Rigorous Integration of Burgers Equation

Jacek Cyranka

Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland,
e-mail: jacek.cyranka@ii.uj.edu.pl

This paper presents techniques that allows to rigorously integrate dissipative partial differential equations. A full case study of an application to the Burgers equation on the line with periodic boundary conditions is presented.

Keywords and phrases: rigorous numerics, Lohner algorithm, Burgers equation, rigorous integration, dissipative partial differential equations, interval arithmetics.

Introduction

Basically speaking as *rigorous numerics* we mean methods of solving differential equations that operate on *intervals* and produce intervals that always contain the exact solution. Rigorous numerics for ordinary differential equations (ODEs) is a well established and analysed topic. There exist a few algorithms that offer reliable computations of the solution trajectories of ODEs based on interval arithmetics. The approach used in this paper is based on the presentation by Lohner in [1], see also [2]. It has made possible to prove many facts about the dynamics of certain ODEs (e.g. fixed points, periodic orbits and existence of attractors), let us mention the Rossler equation, the Lorenz equation and n-body problem (see [2–5]). In context of rigorous integration of ODEs we consider an abstract Cauchy problem

$$\begin{cases} \dot{x}(t) = f(x(t)), \\ x(0) = x_0. \end{cases} \tag{1}$$

$x\colon [0,\mathcal{T}] \to \mathbb{R}^n$, $f\colon \mathbb{R}^n \to \mathbb{R}^n$, $f \in C^\infty$. Furthermore we denote by $\varphi(t_k, x_0)$ a solution of (1) at time t_k and by $[x]$ an interval set $[x] \subset \mathbb{R}^n$ compact and connected, we also use another notation $[x] = \Pi_{k=1}^n[\hat{x}_k^-, \hat{x}_k^+]$, $[\hat{x}_k^-, \hat{x}_k^+] \subset \mathbb{R}$, $-\infty < \hat{x}_k^- \leq \hat{x}_k^+ < \infty$. The goal of a rigorous ODEs solver is to find a set $[x_k]$ such that

$$\varphi(t_k, [x_0]) \subset [x_k] \tag{2}$$

$t_k \in [0, \mathcal{T}]$, $[x_0]$, $[x_k]$ are interval sets. There are some subtle issues regarding set representation in the Lohner algorithm, which are discussed e.g. in [2]. Let us only mention that it is highly ineffective to store an interval set $(\Pi_{k=1}^n[a_k^-, a_k^+])$ explicitly, because it leads to the so-called wrapping effect. This leads to large over-estimates and prevents us from integrating over a longer time. Instead we store $[x]$ as $m([x])$ and $r([x]) = [x] - m([x])$, the middle point and the remainder separately.

Rigorous numerics for dissipative PDEs

Research considering rigorous numerics for partial differential equations is still at a pioneering stage. As far as I know, for non-stationary dissipative PDEs there exists only one algorithm, described in [6–9]. It has been successfully used for proving existence of fixed points for the Kuramoto-Sivasinsky equation. The analysis below is based on this approach. In Section 2 an application to the Burgers equation is provided.

Motivation

The aim of this section is to answer the question: "Why we want to rigorously integrate dissipative PDEs?". There is a considerable amount of numerical solvers available that are capable of obtaining approximate solutions of PDEs that are based on the Finite Difference and Finite Element Methods. Still, all of them have some limitations.

Generally, when we use them we can never answer the question whether what is being observed is a consequence of the dynamics embedded in an equation or only an illusion created by a loss of some important information as a result of using computer arithmetics that is capable of representing only finite precision numbers. In the present approach interest lies in proving some facts about the dynamics of a system that approximate simulations may suggest. Thus the accuracy of results is of critical importance, since we aim at having a reliable solver providing results that can be used in mathematical proofs. Moreover, from engineering point of view, when some proved facts about dynamics and a-priori bounds are available we can prevent performing many time consuming simulations. Another issue is related to the phenomenon called the "butterfly effect", that may arise when an equation is extremely prone to changes in initial conditions. The consequence is that very small changes in initial conditions leads to significant changes in the solution at an arbitrary time. The well known examples of equations that exhibit the butterfly effect are the Navier-Stokes (one observe it in the weather forecasts) and the Lorenz equations. It is of great interest to detect situations when this phenomenon can significantly affect calculations.

