Numerical solutions of the Falkner-Skan equation based on quasilinearization

Shengfeng Zhu [a,*], Xiaoliang Cheng [a], Qingbiao Wu [a]

[a] *Department of Mathematics, Zhejiang University, Hangzhou 310027, Zhejiang, P.R.China*

Abstract

We present two iterative methods for solving the Falkner-Skan equation based on the quasilinearization method. We formulate the original problem as a new free boundary value problem. The unknown free boundary depending on a small parameter is determined in iteratively. We apply the quasilinearization method to solve the nonlinear equation. Then we propose two different iterative approaches based on a cubic spline solver. The results of numerical experiments for various instances are compared with those reported previously in the literature. The comparisons show the accuracy, robustness and efficiency of the presented methodology.

Key words: Falkner-Skan equation; Free boundary value problem; Quasilinearization method; Cubic spline; Semi-infinite interval

1 Introduction

Generally, no analytical solutions are available for the nonlinear two-point boundary value problems (BVPs). Numerical solutions of these problems have always been of interest for scientists and engineers. The well-known nonlinear third-order Falkner-Skan equation is much more challenging since it is a BVP depicted on an infinite interval. This problem arises in the study of laminar boundary layers exhibiting similarity in fluid mechanics [1]. The solutions of the Falkner-Skan equation are similarity solutions of the two-dimensional incompressible laminar boundary layer equations. The Falkner-Skan equation is given by

$$f''' + \beta f f'' + \gamma[1 - (f')^2] = 0, \qquad 0 < \eta < \infty \tag{1}$$

* Corresponding author.
Email address: shengfengzhu@zju.edu.cn (Shengfeng Zhu).

subject to the boundary conditions

$$f(0) = f'(0) = 0, \tag{2}$$
$$f'(\infty) = 1, \tag{3}$$

where β and γ are constants and the prime stands for differentiation with respect to η. We can deduce easily that the solution of Eq. (1) satisfies the asymptotic condition

$$f''(\eta) \to 0 \quad as \quad \eta \to \infty. \tag{4}$$

Sometimes, the Falkner-Skan equation especially refers to the case when $\beta = 1$.

The first analytical treatment for the Falkner-Skan equation was given by Hartree [2], who found that in the region $-0.19884 < \gamma < 0$, there exists a family of unique solutions whose first order derivative $f'(\eta)$ tends to 1 exponentially. Weyl [3] proved that, for $\gamma > 0$, the problem has a solution $f(\eta)$ whose first derivative $f'(\eta)$ increases with η and whose second derivative $f''(\eta)$ tends decreasingly to zero as $\eta \to \infty$. Coppel [4] presented an elegant proof of the existence and uniqueness for $\gamma > 0$ and showed that the wall shear stress $\alpha := f''(0)$ is an increasing function of γ. In [5,6], extension for $\gamma < 0$ was discussed and properties of solutions were investigated further. New theoretical results for the solutions of the Falkner-Skan equation were given by Padé [7] and Yang [8].

Numerical techniques for the solution of the Falkner-Skan equation fall into the following classes in the literature: finite difference [2,5,9–11], shooting [12–17], finite element [18], group invariance theory [19,20], Chebyshev spectral method [21,22], differential transformation method [23], etc.

One problem for numerically solving the Falkner-Skan equation is how to deal with the infinite boundary. In the literature, there are various ways to overcome this difficulty. In [2,11,14], a finite prescribed truncated boundary instead of infinity was introduced directly. Asaithambi [9,12,13,18] truncated the infinite boundary by an unknown η_∞. The value for η_∞ is determined by imposing an asymptotic condition $f''(\eta_\infty) = 0$, which is an approximation of the condition (4). Fazio [20] proposed a novel approach to define the truncated boundary depending on a small parameter ϵ and formulate the Falkner-Skan equation as a free boundary value problem (FBVP). Recently, methods without truncating the infinite domain are developed, such as homotopy analysis method [24] and Adomian decomposition method [25], both of which aims to obtain series solutions and employ Padé approximation to enlarge the convergence domain and improve the accuracy of solutions.

We apply the treatment for the infinite interval proposed by Fazio [19,20] and solve instead a FBVP as an approximation to the Falkner-Skan equation. Consider the FBVP resulting from the Falkner-Skan equation:

$$\begin{cases} f_\epsilon''' + \beta f_\epsilon f_\epsilon'' + \gamma[1 - (f_\epsilon')^2] = 0, \\ f_\epsilon(0) = 0, \quad f_\epsilon'(0) = 0, \\ f_\epsilon'(\eta_\epsilon) = 1, \quad f_\epsilon''(\eta_\epsilon) = \epsilon, \end{cases} \tag{5}$$

where $0 < \epsilon \ll 1$. The free boundary η_ϵ can be considered as an unknown truncated boundary, which should be determined as part of solution. According to Theorem 1 [20], the solution of FBVP (5) converges uniformly to the solution of Falkner-Skan equation on arbitrarily large interval $[0, \eta_\epsilon]$ and the order of convergence is linear in ϵ at least. This result ensures theoretically the rationality for solving the FBVP instead of the original problem. The free truncated boundary η_ϵ, which is dependent on an arbitrary small parameter ϵ, converges to ∞ as ϵ tends to 0. Therefore, this treatment is more reasonable than other direct truncation methods, such as [9,12–14,18,21].

We wish to compute approximately the solution of the FBVP, which is still nonlinear. Mandelzweig et al. [26] applied a powerful iteration technique called quasilinearization method (QLM) to nonlinear ordinary differential equations in physics. As a generalization of Newton-Raphson method, the QLM's iterations are constructed to yield quadratic and monotonic convergence. We employ the QLM to transform FBVP (5) to a sequence of quasilinear third-order differential equation BVPs, which can be solved much easier than the original FBVP.

The remainder of the paper is organized as follows. In Section 2, using linear mappings, we transform the FBVP to a problem defined on $[0, 1]$. The QLM is employed to obtain the quasilinear form of the problem. Then we split it into two subproblems: a first-order initial value problem (IVP) and a second-order BVP. Section 3 is devoted to present a useful cubic spline solver for a class of two-point BVPs. In Section 4, we propose two different alternating iterative methods to solve the two problems. In every iteration, they both solve the second-order BVP by virtue of the cubic spline solver, but they differ in solving the IVP analytically or numerically. Then we present two corresponding algorithms. In Section 5, we apply our method to various cases of the Falkner-Skan equation by varying the values of β and γ. Finally, we use a brief conclusion to end the paper in Section 6.

2 Application of the QLM to the FBVP

In this section, we will apply the QLM [26] to the problem (5) so as to linearize the FBVP. In order to change the physical domain $[0, \eta_\epsilon]$ to a shorter interval $[0, 1]$, we use the linear mappings $x = \eta/\eta_\epsilon$ and $g = f_\epsilon/\eta_\epsilon$ in Ref. [18] to the FBVP. Then, (5) is transformed to

$$g''' + \beta\, \eta_\epsilon^2 g g'' + \gamma \eta_\epsilon^2 [1 - (g')^2] = 0, \tag{6}$$

$$g(0) = 0, \quad g'(0) = 0, \quad g'(1) = 1, \tag{7}$$

$$g''(1) = \epsilon \eta_\epsilon. \tag{8}$$

Considering the nonlinearity of problem (6)-(8), we solve it iteratively both for $g(x)$ and η_ϵ. We are interested in obtaining a proper value η_ϵ such that the solution $g(x)$ of (6)-(7) would satisfy the boundary condition (8), i.e.,

$$u(\eta_\epsilon) = 0, \tag{9}$$

where $u(\eta_\epsilon) := g''(1;\eta_\epsilon) - \epsilon\eta_\epsilon$ and $g''(1;\eta_\epsilon)$ denotes the value $g''(1)$ is dependent on η_ϵ.

Concretely, we will realize our idea in a nested loop. Our nested iteration framework consists of an outer iteration for η_ϵ and an inner iteration for $g(x)$. We denote by $\eta_\epsilon^{(k)}$ the kth iterate of η_ϵ. Let $g^{(k,n)}$ be the nth iterate for $g^{(k)}$ corresponding to $\eta_\epsilon^{(k)}$. For a given $\eta_\epsilon^{(k)}$, we can apply the QLM to the nonlinear problem (6)-(7) and obtain the $n+1$th iterative approximation $g^{(k,n+1)}(x)$ to the solution $g^{(k)}(x)$ in the inner iteration. More precisely, $g^{(k,n+1)}(x)$ is obtained by solving the BVP

$$\begin{cases} g^{(k,n+1)'''} + \beta\eta_\epsilon^{(k)^2} g^{(k,n)} g^{(k,n+1)''} - 2\gamma\eta_\epsilon^{(k)^2} g^{(k,n)'} g^{(k,n+1)'} \\ \qquad\qquad + \beta\eta_\epsilon^{(k)^2} g^{(k,n)''}\Big(g^{(k,n+1)} - g^{(k,n)}\Big) + \gamma\eta_\epsilon^{(k)^2}\Big[(g^{(k,n)'})^2 + 1\Big] = 0, \\ g^{(k,n+1)}(0) = 0, \quad g^{(k,n+1)'}(0) = 0, \quad g^{(k,n+1)'}(1) = 1, \end{cases} \tag{10}$$

where

$$g^{(k,n)'} := \frac{\mathrm{d}}{\mathrm{d}x} g^{(k,n)}, \quad g^{(k,n)''} := \frac{\mathrm{d}^2}{\mathrm{d}x^2} g^{(k,n)}, \quad g^{(k,n+1)'''} := \frac{\mathrm{d}^3}{\mathrm{d}x^3} g^{(k,n+1)}, \quad \eta_\epsilon^{(k)^2} := (\eta_\epsilon^{(k)})^2$$

and $n = 0, 1, 2, \ldots$.

