EVALUATION OF ALGORITHMS FOR THE COMPUTATION OF QUADRATIC FUNCTION APPROXIMATIONS

by

E. Balakrishnan and **A.W. McInnes**

Department of Mathematics, University of Canterbury, Christchurch, New Zealand.

No. 65 November, 1991.

Abstract

The performance of available methods for computing the polynomial coefficients of the quadratic function approximation is evaluated. By comparing the numerical results to those obtained by symbolic methods, for a variety of functions, the direct solution of the matrix equation and a variety of recursive algorithms are all shown to be numerically unstable.

AMS classification: 41A30, 65D15

Key words : Quadratic function approximation, computational algorithms, recursive algorithms.

1. Introduction

This paper considers the performance of available methods for computing the polynomial coefficients of the quadratic function approximation.

The simplest approach is to reduce the coefficient matrix F (details given in section 2) to its echelon form. However, since the condition numbers (in l_2 norm) of these matrices are usually large, this direct approach does not produce a method which is numerically stable, as one would expect.

An alternative approach is to resort to the recursive methods derived from the two recurrence relations developed in an analogous fashion to the rational case, by Paszkowski [6], and a similar but simpler recursive method developed by Brookes and McInnes [3] for the diagonal forms. It is shown in [3] that the latter method is also numerically unstable.

The objective of this paper is to evaluate the numerical effectiveness of these four methods by comparing the results for a variety of different functions. It should be noted that the specialized recursive method established by Borwein [4] is not considered because of the fact that, unlike the above three recursive methods, it is not clear how the technique can be generalized to other functions.

Since it is empirically assumed that the diagonal forms (n, n, n) give a better approximation, two recursive algorithms to generate these diagonal forms are developed from [6]. These, along with the other algorithm are given in section 3. These methods are illustrated by examples in section 4. Solutions generated by the four methods are given in tabulated form in section 5.

2. The Quadratic Approximation Problem

Let $f(x)$ be a function, analytic in some neighbourhood of origin, whose power series expansion about the origin is known. Let $n_i \in \mathbb{Z}^+$ and $a_i(x)$ with $\deg(a_i(x)) \leq n_i$, $i = 0(1)2$,

be polynomials in x such that

$$P(f,x) \equiv \sum_{i=0}^{2} a_i(x) f(x)^i = O\left(x^N\right), \tag{1}$$

where

$$N = \sum_{i=0}^{2} n_i + 2.$$

The function $P(f,x)$ is called the algebraic form of the quadratic function approximation. (This is a particular case ($p = 2$) of the general algebraic form established in [5].)

The existence of such coefficient polynomials $a_i(x)$ follows since these polynomials contain $(N+1)$ coefficients while (1) leads to N homogeneous equations for these coefficients. The matrix form of the system of linear equations represented by (1) has the coefficient matrix

$$F = [F_{n_0} \ : \ F_{n_1} \ : \ F_{n_2}],$$

where

$$F_{n_0} = \begin{bmatrix} I_{n_0+1} \\ 0 \end{bmatrix},$$

and

$$F_{n_j} = \begin{bmatrix} g_0 & & & \\ g_1 & g_0 & & \\ \cdot & \cdot & & \\ g_{n_j} & g_{n_j-1} & \cdots & g_0 \\ \cdot & \cdot & & \cdot \\ g_{N-1} & g_{N-2} & \cdots & g_{N-1-n_j} \end{bmatrix}$$

for $j = 1, 2$, where the g_k in F_{n_j} are defined by $f(x)^j = \sum_{k=0}^{\infty} g_k x^k$.

Given the coefficient polynomials $a_i(x)$, $i = 0, 1, 2$, we now set $\sum_{i=0}^{2} a_i(x) y(x)^i = 0$ and attempt to solve this equation for $y(x)$ in such a way that $y(x)$ approximates $f(x)$. A detailed study on existence, uniqueness and behaviour of these approximations is given in [2], [5].

3. Recursive Methods

The two recursive algorithms which are developed from the recurrence relations [6], and the algorithm in [3] are given in this section. These algorithms are illustrated by means of examples in section 4. In particular, detailed examples of the eliminations are given in this section of examples.

It is assumed that the power series coefficients of $f(x)$ and $f(x)^2$ are known together with some initial forms. Each algorithm requires different initial forms as mentioned at the beginning of each algorithm.

Notation :

(i) It will be assumed throughout that the quadratic forms are of the type (n_0, n_1, n_2).

(ii) $f_i = f(x)^i$, $i = 0, 1, 2$, and $f_{i,j}$ denotes the power series coefficient of x^j in f_i.

(iii) $C^i(n_0, n_1, n_2)$ denotes the coefficient of x^i in the expansion of the (n_0, n_1, n_2) quadratic form after replacing f_i by its power series.

(iv) $E[(n_0, n_1, n_2), z]$ denotes the coefficient of z in the (n_0, n_1, n_2) quadratic form.

It is noted that these algorithms fail for functions which have the following properties.

Property (A) :

The power series coefficient of x^{2k+1} in f_1 for $k = \{0, 1, ...\}$ are all zero; i.e., the function is an even function.

Property (B) :

Any one of the forms $(i, i, i), (i+1, i, i), (i+1, i+1, i)$ have a positive surplus $S > 0$ [5], for any $i \in \mathbf{Z}^+$.

(The property of having surplus $S = 0$ for all $n_i \in \mathbf{Z}^+$ is called Normal by Paszkowski [6]. The property of normality is necessary for Paszkowski's recurrence relations.)

