Numerical Solution of Linear, Nonhomogeneous Differential Equation Systems via Padé Approximation

Kenneth C. Johnson

KJ Innovation

kjinnovation@earthlink.net

(First posted November 1, 2016, last revised December 21, 2016 [v7].)

http://vixra.org/abs/1611.0002

Abstract

This paper generalizes an earlier investigation of linear differential equation solutions via Padé approximation (viXra:1509.0286), for the case of nonhomogeneous equations. Formulas are provided for Padé polynomial orders 1, 2, 3, and 4, for both constant-coefficient and functional-coefficient cases. The scale-and-square algorithm for the constant-coefficient case is generalized for nonhomogeneous equations. Implementation details including step size initialization and tolerance control are discussed.

1. Introduction

An earlier study [1] investigated solutions of the linear differential equation $F'[x] = D[x]F[x]$ via Padé approximation: $F[h] \approx Q[h]^{-1} Q[-h]F[-h]$, where $Q[h]$ is a polynomial, D and Q are square matrices, and F may be a column vector or a multi-column matrix. (In this paper, square braces "[…]" delimit function arguments while round braces "(…)" are reserved for grouping.)

We consider here the more general nonhomogeneous equation,

$$F'[x] = D[x]F[x] + C[x]. \tag{1}$$

where C is a vector or matrix, size-matched to F. Eq. (1) can be recast in the form of a homogeneous equation,

$$\frac{d}{dx}\begin{pmatrix} F[x] \\ \mathbf{I} \end{pmatrix} = \begin{pmatrix} D[x] & C[x] \\ \mathbf{0} & \mathbf{0} \end{pmatrix}\begin{pmatrix} F[x] \\ \mathbf{I} \end{pmatrix}. \tag{2}$$

where "$\mathbf{I}$" is an identity matrix and "$\mathbf{0}$" is a zero matrix. For this case, Eq's. (7) and (8) in [1] result in relations of the form

$$\begin{pmatrix} Q[h] & R[h] \\ \mathbf{0} & \mathbf{I} \end{pmatrix}\begin{pmatrix} F[h] \\ \mathbf{I} \end{pmatrix} - \begin{pmatrix} Q[-h] & R[-h] \\ \mathbf{0} & \mathbf{I} \end{pmatrix}\begin{pmatrix} F[-h] \\ \mathbf{I} \end{pmatrix} = O\,h^{2n+1};\ Q[0] = \mathbf{I},\ R[0] = \mathbf{0}. \tag{3}$$

where Q and R are matrix polynomials. This simplifies to

$$Q[h]F[h]-Q[-h]F[-h]+R[h]-R[-h]=O\,h^{2n+1}. \tag{4}$$

The Q polynomial has the form given in [1]; it is determined from D and has no dependence on C. The R polynomial depends on both D and C and has a linear dependence on C. In some cases $R[h]$ is an odd function of h (i.e., $R[-h]=-R[h]$), in which case the $R[h]-R[-h]$ term in Eq. (4) is replaced by $2\,R[h]$.

In the case that C is constant, $R[h]$ has a right-factor of C,

$$R[h]=L[h]C \quad (\text{constant } C), \tag{5}$$

where $L[h]$ is a square matrix that has no C dependence. Eq. (4) is replaced by the following for this case,

$$Q[h]F[h]-Q[-h]F[-h]+(L[h]-L[-h])\,C=O\,h^{2n+1} \quad (\text{constant } C). \tag{6}$$

The homogeneous equation ($C=\mathbf{0}$ in Eq. (1)) has solutions of the form $F[x]=\Phi[x]F[0]$, where $\Phi[x]$ is the solution of the initial value problem,

$$\Phi'[x]=D[x]\Phi[x], \quad \Phi[0]=\mathbf{I}. \tag{7}$$

For the nonhomogeneous case, general solutions of Eq. (1) are of the form

$$F[x]=\Phi[x]\left(F[0]+\int_0^x \Phi[t]^{-1}\,C[t]\,dt\right). \tag{8}$$

For the special case of constant D, $\Phi[x]$ is an exponential matrix,

$$\Phi[x]=\exp[x\,D] \quad (\text{constant } D). \tag{9}$$

If C is also constant, Eq. (8) reduces to

$$F[x]=\exp[x\,D]F[0]+(\exp[x\,D]-\mathbf{I})\,D^{-1}\,C \quad (\text{constant } D \text{ and } C). \tag{10}$$

The factor $(\exp[x\,D]-\mathbf{I})\,D^{-1}$ is well defined by its Taylor series even when D is singular, and the factor can be robustly calculated by setting $F[0]=\mathbf{0}$ and $C=\mathbf{I}$ in Eq. (10).

If $C[x]$ can be an arbitrary linear combination of basis functions within a finite basis set, then particular solutions of Eq. (1) can be efficiently calculated by setting $F[0]=\mathbf{0}$ and setting $C[x]$ to a matrix containing all basis functions in its columns. The resulting $F[x]$ columns can be linearly combined to obtain particular solutions for any combination of $C[x]$ basis functions. The result can then be added to $\Phi[x]F[0]$ to obtain general solutions $F[x]$ for any $F[0]$.

Eq. (4) is used to integrate $F[x]$ across a small interval, from $x=-h$ to $x=h$. The independent variable x can be scaled and shifted to convert this to an integration from $x=x_0$ to $x=x_0+\Delta x$ for a sufficiently small Δx, and multiple such integrations are concatenated to calculate $F[x]$ over a large integration interval. For the homogeneous, constant-coefficient case (D constant, $C=\mathbf{0}$), the concatenation is efficiently implemented using a "scale-and-square" technique based on the relation

$$\exp[x\,D] = (\ldots((\exp[2^{-j}\,x\,D])^{\overbrace{2})^2\ldots)^2}^{j\times}. \tag{11}$$

(For some sufficiently large integer j, a Padé approximant is used to calculate $\exp[2^{-j}\,x\,D]$, and the result is squared j times to obtain $\exp[x\,D]$.) This algorithm can be generalized for the nonhomogeneous case with constant D and C.

This paper is organized as follows: Section 2 lists polynomial functions Q and R in Eq. (4) for various Padé polynomial orders. Section 3 outlines the scale-and-square algorithm, generalized for the nonhomogeneous case. Sections 4 and 5 discuss the choice of integration interval size based on error tolerancing. Appendix A derives the Padé polynomials, and Appendix B discusses MATLAB® implementation details, for the constant-coefficient case. Appendix C provides Mathematica code validating the results of section 2.

MATLAB® implementation code and application test cases are posted on the MathWorks File Exchange [2]. The algorithms and code incorporate and extend the functionality of MATLAB's **`expm`** function [3-5], and provide an efficient alternative to MATLAB's differential equation solvers [6] (e.g., **`ode45`**) for linear differential equations.

2. Padé-approximation formulas

The Q and R polynomials in Eq. (4) (or Q and L in Eq. (6)) are listed below for Padé polynomial orders 1, 2, 3, and 4, first for the case of constant D and C, and then for the non-constant case. For Padé order n, the approximation order is $2n$; i.e., the single-step approximation error is of order h^{2n+1}. (A formula for the approximation error is derived in Appendix A for the constant-coefficient case.)

Padé order 1, constant D, C:

$$\begin{aligned} &Q[h] = \mathbf{I} - h\,D \\ &L[h] = -h\,\mathbf{I} \\ &Q[h]\,F[h] - Q[-h]\,F[-h] + 2\,L[h]\,C = O\,h^3 \end{aligned} \tag{12}$$

Padé order 2, constant D, C:

$$\begin{aligned} &Q[h] = \left(\mathbf{I} + \tfrac{1}{3}h^2\,D^2\right) - h\,D \\ &L[h] = -h\,\mathbf{I} \\ &Q[h]\,F[h] - Q[-h]\,F[-h] + 2\,L[h]\,C = O\,h^5 \end{aligned} \tag{13}$$

Padé order 3, constant D, C:

$$\begin{aligned} &Q[h] = \left(\mathbf{I} + \tfrac{2}{5}h^2\,D^2\right) - \left(\mathbf{I} + \tfrac{1}{15}h^2\,D^2\right)h\,D \\ &L[h] = -\left(\mathbf{I} + \tfrac{1}{15}h^2\,D^2\right)h \\ &Q[h]\,F[h] - Q[-h]\,F[-h] + 2\,L[h]\,C = O\,h^7 \end{aligned} \tag{14}$$

Padé order 4, constant D, C:

$$\begin{aligned} Q[h] &= \left(\mathbf{I} + \tfrac{3}{7}h^2\,D^2 + \tfrac{1}{105}h^4\,D^4\right) - \left(\mathbf{I} + \tfrac{2}{21}h^2\,D^2\right)h\,D \\ L[h] &= -\left(\mathbf{I} + \tfrac{2}{21}h^2\,D^2\right)h \\ Q[h]\,F[h] - Q[-h]\,F[-h] + 2\,L[h]\,C &= O\,h^9 \end{aligned} \tag{15}$$

