```
<<Anopt`;

g = Abs[1/(e + 0.5 ) - z[1] ]^2
    + ( 5 + Re[1/(2 + e)] ) Abs[ 1/e + z[2]]^2
    + 2 Im[z[1] z[2]]^2;

Anopt[g,0.000001];
```

In the session above the input is the performance "g" and the number 0.01, which is an error tolerance for stopping calculations. The following is output that is printed on the screen:

```
It :    Current Value     :  Step   : Optimality Tests  :  Error  : Sm. : Grid
   :      SupGamma        :         :   OT1       OT2   :   NED   :     :
-----------------------------------------------------------------------------
 0 : 1.              E+01 :   N/A   : 4.2E-01 : 1. E-02 :   N/A   : NON : 32
 1 : 6.9217870409218E+00 : 2.3E-01 : 2.3E-04 : 7.1E-04 : 5. E-02 : NON : 32
 2 : 6.9214867847569E+00 : 1.9E-03 : 2.4E-05 : 4.7E-05 : 7.1E-04 : NON : 32
 3 : 6.921495036711 E+00 : 0  E+00 : 2.6E-05 : 4.7E-05 : 7.1E-04 : 5A10: 32
 4 : 6.921446086706 E+00 : 1.3E-04 : 4. E-07 : 1.2E-06 : 1.7E-07 : NON : 64
 5 : 6.9214453945937E+00 : 3.4E-06 : 4. E-10 : 1.3E-08 : 2.6E-07 : NON : 64


Summary
-------
Supremum of gamma: SupGamma = 6.921445394593699E+00
Optimality Test  : OT1      = 4.0149612824E-10
Optimality Test  : OT2      = 1.3493172843E-08
Error diagnostic : NED      = 2.6 E-07
```

The final value of SupGamma is the calculated value of γ^* in the statement OPT. The numbers under the headings OT1 and OT2 are the result of optimality tests. At the solution, they should be equal to zero. The analytic function that is the calculated solution is also an output, but is not printed. The reader interested in more details on the output and on computer solutions may read the notebook AnoptTutorial.ma, which is an appendix in this book. Also OT1 and OT2 are based on the theory developed in the next three sections.

15.3 Necessary conditions for solutions

As in the scalar case (Theorem 13), diagnostics and computer algorithms in the vector case are based on a theorem characterizing solutions to OPT.

We now present a result that gives two necessary conditions for a solution. These two conditions already give enough information to derive an algorithm (see **[HMW]**). Additional conditions to insure sufficiency will be presented in Section 17.1.

So far we have mentioned "solutions f in A_N to OPT" that are global solutions. However in the study of conditions for optimality one must consider *local* solutions. This is needed because when $N > 1$, there is no uniqueness of solutions.

A *local solution* in A_N is a solution to OPT in some neighborhood $V \subset A_N$ of f. It is a *strict local solution* if it is unique in V. A function $h \in A_N$ is a *descent direction* relative to $f \in A_N$ if for all $t > 0$ small enough,

$$\sup_\theta \Gamma(e^{j\theta}, f(e^{j\theta}) + th(e^{j\theta})) < \sup_\theta \Gamma(e^{j\theta}, f(e^{j\theta}))$$

The function f is a *directional solution* or *directional minimizer* if there are no descent directions relative to f. Of course, every strict local solution is a directional solution.

Recall that Γ is a function of z where $z = (z_1, \ldots, z_N)$, so we can take partial derivatives such as $\frac{\partial\Gamma}{\partial z_1}(e^{j\theta}, z_1, \ldots, z_N)$ and $\frac{\partial\Gamma}{\partial z_2}(e^{j\theta}, z_1, \ldots, z_N)$.

Theorem 20 *Let Γ be of class C^3 and $f^* = (f_1, \ldots, f_N)$ in A_N be such that*

$$a(e^{j\theta}) := \left(\frac{\partial\Gamma}{\partial z_1}(e^{j\theta}, f^*(e^{j\theta})), \ldots, \frac{\partial\Gamma}{\partial z_N}(e^{j\theta}, f^*(e^{j\theta}))\right) \neq (0, \ldots, 0), \qquad \forall\theta$$

If the function f^* is a directional solution to OPT *then*

I $\Gamma(e^{j\theta}, f^*(e^{j\theta}))$ *is a constant function of $e^{j\theta}$.*

II *There exist $F = (F_1, \ldots, F_N)$ with F_ℓ analytic on the unit disk D and integrable boundary values $F_\ell(e^{j\theta})$, and there exists a positive valued function $\lambda(e^{j\theta})$ such that*

$$\begin{array}{ccc} \frac{\partial\Gamma}{\partial z_1}(e^{j\theta}, f^*(e^{j\theta})) & = & \lambda(e^{j\theta})e^{j\theta}F_1(e^{j\theta}) \\ \vdots & \vdots & \vdots \\ \frac{\partial\Gamma}{\partial z_N}(e^{j\theta}, f^*(e^{j\theta})) & = & \lambda(e^{j\theta})e^{j\theta}F_N(e^{j\theta}) \end{array}$$

15.4 An Example

Consider the performance function

$$\Gamma(e^{j\theta}, z_1, z_2) = |z_1|^2 \quad +|z_2|^2 + |100 + e^{j\theta} z_1 + .1(z_1 + z_2 + z_1 z_2)|^2$$
$$+|100 + e^{j\theta} z_2 + .1(z_1 + z_2 + z_1 z_2)|^2 \tag{15.3}$$

This is a good example because, as we shall see, *there are two strict local minimizers f_1 and f_2, and there exists also a function f_0 that is a sort of " saddle point", i.e., f_0 satisfies conditions (I) and (II) but it is not a minimizer.*

In order to decide whether a candidate is or is not a solution we need sufficient conditions for optimality. Basically the sufficient conditions consist of (I), (II), and an additional condition that will be introduced later. For now we will only verify (I) and (II), leaving the rest to a later section.

