



Unified Integral Transforms Algorithm for Solving Multidimensional Nonlinear Convection-Diffusion Problems

Renato M. Cotta , Diego C. Knupp , Carolina P. Naveira-Cotta , Leandro A. Sphaier & João N. N. Quaresma

To cite this article: Renato M. Cotta , Diego C. Knupp , Carolina P. Naveira-Cotta , Leandro A. Sphaier & João N. N. Quaresma (2013) Unified Integral Transforms Algorithm for Solving Multidimensional Nonlinear Convection-Diffusion Problems, Numerical Heat Transfer, Part A: Applications, 63:11, 840-866, DOI: [10.1080/10407782.2013.756763](https://doi.org/10.1080/10407782.2013.756763)

To link to this article: <https://doi.org/10.1080/10407782.2013.756763>



Published online: 28 Feb 2013.



Submit your article to this journal [↗](#)



Article views: 172



Citing articles: 17 View citing articles [↗](#)

UNIFIED INTEGRAL TRANSFORMS ALGORITHM FOR SOLVING MULTIDIMENSIONAL NONLINEAR CONVECTION-DIFFUSION PROBLEMS

Renato M. Cotta¹, Diego C. Knupp¹,
Carolina P. Naveira-Cotta¹, Leandro A. Sphaier², and
João N. N. Quaresma³

¹Laboratory of Transmission and Technology of Heat, LTTC, Mechanical Engineering Department, Universidade Federal do Rio de Janeiro, POLI & COPPE/UFRJ, and Cidade Universitária, Rio de Janeiro, Brazil

²Laboratory of Theoretical and Applied Mechanics, LMTA, Mechanical Engineering Department, Universidade Federal Fluminense, PGMEC/UFF, Niterói, Brazil

³School of Chemical Engineering, Universidade Federal do Pará, FEQ/UFPA, Campus Universitário do Guamá, Belém, Brazil

The present work summarizes the theory and describes the algorithm related to an open-source mixed symbolic-numerical computational code named unified integral transforms (UNIT) that provides a computational environment for finding hybrid numerical-analytical solutions of linear and nonlinear partial differential systems via integral transforms. The reported research was performed by employing the well-established methodology known as the generalized integral transform technique (GITT), together with the symbolic and numerical computation tools provided by the Mathematica system. The main purpose of this study is to illustrate the robust precision-controlled simulation of multi-dimensional nonlinear transient convection-diffusion problems, while providing a brief introduction of this open source implementation. Test cases are selected based on nonlinear multidimensional formulations of Burgers' equation, with the establishment of reference results for specific numerical values of the governing parameters. Special aspects in the computational behavior of the algorithm are then discussed, demonstrating the implemented possibilities within the present version of the UNIT code, including the proposition of a progressive filtering strategy and a combined criteria reordering scheme, not previously discussed in related works, both aimed at convergence acceleration of the eigenfunction expansions.

Received 14 August 2012; accepted 17 November 2012.

The authors would like to acknowledge the financial support provided by CNPq and FAPERJ, both in Brazil.

Address correspondence to Renato M. Cotta, Mechanical Engineering Department, Universidade Federal do Rio de Janeiro, PEM/COPPE-UFRJ-Cx. Postal 68503, Rio de Janeiro 21945-970, Brazil. E-mail: cotta@mecanica.coppe.ufrj.br

NOMENCLATURE

c	nonlinear parameter in convection term	\mathbf{x}	position vector in general problem (1)
d	dissipation operator coefficient	x_1, x_2, x_3	space coordinates in solution (9)
F	filtering solution	y	transversal coordinate in problems (14, 15)
g	nonlinear source term	z	transversal coordinate in problems (14, 15)
K	diffusion operator coefficient	α	boundary condition coefficient
M	number of sub-regions in semi-analytical or Gaussian integration	β	boundary condition coefficient
n	number of coupled potentials in general problem (1)	ϕ_k	source terms in boundary conditions
N	truncation order in eigenfunction expansion	θ	filtered potential
t	dimensionless time variable	μ	eigenvalues
T	dimensionless potential in general problem (1); potential in Burgers' equation, Eqs. (14) and (15)	ν	diffusion coefficient in Burgers' equation
u	nonlinear function in convection term	ψ	eigenfunctions
u_0	linear parameter in nonlinear convection term	Subscripts & Superscripts	
w	transient operator coefficient	i, m, n, o	order of eigenquantities
x	longitudinal coordinate in problems (14, 15)	k	quantity corresponding to the equation of the k th potential
		-	integral transform
		\sim	normalized eigenfunction

1. INTRODUCTION

Analytic-based approaches for purely diffusive and convection-diffusion problems, despite the extensive progress achieved by discrete numerical methods, have been progressively advanced, unified, and further formalized by a few research groups, in part motivated by offering benchmark results for validation and calibration of numerical schemes. In addition, a number of hybrid analytical-numerical methodologies have appeared in the open literature, which attempt to combine classical analytical methods with modern computational tools, in the search for more accurate, robust and economical options to the nowadays well-established discrete solution methods. For instance, a hybrid method for solving diffusion and convection-diffusion problems that has been advanced along more than two decades is the so-called generalized integral transform technique (GITT) [1–9], based on the classical integral transform method for linear diffusion problems [10]. This hybrid approach was developed to overcome barriers posed by different classes of problems that were before supposed to be tractable solely by discrete-type methods, including situations involving nonlinear physical properties, moving boundaries, irregular geometries, and nonlinear convective terms, to name a few. The relative merits of such an approach over purely numerical procedures include the automatic global error control and the mild increase in computational effort for multidimensional nonlinear situations. The GITT hence complements the available numerical simulation tools, either as a companion in co-validation tasks, or as an alternative approach for analytically oriented users.

Hybrid methods become even more powerful and applicable when symbolic manipulation systems, which were also widely disseminated along the last two decades, are employed. The present algorithm has been implemented within a mixed symbolic-numerical environment, allowing for a marked reduction on formula derivation and analysis effort while providing integrated algorithms for numerical computations and post-processing analyses. One of the main goals of the current work comprised an effort to integrate the knowledge of the GITT application into a symbolical-numerical algorithm named unified integral transforms (UNIT) initially intended to bridge the gap between simple problems that allow for a straightforward analytical solution, and more complex and involved situations that almost unavoidably require specialized software systems. The open-source UNIT code is thus an implementation and development platform for researchers and engineers interested in solutions of convection-diffusion problems by integral transforms.

This contribution is part of an ongoing project entitled UNIT Project: Hybrid Methods in Engineering and Multiphysics, which is, in synthesis, a coordinated effort towards consolidating and constructing computational simulation tools based on hybrid numerical-analytical methodologies for multiphysics engineering problems. In this work, the UNIT methodology is demonstrated for nonlinear multi-dimensional convection-diffusion problems and special aspects on the computational behavior of the algorithm are discussed, focusing on illustrating the implemented features of the UNIT code in its present form. For instance, a progressive filtering procedure is formalized, which is applied to the presented test cases, and critically compared to a more usual Laplacian polynomial filtering. In addition, an alternative reordering scheme is implemented, based on combining estimates of the transformed initial conditions and source terms importance, and adding to the traditional squared eigenvalues criterion, to complement the convergence enhancement tools in the automatic code. Such aspects are then illustrated and critically discussed as applied to the nonlinear Burger's equation in three-, two- and one-dimensional formulations.

