

*Communications in
Applied
Mathematics and
Computational
Science*

OPTIMAL STABILITY POLYNOMIALS FOR
NUMERICAL INTEGRATION OF INITIAL VALUE
PROBLEMS

DAVID I. KETCHESON AND ARON J. AHMADIA

vol. 7 no. 2 2012

OPTIMAL STABILITY POLYNOMIALS FOR NUMERICAL INTEGRATION OF INITIAL VALUE PROBLEMS

DAVID I. KETCHESON AND ARON J. AHMADIA

We consider the problem of finding optimally stable polynomial approximations to the exponential for application to one-step integration of initial value ordinary and partial differential equations. The objective is to find the largest stable step size and corresponding method for a given problem when the spectrum of the initial value problem is known. The problem is expressed in terms of a general least deviation feasibility problem. Its solution is obtained by a new fast, accurate, and robust algorithm based on convex optimization techniques. Global convergence of the algorithm is proven in the case that the order of approximation is one and in the case that the spectrum encloses a starlike region. Examples demonstrate the effectiveness of the proposed algorithm even when these conditions are not satisfied.

1. Stability of Runge–Kutta methods

Runge–Kutta methods are among the most widely used types of numerical integrators for solving initial value ordinary and partial differential equations. The time step size should be taken as large as possible since the cost of solving an initial value problem (IVP) up to a fixed final time is proportional to the number of steps that must be taken. In practical computation, the time step is often limited by stability and accuracy constraints. Either accuracy, stability, or both may be limiting factors for a given problem; see [24, Section 7.5] for a discussion. The linear stability and accuracy of an explicit Runge–Kutta method are characterized completely by the so-called stability polynomial of the method, which in turn dictates the acceptable step size [6; 12]. In this work we present an approach for constructing a stability polynomial that allows the largest absolutely stable step size for a given problem.

The problem of finding optimal stability polynomials is of fundamental importance in the numerical solution of initial value problems, and its solution or approximation has been studied by many authors for several decades. Indeed, it is closely related to the problem of finding polynomials of least deviation, which goes back to the work of Chebyshev. A nice review of much of the early work on

MSC2010: primary 65L06, 65M20; secondary 90C26.

Keywords: absolute stability, initial value problems, Runge–Kutta methods.

Runge–Kutta stability regions can be found in [44]. The most-studied cases are those where the eigenvalues lie on the negative real axis [1; 3; 2; 4; 38; 23; 25; 27; 33; 35; 36; 43; 8], on the imaginary axis [21; 20; 22; 26; 43; 32; 46], or in a disk of the form $|z + w| \leq w$ [15; 46]. Many results and optimal polynomials, both exact and numerical, are available for these cases. Some authors have considered the solution of [Problem 1](#) for other spectra corresponding to PDE semidiscretizations [17; 31; 38; 26; 28; 39].

Two very recent works serve to illustrate both the progress that has been made in solving these problems with nonlinear programming, and the challenges that remain. In [39], optimal schemes are sought for integration of discontinuous Galerkin discretizations of wave equations, where the optimality criteria considered include both accuracy and stability measures. The approach used there is based on sequential quadratic programming (local optimization) with many initial guesses. The authors consider methods of at most fourth order and situations with $s - p \leq 4$ “because the cost of the optimization procedure becomes prohibitive for a higher number of free parameters.” In [28], optimally stable polynomials are found for certain spectra of interest for $2 \leq p \leq 4$ and (in a remarkable feat!) s as large as 14. The new methods obtained achieve a 40–50% improvement in efficiency for discontinuous Galerkin integration of the 3D Maxwell equations. The optimization approach employed therein is again a direct search algorithm that does not guarantee a globally optimal solution but “typically converges... within a few minutes”. However, it was apparently unable to find solutions for $s > 14$ or $p > 4$. The method we present in the next section can rapidly find solutions for significantly larger values of s , p , and is provably globally convergent under certain assumptions (introduced in [Section 2](#)).

In the remainder of this section, we review the stability concepts for Runge–Kutta methods and formulate the stability optimization problem. Our optimization approach, described in [Section 2](#), is based on reformulating the stability optimization problem in terms of a sequence of convex subproblems and using bisection. We examine the theoretical properties of the proposed algorithm and prove its global convergence for two important cases.

A key element of our optimization algorithm is the use of numerical convex optimization techniques. We avoid a poorly conditioned numerical formulation by posing the problem in terms of a polynomial basis that is well-conditioned when sampled over a particular region of the complex plane. These numerical considerations, which become particularly important when the number of stages of the method is allowed to be very large, are discussed in [Section 3](#).

In [Section 4](#) we apply our algorithm to several examples of complex spectra. Cases where optimal results are known provide verification of the algorithm, and many new or improved results are provided.

Determination of the stability polynomial is only half of the puzzle of designing optimal explicit Runge–Kutta methods. The other half is the determination of the Butcher coefficients. While simply finding methods with a desired stability polynomial is straightforward, many additional challenges arise in that context; for instance, additional nonlinear order conditions, internal stability, storage, and embedded error estimators. All of these concerns can be dealt with using the software package RK-opt [19], which also includes the algorithm presented herein. The development of full Runge–Kutta methods based on optimal stability polynomials, using the present approach and additional tools from RK-opt, is conducted in [30].

1.1. The stability polynomial. A linear, constant-coefficient initial value problem takes the form

$$u'(t) = Lu, \quad u(0) = u_0, \quad (1)$$

where $u(t) : \mathbb{R} \rightarrow \mathbb{R}^N$ and $L \in \mathbb{R}^{N \times N}$. When applied to the linear IVP (1), any Runge–Kutta method reduces to an iteration of the form

$$u_n = R(hL)u_{n-1}, \quad (2)$$

where h is the step size and u_n is a numerical approximation to $u(nh)$. The *stability function* $R(z)$ depends only on the coefficients of the Runge–Kutta method; see [9, Section 4.3], [6], [12]. In general, the stability function of an s -stage explicit Runge–Kutta method is a polynomial of degree s

$$R(z) = \sum_{j=0}^s a_j z^j. \quad (3)$$

Recall that the exact solution of (1) is $u(t) = \exp(tL)u_0$. Thus, if the method is accurate to order p , the stability polynomial must be identical to the exponential function up to terms of at least order p :

$$a_j = \frac{1}{j!} \quad \text{for } 0 \leq j \leq p. \quad (4)$$

1.2. Absolute stability. The stability polynomial governs the local propagation of errors, since any perturbation to the solution will be multiplied by $R(z)$ at each subsequent step. The propagation of errors thus depends on $\|R(hL)\|$, which leads us to define the *absolute stability region*

$$S = \{z \in \mathbb{C} : |R(z)| \leq 1\}. \quad (5)$$

For example, the stability region of the classical fourth-order method is shown in Figure 1(b).

given an initial value problem (1), let $\Lambda \in \mathbb{C}$ denote the spectrum of the matrix L . We say the iteration (2) is absolutely stable if

$$h\lambda \in S \quad \text{for all } \lambda \in \Lambda. \quad (6)$$

Condition (6) implies that u_n remains bounded for all n . More importantly, (6) is a necessary condition for stable propagation of errors. Thus the maximum stable step size is given by

$$h_{\text{stable}} = \max\{h \geq 0 : |R(h\lambda)| \leq 1 \text{ for } \lambda \in \Lambda\}. \quad (7)$$

Note that for nonnormal L , it may be important to consider the pseudospectrum rather than the spectrum; see Section 4.3.

As an example, consider the advection equation

$$\frac{\partial}{\partial t} u(x, t) + \frac{\partial}{\partial x} u(x, t) = 0, \quad x \in (0, M),$$

discretized in space by first-order upwind differencing with spatial mesh size Δx

$$U'_i(t) = -\frac{U_i(t) - U_{i-1}(t)}{\Delta x}, \quad 0 \leq i \leq N,$$

with periodic boundary condition $U_0(t) = U_N(t)$. This is a linear IVP (1) with L a circulant bidiagonal matrix. The eigenvalues of L are plotted in Figure 1(a) for $\Delta x = 1$, $N = M = 20$. To integrate this system with the classical fourth-order Runge–Kutta method, the time step size must be taken small enough that the scaled spectrum $\{h\lambda_i\}$ lies inside the stability region. Figure 1(c) shows the (maximally) scaled spectrum superimposed on the stability region.

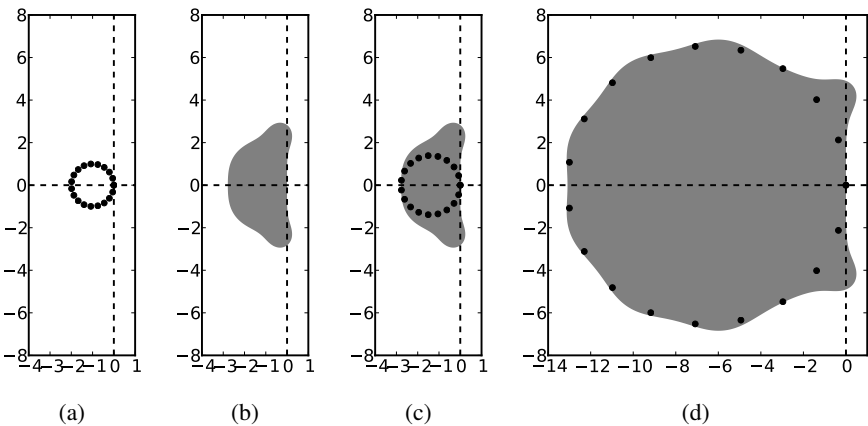


Figure 1. (a) Spectrum of first-order upwind difference matrix using $N = 20$ points in space. (b) Stability region of the classical fourth-order Runge–Kutta method. (c) Scaled spectrum $h\lambda$ with $h = 1.39$. (d) Scaled spectrum $h\lambda$ for optimal ten-stage method with $h = 6.54$.

