Continuum Approximation-Based Power Series and Closed-Form Solutions to Large-Scale Backstepping Kernels Equations*

Jukka-Pekka Humaloja[1] and Nikolaos Bekiaris-Liberis[1]

***Abstract*— We provide two methods for computation of continuum backstepping kernels that arise in control of continua (ensembles) of linear hyperbolic PDEs and which can approximate backstepping kernels arising in control of a large-scale, PDE system counterpart (with computational complexity that does not grow with the number of state components of the large-scale system). In the first method, we provide explicit formulae for the solution to the continuum kernels PDEs, employing a (triple) power series representation of the continuum kernel and establishing its convergence properties. In the second method, we identify a class of systems for which the solution to the continuum (and hence, also an approximate solution to the respective large-scale) kernel equations can be constructed in closed form. We also present numerical examples to illustrate computational efficiency/accuracy of the approaches, as well as to validate the stabilization properties of the approximate control kernels, constructed based on the continuum.**

I. Introduction

Exponentially stabilizing feedback control laws based on the backstepping method [1] have been developed for several classes of hyperbolic PDEs in recent years, see [2]–[10]. The application of the method requires computing the backstepping control (and observer) gains, which involves solving the respective kernel PDEs. In certain instances, as in the case of 2×2 linear, hyperbolic systems with constant coefficients [11], it is possible to derive the solution to the kernel PDEs in closed-form; whereas, in general, the solution to the kernel PDEs has to be computed (i.e., approximated) numerically [12]–[16]. However, for large-scale hyperbolic systems, such numerical schemes for solving the kernel PDEs exhibit computational complexity that grows with the number of state components of the PDEs [17], [18]. Motivated by this, in the present paper, we provide tractable computational tools for constructing exponentially stabilizing feedback laws (by solving the respective kernel PDEs) for large-scale [3] and continua [2] of hyperbolic PDEs. We achieve this in two ways; by adapting the power series method [13] to solve continuum kernel PDEs and by providing closed-form (exact) solutions for a class of such kernel PDEs. As we have shown in [17], [18], the continuum control kernel provides a stabilizing feedback control gain for the respective large-scale hyperbolic PDE system as well.

*Funded by the European Union (ERC, C-NORA, 101088147). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

[1]The authors are with the Department of Electrical and Computer Engineering, Technical University of Crete, Chania, Greece. Emails: jhumaloja@tuc.gr and nlimperis@tuc.gr.

For large-scale, 1-D, linear hyperbolic systems, one of the most efficient approaches to compute backstepping control kernels is to utilize a closed-form, explicit solution. However, an exact, closed-form solution is available only for specific classes of hyperbolic systems, such as, for 2×2 systems with constant coefficients [11]. An alternative in providing explicit, backstepping control kernels, constitutes in computing approximate (yet explicit) solutions to the respective kernels PDEs via, for example, late lumping [14], neural operators [16], and power series [13], [19]–[21] -based approaches. In particular, the power series method provides a simple and flexible approach for computation of approximate backstepping kernels, thus making it a suitable candidate for computation of the solutions to continua and large-scale kernels PDEs.

As the main contribution of the paper, we present a triple power series-based approach to compute the solution to continuum kernel equations on a prismatic 3-D domain, thus, providing a systematic, computational tool for constructing stabilizing feedback laws for continua of hyperbolic PDEs. In turn, utilizing our recent results on continuum approximation of large-scale systems of hyperbolic PDEs [17], [18], the stabilizing feedback laws obtained for the continuum can be directly employed in stabilizing the corresponding large-scale system. We provide theoretical guarantees that the power series approximation is convergent, provided that the parameters of the continuum kernel PDEs are analytic, and present a procedure to compute the solution. We show that the power series method to compute continuum kernels results in providing stabilizing control kernels for the respective large-scale system with computational complexity that does not grow with the number of PDE state components (in contrast to the case of employing the power series approach to solve the large-scale kernel equations). We then apply the proposed approach to a stabilization problem of a large-scale hyperbolic system of PDEs, where an exponentially stabilizing state-feedback gain can be computed efficiently by combining the continuum approximation approach [17], [18] and the proposed triple power series method. We also conduct numerical experiments to illustrate the computational effectiveness/accuracy of the (triple) power series method. In addition to the power series method, we identify sufficient conditions for the parameters of the continuum kernel equations (as well as for the parameters of the corresponding large-scale kernel equations), under which the solution can be expressed in closed form and provide the explicit solution in a constructive manner.

