PROGRAMA DE COMUTAÇÃO BIBLIOGRÁFICA
CAPES / SESU / FINEP / IBICT

FORMULÁRIO DE ENCAMINHAMENTO - PERIÓDICO

Nº PEDIDO PE000474612/2010

IDENTIFICAÇÃO DO PEDIDO

TÍTULO DO PERIÓDICO: BIT

ANO: 1989 VOLUME: 29 FASCÍCULO/MÊS: 1 SUPLEMENTO: ISSN: 0006-3835

AUTOR DO ARTIGO: CHRISTIANSEN, E.; PETERSON, H. G.

TÍTULO DO ARTIGO: ESTIMATION OF CONVERGENCE ORDERS IN REPEATED RICHARDSON EXTRAPOLATION

PÁGINA INICIAL: 48 PÁGINA FINAL: 59 TOTAL DE PÁGINAS: 12 BÔNUS UTILIZADOS: 3

FORMA DE ENVIO: E-MAIL

SITUAÇÃO DO PEDIDO: [ ] Atendido [ ] Repassado [ ] Cancelado

FORMA DO DOC.ORIGINAL: TOTAL DE PÁG.CONFIRMAÇÃO:

MOTIVO:

OBSERVAÇÃO:

FORMULÁRIO DE ENCAMINHAMENTO - PERIÓDICO

BIBLIOTECA-BASE

NOME: UNESP/BRP - BIBLIOTECA

ENDEREÇO: RUA CRISTÓVÃO COLOMBO, 2265

CEP: 15054000 CIDADE-UF: SAO JOSE DO RIO PRETO-SP

Nº PEDIDO PE000474612/2010

USUARIO: CPF: 58202811953

NOME: CARLOS HENRIQUE MARCHI

TEL: (41) 33613126 E-MAIL: marchi@demec.ufpr.br

SOLICITANTE: CÓDIGO ou CPF: 000742-0

NOME: UFPR/BCT - BIBLIOTECA - REFORMA

TEL: (41) 3361-3060 E-MAIL: comut@ufpr.br

IDENTIFICAÇÃO DO PEDIDO

TÍTULO DO PERIÓDICO: BIT

ANO: 1989 VOLUME: 29 FASCÍCULO/MÊS: 1 SUPLEMENTO: ISSN: 0006-3835

AUTOR DO ARTIGO: CHRISTIANSEN, E.; PETERSON, H. G.

TÍTULO DO ARTIGO: ESTIMATION OF CONVERGENCE ORDERS IN REPEATED RICHARDSON EXTRAPOLATION

PÁGINA INICIAL: 48 PÁGINA FINAL: 59 TOTAL DE PÁGINAS: 12 BÔNUS UTILIZADOS: 3

FORMA DE ENVIO: E-MAIL

FORMA DO DOC.ORIGINAL: TOTAL DE PÁG.CONFIRMAÇÃO:

DESTINATÁRIO

NOME: UFPR/BCT - BIBLIOTECA - REFORMA

ENDEREÇO: CENTRO POLITÉCNICO, S/N CAIXA POSTAL 19010

CEP: 81531990 **CIDADE-UF:** CURITIBA-PR

TEL: **E-MAIL** comut@ufpr.br

Recebi o pedido Nº Data _____/_____/_____ Assinatura_____

Voltar

BIT 29 (1989), 48–59

ESTIMATION OF CONVERGENCE ORDERS IN REPEATED RICHARDSON EXTRAPOLATION

EDMUND CHRISTIANSEN and HENRIK GORDON PETERSEN

Mathematics Department, Odense University, Campusvej 55, DK-5230 Odense M, Denmark

Abstract.

Let $A(h)$ be an approximation depending on a single parameter h to a fixed quantity A, and assume that $A-A(h) = c_1h^{k_1}+c_2h^{k_2}+\ldots$. Given a sequence of h-values $h_1 > h_2 > \ldots > h_n$ and corresponding computed approximations $A(h_i)$, the orders for repeated Richardson extrapolation are estimated, and the repeated extrapolation is performed. Hence in this case the algorithm described here can do the same work as Brezinski's E-algorithm and at the same time provide a check whether repeated extrapolation is justified.

Subject classification: AMS (MOS) 65B05.
CR G.1.0.

Keywords: Richardson extrapolation, repeated.

