Chapter 7

NONLINEAR PROGRAMMING AND ENGINEERING APPLICATIONS

Robert J. Vanderbei
Princeton University
Princeton NJ 08544
rvdb@princeton.edu

Abstract The last decade has seen dramatic strides in ones ability to solve nonlinear programming problems. In this chapter, we review a few applications of nonlinear programming to interesting, and in some cases important, engineering problems.

Keywords: Sample, edited book

Introduction

Modern interior-point methods for nonlinear programming have their roots in linear programming and most of this algorithmic work comes from the operations research community which is largely associated with solving the complex problems that arise in the business world. However, engineers and scientists also need to solve nonlinear optimization problems. While it is true that many engineers are well aware of the activities of SIAM and therefore are at least somewhat familiar with the advances that have taken place in optimization in the past two decades, there are many engineers and scientists who think that the only tool for solving an optimization problem is to code up something for MATLAB to solve using a genetic algorithm. Often the results are extraordinarily slow in coming and sometimes not even the results that one was looking for.

Having been immersed for more than a decade in the development of algorithms, and software, for solving nonlinear optimization problems, I eventually

developed a desire to see the other side, the user's side. I became quite interested in applications not just as little toys to illustrate the power of my pet algorithms but rather as important problems that require solution and for which some of my tools might be useful. In this chapter I will chronicle some of these applications and illustrate how modern interior-point methods can be very useful to scientists and engineers.

The next section provides a brief description of an interior-point algorithms for nonlinear programming. Then, in subsequent sections we will discuss the following four application areas:

- Finite Impulse Response (FIR) filter design
- Telescope design—optics
- Telescope design—truss structure
- Stable orbits for the n-body problem

1. LOQO: An Interior-Point Code for NLP

Operations Research, our field of study, got its start more than 50 years ago roughly when George Dantzig invented the simplex method for solving linear programming problems. The simplex method together with the development of the computer provided a new extremely powerful tool for solving complex decision making problems. Even today, the simplex method is an indispensable tool to the operations researcher.

Of course, it was fairly soon after the invention that people began to realize that the linear programming problem was too restrictive for most of the real-world problems that needed to be solved. Many problems have the extra constraint that some or all of the variables need to be integer valued. Thus was born the field of integer programming. Even more problems involve nonlinear functions in the objective function and/or in the constraints and so there also arose the subject of nonlinear programming. The simplex method has played a critical role in both of these directions of generalization. For integer programming, the simplex method is used as a core engine in cutting-plane, branch-and-bound, and branch-and-cut algorithms. Some of these algorithms have proved to be very effective at solving some amazingly difficult integer programming problems. For nonlinear programming, the ideas behind the simplex method, namely the idea of active and inactive variables, were extended to this broader class of problems. For many years, the software package called MINOS, which implemented these ideas, was the best and most-used software for solving constrained nonlinear optimization problems. Its descendent, SNOPT, remains a very important tool even today.

In the mid 1980's, N. Karmarkar NKK invented a new algorithm for linear programming. It was totally unlike the simplex method. He proved that this

algorithm has polynomial time worst-case complexity—something that has not yet been established for any variant of the simplex method. Furthermore, he claimed that the algorithm would also be very good in its average-case performance and that it would compete with, perhaps even replace, the simplex method as the method of choice for linear programming. True enough, this new class of algorithms, which we now call *interior-point methods*, did prove to be competitive. But, the new algorithm did not uniformly dominate the old simplex method and even today the simplex method, as embodied by the commercial software package called CPLEX, remains the most-used method for solving linear programming problems. Furthermore, interior-point methods have not proved to be effective for solving integer programming problems. The tricks that allow one to use the simplex method to solve integer programming problems depends critically on being able to solve large numbers of similar linear programming problems very quickly. The simplex method has the nice feature that solving a second instance of a problem starting from the solution to a first instance is often orders of magnitude faster than simply solving the second instance from scratch. There is no analogous property for interior-point methods and so today the simplex method remains the best method for solving integer programming problems.

So, do interior-point methods have a natural extension to nonlinear programming and, if so, how do they compare to the natural extension of the simplex method to such problem? Here, the answers are much more satisfactory. The answer is: yes, there is a very natural extension and, yes, the methods perform very well in this context. In fact, interior-point methods are really best understood as methods for constrained convex nonlinear optimization.

I have for many years been one of the principle developers of a particular piece of software, called LOQO, which implements an interior-point algorithm for nonlinear programming. In the remainder of this section, I will give a brief review of the algorithm as implemented in this piece of software. The basic family of problems that we wish to solve are given by

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & b \leq h(x) \leq b + r, \\ & l \leq x \leq u, \end{array}$$

where b, h, and r take values in $\mathbb{R}^m$ and l, x, and u take values in $\mathbb{R}^m$. We assume that the functions $f(x)$ and $h(x)$ must be twice differentiable (at least at points of evaluation) but not necessarily convex or concave.

The standard *interior-point paradigm* can be described as follows:

- Add slacks thereby replacing all inequalities with nonnegativities.

- Replace nonnegativities with logarithmic barrier terms in objective; that is, terms of the form $-\mu \log(s)$ where μ is a positive *barrier parameter* and s is a slack variable.
- Write first-order optimality conditions for the (equality constrained) barrier problem.
- Rewrite the optimality conditions in primal-dual symmetric form (this is the only step that requires linear programming for its intuition).
- Use Newton's method to derive search directions. Here is the resulting linear system of equations:

$$\begin{bmatrix} -H(x,y) - D & A^T(x) \\ A(x) & E \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \nabla f(x) - A^T(x)y \\ -h(x) + \mu Y^{-1} e \end{bmatrix}.$$

  Matrices D and E are diagonal matrices involving slack variables,

$$H(x,y) = \nabla^2 f(x) - \sum_{i=1}^{m} y_i \nabla^2 h_i(x) + \lambda I, \text{ and } A(x) = \nabla h(x),$$

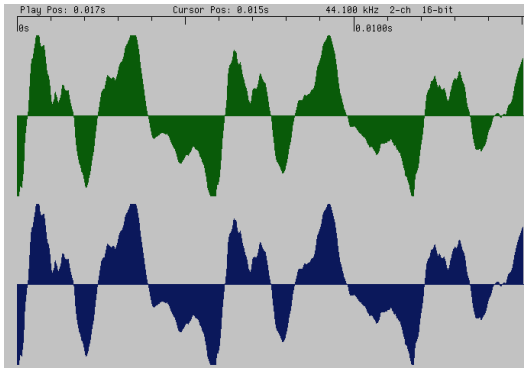
  where λ is chosen to ensure appropriate descent properties.
- Compute a step length that ensures positivity of slack variables.
- Shorten steps further to ensure a reduction in either infeasibility or in the barrier function.
- Step to new point and repeat.

Further details about the algorithm can be found in LMS94; Van97d.

2. Digital Audio Filters

A digital audio signal is a stream of integers, which represents a discretization both in time and in amplitude of an analog signal. For CD-quality sound, there are 44,100 samples per second each sample being a short integer (i.e., an integer between $-32,768$ and $+32,767$). Of course, for stereo there are two such streams of integers. Figure 7.1 shows an example of a very short stretch of music.

A digital audio signal is read off from a CD-ROM device, converted back to analog, amplified, and then sent to a speaker where the induced displacement creates a sound wave that is a replication of the original signal. A speaker that accurately reproduces low frequency signals generally does a bad job at high frequencies and vice versa. Therefore, modern audio equipment generally splits a signal into two, or more often three, frequency ranges and sends them to



0	-32768	8	-23681	16	12111	24	31919	32	28095
1	-32768	9	-18449	17	17311	25	32767	33	28399
2	-32768	10	-11025	18	21311	26	32767	34	28751
3	-30753	11	-6913	19	23055	27	32767	35	28751
4	-28865	12	-4337	20	23519	28	32767	36	26911
5	-29105	13	-1329	21	25247	29	32031	37	24063
6	-29201	14	1743	22	27535	30	29759	38	21247
7	-26513	15	6223	23	29471	31	28399	39	18415

Figure 7.1. A stereo audio signal digitized.

different speakers. Speakers designed for low frequencies are called *woofers*, those for a middle range of frequencies are called *midranges*, and those for high frequencies are called *tweeters*.

Traditionally, the three speakers—woofer, midrange, and tweeter—were housed in the same physical box and the amplified signal was split into three parts using analog filtering components built into the speaker. However, there are limitations to this design. First of all, it is easy to determine the direction of a high frequency signal but not a low frequency one. Hence, the placement of the tweeters is important but the woofer can be put anywhere within hearing range. Furthermore, for stereo systems it is not even necessary to have two woofers—the low frequency signal can be combined into one. Also, woofers are physically large whereas tweeters can be made very small. Hence in a home theater system, one puts small tweeters in appropriate locations on either side of a video screen but puts a single woofer somewhere out of the way, such as under a coffee table.

Once the notion of physically separating the components of a speaker system is introduced, it is natural to consider also using separate amplifiers for each component. The amplifiers can then be designed to work optimally in a narrower range of frequencies. For example, it takes a lot of power to amplify a low frequency signal and much less power to do the same for high frequencies.

Hence, using a hefty amplifier on the high frequency component of a signal is wasteful.

Finally, given that the signal is to be split before being amplified it is now possible to consider splitting it even before converting it from digital to analog. At the digital level one has much more control over how the split is accomplished than can be achieved with an analog signal. The most common type of digital filter is a finite impulse response filter, which we describe next.

Finite Impulse Response (FIR) Filters

A *finite impulse response (FIR) filter* is given by a finite sequence of real (or complex) numbers $h_{-n}, \ldots, h_{-1}, h_0, h_1, \ldots, h_n$. This sequence transforms an input signal, x_k, $k \in \mathbb{Z}$, into an output signal, y_k, $k \in \mathbb{Z}$, according to the following convolution formula:

$$y_k = \sum_{i=-n}^{n} h_i x_{k-i}, \qquad k \in \mathbb{Z}.$$

Since the sequence of filter coefficients is finite, the sum is finite too. Typically n is a small number (less than 100) and so the output signal at any given point in time depends on the values of the input signal in a very narrow temporal range symmetric around this time. With 44,100 samples per second, $n = 100$ corresponds to a time interval that is only a small fraction of a second long. To implement the filter there must be at least this much delay between the input and output signals. Since this delay is small, it is generally unnoticeable.

