MODELLISTICA E CALCOLO SCIENTIFICO

MODELING AND SCIENTIFIC COMPUTING

MOX–Report No. 02/2012

Optimization of the forcing term for the solution of two point boundary value problems

Arioli, G.

MOX, Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox@mate.polimi.it http://mox.polimi.it

Optimization of the forcing term for the solution of two point boundary value problems

Gianni Arioli

MOX–Modellistica e Calcolo Scientifico
Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano
via Bonardi 9, 20133 Milano, Italy
gianni.arioli@polimi.it

Keywords: Boundary value problem for ODE's, Low thrust connecting orbits, Parameter dependence, Taylor series, Chebyshev series

AMS Subject Classification: 65L10, 70Q05

Abstract

We present a new numerical method for the computation of the forcing term of minimal norm such that a two point boundary value problem admits a solution. The method relies on the following steps. The forcing term is written as a (truncated) Chebyshev series, whose coefficients are free parameters. A technique derived from automatic differentiation is used to solve the initial value problem, so that the final value is obtained as a series of polynomials whose coefficients depend explicitly on (the coefficients of) the forcing term. Then the minimization problem becomes purely algebraic, and can be solved by standard methods of constrained optimization, e.g. with Lagrange multipliers. We provide an application of this algorithm to the restricted three body problem in order to study the planning of low thrust transfer orbits.

1 Introduction

We consider a two point boundary value problem for the non autonomous dynamical system in $\mathbb{R}^N$:

$$\begin{cases} y'(t) & = f(y(t)) + \varphi(t) \\ y(-1) & = y_0 \\ y(1) & = y_1 \,, \end{cases} \tag{1}$$

where $f \in C^1(\mathbb{R}^N, \mathbb{R}^N)$ and $\varphi \in C([-1,1], \mathbb{R}^N)$. We introduce a new numerical method for the computation of φ of minimal norm, among the functions representable with a truncated Chebyshev series, such that (1) admits a solution. In order to evaluate the performance of the algorithm, we apply it to a well known problem, that is the study of low thrust orbits between the Earth and the Moon. We refer to [MTBZ] and references there for a discussion of the astrodynamical problem.

Our new approach is derived from the method introduced in [AG1, AG2] to consider the dependence of parameters of the solution of hyperbolic equations. Such method heavily relies on a computer representation of algebras of functions, in such a way that the computer can use as "numbers" such functions in a straightforward and transparent way, reducing to a minimum the computations to be done explicitly by hand. The basic ideas of this method have been used since a few years in the field of computer assisted proofs, see [AK1, AK2] for some examples, but its potentiality in optimization and control appears to have been overlooked. In Section 2 we describe the approach in a generic setting. In Section 3 we describe the specific planar restricted three body problem that we use to present the method. In Section 4 we present the results that we obtained for the RTBP examples, and we discuss the features of the method.

2 Study of the trajectory depending on a parameter

Consider a generic initial value problem in $\mathbb{R}^n$

$$\begin{cases} y_c'(t) & = f(t, y(t)) + \varphi_c(t) \\ y_c(-1) & = y_0 \,, \end{cases} \tag{2}$$

where f is an analytic function, c is an $n \times N$ real matrix and the forcing term $\varphi_c : [-1,1] \to \mathbb{R}^n$ depends on c as follows:

$$(\varphi_c(t))_i = \sum_{j=0}^{N-1} c_{ij} T_j(t)\,, \tag{3}$$

where $T_j(t)$ is the j-th Chebyshev polynomial and $(\varphi_c(t))_i$ denotes the i-th component of $\varphi_c(t)$. Our first step is to compute explicitly the dependence of $y_c(t)$ on c, for $t \in (-1,1]$.

We choose our favorite method to solve the IVP; in order to focus on the issue of the parameter dependence, we choose at first the simplest method, e.g. an explicit first order Euler scheme with constant time step δ, and we consider the case $n = N = 1$. We show later that it is straightforward to adapt the method to a better scheme, e.g. a Runge-Kutta scheme with variable time step, and to the case $n, N > 1$.