1. Introduction.

Let $A(h)$ denote an approximation to a fixed quantity $A = A(0)$, and assume that the discretization error in $A(h)$ satisfies

$$A-A(h) = c_1h^{k_1}+c_2h^{k_2}+\ldots, \quad k_1 < k_2 < \ldots \tag{1}$$

involving only powers in h. The powers need not be integer. We shall assume that (1) is a convergent series for small h-values.

If the exponents k_j are known, it is well known how to perform repeated Richardson extrapolation. Extrapolation has been generalized to more general error terms than simple powers of h. (See [1], [4] and [6] for formalism and algorithms.) The aim of this paper is to move in a different direction. We consider the basic form of Richardson extrapolation (1), and the main goal is to compute the order of the leading term in the error after one or more extrapolations.

If there is a constant ratio between successive h-values, then it is easy to compute approximations to the orders successively. This is so, because each

Received September 1987. Revised June 1988.

extrapolated value depends on only a single h-value, not on a pair as in the general case. More precisely the pair is always of the form say $(h, h/2)$, if the ratio is 2. However, for arbitrary h-values there seems to be no general method for computing more than the first order k_1. In many applications, for example in continuum mechanics, it is not possible to get sufficient results with a constant ratio betwen h-values. For all practical purposes it is then sufficient to consider only the case, when the h-values are of the form $h_i = c/i$ for some constant c. We shall concentrate on this case. It should be emphasized, however, that the algorithm presented here works equally well in the case $h_{i+1}/h_i = \text{constant}$, where the analysis is much simpler.

Even when the convergence orders are known, it is always prudent and in some cases necessary to estimate the convergence orders from the computed values $A(h_i)$ before performing extrapolation. Extrapolation is based on the assumption that $A(h)$ has the form (1) and that the error term to be eliminated is in fact the dominating source of error in $A(h_i)$ for the actual values of h_i. If this condition is not satisfied, extrapolation is at best useless. The most common reasons for this condition to fail are the following:

(a) Several terms of different order may contribute significantly to the error for the h-values used ("h is too large").
(b) An expected "known" order may vanish due to special properties of the solution. (An example is integration with the trapezoidal rule, when the integrand has the same slope at both ends of the interval.)
(c) A singularity in the (unknown) solution may cause a term in the error, which is particular to the application and not to the discretizations used in $A(h)$. (The paper concludes with a classical example of this.)
(d) Round-off error may be of significance relative to the discretization error. (This will certainly be the case after several extrapolations.)

By successively computing approximations to the repeated convergence orders the algorithm presented here can be used to check whether the remaining error after one or several extrapolations is dominated by a term of the form ch^k. The algorithm can then eliminate this error by performing the extrapolation to an order chosen by the user. In short, the present algorithm can in the case (1) replace the E-algorithm in [1] with the addition that it also computes approximate values for the repeated convergence orders. This will reveal, if for some reason the error is not dominated by a single term of the form ch^k.

2. The basic result.

Assume that $A(h)$ has been computed for some sequence of h-values

$$h(1) > h(2) > \ldots > 0$$

The following notation is used:

$$A(i) = A(h(i)) \qquad \text{for } i = 1, 2, \ldots$$

$$\alpha(i,k) = \left(\frac{h(i-1)}{h(i)}\right)^k \qquad \text{for } i = 1, 2, \ldots$$

A sequence of functions $f_n(i,k)$ is defined in the following recursive way:

$$f_1(i,k) = 1 \quad \text{for } i = 1, 2, \ldots \text{ and } k \text{ real.}$$

$$f_{n+1}(i,k) = \frac{\alpha(i,k_n)f_n(i-1,k_n)f_n(i,k) - \alpha(i,k)f_n(i,k_n)f_n(i-1,k)}{\alpha(i,k_n)f_n(i-1,k_n) - f_n(i,k_n)}$$

$$\text{for } n = 1, 2, \ldots, \; i = n+1, n+2 \ldots, \text{ and } k \text{ real.}$$

$f_n(i,k)$ depends on $h(i-n+1), \ldots, h(i)$ and $k_1, \ldots, k_{n-1}$. Hence f_n cannot be computed until k_{n-1} is known (or has been estimated). The definition of f_n will be motivated in lemma 2. The interpretation is as follows: $n-1$ is the number of extrapolations already performed, i is the index of h and A, and the real value k is the order of the next error term (to be found). In the special case (not of interest here), where the convergence orders are known to be $k_j = j$ for $j = 1, 2, \ldots$, the above sequence f_n can be related to the recursively defined functions g_n in [1]. For example

$$f_2(i,k) = h(i)^{-k} g_{1,k}^{(i-1)}.$$

In the definition of f_n the k_j's must be different, but may occur in any order. Corresponding to the easy case we note that if $h(i+1)/h(i)$ is constant, then $f_n(i,k)$ does not depend on i.

