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

Kenneth C. Johnson

KJ Innovation

April 1, 2015

Abstract

This paper reports work-in-progress on the solution of first-order, linear, homogeneous differential equation systems, with non-constant coefficients, by generalization of the Padé-approximant method for exponential matrices.

1. Introduction

A system of first-order, linear, homogeneous differential equations is of the form

$$F'[x] = D[x]\,F[x]\,, \tag{1}$$

where F and D are matrix functions of a scalar argument x, $D[x]$ is a known coefficient matrix, and $F[x]$ is to be determined from a specified initial value (e.g. $F[0]$). (Following the Mathematica convention, square braces "$[\ldots]$" are used in this paper to delimit function arguments, while round braces "$(\ldots)$" are reserved for grouping.) Typically, methods such as Runge-Kutta [1] are used to calculate numerical solutions of Eq. (1). But in the constant-coefficient case (x-independent D) solutions have an exponential-matrix representation, e.g.,

$$F'[x] = D\,F[x] \quad \rightarrow \quad F[x] = \exp[D\,x]\,F[0]\,. \tag{2}$$

The exponential matrix $\exp[D\,x]$ can be calculated using a Padé approximation for small x (using a "scale-and-square" method to build up $\exp[D\,x]$ for large x) [2].

The Padé-approximant method can also be extended for the case of non-constant coefficients. This paper briefly outlines work-in-progress on the method, which may be generalized and elaborated upon in future work. Section 2 introduces Padé approximation in the context of Eq. (1); section 3 summarizes standard exponential matrix approximation methods for the constant-coefficient case; and section 4 presents several Padé-approximant formulas for the case of non-constant coefficients. The Appendix provides Mathematica code validating the results of section 4.

2. Application of the Padé-approximant method to Eq. (1)

Eq. (1) is solved by a multi-step method in which an approximation of $F[x+\Delta x]$ is determined from a previously computed estimate of $F[x]$, for some small increment Δx. It will be convenient to denote the integration step Δx as $2\,h$, and to locate the x origin at the center of the integration interval. Thus, the problem is to find an approximation to $F[h]$ given a predetermined estimate of $F[-h]$. The approximation is represented as

$$F[h] \approx Q[h]^{-1} P[h] F[-h], \tag{3}$$

where $P[h]$ and $Q[h]$ are matrix-valued, polynomial functions of h determined to minimize the error in Eq. (3) under the premise of Eq. (1). Specifically, we require that

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

where $2n$ is the approximation order. (The order is limited to being even, as explained below.)

Making the substitution $h \to -h$ in Eq. (4), we obtain the similar expression

$$P[-h] F[h] - Q[-h] F[-h] = O\, h^{2n+1}, \tag{5}$$

Assuming that P and Q are uniquely determined by some type of definition criteria, it can be inferred from the similarity of Eq's. (4) and (5) that

$$P[h] = Q[-h], \tag{6}$$

Thus, we seek to determine a polynomial function $Q[h]$ such that

$$Q[h] F[h] - Q[-h] F[-h] = O\, h^{2n+1}, \tag{7}$$

$Q[0]$ is set equal to the identity matrix $\mathbf{I}$,

$$Q[0] = \mathbf{I}. \tag{8}$$

Eq. (7) is an odd function of h, so a Taylor series expansion of the expression will contain only odd powers of h and the error order on the right side of Eq. (7) is also an odd power of h. The approximation order (i.e., the error order minus one) is even.

Due to the odd symmetry of Eq. (7), an order-n polynomial $Q[h]$ has sufficient degrees of freedom to achieve order-$2n$ accuracy of Eq. (7). This is a key benefit of the Padé approximation, which remains true for a non-constant coefficient matrix $D[h]$, although the advantage is diminished in this case because the calculation of $Q[-h]$ also entails evaluation of an order-n polynomial. (For the constant-D case, the calculation of $Q[-h]$ adds very little computational overhead because the even and odd parts of the polynomial $Q[h]$ can be computed separately and subtracted to get $Q[-h]$.) Nevertheless, Padé approximants such as those outlined in section 4 can have advantages of computational efficiency and accuracy relative to standard techniques such as Runge-Kutta.

3. The constant-coefficient case; exponential matrices.

For the constant-coefficient case, Eq's. (2) and (7) imply that

$$Q[h] \exp[D\, h] - Q[-h] \exp[-D\, h] = O\, h^{2n+1}, \tag{9}$$

The function Q, denoted as Q_n for a particular approximation order $2n$, is of the form

$$Q_n[h] = \sum_{j=0}^{n} \frac{(2n-j)!n!}{j!(2n)!(n-j)!}(-2hD)^j \,, \tag{10}$$

The polynomials can be calculated from the following recursion relations,

$$\begin{aligned} Q_0[h] &= \mathbf{I}, \\ Q_1[h] &= \mathbf{I} - hD, \\ Q_{n+1}[h] &= Q_n[h] + \frac{h^2 D^2}{(2n+1)(2n-1)} Q_{n-1}[h]. \end{aligned} \tag{11}$$

