FDS-2000 Conference, pp. 1–10
M. Sapagovas *et al.* (Eds)

ACCURATE NUMERICAL METHOD FOR BLASIUS' PROBLEM FOR FLOW PAST A FLAT PLATE WITH MASS TRANSFER*

B. GAHAN[1], *J. J. H. MILLER*[2], and *G. I. SHISHKIN*[3]

University of Dublin [1,2],
Russian Academy of Sciences [3]
Department of Mathematics, *Trinity College*, *Dublin* 2, *Ireland*[1,2]
Institute for Mathematics and Mechanics, *Russian Academy of Sciences*, *Ekaterinburg*, *Russia* [3]
E-mail: *bazzag@maths.tcd.ie*[1], *jmiller@tcd.ie*[2], *shishkin@maths.tcd.ie*[3]

ABSTRACT

We construct a new finite difference method for computing *reference* numerical solutions to the one–parameter family of Blasius' problems arising from incompressible laminar flow past a thin flat plate with mass transfer by both suction and blowing. We show that, by studying several representative problems in the family, the method generates nodal approximations, at a finite number of nodes, to the solution and its derivatives, the piecewise linear interpolants of which provide global pointwise accurate approximations to the solution and its derivatives on the semi–infinite domain $[0, \infty)$. Using an experimental error estimate technique we determine orders of convergence and error constants of the reference numerical solutions and their discrete derivatives. Algebraic formulae for realistic pointwise error bounds, in terms of the number of mesh subintervals used in the discrete problem, determine the number of mesh points required to achieve a given preassigned guaranteed accuracy in the reference numerical solutions of Blasius' problem. Such reference numerical solutions to Blasius' problem can be used to construct Re–uniformly accurate approximations to the components $u_P(x, y)$, $v_P(x, y)$ of the solution of Prandtl's problem and to their first order partial derivatives.

1. INTRODUCTION

Blasius' classical approach to finding the self–similar solution of Prandtl's problem arising from laminar flow past a semi–infinite flat plate with mass transfer by both suction and blowing, see [4], leads to a one–parameter family of problems involving a third order nonlinear ordinary differential equation on the semi–infinite domain $[0, \infty)$. The parameter of this family, which occurs in one of the boundary conditions at the origin,

*This research was supported in part by the National Science Foundation grant DMS-9627244, the Enterprise Ireland grant SC-98-612 and the Russian Foundation for Basic Research grant 98-01-00362.

represents the mass transfer. Positive and negative values of the parameter correspond respectively to suction and blowing.

The relations between the velocity components $u_P(x,y)$, $v_P(x,y)$, the self–similar solution of Prandtl's problem, and the solution f of Blasius' problem are

$$u_P(x,y) = f'(\eta) \tag{1.1}$$

$$v_P(x,y) = \sqrt{\frac{1}{2xRe}}(\eta f'(\eta) - f(\eta)) \tag{1.2}$$

where $\eta = y\sqrt{Re/2x}$ and Re is the Reynolds number of the flow, see [5].

It is well known that Prandtl's problem is singularly perturbed for large Re, which is the reason for the failure of standard numerical methods to generate numerical approximations with Re–uniform pointwise accuracy to the solution, and its first order partial derivatives, of Prandtl's problem. By Re–uniformly convergent, we mean that they satisfy error bounds which are independent of Re. We see from the above relations that to accomplish this we require numerical approximations to Blasius' solution, and its first and second order derivatives, with pointwise accuracy on the semi–infinite domain $[0,\infty)$.

In this paper we construct a numerical method for computing *reference* numerical solutions to Blasius' problem. By reference numerical solutions we mean approximations that have guaranteed pointwise accuracy, not only to the solution, but also to its derivatives, on the semi–infinite domain $[0,\infty)$, for any admissible value of the mass transfer parameter. Such reference numerical solutions to Blasius' problem can be used to construct Re–uniformly accurate approximations to the components $u_P(x,y)$, $v_P(x,y)$ of the solution of Prandtl's problem and to their first order partial derivatives; see [3] for details.

