Published in Journal of Computational Physics vol 204, issue 1 , 20 March 2005, p. 292-301

Optimized point shifts and poles in the linear rational pseudospectral method for boundary value problems [☆]

Jean-Paul Berrut [a,*], Hans D. Mittelmann [b]

[a] *Département de Mathématiques, Université de Fribourg, CH-1700 Fribourg/Pérolles, Switzerland*
[b] *Department of Mathematics, Arizona State University, Tempe, Arizona 85287-1804, USA*

Received 22 March 2004; received in revised form 8 October 2004; accepted 11 October 2004

Abstract

Due to their rapid – often exponential – convergence as the number N of interpolation/collocation points is increased, polynomial pseudospectral methods are very efficient in solving smooth boundary value problems. However, when the solution displays boundary layers and/or interior fronts, this fast convergence will merely occur with very large N. To address this difficulty, we present a method which replaces the polynomial ansatz with a rational function r and considers the physical domain as the conformal map g of a computational domain. g shifts the interpolation points from their classical position in the computational domain to a problem-dependent position in the physical domain. Starting from a map by Bayliss and Turkel we have constructed a shift that can in principle accomodate an arbitrary number of fronts. Its parameters as well as the poles of r are optimized. Numerical results demonstrate how g best accomodates interior fronts while the poles also handle boundary layers.

MSC: 65L10; 65N35; 65N50
Keywords: Two-point boundary value problems; Linear rational collocation; Pole optimization; Mesh generation; Point shift optimization

1. Introduction

The present work is devoted to the numerical solution of linear two-point boundary value problems (BVPs):

[☆] Work partly supported by the Swiss National Science Foundation under Grant #2000-66754.01.

[*] Corresponding author. Tel.: +41 26 300 9196; fax: +41 26 300 9744.
E-mail addresses: jean-paul.berrut@unifr.ch (J.-P. Berrut), mittelmann@asu.edu (H.D. Mittelmann).

0021-9991
doi:10.1016/j.jcp.2004.10.009

$$u''(x)+p(x)u'(x)+q(x)u(x)=f(x),\qquad x\in(-1,1), \tag{1.1a}$$

$$u(-1)=u_\ell,\quad u(1)=u_r, \tag{1.1b}$$

where all arising functions belong to $C^\infty[-1,1]$ and where the boundary values u_ℓ and u_r are given real numbers.

Eq. (1.1) is merely a convenient example: the method to be introduced here could be applied – after modifications as they are demonstrated in [22] – to non-linear and higher dimensional problems (on parallelepipeds).

Our starting point is the polynomial collocation method, also called pseudospectral method when it has *spectral convergence*, i.e. when it converges faster than any power of the number h that characterizes the coarsity of the mesh.

Of the points (nodes) $x_0,x_1,\ldots,x_N$ which define the mesh we only assume for the moment that they are distinct and lie in the interval $[-1,1]$ under consideration. The unique polynomial of degree at most N that interpolates an arbitrary function u between the x_k's can be written in its barycentric form [16],

$$p(x)=\sum_{j=0}^{N}\frac{w_j}{x-x_j}u(x_j)\Bigg/\sum_{j=0}^{N}\frac{w_j}{x-x_j}, \tag{1.2}$$

where the weights w_j are given (up to a constant) by

$$w_j:=1\Bigg/\prod_{k\neq j}(x_j-x_i) \tag{1.3}$$

and thus depend only on the given mesh, not on the interpolated function f. The set of such interpolating polynomials therefore is a linear space $\mathscr{R}_N^{(w)}$, a basis of which is given by the Lagrange fundamental polynomials,

$$\ell_j^{(w)}(x):=\frac{w_j}{x-x_j}\Bigg/\sum_{k=0}^{N}\frac{w_k}{x-x_k},\quad j=0,1,\ldots,N.$$

It has the *Lagrange property*: $\ell_j^{(w)}$ interpolates every function that is 1 at x_j and 0 at all other nodes.

A version of the polynomial pseudospectral method now goes as follows:

- replace the solution u with an unknown polynomial,

$$\widetilde{u}(x)=\sum_{j=0}^{N}\widetilde{u}_j\ell_j^{(w)}(x), \tag{1.4}$$

interpolating the values in (1.1b) and the unknowns $\widetilde{u}_j=\widetilde{u}(x_j)$, $j=1,\ldots N-1$;
- insert $\widetilde{u}$ into (1.1a) and collocate at the $N-1$ points $x_1,\ldots,x_{N-1}$.

