18th International Conference
ENGINEERING MECHANICS 2012
Svratka, Czech Republic, May 14 – 17, 2012

pp. 335–342
Paper **#348**

AN ENHANCED NUMERICAL SOLUTION OF BLASIUS EQUATION BY MEANS OF THE METHOD OF DIFFERENTIAL QUADRATURE

A. Guran*, J. Gwinner**

Abstract: *The differential quadrature method (DQM) is used to solve the two-dimensional Blasius boundary layer problem which is described by a third-order nonlinear differential equation. The governing nonlinear equation of boundary-value Blasius problem is first converted to a pair of nonlinear initial-value problems and then solved by both the DQ method and classical fourth-order Runge-Kutta method (RK4). It is revealed that as compared to the RK4, the DQ method can achieve much higher order of accuracy for the numerical results using larger time step sizes.*

Keywords: ***Differential quadrature method (DQM); Blasius equation; Fourth-order Runge-Kutta method (RK4); Convergence and accuracy of DQ solutions.***

1. Introduction

The Blasius boundary layer is an example of two-dimensional boundary layer problems. The Blasius problem models the behavior of two-dimensional steady state laminar viscous flow of an incompressible fluid over a semi-infinite flat plate. The governing equation of the problem is

$$f'''(\eta)+\frac{1}{2}f(\eta)f''(\eta)=0 \quad , \quad 0\le\eta\le\infty \tag{1}$$

where η and f(η) are the dimensionless coordinate and stream function, respectively (Schiliching, 2004). The boundary conditions for equation (1) are

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

The problem was first solved by Blasius using a series expansions method. But the proposed analytic series solution does not converge at all. In fact, the obtained analytic solution is valid only for small values of η (i.e., the series solution converges only within a finite interval [0, η0] where η0 is an unknown constant which can be determined numerically or analytically). Howarth (Howarth, 1938) solved the Blasius equation numerically and found η0 ≈ 1.8894/0.33206. Furthermore, Asaithambi (Asaithambi, 2005) solved the Blasius equation more accurately and obtained this number as η0 ≈ 1.8894/0.332057336. Due to the limitation of Blasius power series solution, many attempts have been made to obtain the solutions which are valid on the whole domain of the problem. Some researchers have solved the problem numerically and some analytically or semi-analytically. Applying the homotopy analysis method (Liao, 1997), Liao obtained an analytic solution for the Blasius equation which is valid in the whole region of the problem (Liao, 1998; Liao,1999). Using the variational iteration method (He,1997), He constructed a five-term approximate-analytic solution for the Blasius equation which is also valid for large values of η (He,1999). However, the solutions obtained were not very accurate. The Adomian decomposition method (ADM) has also been used by some researchers to find analytic solutions for the Blasius equation (Allan and Syam, 2005; Wang, 2004; Abbasbandy, 2007). A homotopy perturbation solution to this problem was presented by He (He, 2003; He, 2004). Fang et al. (Fang et al., 2006), Cortell (Cortell, 2005), Ahmad (Ahmad, 2007), and Ahmad and Al-Barakati (Ahmad, 2009) also solved the Blasius problem using various numerical and analytical methods. From the review of the proposed schemes in (Liao, 1997; Ahmad and Al-Barakati, 2009), two general limitations may be observed: (1) The proposed approximate-analytic methods can not yield accurate solutions when a rather small number of solution terms are used, (2) Many calculations

* Professor, Technical University of Munich (Germany), Director, Institute of Structronic, 275 Slater Street, 9th Floor,Ottawa, Canada. Corresponding Author, ardeshir.guran@mcgill.ca

should be done to construct the resulting semi-analytic solutions and this increases considerably the CPU time especially when a large number of terms of solutions are to be used. The above-mentioned limitations can be eliminated by using the higher-order methods such as the differential quadrature method (DQM).The DQ method is capable of yielding high accurate numerical solutions using very few grid points. So far, the DQ method has been efficiently employed in a variety of problems in engineering, mathematics, and physical sciences and is emerging as a powerful numerical discretization tool. However, the DQ method has its difficulty in implementation to the problems with irregular and complex geometry. Since the time domain has no such difficulty, the strengths of higher-order accuracy of the DQ method can be fully exploited in approximating the time derivatives.

