Shaping the chameleon

Jonathan A. Pearson*

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

October 17, 2014

Abstract

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

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

Contents

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.

2 The model

In the static regime the chameleon scalar ϕ satisfies

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

and the gravitational potential Φ satisfies Laplace's equation,

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

We compute the forces due to the chameleon and gravitational scalars by taking the gradient of the relevant scalar:

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

3 Numerical methods

- Solve (2.1a) via gradient flow; finite difference derivatives discretized to fourth order accuracy.
- Solve (2.1b) via SoR; discretized to second order accuracy.
- The forces (2.2) are computed with finite difference derivatives discretized to fourth order

3.1 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.

Acknowledgements

A Schemes

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

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

Fourth order finite difference derivatives

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

$$\frac{\partial^2 Q}{\partial x^2} \approx \frac{-Q_{i+2} + 16Q_{i+1} - 30Q_i + 16Q_{i-1} - Q_{i-2}}{12h^2} \tag{A.3}$$