Journal of Computational Physics 352 (2018) 265–284

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A stopping criterion for the iterative solution of partial differential equations

Kaustubh Rao [a], Paul Malan [a], J. Blair Perot [b,*]

[a] Computational Fluid Dynamics Group, Simulia Inc., Johnston, RI, 02919, USA
[b] Theoretical and Computational Fluid Dynamics Laboratory, University of Massachusetts, Amherst, MA, 01003, USA

ARTICLE INFO

Article history:
Received 12 July 2017
Accepted 19 September 2017
Available online 28 September 2017

Keywords:
Convergence estimate
Error estimate
Fluid dynamics
Stopping criteria

ABSTRACT

A stopping criterion for iterative solution methods is presented that accurately estimates the solution error using low computational overhead. The proposed criterion uses information from prior solution changes to estimate the error. When the solution changes are noisy or stagnating it reverts to a less accurate but more robust, low-cost singular value estimate to approximate the error given the residual. This estimator can also be applied to iterative linear matrix solvers such as Krylov subspace or multigrid methods. Examples of the stopping criterion's ability to accurately estimate the non-linear and linear solution error are provided for a number of different test cases in incompressible fluid dynamics.

1. Introduction

The stopping criterion for iterative nonlinear equation solvers is an implementation detail that gets less attention than it warrants. This small aspect of the iterative solver can have an outsized impact on the overall performance of the implementation because performing additional iterations due to overly conservative stopping estimates wastes computational resources. In most situations iteration of a partial differential equation (PDE) problem until errors or residuals are even close to machine precision is computational overkill because the solution already has some level of error due to the discretization process. An efficient solver implementation will stop the iterations once a level of error specified by the user is obtained. Therefore, an efficient implementation of an iterative solution method requires an estimate of the solution error.

Fig. 1 juxtaposes the stopping criterion proposed in this work with the classic 3 decade reduction in residual stopping criterion, in order to highlight the importance of a good stopping criterion. A well-chosen stopping criterion can result in either computational savings or improved solution quality. In this example the user considers a relative solution error below 1% to be sufficiently converged for their needs. The figure depicts the progress of the relative solution error (error norm/solution norm) and the normalized residual (residual/initial residual) as a function of the iteration number for a turbulent diffuser flow simulation (described in section 3.2).

Fig. 1(a) shows the evolution of the x-momentum solution. In this case the initial guess (potential flow) is sufficiently good that a 3-order reduction in the residual (thick black line) results in excessive iterations. In this particular case, the savings produced by stopping at the right time (circle on the thin red line) is around 20%. But it can often be much larger. Fig. 1(b) shows the evolution of turbulent kinetic energy for the same problem. In this case the initial guess for the

* Corresponding author.
E-mail address: perot@umass.edu (J.B. Perot).

https://doi.org/10.1016/j.jcp.2017.09.033
0021-9991/

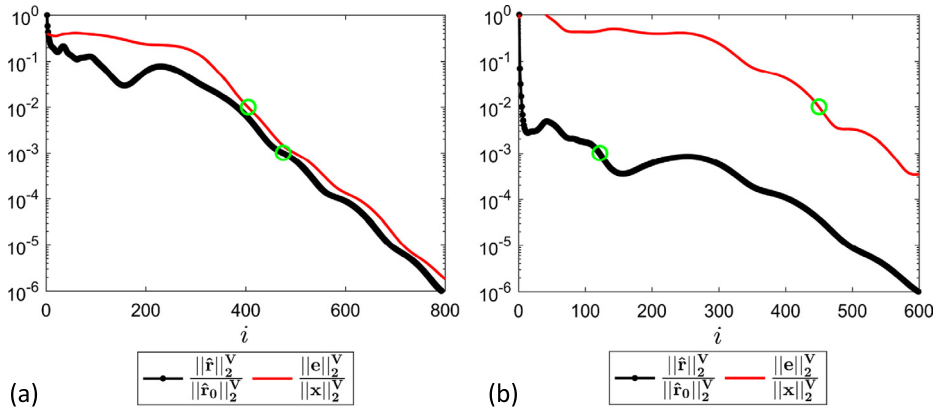

Fig. 1. Normalized residual (thick black lines) and relative error (thin red lines) for the diffusor problem described in section 3.2 (a) x-momentum convergence showing the situation when waiting for a 3 order drop in residuals leads to excessive iterations, (b) turbulent kinetic energy convergence showing the situation when the residuals drops by 3 orders but the error in the solution is still very high.

turbulence solution is bad (it is initialized to a constant value) and a 3-order reduction is easily achieved resulting in an early exit but a final turbulence solution with an excessively large error.

It is important to note that errors in the solution of PDEs arise from different sources. Classical error estimation (sometimes called discretization error, or local error estimation) is used to implement grid refinement and coarsening algorithms. This sort of error estimation is concerned with the error between a discrete approximation and its continuous counterpart (the original PDE). Classical error estimation is a broad and well developed area of research (see references [1–3] for some examples) but is not the area of discussion in this paper. This work focuses on the iterative error (sometimes called linearization error) that exists when an iterative method is converging to its discrete target solution. In the context of the present work, the PDE problem is already considered to be discretized, and the exact solution (when computing and discussing errors in this work) will be considered to be the exact solution to the *discrete* PDE problem. We are not concerned in this work, with how well that discrete solution approximates the continuous PDE solution (the realm of mesh adaptation). We are instead concerned with how close our current iterative solution is to the exact solution to the given discrete system.

Many stopping criteria are based on a norm of the residual vector [4–6]. But stopping iteration based solely on the residual is neither a safe nor a robust solution as shown in Fig. 1. The magnitude of any residual is totally arbitrary (see section 2.4 for details). Normalizing the residual can remove the magnitude problem, but (as shown in Fig. 1) is still problematic. If the initial guess is a good one, iteration may be incapable of achieving the prescribed relative reduction in the residual (due to reaching round-off). Or the iterations may just waste resources (as in Fig. 1(a)). If the initial guess is very bad, the iterative procedure will exit prematurely, when the solution error is still large (as in Fig. 1(b)).

Stopping criteria based on the residual and additional information about the problem perform better but still have issues. For example, a classic stopping criterion is to use the condition number of the Jacobian times the normalized residual to guarantee a certain reduction in the relative error ($\frac{\mathbf{e}}{\mathbf{x}} \leq \kappa \frac{\mathbf{r}}{\mathbf{r}(0)}$). There are two problems with this approach. First it requires a condition number estimate. Second, and much more importantly, the bound being used in this approach is excessively conservative so this stopping criterion can cause excessive iteration. Note that the problem of excessive iteration is more and more likely as mesh sizes (and therefore condition numbers) get larger. It is therefore only more recently, with the advent of large 3D meshes (and large condition numbers) that the inadequacy of this classic stopping criterion has become particularly pressing. The ultimate cost saving for high performance computing problems such as the direct numerical simulations in references [7] and [8] can be as large as 50,000 CPU hours.

This paper takes an unconventional approach to developing the stopping criterion, and abandons the residual (or its norm) as a useful starting point. Instead, the proposed stopping criterion looks at the size of the solution changes between each iteration. Error estimation based on progress (prior solution changes) is a non-trivial task because small changes in the solution do not necessarily mean the solution is converged, it may simply indicate that this is a difficult problem to solve and the method is converging slowly. Nevertheless, this approach is tenable and is not entirely previously unknown. In the past, classic linear iterative solvers such as Jacobi iteration and Gauss–Seidel iteration sometimes used prior solution changes and an extrapolation hypothesis to estimate progress [9,10]. Most modern matrix solution methods (such as Krylov subspace methods) typically have a far more erratic convergence behavior in both the residuals and the solution increments than Jacobi or Gauss–Seidel iteration. This has essentially led to the total abandonment of error estimation via progress extrapolation (though an exception is ref. [11]). However, in this work we will rejuvenate the extrapolation approach by using a robust and parameter-free smoothing approach. We will also focus primarily on the outer nonlinear iterations.

Section 2 of this paper presents the mathematical background for this work. It is shown that for PDE problems, certain vector and matrix norms are particularly attractive for performing the error estimation. Section 3 describes the test cases

and code that are used to demonstrate the usefulness of the proposed error estimator. Classic error estimators based on the residual norm are discussed in Section 4. These are used as the fallback for the smoothed extrapolation error estimation derived in Section 5. A few final considerations of the method, such as its application to linear solvers, are presented in Section 6, followed by a final discussion in Section 7.

2. Background

A nonlinear equation solver is designed to find the answer to the problem, $\mathbf{f}(\bar{\mathbf{x}}) = 0$ where $\mathbf{f}$ is a vector of N nonlinear equations and $\bar{\mathbf{x}}$ is a vector of N solution unknowns. In this work we will assume the nonlinear problem is well posed and has at least one solution. We will also be interested in the case where N is large (at least a million). For some initial guess of the solution, $\mathbf{x}$, the residual $\mathbf{r} = -\mathbf{f}(\mathbf{x})$ tells us how well the equations are satisfied by the guess, and $\mathbf{e} = \bar{\mathbf{x}} - \mathbf{x}$ is the error in the guess. The essential idea of this work is that the residual (error in the equations) is fundamentally different from the error (error in the solution), and the residual is easy to compute but the error which is difficult to compute is what we actually need.

The error and the residual are related. The nonlinear functions can be expanded in a Taylor series about the guess value, $0 = f_i(\bar{x}) = f_i|_{\mathbf{x}} + (\bar{x}_j - x_j)\left.\frac{\partial f_i}{\partial x_j}\right|_{\mathbf{x}} + \frac{1}{2}(\bar{x}_j - x_j)(\bar{x}_k - x_k)\left.\frac{\partial^2 f_i}{\partial x_j \partial x_k}\right|_{\mathbf{x}} + \ldots$ where Cartesian tensor notation has been used, and $J = \left[\frac{\partial f_i}{\partial x_j}\right]$ is the Jacobian matrix, and $\left[\left.\frac{\partial^2 f_i}{\partial x_j \partial x_k}\right|_{\mathbf{x}}\right]$ is a third rank tensor. This expression can be reformulated in terms of the residual and the error and becomes $\mathbf{r} = J\mathbf{e} + \frac{1}{2}\left[\left.\frac{\partial^2 f_i}{\partial x_j \partial x_k}\right|_{\mathbf{x}}\right] : \mathbf{ee} + O(\mathbf{e}^3)$. When the error is small enough the higher order terms can be neglected and

$$\mathbf{r} \approx J\mathbf{e} \tag{1}$$

to a good approximation. Since stopping criteria do not need to be perfect, this relation will be sufficient for our purposes.

If the Jacobian matrix is not singular, then this relation shows that as the residual goes to zero, the error also must go to zero. But the residual does not reveal much more than that. For example, this equation shows that controlling the residual (below a certain bound for example) does not necessarily control the error as well. In particular, consider an eigenvector decomposition of the error and residual vectors into the eigenvectors of the Jacobian matrix. That is, $\mathbf{e} = \sum_i a_i \mathbf{v}_i$ where $\mathbf{v}_i$ is an eigenvector of J and a_i is the amplitude of that particular eigenvector. If the error vector is predominantly constituted by the eigenvectors associated with smallest sized eigenvalues (these are slowest spatially varying modes, or lowest frequency modes) then the residual will be small even when the error is not. $\mathbf{r} \approx J\mathbf{e} = J\sum_i a_i \mathbf{v}_i = \sum_i (\lambda_i a_i)\mathbf{v}_i$. The small eigenvalues remove the low frequency modes from the residual, even when they are large in the error itself. This effect is even worse for least squares problems where eigenvalues are effectively squared [12]. Unfortunately, the scenario of error predominantly in low frequency modes is not an unusual situation. In most non-linear iterative methods, low frequency (small eigenvalue modes) errors that span the whole spatial solution domain, are the last to be removed during the iterative process. Despite the fact that equation (1) looks like a simple linear relation between the residual and the error it is deceptive (even for a constant Jacobian matrix). Due to the action of eigenvalues, a 3 order-of-magnitude drop in the residual during the iterative process does *not* imply that a 3 order-of-magnitude drop in the error has also occurred. As mesh sizes get larger (and condition numbers get larger), and the span between the largest and smallest eigenvalues increases, this disconnect between how the residuals and the errors behave becomes ever more exacerbated. The key take away is that residuals only reflect high frequency (large sized eigenvalue) error modes, and these modes are *not* the ones that usually present themselves in general problems.