In this work, we first convert the Blasius equation to a pair of initial-value problems (Na, 1979) and then solve the pair of initial-value problems by the DQ method. The resultant initial-value problems are also solved by the conventional fourth-order Runge-Kutta method (RK4). The efficiency, accuracy, and convergence of the DQ time integration method are then investigated and analyzed. It is demonstrated that the DQ method gives better accuracy than the RK4 using much larger time steps.

2. Differential quadrature method

Let $f(t)$ be a solution of a differential equation and $0 = t_1, t_2, \ldots, t_m = T$ be a set of sample points in the time domain. According to the DQ method, the first-order derivative of the function $f(t)$ at any sample time points can be approximated by the following formulation (Bellman and Casti, 1971; Shu, 2001; Bert and Malik , 1996)

$$\dot{f}(t_i) = \sum_{j=1}^{m} A_{ij} f(t_j) \qquad \text{or} \qquad \dot{f}_i = \sum_{j=1}^{m} A_{ij} f_j \tag{3}$$

where m is the number of sample points in the time domain, $f(t_j)$ represents the functional value at a sample time point t_j, $\dot{f}(t_i)$ indicates the first-order derivative of $f(t)$ at a time point t_i, and A_{ij} is the weighting coefficient of the first-order derivative. The weighting coefficients can be determined by the functional approximations in the time domain. Using the Lagrange interpolation polynomials as the approximating functions (say test functions), Quan and Chang (Quan and Chang, 1989) obtained the following algebraic formulations to compute the first-order weighting coefficients

$$A_{ik} = \begin{cases} \dfrac{M^{(1)}(t_i)}{(t_i - t_k) M^{(1)}(t_k)} & i \neq k, \quad i,k = 1,2,\ldots,m \\ -\displaystyle\sum_{j=1, j\neq i}^{m} A_{ij} & i = k, \quad i = 1,2,\ldots,m \end{cases} \tag{4}$$

where $M^{(1)}(t)$ is defined as

$$M^{(1)}(t_i) = \prod_{j=1, j\neq i}^{m} (t_i - t_j) \tag{5}$$

In this work, we used the non-uniformly spaced sample points (i.e, the Chebyshev-Gauss-Lobatto sample points) for calculation of weighting coefficients. These points are given by

$$t_i = T/2 \left[1 - \cos\left(\frac{(i-1)\pi}{m-1} \right) \right], \qquad i = 1,2,\ldots,m \tag{6}$$

where T is the time span.

3. Conversion of Blasius boundary-layer problem to a pair of initial-value problems

The boundary-layer problem (1) can be transformed to the following pair of initial-value problems (Na, 1979)

$$\begin{cases} g'''(\alpha) + 0.5 g(\alpha) g''(\alpha) = 0 \\ g(0) = g'(0) = 0, \quad g''(0) = 1 \end{cases} \tag{7}$$

and

$$\begin{cases} f'''(\eta) + 0.5f(\eta)f''(\eta) = 0 \\ f(0) = f'(0) = 0, \quad f''(0) = [g'(\infty)]^{-3/2} \end{cases} \tag{8}$$

4. DQ analogs of the pair initial-value problems

From equations (7) and (8), it may be seen that the only difference between two resultant initial-value problems is related to the value of the second-order derivative initial condition. Thus, the procedure for two initial-value problems is actually the same. To be applicable the procedure for both the initial-value problems, we consider the following equation

$$F'''(\eta) + 0.5F(\eta)F''(\eta) = 0 \tag{10}$$

with general initial conditions

$$F(0) = F_0, \quad F'(0) = F_0', \quad F''(0) = F_0'' \tag{11}$$

where F_0, F_0', and F_0'' are constants.