The three functions are the following:

$$\begin{aligned} f_0(e^{j\theta}) &= (0,0) \\ f_1(e^{j\theta}) &= (30,-30) \\ f_2(e^{j\theta}) &= (-30,30) \end{aligned} \tag{15.4}$$

The calculations for the function f_2 are almost identical to those for f_1, so we present only those corresponding to f_1. We have,

$$\Gamma(e^{j\theta}, f_1(e^{j\theta}))) = \Gamma(e^{j\theta}, 30, -30) = 3800 \qquad \forall e^{j\theta}$$

so f_1^* satisfies (I). To check (II) we begin by calculating partial derivatives:

$$\begin{aligned} \frac{\partial \Gamma}{\partial z_1}(e^{i\theta}, z_1, z_2) &= \\ & \overline{(100 + z_1 e^{i\theta} + 0.1(z_1 z_2 + z_1 + z_2)} \cdot (e^{i\theta} + 0.1 + 0.1 z_2) \\ & + \overline{(100 + z_2 e^{i\theta} + 0.1(z_1 z_2 + z_1 + z_2)} \cdot (0.1 + 0.1 z_2) + \overline{z_1} \\ \frac{\partial \Gamma}{\partial z_2}(e^{i\theta}, z_1, z_2) &= \\ & \overline{(100 + z_1 0.1(z_1 z_2 + z_1 + z_2))}(0.1 + 0.1 z_1) \\ & + \overline{(100 + z_2 e^{i\theta} + 0.1(z_1 z_2 + z_1 + z_2))}(e^{i\theta} + 0.1 + 0.1 z_1) + \overline{z_2} \end{aligned} \tag{15.5}$$

Therefore,

$$\begin{aligned} \frac{\partial \Gamma}{\partial z_1}(e^{i\theta}, 30, -30) &= 10 e^{j\theta} + 2 \\ \frac{\partial \Gamma}{\partial z_2}(e^{i\theta}, 30, -30) &= 10 e^{j\theta} + 2 \end{aligned} \tag{15.6}$$

Note that

$$10e^{j\theta} + 2 \;=\; e^{j\theta}\,|1 + 2e^{j\theta}|^2\,\frac{1}{10 + 2e^{j\theta}} \tag{15.7}$$

and this implies that condition (II) is satisfied by f_1 with $\lambda(e^{j\theta}) = |1{+}2e^{j\theta}|^2$ and $F_\ell(e^{j\theta}) = 1/(10 + 2e^{j\theta})$, for $\ell = 1, 2$.

Now calculations with f_0 are next. Condition I holds since

$$\Gamma(e^{i\theta}, 0, 0) = 20,000 \quad (\forall\, e^{i\theta})\,.$$

To check II, evaluate the partial derivatives of Γ to obtain

$$\left(\frac{\partial\Gamma}{\partial z_1}(e^{i\theta}, 0, 0),\, \frac{\partial\Gamma}{\partial z_2}(e^{i\theta}, 0, 0)\right) = (100\, e^{i\theta} + 20, 100\, e^{i\theta} + 20)\,.$$

Note that

$$100\, e^{i\theta} + 20 = 100\,|e^{i\theta} + 0.2|^2\,\frac{e^{i\theta}}{1 + 0.2e^{i\theta}}$$

so II holds with $\lambda(e^{j\theta}) = 100\,|e^{i\theta} + 0.2|^2$ and $F_\ell(e^{j\theta}) = e^{j\theta}/(1 + 0.2e^{j\theta})$, $\ell = 1, 2$.

15.5 Computer diagnostics

Diagnostics and computer algorithms for the vector case follow immediately from the previous two sections and are a direct generalization of those for the scalar case found in Section 13.

Condition I of Theorem 20 can be easily implemented on a computer, by calculating

$$OT_1(f) = \frac{\sup_\theta\; \Gamma(e^{j\theta}, f(e^{j\theta})) - \inf_\theta\; \Gamma(e^{j\theta}, f(e^{j\theta}))}{\sup_\theta\; \Gamma(e^{j\theta}, f(e^{j\theta}))}$$

To test condition II directly one also needs functions λ and F. If candidates for these functions are not already available, one would have to test every possibility in λ and F, an impossible task. Thus it is desirable to have a test of (II) that does not involve λ and F. The following conjecture is crucial in this respect:

Conjecture. For generic Γ condition II of Theorem 2 is equivalent to:

> Let $a_j = \frac{\partial\Gamma}{\partial z_j}(\cdot, f^*)$. Fix j_0. For each $j = 1, \ldots, N$, the function $\frac{a_j}{a_{j_0}}$ is analytic in the disk.

It is easy to check on the computer when this should apply. Thus the conjecture above allows one to check (for most performance functions) condition (II) of Theorem 20 by testing if each a_j/a_0 is analytic, provided $N > 1$. When $N = 1$ this test does not apply as stated, however it does reduce to the winding number test of Section 13.