We shall need the following expression for f_{n+1}:

$$(2) \qquad f_{n+1}(i,k) = f_n(i,k)\frac{\alpha(i,k_n)f_n(i-1,k_n)/f_n(i,k_n) - \alpha(i,k)f_n(i-1,k)/f_n(i,k)}{\alpha(i,k_n)f_n(i-1,k_n)/f_n(i,k_n) - 1}$$

LEMMA 1. *Let $n \geq 2$ and $k_1, \ldots, k_{n-1}$ (all k_j distinct) be given. Then*

(a) $f_n(i,k)$ is invariant under permutations of $(k_1, \ldots, k_{n-1})$.
$f_n(i,k) = 0$, if k belongs to the set $\{k_1, \ldots, k_{n-1}\}$.

(b) Assume that $h(i)$ is of the form $h(i) = c/i$ for some constant c. For all fixed $k > 1$ we have

$$f_n(i,k) = (1-k/k_1)(1-k/k_2)\ldots(1-k/k_{n-1})(1+O(i^{-1}))$$

$$\frac{\partial f_n}{\partial k}(i,k) = f_n(i,k)\left(\frac{1}{k-k_1} + \frac{1}{k-k_2} + \ldots + \frac{1}{k-k_{n-1}}\right)(1+O(i^{-1}))$$

$$f_n(i,k)/f_n(i+1,k) = 1+O(i^{-2})$$

PROOF. The proof goes by induction on n:

(a) For $n = 2$ the statement is obvious. The induction hypothesis is that $f_j(i,k)$ is invariant under permutations of $(k_1,\ldots,k_{j-1})$ for $j \le n$. Hence if a permutation of $(k_1,\ldots,k_n)$ does not involve k_n, then it follows from the definition that f_{n+1} is unchanged. It is then sufficient to show that an interchange of k_n and k_{n-1} leaves f_{n+1} unchanged. The proof of this is a tedious calculation, and we only indicate the steps. In the definition of $f_{n+1}(i,k)$ we express f_n by f_{n-1} Since f_{n-1} depends only on $(k_1,\ldots,k_{n-2})$ it is now a matter of direct calculation to check that interchanging k_n and k_{n-1} does not change the value of $f_{n+1}(i,k)$. This proves the invariance.

It is clear from the definition that $f_{n+1}(i,k_n) = 0$. From the invariance just proved it then follows that $f_{n+1}(i,k) = 0$, if k belongs to $\{k_1,\ldots,k_n\}$.

(b) We repeatedly use the expansion

$$\left(\frac{i+1}{i}\right)^k = 1 + \frac{k}{i} + \frac{k(k-1)}{2i^2} + O(i^{-3}).$$

Using the equation (2) both the case $n = 2$ and the inductive step are now a matter of direct calculation. ■

Based on numerical computations, but without proof, we also conjecture the following sign variation of $f_n(i,k)$. It will not be used here.

$f_n(i,k) < 0$, if $k \notin \{k_1,\ldots,k_{n-1}\}$ and $k > k_j$ for an odd number of k_j.

$f_n(i,k) > 0$, if $k \notin \{k_1,\ldots,k_{n-1}\}$ and $k > k_j$ for an even number of k_j.

For $n = 1,2,\ldots$ and $i = n+1, n+2,\ldots$, let $R_n(i)$ denote the result of n Richardson extrapolations (removing the first n terms in the error) based on the values $A(i-n),\ldots,A(i)$. (Same definition as in [1]). For convenience let $R_0(i) = A(i)$.