According to [26], the zeroth iteration $g^{(k,0)}(x)$ should satisfy at least one of the boundary conditions. A successful initial guess can generate quadratic convergence for the quasi-linear iteration sequence $\{g^{(k,n)}(x)\}_{n=0}^{\infty}$. See [26] for the detailed proof.

In every iteration, such a third-order two-point BVP (10) can be solved analytically for $g^{(k,n+1)}$ with Green's function. However, the solution procedure is rather complicated after a few iterations. Thus, here we try to evaluate $g^{(k,n+1)}$ numerically instead. First, we split it into two subproblems. Set $y^{(k,n)} = g^{(k,n)'}$, (10) is transformed to a first-order IVP

$$\begin{cases} g^{(k,n)'} = y^{(k,n)}, \\ g^{(k,n)}(0) = 0, \end{cases} \tag{11}$$

and a second-order two-point BVP

$$\begin{cases} y^{(k,n+1)\prime\prime} + \beta\eta_\epsilon^{(k)^2} g^{(k,n)} y^{(k,n+1)\prime} - 2\gamma\eta_\epsilon^{(k)^2} y^{(k,n)} y^{(k,n+1)} + \beta\eta_\epsilon^{(k)^2} y^{(k,n)\prime}(g^{(k,n+1)} - g^{(k,n)}) \\ \quad + \gamma\eta_\epsilon^{(k)^2}\left[(y^{(k,n)})^2 + 1\right] = 0, \\ y^{(k,n+1)}(0) = 0, \quad y^{(k,n+1)}(1) = 1. \end{cases} \tag{12}$$

In the inner iteration, if $y^{(k,n)}$ is known, (11) can be solved either analytically by direct integration or approximated by numerically methods for such first-order IVPs. Considering the smooth property of spline approximation, we shall use a cubic spline solver described in the following section to solve the linear problem (12).

3 A cubic spline solver

This section is devoted to present a cubic spline solver for a class of two-point BVPs. Albasiny and Hoskins [27] derived the cubic spline approximation to a class of two-point BVPs

$$y'' + p(x)y' + q(x)y = r(x), \tag{13}$$
$$y(0) = a, \quad y(1) = b. \tag{14}$$

Consider the uniform grid partition $\Delta := \{0 = x_0 < x_1 < ... < x_N = 1\}$ of the interval $[0, 1]$ with mesh size $h = 1/N$ and the grid points $x_i = ih$, $i = 0, 1, \ldots, N$. The cubic spline $S(x)$ interpolating to the function $y(x)$ at the grid points is given by the piecewise expression

$$S(x) = \begin{cases} S_1(x), & \text{if} \quad x \in [x_0, x_1], \\ S_i(x), & \text{if} \quad x \in [x_{i-1}, x_i], \quad i = 2, 3, \ldots, N-1, \\ S_N(x), & \text{if} \quad x \in [x_{N-1}, x_N], \end{cases} \tag{15}$$

where

$$\begin{aligned} S_i(x) =& M_{i-1}\frac{(x_i - x)^3}{6h} + M_i\frac{(x - x_{i-1})^3}{6h} + \left(y_{i-1} - \frac{h^2}{6}M_{i-1}\right)\left(\frac{x_i - x}{h}\right) \\ &+ \left(y_i - \frac{h^2}{6}M_i\right)\left(\frac{x - x_{i-1}}{h}\right), \quad x \in [x_{i-1}, x_i], \end{aligned} \tag{16}$$

for $i = 1, 2, \ldots, N$; $M_i := S''(ih)$; $y_i := y(x_i)$. Hence,

$$S'(x_i^+) = -\frac{h}{3}M_i - \frac{h}{6}M_{i+1} + \frac{y_{i+1} - y_i}{h}, \quad i = 0, 1, \ldots, N-1 \tag{17}$$

and

$$S'(x_i^-) = \frac{h}{3}M_i + \frac{h}{6}M_{i-1} + \frac{y_i - y_{i-1}}{h}, \quad i = 1, 2, \ldots, N. \tag{18}$$

So the continuity of first derivatives at grid points implies that

$$\frac{h}{6}M_{i-1} - \frac{2h}{3}M_i + \frac{h}{6}M_{i+1} = \frac{y_{i-1} - 2y_i + y_{i+1}}{h}, \quad i = 1, 2, \ldots, N-1. \tag{19}$$

Then, by Eq. (17), (18) and (19), the requirement that the spline approximation should satisfy the differential equation (13) at the grid points leads to a set of relationships from which we can eliminate $M_0, M_1, \ldots, M_N$. More precisely, using an approach similar to one in [27], we obtain a tridiagonal system of equations written as

$$E_i y_{i-1} - F_i y_i + G_i y_{i+1} = H_i \qquad i = 1, 2, \ldots, N-1, \tag{20}$$

where

$$\begin{aligned}
E_i &= d_i\Big(1 - \frac{h}{2}p_{i-1} + \frac{h^2}{6}q_{i-1}\Big),\\
F_i &= c_i\Big(1 + \frac{h}{2}p_{i+1}\Big) + d_i\Big(1 - \frac{h}{2}p_{i-1}\Big) - \frac{2h^2}{3}q_i e_i,\\
G_i &= c_i\Big(1 + \frac{h}{2}p_{i+1} + \frac{h^2}{6}q_{i+1}\Big),\\
H_i &= \frac{h^2}{6}(d_i r_{i-1} + 4e_i r_i + c_i r_{i+1}),\\
c_i &= 1 - \frac{h}{3}p_{i-1} + \frac{h}{3}p_i - \frac{h^2}{12}p_{i-1}p_i,\\
d_i &= 1 - \frac{h}{3}p_i + \frac{h}{3}p_{i+1} - \frac{h^2}{12}p_i p_{i+1},\\
e_i &= 1 - \frac{h^2}{12}p_{i-1}p_{i+1} + \frac{7h}{24}(p_{i+1} - p_{i-1}),
\end{aligned}$$

for $i = 1, 2, \ldots, N-1$ and

$$p_i = p(x_i), \quad q_i = q(x_i), \quad r_i = r(x_i) \quad i = 0, 1, \ldots, N.$$

The boundary conditions (14) will correspond to

$$y_0 = a, \quad y_N = b. \tag{21}$$

Thus, substituting (21) into the first and last equation, the tridiagonal system (20) can be solved fast and efficiently by the famous Thomas algorithm, which is a direct method. Denote

$$\mathbf{p} := [p_0, p_1, \ldots, p_N]^{\mathrm{T}}, \quad \mathbf{q} := [q_0, q_1, \ldots, q_N]^{\mathrm{T}}, \quad \mathbf{r} := [r_0, r_1, \ldots, r_N]^{\mathrm{T}},$$

and

$$\mathbf{y} := [y_0, y_1, \ldots, y_N]^{\mathrm{T}}, \quad \mathbf{y}' := [y'_0, y'_1, \ldots, y'_N]^{\mathrm{T}}, \quad \mathbf{M} := [M_0, M_1, \ldots, M_N]^{\mathrm{T}}.$$

When $\mathbf{y}$ is determined, we can calculate $\mathbf{M}$ by the formulae

$$\begin{aligned} M_{i-1} =& \frac{1}{a_i}\Big\{(1+\frac{hp_i}{3})\Big[r_i - q_i y_i - \frac{p_i}{h}(y_i - y_{i-1})\Big] \\ &+ \frac{hp_{i-1}}{6}\Big[r_{i-1} - q_{i-1}y_{i-1} - \frac{p_{i-1}}{h}(y_i - y_{i-1})\Big]\Big\}, \quad i = 1, 2, \ldots, N \end{aligned} \tag{22}$$

and

$$\begin{aligned} M_{i+1} =& \frac{1}{b_i}\Big\{(1-\frac{hp_i}{3})\Big[r_{i+1} - q_{i+1} y_{i+1} - \frac{p_{i+1}}{h}(y_{i+1} - y_i)\Big] \\ &- \frac{hp_{i+1}}{6}\Big[r_i - q_i y_i - \frac{p_i}{h}(y_{i+1} - y_i)\Big]\Big\}, \quad i = 0, 1, \ldots, N-1. \end{aligned} \tag{23}$$

Thus, the expression of cubic spline function $S(x)$ on $[0, 1]$ is available. It approximates $y(x)$ at all points in $[0, 1]$ to $\mathcal{O}(h^4)$ if $y \in C^4([0, 1])$. Since $y'_i := y'(x_i) = S'(x_i)$ for $i = 0, 1, \ldots, N$, we get $\mathbf{y}'$ from (17) and (18).

Remark 1 *M_0 and M_N should be evaluated by (22) and (23), respectively, while M_1, M_2, $\ldots$, M_{N-1} could be obtained either by (22) or (23). Similarly, y'_0 and y'_N should be evaluated by (17) and (18), respectively, while y'_1, y'_2, $\ldots$, y'_{N-1} could be obtained either by (17) or (18).*

Remark 2 *Based on the analysis above, we can build without difficulty a cubic spline solver in the form of a function* ***CubicSplineBVP*** *(***p***, **q**, **r**, a, b) for solving the two-point BVP class (13)-(14). The output are the vectors* $\mathbf{y}$, $\mathbf{y}'$ *and* $\mathbf{M}$, *from which we also obtain the expression of $S(x)$.*

4 Two iterative methods based on a cubic spline solver

In this section, we will propose two different alternating iterative methods to solve the problem consisting of (11) and (12) by virtue of the cubic spline solver depicted above.