The motivation for this work is that a larger stable step size can be obtained by using a Runge–Kutta method with a larger region of absolute stability. Figure 1(d) shows the stability region of an optimized ten-stage Runge–Kutta method of order four that allows a much larger step size. The ten-stage method was obtained using the technique that is the focus of this work. Since the cost of taking one step is typically proportional to the number of stages s , we can compare the efficiency of methods with different numbers of stages by considering the *effective step size* h/s . Normalizing in this manner, it turns out that the ten-stage method is nearly twice as fast as the traditional four-stage method.

1.3. Design of optimal stability polynomials. We now consider the problem of choosing a stability polynomial so as to maximize the step size under which given stability constraints are satisfied. The objective function $f(x)$ is simply the step size h . The stability conditions yield nonlinear inequality constraints. Typically one also wishes to impose a minimal order of accuracy. The monomial basis representation (3) of $R(z)$ is then convenient because the first $p + 1$ coefficients $\{a_0, a_1, \dots, a_p\}$ of the stability polynomial are simply taken to satisfy the order conditions (4). As a result, the space of decision variables has dimension $s + 1 - p$, and is comprised of the coefficients $\{a_{p+1}, a_{p+2}, \dots, a_s\}$, as well as the step size h . Then the problem can be written as

Problem 1 (stability optimization). Given $\Lambda \subset \mathbb{C}$, order p , and number of stages s ,

$$\begin{aligned} & \underset{a_{p+1}, a_{p+2}, \dots, a_s, h}{\text{maximize}} && h \\ & \text{subject to} && |R(h\lambda)| - 1 \leq 0 \quad \text{for all } \lambda \in \Lambda. \end{aligned}$$

We use H_{opt} to denote the solution of Problem 1 (the optimal step size) and R_{opt} to denote the optimal polynomial.

The set Λ may be finite, corresponding to a finite-dimensional ODE system or PDE semidiscretization, or infinite (but bounded), corresponding to a PDE or perhaps its semidiscretization in the limit of infinitesimal mesh width. In the latter case, Problem 1 is a semi-infinite program (SIP). In Section 4 we approach this by using a finite discretization of Λ ; for a discussion of this and other approaches to semi-infinite programming, see [13].

2. An efficient algorithm for design of globally optimal stability polynomials

Evidently, finding the global solution of Problem 1 is in general quite challenging. Although sophisticated optimization algorithms such as the interior point method can guarantee polynomial time solutions to convex problems, and convex programming techniques are valuable in efficiently seeking minima, the stability constraints in Problem 1 are nonconvex. As a result, suboptimal local minima may exist and

certificates of optimality may require either approximations to the solution of the problem or greater than polynomial time. See [5] for an overview of convex optimization programming techniques, and [29] for an introduction to approximation algorithms and local search heuristics for nonconvex problems.

2.1. Reformulation in terms of the least deviation problem. The primary theoretical advance leading to the new results in this paper is a reformulation of [Problem 1](#). Note that [Problem 1](#) is nonconvex for $s > 2$ since $R(h\lambda)$ is a nonconvex function in h .

Instead of asking for the maximum stable step size we now ask, for a given step size h , how small the maximum modulus of $R(h\lambda)$ can be. This leads to a generalization of the classical least deviation problem.

Problem 2 (least deviation). Given $\Lambda \subset \mathbb{C}$, $h \in \mathbb{R}^+$ and $p, s \in \mathbb{N}$

$$\underset{a_{p+1}, a_{p+2}, \dots, a_s}{\text{minimize}} \quad \max_{\lambda \in \Lambda} (|R(h\lambda)| - 1).$$

We denote the solution of [Problem 2](#) by $r_{p,s}(h, \Lambda)$, or simply $r(h, \Lambda)$. Note that $|R(z)|$ is convex with respect to a_j , since $R(z)$ is linear in the a_j . Therefore, [Problem 2](#) is convex. [Problem 1](#) can be formulated in terms of [Problem 2](#):

Problem 3 (reformulation of [Problem 1](#)). Given $\Lambda \subset \mathbb{C}$, and $p, s \in \mathbb{N}$,

$$\begin{aligned} &\underset{a_{p+1}, a_{p+2}, \dots, a_s}{\text{maximize}} \quad h \\ &\text{subject to} \quad r_{p,s}(h, \Lambda) \leq 0. \end{aligned}$$

2.2. Solution via bisection. Although [Problem 3](#) is not known to be convex, it is an optimization in a single variable. It is natural then to apply a bisection approach, as outlined in [Algorithm 1](#).

Algorithm 1 (simple bisection).

Select h_{\max} (see [Section 2.3](#))

$h_{\min} = 0$

while $h_{\max} - h_{\min} > \epsilon$ **do**

$h = (h_{\max} + h_{\min})/2$

Solve [Problem 2](#)

if $r_{p,s}(h, \Lambda) \leq 0$ **then**

$h_{\min} = h$

else

$h_{\max} = h$

end if

end while

return $H_\epsilon = h_{\min}$

Since $r(0, \Lambda) = -1$ and $\lim_{h \rightarrow \infty} r(h, \Lambda) = +\infty$, then there exists $h_{\max} > 0$ such that $r(h, \Lambda) = 0$ for some $h \in [h_{\min}, h_{\max}]$. Global convergence of the algorithm is assured only if the following condition holds:

$$r_{p,s}(h_0, \Lambda) = 0 \implies r_{p,s}(h, \Lambda) \leq 0 \text{ for all } 0 \leq h \leq h_0. \quad (8)$$

We now consider conditions under which condition (8) can be established. We have the following important case.

Theorem 1 (global convergence when $p = 1$). *Let $p = 1$, $\Lambda \subset \mathbb{C}$ and $s \geq 1$. Take h_{\max} large enough so that $r(h_{\max}, \Lambda) > 0$. Let H_{opt} denote the solution of Problem 1. Then the output of Algorithm 1 satisfies*

$$\lim_{\epsilon \rightarrow 0} H_{\epsilon} = H_{\text{opt}}.$$

Proof. Since $r(0, \Lambda) = 0 < r(h_{\max}, \Lambda)$ and $r(h, \Lambda)$ is continuous in h , it is sufficient to prove that condition (8) holds. We have $|R_{\text{opt}}(H_{\text{opt}}\lambda)| \leq 1$ for all $\lambda \in \Lambda$. We will show that there exists $R_{\mu}(z) = \sum_{j=0}^s a_j(\mu)z^j$ such that $a_0 = a_1 = 1$ and

$$|R_{\mu}(\mu H_{\text{opt}}\lambda)| \leq 1 \text{ for all } \lambda \in \Lambda \text{ and } 0 \leq \mu \leq 1.$$

Let \hat{a}_j be the coefficients of the optimal polynomial:

$$R_{\text{opt}}(z) = 1 + z + \sum_{j=2}^s \hat{a}_j z^j,$$

and set

$$a_j(\mu) = \mu^{1-j} \hat{a}_j.$$

Then

$$\begin{aligned} R_{\mu}(\mu H_{\text{opt}}\lambda) &= 1 + \mu H_{\text{opt}}\lambda + \sum_{j=2}^s \mu^{1-j} \hat{a}_j \mu^j H_{\text{opt}}^j \lambda^j = 1 + \mu \left(\sum_{j=1}^s \hat{a}_j H_{\text{opt}}^j \lambda^j \right) \\ &= 1 + \mu (R_{\text{opt}}(H_{\text{opt}}\lambda) - 1), \end{aligned}$$

where we have defined $\hat{a}_1 = 1$. Define $g_{\lambda}(\mu) = R_{\mu}(\mu H_{\text{opt}}\lambda)$. Then $g_{\lambda}(\mu)$ is linear in μ and has the property that, for $\lambda \in \Lambda$, $|g_{\lambda}(0)| = 1$ and $|g_{\lambda}(1)| \leq 1$ (by the definition of H_{opt} , R_{opt}). Thus by convexity $|g_{\lambda}(\mu)| \leq 1$ for $0 \leq \mu \leq 1$. \square

For $p > 1$, condition (8) does not necessarily hold. For example, take $s = p = 4$; then the stability polynomial (3) is uniquely defined as the degree-four Taylor approximation of the exponential, corresponding to the classical fourth-order Runge–Kutta method that we saw in the introduction. Its stability region is plotted in Figure 1(b). Taking, e.g., $\lambda = 0.21 + 2.3i$, one finds $|R(\lambda)| < 1$ but $|R(\lambda/2)| > 1$. Although this example shows that Algorithm 1 might formally fail, it concerns only the trivial case $s = p$ in which there is only one possible choice of stability

polynomial. We have searched without success for a situation with $s > p$ for which condition (8) is violated.