To implement condition II, consider the Fourier expansion of the functions

$$\frac{a_\ell}{a_{\ell_0}} = \sum_{n=-\infty}^{\infty} c_{\ell,n} e^{jn\theta}$$

and form

$$OT_2(f) = (\sum_{\ell=1}^{N} \sum_{n<0} |c_{\ell,n}|^2)^{1/2}$$

which is calculated on the computer with the Fast Fourier Transform. At optimum both OT_1 and OT_2 should equal zero to nearly machine precision. The program *Anopt* prints these values out at each iteration to monitor how the approximate solutions are progressing. If f is near the solution we must have $OT_1(f) \approx 0$ and $OT_2(f) \approx 0$.

To illustrate the practical use of the diagnostics we consider the performance function

$$\begin{aligned}\Gamma(e^{j\theta}, z_1, z_2) = \quad & |z_1|^2 + |z_2|^2 + |100 + e^{j\theta} z_1 + .1(z_1 + z_2 + z_1 z_2)|^2 \\ & +|100 + e^{j\theta} z_2 + .1(z_1 + z_2 + z_1 z_2)|^2\end{aligned} \tag{15.8}$$

that was studied in Section 15.4. We already know that $f_1(e^{j\theta}) = (30, -30)$ and are directional solutions, and that $f_0(e^{j\theta}) = (0, 0)$ gives a saddle point (I and II are satisfied, but f_0 is not a solution).

Implementing the algorithm *Newton Iteration* (see sections 13.3 and 16) yields computer output, some of which we present below. The output shown in Tables 1 and 2 (except for the last column) was produced by the program Anopt written in Mathematica.

Initializing the iteration at $f_0(e^{j\theta}) = (29.6 + .1e^{j\theta}, -30.4 - .0001e^{j\theta} + .001(e^{j\theta})^2)$ produces the results indicated in Table 1 below. Since in this example we know what the solution is, we include in Table 1 a column that shows the true error $||f^k - f^*||_\infty$.

Table 1

Iter	Value Sup Gamma	Optimality Tests OT1	OT2	True error ||fk-f*||
0	3.8165268771511E+03	8.3E-03	1.6E+00	5.7E-01
1	3.8009177903975E+03	4.9E-04	3.9E-01	5.3E-02
2	3.8000812563274E+03	4. E-05	1.5E-02	2.7E-03
3	3.8000001798719E+03	6.3E-08	2. E-05	4.4E-06
4	3.8000000000006E+03	2.4E-13	3.6E-10	1.5E-11

In the next example the Newton iteration exhibits second order convergence to the constant function $f^*(e^{j\theta}) = (0,0)$, which is not a directional minimizer. The iteration was initialized with $f(e^{j\theta}) = (0.2 + 0.1e^{j\theta}, 1.4 + .1e^{j\theta} + 0.01(e^{j\theta})^2)$.

Table 2

Iter	Value Sup Gamma	Optimality Tests OT1	OT2	True error ||fk-f*||
0	2.0428618081601E+04	3.2E-02	2.9E-01	1.4E+00
1	2.0057600281609E+04	5.7E-03	9.3E-02	1.5E+00
2	2.0097292539204E+04	1.2E-02	3.6E-02	7. E-01
3	2.000858365994 E+04	6.8E-04	3.4E-03	3.1E-02
4	2.0000014412272E+04	1.4E-06	6.6E-06	4.8E-05
5	2.0000000000109E+04	9.7E-12	7.3E-11	7.6E-10

In these tests we have illustrated that, even in this rather elaborate function space setting, Newton's method behaves exactly as it would in traditional R^n optimization. Namely, it converges to a place where the "gradient" vanishes[2], be it a maximum, a minimum, or a saddle point.

15.6 Algorithms for OPT

The algorithms for OPT discussed in this book are instances of the following general scheme for solving nonlinear problems iteratively:

General Iteration Scheme

1. Test f for optimality. If optimal, then stop.
2. Solve a subproblem to obtain an update direction h.
3. Perform a search on $f + th$ with $t \geq 0$ to find a $t = t_0$ that satisfies a preset criterion.

[2] In our setting, this is conditions I and II of Theorem 20 are satisfied.

4. Set $f = f + t_0 h$, and go to step 1.

The tests for optimality OT_1 and OT_2 we have developed are used in Step 1. Another stopping criterion one may want to include in particular implementations is "lack of sufficient progress" (however you want to define this). A typical Step 2 consists of replacing the performance function with a "model", a new performance which is based on the original function but easier to deal with. Step 3 is called *line search.* One obvious example of the linesearch criterion is that t minimizes $\sup_\theta \Gamma(e^{j\theta}, f(e^{j\theta}) + th(e^{j\theta}))$.

The two algorithms discussed later in this book (see Chapter 16) are *Newton Iteration* and *Disk Iteration.* Both are based on models of the performance obtained by taking some terms from the Taylor expansion of $\Gamma(e^{j\theta}, z)$ about $z =$ the current guess f.

This section addresses what we have found to be the most effective means of evaluating the performance of a computer algorithm. The objective is to have a theoretical test, which can be done with pencil and paper at the same time that you are developing an algorithm. This is tremendously helpful in discovering computer algorithms. the reason is that the algorithm itself is presented analytically and having an analytic means of evaluation allows one to see how to make changes which improve performance. What we recommend as the leading indicator of success of an algorithm is actually very conventional. It is the *order of convergence* .

