Ain Shams Engineering Journal 13 (2022) 101713

Contents lists available at ScienceDirect

Ain Shams Engineering Journal

journal homepage: www.sciencedirect.com

A comparative analysis of numerical methods applied to nonsimilar boundary layer-derived infinite series equations

O.M. Amoo [a,*], R.O. Fagbenle [a], M.O. Oyewola [a,b]

[a] Department of Mechanical Engineering, University of Ibadan, Oyo State, Nigeria
[b] School of Mechanical Engineering, Fiji National University, Samabula, Fiji

ARTICLE INFO

Article history:
Received 18 July 2021
Revised 24 September 2021
Accepted 8 January 2022

Keywords:
Boundary layer infinite series ODEs
Finite-element method
Runge–Kutta method
Newton–Raphson iteration
Numerical experiment
Merk–Chao–Fagbenle method

ABSTRACT

Circa 1958, Merk propounded a boundary layer procedure valid for both similarity and nonsimilarity problems. It was notably the first asymptotic expansion to account for boundary layer nonsimilarity. Due to an unfortunate error in the procedure, the method was later ameliorated by Chao and Fagbenle and is today commonly referred to as the Merk-Chao-Fagbenle (MCF) method.

The objective in this work is an investigation to compare two numerical methods—the single-step multistage method known as the fourth-order Runge–Kutta method with the Newton–Raphson shooting iteration as the root-finding algorithm (RK + Newton), and the finite-element method (FEM). In so doing, the characteristic nonsimilar perturbation series boundary layer problem of Merk, Chao, and Fagbenle is employed as a model. The novelty is to assess critical numerical performance indices of both numerical techniques, which constitutes an undertaking that has yet to be elucidated, as far as the authors are aware. Thus, this work departs from the norm and advances beyond previous efforts in literature by emphasizing the numerical performances of two numerical methods rather than the sundry boundary layer solutions, which in any case have been presented in previous works.

It is found that the numerical results obtained using both methods correlate very well with highly accurate benchmarked results. The role of each method to evaluate the velocity functions (fs) and temperature functions (θs) is visually depicted and described numerically. The computation and central processing unit (CPU) times for the evaluation of f_0, f_1, f_2, f_3, and $\theta_0, \theta_1, \theta_2, \theta_3$ according to both the FEM and the RK + Newton methods for element sizes of 10^{-3} and 10^{-4} reveal that the computation time of RK + Newton is significantly less than that of the FEM for both values of the element size. On the other hand, the CPU time of RK + Newton is less than that of the FEM for f_0, and θ_0 only. However, overall, FEM is much more accurate.

1. Introduction

Boundary layer flows are ubiquitous in many transport processes. With the exponential growth of boundary layer research, especially regarding advanced fluids known as nanofluids, there have also been advances in numerical methods to solve the various systems of differential equations that emerge.

Series-based expansion techniques, which date back to Blasius, have continuously being developed. These were characteristically established to solve nonsimilar boundary layer problems. They are advantageous because many coefficients can be solved once and for all. The corrected perturbation series procedure of Merk, Chao, and Fagbenle (MCF, as it is commonly regarded) [1,2], which is valid for both similar and nonsimilar laminar boundary layer transfer problems, is a highly benchmarked technique [3]. Notably, the more advanced form or examination of boundary layer theory invokes an asymptotic framework known as the method of matched asymptotic expansions of the series [4], which is the case in the MCF procedure. Most boundary layer studies in present literature, for the sake of mathematical simplicity, consider similar flows, as opposed to the more complex and physically meaningful nonsimilar flows.

In nonsimilar flows, the external velocity rarely varies according to x; instead, it is a function of both x and y. The surface boundary conditions may not satisfy the requirements of similarity even if the external velocity does. For nonsimilar flows, in comparison to similar flows, the flow quantities are a function of the streamwise

* Corresponding author.
E-mail address: oamoo@alumni.ucla.edu (O.M. Amoo).

https://doi.org/10.1016/j.asej.2022.101713
2090-4479/

Abbreviations and Nomenclature

MCF	Merk–Chao–Fagbenle
FEM	Finite-element method
RK + Newton	Runge–Kutta method with Newton–Raphson iteration
ODE	Ordinary differential equations
PDE	Partial differential equations

Pr	Prandtl number (-)
Λ	wedge variable at the wall for velocity (-), defined in [2]
J	Jacobian (-)
f, f', f''	dimensionless velocity functions (-)
$\theta, \theta', \theta''$	dimensionless velocity functions (-)
η	normal coordinate, a function of x, y (-)

x-direction; that is, the flow quantities change along the streamwise x-direction. Thus, nonsimilar terms or streamwise derivatives on the right-hand side of the equations essentially include the effects of the streamwise boundary layer history and aid in eliminating uncertainties found in many studies in the literature. This is also important because the heat transfer coefficient in a laminar boundary layer is strongly sensitive to the streamwise (or upstream) history of the flow, which is often ignored or unaccounted for in considerations of similar flows. The MCF method is as uniquely suitable for nonsimilar flows as it is for similar flows [5–7], which constitutes an important feature that is also partly responsible for the success of the method. It is a wedge method for obtaining locally similar solutions of the boundary layer equations. For a more expansive discussion of the MCF procedure, the reader is referred to recent work by Fagbenle et al. [8].

The numerical solution of boundary layer differential equations is a fundamental problem in fluid mechanics, such that not all numerical methods may be applicable to or efficient when employed against particular problems. Boundary layer problems may either involve similarity solutions or nonsimilarity solutions. Characteristically, however, most fluid boundary layer problems are nonsimilar. The nonsimilar problems are more complex to solve, but notably more generally valid industrially.

There have been many recent similar and nonsimilar boundary layer studies. In reviewing and discussing these works, we emphasize numerical techniques employed, while de-emphasizing the sundry boundary layer findings. The sundry boundary layer solutions of the MCF problem may be found in [2] and are not repeated here.

Hayat et al. solved their boundary layer problem using the recently proposed homotopy analysis method, indicating its flexibility toward series solutions [9]. Khan et al. employed the usual shooting method to obtain sundry boundary layer solutions to the problem of incompressible, steady magnetohydrodynamic (MHD) Casson flow with the presence of chemical reactions [10]. In another work, regarding both boundary layer and entropy generation for the non-Newtonian Jeffrey fluid, a shooting technique is also employed to obtain solutions to the problem [11]. A dimensionless set of nonlinear PDEs obtained from the usual transformation of the problem of carbon nanotube nanofluid transfer over stretching surfaces is solved using the finite difference method in [12] to obtain solutions, indicating that the Nusselt number decreases for larger values of nanoparticle volume fraction. A fourth-order Runge–Kutta integration scheme is employed for the solution of nonlinear ODEs to emanate from the problem of non-Newtonian nanofluid flow with applications for optimizing solar energy [13].

Another variant of the homotopy analysis method is the homotopy perturbation method, which was employed in [14] to solve entropy generation of three-dimensional hybrid nanofluid flow, showing increases in the performance of both the Nusselt number and skin friction coefficient as compared to a base fluid of water. A fifth-order Runge–Kutta method was employed regarding a bio-fluid problem for nanoparticle three-dimensional crossflow over a cylinder, revealing enhancements in bioconvection [15].

The FEM was employed in [16] for a cavity enclosure problem with hybrid nanofluids, showing significant improvements in the Nusselt number. The FEM is similarly employed in [17].

A fifth-order Runge –Kutta –Fehlberg numerical method was employed in [18–21] for various problems where the boundary layer is applied.

To improve the numerical knowledge of these types of equations, two distinct numerical methods are compared in the computation of the differential equations—the single-step multistage method known as the fourth-order Runge –Kutta method with the Newton –Raphson shooting iteration as the root-finding algorithm (RK + Newton), and the finite-element method (FEM). The former has been widely applied by researchers, while the latter was recently developed in the literature, for the first time, in the context of the MCF-type of equations. This work numerically computes the solutions using the two aforementioned numerical methods and makes comparisons accordingly.

Numerical computations and analyses are par for the course in engineering. In fluid mechanics, it is a critical tool for the assessment and understanding of flow phenomena, when done correctly. Numerical analysis is a practical applied subject of mathematics. Numerical methods for solving differential equations comprise an important theme and tool in theoretical fluid mechanics. Numerical fluid mechanics is an area that continues to gain significance. In boundary layers, many numerical techniques are available to resolve the complex differential equations that emanate from the PDE-to-ODE transformations. In general, numerous techniques exist, such as the Keller box method, finite difference, finite element, and the popular Runge–Kutta method, to name a few. Any of these methods may be used to solve boundary layer problems, however, and as previously mentioned, not all numerical methods may be applicable or efficient when employed against particular problems.

Table 1 depicts the numerical techniques that have to date been employed for some research on the MCF-type equations. It is observed that the fourth-order Runge–Kutta technique is popularly used by researchers to solve the differential equations and advance the solution, whereas various root finding algorithms (e.g., Newton–Raphson) are employed. The Runge–Kutta method is a superbly honed approach to solve a large class of linear and nonlinear differential equations and continues to receive wide acceptance for use in both external and internal flows. The Runge–Kutta method has been supported and made famous by many textbooks and research papers. In principle, any root finding algorithm may potentially be used. Some other notable root finding algorithms are the bisection method, the secant method, and the Nachtsheim–Swigert method [22], to name a few. The reader may also refer to Rogers [23], Carnahan [24], and Kutz [25]. Carnahan [23], Kutz [24], and Rogers [25]. The choice of the root finding algorithm has no bearing per se on the numerical results; however, they differ by the number of iterations or computations employed to

Table 1
Salient summary of numerical techniques employed by researchers to solve the MCF-type differential equations.

No.	Numerical Method	Reference
1	Fourth-order Runge–Kutta with step size control and Newton–Raphson method	[27]
2	Fourth-order Runge–Kutta	[28]
3	A multiple shooting method using subroutine DTPTB from the International Mathematical and Statistics Library	[29]
4	An explicit Runge–Kutta–Fehlberg or *RK*45 method	[30]
5	Fourth-order Runge–Kutta with an automatic interval-halving based step-size adjustment and the shooting method	[31]
6	Fouth-order Runge–Kutta with Newton–Raphson	[32]
7	Fouth-order Runge–Kutta	[33]
8	Fouth-order Runge–Kutta	[34]
9	Fouth-order Runge–Kutta using double precision with Adams–Moulton predictor–corrector method	[35]
10	Fouth-order Runge–Kutta with a least-squares convergence criterion for zeroth-order equations and Newton–Raphson iteration scheme for higher-order equations.	[36]
11	Fouth-order Runge–Kutta with Newton–Raphson shooting technique and method of continuation	[37]
12	Fouth-order Runge–Kutta with a uniform step size	[38]
13	Nachtsheim–Swigert shooting technique	[39]
14	Fouth-order Runge–Kutta	[40]
15	Fouth-order Runge–Kutta with Nachtsheim–Swigert technique	[41]
16	Fouth-order Runge–Kutta	[42]
17	Fouth-order Runge–Kutta with Newton iterative scheme	[43]

obtain the numerical result [8]. Generally, the accuracy of the fourth-order Runge–Kutta integration scheme with any root finding algorithm for boundary layer differential equations is well within allowable engineering limits or requirements, hence its wide applicability to engineering problems in general and fluid mechanics problems in particular. Significant efforts have indeed been expended to develop numerical approaches for a wide array of ordinary differential equations in years past. Recently, for the first time, a FEM technique for solving the MCF-type equations (or higher-order approximations of MCF-type equations) was advanced by Amoo et al. [26]. By contrast, however, relatively little has been done about assessing the performance of the numerical methods employed for solving MCF-type equations, thus creating a knowledge gap to be filled.