Of course the filter coefficients h_i must be determined when the system is designed, which is long before any specific input signal is decided upon. Hence, one treats the input signal as a random process which will only be realized in the future but whose statistical properties can be used to design the filter. To this end, we assume that x_k is a stationary second-order random process. This means that each x_k is random, has mean zero, finite variance, and a covariance structure that is temporally homogeneous. This last property means that the following covariances depend on the difference between the sample times but not on the time itself:

$$s_k = \mathbf{E} x_i \bar{x}_{i+k}.$$

(The bar on the x_{i+k} denotes complex conjugate—most of our processes are real-valued in which case conjugation will plays no role.) The sequence s_k characterizes the input signal and its Fourier transform

$$S(\nu) = \sum_k s_k e^{2\pi j k \nu}$$

($j = \sqrt{-1}$) characterizes it in the frequency domain. The function $S()$ is called the *spectral density*. It is periodic in ν with period 1 and so its domain is usually

taken to be $[-1/2, 1/2)$. Values $\nu \in [-1/2, 1/2)$ are called *frequencies*. They can be converted to the usual scale of cycles-per-second (Hz) using the sample rate but for our purposes we will take them as numbers in $[-1/2, 1/2)$.

An Example. Consider the simplest input process—a complex signal which is a pure wave with known frequency ν_0 and an unknown phase shift:

$$x_k = e^{2\pi j(k+\theta)\nu_0}.$$

Here, θ is a random variable uniformly distributed on $[-1/2, 1/2)$. For this process, the autocorrelation function is easy to compute:

$$s_k = \mathbf{E} x_i \bar{x}_{i+k} = \mathbf{E} e^{2\pi j(i+\theta)\nu_0} e^{-2\pi j(i+k+\theta)\nu_0} = e^{-2\pi j k \nu_0}.$$

The spectral density is given by

$$S(\nu) = \begin{cases} \infty & \nu = \nu_0 \\ 0 & \text{else.} \end{cases}$$

The Transfer Function. We are interested in the spectral properties of the output process. Hence, we introduce the autocorrelation function for y_k

$$r_k = \mathbf{E} y_i \bar{y}_{i+k}$$

and its associated spectral density function

$$R(\nu) = \sum_k r_k e^{2\pi j k \nu}.$$

Substituting the definition of the output process y_i into the formula for the autocorrelation function, it is easy to check that

$$r_k = \sum_l g_l s_{k-l},$$

where

$$g_k = \sum_i h_i h_{i+k}.$$

Similarly, it is easy to relate the output spectral density $R(\nu)$ to the input spectral density $S(\nu)$:

$$R(\nu) = G(\nu) S(\nu), \tag{7.1}$$

where

$$G(\nu) = \sum_k g_k e^{2\pi j k \nu}.$$

Equation (7.1) is called the *transfer equation* and $G()$ is the *transfer function*.

Linear Phase Filters. For simplicity, we assume that the filter coefficients are real and symmetric about zero: $h_{-i} = h_i$. Such a filter is said to be *linear phase*. From these properties it follows that the function $H()$ defined by

$$H(\nu) = \sum_{k=-n}^{n} h_k e^{2\pi j k \nu}$$

is real-valued and symmetric about zero: $H(-\nu) = H(\nu)$. Indeed,

$$H(\nu) = h_0 + 2\sum_{k=1}^{n} h_k \cos(2\pi k \nu).$$

We then see that

$$\begin{aligned} G(\nu) &= \sum_k g_k e^{2\pi j k \nu} = \sum_{i,k} h_i h_{i+k} e^{2\pi j k \nu} \\ &= \sum_{i,k} h_{-i} e^{-2\pi j i \nu} h_{i+k} e^{2\pi j (i+k) \nu} = H(\nu)^2 \end{aligned}$$

and the transfer equation can be written in terms of $H()$:

$$R(\nu) = H(\nu)^2 S(\nu).$$

Power. For stationary signals, the power is defined as the expected value of the square of the signal at any moment in time. So, the input power is

$$P_{\text{in}} = \mathbf{E}|u_0|^2 = s_0 = \int_{-1/2}^{1/2} S(\nu) d\nu$$

and the output power is

$$P_{\text{out}} = \mathbf{E}|y_0|^2 = r_0 = \int_{-1/2}^{1/2} R(\nu) d\nu = \int_{-1/2}^{1/2} H(\nu)^2 S(\nu) d\nu.$$

A signal that is uniformly distributed over low frequencies, say from $-a$ to a has a spectral density given by

$$S(\nu) = 1_{[-a,a]}(\nu).$$

For such a signal, the input and output powers are given by

$$\begin{aligned} P_{\text{in}} &= 2a \\ P_{\text{out}} &= \int_{-a}^{a} H(\nu)^2 d\nu \qquad (7.2) \\ &= \sum_{k,k'} h_k h_{k'} \int_{-a}^{a} e^{2\pi j (k-k') \nu} d\nu \\ &= \sum_{k,k'} 2a h_k h_{k'} \text{sinc}(2\pi (k-k') a), \qquad (7.3) \end{aligned}$$

where

$$\text{sinc}(x) = \begin{cases} \frac{\sin x}{x} & x \neq 0 \\ 1 & \text{else.} \end{cases}$$

Passbands. In some cases, it is desirable to have the output be as similar as possible to the input. That is, we wish the difference process,

$$z_k = y_k - x_k,$$

to have as little energy as possible. Let q_k denote the autocorrelation function of the difference process:

$$q_k = \mathbf{E} z_i \bar{z}_{i+k}$$

and let $Q()$ denote the corresponding spectral density function. It is then easy to check from the definitions that

$$Q(\nu) = (H(\nu) - 1)^2 S(\nu).$$

The output power for the difference process is then given by

$$P_{\text{diff\_out}} = \int_{-1/2}^{1/2} (H(\nu) - 1)^2 S(\nu) d\nu.$$

As before, if the input spectral density $S()$ is a piecewise constant even function, then this output power can be expressed in terms of the sinc function.

Coordinated Woofer–Midrange–Tweeter Filtering

Having covered the basics of FIR filters, we return now to the problem of designing an audio system based on three filters: woofer, midrange, and tweeter. There are four power measurements that we want to be small: for each filter we want the output to be small if the input is uniformly distributed over a range of frequencies *outside* of the desired frequency range and finally when added together the difference between the summed signal and the original signal should be small over the entire input spectrum. Let

$$\begin{aligned} \mathcal{T} &= (-1/2, -b_t) \cup (b_t, 1/2), \\ \mathcal{M} &= (-b_m, -a_m) \cup (a_m, b_m), \\ \mathcal{W} &= (-a_w, a_w) \end{aligned}$$

denote the design frequency ranges for the tweeter, midrange, and woofer, respectively. Of course, we assume that the three ranges cover the entire available spectrum:

$$\mathcal{T} \cup \mathcal{M} \cup \mathcal{W} = (-1/2, 1/2)$$

(or, in other words, that $a_m < a_w$ and $b_t < b_m$). Each speaker has its own filter which is defined by its filter coefficients

$$h_k^{(j)}, \quad k = -n, -n+1, \ldots, n-1, n, \quad j \in \{t, m, w\}$$

and associated spectral density function $H_j(\nu), j \in \{t, m, w\}$. The three constraints which say that for each filter the output power per unit of input power is smaller than some threshold ρ can now be written as

$$\begin{aligned} \frac{1}{|\mathcal{T}^c|} \int_{\mathcal{T}^c} H_t^2(\nu) d\nu &\leq \rho, \\ \frac{1}{|\mathcal{M}^c|} \int_{\mathcal{M}^c} H_m^2(\nu) d\nu &\leq \rho, \\ \frac{1}{|\mathcal{W}^c|} \int_{\mathcal{W}^c} H_w^2(\nu) d\nu &\leq \rho. \end{aligned}$$

It is interesting to note that according to (7.3) the above integrals can all be efficiently expressed in terms of sums of products of pairs of filter coefficients in which the constants involve sinc functions. Such expressions are nonlinear. The fact that these functions are convex is only revealed by noting their equality with the expression in (7.2). Finally, the constraint that the reconstructed sum of the three signals deviates as little as possible from the a uniform response can be written as

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} (H_t(\nu) + H_m(\nu) + H_w(\nu) - 1)^2 d\nu \leq \epsilon.$$

At this juncture, there are several ways to formulate an optimization problem. We could fix ϵ to some small positive value and then minimize ρ, or we could fix ρ to some small positive value and minimize ϵ, or we could specify to proportional relation, such as equality, between ρ and ϵ and minimize both simultaneously. To be specific, for this tutorial, we choose the third approach.

In this paper (and in life), we formulate our optimization problems in AMPL, which is a small programming language designed for the efficient expression of optimization problems FGK93. The AMPL model for this problem is shown in Figure 7.2. The three filters and their spectral response curves are shown in Figure 7.3.

For more information on FIR filter design, see for example WBV97; CS99; LVBL98; Col98.

3. Shape Optimization (Telescope Design)

Until recently the search for extraterrestrial life has been the subject of science fiction stories—science itself was incapable of providing much help. Of course,

```
function sinc;

param n := 23;
param pi := 4*atan(1);

param aw := 0.05;
param am := 0.04;
param bm := 0.25;
param bt := 0.2;

var rho >= 0;
var hw {0..n};
var hm {0..n};
var ht {0..n};

minimize power_bnd: rho;

subject to passband:
    ((hw[0]+hm[0]+ht[0]-1)^2 + 2*sum {k in 1..n} (hw[k]+hm[k]+ht[k])^2)
    <= rho;

subject to wooferband:
    sum {k in -n..n} hw[abs(k)]^2
    -
    sum {k in -n..n, kk in -n..n} 2*aw*hw[abs(k)]*hw[abs(kk)] * sinc(2*pi*(k-kk)*aw)
    <= (1-2*aw)*rho;

subject to midrangeband:
    sum {k in -n..n} hm[abs(k)]^2
    -
    sum {k in -n..n, kk in -n..n} 2*bm*hm[abs(k)]*hm[abs(kk)] * sinc(2*pi*(k-kk)*bm)
    +
    sum {k in -n..n, kk in -n..n}
   2*am*hm[abs(k)]*hm[abs(kk)] * sinc(2*pi*(k-kk)*am)
    <= (1-2*(bm-am))*rho;

subject to tweeterband:
    sum {k in -n..n, kk in -n..n} 2*bt*ht[abs(k)]*ht[abs(kk)] * sinc(2*pi*(k-kk)*bt)
    <= 2*bt*rho;

solve;

printf {k in 0..n}: "%10.6f \n", hw[k] > hw;
printf {k in 0..n}: "%10.6f \n", hm[k] > hm;
printf {k in 0..n}: "%10.6f \n", ht[k] > ht;

printf {nu in 0..0.5 by 1/1000}: "%7.4f %10.3e \n",
    nu, 10*log10((hw[0] + 2* sum {k in 1..n} (hw[k]*cos(-2*pi*k*nu)))^2) > w.out;

printf {nu in 0..0.5 by 1/1000}: "%7.4f %10.3e \n",
    nu, 10*log10((hm[0] + 2* sum {k in 1..n} (hm[k]*cos(-2*pi*k*nu)))^2) > m.out;

printf {nu in 0..0.5 by 1/1000}: "%7.4f %10.3e \n",
    nu, 10*log10((ht[0] + 2* sum {k in 1..n} (ht[k]*cos(-2*pi*k*nu)))^2) > t.out;
```

Figure 7.2. A sample AMPL program for FIR filter design of coordinated woofer, midrange, and tweeter system.

