AIAA 2003–3705

Optimization of Multistage Coefficients for Explicit Multigrid Flow Solvers

Kaveh Hosseini and Juan J. Alonso
Stanford University, Stanford, CA 94305

16th AIAA Computational Fluid Dynamics Conference
June 23–26, 2003/Orlando, FL

AIAA 2003–3705

Optimization of Multistage Coefficients for Explicit Multigrid Flow Solvers

Kaveh Hosseini* and Juan J. Alonso†
Stanford University, Stanford, CA 94305

Explicit Euler and Navier-Stokes flow solvers based on multigrid schemes combined with modified multistage methods have become very popular due to their efficiency and ease of implementation. An appropriate choice of multistage coefficients allows for high damping and propagative efficiencies in order to accelerate convergence to the steady-state solution. However, most of these coefficients have been optimized either by trial and error or by geometric methods. In this work, we propose to optimize the damping and propagative efficiencies of modified multistage methods by expressing the problem within the framework of constrained non-linear optimization. We study the optimization of different objective functions with a variety of constraints and their effects on the convergence rate of a two-dimensional Euler flow solver. We start with a simple scalar one-dimensional model wave equation to demonstrate that the constrained optimization results yield multistage coefficients that are comparable to those that have been used in the past. We then extend the method to the case of the two-dimensional Euler equations and we distinguish each optimization case not only by choosing a different objective function, but also by specifying the conditions in a generic computational cell in terms of three parameters: the local Mach number, the aspect ratio, and the flow angle. Using this approach, it is possible to find the optimized coefficients for different combinations of these three parameters and use suitable interpolations for each of the cells in the mesh. Finally, we present observations regarding the relative importance of the propagative and damping efficiency of explicit schemes and conclude that, although a certain level of damping is required for stability purposes, the major contribution to convergence appears to be the propagative efficiency of the scheme.

Nomenclature

i	Imaginary unit $= \sqrt{-1}$
j, k	Cell indices in the two computational coordinate directions
m	Last (highest) level for multistage scheme
l	Arbitrary level index for multistage scheme
α_l, β_l	Modified Runge-Kutta multistage coefficients
w	State variable for the scalar wave equation
W	State vector for conservative variables
dW_e	Differential state vector for entropy variables
$W_{j,k}$	Discretized state vector (state at mesh points)
F_x, F_y	Flux vectors in the x and y directions
$\mathcal{S}$	Cell surface
dS_x, dS_y	Projections of $\mathcal{S}$ in the x and y directions
u	x-component of velocity
v	y-component of velocity
ρ	Density
p	Static pressure
E	Total energy (internal plus kinetic)
H	Total enthalpy
q	Velocity magnitude
c	Speed of sound
R	Residual
Q	Convective part of the residual
D	Dissipative part of the residual
Δt	Time step
Δx	Spatial increment in the x direction
Δy	Spatial increment in the y direction
a	Wave speed in the x direction
λ	Courant-Friedrichs-Lewy or CFL number
μ	Dissipation coefficient
z, Z	Fourier residual operators
g	Amplification factor
ξ, η	Error frequencies
A, B	Flux Jacobians in conservative variables
A_e, B_e	Flux Jacobians in entropy variables
I	Objective function to optimize
I_4	Four by four identity matrix
M	Mach number
φ	Flow angle
A_R	Cell Aspect Ratio
P	Preconditioning matrix
$\Re$	Real part
$\Im$	Imaginary part

*Ph.D. Candidate, AIAA Student Member
†Assistant Professor, AIAA Member

Introduction

DURING the last decade, explicit multigrid methods have carved their niche in the solution of complex inviscid and viscous flows. In nearly isotropic meshes, such as those used for the solution of the Euler equations, explicit methods achieve rates of convergence that approach the theoretical limit. Despite the fact that these convergence rates degrade significantly

in the solution of the Reynolds Averaged Navier-Stokes equations (due to both the presence of viscous effects and the existence of highly stretched meshes),[12] the cost of solution using these explicit multigrid methods rivals those of implicit solvers. Moreover, explicit methods achieve much more scalable parallel implementations and therefore have been a popular choice for the solution of large-scale, complex configuration problems.

Multistage schemes for the numerical solution of ordinary differential equations are usually designed to yield a high order of accuracy. Since the present objective is simply to obtain a steady state as rapidly as possible, the order of accuracy is not important. This allows the use of schemes selected purely for their properties of stability and damping. For this purpose it pays to distinguish the hyperbolic and parabolic parts stemming respectively from the convective and dissipative terms, and to treat them differently.

A majority of the most efficient existing explicit multigrid solvers are based on the use of a modified Runge-Kutta relaxation scheme which often treats the convective and dissipative portions of the residual separately so that optimum convergence rates at minimum computational cost can be achieved.[5,7] A number of 3-, 4-, and 5-stage Runge-Kutta schemes have been developed in the past and have shown good convergence properties. The choice of the coefficients for these multistage schemes was based on the one-dimensional scalar model equation and the current state of the art coefficients have been found via trial and error. In some cases, simplified geometric methods[13] were used or optimizations were performed for classical Runge-Kutta multistage schemes only, but not for the computationally advantageous modified Runge-Kutta schemes.[8]

In this work, we propose to formalize the choice of coefficients for these multistage schemes, by casting the problem into a constrained optimization framework. The optimum coefficients are those that maximize both the damping and the convergence rates of the scheme subject to appropriate constraints. Using the values of the multistage coefficients as design variables, a non-linear constrained optimizer can be used in such a way that the convergence rates are optimized. It must be noted that the choice of cost function and constraints that maximize the convergence rate of a scheme is never straightforward. In fact, it has been noted in the past that for explicit multigrid schemes, it is important to select coefficients that efficiently damp the high-frequency modes in each grid level. However, little attention has been paid to the more important issue of propagative efficiency.

We first start by defining the governing equations of the problem to which we will apply our optimization procedure and the associated modified Runge-Kutta multistage scheme. We then present the one-dimensional scalar model convection-diffusion equation and the results of the application of the constrained optimization procedure to the choice of multistage coefficients for different objective functions. Finally, we discuss similar results for the Euler equations and outline the procedure that could be used to carry out a coefficient optimization similar to the one presented here.

Governing Equations

Let us start by considering the case of the Euler equations for the two-dimensional flow of an inviscid ideal gas. The finite volume form can be written as

$$\frac{d}{dt}\int_{\mathcal{S}} W d\mathcal{S} + \int_{\partial\mathcal{S}} (F_x dS_x + F_y dS_y) = 0. \qquad (1)$$

Using conservative variables

$$W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \qquad (2)$$

the Euler fluxes in the x and y directions are

$$F_x = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{pmatrix}, F_y = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vH \end{pmatrix}, \qquad (3)$$

where

$$p = (\gamma - 1)\,\rho\left(E - \frac{q^2}{2}\right) \qquad (4)$$

$$H = E + \frac{p}{\rho} = \frac{c^2}{\gamma - 1} + \frac{q^2}{2} \qquad (5)$$

$$q^2 = u^2 + v^2 \qquad (6)$$

$$c^2 = \frac{\gamma p}{\rho}. \qquad (7)$$

If we discretize the spatial derivatives in eq. 1 and add artificial dissipation terms, we can write the semi-discrete form as

$$\begin{aligned} \frac{dW_{j,k}}{dt} + R_{j,k} &= 0 \\ R_{j,k} = R(W_{j,k}) &= Q_{j,k} + D_{j,k}. \end{aligned} \qquad (8)$$

The remainder of this paper will be concerned with the stability, damping, and propagative efficiency of numerical schemes used to drive this equation to a steady-state.

Explicit Multistage Method

Modified Runge-Kutta Equations

Let W^n be the value of W after n time steps. Dropping the subscripts j, k the general m-stage modified

scheme[7] can be written as

$$\begin{aligned} W^{(n+1,0)} &= W^n \\ W^{(n+1,1)} &= W^{(0)} - \alpha_1 \Delta t R^{(0)} \\ &\vdots \\ W^{(n+1,l)} &= W^{(0)} - \alpha_l \Delta t R^{(l-1)} \\ &\vdots \\ W^{(n+1,m)} &= W^{(0)} - \quad \Delta t R^{(m-1)} \\ W^{n+1} &= W^{(n+1,m)}, \end{aligned} \tag{9}$$

with

$$\begin{aligned} R^{(l)} &= Q^{(l)} + D^{(l)} \\ Q^{(0)} &= Q(W^n) \\ D^{(0)} &= D(W^n) \\ Q^{(l)} &= Q(W^{(n+1,l)}) \\ D^{(l)} &= \beta_l D(W^{(n+1,l)}) + (1-\beta_l) D(W^{(n+1,l-1)}). \end{aligned}$$

Note how this method distinguishes between the convective and dissipative terms by using two sets of coefficients α_l and β_l. Such non-standard multistage schemes have much larger stability regions than conventional Runge-Kutta schemes. Moreover, by forcing some of the β_l to be zero, we can avoid the computation of dissipative terms on the corresponding stages and thus reduce computational costs.[5] Note that if we set all β_l to 1 we simply find the classical Runge-Kutta equations.

5-stage Martinelli-Jameson coefficients

One very popular modified Runge-Kutta scheme known for its high propagative and damping efficiencies and large stability domain is a 5-stage scheme proposed by Martinelli and Jameson with the following coefficients

$$\begin{aligned} \alpha_1 &= 1/4 & \beta_1 &= 1 \\ \alpha_2 &= 1/6 & \beta_2 &= 0 \\ \alpha_3 &= 3/8 & \beta_3 &= 14/25 \\ \alpha_4 &= 1/2 & \beta_4 &= 0 \\ \alpha_5 &= 1 & \beta_5 &= 11/25. \end{aligned} \tag{10}$$

We will refer to the above coefficients as the 5-stage MJ coefficients. For these coefficients, we can note that every other β_l is equal to zero, this means that we save on computational costs by computing the dissipative fluxes every other stage only. We can also note that consistency requirements impose that $\alpha_5 = beta_1 = 1$.

We will use the MJ coefficients as our principal reference in order to assess the efficiency of an optimized set of coefficients. Thus for the purpose of this paper we will concentrate only on 5-stage schemes that have $\beta_2 = \beta_4 = 0$.

