The Design and Fabrication of Optimal Light Collectors for the CTF Upgrade

Jason Xavier Prochaska

Physics Department

Princeton University

Advisor: Frank Calaprice

April 19, 1993

To Dad, Mom, Jodi, and Gabielle-Elisabeth

The Design and Fabrication
of Optimal Light Collectors
for the CTF Upgrade

Abstract

The results of over 2 decades of neutrino experiments have revealed neutrino flux rates significantly below those predicted by standard solar models - thus, the Solar Neutrino Problem (SNP). The Borexino experiment, through measurements of the ^{7}Be neutrinos, will provide strong evidence for either an astrophysical explanation to the SNP or a neutrino physics solution. It will also have the capability to detect time variations in the flux rate, thereby confirming or denying the day-night effect prediction offered by the MSW Theory.

The success of Borexino lies in achieving ultra-low background levels. These levels are so low, that a Counting Test Facility (CTF) has been proposed to examine the purity of Borexino's components. Two designs for the CTF have been offered; one, the CTF Upgrade, allows for 3D positional measurements. In order to achieve this 3D resolution while minimizing the number of photomultiplier tubes needed, large light collectors have been proposed.

The design of light collectors optimal for the CTF Upgrade has been investigated. The Edge-Ray Principle, derived for the Winston CPC cone (ideal for planar PMT's and an infinite source), has been extended to the construction of a String Method cone ideal for a spherical PMT and a spherical finite source. The theoretical performance of the String Method cone has been examined by computer Monte Carlo simulations and shown to be optimal for the CTF layout, particularly in terms of its radial transmission. Routines have been presented which calculate the coverage of an array of SM cones and the corresponding value for the number of photoelectrons detected / MeV incident neutrino energy.

A fabrication method, the Panel Technique, has been developed in detail, offering an option for the construction of these large light cones. Two prototype light collectors have been designed and preliminary construction has been made.

Acknowledgements:

Because of the diverse nature of this project, I have a great number of people to acknowledge. In no particular order, then, thanks to Prof. Frank Calaprice for his support and the freedom to investigate whatever avenues of the project I desired. I would like to acknowledge Nick Darton who introduced me to the Cyclotron lab and the Borexino Project. Much thanks to Laszlo Varga, Bill Dix, Glenn Atkinson, and Ole Lillestole for their great patience and insight into all of the machining problems I had. I would also like to thank Professors Bruce Vogelaar and Mike Lowry for their help with my computer simulations. I am very grateful to Martin Moorhead for all of the insight that he shared on ideal light concentrators.

I am indebted to Stan Anderson, Albert Young, Greg Berman, Gordon Jones and Eric Jacobsen for all the "quick" questions they answered. I am grateful to Joe Horvath for his guidance with the evaporative coating. I would like to thank Bob Parsells and Bob Walls who took time out from the war zone at PPL to provide much needed engineering and fabrication advice. I want to acknowledge Ezra Jennings for the extensive use of his computing facilities. Overwhelming thanks to Pat Santorro, Steve Kidner, Fred Loser, Canthi Gorman and Amir Razzaghi without whose guidance I would never had gotten anything done on the B floor. Having hopefully left no one out, I want to express my love and respect for my family (Mom, Dad and Jodi) who have always supported me unconditionally. Finally, I want to acknowledge my love for Gabrielle, who has patiently survived this year long project with me and will likely have to endure many more.

Contents:

Appendix - Programs

1. SM Profile
2. Ray Tracing
    A. 3D Transmission-Radial Curve
    B. 2D Transmission-Radial Curve
    C. 2D Transmission-Angle Curve
    D. 3D Pe/MeV - Radial Curve
3. Fabrication Programs

Chapter 1 - Background

A. Overview

For over 25 years now scientists have been actively investigating the solar neutrino production rate. Although measurement techniques have steadily improved and background noise (e.g. cosmic rays) has been markedly reduced, the measured rate still falls far short of rates calculated from the standard solar model[1]. This empirical discrepancy is infamously knows as the Solar Neutrino Problem (SNP). Granted no significant errors with the experimental techniques, the SNP points to a flaw in our astrophysical understanding of the solar interior or to new properties neutrinos that would alter the standard theory of electroweak interaction. With the fate of such fundamental concepts at stake, scientists search for further evidence that will finally resolve the SNP.

Presently, a team of scientists from Princeton, MIT, AT&T, Italy and Germany have developed a proposal to construct an ultra-low background neutrino detector to further investigate the SNP. The Borexino Project is a scintillator detector allowing real-time analysis of solar neutrinos, in particular the ^{7}Be neutrino of the pp-chain (Section 1.C). Observations of the flux of these neutrinos will provide further insight into the SNP possibly identifying whether the solution lies in new astrophysical or electroweak theories. This thesis focuses on a specific element of the overall Borexino Project, the optics of Borexino's Counting Test Facility (CTF). The CTF is an experiment proposed to test the

collaborations's ability to achieve very low background levels for the Borexino Project. I will begin with a discussion of the standard solar model and the SNP describing the MSW Theory, and the ^{37}Cl, Kamiokande II and Gallium experiments. The goals and significant characteristics of Borexino and the CTF will also be addressed. The following two chapters focus on the design and fabrication of light collectors for the CTF. Although my efforts were directed primarily under the constraints of the CTF, the results will certainly be applicable to Borexino and may offer insight into many other light (radiation) detection projects.

B. Standard Solar Model

Due to the Sun's proximity to Earth, it has historically been the most intensely observed star. As such, precise measurements of the Sun's radius, mass and luminosity have been made with high precision. Armed with a wealth of observational data, scientists have developed models of stellar evolution that predict and explain the present measurements. The standard solar model, the model most widely accepted, offers specific predictions for the neutrino fluxes and energy spectrums observable on Earth. It is the discrepancy between these predictions and the observed experimental data that define the SNP.

The standard solar model is founded on several assumptions. First, the Sun is assumed to exist in hydrostatic equilibrium - its interior radiation and particle pressures balance exactly its internal gravitational forces.

This is essential to prevent free-fall collapse or expansion. Second, energy transport in the solar interior is assumed to be primarily due to photon diffusion. Third, the main source of the radiated photons and neutrinos is nuclear fusion. Specifically, the standard solar model states that the pp chain (Section 1.C) is the main process of nuclear fusion in the Sun. Lastly, the initial solar interior is assumed to have been chemically homogenous, with changes in the chemical abundances a function of nuclear reactions only.

In addition to these 4 basic premises, Bachall and Pinsonneault[2] have recently introduced the notion of helium diffusion to the standard solar model. They point out that the diffusion of helium (and other heavy elements) affects the elemental abundance and radiative opacity in the stellar core, consequently affecting the predicted solar neutrino fluxes. In their paper, Bachall and Pinsonneault introduced the effects of helium diffusion into the standard solar model and calculated the adjusted neutrino fluxes for the most important neutrino experiments. Although some results I quote do not include the helium diffusion effects (i.e. all sources before 1992), all data related to the predicted neutrino fluxes of experiments is from the helium diffusion standard solar model - denoted the SSMH.

As Bachall points out[3], the standard solar model not only coincides with the observational data, but it also predicts several impressive stellar relationships. In particular, the model postulates a stellar mass-luminosity relationship which "is in agreement with observations on almost two orders of

magnitude in mass*. The standard theory of stellar evolution has been applied to stars of a wide range of physical and chemical characteristics without significant deviations. It serves as the foundation for a large part of astrophysics and any changes to its postulates could have widespread effects on our understanding of the universe. Thus, the inconsistencies noted in the SNP between the solar standard model and observed neutrino rates have received great attention.

C. Nuclear Generation and Neutrino Flux

In order to offer a more through understanding of the neutrino predictions given by the standard solar model, I will develop a few of the main points on the theoretical solar nuclear processes. The primary process of energy generation in the Sun is the nuclear fusion of 4p into an alpha particle, two positrons and an electron flavored neutrino,

$$4p \rightarrow \alpha + 2e^{+} + 2\nu_{e} \quad .$$

The total energy released in the process is approximately 26.731 MeV, with ~.6 MeV (on average) carried away by neutrinos. This process is responsible for all of the solar photons and neutrinos that we observe on Earth.

The principal nuclear reaction in the Sun is the proton-proton chain (pp chain). Table 1.1 summarizes the pp chain in the Sun. Reaction 1a, the pp reaction, is the most frequent initial reaction and is therefore responsible for most of the energy generation in the Sun. The neutrinos it produces, pp neutrinos, have the highest predicted flux, yet because of

Reaction	Number	Termination† (%)	ν energy (MeV)
$p + p \rightarrow {}^2H + e^+ + \nu_e$	1*a*	100	≤ 0.420
or			
$p + e^- + p \rightarrow {}^2H + \nu_e$	1*b* (pep)	0.4	1.442
${}^2H + p \rightarrow {}^3He + \gamma$	2	100	
${}^3He + {}^3He \rightarrow \alpha + 2p$	3	85	
or			
${}^3He + {}^4He \rightarrow {}^7Be + \gamma$	4	15	
${}^7Be + e^- \rightarrow {}^7Li + \nu_e$	5	15	(90%) 0.861 (10%) 0.383
${}^7Li + p \rightarrow 2\alpha$	6	15	
or			
${}^7Be + p \rightarrow {}^8B + \gamma$	7	0.02	
${}^8B \rightarrow {}^8Be^* + e^+ + \nu_e$	8	0.02	< 15
${}^8Be^* \rightarrow 2\alpha$	9	0.02	
or			
${}^3He + p \rightarrow {}^4He + e^+ + \nu_e$	10 (hep)	0.00002	≤ 18.77

†The termination percentage is the fraction of terminations of the pp chain, $4p \rightarrow \alpha + 2e^+ + 2\nu_e$, in which each reaction occurs. The results are averaged over the model of the current Sun. Since in essentially all terminations at least one pp neutrino is produced and in a few terminations one pp and one pep neutrino are created, the total of pp and pep terminations exceeds 100%.

Table 1.1: The pp-chain in the Sun. [From BAH 89]

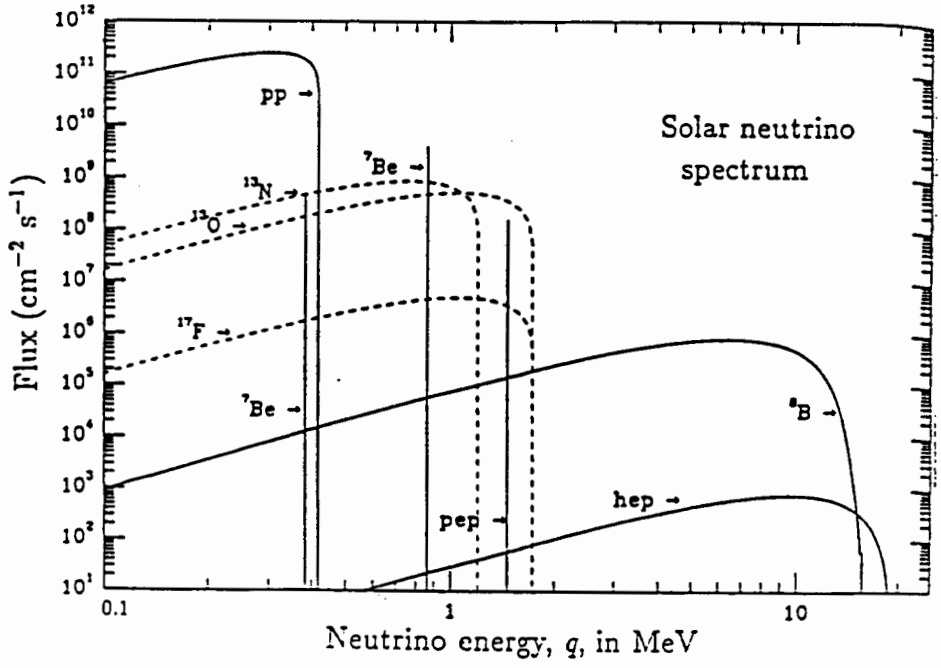

Figure 1.1 Neutrino Spectrum from the Sun: Taken from (BAH 89)

their low energy ($\leq$.420 MeV) they are very difficult to detect in relation to other neutrinos. Reactions that follow, however, lead to the production of other, higher energy neutrinos.

With respect to issues discussed in this thesis, the most important of these reactions involve ^{7}Be and ^{8}B (i.e. reactions 4,5,7 and 8). In reaction 4, a ^{3}He nucleus captures an already excited alpha particle forming ^{7}Be and releasing a gamma ray. Nearly 100% of the time the ^{7}Be nucleus will capture an electron forming ^{7}Li and emit an electron flavored neutrino with energy .861 Mev (90%) or .383 Mev (10%). Very rarely, the ^{7}Be nucleus absorbs a proton forming ^{8}B and releasing a gamma ray (reaction 7). The ^{8}B nucleus will then beta decay to ^{8}Be* releasing a positronium ion and a neutrino with a continuum of energies less than 15 Mev. This reaction occurs approximately once in every 5000 terminations of the pp chain and therefore the ^{8}B neutrinos have a low predicted flux. Figure 1.1 shows the solar neutrino spectrum for the pp chain for the temperatures and densities of the Sun as predicted by the standard solar model. Note the continuum spectrum of the ^{8}B neutrinos with a high maximum energy, yet very small flux. The lines labeled ^{7}Be designate the line fluxes of the two energies of ^{7}Be neutrinos. Because of their higher flux and monoenergetic characteristic, experiments involving the measurement of ^{7}Be have some advantages over ^{8}B detectors. These differences are developed in later sections.

D. MSW Theory

Currently, the most investigated non-astrophysical explanation for the SNP is the MSW Theory, named after Mikheev, Smirnov and Wolfenstein. Developed out of preliminary work by Pontecorvo[4] and Wolfenstein[5], the MSW Theory proposes that neutrino oscillations in matter cause a shift in the ratios of neutrino flavors. Applied to the SNP, it is possible that a significant number of ν_e produced in the solar interior oscillate into another flavor (e.g. muon neutrinos ν_μ). Since most of the neutrino experiments to date are flavor sensitive to ν_e (designated charge-current experiments), neutrino oscillations could explain the reduced observed fluxes. Because of the beautiful simplicity of its explanation and its link to other fields (e.g. cosmological discussions of dark matter), any discussion of the SNP ought to mention its implications. Central to the MSW Theory, however, is the notion that at least one type of neutrino have a finite mass, a hotly debated prediction in high energy physics.

If neutrinos are massive, it is possible that the flavor eigenstates do not exactly match the mass eigenstates, but are actually mixtures (orthogonal combinations) of them. The following is an overview of the physics behind the MSW Effect. To simplify the discussion, we consider only two neutrino flavors to be coupled to one another (e.g. ν_e, ν_μ). In terms of the mass eigenstates, $|\nu_1\rangle$ and $|\nu_2\rangle$, the flavor eigenstates are represented by,

$$|\nu_e\rangle_t = \cos\theta_S\, e^{-iE_1t}|\nu_1\rangle + \sin\theta_S\, e^{-iE_2t}|\nu_2\rangle$$
$$|\nu_\mu\rangle_t = -\sin\theta_S\, e^{-iE_1t}|\nu_1\rangle + \cos\theta_S\, e^{-iE_2t}|\nu_2\rangle \quad ,$$

where $\hbar=1$ and E_i is the energy of the ith mass eigenstate. In this notation the index S indicates that the mixing angle θ is dependent on the medium where the oscillation occurs (i.e. vacuum or matter). We solve for the probability of oscillation by taking the square of the inner product of the two eigenstates. In vacuum, the result is,

$$|\langle\nu_\mu|\nu_e\rangle_t|^2 = \sin^2 2\theta_V \sin^2\left(\frac{\pi R}{L_V}\right) \qquad \text{w/} \quad L_V \equiv \frac{4\pi E}{|m_1^2 - m_2^2|} \quad .$$

R is the distance travelled in time t, L_V is the vacuum oscillation length, and m_i is the mass of the ith mass eigenstate. For oscillation in a vacuum, Bachall & Frautschi[6] calculated the probability that a ν_e remains a ν_e in terms of the mixing angle by averaging over a spectrum of energies characteristic of the Sun. Their result,

$$|\langle\nu_e|\nu_e\rangle_t|_{avg}^2 \approx 1 - \frac{1}{2}\sin^2 2\theta_V \quad ,$$

suggests that large θ_V are necessary to explain the SNP in terms of vacuum oscillations. Although vacuum oscillations would reduce the neutrino flux, it is generally considered that the effect is not large enough to fully account for the SNP. The possibility of neutrino oscillations in matter (i.e. the Sun and Earth), on the other hand, leads to more interesting results and is the basis of the MSW Effect.

It is possible to express the vacuum oscillations in terms of a mass matrix deriving the same relationships as

above[7]. Wolfenstein has shown[8] that in matter, one most consider an additional factor in the mass matrix due to charged-current scattering (a result of the electrons in matter) which affects only ν_e. Wolfenstein described this difference in terms of an index of refraction related to forward scattering by the optical theorem. The expression for n is defined by the relation:

$$k(n - 1) = \frac{2\pi N_e}{k} f(0) = G_F N_e ,$$

with G_F the Fermi coupling constant, N_e the number density of electrons and k the wave vector of an incident particle. Physically, the index of refraction, which exists only for ν_e, causes a phase shift between ν_e and other flavored neutrinos with characteristic length,

$$l_0 = \frac{2\pi}{k(n-1)} = \frac{2.7 \times 10^9 cm}{\rho_e} ,$$

where ρ_e is the electron density in the medium. Therefore, there is a change in both the mixing angle (now written θ_M) and oscillation length L_M given by,

$$\sin^2 2\theta_M = \sin^2 2\theta_V \left(\frac{L_M}{L_V}\right)^2$$

$$L_M^2 = L_V^2 \left[1 + \left(\frac{L_V}{l_0}\right)^2 - 2\frac{L_V}{l_0}\cos 2\theta_V\right]^{-1} ,$$

for a medium with constant density.

Mikheev and Smirnov[9] took the result of Wolfenstein one step further, developing the effects of a medium with varying density, as in the case of the Sun. Their observation is that the value for $\sin^2 2\theta_M$ has a resonance value for small values of

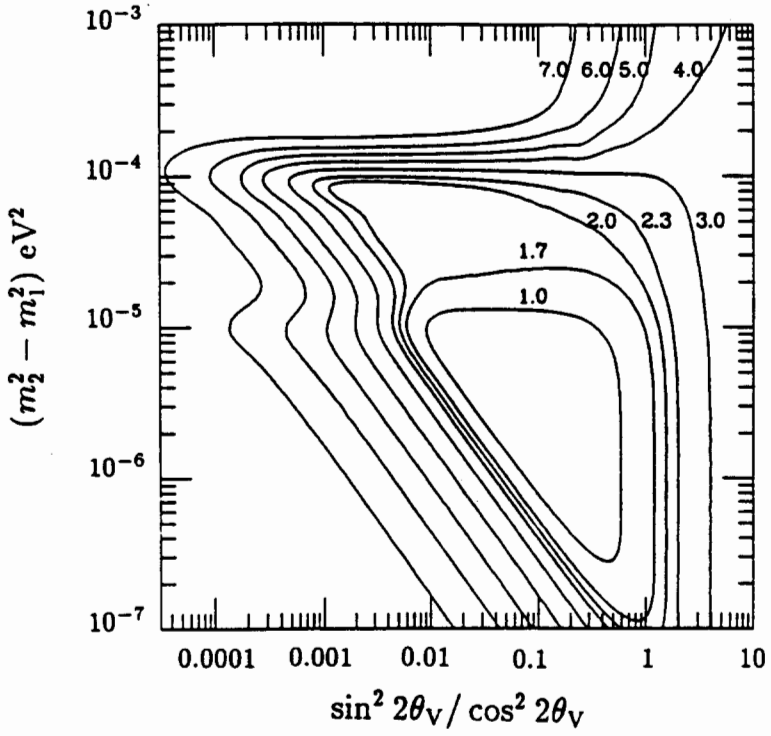

Figure 1.2: Example of an MS Diagram for the ^{37}Cl Experiment. Note that each contour is labeled by specific SNU values. [From BAH 89]

$\sin^2 2\theta_v$. They showed that this resonance occurs at $L_v/l_0 = \cos 2\theta_v$ and proposed a resonance density ρ_{res} that would greatly enhance the oscillations. Furthermore, they proposed that the resonance condition would be met in the varying density of the Sun and suggested this effect could explain the low observed neutrino rates.

Mikheev and Smirnov advanced 3 possible solutions to the varying density case: (1) the adiabatic solution (a slowly changing density) where neutrinos 'adjust' themselves to the varying density as they travel through the medium; (2) the non-adiabatic solution where the density changes too quickly for them to 'adjust'; and (3) the large mixing-angle solution where high values of θ_v are assumed. The 3 solutions have been applied to the data of the major neutrino experiments, constraining the MSW parameters - Δm^2 and $\sin^2\theta_v$ - to specific ranges. The non-adiabatic and large mixing-angle solutions are the most widely accepted while the adiabatic solution has been largely disputed by the work of Bethe[10] and others. The parameters are commonly graphed against one another along SNU contours in what is referred to as an MS diagram (Figure 1.2).

In addition to neutrino oscillations within the solar medium, there ought to be similar interactions within the Earth reversing the process to an extent. The effect, according to Mikheev and Smirnov[11], ought to be great enough that a diurnal variation could be studied experimentally offering positive proof of the MSW Theory. Several experiments, including Borexino, will provide evidence for the 'day-night effect' in the next few years. Observed or not, an

examination of the day-night effect will offer deep insight into the solution of the SNP.

E. Experiments

1. ^{37}Cl Experiment

Historically, the most important neutrino experiment has been the ^{37}Cl experiment designed and constructed by Davis et al.[12] in the 1960's. It is located 4850 ft below ground in the Homestake Mine, in Lead, South Dakota and was the first and only experiment of its kind for nearly two decades. Consequently, it provided the first evidence for the SNP, inspiring the next generation of neutrino detectors.

As discussed in Section 1.C, in approximately 1 of 5000 terminations of the pp chain, a reaction involving ^{8}B occurs resulting in a neutrino with a maximum energy of < 15 MeV. The event rate of the ^{37}Cl experiment is dominated by these ^{8}B neutrinos. They are energetic enough to excite ^{37}Cl to the analogous super-ground state ^{37}Ar via the charged-current reaction,

$$\nu_e + {}^{37}Cl \rightarrow e^- + {}^{37}Ar$$

with threshold energy E_{th} = .814 Mev. Since it is a charged-current reaction, the ^{37}Cl experiment is sensitive only to ν_e. Table 1.2 shows all of the calculated rates predicted from the standard solar model for a ^{37}Cl detector. The SNU is the unit of number of events per target atom per second, chosen equal to 10^{-36} s^{-1} for convenience. The ^{8}B neutrinos dominate the

Neutrino source	Cl (SNU)
pp	0.0
pep	0.2
^{7}Be	1.2
^{8}B	6.2
^{13}N	0.1
^{15}O	0.3
Total	8.0±3.0

Table 1.2: Capture rates predicted by the standard solar model for a ^{37}Cl detector.

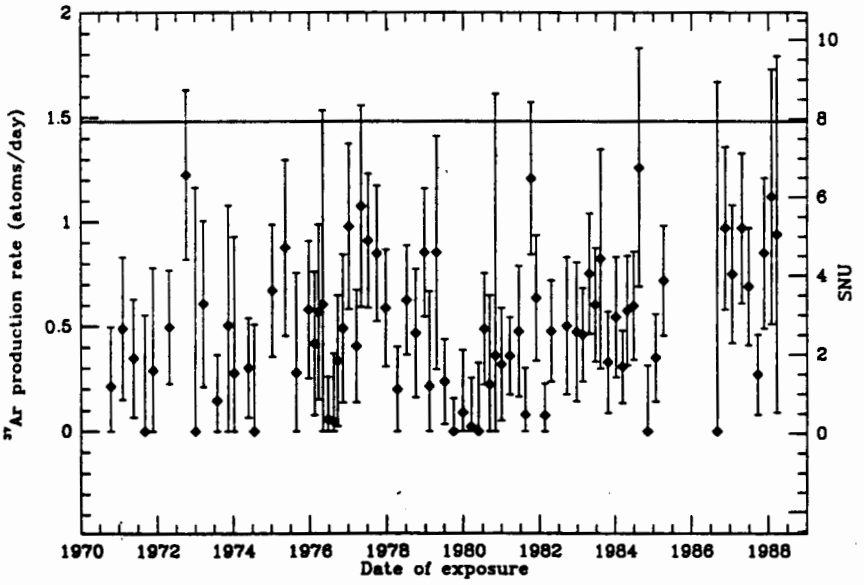

Figure 1.3: Observational results from the chlorine solar neutrino experiment. [From BAH 89]

capture rate because of their higher energy. Unfortunately, because ^{8}B neutrinos have such a low flux, the ^{37}Cl experiment has a very low count rate with correspondingly small statistical power.

By recovering and measuring the levels of ^{37}Ar from the ^{37}Cl detector for a given period of time, Davis et al. has experimentally measured the solar neutrino event rate[13]. Figure 1.2 shows the experimental results over the last 20 years for the chlorine solar neutrino experiment with comparison to the predicted 7.9 SNU predicted by the standard solar model without helium diffusion [The new predicted value given by the SSMH is 8.0 ± 3.0 SNU, a very slight difference.] It is very clear that the observed rate is well below the calculated rate. Even assuming that the entire observed flux is due to ^{8}B neutrinos (reducing the predicted rate to 6.2 SNU), there are still too few neutrinos. Clearly, then, either there is a discrepancy in the prediction of the ^{8}B flux for the standard solar model or the neutrinos are undergoing interactions between the center of the Sun and Earth (as in the MSW theory).

A few final comments on the ^{37}Cl experiment need to be addressed. Because it is an indirect counting experiment - one calculates the rate by measuring ^{37}Ar levels well after the reaction has taken place - there is a limited amount of information one can derive from the observed rates. For instance, it is impossible to differentiate between events (e.g. rates of ^{8}B and ^{7}Be neutrinos) which is essential in testing the SSM. Furthermore, it is difficult to determine

even if the events originate in the Sun or some other galactic phenomenon. The ^{37}Cl experiment was very successful in identifying the SNP, yet it does not have the experimental power to offer conclusive answers to its observations. Thus, the need for further neutrino experiments.

2. Kamiokande II

Unlike the ^{37}Cl experiment, Kamiokande II (KII) is a neutrino-electron scattering experiment which focuses solely on ^{8}B neutrinos. Neutrino-electron scattering is a neutral-current reaction sensitive to any neutrino flavor, yet most sensitive to ν_e. KII is a large water Cherenkov detector located 1 km beneath ground in the Kamohn metal mine in the Japanese Alps. The characteristics of a neutrino-electron scattering experiment offer KII several advantages over the ^{37}Cl experiment. In KII, the mean direction of the scattered electrons is measurable by observing their track paths. Therefore, one can verify the solar origin of neutrino events with statistical accuracy. Secondly, neutrino events are recorded almost instantaneously. This has obvious advantages over the delayed measurements of the ^{37}Cl experiment as well as other indirect counting experiments. Lastly, the threshold energy for KII is high enough such that only ^{8}B can be detected. Thus, by measuring the energy spectrum of the recoil electrons one can derive a fairly accurate measurement of both the energy distribution and flux of the ^{8}B neutrinos and match these results with the predicted values.

The first of the second generation neutrino experiments to record significant data, scientists looked toward KII to

possibly refute the results of the ^{37}Cl experiment. The data from Kamiokande II, however, match closely Davis's results. Totsuku[14] and Nakahatu[15] report a 90% confidence upper limit on the ^{8}B flux of

$$\phi(^{8}B) \leq .55\phi(^{8}B)_{SSM} \ .$$

An improved version of KII, KIII, has recently published data[16] supporting the results reported for KII. The combined result of KII and 395 days of the KIII experiment is .50±.07 of the SSMH prediction. As with the ^{37}Cl experiment, the initial phase of the Kamiokande II experiment was not powerful enough to test theories like the MSW effect, and it is unlikely that KIII could provide definitive evidence either. Another stage of KII - the Super Kamiokande - has also been proposed, and provided low enough backgrounds are achieved, it ought to offer the first evidence of diurnal and temporal variations in the neutrino rate.

3. Gallium Experiments

Two radiochemical experiments, GALLEX[17] and SAGE[18], are measuring the flux of the low-energy pp neutrinos from the pp reaction (reaction 1 of Table 1.1). Because this reaction is the main reaction responsible for energy generation in the Sun, the pp neutrinos have the highest neutrino flux and are very important to investigate. The reaction central to the experiments,

$$\nu_e + {}^{71}Ga \rightarrow {}^{71}Ge + e^- \ ,$$

has a very low threshold energy (E_{th} = .23MeV) and therefore

allows measurement of the low-energy pp neutrinos. The specific details of the experiments are similar to the radiochemical ^{37}Cl experiment and are not worth developing here.

The results from the gallium experiments, as with the ^{37}Cl and Kamiokande experiments, report pp neutrino fluxes far below those predicted from the SSMH. The GALLEX result[19] is 83±21 SNU and the most recent (also the highest) SAGE result[20] is 85^{+35}_{-38} SNU, both well below the predicted SSMH value 132^{+7}_{-6} SNU. Combining all of the SAGE and GALLEX data, the gallium result is 71±15 SNU (.54 of the SSMH prediction).

The gallium results are important, not only because they further underscore the SNP, but because they place further constraints on the core temperature of the Sun (T_c) and the MSW parameters. Bludman et al.[21] have recently evaluated the data for the ^{37}Cl, KII, gallium and KIII experiments. Their analysis suggests that:

> "..a purely astrophysical solution to the solar neutrino problem is strongly disfavored by the data: the Homestake [^{37}Cl] and Kamiokande data together are incompatible with any temperature in the Sun; the central values of both the SAGE and GALLEX results require a large reduction of T_c when they are fit to a cooler Sun."

In short, Bludman et al. refutes the possibility that an astrophysical explanation based on T_c could explain the SNP. Only a radical change in the SSM or new neutrino physics would suffice. Taking the MSW effect as the most likely solution to the SNP, they have fit (Figure 1.4) the experimental data to the MSW parameters, constraining them to two small regions: non-adiabatic oscillations with $\Delta m^2 = (.3\text{-}1.2)\times10^{-5}$ eV^2,

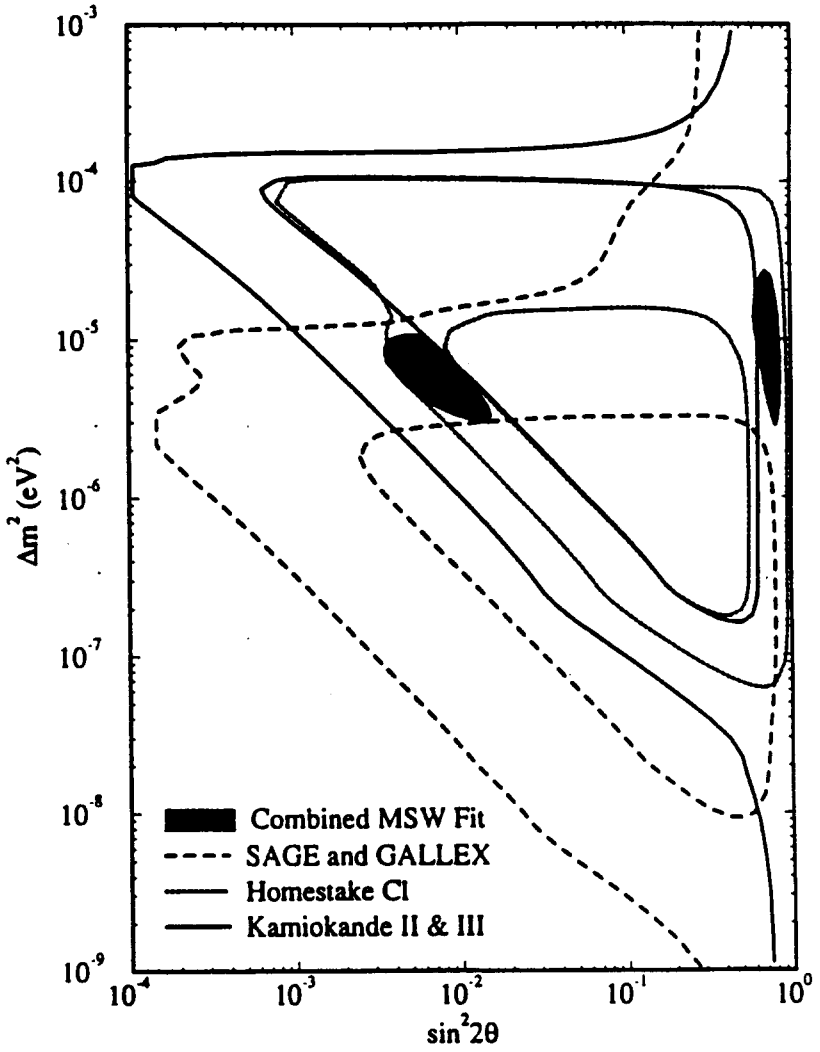

Figure 1.4: The allowed region for Homestake (dotted region), Kamiokande II + III (solid line), and the combined SAGE and GALLEX (dashed line results). The shaded region is the combined fit of the four experiments. [From BLUD 92, PREPRINT]

$\sin^2\theta_v$ = (.4-1.5)x10^{-2}, or large mixing-angle oscillations with Δm^2 = (.3-3)x10^{-5} eV^2, $\sin^2\theta_v$ = .6 - .9. Measurements on yet other neutrinos in the spectrum will reveal deeper insight into the analysis by Bludman. This is just one of the merits of the Borexino experiment, whose focus is upon the ^{7}Be neutrinos.

F. Borexino

The Borexino experiment offers a unique approach to the SNP. Initiated in 1987 by a group from AT&T, Drexel and MIT, Borexino has evolved into an experiment focused on examining low energy solar neutrinos (particularly ^{7}Be neutrinos) by analyzing the photons emitted from an organic scintillator. The success of Borexino's propositions lies in achieving very low backgrounds, both within the scintillator fluid and the supporting experimental apparatus. The experimental site is located in the underground laboratory of the Gran Sasso tunnel, 200km east of Rome, Italy.

