A posteriori analysis for iterative solvers for non-autonomous evolution problems

J. H. Chaudhry*, D. Estep†, V. Ginting‡, and S. Tavener§

Abstract. We derive, implement, and test a posteriori error estimates for numerical methods for a non-autonomous linear system that involve iterative solution of the discrete equations. We consider two iterations: the Picard iteration and the Jacobi iteration for solving the discrete matrix-vector equations. To carry out the analysis, we define an appropriate adjoint problem for the numerical approximations using the matricant. We present a number of examples with interesting characteristics to illustrate the effectiveness of the estimate. We also present a comparison between the a posteriori error estimate and a conceptually simpler estimate obtained with a “pseudo-adjoint” problem.

Key words. a posteriori error estimation, adjoint operator, discontinuous Galerkin finite element method, iterative schemes, matricant, Picard iteration,

1. Introduction. The range of scales and complexity of processes involved in practical scientific and engineering applications mean that obtaining highly accurate numerical results is prohibitively expensive if not impossible. Consequently, numerical error generally has a significant effect on model simulations used in scientific and engineering applications. This includes models involving stochastic differential equations, where numerical error generically introduces significant bias in computed stochastic results, e.g. see [14, 15]. For this reason, accurate computational estimation of the error in information obtained from numerical solutions of differential equations is a key element in a complete uncertainty quantification for computational modeling. Further, error estimates are required to construct adaptive algorithms.

One approach to accurate computational error estimation is an a posteriori analysis that employs adjoint problems, computable residuals, and variational analysis to produce estimates of the error in a specified quantity of interest (QoI) for a particular computation [9, 6, 7, 13, 2, 19, 1]. The introduction of the generalized Green’s function solving an appropriate adjoint (dual) problem is the key component

---

*Department of Mathematics, Colorado State University, Fort Collins, CO 80523 (jehanzeb@math.colostate.edu). J. H. Chaudhry's work is supported in part by the Department of Energy (DE-SC0005304).
†Department of Statistics, Colorado State University, Fort Collins, CO 80523 (estep@stat.colostate.edu). D. Estep’s work is supported in part by the Defense Threat Reduction Agency (HDTRA1-09-1-0036), Department of Energy (DE-FG02-04ER25620, DE-FG02-05ER25699, DE-FG02-07ER54909, DE-SC0001724, DE-SC0005304, INL00120133, DE00000000SC9279), Dynamics Research Corporation PO672TO001, Idaho National Laboratory (00069249, 00115474), Lawrence Livermore National Laboratory (B573139, B584647, B590495), National Science Foundation (DMS-0107832, DMS-0715135, DGE-0221595003, MSPA-CSE-0434354, ECCS-0700559, DMS-1065046, DMS-1016268, DMS-FRG-1065046, DMS-1228206), National Institutes of Health (#R01GM096192).
‡Department of Mathematics, University of Wyoming, Laramie, WY 82071 (vginting@uwyo.edu) V. Ginting's work is supported in part by the National Science Foundation (DMS-1016283), the Department of Energy (DE-SC0004982).
§Department of Mathematics, Colorado State University, Fort Collins, CO 80523 (tavener@math.colostate.edu). S. Tavener's work is supported in part by the Department of Energy (DE-FG02-04ER25620, INL00120133), Idaho National Laboratory (00069249, 00115474), Lawrence Livermore National Laboratory (B590495), National Science Foundation (DMS-1228206).
of this approach as the generalized Green’s function encapsulates stability properties of a solution. Employing numerical approximations of the Green’s function in construction of the error estimate provides the means to accurately account for the effects of stability on accuracy.

The adjoint to a linear operator between Banach spaces is an operator between the related dual spaces that satisfies the bilinear identity, see §3.1. Adjoint operators play a fundamental role in the investigation of properties of linear operators. For nonlinear operators, a standard construction of a useful adjoint problem involves first forming a perturbation equation describing the effects of perturbations on the solution, linearizing the perturbation equation assuming a small perturbation, and forming the adjoint of the linearized problem [24]. This is the approach that is generally used for a posteriori error estimation for nonlinear differential equations [13].

However, while aimed at an eminently practical goal, it is common for a posteriori error analyses to ignore several practical issues that can affect the accuracy of a computed estimate. One such issue is the linearization used to form the adjoint for a nonlinear problem. Abstractly, the problem is linearized around an average of the true and approximate solutions. In practice, of course, it is only possible to linearize around approximate solutions. Theoretically, it is commonly possible to prove that linearization is valid as long as the error remains small, e.g. [9, 13]. In practice, computed error estimates invariably become large at the point when linearization becomes an issue, and in that sense, it is reasonable to ignore linearization in the analysis and when computing estimates.

Another significant issue affecting the accuracy of numerical solutions and computational error estimates arises from the problem of solving the discrete equations that yield an approximate solution. The discretization of a differential equation often results in a system of discrete equations that is solved by some iterative method, e.g. a Picard (fixed) point iteration [22]. It is well known that iterative solution can have a strong effect on the stability and accuracy of a numerical approximation [10, 22]. Generally, if the iterative method converges and the iteration is carried out for a sufficiently large number of iterations, the effects of iteration on the numerical solution can be safely ignored in the a posteriori error analysis. In practice, however, it is common to find that relatively few iterations are used, and it may be unknown if the iteration is actually converging. In that case, the effects of iteration on stability and accuracy cannot be ignored [10, 22].

The difficulty in extending a posteriori error estimates to numerical methods employing some sort of iteration is defining an appropriate adjoint problem [10]. The effect of iterating a finite number of times in the solution of the discrete equation is to produce a discrete evolution operator that may or may not be close to the evolution operator associated with the original problem. This in turn affects the construction of suitable adjoint operators and the form of the a posteriori error analysis.

The subject of this paper is the derivation of accurate a posteriori error estimates for numerical methods employing iteration for the approximate solution of a linear, non-autonomous problem: Find $y \in \mathbb{R}^d$ solving,

$$
\begin{cases}
\dot{y} = A(t) y(t), & t \in (0, T], \\
y(0) = y_0,
\end{cases}
$$

where $\dot{y} = \frac{dy}{dt}$ and $A(t)$ is a $d \times d$ time dependent matrix. The two iterative methods considered are Picard (fixed point) iteration and the Jacobi iteration for solving a matrix-vector equation. The

\[1\] There are classes of discretizations, e.g. explicit, explicit-implicit, and operator-splitting methods, that avoid the necessity of iterative solution of the discrete equations. However, such discretization choices also impact stability and accuracy and the effects need to be treated in a posteriori error analysis [23, 11, 12, 5].
restriction to \((1.1)\) permits a sole focus on the effects of finite iteration, and avoids the complications arising from the linearization point mentioned above.

Our approach to defining the correct adjoint problem makes use of the matricant (or matrizant) \([18, 21]\), which provides a way to represent the solution in an expansion in terms of exponential functions. The original motivation for the matricant arises from the failure of eigenvalue-based analysis to determine the stability of nonautonomous problems. It is related to the WKB approximation (see the beautiful paper by H. and J. Keller [21]). We use the matricant to define an appropriate adjoint solution.

The analysis in this paper is particularly relevant to a posteriori error analyses that employ the solution of an adjoint equation. There are other ways to deal with the effects of iteration in a posteriori error analysis, both using adjoint problems, e.g. \([11, 3, 16, 17, 12, 5, 4]\), and other approaches \([20, 8]\).

An outline of the paper is as follows. In §2, we present two iterative discretizations for \((1.1)\): the iterative Galerkin finite element method and the Jacobi iteration for an implicit finite element method. In §3, we analyze the iterative Galerkin finite element and provide numerical examples. We analyze the Jacobi iteration for the implicit finite element method and provide examples in §4. We present details of various proofs in §5. In §6, we explore the connection between estimates derived with the proper adjoint problem and an analysis employing a “pseudo-adjoint” problem involving the transpose that is often used in the literature. §7 contains the conclusion.

2. Discretization. Using a finite element formulation for the discretization facilitates a posteriori error analysis. We recall the discontinuous Galerkin (dG) finite element method for the numerical solution of \((1.1)\) and introduce two numerical methods that arise in application of iterative solvers. We note that many common finite difference schemes can be formulated as a dG method plus suitable quadrature for the associated integrals, and the analysis is easily extended to these cases. Likewise, the analysis can be applied to the continuous Galerkin (cG) family of discretizations \([13]\).

2.1. The discontinuous Galerkin finite element method. We discretize \([0, T]\) as \(0 = t_0 < t_1 < t_2 < \cdots < t_N = T\) with \(\{\Delta t_n = t_n - t_{n-1}\}_{n=1}^N, \Delta t = \max_{1 \leq n \leq N} \{\Delta t_n\} \) and \(I_n = [t_{n-1}, t_n]\). The numerical approximation is in \(V_n^q\), which is the space of vectors of length \(d\) with each component belonging to the set of polynomials of degree \(q\) on the interval \(I_n\). We denote the jump across \(t_n\) by \([U]_n = U^+_n - U^-_n\), where \(U^+_n = \lim_{t \to t_n^+} U(t)\). The discontinuous Galerkin dG(q) finite element method is: Set \(Y_0^- = y_0\) and, for \(1 \leq n \leq N\), compute \(Y(t) \in V_n^q\) satisfying

\[
\int_{I_n} \left( \dot{Y} - A(t)Y, V \right) dt + ([Y]_{n-1}, V_{n-1}^+) = 0, \quad \forall V \in V_{n-1}^q, \tag{2.1}
\]

where \((\cdot, \cdot)\) denotes the \(\mathbb{R}^d\) Euclidean inner product. This is an implicit discretization method that requires solving a matrix-vector equation in order to produce the numerical solution.