Against this background, it is evident that researchers have employed several numerical techniques for which the numerical performances of the numerical methods have not clearly been demonstrated.

Motivated by the prominent role numerical methods play in boundary layer studies, the present purpose and novelty highlights key performance indices of the FEM technique described in Amoo et al. [26] with the popularly employed RK + Newton technique to solve a coupled system of nonlinear ordinary differential equations. Therefore, this effort is a necessary undertaking in peer-reviewed research that sheds light on the efficiency of numerical methods for particular types of problems or parameters, and on opportunities for future development. In any event, the research is pedagogical, explaining several technical aspects associated with the performances of both numerical methods, thus providing insights hitherto unknown.

2. Method of solution using fourth-order Runge–Kutta with Newton–Raphson shooting iteration

As indicated previously, the aim is to experiment numerically and compare the performances of two methods - the finite element method described in [26] and the Runge–Kutta method with Newton–Raphson shooting iteration (RK + Newton). In this section, we apply the RK + Newton procedure for solving the following MCF coupled systems of ordinary differential equations with corresponding boundary conditions [2]:

$$\begin{cases} f_0''' + f_0 f_0'' + \Lambda[1-(f_0')^2] = 0, \\ f_0(0) = f_0'(0) = 0,\ f_0'(\infty) = 1, \end{cases}$$

$$\begin{cases} f_1''' + f_0 f_1'' - 2(1+\Lambda) f_0' f_1' + 3 f_0'' f_1 = J(f_0', f_0), \\ f_1(0) = f_1'(0) = 0,\ f_1'(\infty) = 0, \end{cases}$$

$$\begin{cases} f_2''' + f_0 f_2'' - 2(2+\Lambda) f_0' f_2' + 5 f_0'' f_2 = f_0' f_1' - f_0'' f_1, \\ f_2(0) = f_2'(0) = 0,\ f_2'(\infty) = 0, \end{cases}$$

$$\begin{cases} f_3''' + f_0 f_3'' - 2(2+\Lambda) f_0' f_3' + 5 f_0'' f_3 = J(f_1', f_0) + J(f_0', f_1) + (2+\Lambda)(f_1')^2 - 3 f_1 f_1'', \\ f_3(0) = f_3'(0) = 0,\ f_3'(\infty) = 0, \end{cases}$$

$$\begin{cases} Pr^{-1}\theta_0'' + f_0 \theta_0' = 0, \\ \theta_0(0) = 0,\ \theta_0(\infty) = 1, \end{cases}$$

$$\begin{cases} Pr^{-1}\theta_1'' + f_0 \theta_1' - 2 f_0' \theta_1 = J(\theta_0, f_0) - 3 f_1 \theta_0', \\ \theta_1(0) = 0,\ \theta_1(\infty) = 0, \end{cases}$$

$$\begin{cases} Pr^{-1}\theta_2'' + f_0 \theta_2' - 4 f_0' \theta_2 = f_0' \theta_1 - f_1 \theta_0' - 5 f_2 \theta_0', \\ \theta_2(0) = 0,\ \theta_2(\infty) = 0, \end{cases}$$

$$\begin{cases} Pr^{-1}\theta_3'' + f_0 \theta_3' - 4 f_0' \theta_3 = J(\theta_1, f_0) + J(\theta_0, f_1) + 2 f_1' \theta_1 - 3 f_1 \theta_1' - 5 f_3 \theta_0', \\ \theta_3(0) = 0,\ \theta_3(\infty) = 0, \end{cases}$$

Here,

$$J(f_0', f_0) := \frac{\partial(f_0', f_0)}{\partial(\Lambda, \eta)} = \frac{\partial f_0'}{\partial \Lambda} \cdot \frac{\partial f_0}{\partial \eta} - \frac{\partial f_0'}{\partial \eta} \cdot \frac{\partial f_0}{\partial \Lambda}$$

is the Jacobian.

The preceding equations (attributable to [2]) emanate from routine boundary layer transformations for an incompressible laminar two-dimensional or axisymmetric steady flow about a body of revolution. While it is possible for an incompressible fluid to have variable density, the problem assumes unit density, for simplicity. The transformed MCF equations are nonlinear ODEs. They are also one dimensional (or 1-D), meaning there is one independent variable and there is one dependent variable. The geometry involved is the domain of the independent variable, which is the simple real semi-axis.

Also, the preceding equations are a set of singularly perturbed problems expressed to four-terms of the series. Notably, the analytical technique of singular perturbation theory (SPT) has a lengthy history and has been applied in different ways in different fields. A consequence of SPT, as observed from the preceding equations, yields a sequence or hierarchy of asymptotic approximations, each with a higher-order of accuracy than its predecessor. The development of numerical methods for these higher-order approximations continues to be an active area of research.

The proposition behind combining the two methods, i.e., Runge–Kutta and Newton–Raphson, is the following. The third-order differential equation with respect to the velocity function f_0 is accompanied by two initial conditions on f_0 and f_0', as well as with a condition on f_0' at infinity. In another regard, in order to be able to apply the Runge–Kutta method to approximate f_0, we need to specify the third initial condition on f_0''. To do so, we employ the Newton–Raphson method to the equation

$1 - f_0'(\infty) = 0.$

Recall that the Newton–Raphson method suggests an iterative process for approximating real roots of nonlinear algebraic equations of the form

$F(x) = 0.$

The n^{th} step of the iterative process is given by

$$x_{n+1} = x_n - \frac{F(x_n)}{F'(x_n)}.$$

It should be remarked that in this particular case, any root finding algorithm may be used, but in general, convergence problems may occur. The Newton–Raphson algorithm is an iterative procedure that is particularly quick, having rapid convergence and used by the majority of computational fluid dynamics codes. Thus, after approximating the initial value f_0'', we obtain a consistent initial-value problem for a third-order nonlinear differential equation with respect to f_0, which can then be solved numerically by employing the fourth-order Runge–Kutta method. For the general steps of the Runge–Kutta method, the reader is referred to [44]. Ascher and Petzold (1998). In the following, we show the key steps applied to discretize the preceding coupled systems of ordinary differential equations.

For f_0, we denote,

$f_0' = U, \quad f_0'' = U' = G.$

Then,

$G' = -f_0 G - \Lambda\left[1 - (U)^2\right].$

Therefore, the discretization coefficients are given as,

$k_{11} = U_n, \quad k_{12} = G_n, \quad k_{13} = -f_{0n}G_n - \Lambda\left[1 - (U_n)^2\right], \quad k_{21} = U_n + d\eta\frac{k_{12}}{2},$

$k_{22} = G_n + d\eta\frac{k_{13}}{2},$

$k_{23} = -\left(f_{0n} + d\eta\frac{k_{11}}{2}\right)\left(G_n + d\eta\frac{k_{13}}{2}\right) - \Lambda\left[1 - \left(U_n + d\eta\frac{k_{12}}{2}\right)^2\right],$

$k_{31} = U_n + d\eta\frac{k_{22}}{2}, \quad k_{32} = G_n + d\eta\frac{k_{23}}{2},$

$k_{33} = -\left(f_{0n} + d\eta\frac{k_{21}}{2}\right)\left(G_n + d\eta\frac{k_{23}}{2}\right) - \Lambda\left[1 - \left(U_n + d\eta\frac{k_{22}}{2}\right)^2\right],$

$k_{41} = U_n + d\eta k_{32}, \quad k_{42} = G_n + d\eta k_{33},$

$k_{43} = -(f_{0n} + d\eta k_{31})(G_n + d\eta k_{33}) - \Lambda\left[1 - (U_n + d\eta k_{32})^2\right],$

where the unknown coefficients are defined as follows:

$$f_{0(n+1)} = f_{0n} + \frac{d\eta}{6}(k_{11} + 2k_{21} + 2k_{31} + k_{41}),$$

$$U_{n+1} = U_n + \frac{d\eta}{6}(k_{12} + 2k_{22} + 2k_{32} + k_{42}),$$

$$G_{n+1} = G_n + \frac{d\eta}{6}(k_{13} + 2k_{23} + 2k_{33} + k_{43}).$$

The update of the iteration parameter is defined as follows:

$\eta_{n+1} = \eta_n + d\eta.$

Similarly, the procedure proceeds with the solution of the initial-value problem for f_1. The steps of the Runge–Kutta method are as follows:

$f_1' = U, \quad f_1'' = U' = G,$

$$G' = -f_0(\Lambda,\eta)G + 2(1+\Lambda)f_0'(\Lambda,\eta)U - 3f_0''(\Lambda,\eta)f_1 + \frac{\partial(f_0', f_0)}{\partial(\Lambda,\eta)}.$$