A sequence of vectors x_n in a vector space with norm $\|\ \|$ that converges to x_* is said to be p-convergent if there exists $C > 0$ such that

$$\text{(EST)} \qquad \|x_n - x_*\| < C\|x_n - x_*|^p$$

The larger p the faster the sequence converges. An algorithm which generates sequences will be called *order p convergent* provided the sequences it generates are at least *order p convergent.* Here C is a number whose size influences the error estimate (EST) less than p. When $p = 1$ we need $C < 1$ to guarantee improvement from one iteration to the next.

A summary of the performance of algorithms presented in this book is:

Method	*Case*	*Convergence*
Disk Iteration	$N = 1$	order 2
	$N > 1$	order 1
Newton Iteration	all N	order 2

Some technical points:

- Convergence of the Disk Iteration algorithms has not been established. However one can show that if it converges, then the limit function f^* must satisfy the optimality conditions I and II. *We conjecture that if Disk Iteration with a line search converges to f^*, then f^* is a local solution of OPT.* The difficulty with establishing convergence is that in the function spaces over which the optimization is performed, bounded sets are not necessarily compact. In such spaces the distance between elements of a sequence may decrease to 0 and the sequence may still not converge. Our experience indicates that formal order of convergence is easier to analyze and of much more practical importance in predicting what you see on a computer than actual convergence itself.
- We have not specified which norm we are using in the table. The details are rather complicated and do not contribute to our exposition.
- Newton iteration may converge to an f^* which is not a local minimum, but a sort of critical point. See **[HMer1-2]**.

15.7 References

The main theorem in this section is from **[HMer2]**. More results can be found in **[H3]** and in **[HMar]**. See **[H1]** for connections with other areas of Mathematics. Several Complex Variables connections are discussed in **[HMer1]**, see also **[HH]**, **[HMar]** and **[HV]**.

[H1] J. W. Helton. Operator Theory, Matrices and Electrical Engineering. NSF. 1987

[H2] J. W. Helton. "Optimization over spaces of analytic functions and the Corona problem," J. Operator Theory, 15 (1986), No. 2, 359–375.

[H3] J. W. Helton: Optimal frequency domain design vs. an area of several complex variables, plenary address. MTNS, 1989.

[HH] J. W. Helton and R. Howe. "A bang-bang theorem for optimization over spaces of analytic functions," J. Approx. Theory, 47 (1986), No. 2, 101–121.

[HMar] J. W. Helton and D. Marshall: "Frequency domain design and analytic selections," Indiana Univ. Math. Journal, 39 (Spring 1990), No. 1, 157–184.

[HMer1] J. W. Helton and O. Merino: "Optimal Analytic Disks," Collection: Several complex variables and complex geometry, Part 2, (Santa Cruz, CA, 1991) Proc. Sympos. Pure Math., 5 251–262.

[HMer2] J. W. Helton and O. Merino. Conditions for Optimality over $\mathcal{H}^\infty$. SIAM J. Cont. Opt., Vol. 31, No. 6, November 1993.

[HV] J. W. Helton, and A. Vityaev. Preprint

Chapter 16

More Numerical Algorithms

This chapter is one of the more theoretical parts of the book which is ironic in that it derives the formulas which are used in our computer algorithms. The authors are a bit iconoclastic in that they see developing computer algorithms of this type as almost an exercise in pure mathematics. Indeed the next appendix describes pure mathematics implications of this and previous sections. A design engineer can skip this section. An engineer who develops optimization methods might find it interesting.

In this chapter we discuss in some detail two algorithms for solving OPT , Disk Iteration and Newton Iteration. The discussion begins with the statement of classical results on distance problems in Section 16.2. Disk Iteration and Newton algorithms are presented in Sections 16.3 and 16.4. The chapter ends with Section 16.5 which treats the connection between the classical results and the characterization of solutions to OPT.

16.1 Notation

We define the spaces L^p_N, H^p_N, whose elements are N-tuples of functions $(f_1, \ldots, f_N)$ and whose norms are defined in terms of the euclidean norm on C^N, as in

$$\begin{aligned} ||f||_{L^p_N} &= \left(\int_0^{2\pi} ||(f_1(e^{j\theta}), \ldots, f_N(e^{j\theta}))||^p_{C^N} \frac{d\theta}{2\pi}\right)^{\frac{1}{p}} \qquad (1 \leq p < \infty) \\ ||f||_{L^\infty_N} &= \sup_\theta ||(f_1(e^{j\theta}), \ldots, f_N(e^{j\theta}))||_{C^N} \end{aligned} \tag{16.1}$$

If $f \in L^p_N$ for some $1 \leq p \leq \infty$, then the Fourier coefficients $\hat{f}_\ell$ are vectors in C^N.

The orthogonal complement in L^2_N of a subspace $\mathcal{S}$ is denoted $\mathcal{S}^\perp$. If k is an $M \times N$-matrix valued function on the unit circle, the Hankel operator with symbol k is the operator $\mathcal{H}_k : H^2_M \to H^2_{N,0}$ with action $\alpha \to P_{H^{2\perp}_{N,0}}[k\alpha]$. The Toeplitz operator with symbol k is the operator $\mathcal{T}_k : H^2_M \to H^2_N$ with action $\alpha \to P_{H^2_N}[k\alpha]$.

16.2 Nehari's problem: perspective on OPT from a classical viewpoint

The math problems treated and used in this book long preceded engineering application. Some go back to the beginning of this century. A variant which is close to the spirit of this book is the classical Nehari problem which dates from the 1950's.

