We see that if exact representations of the delta function is used then the error would be zero; however, we must used a discretized version of the delta function. For this analysis the discrete delta function is,

$$\delta_h(x) = \begin{cases} 1 - \frac{|x|}{h} & 0 \leq \frac{|x|}{h} \leq 1 \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

which if $x_i = x_k$ recovers $e(x) = 0$. Now, to simulate the stretching, we want $x = x_i$ and to evaluate the above. We see the first term turns into evaluating the the distribution function at the three different points. The second part isn't as clear but we see that we have to actually do the sum. This is done in the *Mathematica* and is in an accompanying section.

III. Debugging PIC without Remapping

A. W_2

For the second-order algorithm the deposition steps require the use of W_2 which is defined as

$$W_2(y) = \begin{cases} 1 - |y|, & 0 \leq |y| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

We check that our W_2 function is written correctly by simply plotting it on the interval $[-3, 3]$. This plot is shown in Figure 1.

B. Particle Deposition

Following the paper, the algorithm for the particle deposition step can be written as

$$\rho_i = \sum_p \frac{q_p}{V_i} W_2\left(\frac{x_i - x_p}{\Delta x}\right)$$

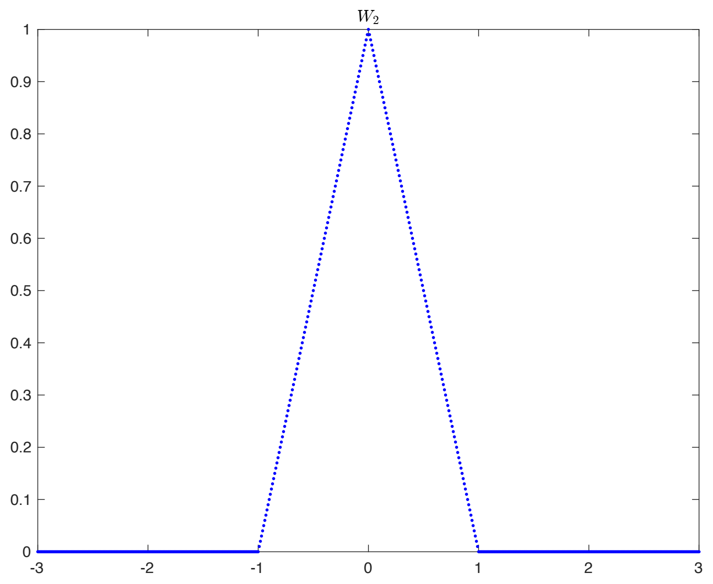

FIG. 1. Plot of W_2 over the interval $[-3, 3]$.

where $V_i = \Delta x$ is the volume of the cell. A simple test of this was done by defining the poisson grid to be $[0, 1/4, 1/2, 3/4]$ and the particles defined as

$$(q_p, x_p) = \begin{pmatrix} 1 & \frac{1}{16} \\ 1 & \frac{3}{16} \\ 1 & \frac{5}{16} \\ 1 & \frac{7}{16} \\ 1 & \frac{9}{16} \\ 1 & \frac{11}{16} \\ 1 & \frac{13}{16} \\ 1 & \frac{15}{16} \end{pmatrix}.$$

Since the charge distribution is uniform, we expect the density to be uniform as well and hence to compare analytically we only need to check the density at one of the poisson grid points. We choose $x_i = 1/4$ and see that then

$$\begin{aligned}\rho &= \frac{1}{1/4}\left(W_2\left(\frac{1/4 - 1/16}{1/4}\right) + W_2\left(\frac{1/4 - 3/16}{1/4}\right) + W_2\left(\frac{1/4 - 5/16}{1/4}\right) + W_2\left(\frac{1/4 - 7/16}{1/4}\right)\right) \\ &= 4(\frac{1}{4} + \frac{3}{4} + \frac{3}{4} + \frac{1}{4}) \\ &= 4(2)\end{aligned}$$

$= 8.$

So we expect the density at every poisson grid point to be 8. The results of running the algorithm on this data is shown below in Figure 2.