Despite the fact that residuals as a proxy for errors constitutes a poor numerical practice, using iterative stopping criteria based on residuals is still extremely common. Probably because residuals are very easy to compute and errors are difficult. It was shown above that prescribed residual drops do not imply corresponding error drops. In addition, prescribed residual values have no implied meaning about the error value. Finally, the magnitude difference between a residual and its corresponding error is completely arbitrary. Typically the two quantities don't even have the same units.

2.1. Approach

There are two quite different approaches to estimating the error of a non-linear iterative process. By far the most common approach is to use ideas from linear systems. Taking the norm of the residual, and using the triangle inequality tells us that,

$$\|\mathbf{e}\| \leq \left\|J^{-1}\right\| \|\mathbf{r}\| \tag{2}$$

This relation provides an error *bound* from a residual calculation. Residuals are already widely used for stopping criteria, so this small modification allows existing methods to now bound the error more rationally. There are two difficulties with

this common approach. First, an error estimate, and not a bound, is what is really desired. This bound is overly conservative and therefore wastes real computational resources. Second, estimating the norm of a matrix inverse can be difficult because it is prohibitively expensive to compute the matrix entries for an inverse. This second problem is surmountable, and this work will address it. The first is a more serious issue.

The second approach to error estimation is essentially observational. By watching past iterative progress, and making the assumption that the future progress will behave similarly, it is possible to extrapolate and obtain an error estimate. This estimate is of course, not perfect, but our tests show that it is usually much better than the bound given by equation (2), and therefore saves substantial computational effort. Ultimately, the final proposed error estimator actually uses a combination of both approaches. The estimator defaults to extrapolation about 95% of the time, but resorts to the bounding approach (eqn. (2)) when extrapolation fails (such as during early iterations or when convergence stalls entirely).

2.2. Vector norms

Equation (2) requires a discussion of vector and matrix norms, which are surprisingly important for this topic. An L_n vector norm is classically defined as $\|\mathbf{e}\|_n = \left(\sum_i |e_i|^n\right)^{\frac{1}{n}}$ where n is any integer from 1 up to and including infinity. The L_1 norm is the sum of all the magnitudes of all the components in the vector. The L_∞ norm is the maximum magnitude of all the items in the vector list. And the most commonly used norm is probably the L_2 norm which simplifies to the relation $\|\mathbf{e}\|_2^2 = \sum_i e_i^2$. These classically defined norms have nice mathematical properties (such as $L_1 \geq L_2 \geq L_3 \geq \cdots \geq L_\infty$, and $L_1 \leq N^{1/2}L_2 \leq N^{2/3}L_3 \leq \cdots \leq NL_\infty$) where N is the number of items in the vector. But these classic norms are not very useful for discretized PDE solvers. The primary issue is that these classic norms are very mesh size dependent. So, for example, a large value for a classic error norm can tell you that the errors are large or alternatively that the number of mesh elements is large. Classical norms do not distinguish between the two possibilities.

The usual solution is to use size-normalized norms, $L_n^{\frac{1}{N}}$, where we use a superscript on the norm to indicate size-normalization, so $\|\mathbf{e}\|_n^{\frac{1}{N}} = \left(\frac{1}{N}\sum_i |e_i|^n\right)^{\frac{1}{n}}$ where N is the number of items in the vector. This is typically the number of mesh points or unknowns in the PDE solution. These size-normalized norms work better for PDE discretization. For a given PDE problem, different mesh types (triangles, quads, etc.) and different mesh resolutions, produce roughly the same norm value for the size-normalized norm. However, size-normalized norms are still not good enough. For highly stretched or refined meshes, size normalization is not sufficient and different meshes give order of magnitude different norm values for a nearly identical field solutions. This is not useful. A good definition for a norm of a field variable $\|\mathbf{x}\|$ should produce nearly the same value irrespective of the underlying discretization of that variable.

The better norms to use for PDE variables, and therefore for this work, are integral norms, L_n^V, represented in this work with the superscript, V. So $\|\mathbf{e}\|_n^V = \left(\frac{\int_\Omega |e_i|^n dV}{\int_\Omega dV}\right)^{\frac{1}{n}}$ where the integral is over the whole domain and the integral is normalized by the domain volume. In theory, computing this norm requires having a prescribed interpolation method available for the discrete unknowns (to be able to produce continuous functions that can be integrated). In practice, this norm is much simpler than it looks to compute. In this work we show that for the purposes of convergence estimation the lowest order integration rule (midpoint integration) is quite sufficient to determine this norm. This norm then becomes effectively a discrete volume weighted norm.

$$\|\mathbf{e}\|_n^V \approx \left(\frac{\sum_i |e_i|^n V_i}{\sum_i V_i}\right)^{\frac{1}{n}} \tag{3}$$

where V_i is the small volume associated with each unknown that is being integrated. For a cell or element based unknown this would be the cell/element volume. For a node based unknown it would the dual-volume surrounding that node (which is usually the summation of some fraction of all the cell/element volumes touching that node).

2.3. Matrix norms

For completeness we should also define the matrix norm that is compatible with the volume weighted vector norm given in equation (3). A consistent matrix norm is given in terms of the vector norm by the expression, $\|A\|_n^V = \max(\mathbf{w} \neq 0)\left(\frac{\|A\mathbf{w}\|_n^V}{\|\mathbf{w}\|_n^V}\right)$ where A is an $N \times N$ matrix [13]. By absorbing the $1/n$ power of the cell volume weights into the search vector $\mathbf{w}$, this can be rearranged into an expression that uses the classical norms $\|A\|_n^V = \max(\tilde{\mathbf{w}} \neq 0)\left(\frac{\|V^{1/n}AV^{-1/n}\tilde{\mathbf{w}}\|_n}{\|\tilde{\mathbf{w}}\|_n}\right)$ where

$V^{1/n}$ is a diagonal matrix containing the $1/n$ power of each cell/element volume and $V^{-1/n}$ is its inverse. This expression is useful because explicit representations for the classical matrix norms are known.

We can therefore write that $\|A\|_1^V = \max_j \left(\sum_i |V_i a_{ij} V_j^{-1}| \right)$, which is the maximum of the sum of every matrix column. And $\|A\|_\infty^V = \max_i \left(\sum_j |V_i^{1/\infty} a_{ij} V_j^{-1/\infty}| \right) = \max_i \left(\sum_j |a_{ij}| \right)$, which is the maximum of the sum of every matrix row. And most useful for this work,

$$\|A\|_2^V = \sigma_{\max}\left(V^{1/2} A V^{-1/2}\right) \tag{4}$$

where $\sigma_{\max}(B) = \{\lambda_{\max}(BB^T)\}^{1/2}$ is the maximum singular value of the matrix B (and $\lambda_{\max}$ is the maximum eigenvalue). Note that volume scaling of the matrix, $V^{1/2} A V^{-1/2}$ (needed for the matrix volume-weighted norm) has no effect on the diagonal entries of the matrix, it only affects the off-diagonal entries (with the square root of a volume ratio).

The volume-weighted matrix 1-norm and infinity-norm look relatively easy to compute compared to the matrix 2-norm, which requires a singular value or eigenvalue calculation. However, this simplicity is deceptive because most often we are interested in the matrix norm of a inverse (such as in equation (2)). In that case computing the matrix 1-norm and infinity-norm are prohibitively expensive because forming the inverse matrix is prohibitively expensive (for N greater than a million), but computing the matrix 2-norm is still possible because $\|A^{-1}\|_2^V = \frac{1}{\sigma_{\min}(V^{1/2} A V^{-1/2})}$ is simply a matter of finding the minimum singular value rather than the maximum one.

2.4. Jacobian ambiguity

Cell/element volumes creep into the analysis in one more place. They appear in the Jacobian itself. We believe there is one version of the Jacobian that is particularly useful, especially in the context of convergence estimates. Specifically, for the case of PDE problems, one very particular scaling of the Jacobian matrix has a minimum singular value that is essentially independent of the mesh size and the discretization type (triangle, quad, etc.). For this specific Jacobian, the minimum singular value is purely a function of the problem physics.

If we apply equation (2) to the specific case of the volume weighted 2-norm (the 2-norm is the most common index choice in practice) then we have $\|\mathbf{e}\|_2^V \leq \frac{\|\mathbf{r}\|_2^V}{\sigma_{\min}(V^{1/2} J V^{-1/2})}$ and it is clear why mesh independence of the minimum singular value is particularly attractive. For one specific Jacobian scaling choice, the constant of proportionality between the residual and the error is determined only by the physics. This could make its estimation much easier. The estimate is now problem dependent but discretization independent.

The scaling ambiguity of a Jacobian is clear. Each equation in the original system $\mathbf{f}(\bar{\mathbf{x}}) = 0$ can be multiplied by a non-zero weight and the solution of the system, $\bar{\mathbf{x}}$, will remain unchanged. But the row in the Jacobian corresponding to that equation will change (it will be multiplied by the weight). In this work, we are only interested in this simple act of weighting each equation. But in general, the original equations can also be added together and even nonlinearly mixed together producing a fascinating variety of Jacobians.

The multiplicative scaling ambiguity in the equations is critical for systems that come from discretized PDEs. The issue is best discussed with a concrete example. The simple Laplace equation will suffice. On a simple 2D Cartesian mesh a finite difference (FD) discretization of Laplace's equation produces a 'neighbor stencil' for the Jacobian that is not the same as the stencil that a finite volume (FV) or finite element (FE) method produces [14],

$$\text{FD:} \quad \begin{matrix} 0 & \frac{1}{\Delta y^2} & 0 \\ \frac{1}{\Delta x^2} & -\frac{1}{\Delta x^2} - \frac{1}{\Delta y^2} & \frac{1}{\Delta x^2} \\ 0 & \frac{1}{\Delta y^2} & 0 \end{matrix} \qquad \text{FE or FV:} \quad \begin{matrix} 0 & \frac{\Delta x}{\Delta y} & 0 \\ \frac{\Delta y}{\Delta x} & -\frac{\Delta y}{\Delta x} - \frac{\Delta x}{\Delta y} & \frac{\Delta y}{\Delta x} \\ 0 & \frac{\Delta x}{\Delta y} & 0 \end{matrix}$$

The FE/FV version of the stencil (or matrix) is the FD version multiplied by the cell volume. In this Laplacian example, the matrices representing the Laplacian's are identical except that each row has a different diagonal scaling. In the FE/FV version each row of the matrix is the FD row multiplied by the cell volume. The matrix formed by the FE/FV method has many attractive properties. For example, for a Laplacian PDE, the matrix is symmetric, even on unstructured and non-uniform meshes. However, the FD matrix also has one attractive property (possibly only this one). The FD matrix has a minimum singular value (and eigenvalue) that is independent of the mesh size and only depends on the problem physics. Formally, it is not perfectly independent of mesh size, there is a perturbation term that depends on the order of accuracy of the discretization. But the leading order term is independent, and this makes the minimum singular value constant enough for error estimation purposes.

The reason this mesh-independence happens for the FD matrix is not totally arbitrary. The FD form produces a set of equations $\mathbf{f}(\bar{\mathbf{x}}) = 0$ in which each discrete equation has exactly the same units as the equations in the original continuous PDE. Similarly, the residuals in the FD method have the same units as the PDE equations. On the other hand, the equations

for the FV/FE approach have a cell/element volume lurking in them. The residual corresponding to a FE/FV solution is the error in the PDE equation integrated over (essentially multiplied by) the cell volume associated with that unknown. So FV/FE residuals have a dependence on cell/element volumes (and therefore on the mesh size).

2.5. Volume weighted error/residual relation

In this work we will continue to think of J as resulting from a FV or FE discretization. These discretization's are the most common and have too many nice properties. But we will define a volume weighted Jacobian, $\hat{J} = V^{-1}J$ that has the same units as the FD version of the Jacobian (and the same units as the original PDE), and that therefore has mesh independent properties for the 2-norm of its inverse. When using a FV or FD discretization we will also define a volume weighted residual, $\hat{\mathbf{r}} = V^{-1}\mathbf{r}$.