Nehari Problem. Let k be a function in L^∞_N, find its distance to H^∞_N.

Here the distance of k to H^∞_N means

$$\operatorname{dist}(k, H^\infty_N) := \inf_{f \in H^\infty_N} ||k - f||_{L^\infty_N} \tag{16.2}$$

and a *best H^∞ approximation to k* is a function $f_0 \in H^\infty_N$ that realizes the supremum in (16.2). Standard theory shows that f_0 exists. Moreover, if $k \in \mathcal{C} + H^\infty_N$ then the best approximation is unique (here $\mathcal{C}$ is the set of continuous functions on the unit circle). We refer to the problem of calculating $\operatorname{dist}(k, H^\infty_N)$ and a minimizer $f_0 \in H^\infty_N$ as *Nehari's problem.* Nehari's problem is also the simplest nontrivial OPT problem with the performance

$$\Gamma(e^{j\theta}, z) = ||k(e^{j\theta}) - z||_{C^N} \tag{16.3}$$

Let $\mathcal{H}_k : H^2_1 \to {H^2_N}^\perp$ be the operator with action $\alpha \to P_{H^{2\perp}_N}[k\alpha]$. This type of operator is called a *Hankel* operator and k is called its symbol. A well known result is:

Theorem 21 *If k is in $\mathcal{L}^\infty_N$, then*

$$||\ \mathcal{H}_k\ ||_{\mathcal{L}} \ = \ \operatorname{dist}(k, H^\infty_N) \tag{16.4}$$

and there exists a best approximant $f_0 \in \mathcal{H}^\infty_N$. If α_0 is a function at which $\mathcal{H}_k$ attains its norm, then

$$k - f_0 = \frac{1}{\alpha_0}\mathcal{H}_k[\alpha_0] \tag{16.5}$$

If the symbol k is in $\mathcal{H}^\infty + \mathcal{C}$, then f_0 is unique. Let $\mathcal{H}_k^*$ be the adjoint of $\mathcal{H}_k$, and set $\tau = \mathrm{dist}(k, H_N^\infty)$; then

$$\mathcal{H}_k^*[\ \mathcal{H}_k[\alpha_0]\ ] = \tau^2 \alpha_0 \tag{16.6}$$

Equation (16.6) says that α_0 is a function associated with τ, the largest singular value of the compact operator $\mathcal{H}_k$ (i.e., the norm of $\mathcal{H}_k$). We will refer to equation (16.6) as the *Nehari - Commutant Lifting formula.* For $N > 1$ this follows from commutant lifting arguments **[A]**, **[S]**, **[NF]**, **[AAK1–3]**.

16.3 Disk Iteration algorithms

The solution of Nehari's problem leads directly to numerical methods for solving OPT problems where the sublevel sets are disks ("disk problems"). To solve more general OPT problems we can successively approximate OPT by disk problems and solve them. We call this Disk Iteration. These are described together with experimental findings in **[HMer1]**, and here we only sketch the basic idea.

16.3.1 The Power Method for solving Nehari's Problem

A common way to solve Nehari's problem is to solve (16.6) for α_0 as an eigenvalue problem, and then solve for f_0 in (16.5). The power method is a natural choice since we are seeking the largest eigenvalue of the operator $\mathcal{H}_k^*\mathcal{H}_k$, and the corresponding eigenvector. It is a well known fact that, provided the multiplicity of this eigenvalue is equal to 1, the power method converges linearly to the solution.

The Power Method. Given k and a guess α a solution to (16.6), update α to α_1 with

P1. $\alpha_+ := P_{H_N^2}[\ \overline{k}^t P_{H_N^{2\perp}}[\ k\alpha\ ]\ ]$.

P2. $\alpha_1 := \frac{1}{\|\ \alpha_+\ \|_{L_1^2}}\alpha_+$.

Suppose that the unit circle is discretized to n (a power of 2) equally spaced points, so that one can calculate projections in $2n\log_2(n)$ operations by using a Fast Fourier Transform algorithm (we do not count additions). Then step P1 takes $N(4n\log_2(n) + 2n)$ operations, and step P2 takes Nn

multiplications, 1 square root and Nn divisions. Thus the power method for solving Nehari's problem is of order $Nn\log_2(n)$. Also, note that memory requirements are low since no matrices have to be stored at any moment.

For comparison we mention that Newton Iteration requires the inversion of a matrix M that arises from a discretization of the operator $T\prime_{f,\beta}$ introduced in Sections 13.3 and 16.4. The matrix M has a number of entries that scale with Nn^2, but it has special structure that can be exploited to reduce the operation count in the inversion (with algorithms of the type found in **[SLK]**), possibly reducing the order of the inversion to Nn^2. Also there exist algorithms for inverting a matrix by iteration, which can be effective in this setting.

The power method is clearly efficient in many respects, and it is superior to the Newton Iteration in memory requirements and operation count. Obviously it is also better when it comes to time actually used to produce an update in step 2 of the iteration scheme. However, there is a crucial limitation of the power method: it is directly applicable only to very simple problems OPT, namely Nehari problems.

The properties of the power method make it an interesting tool for solving OPT problems through approximating Γ by Taylor expansions. This Taylor expansion is used to obtain a Nehari problem, or at least a problem close to it. This has been attempted before in **[HMer1]**. In Subsection 16.3.2 below the method of *Disk iteration*, see **[HMer1]**, is described.

16.3.2 The Algorithm

We now introduce a Disk Iteration algorithm.