Because no sharp estimates of the order of convergence of the numerical solutions generated by this method are at present available, and also no realistic error constant is known, we are forced to use an experimental error analysis technique to determine realistic pointwise error bounds for the reference numerical solutions. The experimental results in this paper show that this new numerical method generates reference numerical solutions with errors of any preassigned accuracy. From algebraic formulae for these realistic pointwise error bounds we determine the number of mesh points required to achieve a given prescribed accuracy in the numerical solutions of Blasius' problem.

2. BLASIUS' PROBLEM

Constructing of the self–similar solution of Prandtl's problem, arising from laminar flow past a semi–infinite flat plate with mass transfer, leads to the following one–parameter family of Blasius' problems

$$(P_B)\begin{cases} \text{Find a function f } \in C^3([0,\infty)) \text{ such that for all } \eta \in (0,\infty) \\ f'''(\eta) + f(\eta)f''(\eta) = 0 \\ \text{with the boundary conditions} \\ f(0) = f_0, \;\; f'(0) = 0 \text{ and } lim_{\eta->\infty} f'(\eta) = 1. \end{cases}$$

where f_0 in one of the boundary conditions at the origin is the mass transfer parameter determining the degree of suction and blowing. The existence and uniqueness of a solution to this third order nonlinear ordinary differential equation is discussed in [5]

The first order derivatives of the velocity components u_P and v_P for prandtl's problem are given by

$$\frac{\partial u_P}{\partial y}(x,y) = \frac{\eta}{y} f''(\eta) \tag{2.1}$$

$$\frac{\partial v_P}{\partial y}(x,y) = -\frac{\partial u_P}{\partial x}(x,y) = \frac{\eta}{2x} f''(\eta) \tag{2.2}$$

$$\frac{\partial v_P}{\partial x}(x,y) = -\frac{1}{2x}[v_P + \sqrt{\frac{1}{2xRe}}\eta^2 f''(\eta)] \tag{2.3}$$

Thus, the construction of approximations to the solution of the original Prandtl problem, and its first order partial derivatives, requires approximations, not only to the solution f of (P_B), but also to its derivatives f' and f'' at each point of the semi–infinite domain $[0,\infty)$.

Although there is no explicit singular perturbation parameter involved, due to the unbounded domain, Blasius' problem is singularly perturbed, see [2] for example. This implies that standard numerical methods are not reliable for the generation, on the semi–infinite domain $[0,\infty)$, of pointwise accurate approximations of the solutions, and especially of their derivatives. In the next section we construct a new numerical method comprising a monotone finite difference operator on a uniform mesh and piecewise linear interpolation both over a finite subset of the semi–infinite domain $[0,\infty)$, and extrapolation procedures over the rest of the domain.

3. NUMERICAL METHOD FOR BLASIUS' PROBLEM

We have to solve (P_B) for f and its derivatives on the semi-infinite domain $[0,\infty)$ for any admissible value of the mass transfer parameter f_0. This is not a trivial matter, since numerical solutions can be obtained at only a finite number of mesh points. For this reason, for each sufficiently large $L \in [1,\infty)$, we consider the following problem on the finite subinterval $(0,L)$

$$(P_{B,L}) \begin{cases} \text{Find a function } f_L \in C^3(0,L) \text{ such that for all } \eta \in (0,L) \\ f_L'''(\eta) + f_L(\eta) f_L''(\eta) = 0 \\ \text{with the boundary conditions} \\ f_L(0) = f_0, \quad f_L'(0) = 0, \quad f_L'(L) = 1. \end{cases}$$

The collection of all such problems forms a new one-parameter family of problems related to (P_B), where the subinterval length L is the parameter of this family.