Note that the matrix volume-2-norm of the modified Jacobian is $\|\hat{J}^{-1}\|_2^V = \frac{1}{\sigma_{\min}(V^{1/2}\hat{J}V^{-1/2})} = \frac{1}{\sigma_{\min}(V^{-1/2}JV^{-1/2})}$. So the matrix volume-2-norm now modifies the FV/FE Jacobian, J, symmetrically when finding the necessary singular value. So if the original FV/FE Jacobian is symmetric (or antisymmetric) then the volume weighted matrix $V^{-1/2}JV^{-1/2}$ is also symmetric (or antisymmetric).

For error estimation, and assuming a FE or FV discretization, the error expression using the appropriate norms and modified residuals for PDEs is therefore,

$$\|\mathbf{e}\|_2^V \leq \left\|J^{-1}V\right\|_2^V \left\|V^{-1}\mathbf{r}\right\|_2^V = \frac{\|V^{-1}\mathbf{r}\|_2^V}{\sigma_{\min}(V^{-1/2}JV^{-1/2})} = \frac{\|\hat{\mathbf{r}}\|_2^V}{\sigma_{\min}(V^{-1/2}JV^{-1/2})} \tag{5}$$

The relevant residual for each cell is actually the FE/FV residual divided by the volume for that cell to produce the volume weighted (or modified) residual, $\hat{\mathbf{r}}$. This volume weighted residual has the same units as the equations of original PDE. The singular value in the denominator of equation (5) is now a mesh-independent quantity. It depends only on the physics and this will make it easier to estimate.

The correct way to volume weight both vector norms and residual/matrices for PDE derived problems is an important basis for the development of good error estimators.

3. Test cases

The performance of the stopping criterion will be evaluated by comparing the error estimator's predictions with the exact error for a variety of test cases. The exact error is calculated by performing all the simulations twice and saving the final solution from the first simulation to calculate the error on the second simulation. The first simulation is typically run for 50% more nonlinear iterations on the first run than on the second run when the error is recorded.

The presented test cases all involve solutions of the incompressible Navier–Stokes equations or the incompressible Reynolds Averaged Navier–Stokes (RANS) equations that include turbulence. We will show error estimates for both the velocity components and for the turbulence model quantities. We hypothesize that the Navier–Stokes equations are a sufficiently complex system to adequately test the proposed error estimator. The iterative method used for these tests is a segregated solver in which each field variable is solved uncoupled from the others sequentially inside each non-linear iteration. The incompressibility condition is enforced using a projection method. The scheme is very similar to the unsteady SIMPLE method [15].

The nonlinear iterative method is essentially a fixed-point iteration for the incremental unknowns. One iteration consists of the two steps.

$$\tilde{J}(\mathbf{x}^n)\delta\mathbf{x}^{n+1} = \mathbf{r}^n$$
$$\mathbf{x}^{n+1} = \mathbf{x}^n + \delta\mathbf{x}^{n+1}$$

where $\tilde{J}(\mathbf{x}^n)$ is an approximation to the Jacobian and $\mathbf{x}^{n+1}$ is the solution vector containing all the fields (velocity, pressure, total energy, and turbulence variables). Our particular approximation to the Jacobian drops all the matrix entries that couple different fields, so that the linear equation inversion process required in the first step can be performed for each field separately and individually.

The test cases used to test the error estimator are summarized and shown below.

3.1. Boundary layer

Incompressible turbulent flow over a smooth flat plate. The Reynolds number based on the plate length is 1.03×10^7. The simulation aims to reproduce the experimental data of Wieghardt [16]. The simulation is solved using two models, RNG K-epsilon [17] and Realizable K-epsilon [18]. The Realizable model converges while for the RNG version convergence stalls (due to highly non-linear turbulence positivity constraints). The RNG model case is used to highlight the strategies used by the error estimator to tackle convergence stall. See Fig. 2a.

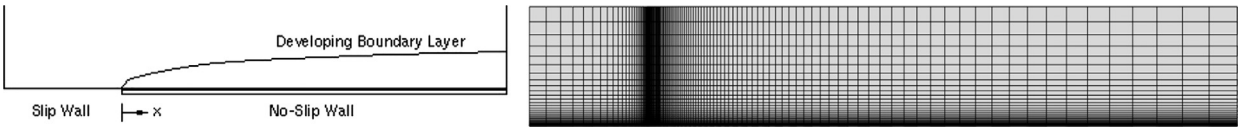

Fig. 2a. Boundary layer.

3.2. Diffuser

For the diffuser case, the K-omega SST turbulence model [19] is used at a Reynolds number based on the inlet opening of 20,000. The flow is based on the experiment referred to in [20]. See Fig. 2b.

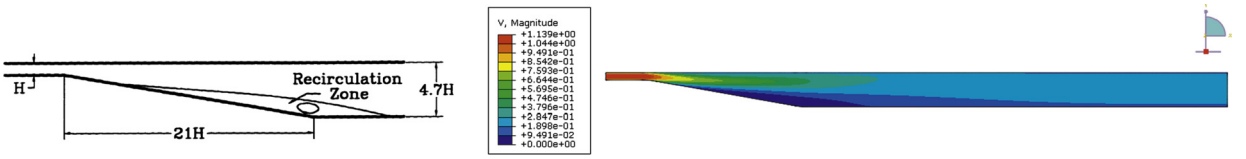

Fig. 2b. Diffuser.

3.3. U-bend

The simulation calculates the turbulent flow through a two-dimensional duct with a U-turn. The simulation is solved using the Spalart–Allmaras turbulence model [21]. The Reynolds number based on the mean velocity and channel width is 10^6. See Fig. 2c.

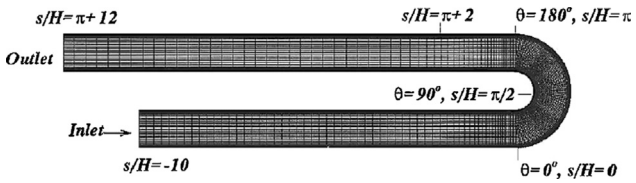

Fig. 2c. U-bend.

3.4. Unsteady shedding

The flow simulated here is a two-dimensional laminar flow past a cylinder. The Reynolds number based on cylinder diameter is 100. The flow is unsteady and is characterized by vortex shedding with frequency that is characterized by a non-dimensional parameter known as the Strouhal number (St). Refer to references [22] and [23]. This simulation takes many timesteps. But in the following results, the convergence of only one of the timesteps of the evolving flow is considered (since it is representative). See Fig. 2d.

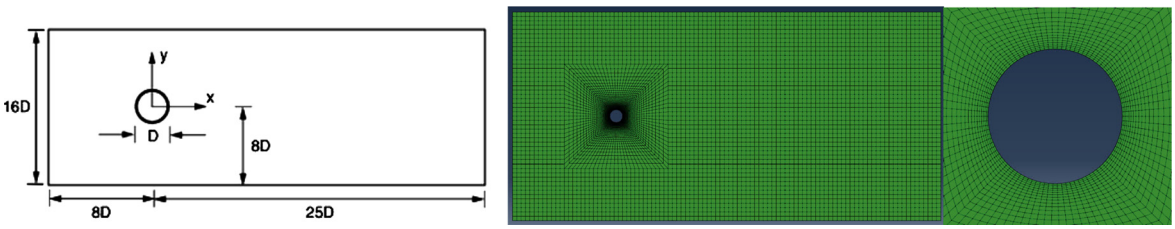

Fig. 2d. Unsteady shedding.

3.5. Conjugate Heat Transfer (CHT)

The flow simulated here involves the heat transfer between a thick conducting pipe (inner radius = 0.5 m, outer radius = 1.0 m, length = 2.0 m, outer wall at a fixed temperature = 400 K) and flow through the pipe (inlet of 1 m/s at 300 K). The problem is solved using the Realizable K-epsilon turbulence model. See Fig. 2e.

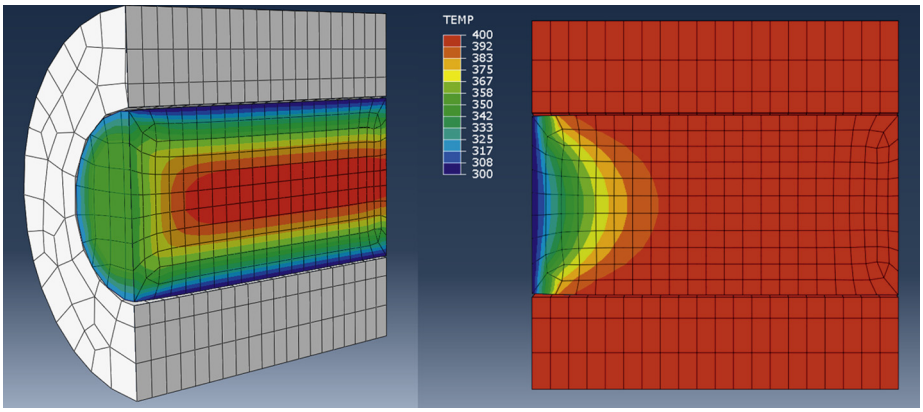

Fig. 2e. CHT.

3.6. Cavity

The flow simulated here is a lid driven square cavity [24] at Re = 1000. To prove mesh independence the cavity is meshed with 32×32, 64×64, 128×128 resolution hexahedral mesh, and one 64×64 (approximately) mixed mesh. The lid velocity is 1 m/s and the domain size is 1 m. See Fig. 2f.

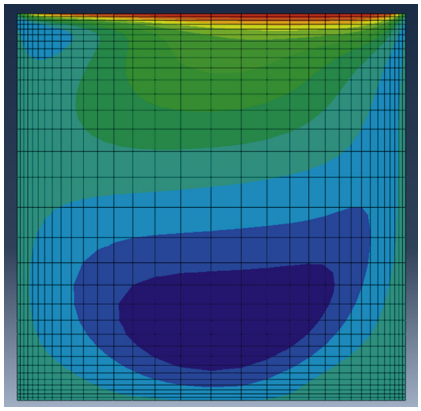

Fig. 2f. Lid driven cavity.

4. Classic error estimation

The most common method for estimating the error is to replace the bound given by equation (5) with an equality and a constant.

$$\|\mathbf{e}\|_2^V = \frac{C}{\sigma_{\min}(V^{-1/2} J V^{-1/2})} \|\hat{\mathbf{r}}\|_2^V \tag{6}$$

where $C \leq 1$. The hope with this approach is that the singular value encapsulates the most important information about the scaling of the problem and that the constant, C, is nearly universal and therefore mesh and iteration and problem independent. We will show via examples that the constant, C in this type of traditional stopping criteria is indeed often surprisingly mesh and iteration independent (varies by less than an order of magnitude) in many practical cases. But it is not independent of the problem.

Fig. 3(a) plots $R \equiv \frac{\|\mathbf{e}\|_2^V}{\|\hat{\mathbf{r}}\|_2^V} = \frac{C}{\sigma_{\min}(V^{-1/2} J V^{-1/2})}$ versus the iteration number for a series of different meshes for the same problem (the cavity flow problem in section 3.6). The ratio of the error and residual of the x-component of the velocity is shown. The minimum singular value does not change with the iteration number at all and is constant for different meshes (using a volume weighting of the Jacobian) to within a perturbation term that is small and proportional to the mesh size (and discussed more in section 4.2). This means the variation with iteration number seen in this plot is almost entirely due to the variation in the constant C.

Fig. 3(a) shows there is some dependence on the iteration number for early iterations. But very little variation in the value for C for different mesh sizes (and types) after many iterations. This is due to the fact that C represents how well the inequality in equation (5) can be represented by the equality assumption in equation (6). At early iterations, the equality level (the constant C) is entirely dependent on the choice of the initial condition. After many iterations the residual and error are converging in a way that loses information about the initial condition and tends towards a constant ratio (that is problem dependent because of the singular value, but is largely iteration and mesh size/type independent).