2. PROBLEM FORMULATION

In order to unify the classes of problems handled by the formal integral transforms procedure programmed into the UNIT platform, a transient convection-diffusion problem of n coupled potentials (for instance velocity, temperature, and/or concentrations) is considered. These potentials are defined in the region V , with boundary surface S and including non-linear coefficients and convective terms, incorporated into the source terms, as follows.

$$w_k(\mathbf{x}) \frac{\partial T_k(x, t)}{\partial t} = G_k(\mathbf{x}, t, \mathbf{T}), \quad \mathbf{x} \in V, t > 0, k = 1, 2, \dots, n \quad (1a)$$

with

$$G_k(\mathbf{x}, t, \mathbf{T}) = \nabla \cdot (K_k(\mathbf{x}) \nabla T_k(\mathbf{x}, t)) - d_k(\mathbf{x}) T_k(\mathbf{x}, t) + g_k(\mathbf{x}, t, \mathbf{T}) \quad (1b)$$

with initial and boundary conditions given, respectively, by

$$T_k(\mathbf{x}, 0) = f_k(\mathbf{x}), \quad \mathbf{x} \in V \quad (1c)$$

$$\left[\alpha_k(\mathbf{x}) + \beta_k(\mathbf{x})K_k(\mathbf{x}) \frac{\partial}{\partial \mathbf{n}} \right] T_k(\mathbf{x}, t) = \phi_k(\mathbf{x}, t, \mathbf{T}), \quad \mathbf{x} \in S, t > 0 \quad (1d)$$

where

$$\mathbf{T} = \{T_1, T_2, \dots, T_k, \dots, T_n\} \quad (1e)$$

where \mathbf{n} denotes the outward-drawn normal to the surface S .

Equation (1) is sufficiently general once all nonlinear and convection terms are grouped within the equations and boundaries source terms. For linear source terms, i.e., $g \equiv g(\mathbf{x}, t)$, and $\phi \equiv \phi(\mathbf{x}, t)$, this example becomes a class I linear diffusion problem according to the classification in reference [10], and exact analytical solutions are readily available via the classical integral transform technique. Otherwise, this problem is not *a priori* transformable, and the ideas in the generalized integral transform technique [1–9] can be utilized to develop hybrid numerical-analytical solutions, as now summarized.

3. FORMAL SOLUTION

Following the solution path previously established for convection-diffusion and purely diffusive nonlinear problems, the formal integral transform solution of the posed nonlinear problem requires the proposition of eigenfunction expansions for the associated potentials. The linear situation mentioned above, that allows for an exact solution via the classical integral transform approach, naturally leads to the eigenvalue problems to be preferred in the analysis of the nonlinear situation as well. They appear in the direct application of the separation of variables methodology to the linear homogeneous purely diffusive version of the above problem. Thus, the recommended set of uncoupled auxiliary problems is given by the following.

$$\nabla \cdot (K_k(\mathbf{x}) \nabla \psi_{ki}(\mathbf{x})) + [\mu_{ki}^2 w_k(\mathbf{x}) - d_k(\mathbf{x})] \psi_{ki}(\mathbf{x}) = 0, \quad \mathbf{x} \in V \quad (2a)$$

$$\left[\alpha_k(\mathbf{x}) + \beta_k(\mathbf{x})K_k(\mathbf{x}) \frac{\partial}{\partial \mathbf{n}} \right] \psi_{ki}(\mathbf{x}) = 0, \quad \mathbf{x} \in S \quad (2b)$$

where the eigenvalues, μ_{ki} , and related eigenfunctions, $\psi_{ki}(\mathbf{x})$, are assumed to be known from exact analytical expressions obtainable for instance through symbolic computation systems [11] or application of computational methods for Sturm-Liouville type problems [4, 10, 12–15]. In fact, the written form of Eq. (1) already reflects this choice of eigenvalue problems, Eq. (2), via prescription of the linear coefficients in both the equations and boundary conditions, since any remaining term is directly incorporated into the general nonlinear source terms without loss of generality. The problem indicated by Eq. (2) allows, through the associated orthogonality property of the eigenfunctions, the definition of the following integral transform pairs.

$$\bar{T}_{ki}(t) = \int_V w_k(\mathbf{x}) \tilde{\Psi}_{ki}(\mathbf{x}) T_k(\mathbf{x}, t) dV \quad \text{transforms} \quad (3a)$$

$$T_k(\mathbf{x}, t) = \sum_{i=1}^{\infty} \tilde{\Psi}_{ki}(\mathbf{x}) \bar{T}_{k,i}(t) \quad \text{inverses} \quad (3b)$$

where the symmetric kernels $\tilde{\Psi}_{ki}(\mathbf{x})$ are given by the following.

$$\tilde{\Psi}_{ki}(\mathbf{x}) = \frac{\Psi_{ki}(\mathbf{x})}{\sqrt{N_{ki}}}; \quad N_{ki} = \int_V w_k(\mathbf{x}) \Psi_{ki}^2(\mathbf{x}) dV \quad (3c, 3d)$$

with N_{ki} being the normalization integral.

The integral transformation of Eq. (1a) is accomplished by applying the operator $\int_V \tilde{\Psi}_{ki}(\mathbf{x})(\cdot) dV$ and making use of the boundary conditions given by Eqs. (1d) and (2b), yielding the following:

$$\frac{d\bar{T}_{ki}(t)}{dt} + \mu_{ki}^2 \bar{T}_{ki}(t) = \bar{g}_{ki}(t, \bar{\mathbf{T}}) + \bar{b}_{ki}(t, \bar{\mathbf{T}}), \quad i = 1, 2, \dots, t > 0, \quad k = 1, 2, \dots, n \quad (4a)$$

where the transformed source term $\bar{g}_{ki}(t, \bar{\mathbf{T}})$ is due to the integral transformation of the equation source term, and the other, $\bar{b}_{ki}(t, \bar{\mathbf{T}})$, is due to the contribution of the boundary source term.

$$\begin{aligned} \bar{g}_{ki}(t, \bar{\mathbf{T}}) &= \int_V \tilde{\Psi}_{ki}(\mathbf{x}) g_k(\mathbf{x}, t, \mathbf{T}) dV; \\ \bar{b}_{ki}(t, \bar{\mathbf{T}}) &= \int_S K_k(\mathbf{x}) \left[\tilde{\Psi}_{ki}(\mathbf{x}) \frac{\partial T_k(\mathbf{x}, t)}{\partial \mathbf{n}} - T_k(\mathbf{x}, t) \frac{\partial \tilde{\Psi}_{ki}(\mathbf{x})}{\partial \mathbf{n}} \right] ds \end{aligned} \quad (4b)$$

The boundary conditions contribution may also be expressed in terms of the boundary source terms, after manipulating Eqs. (1d) and (2b), to yield the following.

