SIAM J. Sci. Comput.
Vol. 43, No. 5, pp. S420–S447

DISCRETIZATION-ERROR-ACCURATE MIXED-PRECISION MULTIGRID SOLVERS[*]

RASMUS TAMSTORF[†], JOSEPH BENZAKEN[†], AND STEPHEN F. MCCORMICK[‡]

Abstract. This paper builds on the algebraic theory in the companion paper *[Algebraic Error Analysis for Mixed-Precision Multigrid Solvers, SISC Vol. 43, No. 5, pp. S392–S419]* to obtain discretization-error-accurate solutions for linear elliptic partial differential equations (PDEs) by *mixed-precision* multigrid solvers. It is often assumed that the achievable accuracy is limited by discretization or algebraic errors. On the contrary, we show that the quantization error incurred by simply storing the matrix in any fixed precision quickly begins to dominate the total error as the discretization is refined. We extend the existing theory to account for these quantization errors and use the resulting bounds to guide the choice of four different precision levels in order to balance quantization, algebraic, and discretization errors in the progressive-precision scheme proposed in the companion paper. A remarkable result is that while iterative refinement is susceptible to quantization errors during the residual and update computations, the V-cycle used to compute the correction in each iteration is much more resilient, and continues to work if the system matrices in the hierarchy become indefinite due to quantization. As a result, the V-cycle only requires relatively few bits of precision per level. Based on our findings, we outline a simple way to implement a progressive precision FMG solver with minimal overhead, and demonstrate as an example that the one-dimensional biharmonic equation can be solved reliably to any desired accuracy using just a few V-cycles when the underlying smoother works well. Additionally, we show that the progressive precision scheme leads to memory savings of up to 50% compared to fixed precision.

Key words. mixed precision, progressive precision, multigrid, rounding error analysis

AMS subject classifications. 65F10, 65G50, 65M55

DOI. 10.1137/20M1349230

1. Introduction.

The *abstract* theory in [6] analyzes rounding-error effects of *algebraic* operations in mixed- and progressive-precision multigrid solvers. The main focus here is applying this analysis to solvers for discretized linear elliptic partial differential equations (PDEs), with the goal of obtaining accuracy on the order of the discretization error in the energy norm. Achieving such algebraic accuracy can often be done by *full multigrid (FMG)* at optimal cost (i.e., comparable to a few matrix multiplies on the finest level), but the approximation property that FMG relies on makes it especially sensitive to rounding errors. One of our main goals is to analyze this sensitivity in order to understand how to achieve optimal results.

To this end, first note that existing rounding-error analyses of linear solvers (e.g., [2, 3, 4, 6]) typically assume that the target matrix is exact. In practice, this assumption is rarely satisfied since forming the matrix itself is subject to rounding errors. We extend the theory in [6] to include errors due to simply storing the linear system in finite precision. Referring to it as *quantization* error, we show that its effect grows much faster under mesh refinement than that of algebraic errors. Our analysis is purely algebraic in nature and therefore applies to linear systems regardless of their origin. It also applies to matrix-free methods because quantization happens regardless of whether the result is stored in main memory or just in a register. Even so, it should be emphasized that the quantization error is the smallest possible error one can

[*] Received by the editors July 1, 2020; accepted for publication (in revised form) February 18, 2021; published electronically June 24, 2021.

[†] Walt Disney Animation Studios, Burbank, CA (Rasmus.Tamstorf@disneyanimation.com, Joseph.Benzaken@disneyanimation.com).

[‡] University of Colorado at Boulder, Boulder, CO (stephen.mccormick@colorado.edu).

consider for the formation process, so in some sense this still represents an optimistic analysis. In particular, we do not consider errors due to numerical quadrature during assembly of the linear system.

To make the theory concrete, we consider classical finite element discretizations of PDEs, which facilitates quantifying the discretization error and comparing it to quantization and algebraic errors. This comparison in turn allows us to explore the optimal relationship between the different precision levels introduced in the progressive-precision multigrid solver in [6]. At first, it might appear as if the quantization error limits the benefit of using three precisions in iterative refinement. However, the benefit remains as long as the various precision levels are chosen carefully. Furthermore, because the inner solver only needs to reduce the residual in the iterative refinement scheme by a small amount, we prove that it is *not* necessary to require that the system matrix remains positive definite when rounded to the lowest precision. Avoiding this requirement allows us to use very low precision for the inner solver where most of the computations are performed. By comparison, [4] also uses low precision for the inner solver, but they require an unknown perturbation to be added to the low precision matrix to recover positive definiteness.

We begin in the next section by introducing nomenclature for the different errors involved in our analyses. While we assume that the reader is familiar with [6], Section 3 summarizes its essential definitions and theoretical estimates for completeness. In Section 4, we consider the effects on the V-cycle of quantizing the multigrid components and computing them by the *Galerkin condition* [1]. The effects on FMG of quantizing the multigrid components are analyzed in Section 5. Section 6 brings the theory together to establish the progression requirements for the different precision levels studied throughout the paper. In Section 7, we continue to leverage the theory and show that the overall memory usage can be reduced by up to 50% when using progressive precision instead of fixed precision. Section 8 introduces a simple model problem based on the one-dimensional biharmonic equation. This problem is then studied in Section 9 with various mesh sizes and approximation orders. We illustrate the behavior of a standard V-cycle and also demonstrate that the problem can be solved reliably to any accuracy by progressive precision $\mathcal{FMG}$ when the precision levels are chosen appropriately. We end the paper with some concluding remarks in the last section.

2. Error definitions.

The numerical solution of PDEs in finite precision involves several different error sources. This section introduces notation and vocabulary to distinguish these basic types of errors.

Consider a positive-definite self-adjoint linear PDE of the form $\mathcal{L}u = f$ subject to some boundary conditions, with source term f and exact solution u. Assume that $A_h x_h = b_h$ represents its discretization on a regular grid of element size h by the Galerkin finite element method, where A_h, x_h, and b_h are exact. Assume also that $x_h = (x_{h,i})$, where the $x_{h,i}$ are the coefficients corresponding to the basis functions $\phi_{h,i}$ so that the finite element solution for grid h is the function $u_h = \sum_i x_{h,i}\phi_{h,i}$. The discretization error in the energy or $\mathcal{L}$ norm for grid h is then represented by $e_{\text{disc}} = \|u_h - u\|_{\mathcal{L}}$. In practice, this is computed through quadrature using the bilinear form for the PDE.

Simply rounding the coefficients $x_{h,i}$ to $\mathcal{B}$ bits is denoted by $\operatorname{fl}(x_h)$ and the corresponding continuous solution (with a slight abuse of notation) by $\operatorname{fl}(u_h) = \sum_i \operatorname{fl}(x)_{h,i}\phi_{h,i}$. Given this, the *floating-point error* due to representing the exact solution at grid level h in $\mathcal{B}$ bits of precision is denoted by $e_{\text{fl}} = \|\operatorname{fl}(u_h) - u_h\|_{\mathcal{L}} =$

$\| \operatorname{fl}(x_h) - x_h\|_{A_h}$. This is the best level of energy error we can expect to obtain in finite precision.

Since computation with matrices and source terms require that they too be represented in working precision, we let A_h and b_h rounded to $\mathcal{B}$ bits be denoted by $\check{A}_h$ and $\check{b}_h$, respectively. We also use the haček diacritical mark for any quantities derived from $\check{A}_h$ and $\check{b}_h$. For example, $\check{x}_h$ represents the *exact* solution of $\check{A}_h x = \check{b}_h$. Note that, in general, $\check{x}_h \neq \operatorname{fl}(x_h)$ because $\check{x}_h$ is based on $\check{A}_h$ while $\operatorname{fl}(x_h)$ is based on A_h and then rounded to $\mathcal{B}$ bits. In addition to hačeks, we use tilde to denote values *computed* from $\check{A}_h$ and $\check{b}_h$ and superscripts in parentheses for iterations. For example, for the numerical solution of $\check{A}_h x = \check{b}_h$, we denote the i^{th} iterate by $\tilde{x}_h^{(i)}$ and the fully converged algebraic solution by $\tilde{x}_h^{(\infty)}$.

Since $\check{A}_h$ and $\check{b}_h$ are assumed to represent the exact quantities A_h and b_h rounded to $\mathcal{B}$ bits, then any computations required to obtain $\check{A}_h$ and $\check{b}_h$ must be accurate to at least $\mathcal{B}$ bits. In particular, any numerical quadrature used must be sufficiently accurate and that any uncertainty in the representation of the coefficients in the PDE must be insignificant relative to the rounding error assumed in $\check{A}_h$ and $\check{b}_h$.

Corresponding to the vectors $\check{x}_h$, $\tilde{x}_h^{(i)}$, and $\tilde{x}_h^{(\infty)}$ are the functions $\check{u}_h$, $\tilde{u}_h^{(i)}$, and $\tilde{u}_h^{(\infty)}$, which allow us to write the *quantization error* as $e_{\text{quant}} = \|\check{u}_h - u_h\|_{\mathcal{L}}$ and the *algebraic error* as $e_{\text{alg}} = \|\tilde{u}_h^{(i)} - \check{u}_h\|_{\mathcal{L}} = \|\tilde{x}_h^{(i)} - \check{x}_h\|_{A_h}$. Note that $e_{\text{alg}} \neq \|\tilde{x}_h - \check{x}_h\|_{\check{A}_h}$ in general. Finally, note that while a multigrid algorithm would presumably converge to $(\check{A}_h)^{-1}\check{b}_h$ in infinite precision, it is generally limited from doing so in finite precision. Accordingly, we decompose the algebraic error into *iteration error* $e_{\text{iter}} = \|\tilde{u}_h^{(i)} - \tilde{u}_h^{(\infty)}\|_{\mathcal{L}}$ and *rounding error* $e_{\text{round}} = \|\tilde{u}_h^{(\infty)} - \check{u}_h\|_{\mathcal{L}}$. It might be argued that quantization error is also a kind of rounding error, but for the purposes of this paper, we consider it separately.

Combined, these definitions allow us to write the total error after i iterations as

$$\begin{aligned} e_{total}^{(i)} &= \|\tilde{u}_h^{(i)} - u\|_{\mathcal{L}} \\ &= \|\tilde{u}_h^{(i)} - \tilde{u}_h^{(\infty)} + \tilde{u}_h^{(\infty)} - \check{u}_h + \check{u}_h - u_h + u_h - u\|_{\mathcal{L}} \\ &\le \underbrace{\|\tilde{u}_h^{(i)} - \tilde{u}_h^{(\infty)}\|_{\mathcal{L}}}_{e_{\text{iter}}} + \underbrace{\|\tilde{u}_h^{(\infty)} - \check{u}_h\|_{\mathcal{L}}}_{e_{\text{round}}} + \underbrace{\|\check{u}_h - u_h\|_{\mathcal{L}}}_{e_{\text{quant}}} + \underbrace{\|u_h - u\|_{\mathcal{L}}}_{e_{\text{disc}}}. \end{aligned} \tag{2.1}$$

In summary, quantities without tildes are exact (in infinite-precision arithmetic), those with tildes are computed, those with hačeks are based on the quantized versions of A_h and b_h, and those with subscript h have been discretized on a grid with element size h. We assume that there are no errors in computing $\check{A}_h$ aside from the quantization itself. Thus, any numerical integration used to compute A_h must be sufficiently accurate and the algebraic error associated with evaluating functions at the quadrature points and summing must be insignificant.

3. Existing Theory.

This section summarizes the notation, conventions, and theory of [6]. Initially, we consider three floating point environments: "standard" precision with unit roundoff $\boldsymbol{\varepsilon}$, "high" precision with unit roundoff $\bar{\boldsymbol{\varepsilon}}$, and "low" precision with unit roundoff $\dot{\boldsymbol{\varepsilon}}$. We also refer to these as $\boldsymbol{\varepsilon}$-, $\bar{\boldsymbol{\varepsilon}}$-, and $\dot{\boldsymbol{\varepsilon}}$-precision, respectively. While it is only formally assumed that $\bar{\boldsymbol{\varepsilon}} \le \boldsymbol{\varepsilon} \le \dot{\boldsymbol{\varepsilon}}$, we address the choice of these precision levels in Section 6.

The theory uses variables and expressions for exact quantities, with δ's added and estimated to represent quantities computed in finite precision. Thus, for $A \in \mathbb{R}^{n\times n}$

symmetric positive definite (SPD) with at most m_A nonzeros per row and $b \in \mathbb{R}^n$, then $A^{-1}b + \delta$ denotes a *computed* approximate solution with error δ of

$$Ax = b, \tag{3.1}$$

and $Ax - b + \delta_r$ denotes its *computed* residual with error δ_r.

In what follows, we let $|\cdot|$ denote the vector or matrix with entries replaced by their absolute values. Inequalities and equalities between vectors and matrices are defined componentwise. Using $\|\cdot\|$ to denote the Euclidean norm for a vector and its induced matrix norm (together with the Euclidean inner product $\langle\cdot,\cdot\rangle$), we frequently use the fact that $\||z|\| = \|z\|$ for any vector z (although this is not generally true for matrices). Our rounding-error estimates are in terms of the discrete *energy* norm $\|\cdot\|_A$ defined by $\|x\|_A = \|A^{\frac{1}{2}}x\|, x \in \mathbb{R}^n$. Following the usual convention in rounding-error analyses, we assume that A and b in (3.1) are exact. To rein in the complexity of our estimates, we take this assumption further by assuming exactness of all of the multigrid components: the intergrid transfer, the system matrix, and the right-hand side on all levels. However, we consider the effects of quantization of these components in Section 5 because they are used specifically within the multigrid solvers. Let $\kappa(\cdot)$ denote condition number. For simplicity, dependence is suppressed for A with possible subscripts that remain (e.g., $\kappa_j = \kappa(A_j)$) and similarly for $\underline{\kappa}$ and ψ defined below, but the dependence is explicitly included otherwise. Then the following notation is used in this paper:

$$\kappa = \|A\| \cdot \|A^{-1}\|,\ \psi = \psi(A) = \||A|\|,\ \underline{\kappa} = \underline{\kappa}(A) = \psi\|A^{-1}\|, \bar{m}_A^+ = \frac{m_A + 1}{1 - (m_A + 1)\bar{\boldsymbol{\varepsilon}}},$$

$$\dot{m}_A = \frac{m_A}{1 - m_A\dot{\boldsymbol{\varepsilon}}}, \dot{\tau} = \kappa^{\frac{1}{2}}\dot{\boldsymbol{\varepsilon}}, \tau = \kappa^{\frac{1}{2}}\boldsymbol{\varepsilon}, \bar{\tau} = \kappa\bar{\boldsymbol{\varepsilon}}, \gamma = \frac{\kappa^{\frac{1}{2}} + \underline{\kappa}}{\kappa}.$$

The mixed-precision approach analyzed theoretically in [6] uses iterative refinement as the outer loop and a generic approximate linear solver as the inner loop. The pseudocode for iterative refinement ($\mathcal{IR}$) is given in Algorithm 3.1 below. The floating-point operations in $\mathcal{IR}$ use all three precisions. The full residual r between successive calls to the inner solver is evaluated in $\bar{\boldsymbol{\varepsilon}}$-precision (red font), while the inner solver uses $\dot{\boldsymbol{\varepsilon}}$-precision (green font). All other operations use $\boldsymbol{\varepsilon}$-precision (blue font).

