UNIVERSITY OF CALIFORNIA, SAN DIEGO

High-Precision Lunar Ranging and Gravitational Parameter Estimation With the Apache Point Observatory Lunar Laser-ranging Operation

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Physics

by

Nathan H. Johnson

Committee in charge:

Professor Thomas W. Murphy, Jr., Chair
Professor Duncan Agnew
Professor William Coles
Professor Kim Griest
Professor Hans Paar

2015

The dissertation of Nathan H. Johnson is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____
Chair

University of California, San Diego

2015

DEDICATION

To my grandmother

Jean Kreizinger

for her inspiration and caring

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

The author gratefully acknowledges Professor Tom Murphy for his many gifts of wisdom and acts of kindness and forbearance.

Gratitude is also due to John Chandler of the Harvard-Smithsonian Center for Astrophysics, whose august voice is always now heard in the author's head when reading technical material, for many patient explanations.

Chapter 2, in part, has been submitted for publication of the material as it may appear in Applied Optics, 2015. Johnson, Nathan H.; Murphy, Thomas W.; Aull, Brian F.; Colmenares, Nicholas R.; Orin, Adam E., OSA Publishing, 2015. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in part, is currently being prepared for submission for publication of the material. Johnson, Nathan H.; Chandler, John F.; Murphy, Thomas W. The dissertation author was the primary investigator and author of this material.

VITA

2008	B. A. in Physics and Classical Studies *magna cum laude*, University of Pennsylvania
2008-2009	Graduate Teaching Assistant, University of California, San Diego
2009-2015	Graduate Student Researcher, University of California, San Diego
2015	Ph. D. in Physics, University of California, San Diego

ABSTRACT OF THE DISSERTATION

High-Precision Lunar Ranging and Gravitational Parameter Estimation With the Apache Point Observatory Lunar Laser-ranging Operation

by

Nathan H. Johnson

Doctor of Philosophy in Physics

University of California, San Diego, 2015

Professor Thomas W. Murphy, Jr., Chair

This dissertation is concerned with several problems of instrumentation and data analysis encountered by the Apache Point Observatory Lunar Laser-ranging Operation. Chapter 2 considers crosstalk between elements of a single-photon avalanche photodiode detector. Experimental and analytic methods were developed to determine crosstalk rates, and empirical findings are presented. Chapter 3 details electronics developments that have improved the quality of data collected by detectors of the same type. Chapter 4 explores the challenges of estimating

gravitational parameters on the basis of ranging data collected by this and other experiments and presents resampling techniques for the derivation of standard errors for estimates of such parameters determined by the Planetary Ephemeris Program (PEP), a solar-system model and data-fitting code. Possible directions for future work are discussed in Chapter 5. A manual of instructions for working with PEP is presented as an appendix.

Chapter 1

Motivation

The desire to understand and describe the motion of heavenly bodies – desire, as it turned out, for a gravitational theory – has been one of the great spurs to the development of scientific inquiry throughout history, from Ptolemy to Copernicus to Kepler to Newton and on to the present day. At every stage there has unfolded the great interplay between measurement and model which is at the heart of the scientific process, as new measurements, often permitted by novel techniques, have challenged the paradigm of the day and ushered in some new understanding. In this way, heliocentrism was superseded by geocentrism, and in turn by Newtonian mechanics and general relativity. We are entitled to ask: Do we today have in hand a perfect theory of gravitation, with which no valid measurement will ever be found to disagree? Or are we at one more waystation on a continuing journey? Today, the experimental pressure on gravitational theory is greater than ever before at a wide range of scales, from the laboratory to the solar system to pulsar systems, with the possible incipience of a new frontier in the coming years with the detection of the first gravitational waves. Also very fertile is the theoretical landscape, where a diverse array of efforts aimed at producing a unified physical

theory have resulted in predictions that may be testably different from those of general relativity. Furthermore, the impetus for understanding gravitation, which has dictated the expansion history of the universe and the formation of structure at all scales, has perhaps never been more present than it is today.

For all its import, gravity is a weak force. For the gravitational interaction between two electrons to be as strong as their electrostatic interaction, they would have to be $2 * 10^{21}$ times more massive than they are, as massive as a small grain of sand. Precise laboratory tests of gravity therefore confront considerable challenges in understanding background effects, although some have certainly done so with great success. Nevertheless it is not surprising that when Einstein first formulated general relativity, only the pristine gravitational laboratory of the solar system offered a venue in which its predictions could be compared with those of Newton. Of course the new theory appeared to pass the tests of the time, of the precession of the perihelion of Mercury and the deflection of starlight by the sun, and so became a standard that would not face another experimental challenge for decades. In the last 50 years, a steady flow of increasingly precise solar-system measurements have greatly increased the rigor with which gravitation can be tested. In the late 1960s and early 1970s, Apollo astronauts and ummanned Soviet missions placed five retroreflector arrays on the surface of the moon, and laser ranging between stations on earth and these arrays has been conducted at a handful of sites in the decades since. In 2006, the Apache Point Observatory Lunar Laser-ranging Operation (APOLLO, appropriately enough) was added to this short list of LLR efforts with the start of its science campaign, bearing the promise of producing measurements accurate at the millimeter level—an order of magnitude more precise than what had previously been achieved [1]. APOLLO has succeeded in reaching this unprecedented level of precision and continues to make measurements. Efforts

to characterize and improve the performance of the APOLLO experiment, and to derive scientifically useful results from its measurements along with those taken by other solar-system ranging endeavors, are the subject of this dissertation.

1.1 Properties of Gravitation

One of the basic building blocks used by Einstein to construct general relativity is the equivalence principle, an important concept that is often framed in one of a few different ways. The most essential of these, from the perspective of the theory, is the invariance of physical laws between inertial frames, such that the results of physical experiments cannot be used to determine (for example) whether one is freely falling in a gravitational potential or distant from any massive object. A consequence of this requirement leads to a perhaps more familiar formulation, that of the equal acceleration by gravity of masses of different compositions. If gravitational acceleration were composition-dependent, the outcome of experiments involving differing masses would depend on the presence of a gravitational field even in a freely falling laboratory, which is exactly what the original formulation of the principle indicates we must not permit.

We may wonder why we would view such a violation as even possible. Objects made of different materials seemed distinct enough to Galileo, but we know today that their differences are superficial in that all are constituted from the same elementary particles. Why, then, would we think it possible that they might be accelerated at different rates? In fact, the total mass-energy of differently constituted objects comes from several sources in composition-dependent proportions: the rest mass of their particles, but also the energy of the electromagnetic bonds between their atoms, the binding energy of their nuclei, and potentially the gravitational

binding energy of the whole. This leads to a third common formulation of the equivalence principle: the universal equivalence of gravitational and inertial mass, specifically that all forms of mass-energy contribute in the same way to each.

When we confine ourselves to the consideration of the space-time trajectories of uncharged masses with negligible self-gravitational energy, requiring only that they be permitted to differ in internal structure and composition, we have the so-called Weak Equivalence Principle (WEP). If we further stipulate the position- and velocity-independence of the outcome of any local nongravitational experiment, then we have formulated the Einstein Equivalence Principle (EEP). The Schiff conjecture holds that the WEP implies the EEP, but is unproven. The validity of the EEP is at any rate a common feature of all metric theories of gravity, in which the gravitational force is ascribed to space-time curvature. The Strong Equivalence Principle (SEP) extends these principles to local gravitational experiments and objects of non-negligible gravitational binding energy. Non-violation of these principles is one of the cornerstones of general relativity, but is not a universal feature of alternative theories, which may be tested by experiments sensitive to such violations.

The most precise tests of the WEP have long been torsion-balance experiments, in which the differential acceleration in a gravitational field of masses of different compositions is measured via the torque on a rod connecting them. Investigations of this type were pioneered by the Hungarian nobleman and physicist Lorand Eötvos, and the figure of merit for WEP violations is the eponymous Eötvos parameter η, being twice the magnitude of the difference of the masses' acceleration divided by the magnitude of the sum of the same. The current limit on the value of η is derived from the almost-eponymous Eöt-Wash torsion-balance experiments conducted by Eric Adelberger and associates at the University of Washington (e.g.

[5]), and is approximately a part in 10^{13}. Measurements of the earth-moon distance do constrain any WEP violation due to the presence of proportionally more metallic elements in the earth than in the moon, but the degree of difference in composition is obviously not what can be achieved in purpose-made laboratory masses, and even millimeter-level lunar laser ranges are not expected to provide competitive WEP limits. However, the earth and moon differ in proportion of self-gravitational mass-energy to total mass by about a factor of 20, far more than is conceivable in laboratory test, and so LLR does furnish the most stringent limits on SEP violation.

The recent discovery of the accelerating expansion of the universe, along with our general ignorance regarding the dark sector, provide a motivation for testing as many properties of gravity as possible with the greatest possible precision, while simultaneously encouraging for the proliferation of novel theories, whose specific predictions may permit falsification by gravitational experiments. Lunar laser ranging is sensitive to numerous aspects of gravitation and has the potential to advance both causes. In particular, the earth-moon distance is sensitive to time variation in the value of G, which is a likely feature of theories in which a scalar field plays the role of dark energy. Millimeter-level LLR also plausibly constitutes the best available constraint on the magnitudes of effects expected in GR, such as gravitomagnetism and geodetic precession, and the most sensitive probe of the inverse-square law for gravitational acceleration.

1.2 APOLLO

The corpus of ranging measurements taken in the nearly-pristine gravitational laboratory of the solar system dates back more than 50 years and includes radar ranges to the surfaces of the inner planets and observations of Mars probes in

addition to the lunar laser-ranging dataset. All of the available information can be leveraged in a grand fit to constrain the aforementioned properties of gravitation. LLR measurements have been taken since 1969 and so provide a long temporal baseline as well as very precise information about the gravitationally complex earth-moon system. Useful ranging data has been produced by projects at the MacDonald Observatory, which later moved to a dedicated telescope and became known as the MacDonald Laser Ranging System (approximately 1969 to present); the Observatoire de la Cote d'Azur in France; (approximately 1984 to present with interruptions); Haleakala in Hawaii (1984-1990); and a handful of other locations that have demonstrated ranging capability but have not produced large volumes of measurements. The typical uncertainty associated with LLR measurements declined steadily over this history to roughly 2 cm in 2005.

APOLLO began its science campaign in 2006 and quickly demonstrated capability of generating millimeter-accuracy ranges, using the 3.5-meter telescope at Apache Point Observatory and a laser system producing 100 ps pulses of 115 mJ at 20 Hz with a wavelength of 532 nm. Although it certainly pays to reduce experimental contributions to measurement uncertainty, the tilt of the retroreflector arrays due to lunar libration imposes an unavoidable single-photon uncertainty of 100-300 ps ($\sim$15 to 45 mm two-way), beyond which statistical reduction of uncertainty by collecting as many photons as possible is key; hence APOLLO's superior capability is due in considerable part to the large collecting area of the telescope and the high peak power of the laser. The real-time precision of the laser fire is approximately 1 μs, good for millimeter-level uncertainty given the instantaneous rate of change in the site-reflector distance, which is dominated by the 400 m/s local rotation velocity of the earth. The initial wavelength of the light from the Nd:YAG laser is 1064 nm, and a second-harmonic generator doubles

the frequency into the green. A fraction of the laser pulse is diverted to a fast photodiode, which serves as the timing anchor for the pulse and alerts the detection system to imminent returns.

The laser fire is coordinated with a transparent rotating optic on which there is a reflecting patch, such that the outgoing pulse strikes this patch and is directed into the optical train of the telescope, is collimated by the primary mirror, and continues on to the lunar surface. A local corner cube attached to the secondary mirror returns through the telescope optics a small amount of the outgoing light, called the fiducial return. It is desired that the fiducial return be as analogous as possible to an actual lunar return, and so its intensity is reduced to approximately 1 photon per pulse by the low transmit rate of the reflective patch on the rotating optic (which has effectively not moved in the tens of nanoseconds it takes the fiducial pulse to return) and some amount of additional neutral density, which can be adjusted as needed. The fiducial pulse also passes through a diffuser to distribute it over the area of the detector elements, again for fidelity with the lunar returns.

The APOLLO detector is a 4-by-4 array of single-photon avalanche photodiode elements, each 40 microns in diameter with 100 microns separating the element centers. Detectors of this type are active only when a potential exceeding their breakdown voltage is applied across the elements; however applying such a voltage continuously may result in thermal damage, and so the potential is held approximately 1 V below the breakdown level until either fiducial or lunar returns are expected, at which point a 'gate' lasting some 200 ns raises it about 5 V above the breakdown level. Photons striking a detector element initiate an electron 'avalanche' by which detections are made. Once initiated, such an avalanche is continuous until quenched by the end of the gate, and so only one detection is

possible per element per gate. This creates the potential for 'first-photon bias', in which a strong return will be detected only at its leading edge, biasing the detection toward early times. APOLLO's use of a 16-element array is invaluable in this regard, as it permits discernment of a strong return from a weak one and thus correction for first-photon bias. The capability of detecting multiple return photons in a single shot is possessed by no other LLR experiment and is another enabler of APOLLO's leap in ranging precision.

The round-trip time to the lunar reflectors is approximately 2.5 seconds but is variable by about 20 percent overall and changes at the rate of about 1 μs/s. A polynomial prediction indicates the expected time of the return at the nanosecond level, permitting the detector to be gated on at the appropriate point. By the time the lunar returns arrive, the rotating optic has moved the reflective patch out of the optical path, and so the lunar returns are able to pass to the detector without undue attenuation. Average rates of one photon per shot are seen but are quite high; 0.1 photons per shot is more typical. Each laser pulse initially contains some 10^{17} photons, but each phase of the journey to and from the moon is characterized by huge losses. Atmospheric seeing at Apache Point is regularly at the arcsecond level, but even so the resulting divergence of the beam means that the footprint of the laser on the lunar surface is a few kilometers in diameter, meaning that relatively little light strikes the targeted retroreflector, the largest of which (Apollo 15) is about the size of a suitcase. Diffraction by the corner cubes of the array imposes divergence on the downlink portion as well; overall, the return rate falls off as the fourth power of the earth-moon separation. The APOLLO system makes use of filters in the wavelength ($\sim$1 nm passband at 532 nm), spatial ($\sim$2 square arcsecond detector field of view), and temporal ($\sim$1 ns foreknowledge of expected return time) to discern the weak lunar return signal from the background.

This is especially challenging when the moon is near full phase, due to the higher background level and an additional factor of 10 degradation of the return rate that is believed to stem from temperature gradients within the corner cubes due to solar heating of dust on their front surfaces.

Once a lunar or fiducial photon initiates an avalanche in a detector element, the resulting current is translated to an ECL-level voltage signal by element-specific electronics on a custom board and passed to a Philips Scientific time-to-digital converter (TDC), by which the detection is recorded. The window in which the TDC is able to record a detection has a duration of 102.4 ns and is coordinated with the gating of the detector array. The timing resolution of most aspects of the detection process is dictated by the 20 ns resolution of a highly stable 50 MHz clock, but the TDC resolution is much finer at 25 ps per bin over 4096 timing bins, corresponding to about 4 mm resolution in two-way range. The extraction of millimeter-level measurements from a system whose fundamental resolution is somewhat larger is accomplished by fitting to a distribution of the photons accumulated during a run. This is not a trivial matter, but it is not a major subject of this work.

APOLLO receives six to eight approximately one-hour sessions per lunation, of which some fraction are lost to poor weather. During a typical ranging session, several circuits are made around the reflectors, and returns from at least three of the five are generally received, with ranging to the Soviet Lunokhod rovers being much more dependent on lunar phase than is the case with the Apollo program arrays. Ranging to a single reflector is called a ‘run’ and lasts several minutes, in the course of which returns of a few hundred photons are typical. This large volume of raw data is not usable by analysis programs. Instead, photons from a run are aggregated and their distribution fit to produce a single ‘normal point’ for

the run, consisting of a single exact timestamp (an even multiple of five seconds, not the actual launch time of any photon), associated two-way range in seconds, and statistical uncertainty in that range.

APOLLO normal points are fundamentally differential, representing a difference between the timing of the fiducial and lunar returns in the course of a run. The identical processing of the fiducial and lunar returns eliminates most systematic effects associated with our instrumentation. The breadth of the fiducial distribution is nevertheless smaller than that of the lunar distribution due to the aforementioned additional spreading imposed by retroreflector tilt. Indeed, returns from the larger Apollo 15 array are visibly more spread than those from the smaller Apollo 11 and 14 arrays due to this effect.

Chapter 2

Detector characterization

2.1 Description

APOLLO uses as its detector a 4-by-4 array of single-photon avalanche photodiode elements, generally called the APD within APOLLO but often known as a single-photon avalanche diode or SPAD in detector-physics circles. Devices like these have some characteristic breakdown voltage, which is about 25 V for the detector currently in use at Apache Point Observatory and for others provided to APOLLO by MIT Lincoln Laboratory. The specific breakdown voltage must be determined for each device. When a potential exceeding the breakdown voltage is applied across the detector elements, an electric field arises in a $p+$ multiplier region buried about 1 μm beneath the surface of the element, which field is sufficiently strong that an electron liberated by an incident photon in this region is accelerated and strikes atoms in the lattice, dislodging additional electrons. In this way an avalanche current rises from which a detection of the photon can be made. This current continues until the avalanche is quenched in some way. So-called 'active' and 'passive' quenching schemes are discussed in [22]. APOLLO uses a gated

quenching method, in which the voltage across the APD is kept just below the breakdown level until a photon from either the local corner cube or the moon is expected, at which point a 'gate' with a duration of several hundred nanoseconds is initiated and the APD voltage rises to several volts above breakdown. During this period, incident photons may result in detections in one or more of the detector elements. When the gate ends, the APD voltage again falls below breakdown, such that the electric field in the multiplier region is no longer strong enough to sustain the avalanche, which therefore ceases.

The avalanche produces a current in the device of a few hundred microamps. APOLLO has gone through two major versions of its APD readout electronics, which are described in Chapter 3. In both versions, the avalanche current produces a change of a few hundred millivolts in a potential that is fed into one side of a comparator. The comparator's other input is a reference voltage close to that of the APD side but which does not change in response to an avalanche. The avalanche signal, then, causes the comparator's ECL-level output to flip, and this signal is directed to the appropriate channel of the time-to-digital converter, producing a detection. The fact that there is inevitably some amount of noise on both the reference and APD signals at the comparator leads to some jitter in the time at which one level becomes higher than the other, and by extension in the time of the detection. This is a source of some tens of picoseconds in the overall error budget, which can be minimized in principle by maximizing the slope of the APD signal at the point at which it crosses the reference voltage.

A comparable source of uncertainty is related to the lateral spread of an avalanche in the detector element. The avalanche current is proportional to the area of the detector element that is avalanching at any particular time. The avalanching area expands linearly in the lateral direction, and so the time evolution of the

avalanche current is quadratic until the avalanching region impinges upon the edge of the detector. Thus, photons striking the center of the element will produce current sufficient to cause a detection faster than those striking near the element's edge. In practice when a detection is made there is no way to know where on the element the precipitating photon arrived, and so there arises a 'spatial jitter' reflecting this uncertainty. This effect is minimized by reducing the difference between the APD-side baseline voltage at the comparator and the reference, so that as small an avalanching area as possible is needed to produce a detection. This has the effect of increasing the area of the detector element in which an originating avalanche will still be expanding quadratically when a detection occurs. However, reducing this threshold voltage will at some point result in spurious events due to noise, or indeed a complete loss of sensitivity if the threshold impinges on a transient preceding the timing window.

2.2 Breakdown characterization

A detector's breakdown voltage, at which it becomes capable of sustaining an avalanche, must be known in order to avoid biasing it above this level when no gate is applied and to obtain the desired degree of excess voltage when the gate is present. The detectors characterized by APOLLO all have breakdown voltages between approximately 25 and 30 V, but there is variation within this range. Breakdown is characterized by initially setting the bias voltage applied to the APD anode at a level well below the plausible breakdown range and initiating a stream of gates of known amplitude. The APD signal is then monitored at the appropriate point on the comparator using an oscilloscope while the APD bias voltage is slowly increased. At some point avalanches will begin to appear on

the scope, and the detector breakdown voltage is the APD bias voltage at that point, plus the gate amplitude. Verification of this value by characterizing it on multiple channels yields agreement within a tenth of a volt. Results of breakdown characterization for the detectors possessed by APOLLO are presented in Table 2.1.

Table 2.1: Breakdown voltages for the APOLLO suite of avalanche photodiode detectors.

Wafer number	Guard Contact?	Element Diameter (μm)	Breakdown (V)
3	No	30	33.27
3	Yes	30	$>$ system limit
6	Yes	30	28.62
6	No	30	28.85
10	Yes	30	27.38
10	No	30	27.37
12	Yes	20	27.89
12	No	20	27.72
12	Yes	30	27.40
12	No	30	27.43
12	Yes	40	27.18
12	No	40	27.05
Original 20-micron			24.6
Original 30-micron			24.4

2.3 Crosstalk characterization

The use in the APOLLO experiment of a 16-element avalanche photodiode is one of its great strengths, permitting detection of multiple photons from a single shot. This both allows good observing conditions to be used to greatest advantage and provides quality information about the signal rate when it is above one photon per shot, which is needed for the characterization of first-photon bias. However, the combination of multiple detector elements as an array makes it vulnerable to crosstalk, in which a photon generated by an avalanche in one element triggers an

avalanche in another element of the detector. This produces a false detection and renders the receiving element incapable of further detections until the avalanche has been quenched. We characterized the extent of this phenomenon in a suite of detectors provided by Lincoln Laboratory as well as its dependence on various properties of the detectors themselves and of the detector environment.

Crosstalk in an SPAD array arises as a byproduct of the photon-detection process [6, 7, 8, 9]. A photon striking the detector has some probability of creating an electron-hole pair. In a device designed to favor electron-initiated avalanches, the photoelectron will drift to a high-field region, where it acquires sufficient energy to impact-ionize other atoms in the lattice, generating the current 'avalanche' by which events are detected.

Once an avalanche is underway in a detector element, it persists until quenched; meanwhile, further detections by that element are impossible. The avalanching element emits some radiation, [10, 11, 12, 13], though inefficiently, and photons with energies close to the band gap may penetrate the silicon between array elements to initiate an avalanche in a neighboring element, or even one farther afield [14]. Clearly this poses a significant challenge for photon-counting applications, as there is no way to distinguish qualitatively between events arising from signal, thermally generated ('dark') events, and crosstalk. As a result, the temporal character and frequency of crosstalk events must be characterized prior to deployment, and considerable effort has been directed at the production of SPAD detectors and attendant electronics with favorable crosstalk properties [15, 16, 17].

2.4 Experimental Setup

We characterized a set of 10 detectors, described here and fabricated at MIT Lincoln Laboratory's microelectronics facility [20, 21]. The devices differed in doping profile, size of the active region, and the presence or absence of a guard contact. We here also briefly describe the electronics used to note the times of detections, as well as other apparatus used in the characterization.

2.4.1 Detector structure

The SPADs characterized in this work were fabricated on six-inch silicon wafers. The silicon substrate is heavily p-doped (10^{18} boron atoms/cm^3) with a lightly p-doped (10^{14} boron atoms/cm^3) epitaxial layer grown on top. The diode is fabricated by ion implantation of p-type (boron) dopants to form a $p-i-p-i-n$ structure (the i layers are actually lightly p-doped; n layers are doped with arsenic). A cross section of the design is shown in Fig. 2.1. The lower i layer, referred to as the absorber, is where most of the photons are absorbed. When reverse-biased at the proper operating voltage, a modest electric field (10^4 V/cm) in the absorber causes the photoelectrons to drift into the upper i layer. The upper i layer, referred to as the multiplier, has a much stronger field (several times 10^5 V/cm), sufficient to cause impact ionization that initiates an avalanche. The photoelectron and the secondary electrons are collected at the top n layer, and the photo-hole and secondary holes are collected at the substrate.

Note that the $n+$ doped region, which defines the extent of the junction, has a larger diameter than the $p+$ implant that separates the absorber from the multiplier. This creates a simple $p-i-n$ structure around the periphery of the APD, where the electric field is much weaker than in the multiplier region. This

peripheral diode serves as a 'guard ring' that performs two functions. First, it tailors the electric field profile so that avalanche breakdown occurs in the central portion of the diode, not at the periphery. Second, it collects electrons generated outside the absorber region, preventing them from initiating avalanches. This minimizes the volume from which dark current is collected, and therefore minimizes the dark count rate. The price paid is that that it also limits the fraction of the chip area that is light-sensitive. Indeed, the depletion region of the guard ring diode encroaches on the absorber, so that the active volume of the absorber has a champagne-glass shape.

The n side of the detector is electrically contacted by etching a contact annulus through the passivating oxide and patterning a ring of metal. The two design variations investigated in this work differ primarily in the extent of the guard contact structure. In the guard-contact (GC) variation, the $n+$ implant extends 9 μm beyond the edge of the $p+$ implant, as shown in Fig. 2.1, whereas in the non-guard-contact (NGC) variation, this spacing is only 5 μm. A p-side contact common to all detector elements is made on the back side of the substrate. Devices tested here came from four wafers; the fabrication process differed from wafer to wafer only in the dose of the p implant. Wafer 3 has 2.6×10^{12} boron atoms/cm^2; wafer 6, 2.7×10^{12}; wafer 10, 2.9×10^{12}; and wafer 12, 3.0×10^{12}. The doping profile is a factor in determining the breakdown voltage, speed and efficiency of photoelectron collection, and avalanche initiation probability.

2.4.2 Equipment and Data Collection

The detectors we have tested are 4-by-4 arrays of circular SPADs with a pitch of 100 μm. We were furnished with 12 such arrays fabricated from four different wafers in matching pairs, one of the GC type and one not. We have two

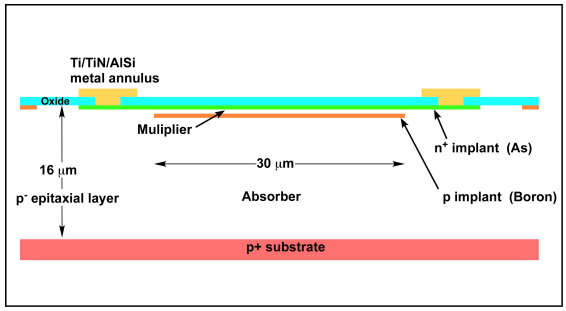

Figure 2.1: Cross sections of a SPAD element with the guard-contact geometry. The drawing is to scale. On the surface is an annular contact made by cutting through the oxide (light blue) and patterning a metal ring. The wider guard contact characteristic of the GC geometry tends to be more effective at collecting dark current generated outside. On the GC device, the arsenic dopant (green) extends beyond the buried boron implant (orange) by 9 μm, whereas this is only 5 μm on the non-GC device. The material is 180 ohm-cm epitaxial layer of thickness 16 μm grown on a $p+$ substrate.

arrays with 30-μm element diameters from each wafer, and then an additional two 20- and two 40-μm arrays from wafer 12. One device—the 30-μm GC device from wafer 3—has a breakdown voltage exceeding what our setup can apply. Another, the 40-μm NGC device, was unavailable for testing, so we have data from 10 arrays. The detectors are mounted in 40-pin dual-inline packages that are inserted into a printed circuit board of our design. Electrically, each detector element has a dedicated cathode, while the anode is common to all elements. The circuit board applies a steady negative voltage to the anode of the detector the magnitude of which is about one volt below the breakdown of the device being characterized. The breakdown voltages of our detectors are all approximately 25 V, but the specific value must be characterized for each.

Depending on the method used to quench avalanching detector elements, continuously reverse-biasing an SPAD above its breakdown voltage may result in thermal damage. Several different active and passive quenching schemes have been proposed and are in use depending on the application[22, 23, 24]. In this

work we have adopted the gated approach, in which the detector is biased below breakdown until the onset of a timed gate, at which point the anode voltage drops by a controllable amount such that its magnitude exceeds the detector's breakdown, putting the element in Geiger mode. The gate lasts for several hundred nanoseconds, and at its end the detector bias magnitude is reduced below breakdown once more. Any avalanches that may be ongoing in its elements are thus quenched, rendering those elements capable of making detections during the next gate. We prefer gated quenching to the other methods for crosstalk characterization, as both passive and active quenching terminate any avalanches soon after they begin, in order to render the element sensitive to additional photons[25, 26, 1, 27]. For our purposes, it is desired to let avalanches persist in order to gauge any increased detection rate in nearby elements attributable to crosstalk during that time. Gates were produced at a rate of 1 kHz for the tests described here. The 1 ms between gates is sufficiently long that afterpulsing due to incomplete quenching of an avalanche should not occur.

The readout of each detector element is governed by dedicated circuitry. We have the capability to electrically disconnect the cathode of each element individually, which prevents avalanches in disconnected elements and permits us to isolate a pair whose mutual rates of crosstalk we wish to determine. The avalanche current arising in an element is converted via preamplification to a voltage signal with a few-ns rise time and few-hundred mV amplitude. The excess voltage—the amount by which the applied voltage exceeds the breakdown voltage—contributes approximately linearly to the strength of the avalanche current, and thus to the amplitude of the voltage signal. Detector-specific characteristics also play a role in determining this amplitude. This signal crosses the level of a reference voltage as measured at a comparator. The comparator directs an ECL-level signal to a

Phillips 7186H 16-channel time-to-digital converter (TDC). We operate the TDC using 4096 25 ps bins, thus producing a 100 ns timing window which lies completely within the somewhat-longer bias gate.

2.5 Determination of Crosstalk Rates

We quantified the crosstalk rates of our SPAD arrays via two approaches, either of which may be appropriate depending on the resources of time and equipment available. In both cases, two elements of the array must be isolated, such that all other elements are prevented from avalanching.

2.5.1 Laser-illumination Approach

Our favored method is to use a narrow-pulse laser to illuminate one element of the array, and then observe the incipience of crosstalk on another element in the wake of the laser fire. This approach requires possession of a short-pulse laser at an appropriate wavelength. We employ a 1064-nm fiber laser producing 6 ps FWHM pulses at 50 MHz, frequency-doubled to 532 nm. Residual infrared light is separated by a prism. An electro-optic intensity modulator negates by interference all but one in 5×10^4 pulses, so that the effective laser-fire rate matches the 1 kHz detector-gating rate and only one unmodulated pulse reaches the array in the course of a timing window. This concentrates the progenitors of subsequent crosstalk events in a narrow time band and endows the temporal rise of crosstalk with a simple exponential form (Fig. 2.2). The component we use to drive the modulator takes input from the detector gating electronics, allowing us to pick out of the train a pulse arriving early in the timing window. The gating and detection circuitry must be synchronized with the laser fire and combined with

an optical setup capable of focusing the beam on a single element of the array. When characterizing crosstalk in this way we focus the beam to a spot of a few μm FWHM, thereby ensuring that the laser pulse is seen minimally if at all by neighboring elements.

It is observed that following the arrival of the laser pulse the crosstalk rate in nearby elements builds up to a steady-state value over a device-dependent period typically of several nanoseconds, short compared to the 100-ns timing window. The rate can be computed by comparing the detection rate in later bins, for which the asymptotic rate has been realized, to that in bins that precede the laser fire. The apparent number of events attributable to crosstalk must then be divided by the number of detections associated with the laser peak on the illuminated channel to get the probability of a crosstalk event conditional on the presence of a progenitor. Only the rate of crosstalk from the illuminated element to the receiving one is measured.

The fact that the considerable majority of events on the emitting channel occur at a specific point in the timing window permits a clear contrast between detection rates on the receiving channel before and after this event. As a result, despite the greater technical challenges, this approach has the advantage of speed. Thirty minutes of operation at 1 kHz is sufficient to constrain a rate in the tens of kHz to within 10 percent. Furthermore, whereas the alternative steady-illumination approach requires some level of signal in both elements to be effective, in the laser-illumination case the background rates can and should be made as low as possible, which blunts the effect of blocking by prepulses, as discussed in Section 2.6.