Discussion :

The basic idea behind these methods is that at each step a higher order form is obtained by using a suitable combination of lower order forms to eliminate the terms involving the appropriate powers of x. For example, the $(2,1,1)$ form which is required to be $O(x^6)$, can be obtained either by eliminating the x^5 term in the combination of the $(2,1,0)$ and the $(1,1,1)$ forms which are both $O(x^5)$, or by eliminating the x^4 and x^5 terms in the combination of the $x(1,0,0),(1,1,0)$ and the $(1,1,1)$ forms which are $O(x^4), O(x^4)$, and $O(x^5)$ respectively.

If property (A) is satisfied then $C^N(n_0,n_1,n_2) = 0$ for some (n_0,n_1,n_2).

If property (B) is satisfied then at least one of

$$C^N(i,i,i),\ C^N(i+1,i,i),\ C^N(i+1,i+1,i)$$

is equal to zero.

With these properties the coefficient $C^N(n_0,n_1,n_2)$ *may* be zero and make the elimination impossible. In such cases the algorithms will fail.

Remark :

To some extent property (B) can be ignored since property (A) includes the functions which are known to have positive surplus (see [5] for more details). However, the $(2,2,2)$ form for $(\log(1+x))^2$ has a positive surplus although the function is not even, and hence demonstrates that the property (B) cannot be totally ignored for the implementation of these algorithms.

3.1 Recursion 1 (R1)

In this case f_1 and f_2 are assumed known, along with the forms $(0,0,0),(1,0,0)$ and $(0,1,0)$. For more details see [3].

Algorithm

Given $(0,0,0)$, $(1,0,0)$, $(0,1,0)$ forms, the following method calculates the (n,n,n) forms.

Initialize : Set

$$(1,1,0) = (0,1,0) - \frac{C^3(0,1,0)}{C^3(1,0,0)}(1,0,0).$$

Set n = 0

Step 1: Let

$$v = (n,n,n), \quad r = (0,0,1), \quad s = (0,1,0), \quad t = (1,0,0),$$

$$z_1 = yx^n, \quad z_2 = x^n.$$

1.1: Set

$$(v+r) = x(v) - \frac{E[(v),z_1]}{E[(v+s),xz_1]}(v+s) - \frac{E[(v),z_2]}{E[(v+t),xz_2]}(v+t).$$

1.2: Set

$$(v+r+t) = (v+r) - \frac{C^{3n+3}(v+r)}{C^{3n+3}(v+t)}(v+t).$$

1.3: Set

$$(v+r+s+t) = (v+r+t) - \frac{C^{3n+4}(v+r+t)}{C^{3n+4}(v+s+t)}(v+s+t).$$

Step 2: Let

$$v = (n+1,n,n), \quad r = (1,0,0), \quad s = (0,0,1), \quad t = (0,1,0),$$

$$z_1 = y^2x^n, \quad z_2 = yx^n.$$

2.1: Set

$$(v+r) = x(v) - \frac{E[(v),z_1]}{E[(v+s),xz_1]}(v+s) - \frac{E[(v),z_2]}{E[(v+t),xz_2]}(v+t).$$

2.2: Set

$$(v+r+t) = (v+r) - \frac{C^{3n+4}(v+r)}{C^{3n+4}(v+t)}(v+t).$$

2.3: Set

$$(v+r+s+t) = (v+r+t) - \frac{C^{3n+5}(v+r+t)}{C^{3n+5}(v+s+t)}(v+s+t).$$

Step 3: Let

$$v = (n+1, n+1, n), \quad r = (0,1,0), \quad s = (1,0,0), \quad t = (0,0,1),$$

$$z_1 = x^{n+1}, \quad z_2 = y^2 x^n.$$

3.1: Set

$$(v+r) = x(v) - \frac{E[(v), z_1]}{E[(v+s), xz_1]}(v+s) - \frac{E[(v), z_2]}{E[(v+t), xz_2]}(v+t).$$

3.2: Set

$$(v+r+t) = (v+r) - \frac{C^{3n+5}(v+r)}{C^{3n+5}(v+t)}(v+t).$$

3.3: Set

$$(v+r+s+t) = (v+r+t) - \frac{C^{3n+6}(v+r+t)}{C^{3n+6}(v+s+t)}(v+s+t).$$

Set n = n+1

Go to step 1.

An example of the implementation of this algorithm is given in [3] and a further example of the implementation is given in section 4.1.

3.2 Recursion 2 (R2)

This algorithm is derived from one set of recurrence relations developed by Paszkowski [6, Theorem 3.2].

The forms $(0,0,0), (1,0,0), (1,1,0)$ and f_1, f_2 are assumed known. Since the three steps of the algorithm calculate, in turn, the $(n,n,n), (n+1,n,n), (n+1,n+1,n)$ forms, it is notationally convenient to denote these by $q_{3n+1}, q_{3n+2}, q_{3n+3}$ respectively.

Algorithm

Given $(0,0,0)$, $(1,0,0)$, $(1,1,0)$ forms. The following method calculates the (n,n,n) forms.

Set n = 0

Step 1: Calculating the $(n+1, n+1, n+1)$ form.

Denote $q_{3n+4} = (n+1, n+1, n+1)$.

Set

$$q_{3n+4} = x(n,n,n) + \alpha_{6n+1}\,(n+1,n,n) + \alpha_{6n+2}\,(n+1,n+1,n),$$

where

$$\alpha_{6n+1} = \frac{-C^{3n+3}(x(n,n,n))}{C^{3n+3}(n+1,n,n)},$$

$$\alpha_{6n+2} = \frac{-\left[C^{3n+4}(x(n,n,n)) + \alpha_{6n+1}.C^{3n+4}(n+1,n,n)\right]}{C^{3n+4}(n+1,n+1,n)}.$$

Step 2: Calculating the $(n+2, n+1, n+1)$ form.

