

Table of Quadrature Results for Regularized Stokeslets

We are solving for the fluid flow around a steadily translating cylinder of radius $a = 0.25$ in an infinite fluid with viscosity $\mu = 1.0$. We use the 2D steady Stokes equations $\mu\Delta\mathbf{u}(\mathbf{x}) = -\nabla p$ with boundary conditions $\mathbf{u} = [1, 0]$ on the circle to approximate the solution near the cylinder. See Cortez (2001) for more details and the formula for the exact solution to the test case. We use a regularization method on the kernel of the integral equation combined with various boundary element methods (BEMs) to solve the problem numerically.

Table 1 lists the errors for a single point slightly outside of the cylinder: $0.26 * (\cos(\pi/6), \sin(\pi/6))$. We discretize the boundary circle into N nodes and represent the circle as an N -polygon. The numerical methods examined are the original method by Cortez (2001) (called “Riemann Sum” in the table), and either hat, constant, or blob basis functions paired with either the trapezoidal rule or Simpson’s rule for numerical integration along each chord comprising the polygon. Quadrature is performed with M points along each chord. In Table 1, $M = 1000$ excluding the “Riemann Sum” column, where $M = 1$ to be equivalent with the original model. Each chord is length $\ell = 2 \sin(\pi/N)$, and the quadrature spacing is ℓ/M . The blob parameter for the regularization is $\ell/4$ in all trials in Table 1, so it is decreasing as chord length decreases.

N	Riemann Sum	Blob, Trap	Blob, Simp	Hat, Trap	Hat, Simp	Cst, Trap	Cst, Simp
4	2.5103	0.0435	0.0435	0.0516	0.0516	4.2663	4.2664
8	0.0280	0.0130	0.0130	0.0065	0.0065	0.0099	0.0099
16	0.0189	0.0117	0.0117	0.0071	0.0071	0.0072	0.0072
32	0.0097	0.0072	0.0072	0.0055	0.0055	0.0055	0.0055
64	0.0049	0.0037	0.0037	0.0035	0.0035	0.0035	0.0035
128	0.0024	0.0020	0.0020	0.0019	0.0019	0.0019	0.0019
N	Err/ ℓ	Err/ ℓ	Err/ ℓ	Err/ ℓ	Err/ ℓ	Err/ ℓ	Err/ ℓ
4	1.7750e+00	3.0794e-02	3.0794e-02	3.6498e-02	3.6498e-02	3.0168e+00	3.0168e+00
8	3.6605e-02	1.6934e-02	1.6934e-02	8.5116e-03	8.5116e-03	1.2994e-02	1.2994e-02
16	4.8429e-02	3.0111e-02	3.0111e-02	1.8181e-02	1.8181e-02	1.8330e-02	1.8330e-02
32	4.9702e-02	3.6751e-02	3.6751e-02	2.8035e-02	2.8035e-02	2.8027e-02	2.8027e-02
64	5.0054e-02	3.7845e-02	3.7845e-02	3.5383e-02	3.5383e-02	3.5377e-02	3.5377e-02
128	4.7881e-02	4.0232e-02	4.0232e-02	3.8575e-02	3.8575e-02	3.8575e-02	3.8575e-02

Table 1: Errors in BEM on the regularized Stokes cylinder example using N chords. The “Riemann Sum” column corresponds to the original method introduced by Cortez (2001). the other trials are boundary element methods using either hat or constant functions as a basis and using either the trapezoid rule or Simpson’s rule to do quadrature over each chord with 1000 points per chord. The values in the upper half of the table are the errors for the point $0.26 * (\cos(\pi/6), \sin(\pi/6))$. The values in the lower half of the table are the errors normalized by the chord length.

There are three things to note about Table 1. First, it can be seen that it doesn’t matter very much which BEM is used with the regularization. In fact, even the original method has comparable accuracy for $N = 128$, implying that using 1000 quadrature points is major over-kill. Second, all the methods are approximately $O(\ell)$, as seen from the lower half of the table, regardless of numerical integration method. Higher order integration methods might allow us to use fewer points per chord to achieve the same level of accuracy, but – as will be seen below – even with only 10 integration points per chord reasonable accuracy can be achieved with the trapezoid rule. Last, using the blob function itself as a basis does not give very good accuracy, even though in some sense it is a natural choice. This is maybe because it is not exactly 0 outside of a single element.