Algorithm 3.1 `Iterative Refinement` ($\mathcal{IR}$)

```
Input: A, b, x initial guess, tol > 0 convergence tolerance.
1: $r \leftarrow Ax - b$                      ▷ Compute $\mathcal{IR}$ Residual and Round
2: if $\|r\| <$ tol then
3:     return $x$                             ▷ Return Solution of $Ax = b$
4: end if
5: $y \leftarrow$ InnerSolve$(A, r)$          ▷ Compute Approximate Solution of $Ay = r$
6: $x \leftarrow x - y$                       ▷ Update Approximate Solution of $Ax = b$
7: goto 1
```

THEOREM 3.1. $\mathcal{IR}$. [6] *Let $x^{(i)}$ be the iterate at the start of the i^{th} cycle of $\mathcal{IR}$ and $r = Ax^{(i)} - b$ its residual computed in $\bar{\boldsymbol{\varepsilon}}$-precision and rounded to $\boldsymbol{\varepsilon}$-precision. Suppose that a $\rho < 1$ exists such that, for any $r \in \mathbb{R}^n$, the solver used in the inner loop of Algorithm 3.1 (line 5) is guaranteed to compute a correction y that satisfies*

$$\|y - A^{-1}r\|_A \le \rho\|A^{-1}r\|_A.$$

Then $x^{(i+1)}$ approximates the solution $A^{-1}b$ of (3.1) with the relative *error bound*

$$\frac{\|x^{(i+1)} - A^{-1}b\|_A}{\|A^{-1}b\|_A} \le \rho_{ir} \frac{\|x^{(i)} - A^{-1}b\|_A}{\|A^{-1}b\|_A} + \chi, \quad \rho_{ir} = \rho + \delta_{\rho_{ir}}, \tag{3.2}$$

where

$$\delta_{\rho_{ir}} = \frac{(1+2\rho)\tau + \gamma(1+\rho)(1+\boldsymbol{\varepsilon})\bar{m}_A^+ \bar{\tau}}{1-\tau}, \quad \chi = \frac{\tau + \gamma(1+\rho)(1+\boldsymbol{\varepsilon})\bar{m}_A^+ \bar{\tau}}{1-\tau}. \tag{3.3}$$

If $\rho + \delta_{\rho_{ir}} < 1$, then the error after $N \ge 1$ $\mathcal{IR}$ cycles with initial guess $x^{(0)}$ satisfies

$$\frac{\|x^{(N)} - A^{-1}b\|_A}{\|A^{-1}b\|_A} \le (\rho + \delta_{\rho_{ir}})^N \frac{\|x^{(0)} - A^{-1}b\|_A}{\|A^{-1}b\|_A} + \frac{\chi}{1 - (\rho + \delta_{\rho_{ir}})}. \tag{3.4}$$

For the inner solver, we use one V$(1,0)$-cycle, with pseudocode $\mathcal{V}$ given in Algorithm 3.2 below. $\mathcal{V}$ uses a nested hierarchy of ℓ levels from the coarsest $j = 1$ to the finest $j = \ell$, $1 \le j \le \ell$. It begins on the finest level and proceeds down to the coarsest level, with one relaxation sweep on each level along the way. Each level is equipped with a system matrix A_j, with $A_\ell = A$. Assume that relaxation on level j applied to $A_j y_j = r_j$ is the stationary linear iteration $y_j \leftarrow y_j - M_j(A_j y_j - b_j)$, where M_j roughly approximates A_j^{-1}. Let P_j denote the interpolation matrix that maps from level $j-1$ to level j with at most m_{P_j} nonzeros per row or column. Let $P_1 = 0$ for the coarsest level, which involves just one relaxation sweep and no further coarsening. Assume further that the Galerkin condition is exactly satisfied on all coarse levels: $A_{j-1} = P_j^t A_j P_j$, $2 \le j \le \ell$. (See Section 4 for analysis of the rounding-error effects when this relationship is used to compute the coarse-level matrices in finite precision.) All computations in $\mathcal{V}$ are performed in low $\dot{\boldsymbol{\varepsilon}}$-precision as shown in green font in the pseudocode. Accordingly, since the input right-hand side (RHS) may be in higher precision, the cycle is initialized with a rounding step.

Algorithm 3.2 `V(1,0)-Cycle (`$\mathcal{V}$`) Correction Scheme`

```
Input: A, r, P, $\ell \ge 1$ $\mathcal{V}$ levels.
$r \leftarrow r$                                            ▷ Round RHS and Initialize $\mathcal{V}$
$y \leftarrow Mr$                                           ▷ Relax on Current Approximation $(y = 0)$
if $\ell > 1$ then                                          ▷ Check for Coarser Level
    $r_{\mathrm{v}} \leftarrow Ay - r$                      ▷ Evaluate $\mathcal{V}$ Residual
    $r_{\ell-1} \leftarrow P^t r_{\mathrm{v}}$              ▷ Restrict $\mathcal{V}$ Residual to Coarse-Level
    $d_{\ell-1} \leftarrow \mathcal{V}(A_{\ell-1}, r_{\ell-1}, P_{\ell-1}, \ell-1)$   ▷ Compute Correction from Coarser Levels
    $d \leftarrow P d_{\ell-1}$                             ▷ Interpolate Correction to Fine Level
    $y \leftarrow y - d$                                    ▷ Update Approximate Solution of $Ay = r$
end if
return $y$                                                  ▷ Return Approximate Solution of $Ay = r$
```

The theory in [5] and the references cited therein establish optimal energy convergence in infinite precision of Algorithm 3.2 under fairly general conditions for fully regular elliptic PDEs discretized by standard finite elements. We simply assume this to be the case by supposing that the error propagation matrix V_j for level j is bounded by a constant $\rho_v^* \in [0,1)$ for all j, that is, $\|V_j\|_{A_j} \le \rho_v^*, 1 \le j \le \ell$.

One aim of this paper is to verify the theory in [6] for multigrid applied to a large class of PDEs, including the model problem introduced below. Accordingly, we have in mind matrices whose condition numbers depend on the mesh size h. (While we do not explicitly exclude coarsening in terms of the degree p of the discretization, our focus is on coarsening in terms of h.) To abstract this dependence, define the *pseudo mesh size* by $h_j = \underline{\kappa}_j^{-\frac{1}{2m}}$, $1 \leq j \leq \ell$, where m is a positive integer, and the *mesh coarsening factor* by $\theta_j = \frac{h_{j-1}}{h_j}$, $2 \leq j \leq \ell$. In the geometric setting, $2m$ correspond to the order of the PDE. Under standard assumptions for finite element discretizations, classical theory shows that the condition number on a given grid is bounded by a constant (depending on the finite element approximation order) times h_{min}^{-2m}, where h_{min} is the smallest element size on that grid (see [8, Sec. 5.2]). In this case, h_j is therefore bounded by that constant times the grid j mesh size.

To allow for a *progressive-precision* V-cycle, where precision is tailored to each level in the hierarchy, assume now that $\dot{\boldsymbol{\varepsilon}}$ varies by letting $\dot{\boldsymbol{\varepsilon}}_j$ denote the unit roundoff used on level j, $1 \leq j \leq \ell$. We use similar notation for other parameters that may now depend on the level, but suppress the subscript when the level is understood. Specifically, $\dot{\boldsymbol{\varepsilon}}_j$-precision is used on level j to store the data, perform relaxation, transfer residuals to level $j-1$ and corrections to level $j+1$, and round residuals transferred from level $j+1$. Define the *precision coarsening factor* by $\dot{\zeta}_j = \frac{\dot{\boldsymbol{\varepsilon}}_{j-1}}{\dot{\boldsymbol{\varepsilon}}_j}$, $2 \leq j \leq \ell$. We will revisit the actual value of $\dot{\zeta}_j$ in Section 6. To accommodate the use of a geometric series involving the rounding-error effects on each level of the V-cycle, denote the *coarsening ratio* by $\vartheta = \min_{1 \leq j \leq \ell}\{\theta_j \dot{\zeta}_j^{-\frac{1}{m}}\}$ and assume that $\vartheta > 1$. To account for rounding errors in relaxation, suppose that a constant α_{M_j} exists such that computing $M_j z_j$ for any vector z_j on level j in $\dot{\boldsymbol{\varepsilon}}_j$-precision yields

$$M_j z + \delta_{M_j}, \qquad \|\delta_{M_j}\| \leq \alpha_{M_j} \dot{\boldsymbol{\varepsilon}}_j \|z_j\|, \; 1 \leq j \leq \ell. \tag{3.5}$$

For example, Richardson iteration with $M_j = \frac{\omega}{\|A_j\|}$, $0 < \omega < 2$, yields $\alpha_{M_j} < \frac{2}{\|A_j\|}$ (see [6]). Assume further that relaxation is monotonically convergent in energy: $\|I_j - M_j A_j\|_A < 1$. To simplify what follows, assume that σ is a constant such that

$$\sigma \geq (1 + \dot{\boldsymbol{\varepsilon}}_j) \max\{\alpha_{M_j}\|A_j\|, |||A_j|||\alpha_{M_j}, |||A_j||| \cdot \|M_j\|\}, \; 1 \leq j \leq \ell.$$

Only the low precision varies by level in the V-cycle because its finest level is fixed. On the other hand, the full multigrid algorithm introduced below uses progressively finer levels for its inner-loop V-cycles. We therefore introduce variable $\boldsymbol{\varepsilon}_j$ and $\bar{\boldsymbol{\varepsilon}}_j$, $1 \leq j \leq \ell$, for this purpose, where ℓ is now the very finest level used in FMG. Finally, we redefine the following parameters to mean their maxima over all levels:

$$\kappa(P^t P) = \max_{1 \leq j \leq \ell} \kappa(P_j^t P_j), \quad \dot{m}_A^+ = \max_{1 \leq j \leq \ell} \frac{m_{A_j}}{1 - m_{A_j}\dot{\boldsymbol{\varepsilon}}_j}, \quad \dot{m}_P^+ = \max_{1 \leq j \leq \ell} \frac{m_{P_j}}{1 - m_{P_j}\dot{\boldsymbol{\varepsilon}}_j},$$
$$\bar{m}_A^+ = \max_{1 \leq j \leq \ell} \frac{m_{A_j}}{1 - m_{A_j}\bar{\boldsymbol{\varepsilon}}_j}, \; m_P^+ = \max_{1 \leq j \leq \ell} \frac{m_{P_j}}{1 - m_{P_j}\boldsymbol{\varepsilon}_j}, \; \mu = 3(\max_{1 \leq j \leq \ell} \dot{\zeta}_j)\kappa^{\frac{1}{2}}(P^t P)\dot{m}_P^+.$$

The next theorem confirms that $\mathcal{V}$ reduces the error optimally toward the solution of the target matrix equation $Ay = r$ provided that the perturbation δ_{ρ_v} of the exact convergence factor satisfies $\delta_{\rho_v} < 1 - \rho_v^*$, meaning that coarsening in the level hierarchy should be fast enough (i.e., large enough θ_j) and progression of the precision should be slow enough (i.e., small enough $\dot{\zeta}_j$) to ensure that $\vartheta \gg 1$. More significantly, it requires the finest-level scale parameter to satisfy $\dot{\tau} \ll 1$, which in turn means that $\kappa \ll \dot{\boldsymbol{\varepsilon}}^{-2}$. Together with Theorem 3.1, we can then conclude that the mixed-precision version of $\mathcal{IR}$ with a $\mathcal{V}$ as the inner loop converges optimally to the solution of (3.1) until to the order of the lower limit χ is reached.

THEOREM 3.2. $\mathcal{V}$. [6] *Define the following cubic polynomial in* $\dot{\tau}_j$*:*

$$\delta_{\rho_v} = \delta_{\rho_v}(\dot{\tau}_j) = \frac{\vartheta^m}{\vartheta^m - 1}\left(a_1\dot{\tau}_j + a_2\dot{\tau}_j^2 + a_3\dot{\tau}_j^3\right), \tag{3.6}$$

where

$$a_1 = 4 + \sigma + 4\mu,\ a_2 = 2(3 + \sigma + \dot{m}_A^+(1+\sigma))\mu_j + 2 + \sigma,\ a_3 = 2(1 + \sigma + \dot{m}_A^+(1+\sigma))\mu_j.$$

If $\dot{\tau}_j$ *is small enough that* $\delta_{\rho_v}(\dot{\tau}_j) < 1 - \rho_v^*$, $1 \le j \le \ell$, *then one cycle of the progressive-precision version of Algorithm 3.2 for solving the* $\mathcal{IR}$ *residual equation converges according to* $\|y - A^{-1}r\|_A \le \rho_v\|A^{-1}r\|_A$, $\rho_v = \rho_v^* + \delta_{\rho_v}$.

Full multigrid uses a special cycling scheme that targets the underlying PDE, with the aim of attaining accuracy comparable to how well the finest-grid solution approximates the PDE solution. FMG starts on the coarsest grid and proceeds to the finest, making sure that enough V-cycles are used on each grid along the way to achieve accuracy comparable to that grid's discretization accuracy. In essence, if grid $j-1$ is solved to within the discretization error Ch_{j-1}^q for some positive constants C and q, then using that result as an initial guess on grid j means that the initial error on grid j is bounded by some small multiple (depending on θ_j) of Ch_j^q. This in turn means that only a few V-cycles are needed to obtain discretization accuracy on grid j (i.e., error below Ch_j^q). For standard finite elements, $q = k - m$, where $2m$ and $k > m$ correspond to the order of the PDE and the order of the finite elements (e.g., polynomials of degree $p = k - 1$), respectively [8, Sec. 2.2].

The full multigrid algorithm based on $N \ge 1$ inner $\mathcal{IR}$ cycles, each using one $\mathcal{V}$, is given below by the pseudocode $\mathcal{FMG}$. Note that $\mathcal{FMG}$ amounts to three nested loops: outer $\mathcal{FMG}$, middle $\mathcal{IR}$, and inner $\mathcal{V}$. The choice of N is critical because it must guarantee convergence to within discretization accuracy on each level. The goal of Section 5 is to determine N in the presence of rounding errors.

Algorithm 3.3 FMG(1, 0)-Cycle ($\mathcal{FMG}$)

```
Input: A, b, P, $N \ge 1$ $\mathcal{IR}$ cycles (using one V(1,0) each), $\ell \ge 1$ $\mathcal{FMG}$ levels.
$x \leftarrow 0$                                                 ▷ Initialize $\mathcal{FMG}$
if $\ell > 1$ then                                               ▷ Check for Coarser Level
    $x_{\ell-1} \leftarrow \mathcal{FMG}(A_{\ell-1}, b_{\ell-1}, P_{\ell-1}, \ell-1, N)$   ▷ Compute Coarse-Level Approximation
    $x \leftarrow Px_{\ell-1}$                                   ▷ Interpolate Approximation to Fine Level
end if
$i \leftarrow 0$                                                 ▷ Initialize $\mathcal{IR}$
while $i < N$ do
    $r \leftarrow Ax - b$                                        ▷ Update $\mathcal{IR}$ Residual and Round
    $y \leftarrow \mathcal{V}(A, r, P, \ell)$                    ▷ Compute Correction by $\mathcal{V}$
    $i \leftarrow i + 1$                                         ▷ Increment $\mathcal{IR}$ Cycle Counter
    $x \leftarrow x - y$                                         ▷ Update Approximate Solution of $Ax = b$
end while
return $x$                                                       ▷ Return Approximate Solution of $Ax = b$
```

To obtain an abstract sense of discretization accuracy, assume that $b_{j-1} = P_j^t b_j$, $2 \le j \le \ell$, are also computed exactly. We characterize the relative accuracy of adjacent levels in the hierarchy by assuming that C is a positive constant such that the following *strong approximation property (SAP)* holds:

$$\|P_j A_{j-1}^{-1} b_{j-1} - A_j^{-1} b_j\|_{A_j} \le C h_{j-1}^q \|A_j^{-1} b_j\|_{A_j}, \quad 2 \le j \le \ell, \quad q = k - m, \tag{3.7}$$

(C and h_j may depend on k, but we assume that this order is fixed in what follows.) While (3.7) characterizes the relative error in a coarse-level solution with respect to the next finer level, it also suggests the following definition. We say that x_j solves $A_j x_j = b_j$ *to the order of discretization error* or simply *to discretization accuracy* if

$$\|x_j - A_j^{-1} b_j\|_{A_j} \le C h_j^q \|A_j^{-1} b_j\|_{A_j}, \quad 1 \le j \le \ell. \tag{3.8}$$

We assume that this level of approximation is achieved on the coarsest level $j = 1$ by just a few relaxation sweeps starting with a zero initial guess.