2.2. Iterative Galerkin finite element method. In the first class of methods, on each time step \(\Delta t_n\), we compute some number of explicit approximations on a sequence of one or more time steps \(\Delta s_n \leq \Delta t_n\) before iterating over the (possibly larger) time step \(\Delta t_n\) for a predetermined number of iterations. We let \(L_n\) denote the number of smaller time steps \(\Delta s_n = \Delta t_n/L_n\) used to solve the system explicitly on each time interval \(I_n\). We set \(I_{t_n} = [t_{t_n-1}, t_{t_n}]\), where \(t_{t_n} = t_{n-1} + l \Delta s_n\). We define the space \(V_{l,n}^q\) as the set of polynomials of degree \(q\) on the interval \(I_{t_n}\). The approximations on each interval are sought in the space \(V^q(I_{t_n})\), where if \(U \in V^q(I_{t_n})\) then \(U|_{I_{t_n}} \in V_{l,n}^q\). After the \(L_n\)
explicit steps, we iterate the approximation over $I_n$. We let $M_n$ be the number of iterations used for each $I_n$. The iterative discontinuous Galerkin dG(q) finite element is given in Algorithm 1.

**Algorithm 1 Iterative Galerkin Finite Element Method**

```plaintext
for $n = 1$ to $N$ do
    Set $Y^{(0)}(t) = Y^{(M_{n-1})}(t_{n-1})$ for $t \in I_n$.
    for $m = 1$ to $M_n$ do
        Set $Y^{(m)}(t_{n-1}) = Y^{(M_{n-1})}(t_{n-1})$.
        for $l = 1$ to $L_n$ do
            Compute $Y^{(m)} \in V_{l,n}$ satisfying
            \[
            \int_{I_{l,n}} \left( \dot{Y}^{(m)} - A(t)Y^{(m-1)} \right) dt + \left[ [Y^{(m)}]_{l-1,n}, V_{l-1}^+ \right] = 0, \quad \forall V \in V_{l,n}.
            \]
        end for
    end for
end for
```

Note that choosing $L_n = M_n = 1$, $q = 0$ and the left-hand quadrature rule in Algorithm 1 corresponds to the explicit Euler method.

**2.3. Jacobi iteration for the implicit finite element method.** In practical implementations of (2.1), it is common to solve the discrete matrix-vector equations using an iterative method. This amounts to using an approximate matrix inverse in the solution of the implicit equations (2.1), and thus is nominally different than the fixed-point type of iteration used for the iterative Galerkin finite element method. However, it turns out to be possible to rewrite this iteration as a kind of fixed point iteration and then extend the a posteriori analysis to handle this different case.

For simplicity, we consider the Jacobi iteration to solve the linear system arising from using a quadrature rule on the dG(0) finite element method (2.1). We let $\{\alpha_{n,r}\}$ denote the quadrature weights associated with quadrature points $\{t_{n,r}\}$ in $[t_{n-1}, t_n]$. Employing the quadrature rule in (2.1) yields the discrete system

\[
(I - \overline{A}_n)Y_n = Y_{n-1}, \quad \overline{A}_n = \sum_{i=1}^{r} \alpha_{n,r} A(t_{n,r}).
\]

where $Y_n$ is the value of the discrete solution $Y$ at $t_n$. Note that for $r = 1$, $t_{n,r} = t_n$ and $\alpha_{n,r} = \Delta t$, we recover the well-known implicit Euler method. To simplify notation, we suppress the index denoting the time step on $\overline{A}_n$ below.

The Jacobi iteration to solve (2.3) is: Set $Y_n^{(0)} = Y^{(K_{n-1})}$ and for $1 \leq k \leq K_n$, compute

\[
Y_n^{(k)} = (I - \overline{D})^{-1}(Y_{n-1} + (\overline{L} + \overline{U}) Y_n^{(k-1)})
\]

where $\overline{D}$, $\overline{L}$, and $\overline{U}$ are the diagonal, lower triangular part and the upper triangular part of $\overline{A}$ on $I_n$, respectively.

**3. A posteriori analysis of the iterative Galerkin finite element method.** In this section, we derive the a posteriori error estimate for the iterative Galerkin finite element method (2.2).
3.1. Adjoint problems and the role of the matricant. The first step in the a posteriori error analysis is defining a suitable adjoint operator. Abstractly, the adjoint operator $\mathcal{L}^* : W^* \to X^*$ of an operator $\mathcal{L} : X \to W$ between Banach spaces $X, W$ is defined by the bilinear identity,

$$\langle \mathcal{L} \alpha, \beta \rangle = \langle \alpha, \mathcal{L}^* \beta \rangle, \quad \alpha \in X, \beta \in W^*, \quad (3.1)$$

where $\langle \cdot, \cdot \rangle$ denotes the duality pairing in the appropriate spaces. The bilinear identity includes the boundary conditions in the definition of the operators $\mathcal{L}$ and $\mathcal{L}^*$ for differential equations. For example, the adjoint associated with (1.1) on $I_n$ is,

$$\begin{cases}
-\dot{\varphi} = [A(t)]^T \varphi(t), & t \in [t_{n-1}, t_n), \\
\varphi(t_n) = \psi_n.
\end{cases} \quad (3.2)$$

Note that (3.2) is solved backwards in time and $\psi_n$ provides the necessary initial condition at $t_n$. Now we identify $X = W^* = H^1(t_{n-1}, t_n)$, $\langle \mathcal{L} \alpha, \beta \rangle = (\alpha(t_{n-1}), \beta(t_{n-1})) + \int_{t_{n-1}}^{t_n} (\dot{\alpha} - A(t) \alpha, \beta) \, dt$ and $\langle \alpha, \mathcal{L}^* \beta \rangle = (\alpha(t_{n-1}), \beta(t_{n-1})) + \int_{t_{n-1}}^{t_n} (\dot{\beta} - [A(t)]^T \beta) \, dt$. This identification satisfies the identity (3.1). Taking $\alpha = y$ and $\beta = \varphi$, (3.1), (1.1) and (3.2) lead to

$$\langle y_n, \psi_n \rangle = \langle z_{n-1}, \varphi_{n-1} \rangle, \quad (3.3)$$

where $y_n = y(t_n)$, $\varphi_{n-1} = \varphi(t_{n-1})$ and $z_{n-1} = y(t_{n-1})$ is the initial value for $y$ in $I_n$. Equation (3.3) is the formulation of the bilinear identity (3.1) for the solution $y$ of (1.1) and adjoint solution $\varphi$ of (3.2).

We note the approximation (2.2) is not a consistent discretization of (1.1) in the sense that for a finite $M_n$, the solution $Y^{(M_n)}(t)$ does not converge to $y(t)$ as $\Delta t$ goes to 0. Rather, (2.2) is a consistent discretization of the following “iterative problem” associated with (1.1): Compute $y^{(m)}(t)$ satisfying

$$\begin{cases}
\dot{y}^{(m)} = A(t) y^{(m-1)}(t), & t \in [t_{n-1}, t_n), \\
y^{(m)}(t_{n-1}) = z_{n-1},
\end{cases} \quad (3.4)$$

for $m = 1, 2, \cdots, M_n$ with initial value $y^{(0)}(t_{n-1}) = z_{n-1} = y^{(M_n-1)}(t_{n-1})$.

The adjoint $\varphi^{(m)}$ to (3.4) should yield the analogous identity, $(y^{(m)}_n, \psi_n) = (z_{n-1}, \varphi^{(m)}_{n-1})$. In order to define the correct adjoint, we use an important construction $Q_m(s, t)$ called the matricant ([18, 21]) that is defined recursively as,

$$Q_m(s, t) = I + \int_s^t \mathcal{A}(\tau) Q_{m-1}(s, \tau) \, d\tau, \quad \text{with} \quad Q_0(s, t) = I, \quad (3.5)$$

or

$$Q_m(s, t) = Q_{m-1}(s, t) + \mathcal{A}_m(s, t), \quad (3.6)$$

with $\mathcal{A}_0(s, t) = I$, and

$$\mathcal{A}_m(s, t) = \int_s^t \mathcal{A}(\tau) \mathcal{A}_{m-1}(s, \tau) \, d\tau, \quad m = 1, 2, 3, \cdots$$

The following lemma establishes important properties of the matricant. The proof of the lemma is given in §5.

**Lemma 3.1.** The following hold:
The solution of (3.4) can be expressed as

\[ y^{(m)}(t) = Q_m(t_{n-1}, t) \, z_{n-1}, \quad t \in (t_{n-1}, t_n), \quad m = 1, 2, 3, \ldots \tag{3.7} \]

(b) For sufficiently small \( t - t_{n-1} \),

\[ z_{n-1} = [Q_m(t_{n-1}, t)]^{-1} \, y^{(m)}(t). \tag{3.8} \]

(c) For sufficiently small \( t - s \),

\[ \lim_{m \to \infty} Q_{m-1}(s, t) \, [Q_m(s, t)]^{-1} = \lim_{m \to \infty} \left[ I + A_m(s, t) \, [Q_{m-1}(s, t)]^{-1} \right]^{-1} = I. \]

The following theorem casts the iterative problem (3.4) in a form having the same unknown on both the left and right hand sides, thus allowing for the definition of an adjoint.

**Lemma 3.2.** For sufficiently small \( \Delta t_n \), the iterative problem (3.4) is equivalent to

\[ \tilde{y}^{(m)} = A(t) \, Q_{m-1}(t_{n-1}, t) \, [Q_m(t_{n-1}, t)]^{-1} \, y^{(m)}(t), \quad t \in (t_{n-1}, t_n). \tag{3.9} \]

**Proof.** This follows from (3.7) and (3.8) in Lemma 3.1.

Lemma 3.2 yields the following adjoint problem and solution representation.