Let $y_i^{(k,n)} = y^{(k,n)}(x_i)$ for $i = 0, 1, ..., N$. Denote $\mathbf{y}^{(k,n)} := \left[y_0^{(k,n)}, y_1^{(k,n)}, \ldots, y_N^{(k,n)}\right]^{\mathrm{T}}$, $\mathbf{y}^{(k,n)\prime} := \left[y_0^{(k,n)\prime}, y_1^{(k,n)\prime} \ldots, y_N^{(k,n)\prime}\right]^{\mathrm{T}}$, and $\mathbf{g}^{(k,n)} := \left[g_0^{(k,n)}, g_1^{(k,n)}, \ldots, g_N^{(k,n)}\right]^{\mathrm{T}}$. All the notations here and in the following stand for the same quantities as those appearing before except for the two superscripts.

We now describe the alternating iterative methods for solving these two subproblems. For (11), if $y^{(k,n)}$ is an integrable function defined on $[0, 1]$, analytical treatment leads to $g^{(k,n)} = \int_0^x y^{(k,n)}(t)\,\mathrm{d}t$ from (11). Hence,

$$g_i^{(k,n)} = \int_0^{x_i} y^{(k,n)}(x)\,\mathrm{d}x \tag{24}$$

for $i = 1, 2, \ldots, N$. Thus, $g^{(k,n)}$ is available if $y^{(k,n)}$ is known.
Alternatively, for (11), we may adopt a numerical integration method, such as Heun's method. Then we have

$$\begin{cases} g_0^{(k,n)} = 0, \\ g_{i+1}^{(k,n)} = g_i^{(k,n)} + \dfrac{h}{2}\Big(y_i^{(k,n)} + y_{i+1}^{(k,n)}\Big), \end{cases} \tag{25}$$

for $i = 0, 1, \ldots, N-1$. Therefore, $\mathbf{g}^{(k,n)}$ is available if $\mathbf{y}^{(k,n)}$ is known.

Remark 3 *Heun's method used here is an explicit scheme. It is equivalent to the central difference discretization of (11) at $x = x_{i+1/2}$ for $i = 0, 1, \ldots, N-1$ (see [9,18]).*

For (12), when $\mathbf{g}^{(k,n)}$ is already known by (24) or (25), it is still difficult to construct iteration schemes from $y^{(k,n)}$ to $y^{(k,n+1)}$ in Eq. (12) directly since $y^{(k,n)\prime}$ and $g^{(k,n+1)}$ are unknown. We use an intermediate iteration step to obtain a function $g^{(k,n+\frac{1}{2})}$ as an approximation to $g^{(k,n+1)}$. To realize it, we first introduce three problems denoted by $\mathcal{Q}_1$, $\mathcal{Q}_2$ and $\mathcal{Q}_3$ as follows.

$$\mathcal{Q}_1 \begin{cases} y^{(k,n+\frac{1}{2})\prime\prime} + \beta\eta_\epsilon^{(k)^2} g^n y^{(k,n+\frac{1}{2})\prime} - 2\gamma\eta_\epsilon^{(k)^2} y^{(k,n)} y^{(k,n+\frac{1}{2})} + \gamma\eta_\epsilon^{(k)^2}\Big[(y^{(k,n)})^2 + 1\Big] = 0, \\ y^{(k,n+\frac{1}{2})}(0) = 0, \quad y^{(k,n+\frac{1}{2})}(1) = 1, \end{cases} \tag{26}$$

$$\mathcal{Q}_2 \begin{cases} g^{(k,n+\frac{1}{2})\prime} = y^{(k,n+\frac{1}{2})}, \\ g^{(k,n+\frac{1}{2})}(0) = 0, \end{cases} \tag{27}$$

$$\mathcal{Q}_3\begin{cases} y^{(k,n+1)\prime\prime}+\beta\eta_\epsilon^{(k)^2}g^{(k,n)}y^{(k,n+1)\prime}-2\gamma\eta_\epsilon^{(k)^2}y^{(k,n)}y^{(k,n+1)} \\ \qquad\qquad +\beta\eta_\epsilon^{(k)^2}y^{(k,n)\prime}\Big(g^{(k,n+\frac{1}{2})}-g^{(k,n)}\Big)+\gamma\eta_\epsilon^{(k)^2}\Big[(y^{(k,n)})^2+1\Big]=0, \\ y^{(k,n+1)}(0)=0, \quad y^{(k,n+1)}(1)=1. \end{cases} \tag{28}$$

In the following, we approximate $\mathcal{Q}_1$, $\mathcal{Q}_2$ and $\mathcal{Q}_3$ successively. Note that the forms of $\mathcal{Q}_1$ and $\mathcal{Q}_3$ belong to the BVP class (13)-(14). Thus, $\mathcal{Q}_1$ and $\mathcal{Q}_3$ can be solved by the cubic spline solver **CubicSplineBVP**.

For $\mathcal{Q}_1$,

$$p^{k,n}(x)=\beta\eta_\epsilon^{(k)^2}g^{(k,n)},\ q^{k,n}(x)=-2\gamma\eta_\epsilon^{(k)^2}y^{(k,n)},\ r^{k,n}(x)=-\gamma\eta_\epsilon^{(k)^2}\Big[(y^{(k,n)})^2+1\Big].$$

For $\mathcal{Q}_3$,

$$\widetilde{p}^{k,n}(x)=p^{k,n}(x),\widetilde{q}^{k,n}(x)=q^{k,n}(x),\widetilde{r}^{k,n}(x)=r^{k,n}(x)-\beta\eta_\epsilon^{(k)^2}y^{(k,n)\prime}\Big(g^{(k,n+\frac{1}{2})}-g^{(k,n)}\Big).$$

Now we present the iteration procedure from $\mathbf{g}^{(k,n)}$ to $\mathbf{y}^{(k,n+1)}$.

First, the cubic spline approximation $S^{(k,n+\frac{1}{2})}(x)$ for $\mathcal{Q}_1$, $\mathbf{y}^{(k,n+\frac{1}{2})}$, $\mathbf{y}^{(k,n+\frac{1}{2})\prime}$ and $\mathbf{M}^{(k,n+\frac{1}{2})}$ are all obtained by calling the function **CubicSplineBVP** ($\mathbf{p}^{k,n}$, $\mathbf{q}^{k,n}$, $\mathbf{r}^{k,n}$, 0, 1).

Second, $\mathbf{g}^{(k,n+\frac{1}{2})}$ can be obtained analytically by (24) or numerically by (25). If we want to solve $\mathcal{Q}_2$ analytically, we have

$$\begin{aligned} g_i^{(k,n+\frac{1}{2})}&=\int_0^{x_i}y^{(k,n+\frac{1}{2})}(x)\,\mathrm{d}x \\ &\approx\sum_{j=1}^{i}\int_{x_{j-1}}^{x_j}S_j^{(k,n+\frac{1}{2})}(x)\,\mathrm{d}x. \end{aligned}$$

Moreover, using (15), (22) and (23), we obtain

$$\begin{aligned} g_i^{(k,n+\frac{1}{2})}&=\sum_{j=1}^{i}\int_{x_{j-1}}^{x_j}\Big[M_{j-1}^{(k,n+\frac{1}{2})}\frac{(x_j-x)^3}{6h}+M_j^{(k,n+\frac{1}{2})}\frac{(x-x_{j-1})^3}{6h}+\Big(y_{j-1}^{(k,n+\frac{1}{2})} \\ &\quad-\frac{h^2}{6}M_{j-1}^{(k,n+\frac{1}{2})}\Big)\Big(\frac{x_j-x}{h}\Big)+\Big(y_j^{(k,n+\frac{1}{2})}-\frac{h^2}{6}M_j^{(k,n+\frac{1}{2})}\Big)\Big(\frac{x-x_{j-1}}{h}\Big)\Big]\,\mathrm{d}x \\ &=\sum_{j=1}^{i}\Big[\frac{h^3}{24}M_{j-1}^{(k,n+\frac{1}{2})}+\frac{h^3}{24}M_j^{(k,n+\frac{1}{2})}+\frac{h}{2}\Big(y_{j-1}^{(k,n+\frac{1}{2})}-\frac{h^2}{6}M_{j-1}^{(k,n+\frac{1}{2})}\Big) \\ &\quad+\frac{h}{2}\Big(y_j^{(k,n+\frac{1}{2})}-\frac{h^2}{6}M_j^{(k,n+\frac{1}{2})}\Big)\Big] \end{aligned} \tag{29}$$

$$= -\frac{h^3}{24}\sum_{j=1}^{i}\left(M_{j-1}^{(k,n+\frac{1}{2})} + M_j^{(k,n+\frac{1}{2})}\right) + \frac{h}{2}\sum_{j=1}^{i}\left(y_{j-1}^{(k,n+\frac{1}{2})} + y_j^{(k,n+\frac{1}{2})}\right)$$

for $i = 1, 2, \ldots, N$. If we approximate $\mathcal{Q}_2$ by Heun's method, we get

$$\begin{cases} g_0^{(k,n+\frac{1}{2})} = 0, \\ g_{i+1}^{(k,n+\frac{1}{2})} = g_i^{(k,n+\frac{1}{2})} + \frac{h}{2}\left(y_i^{(k,n+\frac{1}{2})} + y_{i+1}^{(k,n+\frac{1}{2})}\right), \end{cases} \tag{30}$$

for $i = 0, 1, \ldots, N-1$.

Last, if $\mathbf{g}^{(k,n+\frac{1}{2})}$ is available by (29) or (30), the cubic spline approximation $S^{(k,n+1)}(x)$ for $\mathcal{Q}_2$, $\mathbf{y}^{(k,n+1)}$, $\mathbf{y}^{(k,n+1)\prime}$ and $\mathbf{M}^{(k,n+1)}$ are all obtained by calling the function **CubicSplineBVP** ($\tilde{\mathbf{p}}^{k,n}$, $\tilde{\mathbf{q}}^{k,n}$, $\tilde{\mathbf{r}}^{k,n}$, 0, 1).

