2:[1,1,1,0], 3:[0,0,0,1]}`. If the coefficient list is `None` (as it is by default), the λ specified by `smooth` will be used for all dimensions.
- `monodim`: If set to a non-negative integer, the spline surface will be forced to be monotonic along the corresponding dimension. Note that this involves solving a non-negative least squares problem, and is thus significantly slower than an unconstrained fit.

A.6.2.2 `glam.grideval`

Evaluate a spline surface on a coordinate grid.

Signature:

```
grideval(table, coords, bases=None)
```

Parameters:

- `table`: A `SplineTable` instance (as produced by `glam.fit` or read from disk with `splinefitstable.read`) containing the coefficients and knot vectors of the spline to be evaluated.
- `coords`: A list of vectors, each giving the positions in the corresponding dimension at which to evaluate the spline surface. For example, evaluating a three-dimensional spline at `[[0,1], [1,2], [3,4,5]]` will yield a $2 \times 2 \times 3$ grid of evaluates.
- `bases`: A list of pre-calculated spline bases in each dimension (i.e. with each element being the matrix in Equation (A.5)) with which to evaluate the spline. This can be used to evaluate the derivative of the spline surface along a given dimension by replacing the basis of kth-order B-splines with a basis of the derivatives of kth-order B-splines along the given dimension. This option is not offered in `spglam`.

A.7 Sampling from a B-spline surface

For use in simulation, we need to have an ability to sample photons from functions defined by B-splines in order to sample random photon arrival times. The simplest way to sample from a general distribution is to note that, since by definition first $x\%$ of the distribution are found in the region where the cumulative distribution function (CDF) has a value of $x\%$, the distribution's cumulative distribution function (CDF) is a mapping from the domain of the distribution to a uniform distribution on $(0, 1)$. As such, evaluating the inverse of the CDF at a point drawn uniformly from $(0, 1)$ will give a random variate in the domain of the distribution.

For spline-based functions, this is very difficult. The inverse of the function depends on all the basis functions, making evaluation taxing, and is also difficult to compute. A more promising avenue lies in alternative numerical techniques.

For Photospline, we employ an independence-chain Metropolis-Hastings sampler [60]. This is a Markov Chain-based algorithm that uses numerical values of the probability density function in conjunction with an approximation called the proposal distribution to sample from the spline-based target distribution. Over infinite time, the algorithm will return samples from the target distribution, but the efficiency of the algorithm

and bias in the first samples are both improved as the proposal distribution becomes closer and closer to the true distribution.

The Metropolis-Hastings algorithm works using the acceptance/rejection method. A point $x^{'}$ is sampled from the proposal distribution Q, along with a number α drawn from a uniform distribution on $(0, 1)$. If the condition

$$\alpha < \min\left\{\frac{P(x')Q(x)}{P(x)Q(x')}, 1\right\} \tag{A.30}$$

is satisfied (where x is the previous accepted point, and P the numerical probability density from which we wish to sample), then the sample is accepted and added to list of sampled points. Otherwise, it is rejected, and we repeat the algorithm as often as necessary to accept the next sampled point. For the case where the proposal distribution is the same as the target distribution ($P \equiv Q$), this reduces to $\alpha \in (0, 1) < 1$, which is always true, so all samples are accepted, and the efficiency of the algorithm is 1. When P and Q overlap very little, the likelihood ratio in Equation A.30 will be small, and most samples will be rejected, making the algorithm very inefficient (the number of sampled values is much less than the number of iterations of the algorithm).

As the tables are stored as cumulative distribution functions, and not probability density functions, we must recover the PDF in order to apply the Metropolis-Hastings algorithm. Using the DeBoor recursion relation (Equation A.2) and the multiplication rule, we can easily construct a new set of basis functions that are the derivatives of B-splines:

$$B_n^{i'}(x) = \frac{(x - \vec{k}_i)B_{n-1}^{i'}(x) + B_{n-1}^{i}(x)}{\vec{k}_{i+n} - \vec{k}_i} + \frac{(\vec{k}_{i+n+1} - x)B_{n-1}^{i+1'}(x) - B_{n-1}^{i+1}(x)}{\vec{k}_{i+n+1} - \vec{k}_{i+1}} \tag{A.31}$$

We can unpack one of these recursions in order to write the derivatives as simple linear combinations of two lower-order B-splines:

$$B_n^{i'}(x) = \frac{n}{\vec{k}_{i+n} - \vec{k}_i} B_{n-1}^{i}(x) - \frac{n}{\vec{k}_{i+n+1} - \vec{k}_{i+1}} B_{n-1}^{i+1}(x) \tag{A.32}$$

Because differentiation and addition commute, we can simply interchange these differential spline basis functions on the axis of interest with the original ones, while keeping all the other axes and the coefficient table unchanged.

Because we have quite an accurate approximation to the photon time arrival PDF in the Pandel distribution [42], we employ a version of the Pandel function as the proposal distribution, with the variance slightly widened to improve sampling accuracy in the tails. This provides a quite high accuracy sampling algorithm (Figure A.9) with an 80% acceptance rate and an autocorrelated burn-in period of 0.3 steps.

A.8 Reconstruction with splines

The problem of reconstruction is one of statistical inference. We detect Cerenkov photons at various positions and times in the ice, and wish to extract from this measurement the characteristics of the underlying event (say, the direction of a muon or the energy of an EM shower). The PHOTONICS tables (and their spline parameterizations) provide an approximation to the mean light yield at a given source-receiver orientation and delay time. We can add a model of the statistical fluctuations around the mean to construct a model of the detector response to a given event. For example, we may assume that photons propagate independently through the ice, and that the actual light yield is Poisson-distributed around the mean. For a given event we can invert this formulation to obtain a "likelihood" function that describes the probability with which a given set of event parameters produces the observed data. We can then numerically maximize this function[14] with respect to the event parameters to obtain the most likely event hypothesis.

The pitfalls of this approach are twofold. First, the likelihood function has a characteristic width around its maximum that determines the range of hypotheses compatible with the data, that is, the degree to which the proposed model and the data together constrain the solution to the inference problem. As one adds more constraints (data), this range becomes narrower, which is principle a good thing, as the reconstruction becomes more accurate. However, each new constraint is an extra term in the likelihood function that must be evaluated in each step of the minimizer, slowing the entire process down considerably. Thus, one must carefully choose data to go into the likelihood that are likely to provide the strongest constraints with a necessary minimum of computational effort. Second, if one only has access to the value of the likelihood function and not its functional form, the minimizer may have to spend a significant amount of time in each step probing around the parameter space to numerically estimate the local gradient. If the likelihood function has sudden jumps due to interpolation artifacts in the tabulated light yield, these can also confuse a numerical minimizer.

Our spline surfaces have some attractive features that may ameliorate these problems. First, since we fit the cumulative delay-time distribution $P(t)$ rather than the differential dP/dt, it is possible to bin the pulses extracted from a recorded waveform with arbitrary finesse and still have an accurate estimate of the mean charge expected in each bin (the charge quantile λ_i):

$$q_i = \sum_j q_j : t_i \leq t_j < t_{i+1} \tag{A.33}$$

$$\lambda_i = \int_{t_i}^{t_{i+1}} \frac{dP}{dt} dt = P(t_{i+1}) - P(t_i) \tag{A.34}$$

[14] Equivalently, multiply by -1 and minimize.

This makes it possible to tune the resolution and computational cost of the reconstruction to a desired level. In addition, the minimization may be run in multiple iterations, refining the binning (and making the peaks and valleys of the likelihood function sharper) between each iteration. This is equivalent to simulated annealing (see [60, Section 6]), and may be useful for avoiding local minima.

Second, if the ith dimension of the B-spline basis is constructed from splines of order k, then it has $k-1$ continuous derivatives along that dimension. These can be obtaining analytically by replacing the basis of B-splines along the given axis with a basis of B-spline derivatives, which can be obtained by the DeBoor algorithm (Equation (A.32)). This makes it possible to obtain the local gradient of the light yield exactly in a fixed number of operations. Since evaluating each element of the gradient involves multiplying the same coefficient matrix into a different B-spline basis, the entire gradient can be evaluated at relatively little computational cost by reformulating each scalar-scalar coefficient/basis-element multiplication in the B-spline surface evaluation as a scalar-vector multiplication[15]. In addition, the derivatives of the B-splines can be evaluated at almost no additional cost by pausing the bottom-up calculation of the B-spline values (see Figure A.2) at order $n-1$ and forming the B-spline derivatives (A.32) before proceeding to order n.