The third-order initial-value problem (10) can be converted into a set of first-order initial-value problems as in the following

$$\begin{cases} x' = y \\ y' = z \\ z' = -0.5xz \end{cases} \tag{12}$$

with initial conditions

$$x(0) = F_0, \quad y(0) = F_0', \quad z(0) = F_0'' \tag{13}$$

From the quadrature rule, equation (3), the first-order derivatives of functions x, y, and z can expressed as

$$x_i' = \sum_{j=1}^{m} A_{ij}x_j, \quad y_i' = \sum_{j=1}^{m} A_{ij}y_j, \quad z_i' = \sum_{j=1}^{m} A_{ij}z_j \tag{14}$$

Substituting equation (14) in (12) yields

$$\begin{cases} \sum_{j=1}^{m} A_{ij}x_j = y_i \\ \sum_{j=1}^{m} A_{ij}y_j = z_i \\ \sum_{j=1}^{m} A_{ij}z_j = -0.5x_iz_i \end{cases} \quad i = 1,2,\ldots,m \tag{15}$$

Applying the initial conditions (13) in (15) yields

$$\begin{cases} \sum_{j=2}^{m} A_{ij}x_j + A_{i1}F_0 = y_i \\ \sum_{j=2}^{m} A_{ij}y_j + A_{i1}F_0' = z_i \\ \sum_{j=2}^{m} A_{ij}z_j + A_{i1}F_0'' = -0.5x_iz_i \end{cases} \quad i = 2,3,\ldots,m \tag{16}$$

Equation (16) is a nonlinear system of algebraic equations which can be solved by iterative methods. In this work, we use the Newton-Raphson method to solve system (16). Our numerical experiment for the present problem showed that only 3-5 iterations are sufficient to achieve accurate solutions using the Newton-Raphson method.

5. A step-by-step DQ in time

When very long-term solutions are required, it is more convenient to apply the DQ method as a step-by-step time integration scheme. In this technique, the time domain of interest is first divided into several time elements. The DQ method in then applied to each time element independently. Note that the solutions at the end of each time element will be used as initial conditions for the next time element (for more details of this technique see Refs. (Guran and Ahmadi, 2011)).

6. Numerical results and discussion

As it was mentioned earlier, we should first determine the magnitude of L (defined in equation (9)). This parameter can be obtained using the solution of equation (7). To solve equation (7) using the scheme described in Sec. 5, we divide the time domain into n equal DQM time element with m sample time points (per DQM time element). The total number of sample time points and the average time step can be obtained as, respectively, (Guran and Ahmadi, 2011)

$$M_{tot} = n\,(m\text{-}1) + 1 \tag{17}$$

$$\Delta t = T/(\,M_{tot}\,\text{-}1) = T/(\,n\,(m\text{-}1)) \tag{18}$$

where T is the length of time span. Figure 1 presents the variations of $g(\alpha)$ with α for different values of n (number of time elements) and m (number of time points per time element). It can be seen that by increasing n and/or m, the DQ solutions are converged rapidly. Note that the DQ solution results at m time points are utilized to obtain the solutions at all the time points located in time interval $[0,\ t_m]$ (where t_m is the length of DQ time elements) via the Lagrange interpolation scheme. Thus we are able to find a continuous representation for the function $g(\alpha)$ using the Lagrange interpolation scheme.

