1 Closing Remarks on Fourier Transforms

If f is a function such that the following are true:

$$\|f\|_1 = \int_{-\infty}^{\infty} |f(x)| \, dx < \infty, \qquad \|f\|_2 = \left(\int_{-\infty}^{\infty} |f(x)|^2 \, dx \right)^{\frac{1}{2}} < \infty$$

then we say f is both a member of the L^1 and L^2 spaces; $f \in L^1 \bigcap L^2$. The Fourier transform is defined for functions f in both these spaces by the formula

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

The amazing thing about the transform is that it is an *isometry*- it preserves "lengths". In this context it means that a problem can be transformed, solved in a (hopefully) easier context, and then transformed back to give us a solution.

2 Stability

Suppose a time domain $t \in [0, T]$ with some partition, a space domain with some partition, a state vector u^n that carries the information for the given partition of space and time, and the state change $u^{n+1} = C_{h,k} u^n$ where $C_{h,k}$ is the operator with *mesh parameters* h and k (h is the space partition size, k is the time partition size), we say that the process is *stable* if $\|C_{h,k}^n\|$ is bounded independently of n (the norm here is the Operator norm: $\|A\| = \sup\limits_{\|v\|=1} \|Av\|$). In fact, various norms dominate the operator norm: the elementwise matrix norm, for example, dominates the operator norm. Thus if we can show the elementwise norm of the matrix is bounded independent of n, the operator norm must therefore be bounded.

The next class of stability is *Von Neumann stable* or for short *Neumann stable.* A method is *Von Neumann stable* if $\dfrac{\|u^{n+1}\|}{\|u^n\|} \leq 1$. It can be shown that *stable* $\implies$ *Von Neumann stable*- replace the u^{n+1} with $C_{h,k} u^n$ in the numerator and apply Cauchy-Schwarz. The converse is true in the case that $C_{h,k}$ is a symmetric matrix.

The *spectral radius* of a matrix A is $\rho(A) = \max(|\lambda|)$ for all eigenvalues λ of A. In the case that A is symmetric, the spectral radius $\rho(A)$ is exactly the operator norm $\|A\|$. In the case that A is not symmetric, $\rho(A)$ is a lower bound for $\|A\|$. Thus in any particular method we can examine the eigenvalues of $C_{h,k}$ along with if its symmetric to attempt to determine stability.

3 Discrete Fourier Transform

Let $[x_0, x_1, ..., x_N]$ be a uniform partition of $[0, 2\pi]$ (hence $x_i = i\dfrac{2\pi}{N}$), let $f, g : [x_0, x_1, ..., x_{N+1}] \to \mathbb{C}$ be functions mapping the partition to $\mathbb{C}$ (we can think of them as n-tuples of complex numbers), then $\sigma(f, g) = \displaystyle\sum_{0}^{N-1} f(x_i)\overline{g(x_i)}$ defines a positive definite Hermitian form.

Lemma 3.1. $\sigma(e^{ijx}, e^{ikx}) = \begin{cases} N & \textit{if } \dfrac{j-k}{N} \in \mathbb{Z} \\ 0 & \textit{otherwise} \end{cases}$

Proof. For clarity let $\mathbb{I}$ be the imaginary unit. $\sigma(e^{\mathbb{I}jx}, e^{\mathbb{I}kx}) = \displaystyle\sum_{i=0}^{N-1} e^{\mathbb{I}jx_i}e^{-\mathbb{I}kx_i} = \sum_i e^{\mathbb{I}(j-k)x_i}$. Rewrite it using $x_i = i\dfrac{2\pi}{N}$ (thanks to the uniform partition) to arrive at $\displaystyle\sum_i \left[\left(e^{2\pi\mathbb{I}}\right)^{\frac{j-k}{N}}\right]^i$. Applying the identity $e^{2\pi\mathbb{I}} = 1$ gives us $\displaystyle\sum_i \left(1^{\frac{j-k}{N}}\right)^i$. If $\dfrac{j-k}{N}$ were an integer, then we would simply have $\displaystyle\sum_i 1$ which is N. If not, then we would have *roots of unity*- we already know, however, that summing all of the Nth roots of unity yields 0. $\square$