LEMMA 2. *Assume that $h(i)$ is of the form $h(i) = c/i$ for some constant c. The result after n extrapolations with the correct orders $k_1,\ldots,k_n$ satisfies*

$$R_n(i) = \frac{\alpha(i,k_n)f_n(i-1,k_n)R_{n-1}(i) - f_n(i,k_n)R_{n-1}(i-1)}{\alpha(i,k_n)f_n(i-1,k_n) - f_n(i,k_n)} \tag{3}$$

and

$$A - R_n(i) = \sum_{j=n+1}^{\infty} c_j f_{n+1}(i,k_j)h(i)^{k_j} \tag{4}$$

for $i = n+1, n+2,\ldots$

PROOF. The proof goes by induction on n. For $n = 1$ (the standard step) we

truncate the equations (1) applied for i and $i-1$

$$A - A(h(i-1)) = c_1 h(i-1)^{k_1} + c_2 h(i-1)^{k_2} + \ldots$$

$$A - A(h(i)) \quad = c_1 h(i)^{k_1} \quad + c_2 h(i)^{k_2} \quad + \ldots$$

after 1 term and solve for A. The solution, which by definition is $R_1(i)$, is:

$$R_1(i) = \{\alpha(i, k_1)A(i) - A(i-1)\}/\{\alpha(i, k_1) - 1\}.$$

It is now a straightforward calculation to prove (4) in the case $n = 1$.

The inductive assumption is equation (4). For any $i \geq n+2$ apply (4) for the pair $i-1$ and i:

$$A - R_n(i-1) = \sum_{j=n+1}^{\infty} c_j f_{n+1}(i-1, k_j) h(i-1)^{k_j}$$

$$A - R_n(i) = \sum_{j=n+1}^{\infty} c_j f_{n+1}(i, k_j) h(i)^{k_j}.$$

Multiply the equations by $f_{n+1}(i, k_{n+1})$ and $\alpha(i, k_{n+1}) f_{n+1}(i-1, k_{n+1})$ respectively. Subtraction eliminates the term of order $n+1$ and yields:

$$A = \frac{\alpha(i, k_{n+1}) f_{n+1}(i-1, k_{n+1}) R_n(i) - f_{n+1}(i, k_{n+1}) R_n(i-1)}{\alpha(i, k_{n+1}) f_{n+1}(i-1, k_{n+1}) - f_{n+1}(i, k_{n+1})} +$$

$$+ \sum_{j=n+2}^{\infty} c_j f_{n+2}(i, k_j) h(i)^{k_j}.$$

This holds for all $i \geq n+2$. By lemma 1 (b) the coefficients $f_{n+2}(i, k_j)$ are bounded in i, and hence the first term is $R_{n+1}(i)$. The inductive step is complete, ■

The above lemma shows that formula (3) may be used instead of the E-algorithm in [1] to perform the extrapolations. Our main concern, however, is to compute convergence orders.

DEFINITION 1. Assume that n extrapolations have been performed with the correct convergence orders $k_1, \ldots, k_n$. For $i \geq n+3$ we define the experimental convergence orders $k_{n+1}(i)$ to be the solution for k to the following system of 3 equations in the unknowns A, c_{n+1} and k:

(5a) $$A - R_n(i-2) = c_{n+1} f_{n+1}(i-2, k) h(i-2)^k$$

(5b) $$A - R_n(i-1) = c_{n+1} f_{n+1}(i-1, k) h(i-1)^k$$

(5c) $$A - R_n(i) \quad = c_{n+1} f_{n+1}(i, k) h(i)^k.$$

These equations result from applying formula (4) to the triple $i-2$, $i-1$, i and truncating after the first term on the right hand side.

THEOREM 1. *Let $h(i)$ be of the form $h(i) = c/i$ for some constant c, and assume that n extrapolations have been performed using the correct convergence orders $k_1, k_2, \ldots, k_n$. Then*

(*a*) *The experimental convergence order $k_{n+1}(i)$, $i = n+3, \ldots$, is the solution to the following equation in the single unknown k:*

$$\alpha(i,k)\frac{\alpha(i-1,k)f_{n+1}(i-2,k)-f_{n+1}(i-1,k)}{\alpha(i,k)f_{n+1}(i-1,k)-f_{n+1}(i,k)} = \frac{R_n(i-1)-R_n(i-2)}{R_n(i)-R_n(i-1)} \tag{6}$$

If the right hand side is positive, equation (6) *always has a solution.*

(*b*) *If* (6) *has a unique solution for each positive right hand side, then*

$$k_{n+1}(i) - k_{n+1} = O(i^{k_{n+1}-k_{n+2}}).$$

PROOF. (a) Elimination of A and c_{n+1} in the three equations (5) leads directly to equation (6). It is convenient to denote the left hand side of (6) by LHS(i,k) and the right hand side by RHS(i). A simple inductive argument using equation (2) shows that

$$\text{LHS}(i,k) \to 0 \qquad \text{for } k \to -\infty$$
$$\text{LHS}(i,k) \to +\infty \quad \text{for } k \to +\infty.$$

This proves (*a*).