**Theorem 3.3.** For \( \psi_n \in \mathbb{R}^d \), let \( \tilde{\varphi}^{(m)}(t) \) satisfy

\[
\begin{aligned}
\tilde{\varphi}^{(m)} &= \left[ A(t) \, Q_{m-1}(t_{n-1}, t) \, [Q_m(t_{n-1}, t)]^{-1} \right]^\top \, \tilde{\varphi}^{(m)}(t), \quad t \in (t_{n-1}, t_n), \\
\tilde{\varphi}^{(m)}(t_n) &= \psi_n.
\end{aligned}
\tag{3.10}
\]

Then for \( y^{(m)} \) in (3.4) and sufficiently small \( \Delta t_n \),

\[ (y_n^{(m)}, \psi_n) = (z_{n-1}, \tilde{\varphi}^{(m)}_{n-1}). \tag{3.11} \]

**Proof.** Using integration by parts,

\[
0 = \int_{t_{n-1}}^{t_n} \left( y^{(m)}, \tilde{\varphi}^{(m)} \right) + \left[ A(t) \, Q_{m-1}(t_{n-1}, t) \, [Q_m(t_{n-1}, t)]^{-1} \right]^\top \tilde{\varphi}^{(m)}(t) \, dt \\
= (y_n^{(m)}, \psi_n) - (z_{n-1}, \tilde{\varphi}^{(m)}_{n-1}).
\]

3.2. Error analysis. To begin the analysis, we first define the QoI as,

\[ \mathcal{J}(y) = (y, \Psi), \tag{3.12} \]

where \( \Psi \in \mathbb{R}^d \). Now we decompose the error in the QoI as,

\[ (e^{(MN)}_N, \Psi) = (y_N - Y^{(MN)}_N, \Psi) = (\tilde{e}^{(MN)}_N, \Psi) + (\tilde{e}^{(MN)}_{N-}, \Psi), \tag{3.13} \]

where \( e^{(MN)}_N = y_N - Y^{(MN)}_N \) and \( \tilde{e}^{(MN)}_{N-} = y^{(MN)}_N - Y^{(MN)}_N \) are the iteration and discretization errors in the quantity of interest respectively at the final time. We estimate each component separately.
**Iteration error.** We begin with a representation of the iteration error in \((3.4)\).

**Lemma 3.4.** Given \(\psi_n \in \mathbb{R}^d\),
\[
\left( y_n - y_n^{(m)} , \psi_n \right) = \left( y_{n-1} - z_{n-1}, \varphi_{n-1} \right) + \left( z_{n-1}, \varphi_{n-1} - \tilde{\varphi}_{n-1}^{(m)} \right).
\]

**Proof.** Subtracting \((3.11)\) from \((3.3)\) yields
\[
\left( y_n - y_n^{(m)} , \psi_n \right) = \left( y_{n-1}, \varphi_{n-1} \right) - \left( z_{n-1}, \tilde{\varphi}_{n-1} \right).
\]
Adding and subtracting \((z_{n-1}, \tilde{\varphi}_{n-1})\) yields the result. \(\blacksquare\)

We obtain an error representation for the iteration error at the final time.

**Lemma 3.5.** The iteration error in the quantity of interest at the final time, \(t_N = T\), associated with \((3.4)\) can be written
\[
\left( \tilde{e}_N^{(M_N)}, \Psi \right) = \left( y(t_N) - y_N^{(M_N)}, \Psi \right) = \sum_{n=1}^{N} \left( E_n + F_n \right),
\]
where
\[
E_n = \left( y_{n-1}^{(M_{n-1})}, \varphi_{n-1} - \tilde{\varphi}_{n-1}^{(M_n)} \right), \quad F_n = \left( y_{n-1} - y_{n-1}^{(M_{n-1})}, \varphi_{n-1} - \tilde{\varphi}_{n-1}^{(M_n)} \right).
\]

**Proof.** We use Lemma 3.4 with \(z_{n-1} = y_{n-1}^{(M_{n-1})} = y_{n-1}^{(M_n)}\) and \(m = M_n\). Adding and subtracting terms gives
\[
\left( y_n - y_n^{(M_n)}, \psi_n \right) = \left( y_{n-1} - y_{n-1}^{(M_{n-1})}, \varphi_{n-1} - \tilde{\varphi}_{n-1}^{(M_n)} \right) + E_n + F_n.
\]
We set \(\psi_N = \Psi\) and \(\psi_n = \tilde{\varphi}_{n}^{(M_{n+1})}\) for \(n = N - 1, N - 2, \ldots, 1\) in \((3.15)\) to arrive at a recursive relation. Unwinding this recursive relation and noting that \(y_0 - y_0^{M_0} = 0\) completes the proof. \(\blacksquare\)

**Discretization error.** We set \(B^{(m)} = A(t) Q_{m-1} (t_{n-1}, t) [Q_m(t_{n-1}, t)]^{-1}\) and recall that \(\tilde{e}^{(m)} = y^{(m)} - Y^{(m)}\). Then,

**Lemma 3.6.** The contribution to the discretization error in the quantity of interest over a single time step \(I_n\) is expressed as,
\[
\left( \tilde{e}_n^{(m)} - \tilde{\varphi}_n^{(m)} \right) = \left( e_{n-1}^{(m)} - \varphi_{n-1}^{(m)} \right) + Q_{1,n} + Q_{2,n}
\]
where
\[
Q_{1,n} = \sum_{l=1}^{L_n} \left\{ -\left[ (Y^{(m)})_{l-1,n} , \tilde{\varphi}_{l-1,n}^{(m)} \right] - \int_{I_{l,n}} (Y^{(m)} - A(t) Y^{(m-1)}, \tilde{\varphi}^{(m)} ) dt \right\},
\]
\[
Q_{2,n} = \sum_{l=1}^{L_n} \left\{ \int_{I_{l,n}} (B^{(m)} Y^{(m)} - A(t) Y^{(m-1)}, \tilde{\varphi}^{(m)} ) dt \right\}.
\]
The proof is given in \(\S 5\).
**Total error.** We let \( e = y - Y^{(m)} \) denote the error.

**Theorem 3.7.** The error in the QoI at the final time \( t_N = T \) is given by

\[
\left( e_{(M_N)}^{(n)} , \Psi \right) = E + F + Q_1 + Q_2
\]

where

\[
E = \sum_{n=1}^{N} E_n , \quad F = \sum_{n=1}^{N} F_n , \quad Q_1 = \sum_{n=1}^{N} Q_{1,n} , \quad Q_2 = \sum_{n=1}^{N} Q_{2,n} ,
\]

**Proof.** Assuming \( e_0(M_0) = 0 \), (3.16) implies

\[
\left( e_{(M_N)}^{(n)} , \widetilde{\varphi}_{M_N}^{(n)} \right) = \sum_{n=1}^{N} Q_{1,n} + Q_{2,n}.
\]

Combining with (3.14) completes the proof. \( \blacksquare \)

### 3.3. Evaluation of the a posteriori error estimate.

In this section, we address computational issues that arise in practical evaluation of (3.17). Most of the issues are common to evaluation of the error representations for both the iterative Galerkin method and the Jacobi iteration applied to the implicit finite element method.

**Approximation of the adjoint solutions.** The error representation requires adjoint solutions \( \widetilde{\varphi}_{(m)}^{(n)} \) and \( \varphi \) solving (3.10) and (3.2) respectively. In Lemma 3.8 we show how to compute \( \widetilde{\varphi}_{(m)}^{(n-1)} \) without system inversion.

**Lemma 3.8.** The solution of (3.10) satisfies \( \widetilde{\varphi}_{(m)}^{(n-1)} = \widetilde{\varphi}_{(m)}^{(n-1)}(t) = [Q_m(t_{n-1}, t)] ^ \top \psi_n \).

To approximate \( \widetilde{\varphi}_{(m)}^{(n)}(t) \) at times \( t \in I_n \), we interpolate from the values of \( \widetilde{\varphi}_{(m)}^{(n)} \) in \( G \) intervals adjacent to \( I_n \). We denote this approximation to \( \widetilde{\varphi}_{(m)}^{(n)} \) by \( \widetilde{\varphi}_{(m)}^{(n)} \). In Lemma 3.9 we show how to estimate \( \varphi \) using \( \widetilde{\varphi}_{(m)}^{(n)} \).

**Lemma 3.9.** Provided that \( \Delta t_n \) is sufficiently small, then \( \lim_{m \to \infty} \| \varphi - \widetilde{\varphi}_{(m)}^{(n)} \|_{I_n} = 0 \), with \( \| y \|_{I_n} = \sup_{t \in I_n} \| y(t) \| \).

We denote the approximation to \( \varphi \) by \( \Phi \). The proofs of the lemmas are given in §5.

**Evaluating \( E \) and \( F \) in (3.17).** Equation (3.17) is an exact error representation. In a practical setting, we omit the term \( F_n \) for computing estimates. The justification is the following lemma, whose proof is given in §5.

**Lemma 3.10.** \( F_n \) is a higher order term as compared to \( E_n \).

The term \( E_n \) in (3.17) involves the unknown solution \( y^{(m)} \). In practice, we replace \( y^{(m)} \) by \( Y^{(m)} \), which is justified by observing that

\[
E_n = \left( y_{(n-1)}^{(M_n-1)} , \varphi_{(n-1)} - \widetilde{\varphi}_{(n-1)}^{(M_n-1)} \right) + \left( y_{(n-1)}^{(M_n-1)} - Y_{(n-1)}^{(M_n-1)} , \varphi_{(n-1)} - \widetilde{\varphi}_{(n-1)}^{(M_n-1)} \right),
\]

where the second term on the right is a product of errors, and hence of higher order than the first term.
Evaluating (3.5) for the matricant $Q_m$ requires integration, which is carried out using quadrature in practice. However, naive implementation of quadrature schemes is not suitable for computation of $Q_m$. For example, an $R$-point Gauss quadrature rule is quite expensive, as evaluation of $Q_m(s,t)$ requires further evaluation of $Q_{m-1}(s,t_k)$ at the $R$ quadrature points $t_k$. Now, evaluation of $Q_{m-1}(s,t_k)$ at $R$ points also requires numerical integration, and involves evaluation of $Q_{(m-2)}$ and so on. Thus an $R$-point Gauss quadrature rule applied naively leads to $R^m$ function evaluations, which is computationally expensive. Other quadrature schemes, like the midpoint rule or the trapezoidal rule, may be too inaccurate in evaluating $Q_m$. As an alternative, we use the composite trapezoidal rule to evaluate $Q_m$ efficiently. We divide the interval $(s,t)$ into $R$ subintervals, $s = t_0 < t_1 < t_2 < t_3 \cdots < t_R = t$, and apply the trapezoidal rule in each subinterval $[t_k,t_{k+1}]$ as,

$$Q_m(s,t_{k+1}) = I + \int_s^{t_{k+1}} A(\tau) Q_{m-1}(s,\tau) \, d\tau$$

$$= I + \int_s^{t_k} A(\tau) Q_{m-1}(s,\tau) \, d\tau + \int_{t_k}^{t_{k+1}} A(\tau) Q_{m-1}(s,\tau) \, d\tau$$

$$= Q_m(s,t_k) + \int_{t_k}^{t_{k+1}} A(\tau) Q_{m-1}(s,\tau) \, d\tau$$

$$\approx Q_m(s,t_k) + 0.5(t_{k+1} - t_k) \left[ A(t_k) Q_{m-1}(s,t_k) + A(t_{k+1}) Q_{m-1}(s,t_{k+1}) \right]$$

Our algorithm for computing $Q_m(s,t)$ based on composite trapezoidal rule is given in Algorithm 2.