$$\bar{b}_{ki}(t, \bar{\mathbf{T}}) = \int_S \Phi_k(\mathbf{x}, t, \mathbf{T}) \left[\frac{\tilde{\Psi}_{ki}(\mathbf{x}) - K_k(\mathbf{x}) \frac{\partial \tilde{\Psi}_{ki}(\mathbf{x})}{\partial \mathbf{n}}}{\alpha_k(\mathbf{x}) + \beta_k(\mathbf{x})} \right] ds \quad (4c)$$

The initial conditions given by Eq. (1c) are transformed through the operator $\int_V w_k(\mathbf{x}) \tilde{\Psi}_{ki}(\mathbf{x})(\cdot) dV$, to provide the following.

$$\bar{T}_{ki}(0) = \bar{f}_{ki} \equiv \int_V w_k(\mathbf{x}) \tilde{\Psi}_{ki}(\mathbf{x}) f_k(\mathbf{x}) dV \quad (4d)$$

Equation (4) forms an infinite coupled system of nonlinear ordinary differential equations for the transformed potentials, $\bar{T}_{k,i}(t)$, which is unlikely to be analytically solvable. Nevertheless, reliable algorithms are readily available to numerically handle this ODE system, after truncation to a sufficiently large finite order. The *Mathematica* system [11] provides the routine *NDSolve* for solving stiff ODE systems

such as the one here obtained, under automatic absolute and relative error control. Once the transformed potentials have been numerically computed, the *Mathematica* routine automatically provides an interpolating function object that approximates the t variable behavior of the solution in a continuous form. Then, the inversion formula can be recalled to yield the potential field representation at any desired position \mathbf{x} and time t (or equivalent space coordinate).

4. PROGRESSIVE FILTERING SCHEME

The formal solution above derived provides the basic working expressions for the integral transform method. However, for an improved computational performance, it is always recommended to reduce the importance of the equation and boundary source terms so as to enhance the eigenfunction expansions convergence behavior [6].

One possible approach for achieving this goal is the proposition of analytical filtering solutions, which essentially remove information from the source terms into a desirably simple analytical expression. Several different alternative filters may be proposed for the same problem, and the user experience may be helpful in finding the right combination of analytical involvement and numerical improvement. Thus, the solution is in general proposed as follows.

$$T_k(\mathbf{x}, t) = F_k(\mathbf{x}; t) + \theta_k(\mathbf{x}, t) \tag{5}$$

Where the variable t is a parameter in the filter proposition, $F_k(\mathbf{x}, t)$.

The net effect of the filter is to provide a new filtered problem, with reduced importance of the original problem source terms, written as follows

$$w_k(\mathbf{x}) \frac{\partial \theta_k(\mathbf{x}, t)}{\partial t} = \nabla \cdot (K_k(\mathbf{x}) \nabla \theta_k(\mathbf{x}, t)) - d_k(\mathbf{x}) \theta_k(\mathbf{x}, t) + g_k^*(\mathbf{x}, t, \mathbf{T}), \mathbf{x} \in V, t > 0 \tag{6a}$$

Where the filtered equation source term is given by the following.

$$g_k^*(\mathbf{x}, t, \mathbf{T}) = g_k(\mathbf{x}, t, \mathbf{T}) + \nabla \cdot (K_k(\mathbf{x}) \nabla F_k(\mathbf{x}, t)) - d_k(\mathbf{x}) F_k(\mathbf{x}, t) - w_k(\mathbf{x}) \frac{\partial F_k(\mathbf{x}, t)}{\partial t} \tag{6b}$$

with initial and boundary conditions,

$$\theta_k(\mathbf{x}, 0) = f_k^*(\mathbf{x}) = f_k(\mathbf{x}) - F_k(\mathbf{x}, 0), \mathbf{x} \in V \tag{6c}$$

$$\alpha_k(\mathbf{x}) \theta_k(\mathbf{x}, t) + \beta_k(\mathbf{x}) K_k(\mathbf{x}) \frac{\partial \theta_k(\mathbf{x}, t)}{\partial \mathbf{n}} = \phi_k^*(\mathbf{x}, t, \mathbf{T}), \mathbf{x} \in S \tag{6d}$$

Where the filtered boundary source term is rewritten as follows.

$$\phi_k^*(\mathbf{x}, t, \mathbf{T}) = \phi_k(\mathbf{x}, t, \mathbf{T}) - \alpha_k(\mathbf{x}) F_k(\mathbf{x}, t) - \beta_k(\mathbf{x}) K_k(\mathbf{x}) \frac{\partial F_k(\mathbf{x}, t)}{\partial \mathbf{n}} \tag{6e}$$

Thus, the previously established formal solution is directly applicable to the above filtered problem given by Eq. (6) once the initial conditions, the filtered equation, and boundary source terms have been adequately substituted.

The UNIT code allows for different choices of filtering solutions, either user provided or automatically determined from the symbolic computation feature of the *Mathematica* system. For instance, in the one-dimensional situation, the simplest possible filtering solution is written as a linear function in the space variable that simultaneously satisfies both boundary conditions, i.e.,

$$F_k(x; t) = a_{1,k}(t)x + a_{0,k}(t) \quad (7)$$

For second kind boundary conditions on both sides, one may just multiply the two terms in the above filter expression by x , in order to allow for satisfaction of both prescribed derivatives.

For the two- and three-dimensional cases the definition of such a simple general function that homogenizes all four or six boundary conditions simultaneously becomes less trivial, although still feasible, for instance, through multiplication and combination of powers of the independent variables. Since the filtering of the boundary conditions is a very important step in enhancing the convergence of solutions obtained by means of GITT, it is quite desirable that the UNIT code offers an automatic filtering option for multidimensional problems, similarly to the one-dimensional situation, since the user might not have enough experience to propose an appropriate specific filtering in such cases.

In this context, an automatic filtering alternative is proposed and implemented in the UNIT code, known as progressive linear filtering. The basic idea behind the progressive filtering approach is that two or three-dimensional problems may be partially transformed in each spatial direction successively until all spatial directions have been transformed. So, it is possible to filter the boundary conditions of the first spatial direction to be transformed using any simple analytical function that satisfies both boundary conditions with respect to that spatial direction, and once more, the simplest possibility is a linear function such as the one given by Eq. (7). Once the first spatial direction has been integral transformed, it is possible to employ the same procedure in this first transformed field in order to filter the boundary conditions of the next spatial direction to be transformed. This successive procedure goes on until all spatial directions have been transformed. In this case, assuming that one is dealing with a general three-dimensional problem with space variables $\mathbf{x} = \{x_1, x_2, x_3\}$, bounded by $x_{1,i} \leq x_1 \leq x_{1,f}$, $x_{2,i} \leq x_2 \leq x_{2,f}$, and $x_{3,i} \leq x_3 \leq x_{3,f}$, the eigenfunctions and weighting functions can be written as follows:

$$\tilde{\Psi}_{kmno}(\mathbf{x}) = \tilde{\Omega}_{km}(x_1)\tilde{\xi}_{kn}(x_2)\tilde{\chi}_{ko}(x_3); \quad w_k(\mathbf{x}) = w_{1k}(x_1)w_{2k}(x_2)w_{3k}(x_3) \quad (8a, 8b)$$

where, $\tilde{\Omega}_{km}(x_1)$, $w_{1k}(x_1)$; $\tilde{\xi}_{kn}(x_2)$, $w_{2k}(x_2)$, and $\tilde{\chi}_{ko}(x_3)$, $w_{3k}(x_3)$ correspond to the eigenfunctions and weighting functions in the first, second and third independent variables and corresponding integral transform pairs, respectively.

