2018

Evaluating the Effectiveness of Current Atmospheric Refraction Models in Predicting Sunrise and Sunset Times

Teresa Wilson
Michigan Technological University, tawilson@mtu.edu

EVALUATING THE EFFECTIVENESS OF CURRENT ATMOSPHERIC REFRACTION MODELS IN PREDICTING SUNRISE AND SUNSET TIMES

By

Teresa A. Wilson

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Physics

MICHIGAN TECHNOLOGICAL UNIVERSITY

2018

This dissertation has been approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY in Physics.

Department of Physics

Dissertation Co-advisor:	*Dr. Robert J. Nemiroff*
Dissertation Co-advisor:	*Dr. Jennifer L. Bartlett*
Committee Member:	*Dr. Brian E. Fick*
Committee Member:	*Dr. James L. Hilton*
Committee Member:	*Dr. Claudio Mazzoleni*
Department Chair:	*Dr. Ravindra Pandey*

Dedication

To my fellow PhD students

and the naive idealism with which we started this adventure. Like Frodo, we will never be the same.

Contents

List of Figures . . . . . **xi**

List of Tables . . . . . **xvii**

Acknowledgments . . . . . **xix**

List of Abbreviations . . . . . **xxi**

Abstract . . . . . **xxv**

1 Introduction **1**

1.1 Introduction 1

1.2 Effects of Refraction on Horizon 3

1.2.1 Green Flash 4

1.2.2 Etruscan Vase 5

1.2.3 Fata Morgana 6

1.2.4 Novaya Zemlya Effect 7

1.3 Sunrise and Sunset Times 8

1.4 Applications 12

2 Refraction and Dip 15

2.1 Atmospheric Refraction Models 18

2.1.1 Standard Refraction 18

2.1.2 Garfinkel 19

2.1.3 Hohenkerk & Sinclair 21

2.1.4 Bennett 23

2.2 Dip 27

3 The Ultimate Rise/Set Algorithm 31

3.1 Assessment of Existing Calculators 32

3.1.1 Standards/Community Suggestions 32

3.1.2 Web Survey Findings 37

3.2 The Ultimate Rise/Set Algorithm 39

3.2.1 Relationship to Antecedents 39

3.2.2 Prediction Algorithm 41

3.2.3 Refraction 42

3.2.4 Using the Program 43

3.2.5 External Code Review 44

3.3 Calculator Comparisons 45

3.3.1 Comparison with Standards 45

3.3.2 Comparison with Other Known Calculators 48

3.3.3 Refraction Validation 51

3.4 Conclusion & Future Work 56

4 Analysis of Historical Sunrise and Sunset Data 59

4.1 Mount Wilson Observatory, CA 60

4.1.1 Population Discussion of Novaya Zemlya Effect 64

4.2 Edmonton, Alberta, Canada 66

4.3 Hawai'i and Chile 72

4.4 Smiley 74

4.5 Discussion 77

5 Observing Kepler's Laws with the Sunrise and Sunset 81

5.1 Theory 82

5.2 Video Example 89

5.3 Discussion 91

6 Sunrise & Sunset Observer Citizen Science Project 95

6.1 Smartphone Application 96

6.2 Preliminary Analysis 97

6.3 Conclusion & Future Work 99

7 Conclusion 103

7.1 Conclusion 103

7.2 Future Work & Recommendations 105

References **109**

A Sample Code **119**

A.1 URSA 119

A.2 Horizon Crossing 130

A.3 Hohenkerk & Sinclair Refraction 132

A.4 Bennett Refraction 139

A.5 Schlyter API Request 142

A.6 Meeus Request 145

A.7 USNO API Request 148

B Tables **153**

B.1 Sampson Humidity Values 153

C Letters of Permission **161**

List of Figures

1.1 **Moment of Sunrise or Sunset** The local horizon has an altitude of 0° and a zenith distance of 90°. R_0 is the angular amount of atmospheric refraction. The refracted Sun appears "flattened" because the upper limb experiences less refraction than the lower limb. The geometric Sun is the actual image of the Sun where s is its semi-diameter.[1] 3

1.2 **Green Flash** [2] 4

1.3 **Etruscan Vase** [3] 6

1.4 **Fata Morgana** [4] 6

1.5 **Novaya Zemlya Effect** [5] 7

2.1 **Refraction in a Plane** Path of a light ray through a planar atmosphere.[6] 16

2.2 **Refraction in the Atmosphere** Path of a light ray through the atmosphere from Hohenkerk & Sinclair.[7] 18

2.3 **Bennett-NA Refraction** Atmospheric refraction at height of celestial object. The top line indicates standard conditions, the middle represents conditions in Edmonton, Canada, and the lower line those from Mount Wilson, CA 28

2.4 **Geometric Dip** The angle, d_g between the geometric horizon and the visible horizon, without accounting for refraction.[8] 29

2.5 **Apparent Dip** The angle, d, between the geometric horizon and the visible horizon H, accounting for refraction.[8] 30

3.1 **URSA vs SLSM** These plots show the absolute time differences between the sunrise/set outputs of SLSM and URSA 46

(a) URSA vs SLSM (Rise) 46

(b) URSA vs SLSM (Set) 46

3.2 **URSA vs PAP** These plots show the absolute time differences between the sunrise/set outputs of PAP and URSA 47

(a) URSA vs PAP (Rise) 47

(b) URSA vs PAP (Set) 47

3.3 **URSA vs USNO** These plots show the absolute time differences between the sunrise/set outputs of USNO and URSA 49

(a) URSA vs USNO (Rise) 49

(b) URSA vs USNO (Set) 49

3.4 **URSA vs Schlyter** These plots show the absolute time differences between the sunrise/set outputs of Schlyter and URSA 49

(a) URSA vs Schlyter (Rise) 49

(b) URSA vs Schlyter (Set) 49

3.5 **URSA vs Meeus** These plots show the absolute time differences between the sunrise/set outputs of Meeus and URSA 50

(a) URSA vs Meeus (Rise) 50

(b) URSA vs Meeus (Set) 50

4.1 **Mount Wilson Sunset Data** Time differences between observations and computed predictions by day of year using 34′, Bennett-NA, and H&S model for refraction. Dip is included in all scenarios. 63

(a) Observed - Computed (34′) 63

(b) Observed - Computed (Bennett) 63

(c) Observed - Computed (HS) 63

4.2 **Edmonton Sunrise Data** Time differences between observations and computed predictions by day of year using 34′, Bennett-NA, and H&S's models for refraction. Dip is included in the first three graphs. . . . 68

(a) Observed - Computed (34′ dip) 68

(b) Observed - Computed (Bennett dip) 68

(c) Observed - Computed (H&S dip) 68

(d) Observed - Computed (34′) 68

(e) Observed - Computed (Bennett) 68

(f) Observed - Computed (H&S) 68

4.3 **Edmonton Sunset Data** Time differences between observations and computed predictions by day of year using 34′, Bennett-NA, and H&S's models for refraction. Dip is included for the first three graphs . . . 69

(a) Observed - Computed (34′ dip) 69

(b) Observed - Computed (Bennett dip) 69

(c) Observed - Computed (H&S dip) 69

(d) Observed - Computed (34′) 69

(e) Observed - Computed (Bennett) 69

(f) Observed - Computed (H&S) 69

4.4 **Schaefer & Liller Sunset Data** Time differences between observations and predictions by day of year using 34′ and 34′+dip for refraction. 73

(a) Observed - Computed (34′) 73

(b) Observed - Computed (34′ with dip) 73

4.5 **Smiley Observations** Time differences between observations and predictions by day of year using 34′ 75

5.1 **Variation of Crossing Time with Latitude** These plots show the minimum and maximum crossing times for the year with respect to latitude. When δ is ignored the days fall on aphelion and perihelion and on the days in question. The crossing time for all other days fall between the extremes represented by the perihelion and aphelion lines. 85

(a) $0° - 60°$ lat 85

(b) $61° - 66°$ lat 85

5.2 **Variation of Crossing Time during the Year** This plot shows the deviation in crossing times for the latitudes of 0°, 20°, 40°, and 60° when $\delta = 0°$. The deviation was calculated by subtracting the actual crossing time per day from the mean value for the year. The annual mean times were 127.99 sec, 136.21 sec, 167.08 sec, and 255.98 sec, respectively. 86

5.3 **Days of the Year with Maximum and Minimum Horizon-Crossing Times** These plots show days on which the minimum and maximum crossing times for the year occur and the number of latitudes for which that day is a minimum or maximum. In Fig. 5.3(b), perihelion is represented as day 366. 87

(a) Days with Minimum Crossing Times 87

(b) Days with Maximum Crossing Times 87

5.4 **Variation of Crossing Time at** 40° **Latitude** This plot shows the crossing times throughout the year with and without including declination in the calculations. 88

5.5 **LoggerPro Screenshot** A screenshot of the analysis performed using the LoggerPro software. 91

6.1 **Sunrise & Sunset Observer App** Example pages of what an observer might expect when using the app 97

(a) Google Play Store Entry 97

(b) Sample Page 1 97

(c) Sample Page 2 97

List of Tables

3.1 **Locations and Dates used for URSA Validation** All combinations were tested 48

3.2 **Comparison of NA and Bennett-NA Refraction Values** Standard Conditions 52

3.3 **Refraction Value Comparison 1** URSA and H&S 55

3.4 **Refraction Value Comparison 2** URSA and H&S 56

4.1 **Observation Site Information** Includes Geographic coordinates and height of eye above the horizon 61

4.2 **KS Statistic Results** Mount Wilson 65

4.3 **Mount Wilson Results** Differences between Observed and Computed times for Mount Wilson data separated by Population 66

4.4 **Edmonton Results** Differences between Observed and Computed times for Edmonton data 71

4.5 **Smiley Data** 76

6.1 **Video Analysis Results** 101

B.1 **Sampson Humidity Data** 154

Acknowledgments

To my wonderful parents. Your constant love and support through this endeavor has helped me in more ways than you will ever know. And to my ever supportive siblings: we make a great team.

To my advisor, Robert Nemiroff. Without your APOD ad looking for a graduate student, I would never have ended up at Michigan Tech. You allowed me to explore an ocean of ideas from my little research boat, and that one time we were even famous. I am ever the wiser for this experience.

To my other advisor, Jennifer Bartlett. Your guidance, patience, and encouragement has helped me become the researcher and science writer that I am. Summers at the USNO, thanks to NREIP, were a wonderful experience and provided me the opportunity to see the applications of my work in action. Celestial Navigation training was a real treat.

To Fr. Ben Hasse from St. Albert the Great Catholic Church in Houghton, MI. You helped me stay grounded, keep perspective, and ignore the giraffe.

A big thank you to my fellow graduate students at Michigan Tech: Bethany Klemetsrud, Chad Brisbois, Hugo Ayala Solares, Kevin Waters, Joseph Niehaus, Tyler

Capek, Andrew Chapp, and Scott Rutterbush. You were there with me through the best and worst of times. Afternoon coffee, evenings at the Dog, and broomball season will forever be my fondest memories.

List of Abbreviations

Below is a list of all abbreviations, acronyms, and symbols used in this dissertation and their meanings. They are given once in the document, but are listed here for thoroughness.

α	Right Ascension
ΔT	$TT - UT1$
δ	Declination
Θ	Apparent Sidereal Time
λ	Longitude
ξ	Angle of Refraction
ρ	Air Density
ϕ	Latitude
ψ	Angle of Horizon Crossing
ω	Angular Speed
a	Semi-major axis
API	Application Program Interface
d	Apparent Dip
d_g	Geometric Dip
$d_\odot$	Angular Size of Sun

$D_\odot$	Solar Diameter
DE	Developmental Ephemeris
e	Orbital Eccentricity
EOD	Earth Orientation Department
GPS	Global Positioning System
GST	Greenwich Sidereal Time
h	Height of Observer above Horizon
$h_\odot$	Observed Solar Altitude
h_0	Altitude of Solar Center
HA	Hour Angle
H&S	Hohenkerk & Sinclair
HMNAO	Her Majesty's Nautical Almanac Office
IAU	International Astronomical Union
IBM	International Business Machine
JPL	Jet Propulsion Laboratory
k	Constant of Mean Refraction
KS	Kolmogorov-Smirnov
LHA	Local Hour Angle
MICA	Multiyear Interactive Computer Almanac
MOARSP	Military Operations Analysis Rise Set Program
n_1	Initial Index of Refraction

n_2	Secondary Index of Refraction
NA	Nautical Almanac
NASA	National Aeronautics and Space Administration
NAO	Nautical Almanac Office
NOVAS	Naval Observatory Vector Astrometry Software
NZ	Novaya Zemlya
P	Pressure
PAP	Public Affairs Program
PST	Pacific Standard Time
r	Radial Distance
r_0	Distance from Center of Earth to Observer
R	Refraction Angle
$R_\oplus$	Radius of Earth
$R_\odot$	Radial Distance to Sun
RA	Right Ascension
S&L	Schaefer and Liller
SLAC	Solar-Lunar Almanac Core
SLSM	Standard Lunar-Solar Model
SLSMA	Standard Lunar-Solar Model Algorithm
SQL	Structured Query Language
SSO	Sunrise & Sunset Observer

t	Days since Perihelion
T	Temperature
$T_{\odot}$	Time of Horizon Crossing
TAI	International Atomic Time
TT	Terrestrial Time
URSA	Ultimate Rise/Set Algorithm
USNO	United States Naval Observatory
$UT1$	Universal Time
UTC	Coordinated Universal Time
$v_{\oplus}$	Angular velocity of Earth
z	Zenith Distance
z_0	Topocentric Zenith Distance
z_1	Angle of Incidence
z_2	Angle of Refraction

Abstract

The standard value for atmospheric refraction on the horizon of $34'$, used in all publicly available sunrise and sunset calculators, is found to be inadequate. The assumptions behind atmospheric models that predict this value fail to account for real meteorological conditions. The result is an uncertainty of one to five minutes in sunrise and sunset predictions at mid-latitudes (0° - 55° N/S). A sunrise/set calculator that interchanges the refraction component by varying the refraction model was developed. Two atmospheric refraction models of increasing complexity were tested along with the standard value. The predictions were compared with data sets of observed rise/set times taken from Mount Wilson Observatory in California, University of Alberta in Edmonton, Alberta, observations from various locations in Chile, and on-board the SS James Fergus in the Atlantic Ocean. Increasing the complexity of the model did not yield significantly better results. These observations make up the entirety of documented sunrise and sunset times. A thorough investigation of the problem requires a more substantial data set of observed rise/set times and corresponding meteorological data from around the world. A mobile application, Sunrise & Sunset Observer, was developed so that anyone can capture this astronomical and meteorological data using their smartphone as part of a citizen science project. Data analysis will lead to more complete models that will provide higher accuracy rise/set predictions to benefit astronomers, navigators, and outdoorsmen everywhere.

Chapter 1

Introduction

1.1 Introduction

The periodicity of the heavenly bodies has been well understood for millennia, and the time of their passage through a particular location in the sky was used to set clocks until the advent of the atomic clock. Astronomers were the original time lords. Accurate clocks solved the longitude problem: astronomers could easily determine the time when a celestial body would be in a specific position; by comparing the time and position of those celestial bodies at an observer's location to those at a standard location, such as Greenwich, the longitude of the observer could be calculated.[9]

Sunrise and sunset are defined as the moments when the upper limb of the solar

disk appears or disappears, respectively, on the horizon. Solar right ascension and declination are reported to $0''.001$ and the solar semi-diameter has been measured to $0''.2$, which means calculating sunrise or sunset times to the nearest second should be relatively simple.[10] [11] Why then does the *Astronomical Almanac* report sunrise and sunset times only to the nearest minute and other solar phenomena, such as eclipses, to the nearest 0.1 min (6 seconds)?[12]

The answer lies in the refraction of light as it travels through the atmosphere. As discussed in Chapter 2, according to Snell's law, light bends when it reaches a medium with a different refractive index.This is the case when the Sun's light meets the Earth's atmosphere, and the light continues to bend as the atmosphere's density increases. At near-zenith angles, refraction is negligible; however, at the horizon, when the light is traveling through the most atmosphere, atmospheric conditions are fluid and can change the refractive index significantly. Because the actual conditions cannot easily be predicted, published almanacs and publicly available sunrise/set calculators use a constant of $34'$ for atmospheric refraction at the horizon, and do not account for any seasonal variability based on geographic location. Thus, the atmosphere is assumed to behave the same in Barrow, AK in January as it does in Honolulu, HI in July. Predicting sunrise/set times to no better than the minute reflects the uncertainty in time due to the poorly determined refraction component.

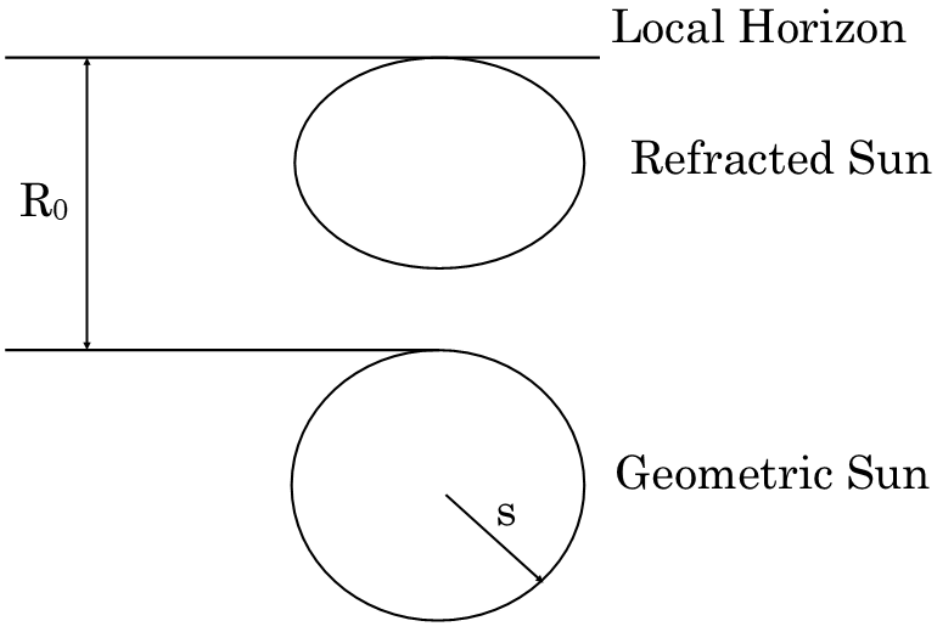


Figure 1.1: Moment of Sunrise or Sunset The local horizon has an altitude of 0° and a zenith distance of 90°. R_0 is the angular amount of atmospheric refraction. The refracted Sun appears "flattened" because the upper limb experiences less refraction than the lower limb. The geometric Sun is the actual image of the Sun where s is its semi-diameter.[1]

1.2 Effects of Refraction on Horizon

Frequent changes in atmospheric conditions near the surface of the earth cause the greatest differences in the refractive index. The atmospheric phenomena discussed

below are fairly common, and knowledge of what meteorological conditions cause them can help predict when they will occur.

Changes in atmospheric conditions cause both superior and inferior mirages: optical phenomena due to refraction. Superior mirages occur when a mirage appears above the erect image, and inferior mirages occur when a mirage appears below. Mirages play an important role in navigation and understanding atmospheric refraction, especially at high latitudes where they are most common. In fact, a cold water mirage, such as one of those described below, might have played a role in the tragedy of the Titanic.[13]

1.2.1 Green Flash

Figure 1.2: Green Flash [2]

The green flash is generally a sunset related phenomenon, but may be seen at sunrise, and is directly related to the wavelength of light being scattered by the atmosphere. While lasting only a few seconds, it adds uncertainty to determining the moment of sunrise or sunset. The traditional green flash can only be seen in the event of an inferior mirage, which requires a warm air layer near the surface of the Earth and a

strong temperature gradient. The atmospheric conditions that are responsible for the mirage magnify the refractive differences between red and green light. The strength of the magnification determines the color of the flash. Sometimes they are closer to yellow, and occasionally, if the air is very clear, they are blue. Even if conditions are right, observers will only see the green flash if they are viewing the event from above the warm air layer. In this case, if seen at sunset, the Etruscan Vase effect mentioned below will usually precede the flash.

The green flash might also be visible in the presence of temperature inversion layers in the form of a mock-mirage. The inversion layers increase the rate of change of the refractive index of the air, refracting the light rays back towards the Earth with a shallow elevation angle. If the curvature of the light's new path is smaller than that of the Earth, the light is trapped and the region in which it propagates is called a duct. The duct causes the Sun to appear in slices on the horizon. Like its inferior mirage counter-part, mock-mirage green flashes can only be observed from above the inversion layer.[14] [8]

1.2.2 Etruscan Vase

An inferior mirage due to a warm air layer near the surface of the Earth causes what is known as the Etruscan Vase effect. An inverted image of the Sun appears directly

Figure 1.3: Etruscan Vase [3]

below the solar disk. This mirage joins the image of the disk to form what Jules Verne called an "Etruscan Vase."[15] As the Sun sinks lower, its shape shifts to appear more like an Ω. As the inverted and erect images join, a green flash appears at the top of the image of the solar disk.[16] The observer's height above sea-level affects how distinctly they will see the effect. The closer they are to sea-level, the stronger the Ω-shape will be. An Etruscan Vase can also appear at sunrise.

1.2.3 Fata Morgana

Figure 1.4: Fata Morgana [4]

The *Fata Morgana* is named after Morgana Le Fay from the Arthurian legends who is known for her magical powers of creating castles in the air.[17] It is aptly named, as the images created by this mirage bear little resemblance to the object that formed them.[18] It occurs when a light duct is formed by calm layers of warm and cool air that cause combinations of superior and inferior mirages.[19]

1.2.4 Novaya Zemlya Effect

Figure 1.5: Novaya Zemlya Effect [5]

The Novaya Zemlya (NZ) effect is an optical ducting phenomemon in the lower atmosphere, and is traditionally thought of as an arctic mirage. It is named after the island of Novaya Zemlya (76°12′N, 67°33′E) where the first known observation was made in 1597 by a member of the Berentz expedition who witnessed a sunrise two weeks early.[20] The Shackleton expedition in 1915 also reported witnessing this phenomenon.[21] The effect is due to a cold surface layer of air, and a strong temperature inversion layer in which the temperature increase occurs in a narrow elevation range. A duct forms and can cause the image of the Sun to be rectangular and split into multiple horizontal pieces.[22] According to the data from Mount Wilson, this "arctic mirage" can be seen at mid-latitudes, and can cause delays in sunset times by 100 seconds or more.

1.3 Sunrise and Sunset Times

Before accounting for refraction, uncertainty in the solar diameter ($0''.2$) is the limiting factor for determining the uncertainty in sunrise and sunset time predictions. This leads to an uncertainty in timing of about $0.013s/\cos\psi$, where ψ the angle between the horizon and the Sun's apparent motion. ψ is dependant on solar declination and the latitude of the observer, and is discussed in further detail in Chapter 5. While the Sun is crossing the horizon in non-Arctic regions, the rate of change in ψ is negligible. Therefore, before accounting for refraction, predictions within 0.1s should be possible.

All sunrise and sunset calculators account for refraction in some way. However, all known, published papers on estimating refraction near the horizon are based on making observations down to a few degrees above the horizon and extrapolating down to the horizon. It has been shown that this extrapolation method inadequately represents reality.[23] At best, the uncertainty in refraction is limited by anomalous refraction, or the amount of refraction seen, but unaccounted for, in any atmospheric refraction model.[24] The value for anomalous refraction is around $10''$, leading to an uncertainty of about $0.7s/\cos\psi$, where ψ is as above.[24] Therefore, with sufficient knowledge of the meteorological conditions and an accurate refraction model, sunrise and sunset predictions to the nearest second should be possible, at least at central latitudes.

Almanacs give sunrise and sunset predictions to the nearest minute because their underlying algorithms cannot adequately predict the meteorological conditions nor the resulting refraction. The differences between observed and calculated sunrise and sunset times with corresponding meteorological observations are a measure of the accuracy of the refraction model used. The results of those comparisons should identify the most influential factors and guide efforts to improve refraction modeling. To explore the relative contributions of different factors satisfactorily, a robust set of observations that span a wide range of meteorological conditions, locations, and dates is necessary.

In order to compare observed times to calculated times, an accurate sunrise and sunset prediction program, the Ultimate Rise/Set Algorithm (URSA), was developed that is capable of incorporating externally computed values for atmospheric refraction. Chapter 3 describes its development and verification along with guidance for its use. Chapter 7 suggests modifications to this calculator which would expand its use to other celestial objects, such as the moon, and increase its potential to contribute to other atmospheric studies, such as illuminance.

For the initial comparisons, three refraction models were chosen that span a range of complexity and expected accuracy: standard refraction, Hohenkerk and Sinclair's high accuracy model, and a lower accuracy formulation by Bennett.[25] [26] Chapter 2 describes each of these models and, in the case of standard refraction and Bennett's

model, documents their history and evolution, as this was not readily available. Hohenkerk and Sinclair's algorithm has sufficient documentation. Chapter 3 describes how the refraction models were implemented for use with the sunrise and sunset prediction program. The hypothesis was that the increasing level of complexity of the refraction models would result in an increased accuracy of the sunrise/set times. However, more complex refraction models require a larger set of meteorological inputs, which limits their practicality for daily use. Chapter 4 also discusses this after evaluating the effectiveness of each refraction model used.

The comparisons required a comprehensive set of sunrise and sunset observations with their associated meteorological data. The observational data sets were available in two flavors: those with matching meteorological conditions, and those without. Observations with matching meteorological conditions consist of

- 250 sunset times from Mount Wilson Observatory near Los Angeles, CA taken between 1987 and 1991 [27]
- 244 sunrise times and 125 sunset times taken from Edmonton, Alberta, Canada between 1990 and 1992 [1]
- 1 sunrise and 4 sunset times acquired through the implementation of the citizen science project described in Chapter 6

Observations without corresponding meteorological conditions consisted of

- 116 sunset times from the paper by Schaefer and Liller taken in 1989 [28]
- 7 sunrise and 29 sunset times taken during the Smiley eclipse expeditions between 1946 and 1951 [29]

Chapter 4 analyzes these data sets. The results indicate that varying the refraction models has little effect on improving the accuracy of sunrise and sunset times. Over an ocean horizon at high elevations, refraction is, in general, much greater than predicted by all three of the refraction models. Over a land horizon, the opposite is true, with the refraction models over-estimating the amount of refraction on the horizon. One must then conclude that current refraction models do not adequately represent the atmosphere near the horizon.

These data sets, however, do not cover a wide range of meteorological conditions nor locations. Therefore, Chapter 6 describes a citizen science project to collect a greater range of observations. As mentioned earlier, Chapter 4 analyzes the observations amassed so far though this program, which demonstrates feasibility of this collection method and the potential value of the data.

1.4 Applications

This study looks at the effectiveness of atmospheric refraction models when used to predict sunrise/set times. Accurate times of these phenomena can provide those using celestial navigation with more accurate positioning, especially at high latitudes where horizontal refraction effects are most common. Global Positioning System (GPS) allows one to determine their location within 7.8 m.[30] However, sailors learn celestial navigation for positioning when GPS is unavailable or suspected to be unreliable. Under normal conditions, they should be able to determine their locations within a nautical mile, or $\approx 1'$ latitude. Due to the rotational speed of the Earth of $15'$ per min, timing errors of phenomenon of 1 minute can affect positioning calculations by as much as 15 nautical miles.

Chapter 3 discusses broadening the use of this rise/set calculator to twilight times, which would benefit astronomers who use twilight conditions to determine when observations should occur. A higher-accuracy prediction model would help them better determine their observing window, especially for the automated telescopes, and perhaps increase observing time: an additional few minutes per day would accumulate to hours of observing time over the course of a year.

