Abstract. In traditional multigrid, the number of unknowns on refined domains can create a memory bottleneck. The memory requirements can also reduce the efficiency of the algorithm due to the communication required between cores. I analyze Segmental Refinement Multigrid (SRMG) a modification to multigrid methods that reduces communication and has logarithmic scaling memory complexity. While previous work has been done on this method, and there have been earlier implementations, they all failed to converge on an arbitrary number of levels. My work provides a theoretical analysis and practical implementation demonstrating the root cause of SRMG's previous failures was due to the underlying low-communication mechanisms of the algorithm. My concluding analysis demonstrates future potential for SRMG, although it imposes requirements that could not be met by the implementation accompanying this work.

Key words. Multigrid, Segmental Refinement

AMS subject classifications. Lookup comp math

1. Introduction. Here we study a low-memory modification of multigrid methods, which are a class of methods for solving partial differential equations. This modified multigrid method, Segmental Refinement (SRMG) [**?**], reduces concurrent memory requirements by creating an overlapping domain decomposition and independently solving the partial differential equation on each subset of the domain. Error imposed by this method exists only in the overlapping region, or buffer region, and requires only modest additional computation in the redundant overlap between domains. This overlapping decomposition is the reason SRMG has logarithmic scaling memory requirements rather than the linear scaling memory requirements of traditional multigrid.

Previous analysis and implementation demonstrated that SRMG could work for some fixed number of levels [**?**], but my work attempts to demonstrate that convergence holds for any number of levels of SRMG, so long as each level eliminates low-frequency error and further levels maintain this high-frequency accuracy.

There are two unanswered questions regarding SRMG. One is whether we have convergence guarantees in spite of communication no longer occuring between patches. I show error induced by no longer communicating with the entire domain is localized and does not extend past the buffer regions of the domain decomposition.

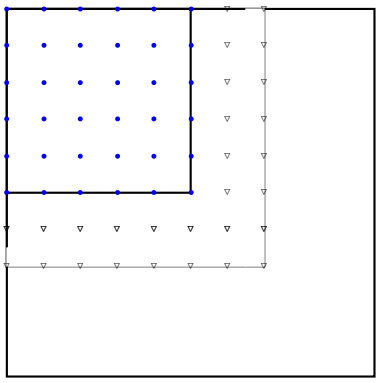

FIG. 1. * **JRT:Get better picture** Significant error is constrainted to triangular black points

The second question is whether SRMG converges on an arbitrary number of refinement levels. I demonstrate a set of requirements which, when satisfied, guarantee convergence on an arbitrary number of levels. I accompanied my analysis by implementing the method with a spectral solver, to investigate the feasibility of these requirements.

2. Background. By 1982, Multigrid was an accepted numerical algorithm for solving elliptic equations [**?**]. The speed at which multigrid solves large problems is the primary reason for its use. One of the problems with multigrid, as well as many other numerical algorithms, is that memory requirements scales linearly.

Models of this SRMG algorithm were presented in detail as early as 1994 [**?**]. However, there is still debate over what conditions must be met for the algorithm

to guarauntee convergence. One question that had been left unanswered by previous research is whether Segmental Refinement can be performed an arbitrary number of times to yield arbitrarily small error. [**?**].

There has been proof of convergence via segmental refinement with a fixed number of levels. However this proof is unaccompanied by numerical implementation on any specific differential equations and lacks strong evidence that unlimited levels of SRMG are possible [**?**]. The theory presented showed that error induced by one level of SRMG decays within the buffer region of each subdomain, but did not demonstrate that a buffer region of fixed size can still allow further levels of SRMG without eventually inducing errors on the highest-resolution domains. However, this was an early work and still served to demonstrate that this algorithm had a potential use in reducing communication complexity during the last few stages of refinement when performing multigrid. In and of itself, this provided a significant, powerful use. Yet, it left unanswered the full potential of SRMG.

In 2015, an actual implementation was investigated [**?**]. A communication model was proposed once again in depth, and with an accompanying numerical implementation, provided very strong support that SRMG had potential as a low-memory but reliable partial differential equation solver.