Since the values of f_L, f_L' and f_L'' are defined only on $[0, L]$, we extend their definition to all points $\eta \in [0, \infty)$ by the following extrapolations for all $\eta \in [L, \infty)$

$$f_L''(\eta) = 0 \tag{3.1}$$
$$f_L'(\eta) = 1 \tag{3.2}$$
$$f_L(\eta) = (\eta - L) + f_L(L), . \tag{3.3}$$

Because L can take arbitrarily large values we need a numerical method which generates L–uniformly convergent approximations to the solution of (P_B), in the sense that the pointwise error bounds are independent of the parameter L. To obtain such approximations to the solution of (P_B), we first compute a numerical solution F_L of $(P_{B,L})$ on the finite subinterval $(0, L)$ and then extrapolate F_L to the semi-infinite domain $[0, \infty)$. We repeat this for an increasing sequence of values of L defined as follows: for each even number $N \geq 4$ we take

$$L_N = lnN$$

(see [2] for a motivation of this choice of L_N). On $[0, L_N]$ a uniform mesh $\overline{I_u^N} = \{\eta_i : \eta_i = iN^{-1}lnN, 0 \leq i \leq N\}_0^N$ with N mesh subintervals is constructed. We denote the set of interior mesh points by I_u^N Then numerical approximations F_L, D^+F_L, $D^+D^+F_L$ to f_L, f_L', f_L'' respectively, are determined at the mesh points in $\overline{I}_u^N$ using the following non-linear finite difference method

$$(P_{B,L}^N) \begin{cases} \text{Find F on } \bar{I}_u^N \text{ such that, for all } \eta_i \in I_u^N, \\ \delta^2(D^-F)(\eta_i) + F(\eta_i) D^+(D^-F)(\eta_i) = 0 \\ F(0) = f_0 \quad D^+F(0) = 0, \text{ and } \quad D^0F(\eta_{N-1}) = 1. \end{cases}$$

In order to simplify the notation, we drop explicit mention of the indices L and N. Thus, we denote the solution of $(P_{B,L}^N)$ by F instead of F_L^N.

Since $(P_{B,L}^N)$ is non-linear, we use the following iterative solver to compute its solution

$$(A_B^N)\begin{cases} \text{For each integer } m, 1 \le m \le M, \text{ find } F^m \text{ on } I_u^N \text{ such that , for all } \eta_i \in I_u^N, \\ \delta^2(D^-F^m)(\eta_i) + F^{m-1}(\eta_i)D^+(D^-F^m)(\eta_i) - D^-(F^m - F^{m-1})(\eta_i) = 0 \\ F^m(0) = f_0, \quad D^+F^m(0) = 0, \text{ and } D^0F^m(\eta_{N-1}) = 1 \\ \text{with the starting values for all mesh points } \eta_i \in \overline{I}_u^N \\ F^0(\eta_i) = \eta_i. \end{cases}$$

We take the function F^M to be the approximation of F. Algorithm (A_B^N) involves the solution of a sequence of linear problems, with one problem for each value of the iteration index m. To obtain a sufficiently accurate approximation of F we must take M sufficiently large. Here, the appropriate value of M is taken to be $M = 8lnN$. The motivation for this choice of M is described in [2]. The final output of the algorithm (A_B^N) is denoted by F, where again we simplify the notation by omitting explicit mention of M.

To ensure that F, D^+F and D^+D^+F are defined at each point of the mesh $\overline{I_u^N}$ we assign the following values at the mesh points η_N and η_{N-1}: $D^+F(\eta_N) = 1$, $D^+D^+F(\eta_{N-1}) = 0$ and $D^+D^+F(\eta_N) = 0$. Then, using piecewise linear interpolation, we define the interpolants $\overline{F}$, $\overline{D^+F}$ and $\overline{D^+D^+F}$, respectively, of F, D^+F and D^+D^+F at each point of the interval $[0, L_N]$. Finally, to define $\overline{F}$, $\overline{D^+F}$ and $\overline{D^+D^+F}$ at each point of the semi–infinite interval $[0, \infty)$, the following extrapolations, analogous to (3.1), (3.2) and (3.3), are introduced for all $\eta \in [L_N, \infty)$

$$\overline{D^+D^+F}(\eta) = 0 \tag{3.4}$$

$$\overline{D^+F}(\eta) = 1 \tag{3.5}$$

$$\overline{F}(\eta) = \overline{F}(L_N) + (\eta - L_N). \tag{3.6}$$

The values of $\overline{F}$, $\overline{D^+F}$ and $\overline{D^+D^+F}$, respectively, obtained in this manner, are the required numerical approximations to f, f', f'' of Blasius' solution and its derivatives at each point of $[0, \infty)$. In what follows we examine the quality of these approximations as a function of N.