Figure 1.5 is an artist's rendition of Borexino within the Gran Sasso tunnel. One notes an inner vessel filled with scintillator fluid being watched by a 3-D PMT array. Recoil electrons emitted via neutrino interactions are absorbed in the organic fluor releasing ~ 12,000 photons/Mev. These photons are detected 'immediately' by the PMT array allowing real-time analysis. Figure 1.6 and Table 1.3 show the predicted neutrino rates for the 100 ton fiducial volume of Borexino as a function of energy range. The .834 MeV ^{7}Be

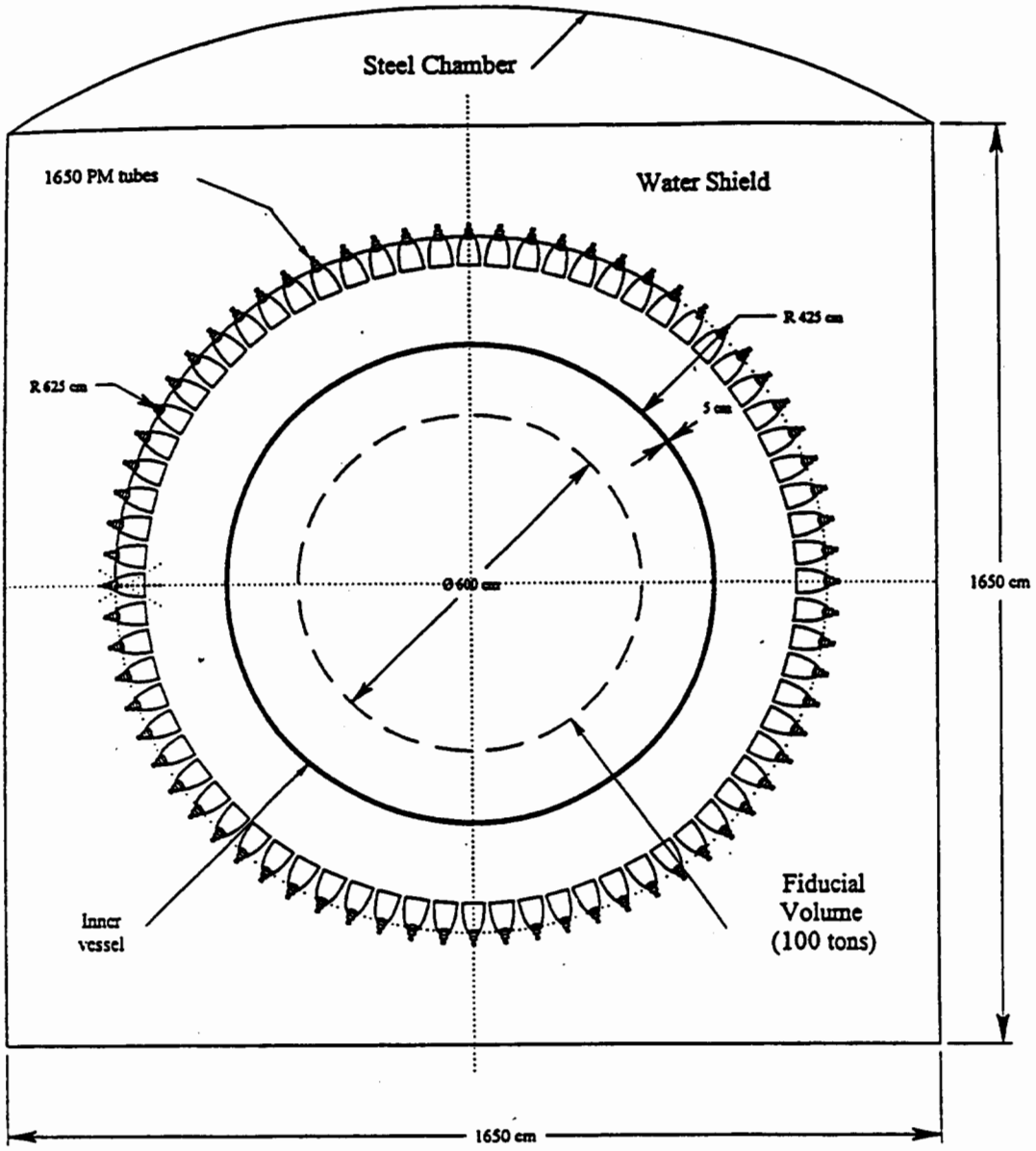

Figure 1.5: Schematic Vertical Cross-Section of the Borexino Detector. [From NSF 92]

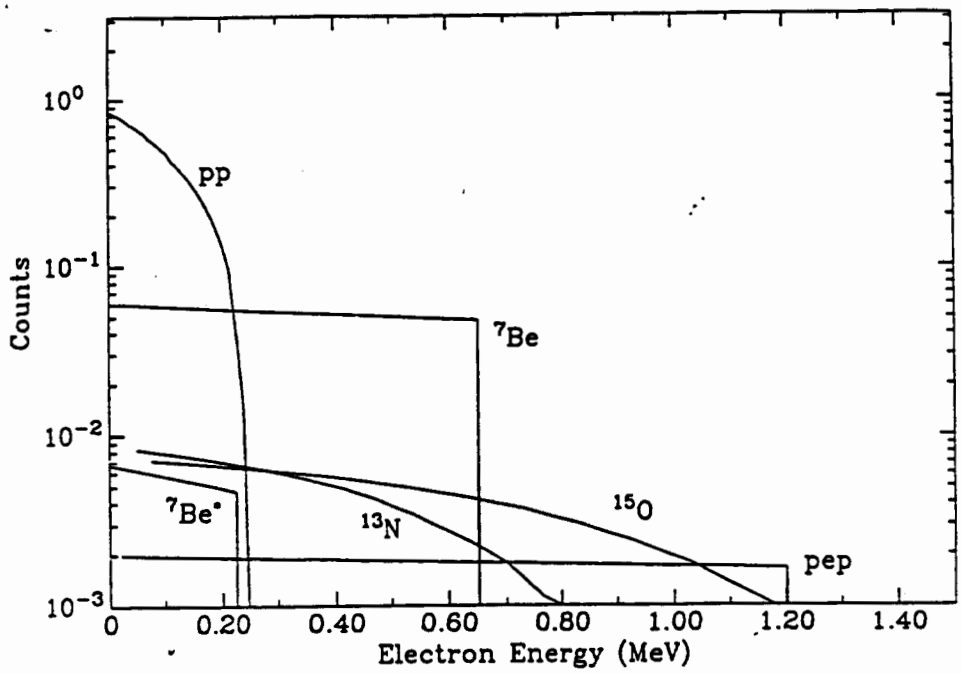

Figure 1.6: The Electron Spectra: Calculated electron spectra in Borexino due to the various solar neutrino groups predicted by the Standard Model. [From NSF 92]

Process	ν flux $10^9/cm^2s$	Events/day in 100 tons ≥0.25 MeV		0.25↔0.66 MeV	
		ν_e	ν_μ	ν_e	ν_μ
pp	6.000	0	0	0	0
^{7}Be	0.423	41.2	8.70	41.2	8.70
pep	0.014	3.5	0.78	1.5	0.30
^{13}N	0.061	4.7	0.90	4.0	0.80
^{15}O	0.052	8.0	1.50	4.4	0.90
^{8}B	0.0006	1.0		0.05	

Table 1.3: Counting rates expected in 100 ton fiducial volume of Borexino [From NSF 92]

neutrinos are expected to dominate the count rate of both electron and muon flavored neutrinos.

If the ultra-low backgrounds are achieved in Borexino the collaboration expects to record from 4,000 - 20,000 neutrino events yearly (10 - 50 events daily) dependent on the Δm^2 parameter associated with the MSW Theory. Borexino would be the first detector of its kind to detect so many neutrino events in real time. Such an experiment has considerable scientific merit. First, Borexino will verify predictions from the Kamiokande, ^{37}Cl, and gallium experiments whose neutrino flux measurements suggest a lower flux of ^{7}Be neutrinos than predicted. An accurate measurement of the ^{7}Be rate would clarify the interpretation of these experiments. In particular, if a high ^{7}Be flux (near SSMH prediction) is observed, it would almost certainly point to the need for a neutrino physics solution to explain the gallium experiments. A low rate of ^{7}Be neutrinos, meanwhile, may suggest a radical astrophysical solution (based on Bludman's results). At the same time, it would place constraints on the MSW parameters, possibly supporting Bludman's results.

Furthermore, Borexino's unique characteristics will lend itself to significant time variation observations. The deviation of the Earth's distance from the Sun ought to create a 7% difference in neutrino flux between aphelion and perihelion. Borexino will be able to measure this difference and thereby offer indisputable evidence of the solar origin of the neutrino events. Borexino will also provide for the investigation of the day/night effect. As discussed in

Section 1.D, the theory predicts a diurnal variation in the observed fluxes due to the oscillations within the Earth. Because ^{7}Be neutrinos are essentially monoenergetic source, the day-night effect ought to appear as a sharp variation the flux as a function of the solar angle. [Figure 1.7 is a graph of the neutrino rate variations for the Borexino project as predicted from the MSW Theory]. Borexino would be able to observe this effect provided the parameter Δm^2 is in the range $3x10^{-8}$ - 10^{-6} eV^2. If such a variation is observed, it would provide conclusive evidence of the MSW effect.

Lastly, and not to be understated, the scientific endeavors which lead to the final design, construction and operation of Borexino will display a variety of techniques pertinent to future low background experiments. It is this notion of low background which lies central to the success of Borexino. Before the construction of Borexino, however, the collaboration needs to demonstrate their ability to achieve very low levels of U, Th, ^{14}C within the experimental design (in particular the scintillator). The levels are so low that present measuring techniques cannot provide sufficient evidence. As such, the Borexino project has proposed the Counting Test Facility (CTF), a smaller scale experiment to investigate several important points of the Borexino background goals.

G. CTF

1. Goals

As outlined in NSF (Dec 92)[22] the CTF is being designed

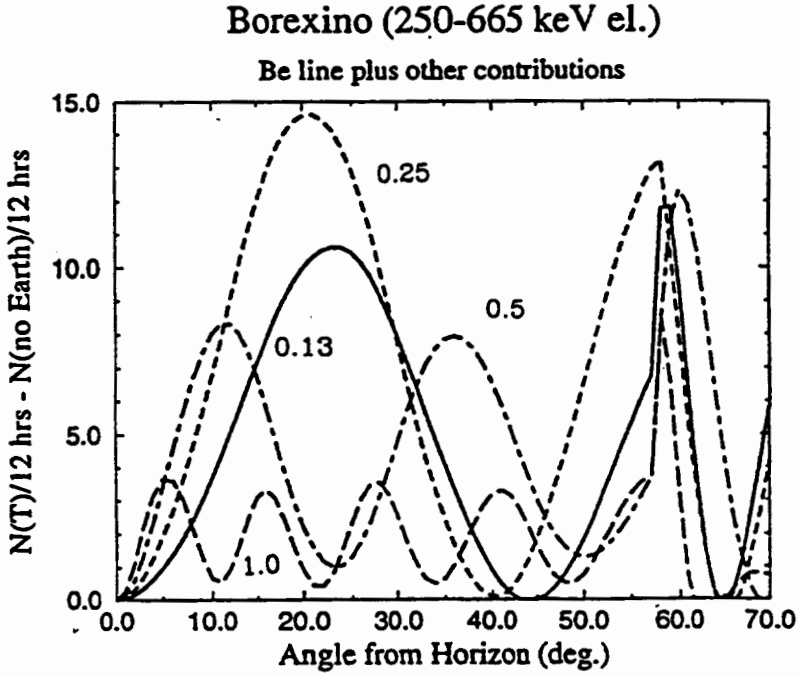

Figure 1.7: Possible Neutrino Rate Variatons: The effect of earth regeneration on the counting rate in Borexino in the Be neutrino signal window (including contributions from all neutrino sources listed in Table 1.3). Plotted are the differences of counts/12 hours for the no-earth and transmitted cases as a function of the solar angle at Gran Sasso (from A. J. Baltz priv. comm). The different curves represent different values of Δm^2 (in 10^{-6} eV^2) for a fixed value of $\sin^2 2\theta = .71$ in the vertical allowed region of the MSW map. [From NSF 92]

to achieve the following goals:

1. Measure the U and Th impurities in the scintillator with a sensitivity of 5×10^{-16} g/g, or better.
2. Measure the K impurity in the scintillator with sensitivity of 10^{-13} g/g.
3. Measure the $^{14}C/^{12}C$ ratio in the scintillator with a sensitivity of 10^{-18} g/g.
4. Ensure that the background of the surrounding apparatus (e.g. inner vessel) is within an acceptable level.
5. Test the effectiveness of proposed scintillator purification techniques.

Presently, there are two versions of the CTF being considered, the 2-arm CTF and the CTF Upgrade. Both designs address the above goals, yet with significantly different approaches.

2. 2-Arm CTF

Figure 1.8 is a schematic representation of the 2-Arm Counting Test Facility as recently proposed by the collaboration to NSF. The scintillator is contained within a 1m vessel (likely a nylon bag) supported by two long support structures - arms - which double as light guides to two PMT arrays. Because the PMT's and their bases have considerable radioimpurities associated with them, they are separated ~2.5m from the scintillator vessel. The efficiency of light detection for each arm has been calculated via ray tracing programs to be ~100 photoelectrons/MeV, a significant yield. Outside of the 2-Arms and PMT array is a 11m diameter x 10m high carbon steel tank filled with highly purified deionized water, which acts as a shield for the fiducial volume.

There are some important drawbacks to the 2-Arm design. First, the 2 Arms do not allow for accurate 3-dimensional positioning measurements. The absence of a 3-dimensional analysis will be very significant if the inner vessel and light

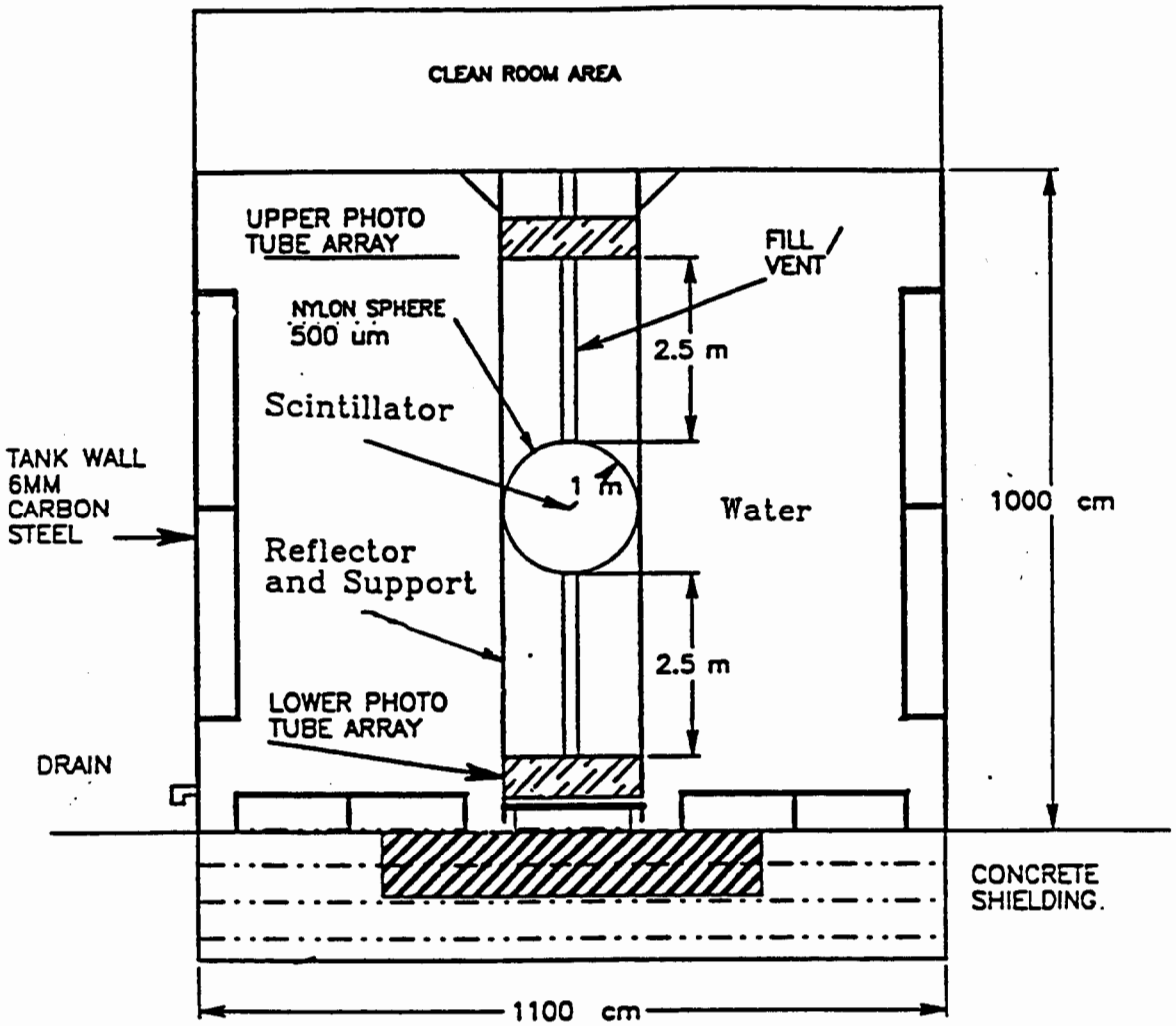

Figure 1.8: General layout of the 2-Arm CTF with a nylon inner vessel

guides have dominant surface radioactivity. It would be necessary to discriminate neutrino events which occur at the edge of the fiducial volume from background events caused by radioimpurities. A proposal has been made to use 6 Arms to provide for 3-dimensional analysis of the decay site, yet this approach would have difficult engineering and costly economic problems associated with it. With growing fears over the surface purity of the inner vessel, it becomes all the more important to design an economical 3-D CTF that would not create significant delays.

3. CTF Upgrade

Figure 1.9 is a schematic representation of one model of the CTF Upgrade. Note the similarities to the Borexino design. Within the carbon steel tank designed for the CTF is a 3.5m stainless steel tank similar to the main tank of Borexino. This tank houses an inner bag filled with scintillator fluid and fluor, suspended against all buoyant forces by nylon straps. The fiducial volume would be defined within this bag. Outside the inner vessel will be 1 of 3 highly purified fluids: deionized water, mineral oil, or xylene. This fluid serves as a shield for the fiducial volume much like the deionized water in the 2-Arm CTF design.

Also similar to the Borexino design and central to the success of this CTF Upgrade proposal is a 3-D PMT array with large light guides. As in the 2-Arm design, the PMT bases have high levels of radioimpurities associated with them and must be placed ~2.5m away from the fiducial volume. At this distance, in order to keep PMT and electronics costs within a respectable

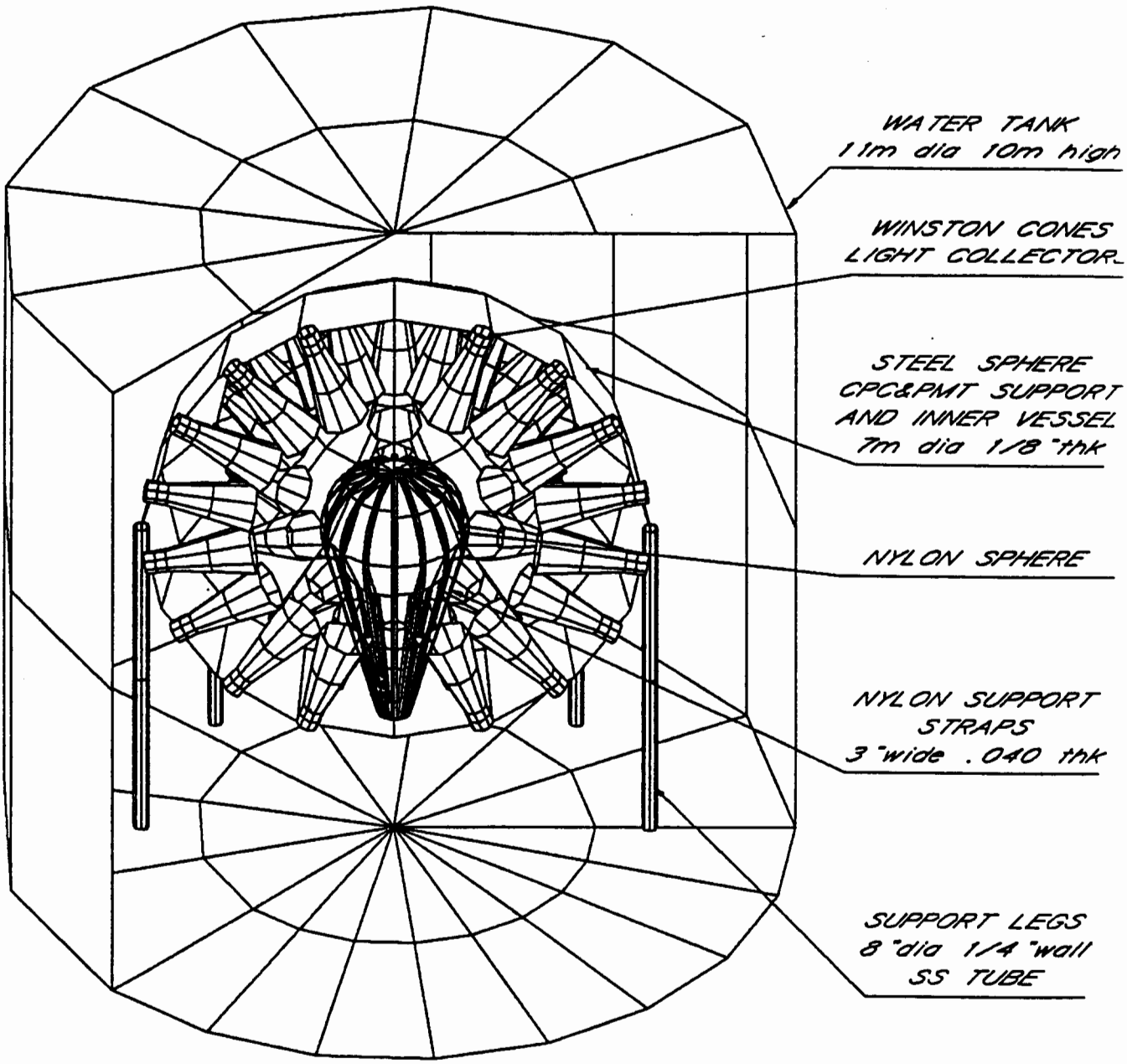

Figure 1.9: General layout of the CTF Upgrade.

level (~64 PMT's) and collect enough photons at the same time, large light cones (1 - 1.5m long) will need to be designed and fabricated. The cones must collect a high percentage of the light within a very specific geometry such that enough events are detected while accurately determining the position of the events. Not only does the CTF Upgrade mirror Borexino in design, it may actually be able to perform some of the same solar neutrino physics. With a photoelectron output of 200/MeV incident neutrino energy, estimates made by Calaprice (priv. comm.) and others suggest a count rate on the order of 10/day for the CTF Upgrade. This would be high enough to perform calculations on the day-night effect, an extremely impressive prospect. These results are very contingent on the background levels achieved in the CTF, and the success of the designed optics (i.e. light cones). The focus of my senior thesis has been the investigation of the design and fabrication of light collectors for the CTF Upgrade. The rest of this paper details this research and offers proposed solutions to the optics for the CTF Upgrade proposal.

Chapter 2 - Design

A. The Maximum Concentration of an Ideal Light Collector

A thorough discussion of the concentrator properties of an optical system must involve principles related to Liouville's Theorem. Extending Liouville's Theorem of Statistical Mechanics to geometric optics, I will derive the maximum concentration ratio of an ideal light concentrator with a flat exit aperture to be used with a planar PMT. In terms of Statistical Mechanics, Liouville's Theorem requires that a specific region of phase space always contain the same number of states, regardless of the coordinate system. Classically, there is a continuum of phase states and therefore one can only consider an *average* density of states which would be proportional to dxdydpdq (in 2D), with p and q the generalized momenta. In quantum mechanics the number of states in a finite element of volume is exact, and is represented by,

$$\frac{1}{h^3} d^3\mathbf{r}\, d^3\mathbf{p} \quad .$$

This quantity is considered an invariant of the coordinate system; the number of states within this differential phase space never changes. In geometric optics, too, it is possible to derive an invariant analogous to the phase space density dxdydpdq.

Applied to a perfect geometric optical system (no losses due to absorption) in equilibrium, the general approach is to

write p,q in terms of the optical direction cosines L,M → p=nL, q=nM. Therefore, the invariant defined by Liouville's Theorem is,

$$n^2\,dx\,dy\,dL\,dM \quad .$$

Winston points out[23] that this invariant can be interpreted as a differential element of radiance, where radiance is defined as the flux per unit solid angle per unit area projected perpendicular to the ray direction.

For a concentrator, we are interested in examining the radiance - the passage of this flux - from the entrance and exit apertures, where the two apertures are chosen, for convenience, to lie in the x-y plane. In the language of Liouville, we can write the radiance as the phase space acceptance (the difference between radiance and acceptance is only a matter of direction) of an aperture,

$$\Psi = \int\int\int\int n^2\,dx\,dy\,dL\,dM \quad .$$

Now assuming that the aperture has cylindrical symmetry, we can express the integral of the direction cosines as,

$$\int\int dL\,dM = \int_0^{\frac{\pi}{2}} \sin(2\theta)\,d\theta \quad ,$$

where θ is the angle of incidence on the aperture as in Figure 2.1 - expected to range from 0 to $\pi/2$.

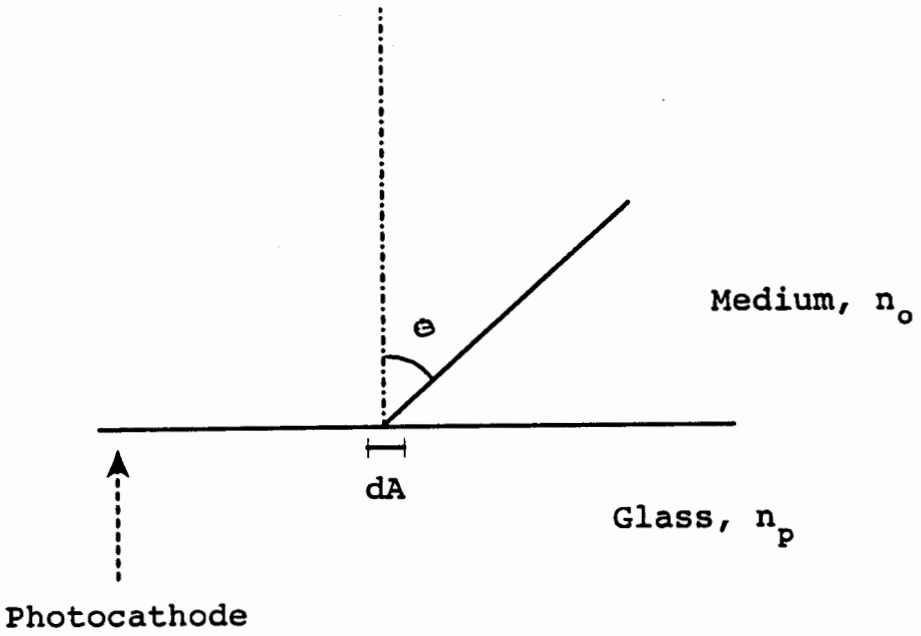

Figure 2.1: Phase space acceptance of an area element dA of the photocathode. [From Moor 92]

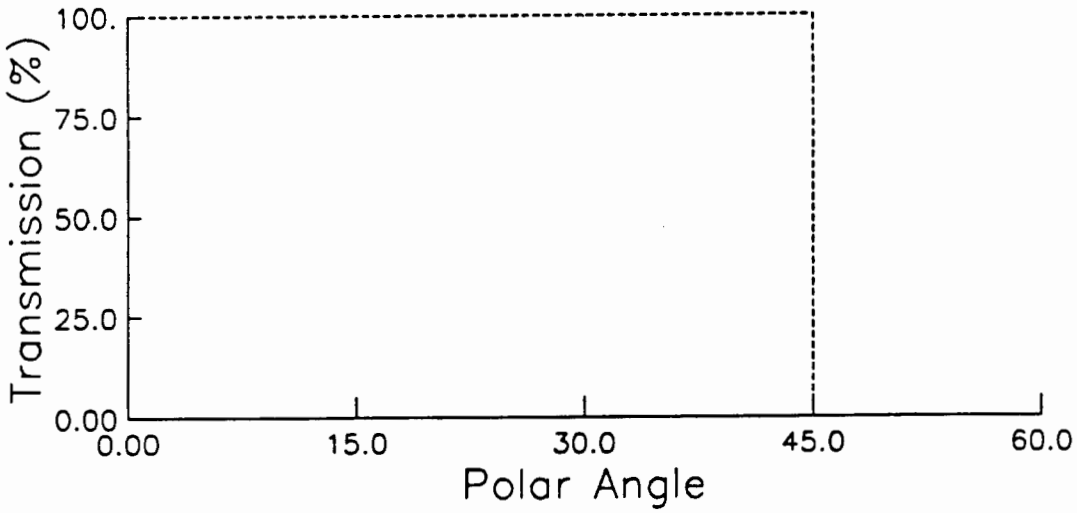

Figure 2.2: Ideal transmission-angle curve of a concentrator with a limiting angle θ_i = 45°. The probability of transmission to the photocathode (at an angle of incidence θ_2, $T(\theta, \mathbf{r}_{ent})$, is averaged over the entrance aperture of the concentrator. [From Moor 92]

Rewriting dxdy = dA, we arrive at,

$$\Psi_{aper} = \int\int_0^{\frac{\pi}{2}} n^2 \sin(2\theta)\, d\theta\, dA \quad .$$

In order to apply this to an exit aperture connected to a planar PMT, we introduce a factor $E_{pmt}(\theta, \mathbf{r}_{pmt})$, the efficiency that the PMT registers an event as a function of the incident angle θ and the position on the PMT. For an ideal PMT, E_{pmt} is considered constant over all angles and over the entire surface of the PMT. Introducing E_{pmt} and integrating with n_p the refractive index of the PMT (usually glass), the phase space acceptance of the exit aperture is,

$$\Psi_{exit} = \int\int_0^{\frac{\pi}{2}} n_p^2\, E_{pmt}(\theta, \mathbf{r}_{pmt})\, \sin(2\theta)\, d\theta\, dA_{exit} \qquad (1)$$

$$\Psi_{exit} = [\int n_p^2\, E_{pmt}\, dA_{exit}]\, [-\frac{1}{2}\cos(2\theta)]_0^{\frac{\pi}{2}}$$

$$= n_p^2\, E_{pmt}\, A_{exit} \quad . \qquad (2)$$

This quantity is invariant under Liouville's Theorem, and therefore to determine the concentration ratio of a light collector (ideally A_{ent}/A_{exit}), we examine the phase space acceptance (invariant) of the entrance aperture with the intention of equating the two.

The approach for the entrance aperture is essentially the same. The only difference is that we must include a factor $T(\theta, \mathbf{r}_{ent})$, the probability for transmission to the exit aperture of a ray that intersects the entrance aperture at $\mathbf{r}_{ent}$, with incident angle θ. For an ideal concentrator, all

rays with incident angle $\leq \theta_i$ are transmitted independent of $\mathbf{r}_{ent}$ while all rays with incident angle $\geq \theta_i$ are rejected. Therefore the transmission-angle curve is flat with 100% transmission out to θ_i and an immediate drop to 0% at θ_i (Figure 2.2). The phase space acceptance of this aperture is defined by the double integral,

$$\Psi_{ent} = \iint_0^{\frac{\pi}{2}} n_o^2 \, T(\theta, \mathbf{r}_{ent}) \, E_{pmt} \sin(2\theta) \, d\theta \, d\mathbf{A}_{ent} \, , \quad (3)$$

with n_o the refractive index of the medium within the concentrator, and E_{pmt} defined as above. Evaluating the integral for an ideal transmission-angle curve with cutoff at θ_i, one finds

$$\Psi_{ent} = n_o^2 \, E_{pmt} \, A_{ent} \sin^2\theta_i \, . \quad (4)$$

Noting that the phase space acceptance of the two apertures is invariant, $\Psi_{exit} = \Psi_{ent}$ we find:

$$\frac{A_{ent}}{A_{exit}} = \frac{n_p^2}{n_o^2 \sin^2\theta_i} \, . \quad (5)$$

As the ideal concentration ratio of a light collector is defined as A_{ent}/A_{exit}, equation (5) is the maximum theoretical concentrator ratio (C_{3D}) for a 3D light concentrator. It is also trivial to show that in 2D, where a ray can vary in one dimension only (the invariant is ndydL), $C_{2D} = n_p/(n_o \sin \theta_i)$. The most important and difficult factor in designing a light collector is to achieve one with the ideal transmission-angle curve. In the next section, I detail the efforts pioneered by Winston[24] to design such a light collector.

B. The Edge-Ray Principle

The key to designing an ideal light concentrator is to devise one with an ideal transmission-angle curve for an extreme angle θ_i. The following treatment is done initially in 2-dimensions because it is easier and because the results carry directly to 3D. Intuitively, one might first guess that a conic collector - essentially the reverse of a megaphone - would have ideal transmission. The conic collector is defined by an extreme ray θ_i off the axis of symmetry as shown in Figure 2.3. The angle γ is given by $2\gamma = (\pi/2) - \theta_i$, and the length of the cone is defined by the intersection of an extreme ray which originates at one edge of the exit aperture with the opposite side of the cone. Although most rays incident on the cone will be transmitted to the exit aperture after one reflection, Figure 2.3 illustrates that for rays with incident equal to θ_i, many will be turned out of the cone after multiple reflections. One can also show that many rays will be rejected with incident angle $\leq \theta_i$. Therefore, the transmission-angle curve is not ideal for the conic collector; it is not an ideal light concentrator.