In their work, the researchers discovered that convergence depended on an additional parameter not previously considered [**?**]. In their implementation of SRMG, the authors found that increasing the resolution of the coarsest level on which SRMG is used proportionally decreased the error induced by SRMG and also the buffer region required to achieve convergence on the interior of each subdomain. Previous analysis made no connection between the resolution on the coarsest grid and the error induced by subsequent levels of SRMG.

Still, there was no evidence suggesting segmental refinement could or could not be performed indefinitely. Errors in the 2015 implementation grew beyond acceptable levels unless the buffer size grew as compensation [**?**]. Increasing the buffer size also increases the memory requirements and computational complexity of SRMG. Therefore, a requirement that buffer size grow with the number of SRMG levels would undermine the advantages offered by using this algorithm in place of parallel multigrid. It was also not clear what was causing convergence failures under different circumstances, and whether they could be ameliorated while preserving the low-memory, high-speed properties of SRMG.

My work builds upon this body of previous work by approaching the analysis of the algorithm from a new perspective, showing that convergence depends on the frequency of accuracy on the coarsest level. In order for SRMG to be effective, one must eliminate all low-frequency errors on coarser patches. This high-frequency accuracy requirement motivated the implementation of a spectral method coupled with SRMG to investigate the convergence of SRMG. Spectral solvers allow exact interpolation via injection, eliminating at least one possible source of low-frequency error: interpolation error.

Even when eliminating the possibility of interpolation error, convergence failed. There were many possible causes, including aliasing error. I performed a second, more abstract analysis in light of these results, which explored convergence without considering any specific solver, interpolation operator, or restriction operator. Error induced by the first set of artificial boundary conditions, which is imposed on patches generated the first level of SRMG, cannot not be eliminated by refinement, only by increasing coarsest resolution or buffer size.

The value of the analysis done in this work is two-fold: finding the cause of previ-

ous failures of multiple-level SRMG and the derivation of explicit error bounds. The analysis supports and clarifies the failures of previous implementations, demonstrating that the required growth of the buffer boundary [**?**] was intrinsic to SRMG and could not be eliminated. Furthermore, it demonstrates exactly what requirements must be met in order to achieve convergence with SRMG when one knowns in advance how many levels of SRMG are desired.

3. Methodology.

3.1. Model Problem. Consider Poisson's equation on the domain $[0,1]$ x $[0,1]$

$$\begin{aligned} \triangle u(x,y) &= f \\ u(0,y) = u_{west}(y) \quad u(1,y) &= u_{east}(y) \\ u(x,0) = u_{south}(x) \quad u(x,1) &= u_{north}(x) \end{aligned} \tag{1}$$

We can attempt to solve this problem numerically using multigrid.

Discretizing (1) by a spectral method and solving using a multigrid method yields an approximate solution with a Fourier representation that is accurate up to the first k modes. To increase the value of this k, one can solve on a more refined grid using the approximation generated on a coarse grid as an initial guess for the solution on the refined discretization.

We consider a modification of multigrid: segmental refinement, which solves for the refined domain by locally solving on patches of an overlapping decomposition of the domain.

A brief outline of the SRMG method is:

1. Solve coarse grid
2. Quarter grid into four patches
3. Extend boundary of patch to include buffer region.
4. Refine each patch
5. Solve on patch, then repeat steps 2-4 on this new patch.

3.2. Justification of Convergence. The SRMG algorithm crucialy relies on the accuracy of the solution far from the boundary. If error decays sufficiently rapidly away from a boundary, then an inexact boundary condition combined with a buffer region can be used to obtain an accurate solution inside the domain.

To this end, we will look at the decay of the solution to the Laplace equation on a square domain with boundary values driven by high-frequency errors. Assume that step 1 of SRMG is completed, meaning a coarse solution $u^{(0)}(x,y)$ is generated as an approximation to the true solution. This coarse solution $u^{(0)}$ is used to impose boundary conditions on the patches generated by steps 2 and 3.