Scalar one-dimensional model wave equation

General formulation

Let us consider the one-dimensional model problem

$$w_t + a w_x + \mu \Delta x^3 w_{xxxx} = 0. \tag{11}$$

In the absence of 3^{rd} order dissipation and for $a = 1$, eq. 11 describes the propagation of a disturbance at unit speed without distortion. The viscous term represents the artificial dissipation added in order to stabilize central differencing numerical methods. Using central differencing our residual has the form

$$\begin{aligned} \Delta t R_j &= \frac{\lambda}{2}\left(w_{j+1} - w_{j-1}\right) \\ &+ \lambda\mu\left(w_{j+2} - 4w_{j+1} + 6w_j - 4w_{j-1} + w_{j-2}\right), \end{aligned} \tag{12}$$

with the CFL number defined as

$$\lambda = \frac{a\Delta t}{\Delta x} = \frac{\Delta t}{\Delta x}. \tag{13}$$

Now, if we consider a Fourier mode $\hat{w} = e^{ipx}$ with $x_j = j\Delta x$ and set $\xi = p\Delta x$, we have

$$\Delta t \frac{d\hat{w}}{dt} = z\hat{w}, \tag{14}$$

with

$$z = -\lambda i \sin\xi - 4\lambda\mu(1-\cos\xi)^2. \tag{15}$$

A single step of the multistage scheme yields

$$\hat{w}^{n+1} = g(z)\hat{w}^n, \tag{16}$$

where $g(z)$ is called the amplification factor. For an m-stage scheme, g is computed iteratively. For classical Runge-Kutta schemes g is a complex polynomial function of z but for modified Runge-Kutta schemes g is the sum of a polynomial function of $\Re(z)$ and a polynomial function of $\Im(z)$

Stability Domain for the MJ coefficients

We can use the stability domain of a given set of multistage coefficients as a visual tool in order to understand some of the mechanisms of damping and propagation. Fig. 1 shows the stability domain of the MJ coefficients, the locus of the residual operator $z(\xi)$ as defined in eq. 15, and the amplification factor $|g|$ for three different sets of λ and μ. On the left, the stability domain is represented by contours lines of the amplification factor with contour levels going from the outside to the inside from 1 to 0 with increments of 0.1. The locus of $z(\xi)$ is represented on the same figure for $-\pi \leq \xi \leq \pi$ as the oval-shaped curve extending in the direction of the imaginary axis. On the right we can see how the locus of $z(\xi)$ translates into the amplification factor $\left|g\left(\frac{\xi}{\pi}\right)\right|$ and since the locus of $z(\xi)$ is symmetric, we will only consider $0 \leq \frac{\xi}{\pi} \leq 1$.

On fig. 1-a we have the locus of $z(\xi)$ for $\lambda = 2.8$ and $\mu = \frac{1}{32}$. This value of $\mu = \mu_0$ is typically used in CFD computations and we will use it as a reference value. It is important to keep μ small in order to avoid introducing too much dissipation, otherwise we would produce a numerical solution that would not represent accurately the physical solution. Fig. 1-b shows the amplification factor $|g|$ corresponding to the locus of z shown on fig. 1-a. We can clearly see that high-frequency errors are damped more efficiently than low-frequency errors. This is a key feature of such multistage schemes. We are indeed interested in damping high-frequency errors efficiently because multigrid techniques transfer low-frequency errors from fine meshes to coarser meshes on which they become high-frequency errors themselves.[6,12] Therefore we will concentrate on improving the damping properties of multistage schemes on the interval $\frac{\pi}{2} \leq \xi \leq \pi$.

Although it is well-known that convergence depends on both propagation and damping, there is no simple analytical way to compare their respective contributions to convergence. We will consider the amplification factor $|g|$ as a measure of damping efficiency and the CFL number λ as a measure of propagative efficiency. To analyze propagative efficiency in a more complete way, one can include the effects of residual smoothing as it can dramatically increase λ. One can also use a more precise measure of propagative efficiency by analyzing the group velocity $c_g(\xi)$ of a wave packet rather than λ only. For the sake of simplicity, we have chosen to leave the impact of residual smoothing and group velocity to further studies.

On figs. 1-c and 1-d we can see that if λ is increased, the locus of z extends in the directions of both axes. These figures clearly show that for $\lambda = 3.93$, the locus of z is extended to its stability limit. On the other hand, figs. 1-c and 1-d show that if μ is increased to $2\mu_0$, the locus of z extends in the direction of the real axis only. In order to avoid too much loss in accuracy, the value $\mu = 2\mu_0 = \frac{1}{16}$ is usually the maximum used in CFD computations.

Optimization Framework

The optimization framework is based on a traditional non-linear programming setup with an objective function to minimize (or maximize) and a set of linear and/or non-linear constraints. We want to maximize convergence speed and all we know is that it is a function of propagation and damping. But since there exists no clear analytical expression linking convergence speed to propagation and damping, we will define several sets of objective functions and constraints. We will use as variables the CFL number λ, the dissipation coefficient μ, and the multistage coefficients α_l and β_l.

The constraints below will be common to all of our objective functions:

$$\lambda > 0 \tag{17}$$

$$\frac{0.5}{32} \leq \mu \leq \frac{2}{32} \tag{18}$$

$$0 \leq \alpha_l, \beta_l \leq 1 \tag{19}$$

$$|g| \leq 1 \tag{20}$$

$$\alpha_5 = \beta_1 = 1 \tag{21}$$

$$\beta_2 = \beta_4 = 0. \tag{22}$$

Note that the inequality in eq. 20 will be strict except for $\xi = 0$. Our 8 variables are α_1, α_2, α_3, α_4, β_3, β_5, λ, and μ. Our optimization results for the scalar one-dimensional wave equation were obtained using MATLAB's gradient-based optimization function `fmincon` for constrained minimization. One of the limitations of gradient-based methods is that they are only guaranteed to find local optima and it is never clear whether we have a global optimum or not. The advantage of such methods on the other hand is that they are very fast compared to methods that have the potential to find global optima such as genetic algorithms. The comparison of our results obtained using both `fmincon` and a genetic algorithm will be part of future work.

Test cases

We will study 7 cases using different objective functions, the constraints in eqs. 17-22, and in some cases additional constraints that will be described later. In each case we will have 2 sub-cases, one where μ will be considered as a variable and another one where we will impose the constraint $\mu = \mu_0$. Therefore, in reality we will have 14 cases. Tab. 1 summarizes the results of these optimizations. We will assign a number to each case followed by the letter V or F designating whether μ was variable or fixed. The values for α_1, α_2, α_3, α_4, β_3, β_5, λ, and μ in the table correspond to the results produced by `fmincon`.

We used Jameson and Martinelli's two-dimensional multigrid viscous flow solver FLO103[4,9,10] to test the optimized coefficients for two Mach numbers: $M = 0.5$ and $M = 0.8$. The test case was a NACA0012 at an angle of attack of 2.25° using 5 levels of multigrid on a 160x32 C-mesh. One of the features of FLO103 is that it computes high-order dissipation fluxes on fine meshes and low-order dissipation fluxes on coarse meshes. Since our optimizations are based on the one-dimensional scalar model equation using high-order dissipation terms, we modified FLO103 so that it computes high-order dissipation fluxes on all meshes. This enables us to have a more correct validation of our optimization approach. If we were to use FLO103 in its original form, then, in order to be rigorous we would need two sets of optimized coefficients: one set obtained with a one-dimensional scalar model using 3^{rd} order dissipation, and another set with a one-

dimensional scalar model using 1^{st} order dissipation. This is currently being studied.

When running FLO103 on Euler test cases, sometimes we had to use a lower λ than that given by `fmincon` in order to achieve convergence. The λ used for the actual computation and the corresponding residuals after 150 multigrid cycles are denoted by $\lambda_{0.5}$, $Res_{0.5}$, $\lambda_{0.8}$ and $Res_{0.8}$ for the two Mach numbers mentioned above. The first line in tab. 1 represents our reference case using the MJ coefficients. A remarkable result is that the fastest convergence using the MJ coefficients is obtained with $\lambda = 3.93$. It turns out that this value corresponds to the theoretical maximum predicted by the one-dimensional model problem. This is in itself a very strong justification of the fact that such a simplified model is well adapted to the study of actual flow solvers.

Case 1: Minimization of amplification factor for all frequencies

In this approach we minimize $I = \int_0^\pi |g(\xi)|\, d\xi$ with no additional constraints. Fig. 2 shows the case with variable μ and fig. 3 shows the case with fixed μ. Compared to the MJ case on figs. 1-c and -d we can see that although the amplification factor is greatly reduced, it is done at the expense of the propagation efficiency: λ is much smaller now. Both cases 1V and 1F achieve convergence but both are slower than the MJ case.

Comparing 1V to 1F, we can see that 1V achieves better damping than 1F but at the same time it has a smaller λ too. Part of the better damping of 1V is certainly due to its larger μ. It turns out that 1F has faster convergence than 1V and this is a first indication that propagation has a more important role in contributing to convergence than damping.

Case 2: Minimization of amplification factor for high frequencies only

Since we use multigrid in our solution procedure, let us concentrate on the high-frequency domain only and minimize $I = \int_{\frac{\pi}{2}}^\pi |g(\xi)|\, d\xi$ with no additional constraints. Fig. 4 shows the case with variable μ and fig. 5 shows the case with fixed μ. Compared to case 1 we can see that we have further reduced damping, especially in case 2V where $|g|$ in the high-frequency domain is in the order of 0.01. Nevertheless, the residuals of 2V and 2F are larger than the residuals of 1V and 1F respectively. This loss in convergence rate is due to the fact that the λ of 2V and 2F are smaller than those of 1V and 1F respectively. This means that the loss in λ is not compensated by the improvements in damping and that reaffirms our first indication about the role of propagation.

Case 3: Maximization of the CFL number λ

Now that propagation seems to have a more important role than damping, let us maximize λ with no additional constraints. Figs. 6 and 7 show that we can barely push λ beyond 3.93, which was the maximum value for the MJ case. In case 3V, it is interesting to note that because of the fact that the extra increase in λ might push the locus of z out of the stability domain, the optimizer decreases μ to keep the locus inside the domain and achieve a slightly higher λ than in case 3F. For both cases 3V and 3F, the very slight gain in CFL is achieved at a catastrophic cost for the stability domain. The stability domain extends very little in the direction of the negative real axis and its limits come in contact with the locus of z in the region of high frequencies now. FLO103 could not achieve convergence for these theoretical values of maximum λ anymore. In order to achieve a stable convergence for such restricted domains of stability, we had to reduce λ to about 1.5. At such a loss of λ, the maximization of λ simply defeats its purpose. This can also be double-checked by taking a look at the values of the residuals at the end of each calculation.

Cases 4 and 5: Maximization of the CFL number λ with additional constraints

It has now become obvious that the optimization of damping alone or the optimization of propagation alone does not lead to better convergence. It becomes more and more apparent that propagation plays a more important role in convergence than damping. Thus we continue maximizing propagation while adding some safeguards to achieve proper damping for high frequencies. For cases 4 and 5, we maximize λ but we impose an additional constraint on $|g|$ in the domain of high frequencies. In fact, what we try to do is a blend of cases 2 and 3.

For case 4 we impose $|g| \leq 0.25$ for $0.5 \leq \frac{\xi}{\pi} \leq 1$. As we can see on figs. 8 and 9, the stability domains in these cases start to approach the general shape of the MJ scheme with good extensions in both directions. The optimized solutions result in rather high λ. For case 4V the practical λ had to be reduced to produce the best results in FLO103, and for the first time, we have convergence rates that approach the MJ case. For case 4F, we did not have to use a reduced λ and we obtained good convergence performance. So far it seems that fixing λ has been a better strategy than using it as a variable.

For case 5 we impose $|g| \leq 0.8$ for $0.5 \leq \frac{\xi}{\pi} \leq 1$. Now that we are being less stringent on the damping constraint, we produce a very high λ, but as we can see in the results of tab. 1 and figs. 10 and 11, we are again faced with problems similar to those in case 3 as we need to use reduced λ in FLO103 calculations. Furthermore, case 5V did not produce good convergence even with reduced λ, probably because of a very small μ. Again this case confirms that using μ as a variable might not be worth the degree of freedom it offers.