The tremendous insight of Winston is the application of a theorem he terms the Edge-Ray Principle. According to the principle, in order to achieve ideal concentration it is necessary that all extreme rays (those with angle θ_i) be reflected to the edge of the exit aperture after one reflection. With this principle satisfied, all rays (in 2D)

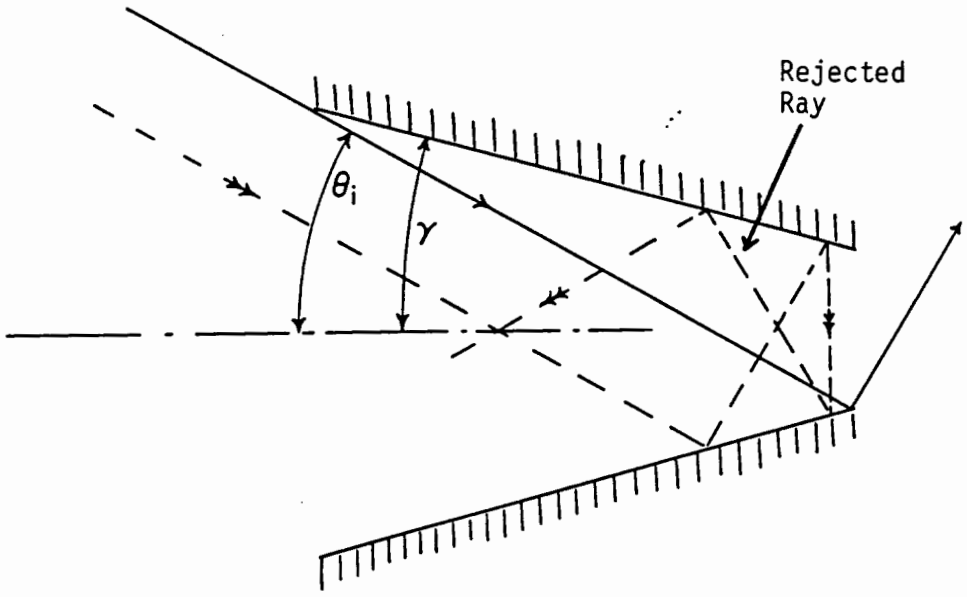

Figure 2.3: The conic concentrator. The dotted line represents a ray that has been rejected.

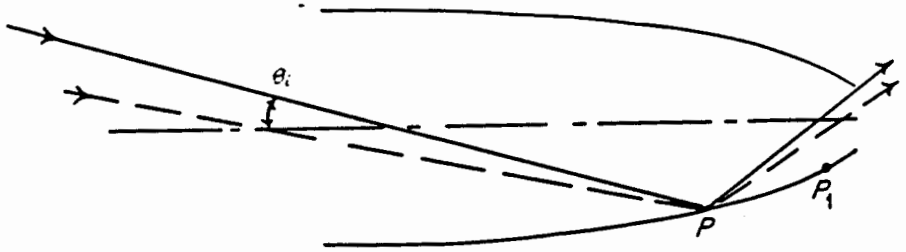

Figure 2.4: Application of the edge ray principle with extreme angle θ_i. Note that the ray with incident angle less than θ_i passes through the exit aperture.

with incident angle less than θ_i are transmitted somewhere in the exit aperture (Figure 2.4). Furthermore, all rays (in 2D) with angle greater than θ_i will reflect several times and be rejected. As such, an ideal transmission-angle curve is achieved. Applying this principle in 2D, we immediately derive the Winston Compound Parabolic Concentrator (CPC). These cones were initially used as light collectors in Cherenkov counters by Hinterberger and Winston[25]. More recently, they have been applied to solar energy collection because of their nearly ideal performance.

Figure 2.5 demonstrates the construction of the CPC (in 2D) through the application of the Edge-Ray Principle. Given a planar exit aperture and a source at infinite distance with maximum input angle θ_i (two conditions in contradiction with the CTF Upgrade), we require that all parallel extreme rays reflect off of the collector to point B' on the edge of the exit aperture. This principle coincides with the definition of a parabola that has a focal point at B' and an axis of symmetry defined parallel to the extreme rays. The curve running from the entrance aperture to the exit aperture is a 'slice' of this parabola. One repeats the process along the other side with B as the new focal point and for parallel extreme rays with angle θ_i off the other side of the symmetry axis. This defines another parabola, the reflection of the first through an axis of symmetry defined as the bisector of the entrance and exit apertures. In 3D, this axis of symmetry is the axis of revolution.

Similar to the conic collector, the length of the CPC is

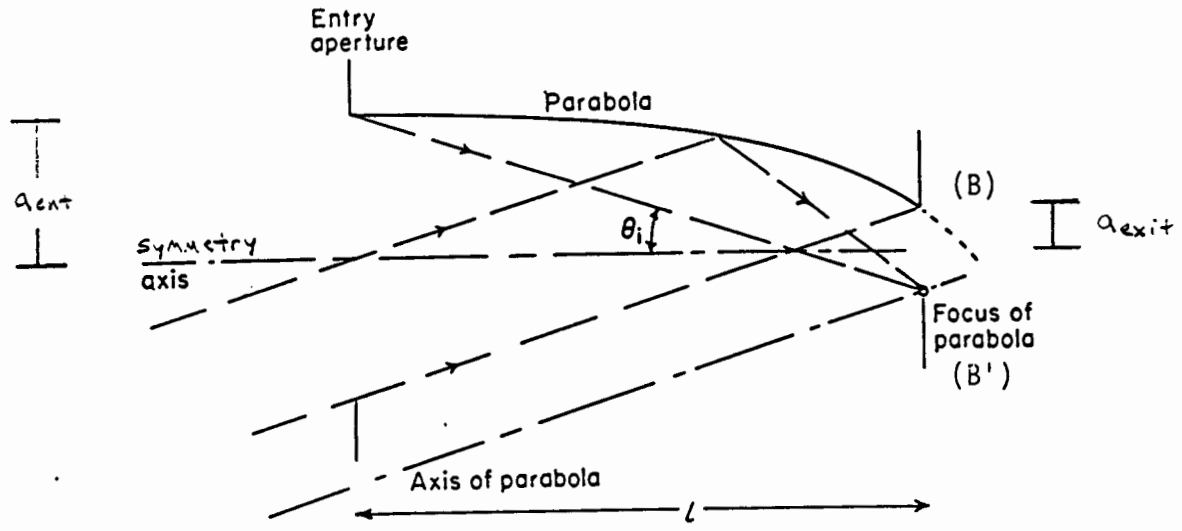

Figure 2.5: Construction of CPC profile from the edge-ray principle. [From Win 89]

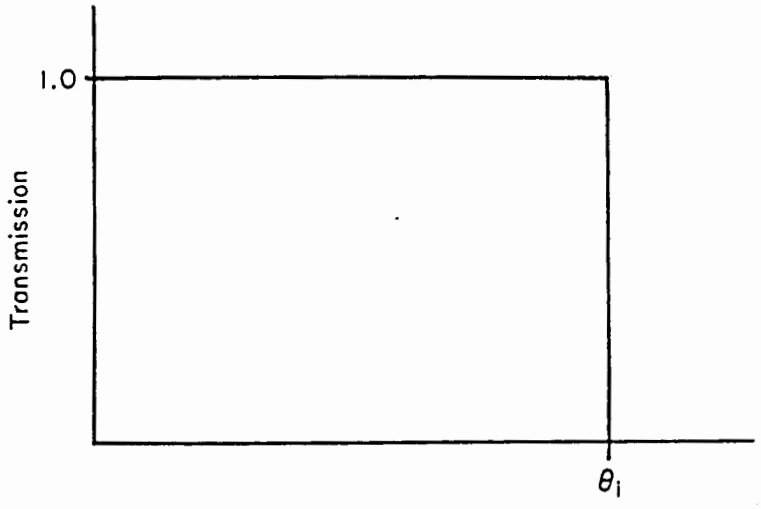

Figure 2.6: The transmission-angle curve for a 2D CPC. [From Win 89]

defined by the extreme ray which originates at the edge of the exit aperture. It is a relatively simple matter (see Appendix D in Win 1989) to show that $L = a'(1+\sin \theta_i)\cos \theta_i / \sin^2 \theta_i$ with a' the exit aperture radius. From Figure 2.5, one sees directly that the length of the cone can be described in terms of the radii of the entrance and exit apertures by $L = (a + a')\cot \theta_i$. Equating these two expressions, we find a relation for the radii of the two apertures, $a_{ent} = a'_{exit}/\sin(\theta_i)$. Because we applied the Edge-Ray Principle, the CPC has an ideal 2D transmission-angle curve. It accepts all rays with incident angle $\leq \theta_i$ and rejects those with incident angle $\geq \theta_i$. Figure 2.6 is the 2D transmission-angle curve calculated by Winston for the CPC, clearly ideal. Consequently, $C_{2D} = a/a' = 1/\sin[\theta_i]$, which is the maximum concentration ratio derived in Section 2.A with $n_p = n_o$. Thus, the 2D CPC is an ideal light collector.

A comparison in 3D further highlights the differences between the conic and CPC cone. Figures 2.7 and 2.8 are the transmission-angle curves for the conic and CPC cone respectively. Compare the curve for a $\theta_{max} = 10°$ conic collector with the $\theta_{max} = 10°$ curve for the CPC cone. Clearly, the CPC cone is flatter much further out to 10° with a much sharper drop after 10° than the conic collector. Therefore, the total transmission is higher, and in terms of defining a specific phase space (e.g. fiducial volume), the definition is much sharper. Note that the 3D CPC does not have an ideal transmission-angle curve (unlike the 2D CPC) because of skew rays which undergo multiple reflections and bounce back out of

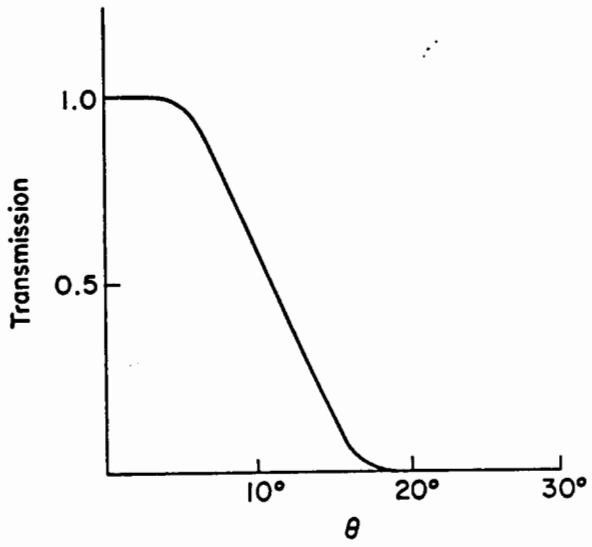

Figure 2.7: Transmission-angle curve for a conic collector; θ_i = 10°. [From Win 89]

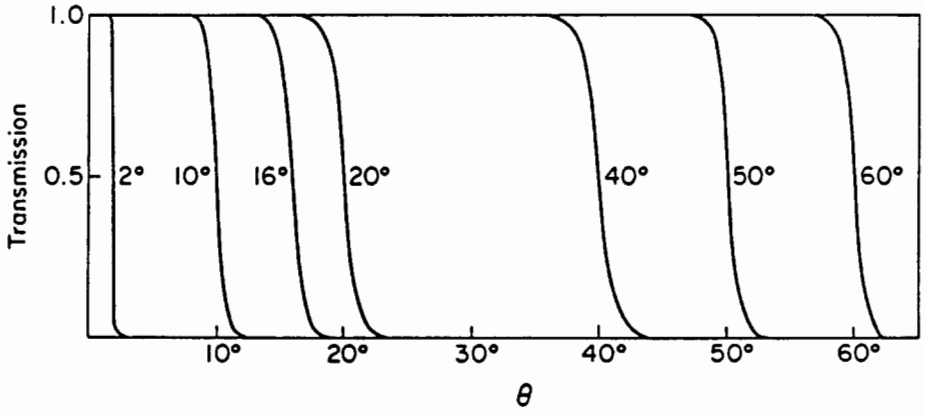

Figure 2.8: Transmission-angle curves for 3D CPC's with θ_i from 2° to 60°. [From Win 89]

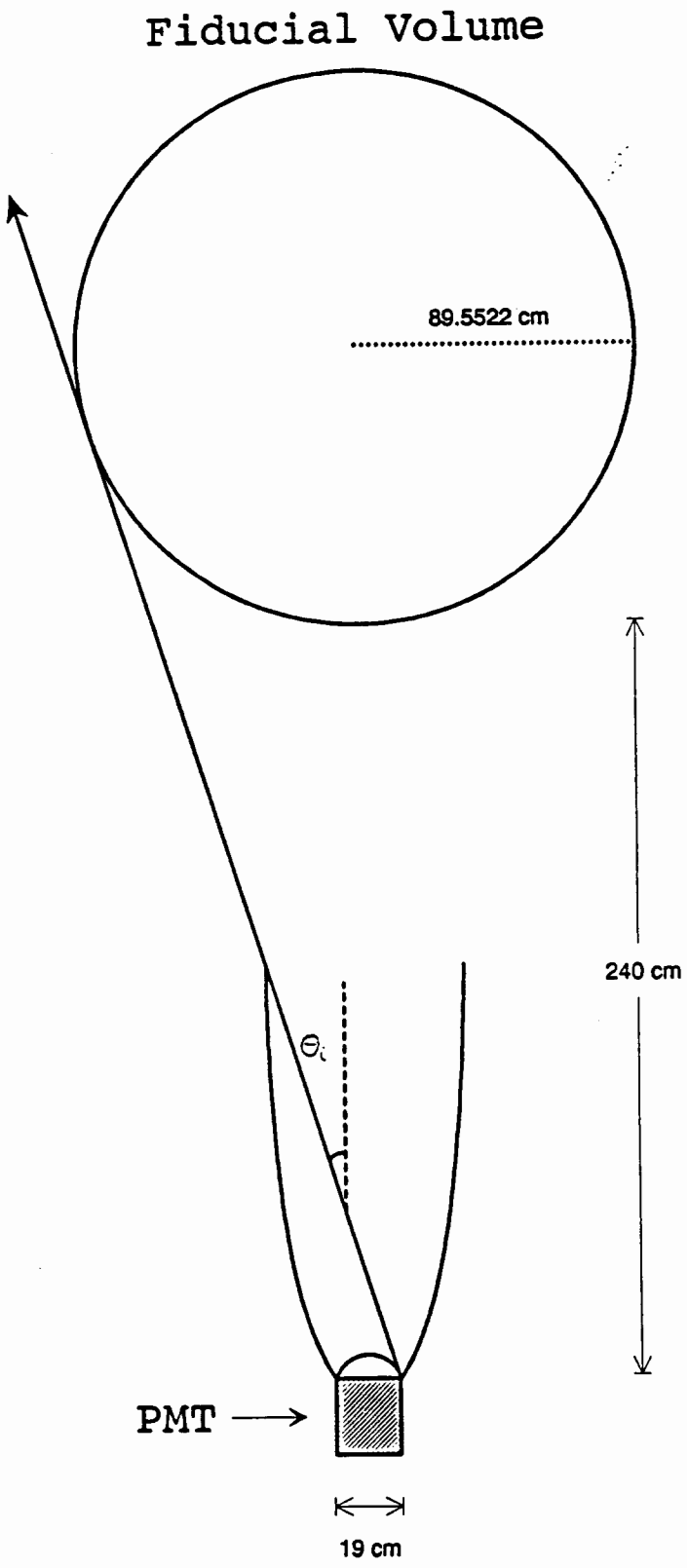

Figure 2.9: Illustration of the CPC cone in the CTF Upgrade design. θ_i is defined by the ray drawn tangent to the fiducial volume from the edge of the exit aperture. Note, the curve representing the PMT is not a semi-circle with radius 9.5cm, but an ~120° arc with radius 11cm.

the light collector. As Winston discusses[26], in extending the CPC to 3D strictly by a rotation about an axis of symmetry, we have no further degrees of freedom and therefore no way to deal with these skew rays. This leads directly to the non-ideal performance. Steps can be taken to optimize the 3D CPC to skew rays, yet the effects are minimal.

In conclusion, I looked initially to the CPC cone as the optimal concentrator for the CTF Upgrade because of the shape of its transmission-angle curve. I took θ_i = 17.3881° defined by a ray from the edge of the exit aperture tangent to the fiducial volume as in Figure 2.9. I expected that an array of CPC's would accurately define a fiducial volume, recording all events that occurred within and rejecting all events outside by the observed pattern of lit and unlit PMT's. Upon further analysis, however, it is apparent that the CPC is not ideal for the CTF Upgrade and that a slightly different cone had to be designed.

C. Non-Planar Photocathode

The Winston Compound Parabolic Concentrator is ideal in 2D for a planar exit aperture and nearly ideal in 3D. In the CTF Upgrade, as well as in many other detectors, a curved PMT window is used because it is very expensive to fabricate a large, flat PMT. With a portion of the PMT protruding into the light cone, the characteristics of the CPC transmission-angle curve are affected. The PMT's proposed for the CTF Upgrade are spherical with an 11 cm radius. Because of fixturing needed to hold the PMT to the light cone, the

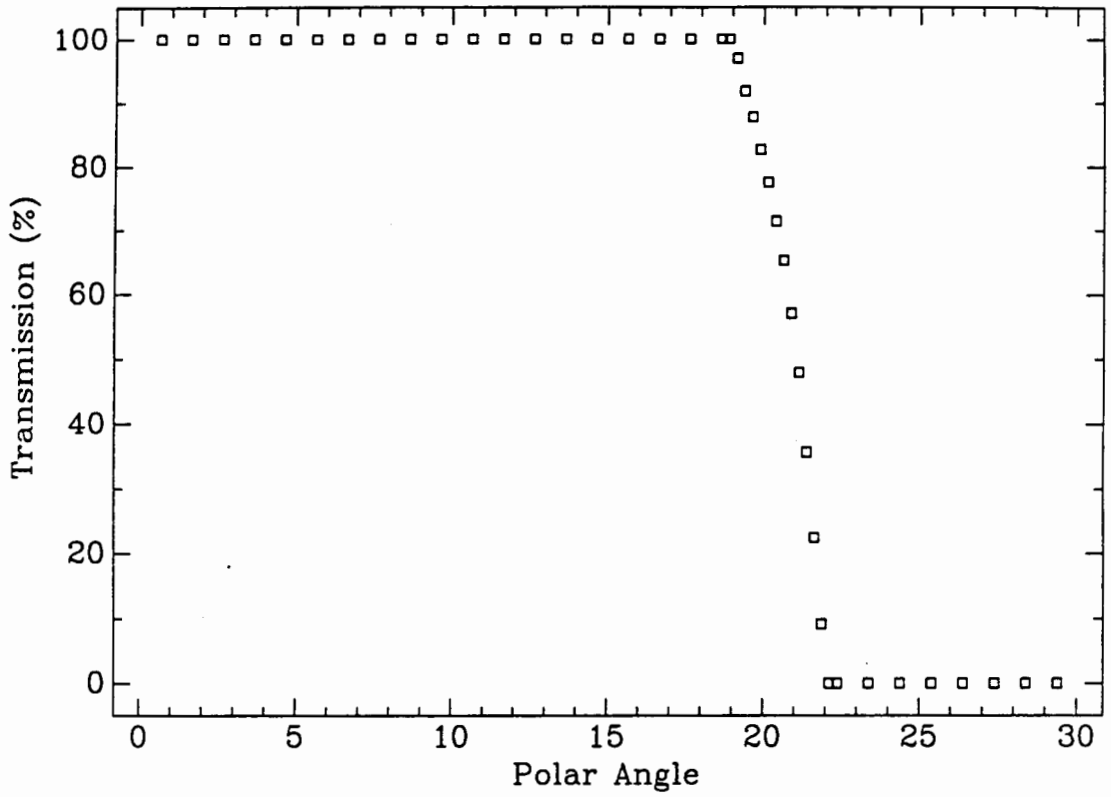

Figure 2.10: Transmission-angle curve for a θ_i = 17.39° 2D CPC with a spherical PMT.

maximum radius for the exit aperture is 9.5 cm.

Figure 2.10 shows the adjusted 2D transmission-angle curve for the experimental design of the CTF Upgrade. The figure was created with a 2D ray trace simulation - program ta2cp.c from Appendix 2.C - that assumed an ideal PMT (i.e. it would detect any photon that intersected its surface). Although the CPC is designed with θ_i = 17.39°, incident rays out to 19° are detected with 100% transmission. Furthermore, the curve has a broader cutoff about θ_i and, therefore, is no longer ideal. Clearly, the profile must be altered so that the new exit aperture, defined by the outer surface of the PMT, can be fully accounted for. This, in turn, alters the maximum concentrator ratio.

If the entire phase space acceptance of the spherical photocathode can be utilized, the concentration is $C' = A_{ent}/A_{pmt}$ and no longer A_{ent}/A_{exit} where A_{pmt} is the surface area of the photocathode that projects into the light collector. Therefore, the concentration is increased by a factor of A_{pmt}/A_{exit}, which is ~4/3 for the CTF's photomultiplier tubes. To achieve this concentrator ratio, it is necessary to apply to the Edge-Ray Principle again, yet with the edge given by the entire surface of the PMT. Moorhead, in his doctoral thesis[27], has developed these cones extensively and called them Compound Tangential Concentrators (CTC's). The CTC's will be implemented in another proposed neutrino experiment, the SNO Experiment. It turns out, however, that these cones, as well, are not fully applicable to the CTF Upgrade.

D. String Method Cones

The other aspect that I did not consider initially in designing light collectors for the CTF Upgrade is the fact that the fiducial volume is not an 'infinite' distance from the cones. Since the edge of the fiducial volume is expected to be only ≈1m from the light cones, there will be a ±10° spread of incident rays at the entrance aperture. Therefore some rays will have incident angles as high as 27°, much greater than $\theta_i \approx 17°$ as defined by the CTF Upgrade design (Figure 2.9). The rays with incident angle > 17° (assuming a planar PMT; > 19° for the spherical PMT) will be rejected by the CPC cone. Therefore, many events will occur within the fiducial volume that would not be completely detected (registered by all cones) by a spherical array of these CPC cones. With the source at a finite distance, it is instructional to examine the transmission levels of the CPC cone as one moves away from the center of the fiducial volume (a transmission-radial curve). To calculate this curve, we consider fiducial 'shells' with radii ranging from 0 to the maximum fiducial radius.

Because the angular 'spread' increases with radius, in considering shells of increasing fiducial radius, one expects to observe fewer and fewer events as the maximum fiducial radius is approached. Figure 2.11 is a 2D transmission-radial curve for a 17.39° CPC cone designed for a maximum fiducial volume of 89.55 cm as before. (The plot was generated using the 2D ray trace program tr2d.c detailed in Appendix 2.B). The 2D data demonstrates conclusively that this cone is not

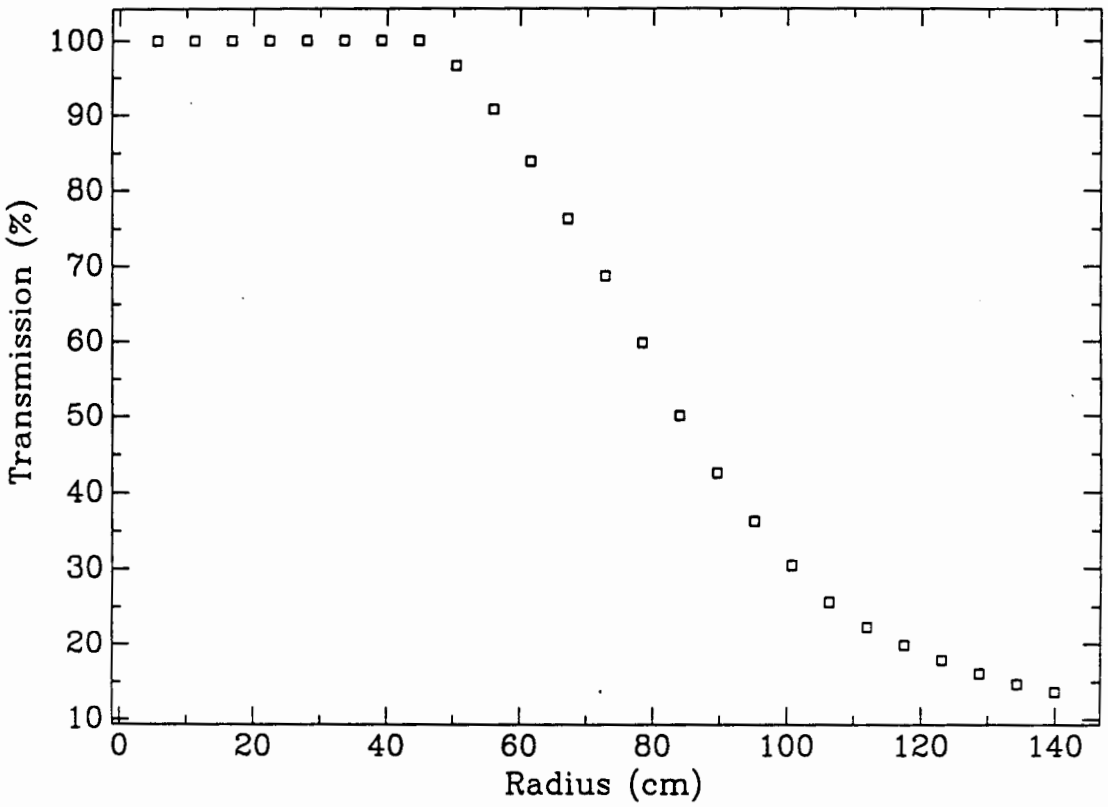

Figure 2.11: Transmission-radial curve for a 2D CPC cone designed to look at an 89.55cm FV (θ_i = 17.39°).

ideal for the CTF Upgrade design. In particular, the cone does not have 100% transmission out to the edge of the defined fiducial volume implying the possible loss of events that occur within the fiducial volume. Therefore, the CPC design fails for the CTF Upgrade and another profile was actively pursued. [Note: a curved PMT is assumed in this simulation which does affect the results as the CPC is designed for a planar exit aperture. The effect, however, only improves the CPC's radial performance and is not to blame for the non-ideal transmission-radial curve.]

An ideal transmission-radial curve is highly desired for the CTF Upgrade for several reasons. First, we want to be able to detect with a high degree of confidence <u>all</u> events which occur within the fiducial volume. Secondly, a sharp transition at the edge of the fiducial volume allows for very high resolution in differentiating between events outside the fiducial volume from those within. This second point is easily as important as the first for the CTF Upgrade because it is directly related to the reduction of background levels. As such, it is developed in great detail below.

The solution to account for both the near fiducial volume and the spherical PMT is the construction of the 'String Method' (SM) Cone. Instead of assuming an infinite source, we include the varying finite distance of the fiducial volume within the design. The general method is similar to the gartner's method of drawing an ellipse. If the PMT and source were planar, the treatment would be exactly as that developed in Chapter 5.4 of Winston's text[28]. There, he applies the

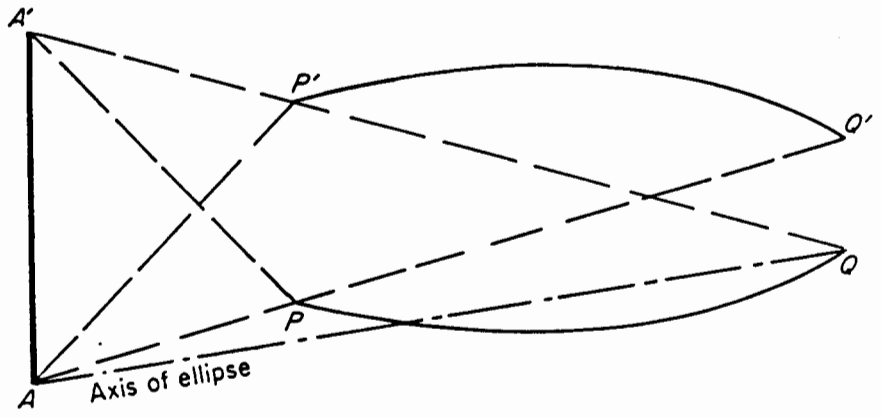

Figure 2.12: Concentrator for a planar source at a finite distance from a planar PMT. The foci of the lower ellipse are A' and Q' while those for the upper ellipse are A,Q. [From Win 89]

Edge-Ray Principle to a planar source located at a finite distance from the light concentrator. He demonstrates that the profile would be 'slices' of two ellipses (Figure 2.12) where the foci of these ellipses are defined by the edge of the exit aperture and the edge of the source. [This ought to sound very familiar to the description given on the CPC cones, except replacing the parabolas by ellipses]. The curvature of the PMT and fiducial volume, however, forces a slightly different approach.

In essence, we look for an array of points that are the set of intersections between rays tangent to the fiducial volume and those tangent to the surface of the PMT. [As with the CPC, the treatment is done first in 2D and then applied to 3D by rotating the 2D curve about its axis of symmetry.] With much credit to Martin Moorhead (priv. comm.), I have written a program which finds the profile of the SM cone given a set of input criteria (program simult_dat of Appendix 1). The method is fairly complicated and therefore I describe it in detail.

We take the first point on the profile to be the point of intersection between the PMT and the exit aperture of the light cone. The second point is located a variable distance - denoted the 'step size' - normal from the PMT surface. From this new point, we draw a ray tangent to the PMT and the fiducial volume. One then calculates the normal $\hat{n}$ at this point whereby the law of reflection would 'connect' the two rays with one reflection (Figure 2.13). The next point is taken a step size away from this one in the direction perpendicular to the normal $\hat{n}$.

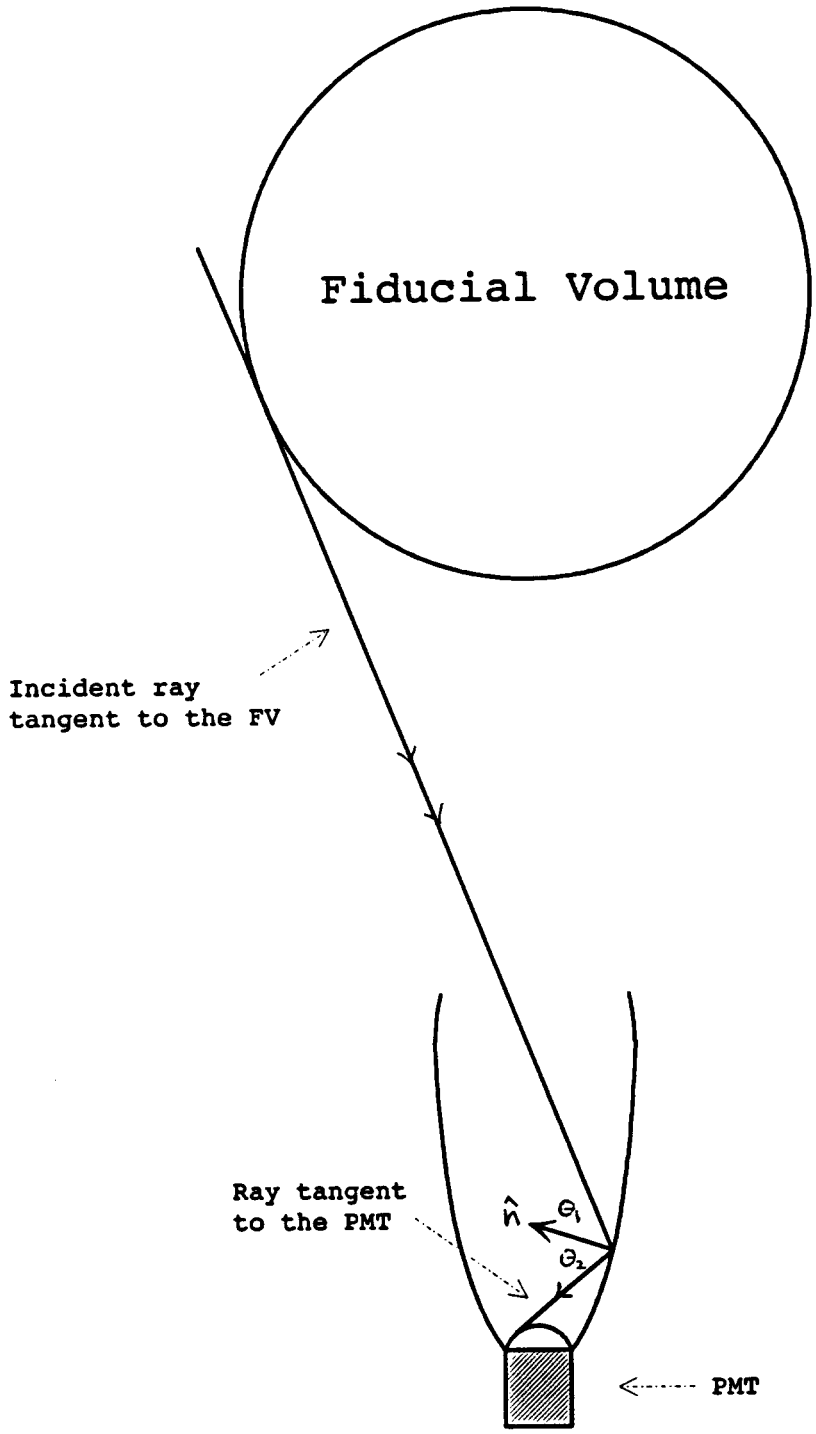

Figure 2.13: 2D SM Cone: Demonstrating the edge-ray principle for a non-planar PMT. n is the normal vector defined such that $\theta_1 = \theta_2$.