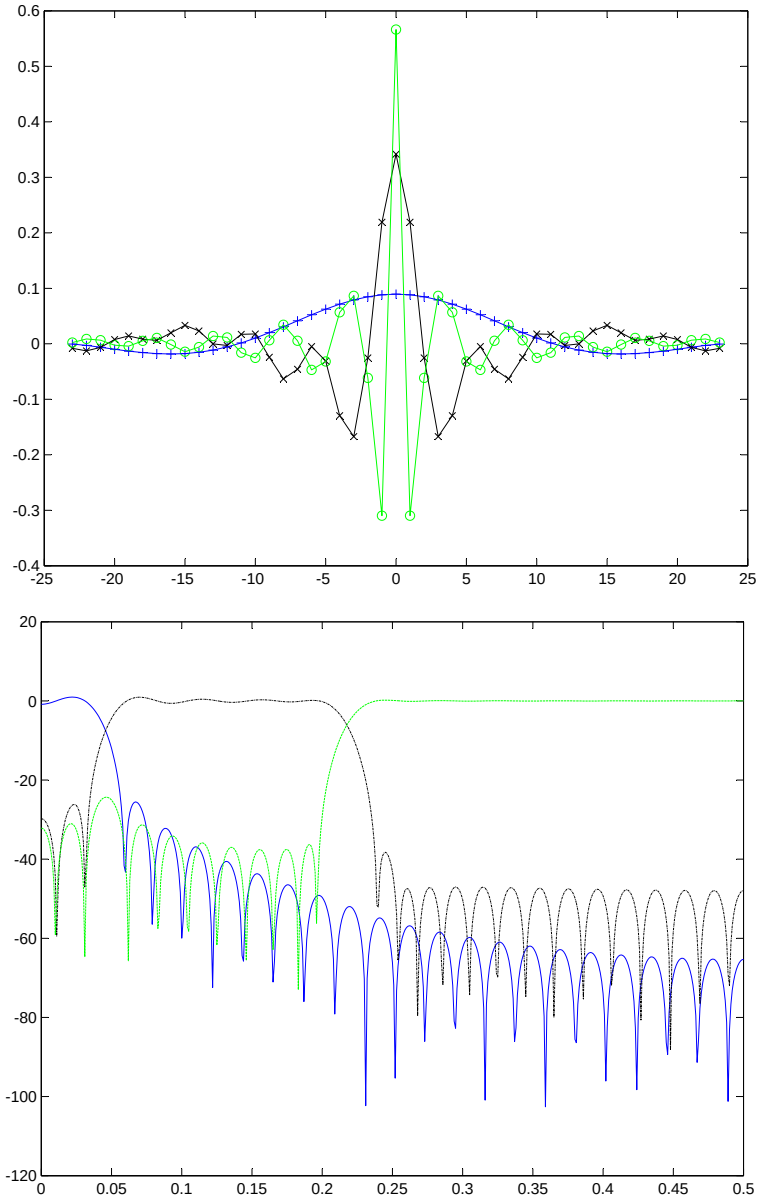


Figure 7.3. *Top.* The optimal filter coefficients. *Bottom.* The corresponding spectral response curves. In the top graph, the $\circ$'s correspond to the tweeter filter, the $\times$'s correspond to the midrange filter, and the $+$'s correspond to the woofer filter. The spectral response curve is a plot of ten times log base ten of power as a function of frequency.

there is so far one exception—the SETI project (SETI stands for *search for extraterrestrial intelligence*), which has been operating for several years now. The idea here is to use radio telescopes to listen for radio transmissions from advanced civilizations. This project was started with the support of Carl Sagan and his book *Contact* was made into a Hollywood movie starring Jodie Foster. But, the universe is big and the odds that there is an advanced civilization in our neck of the woods is small so this project seems like a long shot. And, every year that goes by without hearing anything proves more and more what a long shot it is. Even if advanced civilizations are rare, there is every expectation that most stars have planets around them and even Earth-like planets are probably fairly common. It would be interesting if we could search for, catalog, and survey such planets. In fact, astrophysicists, with the support of NASA and JPL, are now embarking on this goal—imaging Earth-like planets around nearby Sun-like stars. We have already detected indirectly more than 100 Jupiter-sized planets around other stars and we will soon be able to take pictures of some of these planets. Once we can take pictures, we can start answering questions like: is there water? is there chlorophyl? is there carbon dioxide in the atmosphere? etc. But Jupiter-sized planets are not very Earth-like. They are mostly gas and very massive. There's not much place for an ET to get a foothold and if one could the gravity would be crushing. A more interesting but much more difficult problem is to survey Earth-like planets. NASA has made such a search and survey one of its key science projects for the coming decades. The idea is to build a large space telescope, called the *Terrestrial Planet Finder (TPF)* that is capable of imaging these planets. But just making it large and putting it into space is not enough. The planet, which will appear very close to its star, will still be invisible in the glare of its much brighter star.

To put the problem into perspective, here are a few numbers. Consider a star that is say 30 light years from our solar system. A planet that is as far from this star as we are from our Sun will appear to us here on Earth at an angular separation of about 0.1 arcseconds from its star. And the star will be 10^{10} times brighter than the planet. There will be an enormous amount of starlight that will "spill" onto that part of the image where the planet is supposed to be. This spillage is not one of engineering/construction imprecisions. Rather, it is a consequence of fundamental physics. Light is a wave, an electro-magnetic wave, and because of this it is impossible to focus the light from a point source (such as a star) to a perfect point in the image. Instead, for a telescope with a circular opening letting in the light, you get a small blob of light called an *Airy disk* surrounded by *diffraction rings*—see Figure 7.4. The planet is about as bright as the light in the 100th diffraction ring. Earlier rings are much brighter. The spacing of the rings is inversely proportional to the size of the telescope. To make it so that the rings are tight enough that we can image a planet like the one just described would require a telescope with a mirror having a 250

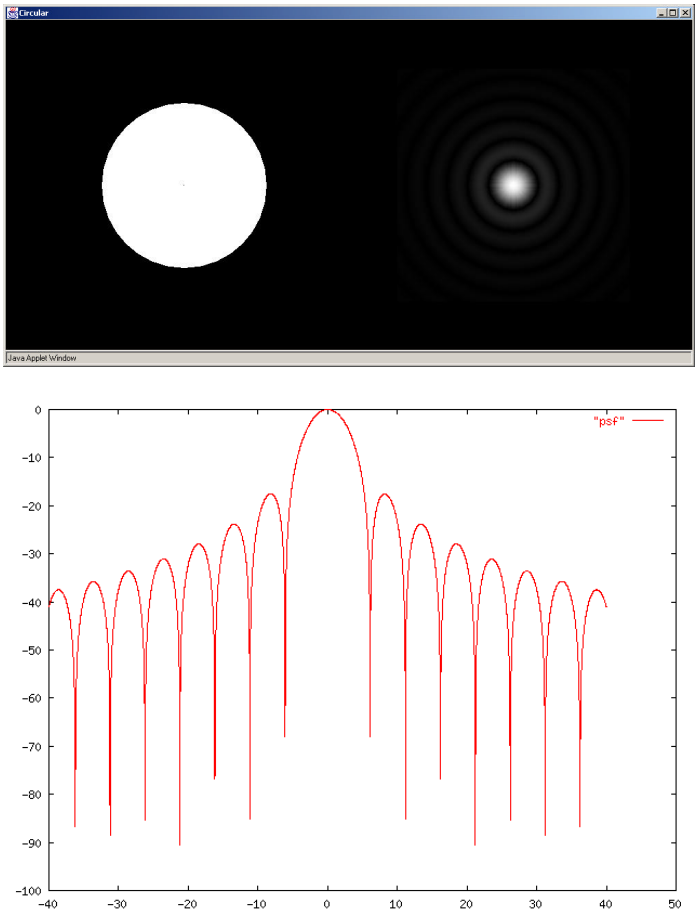


Figure 7.4. The top left shows a circular opening at the front of a telescope. The top right shows the corresponding Airy disk and a few diffraction rings. The plot on the bottom shows the cross-sectional intensity on a $\log$ scale. The desired level of 10^{-10} corresponds to an intensity level of -100 on this log plot. It is way off to the side. It occurs out somewhere around the 100th diffraction ring.

meter diameter. That is more than 10 times the diameter of the mirror in the Hubble space telescope. There are not any rockets in existence, or on the drawing boards, that will be capable of lifting such a large monolithic object into space at any time in the foreseeable future. For this reason some clever ideas are required. A few have been proposed. Perhaps the most promising one exploits the idea that the ring pattern is a consequence of the circular shape of the telescope. Different shapes provide different patterns—perhaps some of them provide a very dark zone very close to the Airy disk. An even broader generalization is to consider a telescope that has a filter over its opening that has light transmission properties that vary over the surface of the filter. If the transmission is everywhere either zero or one then the filter acts to create a different shaped opening. Such filters are called *apodizations*. The problem is to find an apodization that provides a very dark area very close to the Airy disk.

Okay, enough with the words already—we need to give a mathematical formulation of the problem. The diffraction pattern produced by the star in the image is the square of the electric field at the image plane and the electric field at the image plane turns out to be just the Fourier transform of the apodization function A defining the transmissivity of the apodized pupil:

$$E(\xi,\zeta) = \iint_S e^{-2\pi i(x\xi+y\zeta)} A\,(x,y)\,dxdy,$$

where

$$S = \{(x,y):\ 0 \le r(x,y) \le 1/2,\ \theta(x,y) \in [0,2\pi]\}\,,$$

and $r(x,y)$ and $\theta(x,y)$ denote the polar coordinates associated with point (x,y). Here, and throughout this section, x and y denote coordinates on the filter measured in units of the mirror diameter D and ξ and ζ denote angular (radian) deviation from on-axis measured in units of wavelength λ over mirror-diameter (λ/D) or, equivalently, physical distance in the image plane measured in units of focal-length times wavelength over mirror-diameter ($f\lambda/D$).