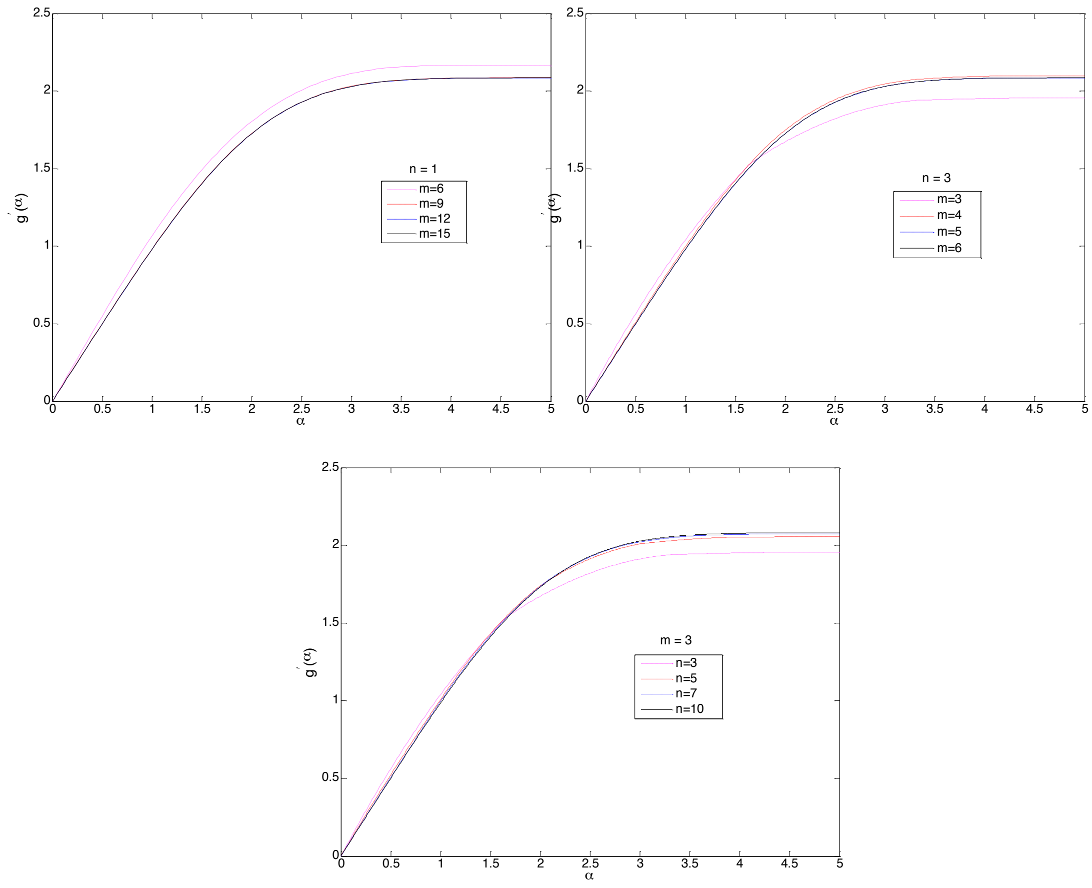

Fig. 1: Convergence and accuracy of DQ solutions with respect to the number of sample time points, m, and number of time elements, n.

It is interesting that the DQ method yields converged and rather accurate solutions using only m=3 time points. Note that m=3 is the smallest number of sample time points which can be used for the solution of present problem using the DQ method. This is due to the fact that the present problem is a third-order nonlinear equation and has three initial conditions given at the initial time point. It can also be seen from figure (1) that as α increases $g(\alpha)$ approaches to a constant value. This constant value is actually the magnitude of L. Note that in the cases shown in figure (1), the values of α is in the range $0 \le \alpha \le 5$. It is clear that to determine the magnitude of L ($= g(\infty)$), it in not required to solve the initial-value problem (7) at all the domain $0 \le \alpha \le \infty$. For example, as it can be seen from figure (1), one can solve the problem at the interval $0 \le \alpha \le 5$ to find an approximation of L. However, through numerical experiments we found that the most accurate L values can be obtained if the problem (7) is solved in the domain $0 \le \alpha \le 10$. Table 1 gives the results for L. An excellent rate of convergence can be observed. It can be seen that the DQ results are converged without instability for an increase in n and m. It can also be observed that by increasing the number of sample time points per time element, i.e., m, a smaller number of time elements, i.e., n, are required to obtain solutions with identical accuracies and vice versa (i.e., by increasing n a smaller m are required). However, when m is too small, the rate of convergence is slow and very large values of n are required to achieve accurate solutions.

Tab. 1: Convergence and accuracy of the DQ solution for $L = g'(10)$

n	m=7	m=9	m=11	m=13	m=15
2	2.07125099256072	2.08616508333382	2.08552625546298	2.08535915160828	2.08542031533568
3	2.08086107493813	2.08573633901879	2.08539005571989	2.08541016274351	2.08540913792848
5	2.08564569949467	2.08540031913574	2.08540939693684	2.08540917319786	2.08540917643640
10	2.08540977491746	2.08540917461438	2.08540917646167	2.08540917643749	2.08540917643791***
15	2.08540923283780	2.08540917639232	2.08540917643792*	2.08540917643791**	2.08540917643791
20	2.08540918655018	2.08540917643358	2.08540917643791	2.08540917643790	2.08540917643791
30	2.08540917732140	2.08540917643774	2.08540917643791	2.08540917643791	2.08540917643791

*Δt_{DQ}=0.06666, **Δt_{DQ}=0.05555, ***Δt_{DQ}=0.07143,

In other words, the rate of convergence of solutions is more sensitive to m than n. Thus, to obtain accurate solutions with a reasonable time step, one should first choose the correct value of m and then increase n to reach to required accuracy. From Table 1, it is also observed that the magnitude of L is found to be converged up to fifteen digits. In Table 2 the DQ solutions are compared with those of RK4. Note that the sizes of time steps for two schemes are set to be equal to provide a reasonable comparison between two schemes.