Consider in specific the patch solution $u^{(1)}$ generated on the lower left patch $[0,L]$ x $[0,L]$, which interfaces with the true boundary on the West and South but is given by artificially boundary conditions on the other two sides.

$$\begin{aligned} \triangle u^{(1)}(x,y) &= f \\ u^{(1)}(0,y) = u_{west}(y) \quad u^{(1)}(L,y) &= u^{(0)}(L,y) \\ u^{(1)}(x,0) = u_{south}(x) \quad u^{(1)}(x,L) &= u^{(0)}(x,L) \end{aligned} \tag{2}$$

Let the error between the true solution and the coarse solution be given by $E^{(0)}(x,y) = u^{(0)}(x,y) - u(x,y)$

We can therefore analyze the error on this lower left patch, $E^{(1)}(x,y) = u^{(1)}(x,y) - u(x,y)$, resulting from this first SRMG decomposition by expressing it as a solution to the following PDE:

$$\triangle E^{(1)}(x,y) = \triangle u^{(1)} - \triangle u = f - f = 0 \tag{3}$$
$$E^{(1)}(0,y) = u_{west}(y) - u_{west}(y) = 0 \quad E^{(1)}(L,y) = u^{(0)}(L,y) - u(L,y) = E^{(0)}(L,y)$$
$$E^{(1)}(x,0) = u_{south}(x) - u_{south}(x) = 0 \quad E^{(1)}(x,L) = u^{(0)}(x,L) - u(x,L) = E^{(0)}(x,L)$$

On $[0,L] \times [0,L]$. We can obtain a generic boundary condition by superposition of solutions satisfying one boundary conditions and three zero boundary value conditions.

Therefore, we further simplify our analysis by individually evaluating the error imposed by one boundary.

$$\begin{aligned} \triangle E(x,y) &= 0 \\ E(0,y) = 0 \quad & u(L,y) = 0 \\ u(x,0) = 0 \quad & u(x,L) = E^{(0)}(x,L) \end{aligned} \tag{4}$$

This partial differential equation has series solution [**?**]

$$\begin{aligned} E(x,y) &= \sum_{n=1}^{\infty} E_n(x,y) \\ &= \sum_{n=1}^{\infty} B_n \sin\left(\frac{n\pi x}{L}\right) \sinh\left(\frac{n\pi y}{L}\right), \end{aligned} \tag{5}$$

where the coefficients B_n can be determined in terms of the inhomogeneous boundary condition, using the orthogonality of trigonometric functions.

$$B_n = \frac{2}{L \sinh(n\pi)} \int_0^L E^{(0)}(x,L) \sin\left(\frac{n\pi x}{L}\right) \, dx$$

As one moves from the top boundary, the solution to this problem decays exponentially, which will motivate the size we require for our buffer used in SRMG.

Thus we have the bound

$$\begin{aligned} |E_n(x,y)| &\leq \frac{2B_n}{L \sinh(n\pi)} \sinh\left(\frac{n\pi y}{L}\right) \\ &\leq \frac{4B_n}{L \cdot exp(n\pi)} exp(\frac{n\pi y}{L}) \\ &\leq \frac{4B_n}{L} exp(\frac{n\pi(y-L)}{L}) \end{aligned}$$

Letting $d = L - y$ be the distance from the boundary, we have

$$|E_n(x,y)| \leq \frac{4C}{L} exp(\frac{-n\pi d}{L}), \tag{6}$$

and we see that the exponential decay away from the boundary is also exponential in the wavenumber of the perturbation.

For our solution procedure, we assume that we obtain the solution on an initial coarse grid to remove any low-frequency errors. Thus along any artificial boundary of some form analogous to $y \in [0, L]$ and $x = L$ we have accuracy in the first N_0 Fourier components. That is the error corresponding to each wavefrequency, (5) is 0 for $n \leq N_0$.

Further assume that each quarter of the domain contains $N_0 \times N_0$ unknowns.