The first several iterations of this recursion yield

$$Q_2[h] = \mathbf{I} - hD + \tfrac{1}{3} h^2 D^2 \,, \tag{12}$$

$$Q_3[h] = \mathbf{I} - hD + \tfrac{2}{5} h^2 D^2 - \tfrac{1}{15} h^3 D^3 \,, \tag{13}$$

$$Q_4[h] = \mathbf{I} - hD + \tfrac{3}{7} h^2 D^2 - \tfrac{2}{21} h^3 D^3 + \tfrac{1}{105} h^4 D^4 \,. \tag{14}$$

The accuracy advantage of the Padé approximant method is illustrated by comparing the accuracy error of Eq. (9) to Runge-Kutta methods. For $n = 2$, the error is approximately $\tfrac{2}{45} h^5 D^5$, which is six times smaller than the error of the classic 4th-order Runge-Kutta method. For $n = 3$, the approximate error is $-\tfrac{2}{1575} h^7 D^7$, which is smaller than the error of the 6th-order Runge-Kutta method described in [1] (top of page 192) by a factor of $3/200$.

4. The non-constant-coefficient case: some illustrative formulas

For non-constant $D[x]$ the first several expressions for $Q_n[h]$ can be generalized by replacing the D factors with linear combinations of $D[x]$ evaluated at different x's,

$$Q_1[h] = \mathbf{I} - hD[0], \tag{15}$$

$$Q_2[h] = \mathbf{I} - h(-\tfrac{1}{6} D[-h] + \tfrac{2}{3} D[0] + \tfrac{1}{2} D[h]) + \tfrac{1}{3} h^2 D[h]^2 \,, \tag{16}$$

$$\begin{aligned} Q_3[h] = {} & \mathbf{I} - h(\tfrac{2}{45} D[-\tfrac{1}{2}h] + \tfrac{2}{15} D[0] + \tfrac{2}{3} D[\tfrac{1}{2}h] + \tfrac{7}{45} D[h]) + \\ & (\tfrac{1}{15} D[-\tfrac{1}{2}h] + \tfrac{1}{5} D[0] + \tfrac{11}{15} D[\tfrac{1}{2}h]) \\ & (\tfrac{2}{5} h^2 (\tfrac{1}{9} D[-\tfrac{1}{2}h] - \tfrac{1}{2} D[0] + D[\tfrac{1}{2}h] + \tfrac{7}{18} D[h]) - \tfrac{1}{15} h^3 D[h]^2). \end{aligned} \tag{17}$$

Eq. (17) illustrates the efficiency characteristics of the Padé approximant method. The calculation of $Q_3[h]^{-1} Q_3[-h]$ (i.e., the $Q[h]^{-1} P[h]$ factor in Eq. (3)) requires four matrix multiplies and one matrix divide, but it actually only needs three multiplies per integration step because the $D[h]^2$ term can be reused for the succeeding step (as $D[-h]^2$). The method requires four $D[x]$ function evaluations per integration step (not counting $D[h]$, which is the starting point for the succeeding step). The Padé approximation samples the function at uniform intervals, which is advantageous because interleaved data points can be added to reduce h by a factor of 2 (e.g. for using Richardson extrapolation). If the sampling does not need to be uniform, then an alternative Padé approximant using only three $D[x]$ samples per step can be used,

$$\begin{aligned} Q_3[h] = \mathbf{I} - h\Big(&(\tfrac{5}{12} - \tfrac{3\sqrt{5}}{20})\, D[-\tfrac{1}{\sqrt{5}} h] + (\tfrac{5}{12} + \tfrac{3\sqrt{5}}{20})\, D[\tfrac{1}{\sqrt{5}} h] + \tfrac{1}{6} D[h]\Big) + \\ &\Big((\tfrac{1}{2} - \tfrac{\sqrt{5}}{6})\, D[-\tfrac{1}{\sqrt{5}} h] + (\tfrac{1}{2} + \tfrac{\sqrt{5}}{6})\, D[\tfrac{1}{\sqrt{5}} h]\Big) \\ &\Big(\tfrac{2}{5} h^2 (\tfrac{1}{12} D[-h] - \tfrac{5}{24}(\sqrt{5} - 1)\, D[-\tfrac{1}{\sqrt{5}} h] + \tfrac{5}{24}(\sqrt{5} + 1)\, D[\tfrac{1}{\sqrt{5}} h] + \tfrac{1}{2} D[h]) - \tfrac{1}{15} h^3\, D[h]^2 \Big). \end{aligned} \tag{18}$$

For approximation orders greater than 6, the constant-D formulas for $Q_n[h]$ cannot be generalized by simply replacing D factors with linear combinations of $D[x]$ terms evaluated at different x's. For $n > 3$ $Q_n[h]$ can be generalized as a multivariate order-n polynomial function of multiple $D[x]$ terms, but finding an optimally efficient polynomial with minimal function evaluations and matrix multiplies remains a challenge for future work.

References

[1] Butcher, John C. "On Runge-Kutta processes of high order." *Journal of the Australian Mathematical Society* 4.02 (1964): 179-194.