Tab. 2: Convergence and accuracy of DQ solution for $L = g'(10)$ *and comparison with RK4 solution*

Method	m=7	m=11	m=15
DQ (n=2)	2.07125099256072	2.08552625546298	2.08542031533568
RK4*	6.67115504023E+4	2.01369359611100	2.08539498816094
DQ (n=5)	2.08564569949467	2.08540939693684	2.08540917643640
RK4	2.08539777120960	2.08540721539089	2.08540861092003
DQ (n=10)	2.08540977491746	2.08540917646167	2.08540917643791
RK4	2.08540817139355	2.08540903063789	2.08540913673422
DQ (n=20)	2.08540918655018	2.08540917643791	2.08540917643791
RK4	2.08540910423540	2.08540916658843	2.08540917381891

*Note that the time steps for RK4 are so chosen that they are equal to those of DQM (i.e., $\Delta t_{RK4} = \Delta t_{DQM} = T/(n(m-1))$ and T=10 here)

To check the accuracy of DQ and RK4 solutions, shown in Tables 1 and 2, the problem is solved again using the RK4 and with very small time steps and the solutions are cited in Table 3.

Tab. 3: Convergence and accuracy of RK4 solution for $L = g'(10)$

Δt=0.1	Δt=0.01	Δt=0.002	Δt=0.001	Δt=0.0005
2.08540903063789	2.08540917642120	2.08540917643787	2.08540917643789	2.08540917643787

By comparing the DQ results with those of RK4, one can conclude that the DQ method needs much larger time steps to achieve accurate solutions. This illustrates the superiority of the DQ time integration method over the classical RK4. From the RK4 solutions shown in Table 2, one can also observe the numerical instability of RK4 when n=2 and m=7. In fact, when the time step is too large, the RK4 may encounter some numerical instability as seen in Table 2.

Tables 4-6 present convergence and accuracy of DQ time integration scheme for the solution of Blasius equation (i.e., equation (8)). A single time element is employed (i.e., n=1). The results of RK4 are also included for comparison. Rapid convergence and stability of the numerical solutions versus increasing DQ number of sample time points are obvious. Note that the DQ solutions are converged up to eight digits. Since the time steps are so large, the RK4 and DQ solutions match only to four decimal positions. This is due to the fact that the RK4 needs much smaller time step sizes than DQM to achieve results with identical accuracies.

Tab.4: Convergence and accuracy of DQ solution for $f(5)$ (n=1) and comparison with the RK4 solution

Method	*m*=6	*m*=7	*m*=9	*m*=11	*m*=13	*m*=15	*m*=20
DQ	3.2522253	3.2717815	3.2842311	3.2832062	3.2832779	3.2832734	3.2832736
RK4	3.2847702	3.2839304	3.2834541	3.2833406	3.2833037	3.2832890	3.2832778

Tab. 5: Convergence and accuracy of DQ solution for $f'(5)$ (n=1) and comparison with the RK4 solution

Method	*m*=6	*m*=7	*m*=9	*m*=11	*m*=13	*m*=15	*m*=20
DQ	0.9874834	0.9886335	0.9917923	0.9915237	0.9915431	0.9915418	0.9915419
RK4	0.9903180	0.9910017	0.9913909	0.9914847	0.9915157	0.9915282	0.9915381

Tab. 6: Convergence and accuracy of DQ solution for $f''(5)$ (n=1) and comparison with the RK4 solution

Method	*m*=6	*m*=7	*m*=9	*m*=11	*m*=13	*m*=15	*m*=20
DQ	0.0164112	0.0159920	0.0158999	0.0159073	0.0159067	0.0159068	0.0159068
RK4	0.0181004	0.0168513	0.0161596	0.0159995	0.0159482	0.0159279	0.0159125

Table 7 gives the convergence and accuracy of RK4 solutions with respect to time step sizes. The DQ results with Δt=1/4 =0.25 are also included for comparison. It can be seen that the DQ solutions with Δt=1/4 and RK4 solutions with Δt=1/20 are coincident to each other. Needless to say the DQ method is more efficient than RK4. We also compared our solutions with those of Howarth (Howarth, 1938) and we found that both results agree well with each other.

