Computers in Biology and Medicine 53 (2014) 30–41

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/cbm

ODE/PDE analysis of corneal curvature

Łukasz Płociniczak [a,*], Graham W. Griffiths [b], William E. Schiesser [c]

[a] Institute of Mathematics and Computer Science, Wroclaw University of Technology, ul. Janiszewskiego 14a, 50-372 Wroclaw, Poland
[b] City University, Northampton Square, London EC1V 0HB, UK
[c] Lehigh University, D118 Iacocca, 111 Research Drive, Bethlehem, PA 18015, USA

ARTICLE INFO

Article history:
Received 29 May 2014
Accepted 7 July 2014

Keywords:
Corneal curvature
Biomechanical shell
Ordinary differential equation (ODE)
Boundary value ordinary differential equation (BVODE)
Partial differential equation (PDE)
Parabolic PDE
R language
Method of Lines (MOL)

ABSTRACT

The starting point for this paper is a nonlinear, two-point boundary value ordinary differential equation (BVODE) that defines corneal curvature according to a static force balance. A numerical solution to the BVODE is computed by first converting the BVODE to a parabolic partial differential equation (PDE) by adding an initial value (t, pseudo-time) derivative to the BVODE. A numerical solution to the PDE is then computed by the method of lines (MOL) with the calculation proceeding to a sufficiently large value of t such that the derivative in t reduces to essentially zero. The PDE solution at this point is also the solution for the BVODE. This procedure is implemented in R (an open source scientific programming system) and the programming is discussed in some detail. A series approximation to the solution is derived from which an estimate for the rate of convergence is obtained. This is compared to a fitted exponential model. Also, two linear approximations are derived, one of which leads to a closed form solution. Both provide solutions very close to that obtained from the full nonlinear model. An estimate for the cornea radius of curvature is also derived. The paper concludes with a discussion of the features of the solution to the ODE/PDE system.

1. Introduction

Mathematical modeling in biological and medical sciences is currently receiving wide attention. Qualitative and quantitative models can provide a major advantage in the diagnosis and prediction of the outcome of therapies for different diseases and related health issues. In the case of vision, ophthalmologists and optometrists generally consider the cornea as one of the most important components of the eye that is crucial for normal vision [1]. For diagnostic purposes, a detailed knowledge of corneal topography and curvature is essential. To understand these features, a sufficient level of detail is achieved only by accurate measurements and correct mathematical models [2].

For corneal topography modeling, models based on various conical sections such as ellipsoids and paraboloids [3] are well established and confirmed as accurate. This approach goes back to Helmholtz [4] who proposed ellipses as descriptions of corneal profiles. Presently, this description along with its many variations is the main basis for the mathematical modeling of corneal topography (for a survey see [5] and references therein; [6] presents some generalizations). Additionally, models and associated numerical simulations of the biomechanical shell that constitutes the cornea are now widely studied (see e.g., [7,8]). These structural models are invaluable for understanding the fine details of many corneal features such as their age-dependent evolution [9] and apex displacement caused by intra-ocular pressure [10]. The greater complexity of biomechanical models provides enhanced insight of the model accuracy and physical interpretations.

This paper considers the numerical analysis of an intermediate (in accuracy and complexity) model between the conical sectional and structural mechanical approaches. The starting point for the analysis of corneal curvature is [11] a static force balance (see also its generalization [12])

$$-T\frac{d^2h/dx^2}{(\sqrt{1+(dh/dx)^2})^3}+kh=\frac{P}{\sqrt{1+(dh/dx)^2}}$$

The variables and parameters of Eq. (1) are given in Table 1.

* Corresponding author. Tel.: +48 713203183.
E-mail addresses: lukasz.plociniczak@pwr.wroc.pl (Ł. Płociniczak), graham@griffiths1.com (G.W. Griffiths), wes1@lehigh.edu (W.E. Schiesser).

http://dx.doi.org/10.1016/j.compbiomed.2014.07.003
0010-4825/

Table 1
Variables, parameters of Eq. (1).

Variable parameter	Interpretation	Units (mks)
$h(x)$	Corneal surface height	m
x	Distance from axisymmetric center	m
T	Tension	N
P	intraocular pressure	N/m
k	Elastic constant	N/m^2

To facilitate computation, it is convenient to rescale all the variables to the non-dimensional form (while retaining the original notation for clarity) and obtain

$$\frac{d^2h/dx^2}{(1+(dh/dx)^2)^{3/2}} - ah + \frac{b}{(1+(dh/dx)^2)^{1/2}} = 0 \tag{1}$$

with $a = R^2k/T, b = RP/T$ and h, x rescaled by R – typical corneal dimension such as its radius. Eq. (1) is a two-point, nonlinear boundary value ODE (BVODE) with the boundary conditions (BCs)

$$\frac{dh(x=0)}{dx} = 0 \tag{2a}$$

$$h(x=1) = 0 \tag{2b}$$

Eq. (2a) reflects the symmetry of the corneal curvature $h(x)$ around $x=0$. Eq. (2b) indicates that the height $h(x)$ is defined relative to the height at $x=1$. Eqs. (1) and (2) constitute the BVODE to be integrated numerically as explained next. For more formal and theoretical developments of the considered problem see [13].

2. PDE formulation

A variety of methods exists for the solution of BVODEs. For example, the derivatives in Eq. (1) could be replaced by finite differences. The resulting system of nonlinear algebraic equations could be solved with a variant of Newton's method. Here we consider an approach that offers significant advantages, that is, conversion of Eq. (1) to a partial differential equation (PDE).