Theorem 3.3. *$\mathcal{FMG}$.* [6] *Assume that $\rho_v + \delta_{\rho_{ir}} < 1$ and that χ is small enough and N is large enough that the following holds on all levels $j \in \{1, 2, \ldots, \ell\}$:*

$$(\rho_v + \delta_{\rho_{ir}})^N \left((\sqrt{2} + \mu\tau)\theta^q C h^q + \mu\tau \right) + \frac{\chi}{1 - (\rho_v + \delta_{\rho_{ir}})} \le C h^q, \tag{3.9}$$

where $h = h_j$, $\theta = \theta_j$, $\tau = \tau_j$, and $\mu = \mu_j = \kappa^{\frac{1}{2}}(P_j^t P_j) m_P^+$, and (with subscript j understood) parameter ρ_v is defined in Theorem 3.2 and parameters $\delta_{\rho_{ir}}$ and χ are defined in (3.3). Then Algorithm 3.3 solves (3.1) to the order of discretization error on each level.

4. Effects of Quantization & Galerkin Construction on $\mathcal{V}$ & $\mathcal{IR}$. The components A_j, P_j, and b_j have so far been assumed to be exact for all j. In this and the next section, we extend the theory from [6] to include *quantization errors* incurred from simply storing the components in finite precision. This is in addition to the *algebraic errors* accumulated during computations and already accounted for in the existing theory.

Dropping subscript j, assume that the system matrices are stored in symmetric form and that $\breve{A} = A + \Delta$ and $\breve{b} = b + \delta$ result from simply rounding the exact A and b, respectively, to some $\breve{\varepsilon}$-precision. The actual value of $\breve{\varepsilon}$ will be determined later. We first obtain the general result that $(A + \Delta)^{-1} b \approx A^{-1} b$ and $\kappa(A + \Delta) \approx \kappa$ to the extent that $\underline{\kappa}\breve{\varepsilon} < 1$.

Theorem 4.1. A and b Quantization Errors. *If $\underline{\kappa}\breve{\varepsilon} < 1$, then $A + \Delta$ is SPD and $(A + \Delta)^{-1}(b + \delta)$ approximates $A^{-1} b$ with relative error bounded according to*

$$\frac{\|(A + \Delta)^{-1}(b + \delta) - A^{-1} b\|_A}{\|A^{-1} b\|_A} \le \phi\breve{\varepsilon}, \quad \phi = \frac{\underline{\kappa} + \kappa^{\frac{1}{2}}}{1 - \underline{\kappa}\breve{\varepsilon}}. \tag{4.1}$$

Proof. The relative error in each entry of $A + \Delta$ and $b + \delta$ is bounded by $\breve{\varepsilon}$, which immediately yields the relative error bound

$$\|\delta\| \le \|b\|\breve{\varepsilon}. \tag{4.2}$$

Using $\cdot$ to emphasize multiplication, a bound for $A + \Delta$ follows by noting that $y^t y = |y|^t |y|$ and $y^t \Delta \cdot y = |y^t \Delta \cdot y| \le |y|^t |\Delta| \cdot |y| \le |y|^t |A| \cdot |y| \breve{\varepsilon}$ for any $y \in \mathbb{R}^n$:

$$\|\Delta\| \le \||A|\|\breve{\varepsilon}. \tag{4.3}$$

By (4.3) and noting that $A + \Delta = A^{\frac{1}{2}} (I + E) A^{\frac{1}{2}}, E = A^{-\frac{1}{2}} \Delta A^{-\frac{1}{2}}$, we have that

$$\|E\| \le \|\Delta\| \cdot \|A^{-1}\| \le \||A|\| \cdot \|A^{-1}\|\breve{\varepsilon} = \underline{\kappa}\breve{\varepsilon} < 1, \tag{4.4}$$

which proves that $I + E$ and, hence, $A + \Delta$ are positive definite. Note also that

$$\| (I + E)^{-1} \| \le \frac{1}{1 - \|E\|} \le \frac{1}{1 - \underline{\kappa}\breve{\varepsilon}}. \tag{4.5}$$

We next obtain the following expression for the perturbation of $A^{-1}b$:

$$\begin{aligned}(A+\Delta)^{-1}(b+\delta) - A^{-1}b &= A^{-\frac{1}{2}}\left[(I+E)^{-1} - I\right]A^{-\frac{1}{2}}b + A^{-\frac{1}{2}}\left(I+E\right)^{-1}A^{-\frac{1}{2}}\delta \\ &= A^{-\frac{1}{2}}\left(I+E\right)^{-1}\left(-EA^{\frac{1}{2}}A^{-1}b + A^{-\frac{1}{2}}\delta\right). \qquad (4.6)\end{aligned}$$

Finally, the theorem is proved as follows:

$$\begin{aligned}&\|(A{+}\Delta)^{-1}(b+\delta) - A^{-1}b\|_A \\ &\quad\le \|\left(I+E\right)^{-1}\|\left(\|E\|\cdot\|A^{-1}b\|_A + \|A^{-\frac{1}{2}}\delta\|\right) && \text{by (4.6)} \\ &\quad\le \frac{1}{1-\underline{\kappa}\breve{\boldsymbol{\varepsilon}}}\left(\|E\|\cdot\|A^{-1}b\|_A + \|A^{-\frac{1}{2}}\|\cdot\|\delta\|\right) && \text{by (4.5)} \\ &\quad\le \frac{1}{1-\underline{\kappa}\breve{\boldsymbol{\varepsilon}}}\left(\underline{\kappa}\breve{\boldsymbol{\varepsilon}}\|A^{-1}b\|_A + \|A^{-\frac{1}{2}}\|\breve{\boldsymbol{\varepsilon}}\|b\|\right) && \text{by (4.2) and (4.4)} \\ &\quad\le \phi\breve{\boldsymbol{\varepsilon}}\|A^{-1}b\|_A,\end{aligned}$$

where the last line follows from noting that $\|b\| \le \|A^{\frac{1}{2}}\|\cdot\|A^{-\frac{1}{2}}b\| = \|A^{\frac{1}{2}}\|\cdot\|A^{-1}b\|_A$. □

When the multigrid components are extracted directly from the discretization, quantization in $\dot{\boldsymbol{\varepsilon}}_j$-precision incurs a relative $\underline{\kappa}_j\dot{\boldsymbol{\varepsilon}}_j$ error in these components, where $\underline{\kappa}_j = \||A_j|\|\cdot\|A_j^{-1}\|$. However, our framework also applies to inherently algebraic problems, with the coarse-level matrices in $\mathcal{V}$ possibly constructed based on the Galerkin condition. For simplicity in illustrating rounding effects for this case, we consider a single level $j-1$ only, assuming that A_j and P_j are exact, that P_j has only nonnegative entries, and that the Galerkin condition is computed in $\dot{\boldsymbol{\varepsilon}}_j$-precision in the order given by $P_j^t\left(A_jP_j\right)$. Our next theorem shows that the resulting rounding errors are also $\mathcal{O}(\underline{\kappa}_j\dot{\boldsymbol{\varepsilon}}_j)$.

Theorem 4.2. Galerkin Rounding Errors. *Fix $j \in \{1, 2, \ldots, \ell-1\}$ and assume that A_j and P_j are exact. Then the coarse-level matrix $A_{j-1} = P_j^tA_jP_j + \Delta$ computed from the Galerkin condition in $\dot{\boldsymbol{\varepsilon}}_j$-precision satisfies the relative error bound*

$$\|\Delta\| \le \underline{\kappa}_j\left(2 + \dot{m}_A^+\dot{\boldsymbol{\varepsilon}}_j\right)\dot{m}_A^+\dot{\boldsymbol{\varepsilon}}_j\|P_j^tA_jP_j\|. \qquad (4.7)$$

Proof. Writing the computed A_jP_j as $A_jP_j + \Delta_1$, $|\Delta_1| \le |A_j|P_j\dot{m}_A^+\dot{\boldsymbol{\varepsilon}}_j$, then the computed A_{j-1} can be written as $P_j^t\left(A_jP_j + \Delta_1\right) + \Delta_2$, $|\Delta_2| \le P_j^t|A_jP_j + \Delta_1|\dot{m}_A^+\dot{\boldsymbol{\varepsilon}}_j$. We thus have that

$$\|\Delta\| = \|P_j^t\Delta_1 + \Delta_2\| \le \left(2 + \dot{m}_A^+\dot{\boldsymbol{\varepsilon}}_j\right)\dot{m}_A^+\dot{\boldsymbol{\varepsilon}}_j\|P_j^t\|\cdot\||A_j|\|\cdot\|P_j\|.$$

Bound (4.7) now follows from noting that

$$\|P_j^t\|\cdot\|P_j\| = \|P_j\|^2 \le (\|A_j^{-\frac{1}{2}}\|\cdot\|A_j^{\frac{1}{2}}P_j\|)^2 = \|A_j^{-1}\|\cdot\|P_j^tA_jP_j\|. \qquad □$$

Theorem 4.1 suggests that $\underline{\kappa}_j\dot{\boldsymbol{\varepsilon}}_j \ll 1$ is needed to ensure good V-cycles performance. After all, if A_j is indefinite, then just computing the residual could expand the error associated with the negative spectrum. But this is not really a concern for $\mathcal{V}$: quantization has negligible effect in $\dot{\boldsymbol{\varepsilon}}_j$-precision on $\mathcal{V}$ because this error expansion is small compared to other rounding errors, as our next theorem shows. Note that a similar result holds when all A_j are computed via the Galerkin condition, where (4.7) would be used recursively to account for errors accumulated over all levels. Note also that quantization of M_j would have a truly negligible effect on the performance of $\mathcal{V}$ because preconditioners only need to be crude approximations to the inverse (e.g., relaxation parameters are typically allowed to be anywhere in the interval $(0, 2)$).

Theorem 4.3. $\mathcal{V}$ Quantization Errors. *Let A_j quantized in $\dot{\boldsymbol{\varepsilon}}_j$-precision be denoted by $A_j + \Delta_j$, $|\Delta_j| \le |A_j|\dot{\boldsymbol{\varepsilon}}_j$, $1 \le j \le \ell$. Then Theorem 3.2 holds with $\dot{m}_A^+$ in a_2 and a_3 replaced by the slightly larger $(\dot{m}_A^+ + 1)(1 + \dot{\boldsymbol{\varepsilon}}_1)$.*

Proof. We treat each level individually because the exact A_j is rounded directly, without error accumulation. Dropping subscript j, since A is only used in $\mathcal{V}$ for computing the residual just before coarsening, all we need to do is establish a quantized version of bound (7.12) in the proof of Theorem 7.2 of [6], which is of the form

$$r^{(\frac{1}{2})} = Ay - r + \delta_1, \quad |\delta_1| \le \dot{m}_A^+ \dot{\boldsymbol{\varepsilon}} \left(|r| + |A| \cdot |y|\right).$$

Substituting in quantized A yields $r^{(\frac{1}{2})} = (A + \Delta)y - r + \delta_2$, where

$$|\delta_2| \le \dot{m}_A^+ \dot{\boldsymbol{\varepsilon}} \left(|r| + |A + \Delta| \cdot |y|\right) \le \dot{m}_A^+ (1 + \dot{\boldsymbol{\varepsilon}}) \dot{\boldsymbol{\varepsilon}} \left(|r| + |A| \cdot |y|\right).$$

Thus, $r^{(\frac{1}{2})} = Ay - r + \delta_3$, with

$$|\delta_3| = |\Delta \cdot y + \delta_2| \le (\dot{m}_A^+ + 1)(1 + \dot{\boldsymbol{\varepsilon}}_1) \dot{\boldsymbol{\varepsilon}} \left(|r| + |A| \cdot |y|\right),$$

where we replaced $\dot{\boldsymbol{\varepsilon}}_j$ by $\dot{\boldsymbol{\varepsilon}}_1 \ge \dot{\boldsymbol{\varepsilon}}_j$ to ensure that the changes in a_2 and a_3 amount to just replacing $\dot{m}_A^+$ by the slightly larger *constant* $(\dot{m}_A^+ + 1)(1 + \dot{\boldsymbol{\varepsilon}}_1)$. This completes the proof. □

Remark 4.4. *Sensitivity of $\mathcal{V}$ to Quantization.* Theorem 4.3 confirms that $\mathcal{V}$ is insulated from the indefiniteness that quantization may create. This insensitivity comes from the fact that V-cycles are basically just a hierarchy of simple relaxation steps that have little effect on the near-kernel error components that indefiniteness may alter. On coarse enough levels, relaxation may begin to significantly affect the near-kernel components, but this is just where the system matrices retain positive definiteness (because the condition numbers are small). Other basic relaxation methods may also be insensitive to quantization, but they tend not to be very efficient solvers for PDEs. On the other hand, while direct solvers can be applied to modest-size discrete PDEs, their reliance on positive definiteness to control the error makes them very sensitive to quantization.

Theorem 4.5. $\mathcal{IR}$ Quantization Errors. *Let A and b quantized in $\breve{\boldsymbol{\varepsilon}}$-precision be denoted by $A + \Delta$, $|\Delta| \le |A|\breve{\boldsymbol{\varepsilon}}$, and $b + \delta$, $|\delta| \le |b|\breve{\boldsymbol{\varepsilon}}$, respectively. Then Theorem 3.1 holds with $(1+\boldsymbol{\varepsilon})\bar{m}_A^+\bar{\tau}$ replaced by $(1+\boldsymbol{\varepsilon})\left(\breve{\tau} + (1+\breve{\boldsymbol{\varepsilon}})\bar{m}_A^+\bar{\tau}\right)$ in the expressions for $\delta_{\rho_{ir}}$ and χ in (3.3) , where $\breve{\tau} = \kappa\breve{\boldsymbol{\varepsilon}}$.*

Proof. The proof is analogous to that of Theorem 4.3, but with three terms quantized in bound (4.7) in the proof of Theorem 4.1 of [6], which for exact A and b reads

$$r = Ax - b + \delta_1, \quad |\delta_1| \le \boldsymbol{\varepsilon}|Ax - b| + (1 + \boldsymbol{\varepsilon})\bar{m}_A^+ \bar{\boldsymbol{\varepsilon}} \left(|b| + |A| \cdot |x|\right).$$

Substituting in quantized A and b thus yields $r = (A + \Delta)x - (b + \delta) + \delta_1$, where

$$\begin{aligned}
|\delta_1| &\le \boldsymbol{\varepsilon}|(A + \Delta)x - (b + \delta)| + (1 + \boldsymbol{\varepsilon})\bar{m}_A^+ \bar{\boldsymbol{\varepsilon}} \left(|b + \delta| + |A + \Delta| \cdot |x|\right) \\
&\le \boldsymbol{\varepsilon}|Ax - b| + \boldsymbol{\varepsilon}(|\delta| + |\Delta| \cdot |x|) + (1 + \boldsymbol{\varepsilon})\bar{m}_A^+ \bar{\boldsymbol{\varepsilon}} \left(|b| + |A| \cdot |x| + |\delta| + |\Delta| \cdot |x|\right) \\
&\le \boldsymbol{\varepsilon}|Ax - b| + \left(\boldsymbol{\varepsilon}\breve{\boldsymbol{\varepsilon}} + (1 + \boldsymbol{\varepsilon})(1 + \breve{\boldsymbol{\varepsilon}})\bar{m}_A^+ \bar{\boldsymbol{\varepsilon}}\right) \left(|b| + |A| \cdot |x|\right).
\end{aligned}$$