This yields the following system of linear equations for the $\widetilde{u}_j$:

$$\sum_{j=0}^{N}\widetilde{u}_j\ell_j^{(w)''}(x_i)+p(x_i)\sum_{j=0}^{N}\widetilde{u}_j\ell_j^{(w)'}(x_i)+q(x_i)\sum_{j=0}^{N}\widetilde{u}_j\ell_j^{(w)}(x_i)=f(x_i),\quad i=1,\ldots,N-1,\quad \widetilde{u}_0=u_r,\quad \widetilde{u}_N=u_\ell, \tag{1.5}$$

or $\mathbf{A}\widetilde{\mathbf{u}}=\mathbf{f}$ with $\mathbf{A}:=\mathbf{D}^{(2)}+\mathbf{P}\mathbf{D}^{(1)}+\mathbf{Q}$ and

$$\widetilde{\mathbf{u}}:=[\widetilde{u}_1,\widetilde{u}_2,\ldots,\widetilde{u}_{N-1}]^{\mathrm{T}},$$

$$\mathbf{D}^{(1)}=(D_{ij}^{(1)}),\quad D_{ij}^{(1)}:=\ell_j^{(w)'}(x_i),$$

$$\mathbf{D}^{(2)} = (D_{ij}^{(2)}), \quad D_{ij}^{(2)} := \ell_j^{(w)''}(x_i),$$

$$\mathbf{P} := \mathrm{diag}(p(x_i)), \quad \mathbf{Q} := \mathrm{diag}(q(x_i)),$$

$$\mathbf{f} := \left[f(x_i) - u_r\left(\ell_0^{(w)''}(x_i) + p(x_i)\ell_0^{(w)'}(x_i)\right) - u_\ell\left(\ell_N^{(w)''}(x_i) + p(x_i)\ell_N^{(w)'}(x_i)\right)\right]^{\mathrm{T}}, \quad i,j = 1,\ldots,N-1.$$

A good formula for the elements of the matrices $\mathbf{D}^{(1)}$ and $\mathbf{D}^{(2)}$ is given in (2.3) below. The approximate solution is then obtained by solving the system and introducing $\widetilde{\mathbf{u}}$ into (1.4).

The quality of the approximation of u by $\widetilde{u}$ depends upon that of an arbitrary function by its interpolating polynomial (in the Galerkin method, $\widetilde{u}$ is the best approximation to u in the so-called energy norm). That is why one should use nodes which accumulate near the boundary (see [12] for a recent discussion). Since the nodes should contain the boundary abscissae for an easy incorporation of the values u_ℓ and u_r, we have used Chebyshev points of the second kind $\cos j(\pi/N)$, which have the further advantage that the w_j's can be simplified to [21],

$$w_j = (-1)^j\delta_j, \quad \delta_j := \begin{cases} 1/2, & j = 0 \text{ or } j = N, \\ 1, & \text{otherwise.} \end{cases}$$

On the other hand, Markov's inequality unfortunately shows that, when u has a steep gradient in $[-1, 1]$, then $\widetilde{u}$ can be a good solution only if N is correspondingly large [8], and this for every set of nodes.

To move around this inherent limitation of the polynomials, we have turned to rational interpolants. Since classical rational interpolation suffers, at least for small N, from unattainable points and uncontrollable poles, we have suggested in [8] to obtain rational interpolants by optimally attaching poles to the interpolating polynomial (see [10] for an explanation of the nomenclature "attaching poles"). These interpolants avoid the two just cited pitfalls. Moreover, starting from the polynomial interpolant, they provide a sequence of approximations whose error decreases as the number P of attached poles grows.

In [9], we have introduced an algorithm for solving BVPs with such rational interpolants with attached poles. It consists in recursively performing two steps: the first collocates a linear rational ansatz with fixed, known poles, the second optimizes the location of the poles by minimizing the residual of the differential equation with respect to these poles. The method works well, as documented by the numerical results in [9]. However, the improvement in the error of the approximate solution $\widetilde{u}$ is often significantly less pronounced than the improvement in the residual error. We conjectured that this is due to the ill-conditioning of the evaluation of the residual by means of derivatives of interpolants between Chebyshev points.

We have then used a suggestion by Kosloff and Tal-Ezer to improve the condition of the differentiation of the interpolating polynomial by replacing the Chebyshev points with their images under a conformal map (the latter preserving the spectral convergence), i.e. by considering the physical space x as the image $g(y)$ of another coordinate space y, which will be called *reference space* and carry the Chebyshev points y_k. g is then chosen in such a way that the shifted nodes $x_k = g(y_k)$ are closer to equidistant than the y_k's. Applying this idea to our pole attaching method, we indeed obtained an improvement of the derivatives [10]; we noticed, however, that for functions with a steep gradient (front) in the center of the interval the improvement in the precision of the derivatives was due more to the motion of the center nodes towards the front than to better conditioning.