To convert the gradient of the B-spline surface to a gradient in the likelihood function, we can apply the Chain Rule. For example, to convert from a gradient in spherical Photonics coordinates to a gradient in detector coordinates, we apply the following transformation:

$$\begin{pmatrix} \frac{\partial\lambda}{\partial x_s} \\ \frac{\partial\lambda}{\partial y_s} \\ \frac{\partial\lambda}{\partial z_s} \\ \frac{\partial\lambda}{\partial t_s} \\ \frac{\partial\lambda}{\partial \theta_s} \\ \frac{\partial\lambda}{\partial \phi_s} \end{pmatrix} = \begin{pmatrix} \frac{\partial r}{\partial x_s} & \frac{\partial \phi}{\partial x_s} & \frac{\partial \cos\theta}{\partial x_s} & \frac{\partial t_d}{\partial x_s} & \frac{\partial \theta_{\mathrm{ph}}}{\partial x_s} & \frac{\partial z_{\mathrm{ph}}}{\partial x_s} \\ \frac{\partial r}{\partial y_s} & \frac{\partial \phi}{\partial y_s} & \frac{\partial \cos\theta}{\partial y_s} & \frac{\partial t_d}{\partial y_s} & \frac{\partial \theta_{\mathrm{ph}}}{\partial y_s} & \frac{\partial z_{\mathrm{ph}}}{\partial y_s} \\ \frac{\partial r}{\partial z_s} & \frac{\partial \phi}{\partial z_s} & \frac{\partial \cos\theta}{\partial z_s} & \frac{\partial t_d}{\partial z_s} & \frac{\partial \theta_{\mathrm{ph}}}{\partial z_s} & \frac{\partial z_{\mathrm{ph}}}{\partial z_s} \\ \frac{\partial r}{\partial t_s} & \frac{\partial \phi}{\partial t_s} & \frac{\partial \cos\theta}{\partial t_s} & \frac{\partial t_d}{\partial t_s} & \frac{\partial \theta_{\mathrm{ph}}}{\partial t_s} & \frac{\partial z_{\mathrm{ph}}}{\partial t_s} \\ \frac{\partial r}{\partial \theta_s} & \frac{\partial \phi}{\partial \theta_s} & \frac{\partial \cos\theta}{\partial \theta_s} & \frac{\partial t_d}{\partial \theta_s} & \frac{\partial \theta_{\mathrm{ph}}}{\partial \theta_s} & \frac{\partial z_{\mathrm{ph}}}{\partial \theta_s} \\ \frac{\partial r}{\partial \phi_s} & \frac{\partial \phi}{\partial \phi_s} & \frac{\partial \cos\theta}{\partial \phi_s} & \frac{\partial t_d}{\partial \phi_s} & \frac{\partial \theta_{\mathrm{ph}}}{\partial \phi_s} & \frac{\partial z_{\mathrm{ph}}}{\partial \phi_s} \end{pmatrix} \begin{pmatrix} \frac{\partial\lambda}{\partial r} \\ \frac{\partial\lambda}{\partial \phi} \\ \frac{\partial\lambda}{\partial \cos\theta} \\ \frac{\partial\lambda}{\partial t_d} \\ \frac{\partial\lambda}{\partial \theta_{\mathrm{ph}}} \\ \frac{\partial\lambda}{\partial z_{\mathrm{ph}}} \end{pmatrix} \tag{A.35}$$

This functionality is implemented in PhotonicsService for both total amplitudes and time-delay quantiles.

A.8.1 Analytic convolution of spline surfaces

While scattering processes account for the vast majority of the time-delay distribution in detected photons, it can be important to account for detector effects as well when reconstructing events. In particular, the photon arrival time extracted from the waveform recorded by an IceCube DOM is not actually the true photon arrival time, but rather the arrival time of a current pulse at the PMT anode, corrected for the average time it takes the pulse to propagate through the dynode stages. There is some variance in this transit

[15] Most modern CPUs are equipped with vector units, and can perform scalar/scalar and scalar/4-element-vector multiplications at nearly the same speed.

time, though, leading to "PMT jitter" or "transit-time spread." Also, while EM and hadronic showers are approximately point-like on the scale of the inter-string spacing in IceCube, they do have some spatial extent. Both of these effects can be approximately accounted for by convolving the photon delay-time distribution with a Gaussian. If we want to reconstruct with a convoluted spline-fit delay-time distribution, we have three options:

- Apply the convolution to the PHOTONICS tables before fitting, doubling the already considerable computational effort required.
- Numerically convolve the time-delay PDF evaluated from the spline every time it is requested, slowing down every spline evaluation.
- Analytically convolve the spline surface with a Gaussian, deriving the coefficients without fitting.

Luckily, it is possible to approximate the third option.

On a uniform knot field, a B-spline basis function of order n is proportional to the n-fold self-convolution of a uniform distribution over two adjacent knots. By the central limit theorem, the nth order uniform B-spline becomes proportional to a normal distribution as $n \to \infty$. B-splines of relatively low order can be used to approximate Gaussian convolution kernels, albeit with limited support. Furthermore, it is possible to analytically convolve a spline surface with a B-spline kernel and expand the result in a new B-spline basis . These are useful properties, as they allow one to apply a pseudo-Gaussian convolution to a B-spline surface merely be manipulating its coefficients.

The procedure for analytically convolving splines is described in [61] in a slightly different form than we require for our application and using terminology not necessarily familiar to anyone outside the field of computer graphics. We will review our adaptation of it here for the sake of clarity. We will use $B_{i,q}$ to denote the ith B-spline of degree (order +1) q, normalized such that they form a partition of unity:

$$\sum_i B_{i,q}(x) = 1 \tag{A.36}$$

These are the usual B-spline basis functions, defined on an arbitrary field of simple knots $\{t_i\}$. $M_{j,k}$ will denote the jth B-spline of degree k, but with each normalized to unity:

$$\int_{-\infty}^{+\infty} M_{j,k}(x)dx = 1 \tag{A.37}$$

These splines, defined on knots $\{\tau_j\}$, will be used as convolution kernels.

The convolution of $B_{i,q}$ and $M_{j,k}$ can be written[16] in terms of dummy variables x and y as

[16]This is a variation on, Equation (13) of [61].

$$(B_{i,q} \star M_{j,k})(z) = (t_{i+q} - t_i)\frac{q!(k-1)!}{(q+k-1)!}[t_i, \ldots, t_{i+q}]_x[t_j, \ldots, t_{j+k}]_y(x+y-z)_+^{k+q-1} \tag{A.38}$$

where $[t_i, t_{i+1}, \ldots, t_{i+k}]_x$ is the divided-difference operator with respect to x and $(\cdot - z)_+^n$ is the truncated power function

$$(a-b)_+^n = \begin{cases} (a-b)^n & b < a \\ 0 & b \geq a \end{cases} \tag{A.39}$$

Furthermore, $B_{i,q} \star M_{j,k}$ is a B-spline surface of degree $k+q$ on the combined knot field $\{t_i, \ldots, t_{i+q}\} \uplus \{\tau_j, \ldots, \tau_{j+k}\}$[17]. Given this fact and the ability to evaluate $(B_{i,q} \star M_{j,k})(z)$ for arbitrary z, we can expand the convolution in its B-spline coefficients.

