# Shaping the chameleon

Jonathan A. Pearson*

*School of Physics & Astronomy*
*University of Nottingham*
*Nottingham, NG7 2RD*

October 22, 2014

**Abstract**

In conservation with Clare Burrage, Ed Copeland, James Stevenson, Adam Moss...

## Contents

**1 Introduction** **1**

**2 The model** **1**
2.1 Specification of the source density profile . . . . . 2
2.2 Boundary conditions and parameter values . . . . . 2

**3 Results** **2**
3.1 Circular sources . . . . . 2
3.2 Ellipsoidal sources . . . . . 2
3.3 Other shapes . . . . . 3

**Acknowledgements** **3**

**A Numerical methods** **4**
A.1 Gradient flow . . . . . 4
A.2 Discretization schemes . . . . . 5
A.2.1 Fourth order finite difference derivatives . . . . . 5
A.2.2 SoR . . . . . 5
A.3 Simulated annealing . . . . . 5

**References** **6**

*E-mail: j.pearson@nottingham.ac.uk

# 1 Introduction

The idea is to understand the differences between the chameleon force and gravitational force for source objects with different shapes – circles, ellipsoids, etc.

The idea is to build on the analytic results and understanding developed by Burrage, Copeland, and Stevenson – they obtained analytic solutions to the equations whilst using some simplifying (and justified) assumptions. They obtained the fields surrounding an ellipsoid source. We intend to reproduce their results, and extend to other – more complicated – shapes.

# 2 The model

In the static regime the chameleon scalar $\phi$ satisfies

$$\nabla^2\phi = -\frac{\Lambda^5}{\phi^2} + \frac{\rho}{M}, \tag{2.1a}$$

and the gravitational potential $\Phi$ satisfies Laplace's equation,

$$\nabla^2\Phi = -\rho \tag{2.1b}$$

The forces due to the chameleon and gravitational scalars are computed by taking the gradient of the relevant scalar:

$$\mathbf{F}_{(\phi)} = \nabla\phi, \qquad \mathbf{F}_{(\Phi)} = \nabla\Phi. \tag{2.2}$$

We solve (2.1a) using numerical methods outlined in Appendix A for a given source density function, $\rho(\mathbf{x})$.

## 2.1 Specification of the source density profile

We use a "step-function" to describe $\rho$, whereby the density has value $\rho_0$ in the interior of the source, and $\rho_{\text{bg}}$ in the exterior of the source (this is supposed to represent the ambient, possibly cosmological, density). To be concrete, we set

$$\rho(\mathbf{x}) = \begin{cases} \rho_0 & \text{inside,} \\ \rho_{\text{bg}} & \text{outside.} \end{cases} \tag{2.3}$$

The spatial locations which are defined as "inside" and "outside" are those inside the given shape under consideration (circles, ellipses, etc).

## 2.2 Boundary conditions and parameter values

For boundary conditions we set

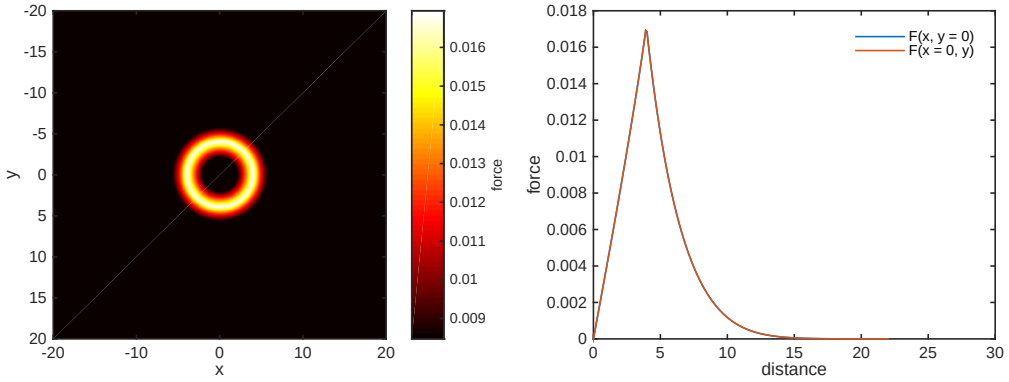
$$\phi(\mathbf{x}_\infty) = \phi_{\text{bg}} = \sqrt{\frac{M\Lambda^5}{\rho_{\text{bg}}}}. \tag{2.4}$$

This is the value of the scalar which minimizes the effective potential.