For circularly-symmetric apodizations, it is convenient to work in polar coordinates. To this end, let r and θ denote polar coordinates in the filter plane and let ρ and ϕ denote the image plane coordinates:

$$\begin{array}{rclcrcl} x &=& r\cos\theta & & \xi &=& \rho\cos\phi \\ y &=& r\sin\theta & & \zeta &=& \rho\sin\phi. \end{array}$$

Hence,

$$\begin{aligned} x\xi + y\zeta &= r\rho(\cos\theta\cos\phi + \sin\theta\sin\phi) \\ &= r\rho\cos(\theta-\phi). \end{aligned}$$

The electric field in polar coordinates depends only on ρ and is given by

$$E(\rho) = \int_0^{1/2} \int_0^{2\pi} e^{-2\pi i r \rho \cos(\theta - \phi)} A(r) r d\theta dr, \tag{7.4}$$

$$= 2\pi \int_0^{1/2} J_0(2\pi r \rho) A(r) r dr, \tag{7.5}$$

where J_0 denotes the 0-th order Bessel function of the first kind. Note that the mapping from apodization function A to electric field E is linear. Furthermore, the electric field in the image plane is real-valued (because of symmetry) and its value at $\rho = 0$ is the *throughput* of the apodization:

$$E(0) = 2\pi \int_0^{1/2} A(r) r dr.$$

As mentioned already, the diffraction pattern, which is called the *point spread function* (psf), is the square of the electric field. The contrast requirement is that the psf in the dark region be 10^{-10} of what it is at the center of the Airy disk. Because the electric field is real-valued, it is convenient to express the contrast requirement in terms of it rather than the psf, resulting in a field requirement of $\pm 10^{-5}$.

The apodization that maximizes throughput subject to contrast constraints can be formulated as an infinite dimensional linear programming problem:

$$\begin{array}{lll} \text{maximize} & E(0) & \\ \text{subject to} & -10^{-5} E(0) \le E(\rho) \le 10^{-5} E(0), & \rho_{\text{iwa}} \le \rho \le \rho_{\text{owa}}, \\ & 0 \le A(r) \le 1, & 0 \le r \le 1/2, \end{array}$$

where ρ_{iwa} denotes a fixed *inner working angle* and ρ_{owa} a fixed *outer working angle*. Discretizing the sets of r's and ρ's and replacing the integrals with their Riemann sums, the problem is approximated by a finite dimensional linear programming problem that can be solved to a high level of precision.

The solution obtained for $\rho_{\text{iwa}} = 4$ and $\rho_{\text{owa}} = 40$ is shown in Figure 7.5. Note that the solution is of a bang-bang type. That is, the apodization function is mostly 0 or 1 valued. This suggests looking for a mask that is about as good as this apodization. Such a mask can be found by solving the following nonlinear optimization problem. A mask consists of a set of concentric opaque rings, formulated in terms of the inner and outer radii of the openings between the

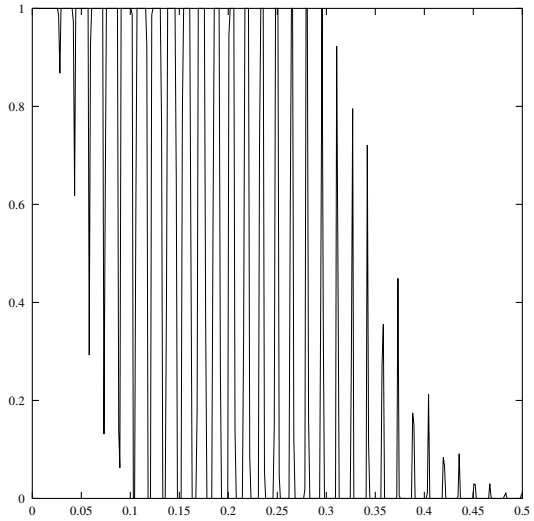


Figure 7.5. The optimal apodization function turns out to be of bang-bang type.

rings:

$$\begin{array}{rl} [r_0, r_1] & \text{first opening} \\ [r_2, r_3] & \text{second opening} \\ [r_4, r_5] & \text{third opening} \\ \vdots & \\ [r_{2m-2}, r_{2m-1}] & m\text{-th opening} \end{array}$$

With this notation, the formula for $E(\rho)$ given in (7.5) can be rewritten as a sum of integrals over these openings:

$$\begin{aligned} E(\rho) &= 2\pi \sum_{k=0}^{m-1} \int_{r_{2k}}^{r_{2k+1}} J_0(2\pi r\rho) r dr, \\ &= \frac{1}{\rho} \sum_{k=0}^{m-1} \left(r_{2k+1} J_1(2\pi r_{2k+1}\rho) - r_{2k} J_1(2\pi r_{2k}\rho) \right) \end{aligned}$$

Treating the r_k's as variables and using this new expression for the electric field, the mask design problem becomes:

$$\begin{array}{ll} \text{maximize} & \pi \displaystyle\sum_{k=0}^{m-1} \left(r_{2k+1}^2 - r_{2k}^2\right) \\ \text{subject to} & -10^{-5}E(0) \le E(\rho) \le 10^{-5}E(0), \qquad \rho_{\text{iwa}} \le \rho \le \rho_{\text{owa}}, \\ & 0 \le r_0 \le r_1 \le \cdots \le r_{2m-1} \le 1/2. \end{array}$$

This problem is a nonconvex nonlinear optimization problem and hence the best hope for solving it in a reasonable amount of cpu time is to use a "local-search" method starting the search from a solution that is already close to optimal. The bang-bang solution from the linear programming problem can be used to generate a starting solution. Indeed, the discrete solution to the linear programming problem can be used to find the inflection points of A which can be used as initial guesses for the r_k's. LOQO was used to perform this local optimization. Figure 7.6 shows an optimal concentric-ring mask computed using an inner working angle of 4 and an outer working angle of 60. Using this mask over a 10 meter primary mirror makes it possible to image the Earth-like planet 30 light-years away from us. Even a telescope with a 10 meter primary mirror is larger than anything we have launched into space to date but it is a size that fits into the realm of possibility. And, if a 10 circular mirror is too large, we could fall back on elliptical designs say using a 4×10 mirror. Such a mirror could be put into space using currently available Delta rockets. The mask designs presented here and many others can be found in the following references: ref:Spergel; ref:kasdin; KVSL02; VSK02; VSK03.

4. Minimum weight truss design

As we saw in the previous section, designing a space telescope with an apodization, or mask, over the mirror makes it possible to image Earth-like planets around nearby stars. However, it is still just on the edge of tractability. The desire remains to launch a much larger telescope. But, given constraints on mirror manufacture and launch capabilities, the prospect of using a huge telescope remains well out of reach for the foreseeable future. A compromise idea it to launch several, say four, individual telescopes, attach them to a common structure so that they are configured along a straight line, and then combine the light from the four telescopes to make one image. In some sense, this is equivalent to making a huge circular-mirror telescope and masking out everything except for four circular areas spread out along the mirror. As explained in the previous section, this masking changes the diffraction pattern but a basic principle holds which says that the further apart you can get the telescopes, the tighter the diffraction pattern will be. One of the main challenges with this design concept is to design and build a truss-like structure that is light enough to

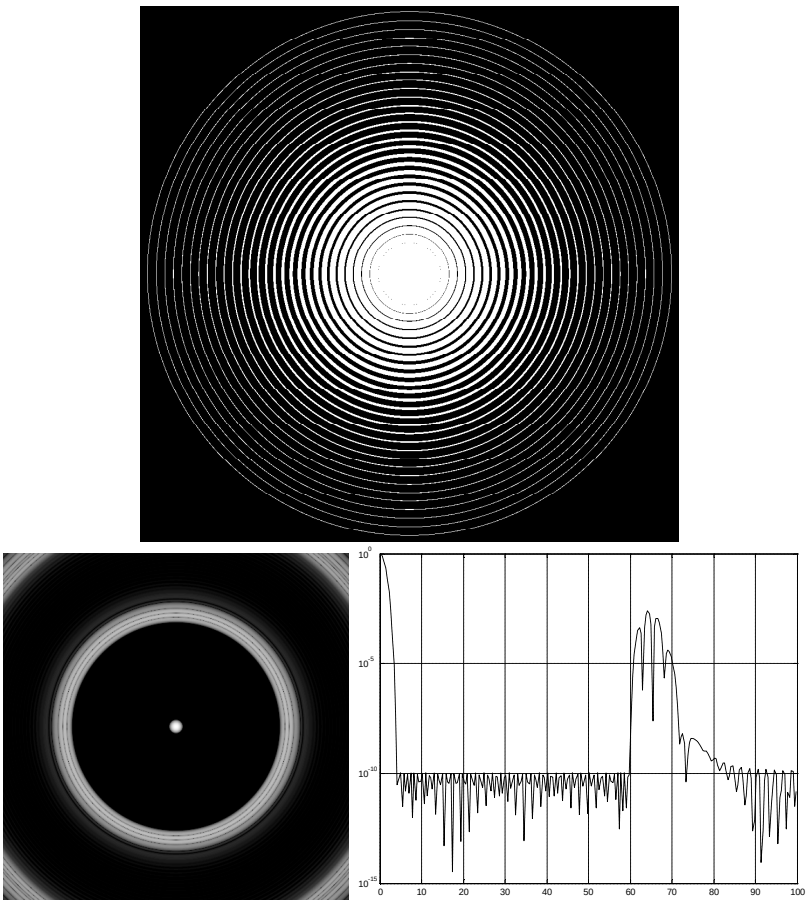


Figure 7.6. At the top is shown the concentric ring apodization. The second row shows the psf as a 2-D image and in cross-section. Note that it achieves the required 10^{-10} level of darkness at a distance of only 4 from the center of the Airy disk.