Set $t_k = k\delta$ and $y_k(c) = y_c(t_k)$ so that the Euler scheme reads

$$y_{k+1}(c) = y_k(c) + \delta f(k\delta, y_k) + \delta\varphi_c(k\delta)\,. \tag{4}$$

Now we write

$$y_k(c) = \sum_{j=0}^{+\infty} y_{k,j} U_j(c)\,,$$

where $\{U_j(c)\}_{j\in\mathbb{N}}$ is a basis of some space of functions X of our choice. In this paper we consider the cases where X is the set of analytic functions in a disc (and then $U_j(c) = c^j$) and where X =C([-1,1]) (and then $U_j(c) = \cos(j \arccos c)$), that is the Taylor and Chebyshev expansions, respectively for analytic and continuous functions, but it will be clear how to generalize the method to other expansions. Now we choose a maximum order M and we approximate $y_k(c)$ with the truncated expansion $\sum_{j=0}^{M} y_{k,j} U_j(c)$. Our aim is to compute the coefficients $y_{k,j}$ using an automatic algorithm. Since $y_0(c) = y_0$, then we have $y_{0,0} = y_0$ and $y_{0,j} = 0$, when $j \geq 1$.

The core of the method consists in a generalization of the Taylor Models approach (see [BM1, BM2, BM3, BM4]). For a detailed description of the technique with different expansions see [AG1, AG2]. Here, we only recall the basic ideas. First, we observe that it is straightforward to implement on a computer an arithmetic of Taylor or Chebyshev polynomials of one variable. Using object-oriented programming and operator overloading, it is possible to define a class `Fun` which represents a Taylor or Chebyshev expansion and a set of methods which perform the basic operations and the computation of the elementary functions. More precisely, an object `Fun` is represented as the list of the coefficients. Then it is straightforward to implement a procedure that, given a scalar α and two objects `Fun` T_1, T_2, computes the object `Fun` corresponding to $\alpha(T_1 * T_2)$, where $*$ is either the addition or the multiplication. Then, given a polynomial $p(x)$, it is possible to compute the object `Fun` corresponding to $p(T_1)$, and since all analytic functions $f(x)$ can be approximated with polynomials, it is also possible to compute the object `Fun` that better approximates $f(T_1)$. A more algebraic oriented way of expressing this idea is the following: let X be the space of functions spanned by $\{U_j(c)\}_{j\in\{0,\dots,M\}}$ and let $T : X \to \mathbb{R}^{M+1}$ be the (invertible) map that extracts the coefficients. Our approach consists in lifting the basic arithmetics of X to $\mathbb{R}^{M+1}$. More precisely, we compute operators

$$\{\oplus, \otimes\} : \mathbb{R}^{M+1} \times \mathbb{R}^{M+1} \to \mathbb{R}^{M+1}$$

and

$$\odot : \mathbb{R} \times \mathbb{R}^{M+1} \to \mathbb{R}^{M+1}$$

such that $T(f+g) = T(f) \oplus T(g)$, $T(fg) = T(f) \otimes T(g)$ and $T(\alpha g) = \alpha \odot T(g)$. Then, we make extensive use of object-oriented programming and operator overloading in order to make this lifting completely transparent to the user.

The extension of this method to the case $n > 1$ is straightforward. The case $N > 1$ requires a little more work, since we have to consider multivariable Taylor or Chebyshev expansions. Clearly, we still can represent a multivariable expansion by a list of coefficients, and we can still implement the sum of two of such objects componentwise. All is left to do is to implement the multiplication. This is not a trivial task, but a very efficient algorithm for the case of the Taylor expansion has been introduced in [BM4], and it is not too hard to extend it to the Chebyshev expansion. We refer to our programs for details.

With the class `Fun` in place, we can use a standard implementation of the algorithm (4), observing that, given y_k, the methods of the class will take care of the computation of $f(k\delta, y_k)$, while $\varphi_c(k\delta)$ is explicitly given by (3). Extending this approach to better integration schemes, say Runge-Kutta or multistep, is straightforward: it suffices to implement the chosen algorithm using objects `Fun` and their methods.

3 Low thrust orbits in the planar RTBP

In order to model the dynamics of a spaceship travelling from an orbit around the Earth to an orbit around the Moon, we consider the forced planar RTBP, that is the system:

$$\begin{aligned} \ddot{x} + 2\dot{y} &= \Omega_x + f(t)\tfrac{\dot{x}}{\sqrt{\dot{x}^2+\dot{y}^2}} \\ \ddot{y} - 2\dot{x} &= \Omega_y + f(t)\tfrac{\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}}\,, \end{aligned} \tag{5}$$

where

$$\Omega(x,y) = \frac{x^2}{2} + \frac{y^2}{2} + \frac{1-\mu}{\sqrt{(x+\mu)^2+y^2}} + \frac{\mu}{\sqrt{(x+\mu-1)^2+y^2}} + C\,.$$