That observation lead us to consider more versatile changes of variable adapted not only to the steepness, but also to the location of the gradient, and able to accomodate not just one, but several fronts. In some examples, the corresponding method in [11] yields for the same N an improvement of up to 11 orders of magnitude in the approximation error and 9 in the error in the second derivative as compared with the interpolating polynomial (1.2). The aim of the present work is to apply these efficient changes of variable to the linear rational pseudospectral method presented in [9] for the solution of BVPs (1.1).

In Section 2 we recall the linear rational pseudospectral method with optimized poles introduced in [9]. Section 3 describes the modifications induced in the method by a point shift as described above. Section 4 presents the test problems we have experimented with and comments on the numerical results obtained when solving them with our method. The final section discusses another method which yields analytic solutions and concludes the paper with some general remarks.

2. The linear rational pseudospectral method with iteratively optimized poles

The first step in generalizing the pseudospectral method presented in the introduction consists in replacing the polynomial ansatz with a rational interpolant $\widetilde{u}$ with preassigned poles. It was noticed in [6] that this is easily realized in the barycentric setting. Let $\mathbf{z}: = [z_1, z_2, \ldots, z_P]^{\mathrm{T}}$ denote the vector of P poles one wishes to attach. Then the denominator of the interpolant will be proportional to $d(z): = (z - z_1)(z - z_2)\cdots(z - z_P)$ (if an interpolant of the given data with these poles exists, see [6]) and the barycentric representation of the interpolant will be (1.2) with the weights w_j replaced with [8],

$$b_j = w_j d_j, \quad d_j := d(x_j) = \prod_{k=1}^{P} (x_j - z_k). \tag{2.1}$$

We search for a solution in the linear space $\mathscr{R}_N^{(b)}$ of all rational interpolants with weights $\mathbf{b} = [b_1, b_2, \ldots, b_N]^{\mathrm{T}}$; the Lagrange fundamental rational functions,

$$\ell_j^{(b)}(x) := \frac{b_j}{x - x_j} \Bigg/ \sum_{k=0}^{N} \frac{b_k}{x - x_k}, \quad j = 0,1,\ldots,N, \tag{2.2}$$

constitute a basis of $\mathscr{R}_N^{(b)}$ with the Lagrange property,

$$\ell_j^{(b)}(x_i) = \delta_{ij}.$$

After replacement of $\ell_j^{(w)}$ with $\ell_j^{(b)}$, the pseudospectral method remains exactly the same as in the polynomial case and is just a generalization thereof. It boils down to the solution of the system (1.5) with $\mathbf{w}$ replaced with $\mathbf{b}$. The elements of the matrices $\mathbf{D}^{(1)}$ and $\mathbf{D}^{(2)}$ can be computed by the formulae,

$$\begin{aligned} D_{ij}^{(1)} &= \begin{cases} \dfrac{b_j/b_i}{x_i - x_j}, & i \neq j, \\ -\sum\limits_{k \neq i} D_{ik}^{(1)}, & i = j, \end{cases} \\ D_{ij}^{(2)} &= \begin{cases} 2D_{ij}^{(1)}\left(D_{ii}^{(1)} - \dfrac{1}{x_i - x_j}\right), & i \neq j, \\ -\sum\limits_{k \neq i} D_{ik}^{(2)}, & i = j, \end{cases} \end{aligned} \tag{2.3}$$

established in [3] and [2], see [12] for a direct proof.

This linear rational pseudospectral method, presented in [4] (see also [24]), delivers exponential convergence of $\widetilde{u}$ toward u when the solution of (1.1) is meromorphic with poles at $z_1, \ldots, z_P$.

In most cases, and in particular when the reason for extending the classical polynomial pseudospectral setting is the presence of steep gradients in the interior of $[-1, 1]$, one does not know the location of the poles a priori. We have therefore suggested in [9] to successively optimize them by applying a *sequence of linear rational collocation procedures*, interlaced with displacements of the poles to adjust their position to the given problem.

More precisely, our algorithm runs as follows: for $k = 1,2,\ldots$ repeat

Step 1.

Compute the approximate solution $\widetilde{\mathbf{u}}^{(k)} = [\widetilde{u}_1^{(k)}, \ldots, \widetilde{u}_{N-1}^{(k)}]^{\mathrm{T}}$ of (1.1) by the linear rational collocation method with $b_j = w_j d_j$, d_j from (2.1) ($d_j \equiv 1$ for $k = 1$). This modifies $\widetilde{\mathbf{u}}$ (for $k > 1$), but neither the poles $\mathbf{z}$ nor the weights $\mathbf{b}$.