On a broader scale, understanding the conditions under which refraction models fall

short can help with our understanding of the atmosphere, and determine how often "standard" conditions actually occur. Complex refraction models such as that by Hohenkerk and Sinclair use an atmospheric model to trace the path of the light-ray. Limitations of the refraction algorithm, discussed in Section 2.1, also reflect how well their atmospheric model describes the actual behavior of the atmosphere. Chapter 5 discusses how long it should take the Sun to cross the horizon, assuming atmospheric conditions remain constant during the crossing. The atmospheric phenomena mentioned in Section 1.2 would increase the crossing times significantly. A study of horizon-crossing times would also indicate limitations of refraction models.

Chapter 2

Refraction and Dip

Understanding how light bends as it passes through any given substance is, in general, as simple as knowing how the index of refraction changes in that material. Applying Snell's law

$$n_1 \sin z_1 = n_2 \sin z_2 \tag{2.1}$$

where n_1 is the initial index of refraction, z_1 is the angle of incidence, n_2 is the second index of refraction, which then gives the angle, z_2, of the refracted light as it encounters the changes in the material. For the atmosphere, the refractive index is proportional to the density of the air. The density depends on the meteorological conditions of any given location, especially temperature and pressure, but also humidity and lapse rate. Lapse rate, also known as the vertical temperature gradient,

is the rate at which the temperature decreases with height above the surface of the Earth. Refraction is also dependent on the wavelength of the light that reaches the observer, and the zenith distance of the observed object.

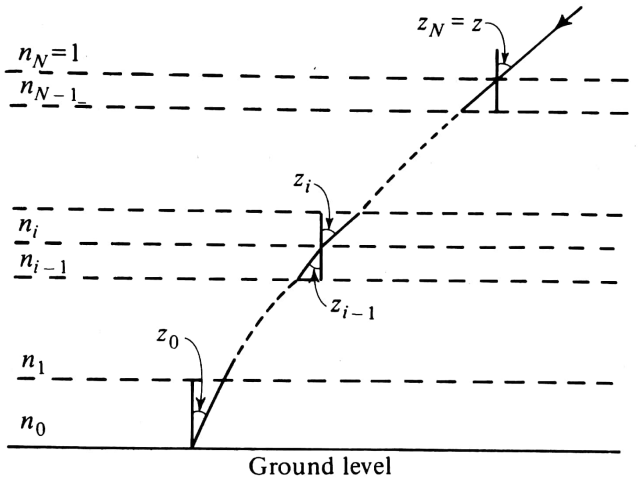


Figure 2.1: Refraction in a Plane Path of a light ray through a planar atmosphere.[6]

If we assume a planar model for the atmosphere, as that pictured in Fig. 2.1, this becomes

$$n_0 \sin z_0 = \sin z$$

where z_0 is the *topocentric zenith distance*, or the zenith distance as measured from the observer's location, of the source, and n is set to 1 since the uppermost layer is outer space and assumed to be a vacuum. The angle of refraction, R is

$$R = z - z_0$$

and using small angle approximations, Eqn. 2.1 becomes

$$R = (n_0 - 1)\tan z_0 = k \tan z_0 \tag{2.2}$$

where k is known as the *constant of mean refraction.*[31] [6] k is empirically determined by astronomical observations, and the value generally accepted is $58''.2$, for conditions of 10°C air temperature and 760 mm Hg (1013.25mb).[31] A humidity value is not given.

If we assume a spherically symmetric atmosphere, Snell's law becomes

$$nr \sin z = n_0 r_0 \sin z_0 \tag{2.3}$$

for the path of the light ray, shown in Fig. 2.2, through the atmosphere to an observer at O, a distance r_0 from the center of the Earth.[31] From a general point P along the path, the zenith angle is z, the radial distance is r, z_0 is the value of z at the observer, and n_0 is the refractive index at the observer.

This chapter discusses four atmospheric refraction models, focusing on refraction at the horizon where its effects are most noticeable.

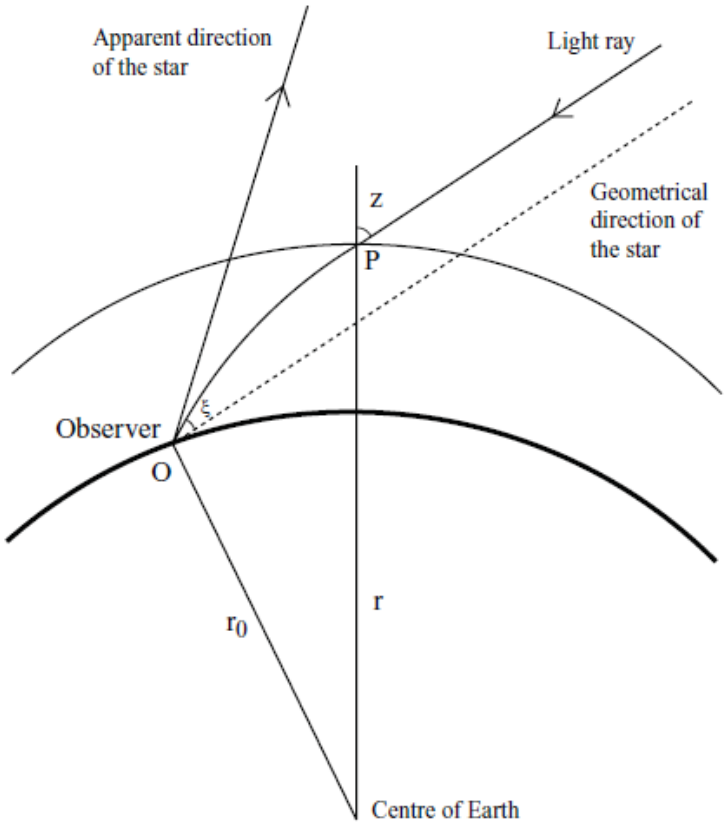


Figure 2.2: Refraction in the Atmosphere Path of a light ray through the atmosphere from Hohenkerk & Sinclair.[7]

2.1 Atmospheric Refraction Models

2.1.1 Standard Refraction

At the horizon, the zenith angle, or the angle between the local zenith and the body's location, is the largest, and light passes through the densest part of the atmosphere at a shallow angle. This means that the strongest effects of refraction happen to

objects at the horizon. The accepted standard value for refraction of light at the horizon is a constant $34'$, meaning that light is bent such that an observer is able to see $34'$ beyond the horizon.[32] This value has been accepted as standard since the 17th century and appears to have been determined by observations, though the meteorological conditions and the location of the observations are unclear.[33] [34] A text from 1733 by Halley includes a table of atmospheric refraction values at a range of apparent altitudes.[35] While the author of the table is unnamed, indications are that Newton is the origin of the $34'$ value.(J. Hilton, private communication, 2018) Texts as far back as 1865 cite it as the mean result of refraction at the horizon using the standard conditions of 50°F (10°C) and 29.6in Hg (1002mb) for temperature and pressure. Wavelength of light, lapse rate, and humidity appear to be lacking from early calculations, but Newcomb cites a lapse rate of 6°.5C per km in 1902 as a generally accepted standard and that an average spectral wavelength from astronomical observations of 590 nm (yellow light) is used.[34] In all cases, the observer is assumed to be at sea-level.

2.1.2 Garfinkel

An analytical model which became the standard used by the British and American Nautical Almanac Offices until the 1980's, was developed by Garfinkel in 1944, and then later modified in 1966 when computers became available for calculations. The

model was one of the first to use meteorological conditions for verification.[36] It is based on a polytropic atmosphere, meaning that the atmosphere is in hydrostatic equilibrium with a constant, nonzero lapse rate. The model takes input of zenith distance, pressure, temperature, altitude above sea level, temperature gradient, and some geophysical constants such as the refractive index.

Garfinkel's algorithm at sea-level for conditions of 0°C, 760mm Hg (1013.25mb), and a temperature gradient of $-0°.005694$C/m produces a refraction value of $2206''.5$ ($36'.8$) at the horizon. The model does not include humidity, so it is assumed to be 0%.[37] In a technical note published in 1985, Hohenkerk and Sinclair ran Garfinkel's algorithm using updated standard conditions of 1010mb and 10°C, and produced refraction value of $2068''.28$ ($34'.5$) for at the horizon, which is much closer to the standard refraction value of $34'$.[7]

Because of its computational complexity, Garfinkel's model has been largely superseded by one derived by Sinclair. The derivation adopted the methods of Auer and Standish, removing the asymptote at the horizon. It is discussed further in Section 2.1.3. Garfinkel's model, while worth mentioning, will not be considered further in this analysis.

2.1.3 Hohenkerk & Sinclair

As an alternative to Garfinkel, Hohenkerk and Sinclair (H&S) of Her Majesty's Nautical Almanac Office (HMNAO) developed an approach to computing refraction that combined the numerical methods of Auer and Standish with an atmospheric model by Sinclair, which was published in a technical note in 1985.[7] [38] [39]. This approach considers local meteorological conditions (temperature, pressure, relative humidity, and temperature lapse rate) as well as the location of the observer (height and latitude). They also provided FORTRAN-77 code to implement their model. The details of the computation can be found in the original technical note, but the general formulae can be found here.[7] According to Snell's law, the bending of the light ray relative to a fixed direction as it passes through zone $r + dr$ of Fig. 2.2, in which the refractive index changes from $n + dn$ to dn is

$$\xi = \int_1^{n_0} \frac{\tan z}{n} dn = \int_\infty^{r_0} \frac{\tan z}{n} \frac{dn}{dr} dr = \int_0^{z_0} \frac{\tan z}{n} \frac{dn}{d(nr)} \frac{d(nr)}{dz} dz \tag{2.4}$$

where the refractive index outside the atmosphere is 1. Transforming Eqn. 2.4 so that z is the independent variable prevents the integrand from becoming infinite at large zenith angles. From Eqn. 2.3, nr is a function of z, so

$$\frac{d(nr)}{dz} = -nr \cot z$$

and

$$\frac{dn}{d(nr)} = \frac{dn}{dz} \Big/ \frac{d(nr)}{dz} = \frac{dn/dr}{n + rdn/dr}$$

This leads to

$$\xi = -\int_0^{z_0} \frac{rdn/dr}{n + rdn/dr} dz \tag{2.5}$$

where dn/dr can be calculated solving Eqn. 2.3 for r.[7]

This atmospheric model used is that described in Sinclair (1982) and assumes [39]:

1. Up to the tropopause (11km), the lapse rate is constant. In the stratosphere, the lapse rate is 0.

2. The atmosphere obeys the perfect gas law for dry air and water vapor separately, as well as for the combined mixture of the two.

3. The atmosphere is in hydrostatic equilibrium.

4. The relative humidity is equal to its value at the observer throughout the troposphere. In the stratosphere, it is assumed to be 0.

In 2008, H&S updated their previous technical note.[25] In particular, they modified the values used for the acceleration due to gravity and other atmospheric constants and implemented some bug fixes. The improved acceleration of gravity is the most significant change and strongly affects objects near the horizon, as it affects the density of the atmosphere near the surface of the Earth. Although HMNAO has not formally

published the new version, this dissertation uses that version for computation and analysis; a copy of the associated code is available in Appendix A.3. Under the accepted standard conditions of 1010mb of pressure, temperature of 283°.15K at the ellipsoid (effectively the Earth's surface), 0% humidity, lapse rate of $-0°.0065\text{Km}^{-1}$ and light at wavelength 0.50169μm, H&S produces a horizontal refraction value of $2041''.04$ ($34'.0$).

One limitation of this algorithm worth noting is in the range of lapse rate values which can be used as input. The lapse rate is used to describe the temperature profile of the atmosphere, which then defines how the light ray is bent. If the magnitude of the lapse rate value is too high (on the order of $0°.01\text{Km}^{-1}$, such as that reported in by Sampson), the refraction value never converges, and the code outputs NaN (not a number) as it exceeds the physically limited value of 9.8Kkm^{-1}.[1]

2.1.4 Bennett

A simple low-accuracy formula for calculating refraction was developed by Bennett in 1982 based on Snell's law and modified empirically to fit the values produced by Garfinkel, the standard at the time. To eliminate the domain error that would occur

at large zenith angles, Bennett began with

$$R = k \cot h_{\odot}$$

where R is the amount of atmospheric refraction, $h_{\odot}$ is the object's observed altitude (in degrees), and k is as above in Eqn. 2.2. Temperature and pressure are assumed to be 10°C and 1010mb respectively. If R is calculated in arcminutes, k is close to unity.[26]

To get better agreement with the values produced by Garfinkel's model, Bennett modified the above equation empirically, resulting in

$$R = \cot\left(h_{\odot} + \frac{7.31}{h_{\odot} + 4.4}\right) \tag{2.6}$$

for standard atmospheric conditions. This produces a horizontal refraction value of $34'.4$.

For non-standard conditions, Bennett provides another, presumably empirically-determined, formula

$$R_{new} = \frac{P - 80}{930}\left(\frac{1}{1 + 8 \times 10^{-5}(R + 39)(T - 10)}\right)R$$

where R is as above in Eqn. 2.6 and P is pressure (mb), which agrees with Garfinkel

within $0'.2$ in the temperature range of -20 to $40°$C and a pressure range of 970 to 1050mb.[26] Rather than use this equation for non-standard conditions, this work uses a modification of Eqn. 2.6 which applies the ideal gas law to allow for other pressure and temperature values, and gives good agreement with the low-accuracy formulas provided in the *Explanatory Supplement to the Astronomical Almanac.* From the ideal gas law, ρ, the air density, is proportional to $\frac{P}{T}$, so

$$\frac{\rho_{new}}{\rho_{old}} = \frac{P_{new}T_{old}}{P_{old}T_{new}}$$

If we add the values above, this gives us a pre-factor of

$$\frac{P(283K)}{(1010mb)(T+273)} = \frac{0.28P}{T+273}$$

and Bennett becomes

$$R = \left(\frac{0.28P}{T+273}\right)\cot\left(h_\odot + \frac{7.31}{h_\odot + 4.4}\right)$$

If k is converted to degrees, Bennett's equation can be further modified to

$$R = \left(\frac{0.28P}{T+273}\right)\frac{0°.0167}{\tan(h_\odot + 7.31/(h_\odot + 4.4))} \tag{2.7}$$

which appears as Eq. 7.90 in the *Explanatory Supplement to the Astronomical Almanac.*[40] For standard conditions, this equation produces a horizontal refraction

value of $34'.4$. Starting with the 2004 edition, the *The Nautical Almanac* refraction tables were generated using the H&S model with updated standard conditions of [7]

T	=	10°C
P	=	1010mb
humidity	=	80%
wavelength of light	=	0.50169μm
latitude	=	45°.

For better agreement with the new refraction tables, Hohenkerk updated Eqn. 2.7 using these parameters to obtain

$$R = \left(\frac{0.28P}{T+273}\right)\frac{0°.0167}{\tan(h_\odot + 7.32/(h_\odot + 4.32))} \tag{2.8}$$

which appears as Eq. 7.93 in the *Explanatory Supplement*, henceforth referred to as Bennett-NA.[41] It produces a horizontal refraction value of $33'.8$ and is used as the first alternative to $34'$ for the remainder of this document.

The Bennett-NA refraction algorithm does have limitations when determining refraction values below the horizontal both due to the nature of the $\tan^{-1}$ function, and the singularity that would occur at an altitude of $-4°.32$. This singularity would only occur if the observer was around 17km above the local terrain while observing sunrise or sunset, which is higher than most commercial planes fly, and can be ignored. However, the nature of the $\tan^{-1}$ function requires some attention. Fig. 2.3 shows the refraction outputs for Bennett-NA, given altitude above the horizon, and

a given set of temperature and pressure conditions. The temperature and pressure conditions were chosen to represent standard conditions, and the average conditions for the Edmonton and Mount Wilson data sets discussed in Chapter 4. As Fig. 2.3 shows, if the apparent altitude of the Sun is lower than 1°.6 below the horizontal, the Bennett-NA refraction outputs begin to decrease. This would occur if the observer was about 2500m above local terrain while observing sunrise or sunset. While this may not be a common occurance, this behavior does not represent real-world conditions, and care should be taken when using the Bennett-NA algorithm when the observer is high above the local terrain.

2.2 Dip

The distance an observer can see on the horizon is not only affected by refraction, but also how high above the surface of the Earth they are. In Fig. 2.4, the observer's geocentric horizon corresponds to the horizontal dashed line through O, while the geometric horizon lies an angle d_g below that. This is called the geometric dip of the horizon. The angle of geometric dip can be calculated using

$$\cos d_g = \frac{R_\oplus}{R_\oplus + h} \tag{2.9}$$

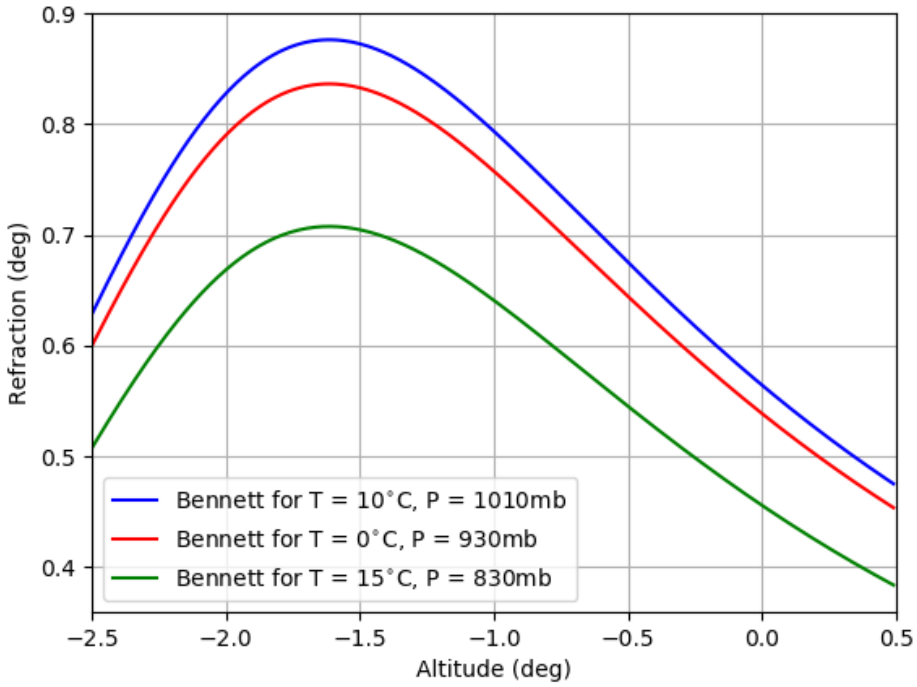


Figure 2.3: Bennett-NA Refraction Atmospheric refraction at height of celestial object. The top line indicates standard conditions, the middle represents conditions in Edmonton, Canada, and the lower line those from Mount Wilson, CA

where $R_\oplus$ is the radius of the Earth, and h is the observer's height above the geometric horizon.[8]

However, because refraction affects everything viewed through the atmosphere, there is a difference between geometric dip and apparent dip, or the full distance an observer can see due to their height and refraction. As can be seen from Fig. 2.5, and as we have noted in Section 2.1.1, refraction allows an observer to see further than they would be able to without it, due to the curvature of the Earth. In this case, apparent

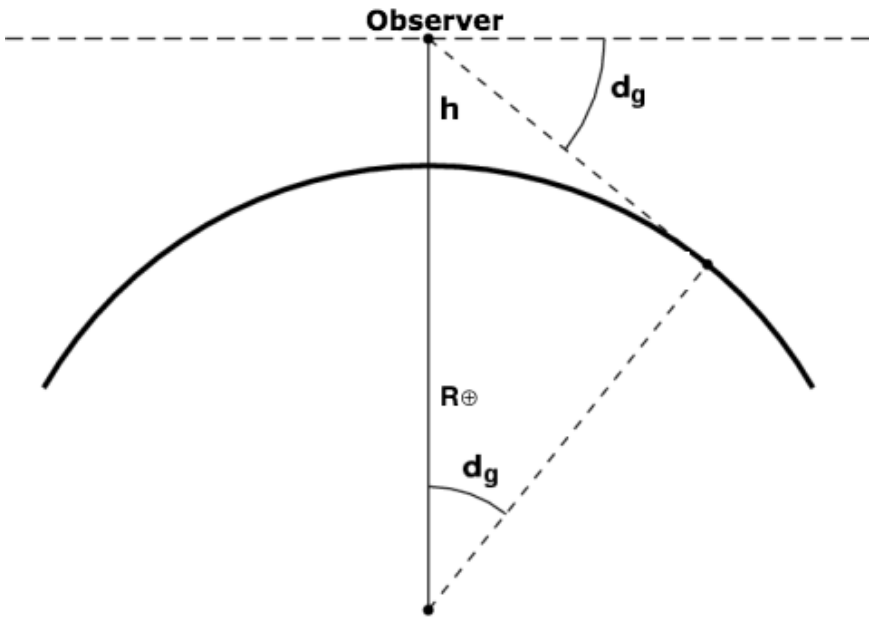


Figure 2.4: Geometric Dip The angle, d_g between the geometric horizon and the visible horizon, without accounting for refraction.[8]

dip can be calculated by

$$d = d_g - R$$

where the refraction angle, R, is negative when compared to the geometric horizon. Since the refraction associated with dip is not always known, Eqn. 2.10 has been developed using the *Standard Atmosphere*.[8]

$$d(h) = 1'.75\sqrt{h} \tag{2.10}$$

where h is in meters.[42]

Determining the angle of atmospheric refraction is no easy task as the meteorological conditions in the atmosphere are inconstant. The models discussed here can not

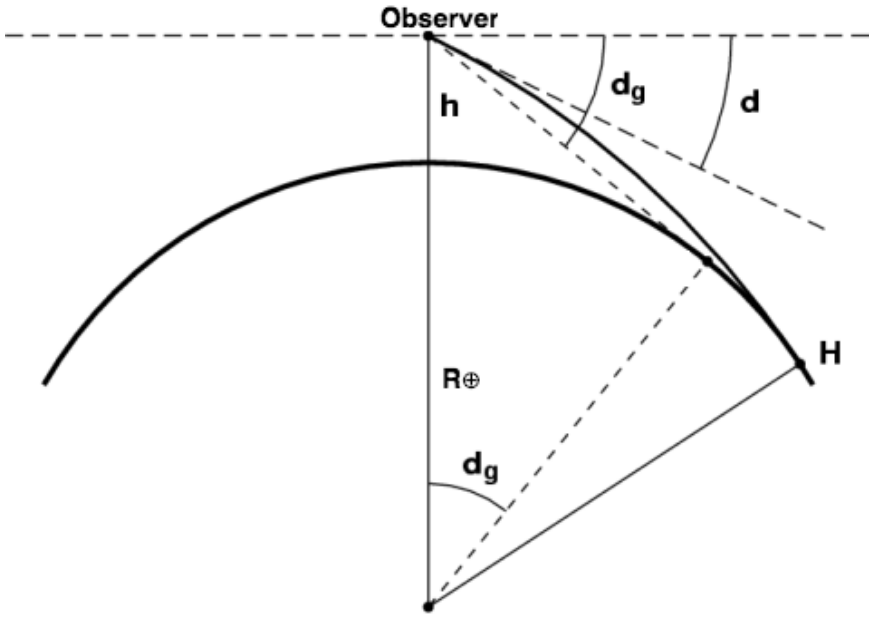


Figure 2.5: Apparent Dip The angle, d, between the geometric horizon and the visible horizon H, accounting for refraction.[8]

account for non-standard phenomena such as inversion layers because of their underlying assumptions. The following chapters discuss the models' implementation in a sunrise/set calculator, and then analyze how well the predictions using these models compare to actual observations.

Chapter 3

The Ultimate Rise/Set Algorithm

To investigate the effects of refraction on times of sunrise or sunset thoroughly, a prediction program is needed that outputs times to the nearest second, uses a rigorous ephemeris and ΔT model[1], has pole-to-pole coverage, and the ability to incorporate meteorological data to account for atmospheric conditions. None of the current publicly available tools is satisfactory for reasons that will be explained in greater detail in Section 3.1. To meet the above requirements, the author developed a new calculator using Python-2.7: "The Ultimate Rise/Set Algorithm (URSA)." Its design facilitates switching atmospheric refraction modules to compare their performance with observations, which is necessary to begin understanding the discrepancies between predicted and observed times. Three refraction models of increasing complexity for use with

[1] $\Delta T = TT - UT1$ where TT is Terrestrial Time and UT1 is Universal Time. ΔT is important for accurate timing and will be discussed in further detail in Section 3.2.2

the prediction program are discussed in Section 3.2.3. Their efficacy is evaluated in Chapter 4.

3.1 Assessment of Existing Calculators

While many sources report sunrise and sunset times, only a few unique prediction algorithms are in common use. A survey of 76 different on-line tools, discussed in Section 3.1.2, revealed that there are three widely-used algorithms.[43] A total of seven existing calculators were surveyed for their compliance with the requirements outlined above: the three found in the survey, two previous U.S. Naval Observatory (USNO) standards, National Aeronautics and Space Administration (NASA) Jet Propulsion Laboratory's (JPL) HORIZONS, and the Astropy library. They were chosen either because of their popularity and widespread use, or their reputation as an authoritative source. Below, each of them is discussed in detail.

3.1.1 Standards/Community Suggestions

U.S. Naval Observatory Standard Lunar-Solar Model (SLSM)

The USNO Standard Lunar-Solar Model (SLSM) is a set of sunrise, sunset, moonrise, and moonset tables intended to serve as a standard that other developers could use to

validate their prediction code.[44] The USNO Standard Lunar-Solar Model Algorithm (SLSMA) generated these tables.[2][44] Although parts of the original algorithm may no longer be considered the best practices (J. Bangert, 2017, private communication), the USNO still uses the SLSM as part of its verification procedures for any sunrise and sunset prediction product. Written in FORTRAN-77, the SLSMA computes the topocentric apparent horizon coordinates of the Sun every two minutes and uses inverse interpolation to determine the time at which the Sun rose or set. It uses a high-accuracy, but dated, ephemeris, NASA JPL's Developmental Ephemeris (DE) 200 and interpolates ΔT from an annual list.[45][46] Updating the SLSMA would require modifying its topocentric apparent place subroutines in accordance with the 1997 and 2000 International Astronomical Union (IAU) resolutions on astrometry, switching its ephemeris to one of the current standards (*e.g.,* DE405 or DE430), and improving its ΔT processing.[47][48][10]

USNO Public Affairs Program 7.1

The USNO Public Affairs Program (PAP) is the latest version of a FORTRAN-77 program that began in the 1980s[3].[49][50] An International Business Machine (IBM) mainframe ran the original version to generate "certified" letter files, which could be printed with a typewriter terminal connected to the mainframe by an RS-232 connection. Automating the calculation of sunrise and sunset times for court cases was the

[2]SLSMA is a more recent name for the Military Operations Analysis Rise Set Program (MOARSP), which was colloquially called the "Mother of All Rise Set Programs."

[3]or possibly earlier; the original developers did not produce any external documentation and the details have faded in the memories of those who remain

original goal of the program. At the time, the staff of old Nautical Almanac Office were manually producing 1 or 2 of these documents per day. (G. Kaplan, private communication, 2017) PAP formed the computational engine of the "Complete Sun and Moon Data for One Day" data service until a new version based on the Solar-Lunar Almanac Core 2.0 (SLAC 2.0, see below) was released in 2014 ; the current version has a more sophisticated user interface, including an Application Programming Interface (API).[51][52]

PAP uses an internal, analytic ephemeris and interpolates or extrapolates ΔT from a list of measured values spaced five years a part. While the underlying algorithm is a good starting point, both the coding language and the format of input variables make updating it difficult. In addition, it does not natively support alternative refraction models. This program served as a model for URSA[4] and was used to validate the output phenomena times.

