Motion Planning for Multi-Agent Systems Using Gevrey Trajectories Based on Burgers’ Viscous Equation
Etienne Servais, Brigitte Andrea-Novel, Hugues Mounier

To cite this version:
Etienne Servais, Brigitte Andrea-Novel, Hugues Mounier. Motion Planning for Multi-Agent Systems Using Gevrey Trajectories Based on Burgers’ Viscous Equation. 19th IFAC World Congress on International Federation of Automatic Control (IFAC 2014), IFAC, Aug 2014, Cape Town, South Africa. hal-01100841

HAL Id: hal-01100841
https://hal-supelec.archives-ouvertes.fr/hal-01100841
Submitted on 7 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Motion Planning for Multi-Agent Systems
Using Gevrey Trajectories
Based on Burgers’ Viscous Equation

Étienne Servais *,** Brigitte d’Andréa-Novel *
Hugues Mounier **

* Mines-ParisTech, Centre de Robotique, 60 bld. Saint-Michel, 75272 Paris Cedex 06, France, brigitte.dandrea-novel@mines-paristech.fr
** L2S-CNRS-Supélec, 3 rue Joliot-Curie, 91192 Gif sur Yvette, France, servais@lss.supelec.fr, mounier@lss.supelec.fr

Abstract: The differential flatness of the one-dimensional heat equation controlled at each boundary is used to propose, through the Hopf-Cole transform, a finite-time motion planning for multi-agent systems. The proposed paths, which are solutions of Burgers’ viscous equation, are smooth while non analytical in initial and final equilibria. The paths may respect various geometrical constraints allowing them to be used for different purposes.

Keywords: Nonlinear partial differential equations, flatness, multi-agent systems, autonomous mobile robots, robot navigation, trajectory planning

1. INTRODUCTION

Since the introduction of computer animations back in the late 80’ and the need of reliably animate swarm of birds, herds and flocks (see Reynolds (1987)), several ways of animating and moving multi-agent systems have been proposed. In the past years, this field has been thoroughly studied due to the appearance of new multi-agent systems: unmanned autonomous vehicles (so called UAV’s) such as quadrotors and mobile robots.

The different methods can be either distributed or centralized depending on the knowledge of a single agent of the state of the global system. Systems needing informations about the states of all the agents are called to be centralized whereas systems where agents can be moved knowing only informations on their nearest neighbors are called decentralized. The main goal of the techniques are collision avoidance (with other agents or obstacles), cohesion of the system and the final stabilization on a specific formation or trajectory.

Various methods either distributed or centralized have been proposed using potential-like solutions (Olafi-Saber and Murray (2002)), geometric approaches (Bruñon Arranz et al. (2011)) or based on graph theory (see Ferrari-Trecate et al. (2006); Fax and Murray (2002)) and receding horizons methods (Murray and Dunbar (2004)).

Our method, based on previous works by Frihauf and Krstic (2011); Meurer and Krstić (2011) considers the multi-agent system as a continuum of fully actuated agents and steers it using Partial Differential Equations (PDE’s). It accomplishes full stabilization of the system and collision avoidance using the physical model described by the chosen PDE. The problem of PDE control is a wide open subject and only few problems are yet solved, using for example either differential flatness (see Laroche et al. (2000); Meurer and Krstić (2011)) or the newly introduced backstepping method by Krstić and Smyshlyaev (2008).

In this paper, we use the Hopf-Cole transformation by Hopf (1950) and Cole et al. (1951) to transform the viscous Burgers’ equation \( u_t + uu_x = \mu u_{xx} \) into the heat equation, which was already proven to be in some case differentially flat (see for instance Laroche et al. (2000)).

The controllability of Burgers’ viscous equation with two boundary controls has been recently broadly studied and various results have been obtained. On the one hand Guerrero and Imanuvilov (2007) proved that both the global exact null controllability for small time and the exact controllability for large time do not hold. On the other hand, Glass and Guerrero (2007) gives a proof of the global exact null controllability to nonzero states and Coron (2007) has shown using the Hopf-Cole transformation a controllability result large constant states.

