Motion Planning for Multi-Agent Systems Using Gevrey Trajectories Based on Burgers' Viscous Equation

Etienne Servais, Brigitte Andrea-Novel, Hugues Mounier

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), Aug 2014, Cape Town, South Africa. Proceedings of the 19th IFAC World Congress 2014. <hal-01100841>

HAL Id: hal-01100841
https://hal-supelec.archives-ouvertes.fr/hal-01100841

Submitted on 7 Jan 2015

Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

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 (Olfati-Saber and Murray (2002)), geometric approaches (Briñon 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 to 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.

[⋆] This work was supported by DIGITEO grant 2011-043D and by the Région Île-de-France

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 a common choice allowing shock-like equilibria (see Krstić et al. (2008)) and has been used for motion-planning tasks (see Krstić et al. (2009); Meurer and Krstić (2011)).

In this paper, we consider a motion planning problem of a continuum of agents between successive equilibria at times t_i. The agents are described by their abscissa, $x = 0$ (resp. $x = 1$) designing the *left-most* (resp. *right-most*) agent. As in Meurer and Krstić (2011), we first consider our system as a continuum to solve the PDE and then discretize the solution to extract trajectories for the agents.

One of the problem arising in such models are the transitions between successive solutions. In Meurer and Krstić (2011) a new term, which is non-analytic at transition times t_i, is added to Burgers' equation to ensure smooth transitions between states. In the present paper we will instead look for non-analytic solutions of Burgers' genuine viscous equation so to allow infinite differentiability of the global solution at equilibria.

Using the Hopf-Cole transform $u = -2\mu\phi_x/\phi$, one transforms Burgers' viscous equation $u_t + uu_x = \mu u_{xx}$, where u denotes a spatial coordinate of the agent, into the standard heat equation $\phi_t = \mu\phi_{xx}$. As shown in the following proposition, if the solution of the heat equation is Gevrey of order α, then the resulting solution to Burgers' equation is Gevrey of order α.

Proposition 1. If ϕ is Gevrey of order α and if there exists $c > 0$ such as $\forall x \in [0,1], \forall t \in [0,1], \phi(x,t) > c$ then $u = \phi_x/\phi$ is Gevrey of order α.

Proof. Let $v(x,t) = \ln\phi(x,t)$. Since ln is analytic on $[c,\infty[$, it is Gevrey of order 1. Hence v is Gevrey (see Yamanaka (1989)) and has same order as ϕ as long as this order is greater than 1. Since $u = v_x$, u is Gevrey and has same order as v and thus, as ϕ.

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

Proposition 2. The heat equation with Cauchy boundary control in $x = 0$ is flat.

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

$$\begin{cases}\widehat{\phi}(x,s) = \widehat{C}_x\widehat{\lambda}_1(s) + \widehat{S}_x\widehat{\lambda}_2(s)\\ \widehat{C}_x = \cosh\left(x\sqrt{s/\mu}\right), \widehat{S}_x = \sqrt{\mu/s}\sinh\left(x\sqrt{s/\mu}\right)\end{cases}$$

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

It is obvious that $\partial_x C_x = \partial_t(S_x)/\mu$ and $\partial_x S_x = C_x$. Furthermore $C_0 = 1$ and $S_0 = 0$. Deriving ϕ according to x leads to $\phi_x(x,t) = \partial_t/\mu\,(S_x\lambda_1(t)) + C_x\lambda_2(t)$. 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 ϕ_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:

$$\phi(x,t) = C_x\phi_0(t) + S_x\phi_{x,0}(t) \tag{1}$$

we thus have:

$$\phi(1,t) = C_1\phi_0(t) + S_1\phi_{x,0}(t) \tag{2}$$

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

$$\phi_{x,0}(t) = (S_1)^{-1}\left(\phi(1,t) - C_1\phi_0(t)\right) \tag{3}$$

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

$$\phi(x,t) = \left(C_x - T_xC_1\right)\phi_0(t) + T_x\phi_1(t) \tag{4}$$

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:

$$\phi(x,t) = \left(C_xT_1 - T_xC_1\right)\phi_0(t) - \left(C_xT_0 - T_xC_0\right)\phi_1(t)$$

Using the Taylor expansion of csch (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 $u_k = 2(1-2^{2k-1})B_{2k}/(2k)!$ 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:

$$T_x = \sum_{k\geq 0}\left(\sum_{l=0}^{k}\frac{x^{2l+1}u_{k-l}}{(2l+1)!}\right)\frac{d^k}{dt^k} \tag{5}$$

We show in the following a meaning of this inversion.