In the next iterative step, $\mathbf{g}^{(k,n+1)}$ can be obtained, similar as (29) or (30), by

$$g_i^{(k,n+1)} = -\frac{h^3}{24}\sum_{j=1}^{i}\left(M_{j-1}^{(k,n+1)} + M_j^{(k,n+1)}\right) + \frac{h}{2}\sum_{j=1}^{i}\left(y_{j-1}^{(k,n+1)} + y_j^{(k,n+1)}\right) \tag{31}$$

for $i = 1, 2, \ldots, N$ or

$$\begin{cases} g_0^{(k,n+1)} = 0, \\ g_{i+1}^{(k,n+1)} = g_i^{(k,n+1)} + \frac{h}{2}\left(y_i^{(k,n+1)} + y_{i+1}^{(k,n+1)}\right) \end{cases} \tag{32}$$

for $i = 0, 1, \ldots, N-1$.

For solving $\mathcal{Q}_1$, $\mathcal{Q}_2$ and $\mathcal{Q}_3$ iteratively, the iterations described above will continue until

$$\left\|\mathbf{y}^{(k,n+1)} - \mathbf{y}^{(k,n)}\right\| < \varepsilon_1 \quad \text{or} \quad \left\|\mathbf{g}^{(\mathrm{k,n+1})} - \mathbf{g}^{(\mathrm{k,n})}\right\| < \varepsilon_1 \tag{33}$$

for some prescribed error tolerance ε_1.

Now we discuss the outer iteration loop for η_ϵ. We assume that condition (33) is satisfied corresponding to some k and denote $\tilde{g}^{(k)}$ and $\tilde{y}^{(k)}$ the converged values for $g^{(k,n)}$ and $y^{(k,n)}$, respectively. η_ϵ may be calculated by solving the nonlinear Eq. (11) by means of an iterative technique such as Newton's method or secant method. Since $u(\eta_\epsilon)$ and its derivative are not known explicitly, we will use secant method for Eq. (11) to obtain iterate for η_ϵ. First we have

$$u(\eta_\epsilon^{(k)}) = g''(1;\eta_\epsilon^{(k)}) - \epsilon\eta_\epsilon^{(k)} = y'(1;\eta_\epsilon^{(k)}) - \epsilon\eta_\epsilon^{(k)} \tag{34}$$

where

$$y'(1;\eta_\epsilon^{(k)}) = \tilde{y}_N^{(k)\prime} = \frac{h}{3}\widetilde{M}_N^k + \frac{h}{6}\widetilde{M}_{N-1}^k + \frac{\tilde{y}_N^k - \tilde{y}_{N-1}^k}{h}. \tag{35}$$

Then, the secant iterate is obtained using

$$\eta_\epsilon^{(k+1)} = \eta_\epsilon^{(k)} + \Delta\eta_\epsilon^{(k)}, \tag{36}$$

where

$$\Delta\eta_\epsilon^{(k)} = -u(\eta_\epsilon^{(k)}) \left[\frac{\eta_\epsilon^{(k)} - \eta_\epsilon^{(k-1)}}{u\left(\eta_\epsilon^{(k)}\right) - u\left(\eta_\epsilon^{(k-1)}\right)} \right]. \tag{37}$$

The iterative procedure is terminated when the condition

$$\left| u(\eta_\epsilon^{(k)}) \right| < \varepsilon_2 \tag{38}$$

is satisfied for some prescribed tolerance ε_2. We will assume that we begin our computations with initial guesses $\eta_\epsilon^{(0)}$ and $\eta_\epsilon^{(1)}$ for η_ϵ.

Now we are ready to present the algorithms of our numerical methods for the Falkner-Skan equation. In our iterative procedure, solving first-order IVPs analytically or numerically leads to two different algorithms below.

Algorithm 1
Step 1. *Input ϵ, ε_1, ε_2, $\eta_\epsilon^{(0)}$, $\eta_\epsilon^{(1)}$, h, and an initial function $y^{(0,0)}(x)$; $N \leftarrow 1/h$.*
Step 2. *$k \leftarrow 1$, repeat through step 8 until (38) is satisfied.*
Step 3. *$n \leftarrow 0$, $\{y_i^{(k,n)}\}_{i=1}^{N-1} \leftarrow \{y^{(0,0)}(x_i)\}_{i=1}^{N-1}$, fix $y_0 \leftarrow 0$ and $y_N \leftarrow 1$; repeat through step 7 until (33) is satisfied.*
Step 4. *If $n = 0$, compute $\mathbf{g}^{(k,n)}$ using (24); else compute $\mathbf{g}^{(k,n)}$ using (31).*
Step 5. *Obtain $\mathbf{y}^{(k,n+\frac{1}{2})}$ and $\mathbf{M}^{(k,n+\frac{1}{2})}$ by **CubicSplineBVP** ($\mathbf{p}^{k,n}$, $\mathbf{q}^{k,n}$, $\mathbf{r}^{k,n}$, 0, 1).*
Step 6. *Compute $\mathbf{g}^{(k,n+\frac{1}{2})}$ by (29).*
Step 7. *Obtain $\mathbf{y}^{(k,n+1)}$, $\mathbf{y}^{(k,n+1)\prime}$ and $\mathbf{M}^{(k,n+1)}$ by **CubicSplineBVP** ($\tilde{\mathbf{p}}^{k,n}$, $\tilde{\mathbf{q}}^{k,n}$, $\tilde{\mathbf{r}}^{k,n}$, 0, 1); $n \leftarrow n+1$.*
Step 8. *Compute $u(\eta_\epsilon^{(k)})$, $\Delta\eta_\epsilon^{(k)}$ and $\eta_\epsilon^{(k+1)}$ by (34) and (36); $k \leftarrow k+1$.*

Algorithm 2
Step 1. *Input ϵ, ε_1, ε_2, $\eta_\epsilon^{(0)}$, $\eta_\epsilon^{(1)}$, h and an initial vector $\mathbf{y}^{(0,0)}$ with $y_0^{(0,0)} \leftarrow 0$ and $y_N^{(0,0)} \leftarrow 1$; $N \leftarrow 1/h$.*
Step 2. *$k \leftarrow 1$, repeat through step 8 until (38) is satisfied.*

Step 3. $n \leftarrow 0$, $\mathbf{y}^{(k,n)} \leftarrow \mathbf{y}^{(0,0)}$*; repeat through step 7 until (33) is satisfied.*
Step 4. Compute $\mathbf{g}^{(k,n)}$ *using (25);*
Step 5. Obtain $\mathbf{y}^{(k,n+\frac{1}{2})}$ *by* ***CubicSplineBVP*** *(*$\mathbf{p}^{k,n}$, $\mathbf{q}^{k,n}$, $\mathbf{r}^{k,n}$, *0, 1).*
Step 6. Compute $\mathbf{g}^{(k,n+\frac{1}{2})}$ *by (30).*
Step 7. Obtain $\mathbf{y}^{(k,n+1)}$ *and* $\mathbf{y}^{(k,n+1)\prime}$ *by* ***CubicSplineBVP*** *(*$\widetilde{\mathbf{p}}^{k,n}$, $\widetilde{\mathbf{q}}^{k,n}$, $\widetilde{\mathbf{r}}^{k,n}$, *0, 1);* $n \leftarrow n+1$.
Step 8. Compute $u(\eta_\epsilon^{(k)})$, $\Delta\eta_\epsilon^{(k)}$ *and* $\eta_\epsilon^{(k+1)}$ *by (34) and (36);* $k \leftarrow k+1$.

Remark 4 *In* `Algorithm 2`*, Heun's method could be replaced by other high order methods such as Runge-Kutta method to improve accuracy.*

Hereafter, we use the tilde notation "$\sim$" placed on top of a symbol to indicate its corresponding final converged quantity. Let $\tilde{S}(x)$ be the final converged cubic spline approximation obtained by `Algorithm 1` or `Algorithm 2`. Then we get its expression as follows:

$$\begin{aligned}
\tilde{g}_i(x) &= \tilde{g}_{i-1} + \int_{x_{i-1}}^{x} \tilde{S}_i(t)\,\mathrm{d}t \\
&= \tilde{g}_{i-1} + \frac{h^2-(x_i-x)^2}{2h}\tilde{y}_{i-1} + \frac{(x-x_{i-1})^2}{2h}\tilde{y}_i \\
&\quad - \frac{[h^2-(x_i-x)^2]^2}{24h}\widetilde{M}_{i-1} + \frac{(x-x_{i-1})^4 - 2h^2(x-x_{i-1})^2}{24h}\widetilde{M}_i \\
&= \tilde{g}_{i-1} + \frac{\tilde{y}_{i-1}}{2}h - \frac{\widetilde{M}_{i-1}}{24}h^3 + \frac{6\tilde{y}_i - \widetilde{M}_i h^2}{12h}(x-x_{i-1})^2 \\
&\quad - \frac{6\tilde{y}_{i-1} - \widetilde{M}_{i-1}h^2}{12h}(x-x_i)^2 + \frac{\widetilde{M}_i}{24h}(x-x_{i-1})^4 - \frac{\widetilde{M}_{i-1}}{24h}(x-x_i)^4
\end{aligned} \tag{39}$$

$x \in [x_{i-1}, x_i]$, for $i = 1, 2, \ldots, N$. Let

$$\tilde{g}(x) = \begin{cases} \tilde{g}_1(x), & \text{if } \ x \in [x_0, x_1], \\ \tilde{g}_i(x), & \text{if } \ x \in [x_{i-1}, x_i], \quad i = 2, 3, \ldots, N-1, \\ \tilde{g}_N(x), & \text{if } \ x \in [x_{N-1}, x_N], \end{cases} \tag{40}$$

then we have

$$f(\eta) \approx f_\epsilon(\eta) = \tilde{\eta}_\epsilon \tilde{g}(x) = \tilde{\eta}_\epsilon \tilde{g}(\eta/\tilde{\eta}_\epsilon). \tag{41}$$

Notice that the final expression (41) is a fourth degree polynomial approximation for the Falkner-Skan equation.