In our case we use two inputs on either side of the system. These inputs describe the trajectories of two leaders of the formation. We use the Hopf-Cole transformation in section 2 and prove some flatness properties of the heat equation. We then propose in section 3 a Gevrey (1918) solution respecting initial and final formations constraints both in the real space (in which we solve Burgers’ equation) and in the Hopf-Cole space (in which we solve the heat equation). When using only the heat equation, trajectory checkpoints might be used by the two boundary agents. We evaluate our controls through numerical computations and propose a formal expression of the solution. We then test our controls for the heat equation against formal solution and constraints. Some examples for the motion planning based on the Hopf-Cole equation are finally given.
2. FLATNESS PROPERTIES

The steering of PDE’s is a problem widely studied in industry. It allows motion planning, i.e. the ability to move a system such as a chemical reactor in Fliess et al. (1998) or a crystal in Rudolph et al. (2004) from a known state to another state on a determined path. Burgers’ equation is a standard PDE of the fluid mechanics. It is sometimes thought as the 1D analogue of Navier-Stokes 2D equations. As one of the simplest non-linear PDE, it is sometimes thought as the 1D analogue of Navier-Stokes equation. It is a standard PDE of the fluid mechanics. It is

\[ \phi \]

is Gevrey of order 1. Hence \( \phi \) has same order as \( u \). Since \( \phi \) is Gevrey of order greater than 1. Since \( u = \phi \) is Gevrey and has same order as \( \phi \) and as thus, as \( \phi \).

Let us recall the following result due to Holmgren (1908):

**Proposition 2.** The heat equation with Cauchy boundary control in 0 is flat.

**Proof.** Using the Laplace transform on the heat equation, we get \( \tilde{s}_0(s,x) = \mu^2 \hat{\phi}(s,x) \). Solving this x ODE follows in the system:

\[
\begin{cases}
\hat{\phi}(s,x) = \hat{C}_x \hat{\lambda}_1(s) + \hat{S}_x \hat{\lambda}_2(s) \\
\hat{C}_x = \cosh(x \sqrt{s/\mu}) \\
\hat{S}_x = \sqrt{\mu/2} \sinh(x \sqrt{s/\mu})
\end{cases}
\]

Using the Laplace inverse transform, we get \( \phi(x,t) = C_x \lambda_1(t) + S_x \lambda_2(t) \) where \( C_x = \cosh(x \sqrt{\delta_1/\mu}) \) and \( S_x = \sqrt{\mu/2} \sinh(x \sqrt{\delta_1/\mu}) \) are differential operators of infinite order (see for example Van Tran and Dinh (1994)).

It is obvious that \( \partial_t C_x = \partial_t (S_x) / \mu \) and \( \partial_t S_x = C_x \). Furthermore \( C_0 = 1 \) and \( S_0 = 0 \). Deriving \( \phi \) according to \( x \) leads to \( \phi_x(x,t) = \delta_t (S_x \lambda_1(t) + C_x \lambda_2(t)) \). We then get \( \lambda_1(t) = \phi(0,t) \) and \( \lambda_2(t) = \phi_x(0,t) \).

Thus \( \phi_0, \phi_x,0 \) is a flat output of the system, where \( \phi_0 \) (resp. \( \phi_x,0 \)) stands for \( \phi(0,t) \) (resp. \( \phi_x(0,t) \)).

Instead of having two controls on one side, we use the following proposition to control the equation on both sides:

**Proposition 3.** The heat equation with controls in \( x = 0 \) and \( x = 1 \) is flat.

**Proof.** For convenience we will set \( \mu = 1 \). Based on the result of the previous proposition, we may write the solution to the heat equation as:

\[ \phi(x,t) = C_x \phi_0(t) + S_x \phi_x,0(t) \]

we thus have:

\[ \phi(1,t) = C_1 \phi_0(t) + S_1 \phi_x,0(t) \]

Let us assume we may invert the \( S_1 \) operator, equation (2) turns into:

\[ \phi_x,0(t) = (S_1)^{-1} (S_1 \phi_0(t) - C_1 \phi(t)) \]

This inversion is formal and no proof of its convergence is given. Injecting equation (3) in (1), we get:

\[ \phi(x,t) = (C_x - T_x C_1) \phi_0(t) + T_x \phi(t) \]

where \( T_x \) stands for \( S_x (S_1)^{-1} \). Thus, assuming the \( T_x \) operator has a meaning and has a sufficient regularity regarding the chosen controls, we get the result.