Smith (2009) claims that using a BEM 1) allows the use of fewer points to get comparable accuracy, and 2) weakens the coupling between the blob parameter ϵ and the discretization mesh parameter choices. The first claim appears to be supported by Table 1, although to truly save computation time, a smaller number of integration points should be used. However, I am not sure that the second claim has much impact. I found that the blob parameter is strongly coupled to the number of chords chosen, N (effectively the discretization of the forces around the boundary), but is maybe decoupled from the number of integration points per chord, M . I did a log 10 space exploration of 100 blob parameters from $1.5\text{e-}4$ to $3.0\text{e-}2$ and found error minima for $N = 8, 16, 32, 64, 128, 256$. These calculations all used hat basis functions and a trapezoid integration rule with $M = 10$. None of the minimal errors occurred at the endpoints of the range. The blob parameters associated with these minimal errors are log-log plotted in Figure 1 against chord length, showing an approximate power law relationship of $3/2$. Table 2 gives the blob parameters themselves.

N	8	16	32	64	128	256
ℓ	7.6537e-01	3.9018e-01	1.9603e-01	9.8135e-02	4.9082e-02	2.4543e-02
“best” ϵ	2.5550e-02	5.4120e-03	1.7590e-03	7.4709e-04	3.3476e-04	1.5825e-04

Table 2: For each N and associated chord length ℓ , ϵ is the blob parameter resulting in the minimal error on the test case. This minimum is based on a search over 100 values of ϵ evenly distributed in log-10 space between $1.5\text{e-}4$ and $3.0\text{e-}2$, using hat basis functions and the trapezoid rule with $M = 10$.

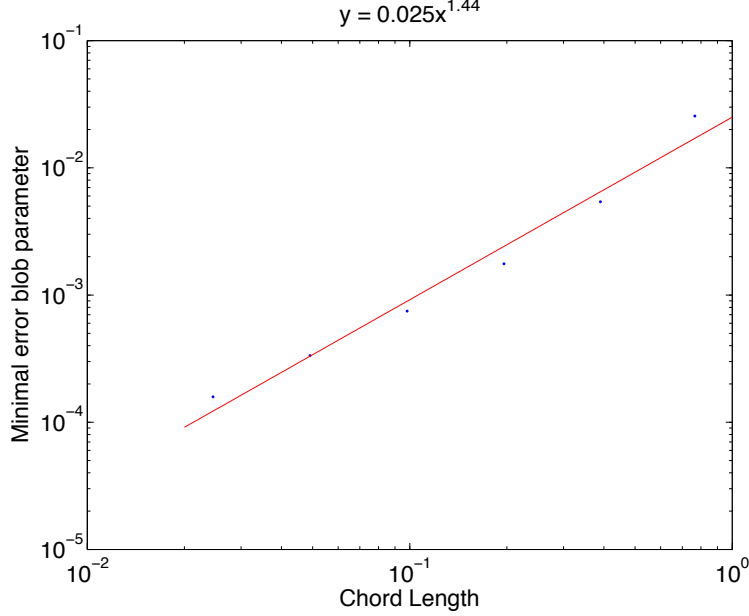


Figure 1: Log-log plot of the minimal error blob parameter versus chord length (dots). The line is the best fit power law. The dots correspond (from left to right) to $N = 256, 128, 64, 32, 16, 8$.

For the blob parameters in Table 2, I calculated the error for M varying between 10 and 250 (Table 3). The errors for $N \leq 32$ monotonically decrease with M , as one would expect in numerical integration. For higher numbers of chords, the minimal error occurs at $M = 20$ and the asymptotic error is higher. I am not sure what to make of this fact, but the differences between $M = 20$ and $M > 20$ are on order of $1.\text{e-}6$, so this might be some kind of negligible numerical artifact.