**Algorithm 2** Composite Trapezoidal quadrature to compute $Q_m(s,t)$

Set $Q_0(s,t) = I$, $A_0 = A(t_0)$.

for $k = 1$ to $R$ do

Set $A_k = A(t_k)$.

for $i = 1$ to $m$ do

$Q_i(s,t_k) = Q_i(s,t_{k-1}) + 0.5(t_k - t_{k-1}) \left[ A_{k-1} Q_{i-1}(s,t_{k-1}) + A_k Q_{i-1}(s,t_k) \right]$. \hfill (3.20)

end for

end for

This method is relatively efficient in that it requires only $R$ evaluations of the matrix $A(t)$ and $2Rm$ matrix-matrix multiplications.

**Evaluation of** $B^{(m)} = A(t)Q_{m-1}(t_{n-1},t) [Q_m(t_{n-1},t)]^{-1}$. This is required for evaluating (3.16). Computing the inverse of the matrix $Q_m(t_{n-1},t)$ is computationally expensive. However, Lemma 3.1 implies that $A(t)Q_{m-1}(t_{n-1},t)[Q_m(t_{n-1},t)]^{-1} \approx A(t)$ for $m$ sufficiently large. In practice, we find this approximation is poor for $m = 1$, but becomes sufficiently good to provide an accurate error estimate when $m = 2$. 


3.4. Numerical experiments. Now we specify the error estimate, \( \eta := (e^{\text{MC}}_N, \Psi) \) taking into account the approximations listed in the previous section,

\[
\eta := \sum_{n=1}^{N} (E_n + Q_{1,n} + Q_{2,n}),
\]

where,

\[
E_n = \left( Y^{(M_{n-1})}, \Phi_{n-1} - \tilde{\Phi}_{n-1} \right),
\]

\[
Q_{1,n} = \sum_{l=1}^{L_n} \left\{ -([Y^{(M_n)})_{l-1,n}, \tilde{\Phi}_{l-1,n}) - \int_{I_{l,n}} (Y^{(M_n)} - A(t)Y^{(M_{n-1})}, \tilde{\Phi}(M_n)) dt \right\},
\]

\[
Q_{2,n} = \sum_{l=1}^{L_n} \left\{ \int_{I_{l,n}} (B^{(M_n)}Y^{(M_n)} - A(t)Y^{(M_{n-1})}, \tilde{\Phi}(M_n)) dt \right\}.
\]

Here \( \Phi \) and \( \tilde{\Phi}(M_n) \) are the discrete approximations to \( \varphi \) and \( \varphi^{(M_n)} \) as discussed in §3.3.

We explore the effectiveness of the error estimate for the dG(0) method using a variety of examples. We summarize some notation in Table 3.1.

<table>
<thead>
<tr>
<th>Label</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Error</td>
<td>True value of ((e^{(M_N)}_N, \Psi))</td>
</tr>
<tr>
<td>Estimated Error</td>
<td>Estimate of ((e^{(M_N)}_N, \Psi)) by (\eta) in (3.21)</td>
</tr>
<tr>
<td>Estimated Iteration Error</td>
<td>Estimate of ((e^{(M_N)}<em>N, \Psi)) by (\sum</em>{n=1}^{N} E_n) in (3.21)</td>
</tr>
<tr>
<td>Estimated Discretization Error</td>
<td>Estimate of ((e^{(M_N)}<em>N, \Psi)) by (\sum</em>{n=1}^{N} Q_{1,n} + Q_{2,n}) in (3.21)</td>
</tr>
</tbody>
</table>

| Table 3.1                    | Notation used in the numerical experiments                            |

To judge the accuracy of the estimate, we compare the estimate to the exact error or an approximation to the error computed using a higher order method. One measure of accuracy is the effectivity ratio defined as,

\[
\rho_{\text{eff}} = \frac{\text{Estimated error}}{\text{True error}},
\]

provided the true error is not zero. An accurate error estimator has an effectivity ratio close to one. The results in the numerical experiments are shown for multiple simulations, with varying values of the final time, \( t_N = T \). Moreover we take the same number of iterations, \( M = M_n \), on each time interval \( I_n \).

3.4.1. A scalar problem. We consider the scalar ode,

\[
\dot{y} = \lambda y, \quad t \in (0, T],
\]

with initial value \( y(0) = 10 \). We set \( \lambda = -2 \), and the QoI is final value of \( y \) at \( t = T \).
Analysis of iterative evolution solvers

The accuracy of the estimate and the contributions of iterative and discretization errors in the quantity of interest are shown in Fig. 3.1 as the number of iterations is varied. The error estimate is quite accurate. For \( M = 1 \), the estimated iteration error is almost equal to the total error, whereas the estimated discretization error is almost zero. In this case, the estimated discretization error is measuring \((y^{(1)}_N - y^{(1)}_{N-1})\) and integration of (3.22) shows that \( y_n^{(1)} = y_n^{(1-)} \) for all \( n \). The estimated discretization error in the quantity of interest plays a dominant role for larger \( M \), since the estimated iteration error decreases as expected.

![Figure 3.1](image)

Figure 3.1. Results for the scalar problem in §3.4.1 as the number of iterations, \( M \), is varied. \( \Delta t = 0.01, L_n = 20 \). Here (+) true error, (−) estimated error, (○) estimated iteration error and (□) estimated discretization error.

### 3.4.2. The Vinograd problem.

The well-known Vinograd problem is,

\[
\begin{bmatrix}
  y_1 \\
  y_2
\end{bmatrix}
= -\begin{bmatrix}
  1 + 9\cos^2(6t) - 6\sin(12t) & -12\cos^2(6t) - \frac{9}{2}\sin(12t) \\
  12\sin^2(6t) - \frac{9}{2}\sin(12t) & 1 + 9\sin^2(6t) + 6\sin(12t)
\end{bmatrix}
\begin{bmatrix}
  y_1 \\
  y_2
\end{bmatrix},
\]

with initial conditions \([y_1(0) \ y_2(0)]^T = [-1 \ 3]^T\).

This equation can be solved analytically [7]. The component \( y_1(t) \) is plotted in Fig. 3.2. The QoI at the final time \( t_N = T \) is obtained by choosing \( \Phi = [1 \ 0]^T \). The performance of the estimator is shown graphically in Fig. 3.2 and in the form of effectivity ratios in Table 3.2. We observe that the estimate is remarkably accurate: the effectivity ratios are quite close to one, even when the error has a large magnitude. For example, in Fig. 3.2, the error is large around \( t = 1.75 \), where it is almost 40% of the solution. However, the error estimate is still accurate. We also note the significant decrease in the estimated iteration error in the quantity of interest as \( M \) increases from 1 to 2. The estimated discretization error in the quantity of interest is not zero for the case \( M = 1 \), but is significantly smaller than the estimated iteration error.

### 3.4.3. A multiscale system.

We consider a multiscale system in which the second component evolves at a relatively faster rate,

\[
\begin{bmatrix}
  y_1 \\
  y_2
\end{bmatrix}
= \begin{bmatrix}
  \sin(t) & \sin(3t) \\
  \sin(10t) & \sin(30t)
\end{bmatrix}
\begin{bmatrix}
  y_1 \\
  y_2
\end{bmatrix},
\]

with initial conditions \([y_1(0) \ y_2(0)]^T = [1 \ 1]^T\). The QoI is chosen to be the value at the final time in the second component, hence \( \Phi = [0 \ 1]^T \). We show the results for varying \( M \) in Fig. 3.3. Once again we see that estimated iteration error in the quantity of interest is dominant for \( M = 1 \), and estimated discretization error in the quantity of interest dominates for \( M = 2 \).
Table 3.2
Effectivity ratios for the Vinograd problem in §3.4.2

\[
\begin{array}{c|cc|c|cc|}
T & \rho_{\text{eff}} \text{ for } M = 1 & \rho_{\text{eff}} \text{ for } M = 2 & T & \rho_{\text{eff}} \text{ for } M = 1 & \rho_{\text{eff}} \text{ for } M = 2 \\
0.5 & 1.02 & 1.00 & 1.3 & 1.06 & 0.99 \\
0.7 & 1.02 & 0.99 & 1.5 & 1.16 & 0.99 \\
0.9 & 1.00 & 1.00 & 1.7 & 1.10 & 0.99 \\
1.1 & 1.06 & 0.99 & 1.9 & 1.10 & 0.99 \\
\end{array}
\]

3.4.4. Large dimensional systems obtained from semidiscretization of a reaction-diffusion equation. We consider the large dimension system of ordinary differential equations that arises upon discretizing the reaction-diffusion system,

\[
\begin{align*}
\frac{u_t(x,t)}{} &= (a(x,t)u_x)_x + r(x,t)u, & x \in (0,1), t \in (0,T], \\
u(1,t) &= u(0,t) = 0, & t \in (0,T], \\
u(x,0) &= \sin(\pi x), & x \in (0,1),
\end{align*}
\]  

(3.24)
in space. We assume sufficient regularity of $a$ and $r$ to insure the solution is twice differentiable in space and differentiable in time pointwise. We also assume $a$ is uniformly coercive. We discretize in space using a standard second order continuous piecewise linear finite element method using lumped mass quadrature (composite trapezoidal rule) on a uniform mesh with spacing $\Delta x$. The result is the system of ordinary differential equations (1.1), where $A(t) = A_a(t) + A_r(t)$ and $y$ is a vector containing nodal values of the spatial finite element approximation. Here $A_a(t)$ and $A_r(t)$ are banded matrices corresponding to spatial discretization of the diffusion and reaction terms respectively. The dimension $d$ is the number of nodes in the space discretization.