The filtering proposal is then given by the following.

$$\begin{aligned}
 T_k(\mathbf{x}, t) &= F_{1,k}(x_1; t) + \sum_{m=1}^{\infty} \bar{F}_{2,km}(x_2; t) \tilde{\Omega}_{km}(x_1) \\
 &+ \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \bar{\bar{F}}_{3,kmn}(x_3; t) \tilde{\Omega}_{km}(x_1) \tilde{\xi}_{kn}(x_2) + \\
 &+ \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \sum_{o=1}^{\infty} \bar{\theta}_{kmno}(t) \tilde{\Omega}_{km}(x_1) \tilde{\xi}_{kn}(x_2) \tilde{\chi}_{ko}(x_3)
 \end{aligned} \tag{9a}$$

Which may be rewritten in the more adequate single summation form as follows.

$$T_k(\mathbf{x}, t) = F_k(\mathbf{x}; t) + \sum_{i=1}^{\infty} \bar{\theta}_{ki}(t) \tilde{\Psi}_{ki}(\mathbf{x}) \tag{9b}$$

where

$$F_k(\mathbf{x}; t) = F_{1,k}(x_1; t) + \sum_{m=1}^{\infty} \bar{F}_{2,km}(x_2; t) \tilde{\Omega}_{km}(x_1) + \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \bar{\bar{F}}_{3,kmn}(x_3; t) \tilde{\Omega}_{km}(x_1) \tilde{\xi}_{kn}(x_2) \tag{9c}$$

Defining the operator \mathcal{B}_k as,

$$\mathcal{B}_k(\cdot) \equiv \alpha_k(\mathbf{x})(\cdot) + \beta_k(\mathbf{x}) K_k(\mathbf{x}) \frac{\partial(\cdot)}{\partial \mathbf{n}} \tag{9d}$$

Where $F_{1,k}(x_1; t)$, $\bar{F}_{2,km}(x_2; t)$, and $\bar{\bar{F}}_{3,kmn}(x_3; t)$ are linear functions (or quadratic in case of both boundary conditions in that coordinate being of the second kind) that satisfy the following.

$$\mathcal{B}_k F_{1,k}(x_1; t) = \phi_k(x_1, x_2, x_3, t) \text{ at } x_1 = x_{1,i} \text{ and } x_1 = x_{1,f} \tag{9e}$$

$$\mathcal{B}_k \bar{F}_{2,km}(x_2; t) = \bar{\Phi}_{km}(x_2, x_3, t) \text{ at } x_2 = x_{2,i} \text{ and } x_2 = x_{2,f} \tag{9f}$$

$$\mathcal{B}_k \bar{\bar{F}}_{3,kmn}(x_3; t) = \bar{\bar{\Phi}}_{kmn}(x_3, t) \text{ at } x_3 = x_{3,i} \text{ and } x_3 = x_{3,f} \tag{9g}$$

with

$$\bar{\Phi}_{km}(x_2, x_3, t) = \int_{x_{1,i}}^{x_{1,f}} w_{1,k}(x_1) \tilde{\Omega}_m(x_1) [\phi_k(x_1, x_2, x_3, t) - \mathcal{B}_k F_{1,k}(x_1; t)] dx_1 \tag{9h}$$

$$\bar{\bar{\Phi}}_{kmn}(x_3, t) = \int_{y_{1,i}}^{y_{1,f}} w_{2,k}(x_2) \tilde{\xi}_n(x_2) [\bar{\Phi}_{km}(x_2, x_3, t) - \mathcal{B}_k \bar{F}_{2,km}(x_2; t)] dx_2 \tag{9i}$$

5. COMPUTATIONAL ALGORITHM

The constructed UNIT code in the *Mathematica* platform [11] encompasses all of the symbolic derivations that are required in the above GITT formal solution, besides the numerical computations that are required in the solutions of the chosen eigenvalue problem and the transformed ODE system. The user essentially needs to specify the problem formulation, according to Eq. (1) together with the required problem parameters, solve the problem using the provided UNIT algorithm, and then choose how to present the results according to the specific needs.

In order to computationally solve the problem defined by Eqs. (1), a straightforward general algorithm can be described as follows.

- The user provides the input and problem formulation module, which includes the equation and boundary condition coefficients in Eq. (1), besides the corresponding source terms, G and ϕ . As the problem is formulated, with particular expressions for w_k , K_k , d_k , α_k , and β_k , there is an implicit choice of the eigenvalue problem then proposed.
- The automatic filtering module is then activated, or the filter is provided as a problem formulation by the user, to be eventually handled via symbolic computation. The option of not providing a filtering solution is also allowed, either because it might not be actually necessary or as a solution strategy to be complemented by the user, such as by an integral balance acceleration *a posteriori* [6].
- The auxiliary eigenvalue problem (2) is solved for the eigenvalues and related normalized eigenfunctions, either in analytic explicit form, when applicable, as obtained for instance by the symbolic routine *DSolve* [11], after separation of variables in multidimensional applications, or through the GITT itself [4, 12–15].
- The transformed initial conditions are computed, either analytically (function *Integrate* [11]) or with a general-purpose procedure through adaptive numerical integration (function *NIntegrate* [11]). Two additional options are here provided to the user; namely, a semi-analytical evaluation where the analytical integration of the eigenfunction oscillatory behavior is preserved [16, 17], and a simplified and cost-effective numerical integration with Gaussian quadrature, automatically exploiting the frequency of oscillation of the eigenfunctions in the choice of subintervals for integration. Similarly, the transformed source term in the transformed ODE system in Eq. (4a), once not dependent on the transformed potentials, can be evaluated in advance. For the more general situation of nonlinear coefficients, there are some computational savings in grouping them into a single integrand, as represented in Eq. (4b). The integration can be obtained analytically, if feasible, or again by the automatic Gaussian quadrature scheme that accounts for the information regarding the eigenfunctions' oscillatory behavior. The alternative semi-analytical integration procedure is also implemented, which is particularly convenient in nonlinear formulations that might require costly numerical integration. For instance, the integral transformation of the equation source term would then be evaluated as follows:

$$\bar{g}_{ki}^*(t, \mathbf{T}) = \int_V \tilde{\Psi}_{ki}(\mathbf{x}) g_k^*(\mathbf{x}, t, \mathbf{T}) dV = \sum_{m=1}^M \int_{V_m} \tilde{\Psi}_{ki}(\mathbf{x}) \hat{g}_{k,m}^*(\mathbf{x}, t, \mathbf{T}) dV_m \quad (10)$$

where $\hat{g}_{k,m}^*(\mathbf{x}, t, \mathbf{T})$ are simpler representations of the source term, defined in sub-regions V_m , for which analytical integration of the eigenfunctions is still obtainable. The simplest choice would be the adoption of uniform values of the source terms within the subdomains (zeroth order approximation), but linear and quadratic representations of the source terms behavior are also implemented [16, 17].