Thus, $r = (A+\Delta)x - (b+\delta) + \delta_1 = Ax - b + \delta_2$, and the theorem follows because

$$\begin{aligned} |\delta_2| &= |\Delta \cdot x - \delta + \delta_1| \\ &\leq \boldsymbol{\varepsilon}|Ax-b| + |\delta| + |\Delta|\cdot|x| + \left(\boldsymbol{\varepsilon}\check{\boldsymbol{\varepsilon}} + (1+\boldsymbol{\varepsilon})(1+\check{\boldsymbol{\varepsilon}})\bar{m}_A^+\bar{\boldsymbol{\varepsilon}}\right)(|b| + |A|\cdot|x|) \\ &\leq \boldsymbol{\varepsilon}|Ax-b| + (1+\boldsymbol{\varepsilon})\left(\check{\boldsymbol{\varepsilon}} + (1+\check{\boldsymbol{\varepsilon}})\bar{m}_A^+\bar{\boldsymbol{\varepsilon}}\right)(|b| + |A|\cdot|x|)\,. \qquad \square \end{aligned}$$

5. Effect of Input Quantization on $\mathcal{FMG}$.

$\mathcal{FMG}$ is more sensitive to quantization because it relies directly in step 4 of Algorithm 3.3 on the SAP (3.7). (We assume from now on that (3.7) holds when A, b, and P are exact on all levels.) Here we analyze the effects on $\mathcal{FMG}$ of rounding A_{j-1}, P_j, and b_{j-1} for a fixed $j \in \{2, 3, \ldots, \ell\}$ to a given *quantization precision* $\check{\boldsymbol{\varepsilon}}$. To clarify where this rounding occurs, note that each recursive call to $\mathcal{FMG}$ means that $j-1$ serves as the finest level for the inner $\mathcal{V}$ calls on $\mathcal{FMG}$ level $j-1$. So A_{j-1} and b_{j-1} rounded to precision $\check{\boldsymbol{\varepsilon}}_{j-1} \leq \boldsymbol{\varepsilon}_{j-1}$ in $\mathcal{FMG}$ means that these rounded quantities are passed into the recursive call to $\mathcal{FMG}$ from level j to the coarser level. Note that the resulting A_{j-1} is further rounded to $\dot{\boldsymbol{\varepsilon}}_{j-1}$-precision in the inner call to $\mathcal{V}$. Similarly, rounding P_j to $\check{\boldsymbol{\varepsilon}}_j$-precision in $\mathcal{FMG}$ means that this occurs in step 4 when the full approximation is interpolated from the current finest level $j-1$ to the new finest level j. All other multigrid components are processed in $\dot{\boldsymbol{\varepsilon}}$-precision within the inner $\mathcal{V}$ solver. Our final theorem extends Theorem 3.3 to account for these quantization errors, at the cost of increased complexity of the logic and estimates. Aligned with our ultimate goal of balancing errors, the aim here is for both the solver and the rounding errors to be smaller than Ch^q, as opposed to bounding their sum as in (3.9). The key to this extension is to establish a SAP that accounts for quantization. Specifically, with $A_j^{-1}b_j + \delta_j$ denoting the exact solution of $A_j x_j = b_j$ when A_j and b_j have been quantized, then the extended SAP asserts existence of a constant $\check{C}$ such that

$$\|P_j A_{j-1}^{-1} b_{j-1} + \delta_{j-1} - (A_j^{-1}b_j + \delta_j)\|_{A_j} \leq \check{C} h_{j-1}^q \|A_j^{-1} b_j + \delta_j\|_{A_j}, \quad 2 \leq j \leq \ell. \tag{5.1}$$

To reduce complexity, we assume for our final theorem that all of the entries of P_j are nonnegative.

Theorem 5.1. $\mathcal{FMG}$ Quantization Errors. *The extended SAP (5.1) holds with*

$$\check{C} =$$
$$\max_{2\leq j\leq\ell}\{\left(C + \kappa_{j-1}^{\frac{q}{2m}}\left(\phi_{j-1}\check{\boldsymbol{\varepsilon}}_{j-1} + \phi_j\check{\boldsymbol{\varepsilon}}_j + \kappa_j^{\frac{1}{2}}\underline{\kappa}_{j-1}^{\frac{1}{2}}\check{\boldsymbol{\varepsilon}}_j\left(1 + \phi_{j-1}\check{\boldsymbol{\varepsilon}}_{j-1}\right)\right)\right)\left(1 + \phi_j\check{\boldsymbol{\varepsilon}}_j\right)\},$$

where $\phi_j = \frac{\kappa_j + \kappa_j^{\frac{1}{2}}}{1 - \underline{\kappa}_j\check{\boldsymbol{\varepsilon}}_j}$ $2 \leq j \leq \ell$. *Moreover,* $\mathcal{FMG}$ *approximates the solution* $A^{-1}b + \delta$ *of the quantized version of (3.1) to the level of discretization accuracy provided* $\rho_v + \delta_{\rho_{ir}} < 1$ *and the following hold on every level:*

$$\frac{\chi}{1 - (\rho_v + \delta_{\rho_{ir}})} < Ch^q \quad \text{and} \quad (\rho_v + \delta_{\rho_{ir}})^N\left(\theta^q C_c h^q + \mu_c\right) \leq Ch^q, \tag{5.2}$$

where the constants $C_c = \max_j\{(1 + \kappa_j^{\frac{1}{2}}\check{\boldsymbol{\varepsilon}}_j)(1 + \phi_{j-1}\check{\boldsymbol{\varepsilon}}_{j-1})(1 + \phi_j\check{\boldsymbol{\varepsilon}}_j)C + \check{C}\}$ *and* $\mu_c = \max_j \check{\boldsymbol{\varepsilon}}_j(1 + \check{\boldsymbol{\varepsilon}}_j)\kappa_j^{\frac{1}{2}}(1 + Ch_{j-1}^q)(1 + \phi_{j-1}\check{\boldsymbol{\varepsilon}}_{j-1})(1 + \phi_j\check{\boldsymbol{\varepsilon}}_j)$, *and subscript* j *is understood for the other terms in (5.2).*

Proof. Dropping subscript j and replacing subscript $j-1$ by c, let $P+\Delta_P$ denote quantized P, where $|\Delta_P \cdot z| \le |\Delta_P| \cdot |z| \le \breve{\varepsilon} P \cdot |z|$ for any coarse-level z. This proof uses the bounds $\|\delta_c\|_{A_c} \le \phi_c \breve{\varepsilon}_c \|A_c^{-1} b_c\|_{A_c}$, $\|\delta\|_A \le \phi \breve{\varepsilon} \|A^{-1}b\|_A$, and $\|A_c^{-1}b_c + \delta_c\|_{A_c} \le (1+\phi_c\breve{\varepsilon}_c)(1+\phi\breve{\varepsilon})\|A^{-1}b+\delta\|_A$ that are implied by (4.1). The proof assumes familiarity with the logic as well as some estimates used in [6], including $\||z|\|_{A_c} \le \|A_c^{\frac{1}{2}}\| \cdot \|z\| \le \underline{\kappa}_{j-1}^{\frac{1}{2}} \|z\|_{A_c}$ and $\|A_c^{-1} b_c\|_{A_c} \le \|A^{-1}b\|_A$.

To establish the extended SAP (5.1), first note that

$$\begin{aligned}\|\Delta_P \left(A_c^{-1} b_c + \delta_c\right)\|_A &\le \|A^{\frac{1}{2}}\| \breve{\varepsilon} \|P \cdot |A_c^{-1} b_c + \delta_c|\| \\ &\le \kappa^{\frac{1}{2}} \breve{\varepsilon} \||A_c^{-1} b_c + \delta_c|\|_{A_c} \\ &\le \kappa^{\frac{1}{2}} \underline{\kappa}_c^{\frac{1}{2}} \breve{\varepsilon} \left(1 + \phi_c \breve{\varepsilon}_c\right) \|A_c^{-1} b_c\|_{A_c}.\end{aligned}$$

Noting that $h_c = \kappa_c^{-\frac{1}{2m}}$ and using the original SAP (3.7) yields (5.1):

$$\begin{aligned}&\| \left(P+\Delta_P\right) \left(A_c^{-1} b_c + \delta_c\right) - \left(A^{-1} b + \delta\right) \|_A \\ &\qquad \le \|P A_c^{-1} b_c - A^{-1} b\|_A + \|\delta\|_A + \|P \delta_c\|_A + \|\Delta_P \left(A_c^{-1} b_c + \delta_c\right)\|_A \\ &\qquad \le (C h_c^q + \phi \breve{\varepsilon}) \|A^{-1} b\|_A + \left(\phi_c \breve{\varepsilon}_c + \kappa^{\frac{1}{2}} \underline{\kappa}_c^{\frac{1}{2}} \breve{\varepsilon} \left(1 + \phi_c \breve{\varepsilon}_c\right)\right) \|A_c^{-1} b_c\|_{A_c} \\ &\qquad \le \left(C + \kappa_c^{\frac{q}{2m}} \left(\phi_c \breve{\varepsilon}_c + \phi \breve{\varepsilon} + \kappa^{\frac{1}{2}} \underline{\kappa}_c^{\frac{1}{2}} \breve{\varepsilon} \left(1 + \phi_c \breve{\varepsilon}_c\right)\right)\right) h_c^q \|A^{-1} b\|_A \\ &\qquad \le \check{C} h_c^q \|A^{-1} b + \delta\|_A.\end{aligned}$$

For FMG convergence, assume for induction purposes that the coarse-level result, x_c, has properly converged: $\|x_c - (A_c^{-1} b_c + \delta_c)\|_{A_c} \le C h_c^q \|A_c^{-1} b_c + \delta_c\|_{A_c}$. Then

$$\begin{aligned}&\| \left(P+\Delta_P\right) \left(x_c - (A_c^{-1} b_c + \delta_c)\right) \|_A \\ &\qquad \le \|P \left(x_c - (A_c^{-1} b_c + \delta_c)\right)\|_A + \|\Delta_P \left(x_c - (A_c^{-1} b_c + \delta_c)\right)\|_A \\ &\qquad \le (1 + \kappa^{\frac{1}{2}} \breve{\varepsilon}) C h_c^q \|A_c^{-1} b_c + \delta_c\|_{A_c},\end{aligned}$$

which implies that

$$\begin{aligned}&\| \left(P+\Delta_P\right) x_c - (A^{-1} b + \delta)\|_A \\ &\qquad \le \| \left(P+\Delta_P\right) \left(x_c - (A_c^{-1} b_c + \delta_c)\right) \|_A \\ &\qquad\qquad + \| \left(P+\Delta_P\right) \left(A_c^{-1} b_c + \delta_c\right) - (A^{-1} b + \delta)\|_A \\ &\qquad \le \left((1 + \kappa^{\frac{1}{2}} \breve{\varepsilon})(1 + \phi_c \breve{\varepsilon}_c)(1 + \phi \breve{\varepsilon}) C + \check{C}\right) h_c^q \|A^{-1} b + \delta\|_A. \qquad (5.3)\end{aligned}$$

Denote $\left(P+\Delta_P\right) x_c$ computed in $\breve{\varepsilon}$-precision by $\left(P+\Delta_P\right) x_c + \delta_x$, where $|\delta_x| \le \breve{\varepsilon} \left(P+\Delta_P\right) |x_c| \le \breve{\varepsilon}(1+\breve{\varepsilon}) P |x_c|$. But $\|\delta_x\|_{A_c} \le \breve{\varepsilon}(1+\breve{\varepsilon}) \kappa^{\frac{1}{2}} \|x_c\|_{A_c}$ and

$$\begin{aligned}\|x_c\|_{A_c} &\le \|A_c^{-1} b_c + \delta_c\|_{A_c} + \|x_c - (A_c^{-1} b_c + \delta_c)\|_{A_c} \\ &\le (1 + C h_c^q) \|A_c^{-1} b_c + \delta_c\|_{A_c} \\ &\le (1 + C h_c^q)(1 + \phi_c \breve{\varepsilon}_c)(1 + \phi \breve{\varepsilon}) \|A^{-1} b + \delta\|_A.\end{aligned}$$

Thus, $|\delta_x\|_{A_c} \le \breve{\varepsilon}(1+\breve{\varepsilon}) \kappa^{\frac{1}{2}} (1 + C h_c^q)(1 + \phi_c \breve{\varepsilon}_c)(1 + \phi \breve{\varepsilon}) \|A^{-1} b + \delta\|_A$, which with (5.3) confirms that $\|x - (A^{-1} b + \delta)\|_A \le C_c \|A^{-1} b + \delta\|_A$, where $x = \left(P+\Delta_P\right) x_c + \delta_x$ is the initial iterate for the V-cycles on the fine level. As in the proof of Theorem 10.1 in [6], we then invoke Theorems 3.1 and 3.2 above to prove the theorem. □

The condition on N in (5.2) can be substantially simplified by assuming that $\underline{\kappa}_{j-1} \leq \kappa_j$, $\kappa_j^{\frac{1}{2}} \ll \kappa_j$, $\underline{\kappa}_j \check{\boldsymbol{\varepsilon}}_j \approx \kappa_j \check{\boldsymbol{\varepsilon}}_j \ll 1$, and $\kappa_j^{\frac{q+2m}{2m}} \check{\boldsymbol{\varepsilon}}_j \lesssim C$, and by deleting negligible terms. We therefore conclude that $1 + \underline{\kappa}_j \check{\boldsymbol{\varepsilon}} \lesssim 1 + \kappa_j^{\frac{q}{2m}} C \approx 1$, $\phi_j \approx \underline{\kappa}_j$, and $\kappa_j^{\frac{q}{2m}} \phi_j \approx C$, and similarly for other analogous terms. We then have that $\check{C} \approx 4C$, that $C_c \approx C + \check{C} \approx 5C$, and that $\mu_c \approx \kappa_j^{\frac{1}{2}} \check{\boldsymbol{\varepsilon}}_j \ll C$, leading to the simplified condition $5{\rho_v^*}^N \theta^q C h^q \leq C h^q$, i.e., $5{\rho_v^*}^N \theta^q \lesssim 1$. N thus requires a relatively modest increase from $\frac{\log_2(\sqrt{2}) + q \log_2(\theta)}{|\log_2(\rho_v^*)|}$ in Remark 10.2 of [6] to the following estimate that accounts for quantization:

$$N \approx \frac{\log_2(5) + q \log_2(\theta)}{|\log_2(\rho_v^*)|}. \tag{5.4}$$

6. Precision Requirements.

Up to this point, we have referred to $\boldsymbol{\varepsilon}$, $\bar{\boldsymbol{\varepsilon}}$, and $\dot{\boldsymbol{\varepsilon}}$ as standard, high, and low precision, respectively, without specifying how to select these precisions. Additionally, we have introduced $\check{\boldsymbol{\varepsilon}}$ for the quantization precision. In this section, we show how the theoretical estimates can guide the selection of all of these precision levels. While the focus is on FMG, most of the tools discussed here also apply to V-cycles.

Our estimates involve discretization, quantization, rounding, and iteration errors. Ideally, all errors should be comparable so that computation is not wasted on reducing one only to have the end result contaminated by the others. As shown in (2.1), the total error is bounded by the sum of the four types of errors, so our guiding principle is to assume that the *bounds* for each of these errors should be comparable[1]. This goal leads to overestimates of the total error and the individual precision levels, but such is the nature of an a priori theoretical analysis. Also, while (2.1) provides a decomposition of the absolute errors, we are able to focus instead on the relative errors by assuming that the total error is small enough that $\|\tilde{u}_h^{(\infty)}\|_{\mathcal{L}} \approx \|\check{u}_h\|_{\mathcal{L}} \approx \|u_h\|_{\mathcal{L}} \approx \|u\|_{\mathcal{L}}$.