Therefore, the discretization coefficients are given as

$k_{11} = U_n, \quad k_{12} = G_n,$

$$k_{13} = -f_0(\Lambda,\eta_n)G_n + 2(1+\Lambda)f_0'(\Lambda,\eta_n)U_n - 3f_0''(\Lambda,\eta_n)f_{1n} + \frac{\partial(f_0', f_0)}{\partial(\Lambda,\eta_n)},$$

$$k_{21} = U_n + d\eta\frac{k_{12}}{2}, \quad k_{22} = G_n + d\eta\frac{k_{13}}{2},$$

$$k_{23} = -f_0\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(G_n + d\eta\tfrac{k_{13}}{2}\right) + 2(1+\Lambda)f_0'\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(U_n + d\eta\tfrac{k_{12}}{2}\right) - 3f_0''\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(f_{1n} + d\eta\tfrac{k_{11}}{2}\right) + \frac{\partial(f_0' f_0)}{\partial\left(\Lambda,\eta_n+\frac{d\eta}{2}\right)},$$

$$k_{31} = U_n + d\eta\frac{k_{22}}{2}, \quad k_{32} = G_n + d\eta\frac{k_{23}}{2},$$

$$k_{33} = -f_0\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(G_n + d\eta\tfrac{k_{23}}{2}\right) + 2(1+\Lambda)f_0'\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(U_n + d\eta\tfrac{k_{22}}{2}\right) - 3f_0''\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(f_{1n} + d\eta\tfrac{k_{21}}{2}\right) + \frac{\partial(f_0' f_0)}{\partial\left(\Lambda,\eta_n+\frac{d\eta}{2}\right)},$$

$$k_{41} = U_n + d\eta k_{32}, \quad k_{42} = G_n + d\eta k_{33},$$

$$k_{43} = -f_0(\Lambda,\eta_n + d\eta)(G_n + d\eta k_{33}) + 2(1+\Lambda)f_0'(\Lambda,\eta_n + d\eta)(U_n + d\eta k_{32}) - 3f_0''(\Lambda,\eta_n + d\eta)(f_{1n} + d\eta k_{31}) + \frac{\partial(f_0' f_0)}{\partial(\Lambda,\eta_n+d\eta)},$$

where the unknown coefficients are defined as follows:

$$f_{1(n+1)} = f_{1n} + \frac{d\eta}{6}(k_{11} + 2k_{21} + 2k_{31} + k_{41}),$$

$$U_{n+1} = U_n + \frac{d\eta}{6}(k_{12} + 2k_{22} + 2k_{32} + k_{42}),$$

$$G_{n+1} = G_n + \frac{d\eta}{6}(k_{13} + 2k_{23} + 2k_{33} + k_{43}).$$

Similarly, for f_2,

$f_2' = U, \quad f_2'' = U' = G,$

$$G' = -f_0(\Lambda,\eta)G + 2(2+\Lambda)f_0'(\Lambda,\eta)U - 5f_0''(\Lambda,\eta)f_2 + f_0'(\Lambda,\eta)f_1'(\Lambda,\eta) - f_0''(\Lambda,\eta)f_1(\Lambda,\eta),$$

the discretization coefficients are given as

$k_{11} = U_n, \quad k_{12} = G_n,$

$$k_{13} = -f_0(\Lambda,\eta_n)G_n + 2(2+\Lambda)f_0'(\Lambda,\eta_n)U_n - 5f_0''(\Lambda,\eta_n)f_{2n} + f_0'(\Lambda,\eta_n)f_1'(\Lambda,\eta_n) - f_0''(\Lambda,\eta_n)f_1(\Lambda,\eta_n),$$

$$k_{21} = U_n + d\eta\frac{k_{12}}{2}, \quad k_{22} = G_n + d\eta\frac{k_{13}}{2},$$

$$k_{23} = -f_0\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(G_n+d\eta\tfrac{k_{13}}{2}\right)+2(2+\Lambda)f_0'\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(U_n+d\eta\tfrac{k_{12}}{2}\right)- -5f_0''\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(f_{2n}+d\eta\tfrac{k_{11}}{2}\right)+f_0'\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)f_1'\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)- -f_0''\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)f_1\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right),$$

$$k_{31}=U_n+d\eta\frac{k_{22}}{2},\quad k_{32}=G_n+d\eta\frac{k_{23}}{2},$$

$$k_{33} = -f_0\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(G_n+d\eta\tfrac{k_{23}}{2}\right)+2(2+\Lambda)f_0'\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(U_n+d\eta\tfrac{k_{22}}{2}\right)- -5f_0''\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(f_{2n}+d\eta\tfrac{k_{21}}{2}\right)+f_0'\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)f_1'\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)- -f_0''\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)f_1\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right),$$

$$k_{41}=U_n+d\eta k_{32},\quad k_{42}=G_n+d\eta k_{33},$$

$$k_{43} = -f_0(\Lambda,\eta_n+d\eta)(G_n+d\eta k_{33})+2(2+\Lambda)f_0'(\Lambda,\eta_n+d\eta)(U_n+d\eta k_{32})- -5f_0''(\Lambda,\eta_n+d\eta)(f_{2n}+d\eta k_{31})+f_0'(\Lambda,\eta_n+d\eta)f_1'(\Lambda,\eta_n+d\eta)- -f_0''(\Lambda,\eta_n+d\eta)f_1(\Lambda,\eta_n+d\eta),$$

where the unknown coefficients are defined as follows:

$$f_{2(n+1)}=f_{2n}+\frac{d\eta}{6}(k_{11}+2k_{21}+2k_{31}+k_{41}),$$

$$U_{n+1}=U_n+\frac{d\eta}{6}(k_{12}+2k_{22}+2k_{32}+k_{42}),$$

$$G_{n+1}=G_n+\frac{d\eta}{6}(k_{13}+2k_{23}+2k_{33}+k_{43}).$$

Similarly, for f_3,

$f_3'=U,\quad f_3''=U'=G,$

$$G' = -f_0(\Lambda,\eta)G+2(2+\Lambda)f_0'(\Lambda,\eta)U-5f_0''(\Lambda,\eta)f_3+\tfrac{\partial(f_1'f_0)}{\partial(\Lambda,\eta)}+\tfrac{\partial(f_0'f_1)}{\partial(\Lambda,\eta)}+ +(2+\Lambda)\left(f_1'(\Lambda,\eta)\right)^2-3f_1(\Lambda,\eta)f_1''(\Lambda,\eta),$$

the discretization coefficients are given as

$k_{11}=U_n,\quad k_{12}=G_n,$

$$k_{13} = -f_0(\Lambda,\eta_n)G_n+2(2+\Lambda)f_0'(\Lambda,\eta_n)U_n-5f_0''(\Lambda,\eta_n)f_{3n}+\tfrac{\partial(f_1'f_0)}{\partial(\Lambda,\eta_n)}+ +\tfrac{\partial(f_0'f_1)}{\partial(\Lambda,\eta_n)}+(2+\Lambda)\left(f_1'(\Lambda,\eta_n)\right)^2-3f_1(\Lambda,\eta_n)f_1''(\Lambda,\eta_n),$$

$$k_{21}=U_n+d\eta\frac{k_{12}}{2},\quad k_{22}=G_n+d\eta\frac{k_{13}}{2},$$

$$k_{23} = -f_0\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(G_n+d\eta\tfrac{k_{13}}{2}\right)+2(2+\Lambda)f_0'\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(U_n+d\eta\tfrac{k_{12}}{2}\right)- -5f_0''\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(f_{3n}+d\eta\tfrac{k_{11}}{2}\right)+\tfrac{\partial(f_1'f_0)}{\partial(\Lambda,\eta_n+\frac{d\eta}{2})}+\tfrac{\partial(f_0'f_1)}{\partial(\Lambda,\eta_n+\frac{d\eta}{2})}+ +(2+\Lambda)\left(f_1'\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\right)^2-3f_1\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)f_1''\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right),$$

$$k_{31}=U_n+d\eta\frac{k_{22}}{2},\quad k_{32}=G_n+d\eta\frac{k_{23}}{2},$$

$$k_{33} = -f_0\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(G_n+d\eta\tfrac{k_{23}}{2}\right)+2(2+\Lambda)f_0'\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(U_n+d\eta\tfrac{k_{22}}{2}\right)- -5f_0''\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(f_{3n}+d\eta\tfrac{k_{21}}{2}\right)+\tfrac{\partial(f_1'f_0)}{\partial(\Lambda,\eta_n+\frac{d\eta}{2})}+\tfrac{\partial(f_0'f_1)}{\partial(\Lambda,\eta_n+\frac{d\eta}{2})}+ +(2+\Lambda)\left(f_1'\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\right)^2-3f_1\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)f_1''\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right),$$

$$k_{41}=U_n+d\eta k_{32},\quad k_{42}=G_n+d\eta k_{33},$$

$$k_{43} = -f_0(\Lambda,\eta_n+d\eta)(G_n+d\eta k_{23})+2(2+\Lambda)f_0'(\Lambda,\eta_n+d\eta)(U_n+d\eta k_{22})- -5f_0''(\Lambda,\eta_n+d\eta)(f_{3n}+d\eta k_{21})+\tfrac{\partial(f_1'f_0)}{\partial(\Lambda,\eta_n+d\eta)}+\tfrac{\partial(f_0'f_1)}{\partial(\Lambda,\eta_n+d\eta)}+ +(2+\Lambda)\left(f_1'(\Lambda,\eta_n+d\eta)\right)^2-3f_1(\Lambda,\eta_n+d\eta)f_1''(\Lambda,\eta_n+d\eta),$$

where the unknown coefficients are defined as follows:

$$f_{3(n+1)}=f_{3n}+\frac{d\eta}{6}(k_{11}+2k_{21}+2k_{31}+k_{41}),$$

$$U_{n+1}=U_n+\frac{d\eta}{6}(k_{12}+2k_{22}+2k_{32}+k_{42}),$$

$$G_{n+1}=G_n+\frac{d\eta}{6}(k_{13}+2k_{23}+2k_{33}+k_{43}).$$

The preceding concludes the expressions for the velocity functions. A similar approach is applied for the temperature functions in what follows. For θ_0,

$\theta_0'=U,\quad U'=-Prf_0(\Lambda,\eta)U,$

the discretization coefficients are given as

$$k_{11}=U_n,\quad k_{12}=-Prf_0(\Lambda,\eta_n)U_n,\quad k_{21}=U_n+d\eta\frac{k_{12}}{2},$$

$$k_{22}=-Prf_0\left(\Lambda,\eta_n+\frac{d\eta}{2}\right)\left(U_n+d\eta\frac{k_{12}}{2}\right),$$

$$k_{31}=U_n+d\eta\frac{k_{22}}{2},$$

$$k_{32}=-Prf_0\left(\Lambda,\eta_n+\frac{d\eta}{2}\right)\left(U_n+d\eta\frac{k_{22}}{2}\right),$$

$$k_{41}=U_n+d\eta k_{32},\quad k_{42}=-Prf_0(\Lambda,\eta_n+d\eta)(U_n+d\eta k_{32}),$$

where the unknown coefficients are defined as follows:

$$\theta_{0(n+1)}=\theta_{0n}+\frac{d\eta}{6}(k_{11}+2k_{21}+2k_{31}+k_{41}),$$

$$U_{n+1}=U_n+\frac{d\eta}{6}(k_{12}+2k_{22}+2k_{32}+k_{42}).$$

For θ_1,

$\theta_1'=U,$

$$U'=Pr\left(-f_0(\Lambda,\eta)U+2f_0'(\Lambda,\eta)\theta_1+\frac{\partial(\theta_0,f_0)}{\partial(\Lambda,\eta)}-3f_1(\Lambda,\eta)\theta_0'(\Lambda,\eta)\right),$$