2.3. Selection of h_{\max} . The bisection algorithm requires as input an initial h_{\max} such that $r(h_{\max}, \Lambda) > 0$. Theoretical values can be obtained using the classical upper bound of $2s^2/x$ if Λ encloses a negative real interval $[x, 0]$, or using the upper bound given in [34] if Λ encloses an ellipse in the left half-plane. Alternatively, one could start with a guess and successively double it until $r(h_{\max}, \Lambda) > 0$ is satisfied. Since evaluation of $r(h, \Lambda)$ is typically quite fast, finding a tight initial h_{\max} is not an essential concern.

2.4. Convergence for starlike regions. In many important applications the relevant set Λ is an infinite set; for instance, if we wish to design a method for some PDE semidiscretization that will be stable for any spatial discretization size. In this case, Problem 1 is a semi-infinite program (SIP) as it involves infinitely many constraints. Furthermore, Λ is often a closed curve whose interior is starlike with respect to the origin; for example, upwind semidiscretizations of hyperbolic PDEs have this property. Recall that a region S is starlike if $t \in S$ implies $\mu t \in S$ for all $0 \leq \mu \leq 1$.

Lemma 1. *Let $\Lambda \in \mathbb{C}$ be a closed curve passing through the origin and enclosing a starlike region. Let $r(h, \Lambda)$ denote the solution of Problem 2. Then condition (8) holds.*

Proof. Let Λ be as stated in the lemma. Suppose $r(h_0, \Lambda) = 0$ for some $h_0 > 0$; then there exists $R(z)$ such that $|R(h\lambda)| \leq 1$ for all $\lambda \in \Lambda$. According to the maximum principle, the stability region of $R(z)$ must contain the region enclosed by Λ . Choose h such that $0 \leq h \leq h_0$; then $h\Lambda$ lies in the region enclosed by Λ , so $|R(h\lambda)| \leq 1$ for $\lambda \in \Lambda$. \square

The proof of Lemma 1 relies crucially on Λ being an infinite set, but in practice we numerically solve Problem 2 with only finitely many constraints. To this end we introduce a sequence of discretizations Λ_n with the following properties:

1. $\Lambda_n \subset \Lambda$.
2. $n_1 \leq n_2 \implies \Lambda_{n_1} \subset \Lambda_{n_2}$.
3. $\lim_{n \rightarrow \infty} \Lambda_n = \Lambda$.
4. $\lim_{n \rightarrow \infty} v_n = 0$ where v_n denotes the maximum distance from a point in Λ to the set Λ_n :

$$v_n = \max_{\gamma \in \Lambda} \min_{\lambda \in \Lambda_n} |\gamma - \lambda|.$$

For instance, Λ_n can be taken as an equispaced (in terms of arc-length, say) sampling of n points.

By modifying [Algorithm 1](#), we can approximate the solution of the semi-infinite programming problem for starlike regions to arbitrary accuracy. At each step we solve [Problem 2](#) with Λ_n replacing Λ . The key to the modified algorithm is to only increase h_{\min} after obtaining a certificate of feasibility. This is done by using the Lipschitz constant of $R(z)$ over a domain including $h\Lambda$ (denoted by $L(R, h\Lambda)$) to ensure that $|R(h\Lambda)| \leq 1$. The modified algorithm is stated as [Algorithm 2](#).

Algorithm 2 (bisection for SIP).

```

 $h_{\min} = 0$ 
 $h_{\max} = 2s^2 / \max |\lambda|$ 
 $n = n_0$ 
while  $h_{\max} - h_{\min} > \epsilon$  do
     $h = (h_{\max} + h_{\min})/2$  ▷ Bisect
    Solve Problem 2
    if  $r(h, \Lambda_n) < 0$  and  $v_n < -2r/L(R, h\Lambda)$  then ▷ Certifies that  $r(h, \Lambda) < 0$ 
         $h_{\min} = h$ 
    else if  $r(h, \Lambda_n) > 0$  then ▷ Certifies that  $r(h, \Lambda) > 0$ 
         $h_{\max} = h$ 
    else ▷  $r(h, \Lambda_n) \leq 0$ 
         $n \leftarrow 2n$  ▷ Reduce the discretization spacing
    end if
end while
return  $H_\epsilon = h_{\min}$ 

```

The following lemma, which characterizes the behavior of [Algorithm 2](#), holds whether or not the interior of Λ is starlike.

Lemma 2. *Let $h^{[k]}$ denote the value of h after k iterations of the loop in [Algorithm 2](#). Then either*

- [Algorithm 2](#) terminates after a finite time with outputs satisfying $r(h_{\min}, \Lambda) \leq 0$, $r(h_{\max}, \Lambda) > 0$; or
- there exists $j < \infty$ such that $r(h^{[j]}, \Lambda) = 0$ and $h^{[k]} = h^{[j]}$ for all $j \geq k$.

Proof. First suppose that $r(h^{[j]}, \Lambda) = 0$ for some j . Then neither feasibility nor infeasibility can be certified for this value of h , so $h^{[k]} = h^{[j]}$ for all $j \geq k$.

On the other hand, suppose that $r(h^{[k]}, \Lambda) \neq 0$ for all k . The algorithm will terminate as long as, for each $h^{[k]}$, either feasibility or infeasibility can be certified for large enough n . If $r(h^{[k]}, \Lambda) > 0$, then necessarily $r(h^{[k]}, \Lambda_n) > 0$ for large enough n , so infeasibility will be certified. We will show that if $r(h^{[k]}, \Lambda) < 0$, then for large enough n the condition

$$v_n < -2r/L(R, h\Lambda) \tag{9}$$

must be satisfied. Since $r(h, \Lambda_n) \leq r(h, \Lambda)$ is bounded away from zero and $\lim_{n \rightarrow \infty} v_n = 0$, (9) must be satisfied for large enough n unless the Lipschitz constant $L(R, h\Lambda)$ is unbounded (with respect to n) for some fixed h . Suppose by way of contradiction that this is the case, and let $R^{[1]}, R^{[2]}, \dots$ denote the corresponding sequence of optimal polynomials. Then the norm of the vector of coefficients $a_j^{[i]}$ appearing in $R^{[i]}$ must also grow without bound as $i \rightarrow \infty$. By Lemma 3, this implies that $|R^{[i]}(z)|$ is unbounded except for at most s points $z \in \mathbb{C}$. But this contradicts the condition $|R^{[i]}(h\lambda)| \leq 1$ for $\lambda \in \Lambda_n$ when $n > s$. Thus, for large enough n we must have $v_n < -2r/L(R, h\Lambda)$. \square

In practical application, $r(h, \Lambda) = 0$ will not be detected, due to numerical errors; see Section 3.1. For this reason, in the next theorem we simply assume that Algorithm 2 terminates. We also require the following technical result.

Lemma 3. *Let $R^{[1]}, R^{[2]}, \dots$ be a sequence of polynomials of degree at most s ($s \in \mathbb{N}$ fixed) and denote the coefficients of $R^{[i]}$ by $a_j^{[i]} \in \mathbb{C}$ ($i \in \mathbb{N}$, $0 \leq j \leq s$):*

$$R^{[i]}(z) = \sum_{j=0}^s a_j^{[i]} z^j, \quad z \in \mathbb{C}.$$

Further, let $a^{[i]} := (a_0^{[i]}, a_1^{[i]}, \dots, a_s^{[i]})^T$ and suppose that the sequence $\|a^{[i]}\|$ is unbounded in \mathbb{R} . Then the sequences $R^{[i]}(z)$ are unbounded for all but at most s points $z \in \mathbb{C}$.

Proof. Suppose to the contrary there are $s + 1$ distinct complex numbers, say, z_0, z_1, \dots, z_s such that the vectors $r_i := (R^{[i]}(z_0), R^{[i]}(z_1), \dots, R^{[i]}(z_s))^T$ ($i \in \mathbb{N}$) are bounded in \mathbb{C}^{s+1} . Let V denote the $(s + 1) \times (s + 1)$ Vandermonde matrix whose k^{th} row ($0 \leq k \leq s + 1$) is $(1, z_k, z_k^2, \dots, z_k^s)$. Then V is invertible and we have $a^{[i]} = V^{-1}r_i$ ($i \in \mathbb{N}$), so if $\|\cdot\|$ denotes the induced matrix norm, then

$$\|a^{[i]}\| = \|V^{-1}r_i\| \leq \|V^{-1}\| \|r_i\|.$$

But, by assumption, the right side is bounded, whereas the left side is not. \square

Theorem 2 (global convergence for strictly starlike regions). *Let Λ be a closed curve that encloses a region that is starlike with respect to the origin. Suppose that Algorithm 2 terminates for all small enough ϵ , and let H_ϵ denote the value returned by Algorithm 2 for a given ϵ . Let H_{opt} denote the solution of Problem 1. Then*

$$\lim_{\epsilon \rightarrow 0} H_\epsilon = H_{\text{opt}}.$$

Proof. Due to the assumptions and Lemma 2, we have that $r(h_{\min}, \Lambda) < 0 < r(h_{\max}, \Lambda)$. Then Lemma 1 implies that $h_{\min} < H_{\text{opt}} < h_{\max}$. Noting that also $h_{\max} - h_{\min} < \epsilon$, the result follows. \square

Despite the lack of a general global convergence proof, [Algorithm 1](#) works very well in practice even for general Λ when $p > 1$. In all cases we have tested and for which the true H_{opt} is known (see [Section 4](#)), [Algorithm 1](#) appears to converge to the globally optimal solution. Furthermore, [Algorithm 1](#) is very fast. For these reasons, we consider the (much slower) [Algorithm 2](#) to be of primarily theoretical interest, and we base our practical implementation on [Algorithm 1](#).