A B-spline surface of degree $k+1$ defined on a knot field $\{\rho\}$ is a linear combination of the B-splines on $\{\rho\}$:

$$f(x) = \sum_i a_i B_{i,k+1,\rho} \tag{A.40}$$

The coefficients of the B-spline expansion are given by

$$a_i = \mathcal{B}(f(\xi_i))(\rho_{i+1}, \ldots, \rho_{i+k}) \tag{A.41}$$

where $\xi_i \in [\rho_i, \rho_{i+k+1})$ and $\mathcal{B}$ is the *multilinear blossom* of f [62]. The multilinear blossom is a function that is linear in each of its k arguments and is identical to f when evaluated on its diagonal, that is, when all its arguments are identical. The blossom of $B_{i,q} \star M_{j,k}$ is given by

$$\begin{aligned}\mathcal{B}((B_{i,q} \star M_{j,k})(z))(\rho_{i+1}, \ldots, \rho_{i+k+q-1}) = (t_{i+q} - t_i)\frac{q!(k-1)!}{(q+k-1)!}[t_i, \ldots, t_{i+q}]_x[t_j, \ldots, t_{j+k}]_y \\ \times\Theta(x+y-z)\prod_{l=i+1}^{i+k+q-1}(x+y-\rho_l)\end{aligned} \tag{A.42}$$

where Θ is the Heaviside step function. By (A.41), the coefficients a_l of the B-spline expansion of degree $k+q$ on the knot field ρ are given in terms of the coefficients of the un-convolved spline surface b_i by

$$a_l = \sum_{i,j} b_i \mathcal{B}((B_{i,q} \star M_{j,k})(x_l))(\rho_{l+1}, \ldots, \rho_{l+k+q-1}) : x_l \in [\rho_l, \rho_{l+k+q}) \tag{A.43}$$

[17] The combined knot field is formed by $q+1$ copies of $\{\tau_j, \ldots, \tau_{j+k}\}$, each centered on an element of $\{t_i, \ldots, t_{i+q}\}$. For example, $\{0,1,2\} \uplus \{-0.1, 0, 0.1\} = \{-0.1, 0, 0.1, 0.9, 1, 1.1, 1.9, 2, 2.1\}$. For a proof that $B_{i,q} \star M_{j,k}$ is contained in the spline space of degree $k+q$, see Theorem 8 of [61].

The blossom of the convolution can be used to construct a matrix that transforms the coefficients on the original knot field to coefficients on the convolved knot field. This is particularly useful for tensor-product spline surfaces: then, the convolution can be carried out along a given dimension by calculating the transformation matrix once and then applying it to each slice of the coefficient grid along that dimension.

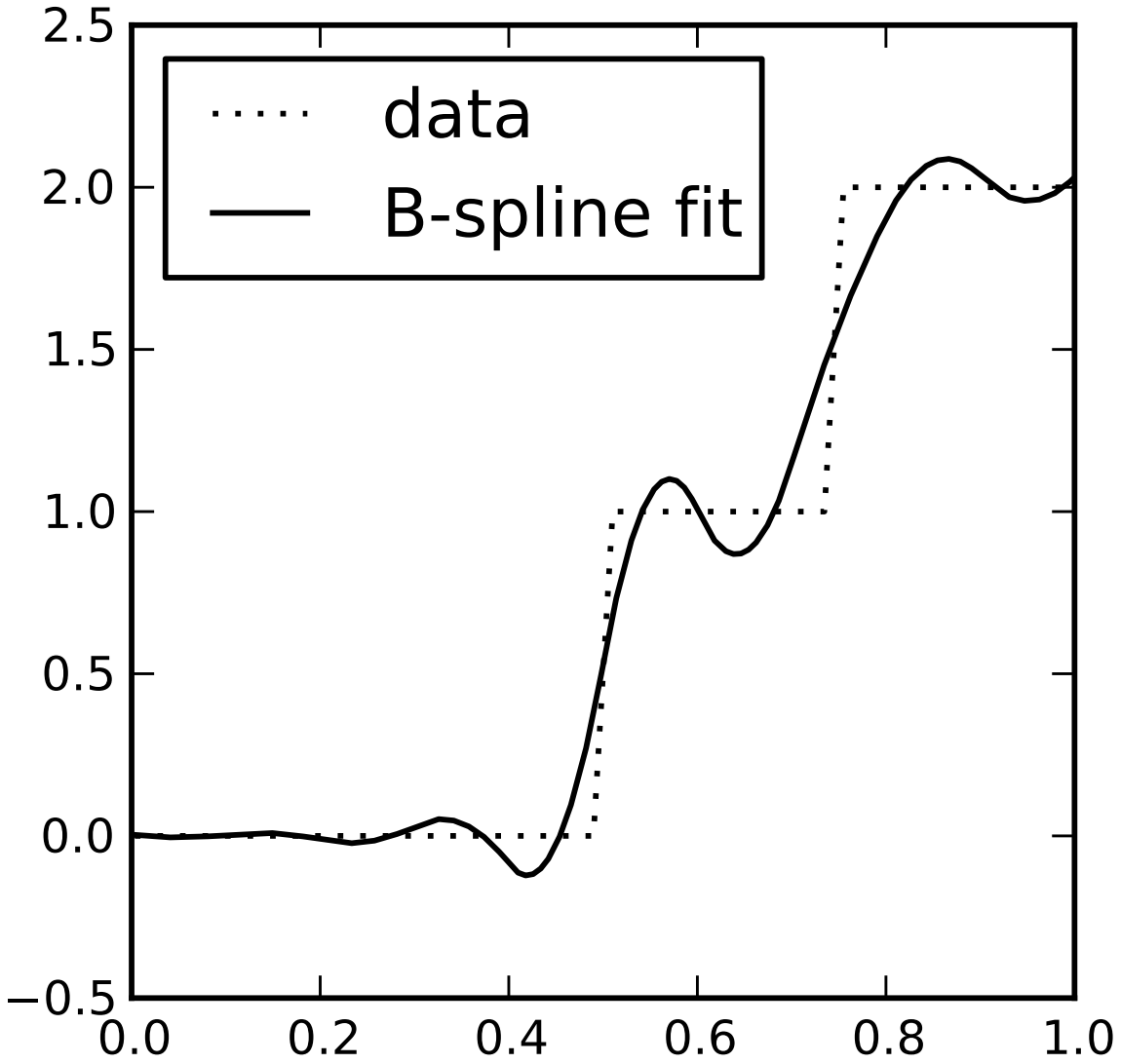
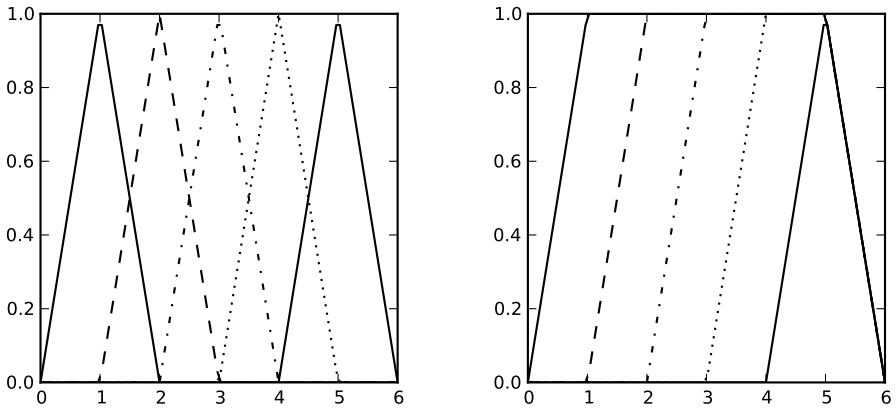
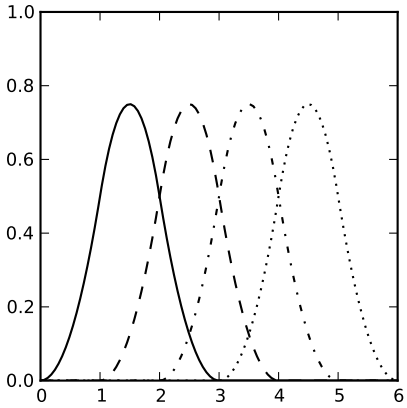


Figure A.5 An example of the "ringing" that can occur as an artifact of least-squares fitting with B-splines. The sharp transitions are unrepresentable in the B-spline basis, so the spline surface must fluctuate around the data in order to minimize the residual. This can cause the spline surface to become negative where the data are strictly positive and non-monotonic where the data are strictly monotonic.