**Remark:** Since \( T_1 = 1, T_0 = 0 \) and \( C_0 = 1 \), we may rewrite equation (4) in a symmetric way:

\[ \phi(x,t) = (C_x T_1 - T_x C_1) \phi_0(t) - (C_x T_0 - T_x C_0) \phi(t) \]

Using the Taylor expansion of csh (Abramowitz and Stegun, 1965, p. 85), we get a formal expression of \( (S_1)^{-1} \):

\[ (S_1)^{-1} = \sum_{k \geq 0} u_k \frac{d^k}{dt^k} \]

where \( u_0 = 1 \) and \( u_k = (1 - 2^{2k-1}) B_{2k} / (2k)! \) where \( B_k \) is the \( k \)-th Bernoulli number. Thus, assuming expression (3) has a meaning, we get a formal expression of the \( T_x \) operator:

\[ T_x = \sum_{k \geq 0} \left( \sum_{l=0}^{k} x^{2l+1} u_{k-l} (2l+1)! \right) \frac{d^k}{dt^k} \]

We show in the following a meaning of this inversion.

**Remark:** Based on (Abramowitz and Stegun, 1965, p. 807), \( u_k = (-1)^{k+1} \frac{\zeta(2k)}{(2k)!} \) where \( \zeta \) is the Riemann zeta function. Thus we get the asymptotic approximation \( u_k \sim 2(-1)^k / \pi^{2k} \).

3. SOLUTION COMPUTATION

3.1 Control building

We consider the motion of a formation of \( N \) fully actuated UAV’s between the times \( t_0 = 0 \) and \( t_1 = 1 \). We describe the initial and final formation with the polynomials:
\[ \begin{aligned}
\phi_0(t) &= \sum_{i=0}^{N-1} p_{2i} \frac{t^i}{i!} + \sum_{i=0}^{N-1} r_{2i} \frac{(t-1)^i}{i!} \\
\phi_1(t) &= \sum_{i=0}^{N-1} q_{2i+1} \frac{t^i}{i!} + \sum_{i=0}^{N-1} s_{2i+1} \frac{(t-1)^i}{i!}
\end{aligned} \]

Both functions interpolate their respective formation. Furthermore, we want in some cases the left-most and right-most agents going through a given set of points.

A way to find appropriate \((p_i, q_i, r_i, s_i)\) regarding to the trajectories and formations constraints will be explained in appendix A.

Since \(f_0(x) = \phi(x, t = 0)\) and \(f_1(x) = \phi(x, t = 1)\), by identification of the coefficients of the left- and right-hand sides in (1) and (6), we have the following conditions:

\[ \begin{aligned}
&\phi_0(0) = p_{2i} \\
&\phi_0(1) = q_{2i+1} \\
&\phi_x(0) = p_{2i+1} \\
&\phi_x(1) = q_{2i+1}
\end{aligned} \quad \text{for } 0 \leq i < N \\
\begin{aligned}
&\phi_0(0) = 0 \\
&\phi_0(1) = 0 \\
&\phi_x(0) = 0 \\
&\phi_x(1) = 0
\end{aligned} \quad \text{for } i \geq N
\]

(7)

Based on the work in Laroche et al. (2000), we now propose trajectories for both left-most and right-most agent as:

\[ \begin{aligned}
\phi_0(t) &= \sum_{i=0}^{N-1} \left( p_{2i} \frac{t^i}{i!} (1 - \Phi(t)) + q_{2i} \frac{(t-1)^i}{i!} \Phi(t) \right) \\
\phi_1(t) &= \sum_{i=0}^{N-1} \left( r_{2i+1} \frac{t^i}{i!} (1 - \Phi(t)) + s_{2i+1} \frac{(t-1)^i}{i!} \Phi(t) \right)
\end{aligned} \]

where \(\Phi(t)\) is a Gevrey function non-analytical (thus of order \(\alpha > 1\)) in \(t = 0\) and \(t = 1\) with \(\Phi(0) = 0\) and \(\Phi(1) = 1\). Since scaling, integration, addition, multiplication and composition of Gevrey functions of order \(\alpha > 1\) is of order \(\alpha\), the latter proposed trajectories are Gevrey of order \(\alpha\).