II. Large-Scale Kernel Equations and Their Continuum Approximation

For $i = 1, \dots, n$, where n is large, we consider kernel equations of the form [3], [17]

$$\mu(x)\partial_x k^i(x,\xi) - \lambda_i(\xi)\partial_\xi k^i(x,\xi) - \lambda_i'(\xi)k^i(x,\xi) = \frac{1}{n}\sum_{j=1}^{n}\sigma_{j,i}(\xi)k^j(x,\xi) + \theta_i(\xi)k^{n+1}(x,\xi), \quad (1a)$$

$$\mu(x)\partial_x k^{n+1}(x,\xi) + \mu(\xi)\partial_\xi k^{n+1}(x,\xi) = -\mu'(\xi)k^{n+1}(x,\xi) + \frac{1}{n}\sum_{j=1}^{n}W_j(\xi)k^j(x,\xi), \quad (1b)$$

on a triangular domain $0 \le \xi \le x \le 1$ with boundary conditions

$$k^i(x,x) = -\frac{\theta_i(x)}{\lambda_i(x)+\mu(x)}, \quad (2a)$$

$$\mu(0)k^{n+1}(x,0) = \frac{1}{n}\sum_{j=1}^{n}q_j\lambda_j(0)k^j(x,0), \quad (2b)$$

for all $x \in [0,1]$. Our aim is to find approximate solutions to (1), (2) by solving the corresponding continuum kernel equations of the form [2], [17]

$$\mu(x)\partial_x k(x,\xi,y) - \lambda(\xi,y)\partial_\xi k(x,\xi,y) - \theta(\xi,y)\bar{k}(x,\xi) = k(x,\xi,y)\partial_\xi\lambda(\xi,y) + \int_0^1\sigma(\xi,\eta,y)k(x,\xi,\eta)d\eta, \quad (3a)$$

$$\mu(x)\partial_x\bar{k}(x,\xi) + \mu(\xi)\partial_\xi\bar{k}(x,\xi) = -\mu'(\xi)\bar{k}(x,\xi) + \int_0^1 W(\xi,y)k(x,\xi,y)dy, \quad (3b)$$

on a prismatic domain $0 \le \xi \le x \le 1, y \in [0,1]$ with boundary conditions

$$k(x,x,y) = -\frac{\theta(x,y)}{\lambda(x,y)+\mu(x)}, \quad (4a)$$

$$\mu(0)\bar{k}(x,0) = \int_0^1 q(y)\lambda(0,y)k(x,0,y)dy, \quad (4b)$$

for all $x \in [0,1]$ and for almost every $y \in [0,1]$.

The kernel equations (1), (2) appear in backstepping control of $n+1$ hyperbolic PDEs [3], where the solution $\left(k^i\right)_{i=1}^{n+1}$ provides an exponentially stabilizing state feedback gain for the system. More recently [2], the backstepping control methodology has been extended to continua of hyperbolic PDEs, where the exponentially stabilizing state feedback gain is obtained from the solution $(k,\bar{k})$ to the continuum kernel equations (3), (4). In our recent work [17], we have shown that the continuum kernel equations (3), (4) can be interpreted as an approximation of (1), (2) when n is sufficiently large Thus, the exponentially stabilizing state feedback gain for the large-scale $n+1$ hyperbolic PDE can be approximated from the solution to (3), (4), without compromising the exponential stability of the closed-loop system. The relation between the parameters of (1), (2) and (3), (4) was established in [17] as

$$\lambda(x,i/n) = \lambda_i(x), \quad (5a)$$

$$W(x,i/n) = W_i(x), \quad (5b)$$

$$\theta(x,i/n) = \theta_i(x), \quad (5c)$$

$$\sigma(x,i/n,j/n) = \sigma_{i,j}(x), \quad (5d)$$

$$q(i/n) = q_i, \quad (5e)$$

for all $x \in [0,1]$ and $i, j = 1,2,\dots,n$, where the parameters of (3), (4) are assumed to be continuous in y and η so that the pointwise evaluations are well-defined. Under (5), we have shown in [17] that the solution to (1), (2) converges to the solution of (3), (4) (in the L^2 sense in y)[1]. Consequently, the solution of (3), (4) can be used to approximate the solution to (1), (2) and to construct an exponentially stabilizing feedback gain for a large-scale $n+1$ hyperbolic system of PDEs, which is our main motivation of studying solving (3), (4).