Consequently, by (41), we get

$$f''(\eta) \approx f''_\epsilon(\eta) = \frac{\tilde{g}''(\eta/\tilde{\eta}_\epsilon)}{\tilde{\eta}_\epsilon}, \tag{42}$$

$$f''(\eta_i) \approx \frac{\tilde{g}''_i(x_{i-1})}{\tilde{\eta}_\epsilon} = \frac{1}{\tilde{\eta}_\epsilon}\Big(-\frac{h}{3}\widetilde{M}_{i-1} - \frac{h}{6}\widetilde{M}_i + \frac{\tilde{y}_{i-1}-\tilde{y}_{i-1}}{h}\Big), \tag{43}$$

for $i = 1, 2, \ldots, N$ and obtain the wall shear stress α approximately by

$$\alpha_\epsilon := f''_\epsilon(0) = \frac{1}{\tilde{\eta}_\epsilon}\Big(-\frac{h}{3}\widetilde{M}_0 - \frac{h}{6}\widetilde{M}_1 + \frac{\tilde{y}_1-\tilde{y}_0}{h}\Big), \tag{44}$$

where

$$\begin{aligned}\widetilde{M}_i =& \frac{1}{a_{i+1}}\Big\{\Big(1+\frac{hp_{i+1}}{3}\Big)\Big[r_i - q_i\tilde{y}_i - \frac{p_i}{h}(\tilde{y}_{i+1}-\tilde{y}_i)\Big] \\ &+ \frac{hp_i}{6}\Big[r_{i+1} - q_{i+1}\tilde{y}_{i+1} - \frac{p_{i+1}}{h}(\tilde{y}_{i+1}-\tilde{y}_i)\Big]\Big\}, \quad i = 0, 1.\end{aligned} \tag{45}$$

Remark 5 *Since $\tilde{g}''(0) = \tilde{S}'(0)$, by (15), we can also obtain the formula for computing α_ϵ exactly the same as (44).*

5 Numerical results

In this section, numerous cases of the Falkner-Skan equation are solved numerically by our methods. We observe, through many experiments, that `Algorithm 2` has almost the same accuracy as `Algorithm 1` but is more efficient than `Algorithm 1` in implementation. This is also why we use Heun's method rather than Runge-Kutta method in `Algorithm 2`. For the sake of simplicity, we only list numerical results obtained by `Algorithm 2`. Comparisons are made between our numerical results and those reported previously in the literature.

5.1 Case 1: Blasius flow($\beta = 1/2, \gamma = 0$)

When $\beta = 1/2$ (or $\beta = 1$) and$\gamma = 0$, we have the so-called Blasius flow. Blasius [28] obtained this problem in the study of a laminar boundary layer along a thin flat plate. The Blasius flow has been either studied individually in [29–31] and the references therein or tested as a special case of Falkner-Skan equation in [13,32], etc.

Generally, in order to obtain an idea of the accuracy involved, the value of the wall shear stress is of physical interest since it defines the skin friction around the plate. Table 1 shows the comparison of the wall shear stress and the truncated boundary obtained by the proposed method to those obtained by [32]. We choose $h = 10^{-3}$, $\varepsilon_1 = 10^{-11}$ and $\epsilon =$

$\varepsilon_2, 10\varepsilon_2$ for various ε_2. Denote $\alpha_{\epsilon,h}$ and $\eta_{\epsilon,h}$ the values of wall shear stress and truncated boundary corresponding to ϵ and h, respectively. Then increased accuracy is accomplished by using extrapolation formulas $\alpha_\epsilon = (4\alpha_{\epsilon,h/2} - \alpha_{\epsilon,h})/3$ and $\eta_\epsilon = (4\eta_{\epsilon,h/2} - \eta_{\epsilon,h})/3$. Note that the present results for α_ϵ are also in good agreement with those reported in [13] and [29].

We estimate the order of accuracy for calculation of α_ϵ by grid refinement analysis. Let α^* denote the exact value of α. Notice that the α^* for Blasius flow is unknown, we estimate the order of accuracy by

$$\text{order} = \log_2 \left(\frac{\alpha_{\epsilon,h/2} - \alpha_{\epsilon,h}}{\alpha_{\epsilon,h/4} - \alpha_{\epsilon,h/2}} \right). \tag{46}$$

Shown in Table 2 are results with $\varepsilon_1 = 10^{-11}$ and $\varepsilon_2 = 10^{-9}$. In Fig. 1, we plot the curves of $\mathrm{d}f_\epsilon/\,\mathrm{d}\eta$ and $\mathrm{d}^2 f_\epsilon/\,\mathrm{d}\eta^2$ in cases $\epsilon = 10^{-7}$ and $\epsilon = 10^{-9}$, respectively.

5.2 Case 2: Pohlhausen flow($\beta = 0, \gamma = 1$)

The exact solution of Pohlhausen [33] flow is known as

$$f'(\eta) = 3\tanh^2\left(\frac{\eta}{\sqrt{2}} + \tanh^{-1}\sqrt{\frac{2}{3}}\right) - 2.$$

Hence, $\alpha^* = 2/\sqrt{3}$ and the exact expression of $f'(\eta)$ is also available. We denote by $E_{\epsilon,h} := |\alpha_{\epsilon,h} - \alpha^*|$ the error in the computed solution corresponding to h and ϵ, then estimate the order of accuracy using the formula

$$\text{order} = \frac{\log(E_{\epsilon,h}/E_{\epsilon,h/2})}{\log 2}. \tag{47}$$

Summarized in Table 3 are results obtained with $\varepsilon_1 = 10^{-11}$ and $\varepsilon_2 = 10^{-9}$. We clearly see the second-order accuracy of our method. In Fig. 2, comparisons of $f'(\eta)$ and $f''(\eta)$ are made between exact values with our numerical results.

5.3 Case 3: Several instances($\beta = 1$)

The Falkner-Skan equation mostly refers to the case corresponding to $\beta = 1$. The solutions corresponding to $\gamma > 0$, $\gamma = 0$, and $\gamma < 0$ are known as accelerating, constant, and decelerating flows, respectively.

5.3.1 *Homann flow($\beta = 1, \gamma = 1/2$)*

The numerical results obtained for the Homman flow, corresponding to the value of $\beta = 1, \gamma = 1/2$, are listed in Table 4. We compare our results with those reported in Fazio [20] using $\varepsilon_1 = 10^{-11}$ and $\varepsilon_2 = 10^{-8}$. In Fig. 3, we plot the curves of $\mathrm{d}f_\epsilon/\,\mathrm{d}\eta$ and $\mathrm{d}^2 f_\epsilon/\,\mathrm{d}\eta^2$ when $\epsilon = 10^{-6}$ and $\epsilon = 10^{-7}$.

5.3.2 *Hiemenz flow($\beta = 1, \gamma = 1$)*

The Falkner-Skan equation turns to the Hiemenz flow when $\beta = 1$ and $\gamma = 1$ [34]. In Fig. 4, we plot the curves of $\mathrm{d}f_\epsilon/\,\mathrm{d}\eta$ and $\mathrm{d}^2 f_\epsilon/\,\mathrm{d}\eta^2$ when $\epsilon = 10^{-6}$ and $\epsilon = 10^{-7}$.

Finally, the scaling curves f_ϵ/η_ϵ of f_ϵ for the above four special flows are presented in Fig. 5.

5.3.3 *Other cases ($\beta = 1, \gamma \in [-0.1988, \infty]$)*

To illustrate the robustness and accuracy of our algorithms further, we perform many other tests for the cases that $\beta = 1$ and $\gamma \in [-0.1988, \infty]$. The present results in Table 5, Fig. 6 and Fig. 7 were obtained with $\epsilon = 10^{-6}$, $\varepsilon_1 = 10^{-11}$, and $\varepsilon_2 = 10^{-7}$. Our calculated α_ϵ are in good agreement with those reported [13] and [32]. We also observe that the free truncated boundary η_ϵ is more accurate than [32] especially as $\epsilon \leq 10^{-6}$. Fig. 8 shows the profile of α_ϵ as an increasing function of γ, namely, we verify numerically Coppel [4]'s theoretical result. The figure was obtained with $\varepsilon_1 = 10^{-11}$, and $\varepsilon_2 = 10^{-7}$.

5.4 *Computational effort*

The amount of computational effort needed mainly depends on ε_1 and ε_2. All the computations are carried out on a PC with 3.0 GHz Pentium 4 CPU and 1G memory. each instance usually costs 50-300 ms. All numerical results are obtained with $\eta^{(0)} = 1$ and $\eta^{(1)} = 2$ for all cases.

In Table 6, we list the number of quasilinearization iterations in inner loop for different flows of Blasius, Pohlhausen, Homman and Hiemenz as test examples. Denote by $\mathcal{N}_1$ and $\mathcal{N}_2$ the minimal and maximal number of quasilinearization iterations, respectively. Our results are obtained when $\epsilon = 10^{-7}$, $\varepsilon_2 = 10^{-8}$ and $\{y_j^{(0,0)}\}_{j=0}^{N-1} = \vec{0}$, $y_N^{(0,0)} = 1$. We find that $\mathcal{N}_1$ and $\mathcal{N}_2$ are both independent of ϵ and ε_2 through various experiments.

In Table 7, we report the number of outer iterations for η_ϵ. We choose flows of Blasius, Pohlhausen, Homman and Hiemenz as test examples. We fix $\varepsilon_1 = 10^{-11}$ as we find the iteration number is independent of ε_2. Denote by $\mathcal{K}_1$, $\mathcal{K}_2$ and $\mathcal{K}_3$ the number of outer iterations corresponding to $\varepsilon_2 = 10^{-8}$, $\varepsilon_2 = 10^{-9}$ and $\varepsilon_2 = 10^{-10}$, respectively.