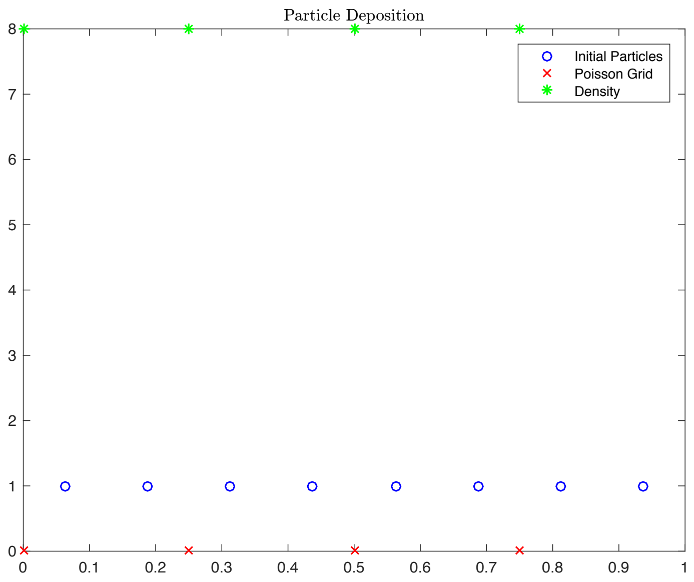

FIG. 2. Plot of results for particle deposition.

C. Field Solve

For the field solve we use a finite difference method on the poisson grid with periodic boundary conditions. The resulting linear system is

$$A\phi = -\rho$$

with

$$A = \frac{1}{\Delta x^2}\begin{pmatrix} -2 & 1 & 0 & 0 & \dots & 1 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \dots & \vdots \\ 0 & 0 & \dots & 1 & -2 & 1 \\ 1 & 0 & 0 & \dots & 1 & -2 \end{pmatrix}.$$

We see that this is singular and that any solution to

$$\Delta\phi = -\rho$$

with periodic boundary conditions can be changed by a constant. To make the solution unique we pick the solution that satisfies

$$\sum_i \phi_i \Delta x = 0$$

and solve the rectangular system with

$$\hat{A}\phi = -\hat{\rho} \tag{18}$$

$$\hat{A} = \frac{1}{\Delta x^2}\begin{pmatrix} -2 & 1 & 0 & 0 & \dots & 1 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \dots & \vdots \\ 0 & 0 & \dots & 1 & -2 & 1 \\ 1 & 0 & 0 & \dots & 1 & -2 \\ \Delta x^3 & \Delta x^3 & \Delta x^3 & \dots & \Delta x^3 & \Delta x^3 \end{pmatrix} \tag{19}$$

$$\hat{\rho} = \begin{pmatrix} \rho \\ 0 \end{pmatrix} \tag{20}$$

using MATLABs mldivide solver.

To test this we let

$$\rho = \left(\frac{2\pi}{L}\right)^2 \sin\left(\frac{2\pi x}{L}\right)$$

which has the exact solution

$$\phi = \sin\left(\frac{2\pi x}{L}\right)$$

where we have enforced the 0 average condition. We do convergence study and get a slope of 2.0304 as we would expect. The convergence plot with a sample solution with 16 grid points is shown in Figure 3.

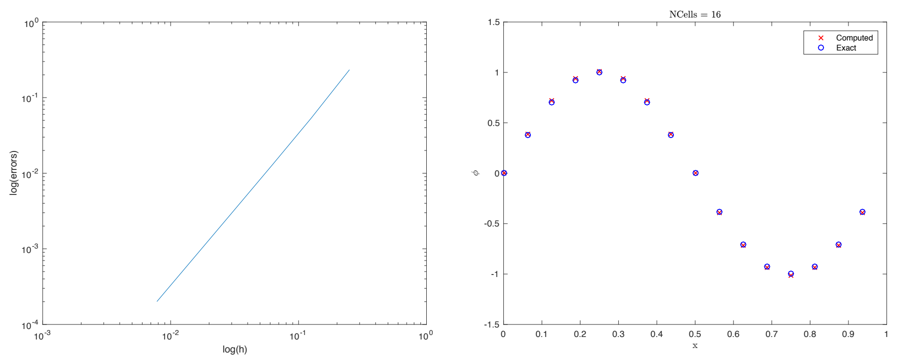

FIG. 3. Plot of poisson solver convergence and sample solution. We get a convergence rate of 2.0304.