Remark: Based on (Abramowitz and Stegun, 1965, p. 807), $u_k = (-1)^{k+1}4(1-2^{2k-1})\zeta(2k)/(2\pi)^{2k}$ where ζ 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:

$$\begin{cases} f_0(x) = \displaystyle\sum_{i=0}^{2N-1} p_i \frac{x^i}{i!} = \sum_{i=0}^{2N-1} r_i \frac{(x-1)^i}{i!} \\ f_1(x) = \displaystyle\sum_{i=0}^{2N-1} q_i \frac{x^i}{i!} = \sum_{i=0}^{2N-1} s_i \frac{(x-1)^i}{i!} \end{cases} \tag{6}$$

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

$$\overbrace{\begin{cases} \phi_0^{(i)}(0) &= p_{2i} \\ \phi_0^{(i)}(1) &= q_{2i} \\ \phi_{x,0}^{(i)}(0) &= p_{2i+1} \\ \phi_{x,0}^{(i)}(1) &= q_{2i+1} \end{cases}}^{0 \leq i < N} \quad \overbrace{\begin{cases} \phi_0^{(i)}(0) &= 0 \\ \phi_0^{(i)}(1) &= 0 \\ \phi_{x,0}^{(i)}(0) &= 0 \\ \phi_{x,0}^{(i)}(1) &= 0 \end{cases}}^{i \geq N} \tag{7}$$

(resp. in $x = 1$ with r_i and s_i).

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

$$\begin{cases} \phi_0(t) = \displaystyle\sum_{i=0}^{N-1} \left(p_{2i} \frac{t^i}{i!}(1 - \Phi(t)) + q_{2i} \frac{(t-1)^i}{i!} \Phi(t) \right) \\ \phi_1(t) = \displaystyle\sum_{i=0}^{N-1} \left(r_{2i} \frac{t^i}{i!}(1 - \Phi(t)) + s_{2i} \frac{(t-1)^i}{i!} \Phi(t) \right) \end{cases} \tag{8}$$

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 α, the latter proposed trajectories are Gevrey of order α. We will write this as:

$$\begin{cases} \phi_0(t) = P_0(t)(1 - \Phi(t)) + Q_0(t)\Phi(t) \\ \phi_1(t) = R_0(t)(1 - \Phi(t)) + S_0(t)\Phi(t) \end{cases}$$

Considering the functions for $t \leqslant 0$ and $t \geqslant 1$, we easily see that we achieve a $\mathcal{C}^\infty$ transition between two polynomials. Furthermore, the proposed solution verifies the condition (7) in $\phi_0(t)$. It verifies similar conditions for $\phi_1(t)$ with r_{2i} and s_{2i}. 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 $\gamma > 0$:

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

where:

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

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

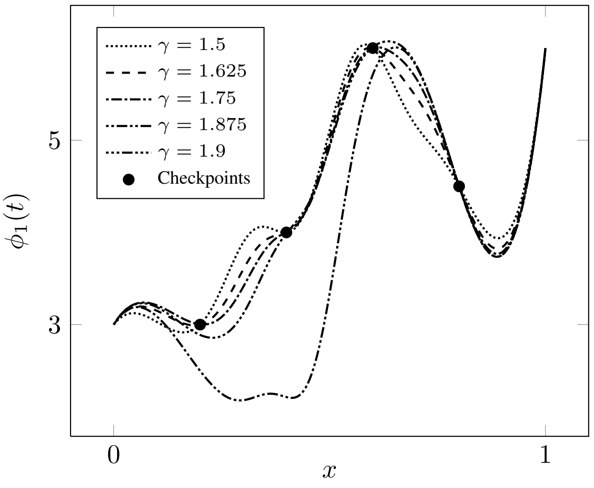

Fig. 1. Control in $x = 1$

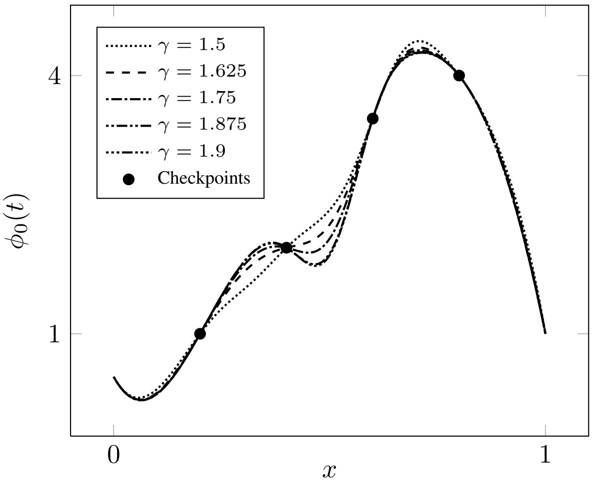

Fig. 2. Control in $x = 0$

3.2 Control validation

It is straightforward to solve the heat equation numerically with given controls ϕ_0 and ϕ_1 and initial condition f_0.

On the one hand, as shown on figure 1 and 2, the bigger γ, the more "bumpy" the control is. In the same time, the bigger γ, 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 γ is of great meaning in order to conform to final formations. While γ decreases toward 0, the controls are more and more Gevrey divergent. While γ increases, the more the Φ_γ 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.

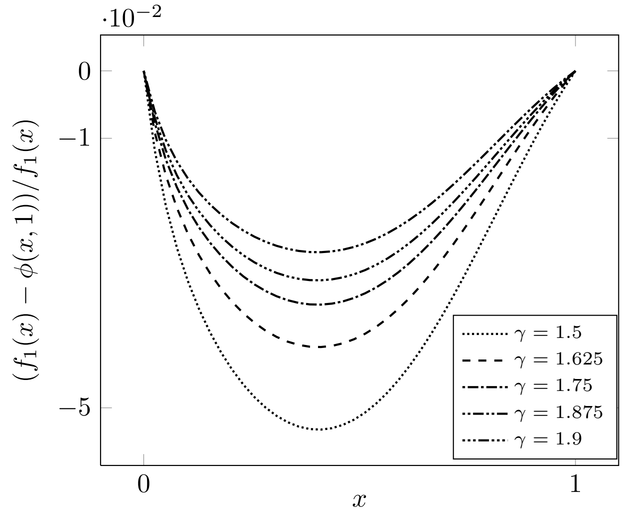

Fig. 3. Error between numerical resolution and model at $t = 1$

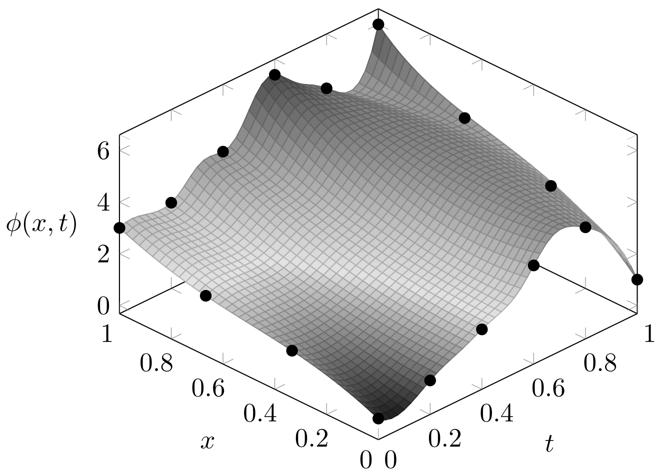

Fig. 4. Numerical resolution ($\gamma = 1.5$)