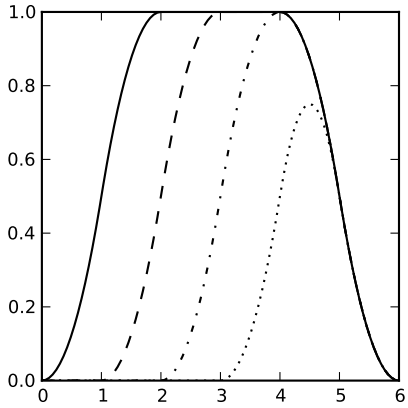


(a) Order-1 B-splines

(b) Order-1 T-splines



(c) Order-2 B-splines



(d) Order-2 T-splines

Figure A.6 B- and T-spline bases of various orders.

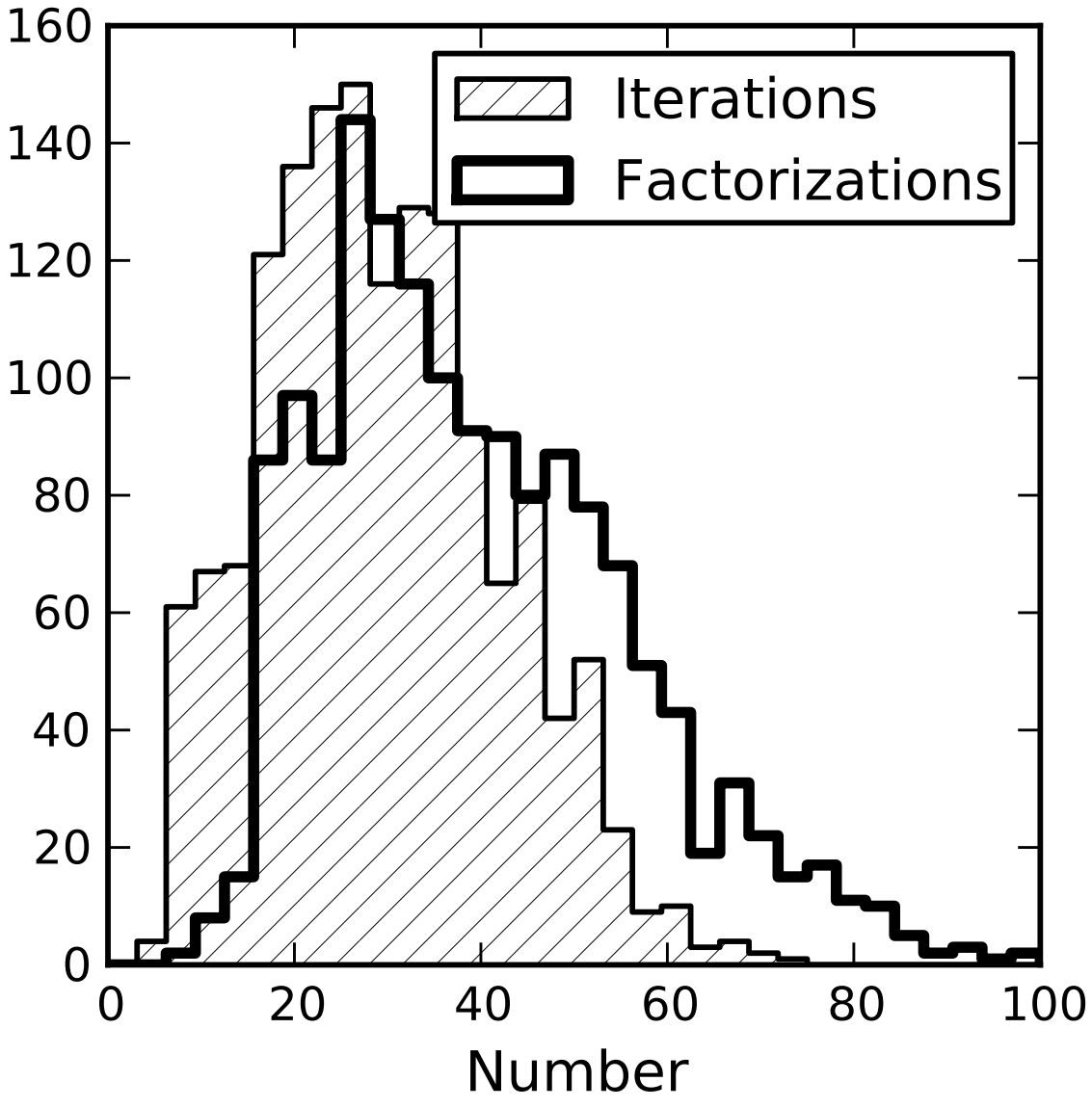
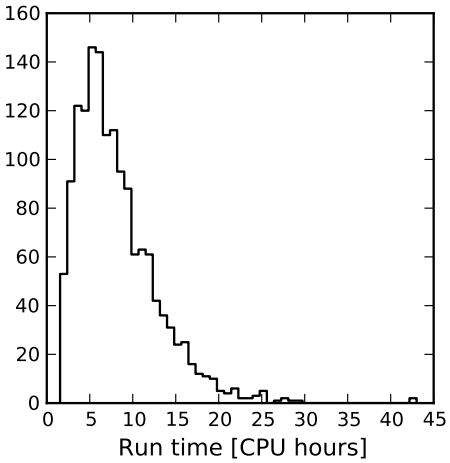
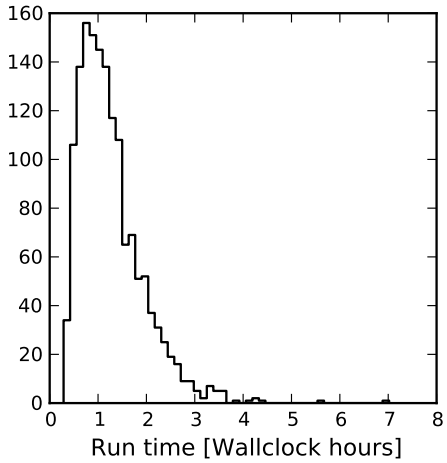


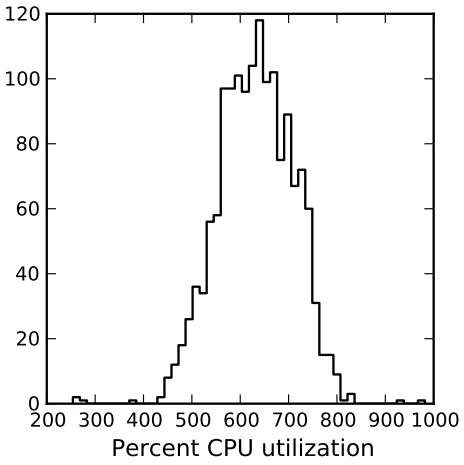
Figure A.7 Number of iterations and complete Cholesky factorizations needed for the BLOCK3 algorithm to converge when fitting the tables described in Section A.6.1.2.



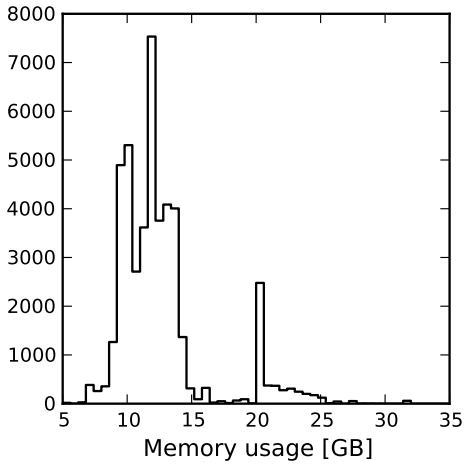
(a) Computation time in CPU hours



(b) Computation time in real hours

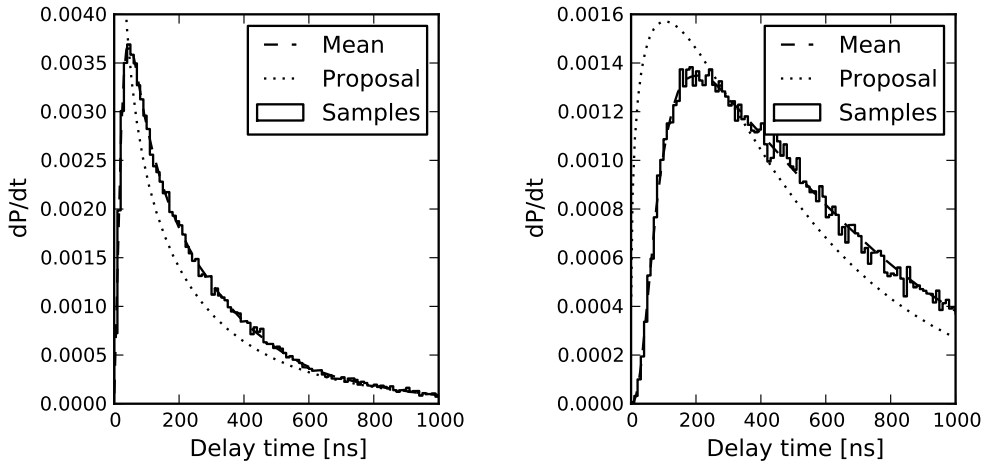


(c) Average CPU utilization



(d) Memory usage sampled every 30 seconds over 400 hours of fitting. The peak at 20 GB represents the memory needed for the initial factorization of the full $A^T A$, while the broader, lower-memory peak represents the memory needed for the factorization of only the unconstrained parts of $A^T A$.