Denote $q_{3n+5} = (n+2, n+1, n+1)$.

Set

$$q_{3n+5} = x(n+1,n,n) + \alpha_{6n+3}\,(n+1,n+1,n) + \alpha_{6n+4}\,(n+1,n+1,n+1),$$

where

$$\alpha_{6n+3} = \frac{-C^{3n+4}(x(n+1,n,n))}{C^{3n+4}(n+1,n+1,n)},$$

$$\alpha_{6n+4} = \frac{-\left[C^{3n+5}(x(n+1,n,n)) + \alpha_{6n+3}.C^{3n+5}(n+1,n+1,n)\right]}{C^{3n+5}(n+1,n+1,n+1)}.$$

Step 3: Calculating the $(n+2, n+2, n+1)$ form.

Denote $q_{3n+6} = (n+2, n+2, n+1)$.

Set

$$q_{3n+6} = x(n+1,n+1,n) + \alpha_{6n+5}(n+1,n+1,n+1) + \alpha_{6n+6}(n+2,n+1,n+1),$$

where

$$\alpha_{6n+5} = \frac{-C^{3n+5}(x(n+1,n+1,n))}{C^{3n+5}(n+1,n+1,n+1)},$$

$$\alpha_{6n+6} = \frac{-\left[C^{3n+6}(x(n+1,n+1,n)) + \alpha_{6n+5}.C^{3n+6}(n+1,n+1,n+1)\right]}{C^{3n+6}(n+2,n+1,n+1)}.$$

Set n = n+1

Go to step 1.

Essentially, at each step this algorithm solves a lower triangular system for 2 variables. This linear system arises from the requirement that the next two powers of x must be eliminated in the expansion of the quadratic forms. See [6] for more details on the recurrence relations which form the basis of this algorithm.

3.3 Recursion 3 (R3)

This algorithm is derived from another set of recurrence relations developed by Paszkowski [6, Theorem 3.1].

The forms $(n,n,n),(n+1,n,n),(n+1,n+1,n)$ are denoted by q_i in the above order where n is a non-negative integer and $i = 1(1)3n+3$. Assume that the last $(2n+4)$ forms along with f_1 and f_2 are known. The notation $q_{i,j}$ denotes the jth polynomial coefficient $(j = 0,1,2)$ of the form q_i, and similarly, P_j denotes the jth polynomial coefficient of the form currently being calculated.

Unlike R1 and R2, this method calculates the coefficients of the coefficient polynomials $a_i(x)$ of (1) individually and the coefficients of the form are obtained by combining the P_j of that form.

Algorithm

Given the $(2n+4)$ initial forms, this method calculates the $a_i(x)$ of (1) of the (n,n,n) form.

Set n (arbitrarily)

Step 1 : Using the known last $(2n+4)$ forms, calculate the polynomials P_j in $(n+2, n+1, n+1)$ form.

Set

$$P_j = \sum_{i=n+1}^{3n+4} \gamma_i q_{i,j} + \begin{cases} x^{n+2} & j=0, \\ 0 & j \neq 0, \end{cases} \quad j = 0(1)2\,,$$

where

$$f_{0,k-n} + \sum_{i=n}^{k} \gamma_{i+1} \cdot \left(C^{k+2}(q_{i+1}) \right) = 0, \quad k = n(1)3n+3.$$

Step 2 : Using the above $(2n+5)$ forms, calculate the polynomials P_j in $(n+2, n+2, n+1)$ form.

Set

$$P_j = \sum_{i=n+1}^{3n+5} \gamma_i q_{i,j} + \begin{cases} x^{n+2} & j=1, \\ 0 & j \neq 1, \end{cases} \quad j = 0(1)2,$$

where

$$f_{1,k-n} + \sum_{i=n}^{k} \gamma_{i+1} \cdot \left(C^{k+2}(q_{i+1}) \right) = 0, \quad k = n(1)3n+4.$$

Step 3 : Using the above $(2n+6)$ forms, calculate the polynomials P_j in $(n+2, n+2, n+2)$ form.

Set

$$P_j = \sum_{i=n+1}^{3n+6} \gamma_i q_{i,j} + \begin{cases} x^{n+2} & j=2, \\ 0 & j \neq 2, \end{cases} \quad j = 0(1)2,$$

where

$$f_{2,k-n} + \sum_{i=n}^{k} \gamma_{i+1} \cdot \left(C^{k+2}(q_{i+1}) \right) = 0, \quad k = n(1)3n+5.$$

Set n = n+1

Go to step 1.

At each step this algorithm solves a lower triangular system for $(2n+3+SN)$ variables where SN stands for the step number.

In step 1, the first $(2n+4)$ coefficients of f_0 are used to eliminate the coefficients of x^{n+2} to x^{3n+5}, in the combination of the initial $(2n+4)$ forms. This elimination process requires a linear system to be solved for $(2n+4)$ variables. With this newly obtained form, and with the help of the first $(2n+5)$ coefficients of f_1, a new linear system is formed in step 2 to generate a new quadratic form by eliminating the coefficients of x^{n+2} to x^{3n+6}. Finally, the diagonal form is obtained by eliminating the coefficients of x^{n+2} to x^{3n+7} in the combination of the known last $(2n+6)$ forms. The linear system of this step requires the first $(2n+6)$ coefficients of f_2. Further details on the recurrence relations of the polynomial coefficients may be found in [6].