the discretization coefficients are given as

$k_{11}=U_n,$

$$k_{12}=Pr\left(-f_0(\Lambda,\eta_n)U_n+2f_0'(\Lambda,\eta_n)\theta_{1n}+\frac{\partial(\theta_0,f_0)}{\partial(\Lambda,\eta_n)}-3f_1(\Lambda,\eta_n)\theta_0'(\Lambda,\eta_n)\right),$$

$$k_{21}=U_n+d\eta\frac{k_{12}}{2},$$

$$k_{22} = Pr\Big(-f_0\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(U_n+d\eta\tfrac{k_{12}}{2}\right)+2f_0'\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\left(\theta_{1n}+d\eta\tfrac{k_{11}}{2}\right)+ +\tfrac{\partial(\theta_0,f_0)}{\partial(\Lambda,\eta_n+\frac{d\eta}{2})}-3f_1\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\theta_0'\left(\Lambda,\eta_n+\tfrac{d\eta}{2}\right)\Big),$$

$$k_{31}=U_n+d\eta\frac{k_{22}}{2},$$

$$k_{32} = Pr\Big(-f_0\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\Big(U_n+d\eta\tfrac{k_{22}}{2}\Big)+2f_0'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\Big(\theta_{1n}+d\eta\tfrac{k_{21}}{2}\Big)+ +\tfrac{\partial(\theta_0 f_0)}{\partial\left(\Lambda,\eta_n+\frac{d\eta}{2}\right)}-3f_1\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\theta_0'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\Big),$$

$$k_{41} = U_n + d\eta k_{32},$$

$$k_{42} = Pr(-f_0(\Lambda,\eta_n+d\eta)(U_n+d\eta k_{32})+2f_0'(\Lambda,\eta_n+d\eta)(\theta_{1n}+d\eta k_{31})+ +\tfrac{\partial(\theta_0 f_0)}{\partial(\Lambda,\eta_n+d\eta)}-3f_1(\Lambda,\eta_n+d\eta)\theta_0'(\Lambda,\eta_n+d\eta)\Big),$$

where the unknown coefficients are defined as follows:

$$\theta_{1(n+1)} = \theta_{1n} + \frac{d\eta}{6}(k_{11}+2k_{21}+2k_{31}+k_{41}),$$

$$U_{n+1} = U_n + \frac{d\eta}{6}(k_{12}+2k_{22}+2k_{32}+k_{42}).$$

For θ_2,

$\theta_2' = U,$

$$U' = Pr(-f_0(\Lambda,\eta)U+4f_0'(\Lambda,\eta)\theta_2+f_0'(\Lambda,\eta)\theta_1(\Lambda,\eta)-f_1(\Lambda,\eta)\theta_0'(\Lambda,\eta)- -5f_2(\Lambda,\eta)\theta_0'(\Lambda,\eta)),$$

the discretization coefficients are given as

$k_{11} = U_n,$

$$k_{12} = Pr(-f_0(\Lambda,\eta_n)U_n+4f_0'(\Lambda,\eta_n)\theta_{2n}+f_0'(\Lambda,\eta_n)\theta_1(\Lambda,\eta_n)- -f_1(\Lambda,\eta_n)\theta_0'(\Lambda,\eta_n)-5f_2(\Lambda,\eta_n)\theta_0'(\Lambda,\eta_n)),$$

$$k_{21} = U_n + d\eta\frac{k_{12}}{2},$$

$$k_{22} = Pr\Big(-f_0\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\Big(U_n+d\eta\tfrac{k_{12}}{2}\Big)+4f_0'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\Big(\theta_{2n}+d\eta\tfrac{k_{11}}{2}\Big)+ +f_0'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\theta_1\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)-f_1\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\theta_0'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)- -5f_2\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\theta_0'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\Big),$$

$$k_{31} = U_n + d\eta\frac{k_{22}}{2},$$

$$k_{32} = Pr\Big(-f_0\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\Big(U_n+d\eta\tfrac{k_{22}}{2}\Big)+4f_0'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\Big(\theta_{2n}+d\eta\tfrac{k_{21}}{2}\Big)+ +f_0'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\theta_1\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)-f_1\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\theta_0'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)- -5f_2\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\theta_0'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\Big),$$

$$k_{41} = U_n + d\eta k_{32},$$

$$k_{42} = Pr(-f_0(\Lambda,\eta_n+d\eta)(U_n+d\eta k_{32})+4f_0'(\Lambda,\eta_n+d\eta)(\theta_{2n}+d\eta k_{31})+ +f_0'(\Lambda,\eta_n+d\eta)\theta_1(\Lambda,\eta_n+d\eta)-f_1(\Lambda,\eta_n+d\eta)\theta_0'(\Lambda,\eta_n+d\eta)- -5f_2(\Lambda,\eta_n+d\eta)\theta_0'(\Lambda,\eta_n+d\eta)),$$

where the unknown coefficients are defined as follows:

$$\theta_{2(n+1)} = \theta_{2n} + \frac{d\eta}{6}(k_{11}+2k_{21}+2k_{31}+k_{41}),$$

$$U_{n+1} = U_n + \frac{d\eta}{6}(k_{12}+2k_{22}+2k_{32}+k_{42}).$$

For θ_3,

$\theta_3' = U,$

$$U' = Pr\Big(-f_0(\Lambda,\eta)U+4f_0'(\Lambda,\eta)\theta_3+\tfrac{\partial(\theta_1 f_0)}{\partial(\Lambda,\eta)}+\tfrac{\partial(\theta_0 f_1)}{\partial(\Lambda,\eta)}+2f_1'(\Lambda,\eta)\theta_1(\Lambda,\eta)- -3f_1(\Lambda,\eta)\theta_1'(\Lambda,\eta)-5f_3(\Lambda,\eta)\theta_0'(\Lambda,\eta)\Big),$$

the discretization coefficients are given as

$k_{11} = U_n,$

$$k_{12} = Pr\Big(-f_0(\Lambda,\eta_n)U_n+4f_0'(\Lambda,\eta_n)\theta_{3n}+\tfrac{\partial(\theta_1 f_0)}{\partial(\Lambda,\eta_n)}+\tfrac{\partial(\theta_0 f_1)}{\partial(\Lambda,\eta_n)}+ +2f_1'(\Lambda,\eta_n)\theta_1(\Lambda,\eta_n)-3f_1(\Lambda,\eta_n)\theta_1'(\Lambda,\eta_n)- -5f_3(\Lambda,\eta_n)\theta_0'(\Lambda,\eta_n)\Big),$$

$$k_{21} = U_n + d\eta\frac{k_{12}}{2},$$

$$k_{22} = Pr\Big(-f_0\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\Big(U_n+d\eta\tfrac{k_{12}}{2}\Big)+4f_0'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\Big(\theta_{3n}+d\eta\tfrac{k_{11}}{2}\Big)+ +\tfrac{\partial(\theta_1 f_0)}{\partial\left(\Lambda,\eta_n+\frac{d\eta}{2}\right)}+\tfrac{\partial(\theta_0 f_1)}{\partial\left(\Lambda,\eta_n+\frac{d\eta}{2}\right)}+2f_1'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\theta_1\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)- -3f_1\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\theta_1'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)-5f_3\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\theta_0'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\Big),$$

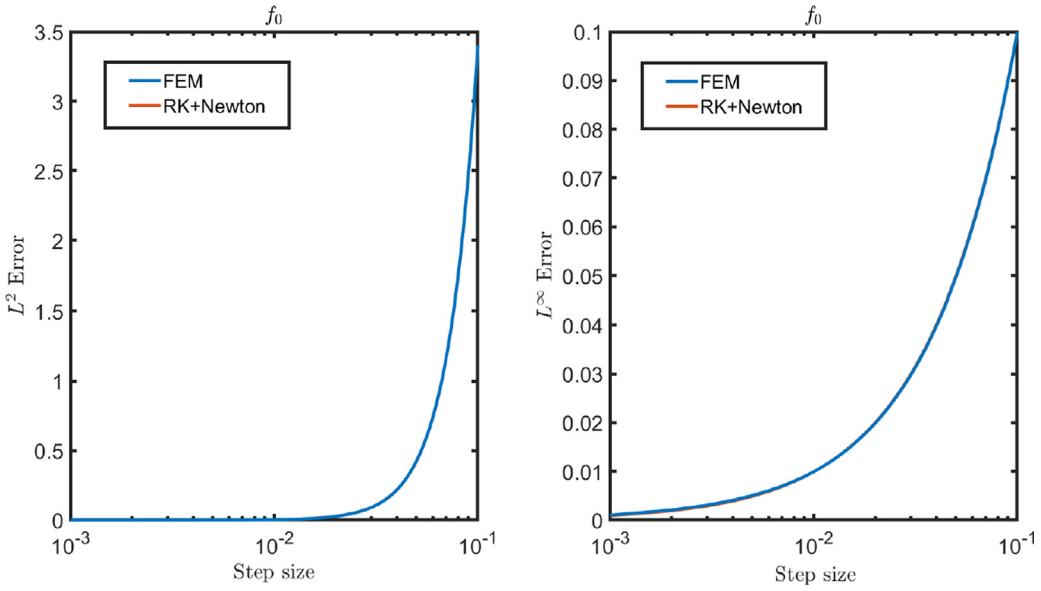

Fig. 1. Comparison of L^2 and L^∞ errors vs. step size for f_0.

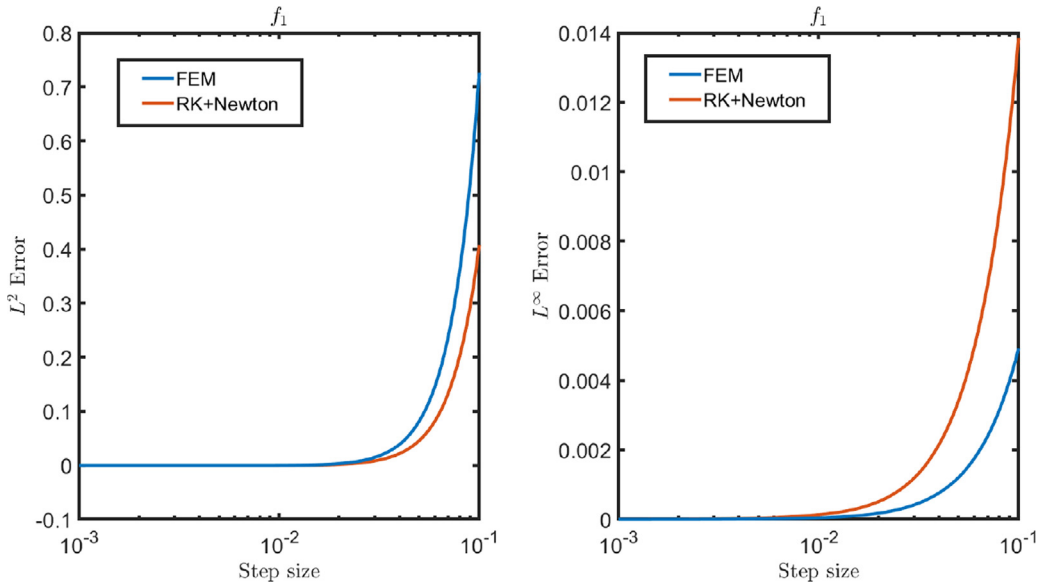

Fig. 2. Comparison of L^2 and L^∞ errors vs. step size for f_1.