Tab. 7: Convergence and accuracy of RK4 for the solution of Blasius equation and comparisons with the DQ results

	Δt=1/4	Δt=1/7	Δt=1/10	Δt=1/12	Δt=1/16	Δt=1/20	DQ (Δt=1/4)
$f(5)$	3.28327697	3.28327398	3.28327374	3.28327370	3.28327367	3.28327367	3.28327367
$f'(5)$	0.99153882	0.99154159	0.99154183	0.99154186	0.99154189	0.99154190	0.99154190
$f''(5)$	0.01591136	0.01590723	0.01590690	0.01590684	0.01590681	0.01590680	0.01590680

Figure 2 presents the accuracy, convergence, and stability of DQ time integration method for computing the long-term solutions. It is seen that the DQ time integration scheme is also very efficient for time integration over long time duration. But care should be taken in choosing proper values of *n* and *m*. From figure 2 it is observed that when *m* is incorrectly chosen, the DQ time integration scheme may also be possible to yield inaccurate solutions. From the results shown in this figure and those cited in Tables 1-6, one can also concluded that the number of time points per time element, i.e., *m*, can not be too small. If *m* be too small, we then should use a large number of time element *n* to obtain accurate solutions and this increases the CPU time considerably. One the other hand, *m* can not be too large. If *m* be too large, the resulting DQ time integration scheme may be increasingly unstable. This is actually due to the fact that the DQ weighting coefficient matrices tend to be ill-conditioned by increasing the number of sample time points. The present authors recommend that *m* be in the range $5 \le m \le 15$. If *m* be in the above range, one needs not to be worried about efficiency and stability of DQ method. In Table 8 the DQ solutions with Δt = 0.34965 are compared with those of RK4 with Δt = 0.25 and Δt = 0.1 for different values of η. It can be seen that the DQ results with Δt = 0.34965 are better in accuracy than those obtained using RK4 with Δt = 0.25. Also, one can observe the numerical instability of RK4 for long-term solutions when Δt = 0.25. In other words, the RK4 solutions with Δt = 0.25 encounter a convergence problem and approach to infinity after η = 35. It can also be seen that the RK4 solutions with Δt = 0.1 are high accurate, since as η increases $f'(\eta)$ approaches to unity and $f''(\eta)$ approaches to zero. The DQ solutions with Δt = 0.34965 are also accurate, but their accuracy is less than those of RK4 with Δt = 0.1.