Case 6: Minimization of a hybrid function of the amplification factor $|g|$ and the CFL number λ

From the results above it appears that convergence is a strong function of propagation and a weak function of damping. Therefore case 6 optimizes the hybrid objective function $I = K\frac{\int_{\frac{\pi}{2}}^{\pi}|g(\xi)|d\xi}{\lambda^n}$ where K is simply a constant chosen large enough in order to avoid numerical difficulties for large values of n. K will be typically chosen of the order of 4^n. We tested many values of n and it appears that for any value above 5, there are only slight differences in the optimization results. In fact, we found that by using very large values of n we could push the maximum λ a little further than the limit value of 3.93 of the MJ scheme without catastrophic negative effects on the stability domain. The following results were obtained with the very large value of $n = 25$. This produces an objective function where a slight increase in λ impacts the minimization procedure much more heavily than a decrease in amplification factor. Fig. 12 and 13 show that the stability domains of case 6 are very similar to the MJ case, except that they are less extended in the direction of the negative real axis. Once again, fixed μ performs better than variable μ. As in case 3V, the optimizer seems to use the degree of freedom offered by a reduction in μ in order to ease the extension of the locus of z in the direction of the imaginary axis and achieve a high λ. Finally, it is interesting to note that case 6F has a slightly higher convergence rate than the MJ case.

Case 7: Minimization of the amplification factor $|g|$ with additional constraint on the CFL number λ

Now that it is apparent that we should try to achieve the highest λ without totally sacrificing on damping, we simply set as large a lower limit on λ as possible while trying to minimize $\int_{\frac{\pi}{2}}^{\pi}|g(\xi)|\,d\xi$. We tested several values for that lower limit and the largest value that the optimizer could find a solution for was for constraint $\lambda \geq 3.99$.

As in the previous case, fixed μ performed better than variable μ. In case 7V, the optimizer again used a reduction in the value of μ in order to achieve a large value of λ. The performance of case 7V is very good, but it is even more interesting to note that case 7F performs better than all the cases studied so far and of course even slightly better than the MJ case as can be seen in tab.1. Fig. 14 is very similar to the figures of case 6, while fig. 15 is very similar to the MJ case with very slightly larger extensions in both directions.

First conclusions

At this point we can affirm that although the MJ coefficients were developed by trial and error, they were nearly perfect and grasped the important mechanisms involved in both damping and propagation. This study showed that the contribution of propagation to convergence is far superior than that of damping, provided that we do not perform at the limits of stability in the high-frequency domain.

We also showed that despite its simplicity, the one-dimensional equation model provides an efficient tool for the analysis of multistage schemes even when used for the Euler equations. Finally we showed that it is not necessarily a wise choice to consider the dissipation coefficient μ as a variable in multistage optimizations.

The next section extends the analysis of the one-dimensional scalar model wave equation to the system of the two-dimensional Euler equations in a similar fashion.

Two-dimensional Euler equations

General formulation

The finite difference form of eq. 1 can be written as

$$\frac{\partial W}{\partial t} + A\frac{\partial W}{\partial x} + B\frac{\partial W}{\partial y} = 0, \tag{23}$$

where A and B are the flux Jacobians of F_x and F_y with respect to W.

For the discretized equations, using central differencing, we have

$$\left(A\frac{\partial W}{\partial x}\right)_{j,k} = \frac{1}{\Delta x}\left(F_{x_{j+\frac{1}{2},k}} - F_{x_{j-\frac{1}{2},k}}\right), \tag{24}$$

and for the scheme with 3^{rd} order matrix dissipation we have

$$F_{x_{j+\frac{1}{2},k}} = A\left(\frac{W_{j+1,k}+W_{j,k}}{2}\right) - \frac{\mu}{2}|A|\left(\Delta W_{j+\frac{3}{2},k} - 2\Delta W_{j+\frac{1}{2},k} + \Delta W_{j-\frac{1}{2},k}\right), \tag{25}$$

and

$$F_{x_{j-\frac{1}{2},k}} = A\left(\frac{W_{j,k}+W_{j-1,k}}{2}\right) - \frac{\mu}{2}|A|\left(\Delta W_{j+\frac{1}{2},k} - 2\Delta W_{j-\frac{1}{2},k} + \Delta W_{j-\frac{3}{2},k}\right). \tag{26}$$

If we denote by T and T^{-1}, the matrices of the left and the right eigenvectors of A, we have $A = T\Lambda T^{-1}$ where Λ is the diagonal matrix of the eigenvalues of A. We then define $|A|$ such that $|A| = T\,|\Lambda|\,T^{-1}$ where $|\Lambda|$ is the diagonal matrix of the absolute values of the eigenvalues of A.

If we use scalar dissipation rather than matrix dissipation, then $|A|$ and $|B|$ must be replaced by $\rho(A)I_4$ and $\rho(B)I_4$ where $\rho(A)$ and $\rho(B)$ are the spectral radii of A and B and I_4 is the 4 by 4 identity matrix.

Now, if we consider a Fourier mode $\hat{W} = e^{ipx}e^{iry}$ and set $\xi = p\Delta x$ and $\eta = r\Delta y$, we have

$$\left(A\frac{\partial \hat{W}}{\partial x}\right)_{j,k} = \frac{1}{\Delta x}\left(iA\sin\xi + 4\mu\,|A|\,(1-\cos\xi)^2\right), \tag{27}$$

and similarly

$$\left(B\frac{\partial \hat{W}}{\partial y}\right)_{j,k} = \frac{1}{\Delta y}\left(iB\sin\eta + 4\mu\,|B|\,(1-\cos\eta)^2\right). \tag{28}$$

Instead of having a scalar z, we have a Z matrix such that

$$\Delta t\frac{d\hat{W}}{dt} = Z\hat{W}, \tag{29}$$

and

$$Z(\xi,\eta) = -\frac{\Delta t}{\Delta x}\left[iA\sin\xi + 4\mu\,|A|\,(1-\cos\xi)^2\right] - \frac{\Delta t}{\Delta y}\left[iB\sin\eta + 4\mu\,|B|\,(1-\cos\eta)^2\right]. \tag{30}$$

Here the eigenvalues of Z play the role of the scalar z of the one-dimensional case. Since they are functions of both ξ and η, each of them will have a locus that will cover a surface on the stability domain instead of producing just a single curve.

Evaluation of the eigenvalues of Z

In order to properly and efficiently compute the eigenvalues of Z, the best strategy is to use the Jacobians A_e and B_e for entropy variables rather than the commonly used conservative variables. We can express the entropy variables in differential form as

$$dW_e = \begin{pmatrix} \frac{dp}{\rho c} \\ du \\ dv \\ dw \\ dp - c^2 d\rho \end{pmatrix}. \tag{31}$$

The flux Jacobians A_e and B_e in entropy variables are simply

$$A_e = \begin{pmatrix} u & c & 0 & 0 \\ c & u & 0 & 0 \\ 0 & 0 & u & 0 \\ 0 & 0 & 0 & u \end{pmatrix} \tag{32}$$

$$B_e = \begin{pmatrix} v & 0 & c & 0 \\ 0 & v & 0 & 0 \\ c & 0 & v & 0 \\ 0 & 0 & 0 & v \end{pmatrix}. \tag{33}$$

With such simple expressions for the Jacobians, it is easy to verify that the eigenvalues of A_e and B_e (thus of A and B) are $(u, u, u+c, u-c)$ and $(v, v, v+c, v-c)$ respectively. As for the spectral radii, if we assume that u and v are positive, we have

$$\rho(A) = u + c \tag{34}$$

$$\rho(B) = v + c. \tag{35}$$

One very important factor in the evaluation of the eigenvalues of Z is the proper definition of the two-dimensional CFL number λ. Many flow solvers use a conservative definition of the time step limit from which λ is derived as follows.

$$\lambda = \Delta t\left(\frac{u+c}{\Delta x} + \frac{v+c}{\Delta y}\right). \tag{36}$$

Using the above definition to eliminate Δt from eq. 30 would allow to produce loci of the eigenvalues of Z that remain in the stability domain even for λ that are unusually large compared to the one-dimensional case and that do not correspond to what can be actually used in a flow solver.

A proper definition of λ for stability analysis should rather be based on the largest eigenvalue of the matrix

$$\frac{1}{\Delta x}A + \frac{1}{\Delta y}B. \tag{37}$$

The proper definition of λ then becomes

$$\lambda = \Delta t\left(\frac{u\Delta y + v\Delta x + c\sqrt{\Delta x^2 + \Delta y^2}}{\Delta x\Delta y}\right). \tag{38}$$

It now becomes apparent that the eigenvalues of Z will be uniquely defined as functions of ξ and η for a given computational cell where we know the Mach number $M = \frac{\sqrt{u^2+v^2}}{c}$, the flow angle $\varphi = \arctan\left(\frac{v}{u}\right)$, and the aspect ratio $A_R = \frac{\Delta x}{\Delta y}$. This means that it is possible to produce a different set of optimized multistage coefficients for each computational cell.

Conclusions

In this paper we have presented a new and systematic optimization framework for the calculation of multistage coefficients for modified Runge-Kutta schemes. In addition, we have developed a number of model equations that can be used in conjunction with a nonlinear constrained optimizer to produce the desired results. We showed that the influence of propagation largely dominates that of damping. Many researchers such as Tai[13] and Lynn[8] had sought to optimize multistage coefficients for classical Runge-Kutta schemes only by focusing on the damping efficiency. Tai's one-dimensional geometric method forces the locus of z to pass through the zeros of the stability domain while Lynn's two-dimensional method minimizes the maximum amplification factor on a portion of the high frequency domain. The coefficients they have produced are therefore not appropriate for the advantageous case of modified Runge-Kutta schemes in addition to producing tight CFL restrictions. Moreover, in most previous multistage optimization work, dissipative terms were often chosen to be very large, thus producing results that are not necessarily well-suited for practical use in flow solvers. This study proved in a systematic way that the MJ coefficients were extremely close to the the optimal solutions found in case 7F and opened the way for future work in the cases of preconditioned Euler and Navier-Stokes equations.

Future work

Adaptive Multistaging

An interesting idea for the future is to produce an array of optimized coefficients for a large number of different combinations of M, φ, and A_R. It would then be possible to feed that array into a flow solver that would use it to interpolate a different set of optimized coefficients for every cell rather than using the same coefficients throughout the whole computational domain.

In order to illustrate the differences between the one-dimensional model and the two-dimensional Euler case, figs. 16 to 21 show the stability domain and loci of the eigenvalues of Z for a scheme using scalar dissipation. These figures compare the stability domain of the MJ coefficients on the left with that of the optimized coefficients on the right. The optimized coefficients were obtained using the non-linear gradient-based optimizer package SNOPT.[3,11] As in the last case of the one-dimensional model, we added a constraint on the lower bound of λ. In this case we chose to minimize $I = \int_{\frac{\pi}{2}}^{\pi}\int_{\frac{\pi}{2}}^{\pi} |g(\xi,\eta)|\, d\xi d\eta$ subject to the constraint that $\lambda \geq 3.9$. Six combinations of M, φ, and A_R are presented where we can see how these parameters influence the clustering of the eigenvalues of Z in different regions of the stability domain. In all cases, the envelope of the loci of the eigenvalues of Z is the same as the locus of z from the one-dimensional model.

For an aspect ratio of 1, figs. 16 to 18 show that the eigenvalues of Z populate the entire surface inside that envelope for $M = 0.5$, but when we reduce M to 0.1, some regions of the stability domain are depopulated. Changing the flow angle has only a small influence on this feature. The optimized solutions seem to produce stability domains that adapt themselves to the way the eigenvalues are clustered in different regions of the stability domain.

Figs. 19 to 21 show that by increasing A_R, we significantly reduce the surface covered by the eigenvalues to one main oval-shaped branch constituting the envelope and several additional smaller branches that extend differently as a function of M and φ. Once again, it seems that the optimizer takes the locations of these branches into account as each stability domain has a different shape.