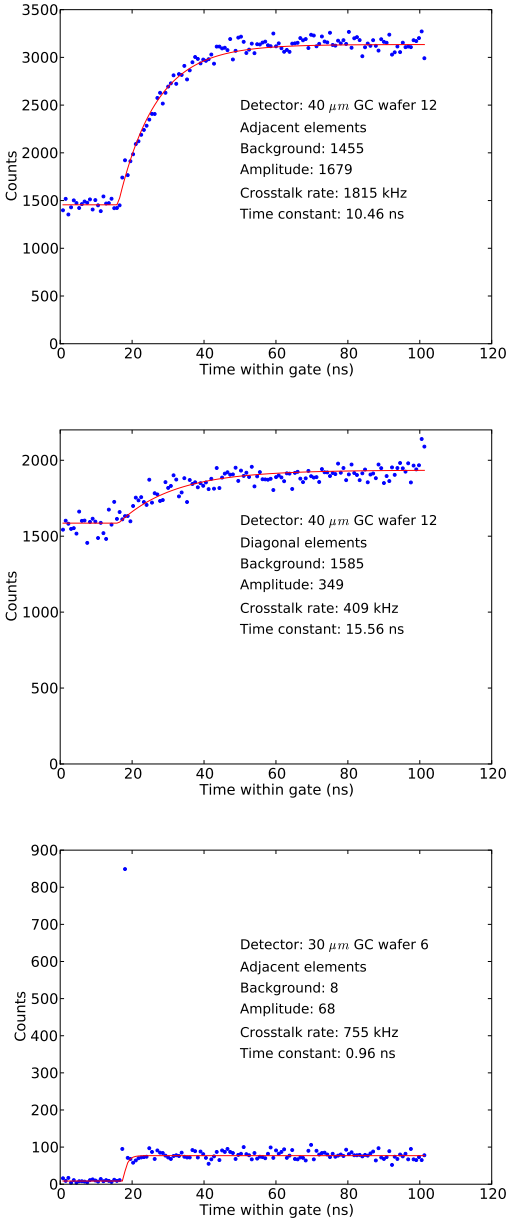

Figure 2.2: Distributions and fits to data collected by illuminating another element of the array with a pulsed laser. Robust fits are possible after a comparatively short period of data collection, even for crosstalk rates below 10^5 Hz. In the wafer 6 data (bottom), note the single high point; this scattered light from the laser fire nicely separates the periods with and without crosstalk events. Direct comparison of these figures is not possible, as data from the laser-illuminated channel is additionally needed to extract the crosstalk rates.

2.5.2 Steady-illumination Approach

A crosstalk rate can also be determined from correlations between the detection times of two elements operating in the presence of a low level of uniform illumination (or no illumination, if the detectors' dark rate is high enough to permit the accumulation of sufficient data in a reasonable amount of time). Higher illumination levels increase the rate at which crosstalk events accumulate, but the crosstalk rates obtained will be biased to a greater extent by prepulse-blocking, discussed later. The technique works as follows. For every gate in which both elements record a detection, we note the difference in TDC bins between the timing of the two avalanches, from -4095 to 4095. In the absence of crosstalk between the elements, a histogram of these differences will have a triangular shape; the timing of the events in the two detectors are uncorrelated, and there are 4096 combinations of timestamps that produce a difference of 0 (bins 0 and 0, 1 and 1, etc.) but only one combination that produces a difference of -4095 (0 and 4095). If detector A has a dark count rate of d_A, the probability that it will fire in a give time bin of width δt is $d_A \delta t$. The probability that the detectors will both fire in a particular time bin is $(d_A \delta t)(d_B \delta t)$. If the gate duration is T, there are $T/\delta t$ time bins in which such a coincidence can occur. If N gates are used to assemble a histogram of the time delay, therefore, its peak value, which occurs at a time delay of zero, is $N(T/\delta t)(d_A \delta t)(d_B \delta t)$. Fig. 2.3 represents data from a simulation of two elements making detections with no crosstalk between them.

In the presence of crosstalk, however, the distribution assumes a different form. Assume for a moment that when element A avalanches, it *instantaneously* causes the count rate of element B to be elevated from its dark count rate d_B to a higher rate $d_B + C_{A \to B}$. Conversely, when element B fires, it instantaneously causes the count rate of element A to be elevated from its dark count rate d_A

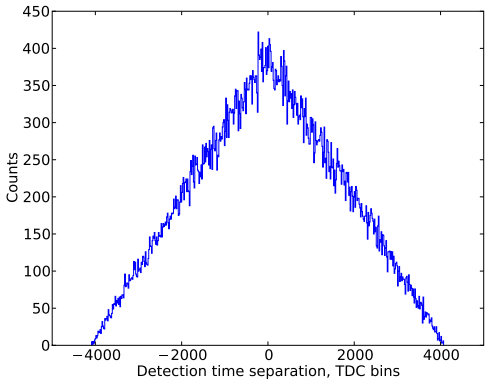

Figure 2.3: Distribution histogram of event-time differentials between two elements in the absence of crosstalk, constructed from simulated data.

to a higher rate $d_A + C_{B\to A}$. Consider a large collection of N gates. As in the case with no crosstalk, one would expect a triangular histogram of the time delay, except that the positive-delay half of the distribution (A fires first) approaches a zero-delay value corresponding to an elevated count rate in detector B: $N(T/\delta t)(d_A\delta t)[(d_B + C_{A\to B})\delta t]$. The negative delay half of the distribution (B fires first) approaches a zero-delay value corresponding to an elevated count rate in element A: $N(T/\delta t)(d_B\delta t)[(d_A + C_{B\to A})\delta t]$.

The observed distribution, however, shows a pronounced dip centered at zero delay due to the fact that crosstalk events cannot occur immediately following their progenitors due to the avalanche rise time. Therefore, after element A avalanches, the count rate of element B rises from d_B to $d_B + C_{A\to B}$ over a time period of several nanoseconds. Thus the zero-delay value of the distribution is not appreciably different from what it would be in the absence of crosstalk. The wings of the distribution, however, are straight lines corresponding to the steady-state elevated count rates and can be extrapolated to zero delay to infer the crosstalk rates, as shown by the dashed lines in Fig. 2.4. That is, the positive-delay wing of the distribution should extrapolate to a zero-delay value of $N(T/\delta t)(d_A\delta t)[(d_B +$

$C_{A\to B})\delta t]$.

This approach has the advantage of not requiring significant equipment other than the SPAD array itself and the circuitry required to quench and read out the detector elements. However, collecting sufficient data to accurately gauge a crosstalk rate of under 100 kHz requires overnight data collection for gating rates not significantly above 1 kHz. Furthermore, the necessity of fitting to irregularly shaped distributions introduces an element of complexity, and makes it difficult to determine a sensible crosstalk rate for low-crosstalk cases in a reasonable amount of time. Fits to simulated data show that this method can reliably determine crosstalk rates and rise times (Fig. 2.4). Additionally, one simultaneously derives the rate of crosstalk from element A to element B and from B to A—a convenient check on the results. The ease and usefulness of such a fit to real data depends on the crosstalk rate (Fig. 2.5), although using a numerical approach discussed presently can discern small rates that elude fitting software.

2.5.3 Numerical Extraction of the Crosstalk Rate

In describing each experimental approach we have had recourse to histograms of detection times so obtained and have extracted the crosstalk rate via fitting to these distributions. There is nothing wrong with a graphical approach in principle, but the possibility of poor fits and failure to account for certain effects in specifying a model can make it less attractive in practice. In the laser-illumination case, for example, we fit a flat line to all the bins prior to the laser event and second flat line to all the bins after, accounting for rise time, and determine the crosstalk rate from the difference between the levels along with information from the emitting channel. It is not clear, however, that a constant rate is a good model for the signal level after the laser event. In general, even if the underlying probability

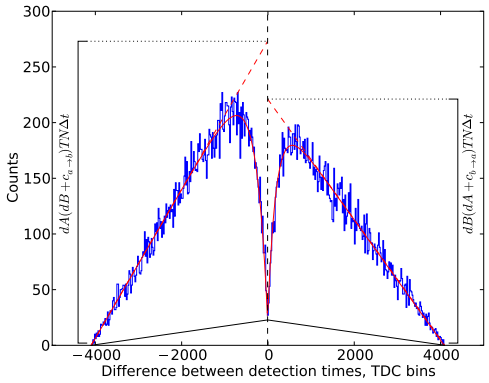

Figure 2.4: Simulated data to which a fit has been made, showing best-fit curve and underlying triangular distribution from randomly distributed events. The asymptotic portion of the crosstalk-generated distribution can be extrapolated to the peak, and—in the case of real data—the crosstalk rate extracted from the parameters as shown. Note that the complementary rates $c_{A\to B}$ and $c_{B\to A}$ may differ and are both determined from the fit. This data was simulated with input crosstalk rates of 1000 kHz on the left and 800 kHz on the right, and rise times of 300 25-ps TDC bins on the left and 200 bins on the right. The fit output determined 996 ± 9 kHz and 800 ± 7 kHz for the crosstalk rates, and 275 ± 11 bins and 197 ± 10 bins for the rise times.

of a detection is in fact constant, the observed rate will decline throughout the gate due to first-photon bias, only one detection being possible in the element per gate. At the same time, the probability of a crosstalk event occurring on the receiving element is actually increasing over this period because not all events on the emitting channel are confined to the main laser pulse; photoelectrons created beneath the depletion region may migrate randomly into it and initiate avalanches tens of nanoseconds after the pulse arrives, and background events continue. As a result the number of potential progenitors slowly but steadily increases.

In view of these vagaries we also present a non-graphical approach to the crosstalk rate that is suitable for data obtained by both methods, with minor modifications. One element is designated the receiving element, which we call element B, while the emitting element is A. In the case of laser illumination the element so illuminated is A, whereas in the case of steady illumination the choice is

arbitrary and reversible. We wish to determine the rate of crosstalk events induced in B by prior events in A. Such events will appear in the data as instances in which both A and B made a detection in the same gate, and the detection on A preceded the detection on B. Within each timing bin, therefore, we count the number of events on B (occurring in that timing bin) that followed an event on A (earlier in the timing window). These 'double events' are potentially crosstalk events; however there is some probability that such a circumstance will arise by chance even in the absence of crosstalk. The number of expected coincidental double events the later of whose detections falls in a given timing bin is given by $I_B(\delta_t/T)I_A f_P N$, where I_A and I_B are the proportion of gates in which an event occurred on each of the two channels, δ_t is the duration of a timing bin, T is the duration of the timing window, f_P is the proportion of the timing window that has elapsed prior to the timing bin being considered, and N is the total number of gates over the course of which the data was generated. It should be noted that this presumes a constant detection probability over the timing window, which presumption becomes more valid for small I_A and I_B due to increased first-photon bias at higher illumination rates. The number of expected coincidences is subtracted from the total number of double events in the bin, Poisson uncertainties being assigned to each number and propagated.

In order to determine a rate, the putative number of crosstalk events in the bin must be divided by the number of potential precursor events on element A and by the width of the timing bin in seconds. The number of potential precursors may be approximated by $I_A f_P N$ with some suitable uncertainty or by counting events in prior bins, in which case no uncertainty need be ascribed.

A final crosstalk rate for the bin in question has now been determined. In the limit of low crosstalk, in which an event on element A is unlikely to result in a

crosstalk event on element B within the timing window, this rate should be the same for all bins, apart from statistical variation. The outlined procedure is therefore repeated for all TDC bins, with the caveat that the Poisson noise is proportionally very large for those bins in which the expected number of crosstalk events is small and so they may be better excluded from this analysis. These would be bins preceding the laser pulse in the laser-illumination case and perhaps the earliest third of bins in the steady-illumination case. The results for all considered bins can then be averaged and their uncertainties propagated accordingly to produce a final estimate and uncertainty for the crosstalk rate. In practice, we find that nominally identical pairs of elements on the same array differ in crosstalk susceptibility at approximately the 15 percent level, so an uncertainty of that scale may be appropriate when making comparisons between elements or detectors.

2.6 Corrections Due to Prior Events

A tacit assumption of both the laser-illumination and steady-illumination experimental approaches is that both elements are capable of making detections during the timing window, but the validity of this view is dependent on the background rate. It is desirable in gated operation to have the timing window begin somewhat after the detector is first biased above breakdown in order to achieve a stable detection rate over the whole timing period and to avoid transient effects associated with gate turn-on. As a result there is some probability that an element is already avalanching, undetected by our system, when the timing window begins. This leads us to underestimate the crosstalk rate. During the timing window, when one element of a pair avalanches we are then looking at the other element to see if it will experience a crosstalk event, and the rate we ascribe to this process

is at some level a ratio of the number of times this is observed to occur to the total number of times it had the opportunity to occur. If the second element is already avalanching, such an opportunity is not actually present, so we inflate the denominator in proportion to the fraction of the time such a condition exists.

It is no trouble to attempt to correct for this by moving the timing window to cover gate turn-on and finding the frequency of events prior to the usual timing start, but this turns out to not be sufficient. Prior to gate turn-on the detector is, in our mode of operation, biased below breakdown for a period of time that is long compared to the actual length of the gate. During this time, the background illumination level liberates electrons in the detector elements, but these do not initiate avalanches due to insufficient electric field strength, and on some timescale they recombine or migrate out of the active volume. If such an electron is present at gate turn-on, however, it will cause a prompt avalanche which will probably not be detected even if the timing window is suitably positioned, as the transient effects of gate turn-on render the detection electronics insensitive at that instant. In principle therefore the crosstalk rate is only correctly determined when the illumination level is as low as possible. Deriving crosstalk rates at different levels of illumination can give us some idea of the level at which prepulses cease to be a significant consideration.

Avalanches occurring promptly at detector turn-on or otherwise before the beginning of the timing window also create a second effect that leads to underestimation of the crosstalk rate, particularly when the steady-illumination approach is used, and perversely in proportion to the crosstalk rate itself. Let us say we want to determine the crosstalk rate from element A to element B. As previously described, this requires determining the probable number of crosstalk events observed on element B and then dividing by the number of events observed

at earlier times within a gate on element A. However, not every event observed earlier in any gate on channel A is a potential progenitor; some are themselves crosstalk events whose own progenitor is a preceding event on element B, which may have occurred before the start of the timing window. When the probability of any particular event spawning a crosstalk event within the 100-ns timing window is small (say, less than 5 percent, corresponding to a crosstalk rate less than 500 kHz), the number of crosstalk events 'masquerading' as potential progenitors is correspondingly small and is subject to a proportionate quantity of neglect. For higher crosstalk rates, however, an increasingly significant fraction of the events identified as potential progenitors are in fact not, leading to proportional underestimation of the crosstalk rate. This effect also produces underestimation by another mechanism, in that supposed background rates are used as described above to compute the number of doubles observed in a bin which are likely to be coincidental rather than crosstalk. If the background rates have included a number of events actually attributable to crosstalk, this calculation will produce an estimated coincidence rate that is too high, in proportion in fact to the square of the fraction by which the background rate has been overestimated.

Such masquerading crosstalk events come in two varieties: those whose progenitor occurred prior to the start of the timing window and those whose progenitor occurred during the timing window. Unfortunately, as noted it is difficult to estimate precisely the number of events that occur promptly at detector turn-on. Moreover, as events prior to the timing window are the cause through different mechanisms of both the masquerading-crosstalk issue and the direct-blocking phenomenon previously discussed, both of which lead to underestimation of the calculated crosstalk rate, it is not possible to draw a direct line between the degree of underestimation of the rate (if the 'true' rate is known by means

of low-background laser-illumination results, which are largely immune to both effects) and the prior fraction. However, with the additional measurement of the illumination level on each channel individually (under identical conditions but with the second element switched off), the several effects can be sussed out and underestimated rates corrected.

We first want to find an expression for the prior fraction, the (unknown) proportion of gates in which a channel avalanches prior to the beginning of the timing window, denoted by P. The observed illumination fraction, the proportion of gates in which the channel was observed to record an event while both channels were operating, is denoted by I_O. Then turning off the second channel we get a 'true' illumination level for the remaining channel that excludes all crosstalk events and is therefore lower, I_T. The events removed in this way can be said to constitute an illumination rate due to crosstalk, I_C, so that $I_O = I_T + I_C$. Of the events that make up I_C, some have progenitors which also occur during the timing window, while some have prior events as progenitors. These two types cause detection rates which we label $I_{C_{TW}}$ and I_{C_P}, so that $I_C = I_{C_{TW}} + I_{C_P}$. Because all progenitors of I_{C_P} occur by definition before the start of the timing window, we assume the events of I_{C_P} are distributed uniformly over its duration. We therefore have $I_{C_P} = PCT$, where C is the crosstalk rate and T is the timing window duration. In contrast, the rate of events of which $I_{C_{TW}}$ is composed rises linearly over the timing span because potential progenitors are accumulating in a correspondingly linear fashion as time goes on. The average rate therefore is equal to the rate at the center of the timing window and we have $I_{C_{TW}} = \frac{1}{2} I_T C T$. We can then write $I_O - I_T = TC(P + \frac{1}{2} I_T)$, and so

$$P = \frac{I_O - I_T}{TC} - \frac{1}{2} I_T.$$

Evidently the prior fraction can be expressed in terms of the combined single-

channel and double-channel illumination rates, as well as the true crosstalk rate C.

However, we have yet to show that C itself can be determined from the information available. Both the numerical and graphical methods for analysis of steady-illumination data produce a calculated crosstalk rate C_C that we know underestimates C, but we want to know by how much. Once I_T has been determined from single-channel operation, it must be incorporated into the analysis pipeline to correct the number of expected 'coincidental' double events which have been subtracted from the total number of double events associated with each bin to yield a number of doubles attributable to crosstalk. As noted the number of coincidences in a bin is proportional to the square of the illumination rate, and so it is subject to a multiplicative correction of $(\frac{I_T}{I_O})^2$. The calculated crosstalk rate we denote by C_C has taken this correction into account already.

As previously described, we expect blocking by priors to cause C_C to understate C by a factor of $(1 - P)$. Masquerading crosstalk events require an additional correction factor which we must now determine. Recall that the crosstalk rate is determined by dividing a number of apparent crosstalk events by the number of its potential progenitors. We have now seen that some of these potential progenitors are in fact themselves crosstalk events and are not able to be the progenitor of subsequent crosstalk themselves; in effect we divided by I_O when we should have divided by I_T, and must now multiply by $\frac{I_O}{I_T}$. In sum, C_C is related to C by $C_C = C(1 - P)\frac{I_T}{I_O}$. We already derived an expression for P, so we have $C_C = C(1 - \frac{I_O - I_T}{TC} + \frac{1}{2}I_T)\frac{I_T}{I_O}$ and, rearranging,

$$C = \frac{1}{1 + \frac{1}{2}I_T}(\frac{C_C I_O}{I_T} + \frac{I_O - I_T}{T}).$$

In this way the true crosstalk rate can be estimated from the crosstalk rate calculated in two-channel operation combined with illumination data obtained in single-channel operation.

For both laser-illumination and steady-illumination experiments, the effect of direct blocking by prior events is related to the background rate and can be minimized by minimizing this rate. In laser mode the background rate is a nuisance that may well be reduced for the sake of reducing it. In steady-illumination mode the background rate is essential to the experiment. However, the data consists of a signal of double events attributable to crosstalk, the rate of which is proportional to the background rate, imposed on noise consisting of coincidental double events, the rate of which is proportional to the square of the background rate. In principle therefore the signal-to-noise ratio actually increases as the background rate goes down. A significant number of crosstalk events are needed to afford reasonable levels of precision, but presuming that P is not more than a factor of a few greater than I_O, setting the background rate so that $I_O \approx 1$ percent will firmly constrain a crosstalk rate as low as tens of kHz in the course of an overnight run with a gating frequency of 1 kHz.

The impact of masquerading crosstalk events depends on the ratio of I_O to P, which is plausibly independent of the actual illumination level in the steady-illumination case. Therefore this effect cannot necessarily be eliminated or even estimated by varying I_O, and recourse to the results of single-channel runs must be had as described. The laser-illumination approach circumvents the issue by breaking the coupling between the illumination rate during the timing window (due primarily to laser-induced events) and the prior fraction (Fig. 2.6). We have shown that crosstalk rates can be corrected to account for the several sources of error, but these methods are themselves subject to correction, for example when

the crosstalk rate is high enough that the occurrence of crosstalk events across the timing window cannot be presumed constant. We therefore view crosstalk rates extracted from laser-illumination experiments conducted at low ($<\sim$1 percent) background illumination levels as the most reliable.

2.7 Results

Using the laser-illumination method, we were able to determine crosstalk rates for our suite of SPAD arrays, comprising a range of doping profiles and element sizes: 20, 30, and 40 μm in diameter. The trends exposed by our work are presented here, and example crosstalk rates for our detectors at $V_{\text{ex}} = 4$ V are given in Table 2.2.

Table 2.2: Crosstalk rates between a pair of adjacent elements from our suite of 10 SPAD detectors, collected at $V_{\text{ex}} = 4$ V using the laser-illumination approach. Uncertainties are determined from propagated Poisson uncertainties on event counts.

Wafer number	Guard Contact?	Element Diameter (μm)	Crosstalk Rate (kHz)
3	No	30	395 ± 13
6	Yes	30	767 ± 12
6	No	30	980 ± 12
10	Yes	30	502 ± 12
10	No	30	522 ± 5
12	Yes	20	12.1 ± 0.3
12	No	20	14.6 ± 0.3
12	Yes	30	266 ± 5
12	No	30	304 ± 16
12	Yes	40	2011 ± 13

2.7.1 Distance Dependence

The decrease in the crosstalk rate as the separation between the emitting and receiving elements becomes greater is due to both the $1/r^2$ dependence of

radiation and absorption of crosstalk photons in the intervening silicon. As this absorption is wavelength-dependent, some sense of the emission spectrum is needed. Measurement of the spectrum of a device with a similar breakdown voltage to those we have characterized indicates a 3300 K blackbody spectrum, which peaks in the near-infrared wavelengths that principally contribute to crosstalk and is broad [19]. Therefore the approximation of a flat spectrum is appropriate.

The receiving element has a volume V and is a distance r away from the emitting element, where the dimensions of V are small compared to r. The emitting element radiates isotropically into the surrounding material. We must take into account the energy dependence of the absorption properties of the intervening silicon, which causes exponential extinction of the emitted radiation. This imposes a factor of $e^{-r\alpha}$, where α is the energy-dependent absorption length scale of the material. This α is known to be quadratic in the difference E between the photon and band gap energies [28]. Finally, a factor of α accounts for the likelihood of absorption by the receiving element itself. This probability is high for short absorption scales (large α) and low for long scales (small α).

Combining these several considerations, the rate at which crosstalk photons are absorbed by the receiving element per energy interval dE is given by

$$R_a \propto \alpha e^{-r\alpha} \times dE/(2\pi r^2).$$

To get the total absorption, this must be integrated from the band gap energy to infinity. The quadratic relationship between α and E permits us to change the variable of integration to α and integrate from zero to infinity, the material being transparent at the bandgap energy. This exponential integral is a common one and contributes a factor of $1/r^{1.5}$ once evaluated. Thus, the expected crosstalk rate is

proportional to $1/(r^{3.5})$.

The results of our detector characterization suggest a somewhat steeper decline of crosstalk with distance between elements. Fits to crosstalk rates derived from laser-illumination tests with several different combinations of detector and excess voltage produce a highly consistent picture of the distance dependence, implying a $1/r^{4.5}$ relationship (Fig. 2.7). This result appears to be robust but is not easy to explain. In principle a deviation of the emission spectrum from flat can have an effect on the relationship. Because the opacity of the material increases with photon energy, we expect the absorbed spectrum of relatively nearby elements to be bluer than that of more distant elements. We have conducted simulations that confirm this intuition, the results of which are represented by Fig. 2.8. Clearly, an apparent power law steeper than $1/r^{3.5}$ should be anticipated if the emission spectrum falls off with increasing wavelength in the relevant range, from about 800 nm to the bandgap. In order to refine this understanding, we performed simulations substituting for a flat spectrum one provided by [12]. When a power-law fit was made to the resulting simulated data, the modified spectrum pushed our estimate of the distance-dependence exponent to approximately -3.75, closer to our observations than the flat-spectrum expectation but still firmly in tension with them. Subsequent simulations of the array using different spectra indicate that a spectrum rising quadratically from 520 nm to 920 nm and then decreasing exponentially with a constant of 25 nm thereafter, as in Fig. 2.9, would be needed to produce results that comport with the relationship we observed, a steep decrease which we believe is firmly excluded as a real possibility.

In general, reflections of emitted photons off the internal surfaces of the detector can impact the distance-dependence of crosstalk, but in our front-illuminated devices the distance from the multiplier to the front interface is negligibly small

compared to the inter-element distance, whereas the back interface is so distant that no near-bandgap photon capable of propagating there and back would have the remotest chance of absorption within a detector element. Moreover, the effects of reflection tend relatively to enhance crosstalk at longer ranges, not diminish it [19]. In principle, it can be imagined that crosstalk would appear enhanced at shorter ranges due to the finite extent of the elements, which causes the nearest parts of two elements to be considerably closer together than their centers. If this were an effect capable of explaining the data, however, we would expect to see an attendant discrepancy in the power-law fits to data from 40-μm and 30-μm devices, whereas this is not observed.

In principle, migration of photoelectrons between detector elements constitutes crosstalk of a different sort that could scale differently with distance, but the distance an electron can diffuse during the 100-ns timing window is small compared to the pixel spacing.

2.7.2 Dependence on Element Size

Variation in the crosstalk rate in otherwise-similar detectors with different element size partially depends on the shape of the avalanching volume and how it changes with element diameter. If our initial presumption is that the volume of both the depletion region in the receiving element and the multiplier region in the emitting element grows as the element area, the crosstalk rate would be expected to grow as the fourth power of element diameter. However, the actual receptive volume of the APD is less than the nominal layout diameter because the depletion region from the guard contact diode encroaches on it, causing the sensitive volume to taper. This encroachment is proportionally more severe for the smallest-diameter device. As a result, the crosstalk rate should increase faster than

the fourth power of the nominal diameter. The stronger-than-expected dependence on element separation is possibly relevant here as well, as the nearest edges of larger elements are considerably closer than those of smaller elements for a given center-to-center distance, the effect being strongest for nearest neighbors.

Considering the three detectors in the wafer 12 guard-contact family, we find that the crosstalk rate is an extremely strong function of the element size, with crosstalk rates for the 40-μm device being over 100 times those of the 20-μm device at a range of excess voltages. The data in each case is well-fit by a power law, with the best fit at each voltage consistent with an exponent of 7 (Fig. 2.10).

2.7.3 Dependence on Excess Voltage

We expect that the crosstalk rate will depend on the excess voltage via several mechanisms. First, additional voltage expands the depletion region, increasing the volume in which a newly-created photoelectron may promptly initiate an avalanche. The expansion of this region stalls at the buried implant until the bias voltage reaches a certain level, at which the depletion region 'punches through' the implant. The detectors we tested were designed such that this punch-through voltage is close to the breakdown voltage, beyond which the depth of the depletion region has been shown in other devices (see Fig. 2.11) to grow roughly as the square root of the excess voltage on the scale of tens of volts of excess. However, the expansion is approximately linear for comparatively small excess voltages such as those used in this characterization.

Other factors combine to make it less clear to what extent we should anticipate crosstalk to grow with excess voltage. As this voltage grows, the corresponding stronger electric field increases the probability that any given photoelectron will succeed in triggering an avalanche. Higher voltages result in greater avalanche

currents and thus more emitted radiation; furthermore, the temperature of the constituent electrons is greater in the presence of the greater electric field, causing them to radiate more power into the surrounding material while also shifting the emitted spectrum, which we have previously assumed to be flat. The aggregate effect of these considerations will not be clear without a more extensive theoretical modeling effort.

Empirically, data from the family of guard-contact detectors from wafer 12 (Fig. 2.12) at four values of the excess voltage are well-fit by a power law, in each case with an exponent of approximately 2.2. There is no indication in this range of a sharp turnover associated with punching through the implant, supporting the notion that the punch-through voltage is approximately equal to the breakdown. A sensible model might be that in the range considered, both the volume of the depletion region and the quantity of emitted radiation scale linearly with excess voltage, resulting in quadratic overall behavior, with other effects making a modest contribution.

2.7.4 Crosstalk Rise Time

As previously discussed, a crosstalk detection cannot occur simultaneously with the primary detection that gives rise to it. Instead, the crosstalk rate rises from the time of the primary event with a roughly exponential character, leveling off at some asymptotic rate. We expect the rise of crosstalk to be connected to the development of the avalanche in both the emitting and receiving elements. We therefore expect it to be swifter in smaller elements and at higher excess voltages, when the electric field is stronger and electrons therefore gain more energy before impacting an atom. Indeed the rise of crosstalk in the 20-μm devices is so quick that we lack the time resolution to make confident statements about it, but rise-time

data for a pair of 30- and 40-μm detectors is presented in Table 2.3. The limited evidence available suggests that the expected correlations exist.

Data from the 30-μm guard-contact device from wafer 6 (not shown) indicate that the rise time of crosstalk in that array is approximately 2 ns, significantly faster than is seen in the analogous wafer 12 device. This suggests that doping levels or other detector-specific factors may have a considerable impact on avalanche rise time.

Table 2.3: Exponential time constants for crosstalk in 30- and 40-μm wafer 12 guard-contact detectors, originating at the time of laser fire onto a neighboring element.

Element Diameter (μm)	V_{ex} (V)	τ (ns)
30	3	8.5 ± 4.7
	4	5.8 ± 1.4
	5	5.1 ± 1.4
40	3	13.8 ± 1.5
	4	11.3 ± 0.7
	5	6.3 ± 0.2

2.8 Simulations

2.8.1 Spread of crosstalk

To explore the implications of our findings for experimental use of SPAD arrays, we conducted a series of simulations on the propagation of crosstalk through a 4-by-4 element grid at various levels of crosstalk. The simulations were run for different numbers of initially avalanching elements of the array, and were advanced in timesteps of 1 ns. The probabilities of crosstalk on these short times are sufficiently small for the crosstalk rates considered that it is reasonable to sum several crosstalk

rates acting on a quiescent element. For example, if the nearest-neighbor crosstalk rate for a simulation was 200 kHz and the only elements already avalanching in the array were two of the nearest neighbors of a given element, that element would be considered subjected to an effective crosstalk rate of 400 kHz, for a 0.04% avalanche probability in the 1 ns timestep. Crosstalk rates were considered to fall off as $1/r^4$ with distance from the receiving element. The rise time of the crosstalk rate has been neglected, so newly avalanching elements begin contributing to the detection probability of the remaining quiescent elements in the next timestep. Beyond the elements avalanching at t = 0, no source of detections other than crosstalk has been taken into account. In practice, the dark rate of the detector produces some additional avalanches in the course of the gate, but the extent of the effect is highly dependent on the detector used and on the operating environment.

We modeled the time required for crosstalk to spread completely across a 16-element square array, at a range of crosstalk rates and for different numbers of initially avalanching elements (Fig. 2.13). Once every element is avalanching, the array is incapable of making further detections, and sensitivity degrades as elements avalanche, so an application that makes use of a gated-quenching scheme should be cognizant of the timescale of crosstalk in specifying a gatewidth. This is less of a consideration in detectors with moderate crosstalk rates and dark rates low enough that few elements are likely to avalanche in the absence of signal; given a 200 kHz crosstalk rate and starting from a single avalanching element, we saw that a detector array retained at least some sensitivity after 10 μs about half the time. However, higher crosstalk and dark rates conspire to drastically reduce the duration of sensitivity. Since both rates are strong functions of element size, applications making use of larger-diameter elements in particular will have to contend with this fact.

Fig. 2.14 shows the spread of crosstalk across an array which initially has one randomly placed element avalanching. The median time y at which x elements had experienced detections is plotted. Note that the slope of the plot is least—indicating that the probability of a crosstalk event is greatest—when there are comparable numbers of emitting and receiving elements. The probability of crosstalk initially rises as more elements avalanche and contribute to crosstalk in their neighbors, but at some point the trend falls victim to diminishing returns, as the addition of a new emitter no longer statistically counterbalances the loss of a potential target.

2.8.2 With lunar-ranging data

We have conducted simulations of the array based on actual APOLLO ranging data to see if the observed rate of detections occurring after the return pulse is consistent with the crosstalk rates and dependencies presented here, and we find that it is.