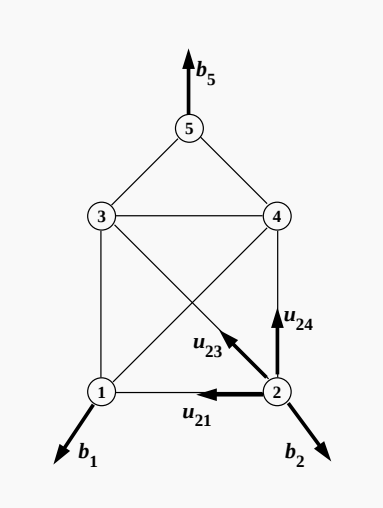


Figure 7.7. A design space showing 5 nodes and 8 arcs. Also shown are three externally applied forces and the tensile forces they induce in node 2.

be launchable into space yet stiff enough that it can hold four massive telescopes in position with a precision that is small relative to the wavelength of light.

Such truss optimization problems have a long history starting, I believe, with Jim Ho's Ph.D. thesis at Stanford which was published in Ho75. In more recent times, Ronnie Ben-Tal has collaborated with Bendsøe and Zowe on just this family of problems. They wrote many papers including the seminal paper BBZ94. Anyway, in the following subsection, I will outline the basic optimization problem and its reduction to a linear programming problem. Then in the following subsection, I will describe how it is being applied to the truss-design problem mentioned above.

Mathematical Formulation

We assume we are given a design space, which consists of a set $\mathcal{N}$ of nodes (aka joints) at fixed locations and a set $\mathcal{A}$ of undirected arcs (aka members) connecting various pairs of nodes—see Figure 7.7. The unknowns in the problem are the tensions x_{ij} in each member. We assume that there are many more members than are needed to make a rigid truss, so it follows that the system is underdetermined and there is lots of freedom as to how the forces "flow" through the structure. The tensions x_{ij} are allowed to go negative. Negative tensions are simply compressions. The constraints in our optimization problem are that force be balanced at each node. For example, if we look at node 2, the

force balance equations are:

$$x_{12}\begin{bmatrix} -1 \\ 0 \end{bmatrix} + x_{23}\begin{bmatrix} -0.6 \\ 0.8 \end{bmatrix} + x_{24}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = -\begin{bmatrix} b_2^1 \\ b_2^2 \end{bmatrix}.$$

To write down the equations in more generality, we need to introduce some notation:

$$\begin{aligned} p_i &= \text{position vector for joint } i \\ u_{ij} &= \frac{p_j - p_i}{\|p_j - p_i\|} \end{aligned}$$

Note that $u_{ji} = -u_{ij}$. With these notations, the general force balance constraints can be written as

$$\sum_{\substack{j: \\ \{i,j\}\in\mathcal{A}}} u_{ij}x_{ij} = -b_i, \quad i = 1,\ldots,m$$

It is instructive to write these equations in matrix form as $Ax = -b$, where

$$x^T = \begin{bmatrix} x_{12} & x_{13} & x_{14} & x_{23} & x_{24} & x_{34} & x_{35} & x_{45} \end{bmatrix}$$

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{bmatrix} \begin{bmatrix}1\\0\end{bmatrix} & \begin{bmatrix}0\\1\end{bmatrix} & \begin{bmatrix}.6\\.8\end{bmatrix} & & & & & \\ \begin{bmatrix}-1\\0\end{bmatrix} & & & \begin{bmatrix}-.6\\.8\end{bmatrix} & \begin{bmatrix}0\\1\end{bmatrix} & & & \\ & \begin{bmatrix}0\\-1\end{bmatrix} & & \begin{bmatrix}.6\\-.8\end{bmatrix} & & \begin{bmatrix}1\\0\end{bmatrix} & \begin{bmatrix}.6\\.8\end{bmatrix} & \\ & & \begin{bmatrix}-.6\\-.8\end{bmatrix} & & \begin{bmatrix}0\\-1\end{bmatrix} & \begin{bmatrix}-1\\0\end{bmatrix} & & \begin{bmatrix}-.6\\.8\end{bmatrix} \\ & & & & & & \begin{bmatrix}-.6\\-.8\end{bmatrix} & \begin{bmatrix}.6\\-.8\end{bmatrix} \end{bmatrix}, \quad b = \begin{bmatrix} b_1^1 \\ b_1^2 \\ b_2^1 \\ b_2^2 \\ b_3^1 \\ b_3^2 \\ b_4^1 \\ b_4^2 \\ b_5^1 \\ b_5^2 \end{bmatrix}.$$

Note that $\|u_{ij}\| = \|u_{ji}\| = 1$ and $u_{ij} = -u_{ji}$. Also, each column contains a u_{ij}, a u_{ji}, and the rest are zero. If the problem were one dimensional, this would be exactly a node-arc incidence matrix. In fact, much of the theory that has been developed for minimum-cost network flow problems has an immediate analogue in these truss design problems. These connections are described at length in Van01.

So far we have only written down the force balance constraints. The optimization problem is to minimize the weight of the final structure. We assume that weight is related to tension/compression in the members by assuming that the cross-sectional area of a member must be proportional to the tension/compression that must be carried by that member (the constants of proportionality can be different for tension and for compression but in what follows

we assume that they are equal). Hence, the minimum weight structural design problem can be formulated like this:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{\{i,j\}\in\mathcal{A}} l_{ij}|x_{ij}| \\ \text{subject to} & \displaystyle\sum_{\substack{j:\\ \{i,j\}\in\mathcal{A}}} u_{ij}x_{ij} = -b_i \qquad i = 1,2,\ldots,m. \end{array}$$

This is not quite a linear programming problem. But it is easy to convert it into one using a common trick of splitting every variable into the difference between its positive and negative parts:

$$\begin{array}{rcl} x_{ij} & = & x_{ij}^+ - x_{ij}^-, \qquad x_{ij}^+, x_{ij}^- \geq 0, \qquad x_{ij}^+ x_{ij}^- = 0 \\ |x_{ij}| & = & x_{ij}^+ + x_{ij}^- \end{array}$$

In terms of these new variables, the problem can be written as follows:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{\{i,j\}\in\mathcal{A}} (l_{ij}x_{ij}^+ + l_{ij}x_{ij}^-) \\ \text{subject to} & \displaystyle\sum_{\substack{j:\\ \{i,j\}\in\mathcal{A}}} (u_{ij}x_{ij}^+ - u_{ij}x_{ij}^-) = -b_i \qquad i = 1,2,\ldots,m \\ & \qquad\qquad x_{ij}^+ x_{ij}^- = 0 \qquad \{i,j\}\in\mathcal{A}, \\ & \qquad\qquad x_{ij}^+,\ x_{ij}^- \geq 0 \qquad \{i,j\}\in\mathcal{A}. \end{array}$$

It is easy to argue that one can drop the complementarity type constraints, $x_{ij}^+ x_{ij}^- = 0, \{i,j\} \in \mathcal{A}$, since these constraints will automatically be satisfied at optimality. With these constraints gone, the problem is a linear programming problem that can be solved very efficiently.

It was shown in BBZ94 that this minimum weight structural design problem is dual to a maximum stiffness structural design problem and therefore that the structure found according to this linear programming methodology is in fact maximally stiff.

Telescope Truss Design

As mentioned at the beginning of this section, it is a very real problem of current interest to design a maximally stiff truss-like structure to support four telescopes. In order to apply the ideas described in the previous subsection, we need to add one extra twist to the model, which is that the design must be stiff relative to a collection of different loading scenarios. In particular, we assume that the structure must be stiff with respect to accelerations in each of the three main coordinate directions and also that it is stiff relative to torques about the

```
param m default 26;  param n default 39;

set X := {0..n}; set Y := {0..m};

set NODES := X cross Y;    # A lattice of Nodes

set ANCHORS within NODES
    := { x in X, y in Y : x == 0 && y >= floor(m/3) && y <= m-floor(m/3) };

param xload {(x,y) in NODES: (x,y) not in ANCHORS} default 0;
param yload {(x,y) in NODES: (x,y) not in ANCHORS} default 0;

param gcd {x in -n..n, y in -n..n} :=
    (if x <  0 then gcd[-x,y] else
    (if x == 0 then y else
    (if y < x  then gcd[y,x] else
    (gcd[y mod x, x])  )));

set ARCS := { (xi,yi) in NODES, (xj,yj) in NODES:
    abs(xj-xi) <= 3 && abs(yj-yi) <=3 &&
    abs(gcd[ xj-xi, yj-yi ]) == 1 &&
    ( xi > xj || (xi == xj && yi > yj) ) };

param length {(xi,yi,xj,yj) in ARCS} := sqrt( (xj-xi)^2 + (yj-yi)^2 );

var comp {ARCS} >= 0;
var tens {ARCS} >= 0;
minimize volume:
    sum {(xi,yi,xj,yj) in ARCS}
        length[xi,yi,xj,yj] * (comp[xi,yi,xj,yj] + tens[xi,yi,xj,yj]);

subject to Xbalance {(xi,yi) in NODES: (xi,yi) not in ANCHORS}:
    sum { (xi,yi,xj,yj) in ARCS }
        ((xj-xi)/length[xi,yi,xj,yj]) * (comp[xi,yi,xj,yj]-tens[xi,yi,xj,yj])
    +
    sum { (xk,yk,xi,yi) in ARCS }
        ((xi-xk)/length[xk,yk,xi,yi]) * (tens[xk,yk,xi,yi]-comp[xk,yk,xi,yi])
    = xload[xi,yi];

subject to Ybalance {(xi,yi) in NODES: (xi,yi) not in ANCHORS}:
    sum { (xi,yi,xj,yj) in ARCS }
        ((yj-yi)/length[xi,yi,xj,yj]) * (comp[xi,yi,xj,yj]-tens[xi,yi,xj,yj])
    +
    sum { (xk,yk,xi,yi) in ARCS }
        ((yi-yk)/length[xk,yk,xi,yi]) * (tens[xk,yk,xi,yi]-comp[xk,yk,xi,yi])
    = yload[xi,yi];

let yload[n,m/2] := -1;

solve;
```

Figure 7.8. A sample AMPL program for minimum weight truss-like structures designed to accommodate a single loading scenario.

