Math 5610 Fall 2020 Term Project[1]

1 Introduction

The primary numerical task in this semester project is the solution of certain systems of nonlinear equations. However, beyond that the project illustrates a recent technological development which I hope you will find interesting and motivating. To get everything to work in this project we will have to put many pieces together, and they have to work just right, as in most real life problems. The results of your work, a couple of programs, will need to work with the programs of your classmates and my own. When we reach that stage we will be able to say with assurance that we truly understand the problem and its solution.

The original US Global Positioning System (GPS) consists of 24 satellites that orbit the earth and constantly transmit their position and the precise time of that transmission, in addition to other data. Commercially available receivers, (*Global Positioning Devices)* process that information and, using the differences in the runtimes of the satellite signals, compute their position and the precise time. The position can be expressed, e.g., as longitude, latitude, and altitude. Its accuracy depends on the type and quality of the device and its software, and other circumstances. It ranges from less than one centimeter to about 100 meters[2].

In this project, we will write two programs that simulate a much simplified version of this system.

GPS was developed originally for the US armed forces, but there are of course also civilian and scientific applications, many of which are still being discovered. As far as volume and resources involved, civilian applications now vastly outstrip military applications. The following is an incomplete list of applications I have read or heard about (or, in the case of the first three applications, actually practiced):

- Navigating in a car. This may be the largest civilian application at present. You can buy devices that will tell you where you are and what's near you, and give you oral turn by turn instructions on how to get to your destination. Many high end models now come with built-in GPS based navigation systems.
- Finding your way in the wilderness.
- Marine navigation.

---

[1] by Peter Alfeld, pa@math.utah.edu

[2] A meter is a little more than 3 feet. Most of the literature on GPS uses the metric system, and we follow that convention.

- *Geocaching* is a high-tech treasure hunting game played throughout the world by adventure seekers equipped with GPS devices. This is a highly popular past time where people search for a cache established by others, given it's coordinates, remove a token item, and replace it with another. The basic idea is to locate hidden containers, called geocaches, outdoors and then share your experiences online. The official web site is http://www.geocaching.com/ .
- Electronically marking a spot (a good fishing place, a buried treasure or mine, man or item overboard) in a featureless area such as the ocean, a lake, a desert, or the Salt Flats.
- Aerial navigation, including precision approach and landing.
- Measuring continental drift and expansion.
- Automatic grading and paving in road construction.
- Telling blind people (via Braille or Audio) where they are.
- Steering a tractor planting vine plants along a suitable trajectory.
- Surveying with high accuracy.
- Controlling a fleet of vehicles (e.g., Police, Taxi, Truck or Bus companies).
- Transfer of extremely accurate time information.
- Installation of several GPS devices in a vehicle, like a ship or plane, and determination not just of position but also of attitude, roll, pitch, and yaw.
- By coupling a cell phone with a GPS device, in an emergency it can tell your location to the dispatcher even if you don't know it yourself. If the phone is installed in your car, and the car is stolen, you might be able to call your car and ask where it is.

  GPS is available anywhere on or near earth, 24 hours a day, in fog, space, or darkness. By making several measurements over time one can compute and display such quantities as speed, bearing, or estimated time of arrival. It's eerie to have your car give you detailed instructions on how to get where you are going, as you are going. Or, for example, you can transform a simple rental boat into a sophisticated yacht with compass and speedometer simply by placing a GPS device next to the wheel.

2 Sources of Information

If you are interested in GPS and like to learn more, here are some places to get started:

  - A standard text on GPS is B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins: *GPS, Theory and Practice*, Springer-Verlag Wien New York, 5th edition, 2001, ISBN 3211835342. This excellent but very technical book gives lots of details and references.

  - Another standard detailed and technical text is Elliott D. Kaplan (ed.), Understanding GPS Principles and Applications, Artech House, Boston, London, 1996, ISBN 0890067937.

- There's a great deal of information on the web. The following Table shows the number of results found by a Google search for *GPS*.

May 2004	17,000,000
May 2005	58,600,000
August 2007	257,000,000
May 2008	407,000,000
August 2009	244,000,000
May 2012	1,330,000,000
May 2013	720,000,000
May 2014	298,000,000
May 2015	620,000,000
July 2017	258,000,000

Table: Number of Google results for ***GPS***

  - There is excellent information in a concise and readable form about the shape of the earth and its gravitational field in the Encyclopedia Britannica, particularly in the article *The Earth* in the macropedia.

  - A useful text giving all kinds of mathematical formulas and information is the *VNR Concise Encyclopedia of Mathematics.* It's published by Van Nostrand Reinhold Company. My copy was published in 1975. The ISBN is 0442226462.

3 Note on Notation

Notation can get quite cumbersome in this project. I use boldface letters to denote vectors in cartesian coordinates. The coordinates of these vectors themselves are lower case Roman letters that may be subscripted. For example, the vector **x** is usually denoted by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{1}$$

but sometimes, for example in the discussion of Newton's method, by

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{2}$$

Subscripts are often used with an S to denote correspondence to a satellite. for example, $\mathbf{x}_{S_i}$ denotes correspondence to the i-th of several satellites.

4 The Space Segment

The number and configuration of the orbiting satellites is evolving. We will consider a set of 24 satellites orbiting in six planes at an inclination of $55°$ to the equator at an altitude of about 20,200 km.

To help you get going on the project, sprinkled throughout this assignment are a total of 15 **exercises**. You will have to solve these exercises or closely related problems during the work on the project, so they collectively form our first home work in this class.