The detector currently used by APOLLO is the 40-μm NGC detector from wafer 12, which was not characterized for this work. However, we expect its crosstalk characteristics to be similar to those of the 40-μm GC device. It is operated at approximately 5 V of excess voltage, and so we infer a nearest-neighbors crosstalk rate of about 3 MHz, which was used for the simulation. The lunar return arrives approximately halfway through the 100 ns timing window. For each lunar return pulse in an observing run of 5000 pulses, we therefore simulated the array's behavior over the subsequent 50 ns, in 1 ns time steps, taking as initially avalanching whichever elements of the array (if any) detected a lunar return photon. The $1/r^{4.5}$ distance dependence observed in the lab was used.

Fig. 2.15 shows lunar return data from a data-collection run taken in March 2015 as well as the results of the crosstalk simulation based on that data. The

quantity and timing of crosstalk events observed in the simulation is consistent with the elevated signal rate observed by APOLLO in the wake of the lunar return. This consistency is more marginal when the theoretically-expected $1/r^{3.5}$ distance dependence is used in the simulation. This does not provide incontrovertible evidence against the $1/r^{3.5}$ rule since the adjacent-pair crosstalk rate of the 40-μm NGC device may be less than that of its GC cousin, although this would be contrary to the trend (Table 2.2).

2.9 Conclusions

Characterization of the SPAD crosstalk rate and understanding of its dependence on the physical parameters of the detector and experiment is an asset in photon-counting applications. Multiple approaches in terms of experimental method and data analysis are capable of determining the crosstalk rate, although the limitations and needed corrections associated with each must be understood. While increasing element size and decreasing the pitch on a detector array increases the fill factor and thus detection rates, the observed steep dependences on element separation (falling off faster than $1/r^4$) and element diameter (going approximately as d^7) suggest that the crosstalk implications of these design decisions may be greater than anticipated and the trade-off may merit re-evaluation for some experimenters. By the same token, our results imply that the crosstalk rate can be arbitrarily minimized by the use of sufficiently small detector elements with a sufficiently large pitch. For signal-limited applications, this loss of fill factor may seem prohibitive, but can be recovered by the use of a lenslet array placed in front of the detector, as is done in APOLLO's case.

Conversely, in cases such as laser ranging in which the signal return time is

precisely known, the few-nanosecond crosstalk rise time permits easy distinction between signal and crosstalk detections if timing resolution is great enough, so that even a high crosstalk rate is not an impediment. Indeed, the much-briefer rise times associated with smaller element sizes seem to make larger elements desirable for such applications, with the additional crosstalk being a small price to pay for a longer crosstalk rise time, a better fill factor, and larger avalanche amplitudes that translate directly into better timing precision when using the type of timing electronics design that we have implemented.

Finally, the benefits of operating at higher excess voltage must be weighed against the resulting higher crosstalk rate, but it may often seem to be worthwhile to do so. The same factors that cause the increased crosstalk—a deeper depletion region and a greater probability of avalanche initiation—enhance data collection as well, providing a greater quantum efficiency with increased timing precision. The avalanche amplitude also grows with increased excess voltage, which may further enhance timing precision as in the case of larger element sizes.

This chapter, in part, has been submitted for publication of the material as it may appear in Applied Optics, 2015. Johnson, Nathan H.; Murphy, Thomas W.; Aull, Brian F.; Colmenares, Nicholas R.; Orin, Adam E., OSA Publishing, 2015. The dissertation author was the primary investigator and author of this paper.

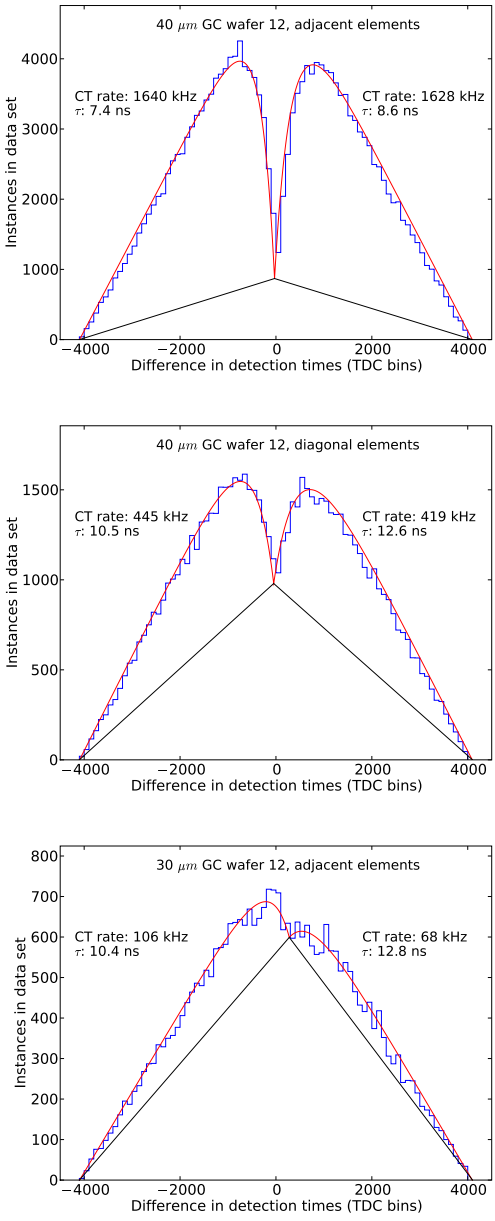

Figure 2.5: The distribution shape characteristic of the steady-illumination approach to crosstalk-rate determination. Crosstalk rates and rise times for the left- and right-hand sides are stated. After eight hours of data collection, the effect is unambiguous for crosstalk rates above 10^6 Hz (top) and down into the $\sim 10^5$ Hz range (middle), but for lower rates the fit (curved lines) to the distribution may be less compelling or essentially impossible. The data in the two upper plots come from the same detector, but the uppermost shows crosstalk between directly adjacent elements, whereas in the other, the elements are diagonally adjacent.

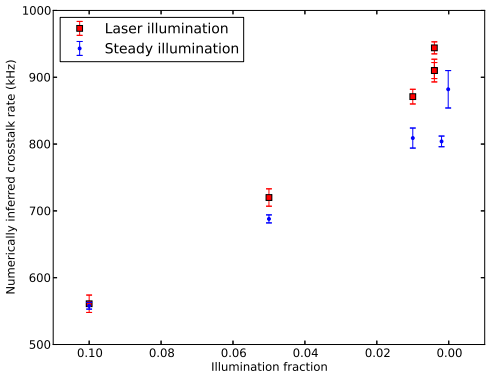

Figure 2.6: Crosstalk rates as a function of background illumination level between a consistent pair of adjacent elements on the 30-μm GC wafer 6 detector at 5 V of excess, inferred from numerical methods and subjected to no corrections. Error bars are derived from propagated Poisson uncertainties on event counts. Whichever experimental approach is used, the inferred crosstalk rate rises as the illumination level decreases due to the decreasing effect of blocking by priors. Both laser- and steady-illumination data are considerably affected by masquerading crosstalk events at relatively high background illumination, but the laser data is little impacted by this effect at low illumination due to the decoupling of illumination during the timing window from the background rate. However, the effect of masquerading crosstalk in the steady-illumination data is likely to depend only weakly if at all on the illumination level, so at low illumination the rates inferred from laser data are both higher and plausibly more accurate.

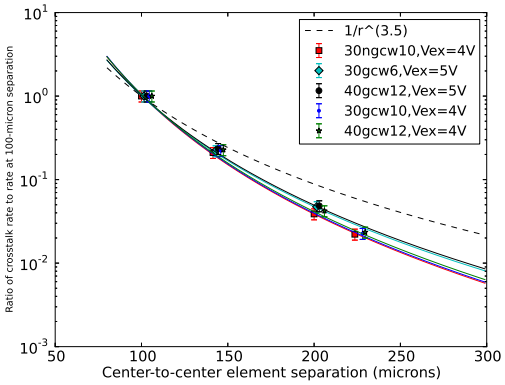

Figure 2.7: The distance-dependence of the crosstalk rate as characterized in a range of detectors at two levels of excess voltage. Absolute crosstalk levels in the several scenarios vary over an order of magnitude for adjacent elements, so rates are here presented as the ratio of the rate to the adjacent-element rate. Uncertainties are set at 15 percent of the calculated rate to account for inherent differences between elements. Even so, the disagreement of the observed distance dependence with the expected $1/r^{3.5}$ power law is evident. The inferred power laws for each detector/excess voltage comparison agree with each other and with a $1/r^{4.5}$ power law at the 1-σ level. The rates used in this figure were derived from laser-illumination data and analyzed with the numerical approach. Rates extracted from the same data by graphical means showed good agreement.

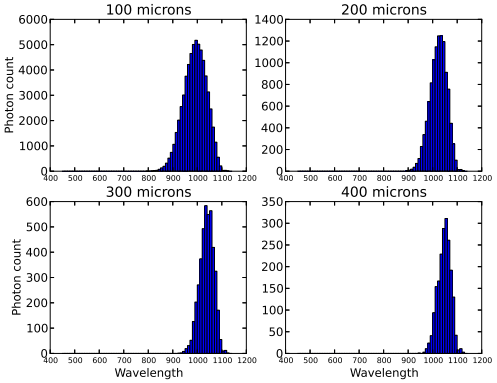

Figure 2.8: Spectrum of photons initiating crosstalk events in detector elements at four different distances from the emitting element. Higher-energy photons are less able to penetrate the intervening material, and so the absorbed spectrum reddens as the element separation increases. This simulation was conducted using elements of 40 μm diameter, and the stated distances are between the centers of the emitting and receiving elements.

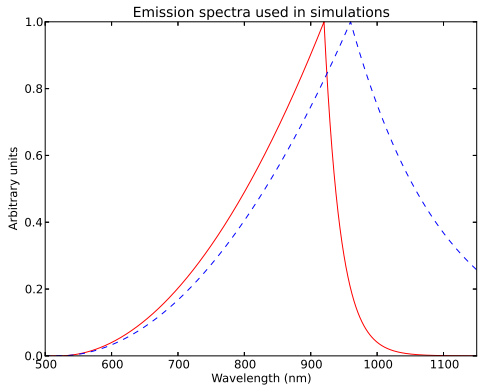

Figure 2.9: An approximation to the emission spectrum found by Rech et al. (dashed) as compared to a spectrum determined in this work to be consistent with our findings regarding the distance dependence of the crosstalk rate (solid).

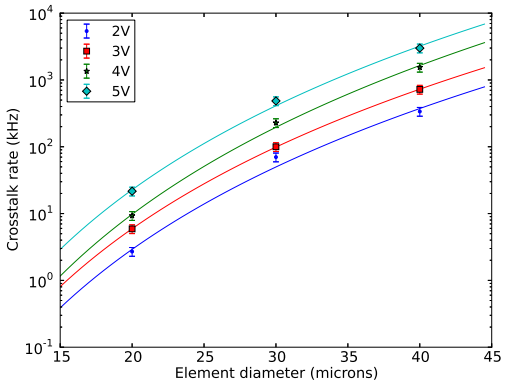

Figure 2.10: Crosstalk rates and best-fit power laws to the family of guard-contact detectors in wafer 12 at a range of excess voltages. In each case the increase in crosstalk is consistent with a dependence on the seventh power of the element diameter. Because observed inherent variability in crosstalk between different element pairs, error bars equal to 15 percent of the calculated rate have been ascribed.

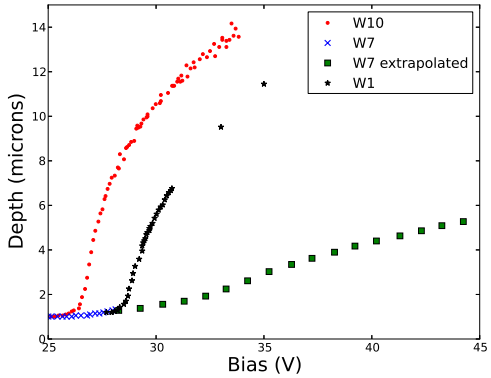

Figure 2.11: In SPAD detectors originating from wafers other than those characterized here, the depth of the depletion region has been observed to expand roughly as the square root of the excess voltage once a 'punch-through' threshold, comparable to the breakdown voltage, has been exceeded.

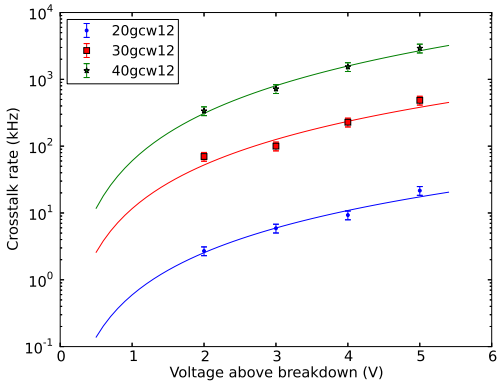

Figure 2.12: Crosstalk rates and best-fit power laws to the family of guard-contact detectors in wafer 12 at a range of excess voltages.

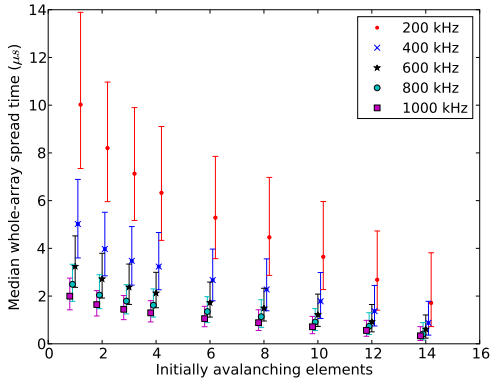

Figure 2.13: Time required for crosstalk to initiate an avalanche in all 16 elements given different numbers of initially avalanching elements, randomly distributed spatially. Each data point represents the results of 1000 simulations. Results are shown for five different crosstalk rates, where the stated rate applies to crosstalk between directly adjacent elements. The physical size of the elements has been taken as small compared to the pitch. Error bars, upper and lower, each capture 34 percent of the outcomes on the corresponding side of the median, rather than representing uncertainty in the location of the median itself.

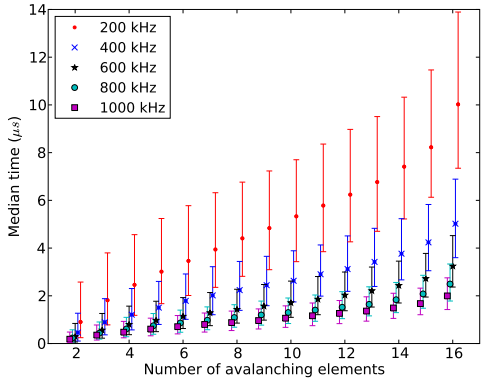

Figure 2.14: Median earliest time within a simulation at which various numbers of detector elements were avalanching, given one randomly placed avalanche at $t = 0$. Each data point again represents 1000 simulations, and the meaning of the stated crosstalk rates and error bars are the same as in Fig. 2.13.

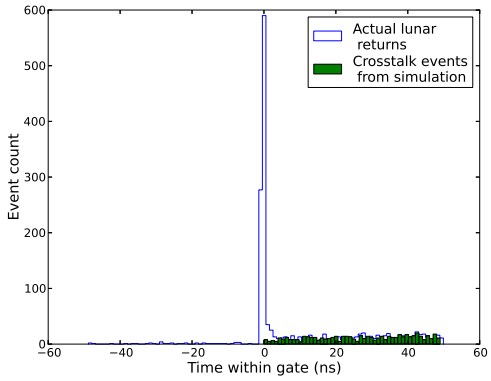

Figure 2.15: Temporal profile of APOLLO lunar returns from a 5000-shot run of March 2015 with events observed in crosstalk simulation overlaid. The rate of crosstalk events rises throughout the timing window due to earlier crosstalk events precipitating additional crosstalk later on, but remains consistent with what was observed in fact.

Chapter 3

Electronics improvements

3.1 Former setup

Until 2013, the APOLLO detector readout electronics (Fig. 3.1) were based fairly closely on a design suggested by [26]. In this version, the APD package was inserted into a motherboard featuring 16 slots for 'daughter boards,' each of which encapsulated the electronics for one APD element. A constant but tunable voltage was applied to all APD channels through the anode, placing them perhaps a volt beneath the breakdown voltage of the device. A positive gate of perhaps 7 V was applied to both the APD cathode (biasing the device above breakdown) and to a 'dummy' channel having the same electrical characteristics as the actual APD, so that a gate of similar form rose on both the detector and dummy channels, with the dummy gate offset to be slightly lower in voltage, by some tens of millivolts, than the APD-side gate. These signals were fed to the two inputs of a comparator. When an avalanche occurred in the appropriate APD element, the avalanche current passed to the daughter board through a coaxial cable and dropped through a 500-Ω resistor isolated from the detector by a transistor, causing the voltage on the APD

side as seen by the comparator to fall. The full amplitude of the drop depended on the particular detector being used and on the amount by which it was biased above breakdown, ranging from about 50 to 200 mV. When the APD-side voltage at the comparator therefore dropped below the level on the dummy side, the ECL-level output of the comparator would flip. This signal was directed to the corresponding channel of the TDC, making a detection.

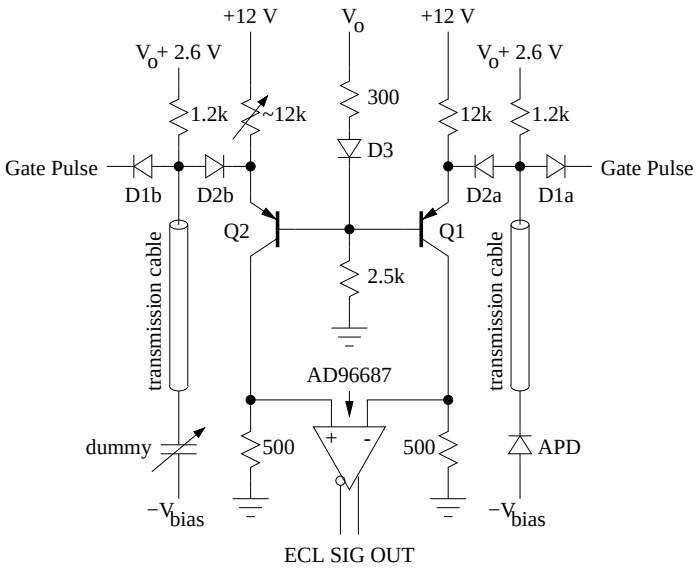

Figure 3.1: Scheme of the former APD readout electronics, in which the avalanche signal was generated by dropping the avalanche current through a transistor-buffered resistor.

The avalanche signal was negative-going with a pronounced RC exponential form, with a time constant of approximately 10 ns. This was believed to be much longer than the time required for the actual spread of the avalanche throughout the detector element, and so it was suspected that resistive and capacitative elements of the daughter boards were the principal determinants of the avalanche speed. We implemented a simple passive-quenching scheme along the lines of [22] using the 20-micron APD array originally provided by Lincoln Lab and were able to observe avalanche signals with a base-to-peak rise of less than 2 ns, albeit with

an amplitude of approximately 8 mV, too small for practical use. Nevertheless this made it clear that the readout electronics were not getting us close to the fundamental limit, and it was expected that if we could achieve a greater avalanche slope with no increase in noise, the corresponding reduction in temporal jitter at the comparator would result in improved timing precision.

We simulated the existing daughter board circuitry using pSPICE in an effort to identify components in which a change of value might decrease the RC time constant of the avalanche signal. These investigations did not identify a way in which timing performance could be improved without compromising the design in some other way. Such alterations as did seem potentially promising were not found to actually result in improved timing when implemented. For example, increasing the value of the resistor through which the avalanche current drops to produce a signal at the comparator increased the amplitude of the drop while also increasing the time constant such that modest improvement in the slope was seen overall, but any potential gains were erased by the corresponding amplification of the noise on the signal. Instead, we opened discussions with the Physics Electronics Shop to explore the possibility of a redesign.

3.2 New setup

The resulting new detector readout system dispenses with daughter boards and coaxial cables, fitting the channel-specific components of the design directly on the main board. Instead of generating the APD signal by dropping the avalanche current through a transistor-buffered resistor, it passes the current to a preamplifier (Fig. 3.3), resulting in a positive-going signal of several hundred millivolts amplitude, a factor of a few larger than was seen for any particular combination of detector and

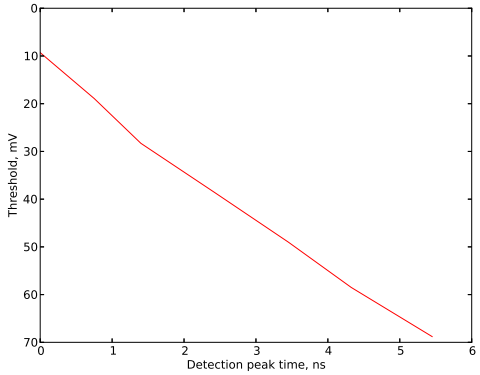

Figure 3.2: Time delay of the detection peak as a function of difference in potential between APD signal baseline and reference, the so-called threshold voltage. A detection is made when the APD signal falls to this threshold level, so this serves as a plot of the voltage drop associated with the avalanche signal. The full amplitude of the avalanche in this case was seen to be 163 mV, and the time constant of the exponential form was calculated to be 10.6 ns. As the threshold decreases, in addition to the depicted shift toward later detection times, the decreasing slope of the avalanche at the intersection point causes the detection peak to spread, reducing the precision that can be ascribed to the detection. Data was taken using the former electronics setup with the 40-μm GC device.

excess voltage when using the old design, with a total rise of 2 to 3 ns. This signal is passed to a comparator as before, where its level is compared to a reference level that is constant rather than an electrical mimic of the APD gate. Other features of the new design include protection circuitry that reduces the voltage across the detector to a safe level if the system is put into DC gate-on state and on-board LEDs around the detector mount for illuminating the APD, which is wanted in the lab for focusing a laser spot on particular elements or points on elements and at Apache Point for alignment of the detector in the optical path. The new board has the same form factor as the old motherboard, although in normal operation it requires three input voltages ($\pm$8 V, 40 V) rather than five. After in-lab testing, the new design was installed at APO in September 2013.

In addition to observation on an oscilloscope, the rise time of the avalanche

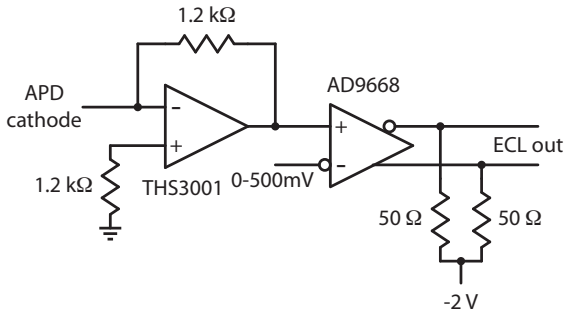

Figure 3.3: Scheme of the current APD readout electronics, in which the avalanche signal is generated by passing the avalanche current to a preamplifier.

signal at the comparator can be probed by observing the change in detection time as a function of the threshold voltage. The results of such a test are presented in Fig. 3.4. The avalanche signal is observed to have a slope of approximately 200 mV/ns. Furthermore, this slope is approximately constant throughout the rise, without the RC exponential form assumed by the avalanche signal under the former design. It seems likely that the rise time of the avalanche signal was formerly limited by the readout electronics, but now we may be closer to the actual timescale of the avalanche as it spreads in the detector element, an idea which is explored in the next section.

3.2.1 Impact on data quality

During a successful data-collection run, each functional element of the APD array records a temporal distribution of lunar returns. Offsets between the different channels are determined and used to combine these distributions, which are then fit, resulting in a single normal point for the run. It is also possible to reduce the data from each channel individually, in which case something like a normal point can be determined for each. These ranges should agree within the uncertainly ascribed to them by the data-reduction pipeline. In practice, however, some inflation of the

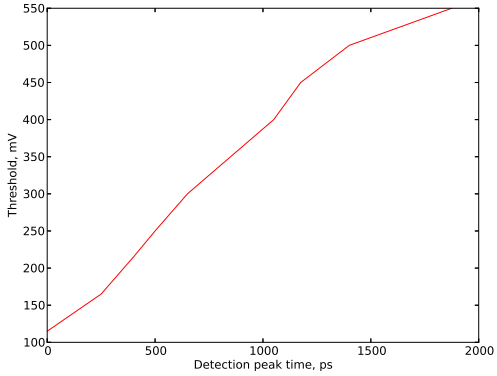

Figure 3.4: Time delay of the detection peak as a function of difference in potential between APD signal baseline and reference, the so-called threshold voltage. A detection is made when the APD signal reaches this threshold level, so this serves as a plot of the rise of the avalanche signal. Detections are not possible at thresholds below a certain level due to noise triggers, nor above the level of the amplitude of the avalanche signal; this figure spans the space between these limits. Data were taken in-lab on the 30-micron device installed at Apache Point prior to 2010, a laser spot being focused on the center of the detector element. The detection peaks at every point were approximately 150 ps full-width at half-maximum.

nominal uncertainties has historically been needed to achieve compatibility. As a figure of merit, we might determine either 1) what size of a root-sum-squares term would we have to add to the single-channel uncertainties from a period in the history of the experiment to achieve inter-channel consistency, or 2) by what factor would we have to scale the uncertainties to achieve the same result? Both approaches have some plausible justification and may serve to indicate the level at which the uncertainties we publish with the normal points may be understated.

With that in mind, Fig. 3.5 gives a sense of the precision and self-consistency of APOLLO data through the life of the experiment. The period approximately coinciding with calendar year 2011 corresponds to a troubled time in APOLLO's history, whereas the period following implementation of the new detector electronics is the most recent one. In terms of raw median normal point uncertainty (top plot), all periods fall close to the millimeter level, with the 2011 span only about 1 mm

higher and the most recent just barely lower. Once the data's self-consistency is considered, however, variations in data quality between the periods are much more apparent. For most time periods during which the original motherboard setup was in use, the size of the required RSS term or scaling factor is such as to put the median normal point uncertainty closer to 2 mm, and for the 2011 data an ultimate median uncertainty of 7 to 9 mm is seen depending on approach. Only in the most recent span is no substantial dilution of normal point precision evidently warranted. On this basis it may be argued that with the new APD electronics, APOLLO is making its most precise measurements by about a factor of two, and its median normal point uncertainty is truly at the single-millimeter level for the first time. As a caveat, the single-channel reduction comparison approach to data self-consistency is only a heuristic, and no absolute calibration of the system has yet been done.

3.3 Spatial-jitter characterization

From the APOLLO perspective, a major purpose of the characterization of the detector suite was to identify a candidate to replace the 30-micron device then installed at APO. It was desired that the new detector have favorable timing characteristics for the determination of precise normal points and a low dark rate to increase the probability of obtaining some identifiable returns in marginal conditions when at least some of the reflectors are in shadow. (When a reflector is in the sun, the lunar background dominates over the detector dark rate.) Timing precision of each of the 16 channels of each candidate detector was determined by focusing a narrow-pulse laser spot in the center of each element, attenuating the laser into the single-photon regime to avoid first-photon bias, and observing the resulting pulsewidth as recorded by the TDC. The dark rates were characterized

by construction of an enclosure which provided a stable, dark environment for the detector. The merits of the candidates being considered, the wafer 6 30-micron guard contact device was selected, with detection peaks of approximately 80 ps uncertainty (1σ) and dark characteristics much more favorable than those of the detector installed at the observatory at the time.

Installation of this detector at APO in August 2010, however, was not a success. The typical 1σ width of the distribution of the fiducial returns ballooned from about 120 ps to the vicinity of 300 ps. Because different sources of uncertainty are added in quadrature to produce this width, the implication was that the contribution of the new detector to the uncertainty was at about that 300 ps level, far greater than what had been observed in the lab. The principal difference between lab operation and the observatory environment is the large amount of electromagnetic interference at APO caused by the firing of the much-more-powerful laser, and at the time this was credited with the observed difference. However, efforts to shield the system from this interference did not produce a major improvement in the overall timing, although calibration of the TDC did show reduced jitter in the operation of that device and of the timing system more generally. We were obligated to revert to the previously-installed detector. The wafer 12 40-micron non-guard contact device was later installed at the observatory, as its steeper avalanche rise would render it less vulnerable to the electronics noise that was believed to be at the heart of the matter.

Armed with the considerably increased avalanche slope seen with the new electronics, we undertook some tests that ended up shedding considerable additional light on the spatial dependence of the timing uncertainty. These tests were of four basic types:

1. Laser spot focused on the center of a detector element (spot size approximately

10 microns) while the amount of neutral density in the beam path is varied. There should be minimal APD-spatial contribution to the resulting peak profile, and we move from the detector-saturating regime to the single-photon regime.

2. Laser spot defocused to fill a detector element and the amount of neutral density is again varied, so we go from the saturation regime to the single-photon but with the APD-spatial component represented in the peak. This is analogous to operation at Apache Point, but the in-lab laser profile should still be peaked at the center since it is Gaussian in form, whereas the APO profile should be completely flat.

3. In the single-photon regime, starting with a laser spot in the center of an element and defocusing until the spot fills the element, gradually introducing the APD-spatial component. This test connects the endpoints of the first two.

4. With a focused spot and in the single-photon regime, scanning across the element from edge to edge in rough steps of approximately 5 microns. This serves to probe the spatial dependence of the timing.

In tests of the first type, we are gradually moving from the multiphoton regime to the single-photon regime. We compare the results of this type of test on the 30-micron device installed at APO until 2010 ('Original 30-micron') and a 40-micron device from wafer 12 that is the 'twin' of the one presently installed ('40GCW12') in Table 3.1. In both cases the peak is observed to broaden by several bins at full-width half-maximum, and the location of the peak shifts toward later times by several hundred picoseconds. At APO a result like this would be attributable to the gradual elimination of first photon bias and the attendant

assertion of the width of the laser pulse in the overall timing uncertainty. The pulse width of the in-lab laser used for these tests is less than that of a single TDC bin, so first-photon effects are not responsible for the shifts observed in this case. Instead, a plausible explanation would stem from the fact that the laser spot is not perfectly focused. As previously described, we expect a detection to be made once a certain volume of the detector element has been filled by an avalanche, and the fraction of the element that must be avalanching for this to occur should depend on both the threshold voltage (the difference between the reference voltage and the APD-side baseline at the comparator) and on the total amplitude of the avalanche signal, which in turn depends on electrical properties of the individual detector and on the degree of excess voltage applied during the gate. If numerous photons are expected to strike the detector element in a single laser pulse, there will be a commensurate number of avalanches that do not initially overlap due to the finite extent of the laser spot but expand independently. As a result the detection threshold will be reached earlier than in the single-photon regime, and the detection peak will be less broad because the avalanche signal is rising more steeply, which means less jitter at the comparator.

Now consider the second case, in which the laser spot has been defocused to fill an entire detector element and we again move from the saturated regime to the single-photon (Table 3.2). The original 30-micron device sees a comparable progression in this case to what was observed when the laser spot was more focused, but in the case of the 40-micron detector the detection peak experienced both more broadening and a greater shift to later times. Some loss of detection precision and delay is expected due to the greater degree of spatial uncertainty in this regime. Avalanches originating near the edge of the element quickly encounter the edge

Table 3.1: Results of tests in which a laser spot was focused in the center of a detector and neutral density was gradually added to move into the single-photon regime. For each detector there is presented the amount of neutral density, the width of the detected peak in ps, and the time with respect to the start of the timing window at which the peak occurred. The peaks are observed to broaden and shift to later times as the level of illumination falls.

	Detector			
	Original 30-micron		40GCW12	
ND	FWHM (ps)	Peak time (ns)	FWHM (ps)	Peak time (ns)
1.5	115	35.68	135	35.95
2	115	35.69	157	36.02
2.5	126	35.74	157	36.08
3	143	35.80	156	36.15
3.5	152	35.85	-	-
4	192	35.92	219	36.25
5	164	35.94	224	36.32

and thereafter grow more slowly; hence they reach the threshold volume at later times and with reduced slope relative to those originating near the center. In the saturation regime, at least one photon likely strikes near the center, but as we move into the single-photon regime the probability of this lessens and the detection peak gets broader. The fact that the 40-micron device saw these effects to a greater degree than the original 30-micron would appear to indicate that the threshold was lower relative to total avalanche height during the tests of the latter. This would mean an increased chance of a detection being made before the growth of the avalanche was limited by contact with an edge.