4. NUMERICAL EXPERIMENTS

In [4] a limiting value for suction is found at $f_0 = 7.07$ and for blowing at $f_0 = -0.875475$. Thus, to illustrate the efficiency of the numerical method described above, we take the representative values $f_0 = 3$, $f_0 = 6$ for suction and $f_0 = -0.25$, $f_0 = -0.5$ for blowing.

Since the exact solution of (P_B) is unknown, we cannot directly determine the exact pointwise errors in the numerical solution. Also, for different values of N the meshes I_u^N

usually do not overlap. Therefore, in the expression for the maximum pointwise errors E^N, we replace the unknown exact solution by the piecewise linear interpolant $\overline{F^*}$ of the numerical solution F^* of (A_B^N) computed on either the finest available mesh or on a sufficiently fine mesh, where the required number of mesh subintervals N^* is determined by the criteria discussed in [2] and the corresponding mesh is denoted by $\overline{I_u^{N^*}}$. Thus, we introduce the following computed maximum pointwise errors on $\overline{I_u^N}$

$$\begin{aligned} E_0^N &= \|F^N - \overline{F^*}\|_{\overline{I_u^N}} \\ E_1^N &= \|D^+F^N - \overline{D^+F^*}\|_{\overline{I_u^N}} \\ E_2^N &= \|D^+D^+F^N - \overline{D^+D^+F^*}\|_{\overline{I_u^N}}. \end{aligned}$$

With $N^* = 65536$ the computed errors E_0^N, E_1^N and E_2^N for F, D^+F and D^+D^+F respectively are shown in Table 1 , for each representative value of f_0. The results in these tables are calculated using quadruple precision arithmetic, because when $N > 2048$ the effect of rounding error is found to be significant when only double precision is used.

Table 1 *Computed maximum pointwise errors E_i^N on I_u^N for i=0,1,2, and various values of N and f_0*

$i = 0$

$f_0 : N$	128	256	512	1024	2048	4096	8192	16384
6	0.036839	0.020977	0.011726	0.006441	0.003468	0.001816	0.000908	0.000413
3	0.035024	0.019937	0.011142	0.006119	0.003295	0.001726	0.000863	0.000392
0	0.018727	0.011030	0.006181	0.003393	0.001826	0.000956	0.000478	0.000217
−0.25	0.010627	0.008069	0.004704	0.002593	0.001396	0.000731	0.000365	0.000166
−0.5	0.032139	0.001552	0.002257	0.001529	0.000842	0.000443	0.000222	0.000101

$i = 1$

$f_0 : N$	128	256	512	1024	2048	4096	8192	16384
6	0.037676	0.022334	0.012765	0.007103	0.003854	0.002027	0.001016	0.000462
3	0.019400	0.011270	0.006371	0.003523	0.001904	0.001000	0.000500	0.000228
0	0.001266	0.000679	0.000386	0.000212	0.000114	0.000060	0.000030	0.000014
−0.25	0.003829	0.001689	0.000909	0.000498	0.000269	0.000141	0.000070	0.000032
−0.5	0.016360	0.004226	0.001621	0.000824	0.000440	0.000230	0.000115	0.000052

$i = 2$

$f_0 : N$	128	256	512	1024	2048	4096	8192	16384
6	0.564052	0.467196	0.325810	0.203342	0.117709	0.064198	0.032822	0.015102
3	0.222855	0.153314	0.095828	0.056134	0.031345	0.016755	0.008472	0.003874
0	0.006279	0.003608	0.002023	0.001112	0.000599	0.000314	0.000157	0.000071
−0.25	0.006269	0.003607	0.002024	0.001113	0.000600	0.000314	0.000157	0.000071
−0.5	0.022472	0.004054	0.002178	0.001189	0.000640	0.000335	0.000167	0.000076

The main conclusion to be drawn from an examination of this table is that the method (P_B^N), together with algorithm (A_B^N), is convergent for F, D^+F and D^+D^+F on I_u^N. The results also suggest that this conclusion holds at least for all values of f_0 between -0.5 and 6.