USNO Solar-Lunar Almanac Core 2.0

The Solar-Lunar Almanac Core 2.0 (SLAC) is a library of C-language functions used in creating the computational engine underlying the "Complete Sun and Moon Data for One Day" data service.[51][52] SLAC steps through a given time period at intervals of either 1 hour, most commonly, or 4 minutes, when conditions exist for the Sun

[4]The USNO does not release the source code for its on-line data services. However, the USNO permitted the author to review this resource during her participation in the Naval Research Enterprise Internship Program (NREIP). The original developer, G. Kaplan (USNO ret.), also provided personal notes and recollections relating to the development of PAP.

or Moon to graze the horizon. SLAC is designed so that it can be used by an observer moving on a platform with respect to the Earth's surface. At each step, SLAC updates the position of the observer taking into consideration any movements during the interval. SLAC then computes the topocentric zenith distance of the Sun or Moon including the contribution of the users preferred atmospheric refraction model. Next, SLAC compares this zenith distance with the zenith distance of the horizon, which may be depressed (i.e. the zenith distance is greater than 90°). If a rise or set occurs during the interval, SLAC uses inverse interpolation to obtain a double precision Julian date, which gives time with micro-second precision. However, the version publicly available as the "Complete Sun and Moon Data for One Day" service truncates the time outputs to the nearest minute. (M. Chizek Frouard, private communication, 2017)

To optimize the performance of its phenomena calculations on personal computers and embedded systems, SLAC uses a mid-accuracy analytic ephemeris for the Sun.[53][54] It also approximates ΔT using a polynomial fit to historical determinations and current predictions; this model is updated approximately annually based on new determinations and revised predictions. The SLAC library source code is restricted to U.S. government agencies and their contractors.[5] Unfortunately for the purposes of this project, the "Complete Sun and Moon Data for One Day" service is restricted to standard refraction. Although the API provides some flexibility in querying the

[5]The USNO does not restrict distribution of the output from programs using this library, such as"Complete Sun and Moon Data for One Day".

service, a user cannot access the alternative refraction models. This service was also used to validate the output times from URSA.

JPL HORIZONS

HORIZONS is a Solar System data and ephemeris computation service provided by the Solar System Dynamics Group of NASA JPL.[55] HORIZONS does use a high-accuracy ephemeris to calculate rise/set times, and their online service allows for approximate variation in refraction based on the height of the observer due to dip. They use a low level refraction model based on Bennett/Saemundsson and assumes standard atmospheric conditions of "yellow-light observations at 10 deg C sea-level with pressure of 1010 millibars."[56][57] They state that the accuracy of their predictions is less than or equal to twice the requested step-size indicated in the search, which could be as short as 1 minute. This means that at best, rise/set times are ± 2 minutes. Furthermore, the interface is not user-friendly. The documentation that explains how to get sunrise/set times is not easy to find, nor is it intuitive to follow. HORIZONS is a powerful ephemeris tool, but it is not explicitly sunrise/set calculator, which makes it a poor choice for those looking specifically for sunrise/set times.

AstroPy

The Astroplan package of the AstroPy library includes a sunrise and sunset time calculator in the Observer class, Observer.sun_rise_time for sunrises and Observer.sun_set_time for sunsets.[58] It uses the high-accuracy ephemeris DE430, and the algorithm for times of rise and set described in Section 12.3.3 of the *Explanatory Supplement to the Astronomical Almanac*. (E. Downey, private communication, 2018) This formula is accurate between 60° North and South and does not allow for pole to pole coverage.[59] It does allow the user to input the position of the body when it reaches the horizon, but the default position is 0 indicating that refraction is not considered.

3.1.2 Web Survey Findings

To understand what sunrise/set models are available to the general public, a survey was conducted using the three most popular search engines (Bing, Yahoo, and Google) which recovered 76 different websites reporting to predict sunrise/set times for any location.[43] For each of these sites, the origin of the data or the method used to calculate rise/set times was identified. The survey indicated that 75% of websites predict their sunrise/set times using one of three sources: the USNO's "Complete Sun and Moon Data for One Day" API, an algorithm by J. Meeus taken from the text *Astronomical Applications*, and an algorithm by P. Schlyter available as open

source code and implemented in the standard PHP function library.[60][61] How the other websites predicted rise/set times was undetermined.

Both Meeus' and Schlyter's algorithms use

$$\cos HA = \frac{\sin h_0 - \sin \phi \sin \delta}{\cos \phi \cos \delta} \tag{3.1}$$

to determine the time of the phenomenon in question, where HA is the solar hour angle, h_0 is the altitude of the solar center for the phenomenon in question, which includes the standard $34'$ for refraction, ϕ is the latitude of the observer, and δ is the solar declination. This formula is accurate between $60°$ North and South and does not allow for pole to pole coverage due to the domain issue that would occur when $\phi = 90°$.[59] Meeus then determines the time of the phenomenon using α (solar right ascension), and Θ_0 (apparent sidereal time). He includes a correction factor which is a function of δ, ϕ, LHA (local hour angle), h_0, and $h_\odot$ (solar altitude). Iterating with this correction can change the time of sunrise or sunset by as much as 15 minutes.[60] Schlyter's general directions acknowledge that this algorithm is low accuracy but suggest iterating over α and δ to decrease the error of HA.[61] Both sources include a low-accuracy ephemeris to calculate α and δ.

3.2 The Ultimate Rise/Set Algorithm

The following section discusses the algorithm used in URSA, including the history of the algorithm, the geometry behind the model, and the refraction component.

3.2.1 Relationship to Antecedents

The USNO PAP program discussed in Section 3.1.1 was translated from its original FORTRAN-77 to Python 2.7. Use of a more modern programming language makes the underlying algorithm accessible to those not familiar with Fortran code. It also allows the program flexibility when it comes to components like the ephemeris, ΔT, or refraction model, as the functions can be easily modified. Although the geometry of the underlying model is still valid, several of the components were updated: the ephemeris functions, the ΔT model, and the refraction-calling sequence.

In place of the internal analytic ephemeris used by PAP, the URSA computes α and δ using the Naval Observatory Vector Astrometry Software Python Edition 3.1.1 (NOVAS Py 3.1.1), which accesses the full JPL DE405.[62][48]

ΔT is the time difference between Terrestrial Time (TT) and UT1. TT is used as the time reference for apparent geocentric ephemerides. It is a coordinate time scale

that can only be determined after-the-fact.[32] UT1 is a conventional astronomical timescale defined by the rotation of the Earth with respect to the Sun, and is nominally equivalent to mean solar time from midnight on the Greenwich meridian.[32] Coordinated Universal Time (UTC) provides the basis of civil time, and seconds of UTC are SI seconds. To ensure that $|UT1 - UTC| < 0.9s$, leap seconds adjustments are occasionally made to UTC.[32]

PAP uses ΔT values taken from a list of measured values spaced once every five years. URSA uses a text file of ΔT values provided monthly by the USNO's Earth Orientation Department (EOD) since 1973.[63] They currently provide quarterly predictions for ΔT from the present through 2024, which are updated annually. EOD also provides historic ΔT values from 1657 to 1984.[64] ΔT currently increases about 1 second each year. The aim of this study is to attribute any error in sunrise/set prediction times to variation in atmospheric refraction, so a more precise ΔT value was used.

The biggest change in URSA is the ability for the program to incorporate values for the refraction at the horizon calculated independently with the user's algorithm of choice. All other rise/set prediction tools considered here use the standard value for refraction on the horizon (34'). This flexibility allows for a determination of the accuracy of each atmospheric refraction model used. The accuracy of the refraction models will be discussed in Chapter 4.

3.2.2 Prediction Algorithm

The algorithm presented calculates the times of sunrise and sunset. Like Meeus' and Schlyter's prediction algorithms, this model uses an iterative method based on an initial guess of when the phenomenon occurs. The initial guess here is based on hours from the Julian date of solar transit (∓ 6 hrs for rise and set respectively) which is calculated separately. It takes α, δ, ΔT, and Greenwich Sidereal Time (GST) as inputs, and approximates the rate of change of the solar hour angle (HA), α, and δ. The apparent altitudes at which the phenomena occur are known and the program uses these positions to iterate over computed times at which they may occur using a Newton-Raphson method to determine the correct time.

The algorithm then obtains the altitude of the phenomenon ($h_\odot$) using the equation[6]

$$\sin h_\odot = \sin\phi \sin\delta + \cos\phi \cos\delta \cos HA \tag{3.2}$$

where ϕ is the latitude, λ is the longitude, and where[59]

$$HA = (GST \times 15.0 + \lambda - RA) \mod 360.0. \tag{3.3}$$

Eqn. 3.2 is a re-arrangment of Eqn. 3.1 that no longer encounters the domain issue when ϕ is 90°.

3.2.3 Refraction

The altitude for sunrise or sunset must include the atmospheric refraction at the horizon to achieve any level of accuracy in phenomena times. The above algorithm calculates the position of the center of the Sun, which means at sunrise and sunset, the refraction at the horizon must be added to the semi-diameter of the Sun to predict accurately when the phenomenon will occur. URSA uses a fixed value of $0°.266667$ for the solar semi-diameter. A brief discussion of some of the models used are presented here.

Standard Refraction The most commonly accepted value for refraction at the horizon is a constant of $34'$. This is the default refraction value for URSA; that is, unless the user provides another value, the code will apply $34'$.

Bennett-NA The atmospheric refraction model developed by Bennett and modified for the NA takes the height of the observed altitude of the object (deg), temperature ($°C$), and pressure (mb), as input.[26] [40] If conditions are unknown, the standard temperature and pressure of $10°C$ and 1010 mb are set as a default. The model has been implemented in Python-2.7 and is provided in Appendix A.4. Refraction values are output to a file which can then be used when running URSA.

Hohenkerk & Sinclair of HMNAO The more sophisticated atmospheric refraction

model written in FORTRAN by Hohenkerk & Sinclair HMNAO takes the observed zenith distance of the object (deg), the pressure (mb), temperature (K), and relative humidity at the observer, the wavelength of light ray (μm), the lapse rate (Km^{-1}), the height of the observer above sea level (m), the latitude of the observer (deg), and the precision required ($'$).[25] All inputs must be supplied by the user. Code for the model is provided in Appendix A.3. Refraction values are output to a file which can then be used when running URSA.

3.2.4 Using the Program

The program is written as a function that takes integers for the Gregorian year, month, and day, and floating point real values for the time-zone, longitude (+ for East, - for West), latitudes (+ for North, - for South), and refraction. The time-zone input is optional; the program uses UT1 unless otherwise specified. The refraction value is also optional; the program uses the standard $34'$ unless otherwise specified. The program requires that the latest version of the Python package NumPy be installed (currently v1.13[6]), as well as NOVAS Py 3.1.1. The program also requires a value for ΔT. Phenomena times are output as a Python dictionary with the Gregorian date and time returned as a tuple: beginning of civil twilight, sunrise, solar transit, sunset, end of civil twilight. Although civil twilight times are not required for the current

[6]URSA has only been tested with v1.13, but should run with all future versions. It is unknown whether earlier version are compatible.

analysis, they are computed for consistency and comparison with PAP and may be useful for future studies of twilight conditions as discussed in Section 3.4.

To ensure the accuracy of the outputs are not affected by ΔT, the program requires a text file of ΔT values. These values are the same as those mentioned in Section 3.2.1 available from the EOD. The current list of ΔT values is accurate from 1900 to the present and should be updated once per year to maintain effectiveness. The program interpolates between the two ΔT values closest to the date in question for use in the sunrise/set calculator.

3.2.5 External Code Review

James Hilton (USNO) reviewed the program for scientific soundness. He recommended using a more accurate date conversion and made notes in two places that will further increase the precision of the calculation and generalize it for future use (J. Hilton, private communication, 2017). Both of these suggestions will be discussed in Section 3.4. Eric Barron (USNO) reviewed the program for compliance with good Python programming practice and made some recommendations that would streamline some of the calculations. (E. Barron, private communications, 2017) All findings were addressed to their satisfaction. (J. Hilton, private communication, 2017; E.

Barron, private communication, 2017) Both gentlemen are members of the Nautical Almanac Office (NAO) within the Astronomical Applications Department, but neither was extensively involved with the development of this program.

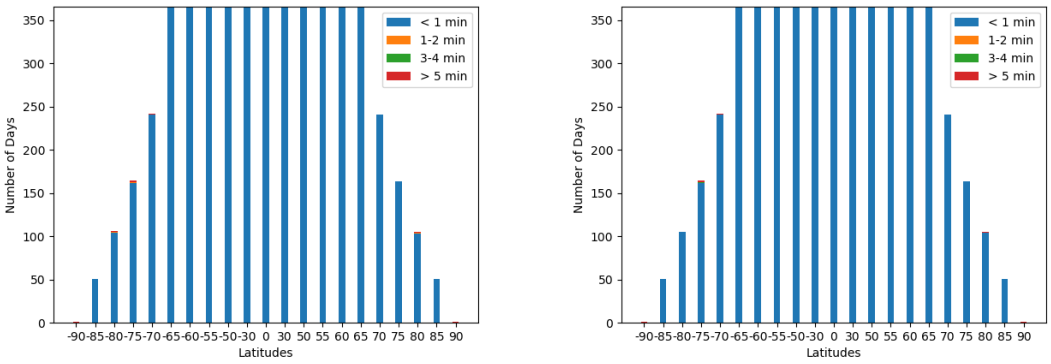
3.3 Calculator Comparisons

To ensure the accuracy of URSA, its outputs were validated against the USNO standards discussed in Section 3.1. The outputs were also compared to the sunrise/set times produced with implementations of Meeus' and Schlyter's algorithms.

3.3.1 Comparison with Standards

The sunrise/set times from SLSM were available in tables for the year 1992 at 21 different latitudes from 90°N to 90°)S: 0°, 30° N and S, and every 5° from 50° to 90° N and S. As can be seen in Fig. 3.1, except for a select number of days, URSA produced values that were within a minute of the times predicted by SLSM. While it did not produce times for rise or set at the poles, it did agree with SLSM on which days the Sun was continuously above or below the horizon. In fact, in this comparison, the dates on which there is a disagreement of greater than 5 min, SLSM reported a phenomena and URSA did not. However, under the circumstances when

two rises or sets are predicted to occur on the same day URSA did correctly produce phenomenon times.



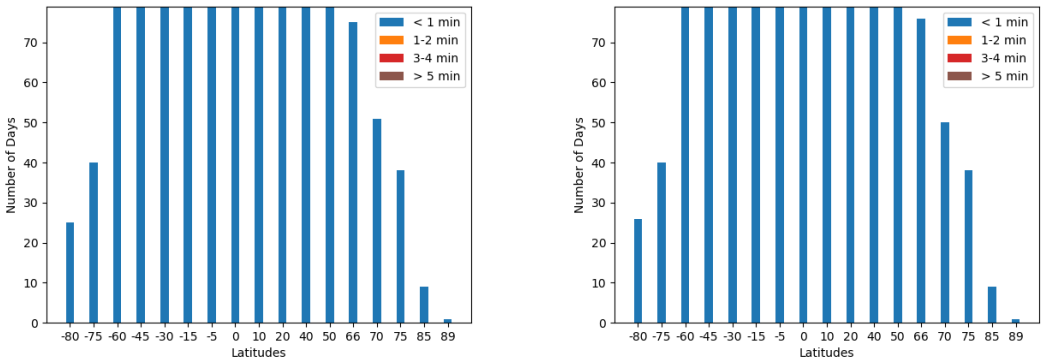
(a) URSA vs SLSM (Rise)

(b) URSA vs SLSM (Set)

Figure 3.1: URSA vs SLSM These plots show the absolute time differences between the sunrise/set outputs of SLSM and URSA

URSA was validated against the original FORTRAN-77 PAP on which it was based. Sunrise/set predictions at 17 locations between 80°S and 89°N for 8 days a year for 10 years between 2000 and 2024 were queried. Dates and locations are listed in Table 3.1. The days were chosen to represent 2 days per season each year and included extreme conditions, *i.e.*, a set of dates at locations within the Arctic or Antarctic circles on which only a sunrise or sunset was expected. Single phenomenon dates were chosen using Multiyear Interactive Computer Almanac (MICA).[65] The mean difference in outputs was 0.6 ± 1.4 sec for sunrises and 0.6 ± 0.9 sec for sunsets. The calculators agreed within a minute of each other for all dates queried, and in fact agreed to the second on all except two of the date-location combinations queried: one day at 89°N

on which they differed by 9 seconds, and the other at $80°S$ on which they differed by 42 seconds. Given how high these latitudes are, these differences are insignificant.



(a) URSA vs PAP (Rise)

(b) URSA vs PAP (Set)

Figure 3.2: URSA vs PAP These plots show the absolute time differences between the sunrise/set outputs of PAP and URSA

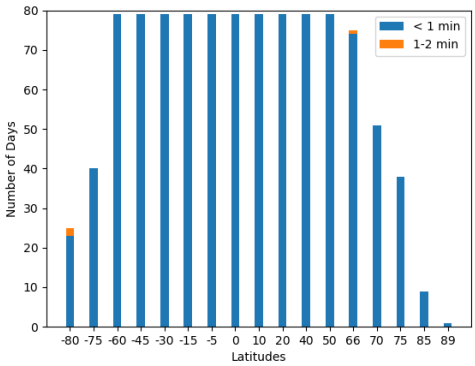
URSA was tested against SLAC 2.0 as implemented by the "Complete Sun and Moon Data for One Day" API. The API was queried using the same set of locations and dates that was used to validate URSA against PAP. Code querying the USNO API is provided in Appendix A.7. Again, as Fig. 3.3 shows, except for a select number of sunrise times above the Arctic and Antarctic circles, the calculators agreed within a minute of each other. In those select cases, the calculators agreed to within 2 minutes.

Table 3.1
Locations and Dates used for URSA Validation All combinations were tested

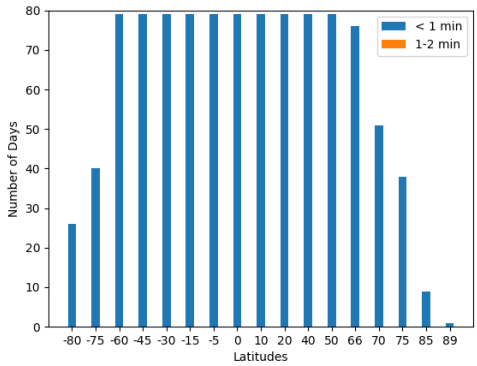
Locations	Dates				
80.0S, 0.0E	01/27/2000	02/25/2006	04/28/2010	05/26/2016	06/25/2022
75.0S, 0.0E	03/06/2000	04/05/2006	06/14/2010	07/13/2016	08/12/2022
60.0S, 0.0E	04/23/2000	05/23/2006	07/24/2010	08/31/2016	09/30/2022
45.0S, 0.0E	01/06/2000	07/01/2006	09/10/2010	10/09/2016	11/17/2022
30.0S, 0.0E	07/19/2000	08/18/2006	10/28/2010	11/26/2016	01/04/2023
15.0S, 0.0E	09/06/2000	10/05/2006	12/15/2010	01/10/2019	02/13/2023
5.0S, 0.0E	10/15/2000	11/23/2006	01/04/2014	03/01/2019	03/23/2023
0.0N, 0.0E	10/15/2000	01/07/2009	02/21/2014	04/18/2019	05/10/2023
10.0N, 0.0E	11/23/2000	02/24/2009	04/02/2014	06/05/2019	06/28/2023
20.0N, 0.0E	01/17/2003	04/04/2009	05/19/2014	07/24/2019	08/15/2023
40.0N, 0.0E	03/06/2003	05/13/2009	07/08/2014	09/10/2019	09/24/2023
50.0N, 0.0E	04/24/2003	06/30/2009	08/25/2014	10/29/2019	11/11/2023
66.0N, 0.0E	06/02/2003	08/18/2009	10/12/2014	12/07/2019	
70.0N, 0.0E	07/19/2003	10/05/2009	11/30/2014	01/01/2022	
75.0N, 0.0E	08/28/2003	11/23/2009	01/11/2016	02/18/2022	
85.0N, 0.0E	10/06/2003	01/01/2010	02/29/2016	03/29/2022	
89.0N, 0.0E	11/23/2003	02/18/2010	04/17/2016	05/16/2022	

3.3.2 Comparison with Other Known Calculators

Using the findings of the survey discussed in Section 3.1.2, implementations of Meeus' and Schlyter's algorithms were selected. Schlyter's algorithm has been implemented in the standard PHP library using the date_sunrise and date_sunset functions. The algorithm is used in several on-line calculators of which the sunrise-sunset.org site was selected as representative because it is API-enabled. The website sunrise-sunset.org API was queried using the same set of locations and dates that was used to query the USNO API.[66]. Code querying this API is provided in Appendix A.5. As can be

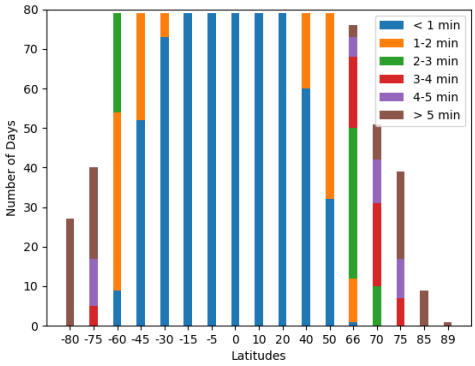


(a) URSA vs USNO (Rise)

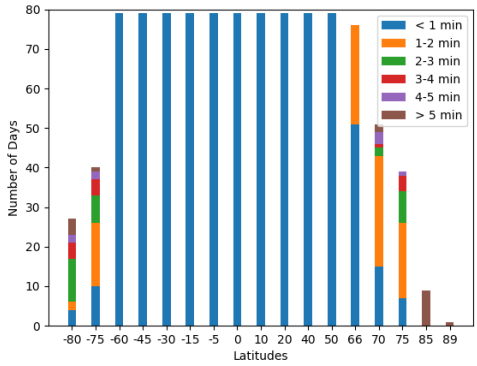


(b) URSA vs USNO (Set)

Figure 3.3: URSA vs USNO These plots show the absolute time differences between the sunrise/set outputs of USNO and URSA



(a) URSA vs Schlyter (Rise)



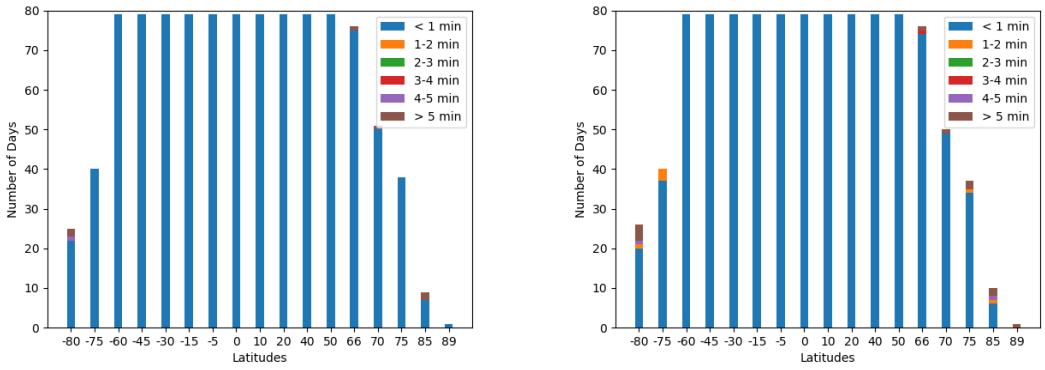
(b) URSA vs Schlyter (Set)

Figure 3.4: URSA vs Schlyter These plots show the absolute time differences between the sunrise/set outputs of Schlyter and URSA

seen in Fig. 3.4, Schlyter's algorithm produced more sunset times that were within a minute of those from URSA than sunrise times, but began to diverge even at low latitudes, around 30°. In the case of days with a single phenomenon, the Schlyter algorithm incorrectly predicted a rise when none of MICA, URSA, or the USNO API

predicted one. The algorithm is also incapable of predicting phenomena at the poles due to the domain error mentioned in Section 3.1.2. However, near the poles on days when no sunrise or sunset occur, it did correctly make the distinction between the Sun being continuously above or below the horizon: 00:00:01 for continuously above, and 00:00:00 for continuously below.

Meeus' algorithm has been implemented in a Python package, pvlib-python using the transit_sunrise_sunset function in the spa library.[67] The same set of dates that was used to query the USNO and Schlyter APIs was used for this function as well. Code querying this package is provided in Appendix A.6. Meeus' algorithm produced



(a) URSA vs Meeus (Rise)

(b) URSA vs Meeus (Set)

Figure 3.5: URSA vs Meeus These plots show the absolute time differences between the sunrise/set outputs of Meeus and URSA

sunrise/set times that were closer to those predicted by URSA than Schlyter's, even at the high latitudes. However, for days on which a single phenomenon occurred, it reported NaN, making no distinction between whether the Sun was continuously

above or below the horizon. It also has the same domain problem that Schlyter's algorithm has when predicting phenomena at the poles.

3.3.3 Refraction Validation

Standard Refraction Computation of the sunrise and sunset tables for *The Astronomical Almanac* uses a fixed refraction term of $34'$.[12] Similar to *The Astronomical Almanac*, PAP, "Complete Sun and Moon Data for One Day," Meeus, and Schlyter also use $34'$ for refraction. Therefore, all of the model inter-comparisons discussed in Section 3.1 and 3.2 use this value. The overall agreement among them indicates that refraction is being implemented properly in URSA.

Bennett-NA Bennett-NA was selected for good agreement with the 2004 and later editions of *The Nautical Almanac*, it was validated by comparison against the tables on pages A2, A3, and A4 of the 2014 NA using apparent altitudes from 10° to 90° with the standard temperature and pressure conditions of $10^\circ C$ and 1010mb, and a set of non-standard conditions with temperatures of $-25^\circ C - 35^\circ C$ and pressure of 980 mb. [68] For the standard conditions, the differences between the tabulated values reported in the NA and revised Bennett were within the accepted error of $\pm 0'.1$, as can be seen in Table 3.2. (C. Hohenkerk, private communication, 2018) Refraction values for non-standard conditions were systematically low for apparent altitudes

above 8°, but matched within −0′.1. For altitudes below that, refraction values varied as much as 3′.7. These differences can be attributed to non-linear effects near the horizon, as well as the fact that the values for non-standard conditions are reportedly good within a "zone" or range of conditions.[69] Calculations show that each "zone" contains conditions that give a range of refraction values within 0′.8. Without better documentation of how the *The Nautical Almanac* non-standard condition table (A4) values were calculated, the values reported this module are considered adequate.