To attempt to disentangle saturation effects from spatial ones we undertake tests of the third kind, gradually defocusing the laser spot while staying in the single-photon regime. This test was done on the two detectors described above and also on the wafer 6 30-micron guard-contact device abortively installed at APO in 2010 ('30GCW6' in Table 3.3). In these cases we see essentially no shift of the

Table 3.2: Results of tests in which a laser spot was defocused to fill a detector element and neutral density was gradually added to move into the single-photon regime. For each detector there is presented the amount of neutral density, the width of the detected peak in ps, and the time with respect to the start of the timing window at which the peak occurred. Relative to the results in Table 3.1, the 40-micron detector sees more time-delay and peak broadening as the illumination decreases.

	Detector			
	Original 30-micron		40GCW12	
ND	FWHM (ps)	Peak time (ns)	FWHM (ps)	Peak time (ns)
1	111	35.61	112	35.69
1.5	109	35.63	-	-
2	110	35.66	112	35.88
2.5	111	35.69	113	35.90
3	142	35.75	148	35.98
3.5	211	35.84	263	36.13
4	223	35.90	386	36.29
5	218	35.94	-	-

peak of the distribution but broadening whose degree varies between the detectors. This makes sense in view of the Gaussianity of the beam. As the beam becomes defocused, he laser spot may to the eye 'fill' the detector element, but it is not uniformly distributed over it. Because the illumination is still greatest at the center, most detections are drawn from the same distribution seen in the focused-laser, single photon case. A smaller number of detections are delayed due to originating near the element edge, and so the combination results in a broadened distribution whose peak is barely delayed compared to the case in which the beam is focused on the element center. This is not a perfect analogy for conditions at APO, where illumination of the element actually is uniform: in the case of the fiducials due to the diffuser, and in the case of lunar returns due to the footprint of the returns on the earth being much larger than the primary mirror.

Noteworthy in this case is that the peak in the original 30-micron device experiences less broadening as the beam is defocused than do the other two.

This may be attributable to differences in threshold voltage relative to avalanche amplitude, or to actual inherent differences between devices. The threshold-to-amplitude ratio was not well controlled in taking these data, so they are of limited use in disentangling the possibilities.

The detection distributions just discussed represent an aggregation of the distributions of photons arriving in all parts of the detector element. We can get a better look at the distributions arising from detections in specific regions of the element by scanning a focused spot across it and taking data at a number of positions while in the single-photon regime. Table 3.4 represents the results of tests of this type. As we might have expected based on the foregoing results, photons striking the element near an edge are detected later and with less timing precision than those striking near the center. This, again, makes sense; if a photon strikes near the edge, the growth of the avalanche signal is quickly limited, and so it takes longer to exceed the level of the reference (later detection) and crosses with a shallower slope (hence more timing jitter). Noteworthy here is the extent to which the variability in the breadth of the distribution depends on the detector; when using the original 30-micron device, detections of photons striking near the edge occur at later times than those striking near the center, but with barely less precision, whereas in the other detectors this effect was more pronounced. Commenting on the defocusing tests, it was noted that the broadening of the peak observed under those conditions is difficult to attribute firmly to either difference in threshold voltage or inherent differences among the devices, but in this case device properties seem more convincingly implicated. If we say that independent of detector there is one slope of the avalanche signal when it is expanding in all directions and another slope once the avalanche is limited by an element edge, then we might conclude that detections of edge-striking photons in the original 30-micron device were seen

to be relatively little delayed because the ratio of threshold to avalanche-signal amplitude was relatively small during those tests. However, we would then expect the loss of precision in detecting such photons to be device-independent, or even greater in lower-threshold circumstances due to the two-dimensional spread of the avalanche, but that is not what is seen here. Instead, it seems possible that the rate of lateral propagation of an avalanche is different from detector to detector. Because uniform illumination of the elements is an unavoidable aspect of ranging with APOLLO, detectors with slower avalanche propagation rates would have a lower bound on their precision that could be a significant source of uncertainty overall.

To further explore the relationship between the threshold voltage and the spatial jitter, we conducted tests of the scan-across type while varying the threshold. The 40-micron guard-contact device from wafer 12 was used. Results are given in Table 3.5. From these data it seems that the extent to which avalanches originating near the element edge are detected later than those starting near the center does depend on threshold voltage, with a delay of perhaps 300 ps when the threshold was 30 mV but closer to 500 ps when it was 150 mV. (No threshold smaller than 30 mV was workable in this case because a transient associated with the start of the gate would cause a false trigger below this level.) This makes sense if we take the view that the avalanche signal rises more slowly when it quickly becomes limited by the element edge. However, these data also offer tentative support to the view that the extent of the delay is also partly attributable to inherent characteristics of each device. Even at the lowest possible threshold, the delay seen in the 40-micron device was larger than what was observed in a scan-across test of the original 30-micron device (Table 3.4). If the contribution of spatial uncertainty to detection time is indeed a strong function of the individual detector, it may be that no amount of

interference mitigation would have resulted in good timing performance with the wafer 6 30-micron device at Apache Point. Spatial uncertainty should be a greater consideration in any future detector-characterization effort.

More immediately, these results imply that the threshold voltage should be set as low as possible. Conventionally, decisions about the threshold have been based on attempting to maximize the slope of the avalanche signal at crossing. With the previous electronics, this signal had a pronounced appearance of exponential decay and minimal threshold was indeed aimed for. In the new system, the avalanche signal has more of a logistic appearance with a slope seemingly greatest when it has risen halfway to its final level. However, it seems likely that the additional precision obtained by setting the threshold at this point is not worth the corresponding deterioration in the spatial uncertainty. Because a change of threshold would impact the fiducial and lunar returns in the same way, it seems plausible that modelers would need apply no separate range-bias parameter to subsequently collected data.

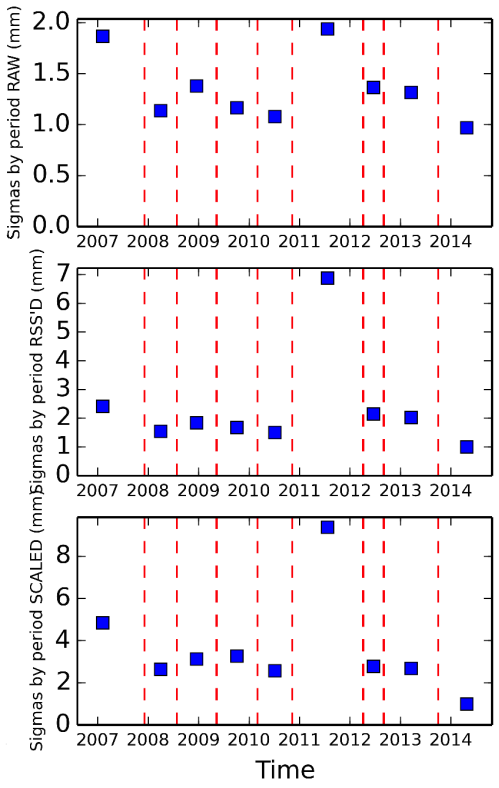

Figure 3.5: Median APOLLO normal point uncertainties by data period. For any span of data we can ask what size of RSS term would we need to add into the uncertainties produced by single-channel data reduction to achieve self-consistency (middle), or alternatively by what factor would we have to inflate the uncertainties to achieve the same result (bottom)? Formerly, the results of such checks provided some reason to think APOLLO uncertainties might be understated by approximately a factor of 2, but in the span of data since the implementation of the new detector electronics, no appreciable inflation of normal-point uncertainties appears needed to achieve self-consistency.

Table 3.3: Results of tests in which a laser spot was gradually defocused to fill a detector element while in the single-photon regime. For each detector there is presented the amount of neutral density, the width of the detected peak in ps, and the time with respect to the start of the timing window at which the peak occurred. The large difference in peak bin number associated with the last detector is due to a difference in the delay between the signals used to drive the pulse-picking modulator and to bias the detector above breakdown, but is not important here. The detection times are observed to change very little, although the peaks broaden as the beam fills the element.

	Detector					
	Original 30-micron		40GCW12		30GCW6	
Lens displacement (microns)	FWHM (ps)	Peak time (ns)	FWHM (ps)	Peak time (ns)	FWHM (ps)	Peak time (ns)
0	161	35.88	251	37.38	258	56.76
10	164	35.85	271	37.39	228	56.74
20	162	35.85	281	37.39	223	56.71
30	168	35.85	300	37.40	237	56.71
40	184	35.85	334	37.42	247	56.72
50	204	35.86	355	37.40	269	56.72
60	227	35.86	370	37.43	-	-
70	239	35.87	386	37.46	299	56.74
80	245	35.88	-	-	324	56.76
90	-	-	-	-	344	56.82

Table 3.4: Results of tests in which a focused laser spot was gradually scanned across a detector element while in the single-photon regime. For each detector there is presented the width of the detected peak in ps, and the time with respect to the start of the timing window at which the peak occurred. The spot positions represent not specific amounts of scanning distance but rather approximately equal increments from one edge of the element to the other, passing through the center. Photons impacting near an element edge are always detected later than those arriving near the center, but the effect seems more pronounced for the wafer 6 30-micron device.

	Detector					
	Original 30-micron		40GCW12		30GCW6	
Spot position	FWHM (ps)	Peak time (ns)	FWHM (ps)	Peak time (ns)	FWHM (ps)	Peak time (ns)
Left edge	210	36.01	310	37.48	440	57.04
Center left	196	35.94	265	37.40	367	56.94
Center	174	35.88	240	37.40	253	56.74
Center right	189	35.95	315	37.49	269	56.75
Right edge	211	36.03	336	37.68	416	57.06

Table 3.5: Detection peak centers and widths at half-maximum as a function of threshold voltage and laser-spot position in the 40GCW12 detector. As before, the spot positions represent not specific amounts of scanning distance but rather approximately equal increments from one edge of the element to the other, passing through the center, which would be between positions 3 and 4. The difference in detection times between edge-impacting and center-impacting photons is observed to be a function of threshold voltage.

Threshold (mV)	Spot Position	FWHM (ps)	Peak time (ns)
30	1	242	55.79
	2	190	55.62
	3	141	55.50
	4	149	55.52
	5	190	55.69
	6	199	55.78
50	1	245	55.99
	2	195	55.88
	3	163	55.71
	4	138	55.66
	5	188	55.77
	6	197	55.97
100	1	280	56.35
	2	∼200	∼55.80
	3	138	55.98
	4	137	55.99
	5	218	56.14
	6	246	56.42
150	1	250	56.64
	2	∼225	∼56.45
	3	170	56.25
	4	144	56.21
	5	204	56.32
	6	294	56.65

Chapter 4

Data analysis

For over 50 years, distance measurements within the solar system have provided among the best empirical constraints on gravitational theory, constraints which have grown more stringent as the number and precision of measurements has increased and experiments have multiplied to include radar ranging to the inner planets, ranges to spacecraft and probes throughout the solar system, and laser ranges to lunar retroreflectors. Since the early days of the field, the agreement or lack thereof of ranging data with the predictions of general relativity has been assessed by fitting the increasingly large and varied data sets to a model of the solar system that attempts to account for all relevant physical effects to a level of precision comparable to that of the data itself. As a result, improvements in data quality have historically spurred parallel developments in model sophistication, and today there exist several advanced codes that are able to derive gravitational parameter estimates from solar-system ranges via a least-squares process. One such model, the freely available Planetary Ephemeris Program (PEP), we used in this work. This approach furnishes formal values for the uncertainties in those parameter estimates, but it has long been understood that the 'true' uncertainties

are larger than what is formally stated. As a result, published constraints derived from solar-system measurements have typically scaled these formal uncertainties by some factor on the order of 10, but no agreed-upon method of doing so exists and settling on one poses challenges for researchers. We discuss the nature of the problem and present a technique of deriving realistic uncertainty estimates based on resampling of the data. This approach has been implemented to work jointly with PEP to produce uncertainty estimates, and some examples of its use are presented.

4.1 Least-squares Modeling

Fitting data to a model via a least-squares process is an old and developed art; see for example [33]. Nevertheless, we summarize the key assumptions, reasons, and procedures here. In the archetypical case, the outcome of a planned series of measurements is precisely expressed as a function of some number of unknown parameters whose values are to be determined. If that function is of the form $F(x) = \sum_{i=1}^{n} a_i f_i(x)$, where a_i are the parameters and the 'basis functions' $f_i(x)$ are functions of none of the parameters but only of some independent variable like the time at which the measurement is taken, then the problem is said to be 'linear.'

The planned measurements are then actually taken with results y_j, and some uncertainty σ_j is ascribed to each measured value. The uncertainty is almost always understood as the standard deviation of a Gaussian probability distribution, a view which is quite consequential to the operation of the least-squares process and so will bear additional explication and scrutiny momentarily. For each measurement then we can write an expression for the σ-denominated or 'normalized' residual as a function of the unknown parameters: $r_j = (y_j - F(x_j))/\sigma_j$, where x_j is the value of the independent variable at which the measurement was made. If the model is

linear in the parameters as defined above, note that the partial derivatives of the residual of a measurement with respect to the model parameters are proportional to the basis functions evaluated at the value of the independent variable when that measurement was taken.

We wish to arrive at some most-probable value, or 'point estimate,' for each parameter. This can be considered to be the set of parameter values in light of which the data that we have taken was most likely to be observed. Obviously parameter values that differ, at least in some degree, from these best-fit values do not render the data that we have in fact observed absolutely impossible to account for, but merely less likely to have been observed; therefore we are also seeking a confidence interval for each parameter, which is to be for each parameter some indication of the span of values of that parameter that are at least plausible in light of the data. In order to arrive at point estimates and confidence intervals that are worthy of the name, we need some numerical expression of the likelihood that a measurement we have made would have occurred, given certain values of the parameters. The assumption of Gaussianity of the probability distribution of the data point, with standard deviation expressed as the measurement uncertainty, suggest that this likelihood for one data point is proportional to e^{-r^2}, where r is the normalized residual as defined above, a function of the parameter values and of the stated measurement uncertainty. (Clearly in order to accurately state the absolute probability of the measurement a normalizing prefactor would be needed, but we are mostly interested in the *relative* likelihoods of the data under different sets of parameter values, so all such factors will cancel out.) The likelihood of multiple data points being the product of their individual likelihoods, we can express the total likelihood of n data points as proportional to $\prod_{i=1}^{n} e^{-r_i^2} = exp(-\sum_{i=1}^{n} r_i^2)$. This expression is monotonic in the argument of the exponent, so maximizing the

likelihood of the observed data is equivalent to *minimizing* the sum of the squares of the normalized residuals. This sum is famous as χ^2.

For a given data set, then, consisting of measurements and (crucially) estimated uncertainties in those measurements, we can compute a value of χ^2 for any set of values of the model parameters. Thus, χ^2 forms a surface in a space whose dimensionality is equal to the number of parameters. Our sought-after point estimates for the parameter values are those at which this surface experiences an absolute minimum, and our confidence intervals will be some expression of how steeply the surface rises in the vicinity of that minimum, which in turn is closely tied to the validity of our estimates the measurement errors and to the assumption that they represent the standard deviations of a Gaussian probability distribution for their associated measurements.

Mathematically, the procedure of minimizing χ^2 is likely a familiar one. The expression for χ^2 is partial-differentiated with respect to each of the model parameters in turn and each resulting expression set equal to zero. This system of equations can then be solved for the parameter values. As the number of parameters, and therefore of equations, may be large, a matrix formulation is desirable. Consider the minimal case in which there are two observations and the model is linear in two parameters. χ^2 is then given by: $\chi^2 = r_1^2 + r_2^2 = (\frac{(y_1 - a_1 f_1(x_1) - a_2 f_2(x_1))}{\sigma_1^2})^2 + (\frac{(y_2 - a_1 f_1(x_2) - a_2 f_2(x_2))}{\sigma_2^2})^2$. For the partial derivatives with respect to the parameters we then have

$$\frac{\delta \chi^2}{\delta a_1} = -2\frac{f_1(x_1)(y_1 - a_1 f_1(x_1) - a_2 f_2(x_1))}{\sigma_1^2} - 2\frac{f_1(x_2)(y_2 - a_1 f_1(x_2) - a_2 f_2(x_2))}{\sigma_2^2} \text{ and}$$
$$\frac{\delta \chi^2}{\delta a_2} = -2\frac{f_2(x_1)(y_1 - a_1 f_1(x_1) - a_2 f_2(x_1))}{\sigma_1^2} - 2\frac{f_2(x_2)(y_2 - a_1 f_1(x_2) - a_2 f_2(x_2))}{\sigma_2^2}.$$

Setting these equal to zero and regrouping, we have

$$\frac{f_1(x_1)y_1}{\sigma_1^2} + \frac{f_1(x_2)y_2}{\sigma_2^2} = a_1(\frac{f_1(x_1)^2}{\sigma_1^2} + \frac{f_1(x_2)^2}{\sigma_2^2}) + a_2(\frac{f_1(x_1)f_2(x_1)}{\sigma_1^2} + \frac{f_1(x_2)f_2(x_2)}{\sigma_2^2}) \text{ and}$$
$$\frac{f_2(x_1)y_1}{\sigma_1^2} + \frac{f_2(x_2)y_2}{\sigma_2^2} = a_1(\frac{f_1(x_1)f_2(x_1)}{\sigma_1^2} + \frac{f_1(x_2)f_2(x_2)}{\sigma_2^2}) + a_2(\frac{f_2(x_1)^2}{\sigma_1^2} + \frac{f_2(x_2)^2}{\sigma_2^2}).$$

In matrix form, these equations are represented as

$$\begin{pmatrix} \frac{f_1(x_1)y_1}{\sigma_1^2} + \frac{f_1(x_2)y_2}{\sigma_2^2} \\ \frac{f_2(x_1)y_1}{\sigma_1^2} + \frac{f_2(x_2)y_2}{\sigma_2^2} \end{pmatrix} =$$

$$\begin{pmatrix} \frac{f_1(x_1)^2}{\sigma_1^2} + \frac{f_1(x_2)^2}{\sigma_2^2} & \frac{f_1(x_1)f_2(x_1)}{\sigma_1^2} + \frac{f_1(x_2)f_2(x_2)}{\sigma_2^2} \\ \frac{f_1(x_1)f_2(x_1)}{\sigma_1^2} + \frac{f_1(x_2)f_2(x_2)}{\sigma_2^2} & \frac{f_2(x_1)^2}{\sigma_1^2} + \frac{f_2(x_2)^2}{\sigma_2^2} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix},$$

which for convenience we write as $\mathbf{Y} = \mathbf{CA}$. Collectively, the relationships embodied by this system are called 'normal equations.' Although the algebra required to explicitly produce this matrix equation becomes more complex as the number of parameters and observations increases, certain salient features of its components are preserved. The column matrix $\mathbf{Y}$ consists of elements that are functions of the measured values, stated uncertainties of those measurements, and basis functions evaluated at the values of the independent parameter at which those measurements were made. Each element of $\mathbf{Y}$ is a sum of terms, and each term corresponds to one measurement. The matrix $\mathbf{C}$ is symmetric. Each of its elements is also a sum of terms corresponding to the measurements, and each of those terms is a product of two basis functions, again evaluated at the value assumed by the independent variable when the corresponding measurement was taken.

Recall that the basis functions are the partial derivatives of the residuals with respect to the parameters. As such, $\mathbf{C}$ quantifies the propensity of the residuals of each data point to grow or shrink in response to changes in the parameter values. In fact, $\mathbf{C}$ is the Hessian or curvature matrix of the χ^2 surface in parameter space. In the linear case, the basis functions from which $\mathbf{C}$ is constructed are independent of the parameter values, and we have nowhere had to make any simplifications of the relationship between the measurements and the parameter values. $\mathbf{C}$ therefore is a complete description of the shape of the χ^2 surface and assumes the same

form everywhere in parameter space. The shape it describes is a paraboloid with dimensionality equal to the number of parameters.

To solve for the parameter values, it is necessary to invert $\mathbf{C}$ and hit both sides of the above matrix equation on the left with the result. Thus,

$\mathbf{C}^{-1}\mathbf{Y} = \mathbf{C}^{-1}\mathbf{C}\mathbf{A} = \mathbf{I}\mathbf{A} = \mathbf{A}$.

$\mathbf{C}^{-1}$ turns out to be the covariance matrix, a symmetric matrix which has the variances of the parameter estimates on its diagonal and covariances, representing the degree of correlation of the model parameters, in its off-diagonal elements. Both are sensitive to the assertions that have been made about the size of the measurement uncertainties.

For the ideal problem that we have been considering, the parameter values produced by this method are the best-fit values. No additional steps are needed, nor must any estimates of the parameters be found beforehand.

4.2 The Uncertainty Problem

Using ranging data to estimate solar-system and gravitational parameters differs in several respects from the simple case just described, and the challenges of assigning realistic uncertainties to the parameter estimates arise from these differences.

A program like PEP incorporates a model with several hundred parameters that can be fit to ranging data. In the particular case of PEP, these include the masses of the eight planets, Pluto, and several of the largest asteroids; the six orbital parameters of each of the planets at the epoch of integration in 1968; rotational parameters of the moon and Mars; coordinates of the launch stations on the surface of the earth; coordinates of ranging targets on the surfaces of the

moon and Mars; and time-delay bias parameters for many of the ranging data series. In the ideal example above, the model-predicted value of the outcome of a measurement could be computed by plugging hypothetical values of the parameters and independent variable into a formula. The basis functions of that model were themselves independent of the parameters, so no prior knowledge about the parameter values was needed to construct the matrices $\mathbf{Y}$ and $\mathbf{C}$.

In a large model of the solar system, on the other hand, not only are the basis functions not independent of the parameters, but there is no function that makes it possible to calculate an observable, like the outcome of a ranging measurement between two known locations at a known time, even if the parameter values are specified. Instead, for starting values of the parameters that are ideally not too far from the eventual best-fit values, the equations of motion for the bodies of the solar system must be integrated from the epoch numerically, creating an ephemeris that describes the positions and, as may be desired, orientations of all the bodies at regular intervals, on the basis of which the predicted outcomes for observables can then be determined. Combined with a data set, numerical integration also produces the values of the basis functions — which is to say the partial derivatives of the residuals with respect to the parameter values — for the parameter values on which it is based. This permits the construction of the matrix $\mathbf{C}$, but unlike in the former example this matrix, effectively a second-order expansion about a point, only describes the curvature of the χ^2 surface in the vicinity of the point in parameter space from which the numerical integration departed. As such, we have again effectively envisioned that surface as a hyperparaboloid, based on the curvature that actually exists at a single point somewhere on the 'real' surface. When $\mathbf{Y}$ is likewise computed, therefore, and the normal equations solved, the parameter values that result are the values corresponding to the minimum of the

imagined surface. The real surface is bound to be a somewhat different, likely more complicated shape, and so these values do not correspond to the actual χ^2 minimum. However, they can be used as feedstock for a new round of numerical integration, and as long as the real surface is moderately well-behaved and not unduly pitted with local minima, and sufficient precision is available in the partial derivatives determined by the integration, over the course of several such iterations the true minimum should be found and the parameter values become stationary, or nearly so.

However, the PEP model does not perfectly describe the solar system or the earth-moon system, and this has ramifications for the uncertainties in the parameter estimates. An entirely complete model of the relevant systems is not possible, since at some level effects become relevant for which no agreed-upon model exists: the minute features of the atmosphere above ranging sites, for example, or the lunar interior. Such unmodeled or partially modeled effects will be safely irrelevant to the analysis if their scale is smaller than the uncertainties in the measurements to which they relate, but in practice, improvements in data quality have historically prompted modelers to bring their codes to a comparable level of precision. In the case of millimeter-level data from APOLLO, still less than 10 years old, refinements in modeling capable of producing residuals comparable to the measurement uncertainties have not yet been completed. The JPL ephemeris code appears capable of producing residuals to APOLLO data about 5 times as large as the RMS uncertainty, whereas when using PEP a scale of 10 to 15 times is typical, with models maintained at the University of Hannover and the Paris Observatory performing comparably or somewhat worse.

Generically, when the residuals produced by a fit to a model are larger than the stated measurement uncertainties, understatement of the measurement

uncertainties may be the cause instead of or in tandem with under-modeling. In the specific case of APOLLO data, checks of internal consistency (Section 3.2.1) indicate that while measurement uncertainties may be understated by a factor of approximately 2, unmodeled or mis-modeled effects account for the majority of the mismatch between residuals and uncertainties. A Fourier decomposition of the residuals is potentially helpful in distinguishing these two effects, as understatement of the uncertainties carries no signal, whereas the residuals might carry the periodic imprint of an unmodeled effect. However, clear signatures in the residuals might be partially or wholly 'soaked up' by some combination of the large number of modeled parameters. Residuals to APOLLO data produced by PEP do not show any unmistakeable signature. Moreover, fits to APOLLO data using the several existing capable models to not produce residuals with clear common features, suggesting that modeling, not understated measurement uncertainties, is the principal cause of the discrepancy.

In any event, it is possible to make some progress on the realistic-uncertainty question without adopting any particular stance on the cause of the residual-uncertainty divergence. Intuitively, we expect the 'true' values of the parameter uncertainties to be larger than what is formally produced by the least-squares process whether modeling or understatement of errors is principally responsible. In the former case, the residuals the sum of whose squares has been minimized are not reliable because the theoretical expectation for the measurements on which the residuals are based is not correct at some unknown level. In the latter, the normalized residuals are stated to be larger than their 'true' values because the σ_i are incorrectly small, and so the value of χ^2 is more sensitive than it ought to be to changes in the parameter values. Admitting both possibilities, the 'true' value of a residual is $r = (y_{\rm obs} \pm \sigma_{\rm obs}) - (y_{\rm true} \pm \sigma_{\rm true}) = y_{\rm obs} - y_{\rm true} \pm \sqrt{\sigma^2_{\rm obs} + \sigma^2_{\rm true}}$. Measurement

uncertainty and modeling uncertainty, then, contribute in indistinguishable ways to the residual uncertainty, so we can avoid deciding at this point in what proportion each is implicated in the residual-uncertainty mismatch as long as our model for any additional uncertainty is independent of the source thereof. Specifically, let us momentarily adopt the view that the outcome of each measurement according to the model differs from the value predicted by a perfect model according to a Gaussian distribution that is the same for each measurement; and that the process by which the data is reduced introduces some additional uncertainty that is also the same for each measurement. In that case, the uncertainty of each residual should be inflated by some new σ arising from these factors, the magnitude of which is unknown but which is the same for every data point. Adopting the approach of [32] (Chapter 9), we can make σ a parameter of the fit and determine a joint probability distribution for it as well as all the physics parameters. Probability distributions for the parameters of interest could then be determined for each parameter of interest by marginalizing this distribution over all other parameters.

In practice, it is not feasible to implement this approach for the solar-system problem. A single additional quadrature uncertainty parameter is probably not a good model for the entire varied data set, but even if it were, implementation of such a meta-parameter would require a major reworking of PEP or one of the other complex ephemeris-generation and data-fitting programs, an effort for which resources are not available and which would probably not be the best use of those resources if they were. Conceivably, PEP in its current form could serve as the engine for a Markov chain Monte Carlo simulation that would determine the joint probability distribution. Ideally beginning from a location in parameter space not far from the χ^2 minimum, a random vector of adjustments to all parameters – including the additional quadrature uncertainty term — would be generated, with

the 'new' value of the uncertainty term being incorporated directly into the data and the 'new' values of the physics parameters being enforced in PEP by a constraint mechanism that is already in place. PEP could then be used to determine the value of χ^2 associated with the 'new' parameter values, and the change in total χ^2 due to the random parameter adjustments would serve as the basis for the transition kernel of the simulation. This is problematic, however, because of the aforementioned non-triviality of the relationship between the parameter estimates and the expectation values of ranging measurements. PEP cannot determine the residuals, and hence total χ^2, associated with a vector of perturbed parameter estimates except by re-integrating the solar system with those estimates as initial conditions, with the result that a single step of such a simulation would take about 30 minutes. Thus, even a modestly sized MCMC simulation would not be complete for months on a single processor, or substantial computing resources would have to be committed to solving it in a reasonable amount of time, both of which approaches seem like an overreaction to the import of the question. PEP does produce an estimate of the value of χ^2 expected from a set of parameter adjustments as a matter of course, but this is achieved by ratifying both the stated measurement uncertainties and the second-order approximation to the χ^2 surface implied by the local curvature matrix, and so does not provide information based on the true shape of the surface as would be wanted.

However, it can be shown for the case of the mean ([32] pg. 225), and seems plausible in general, that the probability distribution for an uncertainty scaling parameter such as we have described peaks at the value that causes the mean measurement uncertainty to be equal to the RMS residual, since the likelihood of the data given the best-fit model will then be at or near its peak. Given the challenges of actually determining the joint probability distribution, if the

measurement uncertainties are of roughly the same size it is convenient to simply scale them by a suitable factor, or inflate each with suitably sized quadrature term, and then use PEP or the equivalent to determine the probable ranges for all of the physics parameters under the new data set. As can be seen from the way in which the measurement uncertainties appear in the information matrix, inflation of all uncertainties by an integer factor will scale the formal parameter uncertainties by the same factor. If the additional uncertainty is added in quadrature instead of as a scaling factor, so as to model the effect of undermodeling, the specific scaling relationship becomes a little muddier but the general effect is to inflate the parameter uncertainties by approximately a factor equal to the RMS normalized residual to the best-fit model when the measurements were fit at full weight. When many datasets are being fit simultaneously it would be inconvenient to determine the appropriate additional uncertainty for each one for a given combination of included data and adjusted parameters, but for a small subset this approach does provide an approximate, easily obtained answer to the central question that can be used as a sanity check for the results of more nuanced approaches.

4.3 Resampling and Bootstrap Methods

Even if PEP or the equivalent provided no hint of the precision of its parameter estimates, it would still be possible to derive confidence intervals from the analysis of many data sets. The parameters of interest could be estimated on the basis of each data set, and the breadth of the distribution of estimates so obtained would be an indication of the precision of each individually, so long as all the data sets possessed comparable constraining power over the parameters. Of course, in reality such numerous realizations of the data do not exist. It would be possible

to create a large number of data sets by subdividing the existing data, but only at the cost of a corresponding loss of constraining power. However, an arbitrary number of realistic data sets can be constructed for this purpose via resampling methods. Resampling in the form of successively removing each observation from a data set and estimating the quantity of interest on the basis of the remainder — later known as the delete-1 jackknife — was pioneered by Quenouille [34] and refined by Tukey [35]. Efron [36] extended the concept by constructing many data sets of the same size as the original through random sampling with replacement, a technique that he memorably christened the 'bootstrap.' Bootstrap methods have since been used to determine standard errors and estimator biases in a wide variety of fields, and a number of different wrinkles on the technique have entered the literature [37, 38, 39, 40, 41, 42, 43, 44, 45].

For a problem of parameter estimation through regression like that which confronts us now, two approaches to the bootstrap suggest themselves, with the potential for each to serve as a check on the other and for differences between them to shed light on the particular nature of our situation. The first is to resample with replacement the measurements themselves, so that any individual measurement from the actual data may appear in any particular bootstrap data set multiple times — effectively strengthening that measurement by reducing its variance by a factor of the number of times it appears — or once, or not at all, keeping the total number of data points constant. Each modified data set is subjected to analysis with PEP, and the resulting parameter point estimates recorded. This approach has the appealing (to laser-rangers) heuristic interpretation of representing the realization of the various data sets that could have occurred if the experiments that collected them had experienced different weather conditions; superior weather during some observing sessions would have strengthened those observations, whereas inclement

weather during others would have resulted in those measurements vanishing from the data set. Since we presume that our conclusions about parameter values cannot depend on such a stochastic phenomenon, except to the extent that those conclusions are themselves uncertain, the dispersion of estimates of the parameter values derived from a large number of such resampled data sets reveals the precision with which the real data permits the parameter values to be determined. This approach is model-free, in that the data sets produced by the resampling do not depend in any way on what model is being used or what the values of the model parameters are.