D. Gradient

To get the electric field from the potential, we need to compute the gradient of the potential. We again use a centered difference approximation of the gradient

$$B = \frac{1}{2\Delta x}\begin{pmatrix} 0 & 1 & 0 & 0 & \dots & -1 \\ -1 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \dots & \vdots \\ 0 & 0 & \dots & -1 & 0 & 1 \\ 1 & 0 & 0 & \dots & -1 & 0 \end{pmatrix}$$

and apply this to the ϕ computed above to get

$$E = -B\phi.$$

Following the above, we let

$$\phi = \sin\left(\frac{2\pi x}{L}\right)$$

and thus have an exact solution of E to be

$$E = -\frac{2\pi}{L}\cos\left(\frac{2\pi x}{L}\right).$$

To check the above we again check the convergence and get a convergence rate of 1.9705. The convergence plot and a sample plot are shown in Figure 4.

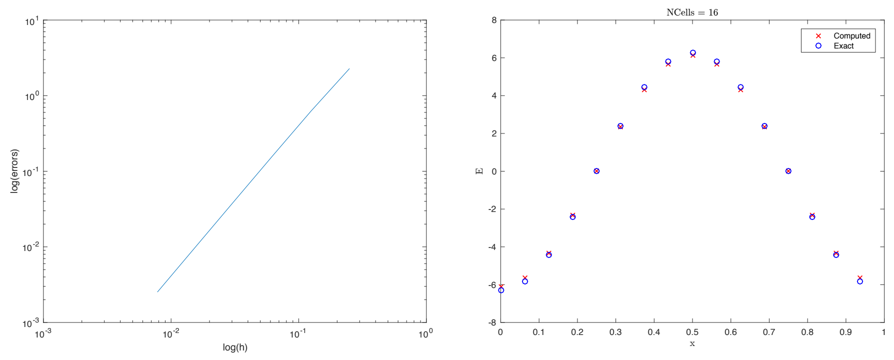

FIG. 4. Plot of gradient convergence and sample solution. We get a convergence rate of 1.9705.

E. Force Interpolation

Following the paper, the algorithm for the force interpolation step can be written as

$$E_p = \sum_i E_i W_2 \left(\frac{x_i - x_p}{\Delta x} \right)$$

where we note the lack of a volume factor. A simple test of this can be done by defining the electric field on the poisson grid as to be

$$(E_i, x_i) = \begin{pmatrix} 8 & 0 \\ 8 & \frac{1}{4} \\ 8 & \frac{1}{2} \\ 8 & \frac{3}{4} \end{pmatrix}$$

and the particles positions defined as

$$x_p = \begin{pmatrix} \frac{1}{16} \\ \frac{3}{16} \\ \frac{5}{16} \\ \frac{7}{16} \\ \frac{9}{16} \\ \frac{11}{16} \\ \frac{13}{16} \\ \frac{15}{16} \end{pmatrix}.$$

Since the field is uniform, we expect the interpolated field to be uniform as well and hence to compare analytically we only need to check the field at one particle.

$$\begin{aligned}\rho &= 8\left(W_2\left(\frac{1/4-1/16}{1/4}\right)+W_2\left(\frac{0-1/16}{1/4}\right)\right)\\ &= 8(\frac{3}{4}+\frac{1}{4})\\ &= 8(1)\\ &= 8.\end{aligned}$$

So we expect the field at every particle to be 8. This is what we expect since the field is a constant. The results of running the algorithm on this data is shown below in Figure 5.

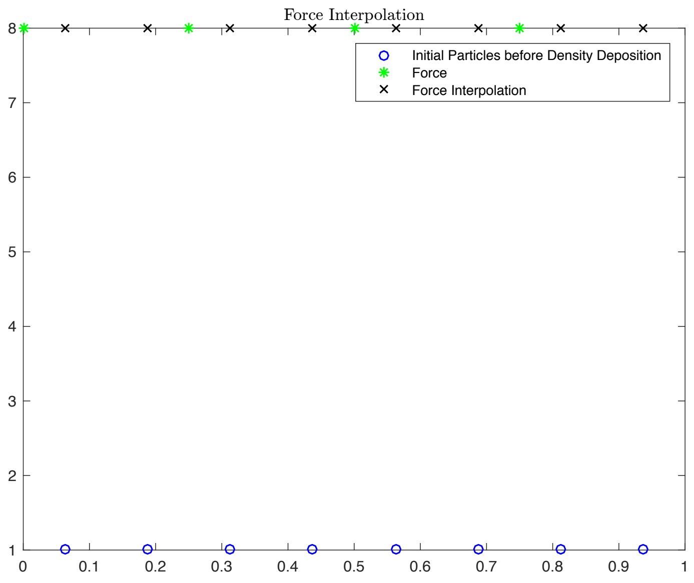

FIG. 5. Plot of results for field interpolation.