Figure A.8 Resource consumption with two fitting jobs running in parallel on a 16-core, 2.3 GHz AMD Opteron system with 64 GB of memory.



(a) Samples 50 m from the source

(b) Samples 100 m from the source

Figure A.9 Sampled photon time delays from a fit EM cascade, showing the proposal distribution (widened Pandel) and the spline-tabulated probability density function.

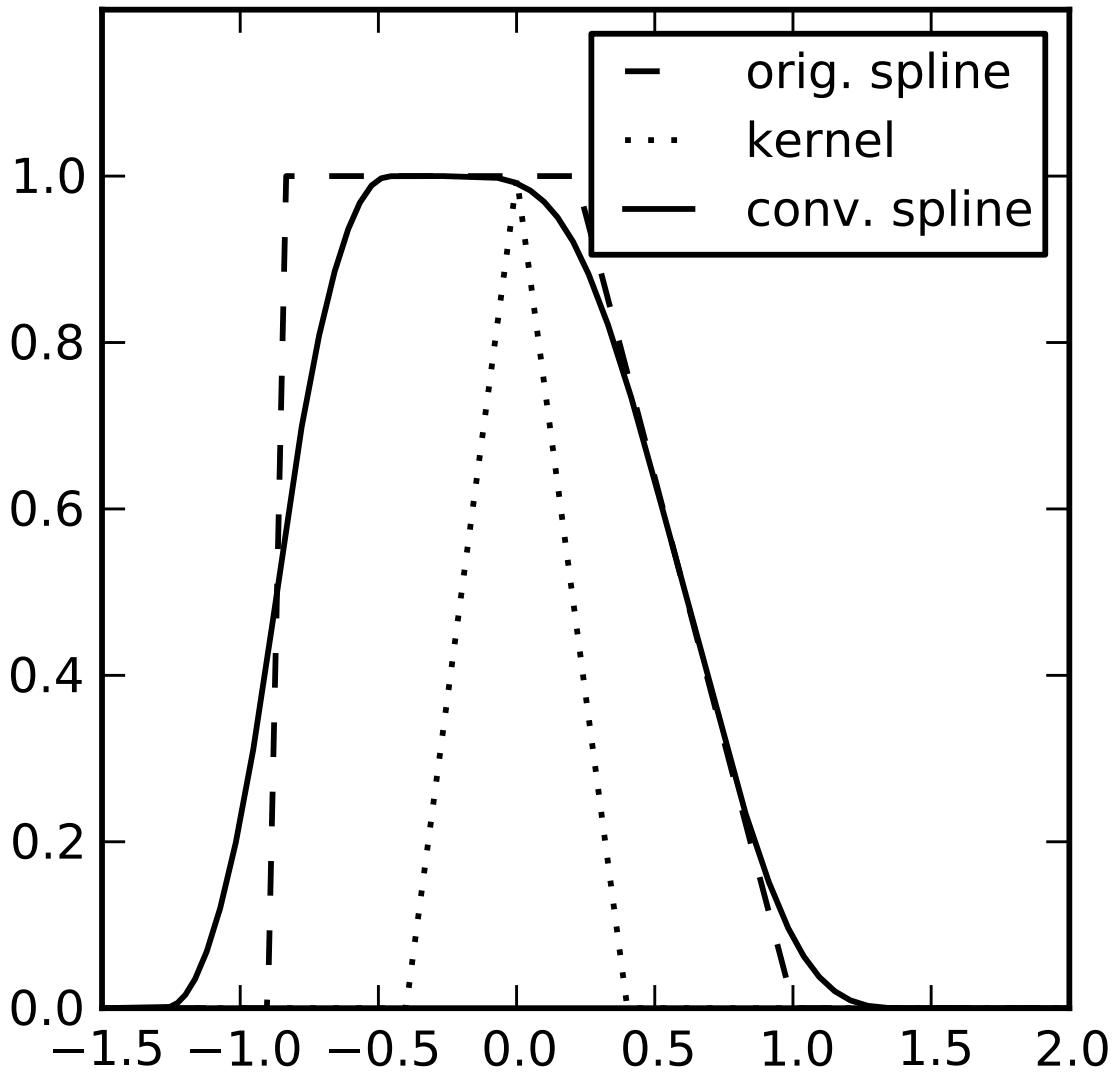


Figure A.10 An example of spline convolution. Here, an order-1 spline surface on a non-uniform knot field is convolved with an order-1 spline, resulting in an order-3 spline surface.

APPENDIX
Burst Catalogs

B.1 IceCube 40

Name	Time (UTC)	T_{start}	T_{stop}	RA (°)	Declination (°)	Position Uncertainty
GRB080408	18:12:48	0.00	20.00	114.66496	33.30414	0.50”
GRB080409	01:22:57	-12.52	10.98	84.32975	5.08484	2.00”
GRB080426	13:23:22	1.04	3.24	26.49890	69.46823	1.80”
GRB080430A	19:53:02	-0.23	21.37	165.31108	51.68567	0.50”
GRB080503	12:26:13	0.62	221.02	286.61956	68.79381	0.25”
GRB080506A	17:46:21	26.02	189.02	329.42433	38.98503	0.50”
GRB080507	07:45:00	0.00	49.03	233.68079	56.43563	0.70”
GRB080513	05:14:32	-21.14	11.11	163.28300	28.19500	10.98’
GRB080514B	09:55:56	0.00	13.17	322.84463	0.70789	0.60”
GRB080515	06:01:13	-1.79	24.81	3.16343	32.57894	3.80”
GRB080517	21:22:51	0.08	69.28	102.24189	50.73496	1.80”
GRB080524	04:13:00	-2.00	8.00	268.44900	80.14300	2.50’
GRB080603A	11:18:11	0.00	180.00	279.40875	62.74407	0.30”
GRB080603B	19:38:13	0.18	75.62	176.53160	68.06111	0.30”
GRB080604	07:27:01	-27.15	67.35	236.96540	20.55780	0.50”
GRB080605	23:47:57	-4.65	30.85	262.12522	4.01555	0.50”
GRB080607	06:07:27	-5.90	154.70	194.94665	15.91965	0.50”
GRB080613A	09:35:21	0.00	30.00	213.27092	5.17319	0.50”
GRB080625	12:28:31	-5.00	80.00	298.40440	56.27790	0.50”
GRB080701A	10:13:37	-2.07	25.23	45.83882	75.47479	1.70”
GRB080702A	11:50:43	0.68	1.18	313.05081	72.31271	1.90”
GRB080707	08:27:53	-1.12	27.28	32.61833	33.10950	1.10”
GRB080710	07:13:10	-96.11	44.09	8.27363	19.50130	0.50”
GRB080726	01:26:10	0.00	12.00	20.39800	13.91300	5.00’