4. Examples

Illustrations of above algorithms are given in this section. The function chosen is $\log(1+x)$. Step 1 of each algorithm is explained.

The required initial forms are

$$\begin{aligned}
(0,0,0) &= y^2, \\
(1,0,0) &= -2x+2y+y^2, \\
(0,1,0) &= -xy+y^2, \\
(1,1,0) &= -6x+(6+2x)y+y^2, \\
(1,1,1) &= 12x-(12+6x)y+xy^2 .
\end{aligned}$$

Since the calculation of $C^i(n_0,n_1,n_2)$ values is essential for these algorithms, the details of obtaining $C^5(1,1,1)$ for $\log(1+x)$ are given as an example.

The $(1,1,1)$ form of $\log(1+x)$ may be written as

$$\begin{aligned}
&12x\times\{1\}-(12+6x)\times\left\{x-\frac{1}{2}x^2+\frac{1}{3}x^3-\frac{1}{4}x^4+\frac{1}{5}x^5+O(x^6)\right\} \\
&\quad +x\times\left\{x^2-x^3+\frac{11}{12}x^4+O(x^5)\right\}.
\end{aligned}$$

The coefficient of x^5 in this form is the sum of coefficients of x^5 in each term and hence

$$C^5(1,1,1) = 0 - \left\{12 \times \frac{1}{5} + 6 \times \left(-\frac{1}{4}\right)\right\} + \left\{1 \times \frac{11}{12}\right\},$$
$$= \frac{1}{60}.$$

Hereafter, the method of calculation of $C^i(n_0, n_1, n_2)$ values is assumed known.

4.1 In algorithm R1, $n = 0$ gives the $(1,1,1)$ form at the end of Step 1. The details of this step are explained below.

Initialize : The coefficient of x^3 is eliminated from the $(0,1,0)$ and $(1,0,0)$ forms to obtain the $(1,1,0)$ form. These coefficients are given by

$$C^3(0,1,0) = -\frac{1}{2},$$
$$C^3(1,0,0) = -\frac{1}{3},$$

and hence the $(1,1,0)$ form (which is $O(x^4)$) is obtained by

$$(1,1,0) = (0,1,0) - \left(\frac{3}{2}\right)(1,0,0),$$
$$= (-xy + y^2) - \left(\frac{3}{2}\right)(-2x + 2y + y^2),$$
$$= -\frac{1}{2}\left[-6x + (6 + 2x)y + y^2\right].$$

Step 1.1 : From the initial forms, the following coefficient values are read off. Substituting the values

$$E[(0,0,0), y] = 0,$$

$$E[(0,1,0), xy] = -1,$$

$$E[(0,0,0), 1] = 0,$$

$$E[(1,0,0), x] = -2,$$

in the expression at step 1.1 gives

$$(0,0,1) = x(0,0,0) - 0 - 0,$$

$$= xy^2.$$

Step 1.2 : The $(1,0,1)$ form is obtained by eliminating the coefficient of x^3 from the $(0,0,1)$

and $(1,0,0)$ forms. Thus

$$C^3(0,0,1) = 1,$$
$$C^3(1,0,0) = -\frac{1}{3},$$

give

$$\begin{aligned}(1,0,1) &= (0,0,1) + 3(1,0,0),\\ &= xy^2 + 3\big(-2x + 2y + y^2\big),\\ &= -6x + 6y + (3+x)y^2,\end{aligned}$$

which is the form of the required type with $O\big(x^4\big)$.

Step 1.3 : The coefficient of x^4 is eliminated from the $(1,0,1)$ and $(1,1,0)$ forms to obtain the $(1,1,1)$ form. Thus

$$C^4(1,0,1) = \frac{1}{4},$$
$$C^4(1,1,0) = \frac{1}{12},$$

give

$$\begin{aligned}(1,1,1) &= (1,0,1) - 3(1,1,0),\\ &= -6x + 6y + (3+x)y^2 - 3\big(-6x + (6+2x)y + y^2\big),\\ &= 12x - (12+6x)y + xy^2.\end{aligned}$$

4.2 In algorithm R2, $n = 0$ gives the $(1,1,1)$ form at the end of Step 1.

Step 1: The constants α_1 and α_2 are chosen to eliminate the coefficients of x^3 and x^4 respectively. Thus

$$C^3(x(0,0,0)) = 1,$$
$$C^3(1,0,0) \quad = -\frac{1}{3},$$

give $\alpha_1 = 3$. The coefficients of x^4 are

$$C^4(x(0,0,0)) = -1,$$
$$C^4(1,0,0) = \frac{5}{12},$$
$$C^4(1,1,0) = \frac{1}{12}.$$

On substitution, the expression for α_{6n+2} in step 1 becomes $\alpha_2 = -[(-1) + \alpha_1.(5/12)]/(1/12)$, which together with $\alpha_1 = 3$, gives $\alpha_2 = -3$. Hence

$$\begin{aligned}(1,1,1) &= x(0,0,0) + 3(1,0,0) + (-3)(1,1,0),\\ &= xy^2 + 3(-2x + 2y + y^2) + (-3)(-6x + (6+2x)y + y^2),\\ &= 12x - (12 + 6x)y + xy^2\,.\end{aligned}$$

4.3 In algorithm R3, $n = 0$ gives the $(2,1,1)$ form at the end of Step 1. Also, the notation q_1, q_2, q_3, q_4 denotes the forms $(0,0,0), (1,0,0), (1,1,0)$ and $(1,1,1)$ respectively. Using f_0, the constants $\gamma_1, \gamma_2, \gamma_3, \gamma_4$ are chosen to eliminate the coefficients of x^2, x^3, x^4, x^5 respectively. Now, these constants are used to obtain the polynomials P_j, $j = 0, 1, 2$, of the $(2,1,1)$ form.