Step 2.

For the $\widetilde{\mathbf{u}}^{(k)}$ inherited from Step 1), optimize the location of the poles $\mathbf{z}$ by minimizing,

$$J(\mathbf{z}) := \|r'' + pr' + qr - f\|_\infty,$$

the norm of the residual of the differential equation for the rational interpolant,

$$r(x) := \sum_{j=0}^{N} \frac{w_j \prod_{\ell=1}^{P}\left(1 - \frac{x_j}{z_\ell}\right)}{x - x_j} \widetilde{u}_j^{(k)} \Bigg/ \sum_{j=0}^{N} \frac{w_j \prod_{\ell=1}^{P}\left(1 - \frac{x_j}{z_\ell}\right)}{x - x_j}. \tag{2.4}$$

This changes $\mathbf{b}$ to yield a new interpolant to the $\widetilde{u}_j^{(k)}$.

When all z_ℓ are at infinity the representation (2.4) expresses that r coincides with the polynomial p_N, recall (1.2). This is the situation at the start of the method, i.e. for $k = 1$.

3. The rational pseudospectral method with a point shift

An alternate way of improving upon the polynomial pseudospectral method for problems with internal fronts is to perform a change of variable in the problem. Let $x = g(y)$, where g maps a domain $\mathscr{D}_1$ containing $[-1, 1]$ in y-space conformally onto a domain $\mathscr{D}_2$ containing $[-1, 1]$ in x-space. This transplants all functions of (1.1a) into y-space [17]: $U(y)\!: = u[g(y)] = u(x)$, $P(y)\!: = p(x)$, etc., and they all remain in $C^\infty[-1, 1]$. By means of the chain rule, the problem is also transplanted into y-space,

$$[y'(x)]^2 U''(y) + [y''(x) + P(y)y'(x)]U'(y) + Q(y)U(y) = F(y). \tag{3.1}$$

The reference nodes $y_0, \ldots, y_N$ are now chosen in the new coordinate space, and the nodes in the physical space are $x_k\!: = g(y_k)$. The idea then is to choose g in such a way that the x_k's accumulate in the vicinity of the fronts, in order to improve upon the approximation of u there.

We will use here the map g constructed in [11] as a generalization to several steep gradients of a map suggested by Bayliss and Turkel [5]. Instead of directly accumulating the x_k's at the fronts, it spreads out the corresponding y_k's there, i.e. it constructs a $g^{[-1]}$ that is very steep at the fronts (see pictures in [5] and [11]) and inverts it whenever g is needed. For Q fronts our $g^{[-1]}$ reads

$$y(x) = g^{[-1]}(x) = \mu + \frac{1}{\lambda} \sum_{q=1}^{Q} \arctan[\alpha_q(x - \beta_q)], \tag{3.2}$$

where λ and μ are the parameters needed for ensuring that $g^{[-1]}(-1) = -1$, $g^{[-1]}(1) = 1$,

$$\lambda = \frac{\gamma + \delta}{2}, \quad \mu = \frac{\gamma - \delta}{\gamma + \delta},$$

with

$$\gamma := \sum_{q=1}^{Q} \arctan[\alpha_q(1 + \beta_q)], \qquad \delta := \sum_{q=1}^{Q} \arctan[\alpha_q(1 - \beta_q)].$$

If $Q = 1$, one has [5]:

$$x = g(y) = \frac{1}{\alpha}\tan[\lambda(y-\mu)] + \beta;$$

for $Q > 1$, $g = x(y)$ is computed pointwise by solving the non-linear equation

$$\sum_{q=1}^{Q} \arctan[\alpha_q(x-\beta_q)] = \lambda(y-\mu);$$

for $Q = 2$, this boils down to the solution of a quadratic equation and the evaluation of an arctangent [11].

The problem (1.1) will now be solved in y-space, to which purpose we adapt the algorithm of Section 2 to the solution of (3.1). The transplanted linear rational ansatz with P prescribed poles $v_k = g^{[-1]}(z_k)$ replacing (1.4) reads

$$\widetilde{U}(y) = \sum_{j=0}^{N} \widetilde{u}_j L_j^{(b)}(y),$$

where $L_j^{(b)}(y)$ is the transplantation of $\ell_j^{(b)}(x)$ in (2.2). With an adequate g and for the same values $\widetilde{u}_j$, the concentration of points in the physical space results in a stretching of the function at the fronts, leading to smaller gradients in the reference space.