Name	Time (UTC)	T_{start}	T_{stop}	RA (°)	Declination (°)	Position Uncertainty
GRB080727B	08:13:24	0.75	10.35	276.85851	1.16272	1.90”
GRB080727C	23:07:35	-2.91	116.69	32.63485	64.13754	1.50”
GRB080810	13:10:12	-20.92	111.68	356.79275	0.31981	0.60”
GRB080816A	12:04:18	-1.00	69.00	156.20000	42.60000	2.00°
GRB080818B	22:40:49	-1.30	10.00	98.10000	7.40000	7.30°
GRB080822B	21:02:52	0.96	64.96	63.56000	25.76000	3.60’
GRB080830	08:50:16	-3.00	28.00	160.10000	30.80000	2.50°
GRB080903	01:12:23	-10.20	68.30	86.79206	51.26433	1.60”
GRB080916B	14:44:47	-2.98	36.02	163.66512	69.06545	5.10”
GRB080920	06:25:48	-6.00	79.00	121.60000	8.90000	5.40°
GRB080925	18:35:55	-8.15	21.85	96.10000	18.20000	1.20°
GRB080927	11:30:32	-1.30	20.20	61.30000	27.40000	4.60°
GRB081003A	13:46:12	-12.00	18.00	262.39082	16.57103	3.90”
GRB081003B	20:48:08	0.00	30.00	285.02560	16.69140	1.50’
GRB081003C	15:27:17	-4.00	51.00	259.10000	35.40000	6.90°
GRB081009	03:20:58	-0.10	49.30	250.50000	18.40000	1.00°
GRB081011	00:28:50	-0.44	9.46	220.34428	33.54364	0.50”
GRB081022	14:23:48	-8.50	208.10	226.58400	12.40900	1.40’
GRB081024A	05:53:08	-0.84	1.16	27.87380	61.33140	1.90”
GRB081024B	21:22:41	-0.47	0.03	322.90000	21.20400	9.60’
GRB081025	08:22:02	61.75	90.10	245.36626	60.47428	6.70”
GRB081028A	00:25:00	30.09	442.29	121.89480	2.30830	1.50”
GRB081102A	17:44:39	-19.68	63.32	331.17240	52.99420	1.50”
GRB081102B	08:45:00	0.28	1.31	225.30000	22.00000	8.60°
GRB081105B	14:43:51	0.22	0.27	215.80000	38.70000	11.40°
GRB081107	07:42:01	-0.09	1.77	51.00000	17.10000	3.50°
GRB081110	14:25:43	0.03	20.03	111.70000	21.40000	1.80°
GRB081119	04:25:27	-0.44	0.06	346.50000	30.00000	15.20°
GRB081122A	12:28:12	-0.79	26.21	339.10000	40.00000	1.00°
GRB081126	21:34:10	-21.12	46.18	323.51496	48.71064	0.50”
GRB081127	07:05:08	-30.30	9.70	332.06420	6.85050	1.70”
GRB081128	17:18:44	-68.30	63.40	20.80410	38.12710	1.80”
GRB081203A	13:57:11	-68.33	405.67	233.03158	63.52081	0.50”

Name	Time (UTC)	T_{start}	T_{stop}	RA (°)	Declination (°)	Position Uncertainty
GRB081203B	13:52:02	-33.05	50.40	228.79863	44.42859	0.50”
GRB081204B	12:24:26	-0.29	-0.10	150.80000	30.50000	10.20°
GRB081206A	06:35:53	-6.12	15.38	120.10000	32.80000	6.40°
GRB081207	16:18:47	9.14	66.54	112.40000	70.50000	1.20°
GRB081209	23:31:56	0.28	0.72	45.30000	63.50000	4.90°
GRB081211B	06:15:02	0.00	102.00	168.26460	53.83000	2.00”
GRB081215A	18:48:37	-1.94	13.42	125.60000	54.00000	1.00°
GRB081216	12:44:00	-0.14	0.82	129.20000	7.60000	4.40°
GRB081217	23:34:49	0.01	18.44	116.80000	26.80000	2.00°
GRB081223	10:03:57	0.07	0.58	112.50000	33.20000	3.80°
GRB081224	21:17:55	0.01	20.01	201.70000	75.10000	1.00°
GRB081226C	03:44:52	-3.19	11.11	193.00000	26.80000	2.40°
GRB081228	01:17:40	0.65	3.65	39.46160	30.85260	2.90”
GRB081229	04:29:02	-0.23	0.09	172.60000	56.90000	8.80°
GRB090102	02:55:45	-13.04	20.24	128.24390	33.11421	1.10”
GRB090107A	04:48:04	0.50	13.20	302.40900	4.74400	2.40’
GRB090107B	16:20:36	-2.00	27.27	284.81750	59.59510	1.80”
GRB090108A	00:29:02	0.20	1.10	260.80000	46.00000	3.80°
GRB090109	07:58:30	-3.31	0.79	129.60000	51.80000	9.80°
GRB090111	23:58:21	-2.20	26.30	251.67557	0.07728	1.70”
GRB090112B	17:30:15	0.00	11.00	192.30000	25.40000	1.70°
GRB090113	18:40:39	-0.73	10.17	32.05678	33.42857	1.70”
GRB090118	13:54:02	66.00	85.00	49.82760	18.41480	7.10”
GRB090126A	02:01:15	0.00	60.00	3.65770	81.37330	2.20”
GRB090126B	05:26:22	-4.38	5.86	189.20000	34.10000	3.60°
GRB090131	02:09:21	-1.81	49.19	352.30000	21.20000	1.00°
GRB090206	14:52:42	0.11	0.68	156.20000	8.80000	8.70°
GRB090207	18:39:10	-1.21	16.20	252.70000	34.90000	3.80°
GRB090219	01:46:18	-0.03	0.61	26.50000	59.20000	5.20°
GRB090222	04:17:09	-2.11	15.30	118.60000	45.00000	4.30°
GRB090227B	18:31:01	0.29	0.81	11.80000	32.20000	1.80°
GRB090228B	23:25:01	-1.39	5.78	357.60000	36.70000	3.30°
GRB090301A	06:55:55	-16.86	72.34	338.14200	26.63900	1.00’

Name	Time (UTC)	T_{start}	T_{stop}	RA (°)	Declination (°)	Position Uncertainty
GRB090301B	07:33:37	-0.56	4.56	352.80000	9.50000	5.00°
GRB090305B	01:14:35	0.22	3.22	135.00000	74.30000	5.40°
GRB090306C	05:52:05	-3.75	6.49	137.00000	57.00000	4.10°
GRB090313	09:06:27	-20.88	67.02	198.40088	8.09700	0.50”
GRB090320B	19:13:46	-7.30	52.10	183.40000	49.80000	9.50°
GRB090323	00:02:42	-1.37	154.15	190.70954	17.05322	0.50”
GRB090328B	17:07:04	0.81	1.07	155.70000	33.40000	7.90°
GRB090401A	00:00:59	-9.03	160.37	350.92000	29.76200	1.00’
GRB090404	15:56:30	-34.76	95.04	239.23986	35.51615	1.80”
GRB090408B	19:46:38	5.36	10.99	43.98000	26.61000	29.39’
GRB090409	06:54:01	0.44	5.54	302.10000	1.10000	9.60°
GRB090410	16:57:52	-58.61	151.39	334.95600	15.41900	1.80’
GRB090411B	23:47:44	-7.00	21.78	38.50000	5.10000	2.40°
GRB090417B	15:20:03	279.03	617.43	209.69310	47.01700	5.00”
GRB090418A	11:07:40	-8.28	61.32	269.31321	33.40585	0.50”
GRB090418B	08:59:02	20.17	145.00	225.91000	17.22400	1.91’
GRB090424	14:12:09	-0.78	103.53	189.52128	16.83753	0.74”
GRB090425	09:03:30	-0.03	80.64	118.60000	68.10000	2.10°
GRB090426	12:48:47	0.21	1.51	189.07531	32.98600	0.75”
GRB090428A	10:34:38	-0.05	2.50	210.10000	39.50000	4.20°
GRB090428B	13:15:11	-9.14	15.36	0.80000	11.50000	3.90°
GRB090429B	05:30:03	-4.14	2.76	210.66688	32.17064	0.50”
GRB090429C	12:43:25	-1.30	11.70	260.00000	54.30000	4.80°
GRB090429D	18:03:57	-12.74	1.60	124.40000	7.90000	5.00°
GRB090511	16:25:16	-4.73	4.47	161.90000	51.30000	7.00°
GRB090518	01:54:44	-2.13	7.56	119.95404	0.75921	1.60”
GRB090519	21:08:56	-11.98	60.82	142.27917	0.18031	0.50”