F. Particle Push

I haven't figured out a great way to test the Particle Push stage; however, I did look at Andrew's bitbucket and some of the code. One thing I noticed and am sort of wondering about is whether the algorithm is supposed to be

$$\begin{aligned}x^{n+1} &= x^n + v^n\Delta t - k_1\frac{\Delta t^2}{2}\\ v^{n+1} &= v^n - (k_1+k_2)\frac{\Delta t}{2}\end{aligned}$$

$$k_1 = E(x^n)$$
$$k_2 = E(x^n + \Delta t v^n).$$

To compute these k's we use the set of functions that we have checked above. The combination of these is shown in the following Matlab code:

If we input our initial conditions for our poisson solver test we should get out that $k1$ is the same as the exact solution for the gradient test - which is what Figure 6 up to some scaling.

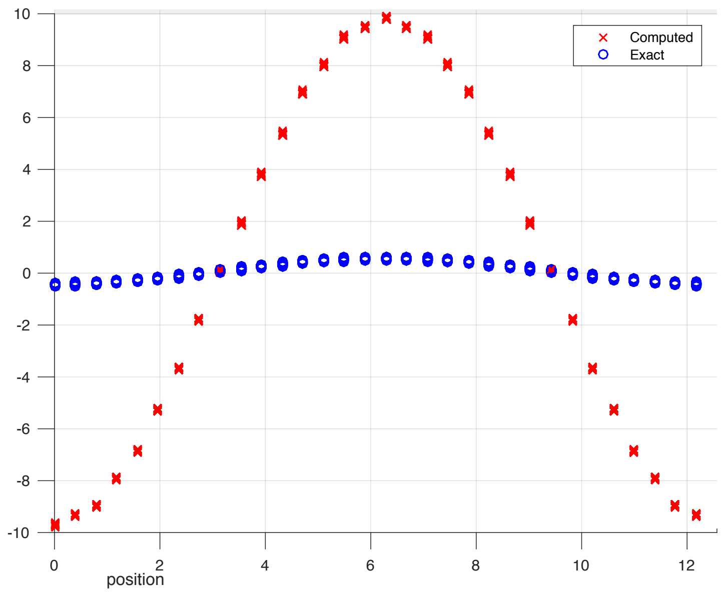

FIG. 6. Plot of k_generator test. We see that the shape is correct up to some scaling that I am not completely sure about at the moment.

G. Linear Landau Damping

To confirm that the code is properly implemented we initialize the particles with

$$q_p = h_x h_v f(x_p, v_p)$$

where

$$f(x, v) = \frac{1}{2\pi} e^{-v^2/2} \left(1 + \alpha \cos(kx)\right).$$

Analytic results show that the amplitude of the electric field is damped with a rate $\gamma = 0.1533$ and oscillates with a frequency $\omega = 1.416$. In Figure 7 we have the computational results plotted with the analytic damping rate. We see that we have good agreement in the damping rate up to time $t \sim 20$.

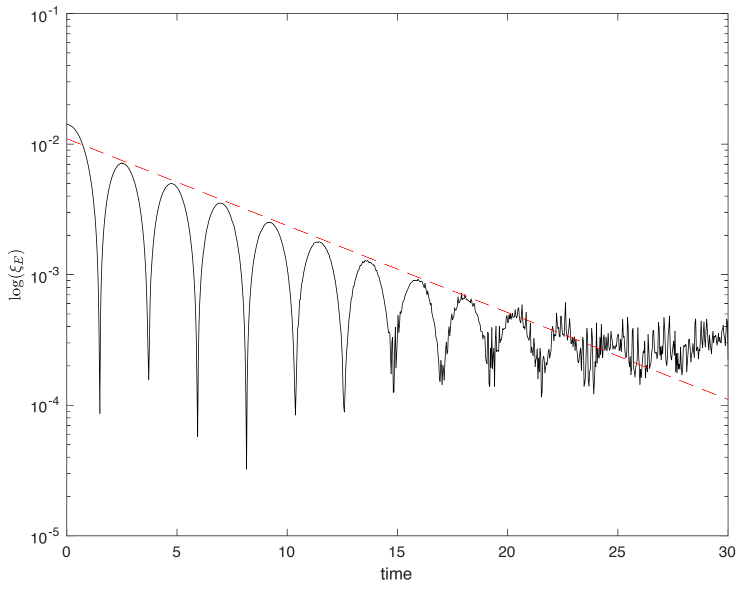

FIG. 7. Semi-log plot of Linear Landau damping against the analytic damping of $0.011exp(0.1533t)$. We see good agreement in the damping rate to time $t \sim 20$.