Thus the Hermitian form σ has an orthogonal set of elements $\{e^{ikx}\}$ for a certain set of k. We will see in a second how to properly choose the numbers k; we will take advantage of this orthogonal property to create the *discrete Fourier transform*:

$$\sum_{k=-M_0}^{M_1-1} C_k e^{ikx}$$

$M_0 = M_1 = \dfrac{N}{2}$ if N is even, $M_0 = M_1 = \dfrac{N-1}{2}$ if N is odd, $C_k = \dfrac{\sigma(f, e^{ikx})}{N}$. Interpret this as projecting f onto the e^{ikx} "vector"; since all of the e^{ikx} are orthogonal, we are decomposing f into different scaled components of the form e^{ikx} (similarly to how we turn an arbitrary vector in R^n into the coordinate n-tuple $\{x_1, x_2, ..., x_n\}$).

In a geometric sense this is akin to using trapezoids as approximations to f and then applying the transform.

Parseval's theorem holds in the discrete case: if f is given by $(f(x_0), f(x_2), ..., f(x_N)$ *samples* of f at x_i, then $\|f\| = \|f\|_\sigma = \left[\sigma\left(f, f\right)^{\frac{1}{2}}\right]$. Note that this suggests an application of the discrete Fourier transform: recall that Neumann stability is given by $\|C_i^{n+1}\| \leq \|C_i^n\|$. Applying Parseval's theorem suggests that we can apply a transformation and use the $\|\cdot\|_\sigma$ norm to make conclusions about stability.

4 Truncation & the Lax Equivalence Theorem

The fundamental problem with any approach for solving something numerically is quite intutive: how do we know if we are approaching the "right answer" without actually knowing the right answer? Any numerical method will give us a sequence of progressively improving estimates; how do we know that the sequence converges, and even more important, that it converges to the true solution? For the finite difference method, we have the following powerful result:

Let $C^n_{h,k}$ represent our finite difference method operator. Start with the error $e_n = u(t_n, \cdot) - Cu^{n-1}$; where $u(t_n, \cdot)$ is our stand-in for the "actual" answer at time t_n and with $\cdot$ representing various other parameters; let Cu^{n-1} represent the answer our numerical method gives us given the state vector u^{n-1} one time-step prior to t_n.

$$\begin{aligned} e_n &= u(t_n, \cdot) - Cu(t_{n-1}, \cdot) + Cu(t_{n-1}, \cdot) - Cu^{n-1} \\ &= u(t_n, \cdot) - Cu(t_{n-1}, \dot) + C\left(u(t_{n-1}, \cdot) - u^{n-1}\right) \end{aligned}$$

The left two terms, $u(t_n, \cdot) - Cu(t_{n-}, \cdot)$, reference only the "true" solution and how the numerical method C is "off" by when given the correct solution as input. We call this term *truncation error*; it is the error induced solely by C, and not by anything else (importantly, it is independent of t_n or any other parameters!).

The right two terms are rewritten as $C\left(e_{n-1}\right)$. So our original error, e_n, is broken down as follows:

$$e_n = \text{truncation error} + Ce_{n-1} = \text{truncation} + C(\text{truncation} + Ce_{n-2})$$

Note that this recurrence relation will therefore apply all the way down; Ce_{n-1} is similarly broken down into truncation error and C^2e_{n-2}, and so on. The term e_n can be thought of as fully breaking down into two parts: a geometric sum truncation $\cdot\, (1 + C + ... + C^{n-1}$ and our numerical error $C^n e_0$ If $\|C^n\|$ is bounded, then we have that the $C^n e_0$ term will go to 0 as we let n go to ∞. On the other hand, for the geometric sum component to go to 0 we have the requirement that the truncation term goes to zero as we improve our approximations (in other words, if h and k go to zero, truncation should go to zero). This requirement is in fact known as *consistency*; it means that a numerical method approximates the partial differential equation correctly. Thus, in order to have e_n go to zero as n goes to ∞, it is required that C both be bounded and consistent. Recall that C being bounded for all n is equivalent to C being stable. This is the Lax-Richtmyer theorem, also known as the Lax Equivalence theorem (named after Peter Lax and Robert D. Richtmyer).

Theorem 4.1. *(Lax Equivalence Theorem)*
If C is a consistent finite difference method, then C converges if and only if C is stable.

#