We choose $\mu \simeq 0.0121506683$, that is the reduced mass of the Earth-Moon system and $C = -1.600172454916536$, so that $\Omega(L1) = 0$. The forcing is given by a scalar function $f(t)$ to be determined times a unit vector directed as the velocity of the spaceship, so we assume that the thrust is parallel to the velocity. We show how to apply the method described in the previous section to the problem of optimizing the travel from an orbit 167km above the Earth to an orbit 100km above the Moon. We choose a connecting orbit passing close to $L1$. Note that the Jacobi function

$$J(x,y,\dot{x},\dot{y}) = \dot{x}^2 + \dot{y}^2 - 2\Omega(x,y)$$

is roughly -60 on an orbit 167km above the Earth and -1.6 on an orbit 100km above the Moon, and in order to "cross" $L1$ it is necessary to raise it above 0. We build the connecting orbit as follows. We start from the Lyapunov orbit around L1 with $J = 0.1$ and approximate periodic orbits around the Moon and the Earth. Then we compute approximate connections from the Lyapunov orbit to the orbit around the Earth (backwards in time) and to the orbit around the Moon (forward in time). This is useful, if not necessary, because of the high instability of the Lyapunov orbit: it is much harder to aim to the Lyapunov orbit than to a stable orbit around a primary. These orbits are obtained by applying a constant low thrust (that is, $f(t) = c$, with small c), chosen but simple trial and error. Once these rough orbits are computed, we apply the techique described above to correct them in order to satisfy some required boundary conditions and at the same time to minimize the thrust. More precisely, we write (5) as a system of 4 first order equations by setting $v = \dot{x}$ and $w = \dot{y}$, we write the modulus of the forcing term as

$$f(t) = c + \sum_{j=0}^{N-1} c_j T_j(t)\,,$$

we choose some initial value on the trajectory at the arbitrary time $t = 0$ and we solve the equation with a fourth order Runge-Kutta scheme up to some time $t = T$, using objects `Fun` as scalars, the coefficients c_j being the independent variables of the objects `Fun`. Referring to Section 2, we have $n = 4$ (the dimension of the system), we choose $N = 4$ and $M = 10$. We rescale the time in the interval $[-1, 1]$ and apply the method described in Section 2 to obtain $(x(T), y(T), v(T), w(T))$ as a function of $\{c_j\}$, expressed either as a Taylor series or a Chebyshev series. Once that expression is known explicitly, it is easy to apply a minimization method to solve some problem, e.g. find $\{c_j\}$ such that $(x(T), y(T), v(T), w(T))$ is the same but $\|f\|$ is minimal. Or find $\{c_j\}$ such that the distance of the spaceship to the Earth or the Moon is assigned, e.g. $(x(T) - \mu)^2 + y(T) = R^2$ together with the final velocity $(v(T))^2 + w(T)^2 = V^2$ and $\|f\|$ is minimal. Or possibly, we can add another constraint, such as the final velocity to be orthogonal to the line connecting the satellite to a primary. We describe in more detail the second of these examples. Since we have a objects `Fun` that represent $(x(T), y(T), v(T), w(T))$ as a function of $\{c_j\}$, then we can use again the (automatic) algebra `Fun` to compute two objects `Fun`, that is two explicit Taylor or Chebyshev polynomials $a(c_1, \ldots, c_N)$ and $b(c_1, \ldots, c_N)$, representing $(x(T)-\mu)^2+y(T)$ and $(v(T))^2+w(T)^2$. So, we reduced the problem to the following: find $\{c_j\}$ of minimal norm under the constraint $a(c_1, \ldots, c_N) = R^2$ and $b(c_1, \ldots, c_N) = V^2$. This can be achieved easily e.g. with Lagrange multipliers.