Eq's. (12)-(15) can be obtained from the following condition for Padé order n, in which Eq. (10) has been substituted with $x = \pm h$:

$$\begin{aligned} &Q[h]\,F[h] - Q[-h]\,F[-h] + 2\,L[h]\,C \\ &= Q[h]\left(\exp[h\,D]\,F[0] + D^{-1}\,(\exp[h\,D] - \mathbf{I})\,C\right) \\ &\quad - Q[-h]\left(\exp[-h\,D]\,F[0] + D^{-1}\,(\exp[-h\,D] - \mathbf{I})\,C\right) \\ &\quad + 2\,L[h]\,C \\ &= O\,h^{2n+1} \end{aligned} \tag{16}$$

Q and L are order-n polynomials of the following form (with L restricted to being an odd polynomial),

$$Q[h] = \sum_{j=0,1,\ldots n} q_j\,(h\,D)^j\,, \quad L[h] = \sum_{j=1,3,\ldots;\; j\le n} r_j\,(h\,D)^{j-1}\,h\,. \tag{17}$$

Considering separately the cases (1) $F[0] = \mathbf{I},\ C = \mathbf{0}$, and (2) $F[0] = \mathbf{0},\ C = \mathbf{I}$, the following two conditions follow from Eq. (16) with substitution from Eq's (17) and replacement of the exponentials with Taylor series,

$F[0] = \mathbf{I}, \quad C = \mathbf{0}$:

$$Q[h]\exp[h\,D] - Q[-h]\exp[-h\,D] = 2\sum_{\substack{k=1,3,\ldots\infty \\ j=0,1,\ldots\min[n,k]}} \frac{q_j}{(k-j)!}(h\,D)^k = O\,h^{2n+1} \tag{18}$$

$F[0] = \mathbf{0}, \quad C = \mathbf{I}$:

$$\begin{aligned} &2\,L[h] + \left(Q[h](\exp[h\,D] - \mathbf{I}) - Q[-h](\exp[-h\,D] - \mathbf{I})\right)D^{-1} \\ &= 2\,L[h] - \left(Q[h] - Q[-h]\right)D^{-1} + O\,h^{2n+1} = O\,h^{2n+1} \end{aligned} \tag{19}$$

Eq. (18) implies n conditions on the $n+1$ coefficients q_0, …, q_n:

$$\sum_{j=0,1,\ldots\min[n,k]} \frac{q_j}{(k-j)!} = 0\,, \quad k = 1,3,\ldots 2n-1\,. \tag{20}$$

With the supplemental condition $q_0 = 1$ (so that $Q[0] = \mathbf{I}$), Eq. (20) has the solution

$$q_j = \frac{n!(2n-j)!(-2)^j}{(2n)!(n-j)!\,j!}, \quad j = 0,1,\ldots n\,. \tag{21}$$

(Eq. (21) is derived in Appendix A.) Eq. (19) implies that

$$L[h] = \tfrac{1}{2}\left(Q[h] - Q[-h]\right)D^{-1}, \quad r_j = q_j \text{ for } j \text{ odd}\,. \tag{22}$$

The generalization of Eq's. (12)-(15) for any Padé order is

Padé order n, constant D, C:

$$\begin{aligned} Q[h] &= \sum_{j=0,1,\ldots n} \frac{n!(2n-j)!}{(2n)!(n-j)!\,j!}(-2hD)^j \\ L[h] &= \sum_{j=1,3,\ldots;\, j\le n} \frac{n!(2n-j)!}{(2n)!(n-j)!\,j!}(-2hD)^{j-1}(-2h) \\ Q[h]F[h] &- Q_n[-h]F[-h] + 2L[h]C = O\,h^{2n+1} \end{aligned} \tag{23}$$

Using an "n" subscript to indicate the Padé order, the polynomial coefficients for Q_n and L_n can be efficiently calculated from the following recursion relations,

$$\begin{aligned} Q_0[h] &= \mathbf{I}, \\ Q_1[h] &= \mathbf{I} - h\,D, \\ Q_{n+1}[h] &= Q_n[h] + \frac{h^2\,D^2}{(2n+1)(2n-1)}Q_{n-1}[h] \end{aligned} \tag{24}$$

$$\begin{aligned} L_0[h] &= \mathbf{0}, \\ L_1[h] &= -h\,\mathbf{I}, \\ L_{n+1}[h] &= L_n[h] + \frac{h^2\,D^2}{(2n+1)(2n-1)}L_{n-1}[h] \end{aligned} \tag{25}$$

For non-constant D and C, general formulas such as Eq's. (23) have not been developed, but several special cases are listed below. (Eq's. (26)-(29) are validated in Appendix C.)

Padé order 1, non-constant D, C:

$$\begin{aligned} Q[h] &= \mathbf{I} - h\,D[0] \\ R[h] &= -h\,C[0] \\ Q[h]F[h] &- Q[-h]F[-h] + 2R[h] = O\,h^3 \end{aligned} \tag{26}$$

Padé order 2, non-constant D, C:

$$\begin{aligned} Q[h] &= \mathbf{I} - h\left(-\tfrac{1}{6}D[-h] + \tfrac{2}{3}D[0] + \tfrac{1}{2}D[h]\right) + \tfrac{1}{3}h^2\,D[h]^2 \\ R[h] &= -h\left(-\tfrac{1}{6}C[-h] + \tfrac{2}{3}C[0] + \tfrac{1}{2}C[h]\right) + \tfrac{1}{3}h^2\,D[h]C[h] \\ Q[h]F[h] &- Q[-h]F[-h] + R[h] - R[-h] = O\,h^5 \end{aligned} \tag{27}$$

Padé order 3, non-constant D, C:

$$
\begin{aligned}
&Q[h] = \mathbf{I} - h\left(\tfrac{2}{45} D[-\tfrac{1}{2}h] + \tfrac{2}{15} D[0] + \tfrac{2}{3} D[\tfrac{1}{2}h] + \tfrac{7}{45} D[h]\right) + \\
&\quad \left(\tfrac{1}{15} D[-\tfrac{1}{2}h] + \tfrac{1}{5} D[0] + \tfrac{11}{15} D[\tfrac{1}{2}h]\right) \\
&\quad \left(\tfrac{2}{5} h^2 \left(\tfrac{1}{9} D[-\tfrac{1}{2}h] - \tfrac{1}{2} D[0] + D[\tfrac{1}{2}h] + \tfrac{7}{18} D[h]\right) - \tfrac{1}{15} h^3 D[h]^2\right) \\
&R[h] = -h\left(\tfrac{2}{45} C[-\tfrac{1}{2}h] + \tfrac{2}{15} C[0] + \tfrac{2}{3} C[\tfrac{1}{2}h] + \tfrac{7}{45} C[h]\right) + \\
&\quad \left(\tfrac{1}{15} D[-\tfrac{1}{2}h] + \tfrac{1}{5} D[0] + \tfrac{11}{15} D[\tfrac{1}{2}h]\right) \\
&\quad \left(\tfrac{2}{5} h^2 \left(\tfrac{1}{9} C[-\tfrac{1}{2}h] - \tfrac{1}{2} C[0] + C[\tfrac{1}{2}h] + \tfrac{7}{18} C[h]\right) - \tfrac{1}{15} h^3 D[h] C[h]\right) \\
&Q[h] F[h] - Q[-h] F[-h] + R[h] - R[-h] = O\, h^7
\end{aligned}
\tag{28}
$$

Padé order 4, non-constant D, C:

$$
\begin{aligned}
&L_1[h, X] = \tfrac{403}{16800} X[-h] - \tfrac{279}{2800} X[-\tfrac{2}{3}h] + \tfrac{99}{800} X[-\tfrac{1}{3}h] \\
&\quad + \tfrac{34}{105} X[0] - \tfrac{333}{5600} X[\tfrac{1}{3}h] + \tfrac{1719}{2800} X[\tfrac{2}{3}h] + \tfrac{1237}{16800} X[h] \\
&L_2[h, X] = \tfrac{57}{1120} X[-h] - \tfrac{243}{560} X[-\tfrac{2}{3}h] + \tfrac{1269}{1120} X[-\tfrac{1}{3}h] - \tfrac{3}{4} X[0] \\
&\quad + \tfrac{891}{1120} X[\tfrac{1}{3}h] + \tfrac{27}{112} X[\tfrac{2}{3}h] - \tfrac{41}{1120} X[h] \\
&L_3[h, X] = -\tfrac{2067}{9680} X[-h] + \tfrac{6021}{4840} X[-\tfrac{2}{3}h] - \tfrac{5805}{1936} X[-\tfrac{1}{3}h] + \tfrac{1863}{484} X[0] \\
&\quad - \tfrac{5697}{1936} X[\tfrac{1}{3}h] + \tfrac{10341}{4840} X[\tfrac{2}{3}h] - \tfrac{727}{9680} X[h] \\
&L_4[h, X] = \tfrac{63}{16} X[-h] - \tfrac{1809}{40} X[-\tfrac{2}{3}h] + \tfrac{2295}{16} X[-\tfrac{1}{3}h] - \tfrac{801}{4} X[0] \\
&\quad + \tfrac{2133}{16} X[\tfrac{1}{3}h] - \tfrac{297}{8} X[\tfrac{2}{3}h] + \tfrac{233}{80} X[h] \\
&L_5[h, X] = \tfrac{123}{160} X[-h] - \tfrac{135}{8} X[-\tfrac{2}{3}h] + \tfrac{2295}{32} X[-\tfrac{1}{3}h] - 132\, X[0] \\
&\quad + \tfrac{3861}{32} X[\tfrac{1}{3}h] - \tfrac{1917}{40} X[\tfrac{2}{3}h] + \tfrac{149}{32} X[h] \\
&L_6[h, X] = -\tfrac{6}{35} X[-h] + \tfrac{27}{10} X[-\tfrac{2}{3}h] - \tfrac{1053}{112} X[-\tfrac{1}{3}h] + \tfrac{57}{4} X[0] \\
&\quad - \tfrac{621}{56} X[\tfrac{1}{3}h] + \tfrac{729}{140} X[\tfrac{2}{3}h] - \tfrac{277}{560} X[h] \\
&Q[h] = \mathbf{I} - h\, L_1[h, D] + L_2[h, D]\left(\tfrac{121}{315} h^2 L_3[h, D] - \tfrac{2}{315} h^3 L_4[h, D] L_5[h, D]\right) \\
&\quad + \left(\tfrac{2}{45} h^2 L_6[h, D] + L_2[h, D]\left(-\tfrac{4}{45} h^3 L_6[h, D] + \tfrac{1}{105} h^4 D[h]^2\right)\right) D[h] \\
&R[h] = -h\, L_1[h, C] + L_2[h, D]\left(\tfrac{121}{315} h^2 L_3[h, C] - \tfrac{2}{315} h^3 L_4[h, D] L_5[h, C]\right) \\
&\quad + \left(\tfrac{2}{45} h^2 L_6[h, D] + L_2[h, D]\left(-\tfrac{4}{45} h^3 L_6[h, D] + \tfrac{1}{105} h^4 D[h]^2\right)\right) C[h] \\
&Q[h] F[h] - Q[-h] F[-h] + R[h] - R[-h] = O\, h^9
\end{aligned}
\tag{29}
$$

Note the commonality of subexpressions between the $Q[h]$ and $R[h]$ formulas in Eq's. (28) and (29). Also, the product factors $D[h]^2$ in Eq's. (27)-(29) can be re-used in the subsequent integration step as $D[-h]^2$ for calculating $Q[-h]$ and $R[-h]$. The products $D[h]C[h]$ in Eq's. (27) and (28) can similarly be carried over to the next step.

3. Scale-and-square algorithm

For the constant-coefficient case, Eq. (10) can be generalized as

$$F[x_0 + \Delta x] = \exp[\Delta x\, D]\, F[x_0] + (\exp[\Delta x\, D] - \mathbf{I})\, D^{-1}\, C\,. \tag{30}$$

This equation is applied with $\Delta x = 2h$ and with the x coordinate origin shifted so that $x_0 = -h$,

$$F[h] = \Phi\, F[-h] + \Gamma\, C\,, \tag{31}$$

where

$$\Phi = \exp[2h\, D]\,, \tag{32}$$

$$\Gamma = (\exp[2h\, D] - \mathbf{I})\, D^{-1}\,. \tag{33}$$

The Φ and Γ matrices can be obtained from the Padé approximation, Eq. (6), for small Δx,

$$F[h] \approx Q[h]^{-1}\, (Q[-h]\, F[-h] - 2\, L[h] C)\,. \tag{34}$$

(The term $L[h] - L[-h]$ in Eq. (6) has been replaced by $2\, L[h]$ because $L[h]$ is an odd function of h for the constant-coefficient case.) The following approximations result from Eq's. (31) and (34),

$$\Phi \approx Q[h]^{-1}\, Q[-h]\,, \tag{35}$$

$$\Gamma \approx -2\, Q[h]^{-1}\, L[h]\,. \tag{36}$$

Eq. (31) generalizes to

$$F[x_0 + \Delta x] = \Phi\, F[x_0] + \Gamma\, C\,, \tag{37}$$

where Φ and Γ are computed from Eq's. (35) and (36) with $h = \Delta x / 2$. Eq. (37) is applied iteratively to integrate $F[x]$ over a large, m-step integration interval,

$$F[x_0 + m\, \Delta x] = \Phi^m\, F[x_0] + \Gamma_m\, C\,, \tag{38}$$

where

$$\Gamma_m = (\mathbf{I} + \Phi + \Phi^2 + \ldots + \Phi^{m-1})\, \Gamma \quad (\Gamma_1 = \Gamma)\,. \tag{39}$$

Given Φ^m and Γ_m for any particular integer m, Φ^{2m} and Γ_{2m} are obtained as

$$\Phi^{2m} = (\Phi^m)^2\,, \tag{40}$$

$$\Gamma_{2m} = \Gamma_m + \Phi^m\, \Gamma_m\,. \tag{41}$$

Eq. (40) is the basis of the standard scale-and-square algorithm for homogeneous linear differential equations, and Eq. (41) generalizes the method for nonhomogeneous equations.

In implementing the Padé approximation it is advantageous to calculate the even and odd parts of $Q[h]$ separately so that $Q[h]$ and $Q[-h]$ can be calculated with minimal computational overhead,

$$Q^{[\text{even}]}[h] = \tfrac{1}{2}(Q[h] + Q[-h]), \quad Q^{[\text{odd}]}[h] = \tfrac{1}{2}(Q[h] - Q[-h]),$$
$$Q[\pm h] = Q^{[\text{even}]}[h] \pm Q^{[\text{odd}]}[h]. \tag{42}$$

Note that the $Q[h]$ definitions in Eq's. (12)-(15) are formatted with the even and odd polynomials separated. Also note that $Q^{[\text{odd}]}[h]$ has a left factor of $L[h]$,

$$Q^{[\text{odd}]}[h] = L[h]\,D\,. \tag{43}$$

Eq. (43) is applied in Eq. (35) to obtain

$$\Phi - \mathbf{I} = -2\,Q[h]^{-1}\,Q^{[\text{odd}]}[h] = -2\,Q[h]^{-1}\,L[h]\,D\,. \tag{44}$$

For small Δx the matrix Γ is approximately proportional to Δx (Eq. (33)), but Φ is approximately equal to $\mathbf{I}$ with a small Δx-proportionate increment (Eq. (32)). To avoid possible precision loss in the Φ diagonal elements, Φ can be calculated with the dominant $\mathbf{I}$ component subtracted off. The $\mathbf{I}$ separation is preserved through the scale-and-square process by modifying Eq's. (40) and (41) as follows,

$$(\Phi^{2m} - \mathbf{I}) = (\Phi^m - \mathbf{I})^2 + 2\,(\Phi^m - \mathbf{I})\,, \tag{45}$$

$$\Gamma_{2m} = 2\,\Gamma_m + (\Phi^m - \mathbf{I})\,\Gamma_m\,. \tag{46}$$

The above calculation procedure can be somewhat simplified by taking advantage of the relation

$$\Phi - \mathbf{I} = \Gamma\,D\,. \tag{47}$$

This is an exact equality based on Eq's. (32) and (33). The relation also holds exactly with Φ and Γ defined by the Padé approximation, based on Eq's. (36) and (44). The same condition holds for Φ^m and Γ_m,

$$\Phi^m - \mathbf{I} = \Gamma_m\,D\,. \tag{48}$$

(This follows from Eq's. (40) and (41), by induction.) Thus, Eq's. (45) and (46) can subsumed by the single equation,

$$\Gamma_{2m} = 2\,\Gamma_m + (\Gamma_m)^2\,D\,. \tag{49}$$

However, Eq. (49) does not provide much efficiency advantage. It requires two matrix multiplies, as do Eq's. (45) and (46). (For homogeneous equations, Eq. (46) is not needed and Eq. (45) only requires one matrix multiply.)