Step 1 of the algorithm remains basically the same. One must just change the matrix $\mathbf{A}$ of the linear system in such a way as to accomodate the change of variable, i.e. to

$$\mathbf{A} := \mathbf{G}_1^2\mathbf{D}^{(2)} + (\mathbf{G}_2 + \mathbf{G}_1\mathbf{P})\mathbf{D}^{(1)} + \mathbf{Q}, \tag{3.3}$$

where $\mathbf{D}_1$ and $\mathbf{D}_2$ are again the Chebyshev differentiation matrices (2.3) with poles in y-space, and where $\mathbf{G}_1$ and $\mathbf{G}_2$ denote the diagonal matrices of the derivatives of $g^{[-1]}$ at the nodes x_i,

$$\mathbf{G}_1 = \operatorname{diag}(y'(x_1),\ldots,y'(x_{N-1})), \qquad \mathbf{G}_2 = \operatorname{diag}(y''(x_1),\ldots,y''(x_{N-1})),$$

while $\mathbf{P}$ and $\mathbf{Q}$ contain the values of P and Q at the y_i's, i.e. of p and q at the x_i's.

In step 2, the residual will be minimized with respect to the parameters in the mapping as well as to the poles. For that purpose, define

$$\alpha := [\alpha_1,\alpha_2,\ldots,\alpha_Q]^{\mathrm{T}}, \qquad \beta := [\beta_1,\beta_2,\ldots,\beta_Q]^{\mathrm{T}},$$

and the new residual norm,

$$J(\mathbf{z},\alpha,\beta) := \|[y']^2R'' + [y'' + Py']R' + QR - F\|_\infty, \tag{3.4}$$

with

$$R(y) := \frac{\displaystyle\sum_{j=0}^{N} \frac{w_j\prod_{k=1}^{P}(y_j - v_k)}{y - y_j}\widetilde{u}_j}{\displaystyle\sum_{j=0}^{N} \frac{w_j\prod_{k=1}^{P}(y_j - v_k)}{y - y_j}} = \frac{\displaystyle\sum_{j=0}^{N} \frac{w_j\prod_{k=1}^{P}\left(g^{[-1]}(x_j) - g^{[-1]}(z_k)\right)}{g^{[-1]}(x) - g^{[-1]}(x_j)}\widetilde{u}_j}{\displaystyle\sum_{j=0}^{N} \frac{w_j\prod_{k=1}^{P}\left(g^{[-1]}(x_j) - g^{[-1]}(z_k)\right)}{g^{[-1]}(x) - g^{[-1]}(x_j)}} =: r(x).$$

In x-space, this is not a rational interpolant in the variable x any longer, rather a rational interpolant in $g^{[-1]}(x)$.

The derivatives, to be used at every step for the computation of $\mathbf{A}$ in (3.3) as well as of the residual (3.4), are obviously given by:

$$y'(x) = \frac{1}{\lambda}\sum_{q=1}^{Q}\frac{\alpha_q}{1+s_q^2},$$

$$y''(x) = -\frac{2}{\lambda}\sum_{q=1}^{Q}\frac{\alpha_q^2 s_q}{(1+s_q^2)^2}, \quad s_q := \alpha_q(x-\beta_q).$$

Their simplicity is the reason we did not write (3.1) wholly in the y-variable.

4. Numerical experience

Two problems will now demonstrate the efficiency of our algorithm. We choose their solutions by considering functions for which a version of the algorithm limited to approximation has proved impressively effective in [11]. Those consist of a sum of functions, each of whom displays a steep gradient: $u_1(x-a) = e^{*\frac{1}{x-a}}$, with an essential singularity at a, the error function u_2 $(x-b) = \text{erf}[\delta(x-b)]$ and $u_3(x-c) = \tanh[\eta(x-c)]$. We chose b and c in the solution interval $[-1,1]$ and $a \in \mathbb{R}$ outside the interval, but close enough to -1 to make for a steep gradient (boundary layer) at -1.

In the construction of his example, subsequently used in [1] and [9], Hemker [15] has taken advantage of the fact that $u_2''(x) = h_2(x)u_2'(x)$ with $h_2(x) = -\epsilon x$ and $\epsilon = 2\delta^2$. Since u_1 and u_3 abide by similar relations $u_i''(x) = h_i(x)u_i'(x)$ with $h_1(x) = -1/x^2 - 2/x$ and $h_3(x) = -2\eta\tanh(x)$, $u := \sum_{i=1}^{3} u_i$ satisfies,

$$u''(x) - h_\ell(x)u'(x) = \sum_{i\neq\ell}(h_i(x) - h_\ell(x))u_i'(x).$$

We chose $\ell = 2$, as it leads to the simplest among the three h_ℓ.