5. REALISTIC ERROR BOUNDS

It is not easy to use the results in Table 1 to determine the value of N required to guarantee a given prescribed accuracy for the computed values of f, f' and f'' on $[0,\infty)$. We now construct realistic error bounds, which can easily be used for this purpose. We note that, in contrast to the previous section, we do not need to replace the exact solution by the solution on the finest available mesh.

Corresponding to $\overline{F}$, $\overline{D^+F}$ and $\overline{D^+D^+F}$, for $i = 0, 1, 2$ respectively and each representative value of f_0, we introduce the pointwise two-mesh differences $\overline{D_i}^N$ and the computed local orders of convergence $\overline{p_i}^N$, where

$$\begin{aligned}
\overline{D_0}^N &= \|\overline{F}^N - \overline{F}^{2N}\|_{[0,\infty)} \\
\overline{D_1}^N &= \|\overline{D^+F}^N - \overline{D^+F}^{2N}\|_{[0,\infty)} \\
\overline{D_2}^N &= \|\overline{D^+D^+F}^N - \overline{D^+D^+F}^{2N}\|_{[0,\infty)} \\
\overline{p_i}^N &= log_2 \frac{\overline{D_i}^N}{\overline{D_i}^{2N}}, \quad i = 0, 1, 2.
\end{aligned}$$

To evaluate these computed formulae we need to find the global two–mesh differences $\overline{D_i}^N$ on the semi–infinite domain $[0,\infty)$. For this purpose it is convenient to consider the three subintervals $[0, L_N)$, $[L_N, L_{2N})$ and $[L_{2N}, \infty)$ separately. In $[0, L_N)$ the two–mesh differences are obtained directly for all three expressions. For $\eta \in [L_N, L_{2N})$ the two-mesh difference at η for $\overline{F}$ is $\overline{F^{2N}} - F^N(L_N) - (\eta - L_N)$, for $\overline{D^+F}$ it is $\overline{D^+F^{2N}}(\eta) - 1$, and for $\overline{D^+D^+F}$ it is $\overline{D^+D^+F^{2N}}(\eta)$. In the subinterval $[L_{2N}, \infty)$ the two-mesh difference at η for $\overline{F}$ is $\overline{F^{2N}}(L_{2N}) - F^N(L_N) - ln2$, while for $\overline{D^+F}$ and $\overline{D^+D^+F}$ it is zero.

We take the computed global orders of convergence to be

$$\overline{p}_i^* = \min_N \overline{p}_i^N. \tag{5.1}$$

Corresponding to these $\overline{p}_i^*$ we define

$$\overline{C}_{\overline{p}_i^*}^N = \frac{\overline{D}_i^N N^{\overline{p}_i^*}}{1 - 2^{-\overline{p}_i^*}} \tag{5.2}$$

and we take the computed global error constants to be

$$\overline{C}_{\overline{p}_i^*}^* = \max_N \overline{C}_{\overline{p}_i^*}^N. \tag{5.3}$$

Thus we obtain the computed error bounds

$$\begin{aligned}
\|\overline{F} - f\|_{[0,\infty)} &\leq \overline{C}_{\overline{p}_0^*}^* N^{-\overline{p}_0^*} \\
\|\overline{D^+F} - f'\|_{[0,\infty)} &\leq \overline{C}_{\overline{p}_1^*}^* N^{-\overline{p}_1^*} \\
\|\overline{D^+D^+F} - f''\|_{[0,\infty)} &\leq \overline{C}_{\overline{p}_2^*}^* N^{-\overline{p}_2^*},
\end{aligned} \tag{5.4}$$

We now apply this procedure to find the computed global error parameters $\overline{p}_i^*$ and $\overline{C}_{\overline{p}_i^*}^*$. For each representative value of f_0, the computed values of $\overline{D_i}^N$ and $\overline{p_i}^N$ respectively are given in Tables 2 and 3 for $i = 0, 1, 2$.