The heat equation. We consider the heat equation with $a = 1$ and $r = 0$. For the QoI, we choose the average value of $y$ over the spatial domain at the final time, $t_N = T$.

For the discretization of the partial differential equation, we require $\Delta t \leq 0.5\Delta x^2$ to ensure stability of an explicit method. In the experiments, we choose $\Delta x = 0.1$, and choose $\Delta t$ small enough to satisfy the stability condition so that the ordinary differential equation has the same stability properties as the original partial differential equation.

The results for varying $M$ are shown in Fig. 3.4. The error estimate is quite accurate.

A reaction-diffusion equation. We choose,

$$a(x, t) = 0.5 \left( \tanh(5.1(x - 0.5(\cos(2\pi t) + 1))) + 1.01 \right), \quad r(x, t) = \exp(1.01 + 1.8 \sin(4\pi t)).$$

The QoI is the average value of $y$ over the spatial domain at the final time $t_N = T$. The results for two different values of $M$ are shown in Fig. 3.5. The error estimate is accurate even through a transient in the solution around $t = 1.2$ that results in a substantial local increase in error.

4. Analysis of the Jacobi Iteration for the implicit discretization. In this section, we extend the analysis in §3 to the Jacobi iteration for the implicit finite element method.

4.1. A reformulation. For a finite $K_n$, (2.4) is not a consistent discretization of (1.1). But, we reformulate (2.4) and show it is a consistent discretization of a different iterative problem.

**Lemma 4.1.** Equation (2.4) is a consistent discretization of,

$$\begin{cases}
    y^{(k)}(t) = D(t)y^{(k)} + \widetilde{A}(t)y^{(k-1)}, & t \in (t_{n-1}, t], \\
y^{(k)}(t_{n-1}) = z_{n-1},
\end{cases} \quad (4.1)$$

**Figure 3.4.** Results for the heat equation in §3.4.4: $\Delta t = 0.001$, $L_n = 1$, $\Delta x = 0.1$. Here (⋆) true error, (−) estimated error, (○) estimated iteration error and (□) estimated discretization error.
where $D$ is the diagonal of $A$ and $\tilde{A} = L + U$ with $U$ and $L$ denoting the upper and lower triangular parts of $A$ respectively.

Proof. The dG(0) method applied to (4.1) is,

$$
\int_{I_n} (-D(t)Y^{(k)} - \tilde{A}(t)Y^{(k-1)}, V) + ([Y^{(k)}]_{n-1}, V^+_{n-1}) = 0, \quad \forall V \in V^0,
$$

Using the same quadrature rule as in (2.3) yields (2.4).

However, we cannot directly apply the earlier analysis to (4.1) because it is not in the same form as (3.4). Therefore, we perform a change of variables. First we define a diagonal matrix,

$$
S(r, t) = I + \int_t^r -D(s)ds + \int_t^r -D(s) \int_s^r D(\tau)d\tau ds + \cdots
$$

(4.2)

Note that $S(r, t)$ can be viewed as the limit for large $m$ of $Q_m(r, t)$ with $-D$ substituted for $A$.

Differentiating (4.2) with respect to $t$ yields,

$$
\frac{d}{dt} S(r, t) = -D(t) \left( I + \int_t^r -D(s)ds + \int_t^r -D(s) \int_s^r d\tau ds + \cdots \right) = -D(t) S(r, t).
$$

It follows,

Lemma 4.2. With $z_{n+1} = y^{(K_{n-1})}_{n-1}$, (4.1) is equivalent to the iterative problem,

$$
\begin{cases}
\dot{w}^{(k)}(t) = \tilde{A}w^{(k-1)}(t), & t \in [t_{n-1}, t_n], \\
 w^{(k)}_{n-1} = y^{(K_{n-1})}_{n-1},
\end{cases}
$$

(4.3)

where

$$
w^{(k)}(t) = S(t_{n-1}, t)y^{(k)}(t), \quad \text{for } t \in [t_{n-1}, t_n], \quad \tilde{A} = S(t_{n-1}, t)\tilde{A}[S(t_{n-1}, t)]^{-1},
$$

(4.4)

and $w^{(k)}_{n-1}$ denotes the initial condition for the interval $[t_{n-1}, t_n]$.

Proof. Differentiating $w$ with respect to $t$

$$
\dot{w} = -D(t)S(t_{n-1}, t)y + S(t_{n-1}, t)\dot{y}
$$

(4.5)
Multiplying (4.1) by \( S(t_{n-1}, t) \) we have,
\[
S(t_{n-1}, t)\dot{y}^{(k)} - D(t)S(t_{n-1}, t)y^{(k)} = S(t_{n-1}, t)\tilde{A}y^{(k-1)} = S(t_{n-1}, t)\tilde{A}[S(t_{n-1}, t)]^{-1}S(t_{n-1}, t)y^{(k-1)}
\]
Further, in a property similar to that of the matrix exponential, the inverse of \( S(t_{n-1}, t) \) is given by,
\[
[S(t_{n-1}, t)]^{-1} = I + \int_t^s D(s)ds + \int_t^s D(s) \int_r^s D(\tau)d\tau ds + \cdots
\]
Now, from the definition of \( w \) (4.4) and its derivative (4.5), we have \( \dot{w}^{(k)} = \tilde{A}w^{(k-1)} \). The initial condition, \( w^{(k)}_{n-1} \), follows by observing that \( S(t_{n-1}, t_{n-1}) = I \).

**4.2. Error analysis.** The analysis follows the line of argument used for the iterative Galerkin method in §3.2.

**Iteration error.** Now, (4.3) is of the same form as (3.4) and the earlier analysis applies. The analogue of (3.15) is,
\[
\begin{align*}
(w_n - w_n^{(K_n)}, \varphi_n^{(K_n)}) &= \left(w_{n-1}^{(K_n)} - w_{n-1}^{(K_n)}, \varphi_{n-1}^{(K_n)}\right) + \left(w_n^{(K_n)}, \varphi_n^{(K_n)}\right) + \left(w_{n-1}^{(K_n)}, \varphi_{n-1}^{(K_n)}\right) + F_n,
\end{align*}
\]
where \( w = S(t_{n-1}, t)y, \varphi^{(K_n)} \) is the adjoint solutions corresponding to (4.3) with initial condition,
\[
\varphi^{(K_n)} = \begin{cases} [S(t_{N-1}, t_N)]^{-1}\psi, & n = N, \\ [S(t_{n-1}, t_N)]^{-1}\varphi^{(K_{n+1})}, & \text{otherwise}. \end{cases}
\]
Here \( \varphi \) is the adjoint solution corresponding to the non-iterative version of (4.3) with the same initial conditions as given in (4.7) and \( F_n \) is the analogue of \( F_n \),
\[
F_n = \left(w_{n-1}^{(K_n)} - w_{n-1}^{(K_n)}, \varphi_{n-1}^{(K_n)}\right).
\]

The motivation for this definition is made clear in the proof of Lemma 4.3. Note that the error representation (4.6) is in terms of \( w^{(k)} \), not \( y^{(k)} \). We get an estimate in terms of \( y^{(k)} \) by substituting the expressions for \( w \) and \( w^{(k)} \).

**Lemma 4.3.** We have a local error representation for the solution to (4.1) as
\[
\left(y_n - y_n^{(K_n)}, \varphi_n^{(K_n+1)}\right) = \left(y_{n-1} - y_{n-1}^{(K_{n-1})}, \varphi_{n-1}^{(K_{n-1})}\right) + \left(y_{n-1}^{(K_{n-1})}, \varphi_{n-1}^{(K_{n-1})}\right) + \tilde{F}_n
\]

**Proof.** Using (4.4) we have,
\[
\left(w_n^{(K_n)}, \varphi_n^{(K_n)}\right) = \left(y_n^{(K_n)}, S(t_{n-1}, t_n)\varphi_n^{(K_n)}\right),
\]
and
\[
\left(w_{n-1}^{(K_n)} - w_{n-1}^{(K_n)}, \varphi_{n-1}^{(K_n)}\right) = \left(y_{n-1} - y_{n-1}^{(K_{n-1})}, \varphi_{n-1}^{(K_{n-1})}\right),
\]
and also,
\[
\left(w_{n-1}^{(K_n)}, \varphi_{n-1}^{(K_n)}\right) = \left(y_{n-1}^{(K_{n-1})}, \varphi_{n-1}^{(K_{n-1})}\right).
\]
Combining (4.6), (4.7), (4.9), (4.10) and (4.11) gives the result.

This leads to the following analogue of Lemma 3.5.

Lemma 4.4. The iteration error in the quantity of interest at the final time associated with iteration (4.1) is

$$
(y(T) - y_N^{(K_n)}, \psi) = \sum_{n=1}^{N} \left( y_{n-1}^{(K_n)}, \varphi_{n-1}^+ - \tilde{\varphi}_{n-1}^{(K_n)} \right) + F_n.
$$

Discretization error. Next we analyze the error arising from discretization of (4.1). We start with a technical result.