5 The Model

The model will consist of three modules, i.e., programs, the `vehicle`, the `satellite`, and the `receiver`. The `vehicle` generates positional data, the `satellite` converts those into data of the kind processed by GPS receivers, and the `receiver` converts these back into positional data. The three modules are piped together in the standard Unix fashion:

$$\texttt{vehicle} \;\;|\;\; \texttt{satellite} \;\;|\;\; \texttt{receiver} \tag{3}$$

If all goes well the standard output of `receiver` will equal (almost) the standard output of `vehicle`. Your job is to write `satellite` and `receiver` (and perhaps a rudimentary `vehicle` for testing purposes).

The model depends on a set of data provided in a file called

`data.dat`

which is read by `satellite`. The beginning part of that file is also read by `receiver`. Figure 1 summarizes the model. Arrows indicate data flow.

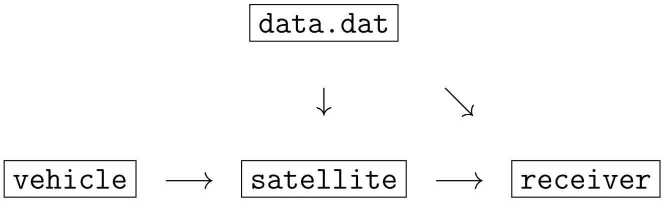

Figure 1: The Model

Following are some of the assumptions we'll make to construct our model. Numerical values are given here for illustration. They will be actually read from `data.dat` and the precise numerical values are subject to change (with appropriate notification). You or I may also change those parameters for the purpose of testing our programs.

1. The Earth is perfectly spherical and has a radius R of

$$R = 6,367,444.50 \text{ m}. \tag{4}$$

2. The Earth turns at a constant rate and completes one revolution in one sidereal day[3] of

$$s = 86,164.09 \text{ seconds}. \tag{5}$$

3. The satellites move at a constant speed in perfectly circular orbits at an altitude of

$$h = 20,200,000 \text{ m} \tag{6}$$

and with an orbital period

$$p = \frac{s}{2} \tag{7}$$

of exactly half a sidereal day.

4. The satellites are evenly spaced in groups of four in six planes, each of which is inclined 55° to the equator. According to Hofmann-Wellenhof et al this provides global coverage of four to eight satellites being more than 15° above the horizon at all times and in all places on earth.

6 Coordinate Systems

We use two coordinate systems: longitude, latitude and altitude to express positions on and near earth, and a cartesian coordinate system to describe the space in which the satellites orbit and the earth rotates. Like commercial global positioning devices, we express angles in degrees, minutes of degree, and seconds of degree. The two coordinate systems are linked by the following conventions:

The North Pole is located at $(0, 0, R)$, the South Pole at $(0, 0, -R)$, (assuming both have altitude 0, which is definitely wrong for the South Pole). At time 0 the point **O** of zero longitude, latitude, and altitude, is at (R,0,0), and the Earth rotates (west to east) once in a sidereal day.

---

[3] A *sidereal day* is the time required for a complete rotation of the earth in reference to a fixed star. It differs from a standard *solar day* of 24 hours because in the course of a day the earth not only rotates, but also changes position in its orbit around the Sun. Hence the sun changes its position in the sky and the earth needs to rotate an additional 4 minutes or so to catch up with the sun before it is noon again.

Exercise 1: Find a formula that describes the trajectory of the point **O** in cartesian coordinates as a function of time.

7 The Programs

As stated above, our model will consist of three programs:

1. **The Vehicle.** This program, which you can download from our Canvas folder, produces a stream of data sets (to standard output) each of which has the form:

$$t_V \quad \psi_d \quad \psi_m \quad \psi_s \quad \mathbf{NS} \quad \lambda_d \quad \lambda_m \quad \lambda_s \quad \mathbf{EW} \quad h \tag{8}$$

   where t_V denotes Universal time[4] in seconds, ψ_d, ψ_m, ψ_s is latitude in degrees, minutes, and seconds, and, similarly, λ_d, λ_m, λ_s is longitude in degrees, minutes, and seconds. More specifically,

   - t_V is a real number given to an accuracy of 10^{-2} seconds ranging from 0 to 10^6. This is the time at which the `vehicle` is at the specified position.
   - ψ_d is an integer ranging from 0° (i.e., the Equator) to +90° (i.e., the North or South Pole).
   - ψ_m is an integer ranging from 0 to 59 minutes of degree.
   - ψ_s is a real number ranging from 0 to 59.9999 seconds of degree. It should be given to an accuracy of 10^{-2} (which corresponds to an accuracy of about a foot).
   - **NS** is an integer that is $+1$ North of the equator and -1 South of the equator.
   - λ_d is an integer ranging from 0° (i.e., the meridian of Greenwich) to 180 (i.e., 180 degrees east, or west, the *date line*).
   - λ_m is an integer ranging from 0 to 59 minutes of degree.

---

[4] The mean solar time of the Greenwich meridian (0° longitude). Universal time replaced the designation Greenwich mean time in 1928; it is now used to denote the solar time when an accuracy of about 1 second suffices. In 1955 the International Astronomical Union defined several categories of Universal Time of successively increasing accuracy. UT0 represents the initial values of Universal Time obtained by optical observations of star transits at various astronomical observatories. These values differ slightly from each other because of the effects of polar motion. UT1, which gives the precise angular coordinate of the Earth about its spin axis, is obtained by correcting UT0 for the effects of polar motion. Finally, an empirical correction to take account of annual changes in the Earth's speed of rotation is added to UT1 to convert it into UT2. Coordinated Universal Time, the international basis of civil and scientific time, is obtained from an atomic clock that is adjusted so as to remain close to UT1; in this way, the solar time that is indicated by Universal time is kept in close coordination with atomic time. [Encyclopedia Britannica, 1992]. We will assume that all the times described here are identical and call them *Universal Time.*

- λ_s is a real number ranging from 0 to 59.9999 seconds of degree, given to the same accuracy as ψ_s.
- **EW** is an integer that is +1 east of Greenwich and −1 west of Greenwich.
- h is a real number giving the altitude in meters, to an accuracy of 1cm.