Disk Iteration. Given a current guess at the solution $f \in A_N$, the Step 2 of the Scheme is:

- Find $h = (h_1, h_2, \ldots, h_N) \in A_N$ that minimizes

$$\sup_{e^{j\theta}} \left\{ \Gamma(\cdot, f) + 2 \text{ Re } \left(\frac{\partial \Gamma}{\partial z}(\cdot, f)^t h \right) + \sum_{k=1}^{N} \frac{\partial^2 \Gamma}{\partial z_k \partial \overline{z_k}}(\cdot, f) \; |h_k|^2 \right\} \tag{16.7}$$

A first glance at the algorithms reveals that the Taylor expansion of Γ about f, the current guess at the solution, is not necessarily of the form of a Nehari problem. However, it can be solved by iterating Nehari problems **[HMer1]**. Thus an implementation of the Disk iteration algorithm that uses the power method requires an inner loop to find a solution to Step 2 [1].

Another feature of (16.7) is that the Taylor expansion it contains does not capture all second order information from most functions Γ. One expects that this would imply first order (bad) convergence for the algorithm. This is indeed true in the vector valued case $N > 1$, but it is surprising that (in practice) second order (good) convergence occurs in the scalar case $N = 1$ (see **[HMer1]**).

Thus Disk iteration gives second order convervence in practice when $N = 1$, and no better than first order when $N > 1$.

16.4 More on the Newton iteration algorithm

This section describes Newton's method of solving OPT when $N > 1$. We follow closely the reference **[HMW]**. Recall that in the scalar case (Section 13.3) the key was a set of Critical Point equations. The vector case (C^N-valued analytic functions) of the Critical Point equations is:

$$\begin{aligned} P_{H^{2\perp}_{N,0}}\left[(1+2\ \mathrm{Re}\ (\chi\beta))\ \tfrac{\partial\Gamma}{\partial z}(\cdot,f)\right] &= 0 \\ P_{H^{2\perp}}\left[\ \Gamma(\cdot,f)\ \right] &= 0 \end{aligned} \tag{16.8}$$

where $\lambda^{-1} = 1 + \ \mathrm{Re}\ (\chi\beta)$ and where β is a scalar valued function which is analytic on the disk. Thus we get the nonlinear equation

$$T\begin{pmatrix} f \\ \beta \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{16.9}$$

where in equation (16.9),

$$T\begin{pmatrix} f \\ \\ \beta \end{pmatrix} = \begin{pmatrix} P_{H^{2\perp}_{N,0}}\left[(1+2\ \mathrm{Re}\ (\chi\beta))\ \frac{\partial\Gamma}{\partial z}(\cdot,f)\right] \\ \\ P_{H^{2\perp}}\left[\ \Gamma(\cdot,f)\ \right] \end{pmatrix} \tag{16.10}$$

Thus our objective is to solve the Critical Point Problem (16.9) with T given by (16.10). Here χ is the function on the unit circle defined by $\chi(e^{j\theta}) = e^{j\theta}$.

Now we turn to the solution of operator equations (e.g. equation (16.9)) with the iterative method of Newton. In terms of T' we have the Newton's

[1] We note that there are many reasonable modifications of the expression (16.7) that preserve intact terms of order up to 1 in h. In fact in **[HMer1]** one was found which eliminates the inner loop at the expense of *all* the second order information.

step for updating a given pair (f, β):

$$\begin{pmatrix} \tilde{f} \\ \tilde{\beta} \end{pmatrix} = \begin{pmatrix} f \\ \beta \end{pmatrix} - T'^{-1}_{f,\beta} T \begin{pmatrix} f \\ \beta \end{pmatrix} \tag{16.11}$$

We refer to (16.11) as *Newton iteration*, and to the repeated application of (16.11) as the *Newton algorithm.*

We now derive the formula for the Jacobian $T'_{f,\beta}$ of T, which we obtain by linearizing (16.10).

$$\begin{aligned} \Gamma(\cdot, f+\epsilon) &= g + 2\ \mathrm{Re}\ (a^t \epsilon) + \overline{\epsilon}^t A\epsilon + \ \mathrm{Re}\ (\epsilon^t B\epsilon) + \cdots \\ \frac{\partial\Gamma}{\partial z}(\cdot, f+\epsilon) &= a + A\overline{\epsilon} + B\epsilon + \cdots \end{aligned} \tag{16.12}$$

where

$$g = \Gamma(\cdot, f), \quad a = \frac{\partial\Gamma}{\partial z}(\cdot, f), \quad A = \frac{\partial^2\Gamma}{\partial\overline{z}\partial z}(\cdot, f), \quad \mathit{and} \quad B = \frac{\partial^2\Gamma}{\partial z^2}(\cdot, f). \tag{16.13}$$

Consider the expression

$$T\begin{pmatrix} f+\epsilon \\ \beta+\delta \end{pmatrix} = \begin{pmatrix} P_{H^{2\perp}_{N,0}}\left[\ (1 + 2\ \mathrm{Re}\ (\chi(\beta+\delta)))\ \frac{\partial\Gamma}{\partial z}(\cdot, f+\epsilon)\ \right] \\ P_{H^{2\perp}_1}\left[\ \Gamma(\cdot, f+\epsilon)\ \right] \end{pmatrix} \tag{16.14}$$