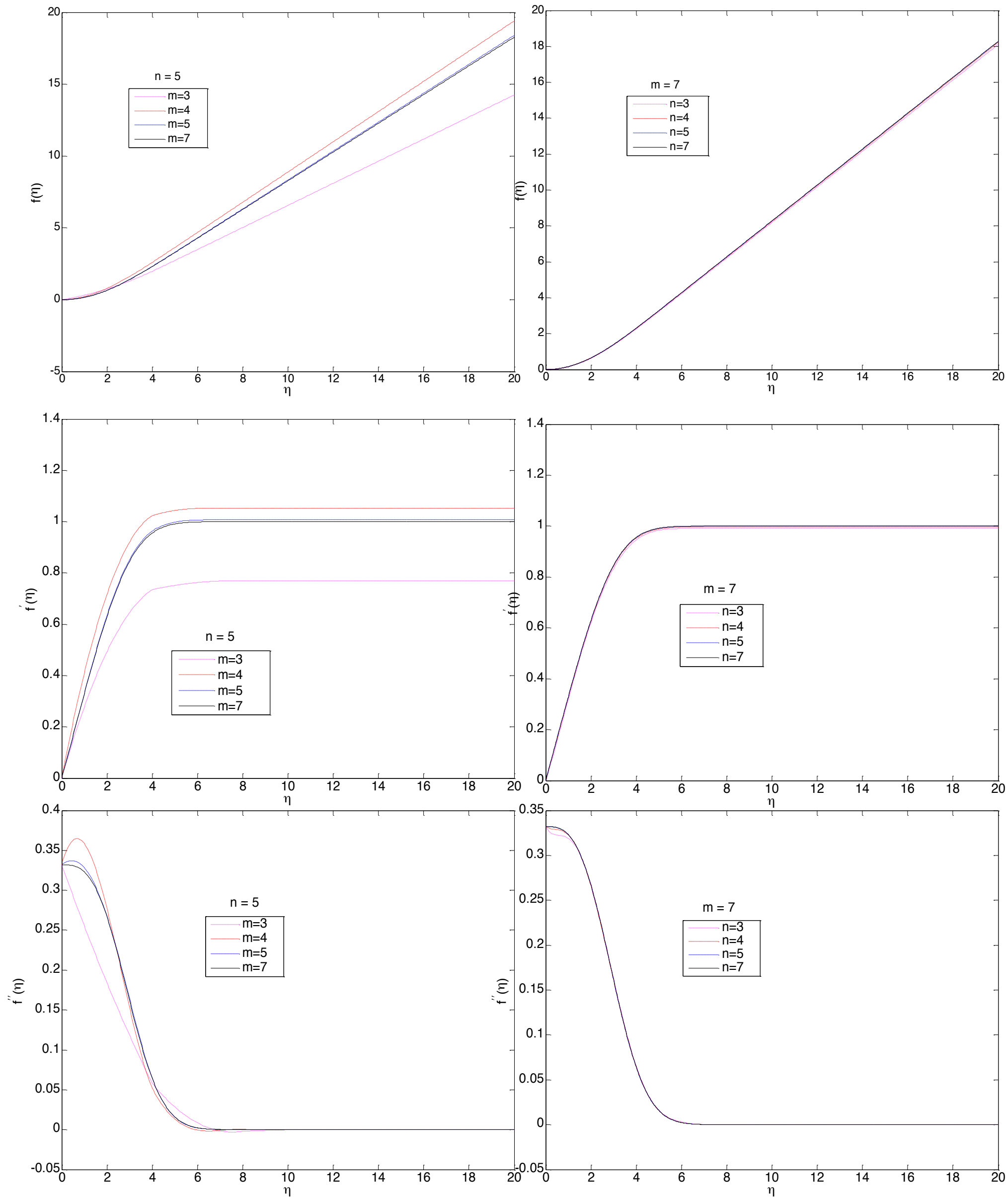

Fig. 2: Convergence and accuracy of the DQ method for the solutions of Blasius equation.

Tab. 8: Convergence and accuracy of RK4 for the solution of Blasius equation and comparisons with the DQ results

η	DQ(n=13,m=12: Δt = 0.34965)			RK4 (Δt = 0.25)			RK4 (Δt = 0.1)		
	$f(\eta)$	$f'(\eta)$	$f''(\eta)$	$f(\eta)$	$f'(\eta)$	$f''(\eta)$	$f(\eta)$	$f'(\eta)$	$f''(\eta)$
0.5	0.0414928	0.1658853	0.3309111	0.0414937	0.1658852	0.3309110	0.0414928	0.1658853	0.3309110
2.5	0.9963114	0.7512599	0.2174116	0.9963139	0.7512585	0.2174121	0.9963112	0.7512597	0.2174116
5.0	3.2832745	0.9915421	0.0159068	3.2832769	0.9915388	0.0159113	3.2832737	0.9915418	0.0159069
7.5	5.7792195	0.9999821	5.5261E-5	5.7792192	0.9999811	5.5849E-5	5.7792180	0.9999818	5.5273E-5
10	8.2792143	1.0000002	8.4437E-9	8.2792122	0.9999994	9.4693E-9	8.2792123	0.99999998	8.4588E-9
15	13.279215	1.0000002	4.361E-16	13.279209	0.9999994	2.862E-19	13.279212	0.99999998	1.736E-20
20	18.279217	1.0000002	6.458E-25	18.279207	0.9999994	9.377E-29	18.279212	0.99999998	1.686E-37
25	23.279218	1.0000002	1.289E-28	23.279204	0.9999994	3.180E-31	23.279212	0.99999998	1.667E-59
30	28.279219	1.0000002	2.892E-33	28.279201	0.9999994	9.181E-26	28.279212	0.99999998	1.404E-85
35	33.279220	1.0000002	8.721E-42	33.279198	0.9999994	2.333E-13	33.279212	0.99999998	9.27E-114
40	38.279221	1.0000002	2.454E-45	-9.8323E2	-9.8592E4	-2.1425E7	38.279212	0.99999998	5.78E-141
45	43.279221	1.0000002	2.417E-49	*- Infinity*	*- Infinity*	*- Infinity*	43.279211	0.99999998	7.72E-164
50	48.279223	1.0000002	1.151E-53	*- Infinity*	*- Infinity*	*- Infinity*	48.279211	0.99999998	6.34E-180