We will write this as:

\[ \begin{aligned}
\phi_0(t) &= P_0(t)(1 - \Phi(t)) + Q_0(t)\Phi(t) \\
\phi_1(t) &= R_0(t)(1 - \Phi(t)) + S_0(t)\Phi(t)
\end{aligned} \]

Considering the functions for \(t \leq 0\) and \(t \geq 1\), we easily see that we achieve a \(C^\infty\) transition between two polynomials. Furthermore, the proposed solution verifies the conditions (7) in \(\phi_0(t)\). It verifies similar conditions for \(\phi_1(t)\) with \(r_{2i+1}\) and \(s_{2i+1}\). We will show in the following it also verifies the conditions in \(\phi_x,0(t)\) and \(\phi_x,1(t)\).

In the following, we will use the \(\Phi_\gamma(t)\) used in Laroche et al. (2000), where \(\gamma > 0\):

\[ \Phi_\gamma(t) = \begin{cases} 0 & \text{if } t \leq 0 \\ 1 & \text{if } t \geq 1 \\ \int_0^1 \phi_\gamma(\tau) d\tau / \int_0^1 \phi_\gamma(\tau) d\tau & \text{if } t \in [0; 1]
\end{cases} \]

where:

\[ \phi_\gamma(t) = \begin{cases} 0 & \text{if } t \leq 0 \text{ or } t \geq 1 \\ \exp(-1/(1-t)^\gamma) & \text{if } t \in [0; 1]
\end{cases} \]

The \(\phi_\gamma\) function (and thus the \(\Phi_\gamma\)) is Gevrey of order \(1 + \frac{1}{\gamma}\) and non analytic in \(t = 0\) and \(t = 1\).

### 3.2 Control validation

It is straightforward to solve the heat equation numerically with given controls \(\phi_0\) and \(\phi_1\) and initial condition \(f_0\).

On the one hand, as shown on figure 1 and 2, the bigger \(\gamma\), the more “bumpy” the control is. In the same time, the bigger \(\gamma\), the less the control follows the checkpoints. Since controls are constructed to follow exactly the checkpoints, it is doubtlessly a matter of computational accuracy in the inversion of matrices in appendix A.

On the other hand, it also appears that the value of \(\gamma\) is of great meaning in order to conform to final formations. While \(\gamma\) decreases toward 0, the controls are more and more Gevrey divergent. While \(\gamma\) increases, the more the \(\Phi_\gamma\) function is a step at \(t = 1/2\) and thus the more the computational accuracy decreases. As shown by figure 3, accuracy increases toward \(\gamma = 1.685\) and decreases afterwards. For further applications, we use \(\gamma = 1.5\), which allows reasonable respect of checkpoints and final formation.

A numerical solution to the heat equation with our controls is given in figure 4. Trajectories are given by \(x\) constant lines.
Fig. 3. Error between numerical resolution and model at \( t = 1 \)

\[ f(x) = \frac{f_1(x) - \phi(x,1)}{f_1(x)} \]

\( \gamma = 1.5 \)
\( \gamma = 1.625 \)
\( \gamma = 1.75 \)
\( \gamma = 1.875 \)
\( \gamma = 1.9 \)

Fig. 4. Numerical resolution \( (\gamma = 1.5) \)

### 3.3 Formal computation

The advantage of the flatness-based methods, is that they allow us to compute formal expressions of the solution. Noticing that \( \Phi^{(2)}(t) = \frac{\gamma^{(1-2t)(t)}}{(1-t)^t} \Phi^{(1)}(t) \), we may write:

\[
\begin{align*}
\phi_0^{(k)}(t) &= P_k(t)(1 - \Phi(t)) + Q_k(t)\Phi(t) + A_k(t)\Phi'(t) \\
\phi_1^{(k)}(t) &= R_k(t)(1 - \Phi(t)) + S_k(t)\Phi(t) + B_k(t)\Phi'(t)
\end{align*}
\]

with the following recurrences:

\[
\begin{align*}
P_k(t) &= P_0^{(k)}(t) = \sum_{l=0}^{N-1-k} p_{l+k} l! \frac{t^l}{(l-1)!} \\
Q_k(t) &= Q_0^{(k)}(t) = \sum_{l=0}^{N-1-k} q_{l+k} l! \frac{t^l}{(l-1)!} \\
A_0(t) &= 0 \\
A_k(t) &= Q_{k-1}(t) - P_{k-1}(t) + A_{k-1}(t) + \frac{\gamma^{(1-2t)(t)}}{(1-t)^t} A_{k-1}(t)
\end{align*}
\]