This is accomplished by appending a derivative to Eq. (1) in a second independent variable t

$$\frac{\partial h}{\partial t} = \frac{\partial^2h/\partial x^2}{(1+(\partial h/\partial x)^2)^{3/2}} - ah + \frac{b}{(1+(\partial h/\partial x)^2)^{1/2}} \tag{3}$$

The approach then is to integrate Eq. (3) forward in t until $\partial h/\partial t \approx 0$ in which case Eq. (3) reduces to Eq. (1). Eq. (3) is a parabolic PDE with the important property that the solution is generally stable and tends to move to the equilibrium solution $\partial h/\partial t \approx 0$. In particular, this is expected when $|\partial h/\partial x| \ll 1$ in which case Eq. (3) is a form of the diffusion equation.

Eq. (3) requires two BCs and an initial condition (IC) in t. The latter can be selected arbitrarily if the solution for $h(x, t\to\infty)$ approaches the desired solution of Eq. (1). Here we will use

$$\frac{\partial h(x=0,t)}{\partial x} = 0 \tag{4a}$$

$$h(x=1,t) = 0 \tag{4b}$$

$$h(x,t=0) = 0 \tag{5}$$

Eqs. (4) follow from Eqs. (2). Eqs. (3)–(5) constitute the PDE MOL formulation of interest.

As an incidental note, t in Eq. (3) is not part of the original BVODE problem, Eqs. (1) and (2). The solution of Eqs. (3)–(5) is termed the *method of false transients* as it takes its name from the introduction of the *pseudo-time* variable t. t can also be considered as a *continuation parameter* that takes (continues) the solution from $t=0$ to the desired solution of Eq. (1) as $h(x, t\to\infty)$.

The principal advantage of this approach is that we can use well established methods for PDEs, in this case, the *method of lines (MOL)* considered next.

3. Method of lines numerical solution of PDE

The MOL is based on the replacement of the PDE boundary value (spatial) partial derivatives, e.g., $\partial^2h/\partial x^2$ in Eq. (3), with algebraic approximations. In the present case, the approximations are finite differences (FDs), although other possibilities include finite volumes, finite elements, spectral and least squares approximations. The algebraic replacement of the boundary value derivatives leaves only one independent variable, the initial value variable (time, t) so that a system of approximating ODEs results. This system of initial value ODEs can then be integrated by a library routine.

These steps in the MOL solution of Eqs. (3)–(5) are illustrated with the ODE/MOL routines `corneal_2.R` in Appendix A1 and the main program is Appendix A2. Some features of these two R routines are considered next (for more information see [14]).

4. ODE/MOL routine

The ODE/MOL R routine for Eqs. (3) and (4) is listed in Appendix A1. We can note a few particular points about this routine.

- A grid in x is defined with `nx=21` points in the main program discussed next, which is the basis for 21 ODEs approximating Eq. (3). The dependent variable vector for the 21 ODEs is `u` (in accordance with the usual convention for PDE numerical analysis, `u` rather than `h` is used for the dependent variable of Eq. (3)). Then, `corneal_2` computes 21 ODE derivatives in t which are placed in the array `ut`. To start, BC (4b) is programmed as

```
#
# BC
  u[nx] = 0;
```

  Note the use of index `nx` corresponding to $x=1$.
- The derivative $\partial h/\partial x = \partial u/\partial x$ is computed by a call to the differentiation routine `dss006`.

```
#
# ux
  ux = dss006(0, xl, nx, u);
```

  A vector of 21 values of the derivative, `ux`, results.
- BC (4a) is programmed as

```
#
# BC
  ux[1] = 0;
```

  Note the use of the index `1` corresponding to $x=0$.
- The derivative $\partial^2 h/\partial x^2 = \partial^2 u/\partial x^2$ is computed by a call to the differentiation routine `dss046`.

```
#
#uxx
  nl = 2; nu = 1;
    uxx = dss046(0, xl, nx, u, ux, nl, nu);
```

  `nl=2` and `nu=1` correspond to Neumann and Dirichlet BCs, respectively, that is BCs (4a,b).
- The MOL approximation of Eq. (3) is programmed as

```
#
#PDE
  tr = sqrt(1+1/3^2); tr3 = tr^3
  sr = sqrt(1+ux^2); sr3 = sr^3

  if(ncase==1)ut = uxx/sr3-a*u+b/sr
  if(ncase==2)ut = uxx-a*u+b/sr
  if(ncase==3)ut = uxx/tr3-a*u+b/tr
#
  ut[nx] = 0;
```

  `ncase=1`(set in the main program) uses the full eq. (3)

$$\frac{\partial h}{\partial t} = \frac{\partial^2 h/\partial x^2}{(1+(\partial h/\partial x)^2)^{3/2}} - ah + \frac{b}{(1+(\partial h/\partial x)^2)^{1/2}}$$

  `ncase=2` uses eq. (3) modified to

$$\frac{\partial h}{\partial t} = \partial^2 h/\partial x^2 - ah + \frac{b}{(1+(\partial h/\partial x)^2)^{1/2}}$$

  `ncase=3` uses eq. (3) modified to

$$\frac{\partial h}{\partial t} = \frac{\partial^2 h/\partial x^2}{(1+(1/3)^2)^{3/2}} - ah + \frac{b}{(1+(1/3)^2)^{1/2}}$$

  In this way, the effect of $1/(1+(\partial h/\partial x)^2)^{3/2}$ is investigated. Since BC 4b specifies a constant value (zero), the ODE derivative at $x=1$ is set to zero, `ut[nx]=0`.

- The vector of 21 derivatives in t, `ut`is returned to the calling main program (Listing 2 in Appendix A2) for use by the ODE integrator.

```
#
  # Return derivative vector
    return(list(c(ut)));
    }
```

The main program in Appendix A2 that calls `corneal_2` is considered next.