The other approach to the bootstrap in a regression problem is to resample with replacement the normalized residuals, which are the differences between the measured ranges and the expectations of those measurements according to the best-fit model, divided by the measurement uncertainties. Thus if a dataset contained two measurements, one 3 units of its measurement uncertainty above the best-fit curve and the other 1 unit below it, the normalized residuals would be 3 and -1. From these residuals four bootstrap data sets could be constructed: one that is identical to the real data, one in which each data point receives the normalized residual of the other, one in which both have a normalized residual of 3, and one in which they both have a normalized residual of -1. The points retain their own measurement uncertainties, so the actual size of the resampled residuals in the units of the measurement depends on the point in question. The measurement values themselves are altered to produce these new residuals with respect to the best fit to the real data, and the model is then refit to the altered data. As in the measurements bootstrap, the spread of the values assumed by the parameters under repeated resampling is taken as a measure of the standard error of the parameters as estimated from the original data.

This type of bootstrap is similar to the measurements bootstrap in that it attempts to create datasets that are comparably plausible to the true data, but unlike the measurements bootstrap it does so from the point of view of an existing best-fit model, on which the method depends. If the residuals are dominated by known or unknown uncertainties in the measurements, then we expect the residual bootstrap to perform comparably to the measurements bootstrap, but if modeling issues dominate the residuals, then resampling those residuals produces datasets that are less plausible than the real data from which they are derived and which could perhaps never have been observed in fact. We should then expect the residual bootstrap to produce larger estimates of the parameter uncertainties than the measurements bootstrap under some circumstances. In particular, if under-modeling results in the presence of some signal in the residuals, this signal will be destroyed by resampling the residuals to a much greater extent than by resampling the measurements, and any parameter that was constrained by the presence of the signal will assume a wider range of values in the former case. Through this mechanism, comparison of the results of the two bootstrap approaches may point the way to deficiencies in the model.

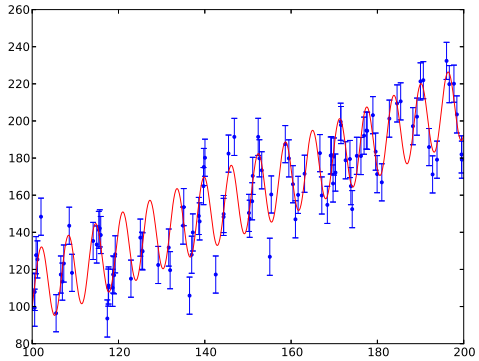

Figure 4.1: Scatter of simulated measurements about a trial function. In this example the measurements' scatter about the function is consistent with their uncertainties.

Table 4.1: Number of trials out of 1000 in which parameter estimates were within 1 and 2 sigma of the true values, uncertainties having been derived from the least-squares process.

Parameter	1σ	2σ
p_0	670	948
p_1	657	952
p_2	689	955

Table 4.2: Number of trials out of 1000 in which parameter estimates were within 1 and 2 sigmas of the true values, uncertainties having been estimated by bootstrap methods.

	Resampling method			
	Measurements		Residuals	
Parameter	1σ	2σ	1σ	2σ
p_0	658	932	664	953
p_1	673	938	663	953
p_2	682	950	661	950

As a demonstration, consider the function $f(x) = 10 + x + 20sin(x)$, $x \in (100, 200)$. Let's initially make 100 measurements of the value of this function at randomly chosen points in its domain. We will initially ascribe an uncertainty of 10 to each of these measurements and have that uncertainty be in fact appropriate, so that the scatter of the measurements about the true function f is given by a Gaussian distribution with mean 0 and standard deviation 10 (Fig. 4.1). Now we fit the measurements with the complete linear model $p_0 + p_1x + p_2sin(x)$. A working approach to estimating the parameters of this fit model must produce point estimates p_* and uncertainties σ_* for all p_i such that the distribution of $(p_T - p_*)/\sigma_*$ is Gaussian with mean 0 and standard deviation 1, where p_T is the known 'true' parameter value used in generating the measurements. Put another way, a successful method will produce point estimates that differ from the true value in accordance with the uncertainties that are ascribed to them. In this case,

in which the model is complete and the measurement uncertainties are accurate, we expect the least-squares method to be successful. Table 4.1 summarizes the results of 1000 simulations in which 100 points were generated and then fit in this way. Approximately sixty-eight percent of estimates for all parameters fell within one standard deviation of the true value, and ninety-five percent within two standard deviations, in accordance with a Gaussian distribution.

Now, how do bootstrap methods fare? We generated 1000 random data sets with the same characteristics as the one just described, each having 100 points with uneven sampling and scatter about the underlying function consistent with their uncertainties, and for each of the 1000 sets we obtained point estimates for the three model parameters from a least-squares fit but determined standard errors via 500 100-point resamples of the set, applying the least-squares process to get parameter estimates for each resample, and taking the standard deviation of the resulting 500 estimates for each parameter. We then repeated this procedure but with resampling of the residuals rather than the measurements. Table 4.2 shows the number of cases in which the resulting confidence intervals captured the true value of the parameters at the 1-σ and 2-σ levels. The resampling techniques performed approximately as well as the least-squares process.

Table 4.3: Number of trials out of 1000 in which parameter estimates were within 1 and 2 sigma of the true values, uncertainties having been derived from the least-squares process. The stated measurement uncertainties were half the RMS scatter about the generating function.

Parameter	1σ	2σ
p_0	389	690
p_1	387	674
p_2	399	695

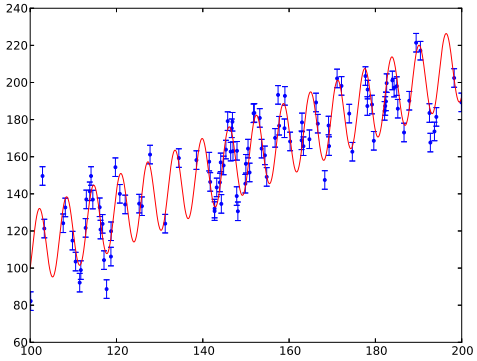

Figure 4.2: Scatter of simulated measurements about a trial function. In this example the measurements' scatter about the function is twice what would be suggested by their uncertainties.

Table 4.4: Number of trials out of 1000 in which parameter estimates were within 1 and 2 sigmas of the true values, with uncertainties estimated by bootstrap methods. The stated measurement uncertainties were half the RMS scatter about the generating function.

	Resampling method			
	Measurements		Residuals	
Parameter	1σ	2σ	1σ	2σ
p_0	650	947	700	959
p_1	644	946	711	959
p_2	686	954	676	950

Obviously we have described a case in which resampling is not the easiest way to produce valid parameter-estimate uncertainties. Consider now the case in which the scatter of the data about the function used to generate it is twice as great as suggested by the uncertainties we ascribe to the measurements (Fig. 4.2). The means by which the stated measurement uncertainties are used to determine the parameter confidence intervals when using the least-squares process leads us to believe that the parameter uncertainty estimates will scale with the measurement uncertainties, and indeed the results of 1000 simulations confirm that the 2-sigma estimate is effectively a 1-sigma estimate, capturing the true value 68 percent of

the time, as shown in Table 4.3. By comparison, resampling techniques are as effective as in the former case in which the ascribed measurement uncertainties were appropriate (Table 4.4). Clearly, as a method of determining standard errors, both bootstrap methods are much less sensitive than least squares to the quality of our assertions about measurement uncertainty.

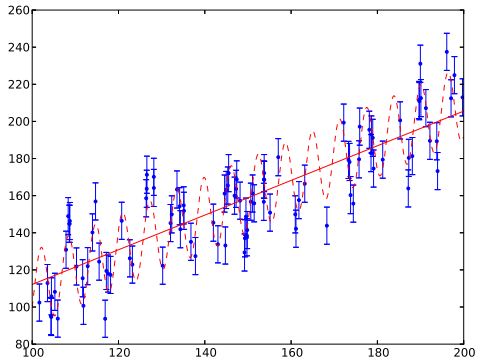

Figure 4.3: Scatter of simulated measurements about a trial function. In this example the measurements' scatter about the generating function (dashed curve) is consistent with their uncertainties, but a model is fit which does not include the sinusoidal term (solid line).

Table 4.5: Number of trials out of 1000 in which parameter estimates were within 1 and 2 sigma of the true values, with uncertainties derived from the least-squares process. The stated measurement uncertainties were consistent with RMS scatter about generating function, but the sinusoidal component was unmodeled.

Parameter	1σ	2σ
p_0	435	748
p_1	425	736

The previous example describes a case in which the model being fit accurately reflects that from which the data were generated, i.e. there are no unmodeled effects and the discrepancy between the measurement uncertainties and the residuals is due to underestimation of the former. What if instead the measurement uncertainties are

formally correct but a discrepancy still exists because the situation is under-modeled by the fitting function? Consider a case in which we use the same function as before to generate the data and ascribe measurement uncertainties consistent with the scatter (Gaussian with a standard deviation of 10), but our model is ignorant of the sinusoidal component, so that we estimate only the coefficients of the constant and linear terms (Fig. 4.3.) As before we then simulate 1000 datasets and determine the number of times the least-squares point estimate of each parameter was within 1 and 2 units of the least-squares estimate of parameter uncertainty of the true value, obtaining the results in Table 4.5. Evidently the least-squares process has poorly estimated the degree to which the parameter estimates differ from their true values, even though the unmodeled sine function does not *bias* the estimate of either the intercept or slope coefficients since its own mean value and slope are 0.

How would bootstrap methods fare? In this case, we expect that the resampling of the measurements will produce bootstrap data sets that are essentially plausible in light of the actual one. In any one instantiation some points will appear multiple times, effectively reducing their associated uncertainty by a factor of $\sqrt{n}$ at the cost of removing others entirely, but every measurement appearing in the sample will appear to be one that could have actually been made, if for example more data had been taken at the corresponding value of the independent variable. Conversely, resampling of the residuals seems certain to create data sets full of measurements that could never actually have been made, by ascribing the large positive residuals associated with points near the sine crest to other points that have large negative residuals in actual fact due to being near the trough. Because these discrepant residuals owe their existence to an unmodeled effect and not to any actual data deficiency, for such a measurement to have actually been made is, at a minimum, highly unlikely. Since the data sets derived from the residual

bootstrap appear to span a wider space of (im)possibilities in this way, we might expect the resulting parameter uncertainty estimates to be too large. However, this is not observed. Again both approaches to resampling are successful in producing realistic uncertainty estimates (Table 4.6). Why does the residual bootstrap appear to work just as well? One way of looking at it is that the modified measurements produced by resampling only look implausible if the true generating function is known, whereas the model we are actually fitting may be said to be ignorant of its own deficiencies, and so from its 'perspective' the modified measurements look just as plausible as the originals. The discrepancy between the model's 'perception' of plausibility and the fact of the matter as seen by someone who knows the generating function grows with the model deficiencies, and so the residual bootstrap may be viewed as incorporating the degree of under-modeling in a natural way. Viewed from this perspective, the good performance of the residual bootstrap is less mystifying.

Table 4.6: Number of trials out of 1000 in which parameter estimates were within 1 and 2 sigmas of the true values, with the uncertainties estimated by bootstrap methods. The stated measurement uncertainties are equal to the scatter about the generating function, but the sinusoidal component is unmodeled.

	Resampling method			
	Measurements		Residuals	
Parameter	1σ	2σ	1σ	2σ
p_0	650	940	658	942
p_1	651	934	649	939

As it happens, there is also a least-squares route to accurate parameter uncertainties in this case. As noted, the stated measurement uncertainty and RMS scatter about the generating function are both 10. The RMS of the unmodeled sine wave is $20/\sqrt{2}$, and so we might suppose that the RMS scatter of the measurements about the best-fit model (which excludes the sine term in this example)

would be the quadrature sum of these: $\sqrt{10^2 + (\frac{20}{\sqrt{2}})^2} = 10\sqrt{3}$. And indeed if we inflate the measurement errors by $\sqrt{3}$ in order to comport with this result the resulting estimates of parameter uncertainty are consistent with the spread of the point estimates (Table 4.7). This isn't because the measurement uncertainties are underestimated; in this case, they are exactly right. Rather, this is a demonstration of a principle cited earlier: that if measurement uncertainties and scatter about a modeling function are not in agreement, a plausible route to realistic parameter uncertainties is to inflate the measurement uncertainties until the tension goes away, even if part or all of that tension is ascribable to model defects rather than actual problems with the measurements. The discomfiting implication for experimenters is that measurement precision is largely wasted if it is considerably better than the precision achievable by the model by which the data are analyzed, except to the extent that it serves as a motivation for further model development. This is the present position of APOLLO.

Table 4.7: Number of trials out of 1000 in which parameter estimates were within 1 and 2 sigma of the true values, with uncertainties derived from the least-squares process. The stated measurement uncertainties were consistent with RMS scatter about generating function, but the sinusoidal component was unmodeled. However, the measurement uncertainties were then inflated by $\sqrt{3}$ to correspond with total scatter, and the resulting parameter uncertainties are valid.

Parameter	1σ	2σ
p_0	677	955
p_1	671	953

The residual bootstrap does have the potential to present misleading uncertainty estimates in the event that an unmodeled aspect of the physical situation creates a signal in the residuals that constrains the estimate of another model parameter. Consider the same function we have used in the preceding examples.

We now also want to estimate the period of the sine wave, but we incorrectly believe that we know its amplitude is 10, instead of the actual 20. This could be due either to mismodeling of an actual effect or to a systematic measurement issue. A sinusoidal signal will be evident in the residuals as a result, but the ability of the fit to constrain the period of the function should be unaffected (Fig. 4.4). Note that the model is no longer linear in the parameters, but least-squares fitting via numerical methods is still possible. Resampling of the residuals will now create data sets in which the sinusoidal aspect of the generating function is much less evident, leading to reduced ability to constrain its period and hence to overestimated uncertainly ascribed to that parameter, as shown in Table 4.8. If the addition of a single parameter to the fit puts residual-bootstrap uncertainty estimates in tension with those derived from the measurements bootstrap when they were previously compatible, it may be that the new parameter was constrained by a signal in the residuals in this way. In a model with many parameters, a broadening of the confidence interval of one may easily be reflected in others due to correlation.

In situations like a fit to solar-system data in which there are many periodic effects or a strong baseline is valuable for other reasons, the measurements bootstrap has the potential to overstate parameter uncertainties on account of having sacrificed some degree of coverage. The design of the resampling method aims to produce bootstrap datasets with similar constraining power to the original by eliminating some measurements but strengthening others proportionally. This is fine if we are trying to find the uncertainty in the mean of a number of measurements, but when trying to estimate effects that vary in a complex way with the independent parameter, two points are far better than one, even if the 'one' has half the associated uncertainty. This may be especially important if the sampling was already uneven,

Table 4.8: Number of trials out of 1000 in which parameter estimates were within 1 and 2 sigmas of the true values, uncertainties having been estimated by bootstrap methods. The stated measurement uncertainties are consistent with the scatter about generating function, but the amplitude of the sinusoidal component was mismodeled. $p2$ here indicates a new parameter modeling the frequency of the sine. Despite the mismodeling this parameter is constrained by the signal in the residuals, which is considerably attenuated by the residual bootstrap, leading that method to overstate the uncertainty in that parameter. The measurements bootstrap, by comparison, provides reliable standard errors for all parameter estimates in this case.

	Resampling method			
	Measurements		Residuals	
Parameter	1σ	2σ	1σ	2σ
p_0	684	940	662	942
p_1	656	943	651	929
p_2	677	954	**990**	**1000**

as is the case with lunar laser-ranging measurements. The APOLLO experiment, for example, is unable to take data at new moon, poorly able near full moon, and frequently shut out during the summer months due to engineering work and weather patterns at Apache Point Observatory.

Regardless of the methods used to assign standard errors to parameter estimates, the existence of physical effects not addressed by the model creates the possibility that the signal of these effects in the data could be 'soaked up' at the fitting stage by unphysical adjustments to model parameters. For example if in the example just discussed a term of the generating function were proportional to $\sin^2 x$ rather than $\sin x$ and its amplitude were mismodeled, the resulting discrepancy would be made up as well as possible by adjustments to the constant term, since the mean of the mismodeled function is not zero. The resulting offset in the point estimate of the constant term could never be made whole through uncertainty estimation, as it is not possible to account after the fact for what has gone unaccounted for at the modeling stage. Particularly vulnerable to this possibility are parameters

whose total effect on the residuals is comparable to the scatter of the measurements about the best-fit model, as it is then challenging to assert with confidence that unmodeled aspects of the physical system and systematic effects in the data are not producing apparent signals that may be absorbed by proportionally large changes in the parameter value. This effect is exacerbated for parameters whose impact is seen principally or exclusively in one portion of the dataset, as systematic and mismodeled components are unlikely to mimic the impact of adjusting a modeled parameter across multiple types of measurements.

This type of parameter-estimate bias was likely implicated in PEP estimates of higher-order coefficients of the lunar potential, a rare case in which the true values of a set of parameters were known to considerably higher precision than could be inferred from PEP results due to the recent results of the GRAIL lunar-gravity experiment. Of the datasets incorporated by PEP, only lunar laser-ranging data is sensitive to these parameters. PEP estimates of the relatively significant quadrupole moment of the potential (J2) were in basic agreement with the GRAIL value, but estimates of some of the less impactful, higher-order coefficients could not be reconciled with GRAIL results (FIg. 4.5). The effect on the goodness of fit of fixing the lunar gravity model at the GRAIL values was found to be measurable but not prohibitive, implying that the signal being absorbed by the unphysical estimates was small compared to the overall scale of the residuals. Ultimately it was decided to use the GRAIL values in future solutions rather than fitting for them. This reduces the goodness of fit, but whatever signal was previously soaked up by mis-estimating the gravity model now awaits attribution to its actual cause or obsolescence at the hands of a rescaling of the measurement uncertainties.

Because the science parameters at which our efforts are directed are constrained by all of the datasets, we have some reason to be confident that we are

protected from mimicry of their signal by unmodeled effects. Nevertheless, the absorption by a scientifically interesting parameter of even a signal or systematic small in comparison to the overall impact of that parameter has the potential to produce deviations from nominal parameter values that might be judged statistically significant, if still objectively small. This ought to be a motivation to bring model precision in line with what is possible experimentally.

4.4 PPN formalism

The model of the solar system to which ranging data will be fit must incorporate the physics of gravitation in a parameterized fashion in order to both express our null hypothesis as embodied by general relativity and to permit and quantify deviations from that hypothesis if the data appear to demand it. Fortunately a such a parameterization for metric theories of gravity was made early in the history of computational solar-system modeling, by Ken Nordtvedt and Clifford M. Will in papers of the late 1960s and early 1970s [46, 47, 48]. The resulting construct is called the parameterized post-Newtonian formalism, and it is appropriate for the slow-motion, weak-field case of the planets and their satellites. In the PPN framework, the coefficient of each term in the metric is replaced by a free parameter; therefore metric theories of gravity that differ in their physical implications will also be associated with different values of these parameters.

In total there are 10 PPN parameters, and PEP is currently able to fit for two of them. The first is β, which indicates the quantity of curvature induced by a unit rest mass; its value in general relativity is 1, which represents our null hypothesis. The second is γ, standing for the nonlinearity of gravitation, or the extent to which gravitational mass-energy itself gravitates; its value in general

relativity is also 1. The other PPN parameters are 0 in general relativity and are not fit for by PEP in its present incarnation. Non-zero values of these parameters would indicate the existence of preferred-frame effects or the violation of one or more of the conservation laws (energy, momentum, and angular momentum). Will [49] gives a thorough treatment of the PPN formalism as well as many other subjects of interest.

4.5 Application to ranging data

Ascribing uncertainties to parameters estimated from fits to ranging data has been a fraught question and has provoked a range of responses. The modelers at JPL (e.g. [50]) prefer to add to the uncertainties of measurements whose precision considerably exceeds the fitting capability of the model a root sum square term in order to obtain in the final analysis a reduced χ^2 value close to 1. This amounts to an assertion that the discrepancy between measurement uncertainties and residuals is due principally to model defects or other factors that impact each point equally such that every measurement is subject to an RSS uncertainty adjustment of the same size. However, it has the effect of downweighting extremely precise APOLLO measurements by approximately a factor of 15, giving measurements of disparate quality effectively equal weight and affording them no more impact on the fit than LLR measurements of previous decades whose actual uncertainty may be on the order of ten times larger. In a fit to the JPL model using all historical LLR data in addition to that from APOLLO, the APOLLO residuals show separation by reflector on any given night, suggesting the lunar orientation is improperly modeled at some level. When the APOLLO data is fit by itself at full weight, this separation by reflector largely disappears, implying that the full-weight high-

precision measurements have the potential to correct the orientation, but at the expense of worsening the fit to other measurements in the LLR dataset. It seems likely that the lunar-orientation model is not currently able to maintain the desired level of fidelity over a the long span of measurements. This being understood to some degree, applying an RSS term to secure a favorable reduced χ^2 perhaps does not best exploit the available level of measurement precision, but it has the virtues of being straightforward and conservative.

Nevertheless we feel that this approach has several shortcomings in the context of a fit to all available ranging measurements. Technically a suitably sized term should be applied to the uncertainties in every data series in order to bring them in line with the scale of the residuals. It may even be appropriate to scale down the uncertainties of data series that are 'overfit' by the model. As the goodness of fit depends on the dataset and parameters being adjusted, the needed values are subject to change and would have to be re-estimated frequently. Moreover, portions of the data that were considerably downweighted initially would have less influence in a fit of the re-weighted data and so would probably still not be fit at a level consistent with even their inflated uncertainties, leading to an ambiguous iterative process in which the data is repeatedly reweighted and refit.

Others (e.g. [51]) appear to fit all data at full weight but inflate the uncertainties of the resulting parameter estimates as given by the least squares process by a common factor seemingly derived from some combination of the residuals-uncertainties discrepancy and comparison of the results of different solutions. As far as the parameter uncertainties go, this is equivalent to increasing the uncertainties of all the measurements by the same factor prior to fitting. (Whether such factor is applied before or after solving would presumably affect the point estimates, however). This has the effect of inflating the uncertainty of even those portions

of the data where the residuals are already consistent with the error bars, and by extension of overestimating the uncertainty of parameters principally constrained by such data sets, while perhaps understating the uncertainty of parameters principally constrained by measurements whose normalized residuals are greater than average. Our interest in practice being mainly directed at a small number of the parameters, it is hard to say in any individual case whether the inflation factor is too large or too small, with the result that apparent but arguably marginal detections of non-nominal values – and only marginal detections are really conceivable – are likely to be ascribed to mismodeling of correlated effects rather than viewed as stakes in the ground in need of serious follow-up, calling into question the power of ranging measurements to advance the aims at which they are directed.

Within the PEP collaboration, a light version of resampling has previously been used. A distinction must be made here between 'science parameters' at which the analysis effort is directed and 'nuisance parameters' that must be estimated for the procedure to work but are not of interest to the operator. This is a subjective distinction; any parameter can be a science parameter if we decide we are interested in it. In a typical approach, a solution adjusting both types of parameters is iterated until it converges. The data set is then truncated or modified in various ways to produce bases for alternative but plausible additional solutions; for example half of a major subset of the data might be used at a time, or another large subset might be considerably downweighted. Various combinations of the science parameters are also considered, adjusting all of them or various subsets that are judged to be appropriate for simultaneous estimation. Using the fully converged solution as a starting point, another solution is then executed for each combination of the data and parameter sets so devised, resulting in an ensemble of perhaps several dozen alternative solutions.

From this point, two avenues have been taken. In the first, a typical excursion for each nuisance parameter is determined from the ensemble, and this is then extrapolated into an impact (defined to be positive) on each science parameter by means of the correlation between the two. A distribution of these impacts for all nuisance parameters is then constructed for each science parameter. At this point there is some degree of uncertainty on how to proceed. It is not realistic to treat the quadrature sum of all the impacts at the total expected excursion of the science parameters, as in any particular realization of the nuisance parameters, some will see positive adjustments relative to the original converged solution and some negative. It has been suggested that the aggregate impact on the science parameter is probably not larger than the single largest impact implied by the excursions of the nuisance parameters, for which we will substitute the mean of the five largest impacts in order to increase the stability of this measure. Another approach is to take the RMS of all the impacts in the distribution, which is more stable still but may underestimate the parameter uncertainty as many nuisance parameters are correlated barely if at all with those of principal interest. These methods are denoted by 'By Correlations, Largest' and 'By Correlations, RMS' in Table 4.9.

Taking the other route, the values assumed by the science parameters in the ensemble of solutions are aggregated and some measure of their spread taken as a standard error. This is easier to implement and understand than the procedure just described, and it is not unlike the bootstrap in principle. However, the relatively small number of different solutions as well as the arbitrariness with which the variant data sets were constructed tend to work against the reliability of such estimates. Furthermore, the introduction and removal of subsets of the science parameters between solutions in the ensemble imply that the estimates of those parameters

are not drawn from a common distribution, as the corresponding correlations are alternatively present and absent. This method is denoted by 'By Observed Range' in Table 4.9. In any event, because there is some degree of arbitrariness in the way the data is subsetted and downweighted, and in which parameters are included and excluded to create an ensemble of solutions, no approach to the uncertainty problem previously used by the PEP collaboration produces estimates that can be reproducibly compared with results from other modeling efforts.

The dataset usable by PEP for parameter estimation comprises a variety of solar system ranging measurements. In the broad brush, these can be broken down into an inner-planets dataset consisting of radar ranges to Mercury and Venus; a moon dataset consisting of the normal points produced by lunar laser-ranging stations in Texas, France, Hawaii, and New Mexico; and a Mars dataset consisting of radio ranges to a succession of Mars landers and orbiters including Viking 1 and 2, Mariner 9, Mars Pathfinder, Mars Global Surveyor, and Mars Odyssey. The Mars dataset is by far the largest by number of observations, although the lunar ranging data possesses both the longest baseline and the most precise observations as well as the greatest sensitivity to the equivalence principle and $\dot{G}$. The entire dataset is divided for PEP purposes into approximately 20 observation libraries. Each such 'obslib' is further subdivided into one or more observation series, each of which represents a consistent operating state for the ranging operation to which the obslib pertains. Thus, observations within a series may be treated as a unit for example for the purpose of applying a range-bias parameter. For this reason we have conducted bootstrap procedures at the series level, creating new data series of the same size as the originals by sampling with replacement among the observations in that series. Vis-a-vis resampling at the level of the entire dataset, this reduces the potential for the measurements bootstrap to create single unreasonably strong

points by including a measurement many times in a particular resample, and in the case of the residual bootstrap it avoids the application of normalized residuals from one series to another in which their original scale may have been quite different.

Within the series of the obslibs are the records representing individual measurements. These include the timestamp, round-trip time, and uncertainty of the measurement. Prior to formation of normal equations, PEP supplements these records with the appropriate values of the partial derivatives as determined from the ephemeris. Resampling capability is not native to PEP, but a utility was already extant to convert the obslibs from their usual binary format to text and back. As part of this dissertation effort, software was written to implement both versions of the bootstrap. In the case of the measurements bootstrap, it is necessary to parse the text form of each obslib to identify the individual records and then write a new version of the obslib with new records drawn randomly with replacement from those in each series. For the residual bootstrap, we extract all of the residuals in each series and divide by the associated measurement uncertainty to normalize them. For each record we then select a random normalized residual from those associated with that series and multiply by the uncertainty of that measurement in order to generate a new residual and modify the round-trip time and residual components of the record accordingly. The time required for the resampling procedure is a strong function of obslib size but is comparable to that required by PEP to form normal equations from the obslibs and is not prohibitive. Much of the contents of the obslibs must be stored in memory throughout repeated resampling and solution procedures as the bootstrap distributions of the parameters are built up, which may test the memory capability of some systems. The text-format size of the entire dataset is several gigabytes. Once the resampled obslibs are created, they are converted back to binary, and PEP uses them to form normal equations which are then solved for

parameter adjustments. The resulting parameter values are accumulated and their standard deviation taken in the end to serve as the estimate of standard error for each parameter. Distributions of the estimates of each parameter are observed to be generally Gaussian in form (although see below for an exception), and when working with the entire dataset 100 resamples strikes a desirable balance between statistical sufficiency and runtime (approximately overnight). With any dataset that does not include the largest obslibs, representing the ranges to Mars Global Surveyor and Mars Odyssey, 1000 resamples is no obstacle and a much cleaner distribution can be obtained.

The residuals of a series often appear to show some type of structure, which is obliterated by the residual bootstrap as described earlier; see Fig. 4.6.

Once the machinery to implement bootstrap methods has been devised, its application to any particular solution is straightforward, without the need to estimate the size of a suitable RSS term for different subsets of the data presently in use, or for the aggregation of the results of solutions using somewhat modified inputs.

As proof of concept we used PEP to perform a fit using only the APOLLO dataset up to 2012, fixing the science parameters at their nominal values. For purposes of this demonstration, we treated six parameters of the system as 'science' parameters and estimated their uncertainties using various methods (Table 4.9). We first degraded the uncertainty of each normal point with an RSS term of 350 ps, equal to about 5 cm of one-way range, in order to get a χ^2 value close to 1 for this particular set of data and adjusted parameters.

Clearly the addition of an RSS term inflates the uncertainty estimates in a very stable manner. By comparison, the methods of correlations and of observed range produce uncertainty scalings that vary considerably from parameter

to parameter and are generally significantly larger than what is suggested by the RSS-term strategy. This may be attributable at least in part to the fact that the subsets of the data used the generate the ensemble of solutions needed by these methods necessarily have less constraining power than the data taken as a whole. Both bootstrap methods produce uncertainty scaling estimates that are generally in keeping with the results of the RSS-term approach and are fairly stable across the parameters listed. The measurements bootstrap indicates somewhat larger scaling factors than the residual bootstrap, especially for the parameters of the lunar orbit, an effect attributable to the sacrifice of baseline inherent in the former approach.

In order to secure the most comprehensive constraints on parameters of interest it is desired to perform a simultaneous fit to as much data as possible while likewise adjusting as many parameters as possible to expose their correlations. In PEP, unfortunately, true convergence is currently ultimately not observed for the global data and parameter set, and for many subsets thereof. Typical behavior involves a gradual reduction in the size of parameter adjustments over the course of many iterated solutions, arriving in the end at a regime in which a subset of parameters experience essentially identical adjustments in solution after solution, with correlated parameters being dragged along as necessary. This type of behavior is not expected given the principles of iteration as described previously and is generally ascribed to insufficient precision in the partial derivatives (although see the following section) and is an active area of PEP development. While an eventual cessation of these adjustments after hundreds of solutions may occur in some cases, it is not clear whether the values to which the parameters have 'converged' as a result are grounded in physical reality. For example, as previously noted the lunar gravity coefficients converge to values in many cases compatible with their GRAIL value in a solution series using only LLR data, but if adjusted in a global solution

series, they gradually drift to unphysical values even though no data other than LLR constrains them.

Fig. 4.7 shows the results of on-axis χ^2 exploration originating from an unconverged solution using all datasets and adjusting all parameters. The solution has been subjected to dozens of iterations prior to the exploration but many parameters are experiencing ongoing adjustments, generally of less than a single unit of the parameter formal uncertainty per iteration. To perform such an exploration, we begin with the current values of all parameters and forcibly displace the value of one parameter, thereby moving along the axis of that parameter in parameter space (hence 'on-axis'). PEP can then be used to calculate the value of total χ^2 associated with the new parameter values, the χ^2 surface can be mapped out in one dimension. The results for the earth-moon barycenter mass show the surface to be nearly flat along that axis, indicating that that parameter may be near a minimum (subject to ongoing changes in the value of correlated parameters). The other three depicted parameters, however (the semimajor axis of the lunar orbit, a relativity scaling coefficient, and the rate of change of G) show an apparently linear dependence of χ^2 on the parameter value, indicating that the minimum is not nearby despite the small ongoing adjustments to the parameter values, with the caveat that the actual direction of ongoing iterative travel in parameter space in not on-axis but involves virtually every parameter.

In Fig. 4.8, resampling distributions for two parameters of interest can be seen. The unconverged global solution described in the previous paragraph served as the basis, and the resulting residuals were resampled 250 times. Because lack of convergence has been seen to destabilize the resampling procedure, the implied standard errors should be taken with a grain of salt, but they are unlikely to be unrealistically small. The limit on $\dot{\mathrm{G}}$ would be about four parts in 10^{14} per year

for GMVARY, which would be quite stringent compared for example to the limit claimed by Williams et al. [52], but in the absence of convergence of course there are no confidence intervals to be had, and furthermore since only one solution is performed on each resampled data set the 'real' resampling distribution could be much broader if convergence is slow. At any rate, the prevailing lack of consistency in the determination of standard errors renders direct comparison suggestive at best.