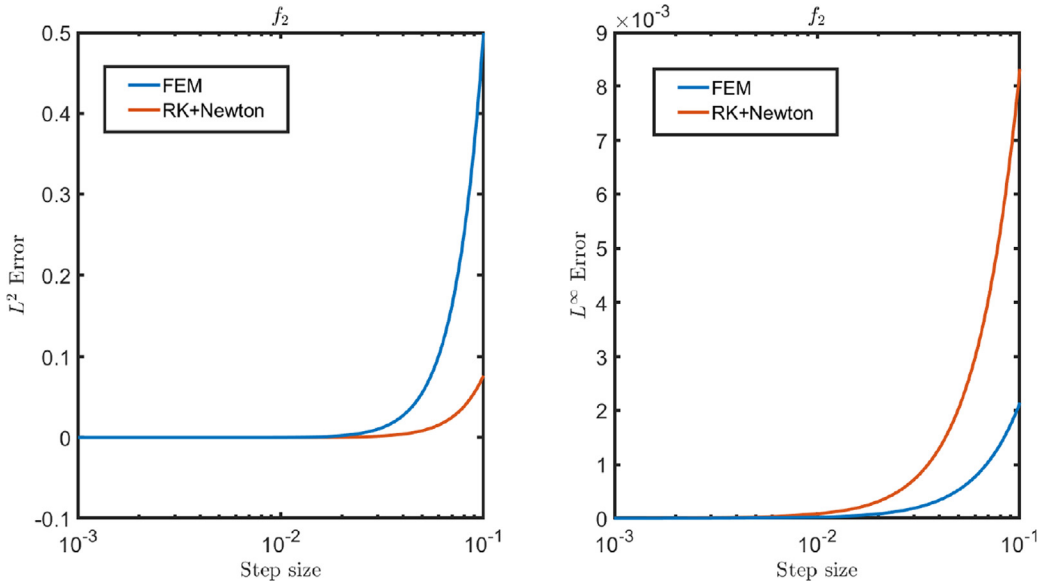

Fig. 3. Comparison of L^2 and L^∞ errors vs. step size for f_2.

$$k_{31} = U_n + d\eta\frac{k_{22}}{2},$$

$$\begin{aligned} k_{32} &= Pr\Big(-f_0\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\Big(U_n+d\eta\tfrac{k_{22}}{2}\Big)+4f_0'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\Big(\theta_{3n}+d\eta\tfrac{k_{21}}{2}\Big)+\\ &+\tfrac{\partial(\theta_1 f_0)}{\partial\left(\Lambda,\eta_n+\frac{d\eta}{2}\right)}+\tfrac{\partial(\theta_0 f_1)}{\partial\left(\Lambda,\eta_n+\frac{d\eta}{2}\right)}+2f_1'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\theta_1\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)-\\ &-3f_1\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\theta_1'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)-5f_3\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\theta_0'\Big(\Lambda,\eta_n+\tfrac{d\eta}{2}\Big)\Big), \end{aligned}$$

$$k_{41} = U_n + d\eta k_{32},$$

$$\begin{aligned} k_{42} &= Pr(-f_0(\Lambda,\eta_n+d\eta)(U_n+d\eta k_{32})+4f_0'(\Lambda,\eta_n+d\eta)(\theta_{3n}+d\eta k_{31})+\\ &+\tfrac{\partial(\theta_1 f_0)}{\partial(\Lambda,\eta_n+d\eta)}+\tfrac{\partial(\theta_0 f_1)}{\partial(\Lambda,\eta_n+d\eta)}+2f_1'(\Lambda,\eta_n+d\eta)\theta_1(\Lambda,\eta_n+d\eta)-\\ &-3f_1(\Lambda,\eta_n+d\eta)\theta_1'(\Lambda,\eta_n+d\eta)-5f_3(\Lambda,\eta_n+d\eta)\theta_0'(\Lambda,\eta_n+d\eta)), \end{aligned}$$

where the unknown coefficients are defined as follows:

$$\theta_{3(n+1)} = \theta_{3n} + \frac{d\eta}{6}(k_{11}+2k_{21}+2k_{31}+k_{41}),$$

$$U_{n+1} = U_n + \frac{d\eta}{6}(k_{12}+2k_{22}+2k_{32}+k_{42}).$$

The developed steps in this section are translated into the MATLAB computer program. MATLAB facilitates implementation through the several toolboxes available for numerical analysis.

3. Results and discussion

This section discusses the findings of the numerical analysis, as well as their comparisons. It is noteworthy that according to Chao and Fagbenle [2], all derivatives with respect to the wedge parameter, required for the evaluation of the Jacobians, are approximated according to the composite finite difference rule

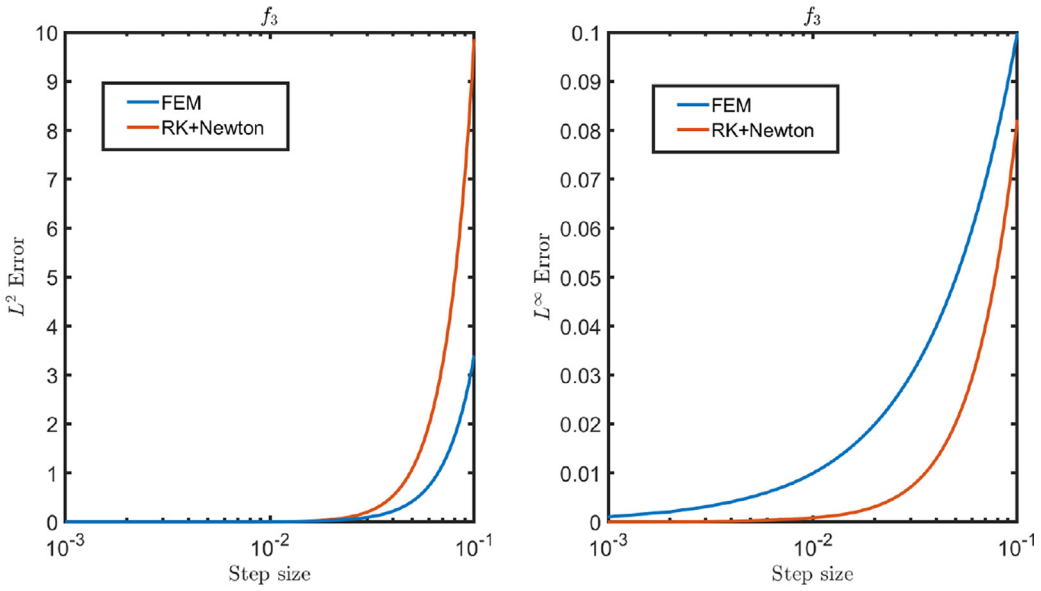

Fig. 4. Comparison of L^2 and L^∞ errors vs. step size for f_3.

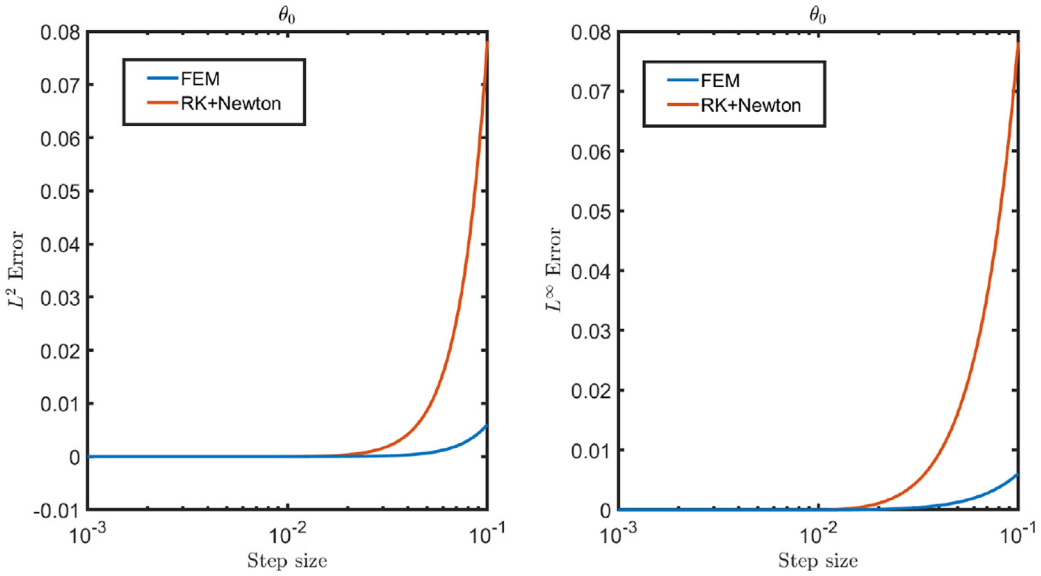

Fig. 5. Comparison of L^2 and L^∞ errors vs. step size for θ_0.

$$\frac{\partial f}{\partial \Lambda} \approx \frac{-f(\Lambda+2\Delta\Lambda,\eta)+8f(\Lambda+\Delta\Lambda,\eta)-8f(\Lambda-\Delta\Lambda,\eta)+f(\Lambda-2\Delta\Lambda,\eta)}{12\Delta\Lambda}$$

and the choice was $\boldsymbol{\Delta\Lambda = 10^{-3}}$.

3.1. Comparison of RK + Newton and FEM

In this sub-section, the performances of both techniques are assessed. Figs. 1–8 show L^2 and L^∞ error plots for solutions obtained here and that obtained by FEM in [13] as a function of the step size. For comparisons, we have chosen Δ = 0.0001: 0.1. The L^2 and L^∞ errors are given as follows:

$$||f||_{L^2[0,\eta_\infty]} = \int_0^{\eta_\infty} f^2(\eta)d\eta, \quad ||f||_{L^\infty[0,\eta_\infty]} = \sup_{\eta\in[0,\eta_\infty]}|f(\eta)|.$$

This form of the L^2 and L^∞ errors is mathematically quite elegant. The L^2 error describes the integral of the square of the difference between the two functions (n and n - 1), whereas the L^∞ error describes the maximal absolute value of the difference of these functions.