For example, according to my Magellan Trailblazer the street light labeled B12[5] in front of the South Window of my office (at time t) is located at:

$$t \quad 40 \quad 45 \quad 55.0 \quad 1 \quad 111 \quad 50 \quad 58.0 \quad -1 \quad 1372.00 \tag{9}$$

i.e., at latitude 40° 45' 55" North, longitude 111° 50' 58" West, and an altitude of 1372 m.

Exercise 2: Write a program that converts angles from degrees, minutes, and seconds to radians, and vice versa. Make sure your program does what it's supposed to do.

For the following four exercises assume that t_V equals true Universal time and denote it by t.

Exercise 3: Find a formula that converts position as given in (8) at time $t = 0$ into cartesian coordinates.

Exercise 4: Find a formula that converts position and general time t as given in (8) into cartesian coordinates.

Exercise 5: Find a formula that converts a position given in cartesian coordinates at time $t = 0$ into a position of the form (8).

Exercise 6: Find a formula that converts general time t and a position given in cartesian coordinates into a position of the form (8).

Exercise 7: Find a formula that describes the trajectory of lamp post B12 in cartesian coordinates as a function of time.

---

[5] If you look around on campus you'll see that all street lights have been labeled with a letter and a number. On some lights, the labels have disappeared, or, as in the case of B12, obstructed by an attachment.

The `vehicle` should not produce impossible positions (like a latitude of 90° 59' 59"). For testing purposes we will use `vehicle`s that produce a stream of data corresponding, for example, to a walk from JWB to the Marriott Library, a hike up to Mount Olympus, or a flight from Salt Lake City to the North Pole. The data sets should be spaced at least 1 second (of time) apart.

In addition `vehicle` writes a copy of the standard input and the standard output into the file

`vehicle.log`.

2. **The Satellite.** This program reads data from `data.dat` and then reads the data generated by a `vehicle` and processes those data as follows:

   A. It computes the position $\mathbf{x}_V$ of the vehicle in cartesian coordinates (measured in meters) at the time t_V.

   B. For each satellite S that is above the horizon[6] at the vehicle's position it computes the time t_S at which a signal needs to be sent to reach the vehicle at time t_V and position $\mathbf{x}_V$, assuming the signal moves at the speed c of light in vacuum where

   $$c = 2.99792458 \times 10^8 \text{m/s}. \tag{10}$$

   The satellite needs to be above the horizon at time t_S, The satellite also computes the position $\mathbf{x}_S$ of the satellite at the time t_S It then writes the following data to standard output:

   $$i_S \quad t_S \quad \mathbf{x}_S \tag{11}$$

   where i_S is the reference number of the satellite (ranging from 0 to 23). The time t_S should be given to an accuracy of 10^{-11} seconds, and the position $\mathbf{x}_S$ should be given as a triple of cartesian coordinates with an accuracy of 1cm.

   C. In addition, `satellite` writes a copy of the standard input and the standard output into the file

   `satellite.log`.

   It should terminate gracefully when the data stream from the satellite program is terminated.

Exercise 8: Given a point $\mathbf{x}$ on earth and a point $\mathbf{s}$ in space, both in cartesian coordinates, find a condition that tells you whether $\mathbf{s}$ as viewed from $\mathbf{x}$ is above the horizon.

Exercise 9: Discuss how to compute t_S and $\mathbf{x}_S$.

---

[6] My own `receiver` will check this condition. I figured that assuming a transparent earth would be too much of a simplification. However, calculations in the actual GPS system are usually based on satellites at least 15° above the horizon.

3. **The Receiver.** This program reads the data written by `satellite` and converts them into position and time data in the same form as the data generated by the vehicle. If all goes well those data should closely approximate the data sent by `vehicle`. Note that `satellite` does not transfer t_V. This has to be reconstructed by `receiver`, thus simulating the transfer of precise time. In addition the program writes a copy of the standard input and the standard output into the file

   `receiver.log`.

   It should terminate gracefully when the data stream from the satellite program is terminated.

   Letting $\mathbf{x}_v$ denote the position at which the vehicle receives the satellite signal, t_V the time that it receives the signal, and, similarly, $\mathbf{x}_S$ and t_S the position and time at which the satellite sends the signal, the basic equation for our system is

   $$\|\mathbf{x}_V - \mathbf{x}_S\| = c(t_V - t_S). \tag{12}$$

   In this equation we know t_S and $\mathbf{x}_S$, and we don't know t_V and $\mathbf{x}_V$. We have one such equation for each satellite from which we receive a signal.

   The time t_V enters these equations linearly, and can be eliminated. It also plays a different role than the coordinates of $\mathbf{x}_V$. It needs to be computed to a much higher accuracy than the location. So it is indeed best to eliminate it. Suppose we have equations of the form (12) for two satellites S_1 and S_2. Taking the difference gives the equation

   $$\|\mathbf{x}_V - \mathbf{x}_{S_1}\| - \|\mathbf{x}_V - \mathbf{x}_{S_2}\| = c(t_{S_2} - t_{S_1}). \tag{13}$$

Exercise 10: Suppose you have data of the from (11) from 4 satellites. Write down a set of four equations whose solutions are the position of the vehicle in cartesian coordinates, and t_V.