- The truncated ODE system (4) is then numerically solved through the *NDSolve* routine of the *Mathematica* system [11]. In general, such initial value problem solvers are recommended to work under the automatic selection of a stiff system situation, such as with the BDF method [11], since the resulting system is likely to become stiff, especially for increasing truncation orders in the expansion. This subroutine offers an interesting combination of accuracy control, simplicity in use, and reliability.
- Once all the intermediate numerical tasks are accomplished within user-prescribed accuracy, one is left with the need of reaching convergence in the eigenfunction expansions and controlling the truncation order N for the requested accuracy in the final solution. The analytic nature of the inversion formula allows for a direct testing procedure at each specified position within the medium where a solution is desired, and then the truncation order N can be increased (or eventually decreased), to fit the user global error requirements over the entire solution domain. The simple tolerance testing formula employed is written as follows.

$$\varepsilon = \max_{\mathbf{x} \in V} \left| \frac{\sum_{i=N^*}^N \tilde{\Psi}_{ki}(\mathbf{x}) \bar{\theta}_{ki}(t)}{F_k(\mathbf{x}; t) + \sum_{i=1}^N \tilde{\Psi}_{ki}(\mathbf{x}) \bar{\theta}_{ki}(t)} \right| \tag{11}$$

where F_k is the filtering solution. Therefore, this testing can be implemented by choosing the value of N^* so as to have a small odd number of terms in the numerator sum, then offering error estimations at any of the selected test positions within the domain.

6. REORDERING SCHEMES

In multidimensional applications, the final integral transform solution for the related potential is expressed as double or triple infinite summations for two or three-dimensional transient problems, respectively. From a computational point of view, only a truncated version of such nested summations can be actually evaluated. However, if one just truncates each individual summation to a certain prescribed finite order, computations become quite ineffective, and even risky. By following this path some still important information to the final result can be disregarded due to the fixed summations limits, while other terms are accounted for that have essentially no contribution to convergence of the potential in the relative accuracy required. Therefore, for an efficient computation of these expansions, the infinite multiple summations should first be converted to a single sum representation with the appropriate reordering of terms according to their individual contribution to the final

numerical result [6, 18, 19]. Then, it becomes possible to evaluate the minimal number of eigenvalues and related derived quantities required to reach the user-prescribed accuracy target [6]. Since the final solution is not, of course, known *a priori*, the parameter which shall govern this reordering scheme must be chosen with care. Once the reordering is completed, the remainder of the computational procedure becomes as straightforward and cost-effective as in the one-dimensional case. The most common choice of reordering strategy is based on arranging in increasing order the sum of the squared eigenvalues in each spatial coordinate, which is implemented as a default in the UNIT code and offers a good compromise between the overall convergence enhancement and simplicity in use. However, individual applications may require more elaborate reordering that accounts for the influence of transformed initial conditions and transformed nonlinear source terms in the ODE system.

To more clearly understand the possible reordering schemes, let us start from the formal solution of the transformed potentials equations (4), which is written as follows.

$$\bar{T}_{ki}(t) = \bar{f}_{ki} \exp(-\mu_{ki}^2 t) + \int_0^t \hat{\mathbf{g}}_{ki}(t', \bar{\mathbf{T}}) \exp[-\mu_{ki}^2(t-t')] dt' \quad (12a)$$

Where the nonlinear transformed source term $\hat{\mathbf{g}}_{ki}(t, \bar{\mathbf{T}})$ includes the contributions of both the equation and boundary conditions source terms.

$$\hat{\mathbf{g}}_{ki}(t, \bar{\mathbf{T}}) = \bar{\mathbf{g}}_{ki}(t, \bar{\mathbf{T}}) + \bar{\mathbf{b}}_{ki}(t, \bar{\mathbf{T}}) \quad (12b)$$

Integration by parts of Eq. (12a) provides an alternative expression that allows the understanding of the influence of the transformed initial conditions and source terms in the choice of terms reordering in rewriting the multiple series as a single one.

$$\begin{aligned} \bar{T}_{ki}(t) = & \bar{f}_{ki} \exp(-\mu_{ki}^2 t) + \frac{1}{\mu_{ki}^2} [\hat{\mathbf{g}}_{ki}(t, \bar{\mathbf{T}}) - \hat{\mathbf{g}}_{ki}(0, \bar{\mathbf{T}}) \exp(-\mu_{ki}^2 t)] \\ & - \frac{1}{\mu_{ki}^2} \int_0^t \frac{d\hat{\mathbf{g}}_{ki}}{dt'} \exp[-\mu_{ki}^2(t-t')] dt' \end{aligned} \quad (13)$$

It is evident that the squared eigenvalues, which involve the combination of the eigenvalues in each spatial coordinate, play the most important role in the decay of the absolute values of the infinite summation components, both through the exponential term $\exp(-\mu_{ki}^2 t)$ and, at a lower convergence rate, through the inverse of the squared eigenvalues, $1/\mu_{ki}^2$. Therefore, this traditionally employed reordering scheme should be able to account for the most important terms in the adequate reordering of the expansion. Nevertheless, supposing that the last integral term in Eq. (13) plays a less important role in the reordering choice, and in fact it vanishes when the source term is not time dependent, one concludes that the decay of the transformed initial condition and the transformed source term evolution from its initial value, play a complementary role in the selection of terms in the eigenfunction

expansion for a fixed truncation order. Thus, a more robust selection can be proposed, based on adding to the initially reordered terms, according to the squared eigenvalues criterion, a few extra terms that are of possible contribution to the final result under the analysis of the initial condition decay and/or the transformed source term behavior. In the first case, for the lowest time value of interest, $t = t_{min}$, the criterion that reorders the terms based on the decay of the initial conditions is based on sorting in decreasing order from the expression $\tilde{f}_{ki} \exp(-\mu_{ki}^2 t_{min})$. In the second case, for the general case of a nonlinear transformed source term, the estimation of the terms importance is more difficult, since the source terms are not known *a priori*. One possible approach is to consider the limiting situation of an uniform unitary source term, representing for instance its normalized maximum value, and analyzing the reordering of terms in descending absolute value based on the expression $\frac{1}{\mu_{ki}^2} \int_V \tilde{\phi}_{ki}(\mathbf{x}) dV$. Therefore, combining the three criteria, and eliminating the duplicates with respect to the traditional reordering scheme based on the squared eigenvalues, a few extra terms are added to the initially reordered terms that may have still some relevant effect in the final truncated summation, as will later on be illustrated.