(b) Recall that (6) resulted from (4) applied to the triple $i-2$, $i-1$, i and truncated to the first term on the right hand side. We repeat this procedure without truncating the right hand side. The result is:

$$\alpha(i,k_{n+1})\frac{\alpha(i-1,k_{n+1})f_{n+1}(i-2,k_{n+1})-f_{n+1}(i-1,k_{n+1})}{\alpha(i,k_{n+1})f_{n+1}(i-1,k_{n+1})-f_{n+1}(i,k_{n+1})} = \tag{7}$$

$$= \frac{R_n(i-1)-R_n(i-2)+\sum_{j=n+2}^{\infty} c_j(f_{n+1}(i-1,k_j)h(i-1)^{k_j}-f_{n+1}(i-2,k_j)h(i-2)^{k_j})}{R_n(i)-R_n(i-1)+\sum_{j=n+2}^{\infty} c_j(f_{n+1}(i,k_j)h(i)^{k_j}-f_{n+1}(i-1,k_j)h(i-1)^{k_j})}$$

The left hand side of (7) is LHS(i,k_{n+1}). Denote the right side by RHS(i) + Δ(RHS). We claim the following two facts, which will complete the proof:

$$\Delta(\text{RHS}) = C_1 i^{k_{n+1}-k_{n+2}-1}(1 + O(i^{-1}))$$

$$\frac{\partial(\text{LHS}(k))}{\partial k} = C_2 i^{-1}(1 + O(i^{-1}))$$

The proof of these consists in tedious, but elementary calculations using lemma 1 (b). ■

We conjecture that equation (6) has exactly one solution, if the right hand side is positive. In fact, based on numerical computations we conjecture that the left hand side is monotonically increasing as a function of k, but we have no proof of this. Note that the right hand side in (6) is positive, if the error is dominated by a term of the form ch^k.

3. The algorithm.

The results of section 2 may be incorporated in an algorithm as follows:

Input $h(i)$ and $A(i) = A(h(i))$ values for $i = 1, .., i_{max}$.
Set $n := 0$.
Loop
  Set $n := n + 1$.
  Compute $k_n(i)$ for $i = n+2, \ldots, i_{max}$ from equation (6).
  Output $k_n(i)$ for $i = n+2, \ldots, i_{max}$.
  The user has the option to STOP here.
  Prompt for value of k_n to use in next extrapolation.
  Compute $R_n(i)$ for $i = n+1, .., i_{max}$ from equation (3).
  Output $R_n(i)$ for $i = n+1, \ldots, i_{max}$.
End of loop.

The crucial step is to enter the (preferably exact) value for k_n based on the experimental values $k_n(i)$. In typical cases the experimental values will confirm an expected value. This will establish the presence of a dominating term of the expected order in the error. Then we are on solid ground and may continue to extrapolate or stop with an accurate error estimate. If the order of the dominating term can not be easily recognized, we may try to enter an order, which is known to be present, although it is not dominating. After extrapolation a single term may now dominate, which will be disclosed by the $k(i)$-values in the next step. This trick is based on lemma 1 (a). It is demonstrated in the last section.

A last resort, which we do not recommend, is to enter for k_n the value of $k_n(i_{max})$, assumed to be the best approximation to k_n. From theorem 1(b) we know that this value will converge to k_n as h tends to zero. However, for a fixed set of h-values this procedure will only result in new numbers, not in new information.

The above description must be supplemented with an algorithm for solving eq. (6). Both the Newton method and the secant method are applicable. The Newton method will require a recursive calculation of $\partial f/\partial k$ parallel with f, so the secant method is most easily implemented. It is also faster by the standard rule for comparing these two methods, so in our implementation we chose the secant method. It is a consequence of the conjecture following theorem 1 that there is a unique simple solution. We have never observed convergence problems for positive right hand sides. The starting guesses for $k_n(i)$ are built into the

program as follows: $k_1(3)$: 1.0, $k_n(n+2)$: $k_{n-1}+1$, $k_n(i)$: $k_n(i-1)$ for $i \geq n+3$. This always seems to work very well.