Usually you will have data from more than four satellites. To use the above approach one would have to choose four specific ones of those satellites. This is in itself a nontrivial problem. A better way is to use all observations and to think of the (possibly overdetermined) nonlinear system of equations

$$F(\mathbf{x}) = 0 \tag{14}$$

as a nonlinear Least Squares problem

$$f(\mathbf{x}) = F(\mathbf{x})^T F(\mathbf{x}) = \min . \tag{15}$$

The least squares problem can be solved by setting the gradient of f in (15) to zero, which gives rise to another system of four equations in four unknowns.

If m is the number of satellites being received, I recommend that you form $m - 1$ equations of the form (13) and apply the Least Squares approach to find $\mathbf{x}_V$. Once you have $\mathbf{x}_V$ you can compute t_V.

Exercise 11: Suppose you have data of the form (11) from more than 4 satellites. Write down a least squares problem whose solution the position of the vehicle in cartesian coordinates, and t_V.

Exercise 12: Think about the *ground track* of satellite 1, i.e., the position in geographic coordinates directly underneath the satellite on the surface of the earth, as a function of time. Do you notice anything particular? What is the significance of the orbital period being exactly one half sidereal day?

8 Mathematical Approach

Mathematically, we have four unknowns: three coordinates of the location, and time. Each satellite signal provides a range $c(t_S - t_V)$ where we don't know t_V. Given four data we should be able to compute position and time. So `receiver` has to solve a system of four nonlinear equations for the four unknowns. In addition the programs will have to do some coordinate conversions.

The natural way (why?) to solve the nonlinear equations arising in this project is by Newton's Method (for systems of nonlinear equations). We will discuss it in detail in class. However, for reference here is a preliminary description: Let $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ be a given function, and suppose we want to solve the nonlinear system of equations

$$F(\mathbf{x}) = 0. \tag{16}$$

Let J denote the *Jacobian Matrix* of F. Thus the (i, j) entry of J is the partial derivative of the i-th component of F with respect to the j-th component of $\mathbf{x}$, i.e.,

$$J(\mathbf{x}) = \left[\frac{\partial F_i}{\partial x_j}(\mathbf{x})\right]_{i,j=1,\ldots,n.} \tag{17}$$

Given a starting point $\mathbf{x}^{(0)}$, Newton's method proceeds iteratively by defining

$$\begin{aligned} &\text{for} \quad k = 0, 1, 2, \cdots, \quad \text{until satisfied} \\ &\qquad \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \left[J\left(\mathbf{x}^{(k)}\right)\right]^{(-1)} F\left(\mathbf{x}^{(k)}\right). \end{aligned} \tag{18}$$

Of course, we never invert a matrix. Instead, your program should do something like this

$$\begin{aligned} &\text{For } k = 0, 1, 2, \cdots \text{ until satisfied :} \\ &\qquad \text{Solve} \quad J\left(\mathbf{x}^{(k)}\right) \mathbf{s}^{(k)} = -F\left(\mathbf{x}^{(k)}\right) \\ &\qquad\qquad \text{Let} \quad \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)} \end{aligned} \tag{19}$$

The performance of Newton's method is very sensitive to the choice of the starting point $\mathbf{x}^{(0)}$. As always, the precise choice of that point depends on the problem. For our purposes you may assume that at least initially you are operating your GPS device in the vicinity of Salt Lake City, i.e., you can use (9) as a starting point.

Exercise 13: Find a precise description of Newton's method as it is applied to the nonlinear system obtained by processing data from 4 satellites, as derived in an earlier exercise. Your answer should include an explicit specification of the derivatives involved.

Exercise 14: Similarly, find Newton's method for the nonlinear system obtained from the least squares approach. Again, your answer should include an explicit specification of the derivatives involved.

A common problem with nonlinear equations is non-existence of a solution, or existence of multiple solutions. For our project, for example, if the difference in run times of two satellite signals exceeds the time corresponding to the maximum distance of the satellites then there is no solution. On the other hand, the way we set up the problem there should always be a solution since we start with a well defined position. Non-existence of a solution would therefore indicate a programming error of some sort. It is also possible to have several solutions. For simplicity, consider a situation where you are given the *distances* (i.e., the receiver has an exact clock) from three satellites at known positions. So your location is on the intersection of three spheres (with the satellites as their centers and the known distances as their radii). Two spheres intersect in a circle, and the third sphere intersects the circle in two points. There are thus two solutions of the nonlinear equations. Which should we decide on as our current location? In practice, global positioning devices use the following criteria: closeness to the previously computed position, closeness to the surface of the earth, and slow motion.

Exercise 15: Think about the number of solutions obtained by analyzing four satellite signals with an unknown vehicle time t_V. This is an open ended question that will not be graded!

Here are some technical comments on writing the programs:

- Use the highest available accuracy throughout, e.g., use the *double* data type in java. Keep in mind that roughly speaking a meter makes a difference in the 7th or 8th digit of the distances involved. Light takes about 3×10^{-9} seconds to pass that distance, and since we contemplate times ranging up to one million seconds we need to process time information accurately to at least 15 digits.
- Make sure your standard output carries enough digits.

- Since your programs will have to work with those written by others you have to follow carefully the conventions laid out here. `vehicle` and `satellite` can't write anything to standard output other than what's specified (and for better comparability, `receiver` also should not write anything else).

- Your `satellite` should read the data file in the form given in the appendix and should not be deterred by the embedded comments.

