

# 7 Mathematical Methods

## 7.5 Padé Approximants

(8 units)

*This project is essentially self-contained, though the Part II course Asymptotic Methods provides background to Question 4.*

### 1 Introduction

A Padé approximant is a rational function, i.e., a function expressed as a fraction whose numerator and denominator are both polynomials, whose power series expansion agrees with a given power series to the highest possible order.

The primary application of Padé approximants is to problems where it is possible to derive the solution formally as a power series expansion in some parameter. The corresponding Padé approximants often turn out to be much more useful than the power series itself (in a sense to be explored in this project).

Given the power series

$$f(x) = \sum_{k=0}^{\infty} c_k x^k \quad (1)$$

the  $[L, M]$  Padé approximant  $R_{L,M}(x)$  is defined by

$$R_{L,M}(x) = \frac{\sum_{k=0}^L p_k x^k}{1 + \sum_{k=1}^M q_k x^k} \quad (2)$$

such that

$$f(x) - R_{L,M}(x) = O(x^{L+M+1}), \quad (3)$$

i.e., the first  $L + M + 1$  terms of the power series of  $R_{L,M}(x)$  match the first  $L + M + 1$  terms of the power series of  $f(x)$ .

Equations for the coefficients  $p_k$ ,  $k = 0, \dots, L$  and  $q_k$ ,  $k = 1, \dots, M$  can be obtained by multiplying (3) by the denominator of  $R_{L,M}(x)$  and equating coefficients of  $x^k$  for  $k = 0, \dots, L + M$ .

The result is  $M$  simultaneous equations for the  $q_k$ ,  $k = 1, \dots, M$ ,

$$\sum_{k=1}^{\min(r,M)} q_k c_{r-k} = -c_r \quad (r = L + 1, \dots, L + M) \quad (4)$$

and  $L + 1$  expressions for the  $p_k$ ,  $k = 0, \dots, L$ ,

$$p_k = c_k + \sum_{s=1}^{\min(k,M)} q_s c_{k-s} \quad (k = 0, \dots, L) \quad (5)$$

In many cases it is convenient to consider only ‘diagonal’ Padé approximants with  $L = M$ . But sometimes this may not be possible, e.g., for special forms of the power series the simultaneous equations (4) corresponding to diagonal approximants may not have a solution. In that case it may be convenient to choose  $M = L + 1$ , or something similar.

**Programming Task:** You will need to write two general purpose programs for this project. You should use MATLAB's 64-bit (8-byte) double-precision floating-point values or the equivalent in other programming languages.

Program A should solve equations (4) and (5) given the coefficients  $c_k$  and the values of  $L$  and  $M$  and evaluate the resulting Padé approximant  $R_{L,M}(x)$  for a specified set of values of  $x$ . You may use a library routine to solve the simultaneous equations (4). For example, if using MATLAB you can use the built-in matrix division routines such as `mldivide`. If you are not using MATLAB it may be worth using iterative improvement. (See Appendix.)

Program B should find the (possibly complex) roots of a polynomial, given the coefficients. If using MATLAB, the `roots` routine makes this program particularly easy to write. Alternatively, two straightforward possibilities are discussed in section 9.5 of [3]. If a (possibly complex) roots of a polynomial routine is available with the programming language you choose, just use it.

## 2 Estimating functions defined by power series

Consider the function  $f_1(x) = (1+x)^{1/2}$ .

**Question 1** Derive the power series expansion for  $f_1(x)$  about  $x = 0$ , deducing a formula for the coefficients  $c_k$  for arbitrary  $k$ . What is the radius of convergence of the power series and what limitations does this put on using the power series to estimate  $f_1(x)$ ? Noting that the power series converges for  $x = 1$  investigate the convergence of the partial sums  $\sum_{k=0}^N c_k$  as  $N$  increases and display selected results. Regarding the partial sum as an estimate for  $\sqrt{2}$ , how does the error vary with  $N$  as  $N$  increases?

**Question 2** Use your program to determine the Padé approximant  $R_{L,L}(x)$  and evaluate this for  $x = 1$ . Again regarding this as an estimate for  $\sqrt{2}$ , how does the error vary with  $L$  as  $L$  increases? What is the smallest value to which the error can be reduced? What determines this smallest value? Does iterative improvement to the solution of (4) make any difference?

Compare the results for the power series and for the Padé approximant. Which method would you recommend to give an estimate for  $\sqrt{2}$  to specified accuracy?

**Question 3** Now consider  $x$  in the range  $1 < x \leq 100$ . Compare power series estimates and Padé approximant estimates for  $f_1(x)$  for a few choices of  $N$  and  $L$ . Display the results graphically and discuss. For two chosen values of  $x$  (e.g.,  $x = 10$  and  $x = 100$ ) investigate carefully how the error in the Padé approximant estimates varies as  $L$  increases, display the results graphically and discuss. What are the implications for using the Padé approximant to estimate  $f_1(x)$  for large  $x$ ?