Typically we set

$$\Lambda = 1, \qquad M = 10^3, \qquad \rho_0 = 12, \qquad \rho_{\text{bg}} = 0.1. \tag{2.5}$$

This is the hierarchy of scales required to obtain the screening solutions of interest. Note that we require a large density contrast, $\delta \equiv \frac{\rho_0}{\rho_{\text{bg}}} - 1$, and large $M$.




(a) Force density (2D) (b) Force as function of distance from origin

*Figure 1: The force due to the chameleon around a circular source. This source has size $R = 4$. The other parameter values are those given in (2.5). Note that the force increases linearly from the origin, until some maximum is reached at the objects surface: the force then decreases as $r^{-2}$. [replace with converged versions]*

# 3 Results

## 3.1 Circular sources

Here we define the shape to be inside the circle

$$x^2 + y^2 = R^2. \tag{3.1}$$

See Figure 1

## 3.2 Ellipsoidal sources

Here we define the shape to be inside the ellipse

$$\left(\frac{x}{b}\right)^2 + \left(\frac{y}{a}\right)^2 = R^2 \tag{3.2}$$

See Figure 2.

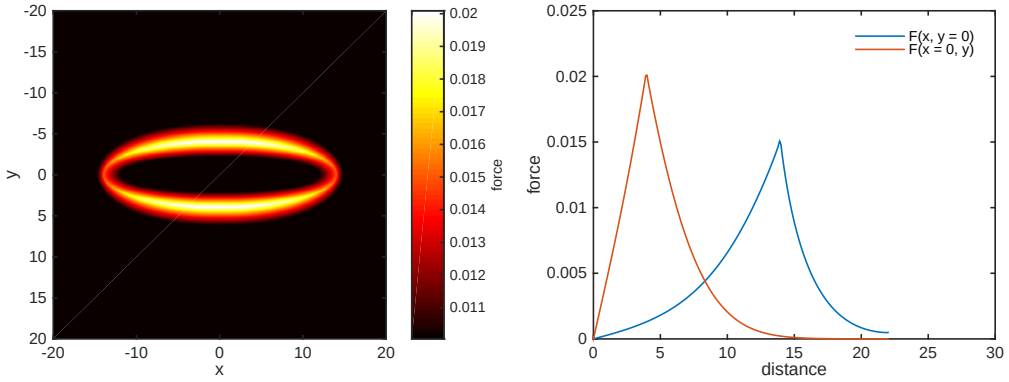
## 3.3 Other shapes

Figure 3.

# Acknowledgements

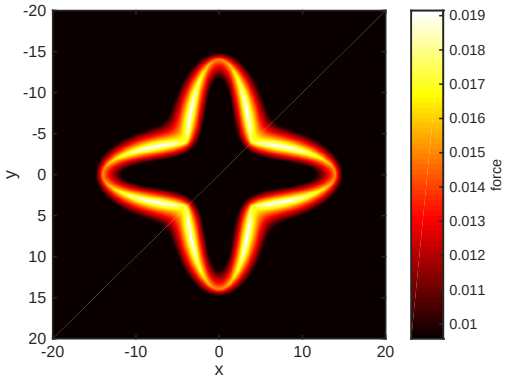
# A Numerical methods

In summary, our numerical methodology is:




(a) Force density (2D) (b) Force as function of distance from origin

*Figure 2: Force due to the chameleon around an ellipsoidal source. This source has* $a = 3.5, b = 1$, *and* $R = 4$ *as defined in (3.2); the densities and all other parameters are the same as the source shown in Figure 1. Here we observe that the force profiles along the* $x$- *and* $y$-*axes are rather different. The force at the object's surface at* $y = 0$ *is lower than the force at the object's surface at* $x = 0$. *Additionally, the force profile inside the object is markedly different along the two axes: along the* $y$-*axis the force profile is approximately linear, but along the* $x$-*axis it is parabolic. [replace with converged versions]*