Preconditioning

The adaptive multistaging approach described above can be extended to the case of the preconditioned Euler or Navier-Stokes equations. Although the MJ coefficients happened to be practically optimal for Euler computations, it is not clear what coefficients would be optimal in the case of the preconditioned Euler equations. In this case we simply have

$$P^{-1}\frac{\partial W}{\partial t} + A\frac{\partial W}{\partial x} + B\frac{\partial W}{\partial y} = 0, \qquad (39)$$

or

$$\frac{\partial W}{\partial t} + P\left(A\frac{\partial W}{\partial x} + B\frac{\partial W}{\partial y}\right) = 0, \qquad (40)$$

where P is the preconditioning matrix. Therefore we simply need to replace Z by PZ and we will have

$$\Delta t\frac{d\hat{W}}{dt} = PZ\hat{W}. \qquad (41)$$

We can clearly see that in this case our amplification factors are a function of the eigenvalues of PZ instead of Z. The optimized coefficients will be different for each type of preconditioner. In the future we will consider different types of preconditioners such as the Block-Jacobi preconditioner,[1] the low-Mach Turkel preconditioner,[14] and combinations of low-Mach and Block-Jacobi.[2]

Acknowledgments

The authors acknowledge the support of the Department of Energy as part of the ASCI program under contract number LLNL-B341491.

References

[1]S.R. Allmaras. Analysis of a local matrix preconditioner for the 2-D Navier-Stokes equations. *AIAA Paper* 93-3330, Orlando, FL, June 1993.

[2]D.L. Darmofal and K. Siu. A robust multigrid algorithm for the Euler equations with local preconditioning and semi-coarsening. *Journal of Computational Physics*, 151:728–756, 1999.

[3]Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization*. Academic Press, San Diego, CA 92101, 1981.

[4]A. Jameson. Solution of the Euler equations by a multigrid method. *Applied Mathematics and Computations*, 13:327–356, 1983.

[5]A. Jameson. Transonic flow calculations. Princeton University Report MAE 1650, April 1984.

[6]A. Jameson. Multigrid algorithms for compressible flow calculations. In W. Hackbusch and U. Trottenberg, editors, *Lecture Notes in Mathematics, Vol. 1228*, pages 166–201. Proceedings of the 2nd European Conference on Multigrid Methods, Cologne, 1985, Springer-Verlag, 1986.

[7]A. Jameson. Analysis and design of numerical schemes for gas dynamics 1, artificial diffusion, upwind biasing, limiters and their effect on multigrid convergence. *Int. J. of Comp. Fluid Dyn.*, 4:171–218, 1995.

[8]J. F. Lynn. *Multigrid Solution of the Euler Equations with Local Preconditioning*. PhD thesis, University of Michigan, Ann Arbor, MI, 1995.

[9]L. Martinelli. Calculations of viscous flows with a multigrid method. *Ph. D. Dissertation*, Princeton University, Princeton, NJ, October 1987.

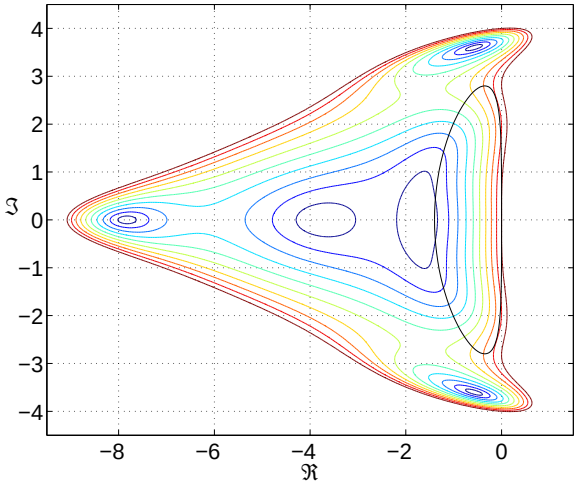
[10]L. Martinelli and A. Jameson. Validation of a multigrid method for the Reynolds averaged equations. *AIAA paper 88-0414*, 1988.

[11]Walter Murray Philip E. Gill and Michael A. Saunders. *User's Guide for SNOPT Version 6, A Fortran Package for Large-scale Nonlinear Programming*. Department of Mathematics, University of California, San Diego, CA, 92093-0112, December 2002.

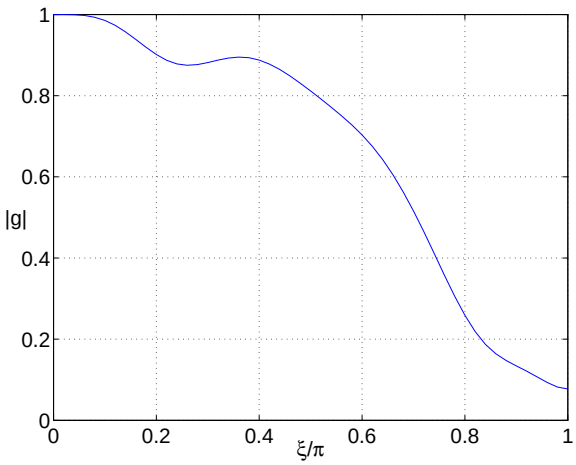
[12]N. A. Pierce. *Preconditioned Multigrid Methods for Compressible Flow Calculations on Stretched Meshes*. PhD thesis, University of Oxford, Oxford, UK, 1997.

[13]C. H. Tai. *Acceleration Techniques for Explicit Euler Codes*. PhD thesis, University of Michigan, Ann Arbor, MI, 1990.

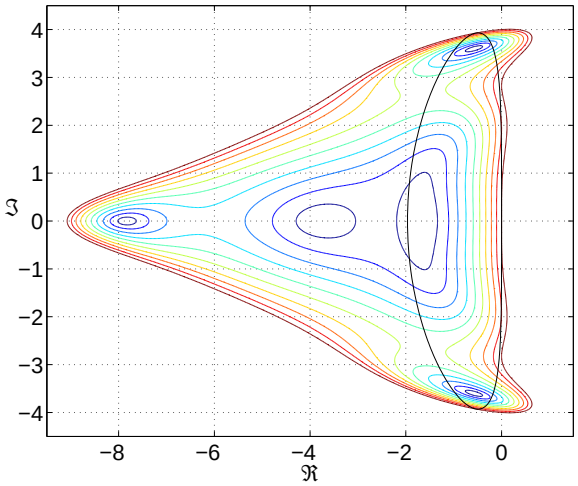
[14]E. Turkel. Preconditioning techniques in computational fluid dynamics. *Annu. Rev. Fluid Mech.*, 31:385–416, 1999.



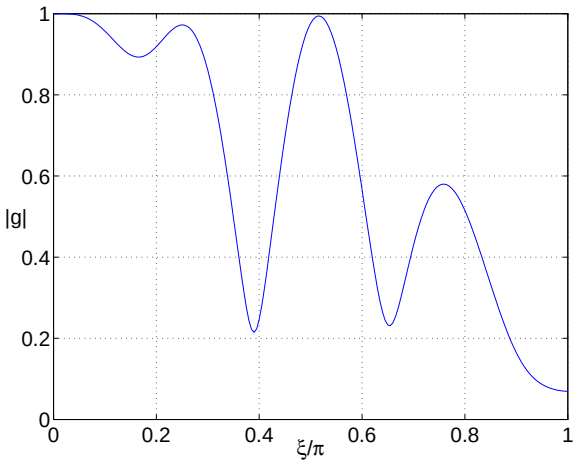
a) Stability domain and locus of z for $\lambda = 2.8$ and $\mu = \frac{1}{32}$



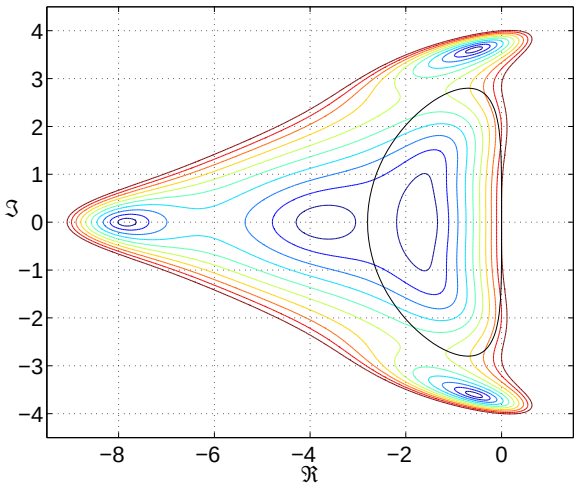
b) Amplification factor $|g|$ for $\lambda = 2.8$ and $\mu = \frac{1}{32}$



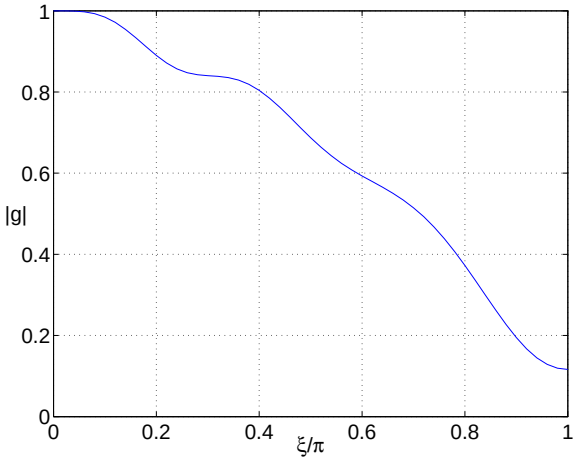
c) Stability domain and locus of z for $\lambda = 3.93$ and $\mu = \frac{1}{32}$



d) Amplification factor $|g|$ for $\lambda = 3.93$ and $\mu = \frac{1}{32}$



e) Stability domain and locus of z for $\lambda = 2.8$ and $\mu = \frac{1}{16}$



f) Amplification factor $|g|$ for $\lambda = 2.8$ and $\mu = \frac{1}{16}$

Fig. 1 Stability domain for the MJ coefficients and the influence of λ and μ on the amplification factor.