- `satellite` will send data for all visible satellites. `receiver` does not know beforehand the number of those satellites. It will therefore have to decide when a group of signals starts and ends, and it will have to take reasonable action if the number of satellite signals is different from 4. If it's less it could write a message saying that there isn't enough information, if there are more it could select a subset of 4 signals, or it could solve a least squares problem. My own inclination is to set up a least squares problem and solve it whenever the number of satellite signals is 4 or greater.

- You should have your `receiver` check whether or not your `satellite` sends only signals from satellites that are above the horizon. Otherwise you may be simulating a technical breakthrough that renders the earth transparent, but your `satellite` will not work with my less advanced `receiver`.

- Both `receiver` and `satellite` should terminate gracefully when they receive no more input. Thus they should shut down without irrelevant system messages, and they should close their log file before terminating.

9 What To Do

1. Work the exercises in this assignment, and hand in your answers.

2. Write the programs `satellite` and `receiver`. You will need a rudimentary `vehicle` for testing purposes. You can download one from the Canvas folder or you can write your own. Your programs must run on all or some of our Unix systems. They may be written in any language, but of course they must accommodate the Unix standard input and output concepts. E-mail your programs and a pdf file with a written report of your work to me, by the deadline of this project. The report should be suitable for distribution among your class mates. Describe your mathematical approach, your software, how to compile and use it, the lessons you learned in this project, and anything else you may find worth mentioning.

3. After I test and process your programs we will have a discussion of this project. I will ask some (or all if practical) teams or individuals to present part of their work to the whole class. Details will be announced as we approach the deadline.

10 Note on Team Work

As mentioned in class, I recommend you work with one or two partners, to study together, and to work on the term project and the home works together. If you do work in a team hand in just one set of answers. Each of you will get the same score on all parts of an assignment.

11 Simplifying Assumptions

As additional information this sections contains an incomplete list of phenomena that have to be incorporated to make GPS viable in practice. This information is taken mostly from Hofmann-Wellenhof et al which gives detailed mathematical descriptions of most of these effects.

- **Signal Acquisition.** The satellite signals are weak and the signal to noise ratio is low. As a consequence the receivers work with a very narrow bandwidth. Due to the Doppler effect the frequency at which the signal is sent varies across a spectrum significantly wider than the bandwidth. Thus the receiver doesn't know the precise frequency at which the satellite is transmitting unless it knows the satellite's position and velocity. As a consequence, if starting from scratch, signal acquisition may take quite a while. The problem is alleviated by the fact that each satellite sends information on the current location of all other satellites (this information is called *ephemerides*).

- **Positioning with Doppler data or Carrier Phases.** Our programs model positioning with *range data*. More accurate positioning can be accomplished by taking into account Doppler data and carrier phases. The latter determine distance only within an integer multiple of the wavelength (which is 19.0cm or 24.4 cm). However, phases can be measured to better than 1% of wavelength, which potentially gives GPS an accuracy of about 1mm or less.

- **Relative Positioning.** In many applications the important item to be measured is the vector between two points, rather than the location of a point with respect to a global coordinate system. This can usually be done more accurately than the determination of a position.

- **Selected Availability and Anti Spoofing.** In the interest of national security the Department of Defense used to limit availability and accuracy of GPS to civilian users by encrypting the high accuracy version of the satellite signal (selected availability) and by dithering (distorting) the public signal (anti-spoofing). On May 2, 2000, these measures were turned off and the full accuracy of GPS is now available to the public. Interestingly, while selected availability or anti-spoofing were still in effect, they were turned off during crises like the first gulf war or the invasion of

Haiti, to make it possible for the armed forces to make full use of commercially available GPS devices.

- **Spherical Earth.** The Earth is of course not spherical. It is more accurately described as an ellipsoid with a semimajor axis of 6,378,137 m and a semiminor axis of 6,356,752 m. When contemplating the determination of altitude within a cm or so that definition is also inadequate. What's really involved here is a precise definition of the term "sea level". The resulting shape of the earth is called the "datum" underlying a map or other navigational tool. For example, the Magellan Trailblazer can use any of 12 built-in datums.

- **Constant axis of rotation.** The North and South Poles move (at the *Chandler period* of about 430 days) within an area that has a diameter of roughly 6 m.

- **Circular Orbits.** The satellites do not move along circular or even elliptical orbits. The orbits are disturbed by a number of factors: non-spherical earth, gravitational anomalies, tidal forces, solar radiation pressure, and even air drag. The problem is overcome by the satellites broadcasting their actual position (rather than one based on an orbit calculation), but this needs to be determined using a sophisticated ground and space based infrastructure.

- **Time Dilation.** Time passes more slowly for an object that is moving at a high speed. This affects the accuracy of the satellite clocks.

- **Time of transmission.** We have to define precisely what we mean by the time of transmission of a signal. Each relevant signal consists of 1023 bits that are separated by about 10^{-6} seconds. Thus at the speed of light the distance between two bits is about 300m. This is called the "chip length" (and the bits are called "chips"). Current timing accuracy is between 0.1% and 1% of the chip length.

- **Ionospheric Refraction.** This is a major cause of inaccuracies. Its precise effects depend for example on current sunspot activity.

- **Tropospheric Refraction.** This depends largely on the amount of water in the various parts of the troposphere, i.e., on the weather.

- **Multipath Effects.** A receiver may receive more than one version of the signal due to reflection (e.g., on walls or cliffs).

- **Position of what?** GPS receivers are larger than the potential accuracy with which they can be used. Thus one has to figure out exactly which point (called the *phase center,* and located usually within the antenna) corresponds to the computed solution. The location of the phase center of course depends on the geometry of the antenna and receiver, but also on the frequency and the intensity of the signal.