Finally a note on the implementation. In our program the $f_n(i,k)$-values are computed recursively exactly as defined here. This is the most elegant and convenient way, and the storage requirements are reduced to a minimum ($h(i)$, $k_1 \ldots, k_{n-1}, R_{n-1}(i)$ and $R_n(i)$ are stored at level n). Of course, the recursion is slow. Computing time increases by approximately a factor 6 for each n-level. Much time can be saved by storing more intermediate values (such as the $f_j(i, k_{j-m})$ values), but the price is elegance and more storage. The results reported in each table in the next section have been achieved within 30 minutes on a MacIntosh and a few seconds on a SUN, using in both cases a compiled language, not an interpreter.

4. Two applications.

As an illustration the integral

$$\int_0^1 x^{1/2} dx$$

is approximated by the rectangular method:

$$A(i) = i^{-1} \sum_{j=1}^{i} [(j-1/2)/i]^{1/2}.$$

It is easy to see that the leading term in the error is of order $k_1 = 1.5$, after which we expect to find the orders $2, 4, 6, \ldots$ as usual for this method applied to smooth integrands. The final result is shown in table 1. Note that the expected exact orders are used, but only after they have been verified. It must be remembered that after each column has been computed and listed there is an interactive interface between the user and the program. The expected orders of convergence are successively confirmed, and high accuracy is easily achieved and documented.

The second application is more ambitious: The smallest eigenvalue for the L-shaped membrane over 3 unit squares is approximated by discretizing the Laplace operator. This eigenvalue has been computed by several methods, both general and ad hoc (see [2] and [3]). The value is

$$\lambda = 9.63972385 \pm 5 \times 10^{-8}.$$

For the present purpose the Laplace operator is discretized by the finite element method with piecewise bilinear functions over a rectangular uniform grid. Because of the singularity in the eigenfunction at the re-entrant corner the discretization error decreases quite slowly.

Table 2 shows the final output from the program. After seeing the approximated

Table 1. *Experimental convergence orders and extrapolations to order* 1.5, 2, 4, 6, *and* 8 *for the integral of* $\sqrt{x}$ *approximated by the rectangular method. (The lower half of the table should be placed to the right of the upper half.)*

$1/h$	$A(h)$	k_1	Rich. order 1.500000	k_2	Rich.order 2.000000	k_3
1	0.707106781187					
2	0.683012701892		0.669835212361			
3	0.676075333609	1.25	0.667788121978		0.666723611018	
4	0.672977397006	1.32	0.667236232289	1.91	0.666676794931	
5	0.671280085859	1.35	0.667010577045	1.96	0.666669762994	3.66
6	0.670231247722	1.37	0.666896671247	1.98	0.666667915350	3.83
7	0.669529353640	1.39	0.666831255604	1.98	0.666667263836	3.89
8	0.669032172130	1.40	0.666790249414	1.99	0.666666987394	3.93
9	0.668664662639	1.40	0.666762859863	1.99	0.666666854063	3.95
10	0.668383841146	1.41	0.666743662318	1.99	0.666666783399	3.96
11	0.668163481710	1.42	0.666729687585	2.00	0.666666743133	3.97
12	0.667986761516	1.42	0.666719199408	2.00	0.666666718833	3.97

$1/h$	Rich.order 4.000000	k_4	Rich. order 6.000000	k_5	Rich.order 8.000000	k_6
1						
2						
3						
4	0.666668406436					
5	0.666666865511		0.666666728988			
6	0.666666708147	5.27	0.666666672043		0.666666669060	
7	0.666666678748	5.60	0.666666667506	6.80	0.666666666839	
8	0.666666671027	5.74	0.666666666855	7.28	0.666666666688	8.28
9	0.666666668495	5.82	0.666666666720	7.51	0.666666666671	8.90
10	0.666666667523	5.86	0.666666666685	7.65	0.666666666668	9.24
11	0.666666667104	5.90	0.666666666674	7.95	0.666666666667	9.39
12	0.666666666906	5.90	0.666666666669	6.07	0.666666666667	9.78

k_1 – values the user is asked to provide the exact value. This cannot be done based on the approximations alone. We know from [5] that the singularity near the re-entrant corner is of the form

$$u(r,\varphi) = c_1 r^{2/3}\sin(2\varphi/3)+c_2 r^{4/3}\sin(4\varphi/3)+c_3 r^{8/3}\sin(8\varphi/3) + c_4 r^{8/3}\sin(2\varphi/3)+O(r^3) \tag{8}$$

where (r, φ) are polar coordinates with origin at the re-entrant corner and one boundary as axis. Hence we expect (1) to hold, but clearly more than one term is significant. We can now proceed in two ways: Based on (8) we can expect (or guess) the leading error term to be of order 4/3, which is the order provided in table 2. The next order strongly verifies the value 2 as expected (see below) confirming the first estimate of 4/3.