Substituting (16.12) into (16.14) and dropping all terms which are not linear or constant in δ or ϵ yields

$$T\begin{pmatrix} f \\ \beta \end{pmatrix} + T'_{f,\beta}\begin{pmatrix} \epsilon \\ \delta \end{pmatrix} =$$
$$\begin{pmatrix} P_{H^{2\perp}_{N,0}}\left[\ (1 + 2\ \mathrm{Re}\ (\chi(\beta+\delta)))\,(a + A\overline{\epsilon} + B\epsilon) + 2\ \mathrm{Re}\ (\chi\delta)a\ \right] \\ P_{H^{2\perp}_1}\left[\ g + 2\ \mathrm{Re}\ a^t\epsilon\ \right] \end{pmatrix} \tag{16.15}$$

Thus we have that the Jacobian of T is given by

$$T'_{f,\beta}\begin{pmatrix} \epsilon \\ \delta \end{pmatrix} = \begin{pmatrix} P_{H^{2\perp}_{N,0}}\left[\ (1 + 2\ \mathrm{Re}\ (\chi\beta))\,(A\overline{\epsilon} + B\epsilon) + 2a\ \mathrm{Re}\ (\chi\delta)\ \right] \\ P_{H^{2\perp}_1}\left[\ 2\ \mathrm{Re}\ a^t\epsilon\ \right] \end{pmatrix} \tag{16.16}$$

An elegant way to write T' appears in the following proposition:

Proposition 4 *The operator $T'_{f,\beta}$ in (16.16) can be written in the form*

$$T'_{f,\beta} = \begin{pmatrix} I_N & 0 \\ 0 & \overline{\chi} \end{pmatrix} \mathcal{C}\mathcal{T}_{M_1} + \begin{pmatrix} \chi I_N & 0 \\ 0 & 1 \end{pmatrix} \mathcal{H}_{M_2} \tag{16.17}$$

where $\mathcal{T}_{M_1}$ *is a Toeplitz operator with symbol* $M_1 = \begin{pmatrix} wA & \chi\overline{a} \\ \overline{\chi}a^t & 0 \end{pmatrix}$, $\mathcal{H}_{M_2}$ *is the Hankel operator with symbol* $M_2 = \begin{pmatrix} \overline{\chi}wB & a \\ a^t & 0 \end{pmatrix}$, $\mathcal{C}$ *is the operator* $f \to \overline{f}$, I_N *is the* $N \times N$ *identity matrix, and* $w = 1 + 2\ \mathrm{Re}\ \chi\beta$.

A proof of Proposition 4 follows from the relations $P_{H^{2\perp}_{N+1,0}}[\overline{M\eta}] = \mathcal{C}P_{H^2_{N+1}}[M\eta]$ and $\overline{\chi}P_{H^{2\perp}_{N+1,0}}[\chi M\eta] = P_{H^{2\perp}_{N+1}}[M\eta]$ applied to (16.16).

One can see in representation (16.17) that $T'_{f,\beta}$ has the form of a "Conjugate of Toeplitz plus Hankel" operator (except for a small change in the range, which turns out to be unimportant for our analysis). One consequence of this as pointed out to us by Ali Sayed is that $T'_{f,\beta}$ can be inverted numerically with fast algorithms similar to those described in **[SLK]**. This enhances the practical appeal of Newton's method applied to the operator T.

An important issue is whether or not $T'_{f,\beta}$ is invertible for almost all (f, β). This is required for Newton's method to work well. A technical point is the space on which $T'_{f,\beta}$ is defined. This certainly influences invertibility. For Newton iteration it is appropiate to define T is defined not on all of an H^2 type space but on a subspace of it consisiting of smooth functions. This causes mathematical problems in analyzing the behavior of T but not in computer implementation. See Theorem 6 in **[HMW]** for details.

16.5 Derivation of the Nehari-Commutant Lifting and other formulas

In this subsection we derive relations (16.4), (16.5), and (16.6) that appear in Theorem 21 directly from the optimality conditions in Theorem 20.

Recall that the Nehari problem is of the type OPT with performance as in (16.3). To avoid technical issues we assume that $k \notin H^\infty_N$ and that k is continuous, hence there is a unique best H^∞ approximation f_0 to k. We also assume that $k - f_0$ is bounded away from 0. Thus Γ and f_0 satisfy the hypotheses of Theorem 20.

Set $\tau = \mathrm{dist}(k, H^\infty_N)$. The optimality conditions I and II of Theorem 20 in this case say that there exists λ measurable and positive on T, and a function $F \in H^1_N$ such that

$$\begin{aligned} ||k(e^{j\theta}) - f_0(e^{j\theta})||^2_{C^N} = &\quad \tau^2 && (16.18) \\ \frac{1}{\lambda}\overline{(k - f_0)} = &\quad \chi F && (16.19) \end{aligned}$$

Let $\alpha_0 \in H_1^2$ be the outer spectral factor of $1/\lambda$. From (16.19) we have that

$$k - f_0 = \frac{1}{\alpha_0}\frac{\overline{\chi F}}{\overline{\alpha_0}} = \frac{1}{\alpha_0} P_{H_N^{2\perp}}[\frac{\overline{\chi F}}{\overline{\alpha_0}}] \tag{16.20}$$

Combining (16.19) and (16.20) one obtains

$$k - f_0 = \frac{1}{\alpha_0} P_{H_N^{2\perp}}[(k - f_0)\alpha_0] = \frac{1}{\alpha_0} P_{H_N^{2\perp}}[k\alpha_0] \tag{16.21}$$

This proves the formula (16.5).