While the result of an approximately mesh size/type and large iteration number independent C is universal to all our tested cases, it is not mathematically guaranteed. There are some exceptional linear problems where we know mathematically that the residual is constant until the last iteration, when it suddenly drops to machine precision zero. Such a convergence behavior will not have a constant C that is independent of the large iteration numbers (particularly the last one). It is likely that C is largely constant in our test cases because we are looking at nonlinear iterations in these test cases (the performance of the error estimator for linear solvers is evaluated in section 6.2). The nonlinear solvers we use are either Richardson iteration (with under-relaxation) or residual based line-searches. These are robust but simple iterative methods.

Fig. 3(b) looks at the same ratio, R but for a number of different problems (problems 3.1–3.6). Again, the ratio is shown to be largely independent of the iteration number for all the different problems (except for the early iterations). But note the log scale for the y-axis of the plot, so the various values for the ratio vary quite considerably from one problem to another. This large variation is mostly captured by the variation in the minimum singular value, but we will show later that the constant C is also somewhat problem dependent as well.

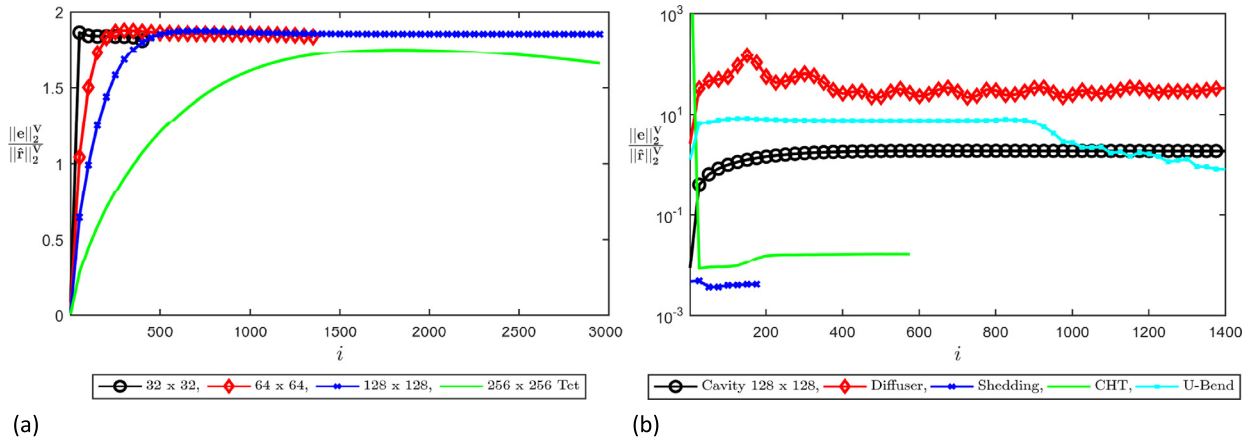

Fig. 3. Ratio of error to residual for the x-momentum equation. (a) Cavity flow using different mesh sizes and shapes. (b) For a variety of different flow cases.

These results show that R is nearly constant as a function of iteration number. R differs only at very early iterations (that are sensitive to the initial guess) and sometimes at very late iterations (when machine precision in the error and/or residual have been reached). R is also seen to be nearly mesh independent (Fig. 3(a)). R is not problem independent because it contains the extremely problem dependent minimum singular value in its definition and a somewhat problem dependent constant C.

Note that the goal in error estimation is not perfection. Any error estimate that is within an order of magnitude of the actual error will be sufficient to produce a reasonably effective stopping criterion. So when we refer to C being "nearly" constant we mean that it varies within the order of magnitude range needed for an inexpensive stopping criteria.

4.1. Physical estimation

There are a number of ways to estimate $\sigma_{\min}(V^{-1/2} J V^{-1/2})$. The simplest is physical intuition. Despite its mathematical look, the singular value $\sigma_{\min}(V^{-1/2} J V^{-1/2})$ depends almost entirely on the physics of the PDE system and not on its numerics (this is shown in the next section). It can therefore be estimated from knowledge of the PDE.

Singular values (and singular modes) are similar to eigenvalues and eigenmodes. But the distinction is important for non-symmetric real matrices J, such as arise in PDEs with advection (such as the Navier–Stokes equations). The eigenvalues of non-symmetric real matrices are often complex-valued. Singular values and singular modes, on the other hand, are always real valued and have physical significance. A singular value σ_i and singular mode $\mathbf{u}_i$ of the matrix B are the solution of the equation $B^T B\mathbf{u}_i = \sigma_i^2 \mathbf{u}_i$. This is an eigenvalue/eigenmode problem for the related matrix $B^T B$. The smallest singular value therefore corresponds to the square root of the smallest eigenmode of the symmetric matrix $B^T B$. The smallest eigenmode of a symmetric semi-positive definite matrix $B^T B$ is the "smoothest" eigenmode with the fewest zero-crossings. It therefore scales on the largest physical scales in the problem that generated the matrix B. Using this physical understanding of the

smallest singular mode and singular value allows one to make estimates of the smallest singular value, knowing the largest physical scales in the problem.

For example, for Laplace's equation in a rectangular domain, the FD matrix (or the symmetrically normalized FV/FE matrix, $V^{-1/2}JV^{-1/2}$) has a minimum singular value that is proportional to $\frac{1}{L_x^2}+\frac{1}{L_y^2}$ where L_x and L_y are the size of the solution domain. This is because for the Laplace equation in a rectangle, the largest physical length scales are dictated by the domain boundaries. Using another example, for an advection dominated Navier–Stokes PDE matrix, the minimum singular value scales like $\rho\frac{U}{L}$ where U and L are the characteristic velocity scale and characteristic length scale in the problem and ρ is the fluid density. So for the flow over an airfoil, the minimum singular value will be found to be proportional to the free-stream flow speed and inversely proportional to the boundary layer thickness (which depends indirectly on the viscosity).

However, results that demonstrate the validity of this physical estimation process are not provided in this work, because this approach is not general enough to be a valid method. This approach is impossible to implement for complex PDE systems and geometries where multiple velocity and length scales are present. We present this section, primarily to remind the reader that a PDE and the matrix mathematics underlying its numerical solution are not uncoupled. The minimum singular value appears in the mathematical bound (given by equation (5)) for a physical reason, and not just a mathematical one. It represents the largest physical scales.

For completeness of discussion, for a FD matrix (or symmetrically volume weighted FE/FV matrix) the *maximum* singular value scales like the maximum matrix entry (and is therefore extremely mesh dependent). For the Laplace equation this would scale like $\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}$ (or equivalently the maximum matrix diagonal element). For an advection dominated Navier–Stokes matrix the maximum singular value would scale as the maximum value for $\rho\frac{u}{\Delta x}$ where u is the local velocity at the mesh location of Δx. The largest singular value is related to the smallest computable scales.

4.2. Coarse mesh estimation

Because R is mesh independent and nearly iteration independent (as shown in Fig. 3), one method to estimate the value of R (which includes the singular value and the constant C) is to run one very coarse mesh simulation first. For the coarse mesh simulation, the initial residual and initial solution guess are saved. Once the coarse simulation completes the error for the initial guess can be computed, and the ratio determined.

The coarse mesh works because the minimum singular value is related to the smoothest and most slowly varying mode in the problem. So any coarse mesh that can reasonably resolve the smoothest possible eigenmode will be sufficient to get a good estimate for the ratio R.

Let us consider here, just how invariant the minimum singular value is to changes in the mesh size. To do this consider the modified PDE that the numerical method actually solves. This is sometimes called backward error analysis or modified equation analysis. If the exact PDE that we wish to solve is $L(\mathbf{v})=\mathbf{b}$, then a numerical method actually solves a modified PDE $L(\mathbf{v})+E(\mathbf{v})=\mathbf{b}$. If the numerical method is convergent then the perturbation of the PDE is $E(\mathbf{v})\propto(\Delta x)^p$ where p is the order of accuracy of the numerical method. The century old theory of Weyl [25] tells us that singular values are perfectly conditioned. The singular value of a perturbed matrix cannot be in error more than the norm of the perturbation itself. This means that $|\tilde{\sigma}_{\min}-\sigma_{\min}|\leq\|E\|_2$ where $\tilde{\sigma}_{\min}$ is the minimum singular value for the PDE (or equivalently the limit of the minimum singular as the mesh size of the discretization goes to zero). By its limit definition, this value is a constant independent of the mesh size. $\sigma_{\min}$ is the minimum singular value for a particular mesh. Since the perturbation scales with the mesh size, the minimum singular value for any particular mesh behaves like $\sigma_{\min}=\tilde{\sigma}_{\min}+O(\Delta x^p)$.

It is in this sense that this paper asserts that the minimum singular value is nearly mesh independent. The leading order term is mesh independent and the additional term is small (except in the case of very coarse meshes). Very coarse meshes are those for which the smallest singular mode (smoothest solution with the fewest zero crossings) is not well resolved by the mesh.

4.3. Rayleigh Quotient estimation

Another way to find a minimum eigenvalue (or singular value) is via repeated matrix inversion using what is referred to as 'inverse power' iteration. As above, this could be done with a coarse matrix in order to speed the process. However, we still consider this to be a fairly expensive operation for a convergence test and do not pursue this further.

An explicit and less expensive option for estimating an eigenvalue is the Rayleigh Quotient. Here the basic idea will be extended to find the minimum singular value (rather than eigenvalue). As noted earlier, the singular value, σ_i, and singular vector, $\mathbf{u}_i$, for a general matrix B is given by the equation, $B^TB\mathbf{u}_i=\sigma_i^2\mathbf{u}_i$. This is closely related to finding the eigenvalues of the symmetric matrix B^TB. Symmetric matrices have orthonormal eigenvectors $\mathbf{u}_i$, and real positive eigenvalues. Consider a test vector which is predominantly made up of one of these orthonormal singular vectors with small amounts of the others added in. Then this vector is given by $\mathbf{t}=\sum a_i\mathbf{u}_i$ where the a_i are all small except for the one a_i corresponding to the dominant singular mode. The generalized Rayleigh Quotient $\frac{\|B\mathbf{t}\|_2}{\|\mathbf{t}\|_2}$ is a then a good approximation for the singular value. Note that $\left(\frac{\|B\mathbf{t}\|_2}{\|\mathbf{t}\|_2}\right)^2=\frac{\mathbf{t}^TB^TB\mathbf{t}}{\mathbf{t}^T\mathbf{t}}=\frac{\left(\sum a_k\mathbf{u}_k^T\right)\left(\sum a_j\sigma_j^2\mathbf{u}_j\right)}{\left(\sum a_k\mathbf{u}_k^T\right)\left(\sum a_j\mathbf{u}_j\right)}=\frac{\sum a_j^2\sigma_j^2}{\sum a_j^2}\approx\sigma_{dominant}^2$ where the summations are over all the singular

modes, and the third equality is a result of the singular modes being orthonormal. The final near equality is due to the fact that all the amplitudes a_i are small except for the dominant one. Note that the Rayleigh quotient will find an approximation for *any* dominant singular mode in the test vector $\mathbf{t}$. It is not restricted to finding the maximum or minimum singular mode.

One method to estimate the minimum singular value is to therefore to guess the vector $\mathbf{t}$ that approximates to the minimum singular mode. This seems a difficult task at first, unless we remember that we happen to know a great deal about the physics (and PDE) of where the matrix B came from. In particular, we are looking for the lowest frequency (smoothest) solution field that fits in the domain and satisfies homogeneous boundary conditions. The amplitude of this low frequency solution is not important (because of the normalization happening in the Rayleigh quotient), we just need its shape.

A constant field works OK for $\mathbf{t}$, but has high frequency components near Dirichlet boundaries where the field values drop suddenly to zero. It tends to overestimate $\sigma_{\min}$ quite a bit.