6 Conclusions

We have presented two iterative methods for solving the Falkner-Skan equation based on the quasilinearization technique and a cubic spline solver. We formulate the original problem as a new FBVP. The truncated unknown free boundary depending on a small parameter is determined in iteratively. The quasilinearization method gives a sequence of linear differential equations. Every linear third-order BVP is split into a IVP and a second-order BVP. Solving the IVP analytically or numerically leads to two different iterative methods. The resulting two algorithms both have a nested iteration framework. In every inner iterative step, the cubic spline solver is employed to solve an intermediate step second-order BVP. The numerical results obtained with the second algorithm are presented for various instances of the Falkner-Skan equation. The comparisons between our results with those obtained by others illustrate the accuracy, robustness and efficiency of our methodology.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 10871178) and Natural Science Foundation (No. Y606154) of Zhejiang Province.

References

[1] V.M. Falkner, S.W. Skan, Some approximations of the boundary layers equations, Philos. Mag. Soc. 12 (1931) 865-896.

[2] D.R. Hartree, On an equation occurring in Falkner-Skan approximate treatment of the equations of the boundary layer, Proc. Camb. Phil. Soc. 33 (1937) 223-239.

[3] H. Weyl, On the differential equations of the simplest boundary-layer problem, Ann. Math. 43 (1942) 381-407.

[4] W.A. Coppel, On a differential equation of boundary layer theory, Philos. Trans. Roy. Soc. London Ser. A 253 (1960) 101-136.

[5] B. Oskam, A.E.P. Veldman, Branching of the Falkner-Skan solutions for $\lambda < 0$, J. Engg. Math. 16 (1982) 295-308.

[6] A.E.P. Veldman, A.I. Van de Vooren, On a generalized Falkner-Skan equation, J. Math. Anal. Appl. 25 (1980) 102-111.

[7] O. Padé, On the solution of Falkner-Skan equation, J. Math. Anal. Appl. 285 (2003) 264-274.

[8] G.C. Yang, New results of Falkner-Skan equation arising in boundary layer theory, Appl. Math. Comput. 202 (2008) 406-412.

[9] A. Asaithambi, A finite-difference method for the solution of the Falkner-Skan equation, Appl. Math. Comput. 92 (1998) 135-141.

[10] A. Asaithambi, A second-order finite-difference method for the Falkner-Skan equation, Appl. Math. Comput. 156 (2004) 779-786.

[11] P.M. Beckett, Finite difference solution of boundary-layer type equations, Int. J. Comput. Math. 14 (1983) 183-190.

[12] A. Asaithambi, A numerical method for the solution of the Falkner-Skan equation, Appl. Math. Comput. 81 (1997) 259-264.

[13] A. Asaithambi, Solution of the Falkner-Skan equation by recursive evaluation of Taylor coefficients, J. of Comput. Appl. Math. 176 (2005) 203-214.

[14] T. Cebeci, H.B. Keller, Shooting and parallel shooting methods for solving the Falkner-Skan boundary-layer equation, J. Comp. Phys. 7 (1971) 289-300.

[15] C.W. Chang, J.R. Chang, C.S. Liu, The Lie-group shooting method for boundary layer equations in fluid mechanics, J. Hydrodynamics Ser.B 18 (2006) 103-108.

[16] C.S. Liu, J.R. Chang, The Lie-group shooting method for multiple-solutions of Falkner-Skan equation under suctioninjection conditions, Int. J. Non-Linear Mech. 43 (2008) 844-851.

[17] I. Sher, A.Yakhot, New approach to the solution of the Falkner-Skan equation, AIAA J. 39 (5) (2001) 965-967.

[18] A. Asaithambi, Numerical solution of the Falkner-Skan equation using piecewise linear functions, Appl. Math. Comput. 159 (2004) 267-273.

[19] R. Fazio, The Falkner-Skan equation: Numerical solutions within group invariance theory, Calcolo, 31 (1994) 115-124.

[20] R. Fazio, A novel approach to the numerical solution of boundary value problems on infinite intervals, SIAM J. Numer. Anal. 33 (1996) 1473-1483.

[21] E.M.E. Elbarbary, Chebyshev finite difference method for the solution of boundary-layer equations, Appl. Math. Comp. 160 (2005) 487-497.

[22] H. Nasr, I.A. Hassanien, H.M. El-Hawary, Chebyshev solution of laminar boundary layer flow, Int. J. Comput. Math. 33 (1990) 127-132.

[23] B.L. Kuo, Application of the differential transformation method to the solutions of the Falkner-Skan wedge flow, Acta Mech. 164 (2003) 161-174.

[24] S.J. Liao, A non-iterative numerical approach for two-dimensional viscous flow problems governed by the Falkner-Skan equation, Int. J. Numer. Meth. Fluids 35 (2001) 495-518.

[25] N.S. Elgazery, Numerical solution for the Falkner-Skan equation, Chaos Solit. Fract. 35 (2008) 738-746.

[26] V.B. Mandelzweig, F. Tabakin, Quasilinearization approach to nonlinear problems in physics with application to nonlinear ODEs, Comput. Phys. Comm. 141 (2001) 268-281.

[27] E.L Albasiny, W.D. Hoskins, Cubic splines solutions to two-point boundary value problems, Comput. J. 12 (1969) 151-153.

[28] H. Blasius, Grenzschichten in Flüssigkeiten mit kleiner Reibung, Z. Math. Phys. 56 (1908) 1-37.

[29] R. Fazio, The Blasius problem formulated as a free boundary value problem, Acta Mech. 95 (1992) 1-7.

[30] S.J. Liao, A uniformly valid analytic solution of two-dimensional viscous flow over a semi-infinite flat plate, J. Fluid Mech. 385 (1999) 101-128.

[31] L. Wang, A new algorithm for solving classical Blasius equation, Appl. Math. Comp. 157 (2004) 1-9.

[32] A.A. Salama, Higher order method for solving free boundary problems, Numer. Heat Transfer, Part B: Fundamentals 45 (2004) 385-394.

[33] K. Pohlhausen, Zur näherungsweisen Integration der Differentialgleichung der laminaren Grenzschicht, J. Appl. Math. Mech. (ZAMM) 1 (1921) 252-268.

[34] K. Hiemenz, Die Grenzschicht an einem in den gleichförmigen Flüssigkeitsstrom eingetauchten geraden Kreiszylinder, Dingl. Polytech. J 326 (1911) 321-410.

Table 1
Comparisons of truncated boundary η_ϵ and the wall shear stress α for the Blasius equation.

ε_2	$\epsilon = 10\varepsilon_2$		$\epsilon = \varepsilon_2$		α [32]	η_∞ [32]
	α_ϵ	η_ϵ	α_ϵ	η_ϵ		
10^{-3}	0.335	5.2627	0.332	6.4020	0.332	6.6798
10^{-5}	0.33207	7.2909	0.33206	8.0648	0.33205	8.1847
10^{-7}	0.3320575	8.7527	0.3320573	9.3786	0.3320573	9.3867
10^{-9}	0.332057337	9.9589	0.332057336	10.5001	0.332057336	10.5764
10^{-11}	0.33205733623	11.0101	0.33205733622	11.4889	0.33205733663	11.6890

Table 2
Estimating order of accuracy by grid refinement analysis when α^* is unknown.

h	ϵ	$\alpha_{\epsilon,h}$	$\alpha_{\epsilon,h/2}$	$\alpha_{\epsilon,h/4}$	order
1/16	10^{-8}	0.331936904	0.332024508	0.332049017	1.84
	10^{-6}	0.331955669	0.332031660	0.332051017	1.97
1/128	10^{-8}	0.332055251	0.332056816	0.332057207	2.00
	10^{-6}	0.332055859	0.332057069	0.332057372	2.00
1/1024	10^{-8}	0.332057305	0.332057329	0.332057335	2.00
	10^{-6}	0.332057448	0.332057466	0.332057471	2.00

Table 3
Estimating order of accuracy by grid refinement analysis when α^* is known.

h	ϵ	$\alpha_{\epsilon,h}$	E_h	$\alpha_{\epsilon,h/2}$	$E_{h/2}$	order
1/32	10^{-8}	1.1486964	6.004×10^{-3}	1.1530518	1.649×10^{-3}	1.86
	10^{-6}	1.1509907	3.710×10^{-3}	1.1537211	9.795×10^{-4}	1.92
1/128	10^{-8}	1.1542786	4.219×10^{-4}	1.1545944	1.061×10^{-4}	1.99
	10^{-6}	1.1544523	2.482×10^{-4}	1.1546383	6.227×10^{-5}	1.99
1/512	10^{-8}	1.1546740	2.656×10^{-5}	1.1546939	6.643×10^{-6}	2.00
	10^{-6}	1.1546850	1.558×10^{-5}	1.1546966	3.896×10^{-6}	2.00

Table 4
Comparison of the truncated boundary x_ε and wall shear stress α_ϵ for the Homann flow.

ϵ	Present α_ϵ	α [20]	Present η_ϵ	η_∞ [20]
10^{-1}	0.943096	0.943081	2.216339	2.216707
10^{-2}	0.928477	0.928476	3.183995	3.184503
10^{-3}	0.927733	0.927733	3.924831	3.925363
10^{-4}	0.927684	0.927684	4.543635	4.550585
10^{-5}	0.927680	0.927680	5.084266	5.085175
10^{-6}	0.927680	0.927680	5.569661	5.571160

Table 5
Comparison of the free boundary x_ε and wall shear stress α when $\beta = 1$.