To check that the y-intercept is correct I ran this with all the grid and time step parameters at half the resolution. Figure 8 shows that the y-intercept does not change with the grid size and also shows the improvement with refinement.

One thing to keep in mind is the recurrence time. This time does scale with the grid that one uses and, according to what I have read, is a numerical artifact that is caused by filamentation of the plasma. Note, this recurrence time can be found by

$$T_{\text{rec}} = \frac{2\pi}{k\Delta v} = \frac{L}{\Delta v}.$$

IV. Debugging with Remapping

A. Debugging Remapping without Positivity Preservation

To debug the case with remapping we continue to use the Linear Landau Damping as a test problem. We first implemented the case without the positivity preservation algorithm since it is

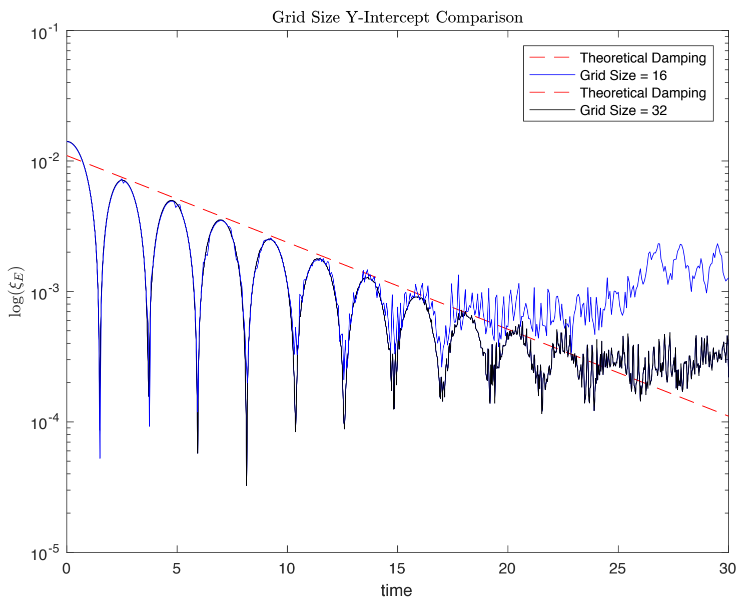

FIG. 8. Semi-log plot of Linear Landau damping against the analytic damping of $0.011exp(0.1533t)$. This plot has two different grid sizes, 16 and 32. We see that for both of these cases the y-intercept is unchanged.

mentioned that it doesn't have a large effect on the problem at this level of detail (it does show up in the order of convergence of the method). We see in Figure 9 remapping every 5th time step drastically improves the performance of the method. This shows that the remapping step appears

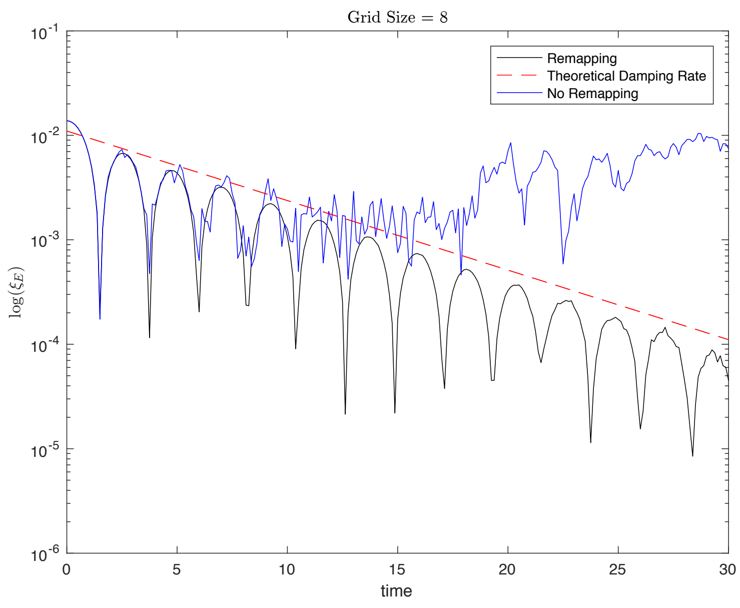

FIG. 9. Semi-log plot comparison of the remapping vs no remapping with a grid size of 8.

to be working.

$$N_{cells} = 64$$
$$N_x = 128$$
$$N_v = 256,$$

so that the runs are comparable to those that Andrew did in [7] on page B478. Note, that the total number of remaps was 23 and that the solution with only second order solvers looks like it has quality that is somewhere between remapping every 50 steps or every 5 steps. For 50-step remapping, this means there is a factor of 3 savings on the number of remaps and for 5-step remapping a savings of 33. The pattern of the remaps seems to imply that it is important to remap at the beginning of the simulation but at later times the number of remaps are not as important.