At this point, I am willing to tentatively agree with Smith (2009) that the number of integration points is not coupled to ϵ . However, the number of chords *is* coupled to ϵ , and both of sets of points comprise the discretization mesh. I suppose he is right in saying that the coupling is weakened, but I am not sure how much we actually gain from this weakening. If the blob parameter corresponds to a physical quantity, then the number of chords should be calculated from the choice of blob parameter, based on some figure like Figure 1, and the number of integration points M should be chosen to reduce the integration error to a level below the regularization error. Comparing the Riemann sum errors to the BEM errors in Table 1, it is unclear that we gain that much from this procedure in this example, except at very low chord numbers.

N	$M = 10$	$M = 20$	$M = 30$	$M = 40$	$M = 50$	$M = 100$	$M = 150$	$M = 200$	$M = 250$
8	2.0385e-03	2.0050e-03	1.9990e-03	1.9970e-03	1.9960e-03	1.9947e-03	1.9945e-03	1.9944e-03	1.9943e-03
16	1.0802e-03	8.6211e-04	8.5578e-04	8.5495e-04	8.5461e-04	8.5417e-04	8.5409e-04	8.5406e-04	8.5404e-04
32	5.9662e-04	4.0789e-04	4.0165e-04	4.0086e-04	4.0064e-04	4.0041e-04	4.0037e-04	4.0036e-04	4.0035e-04
64	3.5328e-04	2.7316e-04	2.7381e-04	2.7386e-04	2.7383e-04	2.7378e-04	2.7377e-04	2.7376e-04	2.7376e-04
128	1.9975e-04	1.6297e-04	1.6485e-04	1.6514e-04	1.6517e-04	1.6516e-04	1.6516e-04	1.6516e-04	1.6516e-04
256	1.0707e-04	8.7104e-05	8.8272e-05	8.8477e-05	8.8507e-05	8.8510e-05	8.8509e-05	8.8509e-05	8.8509e-05

Table 3: Errors in BEM on the regularized Stokes cylinder example using hat basis functions with the trapezoid rule. The blob parameter is the same as in Figure 1 (blue dots). M is the number of integration points per chord; the number of chords is N . The values in the table are the errors for the point $0.26 * (\cos(\pi/6), \sin(\pi/6))$.

Since I have only looked at the error at a single point, further validation is required to make sure that the numerical implementation is working properly. Figure 2 shows the u and v velocity contours for a numerical solution with hat basis functions, a trapezoidal quadrature rule, $N = 256$, $M = 20$, and $\epsilon = 1.5825\text{e-}04$ (black lines). These results are superposed over the exact solution (red lines). Note that the solution is quite poor inside the cylinder, but is robust outside of the cylinder. I do not why this is happening, but the point used in all the experiments above should be fine.