3. Numerical implementation

We have made a prototype implementation of [Algorithm 1](#) in Matlab. The implementation relies heavily on the CVX package [[11](#); [10](#)], a Matlab-based modeling system for convex optimization, which in turn relies on the interior-point solvers SeDuMi [[37](#)] and SDPT3 [[42](#)]. The least deviation problem ([Problem 2](#)) can be succinctly stated in four lines of the CVX problem language, and for many cases is solved in under a second by either of the core solvers.

Our implementation re-attempts failed solves (see [Section 3.2](#)) with the alternate interfaced solver. In our test cases, we observed that the SDPT3 interior-point solver was slower, but more robust than SeDuMi. Consequently, our prototype implementation uses SDPT3 by default.

Using the resulting implementation, we were able to successfully solve problems to within 0.1% accuracy or better with scaled eigenvalue magnitudes $|h\lambda|$ as large as 4000. As an example, comparing with results of [[4](#)] for spectra on the real axis with $p = 3, s = 27$, our results are accurate to 6 significant digits.

3.1. Feasibility threshold. In practice, CVX often returns a small positive objective ($r \approx 10^{-7}$) for values of h that are just feasible. Hence the bisection step is accepted if $r < \epsilon$ where $\epsilon \ll 1$. The results are generally insensitive (up to the first few digits) to the choice of ϵ over a large range of values; we have used $\epsilon = 10^{-7}$ for all results in this work. The accuracy that can be achieved is eventually limited by the need to choose a suitable value ϵ .

3.2. Conditioning and change of basis. Unfortunately, for large values of $h\lambda$, the numerical solution of [Problem 2](#) becomes difficult due to ill-conditioning of the constraint matrix. Observe from ([3](#)) that the constrained quantities $R(h\lambda)$ are related to the decision variables a_j through multiplication by a Vandermonde matrix. Vandermonde matrices are known to be ill-conditioned for most choices of abscissas. For very large $h\lambda$, the resulting CVX problem cannot be reliably solved by either of the core solvers.

A first approach to reducing the condition number of the constraint matrix is to rescale the monomial basis. We have found that a more robust approach for many types of spectra can be obtained by choosing a basis that is approximately

orthogonal over the given spectrum $\{\Lambda\}$. Thus we seek a solution of the form

$$R(z) = \sum_{j=0}^s a_j Q_j(z), \quad \text{where } Q_j(z) = \sum_{k=0}^j b_{jk} z^k. \quad (10)$$

Here $Q_j(z)$ is a degree- j polynomial chosen to give a well-conditioned constraint matrix. The drawback of not using the monomial basis is that the dimension of the problem is $s + 1$ (rather than $s + 1 - p$) and we must now impose the order conditions explicitly:

$$\sum_{j=0}^s a_j b_{jk} = \frac{1}{k!} \quad \text{for } k = 0, 1, \dots, p. \quad (11)$$

Consequently, using a nonmonomial basis increases the number of design variables in the problem and introduces an equality constraint matrix $B \in \mathbb{R}^{p \times s}$ that is relatively small (when $p \ll s$), but usually very poorly conditioned. However, it can dramatically improve the conditioning of the inequality constraints.

The choice of the basis $Q_j(z)$ is a challenging problem in general. In the special case of a negative real spectrum, an obvious choice is the Chebyshev polynomials (of the first kind) T_j , shifted and scaled to the domain $[hx, 0]$ where $x = \min_{\lambda \in \Lambda} \operatorname{Re}(\lambda)$, via an affine map:

$$Q_j(z) = T_j \left(1 + \frac{2z}{hx} \right). \quad (12)$$

The motivation for using this basis is that $|Q_j(h\lambda)| \leq 1$ for all $\lambda \in [hx, 0]$. This basis is also suggested by the fact that $Q_j(z)$ is the optimal stability polynomial in terms of negative real axis inclusion for $p = 1, s = j$. In [Section 4](#), we will see that this choice of basis works well for more general spectra when the largest magnitude eigenvalues lie near the negative real axis.

As an example, we consider a spectrum of 3200 equally spaced values λ in the interval $[-1, 0]$. [Figure 2](#) shows the relative error as well as the condition number of the $3200 \times s$ inequality constraint matrix obtained by using the monomial [\(3\)](#) and Chebyshev [\(12\)](#) bases for $p = 1$ and s ranging from 2 through 50. The optimal objective value is $h = 2s^2$, and the condition number of the inequality constraint matrix is measured for the feasibility problem at this value. The condition number of the monomial basis scales exponentially, while the condition number of the Chebyshev basis constraint matrix has a weak linear dependence on s . Typically, the solver is accurate until the condition number reaches about 10^{16} . This supports the hypothesis that it is the conditioning of the inequality constraint matrix that leads to failure of the solver. The Chebyshev basis keeps the condition number small and yields accurate answers even for very large values of h .

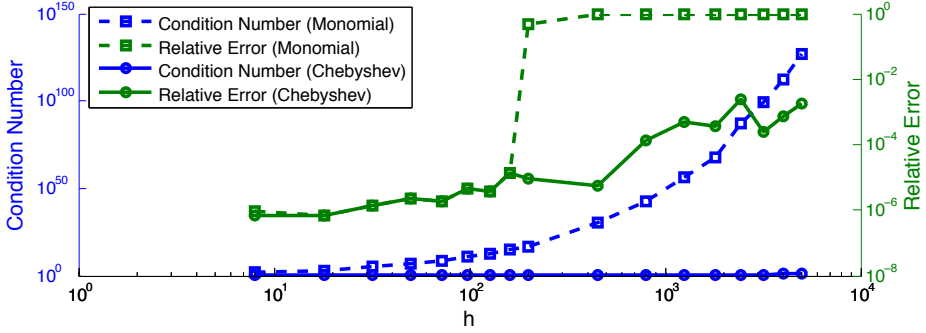


Figure 2. Condition number of principal constraint matrix and relative solution accuracy versus optimal step size. The points along a given curve correspond to different choices of s . The values plotted correspond to $s = 2, 3, \dots, 9, 10, 15, 20, \dots, 45, 50$ and a spectrum of 3200 equally spaced values in the interval $[-1, 0]$. The constraint matrix is formed using the optimal value $h = 2s^2$. The Chebyshev basis keeps the condition number small and yields accurate answers even for very large values of h .

4. Examples

We now demonstrate the effectiveness of our algorithm by applying it to determine optimally stable polynomials (i.e., solve [Problem 1](#)) for various types of spectra. As stated above, we use [Algorithm 1](#) for its simplicity, speed, and effectiveness. When Λ corresponds to an infinite set, we approximate it by a fine discretization.

4.1. Verification. In this section, we apply our algorithm to some well-studied cases with known exact or approximate results in order to verify its accuracy and correctness. In addition to the real axis, imaginary axis, and disk cases below, we have successfully recovered the results of [\[28\]](#). Our algorithm succeeds in finding the globally optimal solution in every case for which it is known, except in some cases of extremely large step sizes for which the underlying solvers (SDPT3 and SeDuMi) eventually fail.

Negative real axis inclusion. Here we consider the largest h such that $[-h, 0] \in S$ by taking $\Lambda = [-1, 0]$. This is the most heavily studied case in the literature, as it applies to the semidiscretization of parabolic PDEs and a large increase of H_{opt} is possible when s is increased (see, e.g., [\[33; 44; 27; 4; 35\]](#)). For first-order accurate methods ($p = 1$), the optimal polynomials are just shifted Chebyshev polynomials, and the optimal timestep is $H_{\text{opt}} = 2s^2$. Many special analytical and numerical techniques have been developed for this case; the most powerful seems to be that of Bogatyrev [\[4\]](#).

We apply our algorithm to a discretization of Λ (using 6400 evenly spaced points) and using the shifted and scaled Chebyshev basis [\(12\)](#). Results for up to $s = 40$ are shown in [Table 1](#) (note that we list H_{opt}/s^2 for easy comparison,

Stages	H_{opt}/s^2				
	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 10$
1	2.000				
2	2.000	0.500			
3	2.000	0.696	0.279		
4	2.000	0.753	0.377	0.174	
5	2.000	0.778	0.421	0.242	
6	2.000	0.792	0.446	0.277	
7	2.000	0.800	0.460	0.298	
8	2.000	0.805	0.470	0.311	
9	2.000	0.809	0.476	0.321	
10	2.000	0.811	0.481	0.327	0.051
15	2.000	0.817	0.492	0.343	0.089
20	2.000	0.819	0.496	0.349	0.120
25	2.000	0.820	0.498	0.352	0.125
30	2.001	0.821	0.499	0.353	0.129
35	2.000	0.821	0.499	0.354	0.132
40	2.000	0.821	0.500	0.355	0.132

Table 1. Scaled size of real axis interval inclusion for optimized methods.

since H_{opt} is approximately proportional to s^2 in this case). We include results for $p = 10$ to demonstrate the algorithm’s ability to handle high-order methods. For $p = 1$ and 2, the values computed here match those available in the literature [43]. Most of the values for $p = 3, 4$ and 10 are new results. Figure 3 shows some examples of stability regions for optimal methods. As observed in the literature, it seems that H_{opt}/s^2 tends to a constant (that depends only on p) as s increases. For large values of s , some results in the table have an error of about 10^{-3} due to inaccuracies in the numerical results provided by the interior point solvers.