We begin by recalling the basic requirement that $\bar{\boldsymbol{\varepsilon}} \leq \boldsymbol{\varepsilon} \leq \dot{\boldsymbol{\varepsilon}}$. Additionally, we must have $\bar{\boldsymbol{\varepsilon}} \leq \check{\boldsymbol{\varepsilon}}$ since $\bar{\boldsymbol{\varepsilon}}$ is the highest precision used for any computation. If $\bar{\boldsymbol{\varepsilon}} > \check{\boldsymbol{\varepsilon}}$, then any computation will effectively include a rounding operation to at least $\bar{\boldsymbol{\varepsilon}}$-precision, which makes the choice of higher precision for $\check{\boldsymbol{\varepsilon}}$ pointless. On the other hand, if $\check{\boldsymbol{\varepsilon}}$ is strictly greater than $\bar{\boldsymbol{\varepsilon}}$, $\boldsymbol{\varepsilon}$, and/or $\dot{\boldsymbol{\varepsilon}}$, then operations can still be performed in the specified precision by extending all $\check{\boldsymbol{\varepsilon}}$-numbers with trailing zeros.

To determine the required precision levels, we first illustrate in Figure 1 the different types of errors along with an assumed desired error-level, e_{goal}. Given that there are four contributions to the total error, each must be on the order of $\frac{1}{4}e_{\text{goal}}$. This level is shown as a one of the dashed horizontal lines in Figure 1. The energy norm of the discretization error for a standard finite element discretization is given by $e_{\text{disc}} = Ch^q$, where $q = k - m$ [8, Sec. 2.2]. The intersection between this discretization error line and the $\frac{1}{4}e_{\text{goal}}$ line defines the mesh size, h^*, required to obtain the desired accuracy. It may be necessary to round h^* down to the nearest available mesh size. To be in balance, all errors must then be comparable *at this grid resolution.*

The estimates for the quantization and rounding errors that we need in the following involve the matrix condition number, κ, which is bounded according to $\kappa \leq c_\kappa h^{-2m}$ [8, Theorem 5.1], where c_κ is a constant. Note that since κ depends on k, then so must c_κ.

[1] If the cost of reducing different types of errors vary widely, then one could conceivably include weights when allocating the error-budget, but for simplicity we will not include that here.

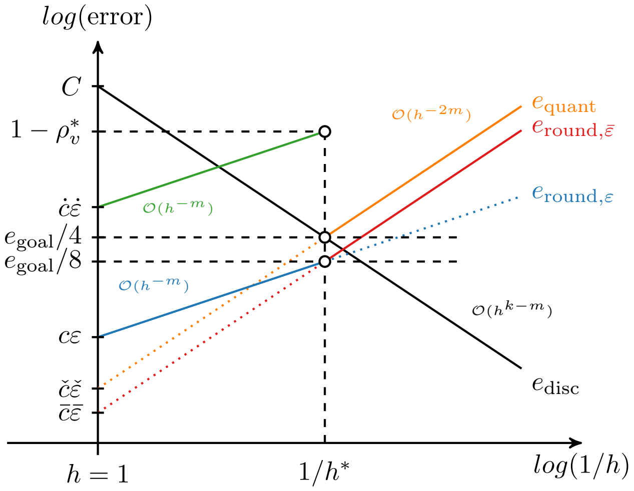

Fig. 1. *Error balance diagram. The color coding corresponds to that used in our pseudo code.*

Theorem 4.1 effectively states that $e_{\text{quant}} \leq \phi\breve{\boldsymbol{\varepsilon}}$. Assuming that $\underline{\kappa}\breve{\boldsymbol{\varepsilon}} \ll 1$ and $\underline{\kappa} \approx \kappa \gg \kappa^{\frac{1}{2}}$, it follows that $\phi \approx \kappa \leq c_\kappa h^{-2m}$ and therefore that $e_{\text{quant}} \lesssim \breve{c}\breve{\boldsymbol{\varepsilon}}h^{-2m}$ for $\breve{c} = c_\kappa$. This bound for e_{quant} is shown as the orange line in Figure 1. The constant $\breve{c}$ does not depend on $\breve{\boldsymbol{\varepsilon}}$, so assuming that we know $\breve{c}$, it follows that we can choose $\breve{\boldsymbol{\varepsilon}}$ to obtain the desired quantization error at $h = h^*$.

A bound for e_{round} is provided by the last term in (3.4), which by (3.3) is the sum of a term proportional to $\tau = \kappa^{\frac{1}{2}}\boldsymbol{\varepsilon}$ and a term proportional to $\bar{\tau} = \kappa\bar{\boldsymbol{\varepsilon}}$. Considering the case that $\bar{\tau} \ll \tau \ll 1$, then χ is bounded approximately by τ, yielding the estimate $e_{\text{round},\boldsymbol{\varepsilon}} \lesssim \tau \leq c\boldsymbol{\varepsilon}h^{-m}$ for $c = \sqrt{c_\kappa}$. For the opposing case where $\tau \ll \bar{\tau} \ll 1$, we can neglect τ in (3.3) to conclude that $e_{round,\bar{\boldsymbol{\varepsilon}}} < \gamma(1+\rho)(1+\boldsymbol{\varepsilon})\bar{m}_A^+\underline{\kappa}\boldsymbol{\varepsilon} \approx \gamma(1+\rho)\bar{m}_A^+\underline{\kappa}\boldsymbol{\varepsilon}$. If we again assume that $\underline{\kappa} \approx \kappa \gg \kappa^{\frac{1}{2}}$ and note that $\gamma \approx 1$ and $\rho < 1$, then $e_{round,\bar{\boldsymbol{\varepsilon}}} \approx \bar{c}\bar{\boldsymbol{\varepsilon}}h^{-2m}$, where $\bar{c} \approx 2\bar{m}_A^+c_\kappa$.

The two bounds for $e_{\text{round},\boldsymbol{\varepsilon}}$ and $e_{\text{round},\bar{\boldsymbol{\varepsilon}}}$ are shown in Figure 1 as blue and red lines, respectively. In mixed precision, $\boldsymbol{\varepsilon} > \bar{\boldsymbol{\varepsilon}}$, which means that $e_{\text{round},\boldsymbol{\varepsilon}}$ dominates the rounding error for small values of κ, while $e_{\text{round},\bar{\boldsymbol{\varepsilon}}}$ dominates for sufficiently large values of κ. To ensure that all errors are balanced, we choose $\boldsymbol{\varepsilon}$ and $\bar{\boldsymbol{\varepsilon}}$ such that $e_{\text{round},\boldsymbol{\varepsilon}} = e_{\text{round},\bar{\boldsymbol{\varepsilon}}} = \frac{1}{8}e_{\text{goal}}$ for $h = h^*$, meaning that $e_{\text{round}} = \frac{1}{4}e_{\text{goal}}$. Once again, assuming that we know c and $\bar{c}$, we can then determine $\boldsymbol{\varepsilon}$ and $\bar{\boldsymbol{\varepsilon}}$. While Figure 1 shows that $\bar{c}\bar{\boldsymbol{\varepsilon}} \leq \breve{c}\breve{\boldsymbol{\varepsilon}}$, it is important to note that this by itself does not necessarily imply that $\bar{\boldsymbol{\varepsilon}} < \breve{\boldsymbol{\varepsilon}}$. However, since $2e_{\text{round},\bar{\boldsymbol{\varepsilon}}} = e_{\text{quant}}$ at $h = h^*$, it follows that if we use the rough estimates for $\bar{c}$ and $\breve{c}$, then we expect that $\breve{\boldsymbol{\varepsilon}} = 4\bar{m}_A^+\bar{\boldsymbol{\varepsilon}}$, where $\bar{m}_A^+$ can be relatively large for high-order discretizations. This is consistent with the basic requirement that $\bar{\boldsymbol{\varepsilon}} \leq \breve{\boldsymbol{\varepsilon}}$.

The choice of $\dot{\boldsymbol{\varepsilon}}$ differs from the other precision levels in that we do not need to achieve $\frac{1}{4}e_{\text{goal}}$ accuracy in $\dot{\boldsymbol{\varepsilon}}$-precision, but instead simply need the V-cycle to be convergent in $\dot{\boldsymbol{\varepsilon}}$-precision. This means that we must have $\rho_v = \rho_v^* + \delta_{\rho_v} < 1$ at $h = h^*$. For sufficiently small $\dot{\tau}$, it follows from Theorem 3.2 that $\delta_{\rho_v} = \mathcal{O}(\dot{\tau})$, where $\dot{\tau} = \kappa^{\frac{1}{2}}\dot{\boldsymbol{\varepsilon}}$, so that $\delta_{\rho_v} \approx \dot{c}\dot{\boldsymbol{\varepsilon}}h^{-m}$ for some constant $\dot{c}$. This leads to the green line shown in Figure 1. In practice, we have not found a reliable way to determine $\dot{c}$ accurately, but simply choosing $\dot{\boldsymbol{\varepsilon}}$ such that $\dot{\tau} \ll 1$ appears to work well. Concretely, we choose

$\dot{\boldsymbol{\varepsilon}} = 0.1/\kappa^{\frac{1}{2}}$.

Balancing e_{iter} with the other three errors amounts to restricting N rather than any precisions. Our goal is to allow discretization accuracy to include a balanced quantization error, so (5.4) provides the choice for N that we need and it ensures that the *four* errors bounding e_{total} in (2.1) are approximately balanced.

We have thus far assumed a single target accuracy. To extend the estimates to progressive precision for the V-cycle, we can choose h^* (instead of e_{goal}) as the independent parameter and then repeat the exercise for every level in the multigrid hierarchy. This leads to $\delta_{\rho_v,j} \approx \dot{c}\dot{\boldsymbol{\varepsilon}}_j h_j^{-m} = 1 - \rho_v^*$ for $1 \leq j \leq l$. It then follows easily by considering $\delta_{\rho_v,j-1}/\delta_{\rho_v,j}$ that $\dot{\zeta}_j = \dot{\boldsymbol{\varepsilon}}_{j-1}/\dot{\boldsymbol{\varepsilon}}_j = (h_{j-1}/h_j)^m = \theta_j^m$. Thus, the precision coarsening factor for $\dot{\boldsymbol{\varepsilon}}$ is directly related to the mesh coarsening factor and, given $\dot{\boldsymbol{\varepsilon}}_j$ for any level, it is straightforward to compute $\dot{\boldsymbol{\varepsilon}}_j$ for the other levels.

For FMG, the values of $\boldsymbol{\varepsilon}$, $\bar{\boldsymbol{\varepsilon}}$, and $\breve{\boldsymbol{\varepsilon}}$ are also level dependent. Given h_j, the value for $\breve{\boldsymbol{\varepsilon}}_j$ follows easily by equating the bounds for e_{quant} and e_{disc}, which yields $\breve{\boldsymbol{\varepsilon}}_j \leq (C/\breve{c})h_j^{k+m}$. Similarly, we can equate $\frac{1}{2}e_{\text{disc}}$ with $e_{\text{round},\boldsymbol{\varepsilon}}$ and $e_{\text{round},\bar{\boldsymbol{\varepsilon}}}$ to get $\boldsymbol{\varepsilon}_j \leq \frac{1}{2}(C/c)h_j^k$ and $\bar{\boldsymbol{\varepsilon}}_j \leq \frac{1}{2}(C/\bar{c})h_j^{k+m}$. If $\theta = 2$, as is typical for geometric multigrid, this means that the sizes of the mantissas needed for $\dot{\boldsymbol{\varepsilon}}$ and $\boldsymbol{\varepsilon}$ grow with m and k bits per level, respectively, while the growth for both $\bar{\boldsymbol{\varepsilon}}$ and $\breve{\boldsymbol{\varepsilon}}$ is $k+m$ bits per level. Since m is typically one or two, this means that the precision required for $\dot{\boldsymbol{\varepsilon}}$ grows quite slowly while that for $\bar{\boldsymbol{\varepsilon}}$ and $\breve{\boldsymbol{\varepsilon}}$ can grow rather quickly for high-order discretizations.

To estimate the absolute precisions required for a given level, C, $\dot{c}$, c, $\bar{c}$, and $\breve{c}$ must all be determined. We do this empirically in Section 9 for our model problem.

7. Memory Requirements. One of the advantages of using mixed and progressive precision is that it generally reduces the overall memory consumption of the solver. Not only does this reduce the storage cost, but it also reduces the communication cost, which is often the bottleneck for multigrid solvers.

The progressive precision scheme can be applied to matrix-free methods and problems where the matrix can be represented by a small number of stencils. However, memory consumption is typically not a concern in those cases. In this section, we therefore consider frameworks where the matrix has to be stored explicitly. For simplicity, we consider a scalar problem on $[0,1]^d$ using a uniform grid with just one element on the coarsest level and a coarsening factor of 2 per dimension, d. Furthermore, we will only consider the storage requirement for the mantissa bits. The memory requirement in this case can be estimated by multiplying the maximum number of non-zeros per row of A_j (m_{A_j}) by the number of rows ($2^{d(j-1)}$) and the number of bits required per element (see below), and then summing over the level index (j). We focus on A_j here because the requirements for b_j, x_j, and other vector-valued variables are typically dominated by those for A_j. The analysis for P_j is very similar to that for A_j except for being more advantageous for progressive precision because P_j only needs to be stored at most in standard precision compared to high precision for A_j.

As discussed above, the storage precision, which is the basis for high-precision computation, requires $\breve{\boldsymbol{\varepsilon}}_j \leq (C/\breve{c})h_j^{k+m}$, where $h_j = 2^{-(j-1)}$. Assuming for simplicity that we want equality here and that $C = \breve{c}$, then the number of required mantissa bits is $\breve{\mathcal{B}}_j = -\log_2 \breve{\boldsymbol{\varepsilon}}_j = (k+m)(j-1)$. Similarly for low-precision computations, we arrive at $\dot{\mathcal{B}}_j = m(j-1)$.

Using the relationships in Section 6, the number of levels, ℓ, required to achieve the desired accuracy can be determined. If the computation is to be done in fixed precision, then $\breve{\boldsymbol{\varepsilon}}_\ell$-precision must be used for all the levels in the hierarchy. The total memory usage for either $\mathcal{IR}$-$\mathcal{V}$ or $\mathcal{FMG}$ (ignoring the possible saving due to symmetry of A_j) then becomes

$$\mathcal{M}_{\text{fix}} = m_A(k+m)(\ell-1)\sum_{j=1}^{\ell} 2^{d(j-1)}.$$

For progressive precision, we begin by considering just $\mathcal{IR}$-$\mathcal{V}$. In this case, *only* A_ℓ needs to be stored in high-precision since $\mathcal{IR}$ is only invoked at the finest level. On the other hand, $\mathcal{V}$ uses the entire hierarchy, so A_j must be stored in low precision for all j. This means that A_ℓ must be stored in both high and low precision while all others are only stored in low precision. The memory usage estimate for $\mathcal{IR}$-$\mathcal{V}$ is therefore

$$\mathcal{M}_{\text{prog}} = m_A\left((k+m)(\ell-1)2^{d(\ell-1)} + m\sum_{j=1}^{\ell}(j-1)2^{d(j-1)}\right).$$

The estimate for $\mathcal{FMG}$ is actually the same. The key insight here is that the outer $\mathcal{FMG}$ loop progresses from coarse to fine using $\mathcal{IR}$-$\mathcal{V}$ along the way. So only the current finest level in $\mathcal{IR}$-$\mathcal{V}$ needs A_j to be constructed and stored in high precision, and it can be discarded as the outer loop proceeds to finer levels. Thus, one allocation that can hold A_ℓ in high precision can be used for all A_j along the way to level ℓ.