γ	Present α_ϵ	Asaithambi [13]	α [32]	Present η_ϵ	η_∞ [32]
40	7.314785	7.314785	7.314785	1.80	1.37
30	6.338209	6.338209	6.338208	2.03	1.54
20	5.180718	5.180718	5.180718	2.40	1.83
15	4.491487	4.491487	4.491487	2.67	2.50
10	3.675234	3.675234	3.675234	3.09	2.39
2.0	1.687218	1.687218	1.687218	4.68	3.67
1.0	1.232588	1.232589	1.232588	5.19	4.30
0.5	0.927680	0.927680	0.927680	5.57	4.55
0.0	0.469600	0.469600	0.469600	6.26	5.29
-0.1000	0.319270	0.319270	0.319270	6.55	5.56
-0.1500	0.216362	0.216361	0.216362	6.79	5.79
-0.1800	0.128637	0.128637	0.128637	7.03	6.03
-0.1988	0.005229	0.005225	0.005226	7.51	6.68

Table 6
The number of quasilinearization iterations.

ε_1	*Blasius*		*Pohlhausen*		*Homman*		*Hiemenz*	
	$\mathcal{N}_1$	$\mathcal{N}_2$	$\mathcal{N}_1$	$\mathcal{N}_2$	$\mathcal{N}_1$	$\mathcal{N}_2$	$\mathcal{N}_1$	$\mathcal{N}_2$
10^{-9}	5	11	4	10	6	9	6	9
10^{-10}	5	12	5	10	6	10	6	9
10^{-11}	6	13	5	10	7	10	7	10
10^{-12}	6	14	5	10	7	11	8	10

Table 7
The number of outer iterations for η_ϵ.

ε_1	*Blasius*			*Pohlhausen*			*Homman*			*Hiemenz*		
	$\mathcal{K}_1$	$\mathcal{K}_2$	$\mathcal{K}_3$	$\mathcal{K}_1$	$\mathcal{K}_2$	$\mathcal{K}_3$	$\mathcal{K}_1$	$\mathcal{K}_2$	$\mathcal{K}_3$	$\mathcal{K}_1$	$\mathcal{K}_2$	$\mathcal{K}_3$
10^{-5}	16	17	17	18	18	18	17	18	18	17	18	18
10^{-6}	19	19	20	20	21	21	20	21	21	20	20	21
10^{-7}	21	22	23	23	23	24	22	23	24	22	23	23
10^{-8}	23	24	25	25	26	27	24	26	26	24	25	26

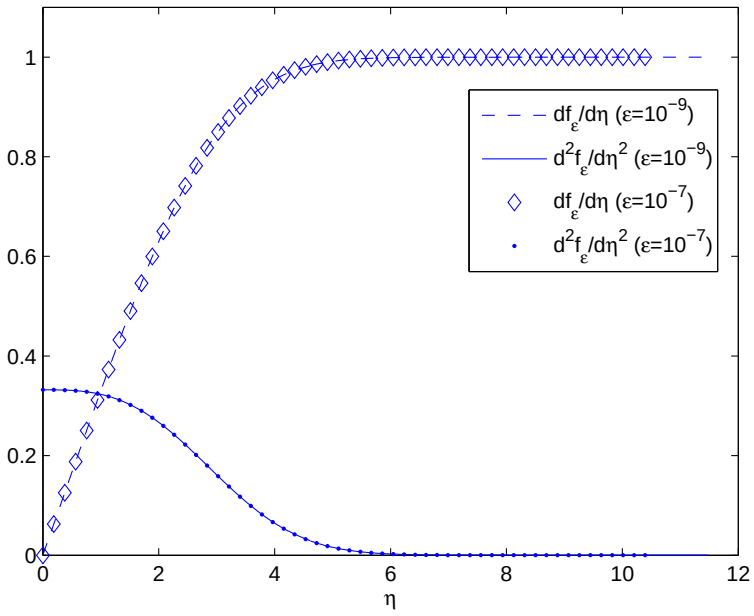


Fig. 1. Numerical solution of the Blasius flow.

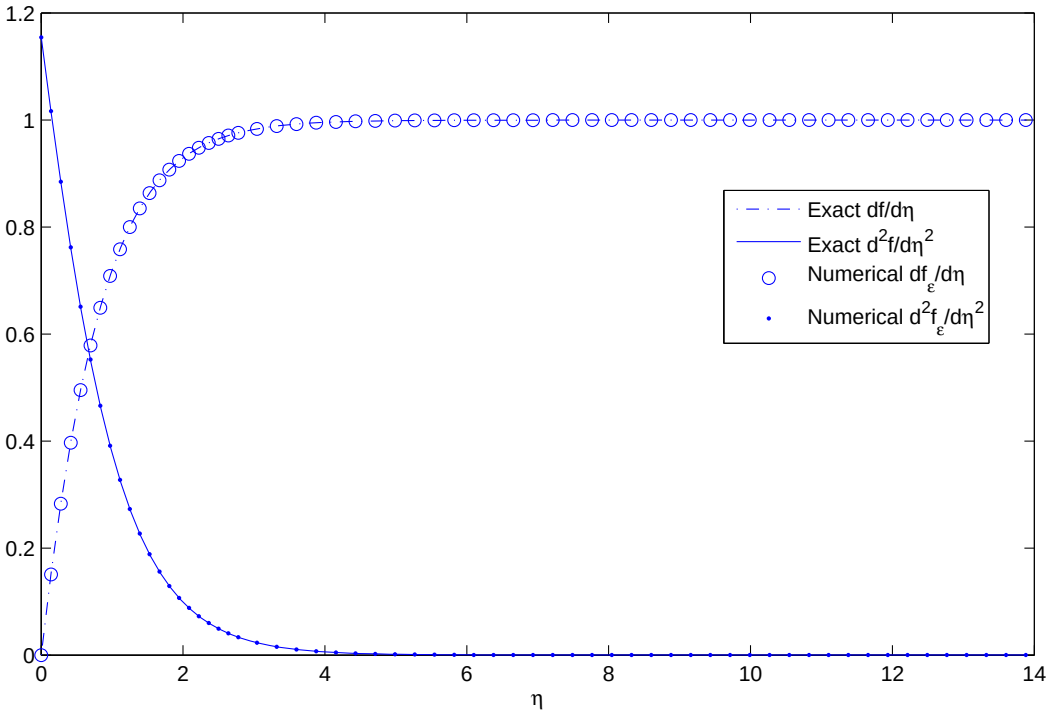


Fig. 2. Comparison between numerical solution and exact solution for the Pohlhausen flow.

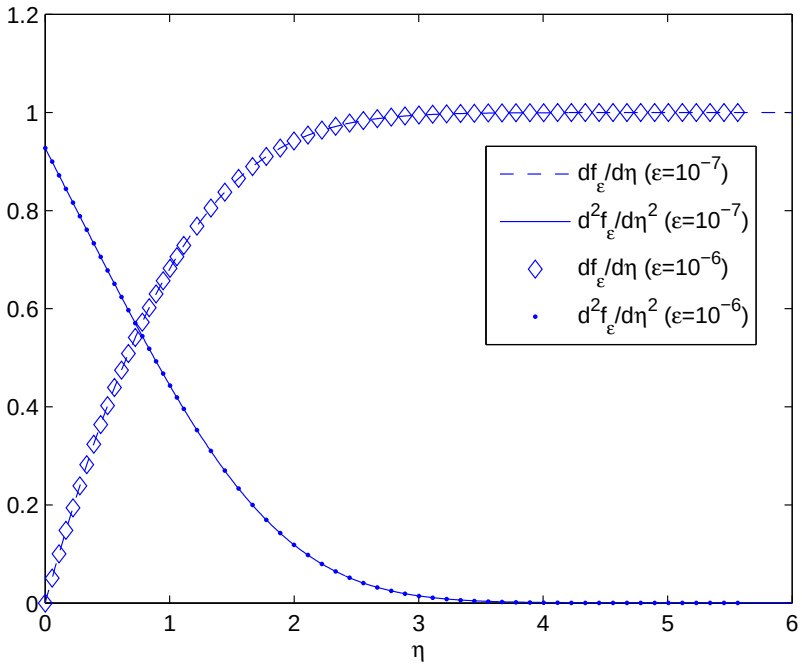


Fig. 3. Numerical solution of the Homman flow.

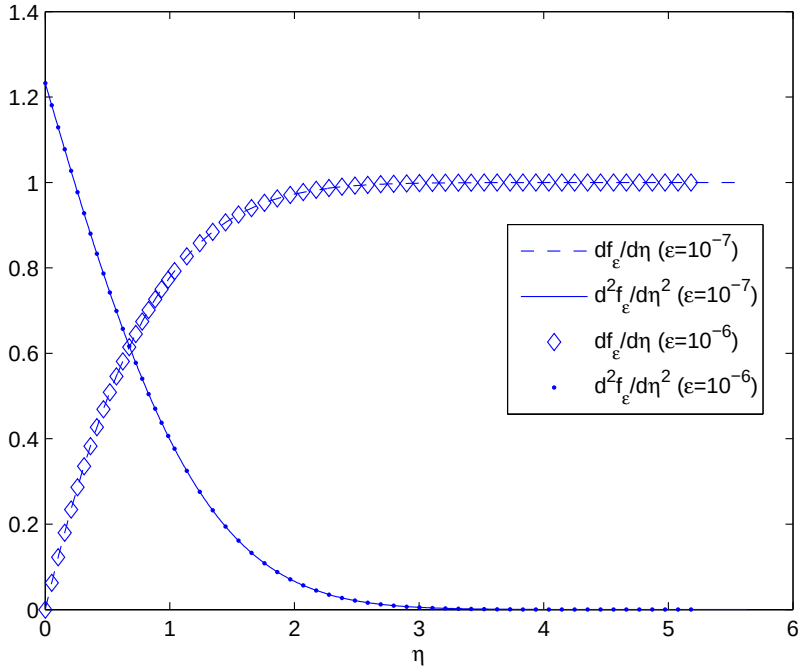


Fig. 4. Numerical solution of the Hiemenz flow.