Table 2 *Computed global two-mesh differences* $\overline{D_i}^N$ *for i=0,1,2 and various values of* N *and* f_0.

$i = 0$

$f_0 : N$	128	256	512	1024	2048	4096	8192	16384
6	0.015863	0.009251	0.005285	0.002973	0.001651	0.000908	0.000495	0.000268
3	0.015087	0.008794	0.005023	0.002825	0.001569	0.000863	0.000471	0.000255
0	0.007740	0.004851	0.002788	0.001567	0.000870	0.000478	0.000261	0.000141
-0.25	0.003050	0.003384	0.002112	0.001197	0.000665	0.000365	0.000199	0.000108
-0.5	0.033054	0.004036	0.000759	0.000688	0.000400	0.000221	0.000121	0.000065

$i = 1$

$f_0 : N$	128	256	512	1024	2048	4096	8192	16384
6	0.017585	0.010224	0.005884	0.003320	0.001849	0.001018	0.000556	0.000301
3	0.008759	0.005096	0.002915	0.001638	0.000911	0.000501	0.000273	0.000148
0	0.000607	0.000296	0.000174	0.000098	0.000054	0.000030	0.000016	0.000009
-0.25	0.002172	0.000781	0.000411	0.000230	0.000128	0.000070	0.000038	0.000021
-0.5	0.012302	0.002633	0.000798	0.000384	0.000210	0.000115	0.000063	0.000034

$i = 2$

$f_0 : N$	128	256	512	1024	2048	4096	8192	16384
6	0.283927	0.225431	0.154714	0.096731	0.057043	0.032430	0.018013	0.009854
3	0.103096	0.070608	0.044320	0.026303	0.015059	0.008421	0.004635	0.002523
0	0.0026697	0.001585	0.000910	0.000513	0.000285	0.000157	0.000086	0.000046
-0.25	0.003075	0.001586	0.000911	0.000513	0.000286	0.000157	0.000086	0.000046
-0.5	0.021879	0.003680	0.000990	0.000549	0.000305	0.000168	0.000091	0.000049

Table 3 *Computed global order of convergence* $\overline{p_i}^N$ *for i=0,1,2 and various values of* N *and* f_0.

$i = 0$

$f_0 : N$	128	256	512	1024	2048	4096	8192	16384
6	0.78	0.81	0.83	0.85	0.86	0.87	0.88	0.89
3	0.78	0.81	0.83	0.85	0.86	0.87	0.88	0.89
0	0.67	0.80	0.83	0.85	0.86	0.87	0.88	0.89
-0.25	-0.15	0.68	0.82	0.85	0.86	0.87	0.88	0.89
-0.5	3.03	2.41	0.14	0.78	0.85	0.87	0.88	0.89

$i = 1$

$f_0 : N$	128	256	512	1024	2048	4096	8192	16384
6	0.78	0.80	0.83	0.84	0.86	0.87	0.88	0.89
3	0.78	0.81	0.83	0.85	0.86	0.87	0.88	0.89
0	1.04	0.77	0.83	0.85	0.86	0.87	0.88	0.89
-0.25	1.47	0.93	0.84	0.85	0.86	0.87	0.88	0.89
-0.5	2.22	1.72	1.05	0.87	0.87	0.88	0.88	0.89

$i = 2$

$f_0 : N$	128	256	512	1024	2048	4096	8192	16384
6	0.33	0.54	0.68	0.76	0.81	0.85	0.87	0.89
3	0.55	0.67	0.75	0.80	0.84	0.86	0.88	0.89
0	0.75	0.80	0.83	0.85	0.86	0.87	0.88	0.89
-0.25	0.96	0.80	0.83	0.85	0.86	0.87	0.88	0.89
-0.5	2.57	1.89	0.85	0.85	0.86	0.87	0.88	0.89

From Tables 2 and 3 we see that the computed values of $\overline{D_i}^N$ and $\overline{p_i}^N$, for i=0,1,2 and each representative value of f_0, are stable for $N \geq 2048$, consequently we determine the error parameters only for this range of N. By inspection of Table 3 for the range of values $N \geq 2048$ we obtain the values of $\overline{p_i}^*$, defined by (5.1), given in Table 4.