Step 1 : For the coefficient of x^2,

$$C^2(0,0,0) = 1,$$

and using the equations in step 1 of this algorithm, with $k = 0$, give

$$1 + \gamma_1 = 0 \qquad \Rightarrow \gamma_1 = -1.$$

For the coefficients of x^3,

$$\begin{aligned}C^3(0,0,0) &= -1,\\ C^3(1,0,0) &= -\frac{1}{3},\end{aligned}$$

and hence substituting in the equations with $k = 1$ give

$$0 - \gamma_1 - \frac{1}{3}\gamma_2 = 0 \qquad \Rightarrow \gamma_2 = 3.$$

For the coefficients of x^4,

$$\begin{aligned}C^4(0,0,0) &= \frac{11}{12},\\ C^4(1,0,0) &= \frac{5}{12},\\ C^4(1,1,0) &= \frac{1}{12},\end{aligned}$$

and hence the equations with $k = 2$ give

$$0 + \frac{11}{12}\gamma_1 + \frac{5}{12}\gamma_2 + \frac{1}{12}\gamma_3 = 0 \qquad \Rightarrow \gamma_3 = -4.$$

For the coefficients of x^5,

$$C^5(0,0,0) = -\frac{5}{6},$$
$$C^5(1,0,0) = -\frac{13}{30},$$
$$C^5(1,1,0) = -\frac{2}{15},$$
$$C^5(1,1,1) = \frac{1}{60},$$

and hence

$$0 - \frac{5}{6}\gamma_1 - \frac{13}{30}\gamma_2 - \frac{2}{15}\gamma_3 + \frac{1}{60}\gamma_4 = 0 \qquad \Rightarrow \gamma_4 = -4.$$

Recalling that $q_{i,j}$ denotes the jth polynomial coefficient of the form q_i, and using the equations in step 1 for the polynomial coefficients P_j, the polynomial coefficient P_0 of the $(2,1,1)$ form is given by

$$\begin{aligned} P_0 &= \gamma_1.q_{1,0} + \gamma_2.q_{2,0} + \gamma_3.q_{3,0} + \gamma_4.q_{4,0}, \\ &= \gamma_1.(0) + \gamma_2.(-2x) + \gamma_3.(-6x) + \gamma_4.(12x) + x^2, \\ &= -30x + x^2. \end{aligned}$$

Similarly,

$$\begin{aligned} P_1 &= \gamma_1.(0) + \gamma_2.(2) + \gamma_3.(6+2x) + \gamma_4.(-12-6x), \\ &= 30 + 16x, \end{aligned}$$

and

$$\begin{aligned} P_2 &= \gamma_1.(1) + \gamma_2.(1) + \gamma_3.(1) + \gamma_4.(x), \\ &= -(2+4x). \end{aligned}$$

Hence

$$\begin{aligned} (2,1,1) &= P_0 + P_1 y + P_2 y^2, \\ &= -\left[30x - x^2 - (30+16x)y + (2+4x)y^2\right]. \end{aligned}$$

5. Results

These algorithms, along with the direct method, were implemented in PRO-MATLAB with 15 digit precision, for different functions and the solutions are given below in tabulated form. The numerical stability of these methods are tested against the solutions obtained from symbolic computation using MACSYMA.

The implementation of these algorithms in Matlab is carried out in the following way. As the first step, the known initial forms are stored as partitioned vectors where each partition corresponds to the polynomial coefficients of $1, f(x), f(x)^2$ in the (n_0, n_1, n_2) quadratic form. Once the power series coefficients of f_1 are also stored as a vector, the same length vector of f_2 is obtained after using convolution on f_1. Now, using the appropriate components of these vectors, the necessary $C^i(n_0, n_1, n_2)$ values can be easily obtained. For example, if the $(1,1,1)$ form of $\log(1+x)$ is stored as $\mathbf{r} = [0, 12; -12, -6; 0, 1]$, then (as in the example at the beginning of section 4)

$$C^5(1,1,1) = \mathbf{r}(3) * \mathbf{f1}(6) + \mathbf{r}(4) * \mathbf{f1}(5) + \mathbf{r}(5) * \mathbf{f2}(6) + \mathbf{r}(6) * \mathbf{f2}(5),$$

where $\mathbf{f1} = [0, 1, -\frac{1}{2}, \frac{1}{3}, -\frac{1}{4}, \frac{1}{5}, -\frac{1}{6}]$, and $\mathbf{f2} = [0, 0, 1, -1, \frac{11}{12}, -\frac{5}{6}, \frac{137}{180}]$.

Hence $C^5(1,1,1) = \frac{1}{60}$.

The rest of this simple technique is straightforward. It is worth noting that this technique of getting the required $C^i(n_0, n_1, n_2)$ values, does not require a linear system to be solved nor depends on the matrix F (see [6]).

The functions chosen are

(i) $\sin(x)$, (ii) $\log(1+x)$, (iii) $\exp(x)$, (iv) $\log((1+x)/(1-x))$, and (v) $\exp(-x) * \sin(x)$. Recall that the algorithms R1, R2 and R3 fail for functions which satisfy the properties (A) or (B) in section 3. In this respect, even functions are excluded.

It should be noted that because some diagonal forms have very large coefficients, ℓ_2 normalization of each coefficient is carried out for all diagonal forms to give $\|(n,n,n)\|_2 = 1$

where $\|(n,n,n)\|_2$ means the ℓ_2 norm of the vector formed from the coefficients of (n,n,n) form.