Table 3.2
Comparison of NA and Bennett-NA Refraction Values Standard Conditions

App Alt.	NA reported	Bennett-NA calculated		App Alt.	NA reported	Bennett-NA calculated
°	′	′		°	′	′
0 00	33.8	33.8		5 50	8.7	8.7
0 03	33.2	33.2		5 55	8.6	8.6
0 06	32.6	32.7		6 00	8.5	8.5
0 09	32.0	32.0		6 10	8.3	8.3
0 12	31.5	31.5		6 20	8.1	8.1
0 15	30.9	31.0		6 30	7.9	7.9
0 18	30.4	30.4		6 40	7.7	7.8
0 21	29.8	29.9		6 50	7.6	7.6
0 24	29.3	29.4		7 00	7.4	7.4
0 27	28.8	28.9		7 10	7.2	7.3
0 30	28.3	28.4		7 20	7.1	7.2
0 33	27.9	27.9		7 30	6.9	7.0
0 36	27.4	27.5		7 40	6.8	6.9
0 39	26.9	27.0		7 50	6.7	6.7
0 42	26.5	26.6		8 00	6.6	6.6
0 45	26.1	26.1		8 10	6.4	6.5
0 48	25.7	25.7		8 20	6.3	6.4
0 51	25.3	25.3		8 30	6.2	6.3
0 54	24.9	24.9		8 40	6.1	6.2

Continued on next page

Table 3.2 – continued from previous page

App Alt.	NA reported	Bennett-NA calculated		App Alt.	NA reported	Bennett-NA calculated
°	′	′		°	′	′
0 57	24.5	24.5		8 50	6.0	6.0
1 00	24.1	24.1		9 00	5.9	6.0
1 03	23.7	23.8		9 10	5.8	5.9
1 06	23.4	23.4		9 20	5.7	5.8
1 09	23.0	23.0		9 30	5.6	5.7
1 12	22.7	22.7		9 40	5.5	5.6
1 15	22.3	22.4		9 50	5.4	5.5
1 18	22.0	22.0		10 00	5.3	5.4
1 21	21.7	21.7		10 05	5.3	5.3
1 24	21.4	21.4		10 15	5.2	5.3
1 27	21.1	21.1		10 25	5.1	5.2
1 30	20.8	20.8		10 45	5.0	5.0
1 35	20.3	20.3		10 55	4.9	5.0
1 40	19.9	19.8		11 05	4.8	4.9
1 45	19.4	19.4		11 25	4.7	4.8
1 50	19.0	19.0		11 35	4.6	4.7
1 55	18.6	18.5		11 55	4.5	4.6
2 00	18.2	18.1		12 15	4.4	4.4
2 05	17.8	17.8		12 25	4.3	4.4
2 10	17.4	17.4		12 45	4.2	4.3
2 15	17.1	17.0		13 05	4.1	4.2
2 20	16.7	16.7		13 25	4.0	4.1
2 25	16.4	16.4		13 45	3.9	4.0
2 30	16.1	16.0		14 05	3.8	3.9
2 35	15.8	15.7		14 25	3.7	3.8
2 40	15.4	15.4		14 45	3.6	3.7
2 45	15.2	15.1		15 15	3.5	3.6
2 50	14.9	14.8		15 45	3.4	3.5
2 55	14.6	14.6		16 15	3.3	3.4
3 00	14.3	14.3		16 45	3.2	3.3
3 05	14.1	14.1		17 15	3.1	3.2
3 10	13.8	13.8		17 45	3.0	3.1
3 15	13.6	13.6		18 25	2.9	2.9
3 20	13.4	13.3		18 55	2.8	2.9
3 25	13.1	13.1		19 35	2.7	2.7
3 30	12.9	12.9		20 25	2.6	2.6
3 35	12.7	12.7		21 05	2.5	2.6

Continued on next page

Table 3.2 – continued from previous page

App Alt.	NA reported	Bennett-NA calculated		App Alt.	NA reported	Bennett-NA calculated
°	′	′		°	′	′
3 40	12.5	12.5		21 55	2.4	2.4
3 45	12.3	12.3		22 55	2.3	2.3
3 50	12.1	12.1		23 45	2.2	2.2
3 55	11.9	11.9		24 45	2.1	2.1
4 00	11.7	11.7		25 55	2.0	2.0
4 05	11.5	11.5		26 55	1.9	1.9
4 10	11.4	11.4		28 15	1.8	1.8
4 15	11.2	11.2		29 35	1.7	1.7
4 20	11.0	11.0		31 15	1.6	1.6
4 25	10.9	10.9		32 55	1.5	1.5
4 30	10.7	10.7		34 45	1.4	1.4
4 35	10.6	10.6		36 35	1.3	1.3
4 40	10.4	10.4		37 55	1.2	1.3
4 45	10.3	10.3		41 25	1.1	1.1
4 50	10.1	10.1		43 15	1.0	1.0
4 55	10.0	10.0		46 25	0.9	0.9
5 00	9.8	9.9		50 35	0.8	0.8
5 05	9.7	9.7		54 15	0.7	0.7
5 10	9.6	9.6		58 15	0.6	0.6
5 15	9.5	9.5		62 45	0.5	0.5
5 20	9.3	9.4		67 35	0.4	0.4
5 25	9.2	9.3		73 05	0.3	0.3
5 30	9.1	9.1		78 25	0.2	0.2
5 35	9.0	9.0		84 05	0.1	0.1
5 40	8.9	8.9		88 35	0.0	0.0
5 45	8.8	8.8				

Hohenkerk & Sinclair The transcription of the FORTRAN code for the H&S refraction model was verified using tables provided in the technical note, which contain 47 values at zenith angles from 10° to 90° with two different sets of pressure, temperature, relative humidity, wavelength of light, and lapse rates. Two of the values

in Tables 3.3 and two in Table 3.4 have a 0.01 difference between the calculated and reported times which can be attributed to rounding differences.

Table 3.3
Refraction Value Comparison 1 URSA and H&S

Pressure at observer	:	1005 mb
Temperature at observer	:	280°K
Humidity at observer	:	80%
Wavelength of ray	:	0.574 μm
Lapse rate, α	:	$0°.0065\text{Km}^{-1}$
Height of observer	:	sea level
Latitude of observer	:	50°

Observed zenith angle	H&S reported	H&S calculated
°	″	″
10	10.27	10.27
20	21.19	21.20
30	33.61	33.61
40	48.83	48.83
45	58.17	58.18
50	69.30	69.30
55	82.98	82.98
60	100.53	100.53
65	124.25	124.25
70	158.67	158.67
72	177.56	177.36
74	200.39	200.39
76	229.50	229.50
78	267.51	267.50
80	319.22	319.22

Table 3.4
Refraction Value Comparison 2 URSA and H&S

Pressure at observer	:	1010 mb
Temperature at observer	:	283°.15K
Humidity at observer	:	0%
Wavelength of ray	:	0.50169 μm
Lapse rate, α	:	see tabulation
Height of observer	:	sea level
Latitude of observer	:	50°

Observed zenith angle	H&S reported $\alpha = 0.005694$	H&S calculated $\alpha = 0.005694$	H&S reported $\alpha = 0.0065$	H&S calculated $\alpha = 0.0065$
°	″	″	″	″
75	214.22	214.22	214.21	214.22
76	229.68	229.68	229.68	229.68
77	247.35	247.35	247.34	247.34
78	267.71	267.71	267.70	267.70
79	291.44	291.44	291.43	291.43
80	319.44	319.44	319.43	319.43
81	352.96	352.96	352.93	352.93
82	393.74	393.74	393.69	393.69
83	444.34	444.34	444.25	444.25
84	508.57	508.57	508.42	508.42
85	592.38	592.38	592.09	592.09
86	705.38	705.38	704.78	704.78
87	863.87	863.87	862.54	862.54
88	1097.02	1097.02	1093.79	1093.79
89	1460.42	1460.42	1451.88	1451.88
90	2068.97	2068.98	2044.24	2044.25

3.4 Conclusion & Future Work

Much of the legacy code in the astronomy community is written in older languages such as FORTRAN, which is less commonly learned by the new generation of scientists. To create a new standard, the code needed to be written in a more modern

language such as Python. URSA is written in Python 2.7, and could easily be upgraded to Python 3.0. The code relies on the latest version of NOVAS Py, which currently uses the JPL ephemeris DE405. JPL released DE430 in 2014, but NOVAS is not yet compatible. The upgrade in the ephemeris only changes the sun's apparent position by a small fraction of an arcsecond.[10] URSA will use DE430 when NOVAS is compatible.

Astropy is commonly used in the astronomy community, but uses a low-level accuracy formula for computing sunrise/set times. A detailed comparison between the two calculators would help determine the level of accuracy of the Astropy sunrise/set calculator. While many calculations may not require high-accuracy sunrise/set times, if the need arises, such a comparison would help determine when to switch to a higher-accuracy calculator such as URSA.

URSA currently computes the time of sunrise or set using a fixed value of $0^\circ.266667$ for the solar radius. As mentioned in Chapter 5, the semi-diameter changes slightly throughout the year because of the Earth's elliptical orbit. From perihelion to aphelion, the difference in the solar radius is $0'.5083$, which would affect calculations by 2 seconds. Computing the semi-diameter for the day of the year would eliminate this error.

URSA currently computes the beginning and end of civil twilight for consistency with the original algorithm on which it is based. These values would be useful for

future analyses of illuminance or other refraction studies, and could be supplemented by times of nautical and astronomical twilights. Since URSA is a general rise/set calculator, it could also be used to compute lunar events with some modification.

URSA is based on a former USNO standard algorithm. Common rise/set algorithms such as that published in *Astronomical Algorithms* calculate an approximate time that causes a domain error at the poles. URSA avoids this problem by estimating the time of rise/set from transit for the first iteration. Unlike its predecessors, the code also allows the user to change the value for atmospheric refraction to best fit the conditions of the observations. With the improvements suggested here, URSA may well be on its way to becoming a standard itself.

Chapter 4

Analysis of Historical Sunrise and Sunset Data

In this chapter, observed sunrise and sunset times from 30 different sets of longitude and latitude coordinates are compared with predictions computed with URSA using up to three refraction models. The observations were taken by L. Rarogiewicz, R. Sampson, B. Schaefer, W. Liller, and C. Smiley.[27][1][28][29] Single observations were taken at 21 of these locations: 14 from the middle of the Atlantic Ocean recorded, and 7 from various beaches in Hawai'i. The data sets from Mount Wilson, CA and Edmonton, Canada both contain 2 different locations within $10''$ of each other. Of the remaining 4 locations, 3 are in Chile, with the final observing site in Providence, RI. Observations from Edmonton and Rhode Island were made with land horizon;

all other observations were made with a sea-level horizon. The availability of meteorological data for a particular site and date determined the number of refraction models used when computing predicted sunrise or sunset times. Specific coordinates are provided in Table 4.1. When height of eye is unknown, it is noted as 0m.

The main purpose of this analysis is to determine how well the refraction models discussed in Chapter 2 predict sunrise and sunset times when compared to observations, and how differently they produce results when compared to each other. The question of the importance of including dip will also be discussed. Ignoring dip, as is done with those observations that did not include height information, may introduce a systematic error in these results, especially for those with a water horizon. The first few meters above the ground are important when calculating dip, which goes as the square root of height, as stated in Eqn. 2.10.

4.1 Mount Wilson Observatory, CA

Sunset observations over the Pacific ocean, during which possible observations of the Novaya Zemlya (NZ) effect were noted, were made available by the USNO in an unpublished technical note.[27] Lucian Rarogiewicz [1] made a total of 250 sunset

[1]Rarogiewicz held several part-time positions on Mount Wilson, including (paid) weather observer, postmaster, and observer on the Mark III Steller Interferometer (G. Kaplan, private communication, 2017).[70]

Table 4.1
Observation Site Information Includes Geographic coordinates and height of eye above the horizon

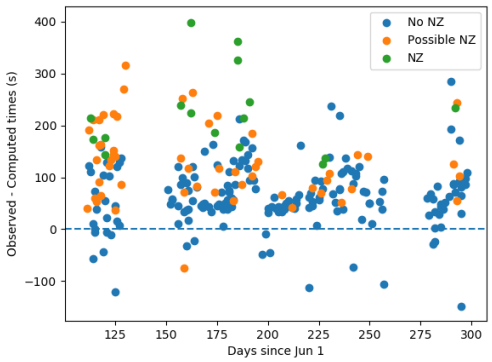
Observation Set	Latitude	Longitude	Height (m)
Mount Wilson	34°13′34″.6 N	118°03′56″.2 W	1739
	34°13′37″.0 N	118°03′56″.2 W	1725
Edmonton, Canada	53°31′34″ N	113°29′14″ W	22
	53°31′34″ N	113°29′14″ W	39
	53°31′41″ N	113°31′10″ W	50
Hawai'i	21°16′40.8″ N	157°49′48″.0 W	5
	20°55′55″.2 N	156°41′45″.6 W	40
	19°37′40″.8 N	155°59′31″.2 W	3
	19°33′46″.8 N	155°57′54″.0 W	3
	19°25′4″.8 N	155°53′6″.0 W	280
	19°56′9″.6 N	155°41′24″.0 W	800
	19°49′22″.8 N	155°28′19″.2 W	4205
Chile	32°57′0″.0 S	71°33′0″.0 W	200
	29°54′0″.0 S	71°18′0″.0 W	300
	30°9′54″.0 S	70°48′54″.0 W	2215
Atlantic	10°32′36″.0 N	58°22′48″.0 W	0
	5°4′42″.0 N	44°52′12″.0 W	0
	18°26′36″.0 S	38°30′36″.0 W	0
	23°43′12″.0 S	43°3′6″.0 W	0
	20°48′18″.0 S	39°51′0″.0 W	0
	13°4′6″.0 S	36°36′30″.0 W	0
	4°33′24″.0 S	35°54′12″.0 W	0
	1°52′12″.0 S	35°55′12″.0 W	0
	0°30′24″.0 S	42°59′48″.0 W	0
	22°29′18″.0 N	65°14′12″.0 W	0
	24°49′48″.0 N	66°15′42″.0 W	0
	28°56′12″.0 N	68°7′0″.0 W	0
	34°52′0″.0 N	70°59′48″.0 W	0
	39°8′30″.0 N	73°9′24″.0 W	0
Rhode Island	41°48′46″.0 N	71°41′11″.0 W	0

observations from Mount Wilson, California between March 1987 and November 1991. Along with the time of the observation, he noted the atmospheric temperature, dew point, and pressure at the observation site. He took all the weather data within 5 min

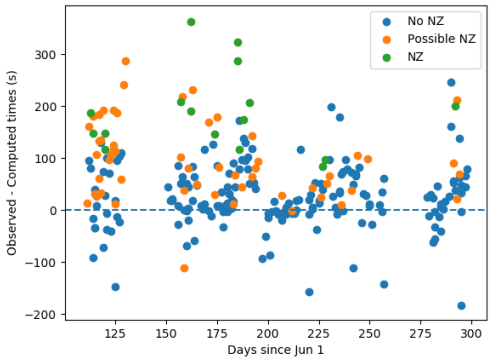
of 1550 PST (2350 UTC) at the National Weather Service station at Mount Wilson. The earliest sunsets recorded were around 1650 PST in December and the latest were around 1810 PST in March, making the time difference between weather observations and sunset observations between 1 and 2.5 hours. He made observations from two nearby locations, for which latitude, longitude, and height were given. The observer made 246 observations with the use of binoculars, and 4 with the unaided eye. He also categorized the observations, noting when the NZ effect was not present, when it was possibly present, and when it was definitely present. Sunsets over the ocean are ideal for this analysis because rise/set time prediction models do not account for an elevated horizon. Due to Mount Wilson's location in relation to the California coast, sunsets over the ocean only happen during the Fall and Winter months, specifically between September 1 and April 1.

Predicted time of sunset was calculated using refraction values from the three different models discussed in Chapter 2: standard refraction, Bennett, and H&S. The H&S model requires a lapse rate value, which was not included in the data, so the standard constant 0.0065 K/m for the troposphere from the *U.S. Standard Atmosphere, 1976* was substituted.[71] Observed and Computed times were then compared, maintaining the NZ observation categories.

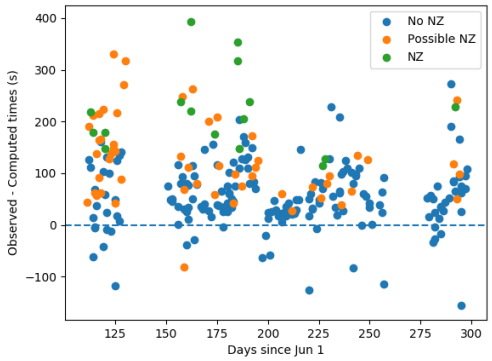
Fig. 4.1(a), shows the use of $34'$ of refraction with dip, when calculating predicted times. The difference between observed and computed times was 88.4 ± 78.4 sec



(a) Observed - Computed (34′)



(b) Observed - Computed (Bennett)



(c) Observed - Computed (HS)

Figure 4.1: Mount Wilson Sunset Data Time differences between observations and computed predictions by day of year using 34′, Bennett-NA, and H&S model for refraction. Dip is included in all scenarios.

indicating that sunset was observed later than the computed time. This means that refraction on the horizon was greater than $34'$. Fig. 4.1(b) shows the use of the Bennett-NA model, which incorporates temperature and pressure, while accounting for dip. The difference between observed and computed times using this model is 51.5 ± 80.0 sec. The Bennett model produced, on average, times within a minute of observed times, however, the standard deviation is just as large as that for $34'$ of refraction. One thing to note with the use of Bennett-NA is that, as discussed in Section 2.1.4 and shown in Fig. 2.2, if the apparent zenith distance of the Sun, with dip, is greater than $91°.6$, the Bennett-NA refraction outputs begin to decrease. The horizon is depressed by about $2°.4$ from Mount Wilson, which means at that height refraction values calculated with Bennett will be systematically low. The H&S algorithm shown in Fig. 4.1(c) yielded larger refraction values than the standard $34'$, but the differences between observed and computed times was 81.1 ± 81.5 sec. Despite the expected systematically low refraction values, Bennett-NA produced the largest refraction values overall, creating smaller differences between observed and computed times by almost a minute.

4.1.1 Population Discussion of Novaya Zemlya Effect

Distribution of the three different populations noted in Fig. 4.1 indicates that the NZ effect did cause a time delay in the observed time of sunset. A two sample

Kolmogorov-Smirnov (KS) test was performed on the populations to verify this. The test determines whether two data sets come from the same distribution and would indicate whether or not observations on days when the NZ effect was thought to be present come from the same population as those on days it was not.[72] As Table 4.2

Table 4.2
KS Statistic Results Mount Wilson

Populations	KS Statistic	p-value
no-NZ, NZ	0.85	2.5e-10
no-NZ+possible-NZ, NZ	0.77	1.06e-8
no-NZ, NZ+possible-NZ	0.48	3.42e-11
no-NZ, possible-NZ	0.42	2.75e-7
possible-NZ, NZ	0.55	0.0006

shows, observations from days when the NZ effect was not present, and those when it was present are two separate populations. The test revealed similar results when adding the possible-NZ days to either population. The KS test was also performed to determine whether the possible-NZ days were a part of either population. The results in Table 4.2 show that possible-NZ days were part of neither population, indicating that some other atmospheric phenomenon might have been occurring. Analysis of the lower-limb measurements on those days may provide more insight. Table 4.3 breaks down the differences between observed and computed times by population. While the standard deviations remain in the minute range, the average difference in the populations essentially doubles each time.

Table 4.3
Mount Wilson Results Differences between Observed and Computed times for Mount Wilson data separated by Population

Population	Refraction Model	Time Difference
no-NZ	$34'$	63.8 ± 61.5 sec
	Bennett	25.8 ± 61.5 sec
	H&S	54.6 ± 61.8 sec
possible-NZ	$34'$	128.9 ± 72.8 sec
	Bennett	95.2 ± 74.7 sec
	H&S + dip	127.0 ± 78.8 sec
NZ	$34'$	222.5 ± 77.2 sec
	Bennett	187.7 ± 76.7 sec
	H&S	217.8 ± 76.7 sec

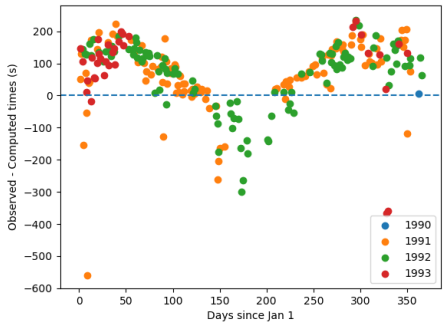
4.2 Edmonton, Alberta, Canada

As part of a Master's thesis in which he calculated the astronomical refraction angle at the horizon, Russell Sampson made a total of 244 sunrise and 125 sunset observations between December 1990 and December 1993, in Edmonton, Alberta, Canada. He published the observation times in an appendix to his thesis.[1] Temperature and pressure data accompanied all of the observations. Relative humidity data were recorded with the observations, but not published. However, Sampson provided a copy of the values in a personal communication. (R. Sampson, private communication, 2018) They are provided in Appendix B.1. Missing values were supplemented with historical values obtained from Weather Underground recorded at a nearby airport.

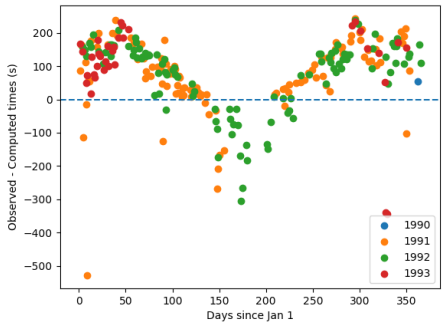
Temperature gradient data were also provided for most of the observations; however,

it was not used for this analysis. The values for vertical temperature gradient were obtained using a radiosonde balloon at heights of 26-695 m, with most heights in the 100-200 m range. They are, on average, an order of magnitude larger than the standard lapse rate of 0.0065 K/m. The H&S algorithm traces the light ray through the entire troposphere, assumes a constant lapse rate throughout, and, as mentioned in Section 2.1.3, cannot handle large lapse rate values.[7] Additionally, the provided values were all observed in the lowest 0.5km of the troposphere, and do not adequately represent the entire troposphere, which spans about 10km. Therefore, the standard lapse rate of 0.0065 K/m was used when calculating refraction with the H&S algorithm.

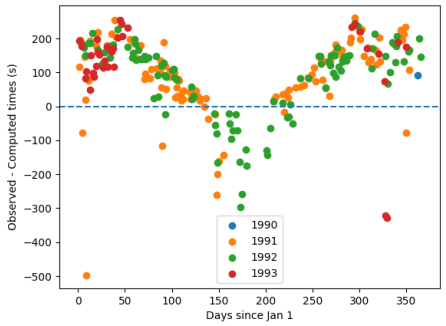
Sampson took observations from two different multi-story buildings in Edmonton. In one building, he used two different floors to make his observations depending on the time of year, providing total elevation and height above the ground floor for one (687 $\pm$ 1m or 22m above ground level), and total elevation for the other (704 $\pm$ 1m).[1] Simple subtraction yields that the height of eye for the second site was 39m above ground level. He provided total elevation (728 $\pm$ 1m) and geographic coordinates for the second building.[1] A topography map of Edmonton provided an elevation of 678m near this building, and a quick calculation determined the height above the ground to be 50m.[73] In all calculations, a flat horizon was assumed.



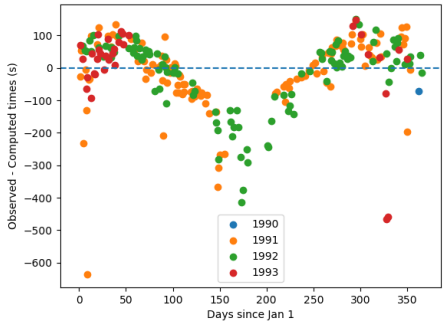
(a) Observed - Computed (34′ dip)



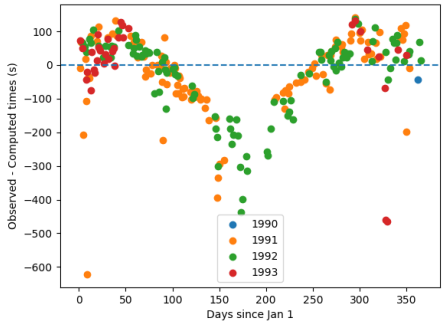
(b) Observed - Computed (Bennett dip)



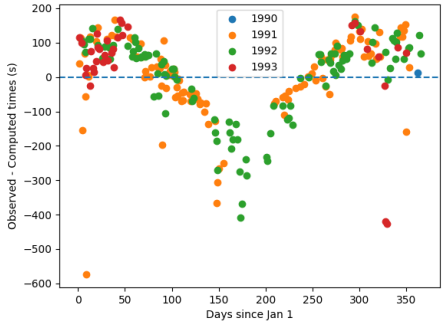
(c) Observed - Computed (H&S dip)



(d) Observed - Computed (34′)

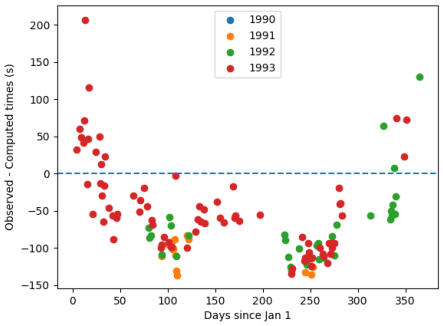


(e) Observed - Computed (Bennett)

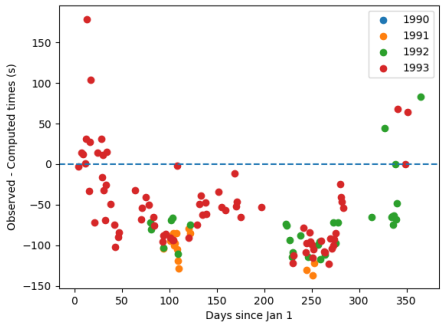


(f) Observed - Computed (H&S)

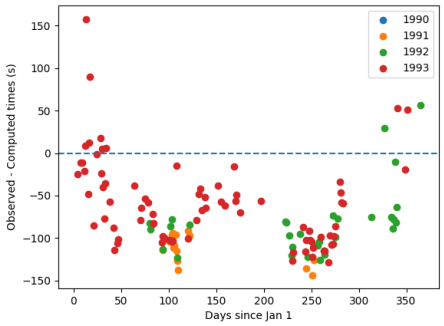
Figure 4.2: Edmonton Sunrise Data Time differences between observations and computed predictions by day of year using 34′, Bennett-NA, and H&S's models for refraction. Dip is included in the first three graphs.



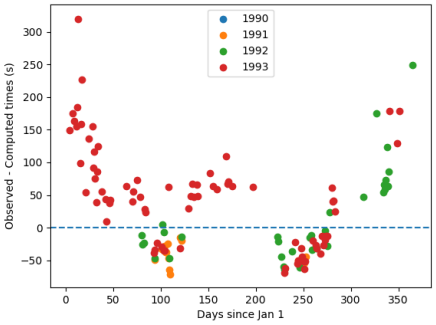
(a) Observed - Computed (34′ dip)



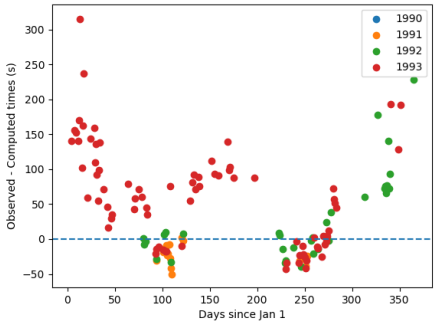
(b) Observed - Computed (Bennett dip)



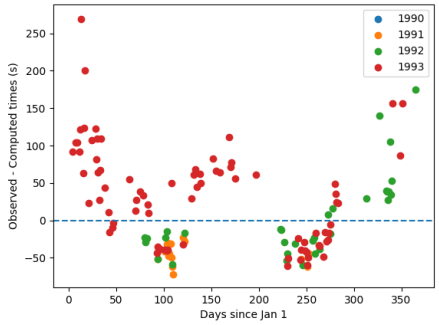
(c) Observed - Computed (H&S dip)



(d) Observed - Computed (34′)



(e) Observed - Computed (Bennett)



(f) Observed - Computed (H&S)

Figure 4.3: Edmonton Sunset Data Time differences between observations and computed predictions by day of year using 34′, Bennett-NA, and H&S's models for refraction. Dip is included for the first three graphs

The differences between observed and computed times were larger at sunrise, on average within 100 seconds, than at sunset, which was on average within 50 seconds. Table 4.4 summarizes the results. When using $34'$ with dip (approx. $13'$), Fig. 4.2(a) and Fig. 4.3(a) show that during the summer months, predictions for sunrises underestimated the amount of refraction causing an average 100 sec delay in sunrise time, while for sunsets, refraction was overestimated, but predictions were only around 50 sec early. Fig. 4.2(a) has maximums in the spring and fall, both around 200 sec, indicating refraction was overestimated by about $45'$ for sunrises at those times of the year. Fig. 4.3(a) has minimums at those same times of the year, but around 100 sec, indicating that refraction was overestimated by about $23'$ then. When using the Bennett-NA refraction model with dip, Fig. 4.2(b) shows that for sunrises, including the daily temperature and pressure information to calculate refraction changed the results in a negligible way. However, for sunsets, Fig. 4.3(b) shows that including the temperature and pressure information shifted the maximum and minimum Observed-Computed times by about 30 sec towards the 0-line. Fig. 4.2(c) and Fig. 4.3(c) show that calculating refraction with the H&S algorithm and including dip give improved predictions over those calculated with Bennett-NA, but only by about 30 seconds. When including dip, sunrise predictions are consistently late, and sunset predictions are consistently early. This means that in both cases, refraction is, in general, less than what is assumed for predictions.