It follows for either $\mathcal{IR}$-$\mathcal{V}$ or $\mathcal{FMG}$ that the memory usage of progressive precision relative to fixed precision is

$$\mathcal{M}_{\text{prog/fix}} = \frac{\mathcal{M}_{\text{prog}}}{\mathcal{M}_{\text{fix}}} = \frac{(k+m)(\ell-1)2^{d(\ell-1)} + m\sum_{j=1}^{\ell}(j-1)2^{d(j-1)}}{(k+m)(\ell-1)\sum_{j=1}^{\ell}2^{d(j-1)}}.$$

For large k, the first term in the numerator dominates and the ratio is approximately $1-2^{-d}$. Thus, for high-order elements, progressive precision leads to approximately 50% in memory savings for $d=1$ and approximately 12.5% for $d=3$. For low-order elements with $k=2$ and $m=1$ (corresponding to a second-order equation discretized with linear elements), simple evaluation of the ratio using Mathematica shows that the savings range between roughly 50% and 38% for $d=1$ and $d=3$, respectively. These numbers do not include the size of the indexing data for the sparse matrix data structures, which is independent of the precision. Also, we have not included the size of the exponent bits, so this analysis is only intended to show that the memory consumption for a mixed- and progressive-precision solver generally is expected to be lower than for a fixed-precision solver.

8. Model Problem.

To illustrate our rounding-error estimates, we consider the 1D biharmonic equation given by the following fourth-order ordinary differential equation (ODE) on $\Omega = (0,1)$ with homogeneous Dirichlet conditions on the boundary $\partial\Omega = \{0,1\}$:

$$\left\{\begin{array}{l} \text{Given } f \in L^2(\Omega), \text{ find } u \in C^4(\Omega) \text{ such that} \\ \qquad\begin{aligned} u'''' &= f \ \text{ in } \Omega \\ u = u' &= 0 \ \text{ on } \partial\Omega. \end{aligned} \end{array}\right.$$

This fourth-order model problem is useful because it leads to very ill-conditioned matrices with severe sensitivity to rounding errors. Although we do not present the results in detail here, we have also studied the 2D biharmonic and found the results to be qualitatively similar but computationally more expensive to obtain.

To discretize the ODE, we apply a standard Bubnov-Galerkin finite element method to its weak form based on the same trial and test spaces given by

$$\mathcal{U} = \left\{ u \colon \Omega \to \mathbb{R} \middle| u \in H^2(\Omega),\, u|_{\partial\Omega} = 0, \text{ and } u'|_{\partial\Omega} = 0 \right\}.$$

The variational form then arises via the L^2-projection of $u'''' = f$ onto an arbitrary test function $v \in \mathcal{U}$ followed by two applications of Green's first identity:

Given $f \in L^2(\Omega)$, find $u \in \mathcal{U}$ such that

$$a(u, v) = \ell(v)$$

for every $v \in \mathcal{U}$, where $a \colon \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ is the bilinear form defined by

$$a(u, v) = \int_\Omega u''v''\, d\omega$$

and $\ell \colon \mathcal{U} \to \mathbb{R}$ is the linear form defined by

$$\ell(v) = \int_\Omega fv\, d\Omega$$

for all $u, v \in \mathcal{U}$.

To discretize this variational form, we use H^2-conforming B-spline finite elements of order $k \geq 4$. (We do not consider splines of order $k = 3$ because, although they are smooth enough for this variational form, the jump discontinuities in the second derivative across quadratic spline elements hinders the optimal convergence rates in the L^2 norm [8, Sec. 2.2].) A set of n univariate B-spline basis functions of order k, $\{B_i^k\}_{i=1}^n$ is defined by first providing a knot vector $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+k}\}$, where $\xi_1 = 0$, $\xi_{n+k} = 1$, and $\xi_i \leq \xi_{i+1}$, $i = 1, 2, \ldots, n + k - 1,$. To facilitate strong enforcement of the Dirichlet boundary conditions, we use an open knot vector, i.e., a knot vector with the first and last knots repeated k times: $\xi_1 = \ldots = \xi_k = 0$ and $\xi_{n+1} = \ldots = \xi_{n+k} = 1$. The interior knots are distinct and, in fact, uniformly spaced. The Cox-de Boor recursion formula given below for $i = 1, 2, \ldots, n$ uses this open knot vector to define the univariate B-spline basis for intermediate $\xi \in (0, 1)$.

$$B_i^k(\xi) = \frac{\xi - \xi_i}{\xi_{i+k} - \xi_i} B_i^{k-1}(\xi) + \frac{\xi_{i+k+1} - \xi}{\xi_{i+k+1} - \xi_{i+1}} B_{i+1}^{k-1}(\xi),\; B_i^0(\xi) = \left\{ \begin{array}{ll} 1, & \xi \in [\xi_i, \xi_{i+1}) \\ 0, & \text{elsewhere} \end{array} \right.$$

For notational ease, we henceforth drop the superscript in B^k that denotes the explicit k-dependence on the B-spline basis.

B-spline h-refinement is done by *knot insertion*, where new equi-spaced interior knots $\frac{1}{2}(\xi_i + \xi_{i+1})$, $k - 1 < i < n$, are added to the original knot vector and the new set of basis functions are computed accordingly. Note that although knot insertion affects neighboring basis functions, it is still a relatively local process, which a variety of knot-insertion strategies exploit. Knot insertion also enables direct construction of prolongation and restriction operators that are naturally transposes of each other, and together with the system matrices they satisfy the Galerkin condition. See [7] for a discussion on spline basis functions and relevant algorithms.

The B-spline basis functions allow us to define the finite-dimensional trial space and test space, $\mathcal{U}_h \subset \mathcal{U}$, used for our Galerkin discretization:

$$\mathcal{U}_h = \left\{ u_h \in \mathcal{U} \middle| u_h = \sum_i u_i B_i(\xi) \right\},$$

where u_i are the so-called control points. The discrete variational form of the ODE is then expressed as

$$\begin{cases} \text{Given } f \in L^2(\Omega), \text{ find } u_h \in \mathcal{U}_h \text{ such that} \\ \qquad\qquad a(u_h, v_h) = \ell(v_h) \\ \text{for every } v_h \in \mathcal{U}_h. \end{cases}$$

Obtaining the discrete solution amounts to solving linear system (3.1) with $A_{ij} = a(B_j, B_i)$, $\xi_i = u_i$, and $b_i = \ell(B_i)$.

To assess the efficacy of iterative refinement, we consider the exact solution field:

$$u = 1 - \cos 2\pi\xi. \tag{8.1}$$

The corresponding forcing function is easily obtained by applying the differential operator, yielding:

$$f = -16\pi^4 \cos 2\pi\xi.$$

This forcing function along with the known solution field u enable an exact measure of the total error in our numerical results. We have experimented with other solution fields, but not found anything leading to different conclusions than what we present below.

9. Numerical Experiments.

We validate the theory here by studying convergence under grid refinement for the model problem. Accordingly, the multigrid solvers we study coarsen only in the mesh size as opposed to the degree of the basis functions. We use Matlab R2019a and the Advanpix toolbox for our experiments. The Advanpix toolbox allows for variable precision computations, although the interface only allows the number of *decimal* digits, d, to be specified. For our "exact" computations, we use 34 decimal digits of precision, which corresponds to 113 bits. While slightly more than the 112 bits in IEEE quad precision, we nevertheless refer to $d = 34$ as "quad precision" in what follows. For this precision level, Advanpix provides 15 bits for the exponent, which is consistent with IEEE quad precision. For *all* other levels of precision, Advanpix provides 64 bits for the exponent. The numerical results reported here are therefore unaffected by the limited dynamic range typically encountered in low precision environments. This aspect of the precision environment corresponds with the theory, which assumes that all computations stay in the dynamical range. Throughout, we use FP16 $= 2^{-11}$, FP32 $= 2^{-24}$, FP64 $= 2^{-53}$, and FP128 $= 2^{-112}$ to denote one unit in last place (ulp) for half, single, double, and quad precision, respectively.

All stiffness matrices and forcing vectors are formed and assembled in quad precision and, as such, are susceptible to quantization (and other) errors at this precision level. The exact solution x_h associated with u_h for a given mesh size h is computed using A_h and b_h formed in quad precision followed by a solve in quad precision.

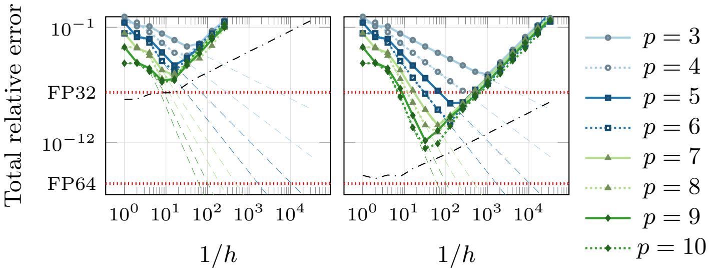

Fig. 2. *The total error in the energy norm when using FMG in fixed precision with* 7 *digits (left) and* 15 *digits (right) corresponding roughly to single and double precision, respectively. Additionally, the discretization error for each polynomial degree is shown using dashed lines, while the black dash-dotted line shows* $\| \mathrm{fl}(x_h) - x_h \|_{A_h}$. *This latter quantity is the smallest error one can hope to achieve because it is obtained by simply rounding the exact solution to the chosen precision.*

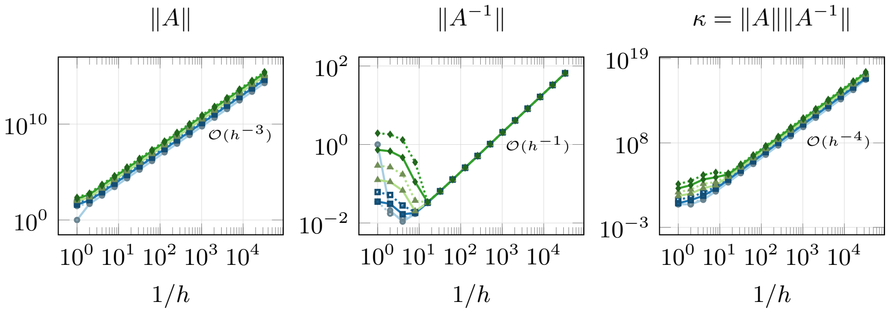

Fig. 3. *Asymptotically, the condition number grows as* $\mathcal{O}(h^{-4})$ *as expected except for the first* 5 *levels. For these coarse levels,* $\|A^{-1}\|$ *shows a distinct pre-asymptotic behavior due to the influence of the boundary conditions. Each plot shows curves for polynomial degrees from* $p = 3$ *(bottom) to* $p = 10$ *(top). The legend is the same as for Figure 2.*

(While this solution is not truly exact of course, its error is insignificant compared to the other errors we consider.) All exact solutions are obtained using Matlab's direct solver with quad precision. The stiffness matrices and forcing vectors pertaining to lower-precision quantizations are obtained by simply rounding their quad-precision counterparts to the desired precision, $\breve{\varepsilon}$. We compute $\breve{x}_h$ from the lower-precision coefficients by first adding trailing zeros to all numbers to extend them back to quad precision and then solving the resulting system in quad precision. This ensures that the difference between $\breve{x}_h$ and x_h is primarily due to quantization rather than algebraic errors. The algebraic solution, $\tilde{x}_h$, is obtained as the solution of $\breve{A}_h x_h = \breve{b}_h$, where the solvers and precision levels used are specified below for each experiment. Given the true solution, u, from (8.1), along with x_h, $\breve{x}_h$, and $\tilde{x}_h$, we evaluate the energy norm of the errors shown in (2.1) in quad precision using k^2 quadrature points per element. For familiarity's sake, we use $p = k - 1$ to refer to the polynomial degree of the finite element basis functions.

We begin by confirming in Figure 2 that $\mathcal{FMG}$ in fixed precision is susceptible to multiple types of errors that prevent it from obtaining discritization-error accuracy. For reference, we also show the error from simply rounding the exact solution to the available fixed precision. This is the smallest error we can hope to achieve for a given fixed precision, but $\mathcal{FMG}$ is clearly unable to achieve this level of accuracy except

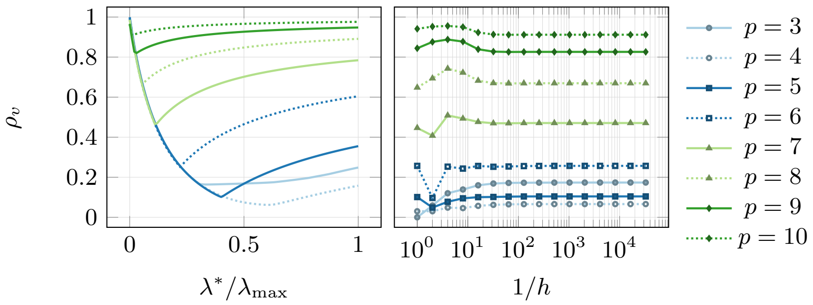

FIG. 4. *The convergence rate of a 2^{nd}-order Chebyshev smoother as a function of percentage of the full spectrum that is targeted (left). Given the optimal fraction of the spectrum to target, the right plot shows the convergence rate as a function of mesh size. All computations are done using 34 digits. For the left plot, $h = 1/64$. The curves for $p = 3$ are slight outliers because the coarsest level in this case contains no degrees of freedom after the boundary conditions have been imposed.*

possibly for very high order basis functions.

Next, we graph the norm and condition number of A_h in Figure 3. The important observation is that while the asymptotic behavior follows the theory, a pre-asymptotic region exists where the boundaries influence the results. As a consequence, we generally do not expect to see optimal results until after level 4. We also note that $\|A\|$ clearly depends on p, which also carries into κ. More specifically, $\|A\|$ is approximately proportional to p^3h^{-3}. Some of the other quantities that show up in the theory are given by $\kappa(P^tP) = 2^p$, $m_A = 2p+1$, and $m_P = p+2$ for all h past the pre-asymptotic region.

Throughout, we use V$(1, 0)$-cycles and a 2^{nd}-order Chebyshev smoother based on $\check{D}_h^{-1}\check{A}_h$, where $\check{D}_h$ is the diagonal of $\check{A}_h$. For simplicity, our initial experiments do *not* use progressive precision. On the coarsest level consisting of a single element, we use a single sweep of the smoother. The cost of a direct solver at that level is insignificant, but also not necessary. The Chebyshev smoother depends on knowing the largest eigenvalue of $\check{D}_h^{-1}\check{A}_h$ as well as what percentage of the spectrum to target. We compute the largest eigenvalue using Matlab's standard `eigs`-function applied to $\check{A}_h$ in double precision. For each polynomial degree, we then determine the lower end of the spectrum, λ^*, to target by evaluating the convergence rate for a range of different values and picking the one that produces the smallest ρ_v. The results are shown in Figure 4. Clearly, a 2^{nd}-order Chebyshev smoother is not very effective for high polynomial degrees, but designing effective smoothers is not our aim here, and our theory does not depend on the quality of the smoother.

The convergence rate, ρ_v, is computed in two steps. First, the error propagation matrix, V, is constructed column by column by applying one V-cycle with a zero initial guess to each of the canonical basis vectors. Next, $\rho_v = \|V\|_A$ is computed as the square root of the largest generalized eigenvalue of $V^TAVx = \lambda Ax$. All of this is done in quad precision. We could have approximated ρ_v by solving $Ax = 0$ with a random initial guess $x^{(0)}$ and choosing the largest value over many iterations i of $\|x^{(i+1)}\|_A/\|x^{(i)}\|_A$. However, our eigenvalue approach determines the worst case for the energy convergence rate more effectively.