We will number each grid starting with $l = 0$ for this coarsest grid.

If we use a second order method for solving each finite-resolution patch, then the discretization error [1] at each level is $C_D \left(\frac{1}{2^l n_0}\right)^2$ for some constant C_D. If we double the resolution of our initial grid in each direction, but divide it into four patches, then at level $l = 1$ we have $L = \frac{1}{2}$ [2].

Let $C = \max\left(|B_{n \leq 2N_0 - 1}|, \sum_{n=2N_0}^{\infty} |B_n|\right)$

$$
\begin{aligned}
|E(x,y)| &\leq \sum_{n=N_0+1}^{\infty} |\frac{4B_n}{L} exp(\frac{-n\pi d}{L})| \\
&= \sum_{n=N_0+1}^{2N_0-1} |\frac{4B_n}{L} exp(\frac{-n\pi d}{L})| + \sum_{n=2N_0}^{\infty} |\frac{4B_n}{L} exp(\frac{-n\pi d}{L})| \\
&\leq \frac{4C(N_0-1)}{L} exp(\frac{-N_0\pi d}{L}) + \frac{4C}{L} exp(\frac{-N_0\pi d}{L}) \\
&= \frac{4CN_0}{L} exp(\frac{-N_0\pi d}{L})
\end{aligned}
\tag{7}
$$

Since later levels of SRMG may generate subpatches with up to four artificial boundaries, we require that the total error imposed by up to four erroneous boundaries be less than the discretization error at this level,

$$
\begin{aligned}
4\frac{4CN_0}{L} exp(\frac{-N_0\pi d_1}{L}) &< C_D \left(\frac{L}{N_0}\right)^2 \\
\implies exp(\frac{-N_0\pi d_1}{L}) &< \frac{C_D}{C}\frac{L^3}{16N_0^3} \\
\implies d_1 &> -\frac{L}{N_0\pi} \ln\left(\frac{C_D}{C}\frac{L^3}{16N_0^3}\right) \\
\implies d_1 &> \frac{\ln(C) - \ln(C_D) + (4+3)\ln(2) + 3\ln(N_0)}{2^1 N_0 \pi}.
\end{aligned}
\tag{8}
$$

This way, we can guarantee second-order discretization error on each patch. This is the same guarantee given by a second-order finite difference method, and multigrid

[1] To clarify any ambiguity, here our second order discretization error is defined as second order with respect to the distance between points or number of unknowns per unit area on a boundary, not cell size, which would actually be $(\frac{1}{k})^4$. This is a trivial difference because the additional needed buffer size would at most double, or we could achieve this by doubling the coarse-solve accuracy n_0

[2] Technically, SRMG must create a buffer zone, so $L = \frac{1}{2} + \epsilon$. We disregard this detail in the analysis provided here.

with a finite difference discretization still eliminates low-frequency errors using the coarser grids.

We continue to assume that refining each patch before quartering assures that each subpatch is generated with higher-frequency accuracy. We interpret this as an initial approximation on on all four subpatches that is exact up to the first N_0 components of the true Fourier representation of the given subpatch's boundary. That is (5) is still 0 for $n \leq N_0$, but L has shrunken, meaning the expression for the boundary-imposed error decays faster.

On level l, let d_l refer to the distance requirement at l levels from the coarsest grid in order to have the error imposed by artificial boundaries decay to discretization error with respect to the given refinement. We assume that each level of SRMG is performed by first doubling the resolution of each patch, then quartering the domain. Note that this means discretization error on a patch at level l is given by $\frac{C_D}{2^l N_0}$

We can derive a requirement for d_l of the following form

$$d_l > \frac{\ln(C) - \ln(C_D) + (4+3l)\ln(2) + 3\ln(N_0)}{2^l N_0 \pi} \tag{9}$$

The sum $d = \sum_{l=1}^{\infty} d_l$ converges. This means that a finite but suitably large buffer zone around the initial coarse patches will guarantee convergence of the method on all subsequent levels. Of course, we need that $d < 1/2$ and ideally much less, but we can achieve this by raising N_0, the initial grid refinement. For implementation, we are concerned with $\overline{d}_l$, the total buffer needed at level l which would allow convergence on all subsequent patches.