4. Error analysis and tolerance control, constant coefficients

Continuing with the constant-coefficient case, Eq. (38) will be slightly in error due to the inaccuracy of the Padé approximation. Denoting error terms (approximation minus exact value) by the prefix "δ", the approximation error in $F[x_0 + \Delta x]$ is

$$\delta F[x_0 + m\,\Delta x] = (\delta(\Phi^m))\,F[x_0] + (\delta(\Gamma_m))\,C\,. \tag{50}$$

It follows from Eq. (48) that

$$\delta(\Phi^m) = (\delta(\Gamma_m))\, D\,. \tag{51}$$

Thus, Eq. (50) reduces to

$$\delta F[x_0 + m\,\Delta x] = (\delta(\Gamma_m))(D\,F[x_0] + C)\,. \tag{52}$$

In section 3 the function names F, Φ, and Γ represent exact values (solutions to Eq. (1)) in some contexts and Padé approximations in other contexts, but in this section the functions are consistently defined to be approximations. The "δ" notation denotes the approximation error. For example, Eq. (49) is an approximate relation, and the corresponding error-corrected equation is

$$\Gamma_{2m} - \delta(\Gamma_{2m}) = 2(\Gamma_m - \delta(\Gamma_m)) + (\Gamma_m - \delta(\Gamma_m))^2\, D\,. \tag{53}$$

This is subtracted from Eq. (49) to obtain the error compounding formula,

$$\delta(\Gamma_{2m}) = 2(\delta(\Gamma_m))(\mathbf{I} + \Gamma_m\, D) - (\delta(\Gamma_m))^{\,2} D\,. \tag{54}$$

Eq. (52) can be more usefully reformulated with the $F[x_0]$ factor on the right replaced by $F[x_0 + m\,\Delta x]$,

$$\delta F[x_0 + m\,\Delta x] = (\delta(\Gamma_m)^{[\mathrm{rel}]})(D\,F[x_0 + m\,\Delta x] + C)\,. \tag{55}$$

where $\delta(\Gamma_m)^{[\mathrm{rel}]}$ is a "relative error" factor, which can be derived by using Eq. (38) to eliminate $F[x_0]$ in Eq. (52),

$$\delta F[x_0 + m\,\Delta x] = \Phi^{-m}(\delta(\Gamma_m))(D(F[x_0 + m\,\Delta x] - \Gamma_m\, C) + \Phi^m\, C)\,. \tag{56}$$

The right-hand factor of Φ^m is eliminated using Eq. (48), and the equation reduces to

$$\delta F[x_0 + m\,\Delta x] = \Phi^{-m}(\delta(\Gamma_m))(D(F[x_0 + m\,\Delta x]) + C)\,. \tag{57}$$

This equates to Eq. (55) with

$$\delta(\Gamma_m)^{[\mathrm{rel}]} = \Phi^{-m}(\delta(\Gamma_m))\,. \tag{58}$$

Eq. (54) is reformulated in terms of the relative errors defined by Eq. (58),

$$\delta(\Gamma_{2m})^{[\mathrm{rel}]} = 2\,\Phi^{-m}(\delta(\Gamma_m)^{[\mathrm{rel}]})(\mathbf{I} + \Gamma_m\, D) - (\delta(\Gamma_m)^{[\mathrm{rel}]})^{\,2} D\,. \tag{59}$$

With application of Eq. (48), this reduces to

$$\delta(\Gamma_{2m})^{[\mathrm{rel}]} = 2\,\delta(\Gamma_m)^{[\mathrm{rel}]} - (\delta(\Gamma_m)^{[\mathrm{rel}]})^{\,2} D\,. \tag{60}$$

The error will be controlled by limiting its norm. Given an upper bound on $\left\|\delta(\Gamma_1)^{[\mathrm{rel}]}\right\|$, bounds on $\left\|\delta(\Gamma_2)^{[\mathrm{rel}]}\right\|$, $\left\|\delta(\Gamma_4)^{[\mathrm{rel}]}\right\|$, … are determined from Eq. (60),

$$\left\|\delta(\Gamma_{2m})^{[\mathrm{rel}]}\right\| \le 2\left\|\delta(\Gamma_m)^{[\mathrm{rel}]}\right\| + \left\|\delta(\Gamma_m)^{[\mathrm{rel}]}\right\|^2 \|D\|\,. \tag{61}$$

where $\|\ldots\|$ is the Frobenius norm.

To obtain $\delta(\Gamma_1)^{[\mathrm{rel}]}$, we use Eq. (55) with $m=1$, $x_0=-h$, and $\Delta x=2h$,

$$\delta F[h]=(\delta(\Gamma_1))^{[\mathrm{rel}]}(D\,F[h]+C)\,. \tag{62}$$

$F[h]$ is defined by Eq. (6), in which the error term "$O\,h^{2n+1}$"on the right is replaced by zero. The error term can be replaced by an explicit residual formula to obtain the following exact version of Eq. (6),

$$\begin{aligned} &Q[h]\,F^{[\mathrm{exact}]}[h]-Q[-h]\,F[-h]+(L[h]-L[-h])\,C= \\ &\quad \exp[-h\,D]\left(\frac{D^{2n}}{(2n)!}\int_{-h}^{h}\exp[x\,D](x^2-h^2)^n\,dx\right)(D\,F^{[\mathrm{exact}]}[h]+C), \end{aligned} \tag{63}$$

where $F^{[\mathrm{exact}]}$ denotes the exact solution of Eq. (1) premised on the initial condition $F[-h]$,

$$F^{[\mathrm{exact}]}[h]=F[h]-\delta F[h]\,. \tag{64}$$

Eq. (63) is derived from Eq. (113) in Appendix A with application of Eq. (5) and with the following substitution (from Eq. (10)),

$$F^{[\mathrm{exact}]}[0]=\exp[-h\,D]\,F^{[\mathrm{exact}]}[h]+(\exp[-h\,D]-\mathbf{I})\,D^{-1}\,C\,. \tag{65}$$

(The F function in Eq's. (10) and (113) is $F^{[\mathrm{exact}]}$.)

Eq. (63) is subtracted from Eq. (6) (with substitution from Eq. (64)) to obtain

$$\delta F[h]=A(D(F[h]-\delta F[h])+C)\,, \tag{66}$$

where

$$A=-Q[h]^{-1}\exp[-h\,D]\frac{D^{2n}}{(2n)!}\int_{-h}^{h}\exp[x\,D](x^2-h^2)^n\,dx\,. \tag{67}$$

Eq. (66) is solved for $\delta F[h]$,

$$\delta F[h]=(\mathbf{I}+A\,D)^{-1}\,A(D\,F[h]+C)\,. \tag{68}$$

Comparing Eq's. (62) and (68), we obtain

$$(\delta(\Gamma_1))^{[\mathrm{rel}]}=(\mathbf{I}+A\,D)^{-1}\,A\,. \tag{69}$$

To bound $\left\|(\delta(\Gamma_1))^{[\mathrm{rel}]}\right\|$, we first limit h to bound the $Q[h]^{-1}$ factor in Eq. (67). $Q[h]$ is close to $\mathbf{I}$ for sufficiently small h ($Q[h]=\mathbf{I}-h\,D+O\,h^2$), so $Q[h]^{-1}$ can be bounded by applying the following condition: For matrixes X and V with $\|X\|<1$, $\left\|(\mathbf{I}+X)^{-1}\,V\right\|$ is bounded by $(1-\|X\|)^{-1}\,\|V\|$. A Taylor series for $(\mathbf{I}+X)^{-1}$ is used to obtain this result,

$$\begin{aligned} &\|X\|<1\rightarrow \\ &\quad \left\|(\mathbf{I}+X)^{-1}\,V\right\|=\left\|V-X\,V+X^2\,V-\ldots\right\|\le\|V\|+\|X\|\cdot\|V\|+\|X\|^2\cdot\|V\|+\ldots. \\ &\quad =(1-\|X\|)^{-1}\,\|V\|. \end{aligned} \tag{70}$$

Rather than applying Eq. (70) directly to $Q[h]$, a less constraining condition can be formulated by taking advantage of the fact that $Q[-h]$ is a close approximation to $Q[h]^{-1}$ for small h ($Q[h]Q[-h] = \mathbf{I} - h^2 D^2 / (2n-1) + O h^4$). Eq. (70) implies the following bound on $(Q[h]Q[-h])^{-1}$,

$$\begin{aligned} &\left\| Q[h]Q[-h] - \mathbf{I} \right\| < 1 \rightarrow \\ &\quad \left\| (Q[h]Q[-h])^{-1} V \right\| = \left\| (\mathbf{I} + (Q[h]Q[-h] - \mathbf{I}))^{-1} V \right\| \le (1 - \left\| Q[h]Q[-h] - \mathbf{I} \right\|)^{-1} \|V\|. \end{aligned} \tag{71}$$

for any matrix factor V.

For the constant-D case, $Q[h]$ is as a polynomial function of $h D$ (Eq's. (17), (21)),

$$Q[h] = \hat{Q}[h D], \quad \hat{Q}[X] = \sum_{j=0}^{n} q_j X^j . \tag{72}$$

The product $\hat{Q}[X]\hat{Q}[-X]$ has a Taylor series of the form

$$\hat{Q}[X]\hat{Q}[-X] = \sum_{j=0}^{n} a_j X^{2j}, \quad a_j = \frac{(-1)^j \, j!(2n-2j)!}{(2n-j)!} q_j{}^2 . \tag{73}$$

The following bound is obtained from this series,

$$\begin{aligned} &\left\| \hat{Q}[X]\hat{Q}[-X] - \mathbf{I} \right\| = \left\| \sum_{j=1}^{n} a_j X^{2j} \right\| \le \sum_{j=1}^{n} |a_j| \cdot \left\| X^2 \right\|^j = \sum_{j=1}^{n} a_j \cdot (i\sqrt{\left\| X^2 \right\|})^{2j} \\ &= \hat{Q}[i\sqrt{\left\| X^2 \right\|}]\hat{Q}[-i\sqrt{\left\| X^2 \right\|}] - 1. \end{aligned} \tag{74}$$

(The i factor cancels the $(-1)^j$ factor in a_j.)