Now consider the function  $f_2(x) = \int_0^\infty e^{-t}(1+xt)^{-1}dt$ , which is defined for all real  $x \geq 0$  (in fact, everywhere in the complex  $x$ -plane except the negative real axis). Replacing  $(1+xt)^{-1}$  by its Maclaurin expansion and integrating term-by-term gives the *asymptotic expansion*

$$1 - 1!x + 2!x^2 - 3!x^3 + 4!x^4 - 5!x^5 + \dots \quad (6)$$

This diverges for all  $x \neq 0$ , which is hardly surprising since the Maclaurin expansion of  $(1+xt)^{-1}$  diverges for  $t \geq x^{-1}$ . Nevertheless, *when truncated at a finite number of terms*, the series gives

a “good” approximation to  $f_2(x)$  when  $x$  is “small”. (A more precise statement of this result, and its justification by Watson’s Lemma, can be found in [1] or [2], or any of the books listed in the schedule for the Part II *Asymptotic Methods* course.)

**Question 4** Regarding the asymptotic series (6) as a power series, use program A to generate Padé approximants for  $f_2(x)$ . Compare the truncated power series and the Padé approximants as a basis for calculating  $f_2(x)$  on the range  $0 \leq x \leq 20$ . Note that numerical integration gives the following values, correct to eight decimal places:

$x$	$f_2(x)$
0.1000	0.91563334
0.2000	0.85211088
0.3000	0.80118628
0.4000	0.75881459
0.5000	0.72265723
0.6000	0.69122594
0.7000	0.66351027
0.8000	0.63879110
0.9000	0.61653779
1.0000	0.59634736
2.0000	0.46145532
3.0000	0.38560201
4.0000	0.33522136
5.0000	0.29866975
6.0000	0.27063301
7.0000	0.24828135
8.0000	0.22994778
9.0000	0.21457710
10.0000	0.20146425
11.0000	0.19011779
12.0000	0.18018332
13.0000	0.17139800
14.0000	0.16356229
15.0000	0.15652164
16.0000	0.15015426
17.0000	0.14436271
18.0000	0.13906806
19.0000	0.13420555
20.0000	0.12972152

### 3 Zeros and poles

**Question 5** Use Program B to determine (in the complex  $x$ -plane) the poles and zeros of the Padé approximant  $R_{L,L}(x)$  for  $f_1(x)$ . Investigate carefully how the positions of the poles and zeros change as  $L$  is increased.

Carry out the same investigation for the functions  $f_3(x) = (1+x)^{-1/2}$ ,  $f_4(x) = e^x$ ,  $f_5(x) = e^x/(1+x)$  and  $f_6(x) = (1+x+x^2)^{1/2}$ . [For  $f_5(x)$  and  $f_6(x)$  your program will have to do some straightforward calculation to evaluate the coefficients in the power series.]

On this basis can you suggest how the positions of poles and zeros of the Padé approximants correspond to any poles, zeros, branch points and branch cuts of the approximated functions? Relate your comments carefully to specific properties of each of the functions considered. Provide a selection of results in the form of plots or short tables to support your comments.

Do you find ‘anomalous’ poles and zeros of the approximants that do not match poles, zeros, branch points or branch cuts of the approximated function? You will find many such cases for  $f_4(x)$  and  $f_5(x)$ , but should also find cases for  $f_1(x)$  and  $f_3(x)$ , particularly when  $L$  is large. What do you notice about the anomalous poles and zeros in these latter cases?

Comment on any problems that might be encountered in using Padé approximants to estimate  $f_6$  along the real  $x$ -axis. Display one or two relevant graphs.

## Appendix: Iterative improvement of the solution of linear simultaneous equations

Consider the set of equations  $A\mathbf{x} = \mathbf{b}$ , where  $A$  is a square matrix,  $\mathbf{b}$  is the column vector of right-hand sides and  $\mathbf{x}$  is the column vector of unknowns.

Suppose that numerical solution has generated the approximate solution  $\mathbf{y}$ . Now suppose that the true solution is given by  $\mathbf{x} = \mathbf{y} + \delta\mathbf{y}$ . Multiplying by  $A$  implies that  $A\delta\mathbf{y} = \mathbf{b} - A\mathbf{y}$ . This is a set of equations for the correction  $\delta\mathbf{y}$  to the approximate solution and solving gives an estimate for the correction, and hence a refinement to the solution. This procedure may be repeated until no further improvement is found.

Note that at each refinement the set of simultaneous equations to be solved has the same associated matrix  $A$ . Only the right-hand sides change. Therefore there is advantage in using an approach such as  $LU$  decomposition, since once the  $LU$  decomposition of  $A$  has been calculated it may be used repeatedly to solve the simultaneous equations occurring at each refinement.

## References

- [1] Hinch, E.J., *Perturbation Methods*.
- [2] Bender, C. & Orszag, S.A., *Advanced Mathematical Methods for Scientists and Engineers*.
- [3] Press *et al.*, *Numerical Recipes in C*.