A better option is to use the solution change. Define $\delta\mathbf{x}^{n+1} = \mathbf{x}^{n+1} - \mathbf{x}^n$ to be the change in the solution after the $n+1$ iteration of the iterative method. After a few iterations of a nonlinear iterative solver, the high frequency components of the solution are often converged and only the slowly converging low-frequency modes remain. This is the observation of all our test cases. So $\delta\mathbf{x}^{n+1}$ often quickly becomes a good approximation for the smallest singular mode. For a non-uniform mesh $\mathbf{t} = V^{1/2}\delta\mathbf{x}^{n+1}$ is a reasonable choice for the smallest singular mode. Then $\sigma_{\min}(V^{-1/2}JV^{-1/2}) = \frac{\|V^{-1/2}JV^{-1/2}V^{1/2}\delta\mathbf{x}\|_2}{D\|V^{1/2}\delta\mathbf{x}\|_2} = \frac{\|\hat{J}\delta\mathbf{x}\|_2^V}{D\|\delta\mathbf{x}\|_2^V}$ is an inexpensive estimate for the minimum singular value (and $D > 1$). The Raleigh Quotient always over estimates the minimum singular value because it is slightly contaminated by the other singular values which are always larger than the minimum, hence the fact that the constant $D > 1$. With this estimate $R = \frac{\|\mathbf{e}\|_2^V}{\|\mathbf{r}\|_2^V} = \frac{C}{\sigma_{\min}(V^{-1/2}JV^{-1/2})} = CD\frac{\|\delta\mathbf{x}\|_2^V}{\|\hat{J}\delta\mathbf{x}\|_2^V}$.

Fig. 4 uses Rayleigh quotient estimates using the solution increment to compute $\tilde{R} = 15\frac{\|\delta\mathbf{x}\|_2^V}{\|\hat{J}\delta\mathbf{x}\|_2^V}$ for a wide variety of test problems. In these tests the smallest singular mode estimate, $\mathbf{t} = V^{1/2}(\mathbf{x}^{n+1} - \mathbf{x}^n)$, is being calculated between two non-linear outer iterations. Simple (under-relaxed Richardson) fixed-point iteration is being used for these iterations. This figure assumes that the constant, CD, is 15. This figure should resemble Fig. 3(b), which is the exact value of R for the same problems. The assumption $CD = 15$ gets us almost within an order of magnitude of the exact results. Our final estimate method will determine the value of CD more accurately from prior iteration information (and will not use a hard value of 15).

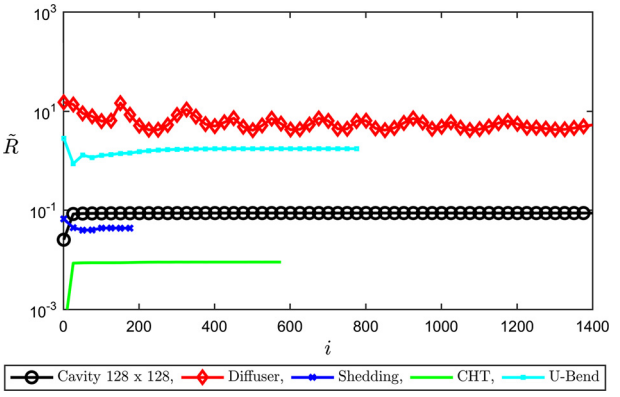

Fig. 4. Rayleigh Quotient estimates, $\tilde{R}$ for the same physics cases used in Fig. 3(b).

Note that if the iterative method uses fixed-point iteration on the incremental form of the equations then it solves $\tilde{J}\delta\mathbf{x}^{n+1} = \mathbf{r}^n$ at each iteration, where $\tilde{J}$ is some approximation of the actual Jacobian. If the approximation of the Jacobian is a good one, and the volumes don't vary rapidly in the mesh then $\sigma_{\min} \approx \frac{\|\hat{J}\delta\mathbf{x}\|_2^V}{D\|\delta\mathbf{x}\|_2^V} = \frac{\|\hat{\mathbf{r}}\|_2^V}{D\|\delta\mathbf{x}\|_2^V}$ and equation (6) becomes approximately,

$$\left\|\mathbf{e}^{n+1}\right\|_2^V \approx (CD)\left\|\delta\boldsymbol{x}^{n+1}\right\|_2^V \tag{7}$$

That is, Rayleigh quotient estimation of the minimum singular value is roughly equivalent to the assumption that the error is roughly some multiple, CD, of the current solution change.

The primary issue with all the classical approaches for error estimation described above (except running a coarse mesh simulation) is that they provide the minimum singular value, but not the constant C or CD, that is necessary in order to turn the error bound into an error estimate. This weakness of the classic methods is overcome in the next section.

5. Extrapolation error estimates

A completely different approach to error estimation is to watch the iterative progress and extrapolate it to deduce the error. Fig. 5 shows plots of the solution increments for different problems as a function of the iteration number. On a log–linear plot there are many regions where the convergence is reasonably well approximated locally by a straight line.

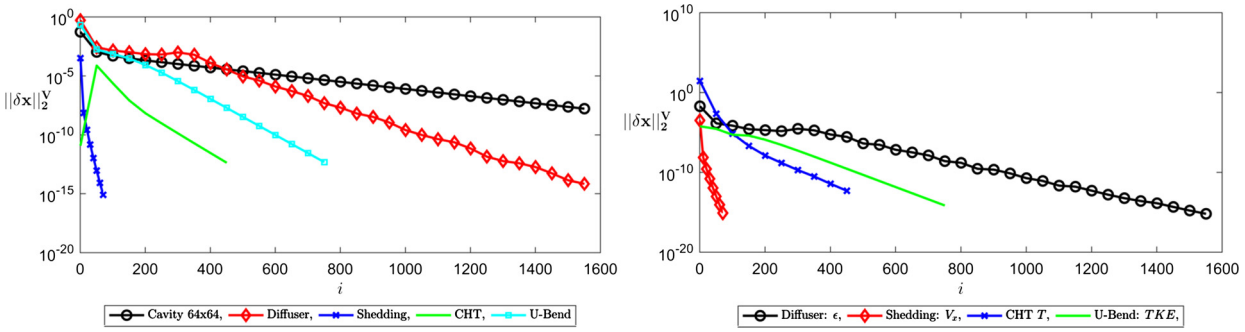

Fig. 5. Change in the solution as a function of iteration number for various problems. (a) Change in the x-momentum (b) change in the turbulent kinetic energy solution.

A straight line on a log plot implies that $\|\delta \mathbf{x}^{n+1}\|_2^V = \alpha \|\delta \mathbf{x}^n\|_2^V$ where $\alpha < 1$. Alternatively this implies $\|\delta \mathbf{x}^n\|_2^V = \alpha^{(n-1)} \|\delta \mathbf{x}^1\|_2^V$ and therefore $\log \|\delta \mathbf{x}^n\|_2^V = \log \|\delta \mathbf{x}^1\|_2^V + (n-1)\log \alpha$. So the slope of the line on the log–linear plot is equal to $\log \alpha$.

We also know that the error at step $n+1$ is equal to all future increments that must be taken. Assume that all future increments will drop by the same value α. Then

$$\left\|\tilde{\mathbf{e}}^{n+1}\right\|_2^V = \left\|\delta \mathbf{x}^{n+2}\right\|_2^V + \left\|\delta \mathbf{x}^{n+3}\right\|_2^V + \cdots = \left(\alpha + \alpha^2 + \cdots\right)\left\|\delta \mathbf{x}^{n+1}\right\|_2^V = \frac{\alpha}{1-\alpha}\left\|\delta \mathbf{x}^{n+1}\right\|_2^V. \tag{8}$$

Is a closed form extrapolation estimate for the error. The tilde on the error indicates that this is an estimate, not the actual error. All that is need to determine the current error estimate from the existing solution increment is some knowledge of the convergence slope (α) in Fig. 5. Also note that $|\frac{\alpha}{1-\alpha}$ is also an estimate for the unknown constant (CD) needed by the classical error estimators described in section 4.

5.1. Two increment extrapolation

In the simplest case the previous increment can be used to estimate the slope, so $\alpha = \frac{\|\delta \mathbf{x}^{n+1}\|_2^V}{\|\delta \mathbf{x}^n\|_2^V}$. Then using two increments of the solution, the extrapolated error estimate is

$$\left\|\tilde{\mathbf{e}}^{n+1}\right\|_2^V = \frac{\left(\|\delta \mathbf{x}^{n+1}\|_2^V\right)^2}{|\delta \mathbf{x}^n\|_2^V - |\delta \mathbf{x}^{n+1}\|_2^V} \tag{9}$$

In this work we make a clear distinction between the actual error (no tilde) and the error estimate (tilde). The actual error is measured using a prior solution of the problem (every test problem is solved twice). The error estimate (tilde) is the goal of the paper, since most application do not want to solve the problem twice in order to get the actual error.

Fig. 6 shows the 2-increment extrapolated error (equation (9)) with a thick blue line versus the actual error (with the smooth thin black lines) for all six test cases. The error estimate can be noisy, but it is usually an over-estimate when it is having trouble extrapolating. This is good for a convergence test because it means the estimator does not give false exits from the iterative procedure when it is having trouble estimating. Other than the noise issue, this simple extrapolation is remarkably accurate.

When there is noise in the convergence of the solution the solution increments can grow in magnitude momentarily rather than decrease. In that situation, the result of equation (9) is a negative error estimate which is easy to ignore/filter. This is why Fig. 6 shows some regions with no estimate shown. Similarly, when convergence stalls and the increments do not change much from one iteration to the next, the denominator gets small and the error estimate tends to become large (overestimates the error). An example of this is the right-hand side of Fig. 6(c).

In what follows we show a reasonable method to smooth the estimator predictions using more data points.

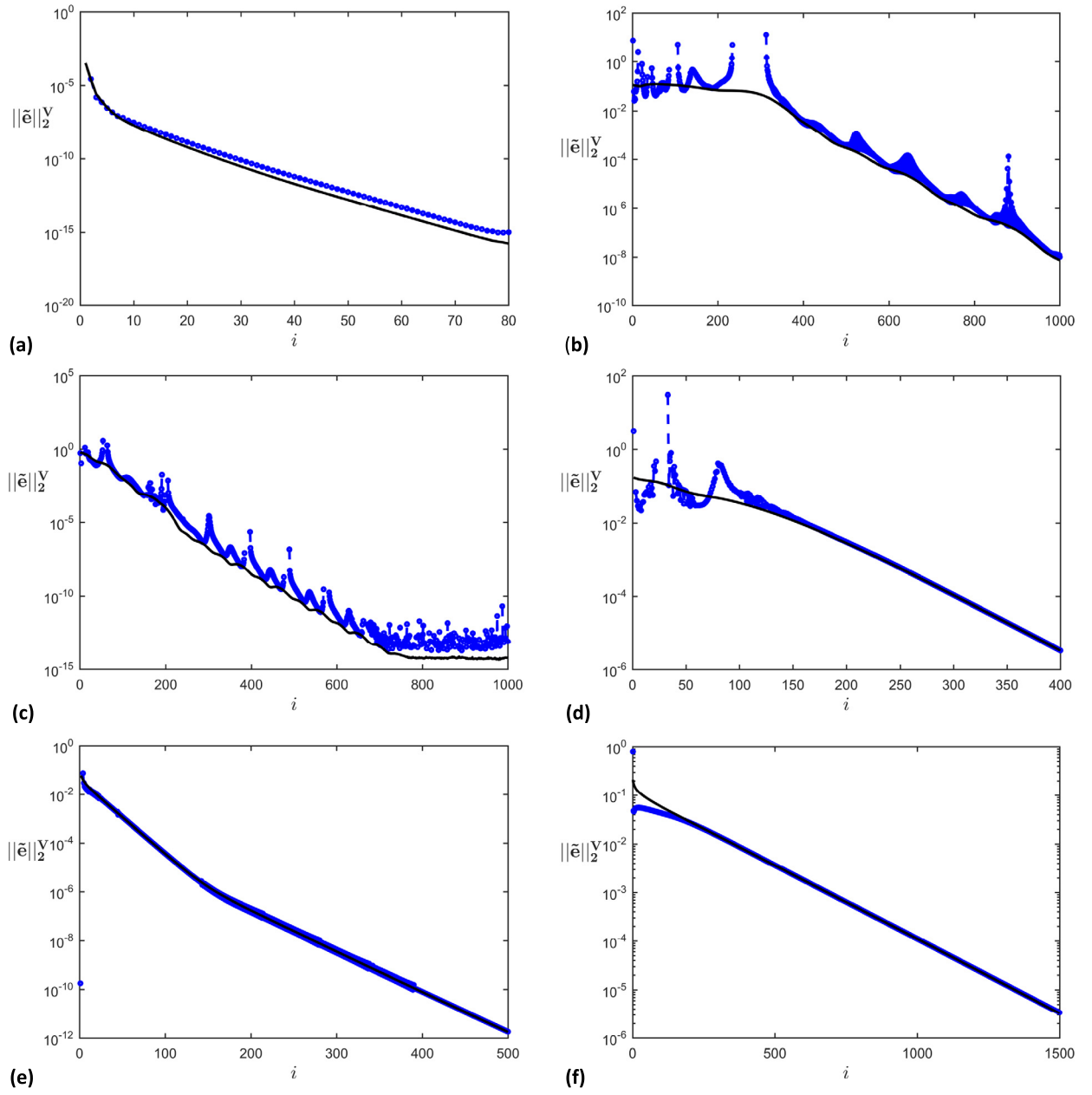

Fig. 6. The 2-increment predicted error (equation (9), thick dotted blue lines) versus the actual error (thin black lines) of the x-momentum for the six test cases: (a) Unsteady shedding, (b) Diffuser, (c) Boundary layer with realizable K-Eps, (d) U-Bend, (e) CHT, (f) Cavity 64×64.