Lemma 4.5. We have the following result,

$$
(w_n^{(k)} - S(t_{n-1}, t_n)Y_n^{(k)}, \tilde{\varphi}_{n-1}^{(K_n)}) = (w_{n-1}^{(k)} - S(t_{n-2}, t_{n-1})Y_{n-1}^{(k)}, \varphi_{n-1}^+) - \left( \varphi_{n-1}^+ + \int_{t_{n-1}}^{t_n} \left( -S(t_{n-1}, t)Y(t) + B(k)S(t_{n-1}, t)Y(t, \tilde{\varphi}) \right) dt, \right.
$$

where $B(k) = \tilde{A}Q_{k-1}(t_{n-1}, t) [Q_k(t_{n-1}, t)]^{-1}$, and $Q_k$ is the matricant corresponding to the matrix $\tilde{A}$. The proof of the above lemma is given in §5.

Combining Lemma 4.5 with the observation that (2.4) is the dG(0) method for (4.1) with a particular choice of quadrature rule gives the following result.

Lemma 4.6. The discretization error in the quantity of interest at the final time $t_n = T$ is given by

$$
(y_N^{(k)} - Y_N^{(k)}, \psi) = \sum_{n=1}^{N} \begin{pmatrix} Q_{1,n} & + & Q_{2,n} & + & Q_{3,n} \end{pmatrix},
$$

where,

$$
Q_{1,n} = -\left( \varphi_{n-1}^+ + \int_{t_{n-1}}^{t_n} (D\varphi_{n-1}^+ + A\varphi_{n-1}) dt \right),
$$

$$
Q_{2,n} = \int_{t_{n-1}}^{t_n} (D\varphi_{n-1}^+ + A\varphi_{n-1}) dt - (D\varphi_{n-1}^+ + A\varphi_{n-1}) dt - (A\varphi_{n-1}) dt,
$$

$$
Q_{3,n} = \int_{t_{n-1}}^{t_n} \left( -W^{(k)} - Y^{(k)} + B(k)W^{(k)} - DY^{(k)} - \varphi_{n-1}^+ \right) dt.
$$

Here $\langle \cdot \rangle_r$ is the approximation of the integral $\int_{t_{n-1}}^{t_n} \cdot \cdot \cdot dt$ by the quadrature rule used in (2.3), and $W^{(k)} = S(t_{n-1}, t_n)Y^{(k)}$.

Proof. From the definition of $\tilde{\varphi}_{n-1}^{(K_n)}$ in (4.7) we have,

$$
(y_N^{(k)} - Y_N^{(k)}, \psi) = (w_n^{(k)} - S(t_{n-1}, t_n)Y^{(k)}_n, \varphi_{n-1}^+).
$$
Now unwinding the recursion in Lemma 4.5 gives,

\[
(y_N^{(k)} - Y_N^{(k)} - \psi) = \sum_{n=1}^{N} \left[ - ([Y^{(k)}]_{n-1}, \tilde{\varphi}_{n-1}^{(k)}) + \int_{t_{n-1}}^{t_n} \left( \frac{d}{dt} (-S(t_{n-1}, t)Y^{(k)}) + B^{(k)} S(t_{n-1}, t)Y^{(k)}, \tilde{\varphi}^{(k)}) \right) dt \right].
\]

Adding and subtracting appropriate terms yields the result.

**Total error.** We combine the component estimates to obtain,

**Theorem 4.7.** The error in the QoI at the final time \( t_N = T \) is,

\[
(e_N^{(K_N)})^-, \psi) = E + \tilde{F} + Q_1 + Q_2 + Q_3
\]

where

\[
E = \sum_{n=1}^{N} (y_{n-1}^{(K_n-1)}, \varphi_{n-1}^{(k)} - \tilde{\varphi}_{n-1}^{(K_n)}) , \quad \tilde{F} = \sum_{n=1}^{N} \tilde{F}_{n},
\]

\[
Q_1 = \sum_{n=1}^{N} Q_{1, n}, \quad Q_2 = \sum_{n=1}^{N} Q_{2, n}, \quad Q_3 = \sum_{n=1}^{N} Q_{3, n},
\]

4.3. **Numerical experiments.** In implementing the a posteriori estimate, we employ the approximations discussed in §3.3. We also drop \( \tilde{F} \), which is justified by noting that \( \tilde{F}_{n} \) is higher order. Finally, we truncate the series defining \( S(r, t) \) in (4.2) after a certain number of terms, which in our case was chosen to be 12. The validity of this truncation is provided by Lemma 3.9. In fact, since \( S(s, t) \) is a diagonal matrix, this computation is relatively cheap.

4.3.1. **The Vinograd problem.** The estimate for the error in the quantity of interest for the Vinograd problem (3.23) is shown in Fig. 4.1. The error estimate for the quantity of interest is quite accurate.

![Figure 4.1](image-url)
4.3.2. Semidiscrete heat equation. The error estimate for the quantity of interest for the semidiscrete heat equation (3.24) with $a = 1$ and $r = 0$ is shown in Fig. 4.2. The error estimate is quite accurate. Note that for $K = 3$ we have significantly more estimated iteration error than for $K = 6$ or $K = 10$, as shown in Table 4.1.

![Figure 4.2](image-url) Results for the Jacobi iteration for the semidiscrete heat equation. $\Delta t = 0.001$, $\Delta x = 0.1$. Here (+) true error; (−) estimated error, (⋄) estimated iteration error, (□) $Q_1$, (○) $Q_2$, (△) $Q_3$.

<table>
<thead>
<tr>
<th></th>
<th>K =3</th>
<th>K =6</th>
<th>K =10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>1.26e-04</td>
<td>8.37e-09</td>
<td>1.01e-14</td>
</tr>
<tr>
<td></td>
<td>5.70e-05</td>
<td>3.78e-09</td>
<td>4.60e-15</td>
</tr>
<tr>
<td></td>
<td>2.50e-05</td>
<td>1.66e-09</td>
<td>2.02e-15</td>
</tr>
<tr>
<td></td>
<td>1.07e-05</td>
<td>7.14e-10</td>
<td>8.69e-16</td>
</tr>
<tr>
<td></td>
<td>4.56e-06</td>
<td>3.02e-10</td>
<td>3.68e-16</td>
</tr>
<tr>
<td></td>
<td>1.90e-06</td>
<td>1.26e-10</td>
<td>1.54e-16</td>
</tr>
</tbody>
</table>

Table 4.1 Estimated Iteration Errors for the Jacobi iteration for the semidiscrete heat equation.

4.3.3. Semidiscrete reaction-diffusion equation. The error estimate for the semidiscrete reaction-diffusion equation (3.24) is shown in Fig. 4.3. Once again, the error estimate is quite accurate. Note that the estimated error for $K = 4$ is almost half to that of $K = 2$, illustrating the effect of finite Jacobi iteration for the implicit finite element method.

5. Proofs. We provide proofs of lemmas and theorems from previous sections.

5.1. Proofs from §3. Proof. [Lemma 3.1]
(a) The proof is based on induction on the iteration $m$ (see Gantmacher [18]). Integrating (3.4) over $(t_{n-1}, t)$ for $m = 1$ gives

$$y^{(1)}(t) = z_{n-1} + \int_{t_{n-1}}^{t} A(s) z_{n-1} ds = \left[ I + \int_{t_{n-1}}^{t} A(s) ds \right] z_{n-1} = Q_1(t_{n-1}, t) z_{n-1}. $$

Next assume that (3.7) is true for $(m - 1)$. We show that it also holds for $m$. We integrate (3.4) over
(t_{n-1}, t) for m to obtain

\[ y^{(m)}(t) = z_{n-1} + \int_{t_{n-1}}^{t} A(\tau) Q_{m-1}(t_{n-1}, \tau) z_{n-1} d\tau = Q_m(t_{n-1}, t) z_{n-1}. \quad (5.1) \]

The recursive relation (3.6) is established by similar induction argument.

(b) From (3.5),

\[
\int_{t_{n-1}}^{t} A(\tau) Q_{m-1}(s, \tau) d\tau = \int_{t_{n-1}}^{t} A(\tau_1) d\tau_1 + \int_{t_{n-1}}^{\tau_1} A(\tau_1) \int_{t_{n-1}}^{\tau_2} A(\tau_2) d\tau_2 + \cdots \\
+ \int_{t_{n-1}}^{\tau_1} A(\tau_1) \int_{t_{n-1}}^{\tau_2} A(\tau_2) \cdots \int_{t_{n-1}}^{\tau_m} A(\tau_m) d\tau_1 d\tau_2 \cdots d\tau_m
\]

Let \( \mu = \sup_{t \in [t_{n-1}, t]} \left( \max_{i,j} |a_{ij}(t)| \right) \), where \( a_{ij}(t) \) is the \((i^{th}, j^{th})\) entry of \( A(t) \). Defining \( \Delta t = (t - t_{n-1}) \), we bound each entry of the matrix \( \int_{s}^{t} A(\tau) Q_{m-1}(s, \tau) d\tau \) as [18],

\[
\Delta t \mu + \frac{d\Delta t^2 \mu^2}{2!} + \cdots + \frac{d^{m-1} \Delta t^m \mu^m}{m!} \leq \frac{1}{d} \sum_{k=1}^{\infty} \left( e^{d\mu \Delta t} - 1 \right)\]

For \( \Delta t \) sufficiently small, \( d\frac{1}{d} \left( e^{d\mu \Delta t} - 1 \right) < 0.5 \). Combining this with (3.5) shows that \( Q_m(t_{n-1}, t) \) is a strictly diagonally dominant matrix, and hence invertible, thus giving (3.8).