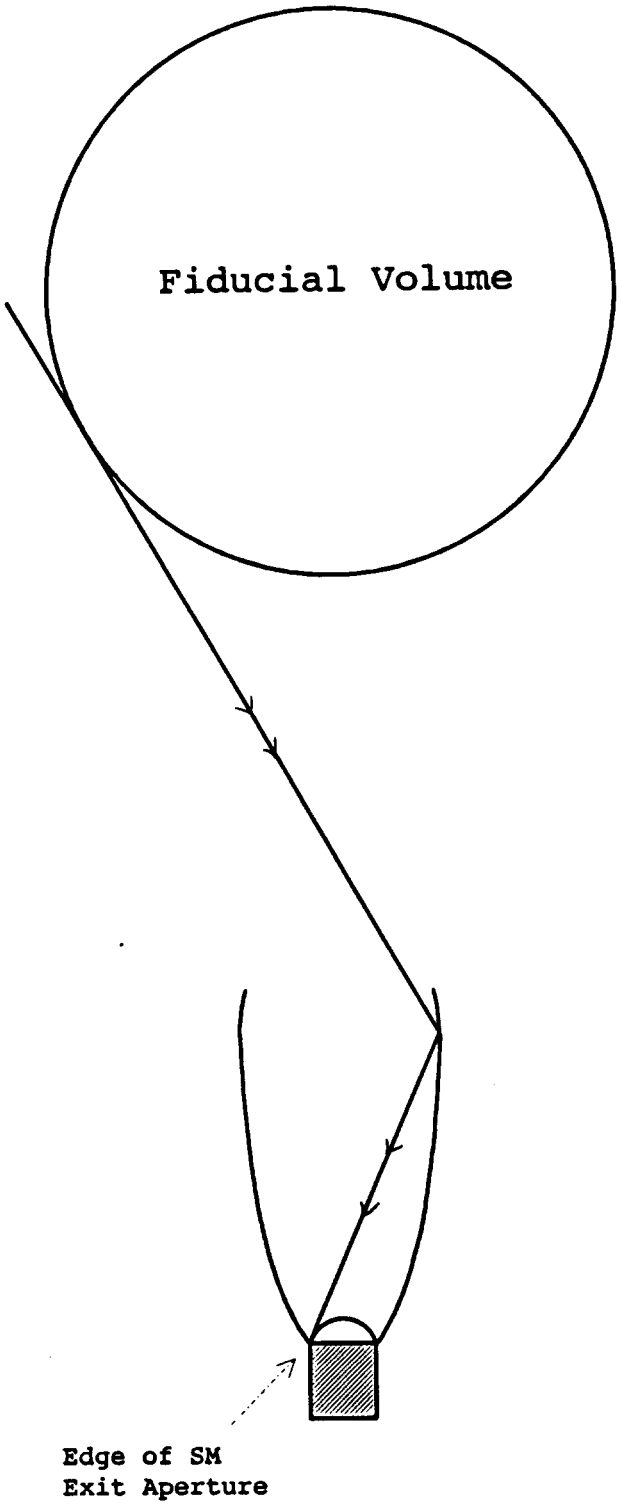

Figure 2.14: 2D SM Cone: Depiction of the upper half of the SM cone where the ray tangent to the fiducial volume is reflected to the edge of the exit aperture as in a CPC cone.

The routine continues in this manner until it is no longer possible to draw a ray tangent to the PMT. At this point, we utilize the 'traditional' Edge-Ray Principle with the edge of the exit aperture as the 'focal' point (Figure 2.14). This second section is not a CPC, however, because we still require that the incoming ray be tangent to the fiducial volume, not originating at some infinite point with fixed extreme incident angle θ_i. The profile continues until it intersects with the ray originating on the edge of the exit aperture that is tangent to the fiducial volume. Figure 2.15 is a 3D computer plot of an SM cone. Note how the curve bends in slightly at the entrance end, a feature unlike the CPC cone which may offer some fabrication difficulties.

With the program detailed above, I designed an SM cone optimal for a 89.55cm Fiducial Volume. As appropriate for the CTF, it has an exit aperture radius of 9.5 cm and the base of the cone (where the PMT lies) was held at 240 cm from the fiducial volume edge. The cone has a length of 117 cm and an entrance aperture radius of 27.4 cm. Employing the same 2D ray trace program used for the CPC cone, I calculated the transmission-radial curve for the SM collector (Figure 2.16). The curve is clearly ideal within program error and therefore maximum light concentration is achieved. Unlike the ideal transmission-angle curve, the ideal transmission-radial curve does not drop to 0 beyond the maximum fiducial radius. This is easily understood. Even for an infinite fiducial radius, there are many rays that can be drawn to the SM cone such that they would be detected giving an average transmission > 0%.

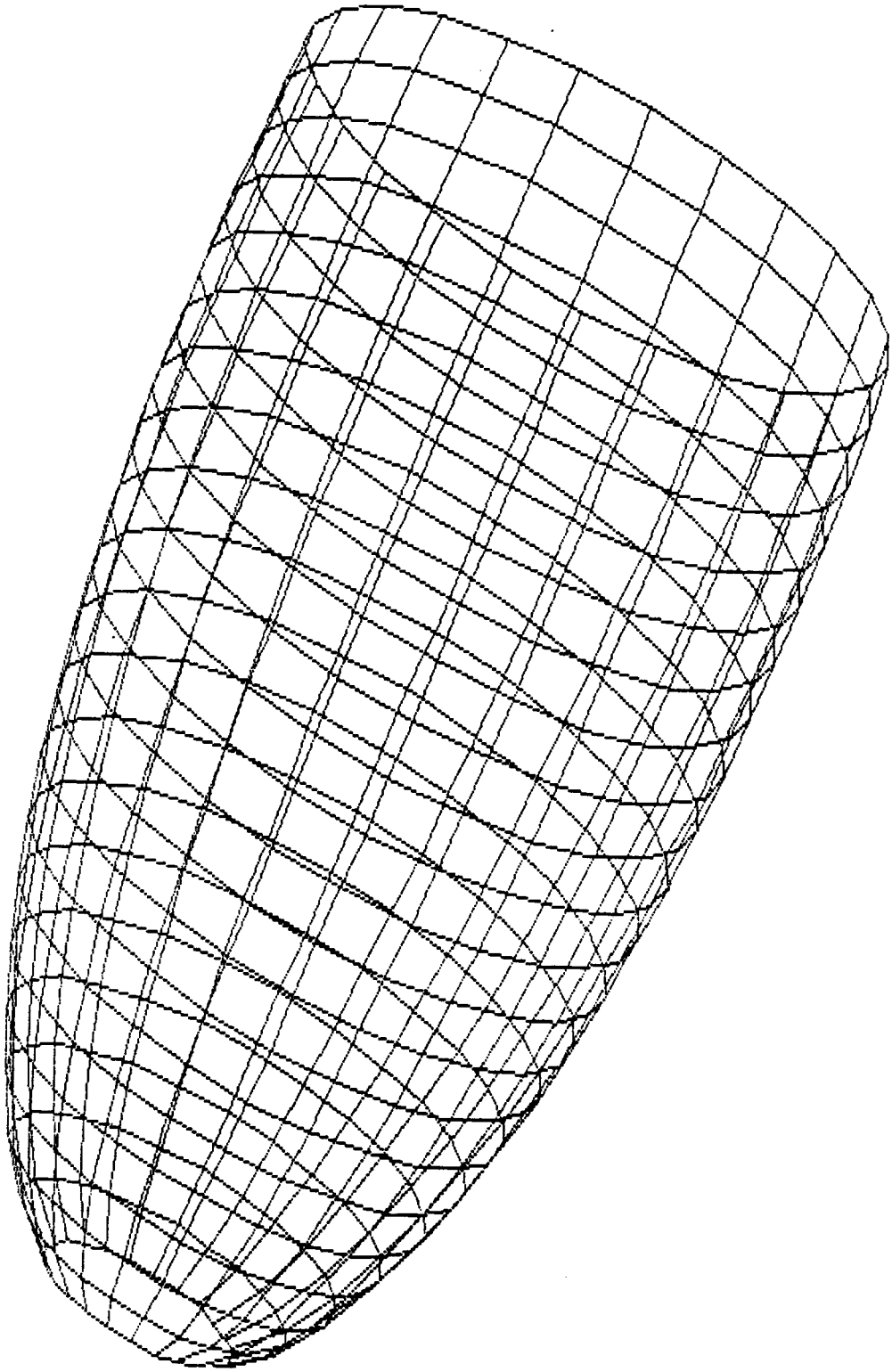

Figure 2.15: 3D Plot of the SM cone.

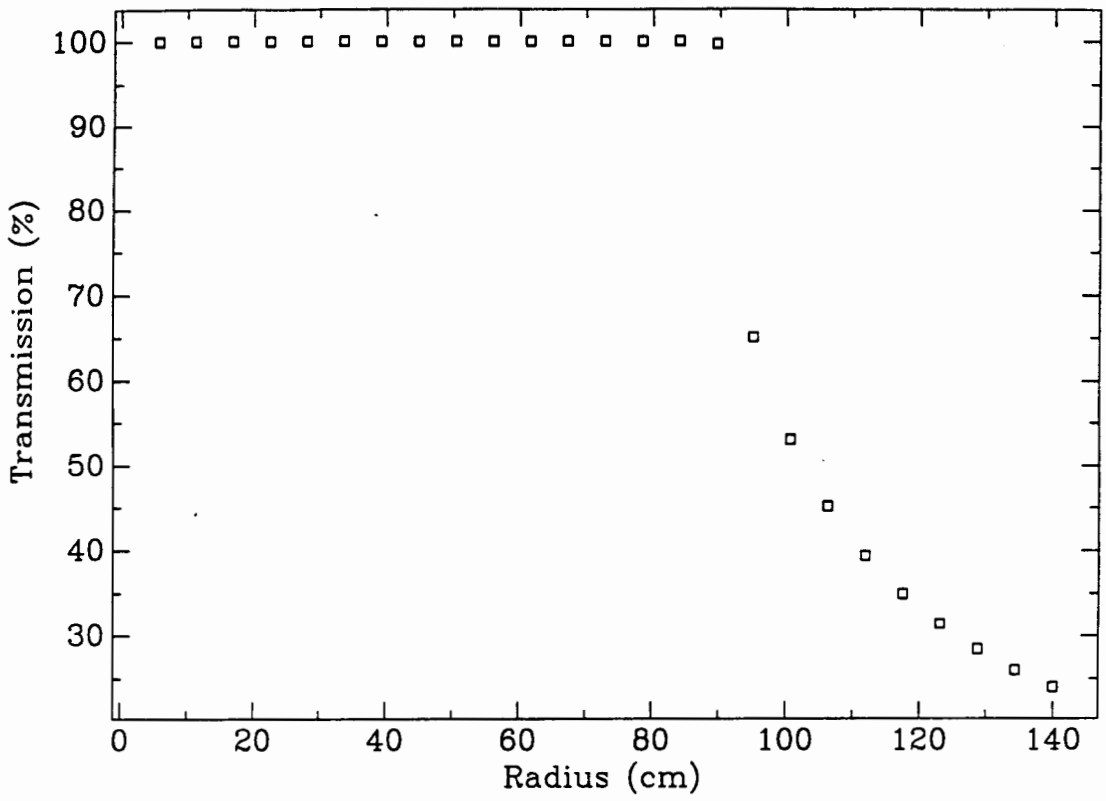

Figure 2.16: Transmission-radial curve for the 2D SM cone designed to look at an 89.55 cm FV.

The success of the SM cone (in 2D) is that it has a flat acceptance out to the maximum fiducial radius where a steep drop in transmission occurs.

I have also written a 3D Monte Carlo ray trace simulation program (trnrad.c - described in Appendix 2.A) for the CPC and String Method cones in order to investigate the effects of skew rays on the transmission-radial curves. The results are somewhat disturbing (see Figures 2.17 and 2.18). Although the SM cone has a greater overall acceptance and a sharper cutoff, the difference is not quite as dramatic as the 2D results. The problem lies in the fact that the 3D SM cone no longer has 100% transmission out to the edge of the fiducial volume (89.55cm).

In order to better understand the significant drop in transmission before the fiducial volume edge, I investigated the polar angular dependence of the transmission for a fixed fiducial radius (89.55cm). Figure 2.19 is a graph of the percentage of photons which are transmitted to the PMT as a function of the polar angle of a fiducial shell with fiducial radius 89.55cm. The polar angle is 0° for rays originating at the lower pole of the shell (the point on the shell closest to the light collector) and 180° for rays originating at the top pole. One observes a minimum in transmission around 65°. As one expects, this corresponds to the portion of the fiducial shell where the extreme rays (those with the greatest incident angle) originate. The effect of skew rays is most significant here. Such a low percentage in this region - relatively near the equator - has a large effect on the average transmission

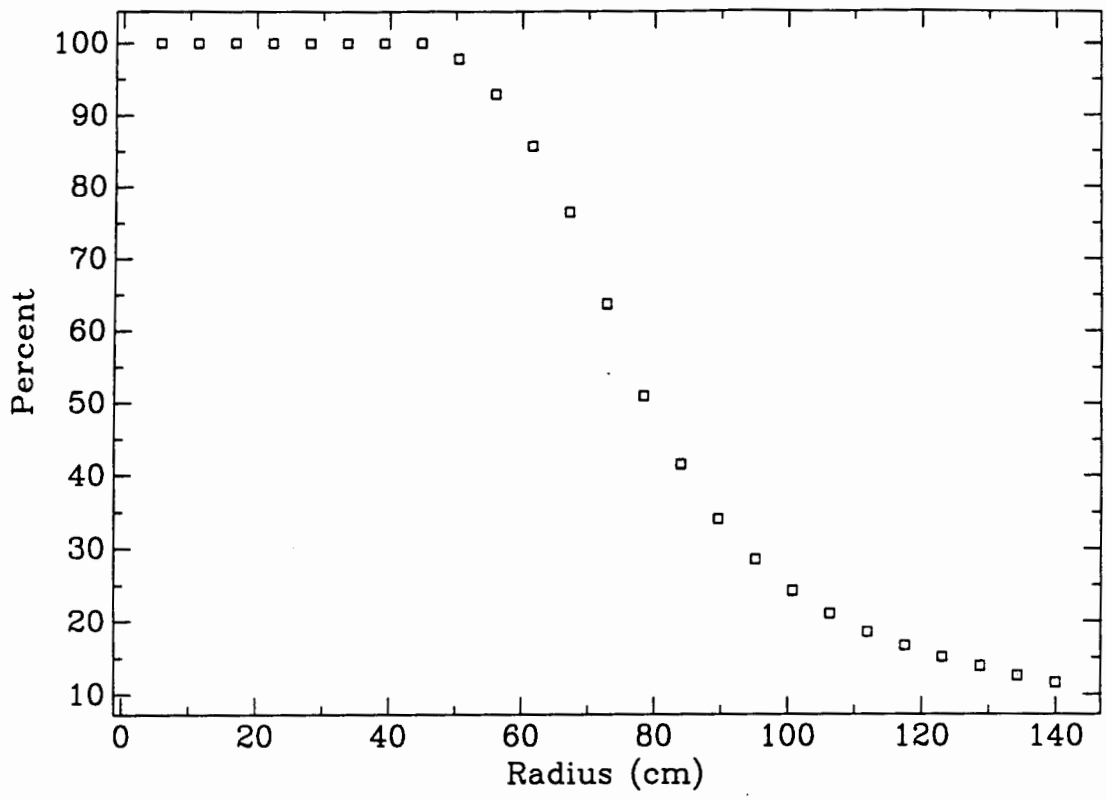

Figure 2.17: Transmission-radial curve for the 3D CPC cone designed to look at an 89.55cm FV (θ_i = 17.39°).

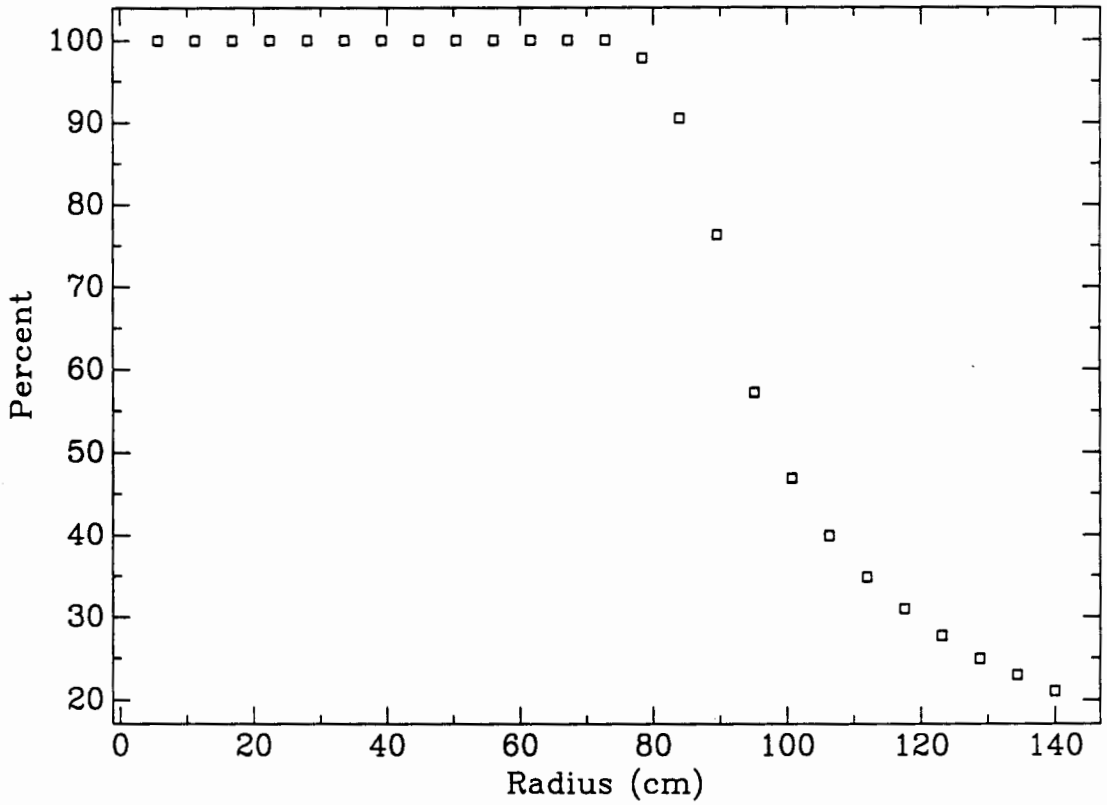

Figure 2.18: Transmission-radial curve for the 3D SM cone designed to look at an 89.55 cm FV.

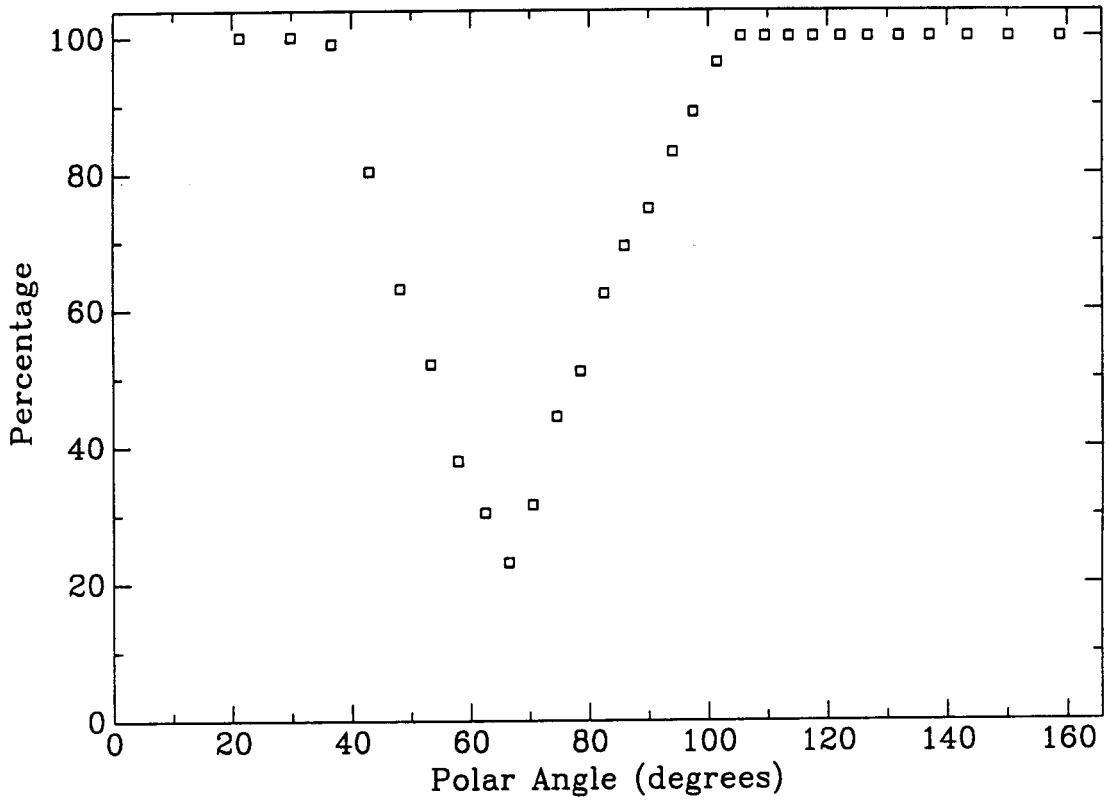

Figure 2.19: Angle-Transmission curve for a fiducial shell with radius 89.55 cm. The Polar Angle is defined to be zero for events originating at the bottom pole of the shell (nearest the light cone).

because more events occur on average in the equatorial range than near the poles. Thus, the explanation for a lower angle transmission near the fiducial volume.

What this 3D transmission-radial curve implies is that we expect poorer position resolution near fiducial edge than we would have expected from the curve for the 2D case. It is at this edge, moreover, that we need the greatest resolution in order to differentiate 'real' events from those associated with the surface impurities of the inner vessel. It ought to be noted, however, that the drop-off before the fiducial edge is not extreme and, in fact, may only minimally affect the 3D resolution in this region. This issue is discussed further in the following section.

Expecting a possible error in my simulation, I have investigated several aspects of my code, including doubling the accuracy of my SM cone and raising immensely the number of events simulated, yet the effects were very minimal. The skew rays associated with the 'misses' very nearly intersect with the PMT, but instead are reflected out of the cone. Because I used the exact same SM profile in my 3D simulation as in the 2D tests (which produced an ideal transmission-radial curve), and as I have checked my code's performance in depth, presently I have high confidence in these results. I do leave open the possibility that there is some error in my analysis, but I consider it quite unlikely. Regardless, the SM cones outperform the CPC cones in terms of radial acceptance. Moreover, I am very confident that any errors discovered in my simulation would only improve the radial performance of the SM

cones.

E. Further Application to the CTF Upgrade

Although the 3D SM cone is not ideal, it clearly outperforms the CPC cone on the basis of its transmission-radial curve. There are, however, several other factors which must be considered with respect to the CTF Upgrade.

The most important issue which demands attention is the coverage that the light collectors will provide. In terms of a single photon incident on the entrance aperture, the coverage per total solid angle ($\Omega = 4\pi$) is expressed by,

$$\frac{d\Omega}{\Omega} = \frac{\cos(\phi)\, dl^2}{4\pi r^2} \quad ,$$

where dl^2 is a finite element of area on the aperture, ϕ is the angle between the normal of the area element and the incident ray, and $\mathbf{r}$ is the vector drawn from the event to the area element (Figure 2.20). If considering the coverage defined by the light cone/PMT system, an additional factor $E_p(\theta, \mathbf{r}, x, y)$ must be introduced. Ep is 1 or 0 dependent on whether the photon is transmitted (via reflections) to the PMT, as a function of the parameters of the ray $(\theta, \mathbf{r})$ and its intersection with the entrance aperture (x,y).

As with our discussion of the transmission-radial curves, it is most instructive to calculate the coverage E_c, as a function of the fiducial radius r_s,

$$E_c(r_s) = \frac{N_{cones}}{N_{events}} \sum_{shell} \frac{\cos(\phi)\, dl^2}{4\pi r^2} E_p(\theta, \mathbf{r}, x, y) \quad ,$$

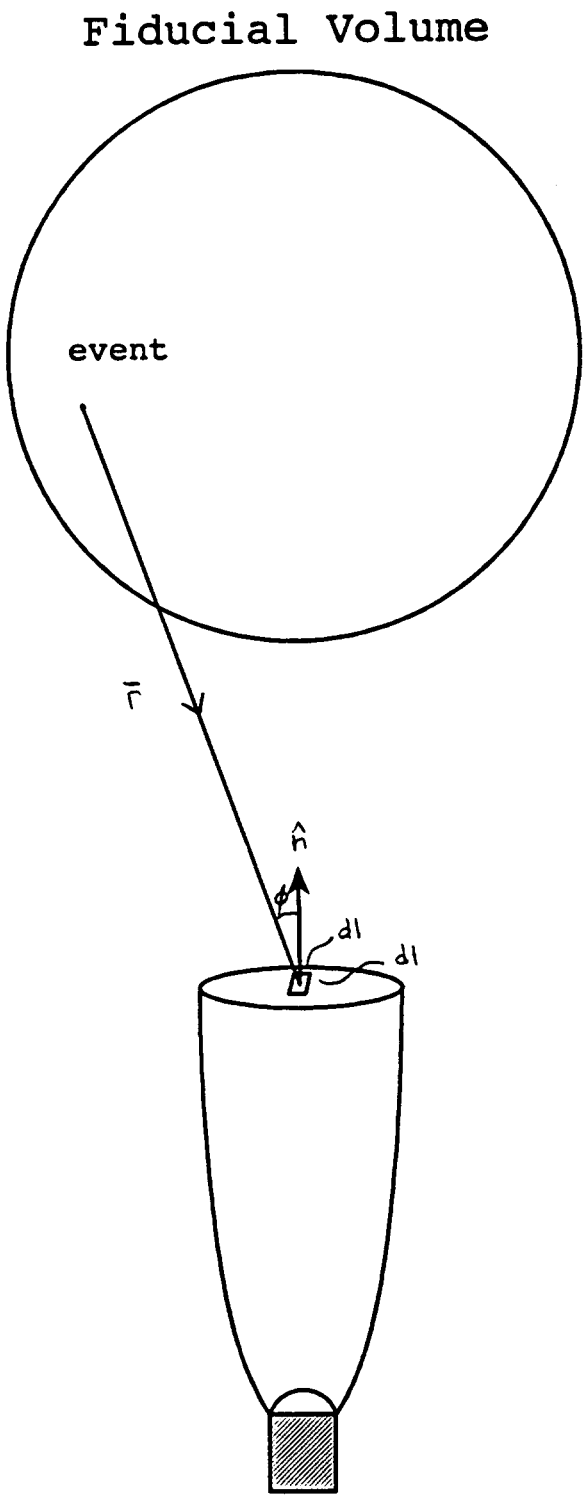

Figure 2.20: Geometry used in the Monte Carlo simulation to determine the coverage of a given light cone - PMT array. (The coverage is expressed in terms of photoelectrons/MeV).

found by averaging a fiducial shell with radius r_s and over the entire aperture entrance. N_{cones} is the number of cones in the array and N_{events} normalizes the sum.

Coverage can be expressed in terms of a more physical factor, namely the number of photoelectrons (pe's) per MeV of incident neutrino energy recorded for a given PMT array. Assuming 10,000 photons/Mev for our scintillator, the factor works out to,

$$\mathbf{Pe/MeV[r_s] = (10000\ photons/MeV)\ (.2)\ (.6)\ (abs)\ (ref)\ (Ec[r_s])}\ .$$

The .2 is the quantum efficiency of the PMT, .6 is the PMT practical factor, abs is an absorption factor associated with the solvent and shielding medium (e.g. water) and ref is a reflection factor dependent on the reflection coefficient of the light collector.

In order to calculate Pe/MeV[r_s], I utilized the ray tracing procedures from program trnrad.c to calculate E_p and added functions to calculate E_c, abs, and ref (see pemev.c described in Appendix 2.D). Figures 2.21 and 2.22 are the Pe/MeV vs. radius curves for 64 CPC and SM cones optimized for a 89.5522 cm fiducial volume. The CPC cone detects many more pe's at the center of the fiducial volume, yet this number drops off dramatically as the fiducial volume extends. The SM cone, meanwhile, records much fewer pe's at the center, yet as one would expect from the transmission-radial curves, this number does not drop off until near the edge of the fiducial volume. The difference in the maximum recorded pe's is solely a function of the length of the two cones and the radius of

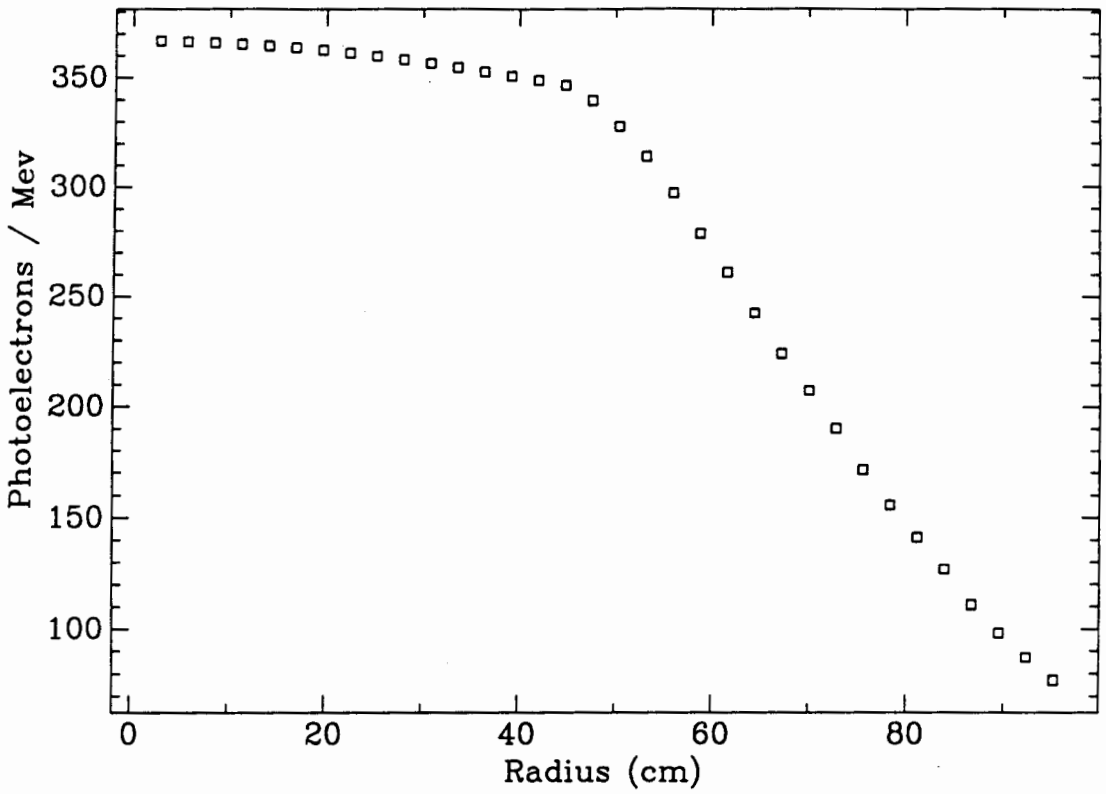

Figure 2.21: Photoelectron/MeV - Radial curve for a 3D CPC cone designed to look at an 89.55cm FV (θ_i = 17.39°). It is assumed that the PMT is ~2.4 m from the FV.

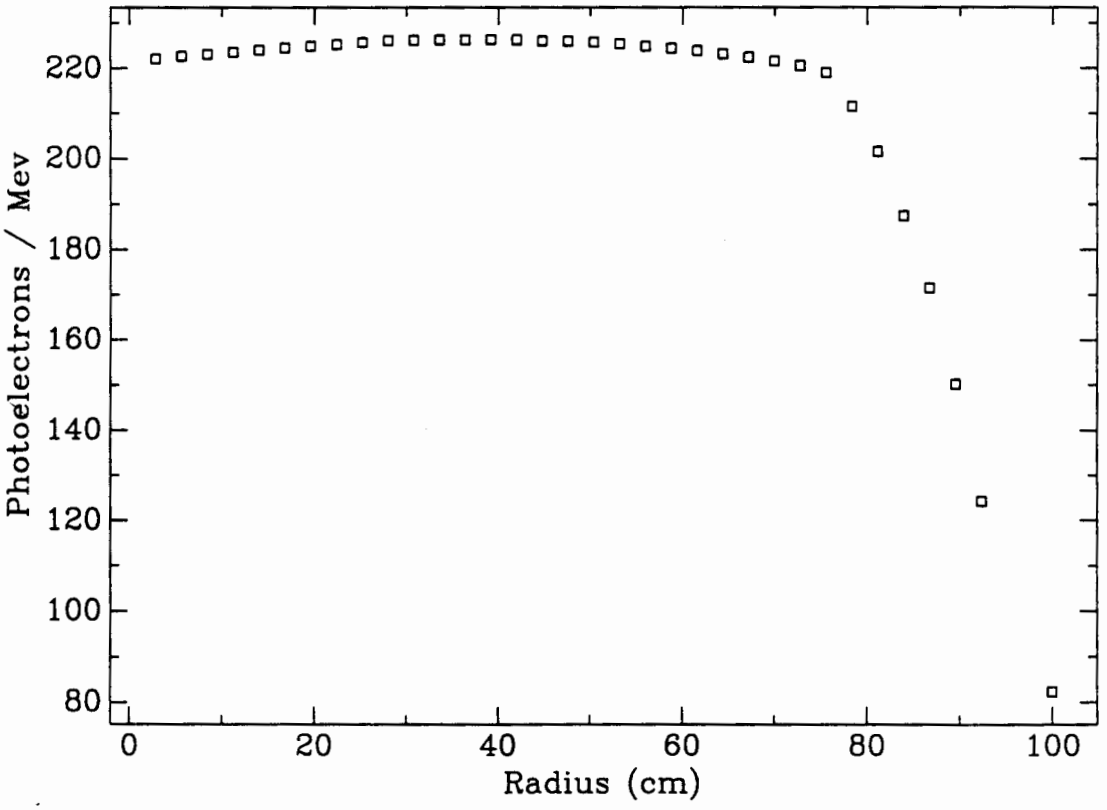

Figure 2.22: Photoelectron/MeV - Radial curve for a 3D SM cone designed to look at an 89.55cm FV. It is assumed that the PMT is ~2.4 m from the FV.

their respective entrance apertures as each can detect all rays incident from these inner fiducial shells.

The number of pe's detected from either array could be increased by implementing more cone and PMT's, yet in order to keep PMT and electronics cost at a minimum, we need to limit the total number of PMT's used. One must examine, therefore, if 220 Pe/MeV is a high enough value or if problems will arise from the implementation of the SM cones. As mentioned in Section 1.G, preliminary calculations suggest that an output of 200 Pe/MeV is satisfactory and, therefore, the coverage of 64 SM cones ought to be sufficient. This fact, combined with the near ideal radial acceptance of the SM cone, makes it the

Although it is important that the cones used for the CTF Upgrade record enough pe's all the way out the edge of the fiducial volume, of greater importance is the characteristics of this coverage at the edge fiducial volume. Specifically, the light collector array chosen for the CTF Upgrade must have accurate 3D resolution at the edge of the fiducial volume, allowing one to differentiate between background events for the inner vessel and actual events within the fiducial volume. As suggested previously, this is exactly the advantage or implementing the SM concentrator over the CPC (and other) cones.