4.1. Main program

We can note a few particular points about the main program.

- The parameters of Eqs. (1) and (2), or (3)–(5), are defined numerically. These values are used in the calculation of the numerical solutions described subsequently.

```
#
  # Parameters
    R = 1; k = 1; P = 1; T = 1; a = R ∧ 2*k/T; b = R*P/T; nx = 21; xl = 1;
```

- The grid in x is defined on 21 points for the interval $0 \le x \le 1$ (`nx=21` is set previously as a parameter). Then IC (5) is programmed as a sequence of 21 zero values.

```
#
  # x grid, initial condition
    xg = seq(from = 0, to = xl, by = xl/(nx-1));
    u0 = rep(0, nx);
```

This completes the programming of Eqs. (3)–(5).

- The 21 ODEs are integrated by a call to `lsodes`.

```
#
  # lsodes ODE integration
    parms = c(rtol = 1e-8, atol = 1e-8)
    out = lsodes(times = tm, y = u0, func = corneal_2, parms = parms);
```

Note the use of `corneal_2`. The 21 ODE solutions are returned in `out`. This solution array is then placed in `u` for subsequent numerical and graphical (plotted) display.

- The numerical solution of Eq. (3), $u(x,t)$, and five terms from Eq. (3) are plotted as a 3×2 matrix of plots.

```
#
  # u(x,t) vs x
    par(mfrow = c(3,2));
  # par(mfrow = c(1,1));
    matplot(x = xg, y = t(u), type = "l", xlab = "x", ylab = "u(x,t)",
      xlim = c(0,xl), lty = 1, main = "u(x,t), t = 0, 0.2, ..., 1", lwd = 2)
  ⋮
      Intermediate code removed to conserve space
  ⋮
  #
  # Plot 1/(1+ux^2)^(3/2)
    matplot(x = xg, y = term1, type = "l", xlab = "x", ylab = "1/(1+ux^2)^(3/2)",
      xlim = c(0,xl), lty = 1, main = "1/(1+ux^2)^(3/2); t = 0, 0.2, ..., 1", lwd = 2);
  #
  # Plotuxx/(1+ux^2)^(3/2)
    matplot(x = xg, y = term2, type = "l", xlab = "x", ylab = "uxx/(1+ux^2)^(3/2)",
      xlim = c(0,xl), lty = 1, main = "uxx/(1+ux^2)^(3/2); t = 0, 0.2, ..., 1", lwd = 2);
  #
  # Plot-a*u
    matplot(x = xg, y = term3, type = "l", xlab = "x", ylab = "-a*u",
      xlim = c(0,xl), lty = 1, main = "-a*u; t = 0, 0.2, ..., 1", lwd = 2);
  #
  # Plot b/(1+ux^2)^(1/2)
    matplot(x = xg, y = term4, type = "l", xlab = "x", ylab = "b/(1+ux^2)^(1/2)",
      xlim = c(0,xl), lty = 1, main = "b/(1+ux^2)^(1/2); t = 0, 0.2, ..., 1", lwd = 2);
  #
  # Plot ut
```

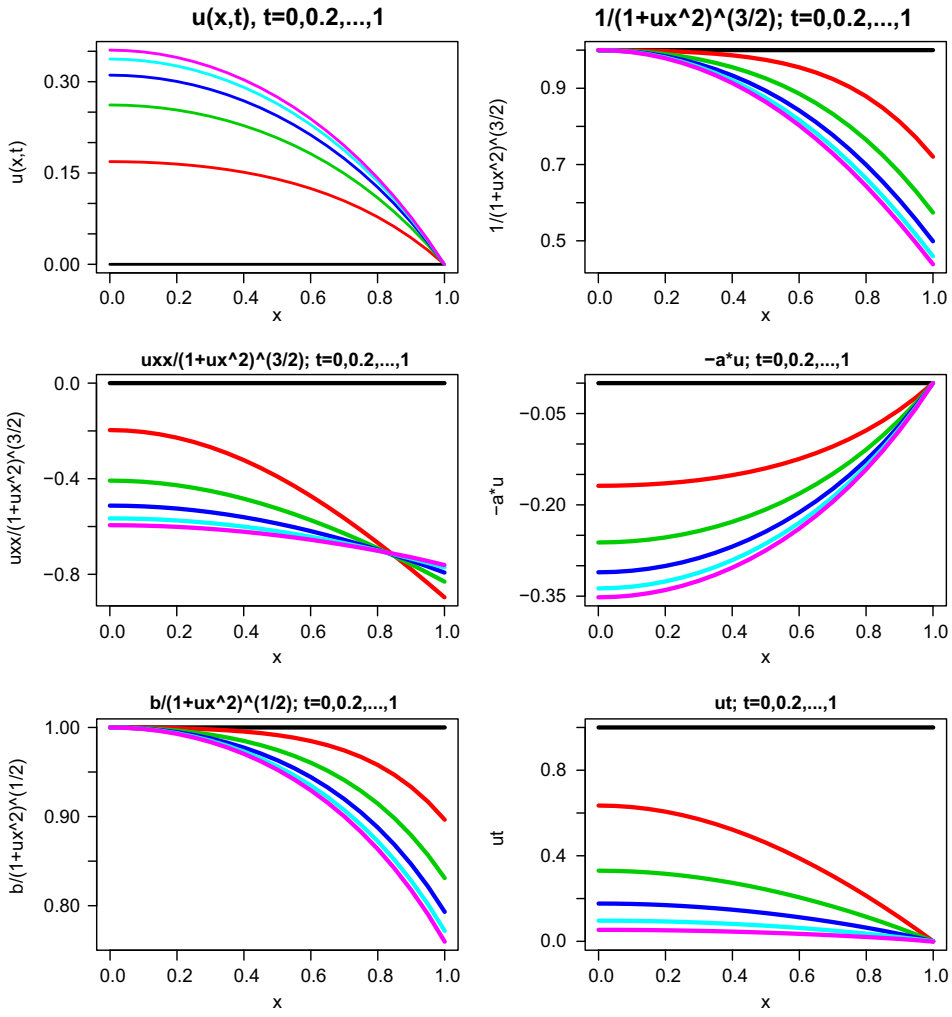

Fig. 1. Output from Listings 1,2.