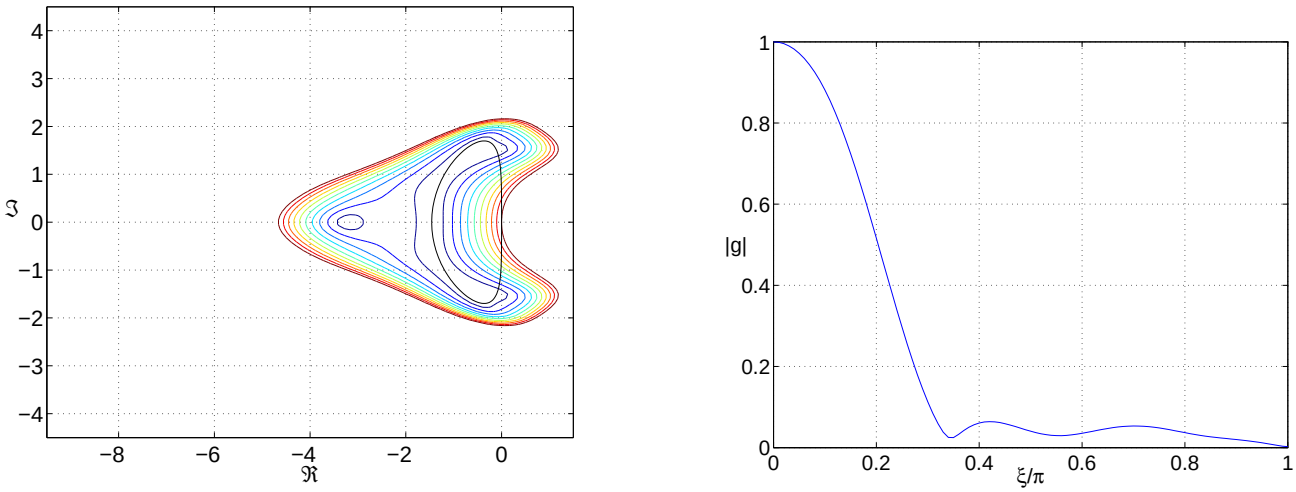


Fig. 2 Minimization of $\int_0^{\pi} |g(\xi)|\, d\xi$ with variable μ

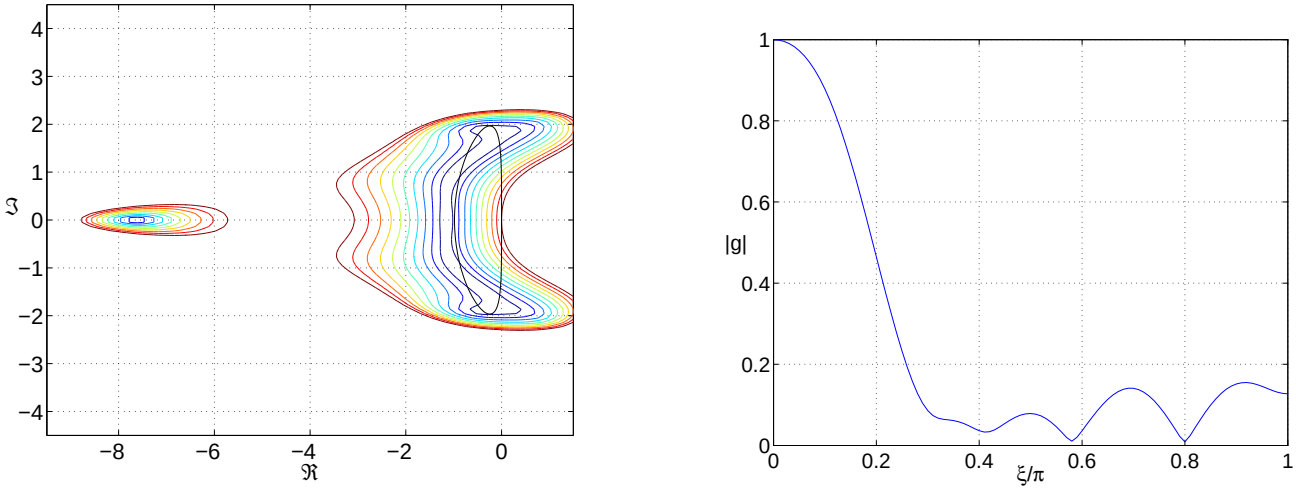


Fig. 3 Minimization of $\int_0^{\pi} |g(\xi)|\, d\xi$ with fixed μ

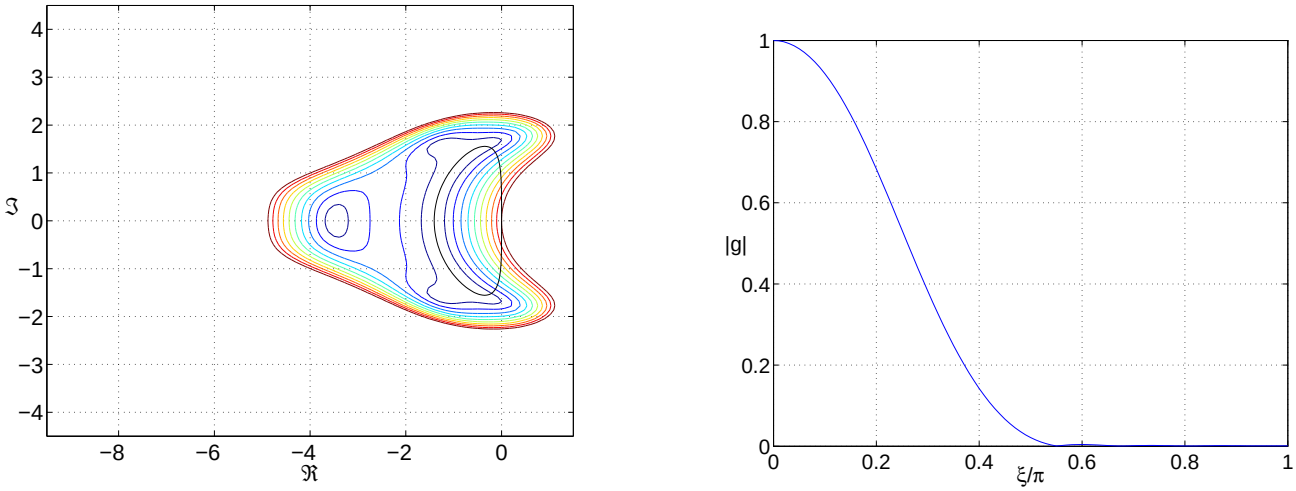


Fig. 4 Minimization of $\int_{\frac{\pi}{2}}^{\pi} |g(\xi)|\, d\xi$ with variable μ