To demonstrate the fact that the number of digits in error increases as n increases in the (n,n,n) form for all tested functions, the normalized coefficients of the $(4,4,4)$, $(5,5,5)$, $(6,6,6)$ forms of $f(x) = \left(1+x+x^2\right)^{1/3}$ are given in Tables 1, 2 and 3. Since it is not easy to analyse the error in this way, the ℓ_2 norm of the error in the normalized coefficients of some diagonal forms are calculated for all six functions and are given in Tables 4, 5, 6, 7, 8 and 9. In every diagonal form the normalized coefficients obtained by symbolic computation are taken to be the true normalized coefficients. The condition number (in the l_2 norm) of the coefficient matrix F is denoted by cond.

function: $\left(1+x+x^2\right)^{1/3}$ form: $(4,4,4)$ cond: 8.7979×10^5

Coeff.	Macsyma	Direct	Recursion-1	Recursion-2	Recursion -3
x^4y^2	0.00017971125166	0.00017971125166	0.00017971125154	0.00017971125174	0.00017971125135
x^3y^2	0.03144808351709	0.03144808351706	0.03144808351658	0.03144808351730	0.03144808351540
x^2y^2	0.13375595753707	0.13375595753732	0.13375595754234	0.13375595753277	0.13375595755189
xy^2	-0.01055874823260	-0.01055874823230	-0.01055874822673	-0.01055874823788	-0.01055874821564
y^2	-0.06834686641991	-0.06834686641985	-0.06834686641765	-0.06834686642262	-0.06834686641499
x^4y	-0.00977871371480	-0.00977871371475	-0.00977871371376	-0.00977871371545	-0.00977871371188
x^3y	-0.14101363490567	-0.14101363490550	-0.14101363490275	-0.14101363490738	-0.14101363489643
x^2y	-0.41693790779769	-0.41693790779783	-0.41693790780079	-0.41693790779476	-0.41693790780614
xy	-0.35262929117739	-0.35262929117759	-0.35262929118159	-0.35262929117342	-0.35262929118893
y	-0.22724933422613	-0.22724933422636	-0.22724933423092	-0.22724933422204	-0.22724933423957
x^4	0.00360635339368	0.00360635339351	0.00360635339037	0.00360635339604	0.00360635338426
x^3	0.19531709756608	0.19531709756566	0.19531709755850	0.19531709757171	0.19531709754331
x^2	0.49623454620279	0.49623454620256	0.49623454619794	0.49623454620698	0.49623454618923
x	0.48450239509864	0.48450239509857	0.48450239509707	0.48450239510038	0.48450239509443
1	0.29559620064604	0.29559620064620	0.29559620064855	0.29559620064469	0.29559620065457

Table 1

function: $(1+x+x^2)^{1/3}$ form: $(5,5,5)$ cond: 8.3869×10^7

Coeff.	Macsyma	Direct	Recursion-1	Recursion-2	Recursion-3
x^5y^2	0.00038144113592	0.00038144113595	0.00038144080447	0.00038144104352	0.00038144111031
x^4y^2	0.00062490973463	0.00062490973526	0.00062490055112	0.00062490717426	0.00062490895609
x^3y^2	-0.11779690040865	-0.11779690040869	-0.11779690301312	-0.11779690131030	-0.11779690086858
x^2y^2	-0.40672391299201	-0.40672391299659	-0.40672384634202	-0.40672389476716	-0.40672390753875
xy^2	-0.30374864876978	-0.30374864877610	-0.30374855556979	-0.30374862331961	-0.30374864110629
y^2	-0.17467802921118	-0.17467802921585	-0.17467796015561	-0.17467801059233	-0.17467802367092
x^5y	-0.00507519317107	-0.00507519317149	-0.00507518752193	-0.00507519160320	-0.00507519271480
x^4y	-0.03277749326540	-0.03277749326901	-0.03277744107174	-0.03277747879903	-0.03277748889409
x^3y	0.13806787248367	0.13806787247668	0.13806797771890	0.13806790176583	0.13806788155849
x^2y	0.48233268511894	0.48233268511194	0.48233278765170	0.48233271332818	0.48233269364468
xy	0.47980580285525	0.47980580285182	0.47980585290750	0.47980581660265	0.47980580700318
y	0.31320636220353	0.31320636220312	0.31320636549889	0.31320636295823	0.31320636226677
x^5	0.03699000376372	0.03699000376498	0.03698998702611	0.03698999917330	0.03699000245307
x^4	0.19906637643752	0.19906637644537	0.19906626374071	0.19906634540533	0.19906636714486
x^3	0.17230290257056	0.17230290258638	0.17230267089993	0.17230283878254	0.17230288326655
x^2	-0.00560300482674	-0.00560300480711	-0.00560329190016	-0.00560308332191	-0.00560302838851
x	-0.16400725534586	-0.16400725533286	-0.16400744573809	-0.16400730720744	-0.16400727087205
1	-0.13852833299235	-0.13852833298727	-0.13852840534595	-0.13852835236605	-0.13852833859609