The sharp drop in Pe/MeV near the fiducial edge, if utilized properly, will result in a very high resolution for identifying background events associated with an inner vessel at 1m. As Figure 2.22 indicates, the number of p.e./MeV likely to be detected with a 64 SM cone array at the inner

vessel (1m) is approximately 1/2 the maximum value. What this ought to allow, then, is the differentiation between an event in the maximum region from one at the inner vessel solely from the pattern of cones that are 'lit' (and unlit) and the distribution of pe's detected. Because the transmission has a complicated dependence on polar angle and because the SM cone's 3D performance is not ideal, determining this pattern with high confidence is far from trivial. The non-ideal radial performance of the SM cone at the fiducial edge does offer a problem, yet can possibly deal with this problem by using the 89.55cm cone while defining the fiducial edge at the point where the curve falls from maximum (~75 cm).

Presently, I am composing a program which will simulate for any N cones the CTF Upgrade enabling me to identify these patterns of lit and unlit cones. I will be able to fix the locations and paths of events both in and outside the fiducial volume and thereby determine which cones are 'lit' (and unlit) as one passes from the radial regime of maximum detection (~75 cm) to the inner vessel (~1m). It is hoped that an identifiable pattern of lit and unlit cones will be revealed, enabling one to differentiate with high confidence an event that occurs at the surface of the inner vessel as an event outside the fiducial volume (i.e. background). This would allow a terrific reduction in background, and therefore a higher probability of performing solar neutrino physics with the CTF. Unfortunately, efforts underway on this project were not completed at the time of this publication and therefore will be detailed in a separate document. Because of the near

ideal radial transmission of the SM cone, I fully expect to succeed. In any case, the String Method concentrator is optimal for the CTF Upgrade and one must therefore pursue fabrication techniques for building the cones under the experiment's strict environmental constraints.

Chapter 3 - Fabrication

A. Materials

Because of the unique constraints of the CTF, there is a very limited selection of materials for the fabrication of the light collectors. In particular, the fact that the light cones will be located very near the fiducial volume requires that they be made of a radiopure material. Presently, acrylic and a special nylon copolymer (Miles-Mobay Nylon 38F), are considered the most likely fabricating materials. Table 3.1 shows the estimated levels of U, Th, and K for the two materials as well as the rest of the CTF materials. Calculations performed by Calaprice (priv. comm.) suggest that light cones made of these two materials would have acceptable background levels for the CTF Upgrade design. There has been, however, some recent discussion concerning the levels of radon found in nylon. Although no conclusions have been drawn, the potential problem must be further considered before the final fabrication of the light cones proceeds.

Another factor important to the choice of material is its resistance to xylene, deionized water and mineral oil - the shielding media most likely for the CTF Upgrade. Tests performed by myself and others on nylon demonstrate its resistance to all 3 mediums. Acrylic, however, is attacked by xylene at a dangerous rate. Within a xylene environment an acrylic light cone would be destroyed well before the completion of the CTF experiment. Furthermore, a leak in the inner vessel could release solvent destroying the entire array

Component	U	Th	K (g/g)
Acrylic	10^{-12}	10^{-12}	10^{-9}
Nylon	$2x10^{-11}$	$2x10^{-11}$	10^{-8}
Water	10^{-13}	10^{-13}	10^{-10}
Scintillator	$5x10^{-16}$	$5x10^{-16}$	$5x10^{-13}$
Steel tank	10^{-8}	10^{-8}	
PMT's and bases	$4x10^{-8}$	$2x10^{-7}$	$1x10^{-4}$

Table 3.1: Estimated Radioactive Impurities in the CTF Materials.

of acrylic light cones regardless of what initial medium it was in. On this basis, then, nylon outperforms acrylic. Other pertinent factors are the cost, material strength, availability, ease of fabrication and ease of coating the material with a reflective surface. These topics will be addressed specifically for nylon in the following sections.

I chose to use nylon as the fabricating material of the light collector for a simple reason. As part of my overall research for the Borexino project, I helped manufacture two prototype inner vessels - nylon bags - for the CTF. The technique we (Mr. Santorro, Mr. Loser, and myself) used involved the chemical welding of thin, .010" thick, nylon panels shaped like orange peels to create a spherical bag. The welding agent, Nylaweld (3:1 H_2O:Eth), dissolves nylon allowing the affected area to flow together. It then evaporates leaving behind pure nylon. Thus the bonds, if done properly, are at least as strong and pure as the original nylon. For the inner vessel, the joints were made by butting edges of two panels and then welding nylon strips over the top of the .

Although the fabrication technique is fairly crude, it turned out to be quite successful and relatively simple to implement. In essence, I took the techniques we developed in our efforts with the inner vessel and applied them to the fabrication of light cones. The properties of nylon lent themselves nicely to my efforts. Nylon is lightweight, very resistant to pulling, easily shaped, and in our case, abundant. In almost all aspects, it seemed the ideal

substance to apply to the fabrication of these large light collectors. Before proceeding with fabrication, however, I wanted some assurance that the nylon would have the necessary tensile strength to hold its shape against gravity.

B. Stress Test

As discussed in Section 3.C, two light cone prototypes were designed and fabricated. This section details theoretical analyses of the strengths and durability of the two structures.

1. CPC Prototype

In order to evaluate the likely success of nylon, steps were taken to determine if nylon would have enough stability to keep its form within rather strict tolerances - less than ~5mm deflection at any point. This tolerance was agreed upon after simulations by Lowry and Moorhead (priv. comm.) suggested that a few millimeter deflection does not significantly affect the characteristics of the light collector. In addition, an engineering rule of thumb is that a deflection of 1/200 the length is optimal. Preliminary analyses by engineer Bob Parsells at PPL (priv. comm.) showed that a CPC cone 135 cm long would buckle critically if restrained at the exit aperture to point towards the horizon (perpendicular to gravity). By introducing a support structure of rings and bands, however, the stresses and deflections associated with the cone were greatly reduced. With this promising analysis, I decided to fabricate a prototype light collector to examine fabrication techniques

and experimentally test deflection levels. At that point, the SM cones had not been fully investigated, and thus, the first prototype was a 13° truncated CPC cone 46.61 cm long with a 3.17 cm radius exit aperture and a 13.36 radius entrance aperture. These parameters were chosen with a particular project in mind that has since been discontinued. A full discussion of the CPC Prototype fabrication follows this section.

2. SM Prototype

Once efforts progressed on the theoretical examination (e.g. computer simulations) of the SM cone, I investigated the stability of a truncated (1m) SM cone designed to look at a fiducial volume of 89.5522 cm. I decided to truncate the SM cone to 1m (from 117 cm) to ease fabrication and because simulations show that the effects of partially truncating an SM cone affect minimally the Pe./MeV - radial curve. [Compare Figure 3.1, the Pe/Mev - radial curve for a the truncated SM cone with Figure 2.22 for a full sized SM cone.] The fact that truncation does not affect greatly the performance of the SM cone may significantly reduce fabricating costs for the design that is finally decided upon.

Using a finite element analysis program, IMAGES, I ran static stress and deflection tests on the proposed design for .020" thick nylon. I input known values for the material properties of nylon (Young's Modulus = 4.5×10^5 psi, Weight Density = .0414 lbs/in^3, Poisson's Ratio = .45), fixed the exit aperture against translational motion and applied a gravity load perpendicular to the axis of symmetry. Without a

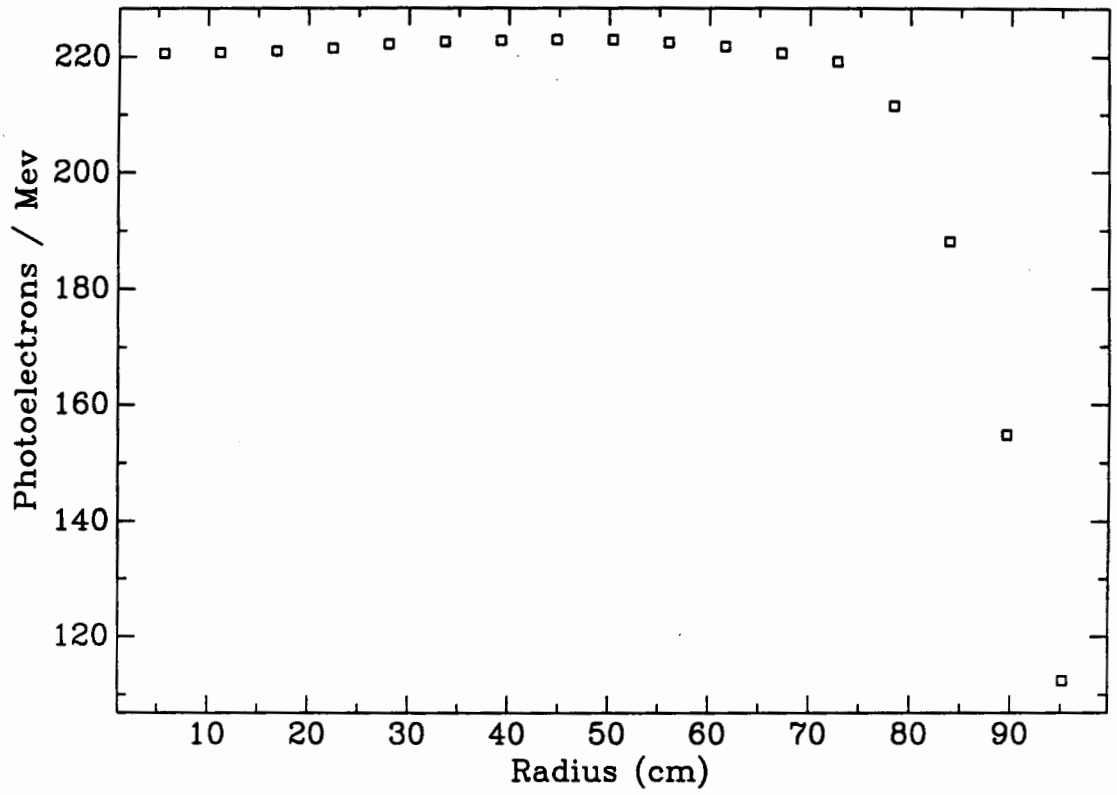

Figure 3.1: Photoelectron/MeV - Radial curve for a truncated (1m) 3D SM cone designed to look at an 89.55cm FV. It is assumed that the PMT is ~2.4 m from the FV.

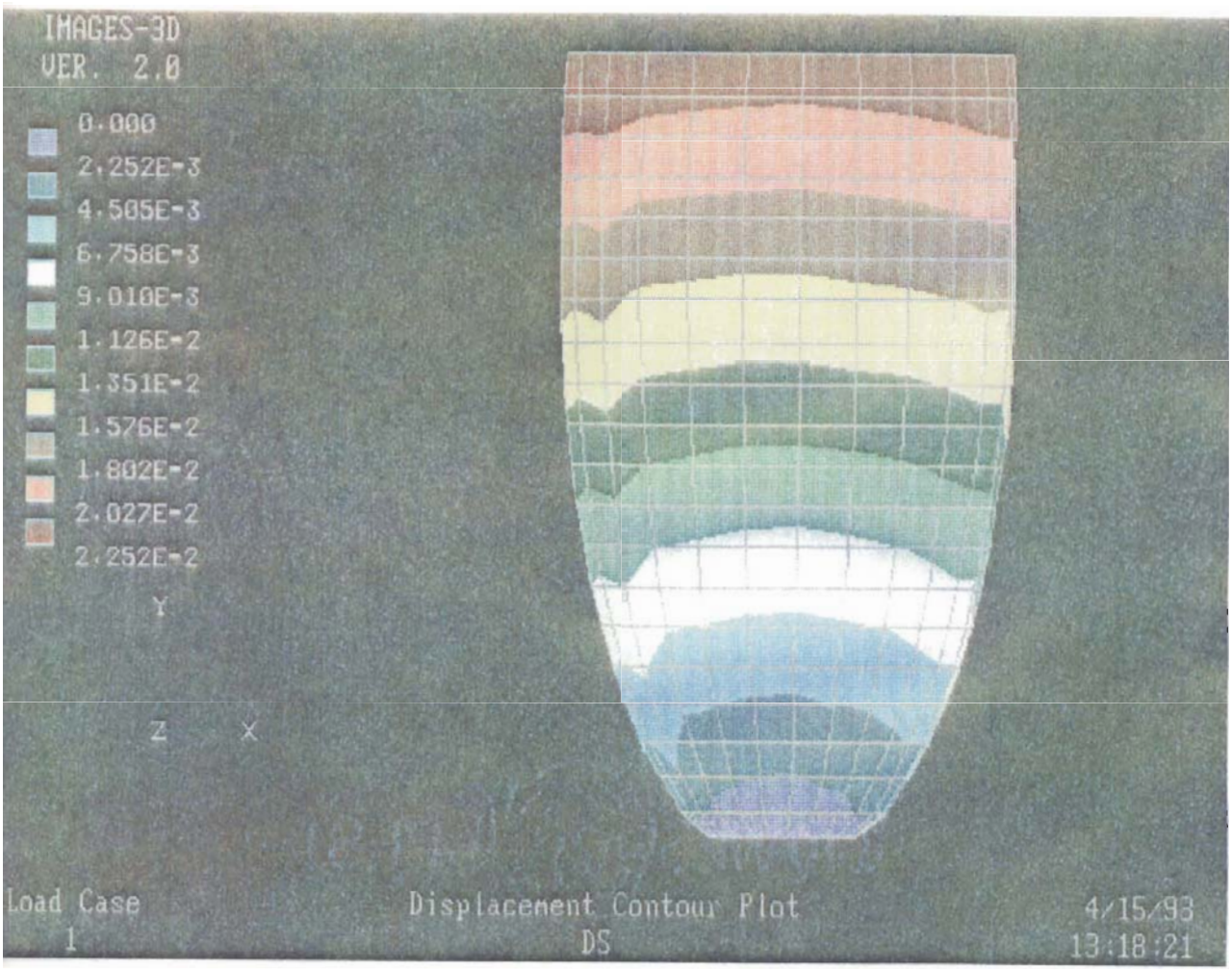

Figure 3.2: Displacement contour plot of the truncated SM cone. (Displacements are all given in inches).

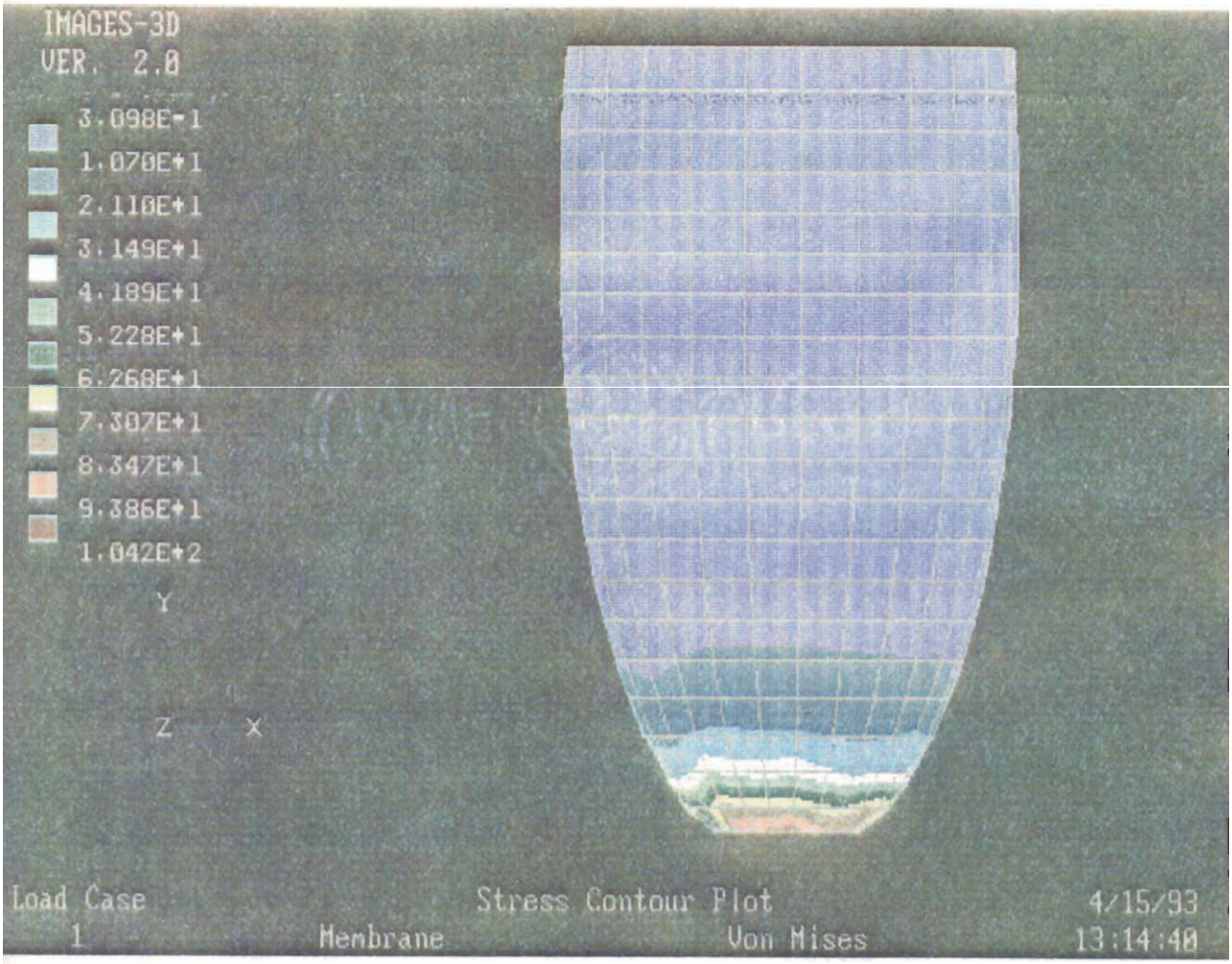

Figure 3.3: Stress contour plot of Von Mises membrane stress. (Stresses are given in psi).

support structure, there were critical deflections in the cone at the entrance aperture. By attaching a nylon ring (.25" square cross-section) to the entrance aperture and 5 longitudinal bands evenly spaced on the outside of the cone, the buckling is significantly reduced.

Figures 3.2 and 3.3 show a color plot of the deflection and stress intensity contours for the supported SM cone. The analysis indicates that the deflection would be within 'safe' levels (.0225" ≈ .57 cm) and the stress levels are far below the 2000 psi critical stress point(determined from pull tests on nylon performed at PPL). Certainly, the finite element analysis can only be taken as an estimate of the actual success, yet the results for this design were quite promising. Therefore, with a relatively successful CPC Prototype already built, I decided to fabricate a SM Prototype to experimentally test the feasibility of fabricating a full sized (1m) collector with nylon.

C. The CPC Prototype - The Panel Technique

The procedure for the fabrication of the CPC Prototype (CPC-P), termed the Panel Technique, can be separated into four steps:

1. Construct a male mold (external surface important)
2. Cut and assemble the panels
3. Coat either the panels or the fully assembled cone
4. Attach support structure

The four steps will be addressed separately so that all of the pertinent options may be more fully discussed.

1. The Mold

Unlike the fabrication of the nylon bag for the inner vessel of the CTF, a full-sized mold is necessary to create a light collector with nylon. With the sphere, as long as the joints connecting panels are made accurately the internal pressure will force a spherical shape. The light cone, however, must be built to the exact shape because there is no internal pressure. Several methods were investigated for the manufacture of a male mold - a mold whose external surface is the desired shape. For my purposes, the mold needed to hold strict tolerances over time and also be heat resistant. One option that was pursued involved spinning a mold out of stainless steel. Another dealt with injection molding a plastic form to the desired shape. On the whole the methods were either too costly or time consuming. After weeks of research one method, the construction of a wood mold, was decided upon.

Wood is relatively lightweight, easily machineable, rather resistant to heat, smooth when sanded, and repairable with wood pulp if nicks or cracks develop. Before lathing a wood mold, we needed a large solid block that could be machined to the desired shape. Several large pieces from a destroyed wooden table were glued together until a large enough block was made. In order to lathe the block to the desired shape, it was necessary to use a follower - a 1" thick metal piece - that has the exact curve of the mold cut into it. An aluminum piece was cut by a Computer Numeric Control (CNC) Mill within approximately .01" tolerance of the designed

CPC curve. The lathe follows the curve on the metal follower and thereby cuts the block into a very accurate wood mold. Figure 3.4 is a photograph of the completed wood mold for the CPC-P next to its follower. This method was very successful in creating an accurate mold as well as time and cost efficient (constructed within 1 week).

2. The Panels

In order to make the panels, a thin metal 'cutout' must be fabricated. Considering the cutout in only 2 dimensions, the vertical axis corresponds to the arc length of the desired curve, and the horizontal axis is 1/8 the circumference of the curve at each point along this arc length. Therefore, a panel made from the cutout ought to compose 1/8 of the light collector when laid flat against the mold. A cutout was made from aluminum and 8 panels were cut out of .02" thick nylon. By using two-sided tape fastened vertically to the mold, two panels are laid on the mold such that their edges are abutting. In order to force the nylon panels to keep the shape of the mold, heat is applied with a heat gun until the panels are sufficiently soft and therefore moldable. The application of heat is a necessary part of the Panel Technique. As the panels are joined, they need to be heat formed to the wood mold such that once removed from the mold they possess the correct shape. Ideally, the panels ought to be heated until they become flexible (~100°C) and then shaped as they cool by applying appropriate pressure. For a large scale project, an oven or array of heat lamps would be more efficient than a heat gun to heat form the panels.

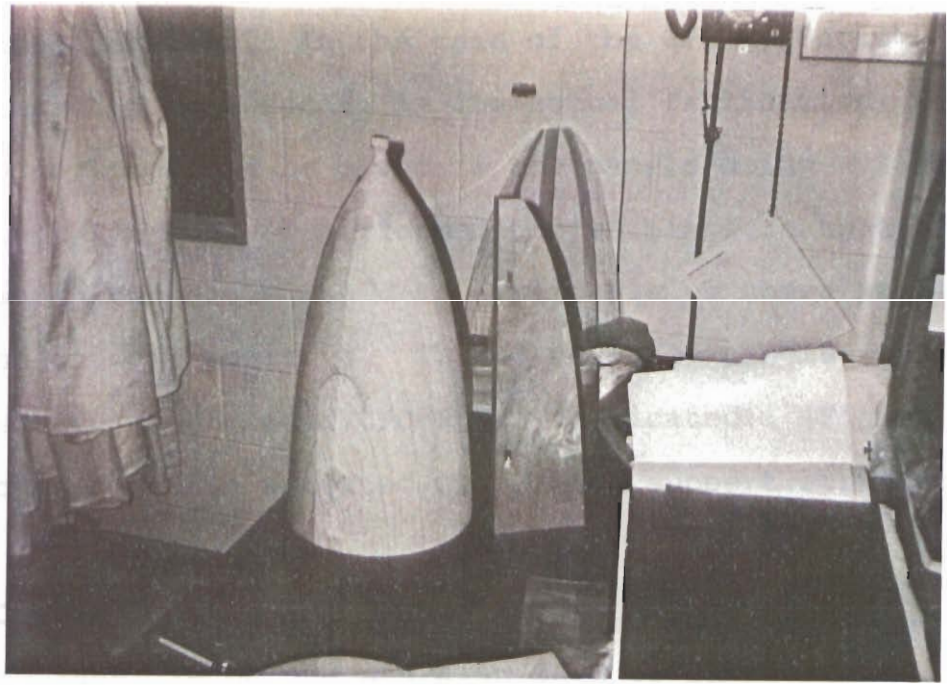

Figure 3.4: Photograph of the wood mold for the CPC-P next to its aluminum follower. A nylon panel can be seen attached to the mold.

Figure 3.5: Photograph of the uncoated CPC-P with support structure.

Once the two panels are flush against one another they are bonded together. In the case of the CPC-P, Scotch fiberglass tape was used. In the actual fabrication, strips of .010" nylon would be glued to the panels using nylaweld as with the nylon bag. I performed tests to ensure that the nylon strip technique would be applicable to these panels and the results were positive. Thus 8 uncoated panels were taped together and an uncoated CPC-P was fabricated. (Figure 3.5 is a photograph of the uncoated CPC-P with support structure).

3. Coating

The Panel Technique was also used to fabricate an aluminum coated CPC-P. Nylon sheets were coated with a reflective coating of aluminum by standard evaporation coating techniques (discussed more thoroughly in Section 3.E). These sheets were cut with the cutout and assembled with glass tape as in the uncoated CPC-P. It is possible that other techniques could be used to coat the light collector. The other avenue that I specifically pursued was to coat all at once a fully assembled uncoated light cone with the evaporative coating technique. Unfortunately, this requires a very large vacuum bell jar as well as an elaborate system of evaporative boats to ensure an even coating. A vendor, EMF Corporation, was found who offers this service, yet their quote was too expensive ($550/piece). Figure 3.6 is two photographs of the coated CPC-P with support structure.

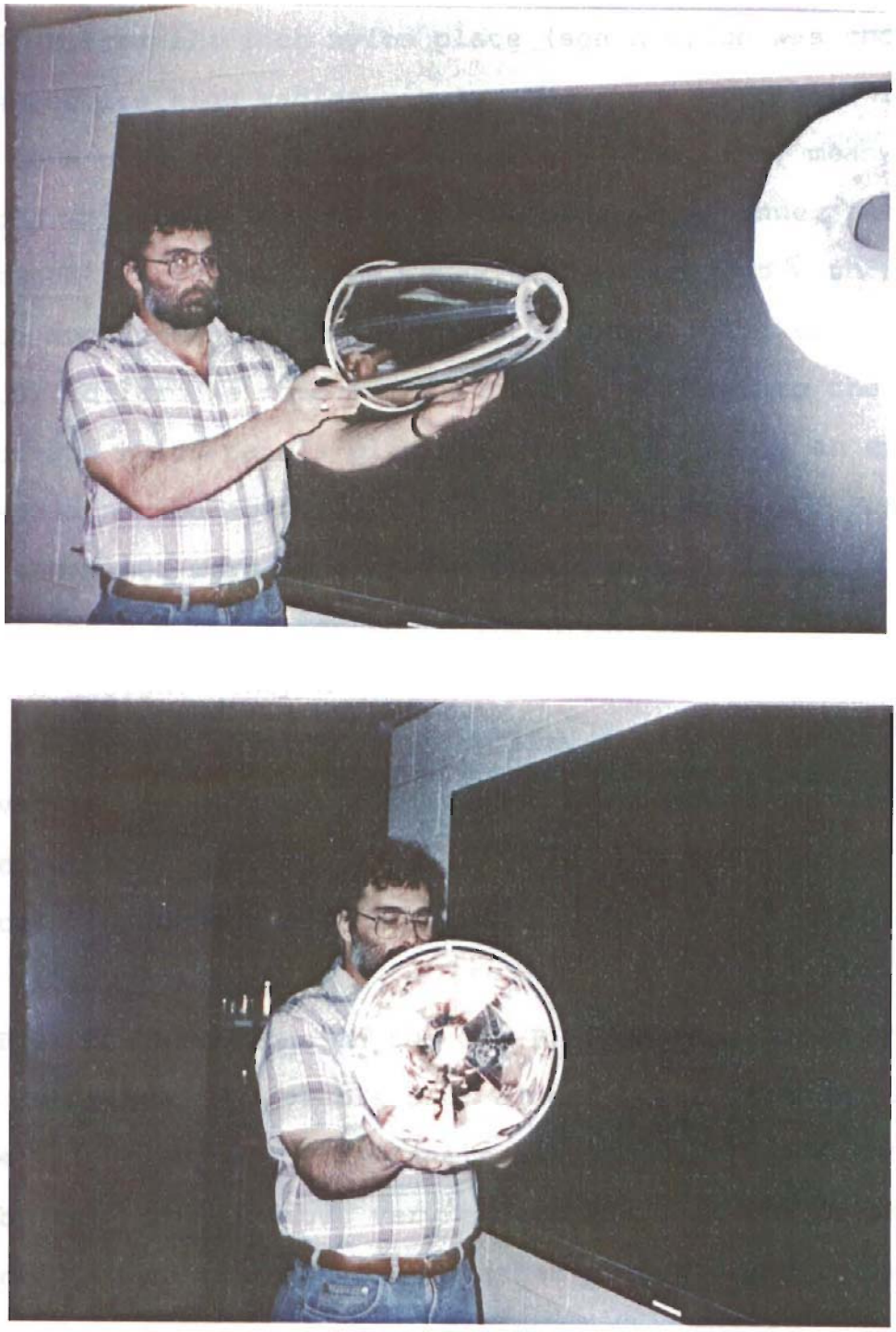

Figure 3.6: Two photographs of the coated CPC-P held at different angles. (Model: Pat Santorro)

4. Support Structure

Two nylon rings and 4 nylon ribs were manufactured for the CPC-P from 1/4 inch nylon plate (again nylon was chosen because of its high radiopurity). The rings were cut with a lathe to a precisely designed diameter. The ribs, meanwhile, required more detailed attention because their inner surface must match the curve of the CPC-P. They were cut with the CNC Mill in much the same manner that the metal follower was constructed for the mold. The ribs can be glued to the outer edge of the nylon panels with nylaweld or possibly an epoxy that would not be attacked in the shielding medium. For the CPC-P the rings were made to fit into the ribs from the outside and are not attached to the nylon panels. This was an error in design. They need to be attached to the panels in order to provide significant stability. This error is observed in the amount of deflection that occurs in the nylon at the entrance aperture end. Although the buckling is not critical, it exceeds tolerant levels.

From the work done on the CPC-P, I was convinced that the Panel Technique is a viable one, albeit labor intensive. It became very clear that the panels used to fabricated the cone need be very flat and cut very accurately. If not prepared correctly, they will not fit together properly and there will be 'gaps' in the cone. This effect is particularly prominent near the exit aperture as observed with the coated CPC-P. I expect that this was the result of a slightly inaccurate cutout.

The overall deflection levels on the two prototypes with support structure were quite low, however, even granted the fact that the structure was not designed ideally. Therefore, with optimistic confidence, I looked to fabricate an SM Prototype (SM-P) 1m in length, with a maximum diameter of 60.344 cm. Although the procedure was essentially the same, a few points will be addressed on the main differences.

D. String Method Prototype (SM-P)

1. Mold

Nearly 3 times the size of the CPC-P, a mold made for the SM-P out of a solid block of wood would be both expensive and extremely unwieldy to lathe. At first, an entirely alternative approach was investigated which remains a very respectable method. In brief, one would lathe a large styrofoam block using a metal follower to produce a mold slightly smaller than the desired cone. Then, the styrofoam is covered with fiberglass strips until a fiberglass form larger than the designed cone is built. This fiberglass form can then be machined (sanded) to the exact shape. It would be sturdy, heat resistant, inexpensive, smooth, and easily repaired. Unfortunately, there are several problems with this method. First, it requires the acquisition of a large block of machineable styrofoam. An estimate from Delaware Valley Pack and Seal places the price at $665 for the amount of styrofoam that would be needed. Secondly, more machine time would be needed (expense and time) and, finally, the glass dust particles released in machining of fiberglass are very

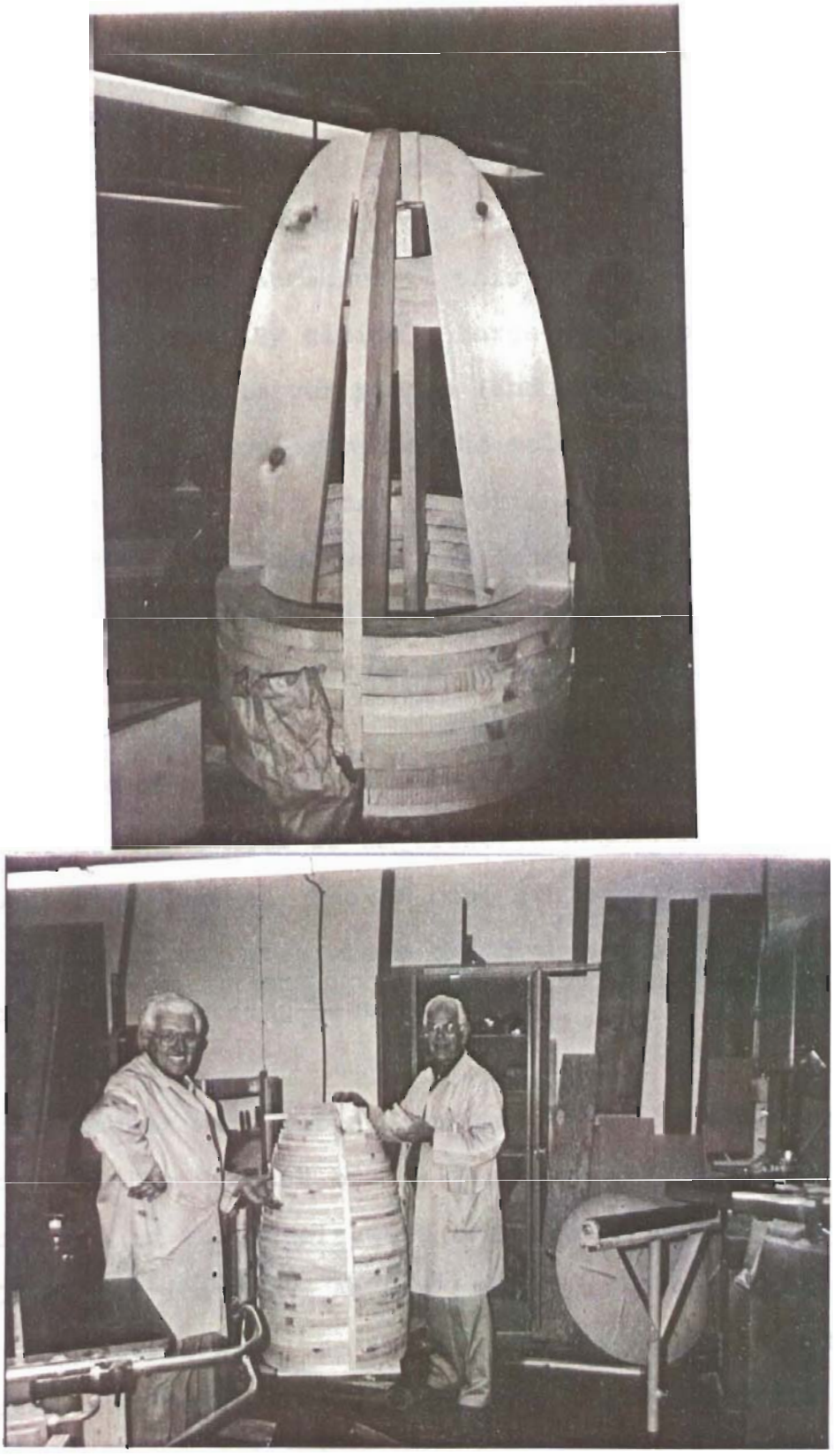

Figure 3.7: Two photographs of the wood form being built. Note the hollow area within the form.

dangerous. With these drawbacks in mind, an alternative approach involving wood was decided upon.