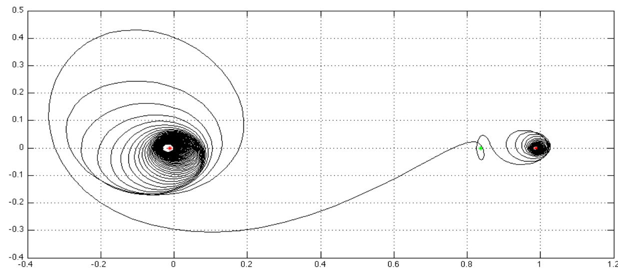

Figure 1: A complete orbit

R	V	ε_T	$\|f_T\|$	ε_C	$\|f_C\|$
0.4	0.9	$1.03 \cdot 10^{-7}$	0.3012	$1.27 \cdot 10^{-7}$	0.3012
0.4	0.85	$2.32 \cdot 10^{-4}$	0.4215	$3.45 \cdot 10^{-4}$	0.4235
0.45	0.9	NA	NA	$7.66 \cdot 10^{-5}$	0.2104
0.43	0.9	$6.58 \cdot 10^{-10}$	0.1540	$1.39 \cdot 10^{-7}$	0.1540

Table 1: Result enforcing R and V

4 Results

Figure 1 represents an example of an orbit computed by using the technique. Tables 1 and 2 collect some results obtained for a trajectory backward in time starting close to a Lyapunov orbit around L1, more precisely at the point $(x(0), y(0), v(0), w(0)) = (0.73356888, -0.017228233, 0.25064405, 0.17220602)$ and ending close to an orbit around the Earth at time $T = -1.7$. The trajectory with a constant thrust $c = 0.2$ ends at $R = \sqrt{(x(T) - \mu)^2 + y(T)} = 0.421852$ and $V = \sqrt{(v(T))^2 + w(T)^2} = 0.876662$. For both tables we computed the thrust of minimal norm necessary to end the trajectory the the values given in the columns R and V, but the results of Table 2 represents the solution with the additional constraint that the final velocity is orthogonal to the segment from the spaceship to the Earth, corresponding to an approximate circular orbit. In order to test the accuracy of the computation, when the optimization is done we repeat the compuation of the orbit with standard floating point numbers, using the explicit expression of the forcing term, and we use the result of such computation to verify that the constraint

R	V	ε_T	$\varepsilon_T^\perp$	$\|f_T\|$	ε_C	$\varepsilon_C^\perp$	$\|f_C\|$
0.42	0.87	$3.04 \cdot 10^{-4}$	$8.48 \cdot 10^{-5}$	0.3627	$1.53 \cdot 10^{-4}$	$8.26 \cdot 10^{-7}$	0.3653
0.42	0.9	$3.18 \cdot 10^{-6}$	$6.00 \cdot 10^{-7}$	0.2540	$4.02 \cdot 10^{-6}$	$9.75 \cdot 10^{-6}$	0.2541
0.44	0.82	$4.18 \cdot 10^{-9}$	$3.37 \cdot 10^{-10}$	0.2044	$2.48 \cdot 10^{-8}$	$2.34 \cdot 10^{-8}$	0.2044
0.46	0.82	$7.32 \cdot 10^{-8}$	$1.43 \cdot 10^{-7}$	0.1439	$3.77 \cdot 10^{-6}$	$2.22 \cdot 10^{-6}$	0.1439

Table 2: Result enforcing R, V and tangential trajectory

are satisfied. The values ε_T and ε_C represent the mean square of the relative errors on R and V, obtained with the Taylor and the Chebyshev computation respectively, while $\varepsilon_T^\perp$ and $\varepsilon_C^\perp$ represent $|(x(T)-\mu)v(T)+y(T)w(T)|$. The columns $\|f_T\|$ and $\|f_C\|$ represent the norm of the forcing terms. It is clear from the tables that the method can achieve very accurate results, and it is also clear that the forcing term obtained with the Taylor and the Chebyshev computation is (almost) the same.

We discuss and compare the features of the two expansion that we tested: Taylor and Chebyshev. Generally speaking (see [AG1, AG2]), the main advantages of the Taylor expansion consist in three features: the algorithms are simpler and faster, they provide directly the derivatives with respect to the parameters and furthermore, and, since one does not need to know the radius of convergence a priori, one can apply them and compute a posteriori the interval of validity of the computation. Taylor expansion have also two main disadvantages: they do not provide uniform errors in the interval where the computation is performed and the radius of convergence may be bounded by poles in the complex plane. The Chebyshev expansion has opposite features: the error is uniform in the interval and the region of convergence is an ellipse as thin as necessary, therefore there are no problems caused by poles with nontrivial imaginary parts. On the other hand, the algorithm for the multiplication is much more complicated and much slower, it does not provide any direct computation of the derivatives and finally, since the Chebyshev polynomials are defined in $[-1,1]$, one needs to choose a priori (via a suitable translation/rescaling) the interval where the parameter ranges, finding out only a posteriori if the approximation is acceptable. In the application considered in [AG1, AG2], the Chebyshev expansion was a clear winner, due to the fact that, when considering the numerical approximation of shock waves and when trying to improve the resolution, the problem of poles becomes the main issue. Here, our judgment is the opposite: since the solutions of the ode's that we are studying are analytic functions, the issue of poles limiting the radius of convergence of Taylor series becomes negligible, while the disadvantages of the Chebyshev expansion, in particular the slower algorithms and the fact that the domain has to be chosen a priori, become very relevant. Additionally, we do not have evidence of a better performance in terms of accuracy. It is easy to find examples when either expansion performs better than the other one, but on average the accuracy is similar.