Table 2
Numerical solution to Eqs. (3)–(5) for $0 \le t \le 1$.

xl=1.00	a=1.00	b=1.00			
t	u(0,t)	u(0.25xl,t)	u(0.50xl,t)	u(0.75xl,t)	u(xl,t)
0.00	0.00000	0.00000	0.00000	0.00000	0.00000
0.20	0.16870	0.16229	0.13992	0.09188	0.00000
0.40	0.26179	0.24885	0.20765	0.13039	0.00000
0.60	0.31077	0.29459	0.24392	0.15146	0.00000
0.80	0.33727	0.31942	0.26382	0.16321	0.00000
1.00	0.35188	0.33314	0.27487	0.16980	0.00000
ncall=131					

```
matplot(x=xg, y=term5, type="l", xlab="x", ylab="ut",
  xlim=c(0,xl), lty=1, main="ut; t=0,0.2,...,1", lwd=2);
```

The solution $u(x,t)=$ `u` is plotted first (using the transpose `t(u)` so that the row dimension of `u` is the same as the length of $x=$`xg`). The five terms `term1`,...,`term5` plotted against `xg` are identified by the comments for each plot. Note in particular the vector of derivatives in t, $\partial u/\partial t=$`ut=term5` computed as the sum of the RHS terms of Eq. (3).

This description of the MOL solution of Eqs. (3)–(5) is abbreviated to conserve space. A more complete description is given in [15], Chapter 8. A discussion of the method applied to 2D problems is given in [16], Chapter 10.

The composite 3×2 plot is in Fig. 1.

Table 3
Numerical solution to Eqs. (3)–(5) for $0 \le t \le 5$.

xl=1.00	a=1.00	b=1.00			
t	u(0,t)	u(0.25xl,t)	u(0.50xl,t)	u(0.75xl,t)	u(xl,t)
0.00	0.00000	0.00000	0.00000	0.00000	0.00000
1.00	0.35188	0.33314	0.27487	0.16980	0.00000
2.00	0.36940	0.34963	0.28824	0.17782	0.00000
3.00	0.37041	0.35057	0.28900	0.17828	0.00000
4.00	0.37046	0.35063	0.28905	0.17831	0.00000
5.00	0.37047	0.35063	0.28905	0.17831	0.00000
ncall=249					

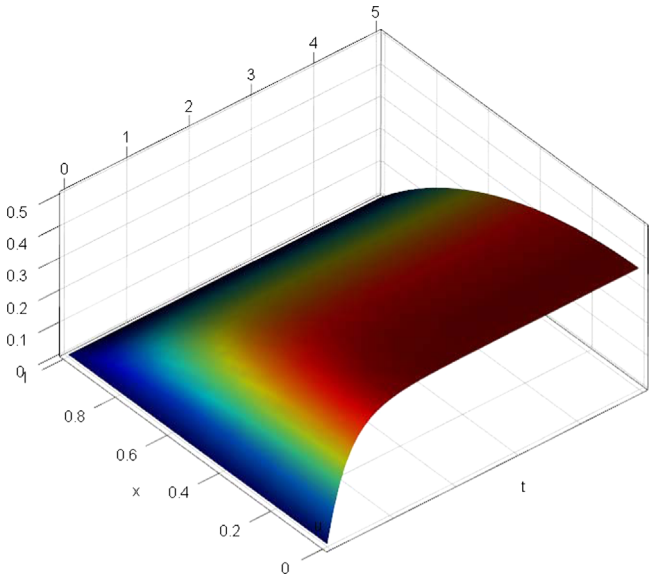

Fig. 2. Complete solution presented as a surface plot.

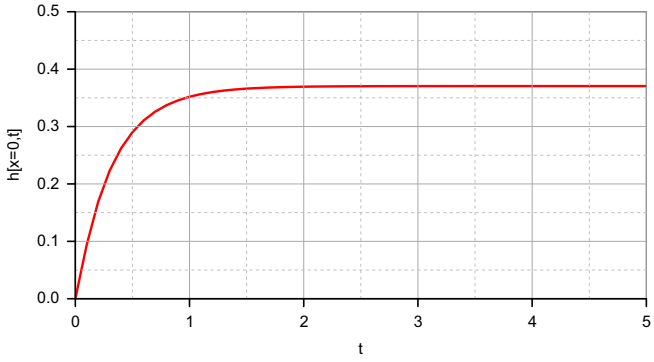

Fig. 3. Evolution of the solution at $x=0$.

(4) Numerical and graphical solutions

We can note the following particular details of Fig. 1.

- All six plots start at $t=0$ with a constant value from the IC `u0[i]=0` discussed previously (note the horizontal line in each plot).
- The curves in all six plots appear to be approaching a final steady state through the six successive values $t=0, 0.2, \ldots, 1$. In particular, the $(1,1)$ plot (top left corner), $u(x=0, t=1) \approx 0.35$. This is confirmed with the following numerical output from Listing 2 with `u(0,t)=0.35188` for $t=1$ (Table 2). Also, `u(xl,t)=0.00000` in accordance with BC (4b). The number of calls to `corneal_2`, `131`, is quite modest indicating `lsodes` efficiently integrated the 21 ODEs.
- As the solution approaches a steady state, $\partial u/\partial t \approx 0$ in the $(3,2)$ plot (lower right corner) as expected.
- The contributions of the three RHS terms of Eq. (3) are clear in the $(2,1), (2,2), (3,1)$ plots. In this way, the properties of a PDE solution can be explained in detail.