Dissipative PDEs

We consider dissipative PDEs are considered (dPDEs), this class contains many of the extensively studied equations (Navier-Stokes, Ginzburg-Landau, Kuramoto--Sivashinsky). PDEs of the following type are being considered:

$$\frac{du}{dt} = Lu + N(u, Du, \dots, D^r u). \tag{3}$$

$u\colon \mathbb{R}^n \times [0, \mathcal{T}] \to \mathbb{R}^n$, $x \in \mathbb{T}^d$ d-dimensional torus, L is linear operator, N is a polynomial, D^s- s-th collection of partial derivatives of order s. Moreover L is diagonal in Fourier basis $\{e^{ikx}\}_{k\in\mathbb{Z}^d}$:

$$Le^{ikx} = \lambda_k e^{ikx}$$

where the eigenvalues satisfy:

$$\lambda_k = -\nu(|k|)|k|^p, \quad 0 < \nu_0 \le \nu(|k|) \le \nu_1, \quad p > r.$$

We impose periodic boundary conditions, then $u(\cdot, x) = \sum_k u_k(\cdot)e^{ikx}$ and we can replace (3) with the following system of ODEs:

$$\frac{da_k}{dt} = \lambda_k a_k + N_k(u). \tag{4}$$

For this class of PDEs we can use the *method of self-consistent bounds* described in [6–9].

Burgers equation

To illustrate the approach we study the Burgers equation with viscosity

$$\frac{du}{dt} = -u \cdot u_x + \nu u_{xx}, \tag{5}$$

with $\nu > 0$ a viscosity constant, dimension $d = 1$, $u\colon \mathbb{R} \times [0, \mathcal{T}] \to \mathbb{R}$, $x \in \mathbb{R}$ and periodic boundary conditions $u(\cdot, x) = u(\cdot, x + 2k\pi)$. Parameters in (4) are as follows: $\lambda_k = -k^2\nu$, $r = 1$, $p = 2$ and $\nu(|k|) = \nu$ is constant. In the Fourier domain it has following form

$$\frac{da_k}{dt} = -i\frac{k}{2}\left(\sum_{k_1\in\mathbb{Z}} a_{k_1}a_{k-k_1}\right) - \nu k^2 a_k, \tag{6}$$

where $a_k \in H_k \subset \mathbb{C}$, further refereed to as the *Fourier modes.* In order to be able to solve numerically such an infinite set of ODEs using rigorous methods presented in Section 1.5 it has to be explained using finite tools somehow. To be able to rigorously integrate a PDE over time we must embed an infinite number of equations to finite structures and express them in a programmable way. One possibility is given in [9] , this approach is called the *method of self-consistent bounds.* The space to which $\{a_k\}_{k\in\mathbb{Z}}$ belongs is a Hilbert space $H = \overline{\oplus_{k\in\mathbb{Z}} H_k}$, ($H_k$ are finite dimensional subspaces of H that are mutually orthogonal) is split into two subspaces $X_m = \oplus_{|k|\le m} H_k = P_m(H)$ (comprising the modes that are the most relevant to the dynamics of the system), and the orthogonal part Y_m containing the remaining modes, m is arbitrarily chosen constant, P_m is the projection onto X_m. We operate on a specific class of sets i.e. the *self-consistent sets* i.e. $W \oplus T \subset H$, $W \subset X_m$, $Y_m \supset T = \Pi_{|k|>m} B_k(c_k, r_k)$, T *is the so-called tail* and where B_k is a ball of center c_k and radius r_k. One may either keep T constant in time or update it at each time step. The issue of choosing a proper size $m \in \mathbb{N}$ of the projection is quite subtle, generally the larger m the more precise the results are, as well as computation time that rises exponentially.

Details

In this section we give specific sets that we use for our purposes, moreover we provide fundamental theoretic result.

$$\begin{gathered} H = \{(a_k) : \textstyle\sum_{k\in\mathbb{Z}} |a_k|^2 < \infty\},\ H_k \subset \mathbb{C}, \\ \hat{W} \subset X_m = P_m(H),\ Y_m \supset \hat{T} = \Pi_{|k|>m}\hat{B}_k, \\ H_k \supset \hat{B}_k = \left[-\tfrac{C}{|k|^s}, \tfrac{C}{|k|^s}\right] \times \left[-\tfrac{C}{|k|^s}, \tfrac{C}{|k|^s}\right] \supset \\ \supset \overline{B_{\mathbb{C}}}(0, \tfrac{C}{|k|^s}),\ |k| \ge m \\ C \in \mathbb{R}_+ \cup \{0\},\ s \in \mathbb{R}_+. \end{gathered} \tag{7}$$

P_m is a symmetric Galerkin projection, i.e. if $k \in P_m$ then $-k \in P_m$, H_k are compact and connected.

