Fast Fourier optimization
Sparsity matters

Robert J. Vanderbei

Received: 21 June 2011 / Accepted: 17 December 2011 / Published online: 18 January 2012
© Springer and Mathematical Optimization Society 2012

Abstract Many interesting and fundamentally practical optimization problems, ranging from optics, to signal processing, to radar and acoustics, involve constraints on the Fourier transform of a function. It is well-known that the fast Fourier transform (fft) is a recursive algorithm that can dramatically improve the efficiency for computing the discrete Fourier transform. However, because it is recursive, it is difficult to embed into a linear optimization problem. In this paper, we explain the main idea behind the fast Fourier transform and show how to adapt it in such a manner as to make it encodable as constraints in an optimization problem. We demonstrate a real-world problem from the field of high-contrast imaging. On this problem, dramatic improvements are translated to an ability to solve problems with a much finer grid of discretized points. As we shall show, in general, the “fast Fourier” version of the optimization constraints produces a larger but sparser constraint matrix and therefore one can think of the fast Fourier transform as a method of sparsifying the constraints in an optimization problem, which is usually a good thing.

Keywords Linear programming · Fourier transform · Interior-point methods · High-contrast imaging · Fast Fourier transform (fft) · Optimization · Cooley–Tukey algorithm

Mathematics Subject Classification (2000) 90C08 · 65T50 · 78A10

The author was supported by a grant from NASA.

R. J. Vanderbei
Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544, USA
e-mail: rvdb@princeton.edu
1 Fourier transforms in engineering

Many problems in engineering involve maximizing (or minimizing) a linear functional of an unknown real-valued design function $f$ subject to constraints on its Fourier transform $\hat{f}$ at certain points in transform space. Examples include antenna array synthesis (see, e.g., [11, 12, 15]), FIR filter design (see, e.g., [3, 22, 23]), and coronagraph design (see, e.g., [6–10, 13, 16, 18, 19]). If the design function $f$ can be constrained to vanish outside a compact interval $C = (-a, a)$ of the real line centered at the origin, then we can write the Fourier transform as

$$\hat{f}(\xi) = \int_{-a}^{a} e^{2\pi i x \xi} f(x) dx$$

and an optimization problem might look like

$$\begin{align*}
\text{maximize} & \int_{-a}^{a} c(x) f(x) dx \\
\text{subject to} & -\varepsilon \leq \Re \hat{f}(\xi) \leq \varepsilon, \quad \xi \in D \\
& -\varepsilon \leq \Im \hat{f}(\xi) \leq \varepsilon, \quad \xi \in D \\
& 0 \leq f(x) \leq 1, \quad x \in C,
\end{align*}$$

where $D$ is a given subset of the real line, $\varepsilon$ is a given constant, and $\Re(z)$ and $\Im(z)$ denote the real and imaginary parts of the complex number $z$. In Sect. 7, we will discuss a specific real-world problem that fits a two-dimensional version of this optimization paradigm and for which dramatic computational improvements can be made.

Problem (1) is linear but it is infinite dimensional. The first step to making a tractable problem is to discretize both sets $C$ and $D$ so that the continuous Fourier transform can be approximated by a discrete Riemann sum:

$$\hat{f}_j = \sum_{k=-n}^{n} e^{2\pi i k \Delta x j \Delta \xi} f_k \Delta x, \quad -n \leq j \leq n. \quad (2)$$

Here, $n$ denotes the level of discretization,

$$\Delta x = \frac{2a}{2n + 1},$$

$\Delta \xi$ denotes the discretization spacing in transform space, $f_k = f(k \Delta x)$, and $\hat{f}_j \approx \hat{f}(j \Delta \xi)$.

Computing the discrete approximation (2) by simply summing the terms in its definition requires on the order of $N^2$ operations, where $N = 2n + 1$ is the number of discrete points in both the function space and the transform space (later we will generalize to allow a different number of points in the discretization of $C$ and $D$).