First, a removable, 4 sided pyramid of plywood was assembled to the approximate dimensions of the mold. 4 Large pieces of pine were attached to this pyramid to provide a support structure. By gluing a large number of small pieces of pine to these 4 larger pieces (and each other), a wood form was built up that approximated the mold. Figure 3.7 is two photographs of the wood form in various stages of assembly. This procedure was time, wood and labor efficient. In addition, the lighter, hollowed-out form is more easily handled both in terms of machining and cone fabrication. This wood from was lathed to the SM shape using a metal follower as with the CPC-P (Figure 3.8). The program used to create the follower with CNC Mill is sm_fllwer.c described in Appendix 3. The procedure used for the SM cone can easily be applied to a cone of almost any size where the dimensions of the lathe are the only real limiting factors.

2. Panels

The panels were cut and assembled in the same manner as for the CPC-P. Because of the larger panel size, the effects of wrinkles in the panels were even more apparent than with the CPC-P fabrication. At the time of publication, I have just begun tests with the SM mold (Figure 3.9). The first uncoated SM-P fabricated was far from ideal. Extreme wrinkling in the initial panels (solely a function of the rolling method used to store the nylon sheet) led to large gaps in the cone at the exit aperture end. Also, it was

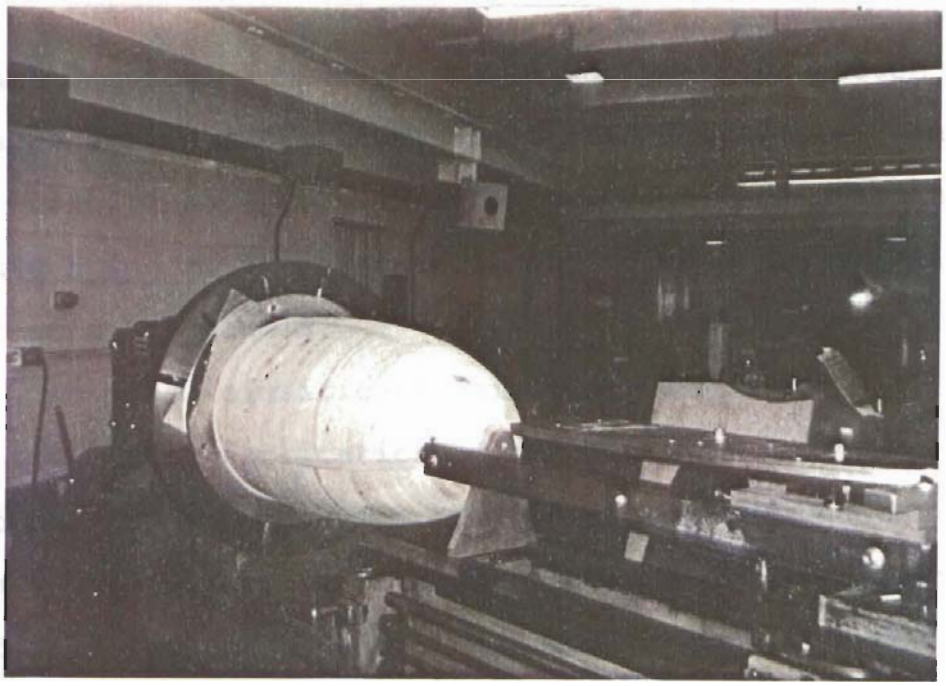

Figure 3.8: Photograph of the wooden form on the lathe. The metal follower can be seen in the foreground of the photograph, lying parallel to the wood mold.

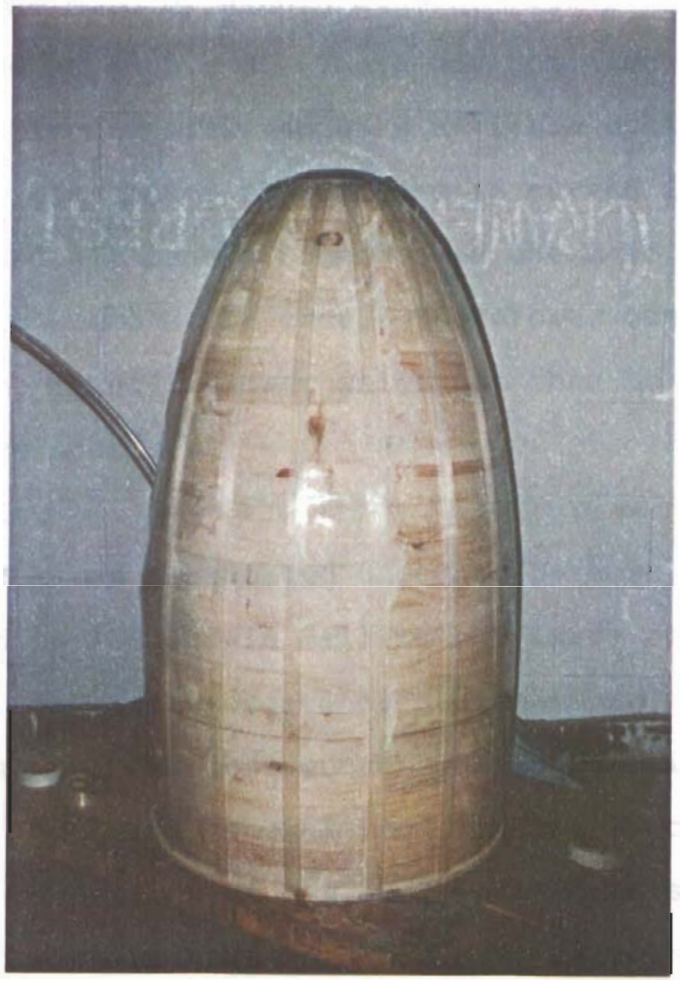

Figure 3.9: Photograph of the completed wood mold with nylon panels.

apparent that the heat gun could not generate enough heat over a large are to sufficiently heat mold the panels. In the next few weeks, I hope to find a flatter supply of nylon sheet and create a second SM-P, possibly exploring other techniques to heat mold the panels.

Two other significant points are worth noting. First, for the 'true' fabrication of the light cones it may be necessary to do away with the two-sided tape because it could attack the reflective coating on the panels. As such, the wood mold for the SM-P was designed to be .5" longer on top and bottom than the designed light cone. One could clamp the panels to the mold by a thin aluminum strip and two c-clamps, one fastened at each end. There was not time to study this method, but it certainly appears a viable option. Second, recall that the end of the SM cone is curved inward a little. If one bonded all panels together on the mold, it would be impossible to remove the cone. Therefore, one makes the final joint on the metal follower which can then be removed through the entrance aperture.

3. Coating

The SM-P was fabricated uncoated because the vacuum bell jar located in 125 Jadwin Hall is not large enough to hold the panels for evaporative coating. Furthermore, the expense of having a vendor coat just enough panels to build the SM-P was too great. To get a general feel for the success of reflection properties of the SM-P, one could spraypaint it with a chrome paint that would be partially reflective.

4. Support Structure

The support structure for the SM-P will be very similar to that for the CPC-P except that the two rings will be attached to the nylon panels. This ought to match the design analyzed with the finite element program described in Section 3.B. If it is found to not provide enough support, extra rings will be added. In addition, I plan to investigate the use of stainless steel rings near the exit aperture where they would be far enough from the fiducial volume to create background problems. These rings would provide much greater stability than the nylon rings and are just as simple to manufacture.

E. Coatings

It is central to the fabrication of a light collector to insure that its reflective surface transmits the highest percentage of light possible for the working wavelengths of the experiment. For Borexino's scintillator, the light spectrum is in the visible region ranging approximately from 350 nm to 500 nm. Therefore, my investigation focused on reflectors ideal for the visible region. The other significant factor associated with the reflective material used is its resistance to the shielding media proposed for the CTF Upgrade: deionized water, mineral oil and solvent (xylene). Because of their very high reflectivity and availability, I most actively pursued aluminum and silver. Table 3.2 summarizes the theoretical and experimental results of tests that I performed on a variety of reflective coatings that involved aluminum and silver. The following is a

discussion of the tests and a further discussion of the results in Table 3.2.

Table 3.2: Summarization of the Coating Tests. A **Y** indicates a successful coating, a **P** a possible solution, and an **N** that the coating was vigorously attacked.

Material	Reflection[29]	Mineral Oil	Xylene	Deionized Water
Al Only	> 90%	**P** (~3 days at 90 °C)	**Y** (3 months at room temp.)	**N** (< 1 day at room T)
Al w/ SiO	> 90%	**Y** (~4 weeks at 90° C)	**Y** (")	**N** (< 1 day at 90° C)
Ag w/ a 100Å Cr underlayer	> 90%	**P** (~12 days at 90° C)	**Y** (")	**N** (< 2 days at 90° C)
Ag w/ SiO (no Cr)	> 90%	**Y** (over 12 dys at 90° C)	**Y** (")	**N** (< 2 days at 90° C)
Au	≥ 40%	-	**Y** (")	-

Many hours were spent preparing coated nylon samples via standard evaporative coating procedures. The evaporation process occurs in a large vacuum bell jar pumped down to ~$1x10^{-5}$ mm Hg. The high vacuum is needed to ensure a smooth and pure coating. Materials are evaporated in the bell jar isotopically, thereby depositing a layer on any substrate placed in the jar. With this technique I evaporated aluminum, chrome, and silver (both alone and in combination) in layers of various thickness. All samples were evaporated on one side only, which allows one to make bonds on the outer side of the panels. It was determined through trials that a total thickness of approximately 1500Å would ensure that the coating is opaque and therefore optimal for reflection. I also observed that the initial surface quality of the nylon samples had direct effect on the final coated surface quality. In particular, scratches and fingerprints initially on the nylon

samples left large areas of unpolished surface. The effects of this surface 'contamination' on the overall reflectivity were not measured, yet for precaution extreme care ought to be taken in handling the nylon sheet. Now, I will report on the specific tests performed for samples in each medium.

1. Solvent

A variety of small (3" x 3" x .020") samples of coated nylon have been kept in a beaker of xylene for over 3 months at room temperature. In all of the cases, the solvent has appeared to do no damage to the reflective surface and therefore aluminum appears the ideal coating in this medium strictly because of availability and costs. It is expected that light cones placed in a xylene environment would last the full length of the CTF experiment.

2. Deionized Water

It is well known that deionized water vigorously attacks aluminum. This was directly observed in the tests that I performed. At room temperature, the 1500Å layer dissolved on the order of one day. Furthermore, a nylon sample prepared by EMF Corporation that had a ~1000Å Al layer protected by ~1300Å of SiO was similarly attacked. Although the SiO layer does offer some protection, nylon is know to be very permeable to water and, therefore, water diffusing from the uncoated side is likely to be dissolving the aluminum. In any case, it became very clear that Al would not be a solution in deionized water.

Therefore, I concentrated my efforts on silver. Before testing the samples, it was determined that a 100Å coating of

Cr adheres the silver to the nylon samples more effectively than straight silver on nylon. As the chrome had no effect on the external silver surface, this two layer technique was used for nearly all of the test samples. Knowing before hand that silver would not be attacked as quickly as the aluminum, steps were taken to simulate the time scale of the CTF Upgrade. I heated beakers containing the deionized water and the coated nylon samples to accelerate the chemical processes between the two substances. For a rough estimate, one may follow a chemistry rule of thumb which states that the corrosion rate is accelerated by a factor of 2 for every 10°C the environment's temperature is increased. In terms of the CTF Upgrade, 3 days at 90°C would approximate a year of operation in the CTF. Therefore, beakers were placed on a large hot plate that was adjusted until the water temperature was very close to 90°C.

The results were very discouraging. Although the silver is not attacked as quickly as aluminum, it is clear that it will not survive in deionized water. First the reflective surface becomes a bit hazy and then eventually the coating begins to flake off the nylon sample. I also tested a silver sample coated with a 1200Å SiO layer by EMF in the hopes that the SiO would protect the silver from the deionized water. The lifetime of the nylon sample was improved, but again the surface was eventually destroyed. It is most likely that water permeating through the uncoated side is swelling the nylon and in this manner ruining the polished surface. In conclusion, after weeks of evaporations and tests presently I

have developed no procedure to make a nylon sample with a reflective coating that can withstand deionized water. This is a serious problem, particularly because deionized water is the most likely shielding medium to be utilized in the CTF. It is not just a problem associated with the procedure I have developed to fabricate nylon light cones, but pertinent to any method that involving metallic reflective coatings. Further research and effort must be made to address this important problem.

3. Mineral Oil

The same procedures describe for deionized water (i.e. raised temperatures to accelerate the corrosion) were applied to mineral oil. It was observed that samples coated with aluminum only were ruined after ~7 days, but the aluminum sample with an SiO overcoat lasted weeks (30 days) at 90°C. Unprotected silver lasted about 14 days, while the sample with an SiO overcoat shows a promising resistance to corrosion (perfect after 12 days now). In all cases (with the possible exception of the Ag-SiO sample), the samples were eventually corroded which raises some alarm, yet it occurred over a time scale that far exceeds the proposed CTF lifetime Again, it is likely that after several weeks, the mineral oil had permeated the nylon enough to cause it to swell thereby ruining the reflective coating.

F. Costs and Time

Not only must the fabrication technique produce light collectors within strict tolerances that will last for over a

year, the technique must be assembled cost and time effective. As with most products, the costs can be divided between materials and labor. In terms of material costs, the Miles-Mobay Nylon 38F is an expensive copolymer which must be extruded on a custom basis. Using the cost of our first extrusion as a basis, I estimate that the .020" nylon cost for 100 cones (~2500 sq. ft) to be $2.00/sq. ft or $50/cone. The nylon plate needed for the support structure ought to be a comparable price provided the nylon plate is bought from stock in large quantity ($25/cone). The wood mold is very expensive (>$1000 with labor), yet it is a one-time expenditure and therefore not too influential on the overall cost. Other one-time costs such as heat lamps or nylaweld would be marginal significant and are therefore ignored.

The biggest unknown cost of fabrication is the price to coat a cone. If a standard evaporative coating method can be done in-house, the costs are strictly labor. If a vendor is required, however, the costs could be very high (several hundred dollars per cone). The actual cost will be strictly dependent on the type of coating, the final design of the cone (size), and the actual technique employed. These variables all remain very significant questions in the fabrication of the light cones.

The labor costs, as well, can only be roughly assessed. From my experience with the fabrication of the prototypes, I estimate it would take an experienced worker 1 man day to cut and assemble one light cone. I do not believe that skilled labor is necessary, greatly reducing the labor costs.

Therefore, the labor cost of the assembly of one light cone would range from $50 - $100 provided that unskilled labor is sufficient. There are also some machine shop costs involved with producing the support structure, yet they ought to be minimal because many bands and rings can be cut simultaneously.

In terms of production time, the building of a wood mold and the extrusion of the nylon sheet would take about 3 weeks. At that point, the time of fabrication is solely a function of the coating process used and the number of workers assigned to the project. Even if only one wood mold is used, in one 8 hour shift, it is possible that 2 cones could be built. For 100 cones, this would relate to 10 working weeks. Given another mold, or more shifts, this time scale could be greatly reduced.

Chapter 4 - Conclusions

The optimal light collector for the CTF Upgrade design has been demonstrated to be the String Method Cone. Not only does it have nearly 100% transmission out to the edge of the fiducial volume, the steep drop in its transmission at this edge ought to allow for accurate 3 dimensional position resolution. The SM cone, however, is not ideal and it is possible that some modifications will be necessary to further optimize its performance. In particular, I expect that one would define the edge of the fiducial volume to be the point where the 3D transmission-radial curve falls from maximum and not where it supposed to drop ideally.

Although the SM cone is ideal for the CTF Upgrade, the final profile remains to be determined. Many factors need to be addressed first (e.g. number of PMT's, fiducial volume size, actual shielding length for the PMT bases) before the exact parameters for the SM Cone will be determined. The programs presented in this thesis will offer the capability to both create and analyze SM cones for any CTF Upgrade configurations.

In addition, it still remains to be demonstrated that an array of SM cones will produce identifiable patterns of 'hits' near the edge of fiducial volume. Such patterns would allow one to differentiate between events originating at the inner vessel from those actually within the fiducial volume. I expect to create a program which will simulate an spherically

isotropic array of 'N' SM cones in order to investigate these patterns. The results of this research will be published in a separate document.

In terms of fabrication, the Panel Technique offers mixed results. The success observed with the CPC-P demonstrates that the method is easily applicable to a collector of its size. For the full sized SM-P, however, concerns remain over the ease with which the panels can be molded to the shape of the wood mold. The fact that an accurate wood mold has been fabricated, however, will allow for further investigations into the technique.

The issue of reflective coatings also remains unresolved. For a xylene or mineral oil environment, a one sided aluminum coating (perhaps protected by SiO) is a very attractive solution. Arrangements, however, would need to be made with a vendor (e.g. EMF Corporation) for the evaporative coating of large nylon sheets for the SM cone panels unless a very large bell could be procured. With respect to a deionized environment, there is no solution at this stage for coating nylon. I believe the focus should center on preventing the water from permeating the open side of the nylon panels. If this can be achieved, I expect that panels coated with silver would last the duration of the CTF Upgrade.

It also remains to be demonstrated that a fabricated cone actually has the correct optical properties. I don't believe that this is a difficult point, but it will involve the design of an experiment (perhaps a setup involving a He laser) that can test the transmission properties of the collector.

In summation, it is clear that the SM cone is optimal for the CTF Upgrade design. Intensive research needs, however, to be focused on the fabrication of these large cones. It is my opinion that the Panel Technique is a viable option, but large advancements must be made in the upcoming weeks to prove its success. The research performed for this thesis was directed specifically to the CTF Upgrade design. It would be a simple task, however, to generalize its results to many types of radiation collection detectors. The programs presented are easily understood and therefore altered to correspond to other designs. The fabrication methods outlined may also be applicable to other projects, particularly ones with strict radiopurity constraints. The String Method cones, derived from principles pioneered by Winston, are powerful tools which once properly understood can alter the entire outlook of an experiment. With respect to the two designs for the CTF, they may allow one (the CTF Upgrade) to perform serious solar neutrino physics; perhaps even first verification of the day-night effect as predicted by the MSW Theory.

Appendix

This appendix gives a more detailed description of the individual programs that I wrote to investigate the performance of light collectors for the CTF Upgrade. In order to minimize the size of this Appendix, I take full use of the fact that many programs share the same procedures. These procedures are detailed once, and then assumed to be common knowledge in the programs that follow. All code was written in standard ANSI C language. Finally, a working understanding of the CTF Upgrade design is assumed throughout the Appendix.

1. SM Profile - simult_dat.c [Located in /home/prochska/thesis/c/ctfcpc]

The following program calculates the x,y coordinates (in cm) of the String Method cone for a given fiducial volume and shielding length. The program takes the fiducial volume and calculates the effective fiducial volume due to refraction between the solvent (n = 1.5) and the water (n = 1.34) outside the inner vessel. It assumes that the exit aperture - aprim - has a radius of 9.5cm and that the PMT is spherical with an 11cm radius. Figures A.1 and A.2 will be very useful in understanding the following description.
Variables:
reff - Effective fiducial volume (f.v.) = (input fiducial volume) * n_scin/n_water
r2 - Distance from the center of the f.v. to the exit aperture
dcone - Distance from the center of the f.v. to the x,y point on the concentrator profile
y2 - Distance from the center of the f.v. to the center of the photocathode
cathang - Angle off the vertical axis to the point of intersection between the exit aperture and the photocathode
cathrad - Radius of the photocathode = 11cm
thetastop - Limiting angle which indicates the end of the profile. Defined by the ray originating at x[1],y[1] which is tangential to the f.v.
x[1],y[1] - First horizontal, vertical point on the concentrator profile where the exit aperture and PMT meet
x[i],y[i] - Point on the concentrator profile
xlast, ylast - Variables used to hold the last output x,y point
step - The distance between output x,y points
dl - Step size for the individual steps in the loop. The smaller the value of dl, the more accurate the profile
d1 - Distance from the photocathode center to the point on the profile
(See Figure A.1 for theta1, theta2, phi1, phi2, totang and tang)
(See Figure A.2 for sumang)
nmx - Horizontal component of the normal vector
nmy - Vertical component of the normal vector
covg - Coverage of a single SM cone determined as a function of the size of the cone

Description: The program generates the profile in the following manner. It takes x[1],y[1] as its first point and steps a distance dl normal to the PMT. This is the second point on the profile. The program then 'draws' two rays: one tangent to the PMT and one tangent to the FV. It calculates the angle subtended by these two rays, and if we consider a normal vector bisecting this angle, the program calculates the angle off the vertical axis (tang) of the vector perpendicular to this normal. Therefore, if an incoming ray from the FV were to strike a infitessimal mirror at this x,y point on the concentrator that is parallel to the vector defined by tang, the ray would reflect off the mirror tangent to the PMT. The program calculates the next point on the concentrator profile by moving a step size (dl) in the direction of the vector defined by tang. The procedure than repeats, with two new rays drawn tangent to the FV and the PMT, and therefore a new angle tang defining a new 'mirror'.

At some point on the concentrator profile, however, it becomes impossible to draw a ray tangent to the PMT [it occurs when phi1 + theta1 > (π/2 - cathang)]. If that is the case, the program draws a ray to the intersection of the exit aperture and the PMT (the reflection of x[1],y[1] through the axis of symmetry). The calculates sumang and queries whether the end of the concentrator has been reached (is sumang > thetastop?). If it has not, it finds the new angle tang as described above and then calculates the new x,y point on the concentrator.

By varying the size of step and dl, one varies the distance between output points and the accuracy of the profile (the smaller dl is, the smaller the steps between x,y points and therefore the greater the accuracy of the profile).

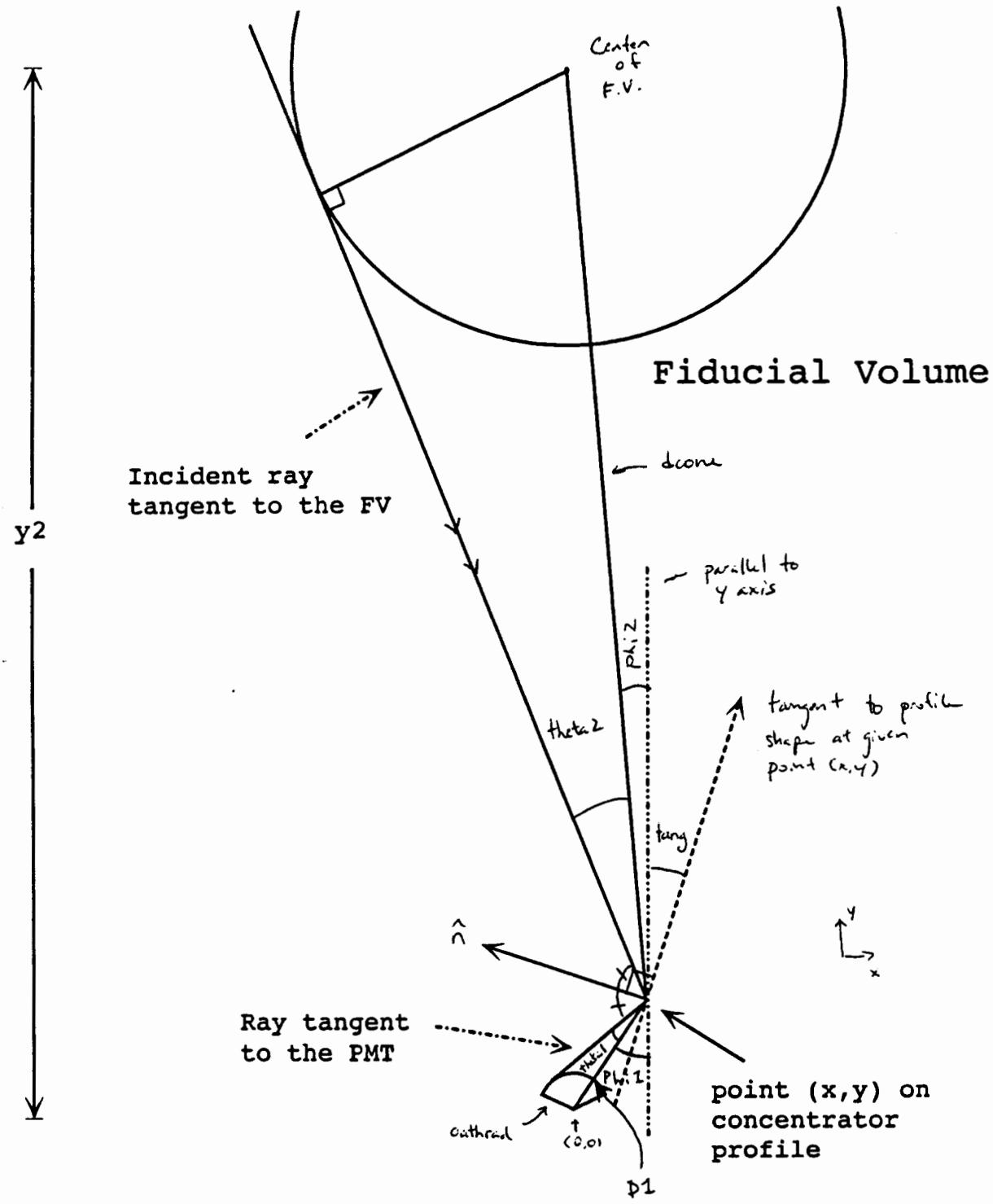

Figure A.1: Explanation of the variables for construction of the lower section of the profile where the SM tangential construction is used. The tangent ray from the FV is reflected so as to be tangent to the PMT.

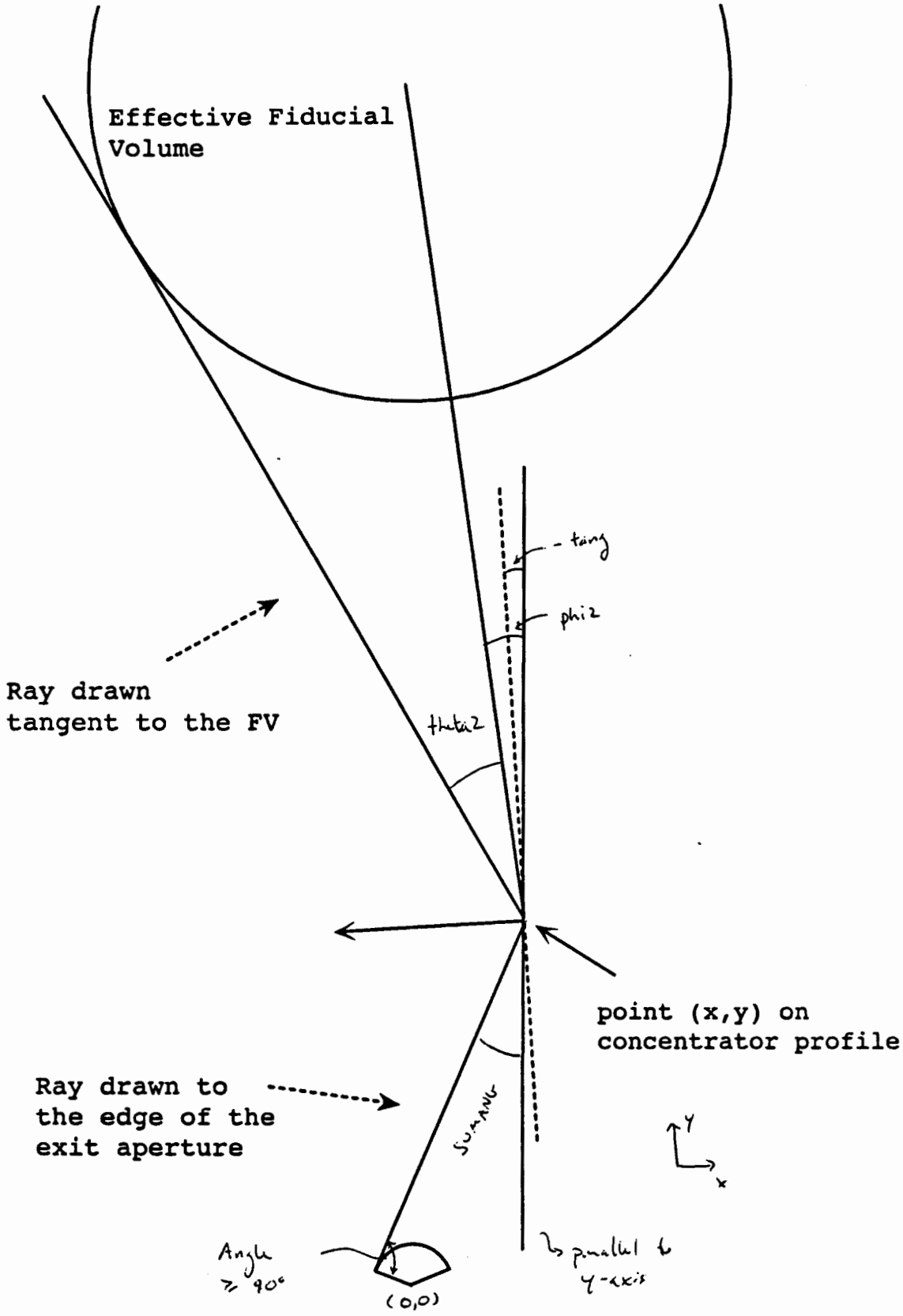

Figure A.2: Depiction of the construction of the upper section of the profile where the CPC-like method is used.

```
/*  This is the program simult_dat.c used to define the profile of the SM
  cone in terms of r,z points.  It is located in /home/prochska/thesis/c/ctfcpc
  A more detailed discussion of the program can be found in Appendix A.1 of
  the 1993 Physics Senior Thesis by Jason X. Prochaska  */

#include <stdio.h>
#include <math.h>
#define aprim 9.5   /*  Radius of the exit aperture  */
#define Pi 3.141592654
#define deg (Pi/180.0)
#define NR_END 1
#define n_scin 1.5   /*  Indices of reflection  */
#define n_water 1.34

void main ()
{
        double rfv, reff, r2, dcone, d1, theta1, theta2, phi1, phi2, totang,
          tang, x[100000], y[100000], diff, thetastop, sumang, cathang, cathrad,
          covg, step, xlast, ylast, nmx, nmy, y2, dl, dpmt;

        int i, j, cones, flag;

/*  Determine inital values of the detector configuration */
        printf("\n Input the radius of the Fiducial Volume (in cm): ");
        scanf("%lf", &rfv);
        /*  reff is the effective fiducial volume with refraction taken into
         account  */
        reff = rfv * n_scin / n_water;
        printf("\n The Effective Fiducial Volume is %.7f cm \n", reff);
        printf("\n How many cones are to be used? ");
        scanf("%ld", &cones);
        printf("\n What is the shielding length?  (in cm) ");
        scanf("%lf", &dpmt);
        r2 = reff + dpmt;

/*  Set initial geometry of the simulation  */
        cathrad = 11.0;
        cathang = asin(aprim/cathrad);
        printf("\n cathang = %.2lf", cathang/deg);
        y2 = r2 + cos(cathang)*cathrad;
        dcone = sqrt(x[1]*x[1] + (y[1] - y2)*(y[1] - y2));
        thetastop = asin(reff/dcone) + asin(x[1]/dcone);

        y[1] = cathrad * cos(cathang);
        x[1] = cathrad * sin(cathang);
        xlast = -100;
        ylast = -100;
        flag = 1;
/*  Set the step lengths for the routine  */
        step = .015;
        dl = 0.003;
/*  Main loop of the program  */
        for (i = 1; i <= 100000; i++)
        {
                d1 = sqrt(x[i]*x[i] + y[i]*y[i]);
                dcone = sqrt(x[i]*x[i] + (y[i] - y2)*(y[i] - y2));

                theta1 = asin(cathrad/d1);
                theta2 = asin(reff/dcone);
                phi1 = asin(x[i]/d1);
                phi2 = asin(x[i]/dcone);
        /*  This conditional determines if the ray can intersect tangentially
          with the PMT.  If so, it does the SM tangential construction.  If not,
          it uses the edge-ray principle to construct a CPC-like section  */
                if (phi1 + theta1 > (90*deg - cathang))
                {
                  /*  SM tangential construction  */
                        totang = (Pi - theta1 - theta2 - phi1 - phi2) * .50;
```

```
                tang = totang + phi1 + theta1 - Pi/2;
            }
            else
            {
              /* CPC-like construction via edge-ray principle  */
                sumang = atan((x[i] + x[1])/(y[i] - y[1]));
                if (sumang < thetastop)
                {
                /*  Breaks if the end of the cone has been reached */
                    flag = 0;
                    break;
                }
                totang = (Pi - sumang - theta2 - phi2) * .50;
                tang = totang + sumang - Pi/2;
            }
 /* Outputs value of point if the step size from the last output point is large enough  */
            if ((x[i] - xlast)*(x[i] - xlast) + (y[i] - ylast) *
              (y[i] - ylast) > (step * step))
            {
                nmx = -cos(tang);
                nmy = sin(tang);
                covg = (cones/2) * (1.0 - cos(phi2)) * 100;
                printf("\n%.8f  %.8f  %.8f", x[i],
                  y[i] - y[1], nmy/nmx);
                xlast = x[i];
                ylast = y[i];
            }
 /*  Calculates the next point on the cone by travelling perpendicular to the
  normal calculated above (defined by tang)  */
            j = i + 1;
            x[j] = x[i] + sin(tang)*dl;
            y[j] = y[i] + cos(tang)*dl;
        }
        if (flag)
        { printf("\n We never reached the end.");
        }
        else
        {
            nmx = -cos(tang);
            nmy = sin(tang);
            covg = 32.0 * (1.0 - cos(phi2)) * 100;

            printf("\n%.8f  %.8f  %.8f", x[i], y[i] - y[1], nmy/nmx);
            printf("\n We have reached the end of the profile. \n");
        }
}
```

2. Ray Tracing

Because the 3D simulations are more complex, I decided to give a detailed account of one of them which applies to the rest of the ray tracing routines. Consider trnrad.c the Rosetta Stone of all of the ray tracing routines that I have developed.