Table 2

function: $(1+x+x^2)^{1/3}$ form: $(6,6,6)$ cond: 1.5616×10^9

Coeff.	Macsyma	Direct	Recursion-1	Recursion-2	Recursion-3
x^6y^2	0.00009397568467	0.00009397568447	0.00009397451124	0.00009397511825	0.00009397598935
x^5y^2	0.00535234204844	0.00535234204192	0.00535230853629	0.00535232374803	0.00535235662246
x^4y^2	0.00930280478926	0.00930280477780	0.00930278633566	0.00930277605554	0.00930286908213
x^3y^2	-0.12392705142926	-0.12392705148388	-0.12392719631612	-0.12392719240834	-0.12392679871065
x^2y^2	-0.31363747184769	-0.31363747206651	-0.31363846278124	-0.31363807707750	-0.31363685833230
xy^2	-0.28740731619048	-0.28740731635140	-0.28740807003357	-0.28740776340789	-0.28740688875468
y^2	-0.16368974433506	-0.16368974444708	-0.16369031461578	-0.16369005946418	-0.16368948958539
x^6y	-0.00243519988897	-0.00243519988533	-0.00243517937266	-0.00243518954288	-0.00243520635600
x^5y	-0.03936878102358	-0.03936878099080	-0.03936860403428	-0.03936868810961	-0.03936884624543
x^4y	-0.13077898716909	-0.13077898710186	-0.13077869627800	-0.13077880298277	-0.13077918990762
x^3y	-0.08540328298434	-0.08540328278902	-0.08540246853040	-0.08540275073740	-0.08540390092463
x^2y	0.04217564886276	0.04217564924892	0.04217738546061	0.04217671605489	0.04217455498656
xy	0.12721794768448	0.12721794802171	0.12721950026005	0.12721888307011	0.12721702734734
y	0.08747022171447	0.08747022192226	0.08747123479814	0.08747080424184	0.08746971050123
x^6	0.00867257441154	0.00867257440259	0.00867251614027	0.00867254826719	0.00867258282277
x^5	0.11406468743188	0.11406468738746	0.11406439963720	0.11406455704028	0.11406472964559
x^4	0.36664257048176	0.36664257045345	0.36664237798903	0.36664248692718	0.36664258957310
x^3	0.52150988493618	0.52150988484108	0.52150943494776	0.52150962104017	0.52151013481198
x^2	0.49216163768891	0.49216163753250	0.49216096876969	0.49216120769734	0.49216211195061
x	0.24015912415788	0.24015912398699	0.24015836791418	0.24015865190001	0.24015961762892
1	0.07621952262059	0.07621952252482	0.07621907981574	0.07621925522225	0.07621977908177

Table 3

function : $\sin(x)$

(4,4,4)	Cond:	2.2152×10^{8}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		1.0048×10^{-14}	1.4092×10^{-14}	1.4092×10^{-14}	9.9365×10^{-15}
(5,5,5)	Cond:	5.8489×10^{10}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		0	0	0	0
(6,6,6)	Cond:	1.0372×10^{15}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		0	1.0766×10^{-13}	3.0089×10^{-14}	3.0089×10^{-14}
(7,7,7)	Cond:	1.9230×10^{17}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		0	0	0	1.0002×10^{-14}
(8,8,8)	Cond:	2.5583×10^{22}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		5.7847×10^{-13}	3.5545×10^{-12}	1.7358×10^{-12}	3.3064×10^{-12}
(9,9,9)	Cond:	9.2488×10^{24}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		9.9989×10^{-15}	4.2427×10^{-14}	0	0
(10,10,10)	Cond:	1.6316×10^{27}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		6.1395×10^{-12}	3.7743×10^{-12}	6.2254×10^{-12}	2.7219×10^{-11}

Table 4

function : $\log(1+x)$

(2,2,2)	Cond:	1.3614×10^{5}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		4.4449×10^{-13}	1.0833×10^{-11}	6.9408×10^{-13}	3.4022×10^{-11}
(3,3,3)	Cond:	2.1669×10^{7}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		1.5694×10^{-11}	5.8407×10^{-10}	3.9113×10^{-11}	3.1394×10^{-9}
(4,4,4)	Cond:	1.9212×10^{10}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		2.1320×10^{-10}	2.8009×10^{-7}	2.5135×10^{-8}	1.5847×10^{-5}
(5,5,5)	Cond:	8.3646×10^{12}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		1.1338×10^{-5}	5.9262×10^{-4}	1.4628×10^{-5}	0.0397
(6,6,6)	Cond:	3.2681×10^{15}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		0.0037	0.1837	0.0053	0.8920
(7,7,7)	Cond:	5.7086×10^{17}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		0.6299	0.1837	0.6066	0.5035

Table 5

function: $\exp(x)$

(2,2,2)	Cond:	7.6554×10^{4}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		2.4920×10^{-13}	2.1725×10^{-13}	1.8001×10^{-13}	3.0332×10^{-13}
(3,3,3)	Cond:	1.7432×10^{8}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		9.1380×10^{-12}	8.0455×10^{-12}	1.3591×10^{-12}	1.9235×10^{-11}
(4,4,4)	Cond:	5.3691×10^{11}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		6.4934×10^{-10}	6.5283×10^{-10}	4.6075×10^{-10}	3.8246×10^{-10}
(5,5,5)	Cond:	5.8321×10^{15}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		1.8532×10^{-7}	1.9564×10^{-7}	1.9123×10^{-7}	2.9783×10^{-8}
(6,6,6)	Cond:	6.1882×10^{19}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		1.0496×10^{-5}	7.1367×10^{-6}	7.3656×10^{-6}	8.4475×10^{-7}
(7,7,7)	Cond:	4.0631×10^{22}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		3.8575×10^{-4}	4.4644×10^{-4}	6.5574×10^{-4}	9.9262×10^{-5}
(8,8,8)	Cond:	1.2094×10^{25}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		0.0168	0.0046	0.0137	0.0012