- The approach to a steady state is further confirmed in Table 3 for $0 \le t \le 5$ (by changing `tf=1` to `tf=5` in the main program of Listing 2). `u(0,t)=0.37046` for $t=4$ and `u(0,t)=0.37047` for $t=5$ infers convergence of the solution to five significant figures.

The complete solution is presented in Fig. 2 as a surface plot that illustrates how the solution evolves in space and pseudo-time. It is clear that the solution quickly and smoothly approaches a steady state. It is particularly interesting to observe how the solution at $x=0$ evolves, as this location is subject to the *Neumann* boundary condition $\partial h(x=0,t)/\partial x = 0$. The method of lines handles this constraint with ease, as demonstrated in Fig. 3.

5. Analytical approximation

A simple analytical demonstration of the method of false transients follows from a linearization of the nonlinear equation (1). According to the Gullstrand eye model, the typical cornea has the radii of curvature ρ equal to 7.7 mm for anterior and 6.8 mm for posterior surfaces respectively [17]. Moreover, since the typical diameter d of the cornea is equal to 11.5 mm, by the formula for circular segment

$$h = \rho - \sqrt{\rho^2 - \frac{d^2}{4}},$$

we can calculate that the height of the cornea is approximately equal to 2.5 mm for an ordinary cornea. Using that value we have $|dh/dx| \approx 2.5/5.75 \approx 0.43$ and thus

$$\frac{1}{(1+(\partial h/\partial x)^2)^{1/2}} \approx 0.92, \quad \frac{1}{(1+(\partial h/\partial x)^2)^{3/2}} \approx 0.78. \tag{6}$$

These quantities are only roughly close to 1, but in the first approximation it is reasonable to neglect the derivative $\partial h/\partial x$ and obtain the approximation of Eq. (1)

$$-\frac{d^2h}{dx^2} + ah = b \tag{7}$$

and its PDE generalization (3)

$$\frac{\partial h}{\partial t} = \frac{\partial^2 h}{\partial x^2} - ah + b \tag{8}$$

along with initial-boundary conditions (4) and (5). Due to the linearity of Eq. (8), a solution follows easily by the separation of variables. To this end, we note that as $t \to \infty$ the solution of Eq. (8) should approach the steady-state that satisfies Eq. (7), that is

$$h_p(x) = \frac{b}{a}\left(1 - \frac{\cosh(\sqrt{a}x)}{\cosh\sqrt{a}}\right) \tag{9}$$

It is thus reasonable to look for a solution of Eq. (8) in the form of

$$h(x,t) = u(x,t) + h_p(x) \tag{10}$$

where $u(x,t)$ is the homogeneous solution and $h_p(x)$ is a particular solution. Substituting Eq. (10) into Eq. (8) we find that u is a solution of the following homogeneous equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - au \tag{11}$$

with boundary conditions (4) and initial condition

$$u(x, t=0) = -h_p(x) \tag{12}$$

By assuming the separated solution $u(x,t) = X(x)T(t)$ we obtain two ordinary differential equations

$$T' = -(a+\lambda)T, \quad X'' = -\lambda X, \quad \lambda > 0 \tag{13}$$

where primes denote differentiation with respect to the corresponding ODE independent variables and λ is a separation constant chosen positive in anticipation of the decaying transient. We of course have to have $X'(0) = X(1) = 0$, which along with Eq. (13) gives us a solution as a constant multiple of $\cos\sqrt{\lambda}x$, where $\lambda = (\pi/2 + k\pi)^2$. By solving also for T and using the linearity of Eq. (11) we arrive at

$$u(x,t) = \sum_{k=0}^{\infty} A_k \cos\left(\left(\frac{\pi}{2} + k\pi\right)x\right) e^{-(a+(\pi/2+k\pi)^2)t} \tag{14}$$

where A_k is calculated from the initial condition

$$A_k = -2\int_0^1 h_p(x) \cos\left(\left(\frac{\pi}{2} + k\pi\right)x\right) dx = \frac{(-1)^{k+1} 16b}{\pi(1+2k)(4a+(1+2k)^2\pi^2)} \tag{15}$$

This finally gives us the solution of Eq. (8) with Eqs. (4) and (12)

$$u(x,t) = \frac{16b}{\pi} \sum_{k=0}^{\infty} \frac{(-1)^{k+1} \cos\left(\left(\frac{\pi}{2} + k\pi\right)x\right) e^{-(a+(\pi/2+k\pi)^2)t}}{(1+2k)(4a+(1+2k)^2\pi^2)} \tag{16}$$

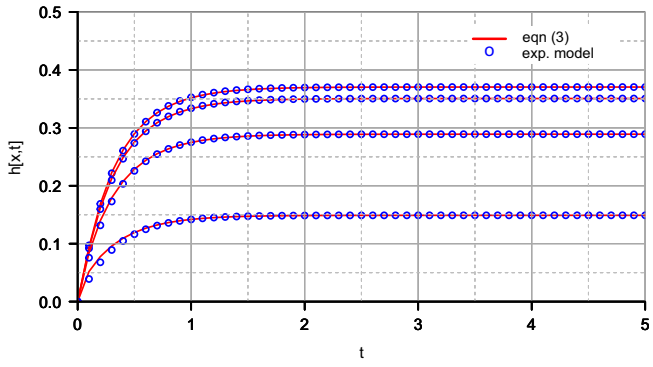

Fig. 4. Plots of $h(x_i,t)$ (solid) overlaid with $\hat{h}_i(t)$ (circles) from Eq. (18) for $x_i \in (0, 0.25, 0.5, 0.75)$ with the top plot being for $x_i = 0$.