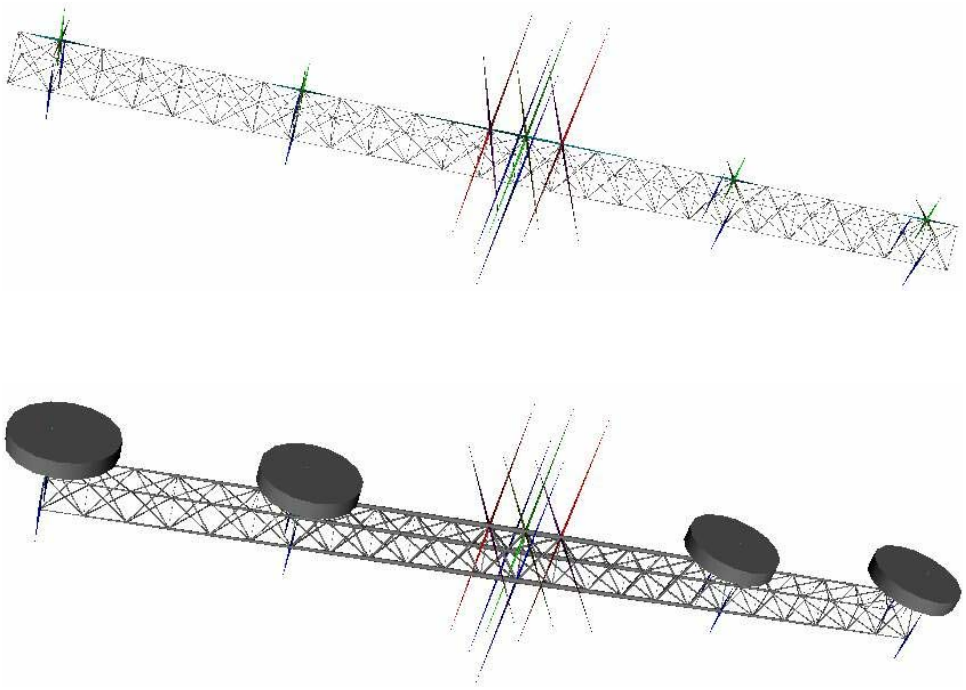


Figure 7.9. Top. The design space. *Bottom.* The optimal design.

three principle axes of rotation. Torques are modeled as pairs of forces that are equal and opposite but applied at points that are not colinear with the direction of the force. So, our basic model has six load scenarios. Forces must be balanced in each scenario. Of course, the tensions/compressions are scenario dependent but the beam cross-sections must be chosen independently of the scenario (since one physical structure must be stiff under each scenario).

The underlying design space consists of 26 trusslike cells, 1.6 m wide with triangular cross-section 2 m on a side. The middle two cells attach to the satellite. Forces are applied by thrusters on the satellite. Of course, in the real satellite, forces are applied to perform various motions but in our model we don't want, or need, to model moving objects. Instead we assume that countervaling forces are applied exactly where the massive objects are—i.e., at the point of attachment of the four telescopes. Figure 7.9 shows the underlying design space together with the optimal design. The force vectors corresponding to the six load scenarios are shown as elongated, shaded cones.

5. New orbits for the n-body problem

Since the time of Lagrange and Euler precious few solutions to the n-body problem have been discovered. Lagrange proved that two bodies being mutually attracted to the other by gravity will execute elliptical orbits where each of the two ellipses has a focus at the center of mass of the two-body system. This can be proved mathematically. Not only does this solution to Newton's equations of motion exist, but it is also stable. Euler pointed out that a third body can be placed stationarily at the center of mass and this makes a solution to the three-body problem. However, this 3-body system is unstable—if the third body is perturbed ever so slightly the whole system will fall apart. A few other simple solutions have been known for hundreds of years. For example, you can distribute n equal-mass bodies uniformly around a circle and start each one off with a velocity perpendicular to the line through the center and this system will behave much like the 2-body system. But, for three bodies or more, this system is again unstable. So, it was a tremendous shock a few years ago when Cris Moore at the Sante Fe Institute discovered a new, stable solution to the equal-mass 3-body problem. This discovery has created a tremendous level of interest in the celestial mechanics community. Not only was the solution he discovered both new and stable, it is also aesthetically beautiful because each of the three bodies follow the exact same path. At any given moment they are at different parts of this path. Such orbital systems have been called *choreographies*. Many celestial mechanics have been working hard to discover new choreographies. The interesting thing for us is that the main tool is to minimize the so-called action functional.

In this section, we will describe how it is that minimizing the action functional provides solutions to the n-body problem and we will illustrate several new solutions that we have found.

Least Action Principle

Given n bodies, let m_j denote the mass and $z_j(t)$ denote the position in $\mathbb{R}^2 = \mathbb{C}$ of body j at time t. The *action functional* is a mapping from the space of all trajectories, $z_1(t), z_2(t), \ldots, z_n(t)$, $0 \leq t \leq 2\pi$, into the reals. It is defined as the integral over one period of the kinetic minus the potential energy:

$$A = \int_0^{2\pi} \left(\sum_j \frac{m_j}{2} \|\dot{z}_j\|^2 + \sum_{j,k:k<j} \frac{m_j m_k}{\|z_j - z_k\|} \right) dt.$$

Stationary points of the action function are trajectories that satisfy the equations of motions, i.e., Newton's law gravity. To see this, we compute the first

variation of the action functional,

$$\begin{aligned}\delta A &= \int_0^{2\pi} \sum_\alpha \left(\sum_j m_j \dot{z}_j^\alpha \dot{\delta z_j}^\alpha - \sum_{j,k:k<j} m_j m_k \frac{(z_j^\alpha - z_k^\alpha)(\delta z_j^\alpha - \delta z_k^\alpha)}{\|z_j - z_k\|^3} \right) dt \\ &= -\int_0^{2\pi} \sum_j \sum_\alpha \left(m_j \ddot{z}_j^\alpha + \sum_{k:k\neq j} m_j m_k \frac{z_j^\alpha - z_k^\alpha}{\|z_j - z_k\|^3} \right) \delta z_j^\alpha dt,\end{aligned}$$

and set it to zero. We get that

$$m_j \ddot{z}_j^\alpha = -\sum_{k:k\neq j} m_j m_k \frac{z_j^\alpha - z_k^\alpha}{\|z_j - z_k\|^3}, \qquad j = 1, 2, \ldots, n, \quad \alpha = 1, 2 \quad (7.6)$$

Note that if $m_j = 0$ for some j, then the first order optimality condition reduces to $0 = 0$, which is *not* the equation of motion for a massless body. Hence, we must assume that all bodies have strictly positive mass.

Periodic Solutions

Our goal is to use numerical optimization to minimize the action functional and thereby find periodic solutions to the n-body problem. Since we are interested only in periodic solutions, we express all trajectories in terms of their Fourier series:

$$z_j(t) = \sum_{k=-\infty}^{\infty} \gamma_k e^{ikt}, \qquad \gamma_k \in \mathbb{C}.$$

Abandoning the efficiency of complex-variable notation, we can write the trajectories with components $z_j(t) = (x_j(t), y_j(t))$ and $\gamma_k = (\alpha_k, \beta_k)$. So doing, we get

$$\begin{aligned}x(t) &= a_0 + \sum_{k=1}^{\infty} (a_k^c \cos(kt) + a_k^s \sin(kt)) \\ y(t) &= b_0 + \sum_{k=1}^{\infty} (b_k^c \cos(kt) + b_k^s \sin(kt))\end{aligned}$$

where

$$\begin{aligned}a_0 = \alpha_0, \qquad a_k^c &= \alpha_k + \alpha_{-k}, \qquad a_k^s = \beta_{-k} - \beta_k, \\ b_0 = \beta_0, \qquad b_k^c &= \beta_k + \beta_{-k}, \qquad b_k^s = \alpha_k - \alpha_{-k}.\end{aligned}$$

Since we plan to optimize over the space of trajectories, the parameters a_0, a_k^c, a_k^s, b_0, b_k^c, and b_k^s are the decision variables in our optimization model. The objective is to minimize the action functional.

```
param N := 3;  # number of masses
param n := 15;  # number of terms in Fourier series representation
param m := 100;  # number of terms in numerical approx to integral

set Bodies := {0..N-1};
set Times  := {0..m-1} circular; # "circular" means that next(m-1) = 0

param theta {t in Times} := t*2*pi/m;
param dt := 2*pi/m;

param a0 {i in Bodies} default 0;        param b0 {i in Bodies} default 0;
var as {i in Bodies, k in 1..n} := 0;    var bs {i in Bodies, k in 1..n} := 0;
var ac {i in Bodies, k in 1..n} := 0;    var bc {i in Bodies, k in 1..n} := 0;

var x {i in Bodies, t in Times}
  = a0[i]+sum {k in 1..n} ( as[i,k]*sin(k*theta[t]) + ac[i,k]*cos(k*theta[t]) );
var y {i in Bodies, t in Times}
  = b0[i]+sum {k in 1..n} ( bs[i,k]*sin(k*theta[t]) + bc[i,k]*cos(k*theta[t]) );

var xdot {i in Bodies, t in Times} = (x[i,next(t)]-x[i,t])/dt;
var ydot {i in Bodies, t in Times} = (y[i,next(t)]-y[i,t])/dt;

var K {t in Times} = 0.5*sum {i in Bodies} (xdot[i,t]^2 + ydot[i,t]^2);

var P {t in Times}
  = - sum {i in Bodies, ii in Bodies: ii>i}
            1/sqrt((x[i,t]-x[ii,t])^2 + (y[i,t]-y[ii,t])^2);

minimize A: sum {t in Times} (K[t] - P[t])*dt;

let {i in Bodies, k in 1..n} as[i,k] := 1*(Uniform01()-0.5);
let {i in Bodies, k in 1..n} ac[i,k] := 1*(Uniform01()-0.5);
let {i in Bodies, k in n..n} bs[i,k] := 0.01*(Uniform01()-0.5);
let {i in Bodies, k in n..n} bc[i,k] := 0.01*(Uniform01()-0.5);

solve;
```

Figure 7.10. AMPL program for finding trajectories that minimize the action functional.

Figure 7.10 shows the AMPL program for minimizing the action functional.

Note that the action functional is a nonconvex nonlinear functional. Hence, it is expected to have many local extrema and saddle points. We use the author's local optimization software called LOQO (see SOR9708, Van97d) to find local minima in a neighborhood of an arbitrary given starting trajectory. One can provide either specific initial trajectories or one can give random initial trajectories. The four lines just before the call to `solve` in Figure 7.10 show how to specify a random initial trajectory. Of course, AMPL provides capabilities of printing answers in any format either on the standard output device or to a file. For the sake of brevity and clarity, the print statements are not shown in Figure 7.10. AMPL also provides the capability to loop over sections of code. This is also not shown but the program we used has a loop around the four initialization statements, the call to solve the problem, and the associated print statements. In this way, the program can be run once to solve for a large number of periodic solutions.