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 $\Phi^{(2)}(t) = \frac{\gamma(1-2t)}{((1-t)t)^{\gamma+1}}\Phi^{(1)}(t)$, we may write:

$$\begin{cases} \phi_0^{(k)}(t) \;=\; P_k(t)(1-\Phi(t)) + Q_k(t)\Phi(t) \\ \qquad\qquad\qquad\qquad\qquad + A_k(t)\Phi'(t) \\ \phi_1^{(k)}(t) \;=\; R_k(t)(1-\Phi(t)) + S_k(t)\Phi(t) \\ \qquad\qquad\qquad\qquad\qquad + B_k(t)\Phi'(t) \end{cases} \tag{9}$$

with the following reccurences:

$$\begin{cases} P_k(t) \;=\; P_0^{(k)}(t) = \sum_{l=0}^{N-1-k} p_{2(l+k)}\frac{t^l}{l!} \\ Q_k(t) \;=\; Q_0^{(k)}(t) = \sum_{l=0}^{N-1-k} q_{2(l+k)}\frac{(t-1)^l}{l!} \\ A_0(t) \;=\; 0 \\ A_k(t) \;=\; Q_{k-1}(t) - P_{k-1}(t) + A'_{k-1}(t) \\ \qquad\qquad\qquad + \frac{\gamma(1-2t)}{((1-t)t)^{\gamma+1}}A_{k-1}(t) \end{cases} \tag{10}$$

(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 ϕ_0 to be:

$$C_x - T_xC_1 = \sum_{k\geqslant 0}\left(\frac{x^{2k}}{(2k)!} - \sum_{l=0}^{k} v_{k-l}\frac{x^{2l+1}}{(2l+1)!}\right)\frac{d^k}{dt^k} \tag{11}$$

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

$$\phi(x,t) = D(x,t)(1-\Phi(t)) + E(x,t)\Phi(t) \\ + F(x,t)\Phi'(t) \tag{12}$$

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

$$D(x,t) = \sum_{l=0}^{N-1}\frac{t^l}{l!}f_0^{(2l)}(x), E(x,t) = \sum_{l=0}^{N-1}\frac{(t-1)^l}{l!}f_1^{(2l)}(x)$$

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

$$A_k(t) = k(Q_{k-1}(t) - P_{k-1}(t)) + \sum_{i=1}^{k}\frac{\mathcal{A}_k^i(t)}{(t(1-t))^{k+i\gamma}} \tag{13}$$

Where $\mathcal{A}_k^i$ 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 $\mathcal{A}_1^i$ 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:

$$F(x,t) = \sum_{k\geqslant 0}\left[\left(\frac{x^{2k}}{(2k)!} - \sum_{l=0}^{k} v_{k-l}\frac{x^{2l+1}}{(2l+1)!}\right)A_k(t) \right. \\ \left. + \left(\sum_{l=0}^{k} u_{k-l}\frac{x^{2l+1}}{(2l+1)!}\right)B_k(t)\right] \tag{14}$$

In future work, we will show that $\Phi_\gamma^{(1)}(t)F(x,t)$ 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 $\Phi_\gamma^{(1)}(t)F(x,t)$ term. This is done and plotted in figure 5.

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

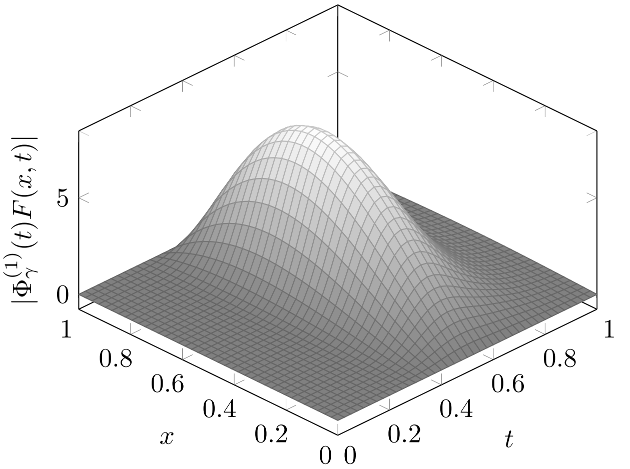

Fig. 5. Numerical evaluation of $|\Phi_\gamma^{(1)}(t)F(x,t)|$ ($\gamma = 1.5$)