Next, we study the algebraic error for $\mathcal{IR}$-$\mathcal{V}$ with and without mixed precision. For simplicity, we set $\dot{\boldsymbol{\varepsilon}} = \boldsymbol{\varepsilon}$ because it allows us to isolate the effects of the other precision levels. Figure 5 shows that the error is initially dominated by iteration error

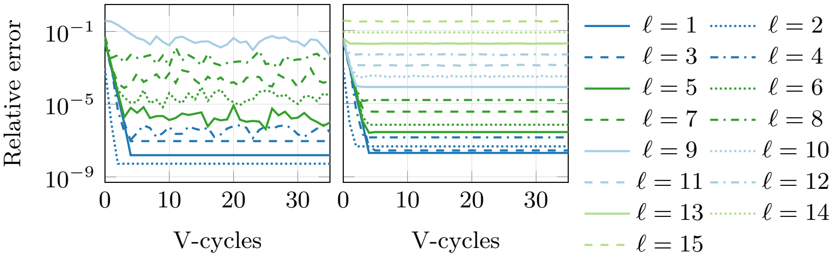

Fig. 5. *The relative algebraic error in the energy norm, i.e., $e_{iter} + e_{round}$, when solving $\check{A}_h x_h = \check{b}_h$ for $p = 4$ with a random initial guess using $\mathcal{IR}$-$\mathcal{V}$ and 1 to 15 levels in the hierarchy. The figure on the left illustrates fixed precision using 7 decimal digits, while the figure on the right illustrates mixed precision using 7 and 34 decimal digits for $\boldsymbol{\varepsilon}$ and $\bar{\boldsymbol{\varepsilon}}$, respectively. For both of these experiments, $\dot{\boldsymbol{\varepsilon}} = \boldsymbol{\varepsilon}$ and $\breve{\boldsymbol{\varepsilon}} = \bar{\boldsymbol{\varepsilon}}$. Initially, e_{iter} dominates until the limiting accuracy is reached. Furthermore, the error generally increases with the grid resolution, and for $\ell > 9$ the fixed precision solver fails to converge.*

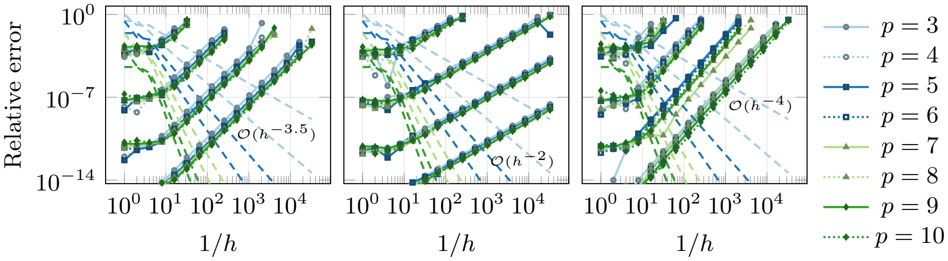

Fig. 6. *The relative rounding error in the energy norm when using fixed precision (left) and mixed precision (middle) as well as the relative quantization error (right). In all three plots, the energy norm of the relative discretization error is shown using dashed lines. The four groups of solid and dotted lines (top to bottom) in each plot correspond to using 3, 7, 11, and 15 digits for $\bar{\boldsymbol{\varepsilon}}$ (left), $\boldsymbol{\varepsilon}$ (middle), and $\breve{\boldsymbol{\varepsilon}}$ (right). This is a representative sample of the full data set, which included results for all precisions between 3 and 34 digits. For fixed precision (left), $\boldsymbol{\varepsilon} = \bar{\boldsymbol{\varepsilon}} = \breve{\boldsymbol{\varepsilon}}$, while for mixed precision (middle), $\bar{\boldsymbol{\varepsilon}}$ and $\breve{\boldsymbol{\varepsilon}}$ both use 34 digits. Referring back to Section 2, these results are compared to $\breve{x}_h$, which is computed as described earlier in this section. Finally, for the quantization error experiment (right), $\boldsymbol{\varepsilon} = \bar{\boldsymbol{\varepsilon}} = \breve{\boldsymbol{\varepsilon}}$, but the results are compared to the true solution, x_h. All the precisions are chosen to ensure that the error is dominated by the choice of $\bar{\boldsymbol{\varepsilon}}$, $\boldsymbol{\varepsilon}$, and $\breve{\boldsymbol{\varepsilon}}$, respectively. All plots show the max relative error over the last 50 iterations when solving the model problem using 1000 V-cycles. For high precisions and higher polynomial degrees (for which the convergence rate deteriorates), more V-cycles would be necessary to recover the true error.*

before eventually being dominated by rounding error. In fixed-precision, the rounding error never stabilizes, which illustrates why it can be difficult to develop a reliable stopping criterion, but this is much less of an issue in mixed precision. It is also evident that the limiting accuracy, χ, depends on the number of levels in the hierarchy. This is illustrated further in Figure 6 (left and middle), where the relative algebraic error after 1000 V-cycles is shown as a function of $1/h$. As theory predicts, the relative algebraic error grows faster for fixed than for mixed precision, which illustrates the benefit of mixed-precision $\mathcal{IR}$-$\mathcal{V}$. By comparing algebraic and discretization errors, Figure 6 also confirms that, in the absence of progressive precision, multigrid is ultimately dominated by rounding errors.

As predicted by our theory, the growth of rounding error for mixed precision is proportional to $\kappa^{1/2}(A)$, or equivalently $\mathcal{O}(h^{-2})$, while the observed growth for fixed

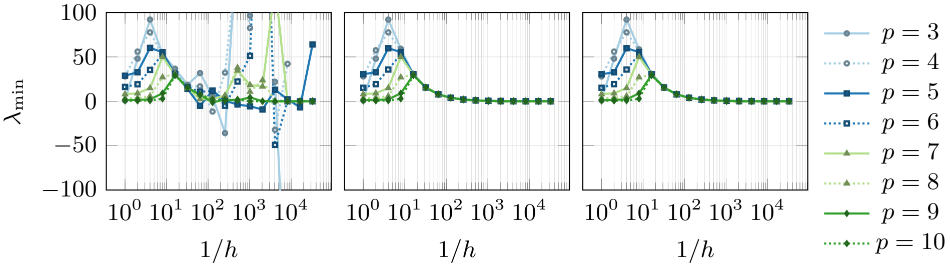

Fig. 7. *The smallest eigenvalue of $\check{A}_h$ when $\check{A}_h$ is quantized to $\dot{\boldsymbol{\varepsilon}}$ (left), $\boldsymbol{\varepsilon}$ (middle), and $\check{\boldsymbol{\varepsilon}}$ (right) precision. For $\dot{\boldsymbol{\varepsilon}}$-precision, it is clear that $\check{A}_h$ becomes indefinite for fine levels, and it should be noted that the smallest eigenvalue can be orders of magnitude below zero, which means that no small diagonal pertubation is likely to recover definiteness. However, it should also be noted that we do not encounter any indefinite matrices in $\boldsymbol{\varepsilon}$ or $\check{\boldsymbol{\varepsilon}}$ precision. Thus, we can reasonably estimate κ_j in $\boldsymbol{\varepsilon}$ precision using the Lanczos method, for example.*

precision is $\mathcal{O}(h^{-3.5})$, which is slightly better than the rate predicted by theory. Also shown in Figure 6 is the quantization error obtained by solving $\check{A}_h x_h = \check{b}_h$ "exactly" for various $\check{\boldsymbol{\varepsilon}}$ and comparing the result to u_h. As predicted by Theorem 4.1, this error grows as $\mathcal{O}(h^{-4})$.

In Figure 7, we confirm that quantization of A_h to $\dot{\boldsymbol{\varepsilon}}$-precision can cause it to become indefinite and it can, in fact, become very indefinite for fine levels.

To implement progressive precision FMG, we need to establish the precisions used at each level, which requires estimating the values for the constants C, c, $\bar{c}$, $\check{c}$, $\dot{c}$ discussed in Section 6. The choice of $\dot{c}$ was discussed in Section 6, while the values for c, $\bar{c}$, and $\check{c}$ can all be estimated based on the data shown in Figure 6. In Section 6, we established bounds for $e_{\text{round},\boldsymbol{\varepsilon}}$, $e_{\text{round},\bar{\boldsymbol{\varepsilon}}}$, and e_{quant}. Here, we treat those expressions as strict equalities to account for the worst case, which yields $e_{\text{round},\boldsymbol{\varepsilon}} = c\boldsymbol{\varepsilon}h^{-m}$, $e_{\text{round},\bar{\boldsymbol{\varepsilon}}} = \bar{c}\boldsymbol{\varepsilon}h^{-2m}$, and $e_{\text{quant}} = \check{c}\check{\boldsymbol{\varepsilon}}h^{-2m}$. Generically, and with a slight abuse of notation, this gives us $e = c\boldsymbol{\varepsilon}h^{-\alpha}$, where e is one of the errors and c, $\boldsymbol{\varepsilon}$, and α are the corresponding constant, precision, and exponent, respectively. It then follows that $\log(e/\boldsymbol{\varepsilon}) = -\alpha\log(h) + \log(c)$. From this expression, we can compute a linear least squares estimate for $\log(c)$ and α using the data points in Figure 6 past the pre-asymptotic region (which in practice we take to be where $1/h > 4$). While Figure 6 only shows data for 4 different precisions, we have conducted the experiments for all precisions between 3 and 15 decimal digits, and we use the data from all the experiments for the least squares estimates except that we omit the data from the pre-asymptotic region ($1/h \leq 16$). The estimates for c, $\bar{c}$, and $\check{c}$ are shown in Figure 8.

Unfortunately, it is computationally quite expensive to obtain all the data required for these least squares estimates. As an alternative, given $c_\kappa \geq \kappa h^{2m}$, we can estimate all the constants quite cheaply by noting from Section 6 that $\check{c} = c_\kappa$, $\bar{c} \lesssim 4\bar{m}_A^+ c_\kappa$, and $c = \sqrt{c_\kappa}$. Furthermore, from Figure 3, we see that this can be estimated reliably as soon as we get past the pre-asymptotic region. In practice, we therefore only have to estimate the condition number for a few small matrices. Technically, we have a lower bound for c_κ that is quite a bit higher in the pre-asymptotic region. However, the bounds for c, $\bar{c}$, and $\check{c}$ based on c_κ are rather conservative to begin with, so we find in practice that it is safe to ignore this technicality and use the asymptotic value of c_κ for all h. In fact, Figure 8 shows that the constants obtained using c_κ can be several orders of magnitude larger than the least squares estimates. This may seem

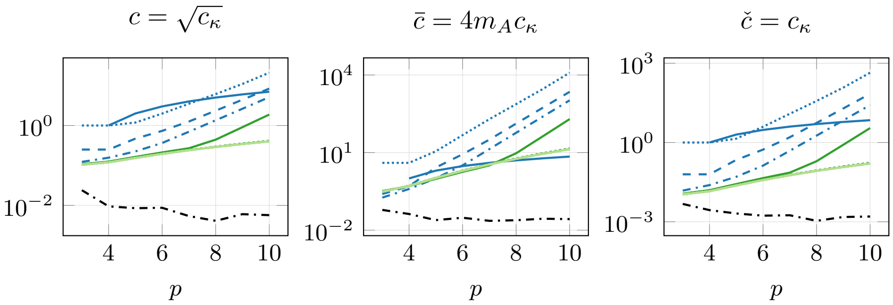

Fig. 8. *Estimates for c, $\bar{c}$, and $\check{c}$. The legend here is the same as for Figure 5 with the colored lines representing estimates based on c_κ. After level 5, all the lines are indistinguishable as the estimates have converged. The dash-dotted line in black is the linear least squares estimate based on the data partially shown in Figure 6. The graphs here suggest that the true constants depend rather weakly (and inversely) on p, while the estimates based on c_κ suggest a significant growth with p.*

p	3	4	5	6	7	8	9	10
Theoretical N	2	2	2	4	8	17	38	85
Minimal N	1	1	1	2	4	9	28	50

Table 1
Number of V-cycles required inside $\mathcal{FMG}$ as a function of the polynomial degree, according to the theory in (5.4) and given the convergence rates obtained in Fig. 4. Also shown is the smallest number of V-cycles for which $\mathcal{FMG}$ actually converges when using the constants estimated from c_κ. This shows that the theory is somewhat conservative, but mostly for high polynomial degrees where the convergence rate of the smoother is poor. The minimal number of V-cycles increases by one in a few cases if the smaller constants obtained from least squares estimation are used instead.

concerning, but each order of magnitude translates to using one additional decimal digit for the corresponding precision level, and this fixed amount of extra precision is relatively insignificant for the higher levels that tend to account for most of the computational cost. The entire approach for computing the constants is captured in Algorithm 9.1. Also included is the computation of N based on (5.4), with the results shown in Table 1.

It remains to estimate C. Given the discretization error as plotted in Figures 2 and 6, C can easily be obtained by linear regression. However, those curves are based on computations in exact arithmetic and knowledge of the exact solution. Fortunately, we can estimate C in the course of running $\mathcal{FMG}$ based on the strong approximation property in (3.7). This approach ultimately leads us to the progressive FMG algorithm outlined in Algorithm 9.2, where orange is used for computations in $\breve{\varepsilon}$ precision. Developing all the details to deal robustly with any pre-asymptotic region is beyond the scope of this paper. Still, this algorithm is notable by starting out in low precision and only advancing to higher precision as necessary in order to achieve the specified error goal.

Algorithm 9.1 Compute constants for progressive FMG

```
Input: $A$, $p$, $m$, $\theta$, tol $< 1$, $\dot{\tau}_{\text{tol}} < 1$.
$q \leftarrow p + 1 - m$                                ▷ Compute $q$
$\kappa_0 \leftarrow \|A_0\| \|A_0^{-1}\|$                ▷ Compute condition number of $A_0$
$j \leftarrow 0$                                        ▷ Initialize level counter
repeat
    $j \leftarrow j + 1$                                ▷ Update level counter
    $\kappa_j \leftarrow \|A_j\| \|A_j^{-1}\|$            ▷ Compute condition number of $A_j$
until $\left| \frac{\kappa_j}{\kappa_{j-1}} \theta^{-2m} - 1 \right| <$ tol     ▷ Stop if in asymptotic region
$c_\kappa \leftarrow \kappa_j \theta^{-2mj}$              ▷ Compute $c_\kappa$
$c \leftarrow \sqrt{c_\kappa}$                           ▷ Compute $c$
$\bar{c} \leftarrow 4 m_A c_\kappa$                       ▷ Compute $\bar{c}$
$\check{c} \leftarrow c_\kappa$                           ▷ Compute $\check{c}$
$\dot{c} \leftarrow \dot{\tau}_{\text{tol}} / \sqrt{c_\kappa}$   ▷ Compute $\dot{c}$
Compute $\rho$ for level $j$                            ▷ Determine asymptotic convergence factor.
$N \leftarrow (\log_2(5) + q \log_2(\theta)) / (|\log_2(\rho)|)$   ▷ Compute theoretical number of V-cycles
return $(c, \bar{c}, \check{c}, \dot{c}, N)$                ▷ Return constants
```