III. Power Series Approach to Solving Continuum Kernel Equations

A. Convergence of Power Series Representation

As per [13], the idea is to find the solution to (3), (4) as a power series, which in case of $k(x,\xi,y)$ is a triple power series

$$k(x,\xi,y) = \sum_{\ell=0}^{\infty}\sum_{i=0}^{\infty}\sum_{j=0}^{i}K_{ij\ell}x^{i-j}\xi^j y^\ell, \quad (6)$$

whereas for $\bar{k}(x,\xi)$ the power series representation is

$$\bar{k}(x,\xi) = \sum_{i=0}^{\infty}\sum_{j=0}^{i}\bar{K}_{ij}x^{i-j}\xi^j. \quad (7)$$

Similarly, the parameters of (3), (4) are represented by the series

$$\lambda(x,y) = \sum_{i=0}^{\infty}\sum_{j=0}^{i}\lambda_{ij}x^{i-j}y^j, \quad (8a)$$

$$\mu(x) = \sum_{i=0}^{\infty}\mu_i x^i, \quad (8b)$$

$$\theta(\xi,y) = \sum_{i=0}^{\infty}\sum_{j=0}^{i}\theta_{ij}\xi^{i-j}y^j, \quad (8c)$$

$$W(\xi,y) = \sum_{i=0}^{\infty}\sum_{j=0}^{i}W_{ij}\xi^{i-j}y^j, \quad (8d)$$

$$\sigma(x,\eta,y) = \sum_{\ell=0}^{\infty}\sum_{i=0}^{\infty}\sum_{j=0}^{i}\sigma_{ij\ell}x^{i-j}\eta^j y^\ell, \quad (8e)$$

$$q(y) = \sum_{i=1}^{\infty}q_i y^i, \quad (8f)$$

where the coefficients are obtained from the Taylor series of the respective parameters. Similarly to [13], we consider the

[1]This relies on interpreting the continuum parameters of (3), (4) as the limits of sequences of functions defined as piecewise constant in y, on intervals of the form $((i-1)/n, i/n]$.

parameters and the kernels appearing in (6)–(8) complex-valued, and by a polydisk we refer to $\mathscr{D}_L \times \mathscr{D}_L \times \mathscr{D}_L$ (or $\mathscr{D}_L \times \mathscr{D}_L$ if only two spatial variables are involved), where $\mathscr{D}_L$ is a complex-valued open disk centered at the origin of radius L, i.e., $\mathscr{D}_L = \{z \in \mathbb{C} : |z| < L\}$. For the power series in (6)–(8) to converge in the domain of definition of the kernel equations (3), the involved functions have to be analytic on polydisks with radius larger than one.

By [2, Thm 3], under sufficient continuity assumptions on the parameters of (3), (4), there exists a unique solution $(k, \bar{k})$ to (3), (4). However, representing the solution as a power series requires the solution to be analytic, so that stronger assumptions have to be imposed on the parameters of (3), (4). On the other hand, if an analytic solution exist, it is uniquely given by the power series (6), (7), because of the uniqueness of the solution and the Taylor series representation. Similar to the results of [13], we utilize the well-posedness result [2, Thm 3] of (3), (4) to state the following.

Theorem 3.1: If the parameters of (3), (4) are analytic on polydisks with radii larger than one, so that they can be represented as the series in (8), and $|\lambda(z_1, z_2)| > 0, |\mu(z_3)| > 0$ for all z_1, z_2, z_3 in these polydisks, then the series defined in (6), (7) converge. That is, they define analytic functions on polydisks with radii larger than one, which are the unique solution to the kernel equations (3), (4).

Proof: See the proof of [22, Thm 1]. ■

B. Solution Computation with Truncated Series

Based on Theorem 3.1, we can construct arbitrarily accurate approximations of the solution to (3), (4), assuming that the involved parameters are analytic, by truncating the infinite series (6)–(8) up to some sufficiently high order N. This results in finding an approximate solution $(k^{\mathrm{a}}, \bar{k}^{\mathrm{a}})$ to (3), (4) as

$$k^{\mathrm{a}}(x, \xi, y) = \sum_{\ell=0}^{N} \sum_{i=0}^{N-\ell} \sum_{j=0}^{i} K^{\mathrm{a}}_{ij\ell} x^{i-j} \xi^{j} y^{\ell}, \tag{9a}$$

$$\bar{k}^{\mathrm{a}}(x, \xi) = \sum_{i=0}^{N} \sum_{j=0}^{i} \bar{K}^{\mathrm{a}}_{ij} x^{i-j} \xi^{j}, \tag{9b}$$

for parameters approximated by truncating the respective power series (8) to order N. Note that the coefficients of (9), may not correspond (for any finite N) to the respective coefficients of (6), (7), as both are unknown. A numerical procedure to compute the coefficients of (9) by least squares is presented in Algorithm 1. For the algorithm, as also noted in [13, Sect. III.D], it is reasonable to express the boundary condition (4a) as

$$(\lambda(x, y) + \mu(x)) k(x, x, y) = -\theta(x, y), \tag{10}$$

so that both sides are polynomials in x and y, because the original form (4a) is not directly compatible with the power series approach.