Imaginary axis inclusion. Next we consider the largest h such that $[-ih, ih] \in S$ by taking $\Lambda = xi, x \in [-1, 1]$. Optimal polynomials for imaginary axis inclusion

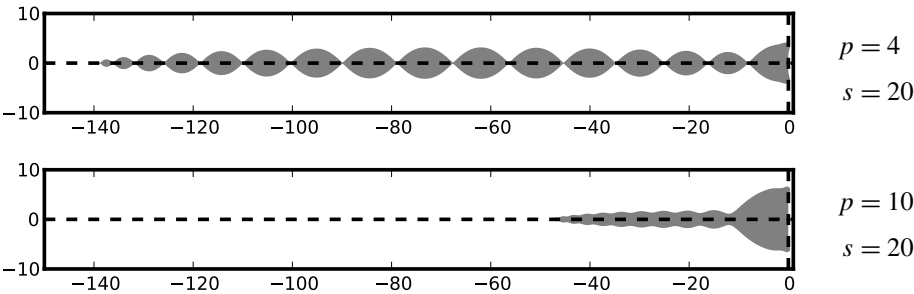


Figure 3. Stability regions of some optimal methods for real axis inclusion.

Stages	H_{opt}/s			
	$p = 1$	$p = 2$	$p = 3$	$p = 4$
2	0.500			
3	0.667	0.667	0.577	
4	0.750	0.708	0.708	0.707
5	0.800	0.800	0.783	0.693
6	0.833	0.817	0.815	0.816
7	0.857	0.857	0.849	0.813
8	0.875	0.866	0.866	0.866
9	0.889	0.889	0.884	0.864
10	0.900	0.895	0.895	0.894
15	0.933	0.933	0.932	0.925
20	0.950	0.949	0.949	0.949
25	0.960	0.960	0.959	0.957
30	0.967	0.966	0.966	0.966
35	0.971	0.971	0.971	0.970
40	0.975	0.975	0.975	0.975
45	0.978	0.978	0.978	0.977
50	0.980	0.980	0.980	0.980

Table 2. Scaled size of imaginary axis inclusion for optimized methods.

have also been studied by many authors, and a number of exact results are known or conjectured [43; 46; 20; 21; 22; 44]. We again approximate the problem, taking $N = 3200$ evenly spaced values in the interval $[0, i]$ (note that stability regions are necessarily symmetric about the real axis since $R(z)$ has real coefficients). We use a “rotated” Chebyshev basis defined by

$$Q_j(z) = i^j T_j \left(\frac{iz}{hx} \right),$$

where $x = \max_i (|\operatorname{Im}(\lambda_i)|)$. Like the Chebyshev basis for the negative real axis, this basis dramatically improves the robustness of the algorithm for imaginary spectra. Table 2 shows the optimal effective step sizes. In agreement with [43; 21], we find $H = s - 1$ for $p = 1$ (all s) and for $p = 2$ (s odd). We also find $H = s - 1$ for $p = 1$ and s even, which was conjectured in [46] and confirmed in [44]. We find

$$H_{\text{opt}} = \sqrt{s(s - 2)}$$

for $p = 2$ and s even, strongly suggesting that the polynomials given in [20] are optimal for these cases; on the other hand, our results show that those polynomials, while third order accurate, are not optimal for $p = 3$ and s odd. Figure 4 shows some examples of stability regions for optimal methods.

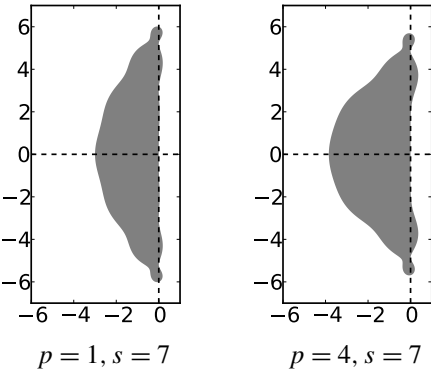


Figure 4. Stability regions of some optimal methods for imaginary axis inclusion.

Disk inclusion. In the literature, attention has been paid to stability regions that include the disk

$$D(h) = \{z : |1 + z/h| \leq 1\}, \tag{13}$$

for the largest possible h . As far as we know, the optimal result for $p = 1$ ($H_{\text{opt}} = s$) was first proved in [15]. The optimal result for $p = 2$ ($H_{\text{opt}} = s - 1$) was first proved in [46]. Both results have been unwittingly rediscovered by later authors. For $p > 2$, no exact results are available.

We use the basis

$$Q_j(z) = \left(1 + \frac{z}{h}\right)^j.$$

Note that $Q_j(z)$ is the optimal polynomial for the case $s = j$, $p = 1$. This basis can also be motivated by recalling that Vandermonde matrices are perfectly conditioned when the points involved are equally spaced on the unit circle. Our basis can be obtained by taking the monomial basis and applying an affine transformation that shifts the unit circle to the disk (13). This basis greatly improves the robustness of

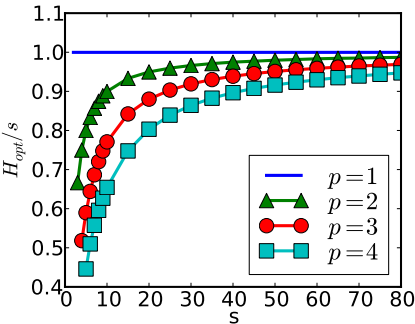


Figure 5. Relative size of largest disk that can be included in the stability region (scaled by the number of stages).

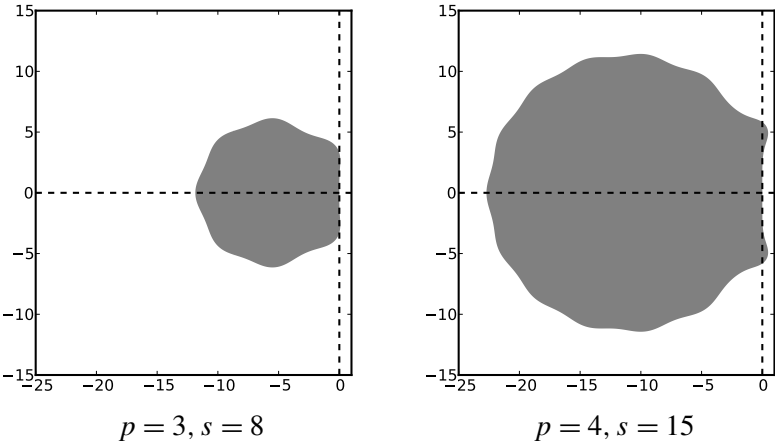


Figure 6. Stability regions of some optimal methods for disk inclusion.

the algorithm for this particular spectrum. We show results for $p \leq 4$ in Figure 5. For $p = 3$ and $s = 5, 6$, our results give a small improvement over those of [16]. Some examples of optimal stability regions are plotted in Figure 6.

4.2. Spectrum with a gap. We now demonstrate the effectiveness of our method for more general spectra. First we consider the case of a dissipative problem with two time scales, one much faster than the other. This type of problem was the motivation for the development of projective integrators in [8]. Following the ideas outlined there we consider

$$\Lambda = \{z : |z| = 1, \Re(z) \leq 0\} \cup \{z : |z - \alpha| = 1\}. \tag{14}$$

We take $\alpha = 20$ and use the shifted and scaled Chebyshev basis (12). Results are shown in Figures 7 and 8. A dramatic increase in efficiency is achieved by adding a few extra stages.

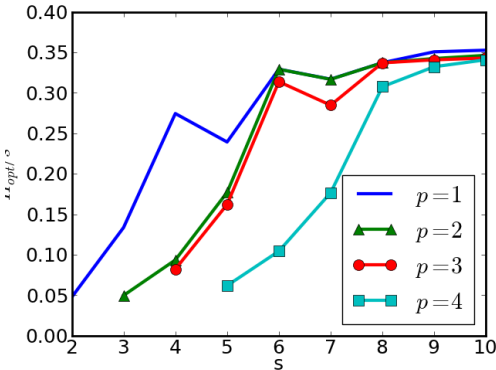


Figure 7. Optimal effective step size for spectrum with a gap (14) with $\alpha = 20$.

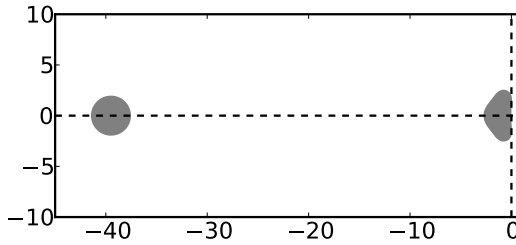


Figure 8. Optimal stability region for $p = 1, s = 6, \alpha = 20$ (stable step size ≈ 1.975).