(c) Using (3.6),

\[
Q_{m-1} Q_m^{-1} = Q_{m-1} Q_{m-1}^{-1} \left[ I + A_m Q_{m-1}^{-1} \right]^{-1} = \left[ I + A_m Q_{m-1}^{-1} \right]^{-1}.
\]

Noting that the leading order in \( A_m [Q_{m-1}]^{-1} \) is \( O(||A|| \Delta t_n)^m \), for all \( \Delta t_n = t_n - t_{n-1} \) sufficiently small, we get the desired limiting behavior in (c). □

**Proof.** [Lemma 3.6] The continuity of \( y^{(m)} \) implies,

\[
\hat{e}_{l-1,n} = (y_{l-1,n}^{(m)} - \hat{y}_{l-1,n}^{(m)}) - (Y_{l-1,n}^{(m)} - Y_{l-1,n}^{(m)}) = \hat{e}_{l-1,n}^{(m)} = [Y^{(m)}]_{l-1,n}.
\]

**Figure 4.3.** Results for the Jacobi iteration for the reaction-diffusion equation. \( \Delta t = 0.001, \Delta x = 0.1 \). Here (\( \ast \)) true error; (\( \ast \)) estimated error; (\( \diamond \)) estimated iteration error; (\( \square \)) Q1, (\( \circ \)) Q2, (\( \triangle \)) Q3.
Using this on the time interval $I_{l,n}$ we have,

$$
0 = \int_{I_{l,n}} \left( \hat{\varphi}^{(m)} - B^{(m)} \right)^T \varphi^{(m)} dt \\
= (e_{l,n}^{(m)} - \tilde{\varphi}_{l,n}^{(m)}) - (e_{l-1,n}^{(m)} - [Y^{(m)}]_{l-1,n} - \tilde{\varphi}_{l-1,n}^{(m)}) + \int_{I_{l,n}} (\varphi^{(m)} - \hat{\varphi}^{(m)} + B^{(m)} \hat{\varphi}^{(m)}) dt. \tag{5.3}
$$

Noting that,

$$
\hat{\varphi} + B^{(m)} \hat{\varphi}^{(m)} = Y^{(m)} - B^{(m)} Y^{(m)},
$$

(5.3) gives,

$$
(e_{l,n}^{(m)} - \tilde{\varphi}_{l,n}^{(m)}) = (e_{l-1,n}^{(m)} - \tilde{\varphi}_{l-1,n}^{(m)}) \\
- ([Y^{(m)}]_{l-1,n} - \tilde{\varphi}_{l-1,n}^{(m)}) - \int_{I_{l,n}} (\tilde{\varphi}^{(m)} - \hat{\varphi}^{(m)} + B^{(m)} Y^{(m)}) dt.
$$

Adding and subtracting $(A(t) Y^{(m-1)} - \tilde{\varphi}^{(m)})$ leads to,

$$
(e_{l,n}^{(m)} - \tilde{\varphi}_{l,n}^{(m)}) = (e_{l-1,n}^{(m)} - \tilde{\varphi}_{l-1,n}^{(m)}) \\
- ([Y^{(m)}]_{l-1,n} - \tilde{\varphi}_{l-1,n}^{(m)}) - \int_{I_{l,n}} (\hat{\varphi}^{(m)} - (A(t) Y^{(m-1)} - \tilde{\varphi}^{(m)})) dt \\
+ \int_{I_{l,n}} (B^{(m)} Y^{(m)} - (A(t) Y^{(m-1)} - \tilde{\varphi}^{(m)})) dt.
$$

Applying this recursive relation proves the Lemma. \[\blacksquare\]

**Proof.** [Lemma 3.8] Expanding the right hand side of (3.10) and multiplying by $[Q_{m}(t_{n-1}, t)]^T$ yields

$$
-[Q_{m}(t_{n-1}, t)]^T \tilde{\varphi}^{(m)} = [Q_{m-1}(t_{n-1}, t)]^T A(t)^T \tilde{\varphi}^{(m)}(t). \tag{5.4}
$$

Taking the derivative of the equation in (3.5) with respect to $t$, we get

$$
\frac{d}{dt} Q_{m}(t_{n-1}, t) = A(t) Q_{m-1}(t_{n-1}, t),
$$

and thus

$$
\frac{d}{dt} [Q_{m}(t_{n-1}, t)]^T = [Q_{m-1}(t_{n-1}, t)]^T A(t)^T.
$$

We use this equality when we integrate (5.4) over $(t, t_n)$ and apply integration by parts on the left hand side to get

$$
-[Q_{m}(t_{n-1}, t_n)]^T \tilde{\varphi}^{(m)}(t_n) + [Q_{m}(t_{n-1}, t_n)]^T \tilde{\varphi}^{(m)}(t) \\
+ \int_t^{t_n} [Q_{m-1}(t_{n-1}, \tau)]^T A^T(\tau) \tilde{\varphi}^{(m)}(\tau) d\tau \\
= \int_t^{t_n} [Q_{m-1}(t_{n-1}, \tau)]^T A(\tau)^T \tilde{\varphi}^{(m)}(\tau) d\tau,
$$
from which we get
\[ [Q_m(t_{n-1}, t)]^\top \tilde{\varphi}^{(m)}(t) = [Q_m(t_{n-1}, t)]^\top \psi_n. \] (5.5)

Since \( Q_m(t_{n-1}, t_{n-1}) = I \), we get the desired result from (5.5).

**Proof.** [Lemma 3.9] Setting \( \xi^{(m)} = \varphi - \tilde{\varphi}^{(m)} \), we get the following equation
\[
\begin{cases}
-\xi^{(m)} = [A(t)]^\top \xi^{(m)}(t) + R^{(m)}(t), & t \in [t_{n-1}, t_n), \\
\xi^{(m)}(t_n) = 0,
\end{cases}
\] (5.6)
where
\[
R^{(m)}(t) = \left([A(t)]^\top - [A(t) Q_{m-1}(t_{n-1}, t) [Q_m(t_{n-1}, t)]^{-1}]^\top\right) \tilde{\varphi}^{(m)}(t)
= [A(t) - A(t) Q_{m-1}(t_{n-1}, t) [Q_m(t_{n-1}, t)]^{-1}]^\top \tilde{\varphi}^{(m)}(t)
= [I - Q_{m-1}(t_{n-1}, t) [Q_m(t_{n-1}, t)]^{-1}]^\top [A(t)]^\top \tilde{\varphi}^{(m)}(t).
\]

Using the same technique as in [18] and as in the previous paragraph, we arrive at the solution of (5.6) expressed as
\[
\xi^{(m)}(t) = \int_t^{t_n} R^{(m)}(\tau) \, d\tau + \int_t^{t_n} [A(\tau)]^\top \int_\tau^{t_n} R^{(m)}(\tau_1) \, d\tau_1 \, d\tau + \int_t^{t_n} [A(\tau)]^\top \int_\tau^{t_n} \int_\tau^{t_n} R^{(m)}(\tau_2) \, d\tau_2 \, d\tau_1 \, d\tau + \cdots.
\]

This gives the bound,
\[
\|\xi^{(m)}\|_{I_n} \leq \left( I + \Delta t_n \|A\|^2 \|I_n + \Delta t_n^2 \|A\|^2 \|I_n^2 + \cdots \right) \Delta t_n \|R^{(m)}\|_{I_n},
\]
from which we conclude the dependence of \( \xi^{(m)} \) on \( R^{(m)} \) for its convergence. Part (c) of Lemma 3.1 confirms the convergence of \( \xi^{(m)} \) to zero as \( m \to \infty \) and thus \( \lim_{m \to \infty} \tilde{\varphi}^{(m)}(t) = \varphi(t) \) for \( t \in I_n \).

**Proof.** [Lemma 3.10] The terms \( E_n \) and \( F_n \) have a common factor \( (\varphi_{n-1}^{(M_n-1)} - \tilde{\varphi}_{n-1}^{(M_n-1)}) \). We show that the factor \( (y_{n-1}^{(M_n-1)} - y_{n-1}^{(M_n-1)}) \) in \( E_n \) is of higher order than \( y_{n-1}^{(M_n-1)} \) in \( E_n \), making \( F_n \) higher order. From [18], we have
\[
\lim_{m \to \infty} (y(t) - y^{(m)}(t)) = 0. \quad (5.7)
\]
For \( t \in I_n = [t_{n-1}, t_n] \), integrating (3.4) leads to
\[
y^{(m)}(t) - y^{(m)}(t_{n-1}) = \int_{t_{n-1}}^t A(\tau) y^{(m-1)}(\tau) \, d\tau.
\]
For \( m = 1 \) and recalling that \( y^{(1)}(t_{n-1}) = y^{(0)}(t) \) yields
\[
\|y^1(t) - y^0(t)\| \leq \int_{t_{n-1}}^t \|A(\tau) y^0(t_{n-1})\| \, d\tau \leq \|A\| \|y^0(t_{n-1})\| \|t - t_{n-1}\| \leq C \tau_n,
\]
where \( \| \cdot \| \) denotes the Euclidean norm, \( C = \| y^{(0)}(t_{n-1}) \| \) and \( \tau_n = \| A \| \Delta t_n \). Now we establish the convergence rate of the iterative scheme. We have,

\[
\| y^2(t) - y^1(t) \| \leq \int_{t_{n-1}}^{t} \| A(\tau)(y^1(\tau) - y^0(\tau)) \| \, d\tau \leq C\tau_n^2.
\]

By induction, \( \| y^m(t) - y^{m-1}(t) \| \leq C\tau_n^m \). Choosing \( \Delta t_n \) so that \( \tau_n < 1 \), (5.7) implies,

\[
\| y(t) - y^{(m)}(t) \| \leq \lim_{l \to \infty} \sum_{k=m+1}^{l} \| y^{(k)}(t) - y^{(k-1)}(t) \| \leq \frac{C\tau_n^{m+1}}{1 - \tau_n}.
\]

Thus, \( \| y(t) - y^{(m)}(t) \| = O(\tau_n^{m+1}) \), and \( F_n \) is higher order than \( E_n \). \( \blacksquare \)