(resp. with \( R_k, S_k, B_k \))

where \( P_k(t), Q_k(t) \) (resp. \( R_k, S_k \)) are polynomials of order \( N - k - 1 \), thus vanishing for \( k \geq N \). Using (5) we may show the operator acting on \( \phi_0 \) to be:

\[
C_x - T_x C_1 = \sum_{k=0}^{\infty} \frac{x^{2k}}{(2k)!} - \sum_{l=0}^{k} v_{k-l} x^{2l+1} \frac{(2l+1)!}{(2l+1)!} d^k dt
\]

where \( v_k = 2^{2n} B_{2n}/(2n)! \). We may then use this to write:

\[
\phi(x, t) = D(x, t)(1 - \Phi(t)) + E(x, t)\Phi(t) + F(x, t)\Phi'(t)
\]

After a few computation, it appears that we may write the solution as:

\[
D(x, t) = \sum_{l=0}^{N-1} \frac{t^l}{l!} f_0^{(2l)}(x), \ E(x, t) = \sum_{l=0}^{N-1} \frac{(t - 1)^l}{l!} j_1^{(2l)}(x)
\]

In fact, we may write \( A_k(t) \) as:

\[
A_k(t) = k(Q_{k-1}(t) - P_{k-1}(t)) + \sum_{l=1}^{k} \frac{A_k(t)}{(l(t-1)+1)^{k+\gamma}}
\]

Where \( A_k^1 \) is a polynomial of order at most \( N + i + k \). It is straightforward to check the expression for \( k = 0 \) and \( k = 1 \) (where all \( A_k^1 \) are zero). The recurrence is verified using relations (10). We get a similar expression for \( B_k \).

Using (4), (5), (11) and (9), we may write \( F \) as:

\[
F(x, t) = \sum_{k=0}^{\infty} \left( \frac{x^{2k}}{(2k)!} - \sum_{l=0}^{k} v_{k-l} x^{2l+1} \frac{(2l+1)!}{(2l+1)!} \right) A_k(t) + \left( \sum_{l=0}^{k} v_{k-l} x^{2l+1} \frac{(2l+1)!}{(2l+1)!} \right) B_k(t)
\]

In future work, we will show that \( \Phi_1^{(1)}(t) F(x, t) \) vanishes on the boundaries of the unity square and is convergent inside this domain.

### 3.4 Solution validation

Using expression (3.3) and previous numerical results, one can find an estimate of the \( \Phi_1^{(1)}(t) F(x, t) \) term. This is done and plotted in figure 5.

This numerical result shows that the \( |\Phi_1^{(1)}(t) F(x, t)| \) vanishes on the boundary of the unity square but that it is clearly not negligible, even for other value of \( \gamma \) in the formal expression of the solution.

Using appropriate coefficients as explained in A, one can find a solution to the Burgers equation. The major issue in finding appropriate controls is to ensure they do not cross the zero-axis. As long as this condition is verified, one can find trajectories between constant states (on figure 6 the agents share a common abscissa on one dimension but not necessarily on other, thus they are aligned).

Finally, one can combine several solutions in order to obtain trajectories in a multi-dimensional space. Each spatial component of the trajectory will be Gevrey and thus the trajectory itself will be Gevrey. A sample trajectory for a two-agents system is drawn for a 2D problem in figure 7.
Simulations are given while experiments using quadrotors are foreseen.

REFERENCES


Hopf, E. (1950). The partial differential equation \( u_t + uu_x = \mu u_{xx} \). *Communications on Pure and Applied Mathematics*, 3(3), 201–230.


Appendix A

In this appendix, we show how to set the coefficients $p_i$, $q_i$, $r_i$, and $s_i$ so that our system achieves the desired tasks. In a first part we propose a way to get trajectories adapted for the heat equation. In a second part we propose a way to get appropriate coefficients to solve Burgers’ equation.