The accuracy of the approximation obtained by following the above procedure can be assessed based on the residual $\|\mathbf{A}\mathbf{x}^{\mathrm{a}} - \mathbf{b}\|_2$ of the least-squares fit after solving the (usually overdetermined) set of linear equations. If the residual is large, then one should increase the approximation order N. Assuming that the solution to (3), (4) is analytic (e.g., under Theorem 3.1), the approximate series (9) tend to (6), (7) as $N \to \infty$, i.e., the coefficients in the approximate series tend to their exact values in the infinite series. Consequently, as the exact coefficients would solve the (infinite) set of linear equations $\mathbf{A}\mathbf{x} = \mathbf{b}$ accurately, the residual $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$ tends to zero as $N \to \infty$. Moreover, as the convergence of power series is uniform, the obtained approximate solution is arbitrarily close to the exact solution pointwise, meaning that we also get arbitrarily accurate approximations for the state feedback gains $k(1, \xi, y)$ and $\bar{k}(1, \xi)$ employed in backstepping control.

Data: Parameters $\lambda, \mu, \sigma, \theta, W, q$ of (3), (4).
Result: Truncated power series approximation (9) for $(k, \bar{k})$.
Initialization:
  Choose approximation order N.
  Construct the approximate solution (9) and truncate the series (8) to order N.
  Insert the truncated series into (3), (4).
  Store the coefficients of each term $x^i \xi^{j-i} y^{\ell}$ from (3), (4) into a list.
  Initialize matrix $\mathbf{A}$ and vector $\mathbf{b}$ based on the number of unknown coefficients $K^{\mathrm{a}}_{ij\ell}, \bar{K}^{\mathrm{a}}_{ij}$ and the length of the list.
while *not at the end of the list* **do**
  Check next element in the list of coefficients.
  if *coefficient has* $K^{\mathrm{a}}_{ij\ell}$ *or* $\bar{K}^{\mathrm{a}}_{ij}$ **then**
    Insert the numerical value to the corresponding position in $\mathbf{A}$.
  else
    Insert the numerical value to the corresponding position in $\mathbf{b}$.
  end
end
Solve for the unknown coefficients from $\mathbf{A}\mathbf{x}^{\mathrm{a}} = \mathbf{b}$ by least squares.
Insert the obtained values $\mathbf{x}^{\mathrm{a}}$ to (9).

Algorithm 1: Computation of the power series approximation (9). A MATLAB implementation of the algorithm is available at https://github.com/jphumaloja/Continuum-Kernels-Power-Series/.

IV. On Explicit Solution to Continuum Kernel Equations

We identify sufficient conditions on the parameters of the kernel equations (1), (2), under which the continuum kernel equations (3), (4) can be solved explicitly. The main idea towards this is to look for separable solutions to the continuum kernel equations, the existence of which requires certain assumptions on the continuum parameters, and respectively, on the parameters of the original large-scale kernel equations.

Assumption 4.1: For the parameters of the kernel equations (1), (2), we assume that $\mu(x) = \mu$ and $\lambda_1(x) = \lambda_2(x) =$

$\ldots = \lambda_n(x) = \lambda$, where μ and λ are positive constants. Moreover, the parameters $W_i(x), \sigma_{i,j}(x), \theta_i(x)$ are of the form

$$W_i(x) = w_i W_x(x), \tag{11a}$$
$$\sigma_{i,j}(x) = s_{1,i} s_{2,j} \sigma_x(x), \tag{11b}$$
$$\theta_i(x) = \vartheta_i \theta_x(x), \tag{11c}$$

for some constants $w_i, s_{1,i}, s_{2,j}, \vartheta_i$ for all $i, j = 1, \ldots, n$.
Under Assumption 4.1, the parameters of the continuum kernel equations (3), (4) can be constructed to be separable, i.e., there exist functions $W_y, \theta_y, \sigma_y, \sigma_\eta$ such that

$$W(x,y) = W_x(x) W_y(y), \tag{12a}$$
$$\sigma(x,y,\eta) = \sigma_x(x)\sigma_y(y)\sigma_\eta(\eta), \tag{12b}$$
$$\theta(x,y) = \theta_x(x)\theta_y(y), \tag{12c}$$

where

$$W_y(i/n) = w_i, \tag{13a}$$
$$\theta_y(i/n) = \vartheta_i, \tag{13b}$$
$$\sigma_y(i/n) = s_{1,i}, \tag{13c}$$
$$\sigma_\eta(i/n) = s_{2,i}, \tag{13d}$$

for all $i = 1, \ldots, n$. Now, if we pose some additional assumptions on the parameters appearing in (12), we can guarantee the existence of a separable, closed-form solution to the continuum kernel equations (3), (4). This is formulated in the following proposition. The motivation for the proposition is that we can impose corresponding conditions on the parameters of the large-scale kernel equations (1), (2), so that the closed-form continuum solution to (3), (4) provides an approximation for the solution to (1), (2). This is discussed in detail after Proposition 4.2 in Remark 4.3.