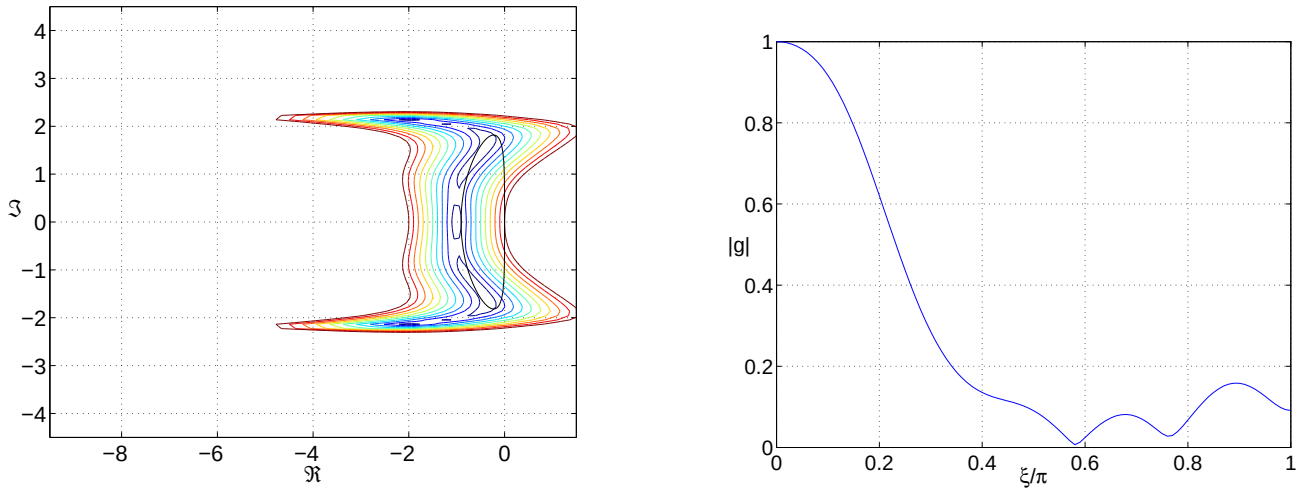


Fig. 5 Minimization of $\int_{\frac{\pi}{2}}^{\pi} |g(\xi)|\, d\xi$ with fixed μ

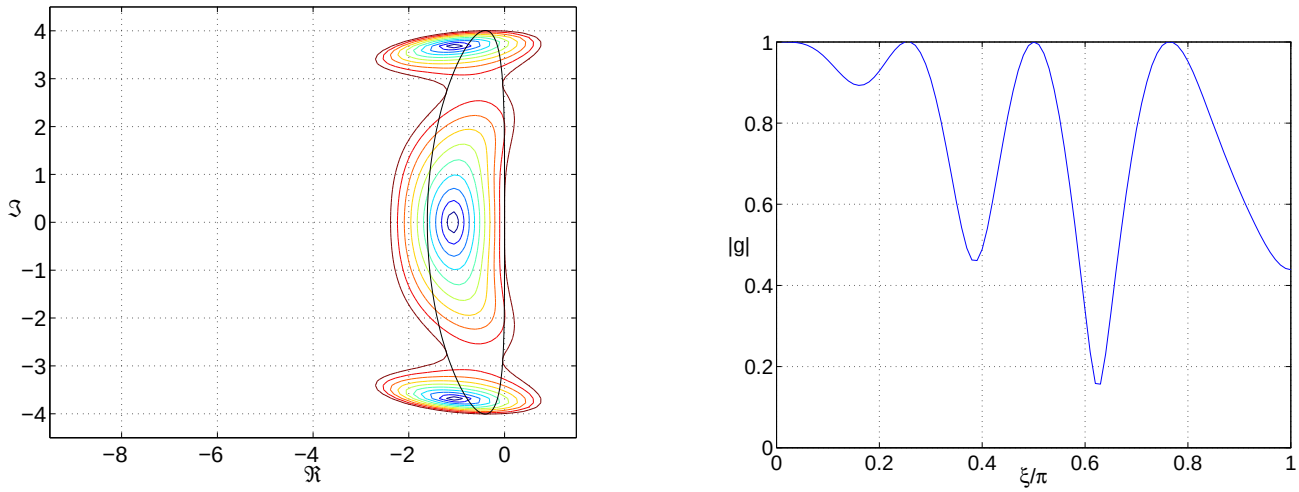


Fig. 6 Maximization of λ with variable μ

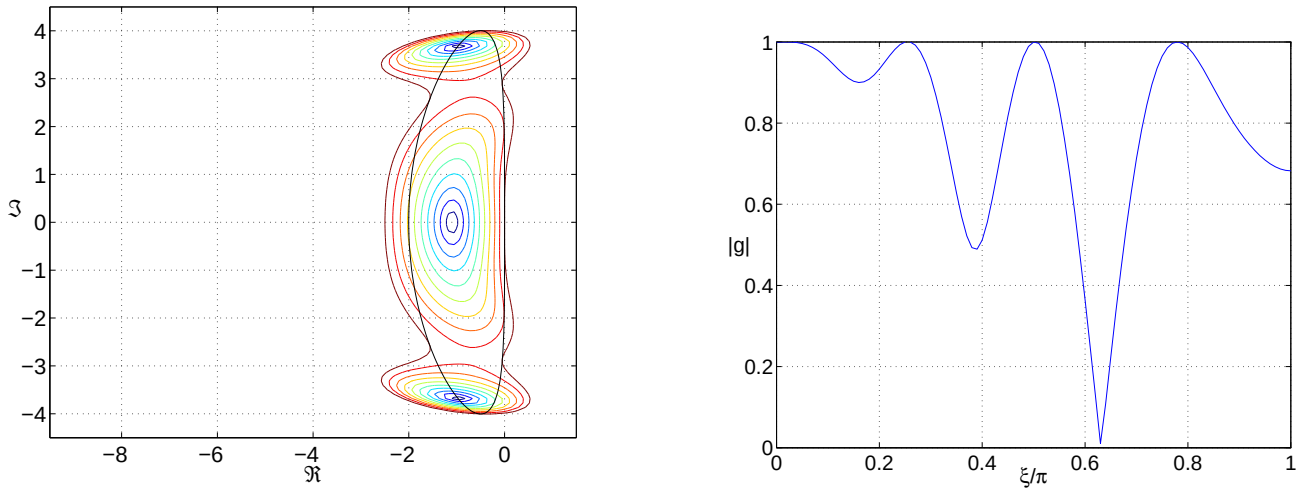


Fig. 7 Maximization of λ with fixed μ

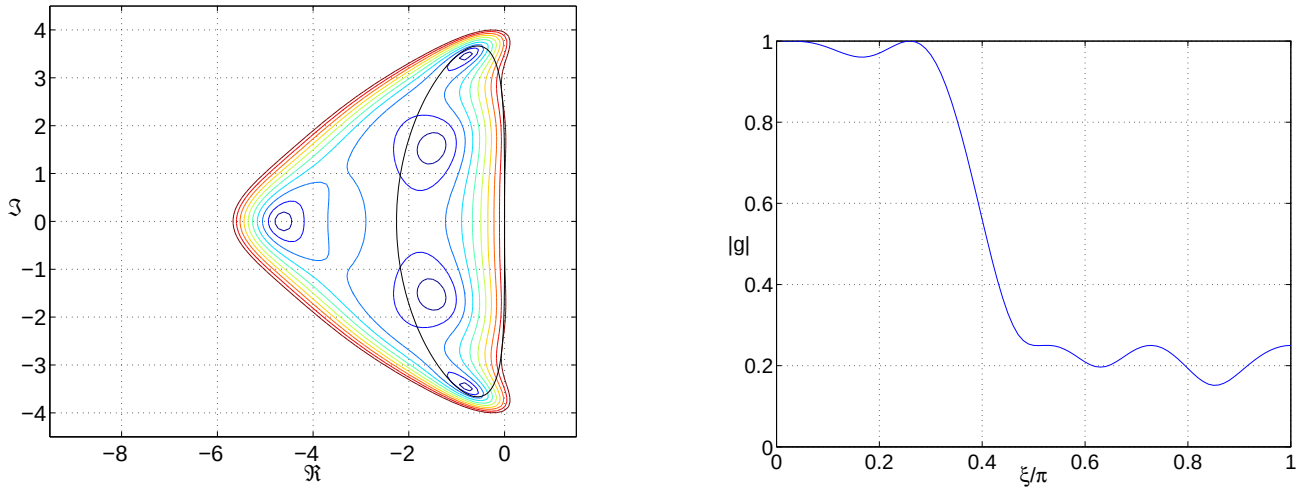


Fig. 8 Maximization of λ with $|g| \leq 0.25$ for high frequencies and variable μ

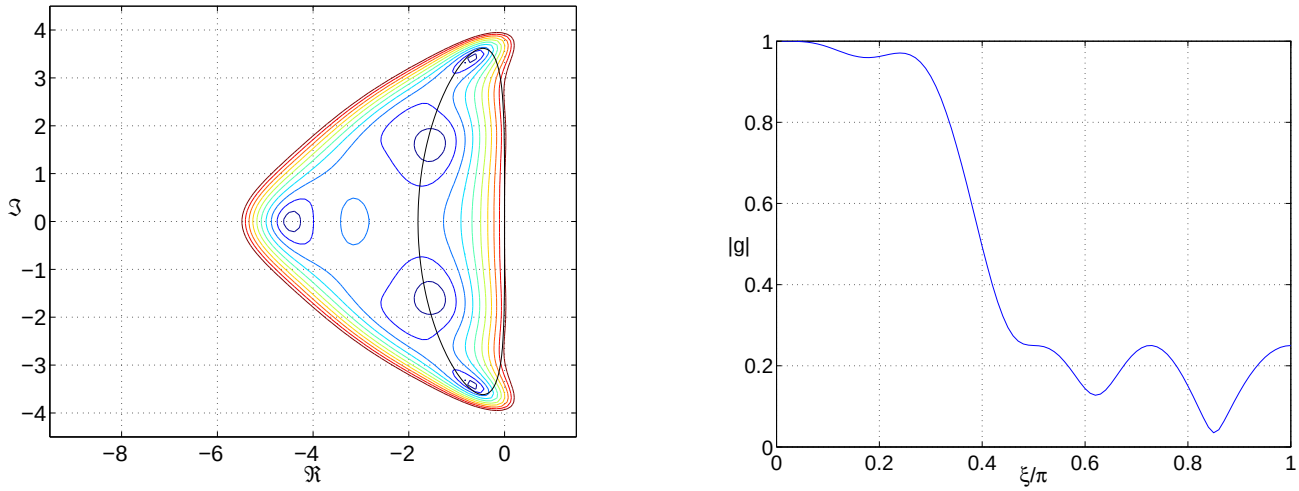


Fig. 9 Maximization of λ with $|g| \leq 0.25$ for high frequencies and fixed μ

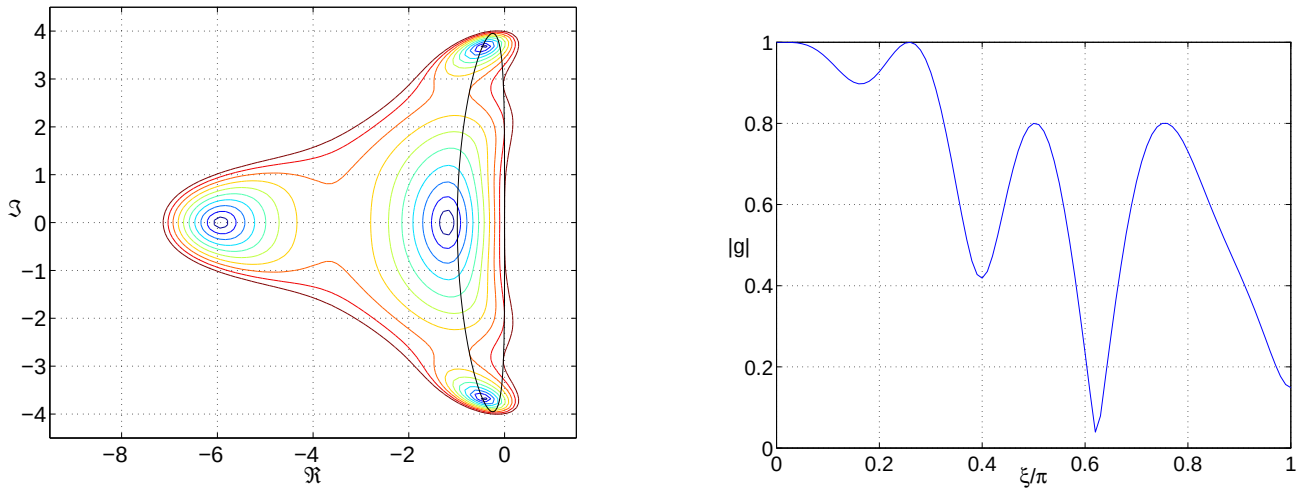


Fig. 10 Maximization of λ with $|g| \leq 0.8$ for high frequencies and variable μ

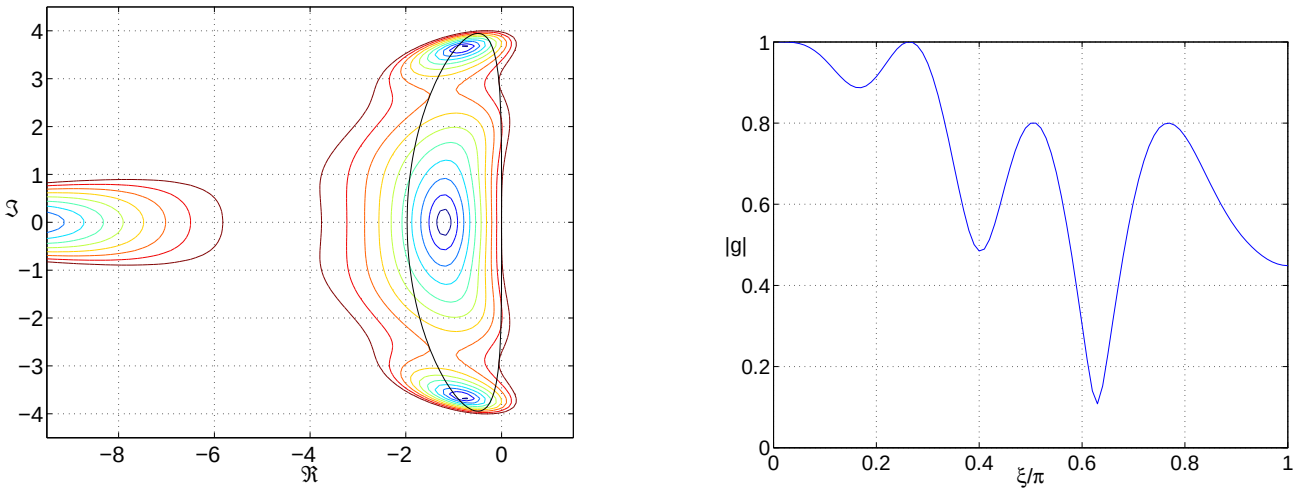


Fig. 11 Maximization of λ with $|g| \leq 0.8$ for high frequencies and fixed μ

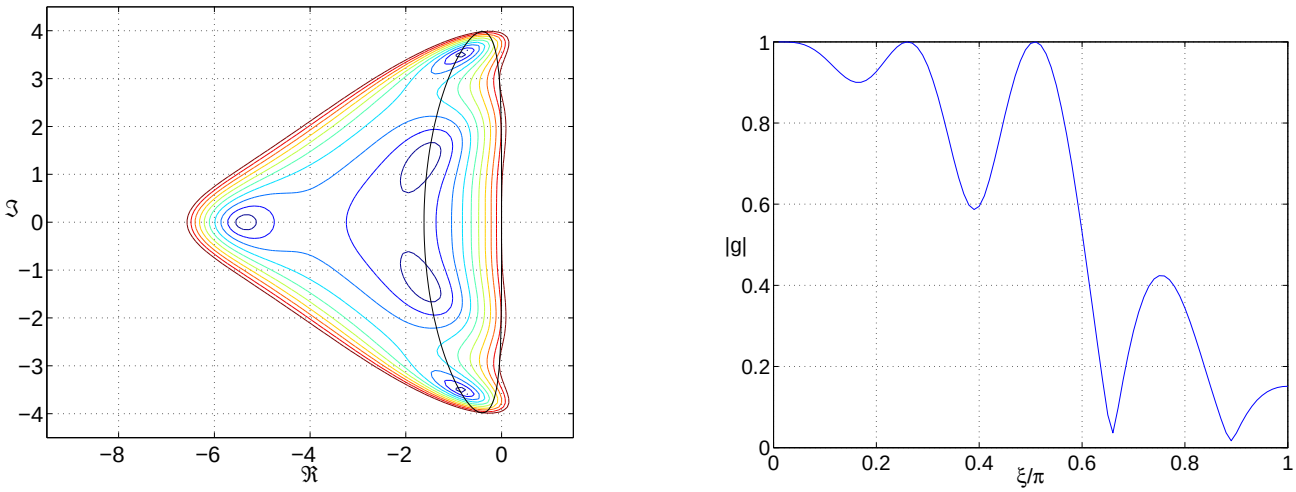


Fig. 12 Minimization of $\frac{\int_{\frac{\pi}{2}}^{\pi} |g(\xi)| d\xi}{\lambda^{25}}$ with variable μ