We write $r_0$ (resp. $s_0$, $p_0$, $p_1$, $q_0$, and $q_1$) the vector $(r_2)$ (resp. $(s_2)$, $(p_{2+i})$, $(q_{2+i})$, $F_0$ and $F_1$) the upper-triangular matrices of generic terms $1/(2(j-i))!$ and $1/(2(j-i)+1)!$. Based on (6) we get the relations:

$$r_0 = F_0 p_0 + F_1 p_1, s_0 = F_0 q_0 + F_1 q_1$$ (A.1)

Let the respective index of every agent be $i/(N-1), 0 \leq i < N$. Let $d^a = (d^a_i)$ (resp. $d^f = (d^f_i)$) be the initial (resp. final) formation, so that $f_0(i/(N-1)) = d^a_i$ (resp. $f_1(i/(N-1)) = d^f_i$). We write $A$ the invertible Vandermonde matrix of generic term $((i/(N-1))^{(2i)})_{0 \leq i < L}$ and $J_0$ (resp. $J_1$) the diagonal matrix of generic term $(1/(2i)!)$ (resp. $1/(2i+1)!$) and $H$ the diagonal non-invertible matrix of generic term $(i/(N-1))$. We have:

$$A J_0 p_0 + H A J_1 p_1 = d^a, A J_0 q_0 + H A J_1 q_1 = d^f$$

Writing $Q = J_0^{-1} A^{-1}$, which is invertible and $R = H A J_1$ which is not, we get:

$$p_0 = Q (d^a - R p_1), q_0 = Q (d^f - R q_1)$$ (A.2)

We also want the left trajectory (resp. right) going through points $(e^{i})_{0 \leq i < N}$ (resp. $(e^{i})_{0 \leq i < N}$) at time $(i^a)_{0 \leq i < N}$ (resp. $(i^f)_{0 \leq i < N}$). Since the positions of the left- and right-most agents at time $t = 0$ and $t = 1$ are already known, we take $0 < i^a, i^f < 1$. Let $i^a = i^f = (i+1)/(N+1)$. Let $P$ be the anti-diagonal matrix of generic term $1$ with $PP = I$. Let $F$ be the diagonal matrix of generic term $(F_{i})_{0 \leq i < N}$. Since $F_{i} (1-i) = 1 - F_{i}, (t_i)$, we have $I - F = P F P$. Let $V$ (resp. $T$) be the Vandermonde matrix of generic term $((t_i)_{0 \leq i < N})$ (resp. $(t_i^a - 1)_{0 \leq i < N}$). Let $B$ be the diagonal matrix of generic term $((-1)^i)_{0 \leq i < N}$ (notice we have $B B = I$). Since $t_i < 1 - t_{i-N-1}$, $i \in \{0, 1\}$, we have $V = PTB$. Let $G$ be the diagonal matrix of generic term $(1/(i!))_{0 \leq i < N}$. We get, using equation (8):

$$e^a = P F V G p_0 + P F V G q_0$$ (A.3)

Using relation (A.1) in the latter and writing $D = P F V G$ and $E = P F V G$, we get:

$$e^a = D p_0 + E q_0$$

Using equations (A.2) in the previous equations, writing $Q = Q^2 D^{-1}$ and $Q = Q^2 F_1^{-1} D^{-1}$ – which are both invertible – we get the global system:

$$p_0 = Q (d^a - R p_1), q_0 = Q (d^f - R q_1)$$

$$Q e^a = d^a - R p_1 + K (d^f - R q_1)$$

$$Q e^f = d^f - R p_1 + Q D F_1 p_1 + L_0 (d^f - R q_1) + L_1 q_1$$

Where $K = Q E Q$, $L_0 = Q E F Q$ and $L_1 = Q E F Q$, are invertible matrices. Subtracting the third line to the fourth, we get:

$$p_0 = Q (d^a - R p_1)$$

$$Q e^a = d^a - R p_1 + K (d^f - R q_1)$$

Then, injecting the fourth line in the third, we get an expression of $q_1$ depending only on $e^a$, $e^f$, $d^a$ and $d^f$. Assuming the matrix $R F_1^{-1} D^{-1} Q^2 (L_0 - K) R - L_1$ is invertible, we get an exact expression of $q_1$. Using the three other equations, we get the three other vectors $r_0$, $s_0$, $r_1$ and $s_1$ are then found using equations (A.1).

In the case of Burgers’ equation, we have to use the Hopf-Cole transform. Since no simple expression of $F(x,t)$ is yet known, one can only set the positions for initial and final formations where an explicit polynomial expression is known. In this case, writing the desired positions as $u_0$ and $u_1$, we have to solve the system:

$$A J_0 p_1 + H A J_1 D p_0 = u_0, A J_0 q_1 + H A J_1 D q_0 = u_1$$

where $D$ is the derivation matrix (all but the first upper diagonal are filled with zeros). In this case $d^a$ and $d^f$ are degrees of freedom and are used though optimization to achieve, in our implementation, trajectories of minimal amplitude.