Proposition 4.2: Let Assumption 4.1 hold so that λ and μ are positive constants and the continuum parameters are separable as in (12), satisfying (13). Additionally, assume that there exists a constant c_y such that

$$c_y = \frac{\sigma_\eta(y)}{\theta_y(y)} \int_0^1 \sigma_y(\eta)\theta_y(\eta) d\eta, \tag{14}$$

i.e., either there exists a constant c such that $\sigma_\eta = c\theta_y$ or the integral appearing in (14) is zero. Moreover, we assume that

$$c_y \sigma_x'(\xi) + \lambda \frac{\theta_x''(\xi)\theta_x(\xi) - \theta_x'(\xi)^2}{\theta_x(\xi)^2} = W_x(\xi)\theta_x(\xi) \int_0^1 W_y(y)\theta_y(y) dy, \tag{15}$$

holds for all $\xi \in [0,1]$. Then, the solution to the continuum kernel equations (3), (4) is given by

$$k(x,\xi,y) = -\frac{1}{\lambda+\mu} \exp\left(\frac{c_x}{\mu} x\right) \exp\left(-\frac{c_x}{\mu}\xi\right) \theta_x(\xi)\theta_y(y), \tag{16a}$$

$$\bar{k}(x,\xi) = \exp\left(\frac{c_x}{\mu} x\right) \bar{k}_\xi(\xi), \tag{16b}$$

where

$$\bar{k}_\xi(\xi) = \left(\frac{c_y}{\lambda+\mu}\sigma_x(\xi) - \frac{c_x}{\mu} + \frac{\lambda}{\lambda+\mu}\frac{\theta_x'(\xi)}{\theta_x(\xi)}\right) \exp\left(-\frac{c_x}{\mu}\xi\right), \tag{17}$$

and

$$c_x = \frac{\mu}{\lambda+\mu}\left(c_y\sigma_x(0) + \lambda\frac{\theta_x'(0)}{\theta_x(0)} + \frac{\lambda}{\mu}\theta_x(0)\int_0^1 q(y)\theta_y(y)dy\right). \tag{18}$$

Proof: See the proof of [22, Prop. 2] and [22, Rem. 2]. ■

Remark 4.3: Under Assumption 4.1, the conditions of Proposition 4.2 can be translated to conditions on the parameters of the large-scale equations shown in (11). Regarding (14), we either need that there exists a constant c such that $\vartheta_i = c s_{2,i}$ for all $i = 1, 2, \ldots, n$, or

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^n s_{1,i}\vartheta_i = 0, \tag{19}$$

where the limit converges to the integral[2] in (14), by construction of the continuum parameters. Similarly, the condition (15) can be written in terms of (11) by replacing the integral with the corresponding infinite sum, i.e.,

$$c_y\sigma_x'(\xi) + \lambda\frac{\theta_x''(\xi)\theta_x(\xi) - \theta_x'(\xi)^2}{\theta_x(\xi)^2} W_x(\xi)\theta_x(\xi) \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^n w_i\vartheta_i, \tag{20}$$

for all $\zeta \in [0,1]$.

Example 1. The simplest case in which the conditions of Proposition 4.2 are satisfied is when $c_y = 0$ and both sides of (15) are zero. Moreover, the solution formula becomes simpler in the particular case $c_x = 0$. These particular conditions (and particular value of c_x) are met in the example considered in [2, Sect. VII], where $\lambda = \mu = 1, W_y(y) = \sigma_y(y) = y - \frac{1}{2}, \theta_y(y) = y(y-1), \theta_x(x) = -70\exp\left(\frac{35}{\pi^2}x\right)$, and $q(y) = \cos(2\pi y)$, in which case the solution (16) becomes

$$k(x,\xi,y) = 35y(y-1)\exp\left(\frac{35}{\pi^2}\xi\right), \quad \bar{k}(x,\xi) = \frac{35}{2\pi^2}. \tag{21}$$

V. Numerical Experiments: Power Series-Based Computation for Example 1

We test the power series procedure with the parameters of Example 1 to get an idea of the practical computational requirements and the accuracy of the method.[3] In Table I, numerical results are presented of computation of the power series (9), where the first column shows the order N of the approximation; the second column shows the number of unknown coefficients (both K and $\bar{K}$ together); the third column shows the number of (linear) equations (that have to be solved for determining the series coefficients); the fourth column shows the computational time (in seconds); the fifth

[2]Formally, to the Riemann integral, and due to continuity of the parameters, this coincides with the standard Lebesgue integral.