4.3. Legendre pseudospectral discretization. Next we consider a system obtained from semidiscretization of the advection equation on the interval $[-1, 1]$ with homogeneous Dirichlet boundary condition:

$$u_t = u_x, \quad u(t, x = 1) = 0.$$

The semidiscretization is based on pseudospectral collocation at points given by the zeros of the Legendre polynomials; we take $N = 50$ points. The semidiscrete system takes the form (1), where L is the Legendre differentiation matrix, whose eigenvalues are shown in Figure 9(a). We compute an optimally stable polynomial based on the spectrum of the matrix, taking $s = 7$ and $p = 1$. The stability region of the resulting method is plotted in Figure 9(c). Using an appropriate step size, all the scaled eigenvalues of L lie in the stability region. However, this method is unstable in practice for any positive step size; Figure 9(e) shows an example of a computed solution after three steps, where the initial condition is a Gaussian. The resulting instability is nonmodal, meaning that it does not correspond to any of the eigenvectors of L (compare [41, Figure 31.2]).

This discretization is now well-known as an example of nonnormality [41, Chapters 30–32]. Due to the nonnormality, it is necessary to consider pseudospectra in order to design an appropriate integration scheme. The ϵ -pseudospectrum (see [41]) is the set

$$\{z \in \mathbb{C} : \|(zI - L)^{-1}\|_2 > 1/\epsilon\}.$$

The ϵ -pseudospectrum (for $\epsilon = 2$) is shown with the eigenvalues in Figure 9(b); note that the pseudospectrum includes small islands around the isolated eigenvalues, though they are not visible at the scale plotted. The instability observed above occurs because the stability region does not contain an interval on the imaginary axis about the origin, whereas the pseudospectrum includes such an interval.

We now compute an optimally stable integrator based on the 2-pseudospectrum. This pseudospectrum is computed using an approach proposed in [40, Section 20], with sampling on a fine grid. In order to reduce the number of constraints and speed up the solution, we compute the convex hull of the resulting set and apply

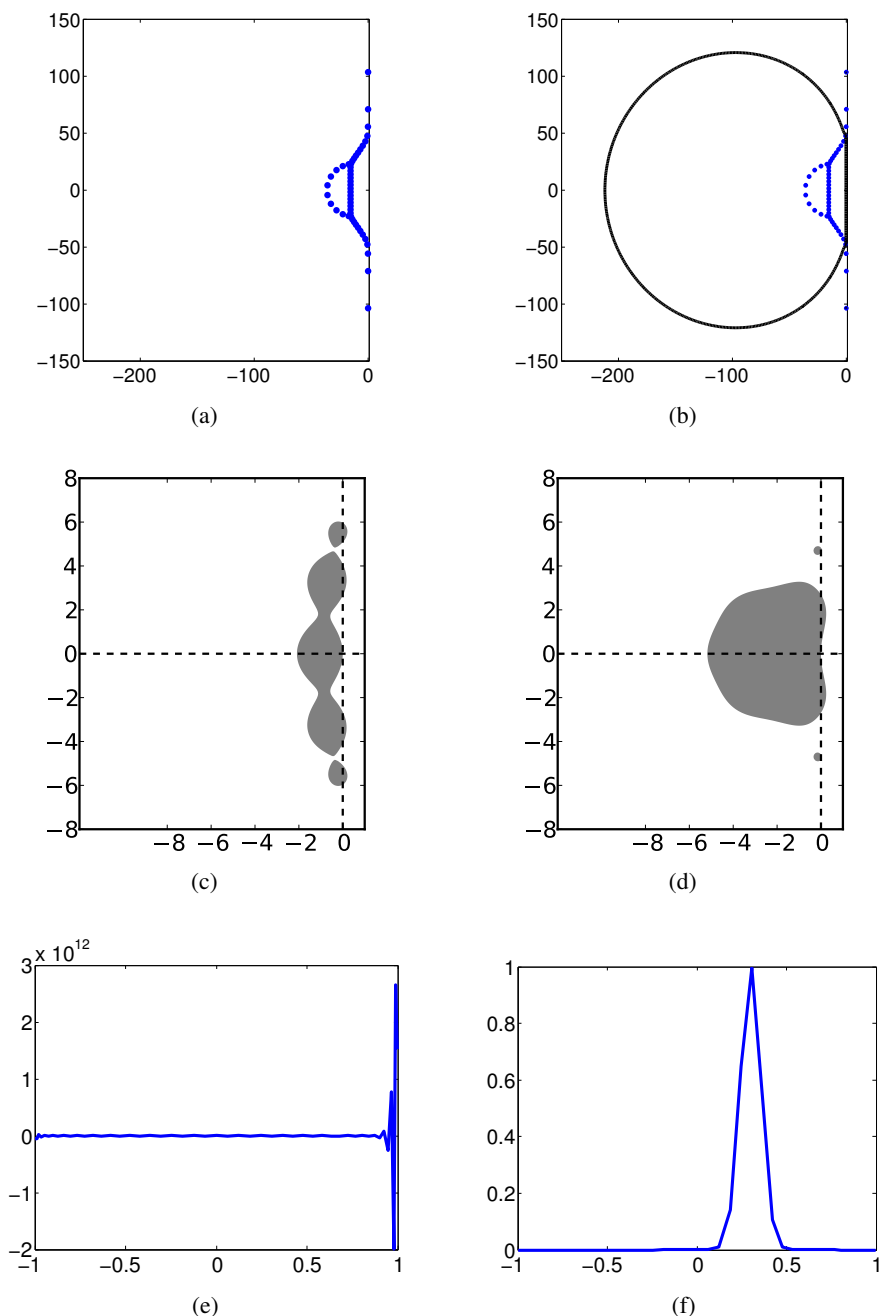


Figure 9. Results for the Legendre differentiation matrix with $N = 50$. Top row: eigenvalues (a) and eigenvalues with pseudospectrum (b). The boundary of the 2-pseudospectrum is plotted. Middle row: Optimized stability region based on eigenvalues (c) and on the pseudospectrum (d). Bottom row: Solution computed with method based on spectrum (e) and with method based on pseudospectrum (f).

our algorithm. The resulting stability region is shown in Figure 9(d). It is remarkably well adapted; notice the two isolated roots that ensure stability of the modes corresponding to the extremal imaginary eigenvalues. We have verified that this method produces a stable solution, in agreement with theory (see Chapter 32 of [41]); Figure 9(f) shows an example of a solution computed with this method. The initial Gaussian pulse advects to the left.

4.4. Thin rectangles. A major application of explicit Runge–Kutta methods with many stages is the solution of moderately stiff advection–reaction–diffusion problems [14; 45]. For such problems, the stability region must include not only a large interval on the negative real axis, but also some region around it, due to convective terms. If centered differences are used for the advective terms, it is natural to require that a small interval on the imaginary axis be included. Hence, one may be interested in methods that contain a rectangular region

$$\Lambda_\kappa = \{\lambda \in \mathbb{C} : -\beta \leq \text{Im}(\lambda) \leq \beta, -\kappa \leq \text{Re}(\lambda) \leq 0\}. \quad (15)$$

for given κ, β . Most methods in the literature do not satisfy this requirement (with the notable exception of those in [47]). Most available approaches for devising methods with extended real axis stability (including those of [38]) cannot be applied to such regions. Because of this, most existing methods are applicable only if upwind differencing is applied to convective terms [45; 38].

For this example, rather than parametrizing by the step size h , we assume that a desired step size h and imaginary axis limit β are given based on the convective terms, which generally require small step sizes for accurate resolution. We seek to find (for given s, p) the polynomial (3) that includes Λ_κ for κ as large as possible. This could correspond to selection of an optimal integrator based on the ratio of convective and diffusive scales (roughly speaking, the Reynolds number). Since the desired stability region lies relatively near the negative real axis, we use the shifted and scaled Chebyshev basis (12).

Stability regions of some optimal methods are shown in Figure 10. The outline of the included rectangle is superimposed in black. The stability region for $\beta = 10, s = 20$, shown in Figure 10 is especially interesting as it is very nearly rectangular. A closeup view of the upper boundary is shown in Figure 11. These regions appear to satisfy the hypothesis stated in [38] that their boundary is tangent to the prescribed boundary at $s - p$ points in the upper half plane.

5. Discussion

The approach described here can speed up the integration of IVPs for which

- explicit Runge–Kutta methods are appropriate;

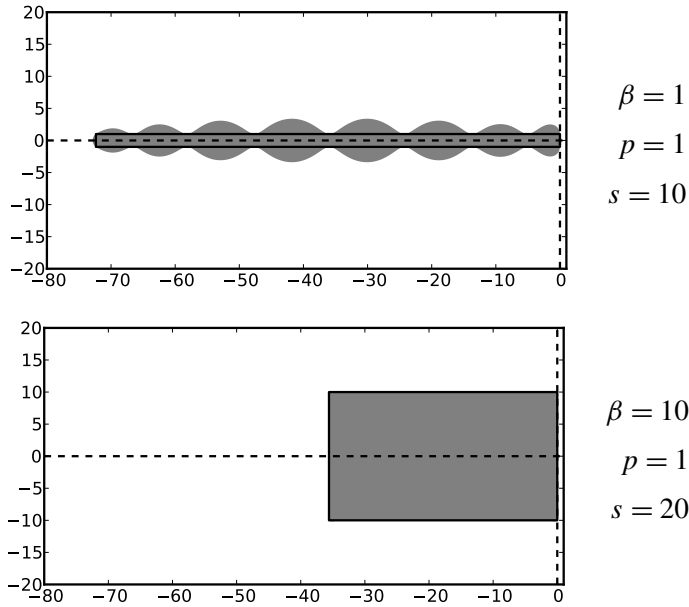


Figure 10. Stability regions of some optimal methods for thin rectangle inclusion.