B.2 IceCube 59

Name	Time (UTC)	RA (°)	Declination (°)	Position Uncertainty (°)	T_{start}	T_{stop}
GRB090520A	01:37:16	11.6	-8	0.05	0	10
GRB090520B	19:57:53	332	43.2	12	-0.384	0.256

Name	Time (UTC)	RA (°)	Declination (°)	Position Uncertainty (°)	T_{start}	T_{stop}
GRB090520C	20:23:19	111.2	-19.7	1.2	-0.512	3.968
GRB090520D	21:01:37	131.3	-18	4.3	-5.12	13.3
GRB090522A	08:15:49	277.7	19.6	4.9	-2	20
GRB090524A	08:17:56	329.5	-67.4	1.3	0.003	71.681
GRB090528A	04:09:01	134.9	-35.8	3.4	-4	29.696
GRB090528B	12:22:31	312.2	32.7	1	-2.048	57.345
GRB090529A	14:12:35	212.5	24.5	0.0005	-49	39.3
GRB090529B	07:26:22	231.2	32.2	7.2	-1.792	5.376
GRB090529C	13:32:00	162.7	47.3	1.5	-0.002	10.048
GRB090530A	03:18:18	179.4	26.6	0.0004	-12.2	51.8
GRB090530B	18:14:24	73.2	13.8	1	-1	219
GRB090531A	01:45:17	178.7	7.8	0.0183	-12.6	29.4
GRB090531B	18:35:56	252.1	-36	0.035	-0.23	67
GRB090602A	13:32:22	248.9	-65	3.4	-1	9
GRB090606A	11:18:08	146.9	-70.5	5.6	-5.1	12.3
GRB090607A	05:30:17	191.2	44.1	0.0317	0.1	2.5
GRB090608A	01:15:26	100.2	-37.4	4.5	-18.4	20.5
GRB090610A	15:33:25	84.2	35.4	5.2	-3.1	3.2
GRB090610B	17:21:31	276	-42.1	9.5	-6.2	161.8
GRB090610C	21:12:07	70.4	30.3	8.2	-4.1	8.2
GRB090612A	14:50:50	81.1	17.8	2.2	-1.024	5.12
GRB090616A	03:45:42	103.1	-3.7	10.4	-0.128	0.512
GRB090617A	04:59:58	78.9	15.7	4.2	-0.032	1
GRB090618A	08:28:26	294	78.4	0.0005	-4.4	213.6
GRB090620A	09:36:23	237.4	61.2	1	-0.003	16.64
GRB090621A	04:26:34	11	61.9	0.0004	-6.1	45.06
GRB090621B	22:07:25	313.5	69	0.0283	-0.028	0.148
GRB090621C	10:00:52	257.5	-28.5	3.3	-3.072	28.672
GRB090621D	10:43:45	12.3	-22.6	4.9	-1.024	20.48
GRB090623A	02:34:17	309	-43.2	2	-1.92	48
GRB090625A	05:37:00	20.3	-6.4	3.1	-6.144	12.288
GRB090625B	13:26:22	2.3	-65.8	0.0283	-2.048	6.144
GRB090626A	04:32:08	169.3	-36.1	1	-0.256	69.377

Name	Time (UTC)	RA (°)	Declination (°)	Position Uncertainty (°)	T_{start}	T_{stop}
GRB090628A	21:20:12	237.1	-16	0.0005	-25	4
GRB090630A	07:27:21	146.6	-46.6	5.8	-1.024	4.096
GRB090701A	05:23:55	114.7	-42.1	4.2	-10	2
GRB090702A	10:40:37	175.9	11.5	0.0417	0	10
GRB090703A	07:54:02	0.8	9.7	5.3	-2.6	6.7
GRB090704A	05:47:48	208.2	22.8	0.0283	0	70
GRB090705A	05:11:31	318.5	62.6	0.05	0	10
GRB090706A	06:47:40	205.1	-47.1	3	-34.8	7.2
GRB090708A	03:38:18	154.6	26.6	0.0283	-1.3	17.7
GRB090709A	07:38:34	289.9	60.7	0.0004	-10	110
GRB090709B	15:07:41	93.5	64.1	0.03	-2	35.9
GRB090711A	20:23:22	139.6	-64.7	1	-20.992	82.433
GRB090712A	03:51:00	70.1	22.5	0.0267	-22.528	49.153
GRB090713A	00:29:28	284.8	-3.3	2.4	-4.1	55.3
GRB090715A	17:25:36	152.1	10	0.04	-0.1	67.8
GRB090715B	21:03:19	251.3	44.8	0.0004	-11.6	82.176
GRB090717A	00:49:32	86.8	-64.2	1	2.8	60
GRB090717B	02:40:31	247	23	3.9	-0.256	0.128
GRB090718A	17:16:42	242.5	-6.6	7.4	0	10
GRB090718B	18:17:42	274.1	-36.4	1.2	0	34
GRB090719A	01:31:26	341.3	-67.9	1	0	16.6
GRB090720A	06:38:08	203.7	-10.3	0.05	-1.856	5.312
GRB090720B	17:02:56	203	-54.8	2.9	0	20.1
GRB090726A	22:42:27	248.7	72.9	0.0004	-34.7	47.3
GRB090727A	22:42:18	316	64.9	0.0004	-1.3	318.9
GRB090728A	14:45:45	29.7	41.6	0.0005	-5	63.2
GRB090802A	05:39:03	51	37.9	4.9	-0.016	0.056
GRB090802B	15:58:23	267	-71.8	10.7	0	10
GRB090807A	15:00:27	273.7	10.3	0.0005	-9.5	151.5
GRB090807B	19:57:59	326.9	7.2	2.6	-0.64	1.408
GRB090809A	17:31:14	328.7	-0.1	0.0002	-1	6.4
GRB090809B	23:28:14	95.3	0.1	1.2	0	14.3
GRB090812A	06:02:08	353.2	-10.6	0.0005	-30	40

Name	Time (UTC)	RA (°)	Declination (°)	Position Uncertainty (°)	T_{start}	T_{stop}
GRB090813A	04:10:42	225.8	88.6	0.0005	-0.8	8.96
GRB090814A	00:52:19	239.6	25.6	0.0005	-16.4	73.6
GRB090814B	01:21:33	64.7	60.6	0.0483	0	50
GRB090814C	08:49:41	332.5	58.9	5.5	-0.064	0.192
GRB090815A	07:12:12	41.7	-2	7.6	-5.12	24.576
GRB090815B	10:30:41	21.4	53.4	5.7	-7.168	23.552
GRB090817A	00:51:26	64	44.1	0.0003	0	41
GRB090820A	00:38:16	87.7	27	1	28	60
GRB090820B	12:13:16	318.3	-18.6	9.6	-0.128	12.16
GRB090823A	16:11:10	128.7	60.6	0.0752	0	14.848
GRB090826A	01:37:31	140.6	-0.1	9.7	-4.096	9.216
GRB090827A	19:06:27	18.4	-50.9	0.0467	0	8.848
GRB090828A	02:22:48	124.4	-26.1	1.2	0	103
GRB090829A	16:07:38	329.2	-34.2	1	-40	92
GRB090829B	16:50:40	355	-9.4	3.2	-1.5	31.2
GRB090831A	07:36:36	145.1	51	1.9	0	49.408
GRB090831B	18:01:14	3.6	28	0.05	0	10
GRB090831C	21:30:25	108.3	-25.1	0.0005	0.6	4.9
GRB090902A	09:38:05	291	53.1	3.8	-0.256	0.896
GRB090902B	11:05:08	264.9	27.3	0.0006	-0.5	24.5
GRB090904A	01:01:06	100.9	50.2	0.0006	29.3	186.3
GRB090904B	01:24:13	264.2	-25.2	0.0005	-2.5	65.5
GRB090910A	19:29:48	296.2	72.3	1	-7.2	44
GRB090912A	15:50:29	188	61.5	0.0005	-15.9	160.1
GRB090915A	15:35:36	238	15.5	0.05	-3	5
GRB090916A	07:00:44	126.6	25.9	0.045	0	68.5
GRB090922A	12:56:42	17.1	74.3	1	0.003	10.24
GRB090925A	09:20:33	333.2	14.3	4.5	-1	20
GRB090926A	04:20:26	354.5	-64.2	1	-0.5	34.5
GRB090926B	21:55:28	46.3	-39	0.0167	-21.6	48.6
GRB090927A	10:07:17	344	-71	0.0006	-0.256	2.4
GRB090929A	04:33:03	56.5	-6	0.05	-1	8.448
GRB090929B	10:09:25	117.7	-0.7	0.0005	-2	371