Table 4.4
Edmonton Results Differences between Observed and Computed times for Edmonton data

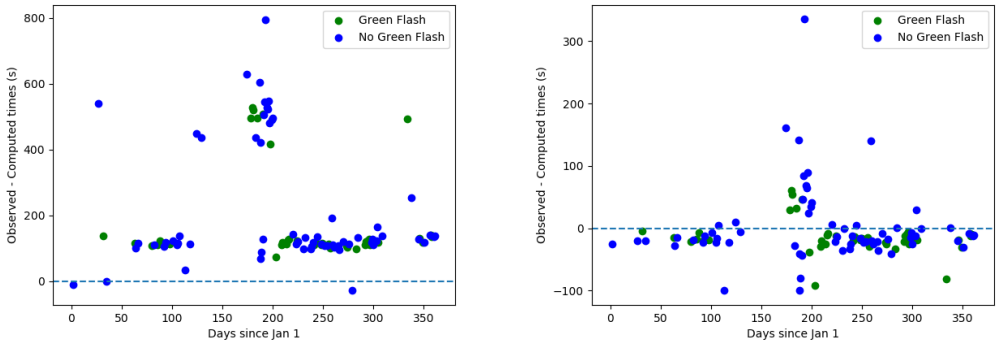
Phenomenon	Refraction Model	Time Difference
Rise	$34'$	-13.1 ± 114.5 sec
	$34'$ + dip	71.8 ± 108.7 sec
	Bennett	-21.6 ± 118.1 sec
	Bennett + dip	80.6 ± 111.0 sec
	H&S	9.7 ± 119.2 sec
	H&S + dip	94.5 ± 113.0 sec
Set	$34'$	28.1 ± 76.7 sec
	$34'$ + dip	-59.9 ± 60.8 sec
	Bennett	43.4 ± 71.0 sec
	Bennett + dip	-62.1 ± 51.7 sec
	H&S	16.3 ± 64.7 sec
	H&S + dip	-71.6 ± 49.2 sec

Both for rise and set times, in excluding the dip portion of refraction, Fig. 4.2(d)- 4.2(f) and Fig. 4.3(d)- 4.3(f) exhibit a shift towards the 0-line when compared to their counterparts, which include dip. This is due to the non-zero altitude of the horizon. Sampson notes that, while the horizon is relatively flat where the observations were made, they did have a non-zero affect on the rise/set observation times. He also calculated the altitude of the horizon at three different points, giving an average elevation of $0°.15$ with respect to the observer. In this work, dip was calculated to be around $0°.19$ assuming a flat horizon. This means that the apparent horizon from the elevated position is at a zenith distance of $90°.04$. This accounts for the shift around the 0-line shown in Fig. 4.2(d)- 4.2(f) and Fig. 4.3(d)- 4.3(f), when dip is excluded.

4.3 Hawai'i and Chile

Sunset observations during which observation of the green flash was noted are available from a paper by Schaefer and Liller (S&L), which discusses astronomical refraction at the horizon. They made 110 sunset observations along with longitude, latitude, and height above sea level. Observations were taken over an ocean horizon between June 1987 and December 1989. Of the observations, eight were taken from eight different locations in Hawai'i, 77 were taken in Viña del Mar, Chile, 4 were taken in La Serena, Chile, and the remaining 21 were taken from Cerro Tololo, Chile. During 45 of the observations, the green flash was noticeable. The time of the sunset was noted using a digital wristwatch with an error of about 2 seconds, and the observations were made with the unaided eye (B. Schaefer, private communication, 2018).[28] Since meteorological data did not accompany the sunset timing observations, and NOAO does not keep records for those locations from that time (Heathcote S., private communication, 2018) the standard $34'$ was used for horizontal refraction. As noted in Chapter 3, Bennett-NA with standard conditions produces $34'$, so using it to calculate predictions times here, with or without dip, would be redundant.

Fig. 4.4 shows the differences between observed and computed times when using $34'$ and $34'$+dip for refraction. Fig. 4.4(a) indicates that, except for three days, the refraction at the time of sunset observations was greater than $34'$, which is to be



(a) Observed - Computed (34′)

(b) Observed - Computed (34′ with dip)

Figure 4.4: Schaefer & Liller Sunset Data Time differences between observations and predictions by day of year using 34′ and 34′+dip for refraction.

expected as many of the data points were taken from a very elevated position (≥200 m). The differences between observed and computed times without accounting for dip is 198.6 ± 170.3 sec. Fig. 4.4(b) indicates that including dip is important to get more accurate sunset times. The difference between observed and computed times is here is -4.6 ± 51.8 sec. However, the large number of negative values indicates that 34′ is an over-estimation of the horizontal refraction at sea level for the locations in question, which verifies Schaefer's results. Similar to the Edmonton data, the trend of larger differences existing in the summer months carries through both plots. A breakdown of the plot by location shows that the positive values in the summer months all come from the Cerro Tololo, which has an elevation of 2215m. Unfortunately, sunset times were only taken there during the summer months, so it is impossible to definitively determine whether observed horizontal refraction was greater than 34′ primarily because of the elevation, or because of the time of year.

According to Fig. 4.4, the green flash seems to have no bearing on the time difference between observed and computed times. To verify that this is the case, observations were separated into days when a green flash was observed, and days when they were not. A two sample KS test was performed on the populations. It revealed a KS statistic of 0.21 and a p-value of 0.16, which verifies the observations come from the same population. This means that atmospheric conditions which cause obsevation of a green flash do not affect the observation of sunset time more than those conditions that do not.

4.4 Smiley

Smiley made a series of hand-written observations of sunrise and sunset times while on expeditions between 1947 and 1952 to study the effects of low atmospheric refraction. His papers are archived at Brown University in Providence, Rhode Island. J. Hilton of the USNO acquired images of these papers and data files. Among his notes are 11 observations over an ocean horizon and 25 observations from a site 18 miles west of Brown University, including 7 sunrises and 29 sunsets.[29] R. Edgar, A. Phlips, and G. Bliss sifted through the images and transcribed the dates, times, and locations of sunrise and sunset observations. They are recorded in Table 4.5 below. Times were noted to be GCT, which Smiley did not define. It is assumed to mean Greenwich Civil Time.(J. Hilton, private communication, 2016) Meteorological data did not

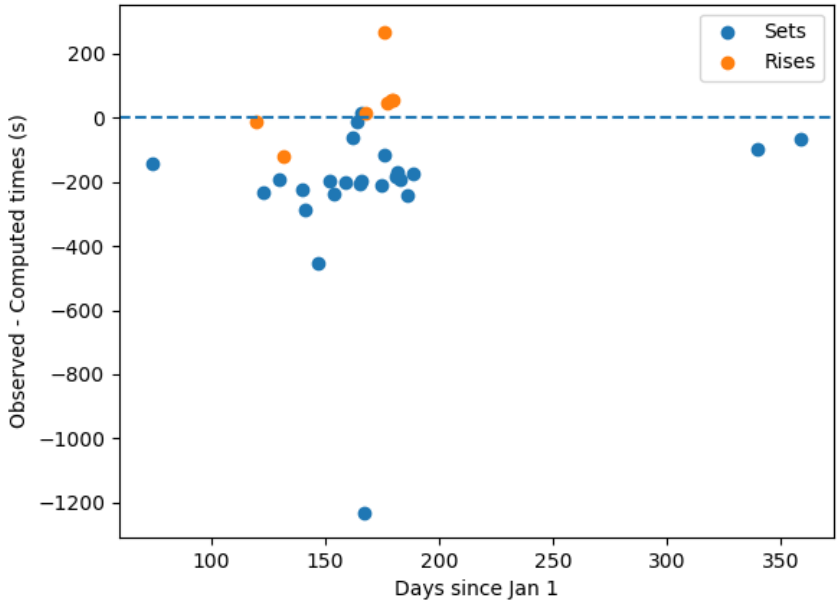


Figure 4.5: Smiley Observations Time differences between observations and predictions by day of year using 34′

accompany these observations, nor did height observations, so the standard 34′ was used.

Fig. 4.5 shows that the difference between observed and computed sunset times was -219.6 ± 226.0, indicating less refraction on the horizon than 34′. The large majority of the sunset observations, taken in RI, were made over land with no indication of what was on the horizon. An elevated horizon could be the the reason for the early sets. The sunrise observations were all taken over the ocean, and the difference between observed and computed times was 43.6 ± 107.7, indicating less than 34′ of refraction here as well. Including dip here would have caused an even grosser over-estimation of refraction on the horizon. Further analysis of Smiley's notes may shed some light

Table 4.5
Smiley Data

Date	Sunrise/Set Time GCT	Latitude	Longitude
03/16/1946	22:51:31	41°48′46″.0 N	71°41′11″.0 W
12/07/1946	21:14:17	41°48′46″.0 N	71°41′11″.0 W
12/26/1946	21:20:35	41°48′46″.0 N	71°41′11″.0 W
05/01/1947	09:35:38	10°32′36″.0 N	58°22′48″.0 W
05/04/1947	21:02:29	5°4′42″.0 N	44°52′12″.0 W
05/11/1947	20:06:13	18°26′36″.0 S	38°30′36″.0 W
05/13/1947	09:17:42	23°43′12″.0 S	43°3′6″.0 W
06/12/1947	20:04:32	20°48′18″.0 S	39°51′0″.0 W
06/14/1947	20:05:57	13°4′6″.0 S	36°36′30″.0 W
06/16/1947	20:20:02	4°33′24″.0 S	35°54′12″.0 W
06/17/1947	20:04:13	1°52′12″.0 S	35°55′12″.0 W
06/18/1947	08:50:13	0°30′24″.0 S	42°59′48″.0 W
06/26/1947	09:42:39	22°29′18″.0 N	65°14′12″.0 W
06/26/1947	23:16:00	24°49′48″.0 N	66°15′42″.0 W
06/27/1947	09:36:28	28°56′12″.0 N	68°7′0″.0 W
06/29/1947	09:33:39	34°52′0″.0 N	70°59′48″.0 W
06/30/1947	09:30:06	39°8′30″.0 N	73°9′24″.0 W
05/21/1949	00:01:46	41°48′46″.0 N	71°41′11″.0 W
06/02/1949	00:12:27	41°48′46″.0 N	71°41′11″.0 W
06/04/1949	00:13:13	41°48′46″.0 N	71°41′11″.0 W
06/15/1949	00:19:48	41°48′46″.0 N	71°41′11″.0 W
06/16/1949	00:20:18	41°48′46″.0 N	71°41′11″.0 W
06/24/1949	00:22:04	41°48′46″.0 N	71°41′11″.0 W
06/25/1949	00:22:04	41°48′46″.0 N	71°41′11″.0 W
06/30/1949	00:22:24	41°48′46″.0 N	71°41′11″.0 W
07/01/1949	00:22:26	41°48′46″.0 N	71°41′11″.0 W
07/02/1949	00:22:27	41°48′46″.0 N	71°41′11″.0 W
07/02/1949	00:21:52	41°48′46″.0 N	71°41′11″.0 W
07/05/1949	00:20:50	41°48′46″.0 N	71°41′11″.0 W
07/08/1949	00:20:17	41°48′46″.0 N	71°41′11″.0 W
05/22/1950	00:01:29	41°48′46″.0 N	71°41′11″.0 W
05/28/1950	00:04:00	41°48′46″.0 N	71°41′11″.0 W
06/09/1950	00:16:49	41°48′46″.0 N	71°41′11″.0 W
07/01/1950	22:28:30	41°48′46″.0 N	71°41′11″.0 W

on conditions that would cause the effect seen here.

4.5 Discussion

While the data sets available are limited, comparing the outputs of sunrise or sunset times computed with varying refraction models yields some interesting results. First, increasing the complexity of the algorithm, and including more meteorological data, does not appear to have a significant effect on decreasing the difference between observed and computed times. Any changes in the models were on the order of 10 sec, but their standard deviations were all on the order of 100 sec nullifying any improvements. As currently implemented, the H&S algorithm consistently computes values around $34'$. With the meteorological data from Mount Wilson, it computed refraction values between $32' - 39'$, and with the data from Edmonton, it returned refraction values between $32' - 44'$, the highest values being at sunrise. This may be in large part due to the problem of using a constant lapse rate in the troposphere, as discussed in Chapter 2, as studies have shown that atmospheric models that use a constant lapse rate produce values around $34'$.[74] However, these findings also raise the question of whether $34'$ should continue to be considered the standard refraction value. The analysis here suggests even with the inclusion of adjustments made for dip, especially at sunset, and regardless of the presence of atmospheric phenomenon, it grossly underestimates the amount of refraction on the horizon. The history of the value and

the conditions under which it was determined need to be better understood if it is going to continue to be considered the "standard" value of atmospheric refraction.

Second, if the Edmonton analysis can be considered representative of a real-world "flat" horizon, even if the observer is at the top of an 8-story building, dip need not be considered. Including dip, even with a slightly elevated horizon, causes a systematic over-estimation of refraction. This causes the sunrise predictions to be early and the sunset predictions to be late.

Third, for refraction values around $34'$, it is clear from the two largest data sets (Edmonton and Mount Wilson) that sunrise and sunset times cannot be predicted to better than 2 min, at least for the cases of a land horizon or a highly elevated observing site. The observations from Cerro Tololo confirm this as well. However, the other observations from S&L all fall well within the minute mark. They were also taken over a water horizon, but from a significantly lower elevation. The other difference between this data set and the other two, is that these observations were taken with the naked-eye as opposed to with binoculars or a welding mask like the Mount Wilson or Edmonton observations. This implies that the eye sees the Sun as having set before the last bit of it has actually left the horizon, perhaps due to saturation, leading the observer to believe that refraction is less than it might be. In this case, $34'$ may be a good estimate when taking naked-eye observations, like those navigators might take when doing positioning. Whether phenomenon times extracted

from videos taken with a camera are susceptible to the same systematic limitations as naked-eye observations should also be explored as it may have a bearing on the effectiveness of the citizen science project outlined in Chapter 6.

Chapter 5

Observing Kepler's Laws with the Sunrise and Sunset

The changing location of sunrise and sunset and number of daylight hours readily demonstrate the tilt of the Earth's spin axis relative to its orbital plane, also known as the obliquity of the ecliptic. However, visualizing the Earth obeying Kepler's laws is a bit more subtle. Observing the time the Sun takes to cross the horizon throughout the year can reveal the first two laws quite well: the elliptical shape of the Earth's orbit with the Sun at one of the foci, and the consequent changes in the Earth's angular speed from perihelion—the point in orbit nearest the Sun—to aphelion—the point in its orbit farthest away. The length of time the full disk of the Sun takes to cross the horizon depends on the Sun's ecliptic latitude and the observer's geographic

latitude.

The exercises outlined here best apply to latitudes that fall between the Arctic and Antarctic Circles (65° N and S respectively). Beyond these latitudes, the Sun spends much of the year either completely above or below the horizon due to the tilt of the Earth. If no sunrise or sunset occurs, the time of crossing would not apply. Large vertical temperature variations near the North and South Poles also affect atmospheric refraction making approximations used here increasingly invalid.[74] The time of horizon crossing should not be confused with the moment of sunrise or sunset. The former refers to the time interval in which the disk of the Sun crosses the apparent horizon. The latter two terms refer to the two moments of the day when the upper limb of the solar disk touches the horizon.[75] This definition applies to both sunrise, at which one would begin recording the crossing time in the morning, and sunset, at which one would stop recording the crossing time in the evening.

5.1 Theory

The length of time the Sun takes to cross the horizon depends on several factors: the most significant are the time of year and the observer's latitude. The angular size of the Sun on the horizon changes through the year because the Earth-Sun distance varies due to the Earth's elliptical orbit. Latitude affects the angle with which with

the Sun crosses the horizon.

The radial distance to the Sun ($R_\odot$) is approximately

$$R_\odot = \frac{a(1-e^2)}{1+e\cos(\omega t)} \tag{5.1}$$

where a is the semi-major axis of the Earth's orbit, e is the eccentricity of Earth's orbit, ω is the angular speed of the Earth's orbital rotation, t is the number of days since perihelion when the Earth-Sun distance is minimized. This equation treats the Earth and Sun as point-objects.[60][76] In the Python program provided in Appendix A.2, t appears as $(t_{day} - t_{perihelion})$ since perihelion currently never occurs on January 1.

The angular size of the Sun ($d_\odot$) on a given day is

$$d_\odot = 2\arctan\left(\frac{D_\odot}{2R_\odot}\right) \tag{5.2}$$

where $D_\odot$ is the diameter of the Sun.

The celestial poles and celestial equator of date correspond with Earth's, so the apparent tilt of the celestial sphere depends on the latitude of the observer on Earth. Consequently, the angle between the horizon and the celestial pole of date is the same as the observer's latitude. The apparent motion of most celestial bodies is approximately parallel to the celestial equator due to the Earth's rotation. Due to

the spherical geometry involved, the angle with which they cross the Earth's horizon depends on their declination, the celestial equivalent of latitude, and apparent rate of motion with respect to the celestial background. For most celestial bodies, the declination changes slowly, and can be considered essentially constant throughout the year. However, the Earth is tilted 23°.45 with respect to its orbit around the Sun, so the Sun's declination varies between ±23°.45, passing through 0° on the equinoxes. The angle, ψ, with which the Sun crosses the horizon with respect to the vertical is

$$\sin(\psi) = \frac{\sin(\phi)}{\cos(\delta)} \tag{5.3}$$

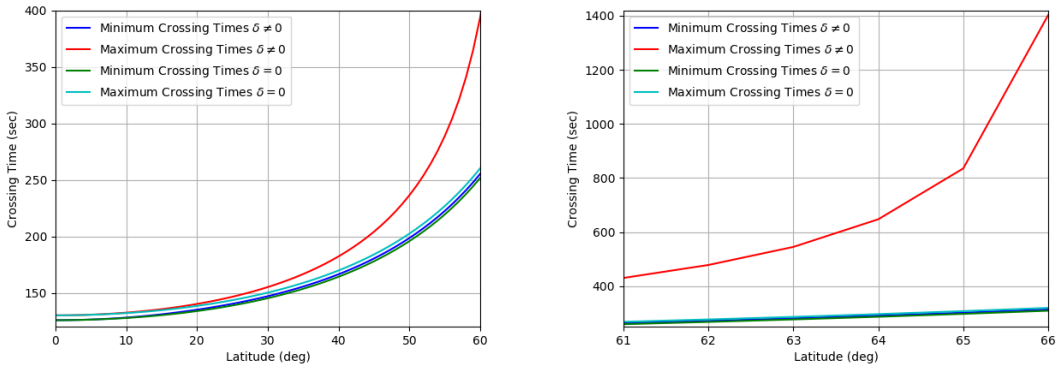
where ϕ is the observer's latitude, and δ is the declination of the Sun.[77]

The time for the Sun to cross the horizon ($T_\odot$) is

$$T_\odot = \frac{d_\odot}{v_\oplus \cos(\psi)} \tag{5.4}$$

where $v_\oplus$ is the angular velocity of Earth. If the declination is not known, then ψ can be approximated as ϕ. This causes errors from 0 sec at the equator up to 1,000 sec at the solstices near the Arctic/Antarctic circles, as can be seen in Fig. 5.1.

The duration of a horizon-crossing is dependent on the angular size of the Sun, the declination of the Sun, and the observer's latitude. If declination is assumed to be zero, the longest and shortest horizon-crossing times occur on perihelion and aphelion,



(a) $0° - 60°$ lat

(b) $61° - 66°$ lat

Figure 5.1: Variation of Crossing Time with Latitude These plots show the minimum and maximum crossing times for the year with respect to latitude. When δ is ignored the days fall on aphelion and perihelion and on the days in question. The crossing time for all other days fall between the extremes represented by the perihelion and aphelion lines.

respectively, and the difference between them can be as much as 8 sec. At perihelion, the radial distance between the Earth and Sun is the shortest and the Sun's angular size is largest, causing a longer horizon-crossing time. At aphelion, the converse is true. The crossing time for all other days fall between the extremes represented by the perihelion and aphelion lines. Fig. 5.1 and 5.2 demonstrate these trends.

If declination is included, the days of the year with the longest and shortest horizon-crossing times vary depending on the latitude. Days with the minimum horizon-crossing times range from aphelion (around July 4), when the angular size of the Sun is smallest, to the autumnal equinox (around September 21), when $\delta = 0$. Days with the maximum horizon-crossing times range from the winter solstice (around December 21), when $\delta = 23.45$ (it's maximum value), to perihelion (around January 4), when

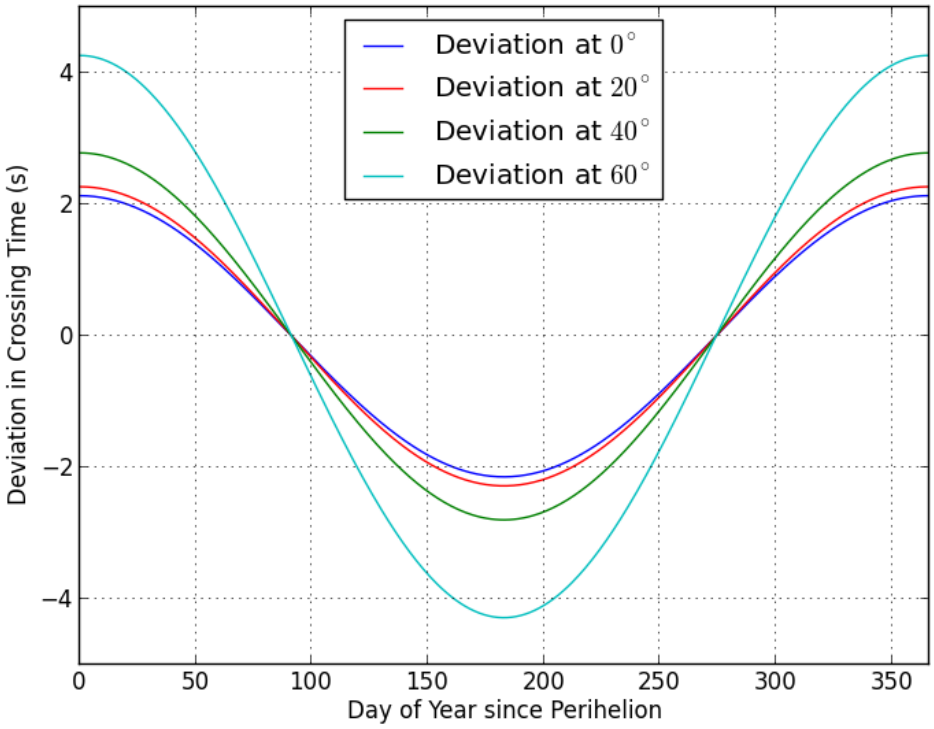
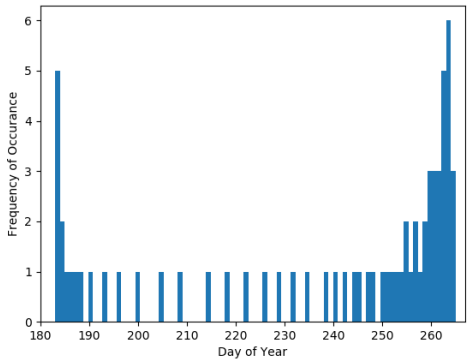


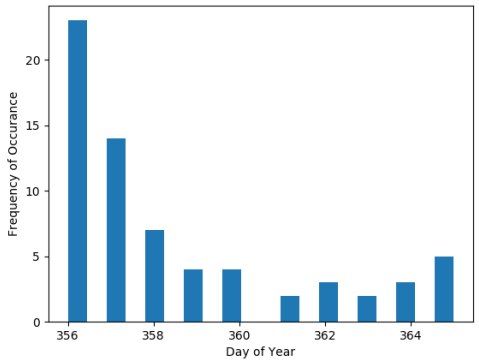
Figure 5.2: Variation of Crossing Time during the Year This plot shows the deviation in crossing times for the latitudes of 0°, 20°, 40°, and 60° when $\delta = 0°$. The deviation was calculated by subtracting the actual crossing time per day from the mean value for the year. The annual mean times were 127.99 sec, 136.21 sec, 167.08 sec, and 255.98 sec, respectively.

the Sun's angular size is the largest. Fig. 5.3 shows the days of the year on which extreme horizon-crossing times occur and the number of latitudes for which that day is an extreme.