Table 2. *Four experimental convergence orders and extrapolations to order* 4/3, 2, 10/3, *and* 4 *for the smallest eigenvalue of the L-shaped membrane approximated using bilinear elements on a uniform rectangular grid.*

1/h	A(h)	k_1	Rich.order 1.333333	k_2	Rich.order 2.000000	k_3	Rich.order 3.333333	k_4	Rich.order 4.000000
20	9.67242140181								
22	9.66779116461		9.63362172180						
24	9.66416115912	1.67	9.63465201595		9.63980608526				
26	9.66125397094	1.66	9.63544136747	2.05	9.63978462610		9.63972122294		
28	9.65888353397	1.65	9.63605954085	2.04	9.63976990089	3.48	9.63972198172		9.63972388042
30	9.65692096548	1.64	9.63655273679	2.04	9.63975948822	3.47	9.63972248248	4.13	9.63972386062
32	9.65527451129	1.63	9.63695254399	2.03	9.63975193429	3.46	9.63972282413	4.10	9.63972384975
34	9.65387728904	1.62	9.63728115787	2.03	9.63974633210	3.46	9.63972306394	4.08	9.63972384377
36	9.65267953377	1.62	9.63755454592	2.03	9.63974209684	3.45	9.63972323639	4.07	9.63972384026
38	9.65164354233	1.61	9.63778443434	2.03	9.63973884052	3.45	9.63972336314	4.04	9.63972383866
40	9.65074028956	1.60	9.63797959230	2.02	9.63973629916	3.44	9.63972345803	4.03	9.63972383777
42	9.64994711007	1.60	9.63814668420	2.02	9.63973428917	3.44	9.63972353028	4.02	9.63972383744
44	9.64924607689	1.59	9.63829084833	2.02	9.63973268031	3.44	9.63972358612	4.00	9.63972383750
46	9.64862284626	1.59	9.63841609808	2.02	9.63973137855	3.43	9.63972362986	3.99	9.63972383767
48	9.64806582067	1.58	9.63852560513	2.02	9.63973031489	3.43	9.63972366453	3.97	9.63972383793
50	9.64756553325	1.58	9.63862190284	2.02	9.63972943799	3.43	9.63972369234	3.95	9.63972383834
52	9.64711418862	1.57	9.63870703451	2.02	9.63972870913	3.43	9.63972371483	3.96	9.63972383860
54	9.64670531620	1.57	9.63878266292	2.02	9.63972809874	3.43	9.63972373323	3.92	9.63972383902
56	9.64633350524	1.56	9.63885015232	2.01	9.63972758403	3.42	9.63972374838	3.93	9.63972383933
58	9.64599420028	1.56	9.63891063059	2.01	9.63972714720	3.42	9.63972376097	3.90	9.63972383968
60	9.64568354164	1.56	9.63896503670	2.01	9.63972677427	3.42	9.63972377150	3.92	9.63972383991

Table 3. *As table* 2, *but with extrapolation orders* 2, 4/3, 4, 10/3 *successively.*