7. APPLICATION

In order to illustrate the methodology and the behavior of the developed implementation, a test case problem based on the nonlinear Burgers' equation formulation is selected, starting from its three-dimensional version. Then, to demonstrate the unifying feature of the UNIT code and inspect the different behavior in each formulation level, the 2-D and 1-D versions were also investigated, which are obtained by simply successively dropping the z and y coordinates in the input module of the UNIT code. The mathematical formulation of the multidimensional nonlinear convection-diffusion problem here considered is given by the following

$$\frac{\partial T(x, y, z, t)}{\partial t} + u(T) \frac{\partial T(x, y, z, t)}{\partial x} = \nu \left[\frac{\partial^2 T(x, y, z, t)}{\partial x^2} + \frac{\partial^2 T(x, y, z, t)}{\partial y^2} + \frac{\partial^2 T(x, y, z, t)}{\partial z^2} \right]$$

$$0 < x < 1, 0 < y < 1, 0 < z < 1, t > 0$$
(14a)

with initial and boundary conditions given by the following.

$$T(x, y, z, 0) = 1, 0 \leq x \leq 1, 0 \leq y \leq 1, 0 \leq z \leq 1 \quad (14b)$$

$$T(0, y, z, t) = 0; T(1, y, z, t) = 0, t > 0 \quad (14c, 14d)$$

$$\frac{\partial T(x, 0, z, t)}{\partial y} = 0; T(x, 1, z, t) = 0, t > 0 \quad (14e, 14f)$$

$$\frac{\partial T(x, y, 0, t)}{\partial z} = 0; T(x, y, 1, t) = 0, t > 0 \quad (14g, 14h)$$

For the present application, the nonlinear function $u(T)$ is taken as follows.

$$u(T) = u_0 + cT \quad (14i)$$

This specific example allows for analytical integration of the coefficients in the transformed system, as presented in references [3, 4], which then provides the most cost-effective GITT solution to this problem, with very accurate predictions, as will be shown. However, since the objective is to demonstrate the performance of the more general and unifying features of the developed code, the focus is shifted towards the illustration of the automatic and cost effective options of the UNIT code, such as the integration schemes, the semi-analytical integration and the Gaussian quadrature, such as the boundary conditions filters, the progressive filtering and the Laplacian polynomial filter, and such as the reordering schemes, the traditional squared eigenvalues criterion and the combined criterion. With respect to the coefficients integration schemes, both options work with an automatic selection of sub-regions, based on the known oscillatory behavior of the eigenfunctions. With respect to the semi-analytical integration scheme, a zeroth order level of approximation of the source term was employed, and the results were compared with those of the Gaussian quadrature scheme.

In order to illustrate the implemented automatic progressive linear filtering for multidimensional applications, we also investigate the same problem but with non-homogeneous boundary conditions, as follows.

$$\frac{\partial T(x, y, z, t)}{\partial t} + u(T) \frac{\partial T(x, y, z, t)}{\partial x} = \nu \left[\frac{\partial^2 T(x, y, z, t)}{\partial x^2} + \frac{\partial^2 T(x, y, z, t)}{\partial y^2} + \frac{\partial^2 T(x, y, z, t)}{\partial z^2} \right] \quad (15a)$$

$$0 < x < 1, 0 < y < 1, 0 < z < 1, t > 0$$

$$T(x, y, z, 0) = 1, 0 \leq x \leq 1, 0 \leq y \leq 1, 0 \leq z \leq 1 \quad (15b)$$

$$T(0, y, z, t) = 1; T(1, y, z, t) = 0, t > 0 \quad (15c, 15d)$$

$$\frac{\partial T(x, 0, z, t)}{\partial y} = 0; T(x, 1, z, t) = 0, t > 0 \quad (15e, 15f)$$

$$\frac{\partial T(x, y, 0, t)}{\partial z} = 0; T(x, y, 1, t) = 0, t > 0 \quad (15g, 15h)$$

where $u(T)$ is again the function given by Eq. (14i).

The inherent choice of the eigenvalue problem is made when the user establishes the equivalence between the general problem, given by Eq. (1), and the considered application. For the first application considered in this work, given by Eq. (14), the UNIT code parameters were taken as follows.

$$\begin{aligned} \mathbf{x} &\rightarrow \{x, y, z\}; n \rightarrow 1; T_k(\mathbf{x}, t) \rightarrow T(x, y, z, t); f_k(\mathbf{x}) \rightarrow 1; w_k(\mathbf{x}) \rightarrow 1; K_k(\mathbf{x}) \rightarrow \nu. \\ d_k(\mathbf{x}) &\rightarrow 0; g_k(\mathbf{x}, t, T_l) \rightarrow -u(T) \frac{\partial T(x, y, z, t)}{\partial x}; \phi_k(\mathbf{x}, t, T_l) \rightarrow 0. \end{aligned} \quad (16a - 16i)$$

$$\begin{aligned} \alpha_k(\mathbf{x}) &\rightarrow 1; \beta_k(\mathbf{x}) \rightarrow 0, \text{ for } x = 0. \\ \alpha_k(\mathbf{x}) &\rightarrow 0; \beta_k(\mathbf{x}) \rightarrow 1, \text{ for } y, z = 0. \\ \alpha_k(\mathbf{x}) &\rightarrow 1; \beta_k(\mathbf{x}) \rightarrow 0, \text{ for } x, y, z = 1 \end{aligned} \quad (16j - 16o)$$

Whereas for the nonhomogeneous example given by Eq. (15), the source terms are given by the following.

$$\begin{aligned} \phi_k(\mathbf{x}, t, T_l) &\rightarrow 1, \text{ for } x = 0 \\ \phi_k(\mathbf{x}, t, T_l) &\rightarrow 0, \text{ for } y, z = 0, x, y, z = 1 \end{aligned} \quad (17a, 17b)$$

8. RESULTS AND DISCUSSION

The UNIT code is here illustrated through its version 2.2.3 (available from the site <http://2009unit.vndv.com>), in *Mathematica* 7.0, which is the presently employed version, and has the following main features.

- System of linear or nonlinear equations (parabolic problems, parabolic-hyperbolic problems, or elliptic problems via pseudo-transient).
- Multidimensional transient formulations.
- Eigenvalue problem analytically solved via *DSolve* routine (Sturm-Liouville problem).
- Transformed coefficients determined by semi-analytical integration (zeroth order), numerical integration (Gaussian quadrature or *NIntegrate* routine) or analytical integration (*Integrate* routine or user supplied).
- User defined or automatic progressive linear filtering.
- Reordering by squared eigenvalues criterion or combination of transformed initial conditions, transformed source term, and squared eigenvalues.
- Non-homogeneous term via Green's 2nd formula.
- Error estimator.

The analysis that will be shown below for the applications detailed in section 7, adopted the following governing parameters values: $u_0 = 1$, $c = 5$, and $\nu = 1$.

The first investigated application, presented in section 7, is given by Burgers' equation formulation with homogenous boundary conditions, in Eq. (9), starting from its three-dimensional version. Then, in order to demonstrate the unifying features of the UNIT code and inspect the different behavior for each formulation level, the 2-D and 1-D versions were also investigated, by simply successively dropping the z and y coordinates, respectively, just changing the statement of the problem at the input module.

In order to offer a benchmark solution for the three dimensional nonlinear Burgers' equation, the UNIT code was first employed with user provided analytical integration of the transformed initial condition and nonlinear source term, and double checked against a dedicated GITT solution of the same problem implemented in Fortran 95. Then, high truncation orders could be achieved with reduced computational cost so as to offer a reliable numerical solution with four fully converged significant digits, as illustrated in Table 1a, where convergence is reached at truncation orders around $N=300$ with the traditional squared eigenvalues reordering criterion.