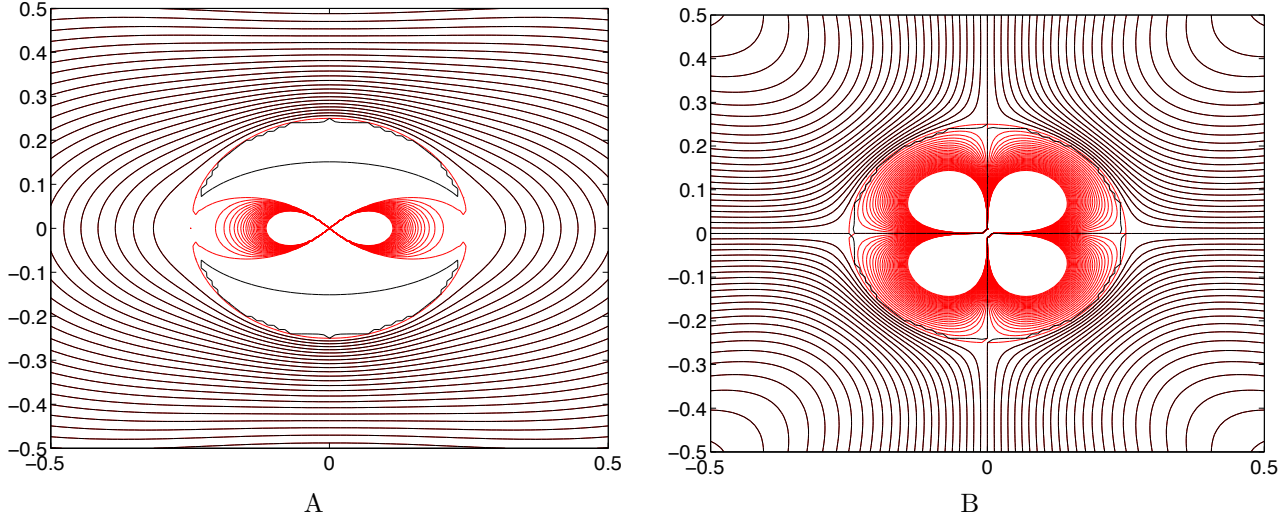


Figure 2: A. The contours of the component of velocity in the x -direction, u , in the square $[-0.5, 0.5] \times [-0.5, 0.5]$. B. The same for the y -component, v .

Blob parameters

We examine the same problem using the L_2 error over square patches of velocity both near and far from the circle as in Fig. 3.

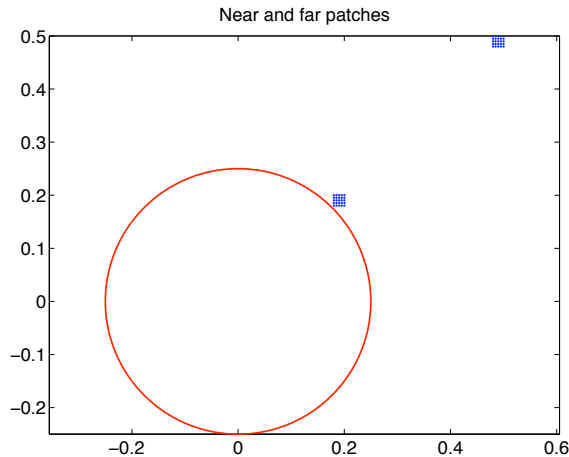


Figure 3: The near and far patches on which the L_2 error is calculated.

First we look at the difference between the original method and a BEM with hat basis functions and the trapezoidal rule with 20 points per chord. We plot error versus blob parameter value for differing numbers of chords (N) in Fig. 4. The original method and the BEM method are very similar for large N or large ϵ , but overall the BEM offers more flexibility in the choice of ϵ for a given error level.

We then fix blob parameters and plot L_2 error versus the number of chords. The results for 4 different blob parameters are in Fig. 5. For a large enough blob parameter, the results of the original method versus the BEM are the same, but at

smaller blob parameters and lower N , the BEM has a much lower error.