The details of the computations were in principle the same as in [9]. L_∞-norms $J(\mathbf{z},\alpha,\beta)$ in (3.4) were approximated by considering the 100 equally spaced points,

$$\widehat{y}_k = -\frac{5}{4} + \frac{k-1}{K-1}\frac{5}{2}, \qquad k = 1(1)K, \quad K = 100, \tag{4.1}$$

on the interval $[-5/4, 5/4]$ and computing the maximal absolute value at those $\widehat{y}_k$ lying in $[-1,1]$. The minimization of $J(\mathbf{z},\alpha,\beta)$ in Step 2) of the algorithm has been performed by the simulated annealing method of [14].

Example 1. Here, we took $u = u_1 + u_2$, i.e. a solution with a boundary layer and an interior front ($Q = 1$). The problem is:

$$u''(x) + \epsilon x u'(x) = k_1(x),$$

$$k_1(x) = \frac{e^{1/(x-a)}}{(x-a)^2}\left[\frac{1}{(x-a)^2} + \frac{2}{x-a} - \epsilon(x-b)\right],$$

$$u_\ell = e^{-1/(1+a)} + \text{erf}[-\delta(1+b)] \qquad u_r = e^{1/(1-a)} + \text{erf}[\delta(1-b)].$$

To allow for comparison with the direct approximation of the exact solution in [11], we have first used the same parameters, namely $a = -1.2$, $b = -0.5$, $\epsilon = 10^4$ and $N = 100$. Fig. 1 in [11] shows the solution u (up to a cosine term) for $\epsilon = 10^6$.

Results are displayed in Table 1. The first two columns give β and α, the location and the intensity of the gradient in the optimized variable change. Stars stand in cases no shift is made. When applicable, the third column gives the optimized poles in x-space (as opposed to [11], where we have forgotten to map them from y-space). The fourth column displays the optimal residual and the last the maximal error of the optimal solution evaluated at equidistant points as in (4.1), but in x-space and with $K = 1000$.

Table 1
Solution of Example 1 with $\epsilon = 10^4$, $a = -1.2$, $b = -0.5$ and $N = 100$

β	α	Poles	Residual	$\Vert\tilde{u} - u\Vert$
*	*		3.154e + 6	3.051e − 1
*	*	(−0.5072, ±3.343e − 2) (−0.5073, ±2.495e − 2) (−0.4863, ±2.275e − 2)	3.374e − 1	8.375e − 3
−0.5211	7.285		6.069e − 3	3.905e − 8
−0.5026	9.065	(−1.031, ±4.233e − 3)	1.113e − 5	7.565e − 11
−0.4978	8.561	(−1.036, ±4.784e − 3) (1.137, ±3.069e − 3)	1.266e − 6	1.141e − 11

Table 2
Solution of Example 1 with $\epsilon = 10^6$, $a = -1.2$, $b = -0.5$ and $N = 200$

β	α	Poles	Residual	$\Vert\tilde{u} - u\Vert$
*	*		5.696e + 10	2.674e + 1
*	*	(−0.5000, ±2.348e − 3) (−0.5016, ±2.728e − 2) (−0.5006, ±1.079e − 2)	4.575	1.313e − 1
−0.5005	34.84		2.135e + 1	1.385e − 6
−0.4998	49.76	(−1.004, ±4.153e − 4)	8.914e − 4	6.559e − 8
−0.4968	55.31	(−1.004, ±6.711e − 4) (1.293, ±6.712e − 3)	1.402e − 5	1.213e − 8

Without point shift (first two rows), the results are consistent with the corresponding ones in Table 4 of [9], just a little worse since about 20% less points are used. The method approximates the solution with an error only about ten times as large as the direct approximation of the exact solution in [11], a splendid performance achieved in all our examples. Without point shift the poles lie in close vicinity of the front, as is to be expected, whereas the presence of the shift enables them to handle the boundary stretches. This is decisive in improving the residual from a mere 6×10^{-3} to an excellent 10^{-6}.

As noticed in [11], the point shift allows for a good approximation of much steeper gradients than the poles alone. Table 2 demonstrates this for $\epsilon = 10^6$, a value the method without point shift of [9] cannot handle. However, the shift alone is not able to bring the residual under 21.4: the poles are required to match the boundary layers. This will be even more pronounced with a closer to -1 in u_1.

Example 2. To add another front, we now consider $u = \sum_{i=1}^{3} u_i$. The right-hand side of the differential equation becomes $k_2(x) = k_1(x) + \eta[\cosh(\eta(x - c))]^{-2}(\epsilon(x - b) - 2\eta \tanh(\eta(x - c)))$ and the quantity $\tanh(\eta(-1 - c))$, resp. $\tanh(\eta(1 - c))$, is to be added to the boundary values.