Table 1b illustrates a comparison of the UNIT code solution for the three-dimensional formulation, with different number of terms in the eigenfunction expansion ($N=45$ and 70) with automatic reordering through the squared eigenvalues criterion, and fixed values for the number of points in the Gaussian integration in each spatial direction ($M=\{12, 12, 12\}$), so as to allow for a comparison just in terms of the truncation orders in the eigenfunction expansions. It is then shown the UNIT solution, starting from a fixed truncation order $N=45$, with the automatic combined reordering scheme, which includes another 24 terms in the final reordering, bringing the truncation order up to $N=69$. Also, a numerical solution obtained by the Method of Lines implemented in the *NDSolve* routine of the *Mathematica* system is shown, not in default mode but controlling the maximum step size in each independent variable to achieve comparable accuracy as the UNIT code, together with the benchmark GITT solution with $N=300$, fully converged to four significant digits, as shown in the previous table. It should be noticed that at the selected positions, all the UNIT results are converged at least to the second significant digit for $t=0.02$ and to the third significant digit for $t=0.1$, thus with the expected improved convergence behavior for larger values of t . At the same time, they agree to at least the same significant digit with the GITT benchmark solution, in a comparable behavior with the numerical solution from the *NDSolve* routine. There is some slight improvement in taking the combined reordering solution with $N=69$ instead of the default solution with $N=70$ terms with the squared eigenvalues reordering

Table 1a. Convergence of GITT solution for three-dimensional Burgers' equation ($N=40$ to 300 terms and user provided analytical integration)

x	$N=40$	$N=70$	$N=100$	$N=140$	$N=180$	$N=220$	$N=260$	$N=280$	$N=300$
$t=0.02, y=0.5, \text{ and } z=0.5$									
0.1	0.2601	0.2868	0.2868	0.2782	0.2814	0.2813	0.2780	0.2797	0.2798
0.3	0.7416	0.7349	0.7361	0.7402	0.7414	0.7415	0.7398	0.7395	0.7396
0.5	0.9208	0.9355	0.9361	0.9351	0.9331	0.9329	0.9330	0.9331	0.9331
0.7	0.9208	0.9157	0.9151	0.9128	0.9140	0.9141	0.9155	0.9157	0.9156
0.9	0.4815	0.4857	0.4858	0.4893	0.4898	0.4898	0.4905	0.4905	0.4905
$t=0.1, y=0.5, \text{ and } z=0.5$									
0.1	0.04898	0.04981	0.04983	0.04920	0.04938	0.04938	0.04923	0.04922	0.04922
0.3	0.1491	0.1491	0.1491	0.1493	0.1495	0.1495	0.1492	0.1492	0.1492
0.5	0.2221	0.2233	0.2233	0.2231	0.2231	0.2231	0.2231	0.2231	0.2231
0.7	0.2237	0.2236	0.2235	0.2231	0.2233	0.2233	0.2235	0.2235	0.2235
0.9	0.1017	0.1024	0.1024	0.1028	0.1030	0.1030	0.1031	0.1031	0.1031

Table 1b. Comparison of UNIT code solutions for three-dimensional Burgers' equation with different reordering schemes (GITT with squared eigenvalues reordering, $N=45$ and 70 terms and $M = \{12, 12, 12\}$ in Gaussian integration, and GITT with $N=69$ terms in combined reordering)

x	UNIT* $N=45$	UNIT* $N=70$	UNIT** $N=69$	$NDSolve$	GITT+ $N=300$
$t = 0.02, y = 0.5, \text{ and } z = 0.5$					
0.1	0.2626	0.2874	0.2859	0.2804	0.2798
0.3	0.7439	0.7363	0.7371	0.7403	0.7396
0.5	0.9250	0.9349	0.9361	0.9334	0.9331
0.7	0.9243	0.9164	0.9159	0.9159	0.9156
0.9	0.4806	0.4836	0.4859	0.4911	0.4905
$t = 0.1, y = 0.5 \text{ and } z = 0.5$					
0.1	0.04922	0.04996	0.05017	0.04947	0.04922
0.3	0.1496	0.1495	0.1497	0.1494	0.1492
0.5	0.2227	0.2238	0.2239	0.2232	0.2231
0.7	0.2239	0.2238	0.2240	0.2236	0.2235
0.9	0.1016	0.1023	0.1024	0.1031	0.1031

(*)UNIT code with Gaussian integration $\{12, 12, 12\}$ points and squared eigenvalues reordering.

(**)UNIT code with Gaussian integration $\{12, 12, 12\}$ points and combined reordering.

(+)UNIT code with user provided analytical integration.

criterion. Nevertheless, at least in the present example and with the adopted truncation order of $N = 45$, the improvement seems to be concentrated at the lower values of the t variable. This observation can be confirmed with the aid of Figures 1a and 1b, where it is shown the variation of the transformed potentials along the t variable. Figure 1a shows the t evolution of the transformed potentials originally selected by the squared eigenvalues reordering criterion, from $N=1$ to 45 , while Figure 1b shows the transformed potentials added by the combination of the other two selection schemes, the decay of the transformed initial condition and the normalized transformed source term, from $N=46$ to 69 . It can be observed that the largest terms in the first criterion are at least one order of magnitude larger than those from the second group, and one may observe the fast decay of the added terms as t increases, from Figure 1b, showing that the contribution of the added terms after the combined reordering, is indeed concentrated within the lower range of the t variable, thus not so evidently relevant in Table 1b above.

Table 1c brings a comparison of the automatically generated results for $N = 45$, with zeroth order semi-analytical integration, with a fixed number of sub-regions $M = \{16,16,16\}$, and those also obtained from the UNIT code solution with Gaussian quadrature, with the same number of quadrature points as sub-regions, against the results obtained with the fully converged solution through GITT with analytical integration and $N = 300$. A column with error estimates automatically provided by the UNIT code, based on the last three terms of the expansions, is also presented. After observing the results, one may conclude that the two sets of results obtained with the UNIT code are in agreement with the benchmark results to at least two significant digits for $t = 0.02$, and to three significant digits for $t = 0.1$, as practically predicted by the error estimates provided. The error estimation through the proposed straightforward testing formula provides reasonable predictions, indicating a