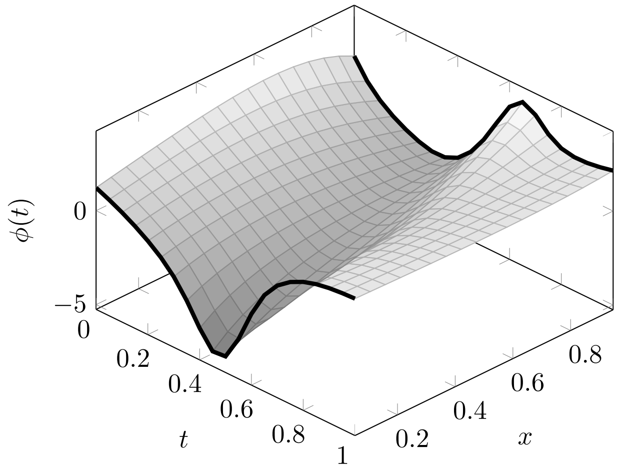

Fig. 6. Trajectories between constant states ($\gamma = 1.5$)

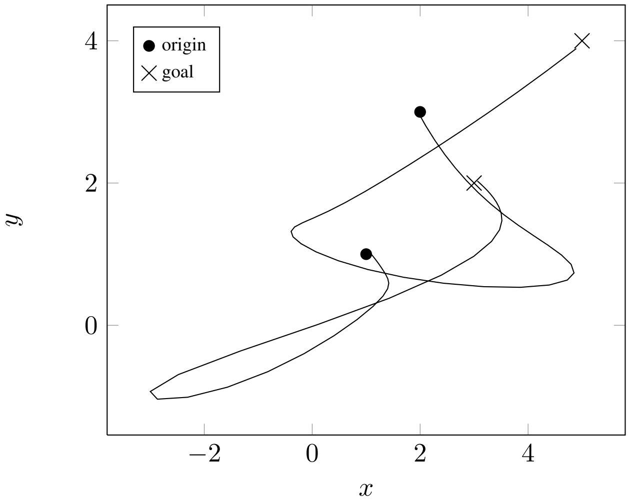

Fig. 7. Trajectories of two robots in a 2D plane ($\gamma = 1.5$)

4. CONCLUSION

We have shown a way to solve Burgers' viscous equation using Gevrey functions allowing infinitely differentiable transitions between successive solutions. These solutions are used for motion planning for multi-agent systems. Simulations are given while experiments using quadrotors are foreseen.

REFERENCES

Abramowitz, M. and Stegun, I.A. (1965). *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, volume 55. Dover publications.

Briñon Arranz, L., Seuret, A., and De Wit, C.C. (2011). Elastic formation control based on affine transformations. In *American Control Conference (ACC), 2011*, 3984–3989.

Cole, J.D. et al. (1951). On a quasi-linear parabolic equation occurring in aerodynamics. *Quart. Appl. Math*, 9(3), 225–236.

Coron, J.M. (2007). Some open problems on the control of nonlinear partial differential equations. In *Perspectives in Nonlinear Partial Differential Equations: In Honor of Haim Brezis (Henri Berestycki, Michiel Bertsch, Bert Peletier and Laurent Véron eds.)*, volume 446, 215–243. American Mathematical Society.

Fax, J.A. and Murray, R.M. (2002). Graph Laplacians and stabilization of vehicle formations. In *World Congress*, volume 15, 88–88.

Ferrari-Trecate, G., Buffa, A., and Gati, M. (2006). Analysis of coordination in multi-agent systems through partial difference equations. *Automatic Control, IEEE Transactions on*, 51(6), 1058–1063.

Fliess, M., Mounier, H., Rouchon, P., and Rudolph, J. (1998). A distributed parameter approach to the control of a tubular reactor: a multivariable case. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, volume 1, 439–442.

Frihauf, P. and Krstic, M. (2011). Leader-enabled deployment onto planar curves: A PDE-based approach. *Automatic Control, IEEE Transactions on*, 56(8), 1791–1806.