Choreographies. Recently, CM00 introduced a new family of solutions to the n-body problem called choreographies. A *choreography* is defined as a solution to the n-body problem in which all of the bodies share a common orbit and are uniformly spread out around this orbit. Such trajectories are even easier to find using the action principle. Rather than having a Fourier series for each orbit, it is only necessary to have one master Fourier series and to write the action functional in terms of it. Figure 7.11 shows the AMPL model for finding choreographies.

Stable vs. Unstable Solutions

Figure 7.12 shows some simple choreographies found by minimizing the action functional using the AMPL model in Figure 7.11. The famous 3-body figure eight, first discovered by Mor93 and later analyzed by CM00, is the first one shown—labeled FigureEight3. It is easy to find choreographies of arbitrary complexity. In fact, it is not hard to rediscover most of the choreographies given in CGMS01, and more, simply by putting a loop in the AMPL model and finding various local minima by using different starting points.

However, as we discuss in a later section, simulation makes it apparent that, with the sole exception of FigureEight3, all of the choreographies we found are unstable. And, the more intricate the choreography, the more unstable it is. Since the only choreographies that have a chance to occur in the real world are stable ones, many cpu hours were devoted to searching for other stable choreographies. So far, none have been found. The choreographies shown in Figure 7.12 represent the ones closest to being stable.

Given the difficulty of finding stable choreographies, it seems interesting to search for stable nonchoreographic solutions using, for example, the AMPL

```
param N := 3;  # number of masses
param n := 15;  # number of terms in Fourier series representation
param m := 99;  # terms in num approx to integral.  must be a multiple of N

param lagTime := m/N;

set Bodies := {0..N-1};
set Times  := {0..m-1} circular; # "circular" means that next(m-1) = 0

param theta {t in Times} := t*2*pi/m;
param dt := 2*pi/m;

param a0 default 0;         param b0 default 0;
var as {k in 1..n} := 0;    var bs {k in 1..n} := 0;
var ac {k in 1..n} := 0;    var bc {k in 1..n} := 0;

var x {i in Bodies, t in Times}
  = a0+sum {k in 1..n} ( as[k]*sin(k*theta[(t+i*lagTime) mod m])
                       + ac[k]*cos(k*theta[(t+i*lagTime) mod m]) );
var y {i in Bodies, t in Times}
  = b0+sum {k in 1..n} ( bs[k]*sin(k*theta[(t+i*lagTime) mod m])
                       + bc[k]*cos(k*theta[(t+i*lagTime) mod m]) );

var xdot {i in Bodies, t in Times} = (x[i,next(t)]-x[i,t])/dt;
var ydot {i in Bodies, t in Times} = (y[i,next(t)]-y[i,t])/dt;

var K {t in Times} = 0.5*sum {i in Bodies} (xdot[i,t]^2 + ydot[i,t]^2);

var P {t in Times}
  = - sum {i in Bodies, ii in Bodies: ii>i}
           1/sqrt((x[i,t]-x[ii,t])^2 + (y[i,t]-y[ii,t])^2);

minimize A: sum {t in Times} (K[t] - P[t])*dt;

let {k in 1..n} as[k] := 1*(Uniform01()-0.5);
let {k in 1..n} ac[k] := 1*(Uniform01()-0.5);
let {k in n..n} bs[k] := 0.01*(Uniform01()-0.5);
let {k in n..n} bc[k] := 0.01*(Uniform01()-0.5);

solve;
```

Figure 7.11. AMPL program for finding choreographies by minimizing the action functional.

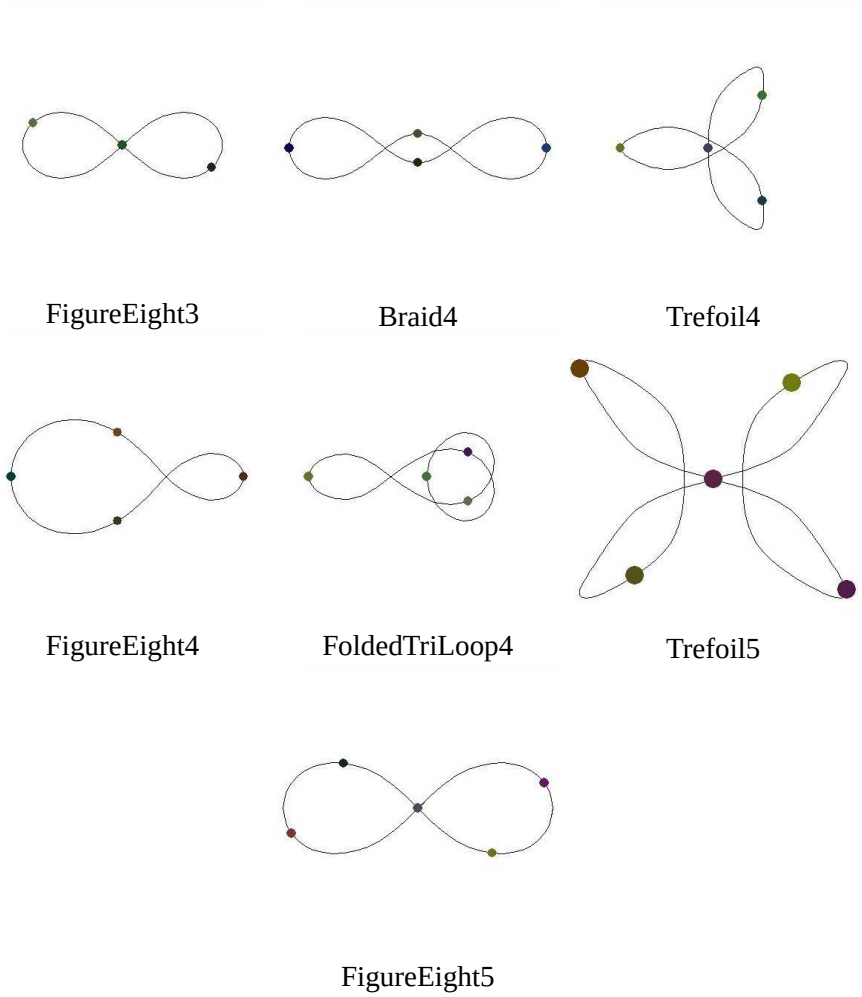


Figure 7.12. Periodic Orbits—Choreographies.

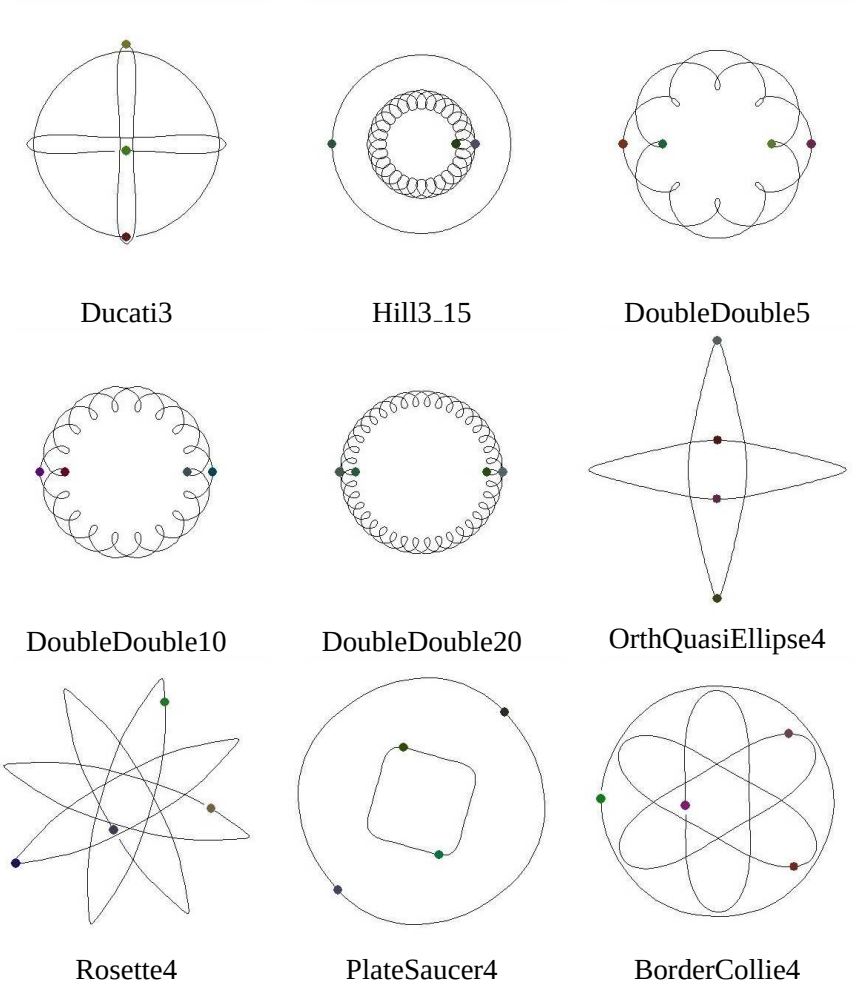


Figure 7.13. Periodic Orbits–Non-Choreographies.

model from Figure 7.10. The most interesting such solutions are shown in Figure 7.13. The one labeled Ducati3 is stable as are Hill3_15 and the three DoubleDouble solutions. However, the more exotic solutions (OrthQuasiEllipse4, Rosette4, PlateSaucer4, and BorderCollie4) are all unstable.

For the interested reader, a JAVA applet can be found at GravityApplet that allows one to watch the dynamics of each of the systems presented in this paper (and others). This applet actually integrates the equations of motion. If the orbit is unstable it becomes very obvious as the bodies deviate from their predicted paths.