Algorithm 9.2 `Progressive FMG`$(1,0)$`-Cycle` $(\mathcal{PFMG})$

```
Input: $\mathcal{L}$, $f$, $m$, $k$, $\theta$, $c$, $\bar{c}$, $\check{c}$, $\dot{c}$, $N \geq 1$, $e_{\text{goal}} < 1$
$q \leftarrow k - m$
$x_0 \leftarrow 0$
$j \leftarrow 1$
loop
    $(\boldsymbol{\varepsilon}, \bar{\boldsymbol{\varepsilon}}, \breve{\boldsymbol{\varepsilon}}, \dot{\boldsymbol{\varepsilon}}) \leftarrow$ComputePrecisions$(c, \bar{c}, \check{c}, \dot{c}, j)$   ▷ Update all precision levels
    $(\check{A}_j, \check{b}_j, \check{P}_j, h_j) \leftarrow$ Discretize$(\mathcal{L}, f, k, j)$   ▷ Discretize PDE at level $j$
    $x_j \leftarrow \check{P}_j x_{j-1}$                  ▷ Interpolate from previous level
    $i \leftarrow 0$                                    ▷ Initialize $\mathcal{IR}$
    while $i < N$ do
        $r_j \leftarrow \check{A}_j x_j - \check{b}_j$      ▷ Update $\mathcal{IR}$ residual and round
        $y_j \leftarrow \mathcal{V}(\check{A}_j, r_j, \check{P}_j, j)$   ▷ Compute correction by $\mathcal{V}$
        $x_j \leftarrow x_j - y_j$                      ▷ Update approximate solution of $A_j x_j = b_j$
        $i \leftarrow i + 1$                            ▷ Increment $\mathcal{IR}$ cycle counter
    end while
    if $j > 4$ then                                     ▷ In asymptotic region ?
        $C \leftarrow \frac{\|\check{P}_j x_{j-1} - x_j\|_{\check{A}_j}}{h_{j-1}^q \|x_j\|_{\check{A}_j}}$   ▷ Estimate discretization constant
        $\ell \leftarrow \left\lceil \frac{1}{q} \log_\theta \left( \frac{C}{e_{\text{goal}}} \right) \right\rceil$   ▷ Compute required number of levels
        if $\ell \leq j$ then
            return $x_j$                                ▷ Return solution with error less than $e_{\text{goal}}$
        end if
    end if
    $j \leftarrow j + 1$
end loop
```

Using the proposed algorithm, the precision requirements and the accuracy actually achieved are shown in Figure 9. Most importantly, we observe that $\mathcal{PFMG}$ does in fact achieve discretization-error accuracy. However, we also note that the use of standard floating point types available in hardware can be surprisingly restrictive in terms of h. In Figure 10, we extrapolate the results to second-order PDEs since these are quite common in real applications. For high-order basis functions, the order of the PDE does not matter much, and we notice that while there is a difference between $\bar{\boldsymbol{\varepsilon}}$ and $\breve{\boldsymbol{\varepsilon}}$, it is relatively insignificant in this regime. For lower-order basis functions, a final observation is that $\breve{\boldsymbol{\varepsilon}} \ll \boldsymbol{\varepsilon}$, meaning that $\check{A}_h$ used in the residual computation in

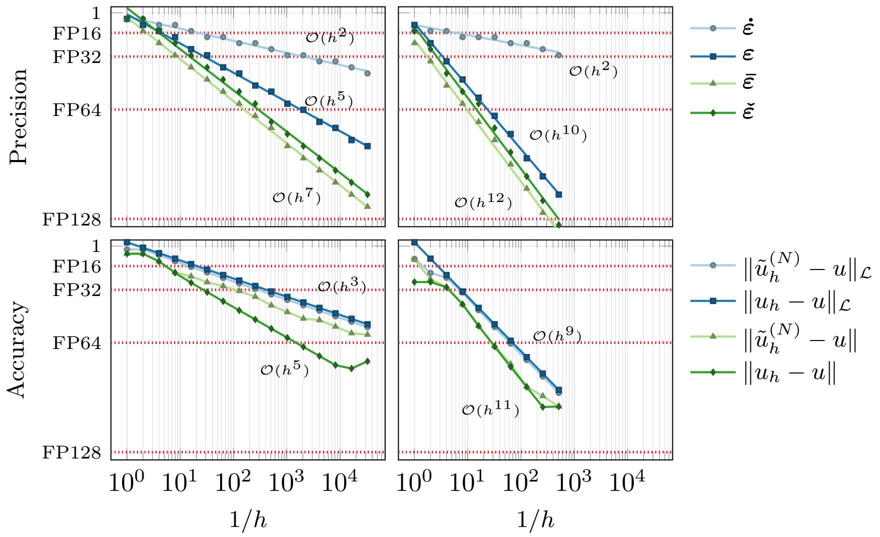

Fig. 9. *Precision requirements for progressive precision FMG in order to reach discretization error accuracy for the model problem (top), and the actual accuracy obtained compared to the true discretization error (bottom). The graphs shown here are for $p = 4$ (left) and $p = 10$ (right). For reference, we include the L^2 error in the accuracy plots, and notice that we generally do not achieve optimal convergence in the L^2-norm. However, for $p = 10$, the large number of V-cycles we use due to the conservative nature of the estimate for N probably accounts for achieving close to optimal results in the L^2-norm too.*

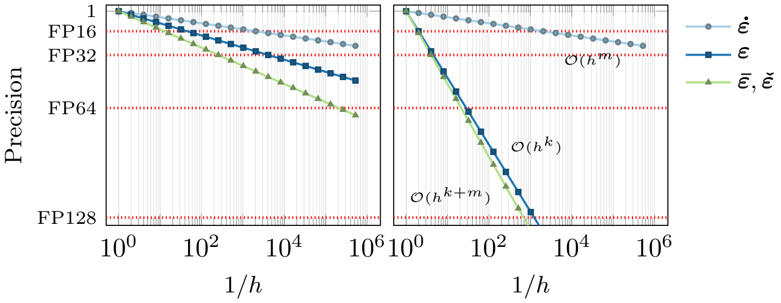

Fig. 10. *Predicted precision requirements for progressive precision FMG for a second-order PDE (assuming that $C = \dot{c} = c = \bar{c} = \check{c} = 1$ for simplicity). The graphs shown here are for $p = 1$ (left) and $p = 10$ (right). For $p = 1$, double precision suffices up to $1/h = 2^{18}$, while, for $p = 10$, anything beyond $1/h = 2^4$ is contaminated by quantization errors when using double precision. Since quantization errors do not depend on the choice of solver, these limits apply to any kind of solver and not just FMG. Notice also that these limits on h apply to problems in all dimensions.*

$\mathcal{IR}$ must be of sufficiently high precision.

10. Conclusions. This paper has successfully shown the potential of using progressive precision multigrid methods for solving linear elliptic PDEs to arbitrary accuracy given sufficient but parsimoniously chosen precisions in all computations. Compared to fixed precision, the accuracy is obtained while using up to 50% less memory. The key to this success on one hand is the observation that quantization errors play a critical role that must be accounted for. On the other hand is the observation that the V-cycle is very resilient and will work correctly even when it is being run in such low precision that the matrices involved may become indefinite simply from rounding them to working precision. The limitations introduced by quantization error ultimately lead to fairly strict limitations on the grid size that can be used to discretize the PDE for any given precision budget. This is worth noting because many computations in practice are limited to standard IEEE double precision at the high end. Insofar as the PDE solution is sufficiently smooth, higher-order elements generally allow for higher accuracy, and as p approaches infinity, it is easy to see from Fig. 1 that the accuracy will approach $4\check{c}\check{\varepsilon}$. However, even for moderate values of p, the benefit of the improvement in accuracy obtained by increasing p further may be outweighed by the additional cost of the higher-order method.

In order to choose all the precision levels, we have introduced a heuristic that balances all the different types of errors. This approach ensures that we avoid "overcomputation", where one type of error is reduced only to be swamped by some other type of error. Assuming that one has appropriate bounds, this idea can easily be generalized to include other types of errors such as those from matrix assembly or even modeling errors. Given an appropriate performance model, it can also be generalized to account for different costs associated with different types of errors. Both of these extensions are interesting topics for future work. Other topics for future work include the extension of the ideas presented here to algebraic multigrid, and a proper analysis of any effects due to overflow or underflow.

Appendix A. Summary of notation. The basic quantities and abbreviations used throughout the paper are as follows:

PDE	partial differential equation
ODE	ordinary differential equation
RHS	right-hand side
SAP	strong approximation property
$\mathcal{IR}$	iterative refinement algorithm
$\mathcal{V}$	V-cycle algorithm
$\mathcal{FMG}$	full multigrid algorithm
$\mathcal{PFMG}$	progressive full multigrid algorithm
$\boldsymbol{\varepsilon}$	"standard" precision unit roundoff
$\bar{\boldsymbol{\varepsilon}}$	"high" precision unit roundoff
$\dot{\boldsymbol{\varepsilon}}$	"low" precision unit roundoff
$\mathrm{fl}(x \circ y)$	finite-precision result of the operation $x \circ y$
$\mathcal{B}$	number of bits of precision
δ	perturbation of a scalar or vector due to finite precision
Δ	perturbation of a matrix due to finite precision
$n = n_j$	dimension of the fine level
$n_c = n_{j-1}$	dimension of the coarse level
ℓ	number of levels in the hierarchy
$A \in \mathbb{R}^{n \times n}$	system matrix
$D \in \mathbb{R}^{n \times n}$	diagonal of A
$M \in \mathbb{R}^{n \times n}$	approximation of A^{-1}
$P \in \mathbb{R}^{n \times n_c}$	prolongation or interpolation matrix
$G \in \mathbb{R}^{n \times n}$	error propagation matrix for relaxation
$T \in \mathbb{R}^{n \times n}$	error propagation matrix for coarse-level correction
$V \in \mathbb{R}^{n \times n}$	error propagation matrix for a V-cycle
$Ax = b$	target problem
P^t	transpose of P

$\|\cdot\|$	Euclidean norm
$\|\cdot\|_A = \|A^{\frac{1}{2}}\cdot\|$	energy norm
$\kappa(\cdot)$	condition number
$\kappa = \kappa(A)$	condition number of A
m_A	bound on the number of nonzero row entries of A
m_P	bound on the number of nonzero row entries of P
N	number of $\mathcal{IR}$ cycles
ρ	generic energy convergence factor
ρ_{ir}	energy convergence factor for $\mathcal{IR}$
ρ_v^*	exact energy convergence factor for $\mathcal{V}$
$\rho_v = \rho_v^* + \delta_{\rho_v}$	computed energy convergence factor for $\mathcal{V}$
χ	limit accuracy in energy
h	pseudo mesh size
θ	pseudo mesh-refinement factor
$\dot{\zeta}$	precision coarsening factor
δ_M	rounding error in computing Mz
α_M	constant in $\mathcal{O}(\dot{\boldsymbol{\varepsilon}}\|z\|)$ bound on $\|\delta_M\|$
$2m$	order of the PDE
k	finite element polynomial order
p	finite element polynomial degree
q	mesh-size exponent of coarse-level approximation order
$\phi_{h,i}$	finite element basis functions
e_{disc}	energy discretization error
e_{fl}	floating-point error
e_{quant}	quantization error
e_{alg}	algebraic error
e_{iter}	iteration error
e_{round}	rounding error
e_{goal}	desired error level
h^*	target mesh size for obtaining desired error level
c	constant in $\mathcal{O}(\boldsymbol{\varepsilon}h^{-m})$ bound on rounding error
c_κ	constant in $\mathcal{O}(h^{-2m})$ bound on condition number
$\dot{c}$	constant in $\mathcal{O}(\dot{\boldsymbol{\varepsilon}}h^{-m})$ bound on V-cycle convergence factor
$\check{c}$	constant in $\mathcal{O}(\check{\boldsymbol{\varepsilon}}h^{-2m})$ bound on quantization error
$\bar{c}$	constant in $\mathcal{O}(\bar{\boldsymbol{\varepsilon}}h^{-2m})$ bound on rounding error
Ω	PDE domain
$\mathcal{U}$	PDE test and trial space
$\mathcal{U}_h$	finite-dimensional test and trial space
$a(u,v) = \ell(v)$	weak form of the PDE
B_i^k	B-spline basis functions of order k
ξ_i	B-spline knots
Ξ	B-spline knot vector
u_i	B-spline control points

Additionally, the following convenience parameters are used:

$$\psi = \|\|A\|\|, \;\; \underline{\kappa} = \psi\|A^{-1}\|, \;\; \bar{m}_A^+ = \frac{m_A + 1}{1 - (m_A+1)\bar{\boldsymbol{\varepsilon}}}, \;\; \dot{m}_P^+ = \frac{m_P}{1 - m_P\dot{\boldsymbol{\varepsilon}}},$$

$$\dot{m}_A^+ = \frac{m_A + 1}{1 - (m_A+1)\dot{\boldsymbol{\varepsilon}}}, \;\; \dot{\tau} = \kappa^{\frac{1}{2}}\dot{\boldsymbol{\varepsilon}}, \;\; \tau = \kappa^{\frac{1}{2}}\boldsymbol{\varepsilon}, \;\; \bar{\tau} = \kappa\bar{\boldsymbol{\varepsilon}}, \;\; \gamma = \frac{\kappa^{\frac{1}{2}} + \underline{\kappa}}{\kappa},$$

$$\sigma \geq (1+\dot{\boldsymbol{\varepsilon}}) \max\{\alpha_M\|A\|, \psi\alpha_M, \psi\|M\|\}, \;\; \vartheta = \min_{1\leq j\leq \ell}\{\theta_j \dot{\zeta}_j^{-\frac{1}{m}}\},$$

$$\mu = 3(\max_{1\leq j\leq \ell} \dot{\zeta}_j)\kappa^{\frac{1}{2}}(P^tP)\dot{m}_P^+, \;\; \delta_{\rho_v} = \delta_{\rho_v}(\dot{\tau}_j) = \frac{\vartheta^m}{\vartheta^m - 1}\left(a_1\dot{\tau}_j + a_2\dot{\tau}_j^2 + a_3\dot{\tau}_j^3\right),$$

$$a_1 = 4+\sigma+4,\, a_2 = 2(3+\sigma+\dot{m}_A^+(1+\sigma))\mu + 2 + \sigma,\, a_3 = 2(1+\sigma+\dot{m}_A^+(1+\sigma))\mu.$$

Subscript j on various quantities indicates the current (fine) level and $j-1$ or c its adjacent coarse level in the hierarchy, with $j=1$ the coarsest and $j=\ell$ the finest. These subscripts may be suppressed when there is no risk of ambiguity. Superscripts in parentheses like (1), (i), and (∞) are used to denote iteration count. We let $\check{A}_h$ and $\check{b}_h$ denote A_h and b_h rounded to $\mathcal{B}$, respectively, and use this haček diacritical mark to denote any quantity derived from them. We use tilde to denote values *computed* from $\check{A}_h$ and $\check{b}_h$.

REFERENCES

[1] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, SIAM Books, Philadelphia, 2000, https://doi.org/10.1137/1.9780898719505. Second edition.

[2] E. Carson and N. J. Higham, *A New Analysis of Iterative Refinement and Its Application to Accurate Solution of Ill-Conditioned Sparse Linear Systems*, SIAM Journal on Scientific Computing, 39 (2017), pp. A2834–A2856, https://doi.org/10.1137/17M1122918.

[3] E. Carson and N. J. Higham, *Accelerating the Solution of Linear Systems by Iterative Refinement in Three Precisions*, SIAM Journal on Scientific Computing, 40 (2018), pp. A817–A847, https://doi.org/10.1137/17M1140819.

[4] N. Higham and S. Pranesh, *Exploiting Lower Precision Arithmetic in Solving Symmetric Positive Definite Linear Systems and Least Squares Problems*, Tech. Report MIMS Preprint 2019.20, University of Manchester, 2019, http://eprints.maths.manchester.ac.uk/2736/.

[5] J. Mandel, S. McCormick, and R. Bank, *Variational Multigrid Theory*, SIAM, Philadelphia, 1987, ch. 5, pp. 131–177, https://doi.org/10.1137/1.9781611971057.ch5.

[6] S. F. McCormick, J. Benzaken, and R. Tamstorf, *Algebraic Error Analysis for Mixed-Precision Multigrid Solvers*, SIAM Journal on Scientific Computing, 43 (2021), pp. S392–S419, https://doi.org/10.1137/20M1348571.

[7] L. Piegl and W. Tiller, *The NURBS book*, Springer Science & Business Media, 2012, https://doi.org/10.1007/978-3-642-97385-7.

[8] G. Strang and G. Fix, *An Analysis of the Finite Element Method*, Wellesley-Cambridge Press, second ed., 2008.