Available online at www.sciencedirect.com
ScienceDirect

Journal of Computational Physics 220 (2006) 6–18

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Short note

Fast geodesics computation with the phase flow method

Lexing Ying *, Emmanuel J. Candès

Applied and Computational Mathematics, California Institute of Technology, Caltech, MC 217-50, Pasadena, CA 91125, United States

Received 7 February 2006; received in revised form 9 May 2006; accepted 31 July 2006
Available online 12 September 2006

Abstract

This paper introduces a novel approach for rapidly computing a very large number of geodesics on a smooth surface. The idea is to apply the recently developed phase flow method [L. Ying, E.J. Candès, The phase flow method, J. Comput. Phys., to appear], an efficient and accurate technique for constructing phase maps for nonlinear ordinary differential equations on invariant manifolds, which are here the unit tangent bundles of the surfaces under study. We show how to rapidly construct the whole geodesic flow map which then allows computing any geodesic by straightforward local interpolation, an operation with constant complexity. A few numerical experiments complement our study and demonstrate the effectiveness of our approach.

Keywords: Geodesics; Weighted geodesics; Geodesic flow; The phase flow method; Manifolds; Tangent bundles; Charts; Surface parameterization; Spline interpolation

1. Introduction

1.1. The problem

This paper introduces a new method for rapidly computing the geodesic flow on smooth and compact surfaces. Suppose that Q is a smooth surface, then the geodesics of Q obey certain types of differential equations, which may take the form

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t} = F(\boldsymbol{y}), \quad \boldsymbol{y} = \boldsymbol{y}_0; \tag{1.1}$$

$\boldsymbol{y} := (\boldsymbol{x}, \boldsymbol{p})$ where $\boldsymbol{x}$ is a running point on the surface Q, and $\boldsymbol{p}$ is a point in the tangent space so that $\boldsymbol{p}(t)$ is the vector tangent to the geodesic at $\boldsymbol{x}(t)$. Standard methods for solving such ordinary differential equations (ODEs) are based on local ODE integration rules such as the various Runge–Kutta methods. Typically, one chooses a small step size τ and makes repeated use of the local integration rule. If one wishes to integrate the equations up to time

[*] Corresponding author. Tel.: +1 626 395 5760.
E-mail address: lexing@acm.caltech.edu (L. Ying).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.07.032

T, the accuracy is generally of the order of τ^{α} – α is called the order of the local integration rule – and the computational complexity is of the order of T/τ, i.e. proportional to the total number of steps.

In many problems of interest, one would like to trace a large number of geodesics. Expressed differently, one would like to integrate the system (1.1) for many different initial conditions. Examples arising from geometric modeling and computational geometry include mesh parameterization, the segmentation of a surface into several components, shape classification [6], and the interpolation of functions defined on surfaces [18]. The range of applications are of course not limited to computer graphics and spans many areas of science and engineering – especially computational physics and computational mechanics. Consider an arbitrary dynamical system with holonomic constraints. Then it is often the case that the particle trajectories are the geodesics of the smooth manifold defined by these constraints. In the theory of general relativity, the trajectory of a free particle is the spatial projection of a geodesic traced on the curved four-dimensional space–time manifold. In order to understand the underlying dynamics of these physical problems, one often calculates a large number of trajectories which are nothing else than geodesics.

An interesting application which involves computing a large number of geodesics and will be studied in this paper comes from the field of high-frequency wave propagation. Suppose that a smooth body is "illuminated" by an incoming planar wave with an arbitrary direction of propagation. We wish to compute the scattered wavefield. Now the geometric theory of diffraction [8] asserts that straight diffraction rays are emitted from the so-called *creeping rays*. A creeping ray is a geodesic curve on the scatterer which starts from a point on the shadow line and whose initial tangent is parallel to the orientation of the incoming planar wave. To compute the scattered field then, one needs to trace as many geodesics as there are points on the various shadow lines, see Section 2.4 for more details.

In many of these problems, standard methods tracing geodesic curves one by one may be computationally very expensive, and in this paper we introduce a fast and accurate method for computing the whole geodesic flow map over the surface. Our strategy is built upon the phase flow method [19], a newly established method for solving system (1.1) which we review next.

1.2. The phase flow method

Suppose that we are given the system of ordinary differential equation (1.1), where the vector field $\boldsymbol{F}$: $\mathbf{R}^d \to \mathbf{R}^d$ is assumed to be smooth. For a fixed time t, the map g_t: $\mathbf{R}^d \to \mathbf{R}^d$ defined by $g_t(\boldsymbol{y}_0) = \boldsymbol{y}(t, \boldsymbol{y}_0)$ is called the *phase map*, and the family $\{g_t,\ t \in \mathbf{R}\}$ of all phase maps – which forms a one parameter family of diffeomorphisms – is called the *phase flow*. A manifold $M \subset \mathbf{R}^d$ is said to be *invariant* if $g_t(M) \subset M$. In many situations, we are interested in the restriction of the phase flow on an invariant manifold.

We wish to compute the solutions $\boldsymbol{y}(T, \boldsymbol{y}_0)$ of the system (1.1) with many initial conditions $\boldsymbol{y}_0$. Rather than integrating the system one ray at a time, we integrate (1.1) for all the initial conditions at once. The approach consists of two steps:

- First, construct an approximation $\tilde{g}_T$ to the phase map g_T at time T.
- Second, for each $\boldsymbol{y}_0$, the solution $\boldsymbol{y}(T, \boldsymbol{y}_0)$ is calculated by simply evaluating $\tilde{g}_T(\boldsymbol{y}_0)$.

The main difficulty is in the construction of $\tilde{g}_T$. Specifically, we need (1) to construct $\tilde{g}_T$ efficiently and accurately and (2) to represent $\tilde{g}_T$ in a way allowing fast evaluation, which is equally important. This is exactly what the phase flow method achieves.

Algorithm 1 (The phase flow method [19]).

1. *Parameter selection*. Select a grid size $h > 0$, a time step $\tau > 0$, and an integer constant $S \geqslant 1$ such that $B = (T/\tau)^{1/S}$ is an integer power of 2.
2. *Discretization*. Select a uniform or quasi-uniform grid $M_h \subset M$ of size h.
3. *Burn-in*. Compute $\tilde{g}_\tau$. For $\boldsymbol{y}_0 \in M_h$, $\tilde{g}_\tau(\boldsymbol{y}_0)$ is calculated by applying the ODE integrator (single time step of length τ). Then construct a local interpolant based on these sampled values, and for $\boldsymbol{y}_0 \notin M_h$, define $\tilde{g}_\tau(\boldsymbol{y}_0)$ by evaluating the interpolant at $\boldsymbol{y}_0$.