$$\begin{aligned}
\overline{d}_l &= \sum_{m=l}^{\infty} d_m \\
&> \sum_{m=l}^{\infty} \frac{\ln(C) - \ln(C_D) + 4\ln(2) + 3\ln(N_0)}{2^m N_0 \pi} + \frac{3m\ln(2)}{2^m N_0 \pi}, \\
&> \frac{\ln(C) - \ln(C_D) + 4\ln(2) + 3\ln(N_0)}{2^{l-1} N_0 \pi} + \frac{3(l+1)\ln(2)}{2^{l-1} N_0 \pi} \qquad (10)
\end{aligned}$$

so the number of cells $c_l = \overline{d}_l 2^l$ required in the buffer in each direction is

$$c_l = 2\frac{\ln(C) - \ln(C_D) + (7+3l)\ln(2) + 3\ln(N_0)}{N_0 \pi},$$

which means at each level we must increase the number of cells by a constant, but the overall distance from the boundary remains constant.

In summary, we require that the solution perturbation, driven by high-frequency errors in the artificially imposed boundary conditions, decays to discretization error within some distance d_l away from the patch boundary, where l is the level index for the patch. Continuing this process recursively by allowing an additional buffer of size d_l for each level, we can show that segmental refinement at each level finds solutions to a PDE up to discretization error. This is the same guarantee offered by traditional multigrid with a finite difference method.

Since the infinite sequence $\{d_l\}$is summable, a finite buffer $\sum_{i=1}^{\infty} d_i = \overline{d_1}$ on the first level $l = 1$ of SRMG is sufficient to guarantee an accurate solution on all successive patches without communication with the whole domain.

3.3. Implementation. Aspects of the implementation of this method was described in detail by John Boyd [**?**]

In order to solve the partial differential equation $\triangle u(x) = f(x)$ with boundary boundary conditions u(x) = g(x) along the boundary $\partial\Omega$

$$\begin{aligned} \triangle u(x) &= f(x) \\ u(x) &= g(x)\ \forall x \in \partial\Omega \end{aligned} \tag{11}$$

we can derive an equivalent formulation:

$$\begin{aligned} u(x) &= \phi(x) + B(x) \\ \triangle\phi(x) &= f - \triangle B(x) \\ B(x) &= g(x)\ \forall x \in \partial\Omega \\ \phi(x) &= 0\ \forall x \in \partial\Omega \end{aligned} \tag{12}$$

Therefore, we solve the boundary conditions with B(x) and add it to a function $\phi(x)$ satisfying homogenous boundary conditions such that $\triangle(\phi(x)+B(x)) = f = \triangle u$.

If ϕ is constructed from a set of basis elements that are equal to 0 on the boundary of Ω, then there are less computations required to solve for $\phi(x)$ such that $\triangle\phi(x) = f - \triangle B(x)$ on a discrete domain. In this case, we must only satisfy conditions for the interior points $\Omega \cap \partial\Omega^c$, since homogenous boundary conditions are satisfied by choice of basis.

Given this suitable choice of basis functions, and if the domain is two dimensional with N_0 points in each dimension of our discretization, this implementation requires inverting a system $(N_0 - 2) * (N_0 - 2)$ unknowns, rather than $N_0 \times N_0$ unknowns. For large N_0, the computational difference is negligible .

The primary benefit of this implementation is that it improves readability of the code and makes it simpler to enforce the proper boundary conditions, g(x), for a subdomain that shares a boundary with the full domain.

3.4. Construction from Chebyshev Basis. We construct a solution to (11) in the specific case where $\Omega = [-1, 1]x[-1, 1]$.