Ducati3 and its Relatives

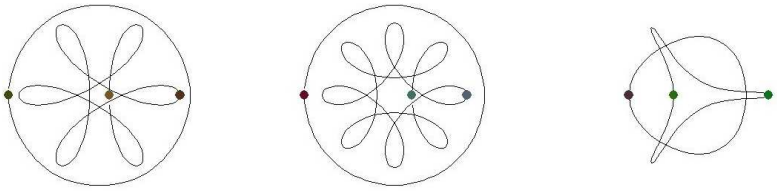
The Ducati3 orbit first appeared in Mor93 and has been independently rediscovered by this author, Broucke Bro03, and perhaps others. Simulation reveals it to be a stable system. The JAVA applet at GravityApplet allows one to rotate the reference frame as desired. By setting the rotation to counter the outer body in Ducati3, one discovers that the other two bodies are orbiting each other in nearly circular orbits. In other words, the first body in Ducati3 is executing approximately a circular orbit, $z_1(t) = -e^{it}$, the second body is oscillating back and forth roughly along the x-axis, $z_2(t) = \cos(t)$, and the third body is oscillating up and down the y-axis, $z_3(t) = i\sin(t)$. Rotating so as to fix the first body means multiplying by e^{-it}:

$$\begin{aligned}
\bar{z}_1(t) &= e^{-it}(-e^{it}) = -1 \\
\bar{z}_2(t) &= e^{-it}\cos(t) = (1 + e^{-2it})/2 \\
\bar{z}_2(t) &= e^{-it} i\sin(t) = (1 - e^{-2it})/2.
\end{aligned}$$

Now it is clear that bodies 2 and 3 are orbiting each other at half the distance of body 1. So, this system can be described as a Sun, Earth, Moon system in which all three bodies have equal mass and in which one (sidereal) month equals one year. The synodic month is shorter—half a year.

This analysis of Ducati3 suggests looking for other stable solutions of the same type but with different resonances between the length of a month and a year. Hill3_15 is one of many such examples we found. In Hill3_15, there are 15 sidereal months per year. Let Hill3_n denote the system in which there are n months in a year. All of these orbits are easy to calculate and they all appear to be stable. This success suggests going in the other direction. Let Hill3_$\frac{1}{n}$ denote the system in which there are n years per month. We computed Hill3_$\frac{1}{2}$ and found it to be unstable. It is shown in Figure 7.14.

In the preceding discussion, we decomposed these Hill-type systems into two 2-body problems: the Earth and Moon orbit each other while their center of mass orbits the Sun. This suggests that we can find stable orbits for the 4-body problem by splitting the Sun into a binary star. This works. The orbits



Hill3_2 Hill3_3 Hill3_0.5

Figure 7.14. Periodic Orbits—Hill-type with equal masses.

labeled DoubleDouble$n$ are of this type. As already mentioned, these orbits are stable.

Given the existence and stability of FigureEight3, one often is asked if there is any chance to observe such a system among the stars. The answer is that it is very unlikely since its existence depends crucially on the masses being equal. The Ducati and Hill type orbits, however, are not constrained to have their masses be equal. Figure 7.15 shows several Ducati-type orbits in which the masses are not all equal. All of these orbits are stable. This suggests that stability is common for Ducati and Hill type orbits. Perhaps such orbits can be observed.

Limitations of the Model

The are certain limitations to the approach articulated above. First, the Fourier series is an infinite sum that gets truncated to a finite sum in the computer model. Hence, the trajectory space from which solutions are found is finite dimensional.

Second, the integration is replaced with a Riemann sum. If the discretization is too coarse, the solution found might not correspond to a real solution to the n-body problem. The only way to be sure is to run a simulator.

Third, as mentioned before, all masses must be positive. If there is a zero mass, then the stationary points for the action function, which satisfy (7.6), don't necessarily satisfy the equations of motion given by Newton's law.

Lastly, the model, as given in Figure 7.10, can't solve 2-body problems with eccentricity. We address this issue in the next section.

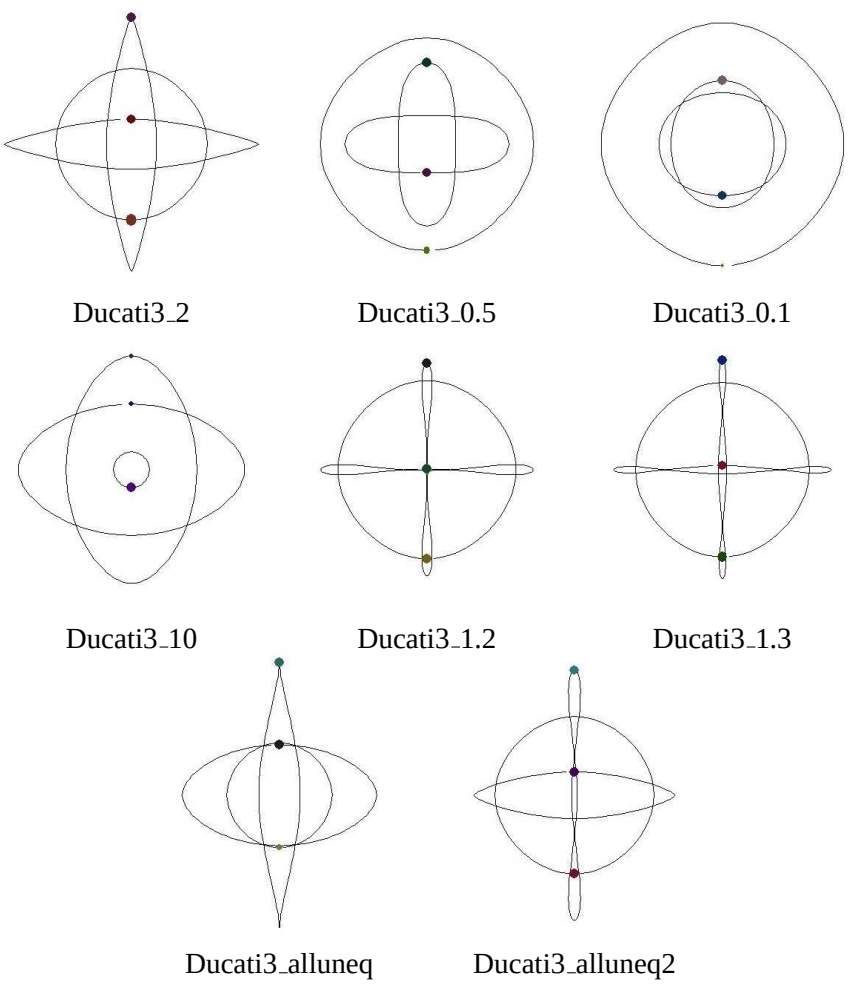


Figure 7.15. Periodic Orbits—Ducati's with unequal masses.

Elliptic Solutions

An ellipse with semimajor axis a, semiminor axis b, and having its left focus at the origin of the coordinate system is given parametrically by:

$$x(t) = f + a\cos t, \qquad y(t) = b\sin t,$$

where $f = \sqrt{a^2 - b^2}$ is the distance from the focus to the center of the ellipse.

However, this is *not* the trajectory of a mass in the 2-body problem. Such a mass will travel faster around one focus than around the other. To accommodate this, we need to introduce a time-change function $\theta(t)$:

$$x(t) = f + a\cos\theta(t), \qquad y(t) = b\sin\theta(t).$$

This function θ must be increasing and must satisfy $\theta(0) = 0$ and $\theta(2\pi) = 2\pi$.

The optimization model can be used to find (a discretization of) $\theta(t)$ automatically by changing `param theta` to `var theta` and adding appropriate monotonicity and boundary constraints. In this manner, more realistic orbits can be found that could be useful in real space missions.

In particular, using an eccentricity $e = f/a = 0.0167$ and appropriate Sun and Earth masses, we can find a periodic Hill-Type satellite trajectory in which the satellite orbits the Earth once per year.

References

M.P. Bendsøe, A. Ben-Tal, and J. Zowe. Optimization methods for truss geometry and topology design. *Structural Optimization*, 7:141–159, 1994.

R. Broucke. New orbits for the n-body problem. In *Proceedings of Conference on New Trends in Astrodynamics and Applications*, 2003.

A. Chenciner, J. Gerver, R. Montgomery, and C. Simó. Simple choreographic motions on n bodies: a preliminary study. In *Geometry, Mechanics and Dynamics*, 2001.

A. Chenciner and R. Montgomery. A remarkable periodic solution of the three-body problem in the case of equal masses. *Annals of Math*, 152:881–901, 2000.

J.O. Coleman. Systematic mapping of quadratic constraints on embedded fir filters to linear matrix inequalities. In *Proceedings of 1998 Conference on Information Sciences and Systems*, 1998.

J.O. Coleman and D.P. Scholnik. Design of Nonlinear-Phase FIR Filters with Second-Order Cone Programming. In *Proceedings of 1999 Midwest Symposium on Circuits and Systems*, 1999.

R. Fourer, D.M. Gay, and B.W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Scientific Press, 1993.

J.K. Ho. Optimal design of multi-stage structures: a nested decomposition approach. *Computers and Structures*, 5:249–255, 1975.

N.K. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

N. J. Kasdin, D. N. Spergel, and M. G. Littman. An optimal shaped pupil coronagraph for high contrast imaging, planet finding, and spectroscopy. *submitted to Applied Optics*, 2002.

N.J. Kasdin, R.J. Vanderbei, D.N. Spergel, and M.G. Littman. Extrasolar Planet Finding via Optimal Apodized and Shaped Pupil Coronagraphs. *Astrophysical Journal*, 582:1147–1161, 2003.

M.S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. Technical report, Electrical Engineering Department, Stanford University, Stanford, CA 94305, 1998. To appear in *Linear Algebra and Applications* special issue on linear algebra in control, signals and imaging.

I.J. Lustig, R.E. Marsten, and D.F. Shanno. Interior point methods for linear programming: computational state of the art. *Operations Research Society of America Journal on Computing*, 6:1–14, 1994.

C. Moore. Braids in classical gravity. *Physical Review Letters*, 70:3675–3679, 1993.

D. N. Spergel. A new pupil for detecting extrasolar planets. *astro-ph/0101142*, 2000.

R.J. Vanderbei. LOQO user's manual—version 3.10. *Optimization Methods and Software*, 12:485–514, 1999.

R.J. Vanderbei. http://www.princeton.edu/∼rvdb/JAVA/astro/galaxy/Galaxy.html, 2001. .

R.J. Vanderbei. *Linear Programming: Foundations and Extensions*. Kluwer Academic Publishers, 2nd edition, 2001.

R.J. Vanderbei and D.F. Shanno. An interior-point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, 13:231–252, 1999.

R.J. Vanderbei, D.N. Spergel, and N.J. Kasdin. Circularly Symmetric Apodization via Starshaped Masks. *Astrophysical Journal*, 599:686–694, 2003.

R.J. Vanderbei, D.N. Spergel, and N.J. Kasdin. Spiderweb Masks for High Contrast Imaging. *Astrophysical Journal*, 590:593–603, 2003.

S.-P. Wu, S. Boyd, and L. Vandenberghe. Magnitude filter design via spectral factorization and convex optimization. *Applied and Computational Control, Signals and Circuits*, 1997. To appear.