5.2. Smooth increment extrapolation

A smooth extrapolation uses more data points. One approach is to use the two-increment estimate of the convergence rate α at many previous locations ($\alpha^{n+1} = \frac{\|\delta\mathbf{x}^{n+1}\|_2^V}{\|\delta\mathbf{x}^{n}\|_2^V}$, $\alpha^{n} = \frac{\|\delta\mathbf{x}^{n}\|_2^V}{\|\delta\mathbf{x}^{n-1}\|_2^V}$, $\alpha^{n-1} = \frac{\|\delta\mathbf{x}^{n-1}\|_2^V}{\|\delta\mathbf{x}^{n-2}\|_2^V}$, ...) and take some average of these estimates. There is ambiguity in what sort of average to use. There also remains a strong (and noisy) dependence on the most recent solution increment, $\|\delta\mathbf{x}^{n+1}\|_2^V$, because the estimate is given by $\|\tilde{\mathbf{e}}^{n+1}\|_2^V = \frac{\bar{\alpha}}{1-\bar{\alpha}}\|\delta\mathbf{x}^{n+1}\|_2^V$.

Much of this ambiguity can be removed by noting that the goal is to curve fit a line on a log–linear plot. The proposed smooth extrapolation approach therefore uses a least squares best-fit line through the data to extrapolate the slope and the intercept (which is approximately $\|\delta\mathbf{x}^{n+1}\|_2^V$). The error is then estimated from both these smooth data values. A least squares fit of exponentially varying data is not common so we repeat the derivation here.

Starting from the previous expression, $\ln\|\delta\mathbf{x}^{n}\|_2^V = \ln\|\delta\mathbf{x}^{1}\|_2^V + (n-1)\ln\alpha$ that is the assumed behavior for the increments, the goal is to find two constants, a and b, that best fit that equation. So

$$\ln\left\|\delta\mathbf{x}^{n+1}\right\|_2^V = a$$
$$\ln\left\|\delta\mathbf{x}^{n}\right\|_2^V = a - 1b$$
$$\vdots$$
$$\ln\left\|\delta\mathbf{x}^{n-M}\right\|_2^V = a - (M+1)b$$

where e^b is the best fit for the slope α, and e^a is the best fit line's approximation for the most recent solution increment $\|\delta\mathbf{x}^{n+1}\|_2^V$. This could be done just as easily with base-10 logs (and then $\alpha = 10^b$ and $10^a \approx \|\delta\mathbf{x}^{n+1}\|_2^V$), or base-2 logs for faster computation (using shift operations).

Note that $M+2$ is the number of data points being used in the curve fit. So $M=0$ for the 2-increment extrapolation (of the previous section). And $M=2$ for a 4 data-point smoothed extrapolation. M is the number of interior (or extra) smoothing data points.

This is an over-determined problem that is solved using least squares

$$\begin{bmatrix} \sum_{i=0}^{M+1} 1 & -\sum_{i=0}^{M+1} i \\ -\sum_{i=0}^{M+1} i & \sum_{i=0}^{M+1} i^2 \end{bmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{M+1} \ln\|\delta\mathbf{x}^{n+1-i}\|_2^V \\ -\sum_{i=0}^{M+1} i \ln\|\delta\mathbf{x}^{n+1-i}\|_2^V \end{pmatrix} \tag{10}$$

The right-hand side vector can be simplified to reduce the number of expensive log operations to just two. For example, $\sum_{i=0}^{M+1} i \ln\|\delta\mathbf{x}^{n-i}\|_2^V = \ln\left[\prod_{i=0}^{M+1} (\|\delta\mathbf{x}^{n-i}\|_2^V)^i\right]$. But this is a false optimization because the product rapidly causes underflow errors. So the form in equation (10) is retained.

The smooth estimated error is the sum of all future increments.

$$\|\tilde{\mathbf{e}}\| = e^{a+b} + e^{a+2b} + \ldots = e^{a+b}\left(1 + e^b + e^{2b} + \ldots\right) = \frac{e^b}{1-e^b} e^a \tag{11}$$

Remember that $e^b = \alpha$ and $e^a \approx \|\delta\mathbf{x}^{n+1}\|_2^V$ (the best fit line estimate for this value). The 2-increment extrapolation (equation (9)) from section 5.1 is equations (10) and (11) with $M=0$.

Fig. 7 shows this smooth error estimate compared to the actual error using different numbers of data points for the extrapolation. In order to better see the behavior a reduced range of iterations (400 to 700) is shown in the figure. This case is the noisiest case from Fig. 6, the boundary layer case (6c) and looks at the x-momentum error. The only free parameter in the least squares approach is the number of points to average over. An obvious trade-off exists here. More points gives smoother results, but takes longer to respond to the changes in the convergence behavior (changes in slopes) that occur as the iterations progress. Fewer points gives more locations where the extrapolation cannot make a good estimate but responds faster. For all averaging intervals, when the extrapolation has trouble it usually defaults to an overly conservative estimate that suggests that iterations should continue to proceed. This does not mean that the estimator can never under-predict the error. It can, but it does not do this by large amounts (that would cause premature stopping).

In this difficult boundary layer problem up to 50 prior points are needed to produce an extremely robust extrapolated error estimate. This interval size (50 iterations) is comparable to the scale at which the error (and solution changes) oscillate during the convergence. We do not suggest that it is imperative that such a long smoothing interval be used in general. The iterations can always continue when the estimator is not producing a reasonable (positive) result.

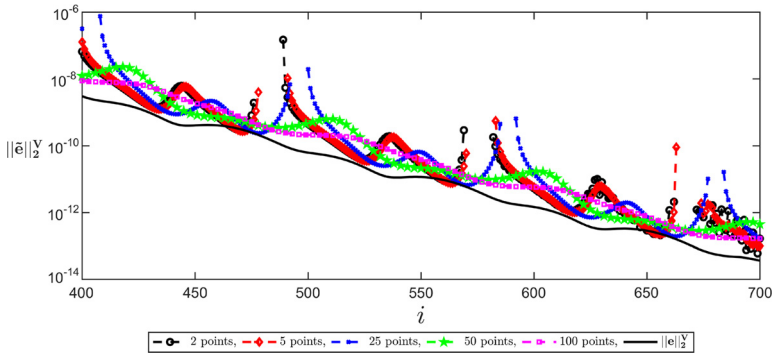

Fig. 7. 2, 5, 25, 50 and 100 – increment extrapolated error versus actual error in the x-momentum for the boundary layer problem with the realizable K-Epsilon turbulence model.

Note that this extrapolation approach differs from the classic approach in that it has completely abandoned the residual as a useful parameter as input for error estimation. It only uses the prior solution increments.

5.3. Hybrid methods

The smoothed extrapolation method of the previous section works remarkably well when it works. But there are a few situations when extrapolation doesn't work well. In most cases the extrapolation then returns an overly conservative estimate of the current error (which is the desired fault behavior for a stopping criterion). But the estimate can be further improved in situations where extrapolation fails by combining the classic estimator ideas from section 4. The primary problem with the classic estimators of section 4 is that the ratio $R \equiv \frac{\|\mathbf{e}\|_2^V}{\|\hat{\mathbf{r}}\|_2^V} = \frac{C}{\sigma_{\min}(V^{-1/2} J V^{-1/2})} = CD \frac{\|\delta\mathbf{x}\|_2^V}{\|\hat{J}\delta\mathbf{x}\|_2^V}$ requires a good estimate for the constants C or CD. In the hybrid method used in this work, the constant CD is determined when the extrapolation approach is working well, and then the saved (and averaged) values of CD are used with classic estimation whenever the extrapolation method is not working well.

The key issue in the hybrid method is therefore determining when the extrapolation is, or is not, working well. This is determined by running multiple error estimators (with different numbers of points). When these estimators produce similar (within 50%) and positive results, extrapolation is used. When they do not agree, the classic estimator is used (using previously calculated and saved values of CD).

Multiple error estimators do not necessarily involve significantly more computational work as many of the calculations overlap. In our estimator we use a 2-interval extrapolation and a 25-interval extrapolation. The 25-interval results are used for the stopping decisions and for calculating CD. The 2-interval extrapolation is used to test if the 25-interval method is working well. The 25-interval extrapolation uses fewer than 25 intervals if 25 intervals are not yet available. When either the 2-point or 25-point extrapolation calculation fails (produces a negative estimate or does not agree within 50% of the other) the algorithm assumes extrapolation is not working well. When extrapolation is not working well, CD is not calculated, and classic error estimation

$$\|\mathbf{e}\|_2^V = \overline{CD} \frac{\|\delta\mathbf{x}\|_2^V}{\|\hat{J}\delta\mathbf{x}\|_2^V} \|\hat{\mathbf{r}}\|_2^V \tag{12}$$

is used, and is based on the average of all previously calculated (from extrapolation) CD values.

With the hybrid method we are particularly interested in being able to capture convergence stall. Fig. 8 shows stall behavior for a turbulent boundary layer when the RNG K-epsilon turbulence model is used. This particular turbulence model causes a "chatter" in the solution. From iteration to iteration the solution flips back and forth but does not converge. Because of chatter, the residual, error, and solution changes do not decrease after roughly iteration 150. They are actually converging, but extremely slowly.

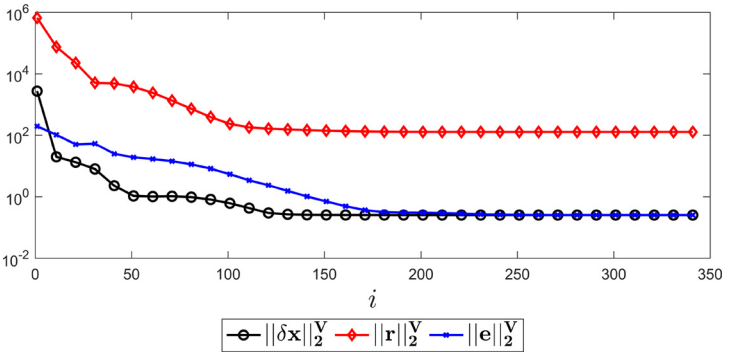

Fig. 8. x-Momentum convergence stall for the boundary layer problem using the RNG k/epsilon model.

The reasons for convergence stagnation are complex. It can be caused by discontinuous terms in the equations, which happens from slope limiters and from some turbulence models. It can also be a result of the underlying physics, when a steady solution is sought from an inherently unsteady or chaotic dynamical system. However, the task of the error estimator is not to diagnose why stagnation happens but to estimate the error reasonably when it does happen and present that information to the user.

Fig. 9 shows how the simple 25-interval estimator and the hybrid error estimator perform in this extreme situation. The figure shows that the actual error in the x-velocity component (thin black line) stagnates and does not converge with more iterations. The original 25-interval extrapolated error estimate (equations (10) and (11)) is the red line with diamonds. That error estimate is now growing with the iteration count. This is because the 25-interval extrapolation still produces a positive estimate in this situation because the solution is converging – just extremely slowly. But the 25-pt estimate is now

an increasingly poor estimate due to the stalled behavior of the solution changes. The estimator is correctly seeing that the error is not converging well but then it is extrapolating that this implies that the error must therefore be very large when it really the actual error is still only O(1). Finally, the Hybrid approach is also shown on Fig. 9 (as blue line with circles). In this case, the 2-interval extrapolation fails (gives a negative result) around iteration 125, so the hybrid method then resorts to a classic error estimator using the average CD found in the first 125 iterations.