Name	Time (UTC)	RA (°)	Declination (°)	Position Uncertainty (°)	T_{start}	T_{stop}
GRB091003A	04:35:45	251.1	37.2	1	-0.384	26.88
GRB091010A	02:43:09	298.7	-22.5	0.05	-0.128	8.448
GRB091018A	20:48:19	32.2	-57.5	0.0005	-0.3	7.7
GRB091020A	21:36:43	175.7	51	0.0001	-7.7	38.3
GRB091024A	08:55:58	339.2	56.9	0.0167	-15.2	135.8
GRB091026A	13:11:30	276.6	-86.1	0.0005	-1	47.9
GRB091029A	03:53:22	60.2	4	0.0002	-1.8	60.2
GRB091030A	19:52:26	41.7	21.5	1.2	-1.7	40
GRB091031A	12:00:28	70.6	-59.1	1	-0.003	41.216
GRB091102A	14:34:38	72.6	-72.5	0.02	-0.5	7.5
GRB091104A	08:49:22	208.8	47.4	0.0004	-1.5	120.5
GRB091109A	04:57:43	309.3	0	0.0002	-0.8	51.5
GRB091109B	21:49:03	112.7	-54.1	0.0008	0	1
GRB091111A	15:21:59	137.8	-45.9	0.0417	0	10
GRB091112A	17:41:15	257.8	-36.7	0.025	-0.512	24.064
GRB091117A	17:44:29	30.9	-16.9	0.0433	-2.5	3.2
GRB091120A	04:34:40	224.8	-24.8	1.32	-3	57.6
GRB091123A	07:08:37	297.1	-29.2	2.4	-5	650
GRB091126A	07:59:24	83.3	-19.3	5.4	-0.256	0.5
GRB091126B	09:19:48	47.4	31.5	14.3	-0.016	0.016
GRB091127A	23:25:45	36.6	2.4	0.0002	-0.4	9.984
GRB091128A	06:50:34	127.7	1.7	1.4	-24.832	71.425
GRB091130A	16:08:10	161.9	11.6	0.05	0	10
GRB091130B	17:59:04	203.1	34.1	0.0004	-4.7	127.4
GRB091202A	23:10:12	138.8	62.5	0.0417	0	50
GRB091208A	08:46:03	0.3	65.7	0.0183	-0.3	32.8
GRB091208B	09:49:57	29.4	16.9	0.0005	-0.128	22.3
GRB091221A	20:52:57	55.8	23.2	0.0005	-2	41.1
GRB091230A	06:27:30	132.9	-53.9	0.0417	0	120
GRB100103A	17:42:46	112.4	-34.5	0.0267	0	41.216
GRB100111A	04:12:49	247	15.6	0.0004	-2.7	8.7
GRB100113A	17:29:24	19.9	-25.2	2.3	-8	96
GRB100115A	11:15:19	3.4	-0.8	0.0011	0	10

Name	Time (UTC)	RA (°)	Declination (°)	Position Uncertainty (°)	T_{start}	T_{stop}
GRB100116A	21:31:00	301.5	16.2	1	0	23.296
GRB100117A	21:06:19	11.3	-1.6	0.0283	0	0.3
GRB100119A	18:25:06	299.3	-53.1	0.0217	-15.8	37.5
GRB100122A	14:47:37	79.2	-2.7	1.3	-2.688	25.984
GRB100130B	18:38:35	77.6	20.4	1.41	-14	89
GRB100131A	17:30:57	120.4	16.5	1.2	-0.5	5.5
GRB100203A	18:31:07	96.2	4.8	0.0333	0	10
GRB100205A	04:18:43	141.4	31.7	0.0005	-12.6	25.6
GRB100205B	11:45:38	133.9	-23	8.2	-1.024	7.168
GRB100206A	13:30:05	47.2	13.2	0.0009	-0.008	0.124
GRB100212A	14:07:29	356.4	49.5	0.0006	-0.51	134
GRB100213A	22:27:48	349.4	43.4	0.0006	-0.7	2.2
GRB100213B	22:58:34	124.3	43.4	0.001	-12.7	35.3
GRB100216A	10:07:00	154.3	35.5	0.11	0	10
GRB100218A	04:38:45	206.6	-11.9	2.2	-4.1	28.67
GRB100219A	15:15:46	154.2	-12.6	0.0004	19	45
GRB100223A	02:38:09	104.1	2.8	3.6	-0.01	0.22
GRB100224A	15:20:10	83.5	-8	0.05	0	10
GRB100224B	02:40:55	269.6	-17.1	1.6	-7.2	71.7
GRB100225A	02:45:31	312.5	-54.9	2.4	-3.8	13.6
GRB100302A	19:53:06	195.5	74.6	0.0005	47.6	67.3
GRB100305A	09:05:38	168.4	42.4	0.0005	-9.2	70.3
GRB100316A	02:23:00	252	71.8	0.0008	-1.2	6.7
GRB100316B	08:01:36	163.5	-45.5	0.0001	-0.4	4
GRB100316C	08:57:59	32.3	-68	0.0005	-3	7
GRB100316D	12:44:50	107.6	-56.3	0.05	0	10
GRB100319A	18:34:50	278.4	-8.5	0.05	0	10
GRB100322A	01:05:09	23.3	-10.2	1.2	-1	42
GRB100322B	07:06:18	76.5	42.7	0.0667	0	10
GRB100324A	00:21:27	98.6	-9.7	0.0417	0	10
GRB100324B	04:07:32	39.672	19.287	0.05796	0	13.824
GRB100325A	06:36:08	332.3	-28.6	1.5	-1.664	7.552
GRB100331B	21:08:38	303	-11.1	0.1	0	10

Name	Time (UTC)	RA (°)	Declination (°)	Position Uncertainty (°)	T_{start}	T_{stop}
GRB100401A	07:07:32	290.8	-8.3	0.0333	-1.9	3.2
GRB100413A	17:33:28	266.2	15.8	0.0005	-2	227.4
GRB100413B	08:42:41	356.8	51.3	0.05	0	10
GRB100414A	02:20:21	192.1	8.7	0.0006	-1.5	28.9
GRB100418A	21:10:08	256.4	11.5	0.0002	-1.1	7.8
GRB100420A	05:22:42	296.1	55.8	0.0006	-30	70
GRB100423A	00:34:57	136.5	21.5	0.0167	-1	99.1
GRB100423B	05:51:25	119.7	5.8	1.5	-2	23.6
GRB100424A	16:32:42	209.4	1.5	0.0005	22.3	148.5
GRB100425A	02:50:45	299.2	-26.4	0.0004	-2.1	39.4
GRB100427A	08:32:08	89.2	-3.5	0.0417	-4.1	11.3
GRB100503A	13:18:03	147.5	4	1.5	-4	89
GRB100504A	19:20:55	255.6	-35.6	0.0007	-6.1	102.2
GRB100508A	09:20:42	76.2	-20.7	0.0005	-8	59
GRB100510A	19:27:06	355.1	-18.9	5	-11	54
GRB100511A	00:49:56	109.3	-4.7	1	-7.4	52
GRB100513A	02:07:08	169.6	3.6	0.0005	-64.1	44.6
GRB100514A	18:53:58	328.8	29.2	0.0005	0.3	30.8
GRB100518A	11:33:35	304.8	-24.6	0.0267	0	30
GRB100522A	03:45:52	7	9.4	0.0004	-0.003	8.848
GRB100526A	16:26:10	230.8	25.6	0.0005	-33.2	80
GRB100526B	19:00:38	0.8	-37.9	0.05	-28.4	35.6
GRB100528A	01:48:01	311.1	27.8	0.0833	0	24.576