We remark that finite number of interval sets from X_m and, as for the tail, only the constants C and s defining the radius of the ball $\overline{B_{\mathbb{C}}}(0, \frac{C}{|k|^s})$ are needed to be stored, thus such a sets can be represented in a computer finite memory. Moreover this set satisfies all assumptions [9] (Definition 3) thus we can deduce:

1. $\hat{W} \oplus \hat{T}$ is a compact subset of H,
2. uniform convergence $\lim_{n\to\infty} P_n(F(u)) = F(u)$ uniformly,
3. $\forall W_1 \oplus T_1$ - self-consistent bounds for (4) $\exists \hat{L}$ $\exists h > 0$ such that for all $l > \hat{L}$ and $\forall u \in P_l(\hat{W} \oplus \hat{T})$ $\varphi^l([0,h], u) \subset W_1 \oplus T_1$, and $|T_{1,k}| \leq \frac{C_1}{|k|^s}$, $|k| > m$.

The last statement is essentially thesis of Theorem 11 of [9], which states that dissipative PDEs have solutions within self-consistent bounds for a sufficiently short time.

Modification of original algorithm

In context of dPDEs instead of (1) we have to solve following ODE:

$$\frac{dx}{dt} = f(x(t)) + g(x(t), y(t)) \tag{8}$$

here $g : \mathbb{R}^l \times \mathbb{R}^n \to \mathbb{R}^l$, it is assumed that additional knowledge about $y(t)$ is available, e.g. is bounded and continuous. We reduce problem of solving (8) to the problem of solving following differential inclusion, meaning that every solution of (8) is contained in a solution of

$$\frac{dx}{dt}(t) \in f(x) + [\delta] \tag{9}$$

Let us fix $y_0 \in [\delta] \subset \mathbb{R}^l$ then denote by $\overline{\varphi}(t, x_0, y_0)$ the solution of the associated Cauchy problem: $\dot{x}(t) = f(x(t)) + y_0$, $x(0) = x_0$. We present a brief description of the suitable algorithm, we find

1. a bound $\overline{[x_t]} \subset \mathbb{R}^l$ such that $\overline{\varphi}(t, x_0, y_0) \subset \overline{[x_t]}$ can be found using original Lohner algorithm, see [2],
2. $[\Delta] \subset \mathbb{R}^l$ such that $\varphi(t, [x_0], [y_0]) \subset \overline{\varphi}(t, x_0, y_0) + [\Delta]$, see [9],
3. $[x_t] \subset \mathbb{R}^l$ such that $\varphi(t, [x_0], [y_0]) \subset [x_t] = \overline{[x_t]} + [\Delta]$, $\quad [x_0] \subset \mathbb{R}^l$, $[y_0] \subset (C_b([0, \infty), \mathbb{R}^n)$.

Let us provide some possible simplifications regarding $g(x(t), y(t))$. Later on in Chapter 2 the first possibility is used and explicit bounds for g are given, only first is presented, because the second one has proved to be ineffective.

First case

We look for $[\delta] = ([\epsilon_1^-, \epsilon_1^+], \ldots, [\epsilon_l^-, \epsilon_l^+])$ i.e. bounds constant in time and separated along coordinates, such that $g_i(x(t), y(t)) \in [\epsilon_i^-, \epsilon_i^+]$ $\forall i = 1 \ldots l$ and $\forall t \in [0, \mathcal{T}]$.

Second case

We look for $[\delta]$ constant in time and uniform for all coordinates $[\delta] = ([\epsilon^-, \epsilon^+], \ldots, [\epsilon^-, \epsilon^+])$. In order to find suitable ϵ we take $\epsilon^- = \min_{i=1\ldots n}\{\epsilon_i^-\}$, $\epsilon^+ = \max_{i=1\ldots n}\{\epsilon_i^+\}$, where $\epsilon_i^-, \epsilon_i^+$ from previous approach.

Example application

Let us analyse example of (5) with a periodic boundary condition, and initial condition $u(0, x) = \sin(x)$. Because the function u is real-valued, we operate on the invariant subspace of all symmetric projections $a_k = \overline{a_{-k}}$. The algorithm has been developed that enables rigorous integration. The more interesting case, studied by numerical analysis ts, taking the initial value u(0, *x*) = sin(*x*) with ν = 0.01 is currently beyond present capabilities, see (11). This case has nontrivial dynamics, there is shock-like discontinuity developing that causes instability no matter what numerical scheme is used. Experiments show that for ν ≈ 0.00001 after a short time all solvers (available in Matlab package) tended to break down.