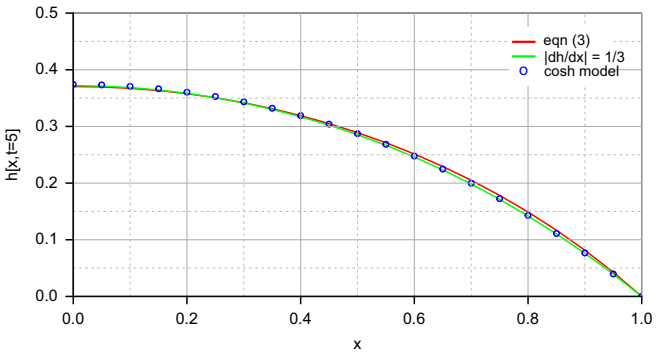

Fig. 5. Comparison of solutions using (red solid) the *full nonlinear* model of Eq. (3); (green solid) the *simple approximation* $|dh/dx| = 1/3$ of Eq. (6); and (blue circles) the closed form *hyperbolic approximation* of Eq. (21). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Notice that as a characteristic feature of parabolic equations an exponential decay in t is present in each term in Eq. (16), which is essential in the false transient method. This means that for $t > 0$ all terms apart from the first vanish rapidly leaving

$$u(x,t) \approx -\frac{16b}{\pi}\frac{\cos\left(\frac{\pi}{2}x\right)e^{-(a+\pi^2/4)t}}{4a+\pi^2} \tag{17}$$

which immediately gives the transient time scale $1/(a+\pi^2/4)$. We see that increasing parameter a increases the rate of decay of the solution. This shows a major advantage of the false transient method: exponential decay forces the solution of parabolic problem (3)–(5) to quickly reach the steady-state (equilibrium solution).

A check on the rate of decay can be made by fitting an exponential model to the evolution of h. As an example, let us focus on the case of $a = b = 1$. Using the built-in function `nls()` in R, the following exponential model was fitted to $h(x = 0, t)$,

$$\hat{h}_0(t) = A_0(1 - \exp(-Bt)), \quad A_0 = h(x = 0, t = 5), \quad B = 3.039 \tag{18}$$

which has a maximum absolute error of 0.003. The value of B compares well with the corresponding value from Eq. (17), i.e. $(a+\pi^2/4) = 3.47$. This idea can be extended over the full range of $h(x,t)$ and Fig. 4 shows $h(x_i,t)$ overlaid with values from the exponential model $\hat{h}_i(t)$, where $A_i = h(x = x_i, t = 5)$ and $x_i \in (0, 0.25, 0.5, 0.75)$. The approximate exponential model fits the MOL solution very well for all values of x_i, even using the same value for B throughout. The differences for $x_i \in (0.5, 0.75)$ can be attributed to the fast decay of the higher order terms of Eq. (16), as mentioned above.

The comparison in Fig. 5 of the solutions computed by the R routines in Appendices A and B with variable dh/dx and $|dh/dx| = 1/3$ for $a = b = 1$ (case 3) demonstrates the validity of our choice of the approximation, $|dh/dx| \approx 1/3$ (when eq. (3) has the analytical solution $ha(x, t = \infty) = 0.9487 - 0.5767\cosh(1.0822x)$). The argument for approximating nonlinearities in Eq. (1) can be made somewhat general. In that equation we can replace dh/dx by the average value of the derivative of the approximation h_p (9), which is

$$\frac{dh}{dx} \approx \int_0^1 \frac{dh_p(x)}{dx}\,dx = h_p(1) - h_p(0) = -\frac{b}{a}\left(1 - \frac{1}{\cosh\sqrt{a}}\right) \tag{19}$$

By taking the particular values of $a = b = 1$ this approach gives $|dh/dx| \approx 0.35$ (close to 1/3). Substituting (19) into (1) linearizes the problem giving an accurate "averaged" approximation

$$h_a(x) = \frac{b}{ac}\left(1 - \frac{\cosh\sqrt{ac^3}x}{\cosh\sqrt{ac^3}}\right) \quad \text{where } c = \sqrt{1 + \frac{b^2}{a^2}\left(1 - \frac{1}{\cosh\sqrt{a}}\right)^2} \tag{20}$$

Inserting the above values for a and b into Eq. (20) gives the following compact hyperbolic form[1] for an approximation to $h(x)$, based on the accurate averaged value for $|dh/dx|$

$$h_a(x) = 0.9433 - 0.5692\cosh(1.0915x) \tag{21}$$

Fig. 5 includes a plot of this approximation which is very close to the solution of the full nonlinear model of Eq. (3), with a maximum absolute error of 0.006.

The closed form solution of Eq. (21) also provides a simple way of estimating the *radius of curvature*, ρ. From

$$\rho = \left|\frac{(1+h'^2)^{3/2}}{h''}\right|$$

we obtain the following estimate for the cornea radius of curvature

$$\rho(x) = \frac{\cosh\sqrt{ac^3}}{bc^2}\left(1+\frac{b^2c}{a\cosh^2\sqrt{ac^3}}\sinh^2\left(\sqrt{ac^3}x\right)\right)^{3/2} \tag{22}$$

In particular, we get a useful estimate for the central radius

$$\rho(0) = \frac{\cosh\sqrt{ac^3}}{bc^2} \tag{23}$$

Using the values of $a = b = 1$ we obtain

$$\rho(x) = 1.475\frac{(1+0.386\sinh^2(1.092x))^{3/2}}{\cosh(1.0915\,x)} \tag{24}$$

giving $\rho(0) = 1.475$ in non-dimensional units. This compares well with the value of

$$\rho(0) = \left|\frac{1}{ah(0)-b}\right| = 1.59 \tag{25}$$

obtained from the full non-linear model of Eq. (1) and BC (2a). Ophthalmologists measure the curvature of cornea in almost every eye examination and can diagnose some corneal problems (like astigmatism) from it. Having an accurate approximation to this radius is very valuable to assist in a diagnosis.