It is observed from Fig. 1 that comparing L^2 and L^∞ of approximation of f_0 for FEM and the present method (RK + Newton) here shows both coincide, which implies that f_0 is approximated with exactly the same accuracy (i.e. no discernible differences) by both methods. This is important because high accuracy is often required for the basic functions f_0 and θ_0 because the higher order functions $f_1, \theta_1, f_2, \theta_2, \cdots$, are sensitive to their variations. However, it is evident from Figs. 2–8 that for larger values of the step size FEM provides better accuracy than the RK + Newton method described here. In another regard, it is observed that FEM requires considerably more computational power as compared to RK + Newton, which is discussed further in sub-Section 3.3. The error plots in Figs. 1–8 essentially only have mathematical meanings that reflect

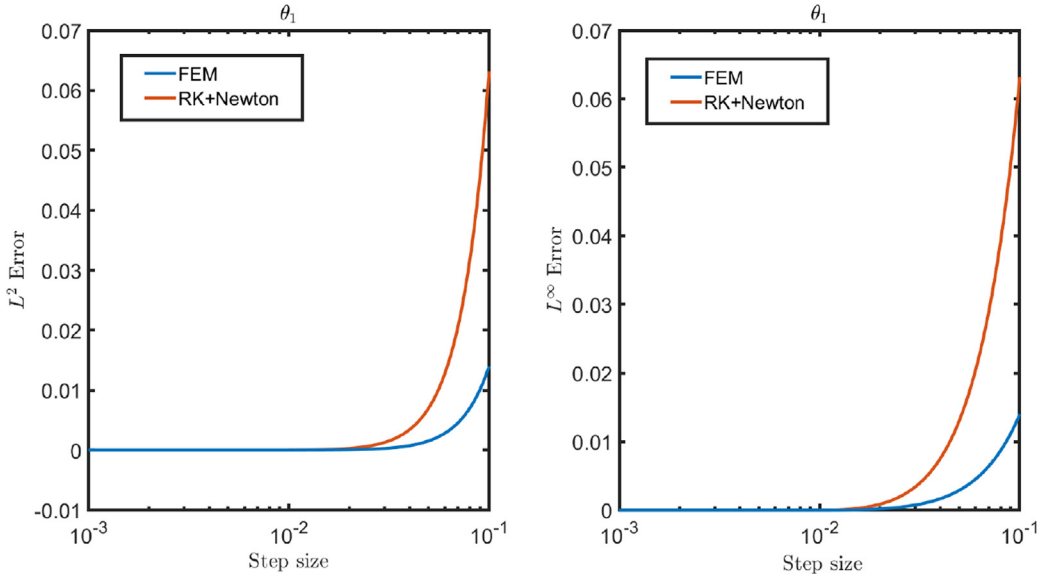

Fig. 6. Comparison of L^2 and L^∞ errors vs. step size for θ_1.

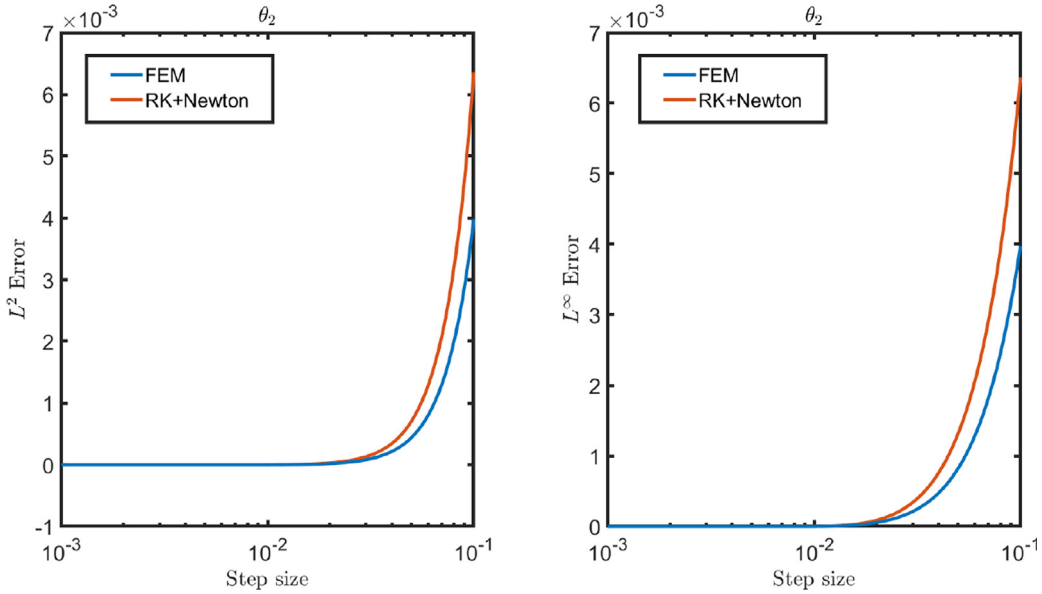

Fig. 7. Comparison of L^2 and L^∞ errors vs. step size for θ_2.

mathematical facts for the respective parameters. These figures simply show the differences between FEM and RK + Newton outputs according to the time step, and serve as evidence for conclusions drawn.

3.2. Fluid boundary layer transfer

The premise in this section is simply to demonstrate good agreement with other findings in the literature for key fluid boundary layer parameters. This sub-section pertains to the fluid boundary layer results of the numerical analyses by comparing the RK + Newton findings here with FEM results in Amoo [26] and the highly accurate benchmark results in work by Chao and Fagbenle [2,3]. The comparisons are made numerically to assess the results obtained qualitatively. It is observed from Tables 2–4 that the findings are in good correlation for the heat transfer, nonsimilar velocity functions, and nonsimilar temperature functions. The tabulated comparisons are reflective of a small step size for accuracy. Larger values of the step size may be necessary in cases where fast computations may be desired, without necessarily caring too much for accuracy, and in this regard the FEM approach performs better. It should also be remarked that the FEM is naturally oriented towards minimizing errors in a global sense (the essence of any numerical scheme is to minimize errors). The FEM is much more powerful than the RK + Newton method for MCF-type problems or problems in which some of the boundary conditions are given at infinity, both theoretically and practically. Overall, the results are just about as accurate as those of other researchers found in the literature. It is important to mention that in a study assessing the performance of methodologies to predict heat trans-

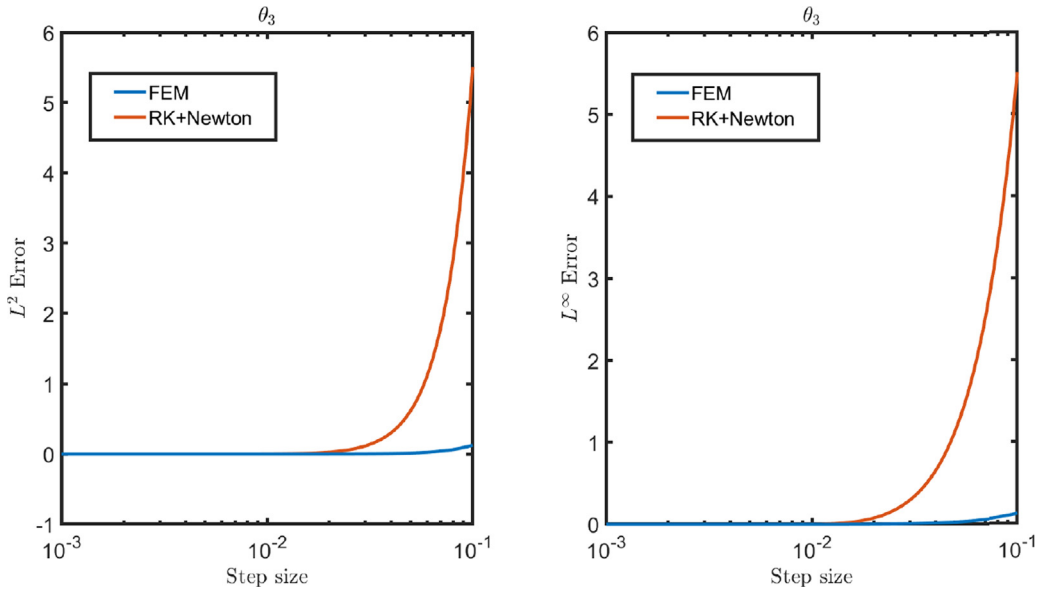

Fig. 8. Comparison of L^2 and L^∞ errors vs. step size for θ_3.

Table 2
Comparison of results between [2], FEM analysis in [26], and present analysis using RK + Newton for local heat transfer parameter, $NuRe^{-\frac{1}{2}}$, for flow over an inclined surface; Pr= 5($Nu = hL/k$, $Re = U_\infty L/\nu$).

		No. of terms in series [2]			
x/L	Λ	1	2	3	4
0.0675	−0.15	0.9408	0.9536	0.9547	0.9921
0.0465	−0.10	1.2237	1.2249	1.2257	1.2319
0.0241	−0.05	1.7989	1.7975	1.7978	1.7989
0.0541	0.10	1.3642	1.3681	1.3687	1.3695
0.4142	0.50	0.6654	0.6704	0.6730	0.6738
0.8257	0.70	0.5765	0.5798	0.5820	0.5824
		No. of terms in series [26]			
x/L	Λ	1	2	3	4
0.0675	−0.15	0.94081	0.95355	0.95473	0.99202
0.0465	−0.10	1.2237	1.2249	1.2257	1.2319
0.0241	−0.05	1.7989	1.7975	1.7978	1.7989
0.0541	0.10	1.3642	1.3681	1.3687	1.3702
0.4142	0.50	0.66538	0.67039	0.67297	0.67667
0.8257	0.70	0.57652	0.57982	0.58201	0.58372
		No. of terms in series (RK + Newton, this work)			
x/L	Λ	1	2	3	4
0.0675	−0.15	0.94081	0.95355	0.95472	0.99258
0.0465	−0.10	1.2237	1.2249	1.2257	1.232
0.0241	−0.05	1.7989	1.7975	1.7978	1.7989
0.0541	0.10	1.3642	1.3681	1.3687	1.3702
0.4142	0.50	0.66538	0.67039	0.67296	0.67676
0.8257	0.70	0.57653	0.57982	0.58201	0.58377

fer, results obtained from the MCF methodology itself are 1– 3% in agreement with experimental and other theoretical results, as established in [3].

3.3. Comparison of computational efficiency

In this sub-section, comparisons are made for both the computation and CPU times for the evaluation of f_0, f_1, f_2, f_3, and $\theta_0, \theta_1, \theta_2, \theta_3$ according to both the FEM and the RK + Newton methods for element sizes of 10^{-3} and 10^{-4}, respectively. The CPU time may be defined as the total time used by MATLAB from the time the simulation was initiated. It may also be regarded as the execution time or operation count. On the other hand, the computational time measures the elapsed time of different steps or parts of the code. It is evident from Tables 5 and 6 that the computation time of RK + Newton is significantly less than that of the FEM for both values of the element size. In another regard, it is observed that the CPU time for RK + Newton is less than that of the FEM for f_0 and θ_0 only.