Eq's. (71) and (74) are combined to obtain the following condition,

$$\begin{aligned} &\hat{Q}[i\sqrt{\left\| h^2 D^2 \right\|}]\hat{Q}[-i\sqrt{\left\| h^2 D^2 \right\|}] < 2 \rightarrow \\ &\quad \left\| (Q[h]Q[-h])^{-1} V \right\| \le \left(2 - \hat{Q}[i\sqrt{\left\| h^2 D^2 \right\|}]\hat{Q}[-i\sqrt{\left\| h^2 D^2 \right\|}] \right)^{-1} \|V\|. \end{aligned} \tag{75}$$

h is constrained to satisfy the premise (first inequality) of Eq. (75). (In practice, the premise $\ldots < 2$ can be replaced by a somewhat tighter limit, e.g. $\ldots < 1.9$, to ensure that the reciprocal factor is not very large.)

The leading factor $Q[h]^{-1} \exp[-h D]$ in Eq. (67) is equal to $(Q[h]Q[-h])^{-1} (Q[-h] \exp[-h D])$. The reciprocal term is bounded by Eq. (75). The factor $Q[-h] \exp[-h D]$ is a difference of even and odd functions,

$$\begin{aligned} &Q[-h] \exp[-h D] = \\ &\quad \tfrac{1}{2}(Q[h] \exp[h D] + Q[-h] \exp[-h D]) - \tfrac{1}{2}(Q[h] \exp[h D] - Q[-h] \exp[-h D]). \end{aligned} \tag{76}$$

The odd function is half the Padé approximation error defined by Eq. (104) in Appendix A. The even function can be bounded by taking advantage of the fact that $Q[\pm h]$ is a close

approximation to $\exp[\mp h\,D]$ for small h ($\exp[\mp h\,D]-Q[\pm h]=h^2\,D^2/(2(2n-1))+O\,h^3$). The following identity is used to take advantage of this approximation,

$$\begin{aligned} &Q[h]\exp[h\,D]+Q[-h]\exp[-h\,D]= \\ &\quad \mathbf{I}+Q[h]\,Q[-h]-(\exp[h\,D]-Q[-h])\,(\exp[-h\,D]-Q[h]). \end{aligned} \tag{77}$$

The two right-hand factors in Eq. (77) are of the form $\exp[X]-\hat{Q}[-X]$. This expression's Taylor series coefficients are all non-negative

$$\exp[X]-\hat{Q}[-X]=\sum_{j=2}^{n}\left(1-\prod_{k=1}^{j-1}\left(1-\frac{k}{2n-k}\right)\right)\frac{X^j}{j!}+\sum_{j=n+1}^{\infty}\frac{X^j}{j!}\,. \tag{78}$$

The $\hat{Q}$ function comprises even and odd parts $\hat{Q}^{[\text{even}]}$ and $\hat{Q}^{[\text{odd}]}$, as in Eq's. (42). The expression $\exp[X]-\hat{Q}[-X]$ correspondingly comprises even and odd parts $\cosh[X]-\hat{Q}^{[\text{even}]}[X]$ and $\sinh[X]+\hat{Q}^{[\text{odd}]}[X]$, both of which have non-negative series coefficients (from Eq. (78)). The right-had product in Eq. (77) is represented in terms of these even/odd expressions,

$$\begin{aligned} &(\exp[h\,D]-Q[-h])\,(\exp[-h\,D]-Q[h])= \\ &\quad (\cosh[h\,D]-Q^{[\text{even}]}[h])^2-(\sinh[h\,D]+Q^{[\text{odd}]}[h])^2\,. \end{aligned} \tag{79}$$

Each of the two squared expressions is an even function of h with non-negative series coefficients; hence Eq. (79) has the following bound,

$$\begin{aligned} &\left\|(\cosh[h\,D]-Q^{[\text{even}]}[h])^2-(\sinh[h\,D]+Q^{[\text{odd}]}[h])^2\right\|\le \\ &\quad (\cosh[\sqrt{\left\|h^2\,D^2\right\|}]-\hat{Q}^{[\text{even}]}[\sqrt{\left\|h^2\,D^2\right\|}])^2+(\sinh[\sqrt{\left\|h^2\,D^2\right\|}]+\hat{Q}^{[\text{odd}]}[\sqrt{\left\|h^2\,D^2\right\|}])^2\,. \end{aligned} \tag{80}$$

This relation is used to bound the first right-hand term in Eq. (76).

The integral in Eq's. (67) and (104) is an even function of D, so the $\exp[h\,D]$ factor can be replaced by $\cosh[h\,D]$. The integral has the following bound,

$$\begin{aligned} &\left\|\int_{-h}^{h}\exp[x\,D](x^2-h^2)^n\,dx\right\|=\left\|\int_{-h}^{h}\cosh[x\,D](x^2-h^2)^n\,dx\right\| \\ &\le\cosh[\sqrt{\left\|h^2\,D^2\right\|}]\int_{-|h|}^{|h|}(h^2-x^2)^n\,dx=\cosh[\sqrt{\left\|h^2\,D^2\right\|}]\frac{n!^2\,|2\,h|^{2n+1}}{(2n+1)!}. \end{aligned} \tag{81}$$

Eq's. (76), (77), and (104) are applied to reformulate Eq. (67), as follows:

$$A=-\tfrac{1}{2}\left(\mathbf{I}+(Q[h]\,Q[-h])^{-1}\left(\mathbf{I}-(\exp[-h\,D]-Q[h])\,(\exp[h\,D]-Q[-h])-D\,B\right)\right)B\,, \tag{82}$$

where

$$B=\frac{D^{2n}}{(2n)!}\int_{-h}^{h}\exp[x\,D](x^2-h^2)^n\,dx\,. \tag{83}$$

Eq's. (75) and (79)-(81) are used to establish a bound on Eq. (82),

$$\hat{Q}[i\sqrt{\|h^2\,D^2\|}]\hat{Q}[-i\sqrt{\|h^2\,D^2\|}] < 2 \rightarrow$$
$$\|A\| \le \tfrac{1}{2}\left(\begin{array}{l} 1+(2-\hat{Q}[i\sqrt{\|h^2\,D^2\|}]\hat{Q}[-i\sqrt{\|h^2\,D^2\|}])^{-1}\cdot \\ \quad\left(\begin{array}{l} 1+(\cosh[\sqrt{\|h^2\,D^2\|}]-\hat{Q}^{[\text{even}]}[\sqrt{\|h^2\,D^2\|}])^2 \\ \quad +(\sinh[\sqrt{\|h^2\,D^2\|}]+\hat{Q}^{[\text{odd}]}[\sqrt{\|h^2\,D^2\|}])^2+\|D\|\cdot\|B\| \end{array}\right) \end{array}\right)\|B\|, \tag{84}$$

where $\|B\|$ is bounded by

$$\|B\| \le \frac{n!^2}{(2n)!(2n+1)!}|2h|^{2n+1}\|D^{2n}\|\cosh[\sqrt{\|h^2\,D^2\|}]. \tag{85}$$

Direct computation of $\|D^{2n}\|$ could potentially result in numeric overflow, but this problem can be avoided by factoring the matrix norm $\|D\|$ out of the D matrix and combining it with h in Eq. (85),

$$|2h|^{2n}\cdot\|D^{2n}\| = |2h\|D\||^{2n}\cdot\|(D/\|D\|)^{2n}\|. \tag{86}$$

The upper bound on $\|A\|$ is used to establish a bound on $(\delta(\Gamma_1))^{[\text{rel}]}$ from Eq. (69). The reciprocal factor in Eq. (69) is bounded using Eq. (70):

$$\|A\|\cdot\|D\| < 1 \rightarrow$$
$$\|(\delta(\Gamma_1))^{[\text{rel}]}\| \le (1-\|A\|\cdot\|D\|)^{-1}\|A\|. \tag{87}$$

(The first inequality can be strengthened somewhat, e.g. $\|A\|\cdot\|D\| < 0.9$, to ensure that the reciprocal factor in Eq. (87) is not excessively large.) Based on this limit, $(\delta(\Gamma_2))^{[\text{rel}]}$, $(\delta(\Gamma_4))^{[\text{rel}]}$, … are bounded by Eq. (61).