Initial bounds

Firstly, we need some estimates regarding the tail influence, because we performed dimension splitting and cut off infinite number of modes, still they have to be present in our calculations in order to produce rigorous bounds. Let us provide a lemma that gives explicit bounds for the *tail influence*, we mean [δ] in (9), moreover it is constant in time.

Lemma 1. *Let (9) be the differential inclusion that rises from (6), l depends on the projection size, $x(t) \mapsto (a_1(t), a_2(t), \ldots, a_l(t))$, $[\delta] \subset \mathbb{R}^l$, $[\delta] = ([\delta_1], \ldots, [\delta_l])$,* $[\delta_i] = \frac{|k|}{2}\left(\frac{C}{(m+1)^s}\sqrt{E_0}\sqrt{2k} + 2C^2\sqrt{\left(1+\frac{1}{2m}\right)\left(1+\frac{1}{2(m+k)}\right)}\frac{1}{(2s-1)}\left(\frac{1}{(m+k)m}\right)^{s-\frac{1}{2}}\right)\cdot$ $[-1, 1]$ *then*

$$\begin{aligned} \max\nolimits_{t\in[0,\mathcal{T}]} g_i(x(t), y(t)) \leq \delta_i^+, \\ \min\nolimits_{t\in[0,\mathcal{T}]} g_i(x(t), y(t)) \geq \delta_i^-, \ i = 1, \ldots, l \end{aligned} \tag{10}$$

Trapping regions

In this section we aim at construction of a set depending on E_0 - initial value of the energy, C, s - constants from the bound $|a_k| \leq \frac{C}{|k|^s}$ and m - Galerkin projection size, noted by N_0, of such property that on the boundary of this set the vector field is pointing inside and for each solution of (6) after application of a symmetric Galerkin projection, if its initial value is inside then the solution will not leave it at any time. This set is used extensively in our algorithm.

Energy

Energy of (4) is defined by $E(\{a_k\}_{k\in\mathbb{Z}} = \sum_{k\in\mathbb{Z}} |a_k|^2$ and corresponding initial value: $E_0 = \sum_{k\in\mathbb{Z}} |a_k(0)|^2$. The lemma below gives exact value of velocity of the energy of (5) on the line. Note that as a corollary we state that $E(\{a_k\}_{k\in\mathbb{Z}}$ converges to 0.

Lemma 2. *For any solution of (5) such that all necessary Fourier series converge and $a_{-k} = \overline{a_k}$*

$$\frac{dE}{dt} = -2\nu \sum_{|k|\in\mathbb{Z}} k^2 a_k a_{-k}$$

Suitable trapping region

We look for sets enclosing values of all modes in (6) of such property that the vector field on boundary is pointing inside.

Lemma 3. *Let $D = 2^s \cdot 7 + 2^{s-1} + \frac{\left(\frac{5}{6}\right)^s \sqrt{\frac{3}{8}}}{\sqrt{2s-1}}$, $C > \sqrt{E_0} N^s$, $N > \left(\frac{\sqrt{E_0} D}{\nu}\right)^{\frac{2}{2s-3}}$ and $E_0 \in \mathbb{R}^+ - \{0\}$ then $N_0(E_0, N, C, s) = \Big\{ \{a_k\} \mid E(\{a_k\}) \leq E_0,$ $|a_k| \leq \frac{C}{|k|^s}, |k| > N \Big\}$ is a trapping region for each symmetric Galerkin projection.*

To prove that the specified sets are in fact trapping regions we have to check that on boundary the vector field is pointing inside, for the details the reader should refer to [10].

Simulations

In this section we present results of a few simulations we performed. Parameters: *m* is size of the Galerkin projection, *C* and s are defining radius of the ball in (7) *h* is the time step, ν is the viscosity constant in (5), *E* is value of the energy at the end of computations, moreover we give our program's running time in minutes. The computer program has been written in C++ language (gnu compiler), using CAPD library [11]. All calculations were performed on Windows XP, Core2Duo 2.20 GHz CPU computer.

nr	ν	*m*	*C*	*s*	*h*	≈ *E*	running time
1	20	8	1.6	4	0.0001	0.0001	2.9
2	25	10	1.6	4	0.0001	0.00001	5.5
3	50	4	1	4	0.00005	0.000001	9.2
4	100	8	1	4	0.00001	0.000001	9.2
5	10	12	4	4	0.0001	0.00017	20.4
6	20	16	1.6	4	0.0001	0.0001	18.7

Algorithms used

Because of the requirement of time efficiency we used C++ language with CAPD library. This library is mainly used for rigorous integration of ODEs. In the problem of the rigorous integration of the Burgers equation it turned out that the algorithms used originally in [11] were ineffective in differentiating the Galerkin projections of high dimension, since the algorithm used there is enforcing a decrease of time step with an increase of the projection size. Therefore it was necessary to derive classes that overshadow original methods and provide more dedicated and flexible algorithms. The rigorous integration of PDEs is at a pioneering stage at the moment.