Fig. 5.4 demonstrates how including the declination can alter the crossing time throughout the year. The solid line demonstrates how crossing times would behave if the distance to the Sun, and therefore its angular size in the sky, was the only thing affecting the crossing time. The dotted line includes the declination, following



(a) Days with Minimum Crossing Times



(b) Days with Maximum Crossing Times

Figure 5.3: Days of the Year with Maximum and Minimum Horizon-Crossing Times These plots show days on which the minimum and maximum crossing times for the year occur and the number of latitudes for which that day is a minimum or maximum. In Fig. 5.3(b), perihelion is represented as day 366.

a similar trend but with other characteristics as well. The peaks of the dashed line represent the solstices when $\delta = 23°.45$. The troughs where the two lines intersect represent days near the equinoxes when $\delta = 0°$. Interestingly, the crossing times of the spring and fall equinoxes are not equal, which indicates that the distance between the Earth and the Sun is shorter at the vernal equinox than it is at the autumnal equinox. Calculations verify that this is indeed the case: the distance to the Sun on 20 March 2017 is 1.49×10^8 km, while on 22 September 2017, the distance is 1.50×10^8 km.[78]

The above calculations ignore the refraction of sunlight by the Earth's atmosphere. When the solar disk is above the horizon near sunrise or sunset, the gradient of the atmosphere refracts light at the bottom of the disk more significantly than at the top,

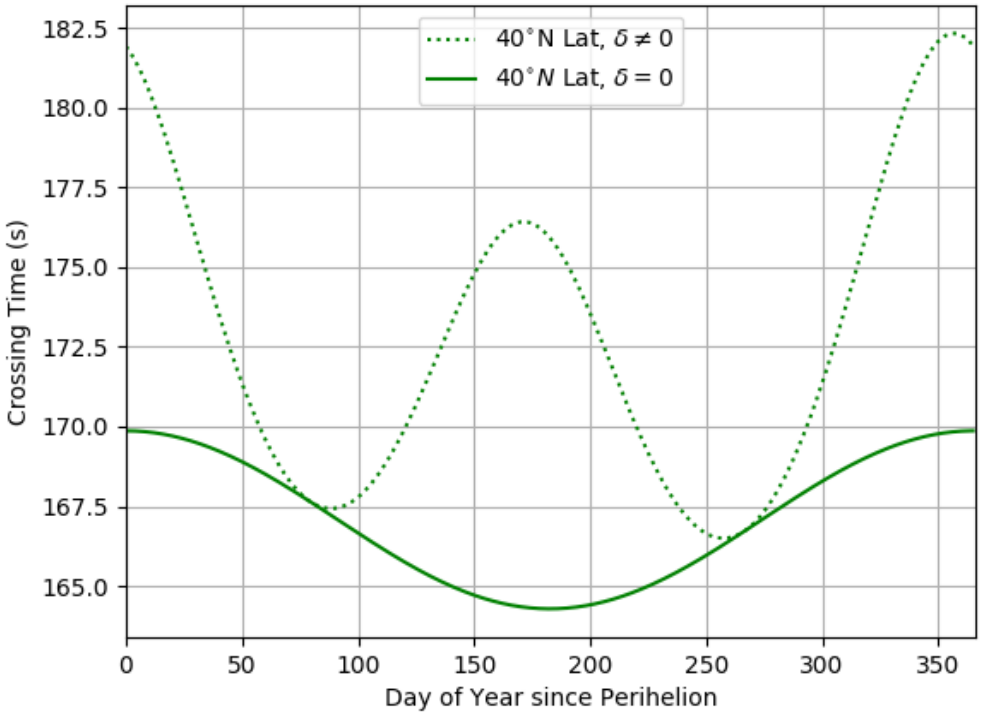


Figure 5.4: Variation of Crossing Time at 40° **Latitude** This plot shows the crossing times throughout the year with and without including declination in the calculations.

causing the Sun to appear "flattened".[79] Because refraction reduces the observed vertical diameter of the Sun, one might expect that the times for horizon crossing would also decrease, but this is not the case.[80] As the Sun rises or sets, the observer witnesses the part of the Sun above the horizon to be flattened. This is most apparent when over half of the Sun is above the horizon. Less than that, and light from the top and "bottom" of the Sun are passing through parts of the atmosphere with similar refraction values, so the flattening becomes less apparent. As the Sun crosses the horizon, say at sunset, the observer witnesses that the lower half of the Sun is more

affected by refraction than the upper half. As the Sun sets, cords of the solar disk that were less affected by refraction now become more so as they get closer to the horizon, until eventually each cord of the solar disk passes through the section of the sky with the greatest refraction. This means that the observer is witnessing the entire solar disk cross the horizon, and not just the flattened image one observes when the disk is above the horizon. As long as meteorological conditions on the horizon remain constant, the magnitude of the refraction value at the horizon is constant. Since the observer is only interested in what is happening at the horizon, the amount of refraction remains unchanged, and the crossing time is unaffected.

5.2 Video Example

This section provides analysis of a YouTube video of a sunset.[81] The software LoggerPro was used to look at the video frame by frame to determine the latitude from the angle of sunset.[82] LoggerPro allows the user to overlay a set of axes and track the motion of objects in the video. In this case, putting the x-axis near the horizon was convenient. After watching the video frame by frame and noting the Sun took 293.5 sec ± 0.5 sec to cross the horizon, points were taken at the upper limb of the Sun at intervals of ~ 30 sec over a period of 10 min to track the motion of the Sun through the sky. A linear fit to the resulting line using a least squares method gave a slope (m) of -0.475 ± 0.003.

Fig. 5.5 shows the points taken, as well as the resulting linear fit. The error reported here indicates the precision of the linear fit and does not account for any systematic error due to changes in atmospheric refraction or random error due to manual plotting. Some basic trigonometry reveals that $\psi = \arctan(1/m)$, also noted on the figure. Without the declination, we could assume that $\psi = \phi$, and conclude that the observer's latitude was $64° \pm 22°$. The uncertainty is calculated using a standard error propagation method, and is large due to the error in the unknown declination. The videographer noted that the video was taken in July 2013, when the average declination was $21° \pm 1°$.[78] Using Eqn. 5.3, we can determine a more accurate latitude of $57° \pm 1°$ The videographer indicated that the video was taken on the beaches of Kaliforien, Germany, on the Baltic Sea, which is at $54°.43 \pm 0°.01$ N latitude. The discrepancy between the calculated and reported latitudes may be due to error in the slope being underestimated, and does not affect the angular size and time calculations that follow since the declination is known.

Together with the time duration of 293.5 sec, solving Eqn. 5.4 estimates the angular size of the Sun to be $0°.525 \pm 0°.003$. The actual angular size of the Sun during that time was $0°.5246 \pm 0°.0003$ [78], and falls within the calculated range. Using this information, we can use Eqn. 5.2 and then 5.1 to determine that the video was taken 167 ± 70 days after perihelion, which occurred on on January 2; the range of calculated dates includes the month of July reported by the videographer.

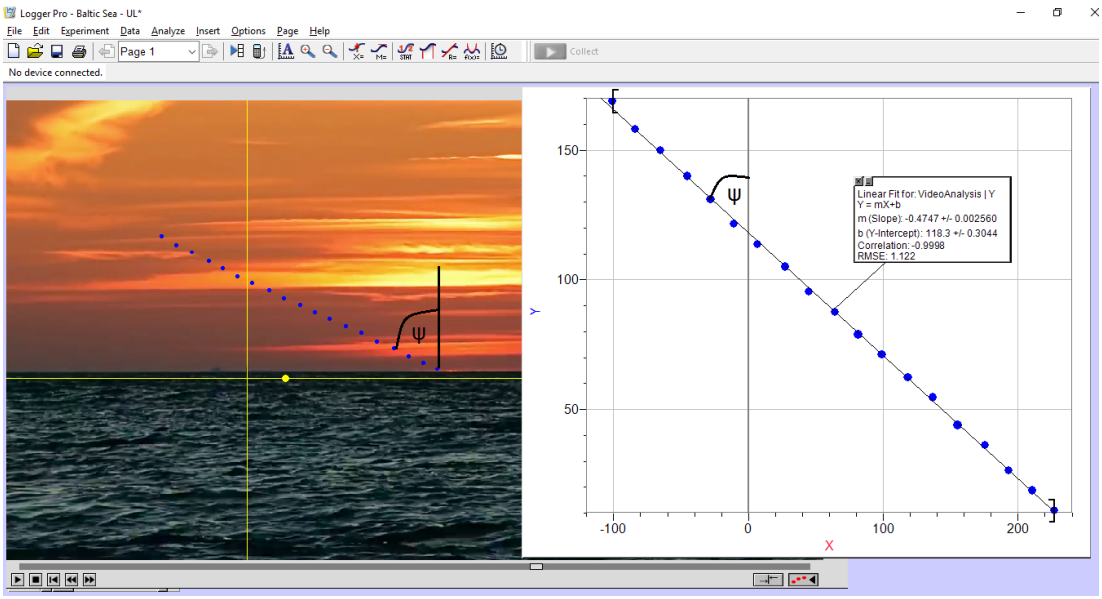


Figure 5.5: LoggerPro Screenshot A screenshot of the analysis performed using the LoggerPro software.

5.3 Discussion

Timing horizon crossings can be incorporated into a number of straightforward learning activities, some of which take only a few minutes. The simplest activity requires a video of sunrise or sunset recorded by a camera steadied by a tripod for consistency in the frames' origin. The observer can hold a protractor up to the video screen and make two measurements: one from a frame showing the Sun above the horizon, and another from a frame showing the Sun at the horizon. One can now estimate the angle of the setting Sun ψ. Assuming δ is near zero, as is true near an equinox, Eqn. 5.3 shows that this angle ψ is a measure of latitude ϕ. If the direction of the sunset is available, one can convert the direction, which is equivalent to the solar azimuth, to

δ by understanding that the Sun sets due west on the equinox when $\delta = 0°$, and south of west on the December solstice, when $\delta = -23°.5$. Many smartphones have compasses that can output this direction. Given the measured angle ψ, Eqn. 5.3 can be used to measure latitude ϕ at any time of the year.

Additionally, one can estimate from the video time display when the Sun has its lower limb, and then upper limb, just at the horizon. To accurately determine the Earth's distance from the Sun, and hence the time of the year, the crossing time should be measured to within a second or better. Section 5.2 showed that a 0.5 sec error in timing can result in a 70 day error in date determination, due to error propagation. This assumes knowledge of declination within 1°. For a more precise exercise, LoggerPro, or a similar software, can be used.[83] One can isolate and superpose video frames to create a composite image that can be used to find the latitude in a few minutes. Error in declination dominates these calculations, as Fig. 5.4 and Section 5.2 demonstrate. Without it, pinpointing time of year can be nearly impossible.

To explore sunsets in even greater detail, students could use the code provided in Appendix A.2 and reproduce the graphs in this paper at various latitudes and times of year to see how different conditions affect the horizon-crossing time. They could then provide a simple analysis of the graphs in the form of answering questions on a worksheet or as a class discussion. Advanced college students could be directed to create their own code to re-produce the results shown here.

Most comprehensively, one could reproduce Fig. 5.2 by recording the crossing times of sunrise or sunset with a stop-watch while keeping track of the date. Alternatively, one could take videos of the sunrises or sunsets over the course of the several months and collect crossing times from the videos.

Timing horizon crossings demonstrates a number of concepts including Kepler's first law concerning the elliptical shape of the Earth's orbit, the latitude of the observer, and the time of the year. It is accessible to people from a wide range of scientific backgrounds, which allows for educational applications on a number of levels, from high-school to college.

Chapter 6

Sunrise & Sunset Observer Citizen Science Project

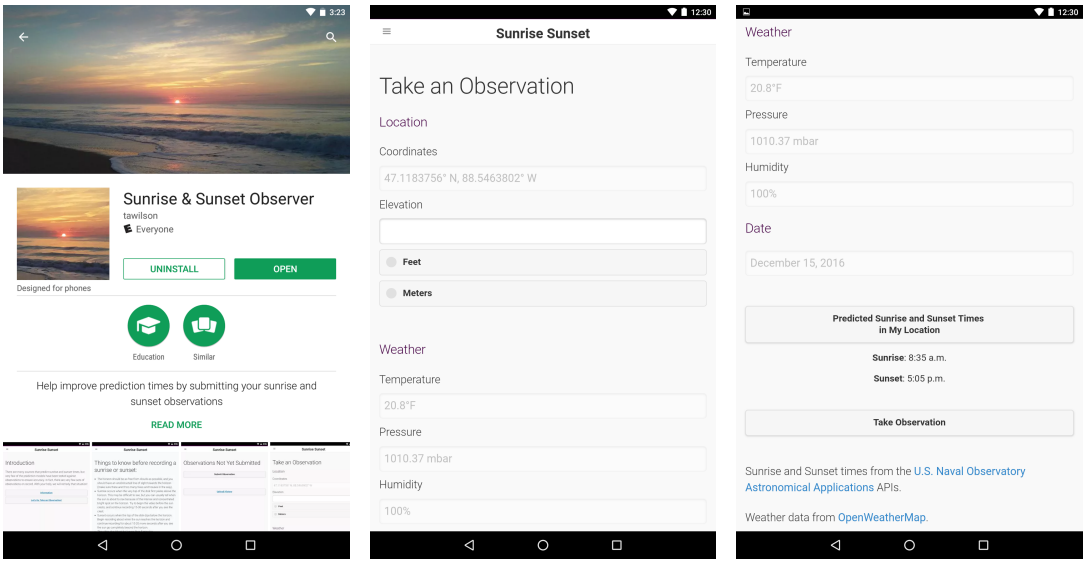
Chapter 4 analyzed the extent of the sunrise and sunset observations currently available. While observations from Mount Wilson and Edmonton provided a sizable data set, they only represent two locations, and the Mount Wilson data is limited to sunset observations. In fact, besides the 7 sunrises provided by the Smiley observations, Edmonton is the only place from which sunrise observations were taken. While these observations can help us begin understanding atmospheric refraction near the horizon, the study would greatly benefit from a more extensive data set. Smartphones have an immense potential for data-collection devices due to their built-in sensors and camera, and their ability to transmit that information over a wireless internet

connection. They allow the user to collect the data they need without requiring them to operate equipment from the confines of an office, and also allow for a wider user base, as almost everyone has one. They are ideal tools for citizen science projects that need information from a wide area over an extended amount of time. The Sunrise & Sunset Observer (SSO) citizen science project was developed to use smartphones' potential to collect a larger data set of sunrise and sunset observations.

6.1 Smartphone Application

To collect a larger set of sunrise and sunset observations, the Sunrise & Sunset Observer (SSO) mobile application for Android was published on the Google Play Store in 2017. The app was written in JavaScript, and once data is collected, hosts it in a SQLite database until it is uploaded to Structured Query Language (SQL) database on the server. It gives the user a predicted sunrise and sunset time using the USNO's "Complete Sun and Moon Data for One Day" API for planning purposes based on their location.[84][52]

At the time of the observation, the SSO app uses the smartphone's GPS to record latitude and longitude coordinates, and records temperature, pressure, and humidity data using the OpenWeatherMap API.[85] The observer is asked to include their height above the horizon so that the observation's analysis can account for dip. The



(a) Google Play Store Entry (b) Sample Page 1 (c) Sample Page 2

Figure 6.1: Sunrise & Sunset Observer App Example pages of what an observer might expect when using the app

SSO app then links to the smartphone's video camera and the observer can take record the sunrise or sunset. After the observation is complete, the observer can submit the observation any time they choose. This allows them to take observations where data-coverage is poor, and then submit them when they have a wireless connection. Once submitted, the time-stamped videos can be analyzed to collect the rise or set time.

6.2 Preliminary Analysis

To verify that the video analysis would yield acceptable observation results, video observations from water horizons were taken by the author, J. Bartlett, and J. Niehaus.

A water horizon, clear of clouds at the moment of sunrise or sunset, is a rarer phenomenon than anticipated at the chosen locations and season. Because of this, it was determined that the observations recorded in Table 6.1 with corresponding weather data would be sufficient for a preliminary analysis.

Computed times were obtained using URSA with the H&S refraction model as meteorological data was taken along with the observations. The observed time was noted by the observer at the time of the observation to help validate the viability of citizen science project. For 3 of the observations, time to nearest second was not available, as the observer used a digital clock which only reported to the nearest minute. In these cases, time to the nearest minute was used.

To determine the time of sunrise or sunset from the videos, they were analyzed frame by frame around the time of crossing. The start time, and the date on which it occured, is embedded in the video's meta-data. When a crossing time is determined, the time from the beginning of the video can then be added to the start time to establish the time of sunrise or sunset. If the horizon was free from clouds, pixelating of the horizon was minimal, and sunrise or sunset times were noted with a confidence of $\pm$2sec. Clouds reflect light from the sun, heavily pixelating the image on the horizon, and made detecting the exact moment of sunrise or sunset from the video difficult. However, working under the assumption that the Sun is only reflected in the water while it is above the horizon, one can use the sun's reflection to determine

the moment of sunrise or sunset. On these days, the confidence in determining when sunset happened was more like ±10 sec. For land observations, this problem will need a different solution.

Of the observations in Table 6.1, 3 were within a minute of the predicted time, and 2 were within 2 minutes. They were also all within a minute of their observed times. Compared to the historic data analyzed in Chapter 4, these observations are much closer to the predicted time by almost a minute. However, they were taken during the summer months, when the difference between observed and computed times was closer to the minute mark in the historic data sets. More observations will be needed to determine whether seasonal variation affects this analysis.

6.3 Conclusion & Future Work

Results from analyzing the videos collected using the SSO citizen science project indicate that it is a viable source for data collection. Times collected using the videos are with a minute of the observation time. This means that, rather than relying solely on direct observations by a single observer, videos with appropriate accompanying information could be used to study the affects of atmospheric refraction on the horizon. Thanks to technological advances, especially in the smartphone industry, taking

videos has never been easier, poviding the potential for a large number of observations from all around the world, and from all times of the year. The development of an iPhone app would increase the project's marketability and would then be able to reach almost the entire smartphone population. Fixed sites with a camera and a Raspberry Pi set up could also be created to submit observations to the database.

With a larger data set, questions such as the seasonal effect of atmospheric refraction on the horizon could then be studied in more depth. The scope of the study could also increase to include land horizons, allowing for comparisons between refraction on land and sea horizons undergoing similar meteorological conditions. Refraction values calculated using these times could then put better constraints on a value for refraction at the horizon, whether by time of year, or by location, or both.

Table 6.1
Video Analysis Results

Type	Location	Date	Video Time	Computed Time	Observed Time
Sunset	50°12′.5N 43°46′.4W	2016-05-21	22:45:06 UT	22:44:09 UT	not available
Sunset	43°8′.8N 58°47′.7W	2016-06-02	23:29:11 UT	23:29:16 UT	23:29:33 UT
Sunrise	38°20′.2N 75°5′.1W	2016-06-18	09:36:05 UT	09:36:33 UT	09:36:00 UT
Sunset	47°16′.6N 88°31′.6W	2017-08-13	01:11:46 UT	01:10:26 UT	01:12:00 UT
Sunset	45°12′.1N 123°57′.7W	2017-08-25	03:06:24 UT	03:05:05 UT	03:06:00 UT

Chapter 7

Conclusion

7.1 Conclusion

The work presented here evaluated the effectiveness current atmospheric refraction models near the horizon by taking a practical approach of using the delay of sunrise and sunset times. Chapter 1 established the need for such a study and presented several atmospheric phenomena due to refraction that current refraction models do not account for. Chapter 2 then discussed the history of the various refraction models of varying complexity, from the constant value of $34'$, which is considered the standard value for refraction at the horizon and is used in all publicly available rise/set calculators, to models that incorporate meteorological conditions. The latter still have

their limitations due to assumptions about the behavior of the troposphere, and it was noted that they generally produce values near $34'$ at the horizon because of this.

To conduct the analysis, a rise/set calculator in the modern programming language of Python 2.7 was developed. The calculator has the capacity to incorporate location-specific refraction values, rather than the standard value, was established as comparable to current standard calculators, and superior to those with less rigorous computations. Chapter 3 discusses the algorithms behind the calculator.

Several sets of direct observations, with various levels of meteorological data, were amassed from the literature, and then compared with computed times using increasingly complex refraction models. Chapter 4 revealed that, regardless of the complexity of the refraction model, predictions of sunrise and sunset times cannot be known within 2 minutes. While there is still much to be understood about the lower atmosphere, it is certain that knowledge of the observer's height above the ground plays a key role in establishing a more accurate rise/set time. As was seen in the S&L data, just including dip yielded much better results. The analysis also yielded that the summer months produce the largest difference in time between observed and computed values, which makes sense as the view of the horizon is affected by the large temperature difference. The S&L data also revealed that the Green Flash does not affect the difference between observed and computed times. However, analysis of the Mount Wilson data revealed that the NZ effect does affect sunset times by at

least as much as 5 minutes. Chapter 6 discussed a method to increase the size of the data-set of observations. The beginning of a citizen science project, Sunrise & Sunset Observer, was established and preliminary results were successfully gathered.

7.2 Future Work & Recommendations

The analysis presented in this work far from exhausts the information that can be gleaned from the data sets discussed here. Meteorological data, along with more sunrise and sunset observations were discovered in the Smiley data while the notes were being transcribed, after the work presented here was complete. Re-analysis of the Smiley data-set using the methods performed on the Mount Wilson and Edmonton data, as well as including estimates for dip, may shed some light on the reason behind the large variance seen in the sunset data, and the low refraction values. The Mount Wilson data set includes 109 observations of contact between the lower limb of the Sun and the horizon, and 217 observations of contact between the center of the Sun and the horizon. Chapter 4 mentions that the lapse rate may be a critical part of understanding refraction near the horizon. A study of how the solar diameter is affected by refraction could reveal something about the behavior of the lapse rate near the surface of the Earth.

URSA could be modified in a number of ways. The upgrades mentioned in Chapter 3 such as using a calculated value for the solar diameter rather than the current constant, might increase URSA's precision. Additionally, the refraction value inputs could be modified to take into account a non-level land horizon. This may increase the accuracy of prediction values for such places as Edmonton, Canada. URSA's use could also be expanded to calculate the rise/set times of other celestial bodies, including the Moon.

The currently available data sets only represent a small portion of locations on the Earth. Further dissemination of the Sunrise & Sunset Observer app would increase the data sets and lead to a better understanding of how to predict what will happen on the horizon and when it will happen. This could lead to the use of different values for refraction depending on the time of year or latitude of the observer.

Much about the refraction on the horizon is yet to be discovered, but the work presented here suggests a few recommendations for now. First, sunrise and sunset calculators should not advertise the accuracy of their predictions to better than two minutes due to refraction. The Edmonton data in Section 4.2 revealed a seasonally-affected variance; the Mount Wilson data in Section 4.1 showed that the NZ-effect, and other currently unknown refraction phenomena, affected observed sunset times. Second, when it comes to incorporating refraction models, if weather data is available, using Bennett-NA may increase the accuracy of the refraction values. However, use of

the H&S model may not be worth the computational complexity. This recommendation could change if the assumption of a constant lapse rate in the troposphere were modified. Third, as the data from Chile and Hawai'i in Section 4.3 demonstrated, existing calculators and other services (such as those used for celestial navigation) should include dip, especially if the observations are made over a water horizon. Incorporating these recommendations can mitigate some of the inaccuracy in refraction values revealed in this work. Newton laid the foundation for much of Physics, but it is time to stand on his shoulders and modify his $34'$ for refraction. Just remember to account for that extra bit of dip.

References

[1] Sampson, R. D. Atmospheric Refraction and its Effects on Sunrise and Sunset Master's thesis, University of Alberta Libraries, Edmonton, Alberta Canada, **1994**.

[2] Brocken Inaglory. *Mock Mirage and Green Flash in San Francisco;* Wikimedia: The Internet, 2002.

[3] Brocken Inaglory. *Sunset Inferior Mirage: Omega Sun;* Wikimedia: The Internet, 2007.

[4] Modiddy. *Fata Morgana as seen off the coast of Manhattan Beach;* Wikimedia: The Internet, 2014.

[5] Brocken Inaglory. *Superior mirage and Green flash in San Francisco;* Wikimedia: The Internet, 2008.

[6] Green, R. M. *Spherical astronomy;* Cambridge University Press, 1985.

[7] Hohenkerk, C.; Sinclair, A. The Computation of Angular Atmospheric Refraction at Large Zenith Angles NAO TN 63, HMNAO, Taunton, UK: PPARC, **1985**.

[8] Young, A. T.; Kattawar, G. W. *Applied Optics* **1998**, *37*(18), 3785–3792.

[9] Sobel, D. *Longitude;* HarperCollins Publishers, 2011.

[10] Folkner, W. M.; Williams, J. G.; Boggs, D. H.; Park, R. S.; Kuchynka, P. *Interplanetary Network Progress Report* **2014**, *196*, 1–81.

[11] Haberreiter, M.; Schmutz, W.; Kosovichev, A. G. *The Astrophysical Journal Letters* **2008**, *675*, L53.

[12] U. S. Naval Observatory.; Royal Greenwich Observatory. *The Astronomical almanac for the year 2017;* U.S. Nautical Almanac Office in the United States (USNO) ; Her Majesty's Nautical Almanac Office (HMNAO) in the United Kingdom: Washington, D.C.; London, 2016.

[13] Maltin, T. *Titanic: A Very Deceiving Night;* BookBaby, 2012.

[14] Lehn, W. H.; German, B. A. *Applied Optics* **1981**, *20*(12), 2043–2047.

[15] Verne, J. *The Green Ray;* Wildside Press, 2002.

[16] Young, A. T. *Journal of the Optical Society of America A* **2000**, *17*, 2129–2139.

[17] Politi, M. *Cronica della nobil' e fedelissima città di Reggio: In Messina Appresso Pietro Brea. M.DCXVII.;* 1617.

[18] Fraser, A. B.; Mach, W. H. *Scientific American* **1976**, *234*(1), 102–111.

[19] Mooney, M. *Ocean Navigator* **1992**, *39*, 89–96.

[20] Van der Werf, Siebren, Y. *Arctic* **1998**, *51*(2), 142–154.

[21] Shackleton, E. H. *South : the story of Shackleton's last expedition, 1914-1917 / by Sir Ernest Shackleton, C.V.O;* W. Heinemann London, 1919.

[22] van der Werf, S. Y.; Können, G. P.; Lehn, W. H. *Applied Optics* **2003**, *42*, 367–378.

[23] Hilton, J. Further Analysis of Mount Wilson Sunset Observations Technical Report Unpublished, USNO/AA, Washington, DC, **2016**.

[24] Taylor, M. S.; McGraw, J. T.; Zimmer, P. C.; Pier, J. R. *The Astronomical Journal* **2013**, *145*(3), 82.

[25] Hohenkerk, C.; Sinclair, A. The Computation of Angular Atmospheric Refraction at Large Zenith Angles NAO TN 63 updated, in preparation, HMNAO, Taunton, UK: PPARC, **2008**.

[26] Bennett, G. G. *Journal of Navigation* **1982**, *35*, 255–259.

[27] Rarogiewicz, L.; Hilton, J.; Kaplan, G. Observations of Sunset from Mount Wilson Observatory 1987-1991 USNO/AA In Preparation, Astronomical Applications Department, USNO, Washington, DC, **2017**.

[28] Schaefer, B. E.; Liller, W. *Publications of the Astronomical Society of the Pacific* **1990**, *102*, 796–805.

[29] Smiley, C. *A Study of Atmospheric Refraction using Solar Disk Flattening;* Unpublished Raw Data, 1950.

[30] DoD, G. N. Global Positioning System Standard Positioning Service Performance Standard Technical report, Department of Defense, Washinton, DC, **2008**.

[31] Smart, W. M.; Green, R. M. *Textbook on Spherical Astronomy;* Cambridge University Press, 1977.

[32] Urban, S.; Seidelmann, P. *Explanatory Supplement to the Astronomical Almanac;* University Science Books, 2013.

[33] Brunnow, F. E. *Spherical astronomy.;* D. Van Nostrand, 1865.

[34] Newcomb, S. *A compendium of spherical astronomy with its application to the determination and reduction of positions of the fixed stars;* New York: The Macmillan Co.; London: Macmillan & Co., Ltd., 1906.

[35] Halley, E. In *Philosophical Transactions;* Printers to the Royal Society, 1733; Vol. VI of *1;* pages 158–160.

[36] Garfinkel, B. *Astronomical Journal* **1944**, *50*, 169–179.

[37] Garfinkel, B. *Astronomical Journal* **1967**, *72*, 235–254.

[38] Auer, L. H.; Standish, E. M. *The Astronomical Journal* **2000**, *119*(5), 2472.

[39] Sinclair, A. The Effect of Atmospheric Refraction on Laser Ranging Data HMNAO TN 59, HMNAO, East Sussex, UK: Royal Greenwich Obs, **1982**.

[40] Hohenkerk, C. In *Explanatory Supplement to the Astronomical Almanac;* University Science Books, 2013; pages 249–304.

[41] Hohenkerk, C. Refraction: Bennetts Refraction Formulae, updated for use in The Nautical Almanac 2005 Onwards Internal memo, unpublished, HMNAO, Chilton, UK: Rutherford Appleton Laboratory, **2003**.

[42] E., W.; Clemence, G. In *Spherical Astronomy;* Academic Press, 1966; page 215.

[43] Phlips, A.; Wilson, T.; Chizek Frouard, M.; Bartlett, J. L. In *American Astronomical Society Meeting Abstracts #231*, Vol. 231 of *American Astronomical Society Meeting Abstracts*, page 150.03, 2018.