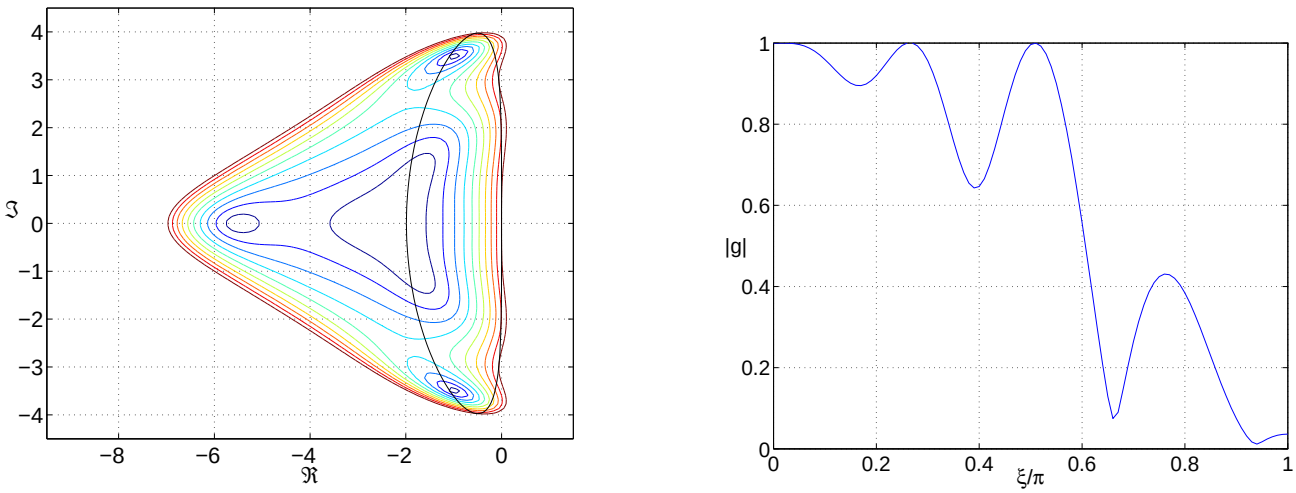


Fig. 13 Minimization of $\frac{\int_{\frac{\pi}{2}}^{\pi} |g(\xi)| d\xi}{\lambda^{25}}$ with fixed μ

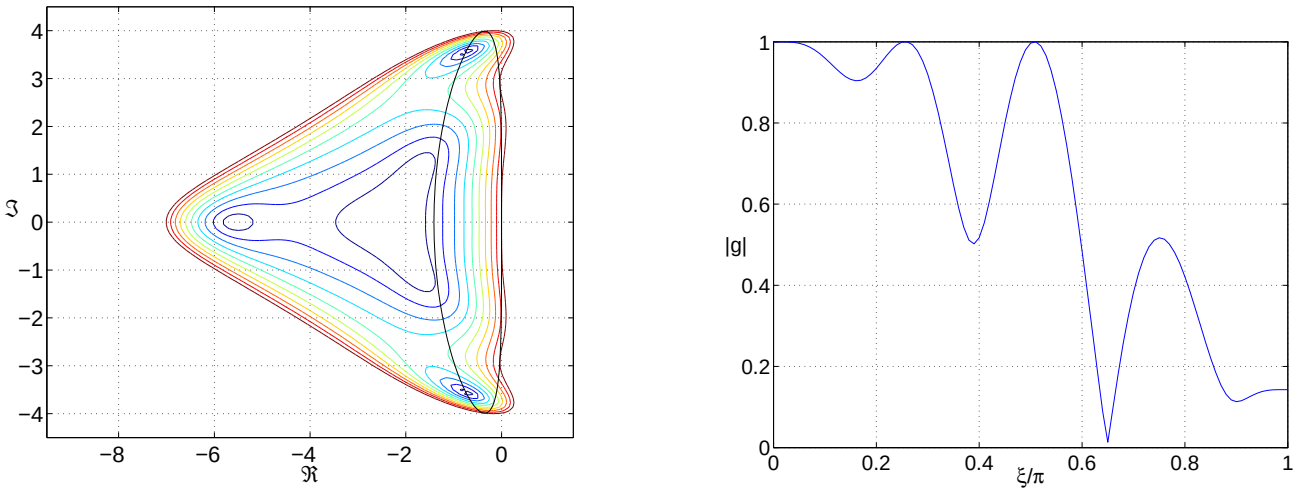


Fig. 14 Minimization of $\int_{\frac{\pi}{2}}^{\pi} |g(\xi)|\, d\xi$ with $\lambda \geq 3.99$ and variable μ

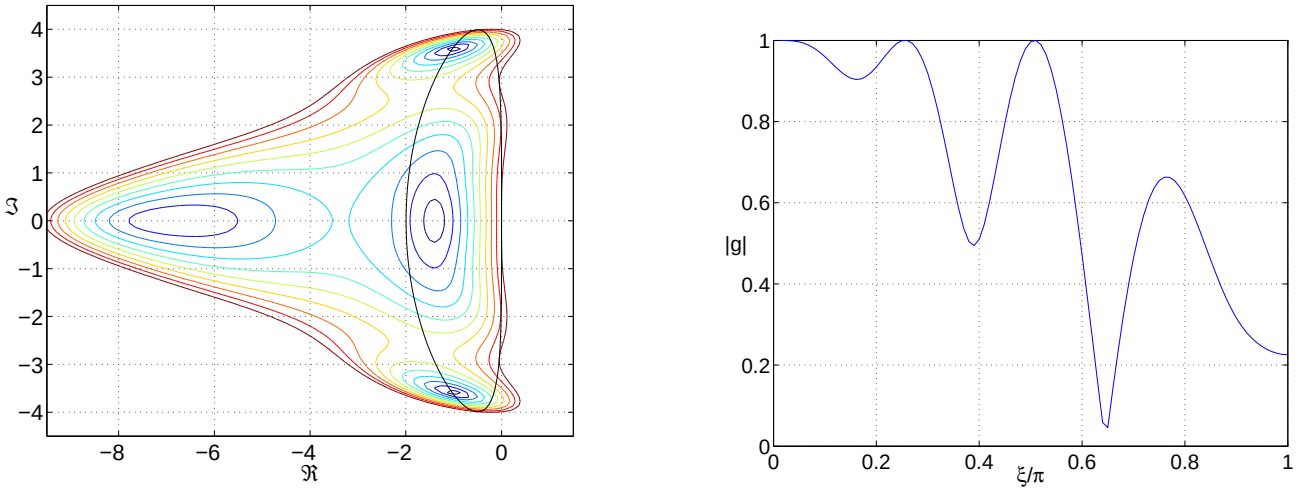


Fig. 15 Minimization of $\int_{\frac{\pi}{2}}^{\pi} |g(\xi)|\, d\xi$ with $\lambda \geq 3.99$ and fixed μ

Case	α_1	α_2	α_3	α_4	β_3	β_5	λ	$\frac{\mu}{\mu_0}$	$\lambda_{0.5}$	$Res_{0.5}$	$\lambda_{0.8}$	$Res_{0.8}$
MJ	0.2500	0.1667	0.3750	0.5000	0.5600	0.4400	3.9300	1.0000	3.9300	0.727E-12	3.9300	0.116E-09
1V	0.5602	0.2778	0.7612	0.8888	0.7719	0.0000	1.6977	1.7189	1.6977	0.135E-04	1.6977	0.210E-03
1F	0.6267	0.2320	0.7739	0.7688	0.4901	0.0000	1.9651	1.0000	1.9651	0.106E-07	1.9651	0.118E-05
2V	0.3323	0.3161	0.6032	0.8409	0.6483	0.0000	1.5555	1.8087	1.5555	0.928E-04	1.5555	0.281E-02
2F	0.6537	0.2359	0.7751	0.7109	0.0000	0.0000	1.8178	1.0000	1.8178	0.117E-07	1.8178	0.220E-05
3V	0.2421	0.1662	0.3746	0.4995	0.4046	0.0000	4.0066	0.8057	1.4000	0.398E-06	1.4500	0.116E-04
3F	0.2339	0.1679	0.3701	0.4995	0.4754	0.0000	4.0053	1.0000	1.4900	0.482E-07	1.5600	0.690E-05
4V	0.1953	0.1779	0.3394	0.4998	0.7793	0.7676	3.6648	1.2317	3.2200	0.333E-11	3.2600	0.625E-09
4F	0.1952	0.1833	0.3401	0.4998	0.7777	0.7928	3.6171	1.0000	3.6171	0.201E-11	3.6171	0.514E-09
5V	0.2391	0.1668	0.3728	0.4995	0.8403	0.0000	3.9522	0.5000	1.3500	0.104E-05	1.3800	0.128E-03
5F	0.2358	0.1658	0.3743	0.4995	0.6083	0.1129	3.9457	1.0000	3.4100	0.146E-10	3.3900	0.179E-08
6V	0.2251	0.1677	0.3675	0.4996	0.7343	0.6171	3.9816	0.8137	3.9816	0.264E-11	3.9816	0.386E-09
6F	0.2216	0.1676	0.3678	0.4996	0.7042	0.5340	3.9751	1.0000	3.9751	0.680E-12	3.9751	0.914E-10
7V	0.2308	0.1678	0.3679	0.4996	0.6993	0.5375	3.9900	0.7078	3.9900	0.368E-10	3.9900	0.939E-09
7F	0.2312	0.1679	0.3678	0.4996	0.5413	0.3222	3.9900	1.0000	3.9900	0.478E-12	3.9900	0.870E-10