$\partial\Omega = \{(x, y) : x = 1\} \cup \{(x, y) : y = 1\} \cup \{(x, y) : x = -1\} \cup \{(x, y) : y = -1\}$

$$u(-1, y) = u_{west}(y),\ u(1, y) = u_{east}(y),\ u(x, -1) = u_{south}(y),\ u(x, 1) = u_{north}(y)$$

We can construct the function B(x, y) as follows:

$$\begin{aligned} B(x, y) = &\frac{1-x}{2}u_{west}(y) + \frac{1+x}{2}u_{east}(y) + \frac{1-y}{2}u_{south}(x) + \frac{1+y}{2}u_{north}(x) \\ &- \frac{(1-x)(1-y)}{4}u(-1, -1) - \frac{(1-x)(1+y)}{4}u(-1, 1) \\ &- \frac{(1+x)(1-y)}{4}u(1, -1) - \frac{(1+x)(1+y)}{4}u(1, 1) \end{aligned}$$

$$\triangle B(x,y) = \frac{1-x}{2}\,\frac{\partial^2 u_{west}(y)}{\partial y^2} + \frac{1+x}{2}\,\frac{\partial^2 u_{east}(y)}{\partial y^2} + \frac{1-y}{2}\,\frac{\partial^2 u_{south}(x)}{\partial x^2} + \frac{1+y}{2}\,\frac{\partial^2 u_{north}(x)}{\partial x^2}$$

Our choice of ϕ is motivated by the Chebyshev polynomials, T_n. Note that Chebyshev polynomials satisfy the following property

$$T_{2n}(\pm 1) = 1$$
$$T_{2n+1}(\pm 1) = \pm 1$$

It is therefore convenient to consider the recombination of Chebyshev polynomials

$$P_{2n}(x) = T_{2n} - 1$$
$$P_{2n+1}(x) = T_{2n+1} - x$$

Therefore $P_n(\pm 1) = 0$ and $P_i(x)P_j(y) = 0\ \forall i,j \in N_0 \times N_0\ \forall x,y \in \partial\Omega$

Also note that $P_0(x) = 0 = P_1(x)$

The construction of ϕ is generated as an element from the space of a tensor product of recombined Chebyshev polynomials.

So $\phi(x,y) = \sum_{i,j} c_{i,j}\ P_i(x)P_j(y)$ is a suitable choice to construct our solution via the implementation described in (12).

Assume that we discretize our domain with N_0 points in each dimension and $N_0 - 2$ points on the interior of the domain. Then ϕ is calculated by solving the linear system

$\sum_{i=2}^{N_0}\sum_{j=2}^{N_0} c_{i,j}\ (P_i''(x)P_j(y) + P_j''(y)P_i(x)) = f(x,y) - \triangle B(x,y)$

3.5. Segmental Refinement: Solving on Subdomains. The method for constructing the solution to (11) as described above is only valid for $\Omega = [-1,1]x[-1,1]$, but we are interested in the process of solving the PDE on subdomains with exact forcing function f given by the problem, and artificial boundary conditions imposed by previous levels of segmental refinement.

Therefore, the domain of each of the patches of Segmental Refinement are not $[-1,1]\times[-1,1]$ but are instead given by some $\tilde{\Omega} = [a,b]\times[c,d]$.

A simple modification allows us to use the same approach as above by considering two linear mappings M_x, M_y such that:

$$M_x([a,b]) = [-1,1]$$
$$M_x([c,d]) = [-1,1]$$

Therefore the solution to a PDE on a subdomain with boundary conditions imposed by the previous levels of SRMG is given by

$$\triangle u(x,y) = f(x,y) \tag{13}$$
$$u(x,y) = g(x,y)\ \forall (x,y) \in \partial(\tilde{\Omega} = \partial\{[a,b]\times[c,d]\})$$

with equivalent expression:

$$\begin{aligned} \triangle u(M_x(x), M_y(y)) &= f(x,y) \\ u(M_x(x), M_y(y)) &= g(x,y)\ \forall (x,y) \in \partial\tilde{\Omega} \end{aligned} \tag{14}$$

Note that via the chain rule, $\triangle u(M_x(x), M_y(y)) = (M_y')^2 \frac{\partial^2 u(M_x(x), M_y(y))}{\partial M_y^2} + (M_x')^2 \frac{\partial^2 u_(M_x(x), M_y(y))}{\partial M_x^2}$

Therefore, with a very simple modification, we can use the same method to solve subdomains or patches generated by Segmental Refinement as we do to solve the coarse domain. The only difference is that boundary conditions will not be imposed by the problem, but by an approximate solution from a previous level of SRMG.