A. trnrad.c - located in /home/prochska/thesis/c/simult/3D

trnrad.c calculates the 3D transmission-radial curve for both CPC and SM cones. It is a Monte Carlo ray trace program that follows the path of individual rays, determining whether or not they strike the PMT. In short, the program considers a 3D fiducial shell in space with variable radius fv_r. A very large number of rays are drawn from the shell to the entrance aperture of the cone. Because of the cylindrical symmetry of the cones and the sphere a simplification is made. The program uses variable z,x on the shell while holding y at 0 always. It then weights the rays of from the shell as a function of the polar angle theta. Therefore, rays originating from near the equator of the shell have more weight than those originating from the poles. This is a logical Monte Carlo simulation as more events will occur within a finite volume taken around the equator than a finite volume around the poles [directly a function of sin(theta)]. One is asked to input a number of variables related to the cone and to the 'depth' of the simulation. In general, the parameters ideal for the CTF Upgrade are assumed, including a simple factor for the refraction of the rays as

they pass from solvent to water. A brief description of the most important variables follows. There are also some detailed comments throughout the code.

(The following variables are best understood by examining the geometry of Figure A.3)
Entrance Aperture Parameters:
aper_area - An array of area elements which make up the aperture entrance. Their size varies with aper_rad.
aper_rad - An array of aperture entrance radius values ranging from ap_max to (ap_max)(.75)/stps_apr.
ap_max - Maximum size of the entrance aperture
stps_apr - Number of radial steps the aperture radius is divided into
stps_ang - Number of angular steps that the entrance aperture is divided into
Light Collector Parameters:
th_max - Extreme incident angle for the CPC cone
r_cp - An array of r values for the CPC cone defined by th_max. Each r value corresponds to an increment of .02cm in z
r_sm, z_sm - An array of r,z values that describe the SM cone profile. These values are input by the user (I used the values determined by simult_dat.c)
rtio_sm - Array of n_z/n_r values defining the normal at each point on the SM profile
nln_sm - Number of lines of input data for the SM cone

hts - Counter used to keep track of the number of photons which reach the PMT
stps_spz - Number of vertical steps that the fiducial shell is divided into
stps_spr - Number of radial steps that the effective fiducial radius is divided into

Variables in Main
L_cone - Length of the light collector
d_apsp - Distance from the center of the sphere to the entrance aperture
fv_eff - Effective fiducial radius with refraction taken into account. This is the radius that the light collectors ought to be designed to look at
fv_r - The radius of a given fiducial shell
N_ev - Number of photon events examined weighted by the Monte Carlo sin(theta) factor
refs - Maximum number of reflections that a ray will undergo before being considered a miss

The comments found within the code are quite extensive and therefore only a brief synopsis of the program is given here. Essentially, the program draws a ray from a fiducial shell (radius fv_r) a point on the entrance aperture with the origin of the simulation taken to be the center of the exit aperture of the light cone. The entrance aperture is simulated by a mesh of finite area elements, each with a known position (Figure A.3). The program determines the direction of the ray, and than through a series of steps calculates whether the path of the ray as it reflects of the light collector eventually hitting the PMT or reflecting out the light cone. Keeping track of the number of 'hits' it than calculates the percentage of rays which were transmitted from the fiducial shell to the PMT, and therefore for a given radius shell, we have a 3D transmission-radial curve for the CTF Upgrade.

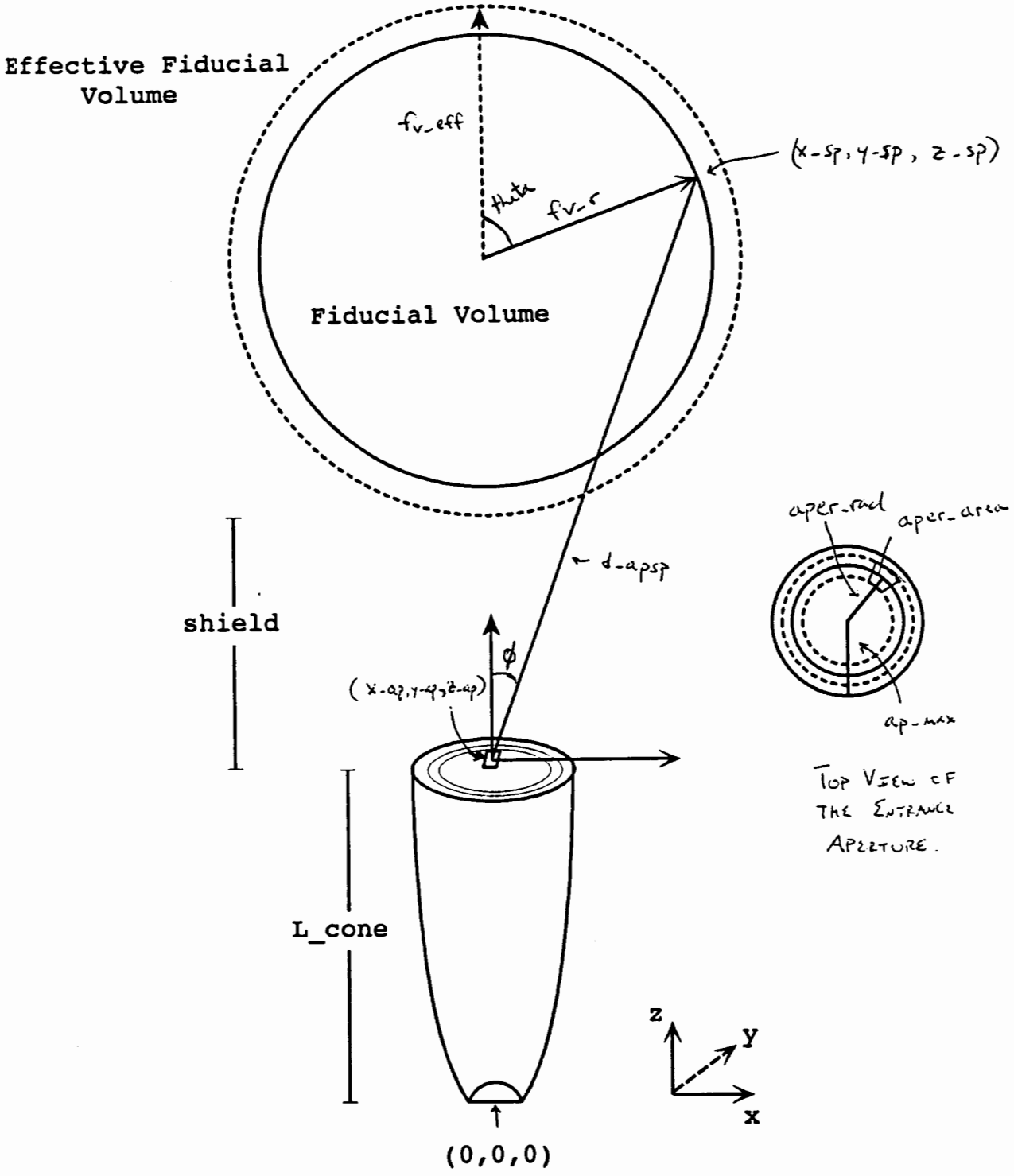

Figure A.3: Description of some of the variables used in trnrad.c

```
/*  This is trnrad.c located in /home/prochska/thesis/c/simult/3D
    This program calculates the 3D transmission-radial curve for both the
    CPC and SM cones.  A more detailed write-up of the program can be found in
    Appendix 2.A of the 1993 Senior Physics Thesis by Jason X. Prochaska
*/
#include <stdio.h>
#include <math.h>
#define aprim 9.5       /*  Exit aperture radius in cm  */
#define Pi 3.141592654
#define deg (Pi/180)
#define NR_END 1
#define n_scin 1.5      /*  Indices of refraction */
#define n_water 1.34
#define refl_cof .90
#define z_pmt -sqrt(121 - 9.5*9.5)  /* Z coordinate of the PMT  */
#define pmt_rad 11.0

/*  Global Variables  */
double *aper_area, *aper_rad, ap_max, th_max, *r_sm, *z_sm, *rtio_sm, *r_cp,
        *hts;
long stps_apr, stps_spz, stps_spr, stps_ang, nln_sm;
int length;
/*  Global Function Declarations  */
void cp_array(double L_cone);
int pmt(double vt_x, double vt_y, double vt_z, double x_ap, double y_ap, double
        z_ap);
double inter_cp(double x_cp, double y_cp, double z_cp, double vt_x,
        double vt_y, double vt_z);
double inter_sm(double x_st, double y_st, double z_st, double vx, double vy,
            double vz, int *ind_sm);
void norm_cp(double *vt_x, double *vt_y, double *vt_z, double x, double y,
        double z);
void norm_sm(int ind_sm, double x, double y, double *vx, double *vy, double *vz);
double *dvector(long nl, long nh);
void nrerror(char error_text[40]);

/*  Main program  */
void main ()
{
        double L_cone, d_apsp, fv_eff, fv_r, N_ev, sum_Ec, *Ec, z, gamma;
        int i, k, flag, npmt, refs;
/*  Function Declarations  */
void inpt(double *L_ip, double *d_ip, double *fv_ip, int *fl_ip, int *npm_ip,
        int *refs);
void init();
void calc_Ec(int k, double fv_r, double L_cone, int fl_cn, double d_apsp,
        double *N_ev, int refs);
/*  Setting Initial Data  */
        inpt(&L_cone, &d_apsp, &fv_eff, &flag, &npmt, &refs);
        stps_ang = 30;
        aper_area = dvector(1, stps_apr - 1);
        aper_rad = dvector(1, stps_apr - 1);
        Ec = dvector(1, stps_spr + 20);
        hts = dvector(1, stps_spr + 20);
        init();
        printf("\nmade it through init\n");
/*  Main Loop of Program  */
        k = 0;
        do {
                k++;
                /*  Define fiducial volume radius  */
                fv_r = fv_eff*k/(stps_spr);
                /*  Calculate the number of hits and the total number of
                  incident rays  */
                calc_Ec(k, fv_r, L_cone, flag, d_apsp, &N_ev, refs);
                /*  Output  */
                printf("\n");
```

```
                printf("fv radius = %.4lf, percent = %.2lf", fv_r,
                  hts[k]*100 / N_ev);
        } while(k < stps_spr + 9);

}

void inpt(double *L_ip, double *d_ip, double *fv_ip, int *fl_ip, int *np_ip,
        int *refs)
  /*  Routine used to define many of the intial values  */
{
        double fv_rad, shield, th, fv_ff, L_cn, r, z, ratio;
        int flag,j;
        printf("\nHow many steps do you want to consider in the aperture entrance?   " );
        scanf("%ld", &stps_apr);
        printf("\nHow many vertical steps do you want to consider in the sphere?   ");
        scanf("%ld", &stps_spz);
        printf("\nWhat is the Fiducial Volume without refraction? (in cm)   ");
        scanf("%lf", &fv_rad);
        fv_ff = fv_rad*n_scin/n_water;
        *fv_ip = fv_ff;
        printf("\nHow many radial steps do you want to consider in the sphere?  ");
        scanf("%ld", &stps_spr);
        printf("\nWhat is the shielding length?  (in cm)  ");
        scanf("%lf", &shield);
        printf("\nWhat is the length of the light collector? (in cm)  ");
        scanf("%lf", &L_cn);
        *L_ip = L_cn;
        *d_ip = shield + fv_ff - L_cn;
        printf("\nWhat is the entrance aperture radius?  (in cm)   ");
        scanf("%lf", &ap_max);
        printf("\nHow many cones are there?  ");
        scanf("%d", np_ip);
        printf("\nHow many reflections do you want to consider?  ");
        scanf("%d", refs);
        printf("\nIs the cone an SM or CPC?  (SM = 1, CPC = 0)  ");
        scanf("%d", &flag);
        if (flag)
        {
        /*  This inputs the profile of the SM cone including the normal vector
        */
                printf("How many lines of data are there?  ");
                scanf("%ld", &nln_sm);
                r_sm = dvector(1, nln_sm);
                z_sm = dvector(1, nln_sm);
                rtio_sm = dvector(1, nln_sm);
                for(j = 1; j <= nln_sm; j++)
                {
                  scanf("%lf %lf %lf", &r, &z, &ratio);
                  r_sm[j] = r;
                  z_sm[j] = z;
                  rtio_sm[j] = ratio;
                }
                *fl_ip = 0;
        }
        else
        {
                printf("\nWhat is theta max? (in degrees)   ");
                scanf("%lf", &th);
                th_max = th*deg;
                cp_array(L_cn);
                *fl_ip = 1;
        }
}
void cp_array(double L_cone)
  /*  This routine creates the CPC array as a function of the length of the
    light collector and extreme angle th_max.  The values are calculated from
    equation 4.6 of Winston's text[30]  */
{
```

```
        double cs, sn, a, b, c, z, cs2, sn2, cssn, b0, c1, c0, radic;
        int i;
        cs = cos(th_max);
        sn = sin(th_max);
        cs2 = cs*cs;
        sn2 = sn*sn;
        cssn = cs*sn;
        b0 = 2*aprim*(1+sn)*(1+sn);
        c1 = (-2)*aprim*cs*(2+sn);
        c0 = (-1)*aprim*aprim*(1+sn)*(3+sn);
        length = ceil(L_cone*50);
        r_cp = dvector(1, length);

        for (i=1; i <= length; i++)
        {
                z = i*.02;
                a = cs2;
                b = 2*z*cssn + b0;
                c = sn2*z*z + c1*z + c0;
                radic = b*b - 4*a*c;
                if (radic < 0) nrerror("Imaginary root in cp_array");
                r_cp[i] = (-b + sqrt(radic))/(2*a);
        }
        printf("\n i = %u", i);
}

void init()
 /*  This procedure initializes the values of the aperture entrance mesh.
    Note that the first and last value of aper_rad and aper_area need special
    consideration so that the mesh remains within ap_max without double
    counting anything   */
{
        int i,j;

        aper_rad[1] = (ap_max)*(.75)/(stps_apr);
        aper_rad[stps_apr - 1] = ap_max*(1-(.75)/(stps_apr));
 /* aper_area is a finite elment of area in the aperture entrance defined
    by 2*Pi*r*dr    */
        aper_area[1] = Pi*((1.5)*ap_max/(stps_apr))*((1.5)*ap_max/(stps_apr))/
                stps_ang;
        aper_area[stps_apr - 1] = Pi*(ap_max*ap_max*(1-(1 - 1.5/stps_apr)*
                (1 - 1.5/stps_apr)))/stps_ang;
        for (i = 2; i < (stps_apr - 1); i++)
        {
                aper_rad[i] = i*ap_max/(stps_apr);
                aper_area[i] = 2*Pi*(ap_max/stps_apr)*(i*ap_max/stps_apr)
                        /stps_ang;
        }
}

void calc_Ec(int k, double fv_r, double L_cone, int fl_cn, double d_apsp,
                double *N_ev, int refs)
 /*  This procedure calculates the number of rays that hit the PMT as a function
   of the fiducial shell radius.  */
{
        double x_sp, y_sp, z_sp, x_ap, y_ap, z_ap, theta, alph, csphi, d_hyp,
           tp, vt_x, vt_y, vt_z, x_in, y_in, z_in, sum_Ec, Er, Ec, vo_x, vo_y,
           vo_z, ab_cof;
        int i, j, l, m, flag, ind_sm, flag2;

        sum_Ec = 0;
        *N_ev = 0;
        hts[k] = 0;
        for (i=1; i < stps_spz; i++)
        {
  /*  x_sp, y_sp, z_sp define a point on a fiducial volume shell.  Theta is
     the azimuthal angle, a strict function of z_sp.  Because of cylindrical
     symmetry in the fiducial shell and the light collector, y_sp = 0 always. */
```

```
        z_sp = ((fv_r)*i*(2.0)/(stps_spz) - fv_r) + d_apsp + L_cone;
        theta = acos(fabs(z_sp - d_apsp - L_cone)/fv_r);
        y_sp = 0;
        x_sp = (fv_r)*sin(theta);
        for (j = 1; j < stps_apr; j++)
        {
                for (l = 0; l < stps_ang; l++)
                {
/*  N_ev keeps track of the number of events in the simulation.  The
    sin(theta) - a Monte Carlo weithing factor - is related
    to the cylindrical symmetric approximation I made in defining
    y_sp = 0 always.  */
                        *N_ev += sin(theta);
                        alph = 2*l*Pi/stps_ang;
        /*  x_ap, y_ap, z_ap define a point on the entrance aperture.
          vt_x, vt_y, vt_z define the vector of the ray as it travels
          throughout the simulation.  */
                        x_ap = aper_rad[j]*cos(alph);
                        y_ap = aper_rad[j]*sin(alph);
                        z_ap = L_cone;
                        vt_x = x_ap - x_sp;
                        vt_y = y_ap;
                        vt_z = z_ap - z_sp;
        /* Checks to see if the ray goes directly to the PMT */
                        flag = pmt(vt_x, vt_y, vt_z, x_ap, y_ap, z_ap);
                        m = 0;
                        if (flag) flag2 = 1;
                        else
                        {
        /*  As the ray did not hit the PMT, we look for the intersection
          of the ray with the light cone and to calculate the ray
          that reflects from the surface  */
                          x_in = x_ap;
                          y_in = y_ap;
                          z_in = z_ap;
                          if (fl_cn)
                          { do
                            {
                /*  This procedure is used for the CPC cones  */
                            m++;
        /*  tp is a parameter which describes a new ray via the
          equation r_new = r_old + tp*vr  */
                            tp = inter_cp(x_in, y_in, z_in,
                              vt_x, vt_y, vt_z);
                /*  Calculate the new point of origin of the ray */
                            x_in = x_in + tp*vt_x;
                            y_in = y_in + tp*vt_y;
                            z_in = z_in + tp*vt_z;
                /*  Find the new vector describing the ray's motion */
                            norm_cp(&vt_x, &vt_y, &vt_z, x_in, y_in, z_in);
                /*  Determine if this new ray hits the PMT  */
                            flag = pmt(vt_x, vt_y, vt_z, x_in, y_in,
                                    z_in);
                            if (flag || z_in < .0001)
                            {
                              flag2 = 1;
                              break;
                            }
        /* If the ray travels upward, it will always escape.
          Therefore, end the routine.  Also break if we have reflected
          our maximum number of times (refs)  without intersection */
                            if (vt_z > 0 || m == refs)
                            {
                              flag2 = 0;
                              break;
                            }
                          } while (m < 1000);
                        }
```

```
                                        else
                                        {  do
                                           {
                                /*  Procedure for the SM cone exactly the same as
                                  that for the CPC cone  */
                                                m++;
                                                tp = inter_sm(x_in, y_in, z_in, vt_x,
                                                   vt_y, vt_z, &ind_sm);
                                                x_in = x_in + tp*vt_x;
                                                y_in = y_in + tp*vt_y;
                                                z_in = z_in + tp*vt_z;
                                                norm_sm(ind_sm, x_in, y_in, &vt_x,
                                                        &vt_y, &vt_z);
                                                flag = pmt(vt_x, vt_y, vt_z, x_in, y_in,
                                                        z_in);
                                                if (flag || z_in < .008)
                                                {
                                                  flag2 = 1;
                                                  break;
                                                }
                                                if (vt_z > 0 || m == refs)
                                                {
                                                  flag2 = 0;
                                                  break;
                                                }
                                           } while (m < 1000);
                                        }
                                      }
                                      if (flag2)
                                      {
                                      /*  Increase the number of hits by 1 weighted
                                       by our Monte Carlo factor  */
                                        hts[k] += 1*sin(theta);
                                      }
                                }
                        }
                }
        return;
}

int pmt(double vt_x, double vt_y, double vt_z, double x_pt, double y_pt, double
        z_pt)
  /*  This routine determines whether a ray hits the PMT directly  */
{
        double a, b, c, z1, z2, radic;
        int flag;
  /*  a,b,c are the parameter of the quadratic roots for the equation z(ray) - z(pmt) = 0  */
        a = vt_x*vt_x + vt_y*vt_y + vt_z*vt_z;
        b = 2*(x_pt*vt_x + y_pt*vt_y + (z_pt - z_pmt)*vt_z);
        c = x_pt*x_pt + y_pt*y_pt + (z_pt - z_pmt)*(z_pt - z_pmt)
                - pmt_rad*pmt_rad;

        radic = b*b - 4*a*c;
        if (radic < 0) flag = 0;
        else
        {
                z1 = ((-b + sqrt(radic))/(2*a))*vt_z + z_pt;
                z2 = ((-b - sqrt(radic))/(2*a))*vt_z + z_pt;
        /*  If an intersection occurs above the base of the light collector
          (z=0) we have a hit!  */
                if (z1 > 0 || z2 > 0) flag = 1;
                else flag = 0;
        }
        return flag;
}

double inter_cp(double x_cp, double y_cp, double z_cp, double vt_x, double vt_y,
                double vt_z)
```

```
/* This finds the intersection of a ray with the CPC cone using the CPC array
   described above. The routine calculates the radial difference (total x,y)
   between the ray and cone. If this distance is small enough, it stores the
   value and calculates the difference for the next point. If this absolute
   difference is smaller than the previous one, the program continues. If not
   it returns the parameter t_last of the previous point */
{
        int i, flag;
        double  r1, r2, t, tlast, x, y, z, diff, dlast, MIN, pass, dum;

        dlast = 2*aprim;
        MIN = (.05)*ap_max/stps_apr;
        flag = 1;
        /* The search begins at the bottom of the CPC cone */
        for(i=0; i < 40000; i++)
        {
                if (i == length) nrerror ("Never found intersection in inter_cp");
                z = i * .02;
                t = (z - z_cp)/vt_z;
                x = x_cp + vt_x*t;
                y = y_cp + vt_y*t;
                r1 = sqrt(x*x + y*y);
                r2 = r_cp[i];

                diff = r1 - r2;
  /* flag keeps track of whether we have come within MIN of the light
     collector yet */
                if (flag)
                {
                  if (fabs(diff) < MIN)
                    {
                        dlast = diff;
                        tlast = t;
                        flag = 0;
                        continue;
                    }
                  else    continue;
                }
                if (fabs(diff) > fabs(dlast)) {  break;}
                else
                {
                   dlast = diff;
                   tlast = t;
                   continue;
                }
        }
        return(tlast);
}

double inter_sm(double x_st, double y_st, double z_st, double vx, double vy,
            double vz, int *ind_sm)
  /*  Essentially the same routine as described for inter_cp  */
{
        double r1, r2, diff, t, x, y, z, dlast, tlast, MIN;
        int i, flag;

        flag = 1;
        MIN = (.05)*ap_max/stps_apr;
        for (i = 1; i <= nln_sm + 1; i++)
        {
                if (i == nln_sm + 1) nrerror("never intersected in inter_sm");
                z = z_sm[i];
                t = (z - z_st)/vz;
                x = x_st + vx*t;
                y = y_st + vy*t;
                r1 = sqrt(x*x + y*y);

                diff = r_sm[i] - r1;
```

```
                if (flag)
                {
                  if (fabs(diff) < MIN)
                  {
                        dlast = diff;
                        tlast = t;
                        flag = 0;
                        continue;
                  }
                  else    continue;
                }
                if (fabs(diff) > fabs(dlast)) { break;}
                else
                {
                   dlast = diff;
                   tlast = t;
                   continue;
                }
        }

  /*  ind_sm defined the point of intersection so that the normal can be found
        in norm_sm  */
        *ind_sm = i - 1;
        return(tlast);
}

void norm_cp(double *vt_x, double *vt_y, double *vt_z, double x, double y,
        double z)
  /*  This procedure calculates the normal vector of the CPC cone and via the
    law of reflection determines the new direction vector of the ray  */
{
        double Fx, Fy, Fz, r, n_norm, rdotn, cs, sn, cs2, sn2, dum;

        cs = cos(th_max);
        sn = sin(th_max);
        cs2 = cs*cs;
        sn2 = sn*sn;
        r = sqrt(x*x + y*y);
   /*  Components of the unnormalized normal vector  */
        Fx = -(2*x*cs2 + 2*x*z*cs*sn/r + 2*aprim*(1+sn)*(1+sn)*x/r);
        Fy = -(2*y*cs2 + 2*y*z*cs*sn/r + 2*aprim*(1+sn)*(1+sn)*y/r);
        Fz = -(2*z*sn2 + 2*r*cs*sn - 2*aprim*cs*(2+sn));
  /*  Magnitude of the normal vector  */
        n_norm = sqrt(Fx*Fx + Fy*Fy + Fz*Fz);
        rdotn = (*vt_x*Fx + *vt_y*Fy + *vt_z*Fz)/n_norm;
  /*  Components of the new ray found via the law of reflection  */
        *vt_x = *vt_x - 2*rdotn*Fx/n_norm;
        *vt_y = *vt_y - 2*rdotn*Fy/n_norm;
        *vt_z = *vt_z - 2*rdotn*Fz/n_norm;
}

void norm_sm(int ind_sm, double x, double y, double *vx, double *vy, double *vz)
  /*  This procedure calculates the normal vector of the CPC cone and via the
    law of reflection determines the new direction vector of the ray  */
{
        double nx, ny, nz, nr, n_norm, rdotn, beta, ratio;

  /*  The normal vector is input in terms of r,z */
        ratio = rtio_sm[ind_sm];
        beta = atan2(y,x);
        nz = 1.0;
        nr = nz/ratio;
  /*  nr is decomposed into nx, ny from the x,y position on the profile  */
        nx = nr*cos(beta);
        ny = nr*sin(beta);
  /*  Magnitude of the normal vector  */
        n_norm = sqrt(nx*nx + ny*ny + nz*nz);
        rdotn = (*vx*nx + *vy*ny + *vz*nz)/n_norm;
```

```
  /*  Components of the new ray found via the law of reflection  */
        *vx = *vx - 2*rdotn*nx/n_norm;
        *vy = *vy - 2*rdotn*ny/n_norm;
        *vz = *vz - 2*rdotn*nz/n_norm;
}

double *dvector(long nl, long nh)
  /* Defines an array  */
{
    double *v;
    v = (double *) malloc((size_t)((nh - nl + 1 +
NR_END)*sizeof(double)));
    if (!v) nrerror("allocation funcetion in dmatrix()");
    return v - nl + NR_END;
}

void nrerror(char error_text[40])
/* Numerical Recipes standard error handler */
{
  printf("Break in code error...\n");
  printf("%s \n",error_text);
  printf("...now exiting to system...\n");
  exit(1);
}
```

B. tr2d.c - located in /home/prochska/thesis/c/simult/2D

This program is the 2D counterpart of trnrad.c. It calculates the 2D transmission-radial curve for both the CPC and SM cones. Instead of a fiducial shell, there is a fiducial circle from which all rays are drawn to a 2D entrance aperture. The PMT is no longer spherical, but a 120° arc of an 11cm radius circle. Any other significant differences between this program and trnrad.c are noted in the code itself.

```
/*  This is program tr2d.c located in prochska/thesis/c/simult/2D.
  It calculates the 2D transmission-radial curve for both the CPC and SM cones.
  A more detailed description of this program can be found in Appendix 2.B
  of the 1993 Senior Physics thesis by Jason X. Prochaska.  This program is
  very similar to trnrad.c located in prochska/thesis/c/simult/3D.  Within
  that program are many comments pertinent to this program.
*/

#include <stdio.h>
#include <math.h>

#define aprim 9.5
#define Pi 3.141592654
#define deg (Pi/180)
#define NR_END 1
#define n_scin 1.5
#define n_water 1.34
#define refl_cof .90
#define z_pmt -sqrt(121 - 9.5*9.5)

/* Global Variables */
double *aper_rad, ap_max, *r_sm, *z_sm, *rtio_sm, th_max, *r_cp;
long stps_apr, stps_spz, stps_spr, nln_sm;
int length;
/*  Global Function Declarations  */
void cp_array(double L_cone);
int pmt(double vr, double vz, double r_pt, double z_pt);
double inter_cp(double r_ct, double z_cp, double vr, double vz);
double inter_sm(double r_st, double z_st, double vr, double vz, int *ind_sm);
void norm_cp(double *vt_r, double *vt_z, double r, double z);
void norm_sm(int ind_sm, double r, double *vr, double *vz);
double *dvector(long nl, long nh);
void nrerror(char error_text[40]);
/* Main Program */
void main ()
```

```
{
        double L_cone, fv_eff, d_apsp, fv_r, *Er;
        int refs, flag, k;
/* Function Declarations */
void inpt(double *fv_ip, double *L_ip, double *d_ip, int *refs, int *fl_ip);
void init();
double calc_Er(int k, double fv_r, double L_cone, int fl_cn, double d_apsp,
        int refs);

/* Setting Initial Data */
        inpt(&fv_eff, &L_cone, &d_apsp, &refs, &flag);
        aper_rad = dvector(1, stps_apr - 1);
        Er = dvector(1, stps_spr + 25);
        init();
        printf("\n Made it through init \n");
/*  This is the Main Loop of the program  */
        k = 0;
        do  {
                k++;
        /*  fv_r is the fiducal circle radius  */
                fv_r = k*fv_eff/(stps_spr);
        /*  Er[k] represents the percentage of rays that hit the PMT after as
          many as (refs) reflections */
                Er[k] = calc_Er(k, fv_r, L_cone, flag, d_apsp, refs);
        /*  Output */
                printf("\nfv rad = %.4lf  Percentage = %.2lf", fv_r, Er[k]*100);
        } while(k < stps_spr + 9);
}

void inpt(double *fv_ip, double *L_ip, double *d_ip, int *refs, int *fl_ip)
/*  Input procedure  */
{
        double fv_rad, shield, th, fv_ff, L_cn, r, z, ratio;
        int flag,j;
        printf("\nHow many steps are in the aperture entrance?   " );
        scanf("%ld", &stps_apr);
        printf("\nHow many vertical steps in the circle?   ");
        scanf("%ld", &stps_spz);
        printf("\nWhat is the Fiducial Volume? (in cm)   ");
        scanf("%lf", &fv_rad);
        fv_ff = fv_rad*n_scin/n_water;
        *fv_ip = fv_ff;
        printf("\nHow many radial steps in the circle?  ");
        scanf("%ld", &stps_spr);
        printf("\nWhat is the shielding length?  (in cm)  ");
        scanf("%lf", &shield);
        printf("\nWhat is the length of the cone? (in cm)  ");
        scanf("%lf", &L_cn);
        *L_ip = L_cn;
        *d_ip = shield + fv_ff - L_cn;
        printf("\nWhat is the aperture radius?  (in cm)   ");
        scanf("%lf", &ap_max);
        printf("\nHow many reflections do you want to consider?  ");
        scanf("%d", refs);
        printf("\nIs the cone an SM or CPC?  (SM = 1, CPC = 0)  ");
        scanf("%d", &flag);
        if (flag)
        {
                printf("How many lines of data are there?  ");
                scanf("%ld", &nln_sm);
                r_sm = dvector(1, nln_sm);
                z_sm = dvector(1, nln_sm);
                rtio_sm = dvector(1, nln_sm);
                for(j = 1; j <= nln_sm; j++)
                {
                  scanf("%lf %lf %lf", &r, &z, &ratio);
                  r_sm[j] = r;
                  z_sm[j] = z;
```

```
                rtio_sm[j] = ratio;
            }
            *fl_ip = 0;
        }
        else
        {
            printf("\nWhat is theta max? (in degrees)   ");
            scanf("%lf", &th);
            th_max = th*deg;
            cp_array(L_cn);
            *fl_ip = 1;
        }
}

void init()
{
        int i;

        aper_rad[1] = (ap_max)*(.75)/(stps_apr);
        aper_rad[stps_apr - 1] = ap_max*(1-(.75)/(stps_apr));
        for (i = 2; i < (stps_apr - 1); i++)
        {
            aper_rad[i] = i*ap_max/(stps_apr);
        }
}

double calc_Er(int k, double fv_r, double L_cone, int fl_cn, double d_apsp,
        int refs)
 /*  This procedure calculates the transmission for a given fiducial radius */
{
        double r_sp, z_sp, r_ap, z_ap, theta, phi, d_hyp, tp, vt_r, vt_z, r_in,
            z_in, N_ev, sum_Er;
        int i, j, l, m, flag, ind_sm, flag2, coeff_rad;

        sum_Er = 0;
        N_ev = 0;

        for (i=1; i < stps_spz; i++)
        {
        /*  z_sp and r_sp are points on the fiducial circle, theta is the
          polar angle defined by z_sp.  r_ap and z_ap define a point on the
          aperutre entrance.  vt_z anr vt_r are the vertical and horizontal
          components of the incident ray.  */
            z_sp = ((fv_r)*i*(2.0)/(stps_spz) - fv_r) + d_apsp + L_cone;
            theta = acos(fabs(z_sp - d_apsp - L_cone)/fv_r);
            r_sp = (fv_r)*sin(theta);

            for (j = 1; j < stps_apr; j++)
            {
             for (l = -1; l < 2; l += 2)
             {
                coeff_rad = (-1)*l;
                r_ap = coeff_rad*aper_rad[j];
                z_ap = L_cone;
                vt_r = r_ap - r_sp;
                vt_z = z_ap - z_sp;
            /*  Does the ray hit the PMT directly?? */
                flag = pmt(vt_r, vt_z, r_ap, z_ap);
                if (flag) flag2 = 1;
            /*  If not, find where it goes...  */
                else
                {
                  m = 0;
                  r_in = r_ap;
                  z_in = z_ap;

                  if (fl_cn)
                  { do
```

```
                {
        /* This is the same routine found in trnrad.c  */
                        m++;
                        tp = inter_cp(r_in, z_in, vt_r, vt_z);
                        r_in = r_in + tp*vt_r;
                        z_in = z_in + tp*vt_z;
                        norm_cp(&vt_r, &vt_z, r_in, z_in);
                        flag = pmt(vt_r, vt_z, r_in, z_in);
                        if (flag)
                        {
                          flag2 = 1;
                          break;
                        }
        /* If the ray travels upward, it will always escape. */
                        if (vt_z > 0 || m == refs)
                        {
                          flag2 = 0;
                          d_hyp = sqrt((r_sp - r_ap)*(r_sp - r_ap) +
                             (z_sp - z_ap)*(z_sp - z_ap));
                          phi = acos(((z_sp - z_ap)/d_hyp));
                          break;
                        }
                    } while (m < 1000);
                  }
                  else
                  {  do
                     {
                        m++;
                        tp = inter_sm(r_in, z_in, vt_r, vt_z,
                                &ind_sm);

                        r_in = r_in + tp*vt_r;
                        z_in = z_in + tp*vt_z;
                        norm_sm(ind_sm, r_in, &vt_r, &vt_z);
                        flag = pmt(vt_r, vt_z, r_in, z_in);
                        if (flag || z_in < .008)
                        {
                          flag2 = 1;
                          break;
                        }
                        if (vt_z > 0 || m == refs)
                        {
                          flag2 = 0;
                          break;
                        }
                    } while (m < 1000);
                  }
                }
        /* sum_Er keeps track of the number of rays which hit the PMT,
           while N_ev is the total number of rays.  Each is weighted
           by Monte Carlo factor [sin(theta)] which underscores the fact
           that more events will occur along the equator of the fiducial
           circle than at the poles  */

                if (flag2)
                {
                        sum_Er += sin(theta);
                        N_ev += sin(theta);
                }
                else N_ev += sin(theta);
              }
            }
        }
        return sum_Er/N_ev;
}

int pmt(double vr, double vz, double r_pt, double z_pt)
/* This procedure determines if the ray hits the PMT without reflection */
```

```
{
        double a, b, c, z1, z2, radic;
        int flag;

        a = vr*vr + vz*vz;
        b = 2*(r_pt*vr + (z_pt - z_pmt)*vz);
        c = r_pt*r_pt+ (z_pt - z_pmt)*(z_pt - z_pmt) - 121;

        radic = b*b - 4*a*c;
        if (radic < 0) flag = 0;
        else
        {
                z1 = ((-b + sqrt(radic))/(2*a))*vz + z_pt;
                z2 = ((-b - sqrt(radic))/(2*a))*vz + z_pt;
                if (z1 > 0 || z2 > 0) flag = 1;
                else flag = 0;
        }
        return flag;
}

double inter_cp(double r_ct, double z_cp, double vr, double vz)
/*  This routine finds the intersection of the ray with a CPC cone  */
{
        int i, flag;
        double  r1, r2, t, tlast, z, diff, dlast, MIN, pass, dum;

        dlast = 2*aprim;
        MIN = (.05)*ap_max/stps_apr;
        flag = 1;
        /*  The search for intersection starts at the bottom of the CPC cone  */
        for(i=0; i < 40000; i++)
        {
                if (i == length) nrerror ("Never found intersection in inter_cp");
                z = i * .01;
                t = (z - z_cp)/vz;
                r1 = fabs(r_ct + vr*t);
                r2 = r_cp[i];
                diff = r1 - r2;
                if (flag)
                {
                  if (fabs(diff) < MIN)
                    {
                        dlast = diff;
                        tlast = t;
                        flag = 0;
                        continue;
                    }
                  else    continue;
                }
                if (fabs(diff) > fabs(dlast)) { break;}
                else
                {
                   dlast = diff;
                   tlast = t;
                   continue;
                }
        }
        return(tlast);
}

double inter_sm(double r_st, double z_st, double vr, double vz, int *ind_sm)
/*  This procedure calculates the intersection of a ray with an SM cone  */
{
        double r1, r2, diff, t, z, dlast, tlast, MIN;
        int i, flag;

        flag = 1;
        MIN = (.05)*ap_max/stps_apr;
```

```
        for (i = 1; i <= nln_sm + 1; i++)
        {
                if (i == nln_sm + 1) nrerror("never intersected in inter_sm");
                z = z_sm[i];
                t = (z - z_st)/vz;
                r1 = fabs(r_st + vr*t);
                r2 = r_sm[i];
                diff = r2 - r1;
                if (flag)
                {
                  if (fabs(diff) < MIN)
                  {
                        dlast = diff;
                        tlast = t;
                        flag = 0;
                        continue;
                  }
                  else    continue;
                }
                if (fabs(diff) > fabs(dlast)) { break;}
                else
                {
                  dlast = diff;
                  tlast = t;
                  continue;
                }
        }
        *ind_sm = i - 1;
        return(tlast);
}

void norm_cp(double *vt_r, double *vt_z, double r, double z)
/*  This routine calculates the new ray that results in a reflection of the
  CPC cone  */
{
        double Fr, Fz, n_norm, rdotn, cs, sn, cs2, sn2, flag;

        cs = cos(th_max);
        sn = sin(th_max);
        cs2 = cs*cs;
        sn2 = sn*sn;
        if (r<0)
        {
                r = r*(-1);
                flag = 1;
        }
        else flag = 0;
        Fr = -(2*r*cs2 + 2*z*cs*sn + 2*aprim*(1+sn)*(1+sn));
        Fz = -(2*z*sn2 + 2*r*cs*sn - 2*aprim*cs*(2+sn));
        if (flag) Fr = Fr*(-1);
        n_norm = sqrt(Fr*Fr + Fz*Fz);
        rdotn = (*vt_r*Fr + *vt_z*Fz)/n_norm;
        *vt_r = *vt_r - 2*rdotn*Fr/n_norm;
        *vt_z = *vt_z - 2*rdotn*Fz/n_norm;
}

void norm_sm(int ind_sm, double r, double *vr, double *vz)
/*  This routine calculates the new ray that results in a reflection of the
  CPC cone  */
{
        double nr, nz, n_norm, rdotn, ratio;

        ratio = rtio_sm[ind_sm];
        nz = 1.0;
        nr = nz/ratio;
        if (r < 0) nr = nr*(-1);

        n_norm = sqrt(nr*nr + nz*nz);
```

```
        rdotn = (*vr*nr + *vz*nz)/n_norm;
        *vr = *vr - 2*rdotn*nr/n_norm;
        *vz = *vz - 2*rdotn*nz/n_norm;
}

double *dvector(long nl, long nh)
/*  This procedure creates an array of variable dimensions  */
{
    double *v;
    v = (double *) malloc((size_t)((nh - nl + 1 +
NR_END)*sizeof(double)));
    if (!v) nrerror("allocation function in dmatrix()");
    return v - nl + NR_END;
}

void nrerror(char error_text[40])
/* Numerical Recipes standard error handler */
{
  printf("Break in code error...\n");
  printf("%s \n",error_text);
  printf("...now exiting to system...\n");
  exit(1);
}
```