In the absence of compelling convergence for something close to the total parameter and data set, point estimates for the parameters of interest under those conditions are not available. Point estimates are obtainable from a subset, for example the LLR data alone, but our experience with the lunar gravity parameters cautions us against putting to much stock in values unconstrained by multiple types of measurements. Furthermore, we have observed that resampling approaches appear unstable when based on a solution that is not fully converged, i.e. when the initial values of the parameters are not associated with a minimum of the χ^2 surface associated with the unresampled measurements.

4.6 Identification of problematic measurements

Considerable effort has been expended over the years in weeding out from the dataset normal points representing spurious detections or presenting other problems that make them unsuitable for use in a solution. Despite these efforts, we find evidence from resampling that some such points are still present and that a subset of these may have no means of identification other than through observation of their effects on the solution. Identification and removal of these points has the potential to increase sensitivity to target parameters and to ameliorate observed

convergence anomalies.

Under the measurements bootstrap, the N points in a series are replaced by N points drawn randomly with replacement from the same series. Thus, the probability that a given point does not appear in a particular resample is given by $(\frac{N-1}{N})^N$. As N grows, this value quickly approaches $\frac{1}{e}$, which is to say that each point appears in approximately 2/3 of resamples of the series with little dependence on how large the series is (Fig. 4.9). If within a series there is one point that, when fit with a particular parameter and data set, strongly affects the best-fit values of one or more parameters, this will be evident through the bootstrap distribution of the values of that parameter, as one-third of the realized values will be drawn from a different distribution than the other two-thirds. This manifests as bimodality of the distribution of parameter estimates, as in Fig. 4.10. An apparent instance of this effect can be followed up by plotting, for each data point, the mean of the best-fit values of that parameter that occurred when that data point was present in the resample. This method was employed to identify a point in the lunar laser-ranging dataset to which an implausibly small uncertainty had been accidentally ascribed, firmly fixing the coordinates of the corresponding ranging site whenever that normal point was present in the fit data. The record in question was excised from the dataset, and the distribution of the site coordinates in subsequent resampling tests were satisfyingly unimodal.

A somewhat more ambiguous case arose in the course of attempting solutions using only the Mercury and Venus radar-ranging data. As is observed in a variety of settings when using PEP, the solution did not fully converge. Most parameters appeared to reach a stable final value, but a small number experienced adjustments of consistent size in succeeding solutions, being the mass of the earth-moon barycenter and its argument of perihelion. Resampling of the measurements was

attempted using this almost-converged solution as the basis, and the distribution of the parameter estimates was observed to be approximately normal, except for those parameters which had not fully converged, in which the presence of an overlapping bimodal distribution with a two-thirds, one-third split seemed possible (Fig. 4.11). As before, a closer look identified a single point whose presence determined to which distribution the subsequent parameter estimate belonged. However, unlike in the previous case there was nothing to our eyes wrong or remarkable about this point. Nevertheless, when it was removed from the data set, not only was the irregular distribution of those two fit parameters resolved, but subsequent solutions using the radar-only dataset saw complete convergence in the value of all parameters. It seems likely that the measurement in question was implicated in both the resampling and convergence problems, but the mechanism is not clear, as the cause of lack of convergence is not fully understood and no one issue may lie at its root in every case. It may be that there is actually some problem with this one record that is not obvious to the eye but emerges in the fit. On the other hand it may be that an issue elsewhere in PEP interacts badly with data points having certain characteristics under certain circumstances but there is nothing actually 'wrong' with these points. Even in the latter case, the removal of a small portion of the dataset would be a small price to pay for improved convergence behavior, but the resampling-based method of identifying such records is something of a blunt instrument and largely depends on there being at most one problematic measurement in a series, as the rate of simultaneous non-occurrence in a bootstrap sample of two or more points is low enough that it would be difficult to notice that anything was amiss. In principle, a binary search could be used to repeatedly subdivide the existing series in an attempt to isolate any additional such points if they exist, but in the absence of a compelling theory tying data problems to

convergence failure in the general case, it has not seemed worthwhile to invest the considerable effort required to implement such an approach.

This chapter, in part, is currently being prepared for submission for publication of the material. Johnson, Nathan H.; Chandler, John F.; Murphy, Thomas W. The dissertation author was the primary investigator and author of this material.

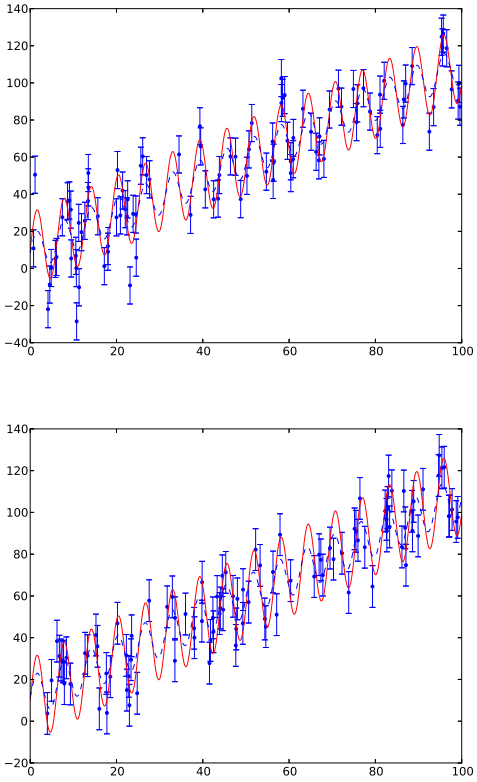

Figure 4.4: Simulated measurements are scattered about a generating function (solid curve), but are fit with a model in which the amplitude of the sine wave is mismodeled (dashed curve). The measurements' scatter about the generating function is consistent with their uncertainties, and a least-squares process is able to faithfully determine the values of the fit parameters, but the mismodeled element leads to a signal in the residuals (top). When these residuals are then resampled, the destruction of this signal makes identification of the period more difficult (bottom). The domain has been shifted to the origin in order to obviate the additional need for a phase term.

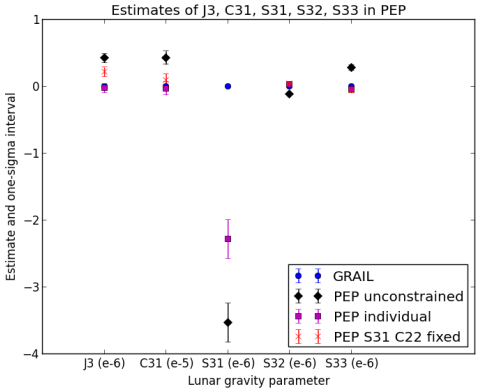

Figure 4.5: Values of five of the lunar gravity coefficients as determined by PEP and by the GRAIL experiment, with the uncertainties on the PEP values being derived from resampling methods. The PEP values and 1σ confidence intervals are shown for the case in which all eight gravity coefficients are adjusted ('unconstrained'), each parameter is estimated individually ('individual'), and when only S31 and C22 are fixed. The depicted uncertainties are about 10 times larger than the PEP formal values, but still many results are not reconcilable with the GRAIL values, presumably due to these parameters soaking up unmodeled effects to some degree.

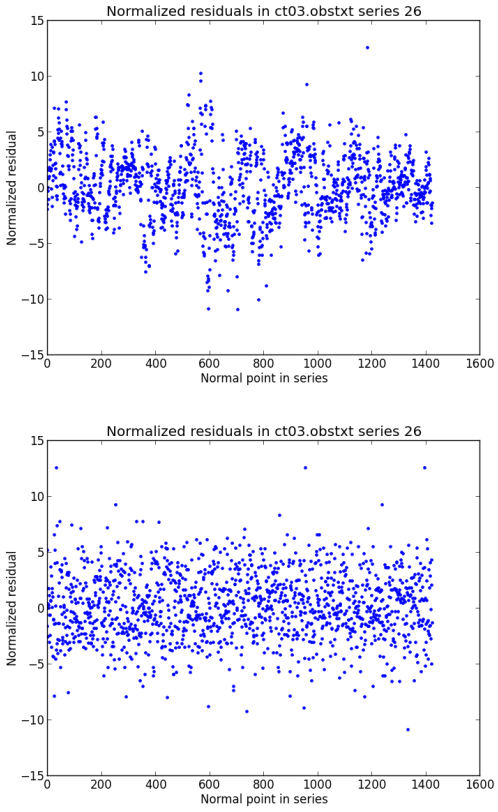

Figure 4.6: Residuals of a series of lunar ranging data from the MacDonald Laser Ranging Station in Texas, before (top) and after application of the residuals bootstrap. The signal remaining after the actual data has been fit as well as possible is removed by the resampling.

Table 4.9: Uncertainties in the estimates of 6 parameter values using different methods.

Method	APORAD	MEMBARY	MMOON	AMOON	EMOON	IMOON
PEP Nominal	1.159e-06	2.087e-17	1.158e-09	2.597e-14	1.300e-11	9.639e-09
PEP RSS 350 ps	10.16	10.94	11.22	10.74	10.93	11.80
By Correlations, RMS	7.58	12.28	5.59	18.49	22.21	28.99
By Correlations, Largest	13.43	22.35	9.61	37.20	45.20	58.12
By Observed Range	34.52	67.08	50.63	20.77	24.47	23.33
By Measurements Bootstrap	13.16	14.90	12.80	17.27	17.64	13.91
By Residual Bootstrap	12.40	13.21	12.44	11.67	11.88	11.88

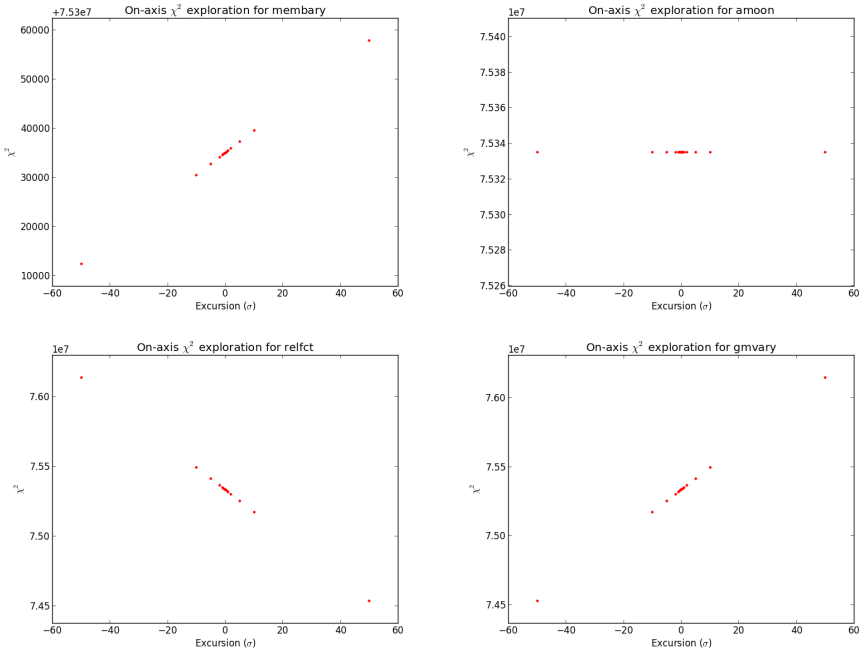

Figure 4.7: On-axis χ^2 exploration in the vicinity of the current parameter values of a solution unconverged after dozens of iterations. Ongoing adjustments to the parameters are less than a few increments of the parameter standard deviation (x-axis) per iteration. Only the indicated parameter is displaced in each plot, with membary being the earth-moon barycenter mass, amoon the semimajor axis of the lunar orbit, relfct a coefficient of all relativistic terms in the equations of motion, and gmvary the rate of change of G. Clearly for some parameters a χ^2 minimum is not nearby, the surface appearing locally flat, in that χ^2 evolves linearly in the parameter value. Ongoing adjustments do appear to be in the direction of the minimum.

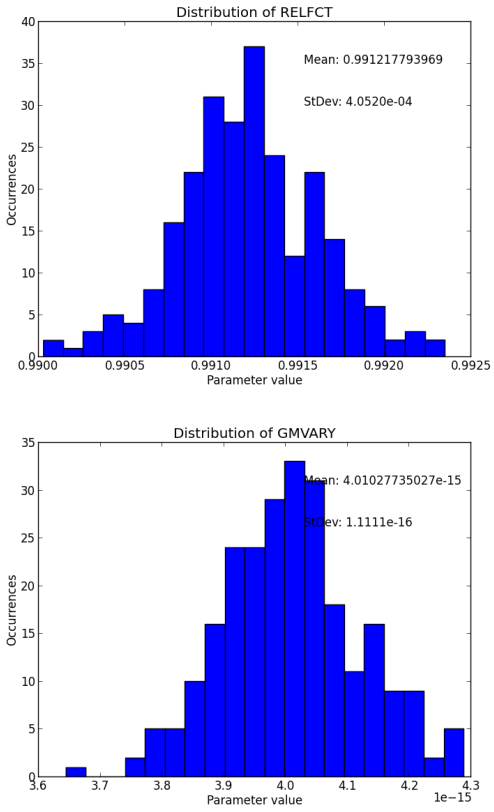

Figure 4.8: Distribution of the values of the relativity coefficient and variation of G under residual resampling on the basis of an unconverged solution. Although the means of these distributions are not to be taken as the point estimates, the use of the standard deviations of these distributions as standard errors for the point estimates arising from the basis solution would indicate a detection of non-nominal values, although obviously such a 'detection' is meaningless in the absence of convergence. At the time the solution series was halted, both of these science parameters were moving in the direction of their nominal values.

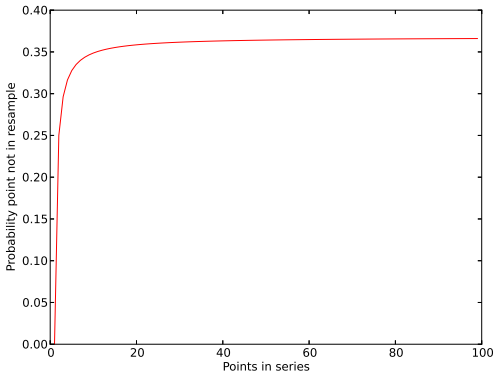

Figure 4.9: The probability of a given measurement not appearing in a bootstrap resample of its series is a function of the size of the series but quickly approaches a value of $1 - \frac{1}{e}$.

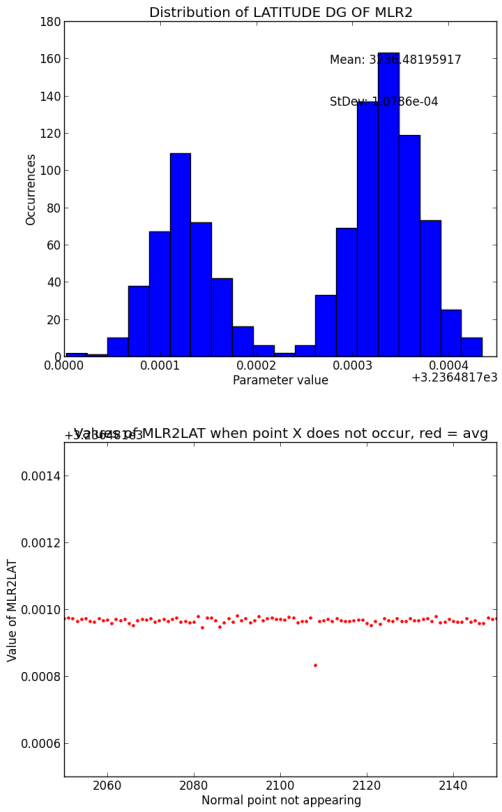

Figure 4.10: The estimate of the latitude of a Texas observing station was found to be bimodally distributed when the LLR normal points taken by that station were resampled 1000 times (top). The fact that about two thirds of the values appeared to fall in one distribution and the other third in a separate distribution led to suspicion that a single 'bad' normal point was present in one of the data series. Follow-up analysis showed that the best-fit value of the same parameter depended greatly on the presence or absence of the 2108th point (bottom).

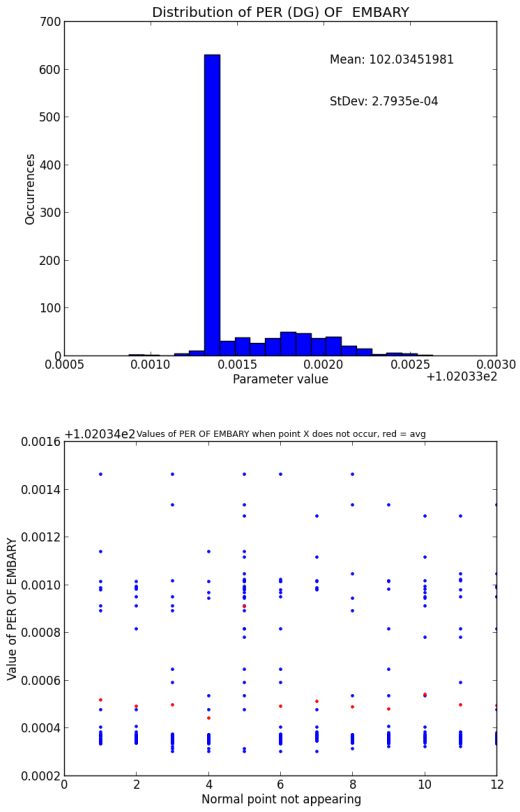

Figure 4.11: Resampling of the radar-ranging measurements 1000 times produced a distribution of the value of the earth-moon argument of perihelion that seemed to be divided into a narrow distribution representing about two thirds of the estimates and a broad one representing the rest (top). Observing the values assumed by this parameter when each point was not in the bootstrap identified a single point whose absence precluded the narrow distribution (the fifth, bottom).

Chapter 5

Future work

5.1 APOLLO experiment

The LLR station at Apache Point Observatory has been producing millimeter-precision normal points for nearly a decade. This level of range uncertainty, which is ascribed to the measurements by the data-reduction pipeline and used in a fit by PEP or another model, is a affirmable to some degree by the single-channel reduction comparison described in Section 3.2.1. Whether the normal points are also accurate, in the sense of faithfully representing the telescope-reflector distance at the reported time, is more challenging to determine. In practice it is not too problematic if the ranges over- or understate the actual distance by a consistent amount, as such a systematic offset will be fit out by a range-bias parameter in PEP or the equivalent. More concerning is the long-term stability of the range bias, whatever its sign and magnitude.

The JPL modeling team has historically inflated APOLLO normal point uncertainties by means of a 15 mm RSS term to secure a value of reduced χ^2 close to unity, and APOLLO residuals in subsequent fits also using the rest of the LLR

dataset have shown a clear separation by reflector within a night's ranges, indicating that the lunar orientation is mis-specified by the fit. When instead APOLLO data is fit at full weight, either by itself or with the other LLR normal points, this separation disappears to within a factor of 2 of the nominal APOLLO uncertainties, an indication of stability over the span of an hour. Less comforting were the results of a test in which a second local corner cube was positioned at a measured distance from the permanent version that provides the fiducial return during normal operation, and the distance between the two measured by the observed difference in return times. The separation as determined by ranging has been observed to conflict with the known distance at the 5 mm level, with the discrepancy dependent on the separation between the corner cubes. We attribute this behavior to the electromagnetically noisy environment associated with the laser fire, which may interfere with the detector-readout electronics. The measurement error has been seen to be stable over the timescale of an hour, but the long-term stability of the offset is again uncertain.

In order to ensure the quality of APOLLO data, an absolute calibration of the system is planned, in which pulses of light separated by a precisely known interval comparable to the lunar round-trip time as determined by a cesium reference will be injected into the same optical path followed by returning lunar and fiducial photons. The pulse separation as reported by the APOLLO instrumentation will then be compared to the known true value. Periodic repetition of this process in the presence of the interference generated by the APOLLO laser will establish the stability of the range bias, or lack thereof, and provide a calibration correction to the data subsequently collected.

5.2 Planetary Ephemeris Program

PEP development remains an area of activity and, it is to be hoped, will ultimately yield a path to improved constraints on gravitational parameters making use of the full precision of APOLLO data. In terms of the physical model, the last few years have seen upgrades to the handling of earth tides, precession/nutation, and atmospheric propagation delay. The coefficients of the lunar gravity model were formerly fit by PEP with considerable success, but the more precise values furnished by the GRAIL experiment are now generally used instead. An ongoing comparison with two European modeling efforts based on simulated data has the potential to yield valuable insights into model deficiencies for everyone involved.

In a practical sense, the convergence problem poses a serious challenge to the realization of science-parameter constraints. Convergence is observed using subsets of the data and parameter sets, but due to likely correlation of model parameters with unmodeled effects in these restricted contexts, the point estimates that arise under such circumstances are properly viewed with suspicion. Although specific 'bad' data points have been determined to prevent convergence in some circumstances, a lack of sufficient precision in the partial derivatives as the parameter set—and therefore the number and degree of correlations between parameters—grows is viewed as a likely general cause. An effort is currently underway to include indirect terms, arising from the dependence of the coordinates themselves upon the partial derivatives, in the partials, and it will soon be seen if convergence behavior improves as a result of this upgrade.

As presently constituted, the programs implementing both versions of the bootstrap treat the parameter adjustments resulting from a single subsequent solution as final. In general it is likely that several iterations would be needed to reach final values, but the general state of convergence behavior is so fraught at

present that it has not seemed worthwhile to closely investigate the issue. In any event, iteration would considerably increase the time required for determination of standard errors if applied to every resample. It would probably be more feasible for a few 'pilot' resamples to be iterated to convergence and some figure of merit determined for the total scale of additional adjustments after the first, which could then be applied as a factor to the standard error determined from many more one-solution resamples. In a regime in which PEP is getting to convergence in a few iterations as desired, something like a 20 percent inflation would appear to be justified if subsequent adjustments have the same sign as the first, but this matter has not been significantly explored.

A large ($\sim$1000) number of potentially adjustable parameters pertaining to the topography of Mercury and Venus and to the earth orientation are currently generally not fit by PEP but instead are fixed at values determined by long-ago solutions. Given the progress that has been made on other fronts it would be desirable to see to what extent changes in the values of other parameters 'reflect back' on those not presently adjusted and to expose their correlations. A recent attempt to iterate on these parameters was aborted due to a complete absence of convergence, making the values currently ascribed to them questionable. If the convergence problem can be resolved, iteration to determine the values of these parameters would be welcome.

It is intended that the PEP development team will submit an independent proposal for funding this year. Success will provide resources for continuing the endeavors just described, but the form taken by the project in the longer term is uncertain. As open-source software, PEP would have better prospects for widespread use and ongoing development if it were shared among a community of researchers. The means of distribution and instructions described in the Appendix

represent one step toward this goal. However, further concerted effort is needed to make available the accumulated wisdom of the current PEP development team. Ideally, a future grant award would include funding for the reincarnation of PEP into a more modern software architecture, which would have the dual benefits of making it more accessible to researchers who might benefit from it and possibly uncovering bugs that are impairing its operation but are difficult even for an expert to discern at present due to the complex structure of the program.

Appendix

A practical guide to PEP

This document is intended as a reference for anyone who would like to install and run the Planetary Ephemeris Program. It is based on my experiences in working with PEP and will hopefully be a source of both theoretical and practical understanding. In preparing it I am deeply indebted to John Chandler of the Harvard/Smithsonian Center for Astrophysics.

1 Installation

If you are attempting to install PEP on a new machine, you will need to get a package of files and follow the instructions below. These files and directions were successfully used to install PEP on grist.ucsd.edu, a Mac running OS X, in October 2013. Different systems behave differently, and some may be incompatible with PEP in ways, for example processor architecture, that will take a long time to track down, so I recommend installing on a Mac. Grist's OS and processor specifications, obtained by typing `uname -a` at the command line, are:

```
Darwin grist.ucsd.edu 12.3.0 Darwin Kernel Version 12.3.0:  Sun Jan
6 22:37:10 PST 2013; root:xnu-2050.22.13~1/RELEASE_X86_64 x86_64
```

First, make sure a few things are in your path. Path information is stored differently on different systems, and the means of updating your path depends to some extent on what shell you use. I use bash, which is the OS X default, and in my home directory on grist I have a file called '.bash_profile', visible to `ls -a`, which defines the path. What you want is to get the current directory ('.') and ∼/bin in your path. For me, this meant adding the following lines to .bash_profile:

- `export PATH=∼/bin:$PATH`
- `export PATH=$PATH:.`

and then saving and quitting that file and typing `source .bash_profile`. This file should be automatically sourced whenever you open a new terminal; you should only have to explicitly source it when you've just changed it. However, you might want to do an `echo $PATH` to see that '.' and '∼/bin' are in fact noted as being in your path, especially if you experience problems later on.

To install PEP,

- Get the package of PEP files from GitHub at
  `https://github.com/NHJohnson/PEP`
- Unzip the package in your home directory (∼). This will create a new folder called PEP-master. This directory contains everything needed by PEP. However, it needs to be called 'peptop', so run
  `mv PEP-master/ peptop/`
- The directory now called peptop contains a directory called bin with certain shell scripts in it. The contents of peptop/bin actually need to be in ∼/bin, which needs to be in your path. If you already have a ∼/bin directory, move the contents of peptop/bin into it; otherwise just move peptop/bin to ∼/.

Either way, you can and probably should delete peptop/bin afterwards to avoid confusion.

- Change into the directory containing the PEP source code: `cd peptop/pep`
- Compile the FORTRAN files by typing `make new`. This may take a minute.
- Make the PEP executable by typing `make pep`. This should be quick.
- You now have the core PEP program, but we need to make several auxiliary programs, some for testing purposes and some that will be needed later.
- Change into peptop/verify and make the PEP internal file-comparison utility by typing `make verify`.
- Change into peptop/peputil, which contains the source code for various auxiliary programs.
- Enter `make bigtest`. We will use bigtest in a moment to verify that PEP is working.
- Enter `make abcps`. This creates the plotting program abc. The make should generate various warnings, which you shouldn't worry about as long as there are no errors.
- Enter `make prepmnpt`, used to turn observations ('normal points') in ASCII format into a binary obslib.
- Enter `make addmoon`, used to combine the lunar ephemeris with the rest of the solar system when integrating.

- Now we have to put some of the executables you just made in a place where they will be findable. I do this by creating soft-links in ∼/bin and then making sure that directory is in my path.

- Change into ∼/bin. Make soft-links to various executables with the following commands:

    1. `ln -s ../peptop/pep/pep pep`
    2. `ln -s ../peptop/verify/verify verify`
    3. `ln -s ../peptop/peputil/abcps abcps`
    4. `ln -s ../peptop/peputil/cpyobs cpyobs`
    5. `ln -s ../peptop/peputil/addmoon addmoon`

If you do an `ls -l` in bin, you should be able to see that the soft-links you just created point to the specified locations in pep, verify, and peputil.

For the next step, do a `pwd` in ∼/ and remember what it says your home directory path is.

Change directory back to ∼/bin. Three of the files that got put here when you unzipped the PEP package are shell scripts that are the means by which you will actually invoke PEP at the command line. Which one you use depends on what you are trying to do; more on that later. These scripts — pepint, pepobs, and pepsol — contain a hard-coded path to your peptop directory, which is obviously not the same as on the machine where the PEP archive was created, so you have to change it.

Open pepint with a text editor, e.g. `vi pepint`

Find the line that says `path="/Users/njohnson/peptop" # path to input files`

Change that path to whatever the path to your peptop directory is. Save and quit.

Do the same thing for pepobs and pepsol. In these files the variable is called 'ppath' instead of just 'path'.

Because of inherent PEP limitations, your path in these files can only be so long, around 50 characters. If your path to peptop is in fact longer, you might need to come up with a clever workaround. This happened to me when I was running PEP on the UCSD physics department machine. What is needed in this case is for you or your system administrator to create a high-level soft link to your home directory, effectively giving the long path-name leading to that directory a short synonym.

PEP is distributed without ephemerides because these files are very large. However, the PEP integrator can produce ephemerides, which have the extension .allpart, if provided with a set if initial values for the parameters. Values of the parameters are contained in files in pepin possessing a common extension. This distribution comes with three sets of initial parameter values that may serve as an initial basis for integration, with the extensions mod3, nhj1, and lock5. I recommend initially integrating from the mod3 parameter values. To do so, run integrate.py from peptop by typing at the command line

```
./integrate.py mod3
```

The subsequent integration process takes about 20 minutes and is described in more detail elsewhere in this document. It will create the ephemeris files and store them in peptop/ephem/. During the process, you may see the message

```
Fortran runtime error:  Sequential READ or WRITE not allowed after
EOF marker, possibly use REWIND or BACKSPACE
```

whenever the lunar integration runstream moonint is invoked. This is nothing to

worry about. However, you should not see other error messages, and if you see a STOP 20, something has definitely gone wrong, possibly relating to PEP's ability to find needed files due to incorrect paths. You can ctrl-C out of the integration if you see a STOP 20, presuming it doesn't stop on its own. PEP output relating to integrations is stored by integrate.py in peptop/integrationfiles. These text files record what PEP was doing during each part of the integration procedure, and they may provide some indication of why an integration failed, indications which will have more meaning for you as you become more experienced with PEP. Because these reports are saved with a timestamp, they are not overwritten each time you run the integrator and tend to build up if you run it regularly, so you may want to clean out peptop/integrationfiles periodically.

If the integration is successful, it will store the final ephemerides as allpart files in peptop/ephem, with peptop/newephem and peptop/itrephem being used to store certain intermediate products of the integration that are not needed afterwards. You may wish to copy the allpart files and the mod3 files to peptop/backup, possibly in a subdirectory you create therein named something appropriate like 'initialsolution.' The mod3 files and the ephemerides produced by integrating from them consititute a consistent solution that you can then put back in place in the future if things go awry. The consistency between the ephemerides and parameter-value files being used at any particular time is very important in PEP.

In the broad-brush, PEP estimates parameter values by fitting solar-system measurements to a model encapsulating the relevant physics. The mod3 files, as well as the other sets of parameter-value files, are the products of solutions that have been performed in the past. The measurements fit by PEP, as they stood at the archiving of the version you downloaded, are stored in peptop/data and are called obslibs (traditionally pronounced OBS-lybe) with the extension .obslib.

These measurements come from diverse sources. For the most part they are ranges between two points in the solar system, one of which is generally an observatory on the surface of the earth. They will be described in more detail later. For now, two of the obslibs need to be reassembled because they are so large that they had to be split up for archiving purposes.

To reconstitute these obslibs, from the command line in peptop/data run

```
cat mgs1aa mgs1ab > mgs1.obslib
cat ody1aa ody1ab ody1ac ody1ad > ody1.obslib
```

You may then delete the pieces mgs1a* and ody1a*.

Your copy of PEP is now nominally ready to run.

2 Bigtest

As you may recall, PEP has an auxiliary program called bigtest that is used to make sure that PEP is working normally. For someone involved in PEP development, bigtest is run to verify that changes made have not broken the program. We will run it at the outset to see if there are any problems with the installation.

To run bigtest, go to peptop/bigtest and type `./bigtest`. The output should be as below:

```
tmi tv1 tv2 tin toc ttr top tfr tmn tpl tfl
Starting tmi
Finished tmi
Starting tv1
Finished tv1
Starting tv2
```

```
Finished tv2
Starting tin
Finished tin
Starting toc
Finished toc
Starting ttr
Finished ttr
Starting top
Finished top
Starting tfr
Finished tfr
Starting tmn
Finished tmn
Starting tpl
Finished tpl
Starting tfl
Finished tfl
```

If this happens, bigtest has at any rate run successfully and your version of PEP is not crashing or complaining. If it does not, you have a problem that you will need to get sorted out. This was how I discovered that the processor architecture of the computer on which I initially tried to install PEP was not suitable for the task.