- **Physical Rigor.** Often GPS devices are employed in demanding conditions (e.g., rain, temperature extremes, dirt, vibrations). They need to be physically rugged.
- **Software.** The quality of a GPS device depends very much on its software. The cost of software development constitutes a very significant portion of the total purchase price.
- **Utility.** The utility of a device depends much on what it can do beyond computing position (and perhaps speed and direction). For example a device might contain tables of magnetic deviations from true North (like the Magellan Trail Blazer) or a database of (almost) all streets in the United States (like the Magellan GPS Map 7000) and so be able to give magnetic directions or a map with your current location.

12 Appendix: The Data File

The whole project depends on a few parameters, like the speed of light, the radius of earth, and the orbits of the satellites. Rather than hardwiring these data into the programs we'll put them into a file that can then be easily modified. For example, if we wanted to run GPS on Jupiter or the Sun we would only have to modify that file.

We express the (perfectly circular) orbit of a satellite as

$$\mathbf{x}(t) = (R+h)\left[\mathbf{u}\cos\left(\frac{2\pi t}{p}+\theta\right)+\mathbf{v}\sin\left(\frac{2\pi t}{p}+\theta\right)\right] \tag{20}$$

where:

- t is time measured in seconds
- $\mathbf{x}(t)$ is the location of a satellite at time t expressed in cartesian coordinates with meters being the unit of length
- R is the radius of the earth
- h is the altitude of the satellite
- $\mathbf{u}$ is a unit vector
- $\mathbf{v}$ is a unit vector that is orthogonal to u
- p is the periodicity of the orbit (assumed to be half a sidereal day)
- θ is the phase of the orbit.