4. Results.

4.1. Accuracy of Method. The coarse resolution was set at $N_0 = 10$

L_1 error was computed with a ten-point Gaussian Quadrature on each patch. Each patch's L_1 error was summed up over all of the patches corresponding to a given level to achieve a total metric for L_1 with respect to a given level.

Here the L_1 error was calculated against the true solution $\sin 2\pi x \sin 2\pi y$

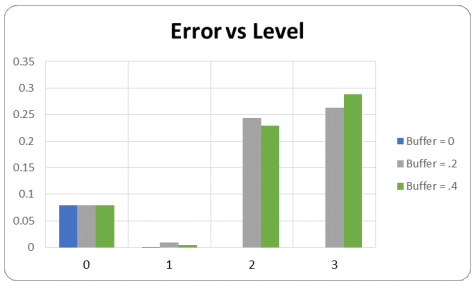

FIG. 2. *Error plateus or increases after level 1*

Here the L_1 error was calculated against the true solution $u = \sin 2\pi y \frac{\sinh 2\pi(x-1)}{\sinh 4\pi}$

FIG. 3. *Error plateus or increases after level 1*

This implementation of SRMG could not consistently achieve error reduction on levels beyond the first level of SRMG, regardless of buffer size, coarse resolution, or number of levels.

5. Explanation of Results. The underlying assumptions of SRMG require that subsequent levels generate higher-frequency approximations of the true solution. However, low-frequency errors coming from the first level of SRMG seemed to propagate into the interior of all subpatches generated at later levels.

A new, more careful analysis was done with this problem in mind, leading to a new condition for iterative SRMG.

5.1. Poisson's Equation on Subdomains. Consider the uniqueness theorem of Poisson's equation [**?**].

THEOREM 5.1. *Let $u(\Omega)$ be a solution to Poisson's equation*

$$\nabla^2 u = f$$

$$u = g \text{ on } \partial\Omega$$

$u(\Omega)$ is unique.

We can also conclude the following corrolary.

COROLLARY 5.2. *On any closed subset of the domain $\tilde{\Omega}_i \subset \Omega$, there is a unique solution v to the following equation.*

$$\nabla^2 v = f$$

$$v = u \text{ on } \partial\tilde{\Omega}_i$$

$v(\tilde{\Omega}_i)$ is unique. Furthermore, u, the solution to Poisson's equation on the full domain, restricted to $\tilde{\Omega}_i$ satisfies this equation $\implies v = u$.

5.2. Recursive SRMG. Assume the following about the mechanics of SRMG: each subdomain generated by the l^{th} level of SRMG, Ω_i^l, is a proper subset of some subdomain Ω_j^{l-1} with $\Omega^0 = \Omega$

Furthermore assume that on each level of SRMG for $l \geq 1$, a patch solution $u_i^{(l)}$ is approximated to the following Poisson equation:

$$\begin{aligned} \triangle u_i^{(l)} &= f \\ u_i^{(l)} &= u_j^{(l-1)} \in \partial\Omega_i^l \end{aligned} \tag{15}$$

Here $u_j^{(l-1)}$ is a solution to a corresponding Poisson equation from a subdomain generated by the previous level of SRMG. That is, each patch is a solution to Poisson's equation with artificial boundary conditions imposed by the solution generated via the previous level of SRMG.