Table 4 *Computed global order of convergence* $\overline{p_i}^*$ *for i=0,1,2,* $N \geq 2048$ *and various values of* f_0*.*

$f_0 : i$	0	1	2
6	0.86	0.86	0.81
3	0.86	0.86	0.84
0	0.86	0.86	0.86
-0.25	0.86	0.86	0.86
-0.5	0.85	0.87	0.86

We use the results in Tables 2 and 4 and the definitions in (5.2) and (5.3) to compute the values of $\overline{C}_{\overline{p}_i^*}^N$ and $\overline{C}_{\overline{p}_i^*}^*$ shown in Table 5.

Table 5 *Computed global error constants* $\overline{C}_{\overline{p}_i^*}^N$ *and* $\overline{C}_{\overline{p}_i^*}^*$ *for i=0,1,2,* $N \geq 2048$ *and various values of* f_0*.*

$i = 0$

$f_0 : N$	2048	4096	8192	16384	$\overline{C}_{\overline{p}_i^*}^*$
6	2.5894	2.58478	2.55757	2.5133	2.5894
3	2.46079	2.45668	2.43357	2.39138	2.46079
0	1.36449	1.36071	1.34854	1.32229	1.36449
-0.25	1.04297	1.03904	1.0282	1.01282	1.04297
-0.5	0.586301	0.583886	0.576231	0.557956	0.586301

$i = 1$

$f_0 : N$	2048	4096	8192	16384	$\overline{C}_{\overline{p}_i^*}^*$
6	2.89994	2.89791	2.87275	2.82277	2.89994
3	1.42879	1.42618	1.41054	1.38794	1.42879
0	0.0846926	0.0854001	0.0826691	0.0844017	0.0854001
-0.25	0.200753	0.199267	0.196339	0.196937	0.200753
-0.5	0.352467	0.352772	0.35321	0.348391	0.35321

$i = 2$

$f_0 : N$	2048	4096	8192	16384	$\overline{C}_{\overline{p}_i^*}^*$
6	63.8700	63.6613	61.9939	59.4579	63.8700
3	20.6311	20.6517	20.3473	19.8262	20.6517
0	0.446989	0.446927	0.444346	0.431387	0.446989
-0.25	0.448557	0.446927	0.444346	0.431387	0.448557
-0.5	0.478356	0.478241	0.47018	0.459521	0.478356

Thus, in Tables 4 and 5, we have numerical estimates of the error constants for the approximations $\overline{F}$, $\overline{D^+F}$ and $\overline{D^+D^+F}$ on the semi–infinite domain $[0, \infty)$, for each representative value of f_0. Using these computed values $\overline{p_i}^*$ and $\overline{C}_{\overline{p}_i^*}^*$ of the error parameters

we obtain realistic error bounds from the algebraic formulae $\overline{C}^*_{\overline{p}^*_i} N^{-\overline{p^*_i}}$ given in (5.4). The resulting error bounds are presented in Table 6.

Table 6 *Computed global error bounds* $\overline{C}^*_{\overline{p}^*_i} N^{-\overline{p}^*_i}$ *for i=0,1,2,* $N \geq 2048$ *and various values of* f_0.