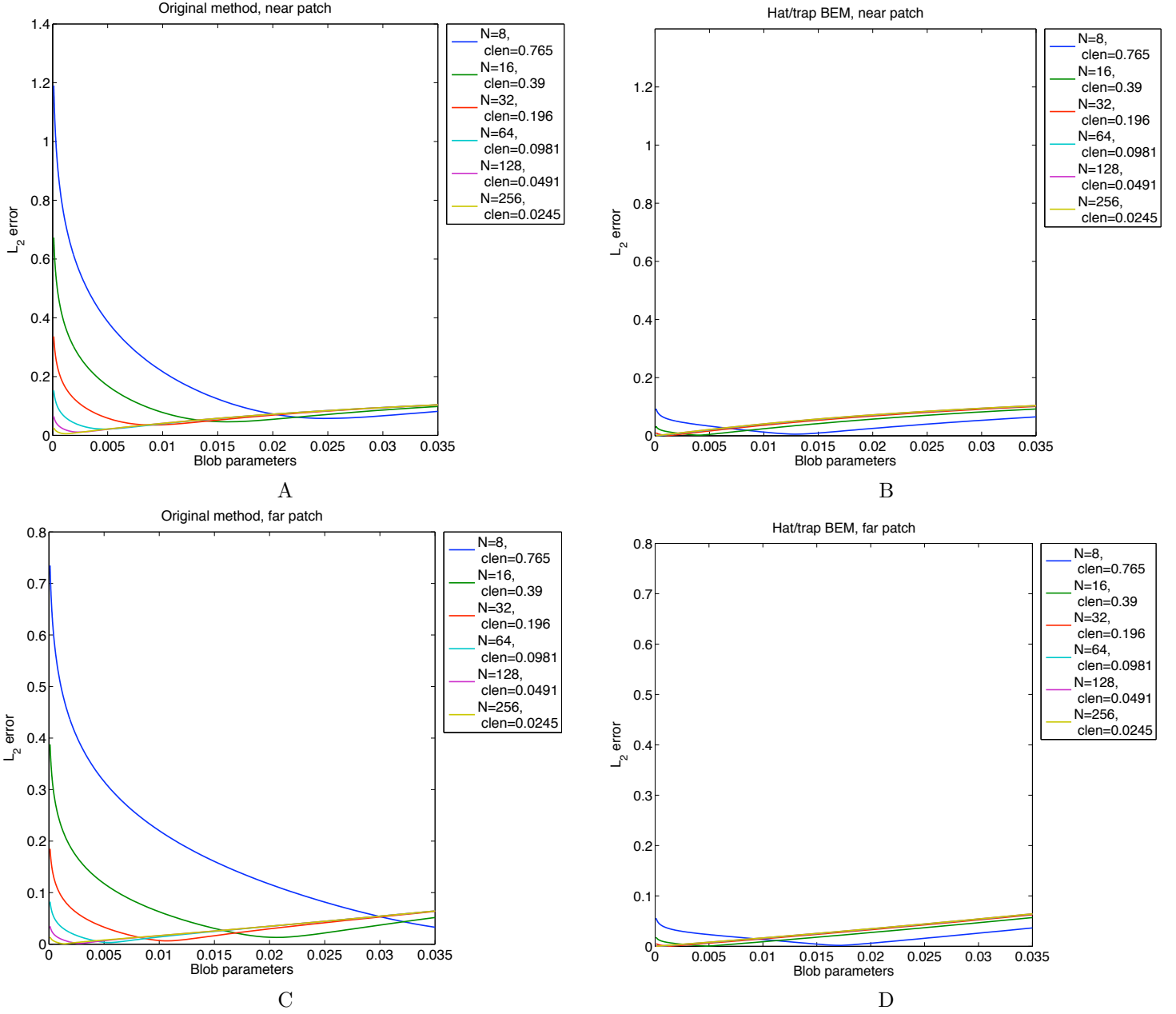


Figure 4: L_2 error versus blob parameter for the near patch (A and B) and the far patch (C and D) using either the original method (A and C) or a BEM using hat functions and the trapezoid rule (B and D).