The integration step size is chosen to limit the error as follows: $F[x]$ is integrated from $x = x_0$ to $x = x_0 + x_{\text{range}}$, and the full range x_{range} is divided into m steps of size Δx, where m is a power of 2,

$$\Delta x = 2h = x_{\text{range}}/m, \quad m = 2^j. \tag{88}$$

Initially, j is large enough to satisfy the first inequality in Eq. (84). An upper bound on $\|A\|$ is calculated from Eq. (84), and j is further increased, if necessary, to satisfy the first inequality in Eq. (87). A bound on $\|\delta(\Gamma_1)^{[\text{rel}]}\|$ is then determined from Eq. (87), and Eq. (61) is applied to bound $\|\delta(\Gamma_m)^{[\text{rel}]}\|$. A bound on $\|\delta F[x_0 + m\Delta x]\|$ is obtained from Eq. (55),

$$\|\delta F[x_0 + m\Delta x]\| \le \|\delta(\Gamma_m)^{[\text{rel}]}\|(\|D\|\cdot\|F[x_0 + m\Delta x]\| + \|C\|). \tag{89}$$

The factors $\|\delta(\Gamma_m)^{[\text{rel}]}\|\cdot\|D\|$ and $\|\delta(\Gamma_m)^{[\text{rel}]}\|\cdot\|C\|$ are both required to be within a specified tolerance threshold, and j is increased until this condition is satisfied. The condition can be formulated as

$$\left\|\delta(\Gamma_m)^{[\mathrm{rel}]}\right\| \cdot \left\|\begin{bmatrix} D & C \end{bmatrix}\right\| \le tol\,, \tag{90}$$

where tol is the tolerance limit and $\begin{bmatrix} D & C \end{bmatrix}$ is the concatenation of matrices D and C ($\left\|\begin{bmatrix} D & C \end{bmatrix}\right\| = \sqrt{\|D\|^2 + \|C\|^2}$).

5. Error analysis and tolerance control, non-constant coefficients

The above formulas are not directly applicable to the non-constant-coefficient case, but the same process can be used to obtain an initial integration step size Δx using values of $D[x]$ and $C[x]$ at the beginning of the integration interval. This initialization is inapplicable when D is zero or when D or C varies significantly over the Δx range. (When D is identically zero, the errors in Eq's. (26)-(29) are proportional to the order-$2n$ derivative of C for Padé order n.) An alternative step initialization criterion may need to be used to accommodate spatial variability of D and C.

After the integration step size Δx is initialized, it is dynamically varied to limit the integration error, which can be estimated by determining $F[x_0 + \Delta x]$ from $F[x_0]$ by two estimation methods and applying Richardson extrapolation to the estimates. A first estimate $F_1[x_0 + \Delta x]$ is obtained by making a single-step Padé approximation with step size Δx, and a second estimate $F_2[x_0 + \Delta x]$ is obtained by making two Padé approximation steps with step size $\frac{1}{2}\Delta x$. The errors in these estimates are approximately

$$\delta F_1[x_0 + \Delta x] \approx A\,\Delta x^{2n+1}, \quad \delta F_2[x_0 + \Delta x] \approx 2\,A\,(\tfrac{1}{2}\Delta x)^{2n+1}, \tag{91}$$

where n is the Padé order, A is an undetermined matrix, and the factor of 2 is included in the second equality to account for the two steps. The following relation is obtained by eliminating A between Eq's. (91),

$$\delta F_1[x_0 + \Delta x] \approx 2^{2n}\,\delta F_2[x_0 + \Delta x]\,. \tag{92}$$

Subtracting the error from both estimates should give the same error-corrected result,

$$F_1[x_0 + \Delta x] - \delta F_1[x_0 + \Delta x] = F_2[x_0 + \Delta x] - \delta F_2[x_0 + \Delta x]\,. \tag{93}$$

$\delta F_1[x_0 + \Delta x]$ is eliminated from Eq's. (92) and (93) to obtain

$$\delta F_2[x_0 + \Delta x] = \frac{F_1[x_0 + \Delta x] - F_2[x_0 + \Delta x]}{2^{2n} - 1}\,. \tag{94}$$

The integration step Δx is decreased or increased by factors of 2 to keep this estimated error within allowed tolerance bounds (i.e. the step is halved if the error significantly exceeds the tolerance, and is doubled if the error times 2^{2n+1} is within the tolerance). Some excursion of the estimated error over the tolerance limit can be allowed because the calculated $F_2[x_0 + \Delta x]$ can be decremented by the error estimate $\delta F_2[x_0 + \Delta x]$ to improve its accuracy.

The $F[h]$ value calculated from Eq. (4) can be represented as

$$F[h] = \Phi\, F[-h] + \Omega\,, \tag{95}$$

where

$$\Phi = Q[h]^{-1}\, Q[-h]\,, \tag{96}$$

$$\Omega = -Q[h]^{-1}\,(R[h] - R[-h])\,. \tag{97}$$

h can be limited to ensure non-singularity of $Q[h]$, e.g.,

$$\left\|Q[h] - \mathbf{I}\right\| \le \tfrac{1}{2} \rightarrow \left\|Q[h]^{-1}\, V\right\| \le 2\|V\| \tag{98}$$

for any matrix V. (Generally the step size initialization and error control will automatically keep $\left\|Q[h] - \mathbf{I}\right\|$ sufficiently bounded.)

With $x_0 = -h$ and $\Delta x = 2h$, the $F_1[x_0 + \Delta x]$ term in Eq. (94) has the form

$$F_1[x_0 + \Delta x] = \Phi_1\, F[x_0] + \Omega_1\,. \tag{99}$$

The same separation is made for $F_2[x_0 + \Delta x]$ in two steps,

$$\begin{aligned} &F_2[x_0 + \tfrac{1}{2}\Delta x] = \Phi_{2,1}\, F[x_0] + \Omega_{2,1}, \\ &F_2[x_0 + \Delta x] = \Phi_{2,2}\, F[x_0 + \tfrac{1}{2}\Delta x] + \Omega_{2,2} = \Phi_{2,2}\,(\Omega_{2,1} + \Phi_{2,1}\, F[x_0]) + \Omega_{2,2}. \end{aligned} \tag{100}$$

This expression is of the form

$$F_2[x_0 + \Delta x] = \Phi_2\, F[x_0] + \Omega_2 \quad \text{with} \quad \Phi_2 = \Phi_{2,2}\,\Phi_{2,1},\ \Omega_2 = \Omega_{2,2} + \Phi_{2,2}\,\Omega_{2,1}. \tag{101}$$

The estimated error $\delta F_2[x_0 + \Delta x]$ in Eq. (94) is correspondingly separated into Φ and Ω components,

$$\delta F_2[x_0 + \Delta x] = (\delta\Phi_2)\, F[x_0] + \delta\Omega_2 \quad \text{with} \quad \delta\Phi_2 = \frac{\Phi_1 - \Phi_2}{2^{2n} - 1},\ \delta\Omega_2 = \frac{\Omega_1 - \Omega_2}{2^{2n} - 1}. \tag{102}$$

The integration step Δx is chosen to keep the errors $\delta\Phi_2$ and $\delta\Omega_2$ within a limit of the form

$$\left\|\begin{bmatrix}\delta\Phi_2 & \delta\Omega_2\end{bmatrix}\right\| \le tol\,\frac{\Delta x}{x_{\text{range}}}, \tag{103}$$

where tol is a specified tolerance threshold, x_{range} is the full integration range, and the factor $\Delta x / x_{\text{range}}$ apportions the tolerance budget between integration intervals.

References

[1] K. Johnson, *Numerical Solution of Linear, Homogeneous Differential Equation Systems via Padé Approximation* (v2, posted April 22, 2016). http://vixra.org/abs/1509.0286.

[2] K. Johnson, Linear differential equation solver (lde.m), posted Nov. 30, 2016. http://www.mathworks.com/matlabcentral/fileexchange/60475-linear-differential-equation-solver--lde-m-.

[3] N. J. Higham, *The Scaling and Squaring Method for the Matrix Exponential Revisited*, SIAM Review, 51 (2009), pp. 747–764.

[4] A. H. Al-Mohy and N. J. Higham, *A new scaling and squaring algorithm for the matrix exponential*, SIAM J. Matrix Anal. Appl., 30 (2009), pp. 970–989.

[5] MATLAB **`expm`** function, https://www.mathworks.com/help/matlab/ref/expm.html.

[6] MATLAB Ordinary Differential Equation solvers, https://www.mathworks.com/help/matlab/math/choose-an-ode-solver.html.

[7] Gautschi, Walter. *Numerical analysis*. Springer Science & Business Media, 2011, section 5.9.2.

Appendix A: Derivation of the Padé polynomial coefficients (Eq. (21))

Considering the homogeneous, constant-coefficient case (constant D, zero C), the following integral both defines the Q polynomial and determines the error in Eq. (4)

$$Q[h]\exp[h\,D]-Q[-h]\exp[-h\,D]=\frac{D^{2n+1}}{(2n)!}\int_{-h}^{h}\exp[x\,D](x^2-h^2)^n\,dx\,. \tag{104}$$

(This formula is adapted from Eq's. (5.149) and (5.150) in [7].) Eq. (104) is right-multiplied by $F[0]$ to obtain Eq. (4), with the "$O\,h^{2n+1}$" term representing the above integral expression.