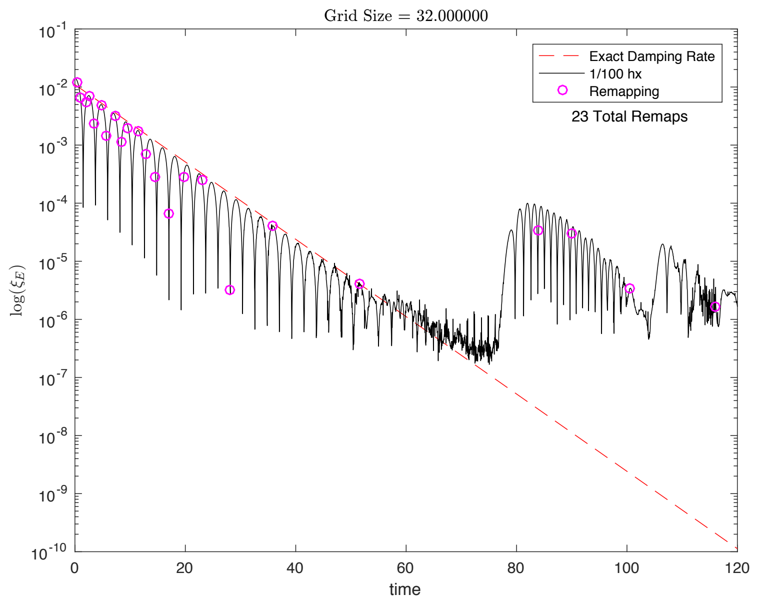

FIG. 23. Plot of Linear Landau Damping where we have set $\epsilon = \frac{1}{100}h_x$. We see good agreement with [7] and note that this was done with a second order method as well as a savings of between a factor of 3 and 33 remaps.

VII. PIC 1D Stretching and Remapping

A. Theory for set up

We want to examine the effects that stretching has on our ability to remap and determine the order of convergence compared to the size of the stretching. This is an analytic test of how much stretching is allowed to achieve a certain order of convergence. To do this we consider a 1D particle

[1] Christopher Anderson and Claude Greengard. On vortex methods. *SIAM journal on numerical analysis*, 22(3):413–440, 1985.

[2] G-H Cottet and P-A Raviart. Particle methods for the one-dimensional vlasov–poisson equations. *SIAM journal on numerical analysis*, 21(1):52–76, 1984.

[3] Georges-Henri Cottet and Petros D Koumoutsakos. *Vortex methods: theory and practice*. Cambridge university press, 2000.

[4] Gerald B Folland. *Real analysis: modern techniques and their applications*. John Wiley & Sons, 2013.

[5] Warren P Johnson. The curious history of faà di bruno's formula. *American Mathematical Monthly*, pages 217–234, 2002.

[6] Boris Lo, Victor Minden, and Phillip Colella. A real-space greens function method for the numerical solution of maxwells equations. *Communications in Applied Mathematics and Computational Science*, 11(2):143–170, 2016.

[7] Andrew Myers, Phillip Colella, and B Van Straalen. A 4th-order particle-in-cell method with phase-space remapping for the vlasov–poisson equation. *SIAM Journal on Scientific Computing*, 39(3):B467–B485, 2017.

[8] HD Victory, Jr and Edward J Allen. The convergence theory of particle-in-cell methods for multidimensional vlasov–poisson systems. *SIAM journal on numerical analysis*, 28(5):1207–1241, 1991.

[9] Bei Wang. *A Particle-in-cell Method with Adaptive Phase-space Remapping for Kinetic Plasmas*. PhD thesis, University of California at Davis, 2011.

[10] Bei Wang, Gregory H Miller, and Phillip Colella. A particle-in-cell method with adaptive phase-space remapping for kinetic plasmas. *SIAM Journal on Scientific Computing*, 33(6):3509–3537, 2011.