Apart from working or not working, bigtest produces files called t*.verout, which amount to a diff (using that verify program you also made) between the bigtest output and some stock output that is known to be fine. Some of these tests are very sensitive and generate a large amount of output even if the differences are

acceptably small, while others are more forgiving. The rub is that if bigtest finds unacceptably large differences between your output and the stock output, it is not clear what you would do to fix this other than install PEP on a better computer, so there is an understandable impulse to cross one's fingers and hope everything is fine as long as bigtest ran without complaint. John Chandler at the CfA can look at the .verout files (they are just text files) and judge whether differences are acceptably small, but I doubt whether a short tutorial would enable anyone else to do the same. Here is the output of an `ls -l` on t*.verout on grist:

```
-rw-r--r-- 1 njohnson staff 196762 Oct 23 14:24 tfl.verout
-rw-r--r-- 1 njohnson staff 48456 Oct 23 14:24 tfr.verout
-rw-r--r-- 1 njohnson staff 14524 Oct 23 14:24 tin.verout
-rw-r--r-- 1 njohnson staff 21102 Oct 23 14:24 tmi.verout
-rw-r--r-- 1 njohnson staff 3878 Oct 23 14:24 tmn.verout
-rw-r--r-- 1 njohnson staff 2864 Oct 23 14:24 toc.verout
-rw-r--r-- 1 njohnson staff 14936 Oct 23 14:24 top.verout
-rw-r--r-- 1 njohnson staff 7374 Oct 23 14:24 tpl.verout
-rw-r--r-- 1 njohnson staff 1990 Oct 23 14:24 ttr.verout
-rw-r--r-- 1 njohnson staff 1990 Oct 23 14:24 tv1.verout
-rw-r--r-- 1 njohnson staff 1990 Oct 23 14:24 tv2.verout
```

If your file sizes (the number right before the month, in bytes) are not radically larger than these, I wouldn't worry. If any of them are, I would consider getting in touch with John. There won't be much point in using PEP on your machine if the results aren't going to be reliable.

Once you are satisfied on this point, rename the files in peptop/bigtest currently called t*.out as biglist.t*, effectively making the files created just now by

bigtest the new baseline for your instantiation of PEP. If you were ever to want the original stock output (biglist.t*) for any reason, I have backed it up for you in peptop/bigtestorig, so overwrite without concern.

3 Running PEP

PEP is no one single thing, but has many different capabilities, and the particulars of working with it ultimately depend on what specifically you are trying to do. However, the actual running of PEP from the command line always has some of the same features. PEP is invoked through one of several shell scripts that sets the user-specified flags for the run and soft-links to the required files, among other things. There are at least three of these scripts: pepint, for integrations; pepobs, for O-C (prefit) and normal-equation formation; and pepsol for solving the normal equations and therefore estimating parameters. These scripts live in home/bin, parallel to peptop, whereas you will invoke PEP from within peptop, so it is necessary to have this bin directory in your path. The first argument to the name of this script is the name of a runstream file. These files live in peptop/pepin and all have the .peprun extension. Do not supply the .peprun piece of the name at the command line, just the part before it, apoomc for example. These runstreams constitute the actual input to PEP—options that are set to determine what will be done, calls to PEP routines and 'includes' of files that those routines will need. Finally, your invocation of PEP will specify various flags that are generally related to which files you want PEP to look at when doing whatever you are asking it to do. These options are listed at the beginning of the shell script. For example, open pepobs. The third line gives the syntax for the flags, and the lines beneath indicate the flag defaults and give some explanation of what each flag is for. For example,

the default of the 'iter' flag is 'lock5,' which means that PEP will take the current parameter values from files with the extension .lock5. If the parameter-value files you want to use have a different extension (and they will), .mod3 for example, you have to say so (or change the default).

Meanwhile, 'num', the number of input obslibs, has no numerical default value, so you need to tell PEP how many obslibs it is supposed to find. So let's say you want to use the runstream apoomc.peprun (Apache Point Observatory Observed Minus Calculated is the way to read that; I will also explain later what this is and why you run it) using parameter files with the .mod3 extension and making use of 1 obslib. The right shell script for doing an O-C is pepobs, so at the command line you would type:

```
pepobs apoomc -iter mod3 -num 1
```

and hit enter. Many of the flags have default values or otherwise don't need to be specified in general, but you need to know what specifications are correct for your application, or PEP may end up doing something other than what you think it's doing (using the .lock5 files instead of the .mod3 ones, for instance) and you may be none the wiser.

Regarding the aforementioned soft-links, PEP draws upon the content of files it needs by soft-linking those files to FORTRAN records in peptop called fort.XX, with the XX being a number of one or two digits that is assigned to a file with a certain purpose. For example, the shell script pepobs links the planetary ephemeris files [planet name].allpart to FORTRAN records 11 through 16. PEP also writes its outputs onto these FORTRAN records. As a result, PEP needs exclusive control of these files while it is running, the major effect of which is that you can't run multiple PEP processes in the same directory at the same time because what one process stores on a FORTRAN record will not be what another process thinks

it's getting when it reads the same. It would be possible to install multiple copies of PEP in different directories and run those at the same time, however.

3.1 Ephemeris and partials

Our aim in using PEP is to estimate the values of certain parameters by making a fit involving those parameters (as well as many others of less interest) to a set of lunar ranges and other solar-system measurements. In order to make such a determination, PEP needs to know the relations among those parameters: how a change in one value will change other values, and how all these changes will ultimately impact the residual of an APOLLO normal point. This is to say that it needs an ephemeris — a list of the positions of the bodies of the solar system at a series of times — and a corresponding set of partial derivatives that express the interrelation of the many parameters as determined by the physics coded in PEP.

PEP gets this information from files with the extension .allpart in peptop/ephem. There is one such file for each planet (or a subset of them), the earth-moon barycenter, the moon, and the moon rotation. No ephemeris is distributed through GitHub because the files are very large, but if you followed these instructions from the beginning you integrated to create ephemerides at the outset. This capability is essential to the iteration of a solution, described later.

3.2 Initial parameter values

Every instantiation of PEP makes use of a set of files containing initial values for a wide variety of PEP variables adjustable and not. These files reside in peptop/pepin and are identified as belonging to the same 'set' by means of a common extension. The three sets of existing parameter files have the extensions .lock5, .mod3, and .nhj1. Each set contains approximately 20 files. The shell scripts

by which PEP is run have 'lock5' as the default extension, but in working with APOLLO data is it advisable to begin with the mod3 set, and then to use one's own files when they are created in the course of iterating.

4 Getting parameter estimates using LLR data

4.1 Normal points

Every run (continuous attempt to range a single reflector, usually lasting several minutes) of the APOLLO experiment that nets lunar photons is, by an established data-reduction procedure, reduced to a single 'normal point.' This consists principally of a single launch time, range in seconds, and associated uncertainty. The actual data, of course, often contains many individual ranges within a single run, but PEP and programs like it are not suited to handle data in that raw format. The launch time stated in a normal point is not the actual launch time of any real photon; indeed, APOLLO normal-point launch times are all multiples of 5 seconds for convenience. Rather, they are a statement based on the run data of how long a photon, if it *had* been launched at that time, would have taken to make the round trip, and how accurate we believe that this round-trip time is. Normal points also contain some other information, notably about atmospheric conditions and launch station/reflector information, and are represented in a standard format that is described on the APOLLO website. Normal points that are downloaded from the APOLLO website are already in this format, and normal points in that format is exactly what we need to get started.

Note that this version of PEP comes with an apo.obslib that already contains all APOLLO normal points up through the third series, ending September 2013. The procedure described here of converting text normal points to binary format is

only necessary if you want to incorporate more recent data.

PEP is expecting to read in normal points in a binary-formatted obslib. Getting the normal points ready for use by PEP is a multi-step process. First, they need to be sorted by reflector. James Battat has kindly furnished a Python program that does this; it is called sortnp.py and is located in peptop/normalPoints. An appropriate application of the Unix 'sort' command can accomplish the same task. Put your text file containing the normal points as they are obtained from APOLLO in the same directory. Then run sortnp, supplying the name of the text file as an argument and directing the output to the filename you want to contain the sorted points. It doesn't matter what this file is called, but you probably want to distinguish it somehow from other datasets you may work with in the future. So you might type:

```
./sortnp.py APOLLO2012unsorted.txt > APOLLO2012nps.txt
```

The named file should now contain the same normal points but in five distinct reflector clumps. The reflector is identified by a number in the normal point; see the format information on the APOLLO website. Next, we need to put some header information in the file that will help both us and PEP keep track of it. Open the file of sorted points with a text editor. You want to add three lines at the beginning of the file. The first provides some plain-language description of the file and can be anything, but you should confine yourself to 72 characters, which is the maximum PEP will pay attention to. The second line needs to say NTAPE, followed by five spaces and then a two-digit number. If PEP is reading in multiple obslibs (as we will see it do soon), this tape number determines the order in which the files are fed in. In this case, it doesn't really matter; 99 is fine.

The third line should say SER and then a space and a four-character series name, like APOL (but it doesn't matter what it is, as long as no other obslib has

the same one, which is unlikely). Lines four and on are the normal points. So, format-wise, your first few lines should look like:

```
SIMULATED NORMAL POINTS BETWEEN CERGA AND AP15 IN FIVE SUBSERIES
NTAPE     99
SER CE15
511986 1 1 359574847633251523662207863019105 00 300300B100000 0 0 5320A 250A
511986 1 6 812195894436241055634013423019105 00 300300B100000 0 0 5320A 250A
511986 111122441553713424462865080099301910500 300300B100000 0 0 5320A 250A
...
```

Finally, we need to run some code that converts this file into a binary format. PEP comes with a program that does this, called prepmnpt (prep-moon-point). If you installed PEP by following the instructions in the first section, you already installed this program. If you are doing this for the first time, you need to make the prepmnpt executable. Go into peptop/peputil and type `make prepmnpt`. This creates the executable in peputil. You can create a softlink in ∼/bin by running

`ln -s ../peptop/peputil/prepmnpt prepmnpt`

in that directory. Now, go back to peptop/normalPoints and feed your sorted, header-containing points into prepmnpt:

`cat [name of file, e.g.  APOLLO2012nps.txt] | prepmnpt`

In true PEP fashion, this creates two outputs stored in files called fort.X. In this case, fort.2 is the binary obslib you want, and fort.8 is an ASCII obslib you may want to look at but that PEP doesn't need and indeed can't use. The proper location for the obslib stored on fort.2 is in peptop/data/apo.iobs0, so move it there (`mv fort.2 ../data/apo.iobs0`). The extension .iobs0 marks this as an

obslib that contains normal points only. We'll see obslibs that have more in them in a moment.

4.2 Observed Minus Calculated (prefit)

Before PEP receives any observational data, it already has a detailed general picture of the solar system over a relatively long time in the form of the ephemeris and the initial parameter values, described above. The latter represent the best estimates from previous PEP runs for parameters that do not change with time, like the masses of the planets, or that are evaluated at the epoch of integration, like the planetary mean anomalies. The ephemerides we work with cover the entire period of conceivably-relevant measurements, from about 1960 to 2020, and contain the positions of all the planets and the moon, recorded at a predetermined interval that depends on the specific body and ephemeris file under consideration. In the case of the N-body ephemeris, for example, the interval is 2 days for Mercury, .5 days for the moon, and 4 days for everyone else. Additionally, the ephemeris files contain partial derivatives recorded at the same cadence that express the instantaneous sensitivity of each body's position to the parameters that it is actually possible to fit. This bridges the gap between the specificity of actual data ('This is where the moon was and how it was oriented at time X') and the generality of the quantities we would like to actually estimate using PEP ('This is the mass of the moon, the semimajor axis and eccentricity of its orbit, and the rate of change of G').

At the first stage of processing, called the O-C or prefit, PEP uses this considerable background information to enhance an obslib in a way that ultimately makes parameter-fitting possible. Based on the initial parameter values provided, PEP is able to calculate what photon travel time it expects for each launch time it receives in normal-point form and compare this result to the round-trip time that

is actually recorded in the normal point. It then reports the difference between these calculated and observed ranges, which is the prefit residual for that normal point. We can use the PEP plotting utility abc in conjunction with the output obslib to produce a plot of the prefit residuals.

Calculating the theoretically expected range requires understanding, by means of the coded physics, how each initial parameter value impacts each ranging measurement. This physics is represented mathematically by the coordinates and partial derivatives contained in the ephemeris. Values in the ephemeris are tabulated at specified intervals, as appropriate for each body. In principle, PEP could be made to record the partial derivatives each second when integrating to produce ephemerides, but then the .allpart files would be absolutely enormous (they are already pretty big). Instead, during the O-C step, PEP interpolates the partials it has to determine their values at the times it is given.

By dint of having calculated ranges based on its encoded physics, PEP can determine the partial derivatives of the ranges with respect to each of the adjustable parameters, and it is this information that is later needed to determine what adjustments to those parameters will reduce the residuals, producing a better fit. Thus, this information is inserted into the obslib at this stage.

Now, in practical terms, this is the procedure for running an O-C calculation using APOLLO data. You want to invoke PEP using the shell script called pepobs, supplying the runstream called apoomc0.peprun. (This is for the case in which you have just created the obslib from text normal points and it therefore has the .iobs0 extension. If you are updating partials in apo.obslib, which already contains them, the runstream is apoomc.peprun.) You need to specify that there is only one obslib to read in (the one we created before, apo.iobs0) and that PEP is to draw on the initial condition files with the .mod3 extension. These files are the appropriate set

for the allpart files that we have created via integration; if you don't specify the set, pepobs will default to the .lock5 files, which are not. There is only one set of .allpart files, located in peptop/ephem, and pepobs knows to pull them in without your saying so. Therefore, what you type from peptop is:

```
pepobs apoomc0 -num 1 -iter mod3
```

The runstream apoomc0 is expecting to find your input obslib at peptop/data/apo.iobs0. As long as it is there, it will automatically be soft-linked to fort.40 and fed to PEP. The command-line printout should indicate what operation PEP is performing:

```
PEPOBS - PERFORM O-C RUN OR FORM NORMAL EQUATIONS
```

This should only take a few seconds, and if it goes normally, you will get a pep.msgs text file that says at the end NORMAL STOP IN MAIN, and often nothing but that. If you get to NORMAL STOP IN MAIN, PEP has run successfully. If it does not, something has gone wrong. I will try to address some errors in a later section.

If the O-C is successful, the output obslib is on fort.60. PEP will later look for it in data/apo.obslib, so move it to that location. Note that the O-C product is also an obslib; it still contains the normal point information, and now it also has the appropriate residuals and partials in it. Obslibs that contain all this get the extension .obslib, as opposed to .iobs0.

A similar O-C process must be run on every set of data that is going to be used as a basis for the solution. Each such set of observations (from other LLR projects, radar ranging to Venus and Mercury, transmissions from Mars landers and orbiters, etc.) has its own obslib in peptop/data and its own runstream for updating the partials in that obslib (the measurements themselves are static). These obslibs are backed up in peptop/backup, but since the only information they contain which

persists from solution to solution is the unchanging data points, there is no need to revert to the 'originals' unless the copies in peptop/data have become corrupted in some way. By default, the obslibs in peptop/data (if you installed PEP as described at the outset) contain partials based on the mod3 initial parameter values and the ephemeris in peptop/ephem, so there is no need to do an O-C to update them if you only mean to produce a single solution. However, if you make an iterated series of solutions, the initial conditions and ephemeris will be changing, so it will be necessary to update the obslibs after every integration in order to maintain consistency.

You may be interested in seeing the prefit residuals, and you can do so by using the PEP plotting utility abc, the use of which is explained later on. For now, though, I will keep moving through the solution procedure.

Note that in addition to the fort.XX files, you now have the brief pep.msgs file and a longer one called pep.out. Every PEP run, O-C or otherwise, produces these latter two files. pep.msgs contains only the briefest information about whether the run was successful or not, and you don't need to do anything with it. However, you probably want to save pep.out somewhere else and rename it so that it doesn't get overwritten by the next run. This file contains the PEP input stream — a long record in text form of all the files that PEP read in in the course of the run — and the results thereof. If your PEP run has been unsuccessful, this file may provide more insight into why than the .msgs file does. If your run was successful, important information about it is presented here in a human-readable format.

4.3 Forming normal equations, solution, and parameter estimation

Once the O-Cs have been done for all the obslibs that will be considered, it is possible to perform a fit to the data and to thereby get new parameter estimates and associated uncertainties. In PEP, this occurs in two stages, of which the formation of normal equations is the first.

At its core, a PEP fit is just the same process of least-squares minimization that is the standard for fitting data in practically every application. The number of parameters potentially fit by PEP is so large, the physics so intricate, and the data set so varied that it can be difficult to discern the workings of the least-squares process as a PEP operator, but the entire edifice should be understood foremost in this light. The basic procedure of least-squares fitting is this: Start with a set of values of an independent variable, like time; the corresponding measured values of a dependent variable, like round-trip times between an observatory and a set of lunar reflectors; and a fitting function that represents physics' understanding of the relationship between the two. This fitting function includes some parameters whose values are not inserted a priori (even if there is some understanding of what they are likely to be), but which are rather the parameters to be fit. In the archetypical case this function is linear in the parameters, which is to say that the value of the dependent variable may be expressed by the model as a sum of terms that are proportional to the value of exactly one parameter, e.g. $y = a \cdot f(t) + b \cdot g(t) - c \cdot h(t) \ldots$, where a, b, c, ... are the fit parameters and t is the independent variable. The functions f, g, h, ... are called the basis functions of the model and cannot be functions of any fit parameter.

The real-life situation modeled by PEP is emphatically not linear in this

way, but rather PEP linear*izes* the problem in order to bring the machinery of linear least-squares to bear on it. The solution is then iterated as will be described later in order to obtain convergence of the parameter estimates, which in a truly linear problem is not necessary.

Each data point (each (t, y) pair in the formulation previously stated) is plugged into the fitting function to create a system of equations for the residuals, one equation for each data point. This is an overdetermined system; there are more equations than parameters. (If there aren't, take more data or be less ambitious about the number of parameters you are trying to estimate.) Add together the squares of all these residual expressions to create a single equation for the sum of the squares of the residuals (which is the quantity we are trying to minimize, see following note). Differentiate this equation with respect to each free parameter and set the result equal to zero. We now have an exactly-determined system with as many equations as parameters. This system is the mathematical expression of the minimization condition, i.e. when the derivative of the sum of the squares with respect to each parameter is zero, the sum of squares is at a minimum (or, hypothetically, a maximum, which is not desired) in parameter space. Solving this system yields the parameter estimates, and the uncertainties in these estimates can be determined by propagating the uncertainties in the measurements. See Bevington and Robinson chapter 6 for a thorough treatment.

The approach of minimizing the sum of the squares of the residuals of the data points with respect to the fit model is a consequence of the presumption of Gaussian uncertainties in the data points themselves, which is a presumption that APOLLO makes about its own normal points. If measurement uncertainties are Gaussian and the model is correct up to the values of the parameters themselves, then the probability of a correctly-measured data point with a one-sigma measure-

ment uncertainty s lying a residual distance r from a fit curve is proportional to $exp(-r^2/(2s^2))$. The probability of a collection of data points having whatever residuals they have is the product of the original probabilities, a product which can be rewritten as a single exponential. The argument of this exponential is a negative-definite fraction whose numerator is the sum of the squares of the residuals. Because we define our best-fit curve as the one that is most probable given the data we have gathered, we want to maximize the exponential, which in turn means minimizing the sum of the squares of the residuals.

A PEP fit may involve hundreds of adjustable parameters and a correspondingly large set of equations, and so it is most convenient to formulate the problem in terms of matrices, an approach which is described in Bevington and Robinson chapter 7, section 2. This method has the additional virtue of providing the covariances between the parameters, which are interesting and important. The system contains n data points and m unknown parameters, $n > m$. We construct an m by n matrix $\mathbf{X}$ that contains the values of the m basis functions at each of the n values of the independent variable. Note that because the expression for each residual is a summed series of terms each consisting of the product of a parameter with a basis function that does not depend on any parameter, the basis functions are by definition the derivatives of the residuals with respect to the parameters; hence $\mathbf{X}$, called in PEP parlance the sensitivity matrix, is constructed from the partial derivatives calculated either when the ephemeris is formed or at the prefit stage. The (yet-unknown) fit parameters form an n-by-1 column matrix $\mathbf{B}$. The values of the dependent variable form another column matrix, this one m-by-1, $\mathbf{Y}$. The system of equations representing the translation of the independent variable to the dependent one by way of the model is represented by $\mathbf{Y} = \mathbf{XB}$. However, we don't expect an exact solution to this system, which is to say that there is no value

of the fit parameters for which the curve will pass exactly through each data point. Even if our model is a perfect description of the system, there is uncertainty in the measurements themselves. In order to get the values of the parameters which minimize the sum of the squares of the residuals, we have to derive and solve the normal equations.

For any values of the parameters, the vector of residuals is given by $\mathbf{R} = \mathbf{Y} - \mathbf{XB}$, where the elements of $\mathbf{R}$ are in units of standard deviations, and the elements of $\mathbf{Y}$ and of each row of $\mathbf{X}$ are therefore divided by the uncertainty of the corresponding data point to properly weight the data. The sum of the squares of residuals is given by $\mathbf{R}^T\mathbf{R}$, which equals $(\mathbf{Y} - \mathbf{XB})^T(\mathbf{Y} - \mathbf{XB})$. Distributing the transposition and expanding, this equals $\mathbf{Y}^T\mathbf{Y} - \mathbf{B}^T\mathbf{X}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{XB} + \mathbf{B}^T\mathbf{X}^T\mathbf{XB}$. Note that every term here works out to a 1-by-1 matrix or scalar and that the two middle terms are transposes of each other. The transpose of a 1-by-1 matrix is the matrix itself, so we combine these terms to get $\mathbf{Y}^T\mathbf{Y} - 2\mathbf{B}^T\mathbf{X}^T\mathbf{Y} + \mathbf{B}^T\mathbf{X}^T\mathbf{XB}$. We want to minimize this quantity with respect to the values of the parameters $\mathbf{B}$, so we differentiate with respect to $\mathbf{B}$ and equate to 0: $-2\mathbf{X}^T\mathbf{Y} + 2\mathbf{X}^T\mathbf{XB} = 0$, or $\mathbf{X}^T\mathbf{XB} = \mathbf{X}^T\mathbf{Y}$. This is the system of normal equations. In PEP parlance the matrix $\mathbf{X}^T\mathbf{X}$ is called the information matrix, and the vector $\mathbf{X}^T\mathbf{Y}$ is known as the 'right-hand side.' The system is solved by inverting the information matrix and hitting both sides of the above equation with it from the left, since the product of a matrix and its inverse (or vice versa) is the identity matrix. Thus if we call the inverted matrix $\mathbf{A}$, the best-fit parameter values are given by $\mathbf{B} = \mathbf{A}\mathbf{X}^T\mathbf{Y}$.

As it turns out, $\mathbf{A}$ is the covariance matrix, which has the variances of the parameter values on the diagonal and covariances in the off-diagonal elements.

Now, the practical part. All of the obslibs in peptop/data, once the appropriate O-C step has been taken, contain partials based on the specified initial

parameter values and the corresponding ephemeris that is found in peptop/ephem. Normal equations are formed for each obslib (or related group) separately and combined in the solution step that follows, so there is a separate runstream for each data set. All of these runstreams are called *sne.peprun, except for the one that uses non-APOLLO LLR data, which is called moonppr.peprun. I will describe the procedure of forming normal equations with reference to the non-APOLLO LLR data.

Normal-equation formation, like the O-C, invokes PEP through the shell script pepobs. Also needed are the number of input obslibs, of which there are 8 in the LLR set but differing numbers in the other sets, and the extension of the initial condition files to be referenced, still mod3. So the command-line invocation to form the LLR normal equations looks like:

```
pepobs moonppr -num 8 -iter mod3
```

PEP doesn't guess which of the .obslib files in peptop/data to use; rather the eight non-APOLLO LLR ones are named in moonppr.peprun and submitted to a program called lnkfort, residing in ∼/bin, that links them to the fort.XX files in which pepobs is expecting to find obslibs. This step shouldn't take too long to run, and success will produce a pep.msgs file saying 'NORMAL STOP IN MAIN' as before. There is also a pep.out file you may want to save. As for the normal equations themselves, there are two binary files, stored on fort.71 and fort.72. You should move fort.71 to peptop/data/llr.tsne and fort.72 to llr.reduced in the same location. It is the file with the .tsne (total saved normal equations) extension that will be wanted later. The other one is a version of the same that has undergone what is called partial pre-reduction (this is what the ppr in moonppr.peprun stands for), to condense the symmetric matrix by eliminating sections that pertain to certain 'nuisance' parameters. This was implemented to permit the estimation of

a large number of nuisance parameters that are not interesting or important to other portions of the dataset while conforming to current PEP limitations on the absolute number of parameters than can be handled. This approach was eventually largely abandoned out of concern that it was contributing to observed issues with iteration of the solution in favor of just fixing the nuisance parameters in advance and not estimating them at each iteration. The non-LLR sets of normal equations have their own names with a .tsne (total saved normal equations) extension.

You may justly wonder why the APOLLO data is not included in the LLR normal equations. It would certainly be possible to do it this way, but we have historically wanted to look at the predicted postfit residuals for this one data set, and as a result the APOLLO obslib is invoked in a special way in the solution step and folded into the normal equations at that point rather than being included here. This treatment does not change its impact on the parameter adjustments.

Once all the normal equations are formed and saved (there is no need for you to manually form and save each set as will be seen, but in principle), we can proceed to solve them and recover parameter estimates and uncertainties. The set of .tsne files, all based on the same ephemeris and initial conditions, are referenced by name for linkage to fort.XX files in the runstream called sssol.peprun. The availability of this large dataset makes it possible to estimate parameters to which the LLR data by itself is not sufficiently sensitive. The solution is run via a new shell script called pepsol, and at this stage all of the information matrices in the 10 .tsne files are combined into a single information matrix which is then inverted in-place. The inverted information matrix is called the covariance matrix, and it has on the diagonal the formal variances in the parameter estimates, and on the off-diagonals the covariances between parameters. The inverse matrix acts on both sides of the matrix-form normal equations to yield the parameter estimates. There

are 10 total .tsne files; syntactically, we enter:

```
pepsol sssol -num 10 -iter mod3
```

Once the solution is run (NORMAL STOP IN MAIN in pepsol.msgs), there will be an output obslib on fort.50 that can be stored in peptop/data with the extension .iobs2, which denotes that it was formed in a solution. Estimates of parameters in the style of the parameter value files are stored on fort.7; we'll need these later, so these are also saved. A general, readable list of parameter estimates and the associated information are found in pepsol.out. To see the estimates, in that file (which is an input stream with the same function that pep.out has at the O-C and normal-equation steps) search for 0ADJ (that's a zero) to skip to the relevant section. All of the parameters adjusted in the fit are listed here, along with their initial values, adjusted values, the magnitude of the adjustment, the formal uncertainty of the new value, and the number of increments of this uncertainty separating the initial and adjusted value.

It is possible that under a given set of conditions PEP is able to arrive at a stable chi-squared minimum just by running through the procedure outlined to this point, regardless of what the original parameter values were; this is called being in the 'linear regime.' However, it may also be that the solution needs to be iterated before final parameter estimates will be reached. This involves using the PEP integrator to produce new ephemerides based on the current solution and then rerun the solution using those ephemerides as a basis. The linear regime is something of a catch-22, as the only way to tell if you are in it is to iterate and see if the parameter estimates change significantly, so one is bound to iterate at least once in any case.

4.4 Iteration

We have described at one point or another the processes of integrating from parameter values to form ephemerides with partial derivatives, performing the O-C step to add these partial derivatives to the obslibs, forming normal equations, and solving these equations for new parameter values. Just accomplishing all these tasks once requires dozens of PEP invocations along with a large amount of file-renaming and other clerical tasks. In order to arrive at parameter estimates in which we can feel confident, it is necessary to repeat this process, perhaps dozens of times, using the new parameter estimates to form new ephemerides and so on. For the operator, doing this again and again is tedious and an invitation to error.

Included with PEP on GitHub are a number of Python scripts that automate the iteration process. These are located in peptop, and they have names containing the word 'iterate.' There are several versions, intended for different portions of the dataset: all of it (allss_iterate), only the LLR portion (llronly_iterate), only the APOLLO data (apoiterate), everything except the LLR data (nollriterate), and no real data at all but rather simulated data (simpts_iterate), about which more later. Each of these programs takes two command line arguments: the extension of the parameter-value files from which the solution series should depart, and the number of solutions that should be produced before the iteration terminates. A plausible invocation might therefore look like:

```
./allss_iterate.py mod3 5
```

Each of these scripts also calls two other scripts also in peptop. The first is solutionsort.py, which distributes the adjusted parameter values stored on fort.7 after a pepsol invocation into a new set of initial-value files with the .nhj1 extension and stores them in peptop/pepin. The change of extension was implemented to avoid overwriting the mod3 files. One might prefer to just back up the original

mod3 files in peptop/backup and then allow the versions in pepin to be overwritten. The alternate version solutionsortmod3.py accomplishes this, but by default the iteration scripts are expecting to find the parameter values in nhj1 files after the first step (and from the very beginning if you specify), so the scripts would have to be changed both to use solutionsortmod3 and to expect files with the mod3 extension at every point, as would integrate.py.

That same integrate.py, which we used at the very outset to form ephemerides not provided with the PEP package, is the other program invoked by all the iteration scripts. It calls PEP to perform N-body integrations of the solar system and individual integrations of the orbits of all the planets and the moon, cycling through three times to refine the results.

Each iteration script starts at the O-C step and is therefore expecting to find ephemerides based on the specified set of parameter-value files already in pepin. This is to spare the operator the time required for integration if such a consistent solution is already in place. If it is not, integration must be accomplished beforehand by calling integrate.py from the command line in peptop with the parameter-value file extension specified. If no extension is given, nhj1 is assumed.

In the course of an iterated series of solutions, the pep.out from each stage of the process in timestamped and saved. Such files arising during integration steps are stored in peptop/integrationfiles. These are generally only needed if the integration has failed and you need help finding out why. Each solution requires many invocations of the integrator, so these tend to accumulate and should be cleared out periodically. PEP output from the O-C, normal-equation formation, and solution steps is stored in peptop/solutionfiles. Unless the iteration has failed, the pepsol.out will be of the most interest because they contain the parameter adjustment information. These files are named sol[timestamp].out.

4.5 Plotting with abc

The collection of PEP files provided include what you need to run PEP's plotting utility, called abc. The program resides in peptop/peputil; to make the executable, if you haven't already, enter that directory and type `make abcps`. You may want to make a soft-link to this executable in ∼/bin or somewhere else in your path so that abcps (the ps stands for postscript, which is the kind of file the program produces) can be comfortably run from peptop.

abc is a versatile program that can make plots of many different PEP-associated quantities. I use it almost exclusively to make plots of pre- and postfit residuals. It plots information contained in an obslib, which it expects to be soft-linked to fort.60, and gets its instructions from a short text file, which is currently located at peptop/abc.omc.config. For documentation related to abc, see peputil/abc.f; For now, I will just explain the content of the config file.

- LOOK=3 This variable is binary coded; '1' means 'print', '2' means 'plot', so '3' means 'both print and plot.'

- OBSLIB=60 This specifies the FORTRAN record number to which abc expects to find an obslib soft-linked.

- NDV=2 This is how you specify the dependent variable to be plotted. A value of 0 here selects the normal-point timestamp. Values less than zero specify the corresponding element of the SAVE array of values associated with the normal point, and values greater than zero specify elements of the DERIV array. The residual of the data point with respect to the solution that preceded and informed the creation of the obslib is DERIV(2), so that is what is being plotted here. Some information about the rest of the content of these arrays is available in peptop/pepvars.lis. What information specifically

is in what places in the array depends on what partial derivatives are included in the obslib being used, which is determined by the PEP runstream that was used to create the obslib. See NAMES below.

- NCODE=1: This selects the first observable of two, which for ranging data is the round-trip time. The second would be the Doppler shift, if applicable, i.e. for planetary radar observations.

- NSERIE=100: The obslib is broken up into series, for example where one reflector's data stops and the next begins. This variable sets the number of such series abc will go through looking for the information it wants to plot before giving up. 100 mostly guarantees it won't give up.

- ONEAXE=TRUE: Plot done on a single set of axes.

- SORT=TRUE: This sorts the data by time tag, producing a single array instead of a separate array for each series in the obslib. This is done to suppress abc's urge to scale the plot according to the data limits of the first series.

- ZLINE=T: Draws the line y=0 on the plot.