Overall, robust numerical experiments and comparisons of two numerical methods (the FEM and the RK + Newton methods) have been carried out on a model nonsimilar fluid boundary layer problem, for which no relevant mathematical results existed. Thus the authors believe this work provides a benchmark for which other numerical methods for solving similar equations can be tested.

Table 3
Comparison of results between [2], FEM analysis in [26], and present analysis using RK + Newton for wall derivatives of velocity functions.

Universal functions: Wall derivatives of velocity functions [2]				
Λ	$f_0''(\Lambda,0)$	$f_1''(\Lambda,0)$	$f_2''(\Lambda,0)$	$f_3''(\Lambda,0)$
−0.15	0.2163614060	−3.47914250	0.053071271	−0.51628
−0.1	0.3192697599	−0.222593940	0.032254361	−0.18908
−0.05	0.4003225954	−0.166581500	0.023117684	−0.10318
0	0.4695999884	−0.133283260	0.017790554	−0.065937
0.05	0.5311296305	−0.110813940	0.014266708	−0.045895
0.1	0.5870352192	−0.094506819	0.011760066	−0.033722
0.2	0.6867081810	−0.072319116	0.008446235	−0.020184
Universal functions: Wall derivatives of velocity functions [26]				
Λ	$f_0''(\Lambda,0)$	$f_1''(\Lambda,0)$	$f_2''(\Lambda,0)$	$f_3''(\Lambda,0)$
−0.15	0.21636	−0.34791	0.053071	−0.51633
−0.1	0.31927	−0.22259	0.032254	−0.18908
−0.05	0.40032	−0.16658	0.023117	−0.094945
0	0.4696	−0.13328	0.017791	−0.065936
0.05	0.53113	−0.11081	0.014267	−0.045895
0.1	0.58704	−0.094507	0.01176	−0.033722
0.2	0.68671	−0.072319	0.0084462	−0.020184
Universal functions: Wall derivatives of velocity functions (RK + Newton, this work)				
Λ	$f_0''(\Lambda,0)$	$f_1''(\Lambda,0)$	$f_2''(\Lambda,0)$	$f_3''(\Lambda,0)$
−0.15	0.21636	−0.34791	0.053071	−0.51633
−0.1	0.31927	−0.22259	0.032254	−0.18908
−0.05	0.40032	−0.16658	0.023118	−0.10318
0	0.4696	−0.13328	0.017791	−0.065937
0.05	0.53113	−0.11081	0.014267	−0.045895
0.1	0.58704	−0.094507	0.01176	−0.033722
0.2	0.68671	−0.072319	0.0084462	−0.020184

Table 4
Comparison of results between [2], FEM analysis in [26], and present analysis using RK + Newton for wall derivatives of temperature functions.

Universal functions: Wall derivatives of temperature functions [2]				
Λ	$\theta_0'(\Lambda,0)$	$\theta_1'(\Lambda,0)$	$\theta_2'(\Lambda,0)$	$\theta_3'(\Lambda,0)$
−0.15	0.68330186	−0.05582850	−0.00279369	0.28400
−0.1	0.74362901	−0.01590380	−0.00546172	0.09337
−0.05	0.78434114	−0.00138098	−0.00605684	0.05020
0	0.81556125	0.00573642	−0.006131	0.03284
0.05	0.84103038	0.00969494	−0.00602159	0.02383
0.1	0.86259826	0.01204010	−0.00584057	0.01842
0.2	0.89789713	0.01433970	−0.00541959	0.01232
Universal functions: Wall derivatives of temperature functions [26]				
Λ	$\theta_0'(\Lambda,0)$	$\theta_1'(\Lambda,0)$	$\theta_2'(\Lambda,0)$	$\theta_3'(\Lambda,0)$
−0.15	0.6833	−0.055828	−0.0027938	0.28394
−0.1	0.74363	−0.015904	−0.0054617	0.09337
−0.05	0.78434	−0.001378	−0.0060576	0.05020
0	0.81556	0.0057364	−0.006131	0.03284
0.05	0.84103	0.0096949	−0.0060216	0.02383
0.1	0.8626	0.01204	−0.0058406	0.01842
0.2	0.8979	0.01434	−0.0054196	0.01232
Universal functions: Wall derivatives of temperature functions (RK + Newton, this work)				
Λ	$\theta_0'(\Lambda,0)$	$\theta_1'(\Lambda,0)$	$\theta_2'(\Lambda,0)$	$\theta_3'(\Lambda,0)$
−0.15	0.6833	−0.055829	−0.0027937	0.28398
−0.1	0.74363	0.015904	0.0054618	0.09337
0.05	0.78434	0.001381	−0.0060568	0.050202
0	0.8155	0.0057364	−0.006131	0.032843
0.05	0.84103	0.0096948	0.0060215	0.023843
0.1	0.8626	0.01204	−0.0058406	0.018417
0.2	0.8979	0.01434	−0.0054196	0.012316

4. Conclusions

In this research we have assessed the performance of two numerical methods – the finite element method (FEM) and the Runge–Kutta with Newton–Raphson shooting method (RK + Newton) – for solving a set of nonsimilar boundary layer derived ordinary differential equations. While it is shown that the RK + Newton method has commonly been employed by researchers for the type of equations considered here, it has been made manifestly evident that for larger values of the step size FEM provides better accuracy than the RK + Newton method, albeit at the expense of more computational power.

The broader practical implication of the findings here is that they offer guidance on what numerical techniques may be appropriate for particular problems. The reader and the student may reference this work as a tool for numerically solving differential equations of various characters. The industrial implications can also not be overlooked, as practical industries seek fast approaches

Table 5
Comparison of results of both computation and CPU times for the FEM and RK + Newton methods: $d\eta = 10^{-3}, \Lambda = -0.15, Pr = 0.7$.

Functions	Element size	Computation time FEM (s)	Computation time RK + Newton (s)	CPU time FEM (s)	CPU time RK + Newton (s)
$f_0(\Lambda,\eta)$	10^{-3}	1.213757e+02	6.170360e−02	8.295313e+01	9.375000e−02
$f_1(\Lambda,\eta)$	10^{-3}	5.200374e+02	1.530066e−01	3.428769e+04	8.400128e+04
$f_2(\Lambda,\eta)$	10^{-3}	7.212769e+02	1.843316e−01	1.866031e+03	6.406250e−01
$f_3(\Lambda,\eta)$	10^{-3}	5.827214e+01	4.064530e−02	3.446670e+04	8.400138e+04
$\theta_0(\Lambda,\eta)$	10^{-3}	1.769856e+02	1.391869e−01	7.100156e+02	2.812500e−01
$\theta_1(\Lambda,\eta)$	10^{-3}	1.998279e+02	9.785790e−02	5.804692e+04	8.400280e+04
$\theta_2(\Lambda,\eta)$	10^{-3}	1.992414e+02	6.832040e−02	1.507781e+03	5.000000e−01
$\theta_3(\Lambda,\eta)$	10^{-3}	3.985892e+01	3.623540e−02	5.820639e+04	8.400291e+04

Table 6
Comparison of results of both computation and CPU times for the FEM and RK + Newton methods: $d\eta = 10^{-4}, \Lambda = -0.15, Pr = 0.7$.

Functions	Element size	Computation time FEM (s)	Computation time RK + Newton (s)	CPU time FEM (s)	CPU time RK + Newton (s)
$f_0(\Lambda,\eta)$	10^{-4}	2.483484e+02	2.632862e−01	1.982344e+02	4.062500e−01
$f_1(\Lambda,\eta)$	10^{-4}	1.649585e+03	9.325260e−01	4.168198e+04	8.403544e+04
$f_2(\Lambda,\eta)$	10^{-4}	2.268270e+03	1.042285e+00	9.988266e+03	1.750000e+00
$f_3(\Lambda,\eta)$	10^{-4}	1.363120e+02	1.325625e−01	4.221048e+04	8.403572e+04
$\theta_0(\Lambda,\eta)$	10^{-4}	4.953056e+02	8.997848e−01	1.882797e+03	1.015625e+00
$\theta_1(\Lambda,\eta)$	10^{-4}	6.810405e+02	5.226423e−01	7.749628e+04	8.403914e+04
$\theta_2(\Lambda,\eta)$	10^{-4}	9.931706e+02	3.156395e−01	5.604484e+03	1.578125e+00
$\theta_3(\Lambda,\eta)$	10^{-4}	2.081671e+02	1.0018998e−01	7.827941e+04	8.403934e+04

to solve problems and deliver improved products to market. Advancing the numerical literature using boundary layer problems as case studies continues to be a necessary undertaking toward the development and improvement of numerical methods.