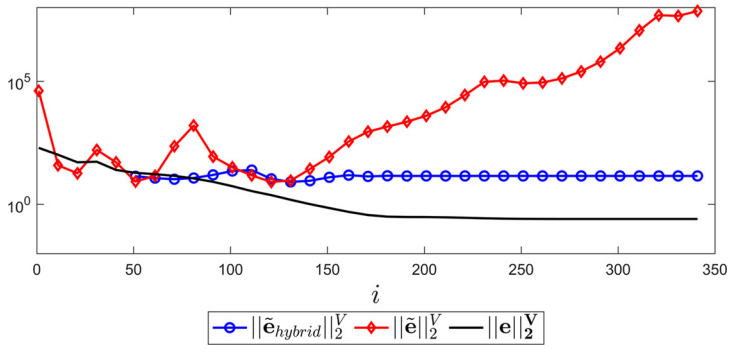

Fig. 9. Comparison of hybrid error estimate (blue circles), simple 25-interval error estimate (red diamonds) and actual error (black).

The predictions of the hybrid method are shown for all the test cases in Fig. 10. This is essentially Fig. 6 but with 25-interval smoothing implemented in the extrapolation (rather than using 2-interval extrapolation as in Fig. 6), and with a reversion to the classical estimation formula (equation (12)) when extrapolation is not working well.

6. Other considerations

6.1. Start-up

Most of the error estimators have large inaccuracy in the early iterations. The extrapolation procedure, for example requires a minimum of 2 iterations before it can even produce a result. For the classic estimators the constant, C, that turns the bound into an estimate is highly initial condition dependent during early iterations. If high predictive accuracy at early iterations is required, probably the only robust solution is to run a coarse mesh problem in order to estimate R. Fortunately, for a stopping criterion application, early iteration accuracy is usually not required. The errors are typically very large. The estimates, while inaccurate, are also very large, so stopping does not occur. A common practice is to put both a minimum and maximum number of iterations on the solver that overrides any error estimates.

6.2. Linear solvers

Non-linear PDE solvers often have one or more linear solvers as the core operations within one non-linear iteration. The linear solvers are also often iterative and also require a stopping criterion. The stopping criterion for the linear solvers is typically less stringent and less critical, because these linear solvers are embedded within a larger non-linear solver which will eventually enforce convergence even if the linear solvers do not. Nevertheless, there still are advantages to precise exiting from the linear solvers, and the stopping criterion methodology developed in this work can also be directly applied to those linear solvers. There is extensive prior work on stopping criteria for linear solvers. Two examples are references [26] and [27]. These methods often use scalars computed within the Krylov method itself [28,29], and often use error norms that include the matrix and the matrix preconditioner [30] which implies they use the residual. These approaches are very clever but are often specific to the particular Krylov solver. They typically estimate the matrix condition number (or minimum singular value) but not the constant C that turns the bound into an estimate. They also require additional coding, whereas the proposed method can be applied to all the solvers in a code, including the nonlinear outer iterations.

Given the linear problem, $A\bar{\mathbf{x}} = \mathbf{b}$ where A is a square invertible matrix, $\bar{\mathbf{x}}$ is the solution vector, and $\mathbf{b}$ is the input data. The residual for an inexact solution guess, $\mathbf{x}$ is given by $\mathbf{r} = \mathbf{b} - A\mathbf{x}$. And the error in the solution is given by $\mathbf{e} = \bar{\mathbf{x}} - \mathbf{x} = A^{-1}\mathbf{r}$. The triangle inequality then gives the classic relation for the error norm, $\|\mathbf{e}\| \leq \|A^{-1}\|\|\mathbf{r}\|$. So the matrix A replaces the Jacobian matrix. We now understand that even for linear systems, it is advantageous to use the volume norms and write,

$$\|\mathbf{e}\|_2^V \leq \frac{1}{\sigma_{\min}(V^{-1/2}AV^{-1/2})}\|\hat{\mathbf{r}}\|_2^V \tag{13}$$

if the matrix A is a FE or FV discretization (with a cell/element volume included in each matrix row).

All the methods discussed previously, including extrapolation, continue to apply. Some Krylov methods for unsymmetric matrices (such as BiCGStab and its variants) can have very noisy residual convergence. But it appears the solution increments are less noisy and smoothing over about 25 points can still provide useful extrapolation estimates. Methods like multigrid

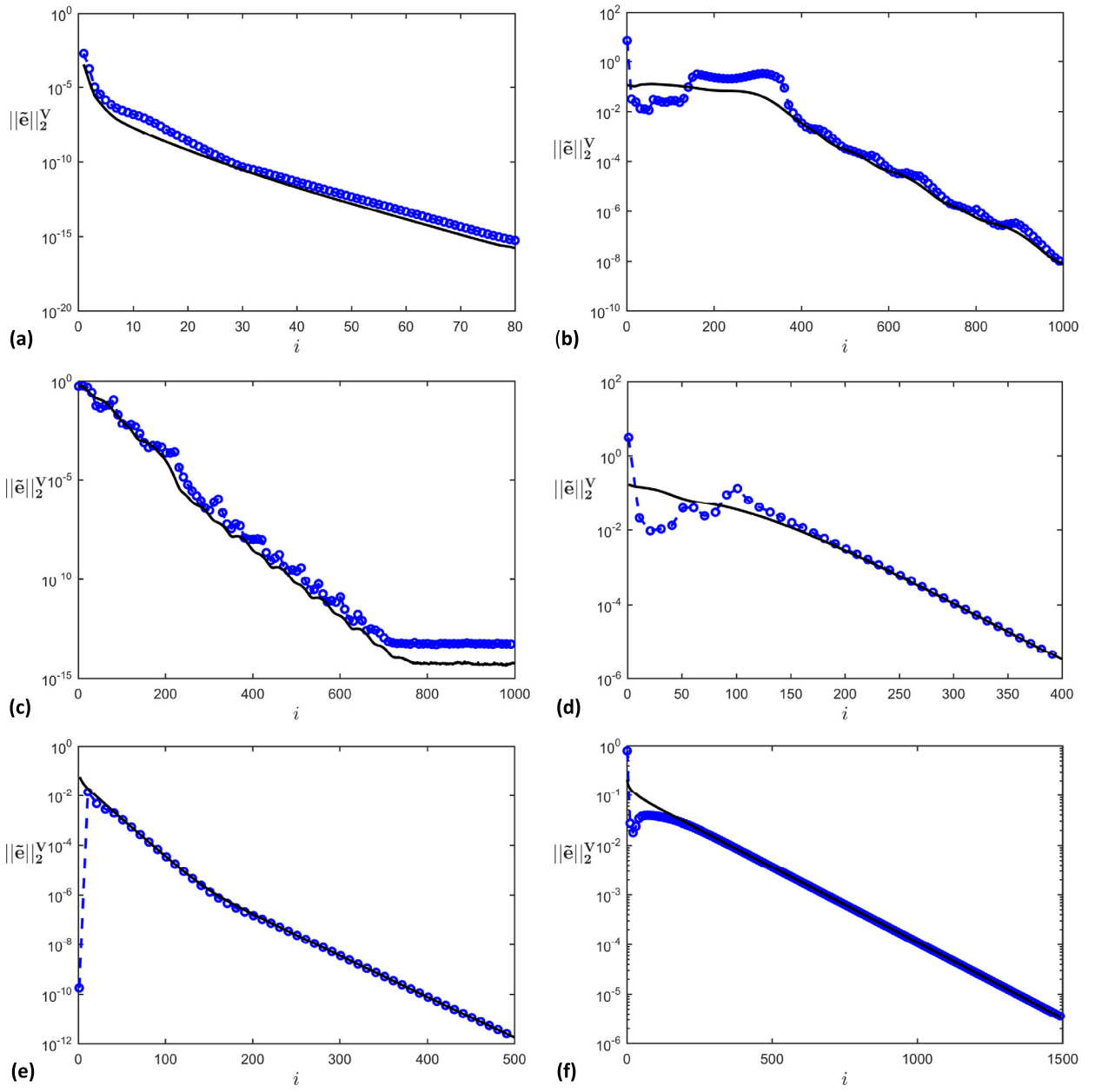

Fig. 10. The 25-increment hybrid error estimator (equation (10)–(12), blue lines with circles) versus the actual error (thin black lines) of the x-momentum for the six test cases: (a) Unsteady shedding, (b) Diffuser, (c) Boundary layer with realizable K-Eps, (d) U-Bend, (e) CHT, **(f)** Cavity 64×64.

and GMRES (or CG for symmetric matrices) tend to show much smoother (but not uniform) convergence. For many Krylov methods, the solution convergence is initially slow and then accelerates at the end. The means the extrapolation methods will over-estimate the error in the early iterations and will estimate it very well when it gets close to the actual stopping point. Note that there is a great deal of information and experience about how residuals behave for linear solvers, but the extrapolation estimates use solution increments where less has been published.

Fig. 11 shows the hybrid error estimator applied to various different linear solver cases. These results are for the 300th iteration of the nonlinear solver in the diffuser problem solved with the K-omega turbulence model (Section 3.2). Figs. 11(a) and 11(b) demonstrate the accuracy of the hybrid error estimator when using a preconditioned GMRES solver. Similarly Figs. 11(c) and 11(d) are error predictions using preconditioned AMG solver and finally Figs. 11(e) and 11(f) have been plotted using a preconditioned BiCGStab solver. All linear solvers use the PetSc implementation.

From the figures it can be seen that the classic estimate (with an assumed value of $CD = 15$, red line with circles) is fairly inaccurate and tends to under predict the error (note the use of a semi-log scale and the large range on the y-axis). On the

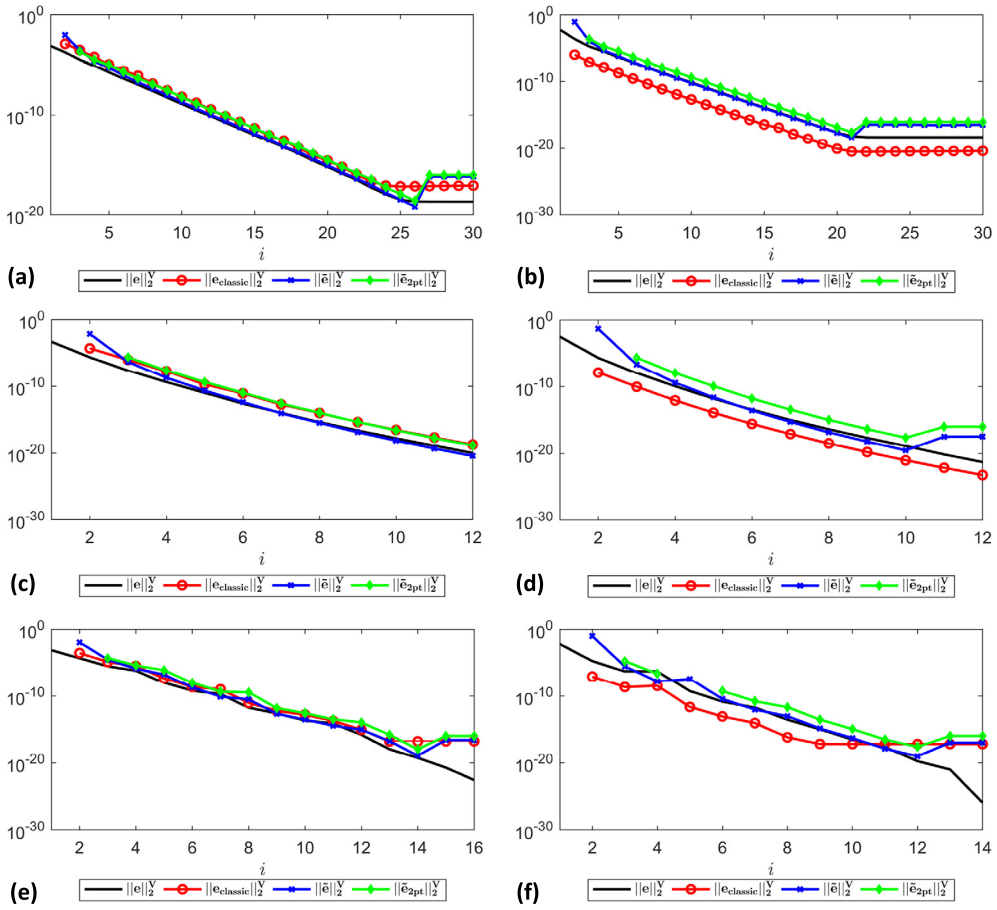

Fig. 11. This figure plots error and estimates for various linear solvers. All figures show the linear solver iterations for the 300th non-linear solver iteration for the diffuser problem using the K-omega turbulence model (discussed in Section 3.2). The actual error is the thin black line. The classic error estimator (eqn. (12) with $CD = 15$) is red circles. The hybrid method is shown with blue crosses, and the two-point extrapolation is shown with green diamonds. (a) Error in the x-momentum using GMRES, (b) error in the omega variable using GMRES, (c) error in the x-momentum using AMG, (d) error in the omega variable using AMG, (e) error in the x-momentum using BiCGStab, (d) error in the omega variable using BiCGStab.

other hand, the hybrid extrapolation estimate (blue line with crosses) and the simple 2-point extrapolation (equation (9)) (green line with diamonds) produce quite accurate error estimates. The only large deviations for the extrapolation estimates occur when the error has reached machine precision, or for the first 2 or 3 iterations of the solver. The results for the BiCGStab case are particularly notable because the erratic residual convergence is present in this test, but the increment and error convergence is much smoother (though sometimes not monotonic) and the method continues to produce useful error estimates.