Crucial to the project is the data file listed in Table 1. (The leading line numbers are not part of the file.) You can download it from the Canvas folder.

```
3.141592653589793116E+00           /= pi
2.997924580000000000E+08           /= c, speed of light, [m/s]
6.367444500000000000E+06           /= R, radius of earth, [m]
8.616408999999999651E+04           /= s, length of a sidereal day, [s]
1.000000000000000000E+00           /= u1 of Sat.  0
0.000000000000000000E+00           /= u2 of Sat.  0
0.000000000000000000E+00           /= u3 of Sat.  0
0.000000000000000000E+00           /= v1 of Sat.  0
5.735764363510461594E-01           /= v2 of Sat.  0
8.191520442889917986E-01           /= v3 of Sat.  0
4.308204499999999825E+04           /= periodicity of Sat.  0 [s]
2.020000000000000000E+07           /= altitude of Sat.  0 [m]
0.000000000000000000E+00           /= phase of Sat.  0 [rad]
1.000000000000000000E+00           /= u1 of Sat.  1
0.000000000000000000E+00           /= u2 of Sat.  1
0.000000000000000000E+00           /= u3 of Sat.  1
0.000000000000000000E+00           /= v1 of Sat.  1
5.735764363510461594E-01           /= v2 of Sat.  1
8.191520442889917986E-01           /= v3 of Sat.  1
4.308204499999999825E+04           /= periodicity of Sat.  1 [s]
2.020000000000000000E+07           /= altitude of Sat.  1 [m]
1.570796326794896558E+00           /= phase of Sat.  1 [rad]
1.000000000000000000E+00           /= u1 of Sat.  2
0.000000000000000000E+00           /= u2 of Sat.  2
0.000000000000000000E+00           /= u3 of Sat.  2
0.000000000000000000E+00           /= v1 of Sat.  2
5.735764363510461594E-01           /= v2 of Sat.  2
8.191520442889917986E-01           /= v3 of Sat.  2
4.308204499999999825E+04           /= periodicity of Sat.  2 [s]
2.020000000000000000E+07           /= altitude of Sat.  2 [m]
3.141592653589793116E+00           /= phase of Sat.  2 [rad]
1.000000000000000000E+00           /= u1 of Sat.  3
0.000000000000000000E+00           /= u2 of Sat.  3
0.000000000000000000E+00           /= u3 of Sat.  3
0.000000000000000000E+00           /= v1 of Sat.  3
5.735764363510461594E-01           /= v2 of Sat.  3
8.191520442889917986E-01           /= v3 of Sat.  3
4.308204499999999825E+04           /= periodicity of Sat.  3 [s]
2.020000000000000000E+07           /= altitude of Sat.  3 [m]
4.712388980384689674E+00           /= phase of Sat.  3 [rad]
5.000000000000000000E-01           /= u1 of Sat.  4
8.660254037844385966E-01           /= u2 of Sat.  4
```

```
 0.000000000000000000E+00          /= u3 of Sat.  4
-4.967317648921540929E-01          /= v1 of Sat.  4
 2.867882181755230797E-01          /= v2 of Sat.  4
 8.191520442889917986E-01          /= v3 of Sat.  4
 4.308204499999999825E+04          /= periodicity of Sat.  4 [s]
 2.020000000000000000E+07          /= altitude of Sat.  4 [m]
 1.000000000000000000E+00          /= phase of Sat.  4 [rad]
 5.000000000000000000E-01          /= u1 of Sat.  5
 8.660254037844385966E-01          /= u2 of Sat.  5
 0.000000000000000000E+00          /= u3 of Sat.  5
-4.967317648921540929E-01          /= v1 of Sat.  5
 2.867882181755230797E-01          /= v2 of Sat.  5
 8.191520442889917986E-01          /= v3 of Sat.  5
 4.308204499999999825E+04          /= periodicity of Sat.  5 [s]
 2.020000000000000000E+07          /= altitude of Sat.  5 [m]
 2.570796326794896558E+00          /= phase of Sat.  5 [rad]
 5.000000000000000000E-01          /= u1 of Sat.  6
 8.660254037844385966E-01          /= u2 of Sat.  6
 0.000000000000000000E+00          /= u3 of Sat.  6
-4.967317648921540929E-01          /= v1 of Sat.  6
 2.867882181755230797E-01          /= v2 of Sat.  6
 8.191520442889917986E-01          /= v3 of Sat.  6
 4.308204499999999825E+04          /= periodicity of Sat.  6 [s]
 2.020000000000000000E+07          /= altitude of Sat.  6 [m]
 4.141592653589793116E+00          /= phase of Sat.  6 [rad]
 5.000000000000000000E-01          /= u1 of Sat.  7
 8.660254037844385966E-01          /= u2 of Sat.  7
 0.000000000000000000E+00          /= u3 of Sat.  7
-4.967317648921540929E-01          /= v1 of Sat.  7
 2.867882181755230797E-01          /= v2 of Sat.  7
 8.191520442889917986E-01          /= v3 of Sat.  7
 4.308204499999999825E+04          /= periodicity of Sat.  7 [s]
 2.020000000000000000E+07          /= altitude of Sat.  7 [m]
 5.712388980384689674E+00          /= phase of Sat.  7 [rad]
-4.999999999999998890E-01          /= u1 of Sat.  8
 8.660254037844385966E-01          /= u2 of Sat.  8
 0.000000000000000000E+00          /= u3 of Sat.  8
-4.967317648921540929E-01          /= v1 of Sat.  8
-2.867882181755230242E-01          /= v2 of Sat.  8
 8.191520442889917986E-01          /= v3 of Sat.  8
 4.308204499999999825E+04          /= periodicity of Sat.  8 [s]
 2.020000000000000000E+07          /= altitude of Sat.  8 [m]
```

```
 2.000000000000000000E+00           /= phase of Sat.  8 [rad]
-4.999999999999998890E-01           /= u1 of Sat.  9
 8.660254037844385966E-01           /= u2 of Sat.  9
 0.000000000000000000E+00           /= u3 of Sat.  9
-4.967317648921540929E-01           /= v1 of Sat.  9
-2.867882181755230242E-01           /= v2 of Sat.  9
 8.191520442889917986E-01           /= v3 of Sat.  9
 4.308204499999999825E+04           /= periodicity of Sat.  9 [s]
 2.020000000000000000E+07           /= altitude of Sat.  9 [m]
 3.570796326794896558E+00           /= phase of Sat.  9 [rad]
-4.999999999999998890E-01           /= u1 of Sat. 10
 8.660254037844385966E-01           /= u2 of Sat. 10
 0.000000000000000000E+00           /= u3 of Sat. 10
-4.967317648921540929E-01           /= v1 of Sat. 10
-2.867882181755230242E-01           /= v2 of Sat. 10
 8.191520442889917986E-01           /= v3 of Sat. 10
 4.308204499999999825E+04           /= periodicity of Sat. 10 [s]
 2.020000000000000000E+07           /= altitude of Sat. 10 [m]
 5.141592653589793116E+00           /= phase of Sat. 10 [rad]
-4.999999999999998890E-01           /= u1 of Sat. 11
 8.660254037844385966E-01           /= u2 of Sat. 11
 0.000000000000000000E+00           /= u3 of Sat. 11
-4.967317648921540929E-01           /= v1 of Sat. 11
-2.867882181755230242E-01           /= v2 of Sat. 11
 8.191520442889917986E-01           /= v3 of Sat. 11
 4.308204499999999825E+04           /= periodicity of Sat. 11 [s]
 2.020000000000000000E+07           /= altitude of Sat. 11 [m]