Table 3 displays results for $a = -1.2$, $b = 0.75$, $c = -0.5$, $\epsilon = 10^4$, $\eta = 100$ and $N = 200$. The corresponding solution u (again up to a cosine term) is given on Fig. 2 in [11]. The numbers are very similar to those of the first example, with a notable exception: 200 points are now too little for the shift to handle the steep gradients well enough: even with shift the poles are better used there than at the boundary.

The solutions are very stable: moving the poles from their optimal position or, as with $P = 4$, from the vicinity of 0.75 to that of -0.5, does not markedly deteriorate residual and error.

5. Further comments and conclusion

The method ours may best be compared with seems to be that of Mulholland, Huang and Sloan [20], which also determines a change of variable $x(y)$ before solving the transformed problem (3.1) with a

Table 3
Solution of Example 2 with $\epsilon = 10^4$, $\eta = 100$, $a = -1.2$, $b = 0.75$, $c = -0.5$ and $N = 200$

β_1	α_1	β_2	α_2	Poles	Residual	$\Vert\widetilde{u} - u\Vert$
*	*	*	*		1.375e + 6	1.040e − 1
*	*	*	*	(−0.5000, ±1.572e − 2) (0.7550, ±5.477e − 2) (0.7542, ±4.313e − 2)	8.703	2.395e − 2
−0.4900	16.09	0.7029	5.241		1.176e − 2	3.074e − 8
−0.4913	16.11	0.7356	4.542	(0.7472, ±5.399e − 2)	6.190e − 3	2.041e − 8
−0.4786	12.56	0.7323	4.768	(−0.5000, ±1.572e − 2) (0.7459, ±5.383e − 2)	7.485e − 4	5.519e − 9

classical pseudospectral method, and also yields an analytic solution (this in contrast with the very efficient method of Lee and Greengard [19]). The infinite complexity part of their method which, like ours, does not use any prior knowledge of the position of the fronts, is an adaptive finite difference method which simultaneously solves the problem equation and a non-linear equation based on an equidistribution principle for points $\bar{x}_k(\eta_k)$ with η_k equidistant in y-space. A smooth mesh is obtained by smoothing the matrices which approximate the non-linear equation.

In a second step, the $\bar{x}_k$ are piecewise linearly interpolated to obtain values x_k at the Chebyshev points y_k, which would in principle yield $x(y)$ as the interpolating polynomial of the x_k. However, the latter is non-monotonic due to oscillations and it must be filtered. Accounting for the boundary values then requires further treatment in Fourier space, which led the authors to choose N as a power of 2. Altogether, the method involves an equioscillation parameter, two smoothing parameters, and filtering parameters which are all chosen by trial and error, but could be optimized with a more sophisticated method such as that we have used. Values of several of these parameters are quite independent of the problems, and the computed solutions are not very sensitive to the others. In view of the cheapness of finite differences, their method seems more effective than ours. However, the many parameters render it much less transparent. Moreover, the chosen examples all have their steep gradients symmetric with respect to the center and no boundary layer with large values like ours – for the latter our pole attachment method could be incorporated into their algorithm. Moreover, Mulholland et al. do not tell how they computed the maximum error, so that our results cannot be compared with theirs, since the error in pseudospectral methods is often much smaller at the collocation points than in-between. Our numerical examples show that our method could also be made much cheaper by a gross optimization of the parameters α_q and β_q, what we have successfully tried with Hemker's original example.

As mentioned at the beginning, we have experimented only with Bayliss and Turkel's point shift. It would be interesting to try the mapping of [18] with two parameters (mentioned, but not tested, in [5]). An extension to problems on parallelepipeds in several dimensions is straightforward, the corresponding pole attachment method is briefly described in [9].

We have reached our goal of approximating by infinitely differentiable functions the solutions of BVPs that display boundary layers and fronts. In our examples, the inclusion of optimized changes of variable into the linear rational pseudospectral method with optimized poles leads to a superb improvement of typically 7 orders of magnitude. As we hoped, the better conditioning of the derivatives with point shift results in a better repercussion of the gain in residual in the L_∞-error than in [9]. The optimization part of the algorithm is expensive and makes it more suited to problems for which the time needed for computing the solution is not an issue. Notice, however, that iterative methods for solving the collocation system in step 1 often converge much faster with an analytic than with a piecewise ansatz, see [13] for the polynomial method and [7] for the linear rational pseudospectral method with Kosloff and Tal-Ezer's point shift (the method in [7] may be improved with GMRES and a Bayliss and Turkel shift [23]).