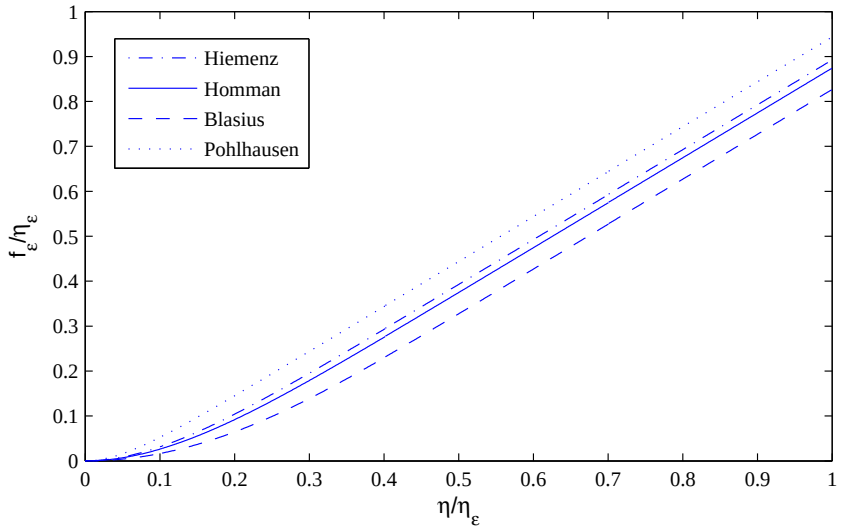
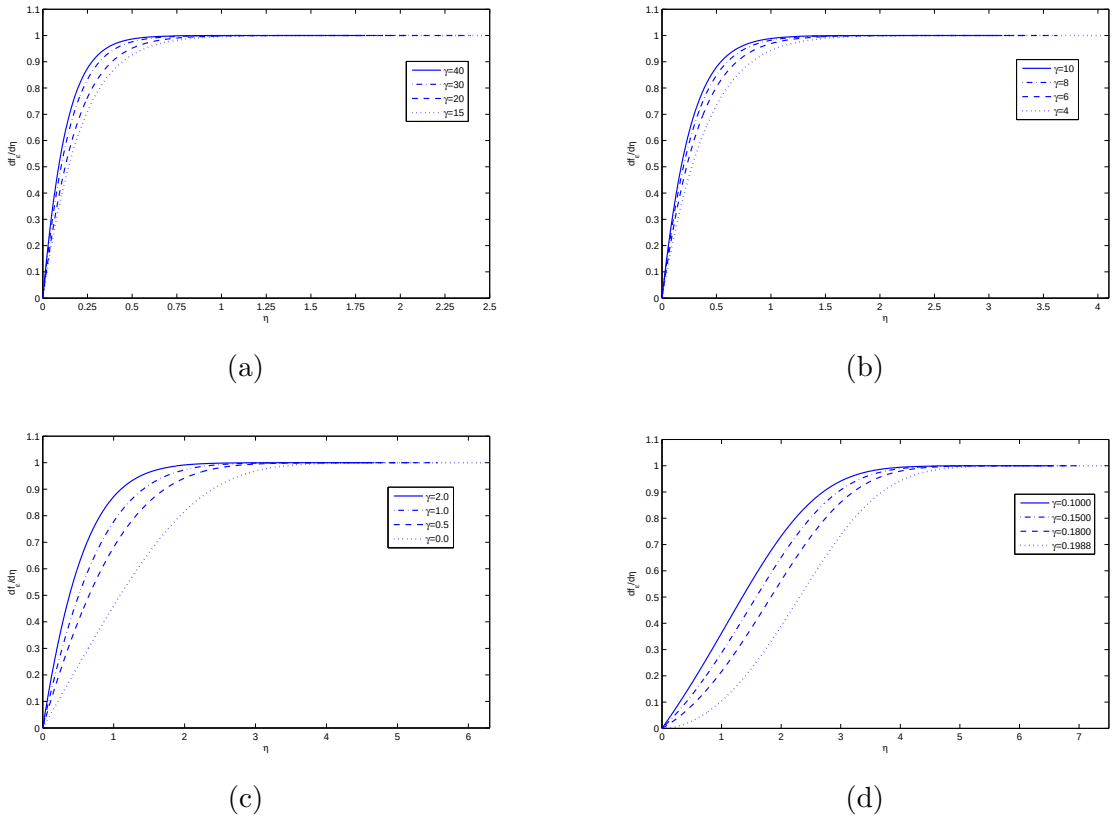


Fig. 5. Numerical f_ϵ for 4 special flows.



(a) (b)

(c) (d)

Fig. 6. Numerical approximation of $\mathrm{d}f/\mathrm{d}\eta$ corresponding to different values for γ.

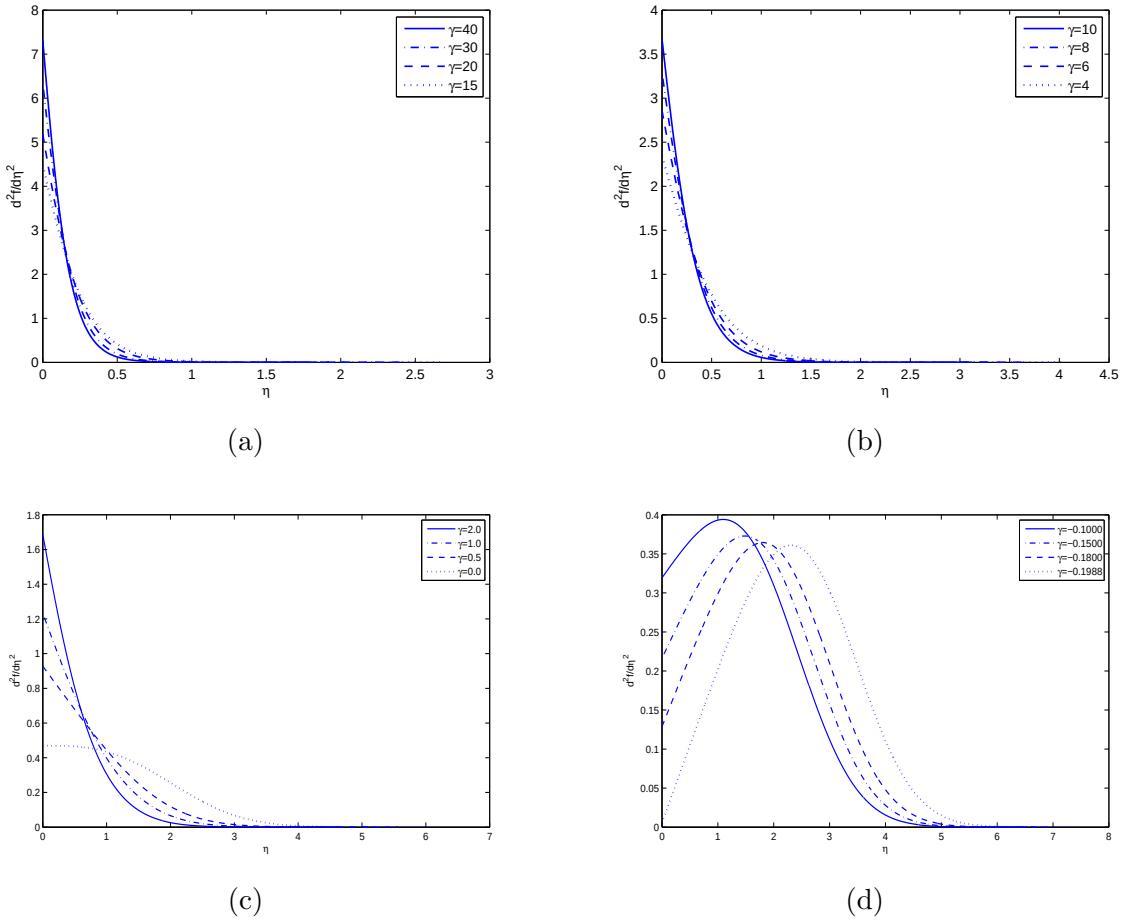


Fig. 7. Numerical approximation of $\mathrm{d}^2 f/\,\mathrm{d}\eta^2$ corresponding to different values for γ.

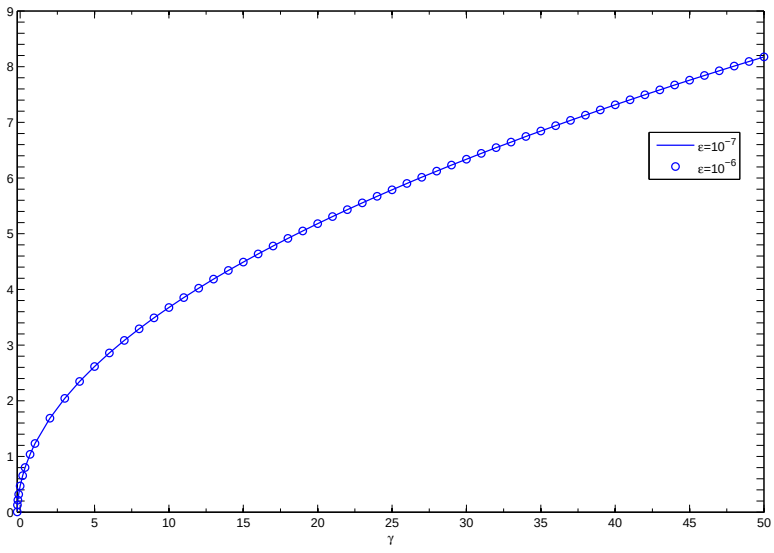


Fig. 8. Profile of α_ϵ as a function of γ.