[2] Higham, Nicholas J. "The scaling and squaring method for the matrix exponential revisited." *SIAM review* 51.4 (2009): 747-764.

Appendix: Approximation orders of Eq's. (15)-(18)

The calculations underlying Eq's. (15)-(18) 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 additional functionality, and verifies Eq. (9) with $Q[x]$ defined by any of Eq's. (15)-(18).

In[1]:=
```
(* Load NCAlgebra package (http://math.ucsd.edu/~ncalg/) *)
<< NC`
<< NCAlgebra`
```

In[4]:=
```
(* 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]
```

In[5]:=
```
(* Dfn, F, and Q represent matrices. ("1" represents the identity matrix.) *)
SetNonCommutative[Dfn, F, Q];
```

In[6]:=
```
(* 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. *)
NCSeries[f_, {x_, x0_, n_}] := NCExpand[Sum[(D[f, {x, j}] / j ! /. x → x0) (x - x0) ^ j, {j, 0, n}]] + O[x - x0] ^ (n + 1);
```

In[7]:=
```
(* substD is a substitution rule for reducing derivatives of F using the relation F'[h]==Dfn[h]**F[h].
   Use "//. substD" to eliminate all F derivatives.
   (Use ":>" here, 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[#] &][x];
```

In[8]:=
```
(* Eq 15 *)
Q[h_] := 1 - h Dfn[0];
NCExpand[Normal[NCSeries[Q[h] ** F[h] - Q[-h] ** F[-h], {h, 0, 2}]] //. substD]
```

Out[9]= 0

In[10]:=
```
(* Eq 16 *)
```

$$\texttt{Q[h\_] := 1 - h}\left(-\frac{1}{6}\,\texttt{Dfn[-h]} + \frac{2}{3}\,\texttt{Dfn[0]} + \frac{1}{2}\,\texttt{Dfn[h]}\right) + \frac{1}{3}\,h^{2}\,\texttt{Dfn[h] ** Dfn[h];}$$

```
NCExpand[Normal[NCSeries[Q[h] ** F[h] - Q[-h] ** F[-h], {h, 0, 4}]] //. substD]
```

Out[11]= 0

In[12]:=
```
(* Eq 17 *)
```

$$\texttt{Q[h\_] := 1 - h}\left(\frac{2}{45}\,\texttt{Dfn}\left[-\frac{h}{2}\right] + \frac{2}{15}\,\texttt{Dfn[0]} + \frac{2}{3}\,\texttt{Dfn}\left[\frac{h}{2}\right] + \frac{7}{45}\,\texttt{Dfn[h]}\right) +$$
$$\left(\frac{1}{15}\,\texttt{Dfn}\left[-\frac{h}{2}\right] + \frac{1}{5}\,\texttt{Dfn[0]} + \frac{11}{15}\,\texttt{Dfn}\left[\frac{h}{2}\right]\right) \texttt{**} \left(\frac{2}{5}\,h^{2}\left(\frac{1}{9}\,\texttt{Dfn}\left[-\frac{h}{2}\right] - \frac{1}{2}\,\texttt{Dfn[0]} + \texttt{Dfn}\left[\frac{h}{2}\right] + \frac{7}{18}\,\texttt{Dfn[h]}\right) - \frac{1}{15}\,h^{3}\,\texttt{Dfn[h] ** Dfn[h]}\right);$$

```
NCExpand[Normal[NCSeries[Q[h] ** F[h] - Q[-h] ** F[-h], {h, 0, 6}]] //. substD]
```

Out[13]= 0

In[14]:=
```
(* Eq 18 *)
Q[h_] :=
```

$$1 - h\left(\left(\frac{5}{12} - \frac{3\sqrt{5}}{20}\right)\texttt{Dfn}\left[-\frac{h}{\sqrt{5}}\right] + \left(\frac{5}{12} + \frac{3\sqrt{5}}{20}\right)\texttt{Dfn}\left[\frac{h}{\sqrt{5}}\right] + \frac{1}{6}\,\texttt{Dfn[h]}\right) + \left(\left(\frac{1}{2} - \frac{\sqrt{5}}{6}\right)\texttt{Dfn}\left[-\frac{h}{\sqrt{5}}\right] + \left(\frac{1}{2} + \frac{\sqrt{5}}{6}\right)\texttt{Dfn}\left[\frac{h}{\sqrt{5}}\right]\right) \texttt{**}$$
$$\left(\frac{2}{5}\,h^{2}\left(\frac{1}{12}\,\texttt{Dfn[-h]} - \frac{5}{24}\left(\sqrt{5} - 1\right)\texttt{Dfn}\left[-\frac{h}{\sqrt{5}}\right] + \frac{5}{24}\left(\sqrt{5} + 1\right)\texttt{Dfn}\left[\frac{h}{\sqrt{5}}\right] + \frac{1}{2}\,\texttt{Dfn[h]}\right) - \frac{1}{15}\,h^{3}\,\texttt{Dfn[h] ** Dfn[h]}\right);$$

```
NCExpand[Normal[NCSeries[Q[h] ** F[h] - Q[-h] ** F[-h], {h, 0, 6}]] //. substD]
```

Out[15]= 0