The integral in Eq. (104) is integrated by parts $2n+1$ times to obtain

$$\int_{-h}^{h}\exp[x\,D](x^2-h^2)^n\,dx=\sum_{j=0}^{j=2n}(-1)^j\,D^{-j-1}\exp[x\,D]\frac{d^j}{dx^j}(x^2-h^2)^n\Bigg|_{x=-h}^{x=h}\,. \tag{105}$$

The derivative expression is expanded via the general Leibniz rule,

$$\begin{aligned}
&\frac{d^j}{dx^j}(x^2-h^2)^n=\frac{d^j}{dx^j}\left((x+h)^n(x-h)^n\right)\\
&=\sum_{k=0}^{k=j}\frac{j!}{k!(j-k)!}\left(\frac{d^k}{dx^k}(x+h)^n\right)\left(\frac{d^{j-k}}{dx^{j-k}}(x-h)^n\right)\\
&=\sum_{k=0}^{k=j}\frac{j!}{k!(j-k)!}\left(\frac{n!}{(n-k)!}(x+h)^{n-k}\right)\left(\frac{n!}{(n-j+k)!}(x-h)^{n-j+k}\right).
\end{aligned} \tag{106}$$

At the lower integration limit, $x=-h$, the $(x+h)^{n-k}$ factor is zero unless $n=k$, and at the upper limit the factor $(x-h)^{n-j+k}$ is zero unless $n=j-k$. Neither condition holds when $j<n$; hence the summation terms $j=0,\ldots n-1$ vanish in Eq. (105) and the sum reduces to

$$\sum_{j=0}^{j=2n}(-1)^j\,D^{-j-1}\exp[x\,D]\frac{d^j}{dx^j}(x^2-h^2)^n\Bigg|_{x=-h}^{x=h}= \\ \sum_{j=n}^{j=2n}(-1)^j\,D^{-j-1}\frac{n!\,j!}{(j-n)!(2n-j)!}\Big((2h)^{2n-j}\exp[h\,D]-(-2h)^{2n-j}\exp[-h\,D]\Big). \tag{107}$$

The summation index j is replaced by $2n-j$ in the last sum, and the result is substituted back into Eq. (105) and Eq. (104),

$$Q[h]\exp[h\,D]-Q[-h]\exp[-h\,D]= \\ \sum_{j=0}^{j=n}\frac{n!(2n-j)!}{(2n)!(n-j)!\,j!}\Big((-2h\,D)^j\exp[h\,D]-(2h\,D)^j\exp[-h\,D]\Big). \tag{108}$$

The Q polynomial coefficients (Eq. (21)) are readily obtained from this expression.

For the nonhomogeneous case (non-zero C), the above process is applied using the formalism of Eq's. (2) and (3). The following substitutions are made in the above equations,

$$D[x]\to\begin{pmatrix}D[x] & C[x]\\ \mathbf{0} & \mathbf{0}\end{pmatrix},\quad Q[h]\to\begin{pmatrix}Q[h] & R[h]\\ \mathbf{0} & \mathbf{I}\end{pmatrix}. \tag{109}$$

Eq. (104) is again obtained with this formalism, and the following additional condition is also obtained,

$$\exp[h\,D](R[h]+D^{-1}C)-\exp[-h\,D](R[-h]+D^{-1}C)= \\ \left(\frac{D^{2n+1}}{(2n)!}\int_{-h}^{h}\big(\exp[x\,D]\big)(x^2-h^2)^n\,dx\right)D^{-1}C. \tag{110}$$

Eq's. (110) and (104) are consistent if

$$R[h]=(Q[h]-\mathbf{I})\,D^{-1}C. \tag{111}$$

However, only the odd part of $R[h]$ is relevant to Eq. (4) so we can alternatively define

$$R[h]=\tfrac{1}{2}(Q[h]-Q[-h])\,D^{-1}C. \tag{112}$$

(This is equivalent to Eq. (43).) In either case, the right side of Eq. (4) is obtained from Eq's. (10) and (104),

$$\begin{aligned}&Q[h]F[h]-Q[-h]F[-h]+R[h]-R[-h]\\ &=(Q[h]\exp[h\,D]-Q[-h]\exp[-h\,D])(F[0]+D^{-1}C)\\ &=\left(\frac{D^{2n}}{(2n)!}\int_{-h}^{h}\exp[x\,D](x^2-h^2)^n\,dx\right)(D\,F[0]+C).\end{aligned} \tag{113}$$

Appendix B: MATLAB Implementation notes, constant-coefficient case

The even part of the $Q[h]$ polynomial ($Q^{[\text{even}]}[h]$, Eq's. (23) and (42)) is of the form

$$Y = \sum_{j=1}^{N} c_j \, X^{j-1}; \;\; X = (h\,D)^2 \,. \tag{114}$$

(X is a square matrix; the coefficients c_j are scalar.) The odd part ($Q^{[\text{odd}]}[h]$) has a similar form, but with an extra factor of $h\,D$. (Omitting the D factor yields the $L[h]$ matrix in Eq. (43), from which $R[h]$ is obtained in Eq. (5).) The polynomial coefficients c_j can be initialized by using the recursion relations in Eq's. (24).

For large N the polynomial can be efficiently evaluated by zero-padding and reshaping the coefficient vector to a rectangular array and reorganizing Eq. (114) as

$$Y = \sum_{k=1}^{N_2} \left(\sum_{j=1}^{N_1} c_{j,k} \, X^{j-1} \right) (X^{N_1})^{k-1}; \;\; N_1 = \text{round}[\sqrt{2\,N}], \;\; N_2 = \text{ceil}[N / N_1] \;\; . \tag{115}$$

A direct implementation of Eq. (114) would require $N-2$ matrix multiplies; but in the context of the matrix exponential algorithm the sum is performed twice (once for $Q^{[\text{even}]}$ and once for $Q^{[\text{odd}]}$), so the number of multiplies would actually be $2(N-2)$. By contrast, Eq. (115) typically requires $N_1 + 2\,N_2 - 3$ multiplies, including one-time precomputation of $X^2, \ldots \; X^{N_1}$ ($N_1 - 1$ multiplies) and the $N_2 - 1$ multiplies by X^{N_1} in the k sum evaluation (by Horner's method) for each of $Q^{[\text{even}]}$ and $Q^{[\text{odd}]}$. This approach reduces the number of multiplies by approximately a factor of $\sqrt{N/2}$. The number can be reduced by 1 if $N_2 = 1$ (in which case X^{N_1} is not needed). Alternatively, if $N_2 > 1$ the number can be reduced by 1 if $c_{j,N_2} = 0$ for $j > 1$ (in which the first step of Horner's method multiplies X^{N_1} by c_{1,N_2}, a scalar). If this condition holds for both $Q^{[\text{even}]}$ and $Q^{[\text{odd}]}$ then the number is reduced by 2.

For example, an order-12 polynomial can be implemented with 5 matrix multiplies as follows,

$$\begin{aligned} c_1 + c_2\,X + \ldots + c_{13}\,X^{12} = {} & c_1 + c_2\,X + c_3\,X^2 + c_4\,X^3 + \\ & \left(c_5 + c_6\,X + c_7\,X^2 + c_8\,X^3 + \left(c_9 + c_{10}\,X + c_{11}\,X^2 + c_{12}\,X^3 + c_{13}\,X^4 \right) X^4 \right) X^4 \,. \end{aligned} \tag{116}$$

Three matrix-matrix multiplies are required for X^2, X^3, and X^4; and two are required for the $(\ldots)\,X^4$ products. Two order-12 polynomials, with the same X and different c_j coefficients, can be evaluated with 7 matrix multiplies.

Eq. (84) contains the additional matrix power D^{2n}, but calculation of this power can be avoided by using the following bounding estimate for the norm of a matrix power,

$$\left\| D^{j_1 + j_2 + \ldots} \right\| \le \left\| D^{j_1} \right\| \cdot \left\| D^{j_2} \right\| \cdot \ldots \;\; . \tag{117}$$

The pre-computed powers of D used for the Q polynomial evaluation are used on the right side of Eq. (117), and a product of this form is substituted in Eq. (85). The number j of squaring operations (Eq. (11)) might be increased by this substitution, but j is proportional to the logarithm of the integration step (Eq. (88)) so the additional squaring steps are typically not significant compared to the runtime penalty of computing D^{2^n}.

Over-estimation of $\|D^{2^n}\|$ using Eq. (117) can potentially lead to numeric precision loss due to overscaling [4], but the $\mathbf{I}$-separation method (Eq's. (45)-(46)) avoids this problem. The following MATLAB test case illustrates the benefit of $\mathbf{I}$ separation:

```
a = -1e20;
b = eps;
c = 1;
A = [a,0,b;0,c,0;-b,0,a];
% Exact matrix exponential:
expA = exp(a)*( ...
    [1,0,0;0,0,0;0,0,1]*cos(b)+ ...
    [0,0,1;0,0,0;-1,0,0]*sin(b))+ ...
    [0,0,0;0,exp(c),0;0,0,0];
disp(num2str(expA))
0           0           0
0      2.7183           0
0           0           0
```

(118)

The standard MATLAB **expm** function (version R2016b) completely loses numeric precision on this example:

```
disp(num2str(expm(A)))
0  0  0
0  1  0
0  0  0
```

(119)

However, a simple code modification implementing the squaring algorithm as in Eq. (45) reduces the error to machine precision ($<10^{-15}$).