Table 6

function : $\log((1+x)/(1-x))$

(3,3,3)	Cond:	3.9966×10^{4}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		9.9365×10^{-15}	4.9087×10^{-14}	1.7387×10^{-14}	7.8776×10^{-14}
(4,4,4)	Cond:	2.2169×10^{6}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		1.7330×10^{-13}	4.4885×10^{-12}	2.3655×10^{-13}	6.5523×10^{-13}
(5,5,5)	Cond:	9.8596×10^{7}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		1.1797×10^{-11}	1.8268×10^{-9}	7.0080×10^{-12}	7.2540×10^{-9}
(6,6,6)	Cond:	5.1231×10^{9}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		3.8743×10^{-10}	1.6423×10^{-7}	2.1621×10^{-10}	7.9490×10^{-7}
(7,7,7)	Cond:	2.4953×10^{11}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		1.0244×10^{-8}	5.9889×10^{-5}	1.0748×10^{-8}	4.4130×10^{-5}
(8,8,8)	Cond:	1.2954×10^{13}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		2.8148×10^{-7}	0.0074	5.7392×10^{-7}	0.0356
(9,9,9)	Cond:	6.4488×10^{14}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		4.2394×10^{-6}	0.4233	2.4588×10^{-5}	0.5420

Table 7

function : $\exp(-x) * \sin(x)$

(3,3,3)	Cond:	2.0534×10^{4}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		9.9998×10^{-15}	1.4141×10^{-14}	1.7315×10^{-14}	1.4136×10^{-14}
(4,4,4)	Cond:	4.7458×10^{7}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		4.7366×10^{-13}	2.7241×10^{-11}	6.0976×10^{-12}	4.3870×10^{-13}
(5,5,5)	Cond:	1.2590×10^{10}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		2.3316×10^{-12}	1.3632×10^{-11}	7.5431×10^{-12}	2.6941×10^{-12}
(6,6,6)	Cond:	1.0032×10^{13}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		2.2881×10^{-11}	5.6358×10^{-12}	3.0353×10^{-11}	1.2445×10^{-11}
(7,7,7)	Cond:	1.4851×10^{16}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		1.0240×10^{-10}	2.2843×10^{-10}	5.3266×10^{-11}	7.8610×10^{-11}
(8,8,8)	Cond:	2.1543×10^{20}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		2.2074×10^{-7}	2.0211×10^{-7}	2.0245×10^{-7}	2.4072×10^{-7}
(9,9,9)	Cond:	1.0281×10^{23}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		6.8338×10^{-9}	7.5116×10^{-9}	2.1606×10^{-9}	6.7458×10^{-8}

Table 8

function : $\left(1+x+x^2\right)^{1/3}$

(2,2,2)	Cond:	1.1209×10^3			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		0	0	2.8262×10^{-14}	2.0033×10^{-14}
(3,3,3)	Cond:	1.8006×10^4			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		1.4150×10^{-14}	9.1692×10^{-13}	4.7403×10^{-13}	7.9487×10^{-13}
(4,4,4)	Cond:	8.7979×10^5			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		7.6789×10^{-13}	1.5103×10^{-11}	1.2564×10^{-11}	4.3533×10^{-11}
(5,5,5)	Cond:	8.3869×10^7			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		3.3163×10^{-11}	4.8518×10^{-7}	1.3292×10^{-7}	3.9999×10^{-8}
(6,6,6)	Cond:	1.5616×10^9			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		7.1694×10^{-10}	3.2667×10^{-6}	1.9848×10^{-6}	2.0058×10^{-6}
(7,7,7)	Cond:	3.2198×10^{10}			
	2-norm of error	Direct	Recursion-1	Recursion-2	Recursion-3
		1.8623×10^{-8}	1.0096×10^{-5}	1.0072×10^{-5}	1.3727×10^{-4}

Table 9

6. Conclusion

This paper has evaluated the numerical effectiveness of available methods for computing the polynomial coefficients of the quadratic function approximation.

From the numerical point of view, two things are apparent.

(i) The numerical results from the recursive methods are usually no better than that from the direct method and are generally even worse.

(ii) All four methods show the error increasing as n increases in the (n, n, n) form as expected, but the rate of increase varies significantly for different functions. In the worst case tested, the $(7, 7, 7)$ form for $\log(1 + x)$ has virtually no significant figures correct.

In this context, the efficiency of these methods becomes immaterial even though the recursive methods each require only $O(n^2)$ operations to generate the (n, n, n) form, against the direct method which of course requires $O(n^3)$ operations.

The failure of the direct method to produce accurate results is no surprise, but on the other hand the failure of the recursive methods (with no direct involvement of the ill-conditioned matrix F at any stage) suggests that any recursive method based on elimination would not yield numerical stability owing to the fact that these processes are essentially similar to the elimination process on the matrix. An accurate evaluation of the numerical coefficients would seem to require a different formulation of the problem.

7. References

1. G.A.Baker, "Essentials of Padé Approximants", Academic Press, New York, 1975.
2. R.G.Brookes, and A.W.McInnes, *The Existence and Local Behavior of the Quadratic Function Approximation*, J. Approx. Theory **62**, 383–395, 1990.
3. R.G.Brookes, and A.W.McInnes, *A Recurrence Algorithm For Quadratic Hermite-Padé Forms*, Research Report **48**, Department of Mathematics, University of Canterbury, 1989.

4. P.B.Borwein, *Quadratic Hermite-Padé Approximation to the Exponential Function,* Constr. Approx., **2**, 291–302, 1986.

5. A.W.McInnes, *Existence and Uniqueness of Algebraic Function Approximations,* Constr. Approx., (to appear), 1991.

6. S.Paszkowski, *Recurrence relations in Padé-Hermite approximation,* J. Comput., Appl. Math., **19**, 99–107, 1987.