Supplemental material for compressed representation of Kohn-Sham orbitals via selected columns of the density matrix

Anil Damle,[1] Lin Lin,[2,3] and Lexing Ying[4,1]

[1]Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305
[2]Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720
[3]Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720
[4]Department of Mathematics, Stanford University, Stanford, CA 94305

We demonstrate the main idea behind Hockney's algorithm [1, 2] for solving Poisson's equation for functions with local support. For clarity we present the algorithm in one dimension, and its generalization to a three-dimensional calculation is straightforward. Here we solve

$$-u''(x) = f(x) \tag{1}$$

on a one dimensional interval $[0, 1]$ with periodic boundary condition. We assume that f has zero mean value, the support of f is localized on a small interval of the computational domain, and we are only interested in the solution u on the support of f. The domain is discretized via the equally spaced set of points $x[j] = j/(N + 1)$ for $j = 0, \ldots, N$ and that f is only nonzero for $0 \leq j \leq M - 1$. We also assume that $M < N/2$, otherwise it would be more efficient to simply solve the problem on the entire domain.

Solving this discrete version of Poisson's equation in the entire domain may be achieved via convolving f with a known kernel. Conceptually, one may view the kernel as the solution to Eq. (1) where the right hand side takes the value one for $j = 0$ and zero otherwise. In fact, the Discrete Fourier Transform (DFT) of this kernel, given the chosen discretization, is known exactly so it need not be computed. Let us denote the kernel in real space as g.

Numerically solving this problem amounts to computing the periodic discrete convolution of f with g. If we let $f[j] = f(x_j)$ we may write the solution on the grid as

$$u[j] = \sum_{i=0}^{N} f[i]g[j - i] \tag{2}$$

where we assume f and g are periodic such that they are well defined for any subscript. If we needed $u[j]$ for $j = 0, \ldots, N$ we would use the relationship between the DFT and periodic convolution coupled with the computational efficiency of the FFT to compute u rapidly. However, here we only need to compute $u[j]$ for $j = 0, \ldots, M - 1$. Consequently we may write Eq. (2) as

$$u[j] = \sum_{i=0}^{M-1} f[i]g[j - i], \tag{3}$$

which implies that the only values of g that we need are $g[-M + 1], \ldots, g[M - 1]$. This observation allows us to write $u[j]$ for $j = 0, \ldots, M-1$ as the cyclic convolution of two vectors of length $2M$. If we let $\tilde{f}[j] = f[j]$ for $j = 0, \ldots, 2M-1$ and, using the periodicity of g,

$$\tilde{g}[j] = \begin{cases} g[j], & j = 0, \ldots, M - 1 \\ g[N - 2M + 1 + j], & j = M, \ldots, 2M - 1 \end{cases} \tag{4}$$

we may write u_j as

$$u[j] = \sum_{i=0}^{2M-1} \tilde{f}[i]\tilde{g}[j - i]. \tag{5}$$

For $j = 0, \ldots, M-1$ this formula yields the exact same values as in Eq. (2) and therefore we may compute the desired values of u via FFTs of length $2M$ rather than N. We could have defined $\tilde{f}$ and $\tilde{g}$ to be longer and still achieve the same result. In our situation we use this to avoid the additional cost of building several different instances of $\tilde{g}$ by fixing the size of the small convolutions we use. We may also choose M to be dyadic, which is beneficial given the use of the FFT. In practice, if the problem size N grows, but M remains fixed, then the cost of numerically solving Poisson's equation at the points of interest remains constant and independent of N.

As an example, we solve the one dimensional Poisson's equation on $[0, 1]$, for $f(x)$ localized in the interval $[0, 0.2]$. Fig. 1 (a) shows $f(x)$, and Fig. 1 (b) shows the solution using full FFT on $[0, 1]$, and using Hockney's algorithm on the double-sized interval $[0, 0.4]$. We observe that the solution provided by Hockney's algorithm fully agrees with that obtained from the full FFT on the interval of interest $[0, 0.2]$, but the two solutions differ in the buffer interval $[0.2, 0.4]$. The computational cost of Hockney's algorithm is strictly smaller due to the reduced size of the FFT required.

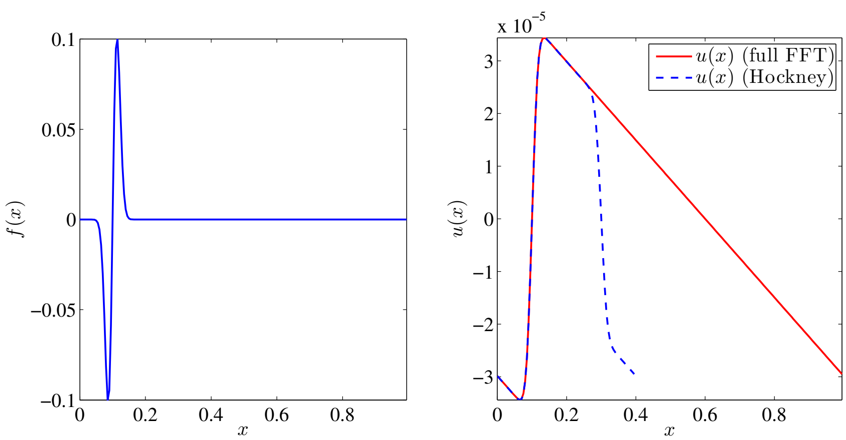

FIG. 1: (a) $f(x)$ for Poisson's equation with support in the interval $[0, 0.2]$. (b) Solution $u(x)$ obtained from Hockney's algorithm compared with the exact solution using FFT on the global domain $[0, 1]$.

[1] R. W. Hockney, J. ACM **12**, 95 (1965).
[2] J. W. Eastwood and D. R. K. Brownrigg, J. Comput. Phys. **32**, 24 (1979).