THEOREM 5.3. *For $l \geq 2$, the analytical solution $u_i^{(l)} \in \Omega_i^l \subset \Omega_j^{l-1}$ is unique and equals: $u_j^{(l-1)} \in \Omega_j^{l-1}$.*

The above theorem follows directly from 5.2

COROLLARY 5.4. *The analytical solution to the equations governing any $u_i^{(l)}$ is given by the solution $u_j^{(1)}$ where $\Omega_i^l \subset \Omega_j^1$*

The direct result of this is that $e_i^{(l)} = u_i^{(l)} - u = u_j^{(1)} - u = e_j^{(1)}$. That is, the pointwise error induced by imposing artificial boundary conditions on level 1 has an effect on all subsequent levels of Segmental Refinement.

This contradicts one of our fundamental requirements of SRMG, that high-frequency errors on the interior of a patch could be eliminated by refinement. Unfortunately, error imposed by the first artificial boundary cannot be eliminated by anything other than a more refined coarse solve, or a suitably large buffer. The required size of this buffer grows to accommodate the stronger convergence requirements on higher-resolution grids.

5.3. Error Analysis Revisited. Note that our original analysis holds the for the first level of Segmental Refinement. If $u^{(0)} - u$ contains primarily high frequncies, then the first artificial boundaries induced by SRMG impose second-order discretization error with a suitably large buffer. The problem is that further levels of SRMG still encode this error, and this error decays no faster on later levels, regardless of refinement.

Therefore, SRMG requires not just that $e^{(0)} \leq \frac{C_D}{N_0^2}$, but that $e^{(0)} \leq \epsilon$, where ϵ is the target accuracy at the finest level of SRMG.

We revisit just the first-level buffer requirements, given by (10), but with the updated requirement on d_1 that $e^{(0)} \leq \epsilon$

$$d_1 > \frac{5\ln 2 + \ln C + \ln N_0 - \ln \epsilon}{2N_0\pi}$$

$$\begin{aligned}
\overline{d_l} &= \sum_{i=l}^{\infty} d_i \\
&= \frac{ln2}{2N_0\pi} 2^{1-l}(l+5) + 2^{1-l}\frac{\ln C + \ln N_0 - \ln \epsilon}{N_0\pi} \\
&= \frac{2^{1-l}}{N_0\pi}\left(\frac{(l+5)\ln 2}{2} + \ln C + \ln N_0 - \ln \epsilon\right)
\end{aligned}$$

An extremely important thing to note is that this analysis crucially relies on no low-frequency error being introduced by any subsequent level of SRMG. This may cause failure to converge not only on this given patch, but on any subsequent subdomain, since my analsis has shown that no method of refinement will eliminate this error.

Therefore, we still must consider an implementation that remedies all sources of low-frequency error including aliasing error, interpolation error, or limitations of the basis used to represent each patch solution.

5.4. Conclusion. Previous analysis and work relied on the assumption that if each level of SRMG could guarantee error on the same order of magnitude as traditional multigrid, then it would also inherit the multigrid property that this error would be restricted to higher frequencies. This property on the frequency of error does not hold in SRMG. Even if we make the extremely optimistic assumption that we can generate the exact solution to each patch's PDE, it is clear that the error induced by the first iteration of Segmental Refinement induces error that cannot be restricted to

higher frequencies. Thus, the first buffer region must grow to accomodate the final desired error.

Computational resources and/or a suitably optimized spectral solver were not available to test the newfound convergence requirements. However, we have shown here several numerical experiments and a rigorous analysis of Segmental Refinement. All are consistent with previous work, where the buffer size had to grow to accommodate further levels of SRMG. I have also shown potential solutions for this issue, although the feasibility of these solutions is yet to be determined.

Iterative Segmental Refinement only guarantees convergence if there is no reintroduction of low-frequency error via interpolation, no low-frequency error due to aliasing, and a buffer size large enough to eliminate error induced by the first (and each subsequent) artificial boundary.

These are requirements that are much more difficult to satisfy than those posed by our original justification. However, in the case that these requirements are not feasible, Segmental Refinement is still suitable for a few of the finest levels assuming a large buffer is chosen. My analysis here demonstrates possible points of failure and methods to ameloriate them.