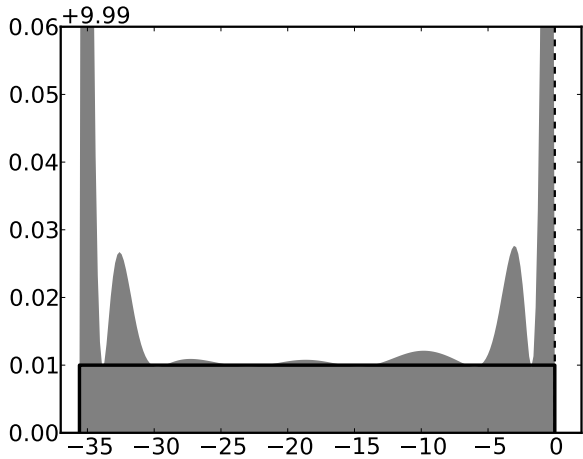


Figure 11. Closeup view of upper boundary of the rectangular stability region plotted in [Figure 10](#).

- the spectrum of the problem is known or can be approximated; and
- stability is the limiting factor in choosing the step size.

Although we have considered only linear initial value problems, we expect our approach to be useful in designing integrators for nonlinear problems via the usual approach of considering the spectrum of the Jacobian. A first successful application of our approach to nonlinear PDEs appears in [\[30\]](#).

The amount of speedup depends strongly on the spectrum of the problem, and can range from a few percent to several times or more. Based on past work and on results presented in [Section 4](#), we expect that the most substantial gains in efficiency will be realized for systems whose spectra have large negative real parts, such as for semidiscretization of PDEs with significant diffusive or moderately stiff reaction components. As demonstrated in [Section 4](#), worthwhile improvements may also be attained for general systems, and especially for systems whose spectrum contains gaps.

The work presented here suggests several extensions and areas for further study. For very high polynomial degree, the convex subproblems required by our algorithm exhibit poor numerical conditioning. We have proposed a first improvement by change of basis, but further improvements in this regard could increase the robustness and accuracy of the algorithm. It seems likely that our algorithm exhibits global convergence in general circumstances beyond those for which we have proven convergence. The question of why bisection seems to always lead to globally optimal solutions merits further investigation. While we have focused primarily on design of the stability properties of a scheme, the same approach can be used to optimize accuracy efficiency, which is a focus of future work. Our algorithm can also be applied in other ways; for instance, it could be used to impose a specific desired amount of dissipation for use in multigrid or as a kind of filtering.

One of the most remarkable aspects of our algorithm is its speed, which opens up the potential for a new kind of adaptive time stepping in which the time integration method itself is designed on-the-fly during the computation. For nonlinear problems, the method could be adapted, for instance, when a significant change in the spectrum of the linearized semidiscretization is detected. Whereas traditional automatic integrators dynamically adjust the step size and scheme order, choosing from a small set of preselected methods, our algorithm could be used as the basis for an implementation that also automatically adjusts details of the stability polynomial at each step. Practical implementation of this idea is dependent on the ability to efficiently approximate this spectrum and might require an implementation of our algorithm in a compiled language.

The problem of determining optimal polynomials subject to convex constraints is very general. Convex optimization techniques have already been exploited to solve similar problems in filter design [\[7\]](#), and will likely find further applications in numerical analysis.

Companion website

The codes used in producing the numerical results in this paper are available at <http://www.runmycode.org/CompanionSite/Site158> [\[18\]](#).

Acknowledgments

We thank Lajos Lóczy for providing a simplification of the proof of [Lemma 3](#). We are grateful to R. J. LeVeque and L. N. Trefethen for helpful comments on a draft of this work. We thank the anonymous referees for their comments that improved this paper.

References

- [1] A. Abdulle, *On roots and error constants of optimal stability polynomials*, BIT **40** (2000), no. 1, 177–182. [MR 2001a:65080](#) [Zbl 0956.65068](#)
- [2] ———, *Fourth order Chebyshev methods with recurrence relation*, SIAM J. Sci. Comput. **23** (2002), no. 6, 2041–2054. [MR 2003g:65074](#) [Zbl 1009.65048](#)
- [3] A. Abdulle and A. A. Medovikov, *Second order Chebyshev methods based on orthogonal polynomials*, Numer. Math. **90** (2001), no. 1, 1–18. [MR 2002i:65071](#) [Zbl 0997.65094](#)
- [4] A. B. Bogatyrev, *Effective solution of the problem of the best stability polynomial*, Mat. Sb. **196** (2005), no. 7, 27–50, In Russian; translated in Sbornik: Math. **196** (2005), 959–981. [MR 2007b:34124a](#)
- [5] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, Cambridge, 2004. [MR 2005d:90002](#) [Zbl 1058.90049](#)
- [6] J. C. Butcher, *Numerical methods for ordinary differential equations*, 2nd ed., Wiley, Chichester, 2008. [MR 2009b:65002](#) [Zbl 1167.65041](#)
- [7] T. Davidson, *Enriching the art of FIR filter design via convex optimization*, IEEE Signal Proc. Mag. **27** (2010), 89–101.
- [8] C. W. Gear and I. G. Kevrekidis, *Projective methods for stiff differential equations: problems with gaps in their eigenvalue spectrum*, SIAM J. Sci. Comput. **24** (2003), no. 4, 1091–1106. [MR 2004c:65065](#) [Zbl 1034.65056](#)
- [9] S. Gottlieb, D. Ketcheson, and C.-W. Shu, *Strong stability preserving Runge–Kutta and multi-step time discretizations*, World Scientific, Hackensack, NJ, 2011. [MR 2012f:65107](#) [Zbl 1241.65064](#)
- [10] M. C. Grant and S. P. Boyd, *Graph implementations for nonsmooth convex programs*, Recent advances in learning and control (V. Blondel, S. Boyd, , and H. Kimura, eds.), Lecture Notes in Control and Inform. Sci., no. 371, Springer, London, 2008, pp. 95–110. [MR 2409077](#) [Zbl 1205.90223](#)
- [11] ———, *CVX: MATLAB software for disciplined convex programming*, Tech. report, 2011.
- [12] E. Hairer and G. Wanner, *Solving ordinary differential equations, II: Stiff and differential-algebraic problems*, 2nd ed., Springer Series in Computational Mathematics, no. 14, Springer, Berlin, 1996. [MR 97m:65007](#) [Zbl 0859.65067](#)
- [13] R. Hettich and K. O. Kortanek, *Semi-infinite programming: theory, methods, and applications*, SIAM Rev. **35** (1993), no. 3, 380–429. [MR 94g:90152](#) [Zbl 0784.90090](#)
- [14] W. Hundsdorfer and J. Verwer, *Numerical solution of time-dependent advection-diffusion-reaction equations*, Springer Series in Computational Mathematics, no. 33, Springer, Berlin, 2003. [MR 2004g:65001](#) [Zbl 1030.65100](#)
- [15] R. Jeltsch and O. Nevanlinna, *Largest disk of stability of explicit Runge–Kutta methods*, BIT **18** (1978), no. 4, 500–502. [MR 80b:65099](#) [Zbl 0399.65051](#)

- [16] R. Jeltsch and M. Torrilhon, *Flexible stability domains for explicit Runge–Kutta methods*, Some topics in industrial and applied mathematics (R. Jeltsch, T.-T. Li, and I. H. Sloan, eds.), Ser. Contemp. Appl. Math. CAM, no. 8, Higher Ed. Press, Beijing, 2007, pp. 152–180. [MR 2008m:65180](#) [Zbl 1171.65415](#)
- [17] C. A. Kennedy, M. H. Carpenter, and R. M. Lewis, *Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations*, Appl. Numer. Math. **35** (2000), no. 3, 177–219. [MR 2001k:65111](#) [Zbl 0986.76060](#)
- [18] D. I. Ketcheson and A. J. Ahmadi, *Optimal stability polynomials for numerical integration of initial value problems*, Tech. report, 2012.
- [19] D. I. Ketcheson and M. Parsani, *RK-opt: Software for the design of optimal runge–kutta methods*, Tech. report, 2012.
- [20] I. P. E. Kinnmark and W. G. Gray, *One step integration methods of third-fourth order accuracy with large hyperbolic stability limits*, Math. Comput. Simulation **26** (1984), no. 3, 181–188. [MR 85k:65069](#) [Zbl 0539.65051](#)
- [21] ———, *One step integration methods with maximum stability regions*, Math. Comput. Simulation **26** (1984), no. 2, 87–92. [MR 85f:65079](#) [Zbl 0539.65050](#)
- [22] ———, *Fourth-order accurate one-step integration methods with large imaginary stability limits*, Numer. Methods Partial Differential Equations **2** (1986), no. 1, 63–70. [MR 89b:65175](#) [Zbl 0623.65077](#)
- [23] J. D. Lawson, *An order five Runge–Kutta process with extended region of stability*, SIAM J. Numer. Anal. **3** (1966), 593–597. [MR 35 #7589](#) [Zbl 0154.40602](#)
- [24] R. J. LeVeque, *Finite difference methods for ordinary and partial differential equations: Steady-state and time-dependent problems*, Soc. Industrial and Applied Math., Philadelphia, PA, 2007. [MR 2009a:65173](#)
- [25] J. Martín-Vaquero and B. Janssen, *Second-order stabilized explicit Runge–Kutta methods for stiff problems*, Comput. Phys. Comm. **180** (2009), no. 10, 1802–1810. [MR 2678453](#) [Zbl 1197.65006](#)
- [26] J. L. Mead and R. A. Renaut, *Optimal Runge–Kutta methods for first order pseudospectral operators*, J. Comput. Phys. **152** (1999), no. 1, 404–419. [MR 2000a:65083](#) [Zbl 0935.65100](#)
- [27] A. A. Medovikov, *High order explicit methods for parabolic equations*, BIT **38** (1998), no. 2, 372–390. [MR 99i:65096](#) [Zbl 0909.65060](#)
- [28] J. Niegemann, R. Diehl, and K. Busch, *Efficient low-storage Runge–Kutta schemes with optimized stability regions*, J. Comput. Phys. **231** (2012), no. 2, 364–372. [MR 2012m:65202](#) [Zbl 1243.65113](#)
- [29] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982, Reprinted Dover, Mineola, NY, 1998. [MR 84k:90036](#) [Zbl 0503.90060](#)
- [30] M. Parsani, D. I. Ketcheson, and W. Deconinck, *Optimized explicit Runge–Kutta schemes for the spectral difference method applied to wave propagation problems*, preprint, 2012, to appear in SIAM J. Sci. Comput. [arXiv 1207.5830](#)
- [31] J. Pike and P. L. Roe, *Accelerated convergence of Jameson’s finite-volume Euler scheme using van der Houwen integrators*, Comput. & Fluids **13** (1985), no. 2, 223–236. [MR 87d:65095](#) [Zbl 0571.76003](#)
- [32] R. A. Renaut, *Two-step Runge–Kutta methods and hyperbolic partial differential equations*, Math. Comp. **55** (1990), no. 192, 563–579. [MR 91d:65128](#) [Zbl 0724.65076](#)