Gevrey, M. (1918). Sur la nature analytique des solutions des équations aux dérivées partielles. Premier mémoire. In *Annales Scientifiques de l'École Normale Supérieure*, volume 35, 129–190.

Glass, O. and Guerrero, S. (2007). On the uniform controllability of the Burgers equation. *SIAM Journal on Control and Optimization*, 46(4), 1211–1238.

Guerrero, S. and Imanuvilov, O.Y. (2007). Remarks on global controllability for the Burgers equation with two control forces. In *Annales de l'Institut Henri Poincare (C) Non Linear Analysis*, volume 24, 897–906.

Holmgren, E. (1908). Sur l'équation de propagation de la chaleur. *Ark. Mat. Astr. Fys.*, 4, 1–28.

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

Krstić, M., Magnis, L., and Vazquez, R. (2008). Nonlinear stabilization of shock-like unstable equilibria in the viscous Burgers PDE. *Automatic Control, IEEE Transactions on*, 53(7), 1678–1683.

Krstić, M., Magnis, L., and Vazquez, R. (2009). Nonlinear control of the viscous burgers equation: Trajectory generation, tracking, and observer design. *Journal of Dynamic Systems, Measurement, and Control*, 131, 021012.

Krstić, M. and Smyshlyaev, A. (2008). *Boundary Control of PDEs: A Course on Backstepping Designs.* Advances in Design and Control. Society for Industrial and Applied Mathematic.

Laroche, B., Martin, P., and Rouchon, P. (2000). Motion planning for the heat equation. *International journal of robust and nonlinear control*, 10(8), 629–643.

Meurer, T. and Krstić, M. (2011). Finite-time multi-agent deployment: A nonlinear PDE motion planning approach. *Automatica.*

Murray, R.M. and Dunbar, W.B. (2004). Distributed Receding Horizon Control With Application to Multi-Vehicle Formation Stabilization.

Olfati-Saber, R. and Murray, R.M. (2002). Distributed cooperative control of multiple vehicle formations using structural potential functions. In *IFAC World Congress*, 346–352.

Reynolds, C.W. (1987). Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH Computer Graphics*, volume 21, 25–34.

Rudolph, J., Winkler, J., and Woittennek, F. (2004). Flatness Based Trajectory Planning for Two Heat Conduction Problems Occurring in Crystal Growth Technology. *e-STA (Sciences et Technologies de l'Automatique).*

Van Tran, D. and Dinh, N.H. (1994). *Differential operators of infinite order with real arguments and their applications.* World Scientific Pub Co Inc.

Yamanaka, T. (1989). A new higher order chain rule and Gevrey class. *Annals of Global Analysis and Geometry*, 7(3), 179–203.

Appendix A

In this appendix, we show how to set the coefficients p_i, q_i, r_i and s_i so 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 $\mathbf{r_0}$ (resp. $\mathbf{s_0}$, $\mathbf{p_0}$, $\mathbf{p_1}$, $\mathbf{q_0}$ and $\mathbf{q_1}$) the vector (r_{2i}) (resp. (s_{2i}), (p_{2i}), (p_{2i+1}), (q_{2i}) and (q_{2i+1})), 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:

$$\mathbf{r_0} = F_0\mathbf{p_0} + F_1\mathbf{p_1}, \mathbf{s_0} = F_0\mathbf{q_0} + F_1\mathbf{q_1} \tag{A.1}$$

Let the respective index of every agent be $i/(N-1), 0 \leq i < N$. Let $\mathbf{d^s} = (d_i^s)$ (resp. $\mathbf{d^f} = (d_i^f)$) be the initial (resp. final) formation, so that $f_0(i/(N-1)) = d_i^s$ (resp. $f_1(i/(N-1)) = d_i^f$). We write A the invertible Vandermonde matrix of generic term $((i/N-1)^{2l})_{0\leq i,l<N}$ 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:

$$AJ_0\mathbf{p_0} + HAJ_1\mathbf{p_1} = \mathbf{d^s}, AJ_0\mathbf{q_0} + HAJ_1\mathbf{q_1} = \mathbf{d^f}$$