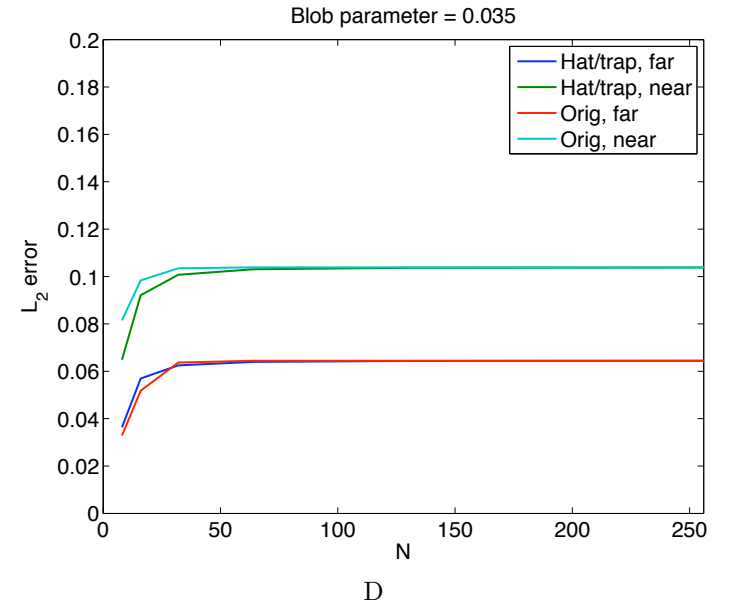
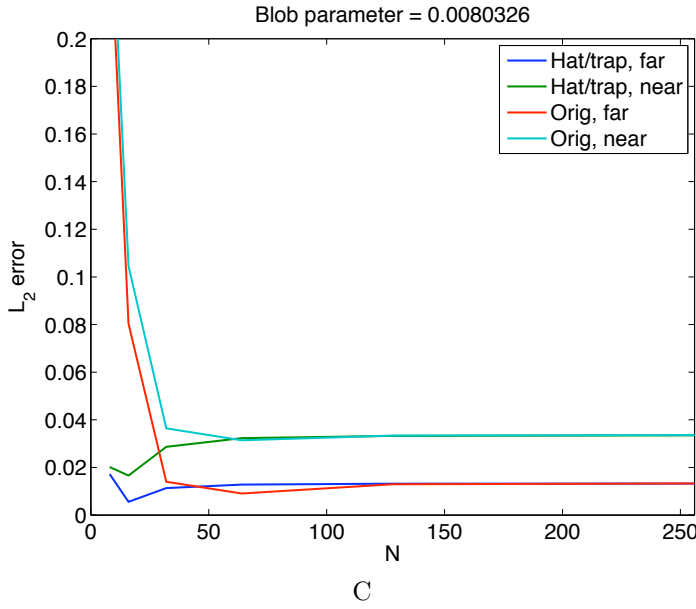
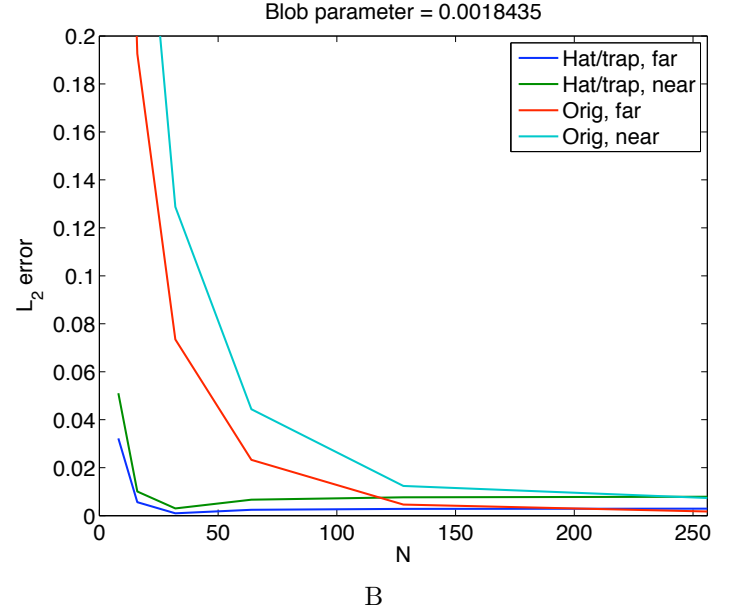
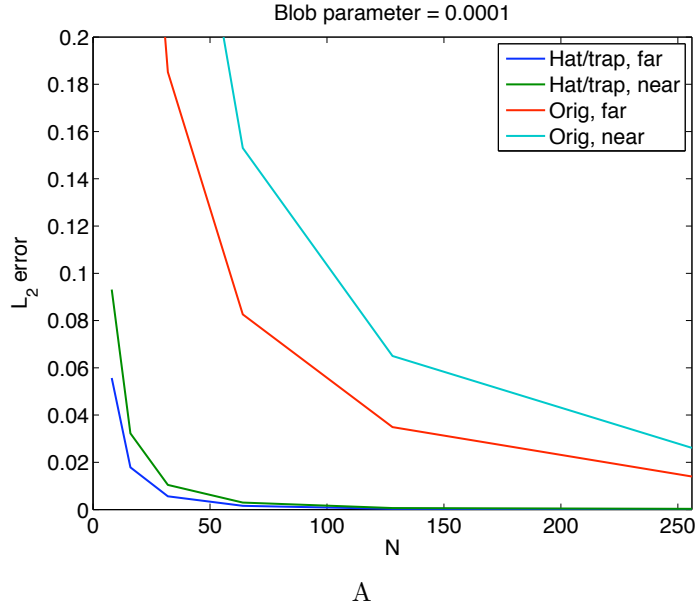


Figure 5: L_2 error versus N for various blob parameters.