- [33] W. Riha, *Optimal stability polynomials*, Computing (Arch. Elektron. Rechnen) **9** (1972), 37–43. [MR 47 #4450](#) [Zbl 0234.65076](#)
- [34] J. M. Sanz-Serna and M. N. Spijker, *Regions of stability, equivalence theorems and the Courant–Friedrichs–Lewy condition*, Numer. Math. **49** (1986), no. 2-3, 319–329. [MR 87i:65140](#) [Zbl 0574.65106](#)
- [35] L. M. Skvortsov, *Explicit stabilized Runge–Kutta methods*, Zh. Vychisl. Mat. Mat. Fiz. **51** (2011), no. 7, 1236–1250, In Russian: translated in Comput. Math. and Math. Phys. **51** (2011), 1153–1166. [MR 2906150](#) [Zbl 1249.65156](#)
- [36] B. P. Sommeijer and J. G. Verwer, *On stabilized integration for time-dependent PDEs*, J. Comput. Phys. **224** (2007), no. 1, 3–16. [MR 2008e:65263](#) [Zbl 1119.65382](#)
- [37] J. F. Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optim. Methods Softw. **11/12** (1999), no. 1-4, 625–653. [MR 1778433](#)
- [38] M. Torrilhon and R. Jeltsch, *Essentially optimal explicit Runge–Kutta methods with application to hyperbolic-parabolic equations*, Numer. Math. **106** (2007), no. 2, 303–334. [MR 2008d:65074](#) [Zbl 1113.65074](#)
- [39] T. Toulorge and W. Desmet, *Optimal Runge–Kutta schemes for discontinuous Galerkin space discretizations applied to wave propagation problems*, J. Comput. Phys. **231** (2012), no. 4, 2067–2091. [MR 2012m:65353](#) [Zbl 1242.65190](#)
- [40] L. N. Trefethen, *Computation of pseudospectra*, Acta numerica, 1999, Acta Numer., no. 8, Cambridge Univ. Press, Cambridge, 1999, pp. 247–295. [MR 2002b:65062](#) [Zbl 0945.65039](#)
- [41] L. N. Trefethen and M. Embree, *Spectra and pseudospectra: The behavior of nonnormal matrices and operators*, Princeton University Press, Princeton, NJ, 2005. [MR 2006d:15001](#)
- [42] R. H. Tütüncü, K. C. Toh, and M. J. Todd, *Solving semidefinite-quadratic-linear programs using SDPT3*, Math. Program. **95** (2003), no. 2, Ser. B, 189–217. [MR 2004c:90036](#) [Zbl 1030.90082](#)
- [43] P. J. van der Houwen, *Explicit Runge–Kutta formulas with increased stability boundaries*, Numer. Math. **20** (1972), 149–164. [MR 47 #6094](#) [Zbl 0233.65039](#)
- [44] ———, *The development of Runge–Kutta methods for partial differential equations*, Appl. Numer. Math. **20** (1996), no. 3, 261–272. [MR 97f:65053](#) [Zbl 0857.65094](#)
- [45] J. G. Verwer, B. P. Sommeijer, and W. Hundsdorfer, *RKC time-stepping for advection-diffusion-reaction problems*, J. Comput. Phys. **201** (2004), no. 1, 61–79. [MR 2005h:65151](#) [Zbl 1059.65085](#)
- [46] R. Vichnevetsky, *New stability theorems concerning one-step numerical methods for ordinary differential equations*, Math. Comput. Simulation **25** (1983), no. 3, 199–205. [MR 85b:65082](#) [Zbl 0573.65052](#)
- [47] C. J. Zbinden, *Partitioned Runge–Kutta–Chebyshev methods for diffusion-advection-reaction problems*, SIAM J. Sci. Comput. **33** (2011), no. 4, 1707–1725. [MR 2012m:65208](#) [Zbl 1245.65120](#)

Received July 12, 2012. Revised November 23, 2012.

DAVID I. KETCHESON: david.ketcheson@kaust.edu.sa

Division of Mathematical and Computer Sciences and Engineering, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia

ARON J. AHMADIA: aron@ahmadia.net

Division of Mathematical and Computer Sciences and Engineering, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia

Communications in Applied Mathematics and Computational Science

map.berkeley.edu/camcos

EDITORS

MANAGING EDITOR

John B. Bell
Lawrence Berkeley National Laboratory, USA
jbbell@lbl.gov

BOARD OF EDITORS

Marsha Berger	New York University berger@cs.nyu.edu	Ahmed Ghoniem	Massachusetts Inst. of Technology, USA ghoniem@mit.edu
Alexandre Chorin	University of California, Berkeley, USA chorin@math.berkeley.edu	Raz Kupferman	The Hebrew University, Israel raz@math.huji.ac.il
Phil Colella	Lawrence Berkeley Nat. Lab., USA pcolella@lbl.gov	Randall J. LeVeque	University of Washington, USA rjl@amath.washington.edu
Peter Constantin	University of Chicago, USA const@cs.uchicago.edu	Mitchell Luskin	University of Minnesota, USA luskin@umn.edu
Maksymilian Dryja	Warsaw University, Poland maksymilian.dryja@acn.waw.pl	Yvon Maday	Université Pierre et Marie Curie, France maday@ann.jussieu.fr
M. Gregory Forest	University of North Carolina, USA forest@amath.unc.edu	James Sethian	University of California, Berkeley, USA sethian@math.berkeley.edu
Leslie Greengard	New York University, USA greengard@cims.nyu.edu	Juan Luis Vázquez	Universidad Autónoma de Madrid, Spain juanluis.vazquez@uam.es
Rupert Klein	Freie Universität Berlin, Germany rupert.klein@pik-potsdam.de	Alfio Quarteroni	Ecole Polytech. Féd. Lausanne, Switzerland alfio.quarteroni@epfl.ch
Nigel Goldenfeld	University of Illinois, USA nigel@uiuc.edu	Eitan Tadmor	University of Maryland, USA etadmor@cscamm.umd.edu
Denis Talay		INRIA, France denis.talay@inria.fr	

PRODUCTION

production@msp.org

Silvio Levy, Scientific Editor Sheila Newbery, Senior Production Editor

See inside back cover or msp.berkeley.edu/camcos for submission instructions.

The subscription price for 2012 is US \$75/year for the electronic version, and \$105/year for print and electronic. Subscriptions, requests for back issues from the last three years and changes of subscribers address should be sent to Mathematical Sciences Publishers, Department of Mathematics, University of California, Berkeley, CA 94720-3840, USA.

Communications in Applied Mathematics and Computational Science, at Mathematical Sciences Publishers, Department of Mathematics, University of California, Berkeley, CA 94720-3840 is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

CAMCoS peer review and production are managed by EditFLOW™ from Mathematical Sciences Publishers.

PUBLISHED BY
 **mathematical sciences publishers**
<http://msp.org/>

A NON-PROFIT CORPORATION

Typeset in L^AT_EX

Copyright ©2012 by Mathematical Sciences Publishers

Communications in Applied Mathematics and Computational Science

vol. 7

no. 2

2012

Discontinuous Galerkin method with the spectral deferred correction time-integration scheme and a modified moment limiter for adaptive grids	133
LEANDRO D. GRYNGARTEN, ANDREW SMITH and SURESH MENON	
Analysis of persistent nonstationary time series and applications	175
PHILIPP METZNER, LARS PUTZIG and ILLIA HORENKO	
Approximation of probabilistic Laplace transforms and their inverses	231
GUILLAUME COQUERET	
Optimal stability polynomials for numerical integration of initial value problems	247
DAVID I. KETCHESON and ARON J. AHMADIA	