Writing $Q = J_0^{-1}A^{-1}$, which is invertible and $R = HAJ_1$ which is not, we get:

$$\mathbf{p_0} = Q\left(\mathbf{d^s} - R\mathbf{p_1}\right), \mathbf{q_0} = Q\left(\mathbf{d^f} - R\mathbf{q_1}\right) \tag{A.2}$$

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

$$\begin{cases} \mathbf{e^0} &= P\mathbf{\Phi}PVG\mathbf{p_0} + \mathbf{\Phi}PVGB\mathbf{q_0} \\ \mathbf{e^1} &= P\mathbf{\Phi}PVG\mathbf{r_0} + \mathbf{\Phi}PVGB\mathbf{s_0} \end{cases} \tag{A.3}$$

Using relation (A.1) in the latter and writing $D = P\mathbf{\Phi}PVG$ and $E = \mathbf{\Phi}PVGB$ (which are both invertible matrices), we get:

$$\begin{cases} \mathbf{e^0} &= D\mathbf{p_0} + E\mathbf{q_0} \\ \mathbf{e^1} &= D(F_0\mathbf{p_0} + F_1\mathbf{p_1}) + E(F_0\mathbf{q_0} + F_1\mathbf{q_1}) \end{cases} \tag{A.4}$$

Using equations (A.2) in the previous equations, writing $\hat{Q} = Q^{-1}D^{-1}$ and $\tilde{Q} = Q^{-1}F_0^{-1}D^{-1}$ – which are both invertible – we get the global system:

$$\begin{cases} \mathbf{p_0} &= Q\left(\mathbf{d^s} - R\mathbf{p_1}\right), \mathbf{q_0} = Q\left(\mathbf{d^f} - R\mathbf{q_1}\right) \\ \hat{Q}\mathbf{e^0} &= \mathbf{d^s} - R\mathbf{p_1} + K(\mathbf{d^f} - R\mathbf{q_1}) \\ \tilde{Q}\mathbf{e^1} &= \mathbf{d^s} - R\mathbf{p_1} + \tilde{Q}DF_1\mathbf{p_1} + L_0(\mathbf{d^f} - R\mathbf{q_1}) + L_1\mathbf{q_1} \end{cases}$$

Where $K = \hat{Q}EQ$, $L_0 = \tilde{Q}EF_0Q$ and $L_1 = \tilde{Q}EF_1$, are invertible matrices. Subtracting the third line to the fourth, we get:

$$\begin{cases} \mathbf{p_0} &= Q\left(\mathbf{d^s} - R\mathbf{p_1}\right), \mathbf{q_0} = Q\left(\mathbf{d^f} - R\mathbf{q_1}\right) \\ \hat{Q}\mathbf{e^0} &= \mathbf{d^s} - R\mathbf{p_1} + K(\mathbf{d^f} - R\mathbf{q_1}) \\ \tilde{Q}DF_1\mathbf{p_1} &= \tilde{Q}\mathbf{e^1} - \hat{Q}\mathbf{e^0} + (K - L_0)\mathbf{d^f} \\ &\quad + ((L_0 - K)R - L_1)\mathbf{q_1} \end{cases} \tag{A.5}$$

Then, injecting the fourth line in the third, we get an expression of $\mathbf{q_1}$ depending only on $\mathbf{e^0}$, $\mathbf{e^1}$, $\mathbf{d^s}$ and $\mathbf{d^f}$. Assuming the matrix $RF_1^{-1}D^{-1}\tilde{Q}^{-1}((L_0-K)-L_1)+KR$ is invertible, we get an exact expression of $\mathbf{q_1}$. Using the three other equations, we get the three other vectors. $\mathbf{r_0}$, $\mathbf{r_1}$, $\mathbf{s_0}$ and $\mathbf{s_1}$ 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 $\mathbf{u}_0$ and $\mathbf{u}_1$, we have to solve the system :

$$AJ_0\mathbf{p_1} + HAJ_1D\mathbf{p_0} = \mathbf{u_0}, AJ_0\mathbf{q_1} + HAJ_1D\mathbf{q_0} = \mathbf{u_1}$$

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