We now check the identity (16.4). We trivially have that

$$||\mathcal{H}_k||^2 = ||\mathcal{H}_{k-f_0}||^2 \;\leq\; ||k - f_0||^2_{L_N^\infty} = \tau^2 \tag{16.22}$$

To prove the reverse inequality, note that by (16.18) the following relation holds for almost all θ:

$$\tau^2 = \frac{|\alpha_0(e^{j\theta})|^2 ||k(e^{j\theta}) - f_0(e^{j\theta})||^2_{C^N}}{|\alpha_0(e^{j\theta})|^2} = \frac{||\alpha_0(e^{j\theta})(k(e^{j\theta}) - f_0(e^{j\theta}))||^2_{C^N}}{|\alpha_0(e^{j\theta})|^2} \tag{16.23}$$

Combine (16.5) and (16.23) to obtain

$$\tau^2 = \frac{||\alpha_0(k - f_0)||^2_{L_N^2}}{||\alpha_0||^2_{L_N^2}} = \frac{||P_{H_N^{2\perp}}[\alpha_0(k - f_0)]||^2_{L_N^2}}{||\alpha_0||^2_{L_N^2}} \leq\; ||\mathcal{H}_k||^2_{\mathcal{L}} \tag{16.24}$$

and this finishes the proof of the equality (16.4).

At this point we are in position of deriving the Nehari - Commutant Lifting formula (16.6). Rewrite (16.19) as

$$\overline{\alpha_0}(\overline{k - f_0}) = (\chi/\alpha_0)\; F \tag{16.25}$$

and notice that by multiplying both sides of (16.25) by $(k - f_0)^t$ and using (16.18) one obtains

$$\tau^2\overline{\alpha_0} = (\chi/\alpha_0)\; (k - f_0)^t F \tag{16.26}$$

Now we reach a crossroads. The strategy behind our approach in Section (16.4) was to project (16.25) onto $H^{2\perp}_{N,0}$. In this case it produces a Toeplitz operator type formula, $P_{H^{2\perp}_{N,0}}[\; \overline{\alpha_0}\overline{k} - \overline{\alpha_0}\overline{f}\;] = 0$, i.e., $\mathcal{T}_{k-f}[\;\alpha\;] = 0$.

The strategy in Nehari and Commutant Lifting approaches is to project both terms in (16.25) onto $H^2_{N,0}$. This yields

$$P_{H^2_{N,0}}[\; \overline{\alpha_0}\overline{k}\;] = (\chi/\alpha_0)\; F \tag{16.27}$$

Multiplying both sides of (16.27) by $(k-f)^t$, using (16.26), and taking conjugates of both sides we have

$$\overline{(k-f)}^t P_{H_N^{2\perp}}[\ \alpha_0 k\ ] = \tau^2 \alpha_0 \tag{16.28}$$

Taking projection of both sides of (16.28) onto H_N^2 gives

$$P_{H_N^2}[\ \overline{k}^t P_{H_N^{2\perp}}[\ \alpha_0 k\ ]\ ] = \tau^2 \alpha_0 \tag{16.29}$$

Now equation (16.29) is precisely the Nehari - Commutant Lifting formula (16.6).

We see that the Nehari- Commutant Lifting formula is remarkable in that it puts together, in a single equality, both relations (16.18) and (16.19), while at the same time eliminates f and F (our approach in Section 3 eliminates τ and F). It is also notable that all this is accomplished with a formula that is *linear* in α. It is unfortunate that these properties do not hold for functions Γ that are more general than (16.3).

16.6 References

[A] Ando

[AAK1] V. M. Adamjam, D. Z. Arov and M. G. Krein. Infinite Hankel matrices and generalized problems of caratheodory-Fejer and F. Riesz, Functional Anal. Appl. 2 (1968), pp. 1-18.

[AAK2] V. M. Adamjam, D. Z. Arov and M. G. Krein. Analytic properties of Schmidt paris for Hankel operator and the generalized Scur-Takagi problem. Math. USSR-Sb. 15 (1972), pp. 15-78.

[AAK3] V. M. Adamjam, D. Z. Arov and M. G. Krein. Infinite block Hankel matrices and related extension problems, Amer. Math. Soc. Translations 111 (1978), 133-156.

[BHM] J. Bence, J. W. Helton and D. E. Marshall. "Optimization over H-infinity," Proc. Conference on Decision and Control, Athens, Greece, December 1986.

[FFHKP] C. Foias, B. Francis, J. W. Helton, H. Kwakernaak, J.B.Pearson. H^∞ -control Theory. Lecture Notes in Mathematics No. 1496. E. Mosca and L. Pandolfi, editors. Springer Verlag, Berlin. 1991.

[HMer1] A Novel Approach to Accelerating Newton's Method for sup-norm optimization arising in H^∞-control. J. Optim. Theory Apps. 1993

[HMer2] J. W. Helton and O. Merino. Conditions for Optimality over $\mathcal{H}^\infty$. SIAM J. Cont. Opt., Vol. 31, No. 6, November 1993.

[HMW] J. W. Helton, O. Merino and T. E. Walker. Algorithms for Optimizing over Analytic Functions. Indiana U. Math. J., Vol 42, No.3 (1993) .

[NF] Sz.-Nagy, B. and Foias, C. Harmonic Analysis of Operators on Hilbert Space. North Holland, 1970.

[S] Sarason, D. Generalized interpolation in H^∞. Trans. Amer. Math. Soc. vol 127, pp. 179-203. 1967.

[SKL] S??, T. Kailath and L??