[44] Bangert, J.; Panossian, S.; Kaplan, G. The U.S. Naval Observatory Standard Lunar-Solar Model Technical Report 02, USNO, Washington, DC, **1993**.

[45] Standish, E. *Astronomy & Astrophysics* **1982**, *114*, 297–302.

[46] Standish, E. *Astronomy & Astrophysics* **1990**, *233*, 252–271.

[47] Kaplan, G. H. *U.S. Naval Observatory Circulars* **2005**, *179*.

[48] Standish, E. *JPL IOM* **1998**, *312.F-98-048*.

[49] United States Naval Observatory Astronomical Applications Department Sunrise/Sunset/Twilight/ Moonrise/Moonset/Phase Public Affairs Program (Version 7 - for Web). Kaplan, G. Fortran 77 source code, Washington, DC: USNO, **1998**.

[50] USNO Public Affairs Program, ver. 7.1. Kaplan, G.; Fredericks, A.; Bartlett, J. L. Washington, DC: USNO, **2013**.

[51] Solar-Lunar Core (SLAC) Software User's Guide, Version 2.0. Bangert, J.; Puatua, W.; Hilton, J.; Lesniak, M.; Bartlett, J.; Harris, W.; Fredericks, A.; USNO, Washinton, DC, **2014**.

[52] Lesniak, M. V.; Pozniak, D.; Punnoose, T. In *American Astronomical Society Meeting Abstracts*, Vol. 225 of *American Astronomical Society Meeting Abstracts*, page 243.09, 2015.

[53] Bretagnon, P.; Simon, J. *Planetary Programs and Tables from -4000 to +2800;* Willmann-Bell: Richmond, VA, 1986.

[54] Bangert, J. Updated Ephemerides of the Sun and Moon for the Solar-Lunar Almanac Core (SLAC) Technical Report 03, USNO/AA, Washington, DC, **2014**.

[55] Group, S. S. D. *HORIZONS;* Jet Propulsion Laboratory, NASA: The Internet, 2018.

[56] Horizons (Version 3.75). NASA JPL, **2013**.

[57] Saemundsson, T. *Sky & Telescope* **1986**, *72*, 70.

[58] Astropy Collaboration.; Robitaille, T. P.; et al. *Astronomy & Astrophysics* **2013**, *558*, A33.

[59] Yallop, B.; Hohenkerk, C.; Bell, S. In *Explanatory Supplement to the Astronomical Almanac;* University Science Books, 2013; pages 505–528.

[60] Meeus, J. *Astronomical Algorithms;* Willmann-Bell, Inc., 2nd ed., 1998.

[61] Schlyter, P. *How to compute planetary positions;* stjarnhimlen.se, 1989.

[62] Barron, E. G.; Kaplan, G. H.; Bangert, J.; Bartlett, J. L.; Puatua, W.; Harris, W.; Barrett, P. In *American Astronomical Society Meeting Abstracts #217*, Vol. 43 of *Bulletin of the American Astronomical Society*, page 344.14, 2011.

[63] Naval Oceanography Portal. *Long-term Delta T;* USNO, 2018.

[64] McCarthy, D. D.; Babcock, A. K. *Physics of the Earth and Planetary Interiors* **1986**, *44*(3), 281 – 292.

[65] USNO. *Multiyear interactive computer almanac, 1800-2050: version 2.2.2;* Willmann-Bell, 2012.

[66] sunrise-sunset.org. *Sunrise Sunset API;* Sunrise-Sunset: The Internet, 2017.

[67] Stein, J.; Holmgren, W.; Forbess, J.; Hansen, C. In *43rd Photovoltaic Specialists Conference*, 2016.

[68] USNO. *The Nautical Almanac for the Year 2014;* Government Printing Office, 2013.

[69] Hohenkerk, C. The Nautical Almanac Altitude Correction Tables A4: Additional corrections for Non-standard Temperature and Pressure Internal memo, unpublished, HMNAO, Chilton, UK: Rutherford Appleton Laboratory, **2001**.

[70] Pool, B. *The Rain Man: Solitary Life Suits Weather Observer Atop Mt. Wilson;* Los Angeles Times: The Internet, 1995.

[71] Dubin, M.; Hull, A.; Champion, K. US Standard Atmosphere, 1976 Technical report, NOAA, NASA, US Airforce, Washington, DC, **1976**.

[72] In *The Concise Encyclopedia of Statistics;* Springer New York: New York, NY, 2008; pages 283–287.

[73] Topographic-map.com. *Topographic Map of Edmonton;* Tessanet: The Internet, 2017.

[74] Young, A. *Astronomical Journal* **2004**, *127*, 3622–3637.

[75] Sampson, R. D.; Lozowski, E. P.; Peterson, A. E.; Hube, D. P. *PASP* **2003**, *115*, 1256–1261.

[76] Danby, J. M. A. In *Fundamentals of celestial mechanics;* Willmann-Bell, 2003; page 125212.

[77] Duffett-Smith, P. *Practical Astronomy with your Calculator;* Cambridge University Press, 3rd ed., 1988.

[78] U. S. Naval Observatory. MICA (Multiyear Interactive Computer Almanac) 1800-2050 Technical report, U. S. Naval Observatory, **2005**.

[79] Young, A. T. *The Observatory* **2006**, *126*, 82–115.

[80] Fisher, W. J. *Popular Astronomy* **1922**, *30*, 213.

[81] BarWal1963. *Sonnenuntergang an der Ostsee - Sunset on the Baltic Sea;* YouTube: The Internet, 2013.

[82] vernier.com. *Logger Pro;* Vernier Software & Technology: The Internet, 2016.

[83] Tracker Video Analysis and Modeling Tool for Physics Education. `http://physlets.org/tracker/`.

[84] Wilson, T.; Chizek Frouard, M.; Bartlett, J. L. In *American Astronomical Society Meeting Abstracts #229*, Vol. 229 of *American Astronomical Society Meeting Abstracts*, page 334.12, 2017.

[85] openweathermap.org. *Current Weather API;* OpenWeatherMap: The Internet, 2018.

[86] wunderground.com. *Weather History & Data Archive;* Weather Underground: The Internet, 2018.

Appendix A

Sample Code

Below are the sample codes of programs used for calculations in this thesis.

A.1 URSA

This program in Python 2.7 computes the time of the beginning of civil twilight, sunrise, transit, sunset, and end of civil twilight for a given date. It uses several calls to the USNO software NOVAS discussed in Chapter (3).

```
import math
import numpy as np
from novas.compat import Object
from novas.compat import app_planet
```

```
from novas.compat import sidereal_time
from novas.compat import cal_date
from novas.compat import julian_date
#The C version of app_planet or sidereal_time needs the ↩
   ephemeris file opened
from novas.compat.eph_manager import ephem_open
#Path to ephemeris file hard-coded here. Eventually this↩
    should be set as a
#global variable
_ = ephem_open('JPLEPH')

#Defining the Object "Sun" to input to NOVAS function "↩
   app_planet"
Sun = Object()
Sun.type = 0
Sun.number = 10
Sun.name = "Sun"

#sidereal rate in degrees/day
sid_rate = 1.0027379 * 360.0
#Rate of change of solar right ascension in degrees/day
ra_dot = 0.985644
#Approximate rate of change of Sun's hour angle (ha) in ↩
   degrees/day
ha_dot = sid_rate - ra_dot

def DeclinationChange(ra):
    '''
    Calculates the approximate rate of change of solar ↩
       declination in degrees
    per day using the right ascension (ra)
    '''
    return 0.403 * math.cos(math.radians(ra))
```

```
def JD_noon(jul_day, time_zone):
    '''
    Calculates Julian Date of local clock noon given a ↩
       Julian Date of midnight
    '''
    return jul_day - time_zone/24.0 + 0.5

def Fractional_Year(jul_day):
    '''
    Input: Julian Date of phenomenon
    Output: Fraction of year in relation to the winter ↩
       solstice 1999

    Creates a zeroth-order approximation for time of ↩
       phenomenon in relation to
    years from winter solstice 1999 (consistency with ↩
       USNO FORTRAN
    PAP subroutine). It serves as a way to calculate the↩
        fraction of the year
    (frac_year)
    '''
    years = (jul_day - 2451535.5)/365.25
    frac_year = math.fmod(years, 1.0)
    if frac_year < 0.0:
        frac_year = frac_year + 1.0
    return frac_year


def sunrise_sunset(year, month, day, time_zone = 0.0, ↩
   lon, lat):
    '''
    Python 2.7
    Rise/Set Calculator
```

```
This routine computes the times of sunrise and ↩
   sunset for one day
Input Arguments:
year: integer, year
month: integer, month
day: integer, day
time_zone: float, hours from UT (East +, West -), ↩
   Default = 0
lon: float, Longitude in degrees (East +, West -)
lat: float, Latitude in degrees (North +, South -)

Output Arguments:
Julian Dates of beginning of civil twilight, sunrise↩
   , transit, sunset,
                end of civil twilight
'''

#Convert Gregorian Date inputs to Julian Dates with ↩
   call to NOVAS
jul_day = julian_date(year, month, day, 0.0)
#Find the Julian Date of local clock noon
jd_noon = JD_noon(jul_day, time_zone)

#Find the correct Delta T value for the Julian Date ↩
   in question
find_delta_t = np.loadtxt('DeltaT_1970-2026.txt', ↩
   usecols=[1, 2])
delta_t = np.interp(jd_noon, find_delta_t[:, 0],
                    find_delta_t[:, 1])

#Computes time (jd_phen) and altitude of transit (↩
   alt_tr) with a
#call to NOVAS
```

```
jd_tr, alt_tr = sun_tr(jd_noon, delta_t, lon, lat, ↩
   Sun)

#Initiate the list of phenomena times
#Altitude of phenomena (alt_phen), computed using ↩
   refraction of Sun and
#horizon, semi-diameter of Sun, and horizontal ↩
   parallax of Sun
phen_times = []
labels = ["Begin civil twilight", "Sunrise", "Sun ↩
   Transit", "Sunset",
          "End civil twilight"]
alt_phen = [-6.0, -0.833333, -0.833333, -0.833333, ↩
   -6.0]
#guess time of phenomena, in number of hours from ↩
   transit
time_guess = [-6.5, -6.0, 0, 6.0, 6.5]

for i in xrange(0, len(alt_phen)):
    #Transit time and altitude already computed ↩
       using sun_tr. Use those
    #values to calculate rise/set/twilight values
    if i == 2:
        if alt_tr < alt_phen[2]:
            jd_phen = -999.0
            phen_times.insert(2, jd_phen)
        else:
            jd_phen = jd_tr
            phen_times.insert(2, jd_phen)
        continue

    #starting iteration
    n = 0
```

```
#del_t is an approximate variable term in ↩
   sunrise/set times and is a
#function of latitude and time of year from ↩
   winter solstice
del_t = (lat/25.7)*math.cos(2.0*math.pi*↩
   Fractional_Year(jd_tr))
if i > 2:
    del_t = -del_t
jd_phen = jd_tr + (time_guess[i] + del_t)/24.0

del_t = 0
n = n + 1
#iterate on algorithm using altitude and ↩
   derivative of altitude
while n < 20:
    #Calculating right ascension (ra) and ↩
       declination (dec) with a call
    #to NOVAS
    ra, dec, distance = app_planet(EphemerisJD(↩
       jd_phen, delta_t), Sun)
    del distance

    #After 1st iteration, make sure phenomenon ↩
       occurs at all
    if n >= 2:
        alt_max = 90.0 - abs(lat) + (math.↩
           copysign(1.0, lat) * dec)
        alt_min = -90.0 + abs(lat) + (math.↩
           copysign(1.0, lat) * dec)
        if alt_max < alt_phen[i]:
        #phenomenon doesn't happen, end program
            jd_phen = -999.0
            phen_times.insert(i, jd_phen)
            break
```

```
        elif alt_min > alt_phen[i]:
        #phenomenon doesn't happen, end program
            jd_phen = 999.0
            phen_times.insert(i, jd_phen)
            break

dec_dot = DeclinationChange(ra)

#Compute Greenwich Sidereal Time (gst) with call to NOVAS
gst = sidereal_time(int(jd_phen), jd_phen - int(jd_phen), delta_t)

sin_lat = math.sin(math.radians(lat))
cos_lat = math.cos(math.radians(lat))

sin_dec = math.sin(math.radians(dec))
cos_dec = math.cos(math.radians(dec))

sin_ha = math.sin(math.radians(HourAngle(gst, ra, lon)))
cos_ha = math.cos(math.radians(HourAngle(gst, ra, lon)))

#Compute altitude and derivative of altitude wrt time
sin_alt = sin_dec * sin_lat + cos_dec*cos_lat * cos_ha
alt = math.degrees(math.asin(sin_alt))
cos_alt = math.sqrt(1.0 - sin_alt**2)
if cos_alt < 1.0e-4:
    cos_alt = 1.0e-4
alt_dot = 1.0 / cos_alt * (sin_lat*cos_dec*dec_dot
```

```
                                        - cos_lat*cos_dec*↩
                                           sin_ha*ha_dot
                                        - cos_lat*sin_dec*↩
                                           cos_ha*dec_dot)

            #Compute residual altitude (del_alt) and ↩
               estimate
            #time correction (del_t)
            del_alt = alt_phen[i] - alt
            del_t = del_alt/alt_dot
            if abs(del_t) > 0.2:
                del_t = math.copysign(0.2, del_t)

            #correct time of phenomenon
            jd_phen = jd_phen + del_t

            #if del_t has converged, add times of ↩
               phenomena to list
            if abs(del_t) <= 3.0e-5:
                phen_times.insert(i, jd_phen)
                break

            n = n + 1

    return phen_times

def Gregorian_Date(jul_day, time_zone = 0.0):
    '''
    Inputs
    Julian Date: float, Julian date
    time zone: (optional) float, in relation to UT

    Outputs
    Year: integer
```

```
    Month: integer
    Day: integer
    Hour: integer
    Minute: integer
    Second: float

    Calculates the Gregorian date and time using a Julian date (UT) input. This
    function makes a call to the NOVAS cal_date function for the year, month,
    day, and hour. It then calculates the minute and seconds using the fractional
    hour provided by cal_date
    '''
    Year, Month, Day, frac_hour = cal_date(time + time_zone/24.0)
    #calculating the hour, minute, and second with the fraction portion of hour
    hour = int(frac_hour)
    frac_min = (frac_hour - hour) * 60.0
    minute = int(frac_min)
    frac_sec = (frac_min - minute) * 60.0
    second = int(frac_sec)
    return (Year, Month, Day, hour, minute, second)

def sun_tr(jd_noon, delta_t, lon, lat, sun, time_prec = 3.0E-5):
    '''
    This subroutine computes the time of sun transit as well as the sun's
    altitude, horizontal parallax, and semi-diameter at transit.
    Inputs:
    jd_noon Julian date of local clock noon
```

```
delta_t Delta T value in seconds
lon Longitude (E +, W -)
lat Latitude (N +, S -)

Outputs:
JdTr Julian date of sun transit
AltTr Altitude of sun at transit, in degrees
Needs expl
'''
#compute transit time
jul_date = jd_noon
#start iteration and initialze the transit time ↩
   corection (del_t) value
n = 0
del_t = 1
while abs(del_t) > time_prec and n < 10:
    #calculate solar right ascension (ra) and ↩
       declination (dec) using
    #call to NOVAS
    ra, dec, distance = app_planet(EphemerisJD(↩
       jul_date, delta_t), Sun)
    del distance
    #Compute Greenwich Sidereal Time (gst) with call↩
        to NOVAS
    gst = sidereal_time(int(jul_date), jul_date-int(↩
       jul_date), delta_t)

    #compute time correction for transit
    del_t = -HourAngle(gst, ra) / 360.0
    jul_date = jul_date + del_t
    n = n + 1
    jul_date_transit = jul_date

#compute transit altitude
```

```
    alt_transit = 90.0 + lat - dec

    return jul_date_transit, alt_transit

def HoursToDegrees(Angle):
    '''
    Converts hours to degress given an angle in hours
    '''
    return Angle*15.0

def EphemerisJD(jul_date, delta_t):
    '''
    Calculates the ephemeris Julian Date using the ↩
       Julian date and delta T
    '''
    eph_jd = jul_date + (delta_t / 86400.0)
    return eph_jd

def HourAngle(gst, ra, lon):
    '''
    Calculates the solar hour angle using Greenwich ↩
       Sidereal Time, solar
    right ascension, and longitude
    '''
    ha = math.fmod(HoursToDegrees(gst) + lon - ↩
       HoursToDegrees(ra), 360.0)
    if ha > 180.0:
        ha = ha - 360.0
    elif ha < -180:
        ha = ha + 360.0

    return ha
```

A.2 Horizon Crossing

This program in Python 2.7 computes the amount of time it takes for the Sun to cross the horizon on a given day of the year. It begins by computing the day of the year from the date, then computes the radial distance to the Sun based on the eccentricity of the orbit, the date the speed of revolution. It then determines the angular size of the Sun and, finally, computes the time to cross the horizon. This final calculation works under the assumption that the angle of crossing is equivalent to the latitude. This means that it does not incorporate the declination of the sun on the day in question. The error is discussed in Chapter 5.

```
'''
This Python program computes the amount of time it takes↩
    for the sun to cross the horizon on a given day of ↩
   the year. It begins by computing the day of the year ↩
   from the date, then computes the radial distance to ↩
   the sun based on the eccentricity of the orbit, the ↩
   date the speed of revolution. It then determines the ↩
   angular size of the sun and finally computes the time ↩
   to cross the horizon.
'''
import math

#Conversion from date to day of year from Astronomical ↩
   Algorithms by Meeus pg 65
month = raw_input("Month: ")
month = float(month)
```

```
day = raw_input("Day: ")
day = float(day)
K = raw_input("Leap Year(1 for yes, 2 for no): ")
K = float(K)
latitude = raw_input("Latitude: ")
latitude = float(latitude)
latitude = math.radians(latitude)
t_start = raw_input("Day of Perihelion this year: ")
t_start = int(t_start)

def datetoday():
  N = int(275 * month/9) - K * int((month + 9)/12) + day↩
      - 30
  return N

#Radial distance to sun calculation
e = 0.0167 #eccentricity of earth's orbit
a = 149598023 #semi-major axis of earth's orbit (km)
w = (2 * math.pi)/365.25 #speed of earth's rotation
t = datetoday() - t_start #day of year shifted so that ↩
   perihelion occurs on correct date
#the following loop ensures that t > 0, ensuring that ↩
   the date is never negative
if t < 0:
  t = t + 365.25

def distance():
  theta = w * t #- math.pi
  r = a * (1 - e**2)/(1 + e*math.cos(theta))
  return r

#Angular diameter of the sun
D = 1.392e6 #diameter of sun (km)
def angularsize():
```

```
  d = 2 * math.atan(D/(2*distance()))
  d = d * (360/(2*math.pi)) #Conversion from radians to ↩
     degrees
  return d

#Time to cross horizon (seconds)
h = (24 * 60 * 60) / 360 #rotational speed of earth
def crossing():
  sec = h *angularsize()/math.cos(latitude)
  return sec

print crossing(), "sec"
```

A.3 Hohenkerk & Sinclair Refraction

This program in Fortran-77 computes atmospheric refraction on the horizon using the algorithm described by H&S in their 2008 technical note.[25] The algorithm is discussed in Chapter 2, and is verified in Chapter 3.

```
      SUBROUTINE hmnaoref(z0, h0, t0, p0, ups, wl, ph, ↩
         as, eps, ref)

      IMPLICIT none
      INTEGER i, j, k, in, is, istart
      DOUBLE PRECISION z0, h0, t0, p0, ups, wl, ph, as, ↩
         eps, ref, refi
      DOUBLE PRECISION r, hepsr, wlsq, z1, psat, a(10), ↩
         gb, z0r, t0c
```

```
      DOUBLE PRECISION pw0, r0, sk0, f0, t00, n0, dndr0,↩
         rt, nt, tt
      DOUBLE PRECISION dndrt, zt, ft, dndrts, nts, zts, ↩
        fts, rs, ns
      DOUBLE PRECISION dndrs, zs, fs, ref0, refp, reft, ↩
        fb, h, step
      DOUBLE PRECISION z, rg, t, tg, n, dndr, f, fe, fo,↩
         ff, ex1, ex2
      DOUBLE PRECISION gcr, md, mw, gamma, z2, s, ht, hs↩
        , dgr

C-----Constants of the atmosphere.
      PARAMETER (gcr = 8314.32D0)
      PARAMETER (md = 28.9644D0)
      PARAMETER (mw = 18.0152D0)
      PARAMETER (gamma = 18.36D0)
      PARAMETER (z2 = 11.2684D-06)

C-----Equatorial radius of the Earth.
      PARAMETER (s = 6378136.6D0)

C-----Height of tropopause and extent of atmosphere in ↩
   metres.
      PARAMETER (ht = 11000.0D0)
      PARAMETER (hs = 80000.0D0)

C-----Degrees to radians.
      PARAMETER (dgr = 1.74532925199433295769236 91D-2)

C-----Integration limits in radians.
      hepsr = 0.5D0*eps*dgr/3600.0D0

C-----Convert ZD to radians
      z0r = z0*dgr
```

```
C-----Set up parameters defined at the observer for the atmosphere.
C      Gravity:
       gb = 9.806248D0*(1.0D0-0.0026442D0*DCOS (2D0*ph*dgr))
       gb = gb-3.086D-6*h0

C-----Factor for optical wavelengths.
       wlsq = wl*wl
       z1 = (287.6155D0+(1.62887D0+0.01360D0/wlsq)/wlsq)
       z1 = z1*273.15D-6/1013.25D0
       a(1) = DABS (as)
       a(2) = (gb*md)/gcr
       a(3) = a(2)/a(1)
       a(4) = gamma

C-----Wet air:
       t0c = t0-273.15D0
       ex1 = (0.7859D0+0.03477D0*t0c)/(1.0D0+0.00412D0*t0c)
       ex2 = (1.0D0+p0*(4.5D-6+6.0D-10*t0c*t0c))
       psat = 10.0D0**(ex1*ex2)
       pw0 = ups*psat/(1.0D0-(1.10D0-ups)*psat/p0)
       a(5) = pw0*(1.0D0-mw/md)*a(3)/(a(4)-a(3))
       a(6) = p0+a(5)
       a(7) = z1*a(6)/t0
       a(8) = (z1*a(5)+z2*pw0)/t0
       a(9) = (a(3)-1.0D0)*a(1)*a(7)/t0
       a(10) = (a(4)-1.0D0)*a(1)*a(8)/t0

C-----At the observer.
       r0 = s+h0
       CALL atmostro(r0, t0, a, r0, t00, n0, dndr0)
```

```
      sk0 = n0*r0*DSIN (z0r)
      f0 = refi(r0, n0, dndr0)

C-----At the Tropopause in the Troposphere.
      rt = s+ht
      CALL atmostro(r0, t0, a, rt, tt, nt, dndrt)
      zt = DASIN (sk0/(rt*nt))
      ft = refi(rt, nt, dndrt)

C-----At the Tropopause in the Stratosphere.
      CALL atmosstr(rt, tt, nt, a(2), rt, nts, dndrts)
      zts = DASIN (sk0/(rt*nts))
      fts = refi(rt, nts, dndrts)

C-----At the stratosphere limit
      rs = s+hs
      CALL atmosstr(rt, tt, nt, a(2), rs, ns, dndrs)
      zs = DASIN (sk0/(rs*ns))
      fs = refi(rs, ns, dndrs)

C-----Integrate the refraction integral in the ↩
   troposphere and
C     stratosphere. ie Ref=Ref troposphere + Ref ↩
   stratosphere.

C-----Initial step lengths etc.
      ref0 = -999.999D0
      is = 16
      DO 400 k=1,2
         istart = 0
         fe = 0.0D0
         fo = 0.0D0
         IF (k .EQ. 1) THEN
            h = (zt-z0r)/DBLE (is)
```

```
            fb = f0
            ff = ft
         ELSE IF (k .EQ. 2) THEN
            h = (zs-zts)/DBLE (is)
            fb = fts
            ff = fs
         END IF
         in = is-1
         is = is/2
         step = h
  100    CONTINUE
         DO 300 i=1,in
            IF (i .EQ. 1 .AND. k .EQ. 1) THEN
               z = z0r+h
               r = r0
            ELSE IF (i .EQ. 1 .AND. k .EQ. 2) THEN
               z = zts+h
               r = rt
            ELSE
               z = z+step
            END IF

C-----Given the zenith distance (z) find r.
            rg = r
            DO 200 j=1,4
               IF (k .EQ. 1) THEN
                  CALL atmostro(r0, t0, a, rg, tg, n, ↩
                     dndr)
               ELSE IF (k .EQ. 2) THEN
                  CALL atmosstr(rt, tt, nt, a(2), rg, n,↩
                      dndr)
               END IF
               rg = rg-((rg*n-sk0/DSIN (z))/(n+rg*dndr))
  200       CONTINUE
```

```
            r = rg

C-----Find refractive index and integrand at r.
            IF (k .EQ. 1) THEN
               CALL atmostro(r0, t0, a, r, t, n, dndr)
            ELSE IF (k .EQ. 2) THEN
               CALL atmosstr(rt, tt, nt, a(2), r, n, ↩
                  dndr)
            END IF
            f = refi(r, n, dndr)
            IF (istart .EQ. 0 .AND. MOD (i, 2) .EQ. 0) ↩
               THEN
               fe = fe+f
            ELSE
               fo = fo+f
            END IF
  300    CONTINUE

C-----Evaluate the integrand using Simpsons Rule.
         refp = h*(fb+4.0D0*fo+2.0D0*fe+ff)/3.0D0

C-----Test for convergence.
         IF (DABS (refp-ref0) .GT. hepsr) THEN
            is = 2*is
            in = is
            step = h
            h = h/2.0D0
            fe = fe+fo
            fo = 0.0D0
            ref0 = refp
            IF (istart .EQ. 0) istart=1
            GOTO 100
         END IF
         IF (k .EQ. 1) reft=refp
```

```
  400 CONTINUE

C-----Refraction in the troposphere + stratosphere in ↩
   degrees.
      ref = (reft+refp)/dgr
      ref = ref*60.0D0

      RETURN
      END

*******************************************************
      SUBROUTINE atmostro(r0, t0, a, r, t, n, dndr)

      IMPLICIT NONE

      DOUBLE PRECISION r0, t0, a(10), r, t, n, dndr, tt0↩
         , tt01, tt02

      t = t0-a(1)*(r-r0)
      tt0 = t/t0
      tt01 = tt0**(a(3)-2.0D0)
      tt02 = tt0**(a(4)-2.0D0)
      n = 1.0D0+(a(7)*tt01-a(8)*tt02)*tt0
      dndr = -1.0D0*a(9)*tt01+a(10)*tt02

      RETURN
      END

*******************************************************
      SUBROUTINE atmosstr(rt, tt, nt, taw, r, n, dndr)

      IMPLICIT NONE
      DOUBLE PRECISION rt, tt, nt, taw, r, n, dndr, b
```

```
      b = taw/tt
      n = 1.0D0+(nt-1.0D0)*DEXP (-b*(r-rt))
      dndr = -1.0D0*b*(nt-1.0D0)*DEXP(-b*(r-rt))

      RETURN
      END

*******************************************************
      DOUBLE PRECISION FUNCTION refi(r, n, dndr)

      IMPLICIT none

      DOUBLE PRECISION r, n, dndr

      refi = r*dndr/(n+r*dndr)

      RETURN
      END
```