C. ta2cp.c - located in /home/prochska/thesis/c/simult/2D

This program calculates the 2D transmission-angle curve for a CPC cone. On the whole, it is very similar to trnrad.c and trnrd.c as it utilizes most of the same procedures. The only difference is that the incident rays do not originate from a fiducial volume, but are assumed to originate at some infinite source with a variable incident angle. The user chooses what range of angles to examine, and therefore what portion of the transmission-angle curve to calculate. Any other differences between this program and trnrd.c are noted within the code.

```
/* This is program tacpc.c located in /home/prochska/thesis/c/simult/2D.
  It calculates the 2D transmission-angle curve for a CPC with a spherical
  PMT (as appropriate for the CTF).  See Appendix 2.C of the 1993 Senior Physics
  Thesis by Jason X. Prochaska for more detail  */

#include <stdio.h>
#include <math.h>

#define aprim 9.5
#define Pi 3.141592654
#define deg (Pi/180)
#define NR_END 1
#define n_scin 1.5
#define n_water 1.34
#define refl_cof .90
#define z_pmt -sqrt(121 - 9.5*9.5)

/* Global Variables */
double *aper_rad, ap_max, th_max, *r_cp, *hit, *theta, step_size;
long stps_apr, stps_the;
int length;
/*  Global Functions */
/*  Note that all of these functions are described in more detail in trnrad.c
located in /home/prochska/thesis/c/simult/3D */
void cp_array(double L_cone);
int pmt(double vr, double vz, double r_pt, double z_pt);
double inter_cp(double r_ct, double z_cp, double vr, double vz);
void norm_cp(double *vt_r, double *vt_z, double r, double z);
double *dvector(long nl, long nh);
void nrerror(char error_text[40]);

/* Main Program */
void main ()
{
        double L_cone;
```

```
        int refs, flag, k;
/* Function Declarations */
void inpt(double *L_ip, int *refs);
void init();
void calc_hit(double L_cone, int refs);

/* Setting Initial Data */
        inpt(&L_cone, &refs);
        aper_rad = dvector(1, stps_apr - 1);
        init();
        printf("\n Made it through init \n");
        /*  This procedure calculates the percentrage of rays which hit the PMT
         */
        calc_hit(L_cone, refs);
        /*  Output  */
        for (k = 1; k < stps_the; k++)
        {
                printf("\ntheta = %.8lf   perecent = %.2lf ", theta[k]/deg,
                  hit[k]*100);
        }
        printf("\n");
}

void inpt(double *L_ip, int *refs)
{
        double th, L_cn;

        printf("\nHow many steps are in the aperture entrance?   ");
        scanf("%ld", &stps_apr);
        printf("\nWhat is the length of the cone? (in cm)  ");
        scanf("%lf", &L_cn);
        *L_ip = L_cn;
        printf("\nWhat is the aperture radius?  (in cm)   ");
        scanf("%lf", &ap_max);
        printf("\nHow many reflections do you want to consider?  ");
        scanf("%d", refs);
        printf("\nWhat is theta max? (in degrees)   ");
        scanf("%lf", &th);
        th_max = th*deg;
        cp_array(L_cn);
        /*  These two input parameters set the range of angles examined around the extreme angle */
        printf("\nHow many steps off of theta_max do you want?  ");
        scanf("%ld", &stps_the);
        theta = dvector(1,stps_the - 1);
        hit = dvector(1, stps_the - 1);
        printf("\nWhat is the step size?  ");
        scanf("%lf", &step_size);
}

void calc_hit(double L_cone, int refs)
/*  This procedure calculates the percentage of rays which hit the PMT after
a maximum of (refs) reflections  */
{
        double tp, vt_r, vt_z, r_in, z_in, r_ap, z_ap;
        int i, j, l, m, flag, flag2, coeff_rad;

        for (i = 1; i < stps_the; i++)
        {
        /*  This for loop steps through a range of theta values around th_max,
          the extreme incident angle  */
                theta[i] = (th_max) - (step_size * deg)*(stps_the - 2*i)/2;
                hit[i] = 0;
                printf("\n theta = %.4lf, i = %u", theta[i]/deg, i);
                for (j = 1; j < stps_apr; j++)
                {
                  for (l = -1; l < 2; l += 2)
                  {
                  /*  This loop insures that both negative and positive values
```

```
                    of aper_rad are used  */
                        coeff_rad = (-1)*l;
                        r_ap = coeff_rad*aper_rad[j];
                        z_ap = L_cone;
                        vt_r = sin(theta[i]);
                        vt_z = -cos(theta[i]);

                        flag = pmt(vt_r, vt_z, r_ap, z_ap);
                        if (flag)   {flag2 = 1; }
                        else
                        {
                          m = 0;
                          r_in = r_ap;
                          z_in = z_ap;
                          do
                                {
                                m++;
                                tp = inter_cp(r_in, z_in, vt_r, vt_z);
                                r_in = r_in + tp*vt_r;
                                z_in = z_in + tp*vt_z;
                                norm_cp(&vt_r, &vt_z, r_in, z_in);
                                flag = pmt(vt_r, vt_z, r_in, z_in);
                                if (flag)
                                {
                                  flag2 = 1;
                                  break;
                                }
                /* If the ray travels upward, it will always escape... */
                                if (vt_z > 0 || m == refs)
                                {
                                  flag2 = 0;
                                  break;
                                }
                                } while (m < 1000);
                        }
                        if (flag2)
                        {
                        /*  If there is a hit, the percentage increases by the
                          following factor */
                                hit[i] += 1.0/(2*(stps_apr - 1.0));
                        }
                    }
                }
        }
}
```

D. pemev.c - located in /home/prochska/thesis/c/simult/3D/event

This program is very similar to trnrad.c, except it calculates the number of photoelectrons that one would expect to record per MeV of incident neutrino energy for a given PMT/light collector array. As with trnrad.c, most of the defined values have been set to match the CTF Upgrade design. The program assumes that for every reflection that occurs off of a light cone, there is a 10% loss of light. Also, the program calculates the lengths that a photon will travel in scintillator and water and then calculates the absorption coefficient as a function of average absorption lengths (8m for xylene, 60m for water). A further discussion of the theory behind the program can be found in Section 2.E of this thesis.

The only difference between pemev.c and trnrad.c is the conversion of 'hits' into a coverage which translates directly into a number of photoelectrons detected. As developed in section ?? the finite solid angle for a finite area element on the entrance aperture is (per total solid angle 4π),

$$\frac{d\Omega}{\Omega} = \frac{\cos\theta \, dl^2}{4\pi r^2}$$

Therefore, for each photon which hits the PMT, $d\Omega/\Omega$ needs to be calculated with reflection and absorption losses taken into account. Than the coverage for all photons is summed giving the total coverage and multiplied by the number of light cones used. The coverage is converted simply to pe/MeV by multiplying by the following factors: (10000 photons/Mev output)(.2 Quantum efficiency of PMT)(.6 PMT "practical factor")

The parts of the program which deal with the above calculations are noted within the code and are

fully described there. The routines which deterimine if a photon hits the PMT are the same as those in trnrad.c and are not included.

```
/*  This is pemev.c found in /home/prochska/thesis/c/simult/3D/event.  A
  detailed write-up of this program can be found in Appendix ?? of the 1993
  Physics senior thesis by Jason X. Prochaska.  This program is very similar
  to trnrad.c found in /prochska/thesis/c/simult/3D, also described in
  Appendix ??.  Because of their close similarity most of the comments found in
  trnrad.c are not included here.    */

#include <stdio.h>
#include <math.h>
#define aprim 9.5
#define Pi 3.141592654
#define deg (Pi/180)
#define NR_END 1
#define n_scin 1.5
#define n_water 1.34
#define refl_cof .90
#define z_pmt -sqrt(121 - 9.5*9.5)

/*   Global Variables   */
double *aper_area, *aper_rad, ap_max, th_max, *r_sm, *z_sm, *rtio_sm, *r_cp,
  *hts;
long stps_apr, stps_spz, stps_spr, stps_ang, nln_sm;
int length;
/*  Global Function Declarations */
void cp_array(double L_cone);
int pmt(double vt_x, double vt_y, double vt_z, double x_ap, double y_ap,
        double z_ap);
double inter_cp(double x_cp, double y_cp, double z_cp, double vt_x,
        double vt_y, double vt_z);
double inter_sm(double x_st, double y_st, double z_st, double vx, double vy,
            double vz, int *ind_sm);
void norm_cp(double *vt_x, double *vt_y, double *vt_z, double x, double y,
        double z);
void norm_sm(int ind_sm, double x, double y, double *vx, double *vy,
        double *vz);
double abcf(double x_as, double y_as, double z_as, double vx, double vy,
        double vz, double L_cone, double d_hyp, double d_apsp);
double *dvector(long nl, long nh);
void nrerror(char error_text[40]);

/*  Main program    */
void main ()
{
        double L_cone, d_apsp, fv_eff, fv_r, N_ev, sum_Ec, *Ec, z, gamma;
        int k, flag, npmt, i, refs;
/*  Function Declarations    */
void inpt(double *L_ip, double *d_ip, double *fv_ip, int *fl_ip, int *npm_ip,
        int *refs);
void init();
double calc_Ec(int k, double fv_r, double L_cone, int fl_cn, double d_apsp,
                double *N_ev, int refs);
/*  Setting Initial Data  */
        inpt(&L_cone, &d_apsp, &fv_eff, &flag, &npmt, &refs);
        stps_ang = 30;
        aper_area = dvector(1, stps_apr - 1);
        aper_rad = dvector(1, stps_apr - 1);
        Ec = dvector(1, stps_spr + 20);
        hts = dvector(1, stps_spz);
        init();
        printf("\nmade it through init\n");

        k = 0;
        do {
                k++;
                fv_r = fv_eff*k/(stps_spr);
```

```
        /* sum_Ec is the coverage of that the light collector has as a function
        of fiducial radius (including reflection and absorption factors)  */
                sum_Ec = calc_Ec(k, fv_r, L_cone, flag, d_apsp, &N_ev, refs);
        /*  Ec converts the coverage into photoelectrons/Mev  */
                Ec[k] = sum_Ec*npmt*(10000*.2*.6)/(N_ev);
        /*  Output  */
                printf("\n");
                printf("fv radius = %.4lf   Photoelectrons = %.4lf", fv_r,
                  Ec[k]);
        } while(k < stps_spr + 2);
}

void inpt(double *L_ip, double *d_ip, double *fv_ip, int *fl_ip, int *np_ip,
        int *refs)
/*  Input routine */
{
        double fv_rad, shield, th, fv_ff, L_cn, r, z, ratio;
        int flag,j;
        printf("\nHow many steps are in the aperture entrance?   " );
        scanf("%ld", &stps_apr);
        printf("\nHow many vertical steps in the sphere?   ");
        scanf("%ld", &stps_spz);
        printf("\nWhat is the Fiducial Volume? (in cm)   ");
        scanf("%lf", &fv_rad);
        fv_ff = fv_rad*n_scin/n_water;
        *fv_ip = fv_ff;
        printf("\nHow many radial steps in the sphere?  ");
        scanf("%ld", &stps_spr);
        printf("\nWhat is the shielding length?  (in cm)  ");
        scanf("%lf", &shield);
        printf("\nWhat is the length of the cone? (in cm)  ");
        scanf("%lf", &L_cn);
        *L_ip = L_cn;
        *d_ip = shield + fv_ff - L_cn;
        printf("\nWhat is the aperture radius?  (in cm)   ");
        scanf("%lf", &ap_max);
        printf("\nHow many cones are there?  ");
        scanf("%d", np_ip);
        printf("\nHow many reflections do you want to consider?  ");
        scanf("%d", refs);
        printf("\nIs the cone an SM or CPC?  (SM = 1, CPC = 0)  ");
        scanf("%d", &flag);
        if (flag)
        {
        /*  This routine inputs the SM profile and normal data */

                printf("How many lines of data are there?  ");
                scanf("%ld", &nln_sm);
                r_sm = dvector(1, nln_sm);
                z_sm = dvector(1, nln_sm);
                rtio_sm = dvector(1, nln_sm);
                for(j = 1; j <= nln_sm; j++)
                {
                  scanf("%lf %lf %lf", &r, &z, &ratio);
                  r_sm[j] = r;
                  z_sm[j] = z;
                  rtio_sm[j] = ratio;
                }
                *fl_ip = 0;
        }
        else
        {
                printf("\nWhat is theta max? (in degrees)   ");
                scanf("%lf", &th);
                th_max = th*deg;
                cp_array(L_cn);
                *fl_ip = 1;
        }
```

```
}

double calc_Ec(int k, double fv_r, double L_cone, int fl_cn, double d_apsp,
               double *N_ev, int refs)
 /* This procedure calculates sum_Ec with the necessary reflection and
  absorption factors. See trnrad.c for a more detailed description of
  most of the parts of this routine.   */
{
        double x_sp, y_sp, z_sp, x_ap, y_ap, z_ap, theta, alph, csphi, d_hyp,
           tp, vt_x, vt_y, vt_z, x_in, y_in, z_in, sum_Ec, Er, Ec, vo_x, vo_y,
           vo_z, ab_cof;
        int i, j, l, m, flag, ind_sm, flag2;

        sum_Ec = 0;
        *N_ev = 0;
        for (i=1; i < stps_spz; i++)
        {
                z_sp = ((fv_r)*i*(2.0)/(stps_spz) - fv_r) + d_apsp + L_cone;
                theta = acos(fabs(z_sp - d_apsp - L_cone)/fv_r);
                y_sp = 0;
                x_sp = (fv_r)*sin(theta);
                *N_ev += sin(theta);

                for (j = 1; j < stps_apr; j++)
                {
                        for (l = 0; l < stps_ang; l++)
                        {
                                alph = 2*l*Pi/stps_ang;
                                x_ap = aper_rad[j]*cos(alph);
                                y_ap = aper_rad[j]*sin(alph);
                                z_ap = L_cone;

                                vt_x = x_ap - x_sp;
                                vt_y = y_ap;
                                vt_z = z_ap - z_sp;

 /*  vo_x, vo_y, vo_z are used to save the intial direction of the ray from
  the sphere to the aperture entrance. This information is needed to
  calculate the absorption lengths  */
                                vo_x = vt_x;
                                vo_y = vt_y;
                                vo_z = vt_z;
                                flag = pmt(vt_x, vt_y, vt_z, x_ap, y_ap, z_ap);
                                m = 0;
                                if (flag) flag2 = 1;
                                else
                                {
                                  x_in = x_ap;
                                  y_in = y_ap;
                                  z_in = z_ap;
                                  if (fl_cn)
                                  { do
                                    {
                        /* m is used to store the number of reflections */
                                        m++;
                                        tp = inter_cp(x_in, y_in, z_in,
                                           vt_x, vt_y, vt_z);
                                        x_in = x_in + tp*vt_x;
                                        y_in = y_in + tp*vt_y;
                                        z_in = z_in + tp*vt_z;
                                        norm_cp(&vt_x, &vt_y, &vt_z, x_in, y_in,
                                                z_in);
                                        flag = pmt(vt_x, vt_y, vt_z, x_in, y_in,
                                                z_in);
                                        if (flag)
                                        {
                                          flag2 = 1;
                                          break;
```

```
                                    }
                /* If the ray travels upward, it will always escape.  */
                                    if (vt_z > 0 || m == refs)
                                    {
                                      flag2 = 0;
                                      break;
                                    }
                                  } while (m < 1000);
                                }
                                else
                                { do
                                  {
                                    m++;
                                    tp = inter_sm(x_in, y_in, z_in, vt_x,
                                            vt_y, vt_z, &ind_sm);
                                    x_in = x_in + tp*vt_x;
                                    y_in = y_in + tp*vt_y;
                                    z_in = z_in + tp*vt_z;
                                            norm_sm(ind_sm, x_in, y_in, &vt_x,
                                            &vt_y, &vt_z);
                                    flag = pmt(vt_x, vt_y, vt_z, x_in, y_in,
                                            z_in);
                                    if (flag || z_in < .008)
                                    {
                                      flag2 = 1;
                                      break;
                                    }
                                    if (vt_z > 0 || m == refs)
                                    {
                                      flag2 = 0;
                                      break;
                                    }
                                  } while (m < 1000);
                                }
                              }
                                    if (flag2)
                              {

        /*  Here we calculate the value of Ec for an individual photon and
          a finite area on the aperture mesh (aper_area) including reflection
          and absorption factors  */
                                /*  Reflection  */
                                Er = pow(refl_cof, m);

                                d_hyp = sqrt((x_sp - x_ap)*(x_sp - x_ap) +
                                  y_ap*y_ap + (z_sp - z_ap)*(z_sp - z_ap));
                                csphi = ((z_sp - z_ap)/d_hyp);
                              /*  Absorption  */
                                ab_cof = abcf(x_sp, y_sp, z_sp, vo_x, vo_y,
                                  vo_z, L_cone, d_hyp, d_apsp);
                        /* Ec is the coverage that a finite element of
                         aperture area (aper_area) has.  By summing over the
                         entire entrance aperture, one calculates the coverage
                         of the cone */
                                    Ec = Er*csphi*aper_area[j]*ab_cof
                                      /(4*Pi*d_hyp*d_hyp);
                        /*  Summing Ec for the overall coverage where sin(theta)
                          is the Monte Carlo factor appropriate for this
                          simulation.  */
                                    sum_Ec += Ec*sin(theta);
                                  }
                                }
                          }
                        }
                        return sum_Ec;
}

double abcf(double x_as, double y_as, double z_as, double vx, double vy,
```

```
        double vz, double L_cone, double d_hyp, double d_apsp)
 /*  This procedure determines the absorption factors as a function of the
     length that the ray travelled in the solved (inner vessel) and the length
     of its path in water  */
{
        double a, b, c, z1, z2, radic, rbag, z_bag, t, x_bn, y_bn, z_bn, cof,
                s1, s2;
        int flag;

     /*  Radius of the bag defined as 1m  */
        rbag = 100.0;
        z_bag = d_apsp + L_cone;

   /* Finds the intersection of the ray with the bag and chooses the lower
     point  */
        a = vx*vx + vy*vy + vz*vz;
        b = 2*(x_as*vx + y_as*vy + (z_as - z_bag)*vz);
        c = x_as*x_as + y_as*y_as + (z_as - z_bag)*(z_as - z_bag) - rbag*rbag;

        radic = b*b - 4*a*c;
        if (radic < 0) nrerror("\n Error in abcf");
        else
        {
                z1 = ((-b + sqrt(radic))/(2*a))*vz + z_as;
                z2 = ((-b - sqrt(radic))/(2*a))*vz + z_as;
                if (z1 > z2) z_bn = z2;
                else z_bn = z1;
        }

   /*  Calculates the lengths travelled in the two mediums.  s1 is the length
     in the scintillator and s2 is the length in the water (an average length of
     L_cone is asumed for the distance the ray travels in the cone)  */
        t = (z_bn - z_as)/vz;
        x_bn = x_as + t*vx;
        y_bn = y_as + t*vy;
        s1 = sqrt((x_as - x_bn)*(x_as = x_bn) + (y_as - y_bn)*(y_as - y_bn) +
          (z_as - z_bn)*(z_as - z_bn));
        s2 = d_hyp - s1 + L_cone;

     /*  Calculates the absorption coefficient with average absorption lengths
        of 8m and 60m for scintillator and water  */
        cof = (exp(-s1/800))*(exp(-s2/6000));
        return cof;
}
```

3. Fabrication Programs:

I wrote a number of programs for the fabrication of the wood molds and ribbing support structure. I don't feel it is necessary to include most of them here as they are not necessarily directly applicable. I do, however, include the program used to create the SM Cone follower, offering it is an example of what needs to be done to create similar pieces (e.g. the cutout and ribbing).

smfllwer.c - located in prochska/thesis/c/design. Basically, the only difference between this program and simult_dat.c is in the fact that the tool used to cut the piece on the CNC Mill has a finite radius. One can only stipulate the position and motion of the center of this tool and therefore in order to create the exact shape, the tool must be kept the radial distance away from the curve (normal distance). The program is very similar to simult_dat.c described above and the comments within the code are more than sufficient for understanding the program.

```
/*  This is program smfllwer.c located in prochska/thesis/c/design
    It was used to calculate the input data that drives the CNC to create
    a follower for the SM wood mold. Because this is America, the units are
    in inches. NOTE: This routine was written before a subtle error was
    discovered in the design of the SM cone. The two lines defining cathang and
    cathrad ought to be changed to match the definitions found in simult_dat.c
    found in prochska/thesis/c/ctfcpc. It is not a great error, but if one
    were to use this program to create the exact SM cone, it would be necessary
    to change those two definitions. A litte more detail about this program
```

```
   can be found in Appendix 3 of the 1993 Senior Physics Thesis by Jason X.
   Prochaska.  On the whole, this program is very similar to simult_dat.c   */

#include <stdio.h>
#include <math.h>

#define aprim .095
#define Pi 3.141592654
#define deg (Pi/180.0)
#define NR_END 1
#define n_scin 1.5
#define n_water 1.34

void main ()
{
        double rfv, reff, r2, dcone, d1, theta1, theta2, phi1, phi2, totang,
          tang, x[50000], y[50000], diff, thetastop, sumang, cathang, cathrad,
          covg, step, xlast, ylast, nmx, nmy, y2, dl, dpmt, ax, ay;

        int i, j, k, cones, flag;

        printf("\n Input the radius of the Fiducial Volume (in cm): ");
        scanf("%lf", &rfv);
        reff = (rfv * n_scin * .3937) / n_water;
        printf("\n The Effective Fiducial Volume is %.4f cm \n", reff);

        printf("\n How many cones are to be used? ");
        scanf("%ld", &cones);

        printf("\n What is the shielding length?  (in cm) ");
        scanf("%lf", &dpmt);
        r2 = reff + dpmt*.3937;

        cathang = 60 * deg;
        cathrad = 11.0*.3937;
        y2 = r2 + cos(cathang)*cathrad;

        y[1] = cathrad * cos(cathang);
        x[1] = cathrad * sin(cathang);

        xlast = -100;
        ylast = -100;
        flag = 1;
        step = .01;
        dl = 0.003;
        k =80;

        dcone = sqrt(x[1]*x[1] + (y[1] - y2)*(y[1] - y2));
        thetastop = asin(reff/dcone) + asin(x[1]/dcone);

        for (i = 1; i <= 50000; i++)
        {
                d1 = sqrt(x[i]*x[i] + y[i]*y[i]);
                dcone = sqrt(x[i]*x[i] + (y[i] - y2)*(y[i] - y2));

                theta1 = asin(cathrad/d1);
                theta2 = asin(reff/dcone);
                phi1 = asin(x[i]/d1);
                phi2 = asin(x[i]/dcone);

                if (phi1 + theta1 > (90*deg - cathang))
                {
                        totang = (Pi - theta1 - theta2 - phi1 - phi2) * .50;
                        tang = totang + phi1 + theta1 - Pi/2;
                }
                else
                {
                        sumang = atan((x[i] + x[1])/(y[i] - y[1]));
```

```
                if (sumang < thetastop)
                {
                        flag = 0;
                        break;
                }
                totang = (Pi - sumang - theta2 - phi2) * .50;
                tang = totang + sumang - Pi/2;
        }

        if ((x[i] - xlast)*(x[i] - xlast) + (y[i] - ylast)*(y[i] - ylast)
          > (step * step))
        {
                k += 10;
                nmx = -cos(tang);
                nmy = sin(tang);

        /*  Because the tool used to make the follower has a finite
         radius = .375 inches, and becuase we can only control the
         position of the center of this tool, we must move a normal
         distance .375 inches normal off the curve.  nmx and nmy define
         the normal direction and ax and ay use this direction to
         calculate where the center of the tool needs to be.  */

                ax = x[i] - nmx*(.375) - 2.26;
                ay = (y[i] - y[1]) - (nmy)*(.375);
                printf("\n");
                printf("N%uX%.4lfY%.4lf", k, ay, ax);
                xlast = x[i];
                ylast = y[i];
        }

        j = i + 1;
        x[j] = x[i] + sin(tang)*dl;
        y[j] = y[i] + cos(tang)*dl;
    }
    if (flag)
    { printf("\n We never reached the end.");
    }
    else
    {
        nmx = -cos(tang);
        nmy = sin(tang);
        covg = 32.0 * (1.0 - cos(phi2)) * 100;

        printf("\n x = %.4f  y = %.4f  %.4f  %.4f", x[i], y[i] - y[1],
          nmx, nmy);
        printf("\n We have reached the end of the profile. \n");
    }
}
```

References:

1. J. N. Bachall, *Neutrino Astrophysics* (Cambridge: Cambridge Univ. Press, 1989).

2. J.N. Bachall, M.H. Pinsonneault, Rev. Mod. Phy. **64**, 885 (1992).

3. J.N. Bachall, *ibid. (1989)*, p.9.

4. B. Pontecorvo, Zh. Eksp. Teor. Fiz. **53**, 1771 (1967).

5. L. Wolfenstein, Phys. Rev. D **17**, 2369 (1978).

6. J.N. Bachall and S.C. Frautschi, Phys. Lett., **B29**, 263 (1969).

7. J. N. Bachall, *ibid. (1989)*, p. 257.

8. L. Wolfenstein, Phys. Rev. D, **20**, 2634 (1979).

9. S.P. Mikheev and A.Yu. Smirnov, Sov. J. Nucl. Phys., **42**, 913 (1985).

10. H. Bethe, Phys. Rev. Lett., **56**, 1305 (1986).

11. S.P. Mikheev and A.Yu. Smirnov in *Proceedings of Twelfth International Conference on Neutrino Physics and Astrophysics*, edited by T. Kitagaki and H. Yuta (Singapore: World Scientific, 1986), p. 177.

12. R. Davis, Jr., Phys. Rev. Lett., **12**, 303 (1964).

13. R. Davis, Jr., in *Proceedings of Seventh Workshop on Grand Unification, ICOBAN'86*, Toyama, Japan, edited by J. Arafune (Singapore: World Scientific, 1987), p. 237.

14. Y. Totsuku, in *Proceedings of Seventh Workshop on Grand Unification, ICOBAN'86*, Toyama, Japan, edited by J. Arafune (Singapore: World Scientific, 1987), p. 118.

15. M. Nakahata *et al.*, Phys. Rev., **55**, 3786 (1986).

16. Y. Suzuki, in *International Symposium on Neutrino Astrophysics*, Takayama, Kamioka, October 19,1992.

17. T. Kristen, in *'86 Massive Neutrinos in Astrophysics and Particle Physics*, Proceedings of the Sixth Moriond Workshop, edited by O. Fackler and J. Tran Thanh Van (Gif-sur_Yvette: Editions Frontieres, 1986), p. 119.

18. V.N. Gavlin, *et al.*, 1992, in Proceedings of the XXVIth International Conference on High Energy Physics, Dallas, TX (to be published).

19. GALLEX Collaboration, P. Anselmann *et al.*, preprints GX 1-1992, GX 2-1992.

20. V. N. Gavlin, *ibid.*

21. S.A. Bludman, S.A. *et al.*, Implications of Combined Solar Neutrino Observations and Their Theoretical Uncertainties, PREPRINT UPR-0516T (revised).

22. NSF proposal, *Borexino - A Real Time Detector for Low Energy Solar Neutrinos*.

23. R. Winston and W.T. Welford, *High Collection Nonimaging Optics*, (New York: Academic Press, Inc., 1989), p. 132.

24. Winston, *ibid. (1989)*, p. 132.

25. H. Hinterberger and R. Winston, *Rev. Sci. Instrum.* **37**, 1094 - 1095 (1966).

26.Winston, *ibid. (1989)*, p. 61.

27. M. Moorhead, 1992, Ph.D. thesis (Oxford University).

28. Winston, *ibid. (1989)*, p. 84.

29. Handbook of Chemistry and Physics 49th, edited by Robert C. Weast, (Cleveland: The Chemical Rubber Co., 1968), p. E-227.

This paper represents my own work
in accordance with University Regulations.