 6.712388980384689674E+00           /= phase of Sat. 11 [rad]
-9.999999999999997780E-01           /= u1 of Sat. 12
 1.110223024625156540E-16           /= u2 of Sat. 12
 0.000000000000000000E+00           /= u3 of Sat. 12
-5.551115123125782702E-17           /= v1 of Sat. 12
-5.735764363510460484E-01           /= v2 of Sat. 12
 8.191520442889917986E-01           /= v3 of Sat. 12
 4.308204499999999825E+04           /= periodicity of Sat. 12 [s]
 2.020000000000000000E+07           /= altitude of Sat. 12 [m]
 3.000000000000000000E+00           /= phase of Sat. 12 [rad]
-9.999999999999997780E-01           /= u1 of Sat. 13
 1.110223024625156540E-16           /= u2 of Sat. 13
 0.000000000000000000E+00           /= u3 of Sat. 13
-5.551115123125782702E-17           /= v1 of Sat. 13
-5.735764363510460484E-01           /= v2 of Sat. 13
```

```
 8.191520442889917986E-01           /= v3 of Sat. 13
 4.308204499999999825E+04           /= periodicity of Sat. 13 [s]
 2.020000000000000000E+07           /= altitude of Sat. 13 [m]
 4.570796326794896558E+00           /= phase of Sat. 13 [rad]
-9.999999999999997780E-01           /= u1 of Sat. 14
 1.110223024625156540E-16           /= u2 of Sat. 14
 0.000000000000000000E+00           /= u3 of Sat. 14
-5.551115123125782702E-17           /= v1 of Sat. 14
-5.735764363510460484E-01           /= v2 of Sat. 14
 8.191520442889917986E-01           /= v3 of Sat. 14
 4.308204499999999825E+04           /= periodicity of Sat. 14 [s]
 2.020000000000000000E+07           /= altitude of Sat. 14 [m]
 6.141592653589793116E+00           /= phase of Sat. 14 [rad]
-9.999999999999997780E-01           /= u1 of Sat. 15
 1.110223024625156540E-16           /= u2 of Sat. 15
 0.000000000000000000E+00           /= u3 of Sat. 15
-5.551115123125782702E-17           /= v1 of Sat. 15
-5.735764363510460484E-01           /= v2 of Sat. 15
 8.191520442889917986E-01           /= v3 of Sat. 15
 4.308204499999999825E+04           /= periodicity of Sat. 15 [s]
 2.020000000000000000E+07           /= altitude of Sat. 15 [m]
 7.712388980384689674E+00           /= phase of Sat. 15 [rad]
-5.000000000000000000E-01           /= u1 of Sat. 16
-8.660254037844383745E-01           /= u2 of Sat. 16
 0.000000000000000000E+00           /= u3 of Sat. 16
 4.967317648921539819E-01           /= v1 of Sat. 16
-2.867882181755230797E-01           /= v2 of Sat. 16
 8.191520442889917986E-01           /= v3 of Sat. 16
 4.308204499999999825E+04           /= periodicity of Sat. 16 [s]
 2.020000000000000000E+07           /= altitude of Sat. 16 [m]
 4.000000000000000000E+00           /= phase of Sat. 16 [rad]
-5.000000000000000000E-01           /= u1 of Sat. 17
-8.660254037844383745E-01           /= u2 of Sat. 17
 0.000000000000000000E+00           /= u3 of Sat. 17
 4.967317648921539819E-01           /= v1 of Sat. 17
-2.867882181755230797E-01           /= v2 of Sat. 17
 8.191520442889917986E-01           /= v3 of Sat. 17
 4.308204499999999825E+04           /= periodicity of Sat. 17 [s]
 2.020000000000000000E+07           /= altitude of Sat. 17 [m]
 5.570796326794896558E+00           /= phase of Sat. 17 [rad]
-5.000000000000000000E-01           /= u1 of Sat. 18
-8.660254037844383745E-01           /= u2 of Sat. 18
```

```
 0.000000000000000000E+00           /= u3 of Sat. 18
 4.967317648921539819E-01           /= v1 of Sat. 18
-2.867882181755230797E-01           /= v2 of Sat. 18
 8.191520442889917986E-01           /= v3 of Sat. 18
 4.308204499999999825E+04           /= periodicity of Sat. 18 [s]
 2.020000000000000000E+07           /= altitude of Sat. 18 [m]
 7.141592653589793116E+00           /= phase of Sat. 18 [rad]
-5.000000000000000000E-01           /= u1 of Sat. 19
-8.660254037844383745E-01           /= u2 of Sat. 19
 0.000000000000000000E+00           /= u3 of Sat. 19
 4.967317648921539819E-01           /= v1 of Sat. 19
-2.867882181755230797E-01           /= v2 of Sat. 19
 8.191520442889917986E-01           /= v3 of Sat. 19
 4.308204499999999825E+04           /= periodicity of Sat. 19 [s]
 2.020000000000000000E+07           /= altitude of Sat. 19 [m]
 8.712388980384689674E+00           /= phase of Sat. 19 [rad]
 4.999999999999996669E-01           /= u1 of Sat. 20
-8.660254037844384856E-01           /= u2 of Sat. 20
 0.000000000000000000E+00           /= u3 of Sat. 20
 4.967317648921540374E-01           /= v1 of Sat. 20
 2.867882181755229132E-01           /= v2 of Sat. 20
 8.191520442889917986E-01           /= v3 of Sat. 20
 4.308204499999999825E+04           /= periodicity of Sat. 20 [s]
 2.020000000000000000E+07           /= altitude of Sat. 20 [m]
 5.000000000000000000E+00           /= phase of Sat. 20 [rad]
 4.999999999999996669E-01           /= u1 of Sat. 21
-8.660254037844384856E-01           /= u2 of Sat. 21
 0.000000000000000000E+00           /= u3 of Sat. 21
 4.967317648921540374E-01           /= v1 of Sat. 21
 2.867882181755229132E-01           /= v2 of Sat. 21
 8.191520442889917986E-01           /= v3 of Sat. 21
 4.308204499999999825E+04           /= periodicity of Sat. 21 [s]
 2.020000000000000000E+07           /= altitude of Sat. 21 [m]
 6.570796326794896558E+00           /= phase of Sat. 21 [rad]
 4.999999999999996669E-01           /= u1 of Sat. 22
-8.660254037844384856E-01           /= u2 of Sat. 22
 0.000000000000000000E+00           /= u3 of Sat. 22
 4.967317648921540374E-01           /= v1 of Sat. 22
 2.867882181755229132E-01           /= v2 of Sat. 22
 8.191520442889917986E-01           /= v3 of Sat. 22
 4.308204499999999825E+04           /= periodicity of Sat. 22 [s]
 2.020000000000000000E+07           /= altitude of Sat. 22 [m]
```

```
 8.141592653589793116E+00           /= phase of Sat. 22 [rad]
 4.999999999999996669E-01           /= u1 of Sat. 23
-8.660254037844384856E-01           /= u2 of Sat. 23
 0.000000000000000000E+00           /= u3 of Sat. 23
 4.967317648921540374E-01           /= v1 of Sat. 23
 2.867882181755229132E-01           /= v2 of Sat. 23
 8.191520442889917986E-01           /= v3 of Sat. 23
 4.308204499999999825E+04           /= periodicity of Sat. 23 [s]
 2.020000000000000000E+07           /= altitude of Sat. 23 [m]
 9.712388980384689674E+00           /= phase of Sat. 23 [rad]
```

Table 1. The file `data.dat`.

You should call it `data.dat` and have your `satellite` read it and use the data. Your `receiver` should use the data contained in the first four lines of that file.