A.4 Bennett Refraction

This program in Python 2.7 computes atmospheric refraction on the horizon using the algorithm described by Bennett and modified for the NA.[26][40] The user can chose to include dip with the apparent_dip and geometric_dip functions.

```
import numpy as np
from numpy import arange
import matplotlib
```

```
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt

def Bennett(H, T, P):
    '''
    This function calculates atmospheric refraction (↩
       degrees) using apparent
    altitude (H, degrees), pressure (P, mb), and ↩
       temperature (T, degrees C).
    This equation is found in the Explanatory Supplement↩
        for the Astronomical
    Almanac EQ 7.93. It is a modification of Bennett↩
       (1982) by Hohenkerk to give
    a better match for the values found in the Nautical ↩
       Almanac put out by the
    USNO.
    '''
    f = 0.28*P / (T + 273)
    R = 0.0167 / np.tan(np.radians(H + 7.32/(H + 4.32)))
    return f*R

def arcmin_to_deg(arcmin):
    '''
    This function takes inputs of arcminute and converts↩
        to degrees. It is
    primarily useful for refraction outputs that need to↩
        be converted to degrees
    for use in the rise/set Calculator
    '''
    return arcmin/60.0

def geometric_dip(h):
    '''
```

```
    This function calculates the geometric dip (degrees)↩
        required to find the
    apparent dip of an object when combined with ↩
       refraction. It takes the height
    of an observer (h, meters) above the geoid.
    '''
    R_earth = 6378136.8
    dip = np.arccos(R_earth/(R_earth + h))
    return np.degrees(dip)

def apparent_dip(h, ref):
    '''
    This function calculates the refraction of the ↩
       horizon (degrees) using
    apparent dip. It takes the height of the observer (h↩
       , meters) above the
    geoid and
    '''
    app_dip = geometric_dip(h)
    app_height = 0.0 - app_dip
    n = 0
    ref_0 = 0.0

    for n in xrange(20):
        if abs(ref_0 - ref) < 0.004167:
            break
        ref_0 = ref
        app_dip = app_dip + ref
        print 'app dip = ', app_dip
        app_height = 0.0 - app_dip
        ref = Bennett(app_height, temp, pres)
    return ref, app_height
```

A.5 Schlyter API Request

This program in Python 2.7 computes sunrise and sunset times produced using the Schlyter algorithm. It does so by querying the sunrise-sunset.org API.[66] The dates and times included here are those used to produce the graphs in Section 3.3.2.

```
import requests
import json
import time
import calendar
import numpy as np
```

```
dates = ("1/27/2000", "3/6/2000", "4/23/2000", "1/6/2000", "7/19/2000", "9/6/2000", "10/15/2000", "11/23/2000", "1/17/2003", "3/6/2003", "4/24/2003", "6/2/2003", "7/19/2003", "8/28/2003", "10/6/2003", "11/23/2003", "2/25/2006", "4/5/2006", "5/23/2006", "7/1/2006", "8/18/2006", "10/5/2006", "11/23/2006", "1/7/2009", "2/24/2009", "4/4/2009", "5/13/2009", "6/30/2009", "8/18/2009", "10/5/2009", "11/23/2009", "1/1/2010", "2/18/2010", "4/28/2010", "6/14/2010", "7/24/2010", "9/10/2010", "10/28/2010", "12/15/2010", "1/4/2014", "2/21/2014", "4/2/2014", "5/19/2014", "7/8/2014", "8/25/2014", "10/12/2014", "11/30/2014", "1/11/2016", "2/29/2016", "4/17/2016", "5/26/2016", "7/13/2016", "8/31/2016", "10/9/2016", "11/26/2016", "1/10/2019", "3/1/2019", "4/18/2019", "6/5/2019", "7/24/2019", "9/10/2019", "10/29/2019", "12/7/2019", "1/1/2022", "2/18/2022", "3/29/2022", "5/16/2022", "6/25/2022", "8/12/2022", "9/30/2022", "11/17/2022", "1/4/2023", "2/13/2023", "3/23/2023", "5/10/2023", "6/28/2023", "8/15/2023", "9/24/2023", "11/11/2023")

latitudes = ("0.0", "10.0", "20.0", "40.0", "50.0", "-5.0", "-15.0", "-30.0", "-45.0", "-60.0", "-75.0", "66.0", "70.0", "75.0", "-80.0", "85.0", "89.0" )

lng = "0.0"

def SchylterRequest(dates, latitudes, longitudes):
    '''
    This program takes a list of dates and geographic coordinates as input. We used 0N for longitude for all calculations. It requests rise/set times using an API. The sunrise-sunset.org API returns them in UNIX epoch time.
```

```
'''

sunrise_times = []
sunset_times = []

for lat in latitudes:
    for date in dates:
        #make sure that the variable names match
        payload = {
        "date": date,
        "lat": lat,
        "lng": lng
        }
        #request the information in the API
        response = requests.get("https://api.sunrise↩
           -sunset.org/json", params = payload)
        #load the data provided above
        data = json.loads(response.content)
        #parse the sting to be read as a date tuple
        date = time.strptime(date, '%m/%d/%Y')

        newsun = data["results"]
        old_sunrise = newsun["sunrise"]
        #parse the returned string to be read as a ↩
           time tuple
        new_sunrise = time.strptime(old_sunrise, '%I↩
           :%M:%S %p')
        #convert the date and time tuples to epoch ↩
           time
        sunrise_epoch = calendar.timegm((int(date↩
           [0]), int(date[1]), int(date[2]),
            int(new_sunrise[3]), int(new_sunrise[4])↩
               , int(new_sunrise[5])))
        sunrise_times.append(sunrise_epoch)
```

```
            old_sunset = newsun["sunset"]
            #parse the returned string to be read as a ↩
               time tuple
            new_sunset = time.strptime(old_sunset, '%I:%↩
               M:%S %p')
            #convert the date and time tuples to epoch ↩
               time
            sunset_epoch = calendar.timegm((int(date[0])↩
               , int(date[1]), int(date[2]),
                int(new_sunset[3]), int(new_sunset[4]), ↩
                   int(new_sunset[5])))
            sunset_times.append(sunset_epoch)

    return sunrise_times, sunset_times
```

A.6 Meeus Request

This program in Python 2.7 computes sunrise and sunset times produced using the Schlyter algorithm. It does so with the Python transit_sunrise_sunset() function of the python package pvlib.[67] The dates and times included here are those used to produce the graphs in Section 3.3.2.

```
import pvlib
from pvlib.spa import transit_sunrise_sunset
import numpy as np
import time
```

```
'''
This program uses the Meeus algorithm for calculating ↩
   sunrise and sunset times. It takes an array of dates (↩
   in epoch format), latitude and longitude coordinates, ↩
   and outputs UNIX epoch times of sunrise and sunsets ↩
   for the dates and times in question. It returns the ↩
   times to a file.
'''

riseset = open("MEEUStimes.txt", "a")

date = np.array([948931200, 952300800, 956448000, ↩
   947116800, 963964800, 968198400, 971568000, 974937600,↩
    1042761600, 1046908800, 1051142400, 1054512000, ↩
   1058572800, 1062028800, 1065398400, 1069545600, ↩
   1140825600, 1144195200, 1148342400, 1151712000, ↩
   1155859200, 1160006400, 1164240000, 1231286400, ↩
   1235433600, 1238803200, 1242172800, 1246320000, ↩
   1250553600, 1254700800, 1258934400, 1262304000, ↩
   1266451200, 1272412800, 1276473600, 1279929600, ↩
   1284076800, 1288224000, 1292371200, 1388793600, ↩
   1392940800, 1396396800, 1400457600, 1404777600, ↩
   1408924800, 1413072000, 1417305600, 1452470400, ↩
   1456704000, 1460851200, 1464220800, 1468368000, ↩
   1472601600, 1475971200, 1480118400, 1547078400, ↩
   1551398400, 1555545600, 1559692800, 1559692800, ↩
   1563926400, 1568073600, 1572307200, 1575676800, ↩
   1640995200, 1645142400, 1648512000, 1652659200, ↩
   1652659200, 1656115200, 1660262400, 1664496000, ↩
   1668643200, 1672790400, 1676246400, 1679529600, ↩
   1683676800, 1687910400, 1692057600, 1695513600, ↩
   1699660800])
```

```
latitude = [0.0 , 10.0, 20.0, 40.0, 50.0, -5.0, -15.0, ↩
   -30.0, -45.0, -60.0, -75.0, 66.0, 70.0, 75.0, -80.0, ↩
   85.0, 89.0]

lng = 0.0

for lat in latitude:
    #the function takes dates, latitude, longitude, ↩
       delta t, and
    data = transit_sunrise_sunset(date, lat, lng, ↩
       63.8285, 3)

    for sunrise in data[1]:
        if not np.isnan(sunrise).any():
            #account for dates in which sunrise does not↩
                occur
            sunrise = time.strftime('%Y-%m-%d %H:%M:%S',↩
                time.gmtime(sunrise))
        else:
            sunrise = "00:00:00"

    for sunset in data[2]:
        #account for dates in which sunset does not ↩
           occur
        if not np.isnan(sunset).any():
            sunset = time.strftime('%Y-%m-%d %H:%M:%S', ↩
               time.gmtime(sunset))
        else:
            sunset = "00:00:00"

    riseset.write('%s %s %s %s\n'%(lat, lng, sunrise, ↩
       sunset))

riseset.close()
```

A.7 USNO API Request

This program in Python 2.7 computes sunrise and sunset times produced using the USNO algorithm. It does so by querying the "Complete Sun and Moon Data for One Day" API. The dates and times included here are those used to produce the graphs in Section 3.3.2.

```
import requests
import json
import time
import calendar
import numpy as np

'''
This program uses the USNO algorithm for calculating ↩
   sunrise and sunset times. It requests rise/set times ↩
   by querying the USNO's "Complete Sun and Moon Data for↩
    One Day" API. It takes strings of dates, and latitude↩
   /longitude coordinates. This program then converts the↩
    output strings to values in UNIX epoch time to more ↩
   easily compare with outputs from other APIs.
'''
```

```
dates = ("1/27/2000", "3/6/2000", "4/23/2000", "1/6/2000↩
   ", "7/19/2000", "9/6/2000", "10/15/2000", "11/23/2000"↩
   , "1/17/2003", "3/6/2003", "4/24/2003", "6/2/2003", "↩
   7/19/2003", "8/28/2003", "10/6/2003", "11/23/2003", "↩
   2/25/2006", "4/5/2006", "5/23/2006", "7/1/2006", "↩
   8/18/2006", "10/5/2006", "11/23/2006", "1/7/2009", "↩
   2/24/2009", "4/4/2009", "5/13/2009", "6/30/2009", "↩
   8/18/2009", "10/5/2009", "11/23/2009", "1/1/2010", "↩
   2/18/2010", "4/28/2010", "6/14/2010", "7/24/2010", "↩
   9/10/2010", "10/28/2010", "12/15/2010", "1/4/2014", "↩
   2/21/2014", "4/2/2014", "5/19/2014", "7/8/2014", "↩
   8/25/2014", "10/12/2014", "11/30/2014", "1/11/2016", "↩
   2/29/2016", "4/17/2016", "5/26/2016", "7/13/2016", "↩
   8/31/2016", "10/9/2016", "11/26/2016", "1/10/2019", "↩
   3/1/2019", "4/18/2019", "6/5/2019", "7/24/2019", "↩
   9/10/2019", "10/29/2019", "12/7/2019", "1/1/2022", "↩
   2/18/2022", "3/29/2022", "5/16/2022", "6/25/2022", "↩
   8/12/2022", "9/30/2022", "11/17/2022", "1/4/2023", "↩
   2/13/2023", "3/23/2023", "5/10/2023", "6/28/2023", "↩
   8/15/2023", "9/24/2023", "11/11/2023")

location = ("0.0N, 0.0E", "10.0N, 0.0E", "20.0N, 0.0E", ↩
   "40.0N, 0.0E", "50.0N, 0.0E", "5.0S, 0.0E", "15.0S, ↩
   0.0E", "30.0S, 0.0E", "45.0S, 0.0E", "60.0S, 0.0E", "↩
   75.0S, 0.0E", "66.0N, 0.0E", "70.0N, 0.0E", "75.0N, ↩
   0.0E", "80.0S, 0.0E", "85.0N, 0.0E", "89.0N, 0.0E" )

tz = 0

riseset = open('USNOepochTimes.txt', 'w+')
riseset = open("USNOtimesNEW.txt", "w+")
riseset.write("#Lat  Long \t Sunrise \t Sunset\n")

sunrise_times = []
```

```
sunset_times = []

for coords in location:
    for date in dates:
        payload = {
        "date": date,
        "coords": coords,
        "tz": tz
        }

        response = requests.get("http://api.usno.navy.↩
           mil/rstt/oneday", params = payload)

        #keep the content returned by the API
        data = json.loads(response.content)
        #Converts the date from a unicode value to a ↩
           time structure
        date = time.strptime(date, '%m/%d/%Y')
        #Look at the sun, ignore the moon
        newsun = data["sundata"]

        for phen in newsun:
            try:
                phen["phen"] == "R"
            #record 00:00:00 when sunrise does not occur
            except NameError:
                sunrise = (0, 0, 0)
                sunrise_epoch = calendar.timegm((int(↩
                   date[0]), int(date[1]), int(date[2]), ↩
                   int(sunrise[0]), int(sunrise[1]), int(↩
                   sunrise[2])))
            #record the time when it does
            else:
                if phen["phen"] == "R":
```

```
                sunrise = phen["time"]
                #converts the unicode values to time↩
                    structures
                sunrise = time.strptime(sunrise, '%H↩
                    :%M')
                #convert the time structure to unix ↩
                    time
                sunrise_epoch = calendar.timegm((int↩
                    (date[0]), int(date[1]), int(date↩
                    [2]), int(sunrise[3]), int(sunrise↩
                    [4]), int(sunrise[5])))
        try:
            phen["phen"] == "S"
        #record 00:00:00 when sunset does not occur
        except NameError:
            sunset = (0, 0, 0)
            sunset_epoch = calendar.timegm((int(date↩
                [0]), int(date[1]), int(date[2]), int(↩
                sunset[0]), int(sunset[1]), int(sunset↩
                [2])))
        #record when it does
        else:
            if phen["phen"] == "S":
                sunset = phen["time"]

                sunset = time.strptime(sunset, '%H:%↩
                    M')
                sunset_epoch = calendar.timegm((int(↩
                    date[0]), int(date[1]), int(date↩
                    [2]), int(sunset[3]), int(sunset↩
                    [4]), int(sunset[5])))

    sunrise_times.append(sunrise_epoch)
    sunset_times.append(sunset_epoch)
```

```
for i in range(len(sunrise_times)):
    riseset.write('{0} \t {1} \t {2} \t {3}\n'.format(↩
       location[i], sunrise_times[i], sunset_times[i]))

riseset.close()
```

Appendix B

Tables

Below are tables of values that were used to conduct the analysis in this dissertation.

B.1 Sampson Humidity Values

Below are the relative humidity values recorded by Sampson at the time of sunrise and sunset. The values were not included in his thesis, but were provided via private communication. Missing values were supplemented using historical weather data provided by Weather Underground from the nearby airport.[86]

Table B.1
Sampson Humidity Data

Sunrise Date	RH		Sunset Date	RH
12/29/1990	0.61		04/05/1991	0.49
01/02/1991	0.58		04/12/1991	0.40
01/03/1991	0.58		04/14/1991	0.26
01/04/1991	0.65		04/15/1991	0.40
01/06/1991	0.62		04/16/1991	0.49
01/08/1991	0.61		04/17/1991	0.34
01/09/1991	0.62		04/18/1991	0.31
01/10/1991	0.63		04/19/1991	0.28
01/11/1991	0.65		04/20/1991	0.27
01/12/1991	0.78		04/21/1991	0.27
01/14/1991	0.59		05/01/1991	0.26
01/20/1991	0.62		05/03/1991	0.36
01/22/1991	0.59		09/03/1991	0.51
02/01/1991	0.51		09/09/1991	0.66
02/04/1991	0.64		09/11/1991	0.45
02/06/1991	0.69		03/21/1992	0.60
02/09/1991	0.79		03/22/1992	0.40
02/11/1991	0.86		03/23/1992	0.38
02/25/1991	0.79		04/04/1992	0.31
03/01/1991	0.62		04/12/1992	0.48
03/05/1991	0.70		04/13/1992	0.39
03/09/1991	0.79		04/19/1992	0.87
03/11/1991	0.77		05/02/1992	0.30
03/13/1991	0.92		08/11/1992	0.46
03/14/1991	0.85		08/12/1992	0.31
03/19/1991	0.86		08/15/1992	0.32
03/20/1991	1.00		08/17/1992	0.29
03/26/1991	0.85		08/18/1992	0.40
03/29/1991	0.64		08/26/1992	0.42
03/30/1991	0.60		09/03/1992	0.54
03/31/1991	0.74		09/14/1992	0.65
04/01/1991	0.75		09/15/1992	0.33
04/02/1991	0.75		09/16/1992	0.49
04/04/1991	0.60		09/21/1992	0.53
04/05/1991	0.86		09/30/1992	0.23
04/08/1991	0.86		10/02/1992	0.38
04/09/1991	0.86		10/05/1992	0.32

Continued on next page

Table B.1 – continued from previous page

Sunrise Date	RH		Sunset Date	RH
04/12/1991	0.69		11/09/1992	0.77
04/13/1991	0.48		11/23/1992	0.73
04/14/1991	0.61		11/30/1992	0.56
04/15/1991	0.80		12/01/1992	0.87
04/16/1991	0.86		12/02/1992	0.59
04/17/1991	0.64		12/03/1992	0.80
04/18/1991	0.48		12/04/1992	0.63
04/19/1991	0.56		12/05/1992	0.52
04/21/1991	0.65		12/06/1992	0.64
04/22/1991	0.60		12/31/1992	0.58
04/23/1991	0.65		01/05/1993	0.65
04/24/1991	0.60		01/08/1993	0.83
04/30/1991	0.59		01/10/1993	0.63
05/01/1991	0.58		01/12/1993	0.67
05/02/1991	0.64		01/13/1993	0.71
05/04/1991	0.44		01/14/1993	0.70
05/07/1991	0.75		01/16/1993	0.59
05/08/1991	0.81		01/17/1993	0.79
05/11/1991	0.66		01/18/1993	0.73
05/12/1991	0.62		01/22/1993	0.74
05/16/1991	0.61		01/25/1993	0.68
05/17/1991	0.54		01/29/1993	0.73
05/20/1991	0.87		01/30/1993	0.49
05/27/1991	0.87		01/31/1993	0.50
05/29/1991	0.76		02/01/1993	0.61
05/30/1991	0.76		02/02/1993	0.49
05/31/1991	0.76		02/03/1993	0.70
06/05/1991	0.66		02/04/1993	0.70
07/29/1991	0.82		02/08/1993	0.65
07/31/1991	0.82		02/12/1993	0.67
08/07/1991	0.82		02/13/1993	0.68
08/09/1991	0.83		02/16/1993	0.66
08/10/1991	0.77		02/17/1993	0.61
08/13/1991	0.94		03/06/1993	0.49
08/14/1991	0.88		03/12/1993	0.54
08/21/1991	0.77		03/13/1993	0.68
08/26/1991	0.81		03/17/1993	0.48
08/30/1991	0.77		03/20/1993	0.56
09/03/1991	0.66		03/25/1993	0.39

Continued on next page

Table B.1 – continued from previous page

Sunrise Date	RH		Sunset Date	RH
09/07/1991	0.93		03/26/1993	0.52
09/10/1991	0.75		04/04/1993	0.71
09/11/1991	0.87		04/05/1993	0.71
09/17/1991	0.80		04/07/1993	0.80
09/19/1991	0.66		04/12/1993	0.87
09/22/1991	1.00		04/14/1993	0.61
09/24/1991	0.62		04/15/1993	0.61
09/26/1991	0.87		04/19/1993	0.81
09/27/1991	0.62		05/01/1993	0.31
09/28/1991	0.71		05/10/1993	0.31
10/02/1991	0.70		05/13/1993	0.38
10/03/1991	0.75		05/14/1993	0.39
10/04/1991	0.69		05/16/1993	0.31
10/08/1991	0.48		05/19/1993	0.58
10/09/1991	0.39		05/20/1993	0.47
10/10/1991	0.76		06/02/1993	0.41
10/11/1991	0.76		06/05/1993	0.39
10/18/1991	0.85		06/09/1993	0.36
10/19/1991	0.81		06/19/1993	0.82
10/23/1991	0.78		06/20/1993	0.48
10/28/1991	0.71		06/21/1993	0.49
10/29/1991	0.78		06/25/1993	0.82
10/30/1991	0.66		07/17/1993	0.94
11/02/1991	0.77		08/19/1993	0.64
11/06/1991	0.75		08/20/1993	0.53
11/07/1991	0.86		08/30/1993	0.63
11/09/1991	0.78		09/02/1993	0.77
11/14/1991	0.80		09/03/1993	0.60
11/15/1991	0.67		09/06/1993	0.48
11/18/1991	0.86		09/07/1993	0.59
11/19/1991	0.93		09/08/1993	0.52
12/10/1991	0.50		09/09/1993	0.55
12/11/1991	0.59		09/10/1993	0.55
12/13/1991	0.72		09/18/1993	0.44
12/14/1991	0.85		09/21/1993	0.53
12/16/1991	0.54		09/22/1993	0.50
12/17/1991	0.79		09/26/1993	0.50
12/20/1991	0.57		09/28/1993	0.47
01/05/1992	0.92		09/29/1993	0.38

Continued on next page

Table B.1 – continued from previous page

Sunrise Date	RH		Sunset Date	RH
01/07/1992	0.92		10/01/1993	0.53
01/09/1992	0.73		10/02/1993	0.66
01/12/1992	0.86		10/03/1993	0.46
01/14/1992	0.77		10/08/1993	0.65
01/16/1992	0.72		10/09/1993	0.58
01/22/1992	0.73		10/10/1993	0.38
01/27/1992	0.68		10/11/1993	0.54
01/29/1992	0.74		12/08/1993	0.58
02/03/1992	0.80		12/16/1993	0.92
02/04/1992	0.69		12/18/1993	0.80
02/05/1992	0.86			
02/06/1992	0.85			
02/07/1992	0.92			
02/12/1992	0.78			
02/23/1992	0.92			
02/28/1992	0.65			
02/29/1992	0.52			
03/01/1992	0.64			
03/02/1992	0.86			
03/03/1992	0.93			
03/05/1992	1.00			
03/08/1992	0.79			
03/09/1992	0.79			
03/12/1992	0.70			
03/13/1992	0.65			
03/15/1992	0.75			
03/16/1992	0.80			
03/22/1992	0.74			
03/24/1992	0.58			
03/26/1992	0.69			
03/27/1992	0.64			
03/30/1992	0.63			
04/01/1992	0.65			
04/02/1992	0.65			
04/03/1992	0.70			
04/04/1992	0.59			
04/10/1992	0.71			
04/12/1992	0.73			
04/13/1992	0.73			

Continued on next page

Table B.1 – continued from previous page

Sunrise Date	RH		Sunset Date	RH
04/15/1992	0.81			
05/01/1992	0.45			
05/02/1992	0.52			
05/03/1992	0.49			
05/25/1992	0.66			
05/26/1992	0.61			
05/28/1992	0.57			
05/29/1992	0.81			
06/10/1992	0.82			
06/11/1992	0.76			
06/12/1992	0.58			
06/14/1992	0.72			
06/17/1992	0.72			
06/19/1992	0.72			
06/21/1992	0.52			
06/22/1992	0.44			
06/24/1992	0.67			
06/28/1992	0.76			
06/29/1992	0.66			
07/20/1992	0.87			
07/21/1992	0.71			
07/24/1992	0.82			
07/27/1992	0.62			
08/06/1992	0.88			
08/11/1992	0.81			
08/13/1992	0.67			
08/14/1992	0.68			
08/17/1992	0.66			
08/25/1992	0.70			
09/03/1992	0.87			
09/14/1992	0.64			
09/15/1992	0.69			
09/16/1992	0.80			
09/21/1992	0.75			
09/24/1992	0.87			
09/26/1992	0.86			
09/27/1992	0.93			
09/29/1992	0.75			
10/01/1992	0.76			

Continued on next page

Table B.1 – continued from previous page

Sunrise Date	RH		Sunset Date	RH
10/02/1992	0.71			
10/03/1992	0.44			
10/05/1992	0.56			
10/07/1992	0.69			
10/09/1992	0.48			
10/11/1992	0.60			
10/13/1992	0.80			
10/14/1992	0.80			
10/16/1992	0.85			
10/22/1992	0.81			
10/25/1992	0.69			
11/09/1992	0.86			
11/10/1992	0.74			
11/12/1992	0.63			
11/24/1992	0.73			
11/26/1992	0.85			
11/29/1992	0.63			
12/01/1992	0.63			
12/03/1992	0.72			
12/04/1992	0.67			
12/05/1992	0.68			
12/06/1992	0.79			
12/13/1992	0.72			
12/18/1992	0.69			
12/19/1992	0.77			
12/30/1992	0.68			
12/31/1992	0.61			
01/02/1993	0.71			
01/05/1993	0.62			
01/09/1993	0.62			
01/10/1993	0.58			
01/14/1993	0.62			
01/15/1993	0.66			
01/17/1993	0.77			
01/18/1993	0.66			
01/20/1993	0.63			
01/21/1993	0.80			
01/22/1993	0.79			
01/24/1993	0.61			

Continued on next page

Table B.1 – continued from previous page

Sunrise Date	RH		Sunset Date	RH
01/28/1993	0.70			
01/29/1993	0.73			
02/01/1993	0.60			
02/02/1993	0.74			
02/06/1993	0.93			
02/07/1993	0.80			
02/08/1993	0.86			
02/12/1993	0.79			
02/15/1993	0.71			
02/16/1993	0.71			
02/17/1993	0.71			
02/23/1993	0.71			
10/20/1993	0.69			
10/23/1993	0.75			
10/29/1993	0.56			
11/05/1993	0.74			
11/17/1993	0.74			
11/24/1993	0.66			
11/25/1993	0.78			
11/26/1993	0.73			
12/08/1993	0.79			
12/17/1993	0.85			

Appendix C

Letters of Permission

Below is the letter of permission to reproduce the unpublished code for Hohenkerk & Sinclair's refraction model, which is provided in Appendix A.3.

Michigan Tech

Teresa Wilson <tawilson@mtu.edu>

2008 Tech Note

Catherine Hohenkerk <catherine.hohenkerk@gmail.com> Wed, Feb 14, 2018 at 11:36 AM
To: Teresa Wilson <tawilson@mtu.edu>
Cc: Jennifer Bartlett <jennifer.bartlett@navy.mil>

Dear Teresa

I have no problem at all, from my point of view, in you including your copy of my code - only too delighted, and i'm sure you will give appropriate references.

However, perhaps Jennifer aught to consult Steve Bell at HMNAO. As you know I am retired and since the end of January, no longer have any "contract" status. I am sure there will be no problems.

Good luck from a cold wet and windy Taunton

Catherine
-

On 14 February 2018 at 15:56, Teresa Wilson <tawilson@mtu.edu> wrote:

> Good morning!
>
> I am Jennifer Bartlett's graduate student who's working on refraction. Thank you so much for all your help with getting all the background straight! I have one more question for you. I transcribed the code you wrote for the 2008 updated version of Tech Note 63 and have been using that for my analysis. Since it's not yet published, may I include the code as an appendix to my dissertation? Thank you!
>
> --
> Teresa Wilson
> PhD Candidate | Physics Department
> 906-487-2078 | Fisher B008
> Michigan Technological University