Acknowledgements

The authors express their gratitude to a referee for his detailed comments on the manuscript.

References

[1] U.M. Ascher, R.M. Mattheij, R.D. Russell, Numerical Solution of Boundary Value Problems for Ordinary Differential Equations, Prenctice Hall, Englewood Cliffs, 1988.

[2] R. Baltensperger, Improving the accuracy of the matrix differentiation method for arbitrary collocation points, Appl. Numer. Math. 33 (2000) 143–149.

[3] R. Baltensperger, J.-P. Berrut, The errors in calculating the pseudospectral differentiation matrices for Čebyšev–Gauss–Lobatto points, Comput. Math. Appl. 37 (1999) 41–48; Errata 38 (1999) 119.

[4] R. Baltensperger, J.-P. Berrut, Y. Dubey, The linear rational pseudospectral method with preassigned poles, Numer. Algorithms 33 (2003) 53–63.

[5] A. Bayliss, E. Turkel, Mappings and accuracy for Chebyshev pseudo-spectral approximations, J. Comput. Phys. 101 (1992) 349–359.

[6] J.-P. Berrut, The barycentric weights of rational interpolation with prescribed poles, J. Comput. Appl. Math. 86 (1997) 45–52.

[7] J.-P. Berrut, R. Baltensperger, The linear rational collocation method for boundary value problems, BIT 41 (2001) 868–879.

[8] J.-P. Berrut, H. Mittelmann, Rational interpolation through the optimal attachment of poles to the interpolating polynomial, Numer. Algorithms 23 (2000) 315–328.

[9] J.-P. Berrut, H.D. Mittelmann, The linear rational collocation method with iteratively optimized poles for two-point boundary value problems, SIAM J. Sci. Comput. 23 (2001) 961–975.

[10] J.-P. Berrut, H.D. Mittelmann, Point shifts in rational interpolation with optimized denominator, in: J. Levesley, I.J. Anderson, J.C. Mason (Eds.), Algorithms for Approximation IV, Proceedings of the 2001 International Symposium, The University of Huddersfield, 2002, pp. 420–427.

[11] J.-P. Berrut, H.D. Mittelmann, Adaptive point shifts in rational approximation with optimized denominator, J. Comput. Appl. Math. 164–165 (2004) 81–92.

[12] J.-P. Berrut, L.N. Trefethen, Barycentric Lagrange interpolation, SIAM Rev 46 (2004) 501–517.

[13] C. Canuto, A. Quarteroni, Preconditioned minimal residual methods for Chebyshev spectral calculations, J. Comput. Phys. 60 (1985) 315–337.

[14] A. Corana, M. Marchesi, C. Martini, S. Ridella, Minimizing multimodal functions of continuous variables with the simulated annealing algorithm, ACM Trans. Math. Software 13 (1987) 262–280.

[15] P.W. Hemker, A Numerical Study of Stiff Two-point Boundary Problems, Math. Centre Tracts No. 80, Math. Centrum, Amsterdam, 1977.

[16] P. Henrici, Essentials of Numerical Analysis, Wiley, New York, 1982.

[17] P. Henrici, Applied and Computational Complex Analysis, vol. 3, Wiley, New York, 1986.

[18] D. Kosloff, H. Tal-Ezer, A modified Chebyshev pseudospectral method with an $\mathcal{O}(N^{-1})$ time step restriction, J. Comput. Phys. 104 (1993) 457–469.

[19] J.-Y. Lee, L. Greengard, A fast adaptive numerical method for stiff two-point boundary value problems, SIAM J. Sci. Comput. 18 (1997) 403–429.

[20] L.S. Mulholland, W.Z. Huang, D.M. Sloan, Pseudospectral solution of near-singular problems using numerical coordinate transformations based on adaptivity, SIAM J. Sci. Comput. 19 (1998) 1261–1289.

[21] H.E. Salzer, Lagrangian interpolation at the Chebyshev points $x_{n,v} = \cos(v\pi/n)$, $v = 0(1)n$; some unnoted advantages, Comput. J. 15 (1972) 156–159.

[22] L.N. Trefethen, Spectral Methods in MATLAB, SIAM, Philadelphia, 2000.

[23] S. Vernay, Résolution par Orthomin et GMRES des systèmes issus d'une méthode pseudospectrale pour problèmes aux limites master's thesis, University of Fribourg, Switzerland, 2002.

[24] J.C. Weideman, Spectral methods based on non-classical orthogonal polynomials, in: W. Gautschi, et al. (Eds.), Applications and Computation of Orthogonal Polynomials, ISNM 131, Birkhäuser, 1999, pp. 239–252.