6. Summary

In conclusion, the solution of BVODE (1) by integration of the parabolic PDE (3) is straightforward by the MOL. An important advantage of this approach is the use of a quality library ODE integrator, `lsodes` from the R package `deSolve` [18], which computed the solution to the 21 MOL/ODEs accurately and efficiently. This conclusion reflects our general experience with the numerical MOL integration of PDE systems. An estimate for the rate of convergence to the solution for $h(x,t)$ is given by Eq. (17) in the form of a truncated series solution. This compares very well to the fitted exponential model of Eq. (18), as illustrated in Fig. 4.

Finally, we have demonstrated two approximations that can be used to linearize the model. The first uses a *simple average* value for the gradient of $|dh/dx| \approx 1/3$ from Eq. (6) that is substituted into Eq. (1) and solved by MOL. The second is a more *accurate gradient* approximation given by Eq. (19) that leads to the *closed-form* approximate solution of Eq. (21). Both these approximations give very good predictions for corneal surface height $h(x)$, the required solution to Eq. (1), as illustrated in Fig. 5. The closed form solution also leads to a simple estimate for the cornea radius of curvature.

Conflict of interest statement

None declared.

Acknowledgments

Ł.Płociniczak was supported by Polish Government funds for science: research project NCN 2012/05/N/ST1/02860. NCN had no involvement in the research.

A. ODE/MOL routine

Listing 1. ODE/MOL routine `corneal_2.R`.

```
corneal_2=function(t,u,parms){
#
```

[1] This closed form model is similar to that of an inverted catenery.

```
# Function corneal_2 computes the PDE t derivative
#
# Declare (preallocate) arrays
  sr=rep(0,nx);
  ux=rep(0,nx);
  uxx=rep(0,nx);
  ut=rep(0,nx);
#
# BC
  u[nx]=0;
#
#  ux
  ux=dss006 (0,xl,nx,u);
#
# BC
  ux[1]=0;
#
#  uxx
  nl=2; nu=1;
    uxx=dss046 (0,xl,nx,u,ux,nl,nu);
#
#  PDE
  tr=sqrt(1+1/3^2); tr3=tr^3;
  sr=sqrt(1+ux^2); sr3=sr^3;

  if(ncase==1)ut=uxx/sr3-a*u+b/sr;
  if(ncase==1)ut=uxx/sr3-a*u+b/sr;
  if(ncase==3)ut=uxx/tr3-a*u+b/tr;
  #
  ut[nx]=0;
#
# Increment calls to corneal_2
  ncall<<-ncall+1;
#
#  Return derivative vector
  return(list(c(ut)));
  }
```

B. Main program

Listing 2. Main program that calls `corneal_2.R`.

```
#
#  Corneal shape ODE/PDE
#
#  Remove previous workspaces
  rm(list=ls(all=TRUE));
#
#  Access deSolve library
   (with lsodes)
  library("deSolve")
#
#  ODE/PDE routines
  setwd("g:/parabolic");
  source("corneal_2.R");
  source("dss006.R");
  source("dss046.R");
#
#  Select case
#
#  ncase=1 - nonlinear ODE
#  ncase=2 - approximate nonlinear ODE
#  ncase=3 - approximate linear ODE
#
```

```
  ncase=1;
#
#  Parameters
  R=1; k=1; P=1; T=1; a=R^2*k/T; b=R*P/T; nx=21; xl=1;
#
#  Write selected parameters
  cat(sprintf('\n xl=%5.2f  a=%5.2f b=%5.2f  \n', xl,a,b))
#
#  x grid, initial condition
  xg=seq(from=0,to=xl,by=xl/(nx-1));
  u0=rep(0,nx);
  for(i in 1:nx){
  u0[i]=0;
  }
#
#  Output sequence in t
  tf=1;nout=6;ncall=0;
  tm=seq(from=0,to=tf,by=tf/(nout-1));
#
#  lsodes ODE integration
  parms=c(rtol=1e-8,atol=1e-8)
  out=lsodes(times=tm,y=u0,func=corneal_2,parms=parms);
#
#  Numerical output
  cat(sprintf("t u(0,t) u(0.25xl,t) u(0.50xl,t) u(0.75xl,t) u(xl,t)"));
  u=matrix(0,nrow=nout,ncol=nx);
  for(it in 1:nout){
    for(i in 1:nx){
      if(it==1){
        u[1,i]=u0[i];
      }else{
        u[it,i]=out[it,i+1];
      }
    }
    u[it,nx]=0;
    cat(sprintf(" tm[it],u[it,1],u[it,6],u[it,11],u[it,16],u[it,21]));
    }
#
#  Calls to corneal_2
  cat(sprintf(" ncall=%5d \n,ncall))
#
# u(x,t) vs x
  par(mfrow=c(3,2));
#  par(mfrow=c(1,1));
  matplot(x=xg,y=t(u),type="l",xlab="x",ylab="u(x,t)",
  xlim=c(0,xl),lty=1,main="u(x,t), t=0,0.2,...,1",lwd=2)
#
#  Supplemental calculations
  term1=matrix(0,nrow=nx,ncol=nout);
  term2=matrix(0,nrow=nx,ncol=nout);
  term3=matrix(0,nrow=nx,ncol=nout);
  term4=matrix(0,nrow=nx,ncol=nout);
  term5=matrix(0,nrow=nx,ncol=nout);
#
#  Step through t
  ux=matrix(0,nrow=nx,ncol=nout);
  uxx=matrix(0,nrow=nx,ncol=nout);
  for(it in 1:nout){
#
#  1/(1+ux^2)^(3/2)
  u[it,nx]=0;
  ux[,it]=dss006(0,xl,nx,u[it,]);
  term1[,it]=1/(1+ux[,it]^2)^(3/2);
#
#  uxx/(1+uxx^2)^(3/2)
```

```
  ux[1,it]=0;
  nl=2;nu=1;
  uxx[,it]=dss046(0,xl,nx,u[it,],ux[,it],nl,nu);
  term2[,it]=term1[,it]*uxx[,it];
#
# -a*u
  term3[,it]=-a*u[it,];
#
# b/(1+ux^2)^(1/2)
  term4[,it]=b/(1+ux[,it]^2)^(1/2);
#
# ut
  term5[,it]=term2[,it]+term3[,it]+term4[,it];
  }
#
# Plot 1/(1+ux^2)^(3/2)
  matplot(x=xg,y=term1,type="l",xlab="x",ylab="1/(1+ux^2)^(3/2)",
  xlim=c(0,xl),lty=1,main="1/(1+ux^2)^(3/2); t=0,0.2,...,1",lwd=2);
#
# Plot uxx/(1+ux^2)^(3/2)
  matplot(x=xg,y=term2,type="l",xlab="x",ylab="uxx/(1+ux^2)^(3/2)",
  xlim=c(0,xl),lty=1,main="uxx/(1+ux^2)^(3/2); t=0,0.2,...,1",lwd=2);
#
# Plot -a*u
  matplot(x=xg,y=term3,type="l",xlab="x",ylab="-a*u",
  xlim=c(0,xl),lty=1,main="-a*u; t=0,0.2,...,1",lwd=2);
#
# Plotb/(1+ux^2)^(1/2)
  matplot(x=xg,y=term4,type="l",xlab="x",ylab="b/(1+ux^2)^(1/2)",
  xlim=c(0,xl),lty=1,main="b/(1+ux^2)^(1/2; t=0,0.2,...,1",lwd=2);
#
# Plot ut
  matplot(x=xg,y=term5,type="l",xlab="x",ylab="ut",
  xlim=c(0,xl),lty=1,main="ut; t=0,0.2,...,1",lwd=2);
```