7. Conclusion

In this paper, the DQ method is employed to solve the well-known Blasius boundary layer problem. The efficiency, accuracy, and convergence of the DQ method are investigated and analyzed. It is shown that the DQ method can predict the behavior of Blasius boundary layer accurately. Numerical comparisons between the DQ time integration method and RK4 reveal that the DQ method is a promising and effective tool for handling the nonlinear systems of ordinary differential equations.

References

Abbasbandy S. A (2007) numerical solution of Blasius equation by Adomian's decomposition method and comparison with homotopy perturbation method. Chaos Solitons & Fractals; 31: pp. 257-60.

Ahmad F. (2007) Degeneracy in the Blasius problem. *Electron J Differ Equations*; 92: pp. 1-8.

Ahmad F, Al-Barakati WH. (2009) An approximate analytic solution of the Blasius Problem. Commun Nonlinear Sci Numer Simul;14: pp. 1021-4.

Allan FM, Syam MI. On the analytic solutions of the nonhomogeneous Blasius problem. *J Comput Appl Math* 2005;182:362-71.

Asaithambi A. (2005) Solution of the Falkner-Skan equation by recursive evaluation of Taylor coefficients. *J Comput Appl Math*;176: pp. 203-14

Bellman RE, Casti J. (1971) Differential quadrature and long term integrations. *J Math Anal Appl*; 34: pp. 235-8.

Bert CW, (1996) Malik M. Differential quadrature method in computational mechanics: a review. *ASME J Appl Mech Rev*;49: pp. 1-28.

Cortell R. (2005) Numerical solution of the classical Blasius flat-plate problem. *Appl Math Comput*; pp. 170:706-10.

Fang T, Guo F, Lee CF. (2006) A note on the extended Blasius Problem. Appl Math Lett; 19: pp. 613-7.

Guran A, Ahmadi G., (2011) Computational Approaches to the Lorenz System, *Proceedings of AMiTaNS'11, American Institute of Physics*.

Guran, A. , J. Gewinner (2011) A fast convergent solution of Blasius equation by means of differential quadrature, *II Annual Meeting of the Georgian Mechanical Union*, 15-17 december, Tbilisi, Georgia.

He JH. (2003) A simple perturbation approach to Blasius equation. Appl Math Comput; 140: pp. 217-22.

He JH. (2004) Comparison of homotopy perturbation method and homotopy analysis method. *Appl Math Comput*; pp. 156:527-39.

He J. (1997) A new approach to nonlinear partial differential equations. Commun Nonlinear Sci Numer Simul;2(4): pp. 230-35.

He J. (1999) Approximate analytical solution of Blasius equation. Commun Nonlinear Sci Numer Simul; 4(1): pp.75-8.

Howarth L. (1938) On the solution of the laminar boundary layer equations. *Proc Roy Soc London*; 164: pp. 547-79

Liao SJ. A (1997) kind of approximate solution technique which does not depend upon small parameters, Part 2: an application in fluid mechanics. *Int J Nonlinear Mech*; 32(5): pp. 815-22.

Liao SJ. (1998) An explicit, totally analytic solution of laminar viscous flow over a semi-infinite flat plate. Commun Nonlinear Sci Numer Simul; 3(2): pp. 53-57.

Liao SJ. A (1999) uniformly valid analytic solution of two dimensional viscous flow over a semi-infinite flat plate. *J Fluid Mech*; 385: pp. 101-28.

Na TY. (1979) Computational methods in engineering boundary value problems. Academic Press;.

Quan JR, Chang CT. (1989a) New insights in solving distributed system equations by the quadrature methods, Part I: Analysis. Comput Chem Eng; 13: pp. 779-788.

Schiliching H. (2004) Boundary layer theory. 8th ed. McGraw-Hill Inc.

Shu C. (2000) Differential quadrature and its application in engineering. New York: Springer publication.

Wang L. (2004) A new algorithm for solving classical Blasius equation. *Appl Math Comput*; 157: pp. 1-9.