Table 1 Results for optimizations and test cases

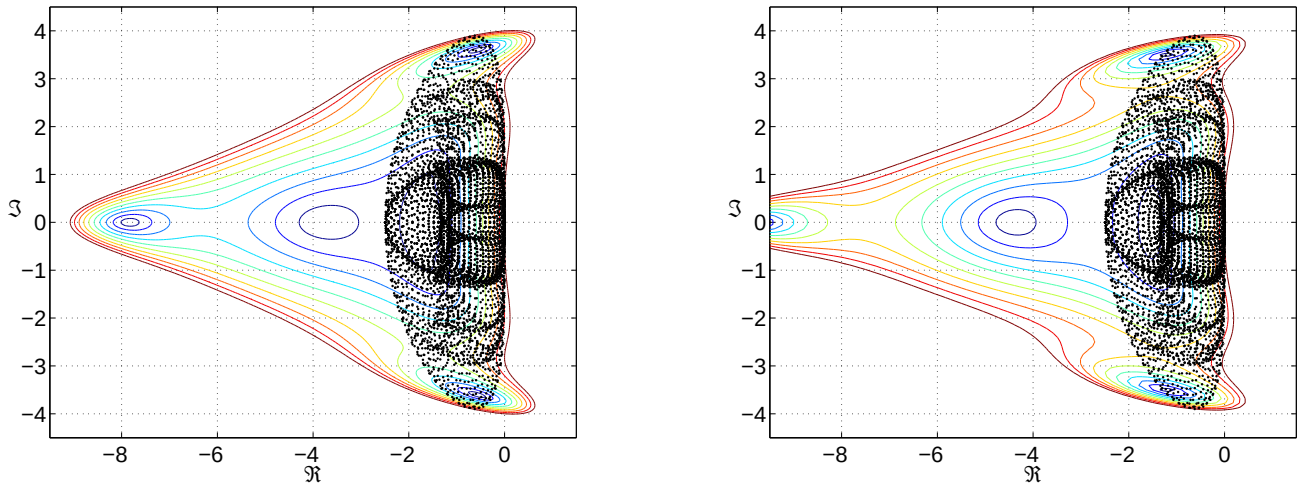


Fig. 16 $\lambda = 3.9$, $M = 0.5$, $\varphi = \frac{\pi}{3}$, **and** $A_R = 1$

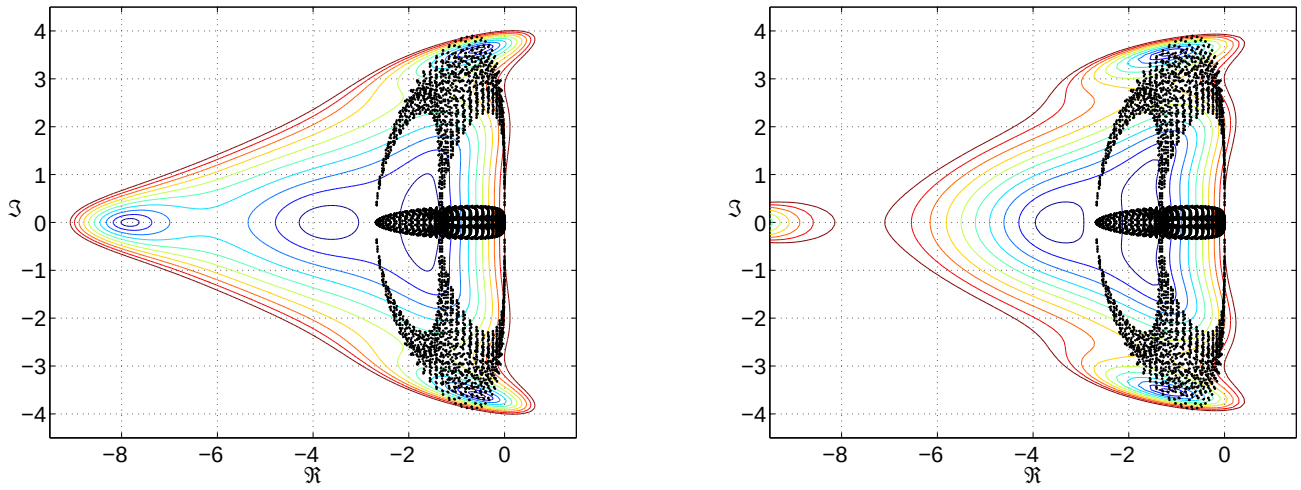


Fig. 17 $\lambda = 3.9$, $M = 0.1$, $\varphi = \frac{\pi}{3}$, **and** $A_R = 1$

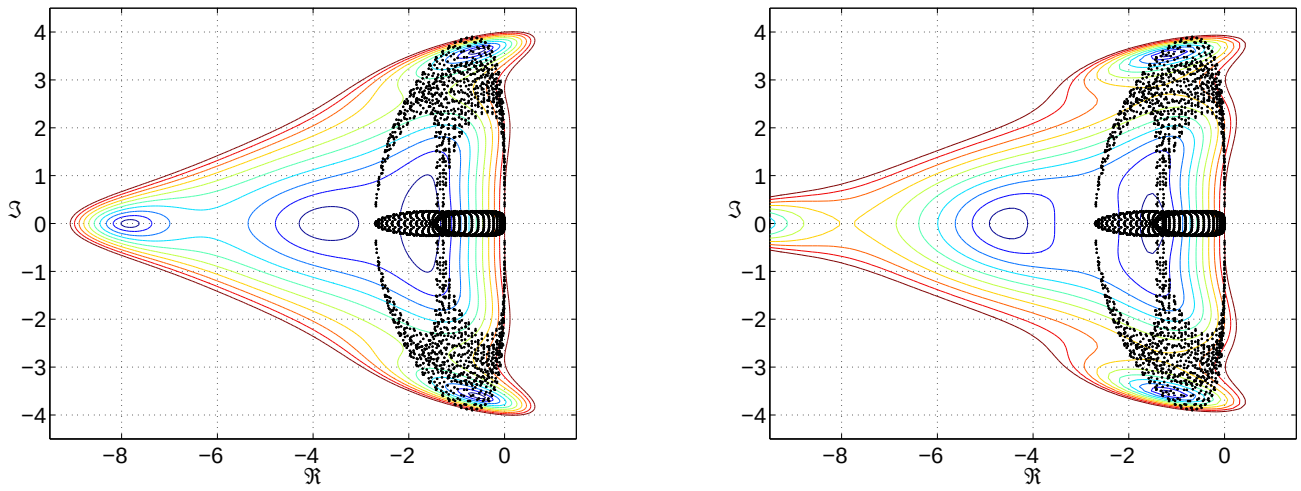


Fig. 18 $\lambda = 3.9$, $M = 0.1$, $\varphi = \frac{\pi}{2}$, **and** $A_R = 1$

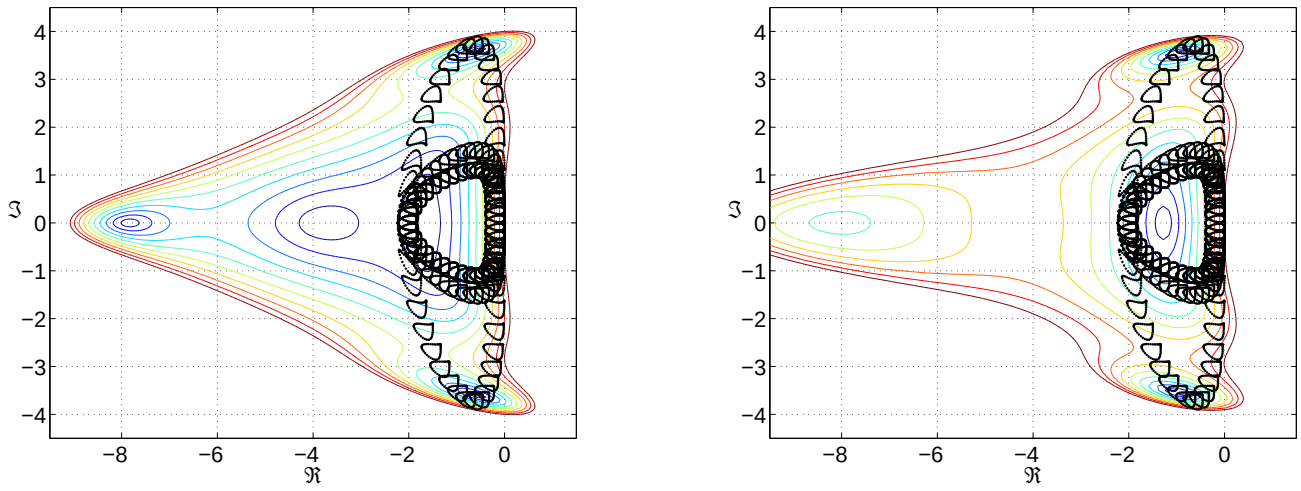


Fig. 19 $\lambda = 3.9$, $M = 0.5$, $\varphi = \frac{\pi}{3}$, **and** $A_R = 4$

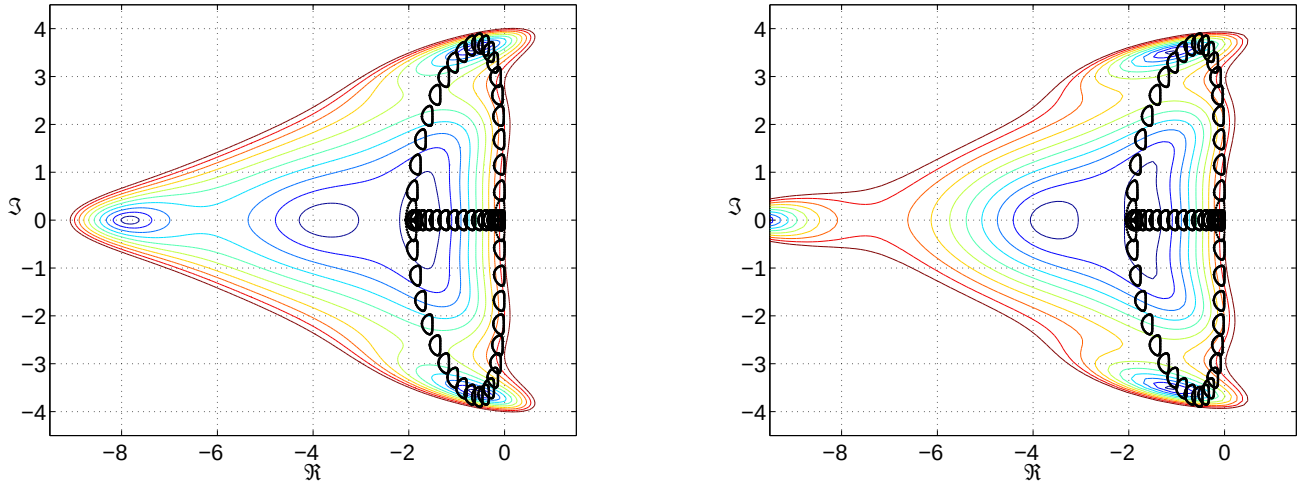


Fig. 20 $\lambda = 3.9$, $M = 0.9$, $\varphi = 0$, **and** $A_R = 16$

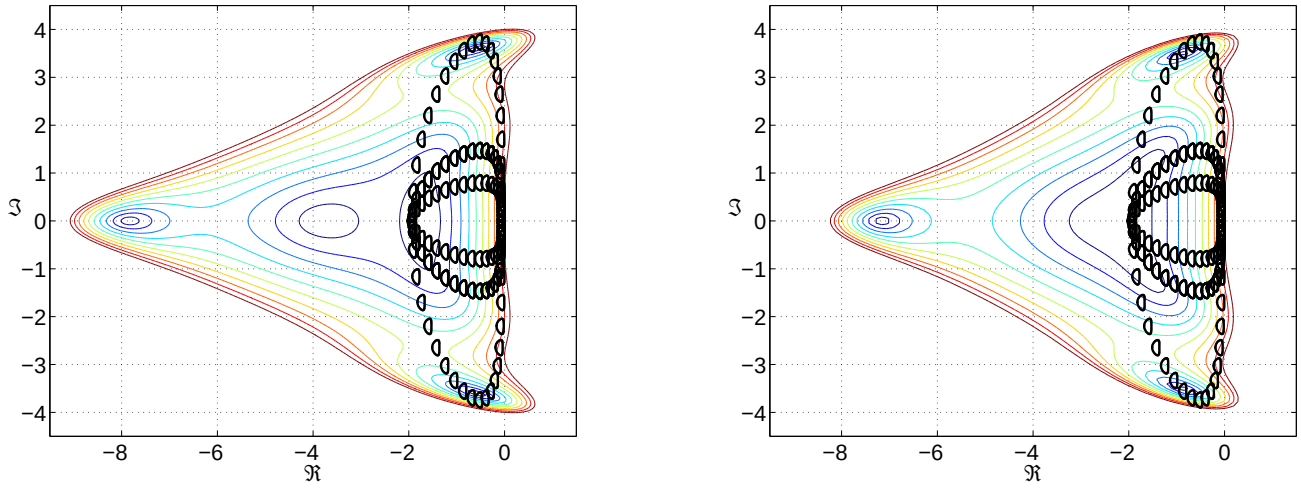


Fig. 21 $\lambda = 3.9$, $M = 1.3$, $\varphi = \frac{\pi}{6}$, **and** $A_R = 16$