- NAMES=T: This is very useful, as it causes the names of the parameters whose partials are available in the obslib to the printed in the abc output. This lets you know what your plotting options are, and it can also be used to cross-reference with the content of the L-vectors in the runstream (below).

- SPOT: These are the PEP 4-character names of the reflectors (or other ranging targets) whose residuals you want to plot. Separate multiple reflector names with commas.

- START, STOP: These are the Julian days that you want to be the beginning and end of the x-axis in your plot. A Google search for 'julian day converter' will bring up a perfectly good one from the U.S. Naval Observatory. If you use the same input file for all of your abc plotting, make sure you change this range if you want to look at some different section of data.

To run abc, designate abc.omc.config as the input and some other file like abc.out as the output. Note that this output is not the plot, but rather a report of what occurred during the abc run. Type for example:

```
abcps < abc.omc.config > abc.out
```

In addition to abc.out, abc will produce a file that is always called plot53.ps, the postscript file containing your plot. A postscript is actually a text file whose contents can be viewed with less or vi, but probably you would like to see the results in graphical form. There is a freeware program called ghostview that can be used to view the postscript plot. I always use the ps2pdf [filename] command to get a PDF, and then look at that. This program may also be called pstopdf on your machine.

5 The runstreams

We should examine the PEP runstreams and how they may be modified. The runstreams, which reside in peptop/pepin and have the extension .peprun, contain instructions for aggregating all the information that will be fed to PEP as it works on the specified task, including control variables that partially determine what exactly is going to be done. As a result, each task you undertake with PEP uses a distinct runstream, even though there are multiple tasks for which the actual call to PEP is made via the same shell script. For example, all invocations of the

PEP integrator begin with the shell script pepint, but a moon integration will use moonint.peprun, and an earth-moon barycenter integration uses embryint.peprun. Note that there is nothing sacred about these names; they are just convenient and descriptive. I could rename moonint.peprun as bargle.peprun (the .peprun extension is, in fact, important), and it would still work fine as long as I called it by this name at the command line when running it with pepint.

Each runstream is directing PEP toward a specific task, so no two are the same, but they do have certain commonalities. You have a runstream used for estimating parameters using the entire solar system dataset, saved as peptop/pepin/sssol.peprun. I'll use it as an example.

The first line of this runstream and many others is '/SPECIAL', and this initiates a section of the file in which shell commands can be issued. In sssol, there are two such, the first a call to the shell script lnkforts (located in ∼/bin, as you may recall). This runstream is used with the shell script pepsol, and the variable denoted by ?PATH is supplied from there, but apart from that, this is a command that could be executed from the command line in peptop. The purpose of lnkforts is to store files that PEP may require on FORTRAN records, fort.XX. This runstream is used when you are estimating parameters based on all of the sets of saved normal equations, and what is being linked here are those total saved normal equation (.tsne) files. The non-APOLLO LLR data, for example, makes up llr.tsne. To see where the APOLLO data comes in, look at the next line, which is a soft-linking shell command putting the APOLLO obslib on fort.40. The APOLLO data is treated differently in this way because this runstream is written so as to get predicted postfit residuals for that data alone, about which more later, but it also gets folded into the normal equations and contributes to the parameter estimates.

That concludes the /SPECIAL section. The main body of the runstream

now commences at the flag '/GO'. Everything from this point on is not a shell command, but either an explicit setting of PEP variables or an include pointing to a file in which such variables are set. All of the information PEP gathers here is reproduced in the first section of the text output of the PEP run that uses this runstream, by default called pepsol.out (if pepsol was the shell script invoked) or pep.out (if either other script was used). The first section of this file, in which the information gathered by the runstream is all printed, is called the input stream, and if you want to know what value of any parameter or control variable or anything else was used at the outset by that run of PEP, that is where you look.

Returning to sssol, the first two lines after /GO provide some information about the runstream which then appears at the top of pepsol.out to indicate what runstream was used. These lines contain no information or instructions for PEP.

The next section sets the control variables, mostly elements of the ICT and JCT arrays, which govern some aspects of PEP operation. A comment in a runstream starts at a $ sign, so you can see that the comments here often explain what the purpose of a given variable designation is. There are a large number of control variables that are basically always set the same, so for convenience these values have been placed in a different file called pepin/nmlst1.include, and you can see an *INCLUDE statement pointing to that file on the first line of this section. For example, there is almost always apriori input (described later) and so ICT(44) is set to 1 in nmlst1.include to instruct PEP to expect it. Other instructions are more specific to the purpose of this particular runstream. We want to get predicted information about the residuals to the APOLLO data based on the parameter adjustments, so JCT(51)=-1, and the line below setting both JCT(51) and JCT(52) to 0 (that's what JCT(51)=2*0, or alternatively JCT(51)=0,0 or JCT(51)=0,JCT(52)=0 means) has been commented out. Note that later input

always supersedes earlier if they conflict, so if this line were un-commented without changing anything else, JCT(51) would be 0, not -1, nor would any error result. This is true throughout the runstream.

Among other things in this section, there are the IOBS and IMAT values, which tell PEP on which FORTRAN records to look for those linked saved normal equations and obslibs. Clearly there are a large number of control variables and the effects of changing them to various values are even more numerous. Information about the variables and the values they assume is given in the long section of comments at the beginning of peptop/pep/prmred.f.

The next section of the runstream begins with the commented line reading 'START PARMS AND L-VECTORS', and it is in this section that current (prior-to-adjustment) values of parameters are read in and parameters are marked for inclusion or exclusion from whatever process is being undertaken, in this case (sssol.peprun) adjustment of the values of the included parameters by solution of the normal equations. There are two major types of statements here. The first is includes of parameter values, in which a *INCLUDE line points to a file containing current values. Most of these files are the common-extension initial-condition files, e.g. .mod3, the actual extension being denoted by ?ITER and inherited from the shell script. The second type of statement is an L-vector, which is a numerical list of included parameters. The two types of statements are grouped, so that an L-vector for certain parameters follows closely on the include for the current values of those same parameters. For example, one of the early includes points to sscon.?ITER, which contains the masses of all the modeled bodies as well as parameters that apply to no body in particular, like time-variation of Newton's constant. Soon after, you can see the statement of LPRM, which is the name of the L-vector for these parameters. In this example, the first through eighth, tenth through

17th, etc., elements of LPRM are marked for adjustment. The correspondence between element of an L-vector and the physical parameter it refers to must be acquired, but this information is available in the documentation at the beginning of peptop/pep/prmred.f. LPRM(1) through (10) are the masses of the eight planets (3 is the earth-moon barycenter mass), Pluto, and the moon. LPRM(32) is the aforementioned time-variation of G, and so on.

After the sscon variables and a few things that are invariably commented out, we get into the parameters that are specific to the major bodies of the solar system. The section for each body starts with an *OBJECT line naming it. Each body has at least six parameters associated with it, whose current values are given in an initial condition file named after the body, e.g. mercury.?ITER. Some bodies about which there are more relevant measurements, like Mars, have considerably more than six, and the rotations of the earth and moon are so heavily parameterized that they are each treated as an object in their own right. The six attributes possessed by each planet (and the moon) are the six parameters of its orbit: semimajor axis, eccentricity, inclination, longitude of ascending node, argument of periapsis, and initial mean anomaly. In stating whether or not they are to be adjusted (or whatever the runstream is intended for with respect to the parameters), there is a slight wrinkle regarding these first six elements. As we saw with LPRM, in general you mark a parameter for inclusion by putting the number of the parameter in the L-vector, so if a hypothetical body has three possible adjustable parameters in the order A, B, C, and you want to adjust the A and C but not B, it would be L=1,3. Regarding these six, however, each one gets either a 1 (mark) or a 0 (don't mark) at the beginning of the L-vector, after which the convention reverts to normal. The next parameter of the body is still number 7, not number 1. So if I wanted to mark for inclusion the semimajor axis, inclination, and eccentricity of Mars, as well as

parameters 7 and 10 (never mind what these are), but nothing else, the L-vector for Mars would be L=1,1,1,0,0,0,7,10, which could also be written as L=3*1,3*0,7,10.

Pluto is here, but it is not a planet. Its initial conditions are stored in a forlorn file with the extension '.nbody311' instead of a mod3 or other parameter-value file.

In the moon section, you can see that there are certain parameters that are marked separately from the usual L-vector. These moon harmonics relate to the lunar gravity model.

Next we get into the sections relating to the parameters that are below the planetary level and are related to the particulars of ranging. Obviously the precise locations of ranging stations, reflectors on the moon, etc., are relevant to what range is actually measured, and PEP is also prepared to fit for these coordinates. The first section is the *SITES, which are the ranging stations; next are the *SPOTS, which are ranging targets like retroreflectors; and after that are the *BIASES, which are offsets applied to all the ranges from each experiment, or from each period of instrumental consistency within a ranging effort, so for example there are several different bias parameters for APOLLO corresponding to different periods in the experiment's history. The sites, spots, and biases are alike in that they are not marked via L-vectors but rather in the initial-condition files themselves. The includes in the runstream do tell you where to look if you want to change a parameter's status. Open for example moonsite.mod3. You can see lines giving the name of an observing station (e.g. TEXL) followed by three coordinates and then after a gap by four numbers that are either 1 or 0. The integers control the marking of the preceding coordinates in the same order. In this file the fourth integer is either a -1 or a 0 and indicates in what coordinate system (cylindrical or spherical) the site coordinates are specified. So if you wanted to estimate the

radius of Apache Point but not the latitude or longitude, you would change the integers at the end of the 'ApachePt' line from '1 1 1 0' to '1 0 0 0'.

Looking back at sssol.peprun, the next section is the *APRIORI includes. PEP is certainly willing to estimate all possible parameters from the data that you give it alone, but often there is some other source of information constraining certain values or a common-sense relationship between them that needs some mathematical expression. The indicated files, also in peptop/pepin and with the extension .apriori, fill this role. The format of the information in the files is described in the comments at the beginning of peptop/pep/acmin.f. Broadly, in an apriori entry you may indicate that a parameter holds a particular value, and then also provide something like an error bar for the constraint, which indicates scale at which the PEP estimate of the parameter is permitted to deviate from the constraint value (if provided), or from the current estimate (if not).

Consider the constraint file sunhar.apriori, which provides a constraint based on published measurements on the value of the solar quadrupole moment, known in PEP as SUNHAR and indicated in LPRM by the number 33. The value of X in the record is the parameter estimate, 2.18e-7, and B is the estimated uncertainty in that measurement, 0.06e-7. If only B were wanted it would be necessary to set APEST=.FALSE., and then the constraint would serve to limit the deviation of the parameter estimate from the value it held prior to the solution, whatever value that may have been. The last line in the record, SUNHAR, is the name of the parameter to which the constraint will be applied. It is by means of such constraints that the values of 'nuisance' parameters are set, thereby obviating the need for partial pre-reduction of the normal equations, as described previously.

Apriori constraints can also be used to enforce a correlation between values that are known to be related without actually specifying either one. A noteworthy

example arises with the coordinates of nearby observing sites, which are known to be a certain distance apart and so should not be made by PEP to appear in different relative positions than those in which they are known to actually be. These are found in pepin/llrsite3.apriori. In this case, each apriori record involves two parameters, whose names are given on consecutive lines at the end of each. The X vector also then has two elements. The first is 0, and this indicates that later elements in the vector represent the offsets of the second and later parameter values with respect to the first, e.g. of MLRS RAD with respect to TEXL RAD, rather than absolute magnitudes. The value of B then indicates the degree to which this separation is constrained.

6 Simulating normal points with DLTREAD

PEP has the additional ability to create a list of simulated normal points, i.e. ranges paired with hypothetical launch times. This is useful if you want to experiment with PEP using a data set that is not subject to the vagaries of real data, e.g. systematics, measurement uncertainties, inseparable unmodeled effects. I used it to create a data series spanning five years in the 1980s at a range of cadences to provide a common jumping-off point for comparing PEP against broadly similar modeling programs that have been developed by French and German collaborations.

In concept, it is easy to understand how PEP does this. The program already calculates a range, based on its current ephemeris and not on any actual measurement, at every O-C step to produce the prefit residuals. This is to say that PEP computes an expected range in seconds for the launch time given in an authentic normal point and subtracts this calculated result from the observed round-trip time. To get a simulated normal point, we just feed in the time we want

instead of the time of any actual measurement and write down the calculated range instead of subtracting it from anything. The PEP subroutine that embodies this capability is called DLTREAD.

The instructions that will be used to produce the simulated points are written into a text file with the extension .iobcon in the peptop/pepin directory. Several examples are already located there (note that the iobcon files that are clearly named after lunar ranging sites are actually for something else). Open for example fiveday.iobcon. The contents look like this (with surrounding demarcations for reference in the following):

```
1 | 2| 3  | 4 |  5 |  6 |  7  | 8   |  9  |     10    |     11        | 12  | 13

1 10 MLRS TEST MLRS AP15  2.00 1.E001.E-09 1    0.0     5.6351966D+14 1 -1  110

2446432 11  0          2448256  0  0                1.0         5 15142

14                    |     15                      |  16      |   17
```

The formatting here is important, so iobcon files for new series of simulated points should be closely based on those already in existence. Here are the meanings of portions of the instructions that might be changed:

1. NCODF, with '1' denoting a 'radar'-type, i.e. two-way ranging observation.

2. NPLNT, denoting the observed body. NPLNT = 10 indicates the moon in PEP, with the proper planets being 1 through 8.

3. Four-letter code of receiving LLR station. MLRS is one incarnation of the MacDonald site in Texas.

4. Four-letter name for the series, which can technically be anything, but some of the additional tools I developed for this process count on it being TEST.

5. Four-letter code of the transmitting station. For the Haleakala observations, two separate but nearby telescopes (MAU1 and MAU2) were used to transmit

and receive the beam, but for all other stations the transmit and receive are the same.

6. Name of the targeted spot on the observed body. This is the Apollo 15 reflector. AP14, AP11, LUN1, LUN2 are other possibilities for LLR.

7. This is a factor by which the normal-point round-trip-time uncertainty will be inflated. This error-bar inflation is another purpose of DLTRED, but as these are dummy observations the quoted uncertainty is unimportant.

8. This is another inflation factor but applies to the second observable, which if present is the Doppler shift for radar-type observations.

9. This is an accuracy time constant whose use in PEP is somewhat technical. Stick with the one given here.

10. Don't change these.

11. This is the frequency of the ranging beam, in this case of the 532 nm laser used by APOLLO. The corresponding wavelength gets inserted by hand when we convert to ASCII normal points, so if you change this frequency, be sure to also change the wavelength later on. The frequency affects propagation speed in various media.

12. Don't change these.

13. This is NTAPE, as in the header information of an ASCII normal point file. It affects the order in which obslibs are read, but that may not matter here. It should be positive, 110 is fine.

14. This is the Julian day, hour, and minute for the start of the observations. Converters from Julian to conventional dates can be found online.

15. This is the end date and time.

16. This is the error itself in seconds, to be multiplied by the error inflator in the first line. It just needs to be non-zero so that PEP will make dummy data with delays. The corresponding Doppler error field is blank, so PEP doesn't generate dummy Doppler shifts.

17. This is the observation cadence in days and seconds, i.e. 5 days 15142 seconds. This file is for dummy points with a five-day cadence, but we wanted the moon to be in the observable sky and at a fairly consistent position over the whole range, so the large number of seconds is a correction for the fact that the moon advances in the sky night to night.

This information and more is available in peptop/pep/dltred.f.

As described above, making dummy points is a wrinkle on the O-C procedure, and so it uses a slightly modified version of the O-C runstream apoomc.peprun. If, in fact, you look at that file (in peptop/pepin) you will see near the bottom a call to the subroutine DLTREAD and various includes of .iobcon files that have been commented out with a $$. We want to put DLTREAD back in and place below it an include that points to the .iobcon file we just made. You already have a peprun file in which this has been done, called testnps.peprun. Note that the include points to secondoneday.iobcon, which is different from the five-day one we were looking at before but is the same in every regard except the cadence of the points to be produced, which is daily. Anyway, you will want to point it at your own iobcon file instead.

Next, run PEP from peptop as though you were doing an O-C but using the modified runstream, therefore:

```
pepobs testnps -num 1 -iter mod3
```

This will produce all of the same outputs that an O-C normally would, but the information you wanted is in the pep.out file, along with a bunch of other stuff including the actual O-C of whatever data was queued up, since we didn't change the runstream in any way that would tell PEP not to do this (although it is possible to do so). The key to extracting the dummy points is the fact that they belong to a series called TEST as specified in the iobcon file. The information also needs to be rewritten in the ASCII normal point format, and I have written some code that does both extraction and reformatting, called testnpfinder.py and located in peptop. Note that some information in the normal points is not assumed by PEP and so is inserted at this stage: the number of photons in the point, the measurement uncertainty, the signal-to-noise ratio, the data quality grade, the atmospheric pressure, the temperature, the humidity, the laser wavelength as I mentioned above, the duration of the observation and two letters whose purpose I do not know. Obviously there are no 'right' values of these things for data that was never really collected, but PEP had to assume something about at least some of them to produce round-trip times, so you can either be consistent with those assumptions or not by modifying the relevant section of testnpfinder. If you are not consistent with the assumptions, residuals of a PEP fit to the dummy data will be nonzero, but that may be desirable depending on what you are trying to do. PEP's defaults in creating the points are pressure = 100000 (hundredths of a millibar), temperature = 0 (C), humidity = 0 (%). None of the other values affects the range except the wavelength, which should be consistent with what you set as the frequency in the iobcon file. The program testnpfinder takes the name of the pep.out file as its sole argument, so run for example ./testnpfinder.py fiveday.out. The output goes into a file in the same directory called [input name]nps.txt, so fivedaynps.txt for the example given. If you are producing multiple series to combine

later as described below, make sure to change the file names so that each new one doesn't overwrite the old.

If you produce a number of series that use the same ranging site and reflector but have different time spans or cadences, you may want to combine them into a single time-ordered file. You can do this by first combining the text files using cat and then running another program I wrote called ordernp.py, located in peptop/normalPoints. It takes the name of the combined text file as its sole argument and puts out a file called [input name]sorted.txt. Every line in the input must be a correctly formatted normal point. You now have an ASCII obslib of dummy normal points. You should append the same three lines of header information that you would put on any other obslib. This obslib is made of real data for all PEP can tell and can be used in all the same ways, for example converted into a binary obslib and used in an O-C (which should show residuals that are zero to within the roundoff error if you left the environment information the same as the PEP defaults as described above).

7 Storing a solution

After a solution has been iterated to convergence or near-convergence, it may be desired to move on to something else while also storing the current solution so that it can be brought back up later or used for reference. To store a solution, enter peptop/backup and create a new directory that will house the files. Enter that directory and copy the program store_solution.py from peptop/backup to the present directory (i.e. **cp ../store_solution.py .**) Then run store_solution.py with no arguments. This will copy the present ephemeris, obslibs, parameter-value files with the .nhj1 extension, and solution output containing parameter estimates

(located at peptop/solutionfiles/sol*.out) to this directory.

To later on put a stored solution back in place, run restore_solution.py from peptop/backup with the name of the directory holding the desired solution as the sole argument. This will put the allpart files constituting the ephemeris back in peptop/ephem and the obslibs back in peptop/data. The parameter-value files will be put back in peptop/pepin, but with the extension .mod3 instead of .nhj1 as currently configured. The solution output files are not put back in peptop/solutionfiles, as these are only for operator reference and are not used by PEP. Note that the use of 'restore' to describe these acts is at variance with the meaning of the notion of 'restoring a solution' as traditionally used by the PEP operators, which refers to something rather technical and different.

8 Bootstrap procedures

Resampling capability resides in two programs that reside in peptop and are run from there. The residual bootstrap is done with full_bootstrap.py, and the measurements bootstrap with npbootstrap.py. Both programs take the number of desired resamples as their sole command-line argument. Both programs have scripting portions that perform normal-equation formation and solution for parameter adjustments after each resample, and these sections are currently written with the presumption that the entire dataset is being used. If some subset of the data is being used instead, some of the pepobs calls forming normal equations will likely have to be commented out, and since it is likely in that case that a solution runstream other than sssol.peprun will be wanted, the name of the desired runstream must be entered into the code and the argument of '-num' will likely also have to be changed.

All obslibs in peptop/data will be resampled. If any obslibs are present there which are not needed for the intended solution, they can be removed to speed up the resampling procedure, but their presence will do no harm. Whether residual or measurements resampling is being performed, the basic course of events is the same:

1. The binary obslibs are converted to a text format with the .obstxt extension using cpyobs.

2. The text obslibs are mined for the information needed to perform the resampling, which differs between the procedures.

3. A temporary text obslib is created by resampling of the mined information.

4. The temporary text obslib is converted to binary format with cpyobs, overwriting the previous version (all the information needed from the original obslib is stored in memory at step 2).

5. Steps 3 and 4 are repeated until every obslib in data has been resampled.

6. Normal equations are formed from the obslibs and solved to produce parameter estimates based on the resampled data, which are stored.

7. Steps 2-6 are repeated as many times as were indicated at the command line.

8. Histograms of the parameter estimates are made and stored as allssbs*.png in peptop/solutionfiles. The standard deviation of each histogram is printed on the plot. This serves as the estimate of standard error for that parameter.

9. When residual resampling is performed, the initial normalized residuals of each series in the resampled data are also plotted and the plots stored in peptop/solutionfiles/*resids.png.

Bibliography

[1] T.W. Murphy, E.G. Adelberger, J.B.R. Battat, L.N. Carey, C.D. Hoyle, P. LeBlanc, E.L. Michelsen, K. Nordtvedt, A.E. Orin, J.D. Strasburg, C.W. Stubbs, H.E. Swanson, and E. Williams, "APOLLO: the Apache Point Observatory Lunar Laser-ranging Operation: instrument description and first detections," Publications of the Astronomical Society of the Pacific, **120**, 20-37 (2008).

[2] H. Dautet, P. Deschamps, B Dion, A.D MacGregor, D. MacSween, R.J. McIntyre, C. Trottier, and P.P. Webb, "Photon counting techniques with silicon avalanche photodiodes," Applied Optics **32**, 3894–3900 (1993).

[3] C. Scarcella, A. Tosi, F. Villa, S. Tisa, and F. Zappa, "Low-noise low-jitter 32-pixels CMOS single-photon avalanche diodes array for single-photon counting from 300 nm to 900 nm," Rev. Sci. Instrum. **84**, 123112 (2013).

[4] F. Villa, B. Markovic, S. Bellisai, D. Bronzi, A. Tosi, F. Zappa, S. Tisa, D. Durini, S. Weyers, U. Paschen, and W. Brockherde, "SPAD smart pixel for time-of-flight and time-correlated single-photon counting measurements," IEEE Photonics Journal, **4**, 795-804 (2012).

[5] S. Schlamminger, K.-Y. Choi, T.A. Wagner, J.H. Gundlach, and E.G. Adelberger, "Test of the equivalence principle using a rotating torsion balance," Phys. Rev. Lett., **100**, 041101 (2008).

[6] R.D. Younger, K.A. McIntosh, J.W. Chludzinski, D.C. Oakley, L.J. Mahoney, J.E. Funk, J.P. Donnelly, and S. Verghese, "Crosstalk analysis of integrated Geiger-mode avalanche photodiode focal plane arrays," Proc. of the SPIE, **7320**, 73200Q (2009).

[7] M.L. Hsia, Z.M. Liu, C.N. Chan, and O.T.C Chen, "Crosstalk effects of avalanche CMOS photodiodes," IEEE Sensors, **11**, 1689-1692 (2011).

[8] E. Sciacca, G. Condorelli, S. Aurite, S. Lombardo, M. Mazzillo, D. Sanfilippo, G. Fallica, and E. Rimini, "Crosstalk characterization in Geiger-mode avalanche photodiode arrays," Electron Device Lett., **29**, 218-220 (2008).

[9] X.J. Chen, E.B. Johnson, C.J. Staples, E. Chapman, G. Alberghini, and J.F. Christian, "Optical and noise performance of CMOS solid-state photomultipliers," Proc. of the SPIE, **7781**, 77810F (2008).

[10] A.N. Otte, "On the Efficiency of Photon Emission during electrical Breakdown in Silicon," 2009, Nucl. Instrum. Meth. A, **610**, 105-109 (2009).

[11] I. Prochazka, K. Hamal, L. Kral, and J. Blazej, "Optical cross-talk effect in a semiconductor photon counting detector array," Proc. of the SPIE, **5956**, 59560Y (2005).

[12] I. Rech, A. Ingargiola, R. Spinelli, I. Labanca, S. Marangoni, M. Ghioni, and S. Cova, "Optical crosstalk in single photon avalanche diode arrays: a new complete model," Optics Express, **16**, 8381-8393 (2008).

[13] N. Akil, S.E. Kerns, D.V. Kerns, Jr., A. Hoffmann, and J. Charles, "A multimechanism model for photon generation by silicon junctions in avalanche breakdown," IEEE Trans. Electron. Devices, **46**, 1022-1028 (1999).

[14] W.J. Kindt, H.W. van Zeijl, and S. Middelhoek, "Optical cross talk in Geiger mode avalanche photodiode arrays: modeling, prevention and measurement," Solid-State Device Research Conference (ESSDERC) **28**, 192 (2008).

[15] C. Cammi, F. Panzeri, A. Gulinatti, I. Rech, and M. Ghioni, "Custom single-photon avalanche diode with integrated front-end for parallel photon timing applications," Rev. Sci. Instrum., **83**, 033104 (2012).

[16] M. Crotti, I. Rech, A. Gulinatti, and M. Ghioni, "Avalanche current read-out circuit for low jitter parallel photon timing," Electronics Lett., **49**, 1017-1018 (2013).

[17] A. Vila, E. Vilella, O. Alonso, and A. Dieguez, "Crosstalk-free single photon avalanche photodiodes located in a shared well," IEEE Electron Device Lett., **35**, 99-101 (2014).

[18] M. Gersbach, Y. Maruyama, R. Trimananda, M.W. Fishburn, D. Stoppa, J.A. Richardson, R. Walker, R. Henderson and E. Charbon, "A time-resolved, low-noise single-photon image sensor fabricated in deep-submicron CMOS technology," IEEE Journal of Solid-state Circuits, **47**, 1394-1407 (2012).

[19] B.F. Aull, D.R. Schuette, D.J. Young, D.M Craig, B.J. Felton, and K. Warner, "A study of crosstalk in a 256 $\times$ 256 photon counting imager based on silicon Geiger-mode avalanche photodiodes," IEEE Sensors Journal, **15**, 2123-2132 (2015).

[20] M.A. Albota, B.F. Aull, D.G. Fouche, R.M. Heinrichs, D.G. Kocher, R.M. Marino, J.G. Mooney, N.R. Newbury, M.E. O'Brien, B.E. Player, B.C. Willard, and J.J. Zayhowski, "Three-dimensional imaging laser radars with Geiger-mode avalanche photodiode arrays," 2002, Lincoln Laboratory Journal, **13**, 351-370 (2002).

[21] B.F. Aull, A.H. Loomis, D.J. Young, R.M. Heinrichs, B.J. Felton, P.J. Daniels, P.J., and D.J. Landers, "Geiger-mode avalanche photodiodes for three-dimensional imaging," Lincoln Laboratory Journal, **13**, 335-350 (2002).

[22] S. Cova, M. Ghioni, A. Lacaita, C. Samori, and F. Zappa, "Avalanche photodiodes and quenching circuits for single-photon detection," Applied Optics, **35**, 1956-1976 (1996).

[23] A. Gallivanoni, I. Rech, and M. Ghioni, "Progress in quenching circuits for single photon avalanche diodes," IEEE Transactions on Nuclear Science, **57**, 3815-3826 (2010).

[24] D. Bronzi, S. Tisa, F. Villa, S. Bellisai, A. Tosi, and F. Zappa, "Fast sensing and quenching of CMOS SPADs for minimal afterpulsing effects," IEEE Photonics Technology Letters, **25**, 776-779 (2013).

[25] S. Cova, A. Lacaita, M. Ghioni, G. Ripamonte, and T.A. Louis, "20-ps timing resolution with single-photon avalanche diodes," Rev. Sci. Instrum., **60**, 1104-1110 (1989).

[26] A. Lacaita, S. Cova, C. Samori, and M. Ghioni, "Performance optimization of active quenching circuits for picosecond timing with single photon avalanche diodes," Rev. Sci. Instrum., **66**, 4289-4295 (1995).

[27] J.D. Strasburg, T.W. Murphy, C.W. Stubbs, E.G. Adelberger, D.W. Miller, and J.I. Angle, "Lunar laser ranging using avalanche photodiode (APD) arrays," Proc. of the SPIE: Astronomical Instrumentation, **4836**, 387-394 (2002).

[28] P.E. Schmid, "Optical absorption in heavily doped silicon," Phys. Rev. B, **23**, 5531-5536 (1981).

[29] A.G Chynoweth and K.G. McKay, "Photon emission from avalanche breakdown in silicon," Phys. Rev., **102**, 369-376 (1956).

[30] W. Spitzer and H.Y. Fan, "Infrared absorption in n-type silicon," Phys. Rev., **108**, 268-271 (1957).

[31] J.D. Strasburg, dissertation, University of Washington, Department of Physics (2004).

[32] Phil Gregory, *Bayesian Logical Data Analysis for the Physical Sciences*, Cambridge University Press (2005).

[33] Philip R. Bevington and D. Keith Robinson, *Data Reduction and Error Analysis for the Physical Sciences* (second edition), McGraw-Hill (1992).

[34] M.H. Quenouille, "Approximate tests of correlation in time series," J. R. Statist. Soc. B, **11**, 18-84 (1949).

[35] J.W. Tukey, "Bias and confidence in not quite large samples (abstract)," Ann. Math. Statist., **29**, 614 (1958).

[36] B. Efron, "Bootstrap methods: Another look at the jackknife," Ann. Statist, **7**, 1-26 (1977).

[37] C.F.J. Wu, "Jackknife, bootstrap and other resampling plans in regression analysis" (with discussion), Ann. Statist., **14**, 1261-1350 (1986).

[38] D.A. Freedman, "Bootstrapping regression models," Ann. Statist., **9**, 1218-1228 (1981).

[39] D.A. Freedman and S.C. Peters, "Bootstrapping a regression equation: Some empirical results," J. Am. Statist. Soc., **79**, 97-106 (1984).

[40] R.A. Stine, "Bootstrap prediction intervals for regression," J. Am. Statist. Assoc., **82**, 1072-1078 (1985).

[41] B. Efron, "Bootstrap confidence intervals for a class of parametric problems," Biometrika, **72**, 45-58 (1985).

[42] B. Efron, "Second thoughts on the bootstrap," Statist. Sci., **18**, 135-140 (2003).

[43] A.C. Davison, D.V. Hinkley, and G.A. Young, "Recent developments in bootstrap methodology," Statist. Sci., **18**, 141-157 (2003).

[44] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*, Chapman and Hall (1993).

[45] M.R. Chernick, *Bootstrap Methods: A Practitioner's Guide*, John Wiley and Sons (1999).

[46] K. Nordtvedt, Jr., "Equivalence principle for massive bodies. II. Theory," Phys. Rev., **169**, 1017-25 (1968).

[47] C.M. Will, "Theoretical frameworks for testing relativistic gravity. II. Parametrized post-Newtonian hydrodynamics and the Nordtvedt effect," Astrophys. J., **163**, 611-628 (1971).

[48] C.M. Will and K. Nordtvedt, Jr., "Conservation laws and preferred frames in relativistic gravity. I. Preferred-frame theories and an extended PPN formalism," Astrophys. J., **177**, 757-774 (1972).

[49] C.M. Will, *Theory and Experiment in Gravitational Physics* (second edition), Cambridge University Press (1993).

[50] J.G. Wiliams, S.G. Turyshev, and D.H. Boggs, "Lunar laser ranging tests of the equivalence principle," Class. Quantum Grav., **29**, 184004 (2012).

[51] J. Müller, F. Hofmann, and L Biskupek, "Testing various facets of the equivalence principle using lunar laser ranging," Class. Quantum Grav., **29**, 184006 (2012).

[52] J.G. Wiliams, S.G. Turyshev, and D.H. Boggs, "Progress in lunar laser ranging tests of relativistic gravity," Phys. Rev. Lett., **93**, 261101 (2004).