[3]The numerical experiments are performed in MATLAB 2023b on a 2023 MacBook Pro with Apple M3 chip and 8 GB of memory.

column shows the residual of the least-squares fit when the set of linear equations is solved; and the sixth column shows the maximal absolute error between the computed and the exact, continuum control gains (i.e., the kernels (21) evaluated along $x = 1$). One can see that both the number of unknowns and the number of equations grow rapidly along with N, and hence, so does the computational time. On the flip side, the residual of the least-squares fit and the maximal error of the control kernel approximation become smaller as N increases.

N	#K	#Eq	C. time	$\|\mathbf{Ax}-\mathbf{b}\|_2$	max error
12	546	1082	11.78 s.	1.90	16.95
13	665	1285	13.71 s.	0.669	8.79
14	800	1510	18.27 s.	0.209	0.668
15	952	1758	25.24 s.	$5.53\cdot10^{-2}$	0.622
16	1122	2030	32.79 s.	$1.13\cdot10^{-2}$	0.116
17	1311	2327	40.90 s.	$3.10\cdot10^{-3}$	$9.14\cdot10^{-2}$
18	1520	2650	50.35 s.	$6.83\cdot10^{-4}$	$7.23\cdot10^{-3}$
19	1750	3000	62.33 s.	$1.42\cdot10^{-4}$	$7.61\cdot10^{-3}$
20	2002	3378	77.70 s.	$2.82\cdot10^{-5}$	$5.68\cdot10^{-4}$

TABLE I

NUMERICAL RESULTS FOR POWER SERIES (9) CORRESPONDING TO EXAMPLE 1.

VI. APPLICATION TO STABILIZATION OF A LARGE-SCALE SYSTEM

A. Parameters and Their Continuum Approximation

We consider a large-scale system of $n+1$ hyperbolic PDEs (see [3], [17, Sect. II]) with parameters

$$\lambda_i(x) = 1, \qquad \mu(x) = 1, \tag{22a}$$

$$\sigma_{i,j}(x) = x^3(x+1)\left(\frac{i}{n}-1\right)\left(\frac{j}{n}-1\right), \tag{22b}$$

$$W_i(x) = 2x(x+1)\frac{i}{n}, \tag{22c}$$

$$\theta_i(x) = -70x\frac{i}{n}\left(\frac{i}{n}-1\right), \tag{22d}$$

$$q_i = \frac{i}{n}\left(\frac{i}{n}-1\right) \tag{22e}$$

for $i, j = 1, 2, \ldots, n$, where we take $n = 30$. As discussed in Section II, when n is sufficiently large, such $n+1$ system can be approximated by a corresponding ensemble of linear PDEs (see also [17]). In particular, the solution to the large-scale kernel equations (1), (2) can be approximated by solving the corresponding continuum kernel equations (3), (4). In order to do this, we need to construct the continuum parameters such that the relations (5) are satisfied. Due to the structure of the parameters (22), this can be achieved by replacing i/n and j/n by y and η, respectively, which results in continuum parameters as

$$\lambda(x,y) = 1, \qquad \mu(x) = 1, \tag{23a}$$

$$\sigma(x,y,\eta) = x^3(x+1)(y-1)(\eta-1), \tag{23b}$$

$$W(x,y) = 2x(x+1)y, \tag{23c}$$

$$\theta(x,y) = -70xy(y-1), \tag{23d}$$

$$q(y) = y(y-1). \tag{23e}$$

B. Computation of Continuum Kernels with Power Series

We note first that for the continuum parameters constructed in Section VI-A, Proposition 4.2 is not applicable for finding the solution to (3), (4) in closed form. Thus, we employ the power series approach to solve the continuum kernel equations.

The results for the power series computations are shown in Table II, which shows the data corresponding to Table I for this case. However, as we do not know the exact solution to the kernel equations (3), (4) for parameters (23), instead of the maximal error to the exact solution, the penultimate column shows the maximal difference d_{n+1} to the solution of the large-scale kernel equations, which we have computed based on a finite-difference approximation of (1), (2). Moreover, the last column shows the maximal difference d_{N-1} to the solution obtained with power series of order $N-1$. The last two columns imply that the power series solutions to (3), (4) converge as N increases. However, as $n = 30$, there persists an approximation error between the solutions to (1), (2) and (3), (4), which would tend to zero as $n \to \infty$ (see [17]), if the solutions to (1), (2) and (3), (4) were exact. The obtained approximate solutions (9a) evaluated along $x = 1$ are displayed in Fig. 1, where visually it is difficult to distinguish the solutions after $N = 20$, further supporting the conclusion of convergent power series approximations.