References

[AG1] G. Arioli, M. Gamba, *An algorithm for the study of parameter dependence for hyperbolic systems*, MOX-Report 05/2011, `http://mox.polimi.it/it/progetti/pubblicazioni/quaderni/05-2011.pdf`

[AG2] G. Arioli, M. Gamba, *Automatic computation of Chebyshev polynomials for the study of parameter dependence for hyperbolic systems*, MOX-Report 07/2011, `http://mox.polimi.it/it/progetti/pubblicazioni/quaderni/07-2011.pdf`

[AK1] G. Arioli, H. Koch, *Computer-assisted methods for the study of stationary solutions in dissipative systems, applied to the Kuramoto-Sivashinski equation*, Arch. Rational Mech. An. 197 (2010) 1033

[AK2] G. Arioli, H. Koch, *Integration of dissipative PDEs: a case study*, SIAM J. of Appl. Dyn. Syst., 9 (2010) 1119-1133

[BM1] K. Makino, M. Berz, *Higher order verified inclusions of multidimensional systems by Taylor models. Proceedings of the Third World Congress of Nonlinear Analysts, Part 5* (Catania, 2000). Nonlinear Anal. 47 (2001), no. 5, 3503–3514

[BM2] M. Berz, K. Makino, *Taylor models and other validated functional inclusion methods.* Int. J. Pure Appl. Math. 6 (2003), no. 3, 239–316

[BM3] M. Berz, K. Makino, *Taylor models and other validated functional inclusion methods.* Int. J. Pure Appl. Math. 4 (2003), no. 4, 379–456.

[BM4] M. Berz, K. Makino, *Higher Order Multivariate Automatic Differentiation, Validated Computation of Remainder Bounds*, Transactions on Mathematics 3(1) (2004) 37-44

[MTBZ] G. Mingotti, F. Topputo, and F. Bernelli-Zazzera, *Numerical Methods to Design Low-Energy, Low-Thrust Sun-Perturbed Transfers to the Moon*, 49th Israel Annual Conference on Aerospace Sciences, Tel Aviv-Haifa, Israel, 4-5 March, 2009

MOX Technical Reports, last issues

Dipartimento di Matematica "F. Brioschi",
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

1/2012 Lassila, T.; Manzoni, A.; Quarteroni, A.; Rozza, G.
A reduced computational and geometrical framework for inverse problems in haemodynamics

47/2011 Antonietti, P.F.; Borz, A.; Verani, M.
Multigrid shape optimization governed by elliptic PDEs

46/2011 Migliorati, G.; Nobile, F.; von Schwerin, E.; Tempone, R.
Analysis of the discrete L2 projection on polynomial spaces with random evaluations

45/2011 Canuto, C.; Nochetto, R. H.; Verani, M.
Adaptive Fourier-Galerkin Methods

44/2011 Fumagalli,A.; Scotti,A.
Numerical modelling of multiphase subsurface flow in the presence of fractures

43/2011 L. Formaggia, A. Quarteroni, C. Vergara
On the physical consistency of the coupling between three-dimensional compliant and one-dimensional problems in haemodynamics

42/2011 Antonietti, P.F.; Quarteroni, A.
Numerical performance of discontinuous and stabilized continuous Galerkin methods or convection-diffusion problems

41/2011 Burman, E.; Zunino, P.;
Numerical Approximation of Large Contrast Problems with the Unfitted Nitsche Method

40/2011 D Angelo, C.; Zunino, P.; Porpora, A.; Morlacchi, S.; Migliavacca, F.
Model reduction strategies enable computational analysis of controlled drug release from cardiovascular stents

39/2011 Antonietti, P.F.; Ayuso de Dios, B.; Brenner, S.C.; Sung, L.-Y.
Schwarz methods for a preconditioned WOPSIP method for elliptic problems