5.2. Proofs from §4. Proof. [Lemma 4.5] Integration by parts gives,

\[
0 = (w_n^{(k)-} - S(t_{n-1}, t_n))Y_n^{(k)-}, \tilde{\varphi}_n^{(k)-}) - (w_n^{(k)+} - S(t_{n-1}, t_n))Y_n^{(k)+}, \tilde{\varphi}_n^{(k)+})
\]

\[
+ \int_{t_{n-1}}^{t_n} \frac{d}{dt}(-w^{(k)} + S(t_{n-1}, t)Y^{(k)}) + B^{(k)}(w^{(k)} - S(t_{n-1}, t)Y^{(k)}), \tilde{\varphi}^{(k)}) \, dt
\]

Rearranging,

\[
(w_n^{(k)-} - S(t_{n-1}, t_n))Y_n^{(k)-}, \tilde{\varphi}_n^{(k)-}) = (w_n^{(k)+} - S(t_{n-1}, t_n))Y_n^{(k)+}, \tilde{\varphi}_n^{(k)+})
\]

\[
- \int_{t_{n-1}}^{t_n} \frac{d}{dt}(-w^{(k)} + S(t_{n-1}, t)Y^{(k)}) + B^{(k)}(w^{(k)} - S(t_{n-1}, t)Y^{(k)}), \tilde{\varphi}^{(k)}) \, dt
\]

From the property that \( S(t_{n-1}, t_{n-1}) = I \), (5.2) and the definitions of \( \tilde{\varphi}_n^{(K_n)-} \) in (4.7), we have,

\[
(w_n^{(k)+} - S(t_{n-1}, t_n))Y_n^{(k)+}, \tilde{\varphi}_n^{(k)+}) = (w_n^{(k)-} - S(t_{n-2}, t_{n-1})Y_n^{(k)-}, \tilde{\varphi}_n^{(K_n)-}) - ([Y^{(k)}]_{n-1}, \tilde{\varphi}_n^{(K_n)+}).
\]

Using the analogue of (3.9) for (4.3) we arrive at,

\[
\int_{t_{n-1}}^{t_n} \frac{d}{dt}(-w^{(k)} + S(t_{n-1}, t)Y^{(k)}) + B^{(k)}(w^{(k)} - S(t_{n-1}, t)Y^{(k)}), \tilde{\varphi}^{(k)}) \, dt
\]

\[
= - \int_{t_{n-1}}^{t_n} \frac{d}{dt}(-S(t_{n-1}, t)Y^{(k)}) + B^{(k)}(S(t_{n-1}, t)Y^{(k)}), \tilde{\varphi}^{(k)}) \, dt.
\]

Combining (5.8), (5.9) and (5.10) leads to the result. \( \blacksquare \)

6. Investigation of an estimate employing a “pseudo-adjoint”. Employing the adjoint of \( A(t) Q_{m-1}(t_{n-1}, t) [Q_m(t_{n-1}, t)]^{-1} \) arising in the Picard iteration problem (3.4) leads to significant complexity in the analysis and computational cost in computing the estimate. This provides motivation to consider the possibility of employing a “pseudo-adjoint problem” using \( A(t)^T \). Moreover, this pseudo-adjoint is often employed in a posteriori analyses found in the application literature.
Unfortunately, this does not yield a proper adjoint problem. However, it is still possible to derive a posteriori error estimates, see e.g. [12, 5].

In this section, we compare the a posteriori estimates corresponding to the true adjoint problem and the “pseudo-adjoint” replacement,

\[
\begin{aligned}
-\dot{y}^{(m)} &= [A(t)]^T \psi^{(m-1)}(t), \quad t \in [t_{n-1}, t_n), \\
\dot{y}^{(m)}(t_n) &= \psi_n.
\end{aligned}
\]  

(6.1)

This pseudo-adjoint problem does not yield a clear analogy of the identity (3.11). Instead, we can derive an estimate that has certain non-computable terms and show these terms are asymptotically smaller than the terms that are computable. In practice, dropping these terms significantly affects accuracy of the estimates.

### 6.1. A modified a posteriori error estimate.

Using the pseudo-adjoint \( \dot{y}^{(m)} \) solving (6.1), the following error representation formula holds ([5]).

**Theorem 6.1.** The error, \( e = y - Y^{(M_N)} \) at the final time \( t_N = T \) is,

\[
(e_N^{(M_N)}, \Psi) = \sum_{n=1}^{N} Q_{1,n} + Q_{2,n} + Q_{3,n} + Q_{4,n}
\]

(6.2)

where

\[
Q_{1,n} = \sum_{l=1}^{L} \left( -[Y^{(M_N)}]_{l-1,n}, \dot{y}^{(M_N)}_{l-1,n} \right) - \int_{t_{l,n}}^{t_{n}} \left( \dot{y}^{(M_N)} - A(t)Y^{(M_N-1)}, \dot{y}^{(M_N)} \right) dt,
\]

\[
Q_{2,n} = \sum_{l=1}^{L} \int_{t_{l,n}}^{t_{n}} \left( A(t)Y^{(M_N-1)} - A(t)Y^{(M_N)}, \dot{y}^{(M_N)} \right) dt,
\]

\[
Q_{3,n} = \left( y_n - y_{n-1}, \dot{y}^{(M_N)}_{n-1} \right),
\]

\[
Q_{4,n} = \left( y_n - y_{n-1}, \dot{y}^{(M_N)}_{n-1} - \dot{y}^{(M_N)}_{n-1} \right) - \sum_{l=1}^{L} \int_{t_{l,n}}^{t_{n}} \left( y^{(M_N)} - Y^{(M_N)}, [A(t)]^T (\dot{y}^{(M_N-1)} - \dot{y}^{(M_N)}) \right).
\]

The terms \( Q_{3,n} \) and \( Q_{4,n} \) are not computable, but are asymptotically small in the limit of large \( m \) so are dropped when computing the estimate [5]. However, ignoring the terms \( Q_{3,n} \) and \( Q_{4,n} \) affects the accuracy of the estimator. We compare (6.2) with (3.17) in a numerical example in §6.2.

Next we examine the difference in the pseudo-adjoint solution and true adjoint solution.

**Lemma 6.2.** The difference between the pseudo- and true adjoints is given by,

\[
\tilde{\varphi}^{(m)}(t) - \dot{y}^{(m)}(t) = \left( [Q_m(t_{n-1}, t_n)Q_m(t_{n-1}, t)^{-1}]^T - P_m(t, t_n), \psi_n \right), \quad t \in [t_{n-1}, t_n),
\]

where \( P_m \) is defined,

\[
P_m(s, t) = I + \int_s^t A^T(\tau)P_{m-1}(\tau, t) d\tau, \quad \text{with} \quad P_0(s, t) = I.
\]
Moreover, if \( A(t) = A \) is independent of \( t \), then \( \tilde{\varphi}(t_{n-1}) = \varphi^{(m)}(t_{n-1}) \).

**Proof.** [Lemma 6.2] From (5.5) we have,
\[
\tilde{\varphi}^{(m)}(t) = [Q_m(t_{n-1}, t_n)Q_m(t_{n-1}, t)^{-1}]^\top \psi_n.
\]

A derivation similar to (5.1) for (6.1) yields,
\[
\varphi^{(m)}(t) = \psi_n + \int_{t_{n-1}}^{t_n} A(\tau)^\top \int_{\tau}^{t_n} \varphi^{(m)}(\tau) \, d\tau = P_m(t, t_n)\psi_n. \tag{6.3}
\]

Subtracting (6.3) from (5.5) yields,
\[
\tilde{\varphi}^{(m)}(t_{n-1}) - \varphi^{(m)}(t_{n-1}) = \left([Q_m(t_{n-1}, t_n)Q_m(t_{n-1}, t)^{-1}]^\top - P_m(t_{n-1}, t_n)\right) \psi_n. \tag{6.4}
\]

Now, if \( A(t) = A \) does not depend on \( t \), then \( P_m(t_{n-1}, t_n) = [Q_m(t_{n-1}, t_n)]^\top \). Thus (6.4) implies,
\[
\tilde{\varphi}^{(m)}(t_{n-1}) - \varphi^{(m)}(t_{n-1}) = \left([Q_m(t_{n-1}, t_n)Q_m(t_{n-1}, t_{n-1})^{-1}]^\top - P_m(t_{n-1}, t_n)\right) \psi_n
\]
\[
= \left([Q_m(t_{n-1}, t_n)]^\top - [Q_m(t_{n-1}, t_n)]^\top \right) \psi_n = 0.
\]

### 6.2. Numerical experiment.

We compare the performance of the a posteriori error estimate (3.17) using the true adjoint with the estimate (6.2) employing the pseudo-adjoint obtained by dropping the terms \( Q_{3,n} \) and \( Q_{4,n} \) for the multiscale problem defined in §3.4.3. The results are shown in Fig. 6.1. We see that estimate using the true adjoint performs well in comparison.

![Figure 6.1](image-url)

**Figure 6.1.** Comparison of error estimate using \( \tilde{\varphi}^{(m)}(\times) \) and \( \varphi^{(m)}(\circ) \). Results for the multiscale problem in §3.4.3. \( \Delta t = 0.1, M = 2, L_n = 1 \). Here (×) is the true error.

### 7. Conclusions.

We developed a posteriori error estimates for a QoI for numerical methods for a non-autonomous linear system that involve iterative solution of the discrete equations. We considered two iterations: the Picard iteration and the Jacobi iteration for solving the discrete matrix-vector equations. To carry out the analysis, we defined the appropriate adjoint problem for the numerical approximations using the matricant. We presented a number of examples with interesting characteristics
to illustrate the effectiveness of the estimate We also presented a comparison between the a posteriori error estimate and a simpler, computationally cheaper estimate obtained with a “pseudo-adjoint” problem.

There are a number of possible future research directions related to this work. Extending the analysis to iterative numerical solution of non-linear partial differential equations is one interesting direction. Another interesting area of future research is the analysis of Newton-Krylov methods for solving the matrix-vector equations for the approximation.

**Acknowledgment.** D. Estep gratefully acknowledges Chalmers University of Technology for the support provided by the appointment as Chalmers Jubilee Professor.

**REFERENCES**