N	#K	#Eq	C. time	$\|\mathbf{Ax}-\mathbf{b}\|_2$	d_{n+1}	d_{N-1}
6	112	221	1.37 s.	4.64	5.96	–
10	352	533	4.75 s.	3.46	4.84	0.162
15	952	1223	18.89 s.	2.07	2.91	0.127
20	2002	2363	38.21 s.	0.414	1.22	0.183
25	3627	4078	95.05 s.	$2.64\cdot10^{-2}$	1.19	$2.09\cdot10^{-3}$
30	5952	6493	233.49 s.	$9.30\cdot10^{-4}$	1.19	$1.62\cdot10^{-5}$

TABLE II

NUMERICAL RESULTS FOR POWER SERIES APPROXIMATION (9) CORRESPONDING TO PARAMETERS (23).

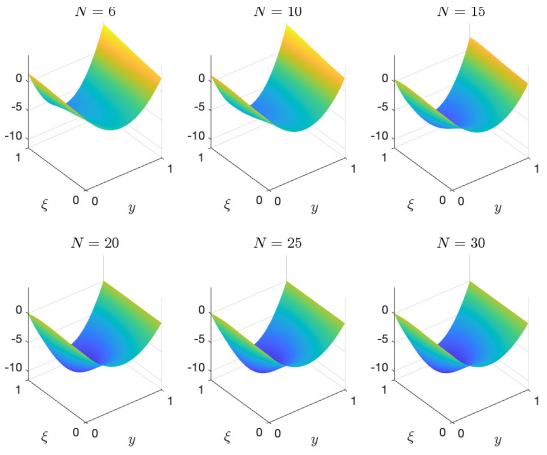

Fig. 1. The control gain $k(1,\xi,y)$ approximated by power series (9a) for $N = 6, 10, 15, 20, 25, 30$.

C. Stabilization Using Power Series-Based Approximate Continuum Kernels

We simulate the $n+1$ system with parameters (22). The $n+1$ hyperbolic PDE system, is approximated using finite differences with 256 grid points in x. For the backstepping control law, we employ the continuum kernels computed in Section VI-B for different orders of approximation N. Note that with parameters (23), it is verified in simulation that the open-loop system is unstable.

Fig. 2 shows the control inputs corresponding to the continuum kernels computed based on power series of order $N = 6, 10, 15, 20, 25$ and the (finite-difference approximation of the) solution to the $n+1$ kernel equations (1), (2). The controls start to diverge for $N = 6$, implying that the controls fail to stabilize the closed-loop system. This is expected based on the numerical experiments of Section VI-B, as the accuracy obtained with such a low-order approximation is not adequate. As the approximation order N increases, one can see that the controls tend to zero, implying that, for larger values of N, the controls are stabilizing. For the lower orders $N = 10$ and $N = 15$ the controls are distinguishable, and the decay rate is relatively slow, whereas for $N = 20$ and $N = 25$ the controls are virtually identical and converge to zero after two seconds, implying that all the state components have converged to zero by that time as well. This is consistent with the control gains displayed in Fig. 1, which are distinctly different for $N = 6, 10, 15$ and virtually identical for $N = 20, 25, 30$. Finally, we see that the approximate controls for $N = 20$ and $N = 25$ are also very close to the controls computed based on the solution to the $n+1$ kernel equations (1), (2), demonstrating efficient stabilization of the $n+1$ system using the approximate continuum kernels.

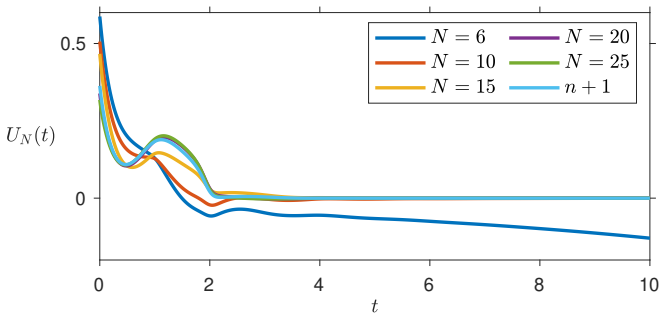

Fig. 2. The controls obtained with approximate continuum kernels computed with power series (9) of order $N = 6, 10, 15, 20, 25$ and by solving (1), (2) for the $n+1$ kernels.

VII. Conclusions

In this paper, we provided computational tools for construction of backstepping-based stabilizing kernels for continua of hyperbolic PDEs, and thus, also for large-scale PDE systems. These tools include both explicit kernels construction, and approximation via power series. We demonstrated the accuracy and efficiency of these tools on two numerical examples. We further demonstrated that the computed continuum kernels provided a stabilizing feedback law for the corresponding large-scale system of linear PDEs, with performance comparable to the case of employing the exact, large-scale control kernels. The computational aspects of the proposed methodology are considered in more detail in [22].

References

[1] M. Krstic and A. Smyshlyaev, *Boundary Control of PDEs: A Course on Backstepping Designs*, SIAM, 2008.