The future

The Navier-Stokes equations and the problems related

We think that the Navier-Stokes equations in three dimensional space are a perfect subject of research.

$$\frac{d}{dt} u_i + \sum_{j=1}^{n} u_j \frac{\partial u_i}{\partial x_j} = \nu \Delta u_i - \frac{\partial p}{\partial x_i} + f_i(x,t),$$
$$i = 1, \ldots, n,$$
$$\mathrm{div} u = \sum_{i=1}^{n} \frac{\partial u_i}{\partial x_i} = 0.$$

It is widely known that they have refused to reveal their secrets for two centuries. Many mathematicians are involved in research aimed at providing an answer to the one of the most splendid problems of whole mathematics: whether smooth solutions of Navier-Stokes exist or there are cases when they do not, see [12] for description. It is hoped that the present research will result in a piece of knowledge that will provide some help in understanding the mysteries that lie beneath the Navier--Stokes equations.

Diffculties of rigorous numerical integration of the Navier-Stokes equations

We consider examining the Burgers equation from the rigorous numerics point of view as a perfect starting point to the final goal, which is the study of the Navier-

-Stokes equation with $n = 2$ and $n = 3$. Especially, it is planned to work on the latter because of the wide scope of problems it involves. Certainly numerical rigorous integration of the Navier-Stokes equation in 3D will require huge amount of computational power and development of new methods and algorithms. The main difficulty is the polynomial increase of storage space and computational time with increase of the dimension, when we switch from one dimensional to three dimensional, generally the complexity changes in a fashion $O(w(N)) \to O(w(N)^3)$.

Remarks

As for the limitations of the size of this paper we do not provide details of the proofs and source codes that will be given in [10].

References

[1] Lohner, R.J. "Computation of Guaranteed Enclosures for the Solutions of Ordinary Initial and Boundary Value Problems". *Computational Ordinary Differential Equations*, J.R. Cash, I. Gladwell Eds., Oxford: Clarendon Press, 1992.

[2] Zgliczyński, P. "C^1-Lohner algorithm". *Foundations of Computational Mathematics* 2 (2002): 429–465.

[3] Mrozek, M., and K. Mischaikow. "Chaos in Lorenz equations: a computer assisted proof". *Bull. Amer. Math. Soc*. (N.S.), 33 (1995): 66–72.

[4] Kapela, T., and P. Zgliczyński. „The existence of simple choreographies for the N-body problem — a computer assisted proof". *Nonlinearity* 16 (2003): 1899–1918.

[5] Tucker, W. "A Rigorous ODE Solver and Smale's 14th Problem". *Found. Comput. Math*. 2 (2002): 53–117.

[6] Zgliczyński, P. and K. Mischaikow. „Rigorous Numerics for Partial Differential Equations: the Kuramoto--Sivashinsky equation". *Foundations of Computational Mathematics* 1 (2001): 255–288.

[7] Zgliczyński, P. "Attracting fixed points for the Kuramoto--Sivashinsky equation — a computer assisted proof". *SIAM Journal on Applied Dynamical Systems* 1(2), 2000: 215–288.

[8] Zgliczyński, P. "Rigorous numerics for dissipative Partial Differential Equations II. Periodic orbit for the Kuramoto-Sivashinsky PDE -a computer assisted proof". *Foundations of Computational Mathematics* 4 (2004): 157–185.

[9] Zgliczyński, P. "Rigorous Numerics for Dissipative PDEs III. An effective algorithm for rigorous integration of dissipative PDEs". *Topological Methods in Nonlinear Analysis.* Preprint available online http://www.ii.uj.edu.pl/zgliczyn/papers/publ.htm

[10] Cyranka, J. "Rigorous integration of Burgers equation" [in preparation].

[11] CAPD — Computer Assisted Proofs in Dynamics, a package for rigorous numeric, http://capd.ii.uj.edu.pl.

[12] Fefferman, Ch. "Existence and Smoothness of the Navier-Stokes Equation". *The Millenium problem description*. Available online http://www.claymath.org/millennium/NavierStokes_Equations/.