$1/h$	$A(h)$	k_1	Rich.order 2.000000	k_2	Rich.order 1.333333	k_3	Rich.order 4.000000	k_4	Rich.order 3.333333
20	9.67242140181								
22	9.66779116461		9.64574241606						
24	9.66416115912	1.67	9.64506417369		9.63980608526				
26	9.66125397094	1.66	9.64450856705	1.29	9.63978462610		9.63973343943		
28	9.65888353397	1.65	9.64404635440	1.29	9.63976990089	3.48	9.63973110770		9.63972388042
30	9.65692096548	1.64	9.64365670951	1.30	9.63975948822	3.47	9.63972946047	3.30	9.63972386062
32	9.65527451129	1.63	9.64332444054	1.31	9.63975193429	3.46	9.63972826660	3.31	9.63972384975
34	9.65387728904	1.62	9.64303823159	1.31	9.63974633210	3.46	9.63972738196	3.31	9.63972384377
36	9.65267953377	1.62	9.64278949735	1.31	9.63974209684	3.45	9.63972671367	3.32	9.63972384026
38	9.65164354233	1.61	9.64257161730	1.31	9.63973884052	3.45	9.63972620025	3.32	9.63972383866
40	9.65074028956	1.60	9.64237941138	1.32	9.63973629916	3.44	9.63972579980	3.33	9.63972383777
42	9.64994711007	1.60	9.64220877362	1.32	9.63973428917	3.44	9.63972548328	3.33	9.63972383744
44	9.64924607689	1.59	9.64205641104	1.32	9.63973268031	3.44	9.63972523007	3.33	9.63972383750
46	9.64862284626	1.59	9.64191965458	1.32	9.63973137855	3.43	9.63972502530	3.34	9.63972383767
48	9.64806582067	1.58	9.64179631989	1.32	9.63973031489	3.43	9.63972485806	3.34	9.63972383793
50	9.64756553325	1.58	9.64168460353	1.32	9.63972943799	3.43	9.63972472025	3.34	9.63972383834
52	9.64711418862	1.57	9.64158300449	1.32	9.63972870913	3.43	9.63972460575	3.34	9.63972383860
54	9.64670531620	1.57	9.64149026415	1.32	9.63972809874	3.43	9.63972450990	3.35	9.63972383902
56	9.64633350524	1.56	9.64140531992	1.33	9.63972758403	3.42	9.63972442911	3.35	9.63972383933
58	9.64599420028	1.56	9.64132726898	1.33	9.63972714720	3.42	9.63972436057	3.35	9.63972383968
60	9.64568354164	1.56	9.64125533973	1.33	9.63972677427	3.42	9.63972430207	3.35	9.63972383991

The other way to proceed heavily uses the strength of Richardson extrapolation. If the order can be correctly estimated, then the corresponding term can be eliminated, even though it does not dominate all other error terms. In the present case with more than one dominating error term this implies that we can start by eliminating the term of our choice, and not necessarily the term of lowest order. With this version of the finite element method (bilinear elements) the leading error for smooth eigenfunctions is known to be of order 2 (the order inherent in the method). Hence we strongly expect the order 2 to be present in (1). We therefore start by extrapolating to order 2. The result of this is seen in table 3. It is now much easier to estimate the order 4/3 of the other (actually leading) term. The results of the second extrapolation in tables 2 and 3 confirm that the two extrapolations commute.

After eliminating the error terms of order 4/3 and 2 we are faced with the same problem again. More than one term dominate the remaining error, assuming still that (1) holds. We are on less solid ground theoretically when estimating the order of the third term, but it appears to be 10/3 (looking for integer multiples of 1/3). This is shown in table 2. It is again safer to eliminate first the term of order 4, known to be present in this finite element discretization (see table 3). After four rounds of extrapolation the roundoff error and lack of a dominating term make continuation questionable. Apparently (since $R_4(h)$ is not monotonic) the next two terms in the error are of opposite sign and of equal magnitude for h about this size. This also makes it difficult to estimate the error after the last extrapolation, but the accuracy obtained is comparable to the accuracy achieved by the *ad hoc* methods in [2] and [3]. In table 2 the error on R_3 ($h = 1/60$) is approximately 7×10^{-8}. The error on R_4 ($h = 1/60$) is considerably smaller, but cannot be accurately estimated.

Finally a remark on experiments with the order of extrapolation. Even with the modest h-values used here the procedure is extremely sensitive to the extrapolation order. If, in the above example, 1.30 is entered for k_1 (instead of 4/3), then the $k_2(i)$-values show no affinity to the value 2.0. This means that the algorithm is very good to disclose the nature of the convergence, but also that the extrapolation orders entered must be exact. Recall our recommendation against using $k_n(i_{\max})$ as extrapolation order. With care it may be used to estimate the error.

REFERENCES

1. C. Brezinski, *A general extrapolation algorithm*, Numer. Math. 35 (1980), p. 175–187.
2. G. Fix, *Higher-order Rayleigh-Ritz approximations*, J. Math. Mech. 18 (1969), p. 645–657.
3. L. Fox, P. Henrici, and C. Moler, *Approximations and bounds for eigenvalues of elliptic operators*, SIAM J. Numer. Anal. 4 (1967), p. 89–102.
4. T. Håvie, *Generalized Neville type extrapolation schemes*, BIT 19 (1979), p. 204–213.
5. R. S. Lehman, *Developments at an analytic corner of solutions of elliptic partial differential equations*, J. Math. Mech. 8 (1959), p. 727–760.
6. J. Wimp, *Sequence Transformations and their Applications*, Academic Press, 1981.