[2] V. Alleaume and M. Krstic, "Ensembles of hyperbolic PDEs: Stabilization by backstepping," *IEEE Trans. Automat. Control*, vol. 70, pp. 905–920, 2025.

[3] F. Di Meglio, R. Vazquez, and M. Krstic, "Stabilization of a system of $n+1$ coupled first-order hyperbolic linear PDEs with a single boundary input," *IEEE Trans. Automat. Control*, vol. 58, pp. 3097–3111, 2013.

[4] H. Yu and M. Krstic, "Output feedback control of two-lane traffic congestion," *Automatica*, vol. 125, pp. Paper No. 109379, 2021.

[5] J. Auriol and D. Bresch-Pietri, "Robust state-feedback stabilization of an underactuated network of interconnected $n+m$ hyperbolic PDE systems," *Automatica*, vol. 136, pp. 110040, 2022.

[6] J. Redaud, J. Auriol, and S.-I. Niculescu, "Output-feedback control of an underactuated network of interconnected hyperbolic PDE-ODE systems," *Syst. Control Lett.*, vol. 154, pp. 104984, 2021.

[7] J.-M. Coron, R. Vazquez, M. Krstic, and G. Bastin, "Local exponential H^2 stabilization of a 2×2 quasilinear hyperbolic system using backstepping," *SIAM J. Control Optim.*, vol. 51, pp. 2005–2035, 2013.

[8] L. Hu, F. Di Meglio, R. Vazquez, and M. Krstic, "Control of homodirectional and general heterodirectional linear coupled hyperbolic PDEs," *IEEE Trans. Automat. Control*, vol. 61, pp. 3301–3314, 2016.

[9] J. Wang and M. Krstic, "Event-triggered output-feedback backstepping control of sandwich hyperbolic PDE systems," *IEEE Trans. Automat. Control*, vol. 67, pp. 220–235, 2022.

[10] A. Diagne, M. Diagne, S. Tang, and M. Krstic, "Backstepping stabilization of the linearized *saint-venant-exner* model," *Automatica*, vol. 76, pp. 345–354, 2017.

[11] R. Vazquez and M. Krstic, "Marcum Q-functions and explicit kernels for stabilization of 2×2 linear hyperbolic systems with constant coefficients," *Syst. Control Lett.*, vol. 68, pp. 33–42, 2014.

[12] H. Anfinsen and O. M. Aamo, *Adaptive Control of Hyperbolic PDEs*, Springer, 2019.

[13] R. Vazquez, G. Chen, J. Qiao, and M. Krstic, "The power series method to compute backstepping kernel gains: theory and practice," in *IEEE Conf. Decis. Control*, 2023, pp. 8162–8169.

[14] J. Auriol, K. A. Morris, and F. Di Meglio, "Late-lumping backstepping control of partial differential equations," *Automatica*, vol. 100, pp. 247–259, 2019.

[15] J. Qi, J. Zhang, and M. Krstic, "Neural operators for PDE backstepping control of first-order hyperbolic PIDE with recycle and delay," *Syst. Control Lett.*, vol. 185, pp. 105714, 2024.

[16] L. Bhan, Y. Shi, and M. Krstic, "Neural operators for bypassing gain and control computations in PDE backstepping," *IEEE Trans. Automat. Control*, 2024.

[17] J.-P. Humaloja and N. Bekiaris-Liberis, "Stabilization of a class of large-scale systems of linear hyperbolic PDEs via continuum approximation of exact backstepping kernels," *IEEE Trans. Automat. Control*, pp. 1–16, 2025.

[18] J.-P. Humaloja and N. Bekiaris-Liberis, "On stabilization of large-scale systems of linear hyperbolic PDEs via continuum approximation of exact backstepping kernels," in *IEEE Conference on Decision and Control*, 2024, pp. 1974–1979.

[19] P. Ascencio, A. Astolfi, and T. Parisini, "Backstepping PDE design: a convex optimization approach," *IEEE Trans. Automat. Control*, vol. 63, pp. 1943–1958, 2018.

[20] R. Vazquez, J. Zhang, J. Qi, and M. Krstic, "Kernel well-posedness and computation by power series in backstepping output feedback for radially-dependent reaction-diffusion PDEs on multidimensional balls," *Syst. Control Lett.*, vol. 177, pp. 105538, 2023.

[21] X. Lin, R. Vazquez, and M. Krstic, "Towards a MATLAB Toolbox to compute backstepping kernels using the power series method," in *IEEE Conference on Decision and Control*, 2024, pp. 446–451.

[22] J-P. Humaloja and N. Bekiaris-Liberis, "On computation of approximate solutions to large-scale backstepping kernel equations via continuum approximation," *Syst. Control Lett.*, vol. 196, pp. 105982, 2025.