6.3. Machine precision

Error that has a size roughly equal to machine precision, μ, produces a residual of size $\|\mathbf{r}\| \approx \mu\|\mathbf{b}\|$. It is therefore impossible for the residual to go below this level, so a stopping criterion should also always terminate when the residual goes below some slightly higher level, such as $1000\mu\|\mathbf{b}\|$. The equivalent lower limit for the error is $1000\mu\|\mathbf{x}\|$. In double precision this is a minimum relative error, $\|e\|/\|\mathbf{x}\|$, of about 10^{-13}. The iterative solver should not allow the user to request a relative error lower than this level. It can be seen that both the linear solvers (Fig. 11(a) and 11(b)) and non-linear solvers (Fig. 10(c)) can be impacted by this machine precision lower bound on the error.

7. Discussion

Stopping criteria for iterative methods are a very small but surprisingly important part of iterative numerical methods. Prior work on the stopping criteria has perhaps been hindered by the desire to black box this problem and treat it purely as a mathematical/numerical issue. In this work, we have attempted to do just the opposite, and use information about the context of the larger problem (PDE solution) to aid and enhance the error estimation process.

The first important manifestation of including the PDE information into the stopping criteria problem was the emphasis on volume norms. These norms are well known to the FE literature, which often present error analysis in terms of integral norms. The volume norms presented in this work have a very similar theoretical basis to integral norms, but are less costly to compute. Cost is always an important consideration when dealing with problems (such as PDEs) that involve a million or more unknowns. Volume norms are simply volume-weighted dot products. Volume norms are useful because they produce a number that depends on the problem but does not depend on the mesh size and type.

Similarly, PDE context was found to be just as important for matrix norms. The popular PDE discretization methods, such as FV and FE methods, produce a discretization in which the matrix (and also the residual) has a cell/element volume in it. There is a great advantage to removing this cell/element volume (or dual element volume depending on the variable in question) from the matrix norm and from the residual norm. The resulting minimum singular value then reflects only the physics of the PDE and not the mesh size or its discretization.

Mesh independence of the matrix and vector norms has numerous advantages. First, it significantly aids the search for an error estimation method that can be universally applied. Second, it allows coarse mesh estimation to be used. Third, it means that on a highly stretched mesh the norm closely aligns with our physical intuition of the "solution variable's magnitude" (that the norm properly accounts for the contributions of the relatively few cells which still cover a large portion of the physical domain).

PDE context (and mesh independence) was used again when showing how the Rayleigh Quotient can be used to estimate the smallest singular value (of the volume weighted Jacobian). In the context of PDEs, the singular values have a physical meaning, and the smallest singular value is the lowest frequency mode of the PDE associated with the slowest spatially varying mode of the PDE. This work suggested using the solution increment as a proxy for this lowest frequency mode and this work demonstrated the surprising error prediction accuracy that such a crude (but physically reasonable) guess, can provide.

Finally, this work demonstrated how well error extrapolation from current iterative progress can work if the extrapolation is appropriately smoothed. In particular, our algorithm performs a least-squares curve fit to an assumed local exponential solution convergence. As a result there is no arbitrariness to the smoothing or averaging of the extrapolation. The only free parameter in the extrapolation error estimate is the number of prior data points to use for that extrapolation. We use this one free parameter to our advantage by computing multiple error estimates with different numbers of data points. Large variation between two extrapolation estimates provides a sanity check on the extrapolation procedure's effectiveness.

Extrapolation estimates differ fundamentally from classical stopping criteria in that they do not use the residual. We have used the fact the methods are fundamentally different to develop a hybrid method that reverts to the classic estimator when the solution convergence is too noisy for extrapolation. This often happens at early iterations or when the convergence stagnates or reaches machine precision. This hybrid method uses the good extrapolation estimates to precompute and store the ratio of the error to the residual, R, so that this constant is available if/when the reversion to classical (residual based) estimation is needed.

Acknowledgements

This work used time on the NSF XSEDE supercomputer, Stampede. The third author was supported in part by the National Science Foundation grants 1353942 and 1336502. The work presented in this article is patent pending and the application has published as U.S. Patent Publication No. 2017/0185707 A1.

References

[1] A.T. Patera, E.M. Ronquist, A general output bound result: application to discretization and iteration error estimation and control, Math. Models Methods Appl. Sci. 11 (2001) 685–712, http://dx.doi.org/10.1142/S0218202501001057.
[2] D. Meidner, R. Rannacher, J. Vihharev, Goal-oriented error control of the iterative solution of finite element equations, J. Numer. Math. 17 (2009) 143–172.
[3] R. Verfürth, A posteriori error estimation and adaptive mesh-refinement techniques, J. Comput. Appl. Math. 50 (1994) 67–83.
[4] M. Arioli, D. Duff, S. Ruiz, Stopping criteria for matrix anal, SIAM J. Matrix Anal. Appl. 13 (1) (1992) 138–144.
[5] T.F. Berry, J. Chan, J.M. Demmel, J. Donato, V. Dongarra, R. Eijkhout, C. Pozo, P.A. Philadelphia, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd ed., 1994.
[6] Y. Saad, Numerical Methods for Large Eigenvalue Problems, 2nd edition, SIAM, Philadelphia, 2011.
[7] J.B. Perot, Determination of the decay exponent in mechanically stirred isotropic turbulence, 1, http://aip.scitation.org/doi/full/10.1063/1.3582815, 2011.
[8] C.J. Zusi, J.B. Perot, Simulation and modeling of turbulence subjected to a period of uniform plane strain, Phys. Fluids 26 (2014), http://dx.doi.org/10.1063/1.4901188.
[9] C.W. Oosterlee, H. Bijl, H.X. Lin, S.W. De Leeuw, J.B. Perot, C. Vuik, P. Wesseling, Fast Solution Methods and Parallelization for Computational Science Applications Lecture Notes for course (wi4145TU) "Computational Science and Engineering," 2004.
[10] M. Zijlema, P. Wesseling, Higher-order flux-limiting schemes for the finite volume computation of incompressible flow, Int. J. Comput. Fluid Dyn. (2007) 89–109, http://dx.doi.org/10.1080/10618569808940844.
[11] C. Vuik, Termination Criteria for GMRES-Like Methods to Solve the Discretized Incompressible Navier–Stokes Equations, TUD Report 92-50, Faculty of Technical Mathematics and Informatics, Delft University of Technology, 1992, http://ta.twi.tudelft.nl/nw/users/vuik/papers/DUT-TWI-92-50.pdf.
[12] X.-W. Chang, C.C. Paige, D. Titley-Peloquin, Stopping criteria for the iterative solution of linear least squares problems, SIAM J. Matrix Anal. Appl. 31 (2009) 831–852, http://dx.doi.org/10.1137/080724071.
[13] G. Golub, J.M. Ortega, Scientific Computing: An Introduction with Parallel Computing, Academic Press, Inc., Boston, 1993.

[14] J. Peiro, S. Sherwin, Finite Difference, Finite Volume Methods for Partial Differential Equations, Handbook of Materials Modeling, ISBN 978-1-4020-287-5 Springer, 2005, p. 2415.
[15] S.V. Patankar, D.B Spalding, A calc. proced. heat mass momentum transf. threedimensional parabol. flows, Int. J. Heat Mass Transfer 15 (1971) 1787–1806.
[16] K. Wieghardt, W. Tillmann, On the Turbulent Friction Layer for Rising Pressure, NACA TM-1314, 1951, https://www.grc.nasa.gov/WWW/wind/valid/fpturb/NACA-TM-1314-Wieghardt-1951.pdf.
[17] V. Yakhot, S. a. Orszag, S. Thangam, T.B. Gatski, C.G. Speziale, Development of turbulence models for shear flows by a double expansion technique, Phys. Fluids 4 (1992) 1510–1520, http://dx.doi.org/10.1063/1.858424.
[18] T.-H. Shih, W.W. Liou, A. Shabbir, Z. Yang, J. Zhu, A new k–ϵ eddy viscosity model for high Reynolds number turbulent flows, Comput. Fluids 24 (1995) 227–238, http://dx.doi.org/10.1016/0045-7930(94)00032-T.
[19] F.R. Menter, Two-equation eddy-viscosity turbulence models for engineering applications, AIAA J. 32 (1994) 1598–1605, http://dx.doi.org/10.2514/3.12149.
[20] C.U. Buice, Experimental Investigation of Flow Through an Asymmetric Plane Diffuser, Stanford University, 1997.
[21] P.R. Spalart, S.R. Allmaras, A one-equation turbulence model for aerodynamic flows, Rech. Aérosp. 1 (1994) 5–21, http://dx.doi.org/10.2514/6.1992-439.
[22] M.S. Engelman, M.-A. Jamnia, Transient flow past a circular cylinder: a benchmark solution, Int. J. Numer. Methods Fluids 11 (1990) 985–1000, http://dx.doi.org/10.1002/fld.1650110706.
[23] M.A. Cruchaga, N.M. Nigro, M.A. Storti, D.J. Celentano, Computing past cylinder flows, Mec. Comput. XXI (2002) 462–475, http://www.cimec.org.ar/ojs/index.php/mc/article/viewFile/904/858.
[24] U. Ghia, K.N. Ghia, C.T. Shin, High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387–411, http://dx.doi.org/10.1016/0021-9991(82)90058-4.
[25] H. Weyl, Das asymptotische Verteilungsgestez der Eigenwert linearer partieller Di erentialgleichungen (mit einer Anwendung auf der Theorie der Hohlraumstrahlung), Math. Ann. 71 (1912) 441–479.
[26] C.C. Paige, M.A. Saunders, LSQR: an algorithm for sparse linear equations and sparse least squares, ACM Trans. Math. Softw. 8 (1982) 43–71, http://dx.doi.org/10.1145/355984.355989.
[27] M. Arioli, D. Loghin, A.J. Wathen, Stopping criteria for iterations in finite element methods, Numer. Math. 99 (2005) 381–410, http://dx.doi.org/10.1007/s00211-004-0568-z.
[28] P. Jiránek, Z. Strakoš, M. Vohralík, A posteriori error estimates including algebraic error and stopping criteria for iterative solvers, SIAM J. Sci. Comput. 32 (2010) 1567–1590, http://dx.doi.org/10.1137/08073706X.
[29] E.F. Kaasschieter, A practical termination criterion for the conjugate gradient method, BIT Numer. Math. 28 (2) (1988) 308–322, http://link.springer.com/article/10.1007/BF01934094.
[30] O. Axelsson, I. Kaporin, Error norm estimation and stopping criteria in preconditioned conjugate gradient iterations, Numer. Linear Algebra Appl. 8 (4) (2001) 265–286.