$i = 0$

$f_0 : N$	2048	4096	8192	16384
6	0.00367667	0.00202567	0.00111605,	0.00061489
3	0.00349406	0.00192506	0.00106062	0.00058435
0	0.00193743	0.00106743	0.000588106	0.000324018
-0.25	0.00148091	0.000815913	0.000449529	0.000247669
-0.5	0.000898442	0.000498442	0.000276528	0.000153413

$i = 1$

$f_0 : N$	2048	4096	8192	16384
6	0.00411761	0.00226861	0.00124989	0.000688632
3	0.00202874	0.00111774	0.000615821	0.000339288
0.	0.000121259	0.0000668081	0.0000368081	0.0000202795
-0.25	0.000285048	0.000157048	0.0000865259	0.0000476717
-0.5	0.000464703	0.000254261	0.000139118	0.0000761179

$i = 2$

$f_0 : N$	2048	4096	8192	16384
6.	0.132776	0.0757331	0.0431968	0.0246386
3.	0.0341538	0.0190798	0.0106588	0.00595448
0.	0.000634677	0.000349677	0.000192655	0.000106144
0.25	0.000636904	0.000350904	0.000193331	0.000106516
0.5	0.000679216	0.000374216	0.000206175	0.000113593

To determine the effectiveness in practice of our procedure for computing the error bounds defined in (5.4), we compare the computed error bounds in Table 6 with the errors E_i^N in Table 1. We see that the entries in Table 6 are only slightly larger than (less than double) the corresponding entries in Table 1. This leads to the conclusion that the computed error bounds in Table 6 are realistic bounds on the maximum pointwise errors in the approximations $\overline{F}$, $\overline{D^+F}$ and $\overline{D^+D^+F}$ to the exact values f, f' and f'' for Blasius' problem (P_B) on the semi–infinite interval $[0, \infty)$ for all $N \geq 2048$. The results also suggest that the same conclusion is valid for all values of f_0 between -0.5 and 6.

We see from these computed error bounds that, for all $N \geq 2048$ and each representative value of f_0, the order of convergence to the solution of Blasius' problem and its first and second derivatives is not less than 0.8 and the error constant is not greater than 64.

In [3] for the particular case $f_0 = 0$ the above error bounds, and similar error bounds for approximations of various more complicated expressions involving f and its derivatives, were used to compute pointwise error bounds for the computed components of the velocity and its first order derivatives for the self-similar Prandtl problem.

6. CONCLUSION

A new finite difference method was constructed for computing *reference* numerical solutions to the one–parameter family of Blasius' problems arising from incompressible

laminar flow past a thin flat plate with mass transfer by both suction and blowing. By examining several representative problems in the family, it was shown that the method generates numerical approximations to the solution and its derivatives, the piecewise linear interpolants of which provide global pointwise accurate approximations to the solution and its derivatives on the semi–infinite domain $[0, \infty)$. Using an experimental error estimate technique we determined orders of convergence and error constants of the reference numerical solutions and their discrete derivatives. It was found that the order of convergence to the solution of Blasius' problem and its first and second derivatives is not less than 0.8 and that the error constant is not greater than 64. It was also found that the algebraic formulae for the pointwise error bounds, in terms of the number of mesh subintervals used in the discrete problem, give realistic estimates of the maximum pointwise errors, and hence lead to realistic estimates of the number of mesh points required to achieve a given preassigned guaranteed accuracy in the reference numerical solutions of Blasius' problem. Such reference numerical solutions to Blasius' problem can be used to construct Re–uniformly accurate approximations to the components $u_P(x, y)$, $v_P(x, y)$ of the solution of Prandtl's problem and to their first order partial derivatives.

REFERENCES

[1] D.J. Acheson, *Elementary Fluid Dynamics* , Oxford: Clarendon, 1990.

[2] P. Farrell, A. Hegarty, J.J.H. Miller, E. O' Riordan, G.I. Shishkin, *Robust Computational Techniques for Boundary Layers* , Series in Applied Mathematics and Mathematical Computation (Eds. R J Knops and K W Morton), Chapman and Hall/CRC Press, 2000

[3] B. Gahan, J. J. H. Miller, G. I. Shishkin: Reynolds–uniform numerical method for Prandtl's problem with suction–blowing based on Blasius' approach. Trinity College Dublin, Maths. Dept. Preprint TCDMATH 00–07.

[4] D.F. Rogers, *Laminar Flow Analysis* , Cambridge University Press, 1992.

[5] H. Schlichting, *Boundary Layer Theory* , 7th Edition, McGraw Hill, 1951.