(a) Crossed ellipses

*Figure 3: The force due to the chameleon around various sources. In (a) there are two ellipses at right-angles each with the same properties as those in Figure 2. [replace with converged versions]*

- Solve for the chameleon scalar (2.1a) via gradient flow; finite difference derivatives discretized to fourth order accuracy.
- Solve Poisson's equation (2.1b) via SoR; discretized to second order accuracy.
- The forces (2.2) are computed with finite difference derivatives discretized to fourth order

In the remainder of this Appendix we shall explain the implementation of each of these methods.

## A.1 Gradient flow

The scalar field profile is obtained by solving the gradient flow equation:

$$\dot{\phi} = \nabla^2\phi - \frac{\mathrm{d}V_{\text{eff}}}{\mathrm{d}\phi}. \tag{A.1}$$

We discretize the time derivative in a very simple manner,

$$\dot{\phi} \approx \frac{\phi(t+1) - \phi(t)}{h_t}. \tag{A.2}$$

This can be used on the LHS of (A.1) to yield an updating algorithm for the scalar; obviously, we also need a scheme to approximate the Laplacian: that is discussed in the next section. Note that we use a "fictitious time-step", $h_t$: this must be chosen to be much smaller than any step-sizes in the spatial dimensions (else the problem is badly defined numerically).

A "solution" to the static problem is obtained when $\dot{\phi} = 0$. This is quantified by computing an error estimate

$$E_\phi \equiv \int \mathrm{d}^2x\, \dot{\phi}, \tag{A.3}$$

and waiting until $E_\phi \ll \varepsilon$, where $\epsilon$ is supposed to be a very small number, before a solution is declared to have been found. We are also estimating the error on a solution by computing the evolution of the integral of the modulus of the force,

$$E^{\text{F}}_\phi \equiv \int \mathrm{d}^2x\, \left|\mathbf{F}_{(\phi)}\right|; \tag{A.4}$$

this quantity should be independant of time, and so we wait until $\dot{E}^{\text{F}}_\phi \ll \varepsilon$ before declaring that a solution has been found.

## A.2 Discretization schemes

Here we explain the discretization schemes used. Space is discretized onto a lattice with step-size $h$.

### A.2.1 Fourth order finite difference derivatives

The first and second derivatives of a quantity $Q$, say, are approximated at a given location via a fourth-order accurate finite difference scheme according to

$$\frac{\partial Q}{\partial x} \approx \frac{-Q_{i+2} + 8Q_{i+1} - 8Q_{i-1} + Q_{i-2}}{12h}, \tag{A.5}$$

$$\frac{\partial^2 Q}{\partial x^2} \approx \frac{-Q_{i+2} + 16Q_{i+1} - 30Q_i + 16Q_{i-1} - Q_{i-2}}{12h^2}. \tag{A.6}$$

There are obvious equivalents for the $y$-derivatives.

### A.2.2 SoR

This is used to solve Laplace's equation. Fictitious "time-steps" are used, and indexed by $n$. Convergence is determined by the parameter $\omega$. The updating algorithm is

$$Q_{i,j}^{n+1} = (1-\omega)Q_{i,j}^{n} + \frac{\omega}{4}\left[Q_{i+1,j}^{n} + Q_{i-1,j}^{n+1} + Q_{i,j+1}^{n} + Q_{i,j-1}^{n+1} + h^2\rho\right] \tag{A.7}$$

## A.3 Simulated annealing

We have a suggestion to obtain the optimal shape via simulated annealing. This would be implemented by first specifying some given matter distribution (e.g., a sphere), and computing the scalar and gravitational forces. A new shape is randomly proposed – by switching "on" or "off" locations which are supposed to have matter. If the force discrepancies are greater for this new shape, it is kept, and the whole process is repeated until an optimal shape is obtained. One needs to be careful to only proposed connected objects.

This strategy is relatively straight-forward, but computationally very intensive – one way to help is to parallelise the code.