Appendix C: Mathematica verification of Eq's. (12)-(15) and (26)-(29)

The calculations underlying Eq's. (12)-(15) and (26)-(29) require non-commutative symbolic algebra. The following results are obtained using the NCAlgebra package for Mathematica, from the University of California, San Diego (http://math.ucsd.edu/~ncalg/). The Mathematica code loads the NCAlgebra package, adds some functionality, and verifies the equations. A Mathematica notebook containing the following code is posted at https://figshare.com/articles/Appendix_2016_12_19_nb/4479707.

```
(* Mathematica calculations for
 "Numerical Solution of Linear,Nonhomogeneous Differential Equation Systems via Padé Approximation",
http://vixra.org/abs/1611.0002
  Updated 12/19/2016 *)

(* Load NCAlgebra package (http://math.ucsd.edu/~ncalg/) *)
<< NC`
<< NCAlgebra`

(* Make all variables commutative by default.
  (Override the default noncommutativity of single-letter lowercase variables.) *)
Remove[a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z]

(* D0, C0, Dfn, Cfn, F, Q, and R represent matrices. D0 and C0 represent constants;
Dfn, Cfn, F, Q, and R represent functions, and "1" represents the identity matrix. *)
SetNonCommutative[D0, C0, Dfn, Cfn, F, Q, R];

(* Series and O (e.g. O[h]^n) do not work with NC types
 (e.g.: try Dfn[h]**F[h]+O[h]^2 or Series[Dfn[h]**F[h],{h,0,1}]). Define a variant that does work. *)
NCSeries[f_, {x_, x0_, n_}] := SeriesData[x, x0, Table[D[f, {x, j}] / j !, {j, 0, n}] /. x → x0, 0, n + 1, 1];

(* substD is a substitution rule for reducing derivatives of F using the relation F'[h]=Dfn[h]**F[h]+Cfn[h].
    Use "... //. substD" to eliminate all F derivatives.
    (The substD definition uses ":>",
     not "->" otherwise the substitutions will not work when x or n has a preassigned value.) *)
substD = Derivative[n_][F][x_] :> Derivative[n - 1][Dfn[#] ** F[#] + Cfn[#] &][x];

(* substD0 is a substitution rule for reducing derivatives of F using the relation F'[h]==
 D0**F[h]+C0. This specializes substD for the case where Dfn and Cfn are constant. *)
substD0 = Derivative[n_][F][x_] :> Derivative[n - 1][D0 ** F[#] + C0 &][x];


(* Eq 12 *)
Q[h_] := 1 - h D0;
R[h_] := -h C0;
Factor[NCExpand[Normal[NCSeries[Q[h] ** F[h] - Q[-h] ** F[-h] + 2 R[h], {h, 0, 2}]] //. substD0]]

0



(* Eq 13 *)
Q[h_] := (1 + 1/3 h^2 D0 ** D0) - h D0;
R[h_] := -h C0;
Factor[NCExpand[Normal[NCSeries[Q[h] ** F[h] - Q[-h] ** F[-h] + 2 R[h], {h, 0, 4}]] //. substD0]]

0



(* Eq 14 *)
Q[h_] := (1 + 2/5 h^2 D0 ** D0) - (1 + 1/15 h^2 D0 ** D0) ** (h D0);
R[h_] := -(1 + 1/15 h^2 D0 ** D0) ** (h C0);
Factor[NCExpand[Normal[NCSeries[Q[h] ** F[h] - Q[-h] ** F[-h] + 2 R[h], {h, 0, 6}]] //. substD0]]

0



(* Eq 15 *)
Q[h_] := (1 + 3/7 h^2 D0 ** D0 + 1/105 h^4 D0 ** D0 ** D0 ** D0) - (1 + 2/21 h^2 D0 ** D0) ** (h D0);
R[h_] := -(1 + 2/21 h^2 D0 ** D0) ** (h C0);
Factor[NCExpand[Normal[NCSeries[Q[h] ** F[h] - Q[-h] ** F[-h] + 2 R[h], {h, 0, 8}]] //. substD0]]

0
```

```
(* Eq 26 *)
Q[h_] := 1 - h Dfn[0];
R[h_] := -h Cfn[0];
NCExpand[Normal[NCSeries[Q[h] ** F[h] - Q[-h] ** F[-h] + 2 R[h], {h, 0, 2}]] //. substD]
```

```
0
```

```
(* Eq 27 *)
Q[h_] := 1 - h (-1/6 Dfn[-h] + 2/3 Dfn[0] + 1/2 Dfn[h]) + 1/3 h^2 Dfn[h] ** Dfn[h];
R[h_] := -h (-1/6 Cfn[-h] + 2/3 Cfn[0] + 1/2 Cfn[h]) + 1/3 h^2 Dfn[h] ** Cfn[h];
NCExpand[Normal[NCSeries[Q[h] ** F[h] - Q[-h] ** F[-h] + R[h] - R[-h], {h, 0, 4}]] //. substD]
```

```
0
```

```
(* Eq 28 *)
Q[h_] := 1 - h (2/45 Dfn[-h/2] + 2/15 Dfn[0] + 2/3 Dfn[h/2] + 7/45 Dfn[h]) +
    (1/15 Dfn[-h/2] + 1/5 Dfn[0] + 11/15 Dfn[h/2]) **
     (2/5 h^2 (1/9 Dfn[-h/2] - 1/2 Dfn[0] + Dfn[h/2] + 7/18 Dfn[h]) - 1/15 h^3 Dfn[h] ** Dfn[h]);
R[h_] := -h (2/45 Cfn[-h/2] + 2/15 Cfn[0] + 2/3 Cfn[h/2] + 7/45 Cfn[h]) +
    (1/15 Dfn[-h/2] + 1/5 Dfn[0] + 11/15 Dfn[h/2]) **
     (2/5 h^2 (1/9 Cfn[-h/2] - 1/2 Cfn[0] + Cfn[h/2] + 7/18 Cfn[h]) - 1/15 h^3 Dfn[h] ** Cfn[h]);
NCExpand[Normal[NCSeries[Q[h] ** F[h] - Q[-h] ** F[-h] + R[h] - R[-h], {h, 0, 6}]] //. substD]
```

```
0
```

```
(* Eq 29 *)
L1[h_, X_] := 403/16800 X[-h] - 279/2800 X[-2 h/3] + 99/800 X[-h/3] + 34/105 X[0] - 333/5600 X[h/3] + 1719/2800 X[2 h/3] + 1237/16800 X[h];
L2[h_, X_] := 57/1120 X[-h] - 243/560 X[-2 h/3] + 1269/1120 X[-h/3] - 3/4 X[0] + 891/1120 X[h/3] + 27/112 X[2 h/3] - 41/1120 X[h];
L3[h_, X_] := -2067/9680 X[-h] + 6021/4840 X[-2 h/3] - 5805/1936 X[-h/3] + 1863/484 X[0] - 5697/1936 X[h/3] + 10341/4840 X[2 h/3] - 727/9680 X[h];
L4[h_, X_] := 63/16 X[-h] - 1809/40 X[-2 h/3] + 2295/16 X[-h/3] - 801/4 X[0] + 2133/16 X[h/3] - 297/8 X[2 h/3] + 233/80 X[h];
L5[h_, X_] := 123/160 X[-h] - 135/8 X[-2 h/3] + 2295/32 X[-h/3] - 132 X[0] + 3861/32 X[h/3] - 1917/40 X[2 h/3] + 149/32 X[h];
L6[h_, X_] := -6/35 X[-h] + 27/10 X[-2 h/3] - 1053/112 X[-h/3] + 57/4 X[0] - 621/56 X[h/3] + 729/140 X[2 h/3] - 277/560 X[h];
Q[h_] := 1 - h L1[h, Dfn] + L2[h, Dfn] ** (121/315 h^2 L3[h, Dfn] - 2/315 h^3 L4[h, Dfn] ** L5[h, Dfn]) +
    (2/45 h^2 L6[h, Dfn] + L2[h, Dfn] ** (-4/45 h^3 L6[h, Dfn] + 1/105 h^4 Dfn[h] ** Dfn[h])) ** Dfn[h];
R[h_] := -h L1[h, Cfn] + L2[h, Dfn] ** (121/315 h^2 L3[h, Cfn] - 2/315 h^3 L4[h, Dfn] ** L5[h, Cfn]) +
    (2/45 h^2 L6[h, Dfn] + L2[h, Dfn] ** (-4/45 h^3 L6[h, Dfn] + 1/105 h^4 Dfn[h] ** Dfn[h])) ** Cfn[h];
NCExpand[Normal[NCSeries[Q[h] ** F[h] - Q[-h] ** F[-h] + R[h] - R[-h], {h, 0, 6}]] //. substD]

0
```