Though the numerical results of the boundary layer parameters (velocity functions, temperature functions, and heat transfer) all show good agreement, using both numerical methods, the FEM method is demonstrated to be much more accurate. Future opportunities may exist to develop or adapt other numerical techniques (and compare their performances) for solving the MCF-type equations (or equations in which some of the boundary conditions are given at infinity). Specifically, future efforts in numerical mathematics may investigate the use of the power series expansion method (to obtain analytical formulas) together with the Runge–Kutta method.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Merk HJ. Rapid calculations for boundary-layer transfer using wedge solutions and asymptotic expansions. J Fluid Mech 1959;5:460–80.
[2] Chao BT, Fagbenle RO. On Merk's methods of calculating boundary layer transfer Int. J Heat Mass Transfer 1974;17:223–40.
[3] Spalding DB, Pun WM. A review of methods for predicting heat transfer coefficients for laminar uniform-property boundary layer flows Int. J Heat Mass Transfer 1962;5:239–49.
[4] Van Dyke M. Perturbation Methods in Fluid Mechanics. Stanford, CA: Parabolic Press; 1975.
[5] Evans HL. Laminar Boundary-Layer Theory. USA: Addison-Wesley; 1968.
[6] Sano T. Extensions of the Merk's method to the calculation of laminar boundary layer heat transfer from a non-isothermal surface. Warme-und Stoffubertragung 1975;8:87–100.
[7] Sano T. Higher approximations to thermal boundary-layer in an inviscid flow Int. J Heat Mass Transfer 1975;18:1257–66.
[8] Fagbenle RO, Amoo OM, Aliu S, Falana A, editors. Applications of Heat, Mass and Fluid Boundary Layers. Cambridge, MA: Elsevier; 2020.
[9] Hayat T, Khan MI, Farooq M, Alsaedi A, Waqas M, Yasmeen T. Impact of Cattaneo-Christov heat flux model in flow of variable thermal conductivity fluid over a variable thicked surface Int. J Heat Mass Transfer 2016;99:702–10. doi: https://doi.org/10.1016/j.ijheatmasstransfer.2016.04.016.
[10] Khan MI, Waqas M, Hayat T, Alsaedi A. A comparative study of Casson fluid with homogeneous-heterogeneous reactions. J Coll Interface Sci 2017;498:85–90. doi: https://doi.org/10.1016/j.jcis.2017.03.024.
[11] Khan MI, Alzahrani F. Nonlinear dissipative slip flow of Jeffrey nanomaterial towards a curved surface with entropy generation and activation energy Math. Comp Sim 2021;185:47–61. doi: https://doi.org/10.1016/j.matcom.2020.12.004.
[12] Ibrahim M, Khan MI. Mathematical modeling and analysis of SWCNT-water and MWCNT-water flow over a stretchable sheet Comp. Meth Prog Biomed 2020;187:105222. doi: https://doi.org/10.1016/j.cmpb.2019.105222.
[13] Nayak MK, Abdul Hakeem AK, Ganga M, Khan MI, Waqas M, Makinde OD. Entropy optimized MHD 3D nanomaterial of non-Newtonian fluid: a combined approach to good absorber of solar energy and intensification of heat transport Comp. Meth Prog Biomed 2020;186:105131. doi: https://doi.org/10.1016/j.cmpb.2019.105131.
[14] Hosseinzadeh K, Mardani MR, Salehi S, et al. Entropy generation of three-dimensional Bödewadt flow of water and hexanol base fluid suspended by Fe_3O_4 and MoS_2 hybrid nanoparticles. Pramana - J Phys 2021;95:57. doi: https://doi.org/10.1007/s12043-020-02075-9.
[15] Hosseinzadeh K, Roghani S, Mogharrebi AR, Asadi A, Waqas M, Ganji DD. Investigation of cross-fluid flow containing motile gyrotactic microorganisms and nanoparticles over a three-dimensional cylinder. Alexandria Eng J 2020;59(5):3297–307. doi: https://doi.org/10.1016/j.aej.2020.04.037.
[16] Hosseinzadeh K, Roghani S, Mogharrebi AR, et al. Optimization of hybrid nanoparticles with mixture fluid flow in an octagonal porous medium by effect of radiation and magnetic field. J Therm Anal Calorim 2021;143:1413–24. doi: https://doi.org/10.1007/s10973-020-10376-9.
[17] Hosseinzadeh K, Montazer E, Shafii MB, Ganji DD. Heat transfer hybrid nanofluid (1-Butanol/MoS_2–Fe_3O_4) through a wavy porous cavity and its optimization. Int J Num Meth Heat Fluid Flow 2021;31(5):1547–67. doi: https://doi.org/10.1108/HFF-07-2020-0442.
[18] Ghadikolaei SS, Gholinia M. 3D mixed convection MHD flow of GO-MoS_2 hybrid nanoparticles in H_2O–$(CH_2OH)_2$ hybrid base fluid under the effect of H_2 bond. Int Comm Heat Mass Trans 2020;110:104371. doi: https://doi.org/10.1016/j.icheatmasstransfer.2019.104371.
[19] Ghadikolaei SS, Hosseinzadeh K, Ganji DD. Investigation on magneto Eyring-Powell nanofluid flow over inclined stretching cylinder with nolinear thermal radiation and Joule heating effect. World J Eng 2019;16(1):51–63. doi: https://doi.org/10.1108/WJE-06-2018-0204.
[20] Ghadikolaei SS, Yassari M, Sadeghi H, Hosseinzadeh K, Ganji DD. Investigation on thermophysical properties of TiO_2–Cu/H_2O hybrid nanofluid transport dependent on shape factor in MHD stagnation point flow Powder Tech. 2017;322: 428–38. doi: https://doi.org/10.1016/j.powtec.2017.09.006.
[21] Ghadikolaei SS, Hosseinzadeh K, Hatami M, Ganji DD. Investigation for squeezing flow of ethylene glycol ($C_2H_6O_2$) carbon nanotubes (CNTs) in rotating stretching channel with nonlinear thermal radiation. J Molec Liq 2018;263:10–21. doi: https://doi.org/10.1016/j.molliq.2018.04.141.
[22] Nachtsheim P R and Swigert P. Satisfaction of asymptotic boundary conditions in numerical solution of systems of nonlinear equations of boundary layer

type. National Aeronautics and Space Administration, Washington DC Rep. NACA TN D-3004; 1965.
[23] Rogers DF. Laminar Flow Analysis. New York, NY: Cambridge University Press; 1992.
[24] Carnahan B, Luther HA, Wilkes JO. Applied Numerical Methods. New York, NY: John Wiley & Sons; 1969.
[25] Kutz JN. Data-Driven Modeling & Scientific Computation. New York, NY: Oxford University Press; 2013.
[26] Amoo OM, Oyewola MO, Fagbenle RO. Application of the finite element method to the solution of nonsimilar boundary layer-derived infinite series equations. Int J Heat Mass Transfer 2020;161. doi: https://doi.org/10.1016/j.ijheatmasstransfer.2020.120244.
[27] Rao JH, Jeng DR, Dewitt KJ. Momentum and heat transfer in a power-law fluid with arbitrary injection/suction at a moving wall. Int J Heat Mass Transfer 1999;42:2837–47.
[28] Lee MH, Jeng DR, DeWitt KJ. Laminar boundary layer transfer over rotating bodies in forced flow. J Heat Transfer 1978;100(3):496–502.
[29] Rajasekaran R, Palekar MG. Viscous dissipation effects on mixed convection about a rotating sphere. Int J Eng Sci 1985;23(8):789–95.
[30] Shokouhmand H, Soleimani M. The effect of viscous dissipation on temperature profile of a power-law fluid flow over a moving surface with arbitrary injection/suction. Energy Convers Manag 2011;52:171–9.
[31] Meissner DL, Jeng DR, DeWitt KJ. Mixed convection to power-law fluids from two-dimensional or axisymmetric bodies. Int J Heat Mass Transfer 1994;37 (10):1457–85.
[32] Kim HW, Esseniyi AJ. Forced convection of power law fluids over a rotating nonisothermal body. J Thermophys Heat Transf 1993;7(4).
[33] Chang TA, Jeng DR, De Witt KJ. Natural convection to power-law fluids from two-dimensional or axisymmetric bodies of arbitrary contour. Int J Heat Mass Transfer 1988;31(3):615–24.
[34] Cameron MR, Jeng DR, DeWitt KJ. Mixed convection from two dimensional and axisymmetric bodies of arbitrary contour Trends in Heat. Mass & Momentum Transfer 1991; 1 1–15.
[35] Howell TG, Jeng DR, Dewitt KJ. Momentum and heat transfer on a continuous moving surface in a power law fluid. Int J Heat Mass Transfer 1997;40 (8):1853–61.
[36] Yang J, Jeng DR, DeWitt KJ. Laminar free convection from a vertical plate with nonuniform surface conditions. Num Heat Transfer, Part A: Appl 1982;5:165–84.
[37] Sahu AK, Mathur MN, Chaturani P, Bharatiya SS. Momentum and heat transfer from a continuous moving surface to a power-law fluid. Acta Mech 2000;142:119–31.
[38] Jeng DR, DeWitt KJ, Lee MH. Forced convection over rotating bodies with non-uniform surface temperature. Int J Heat Mass Transfer 1979;22:89–98.
[39] Kao T, Elrod HG. Rapid calculation of heat transfer in nonsimilar laminar incompressible boundary layers. AIAA J 1976;14(12):1746–51.
[40] Gorla RSR, Tornabene R. Free convection from a vertical plate with nonuniform surface heat flux and embedded in a porous medium. Transp Porous Media 1988;3:95–106.
[41] Fagbenle RO, Karayiannis TG. Mixed free and forced convection laminar flow (ASME IMECE, San Francisco, CA.) 1995, p. 12–17.
[42] Aliu SA, Fagbenle RO. Laminar mixed boundary layer flow over two dimensional and axisymmetric bodies (ASME IMECE, IMECE2013-63447, San Diego, CA.) 2013, p. 15–21.
[43] Falana A, Fagbenle RO. Forced convection thermal boundary layer transfer for non-isothermal surfaces using the modified Merk series. Open J Fluid Dyn 2014;4:241–50.
[44] Ascher UM, Petzold LR. Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations. Philadelphia: SIAM; 1998.

O.M. Amoo Department of Mechanical Engineering, University of Ibadan, Ibadan, Oyo State, Nigeria. Corresponding author: Email address: oamoo@alumni.ucla.edu.

Currently, he is an engineer/scientist at Northrop Grumman Aerospace Systems (Melbourne, FL, USA). He was formerly a senior mechanical engineer at Raytheon Missile Systems in Tuscon, Arizona and UTC Pratt & Whitney in East Hartford, Connecticut, USA. He was also an engineer at Toyota Motor Manufacturing, USA. Together with a colleague at Pratt & Whitney, he obtained a patent for developing a new technique that could optimize fan blades for next-generation low-carbon and advanced jet engines, which continues to be a NASA-funded initiative to this day. He is a member of the American Society of Mechanical Engineers (ASME), the National Society of Black Engineers (NSBE), and the American Institute of Aeronautics and Astronautics (AIAA). His research areas are energy, exergy, nanofluids, machine learning, and laminar boundary layer convective heat transfer.

R.O. Fagbenle The late Prof. Fagbenle is an adjunct professor of mechanical engineering at the Center for Petroleum, Energy Economics and Law, University of Ibadan, Nigeria. He is also the former head of the Mechanical Engineering Department at the University of Ibadan, Nigeria. He has taught engineering courses at several schools, including the University of Illinois (Urbana- Champaign, IL, USA), Iowa State University (Ames, IA, USA), Kwame Nkrumah University of Science and Technology (Kumasi, Ghana), South Bank University (London, UK) and Covenant University (Ota, Ogun State, Nigeria). He is a fellow of the Nigerian Society of Engineers (NSE), the Solar Energy Society of Nigeria (SESN), and the Nigerian Association for Energy Economics (NAEE), and a member of the American Society of Mechanical Engineers (ASME). His areas of research are laminar boundary layers, energy, exergy, climate change, and energy policies for a more sustainable future.

M.O. Oyewola Department of Mechanical Engineering, University of Ibadan, Ibadan, Oyo State, Nigeria.

Professor Oyewola is a scholar with several citations and scientific research papers in turbulent boundary layers, fluidized bed combustion and renewable energy. He has had roles as a visiting scholar at Fiji National University, and the University of Newcastle, Australia where he earned his doctorate in mechanical engineering. He teaches various engineering courses and conducts research in the field of thermo-fluids.