References

[1] W. Trattler, P. Majmudar, J.I. Luchs, T. Swartz, Cornea Handbook, Slack Incorporated, Thorofare, NJ, USA, 2010.
[2] S. Talu, M. Talu, An overview on mathematical models of human corneal surface, IFMBE Proc. 26 (2009) 291–294.
[3] M. Guillon, D.P.M. Lydon, C. Wilson, Corneal topography: a clinical model, Ophthalmic Physiol. Opt. 6 (1986) 47–56.
[4] H. von Helmholtz, H. Southall, P.C. James (Ed.), Helmholtz's treatise on physiological optics, 1924.
[5] R. Lindsay, G. Smith, D. Atchison, Descriptors of corneal shape, Optom. Vis. Sci. 75 (2) (1998) 156–158.
[6] H. Burek, W.A. Douthwaite, Mathematical models of the general corneal surface, Ophthalmic Physiol. Opt. 13 (1993) 68–72.
[7] K. Anderson, A. El-Sheikh, T. Newson, Application of structural analysis to the mechanical behaviour of the cornea, J. R. Soc. Interface 1 (2004) 3–15.
[8] A. Pandolfi, F. Manganiello, A model for the human cornea: constitutive formulation and numerical analysis, Biomech. Model. Mechanobiol. 5 (4) (2006) 237–246.
[9] A. Elsheikh, D. Wang, M. Brown, P. Rama, M. Campanelli, D. Pye, Assessment of corneal biomechanical properties and their variation with age, Curr. Eye Res. 32 (1) (2007) 11–19.
[10] B.L. Boyce, J.M. Grazier, R.E. Jones, T.D. Nguyen, Full-field deformation of bovine cornea under constrained inflation conditions, Biomaterials 29 (28) (2008) 3896–3904.
[11] W. Okrasiński, Ł. Płociniczak, A nonlinear mathematical model of the corneal shape, Nonlinear Anal.: Real World Appl. 13 (2012) 1498–1505.
[12] W. Okrasiński, Ł. Płociniczak, Bessel function model of corneal topography, Appl. Math. Comput. 223 (2013) 436–443.
[13] Ł. Płociniczak, W. Okrasiński, J.J. Nieto, O. Dominguez, On a nonlinear boundary value problem modeling corneal shape, J. Math. Anal. Appl. 414 (1) (2014) 461–471.
[14] R Core Team (2013), R: a language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, Austria. ⟨http://www.Rproject.org⟩.
[15] W.E. Schiesser, Differential Equation Analysis in Biomedical Science and Engineering: Ordinary Differential Equation Applications in R, John Wiley, Hoboken, NJ, 2014.
[16] W.E. Schiesser, G.W. Griffiths, A Compendium of Partial Differential Equation Models: Method of Lines Analysis with Matlab, Cambridge University Press, 2009.
[17] B. Vojnikovi, E. Tamajo, Gullstrands Optical Schematic System of the Eye Modified by Vojnikovi & Tamajo, Coll. Antropol. 37 (1) (2013) 41–45.
[18] K. Soetaert, T. Petzoldt, R.W. Setzer, Solving differential equations in R: package deSolve, J. Stat. Softw. 33 (9) (2010) 1–25.