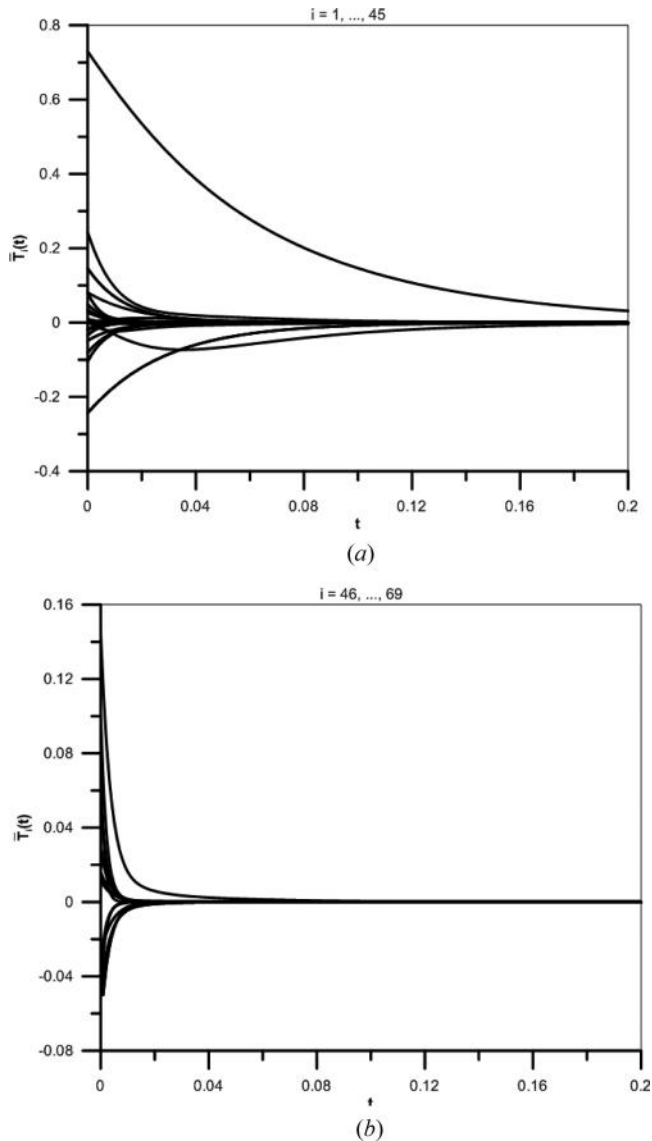


Figure 1. Evolution of the transformed potentials in UNIT solution for the three-dimensional Burgers equation. (a) Orders 1–45, selected from default squared eigenvalues reordering scheme; and (b) orders 46–69, added from the optional combined reordering scheme.

maximum relative error of around 4×10^{-3} for the worst tabulated case at $t=0.02$, thus possibly affecting the second digit to plus or minus one, as confirmed by the benchmark results. These results also show that in this particular case there is no need to increase the number of terms, for instance from three to five terms, in the estimation of the residues.

Table 1c. Comparison of UNIT code solutions with different integration schemes for the three-dimensional Burgers' equation and evaluation of error estimation procedure

x	UNIT ^a	Error estimate ^b	UNIT ^c	Error estimate ^d	GITT ^e
$t = 0.02$					
0.1	0.2615	2.47×10^{-3}	0.2600	2.51×10^{-3}	0.2798
0.3	0.7436	3.56×10^{-3}	0.7432	3.57×10^{-3}	0.7396
0.5	0.9250	4.32×10^{-3}	0.9250	4.33×10^{-3}	0.9331
0.7	0.9242	2.86×10^{-3}	0.9241	2.87×10^{-3}	0.9156
0.9	0.4813	1.34×10^{-3}	0.4823	1.35×10^{-3}	0.4905
$t = 0.1$					
0.1	0.04914	3.44×10^{-4}	0.04902	3.58×10^{-4}	0.04922
0.3	0.1494	6.52×10^{-5}	0.1491	6.77×10^{-5}	0.1492
0.5	0.2224	6.44×10^{-5}	0.2221	6.71×10^{-5}	0.2231
0.7	0.2238	4.35×10^{-5}	0.2237	4.51×10^{-5}	0.2235
0.9	0.1016	1.66×10^{-4}	0.1017	1.72×10^{-4}	0.1031

^aSemi-analytical integration ($N=45$ and $M = \{16, 16, 16\}$).

^{b,d}estimate for 3 terms residue.

^cGaussian quadrature ($N=45$ and $M = \{16,16,16\}$); and

^eUNIT with user provided analytical integration ($N=300$).

Table 2 illustrates the convergence behavior of the UNIT code results for a fixed truncation order ($N=45$), but varying the number of sub-regions in the semi-analytical integration (M). One may see the consistency of such results, which are apparently converged within three to four significant digits around $M = \{12,12,12\}$ for the time $t=0.02$ and to four significant digits for $t=0.1$. The automatic procedure for selecting the number of sub-intervals, available in the UNIT code, considers the eigenfunction of highest frequency (highest eigenvalue) in each direction, and multiplies by four the number of periods of the eigenfunction oscillation in the space variable domain. With $N=45$, in the present case, the code would then automatically find the number of sub-regions to be $M = \{16, 12, 12\}$, which provides very close results.

Next, similar analyses were undertaken for the 2-D and 1-D cases, which are shown in Tables 3–6. From Table 3a one may observe the convergence of the UNIT code solution to plus or minus one in the fourth significant digit for both $t=0.02$ and 0.1, and truncation orders up to $N=60$. From Table 3b one may observe the agreement to at least the third significant digit when comparing the two automatic integration schemes implemented in the UNIT code. Also, the error estimates are compatible with the oscillation in the fourth significant digit and aid in the verification of the numerical results from the *NDSolve* routine. Table 4 shows the convergence behavior of the present UNIT code solution for the two-dimensional Burgers' equation obtained with $N=50$ terms and different numbers of sub-regions (M) in the semi-analytical integration. As can be seen, four significant converged digits are practically reached with $M = \{24,24\}$ for all investigated times and positions, which is also confirmed by the last column with the results from the UNIT code solution itself with automatically generated number of sub-regions, $M = \{32,28\}$.

Table 5a shows that for the one-dimensional case the UNIT results are fully converged to at least four significant digits at $t=0.02$, with a maximum truncation

Table 2. Convergence of UNIT code solution for three-dimensional Burgers' equation (GITT with fixed truncation order, $N=45$, and different number of sub-regions (M) in semi-analytical integration)

x	M			
	{6, 6, 6}	{9, 9, 9}	{12, 12, 12}	{16, 16, 16}
$t = 0.02$				
0.1	0.2692	0.2645	0.2626	0.2615
0.3	0.7458	0.7444	0.7439	0.7436
0.5	0.9251	0.9250	0.9250	0.9250
0.7	0.9247	0.9244	0.9243	0.9242
0.9	0.4762	0.4793	0.4806	0.4813
$t = 0.1$				
0.1	0.04982	0.04938	0.04922	0.04914
0.3	0.1511	0.1500	0.1496	0.1494
0.5	0.2245	0.2232	0.2227	0.2224
0.7	0.2244	0.2240	0.2239	0.2238
0.9	0.1013	0.1015	0.1016	0.1016

order of $N=35$, as expected, with improved convergence behavior with respect to the previous two and three-dimensional situations, since the squared eigenvalues governing terms experience a faster increase in the one-dimensional solution. The results from the two automatic integration schemes, semi-analytical and Gaussian quadrature, are in very good agreement at the same truncation order of $N=35$ and number of sub-regions $M=150$, as shown in Table 5*b*. The *NDSolve* numerical results are comparatively improved, also from Table 5*b*, with at least a three-digits agreement with the converged UNIT results in the tabulated range. Again, Table 6 shows the consistency of the automatic semi-analytical integration, by comparing results obtained with different numbers of sub-regions, M , in the vicinity of the automatically selected number of sub-regions. Here, the automatic number of sub-regions, for a truncation order of $N=35$, would result in $M=140$.

Table 3a. Convergence of UNIT code solution for two-dimensional Burgers' equation ($N=20$ to 60 terms and $M=\{28, 28\}$ in semi-analytical integration)

x	$N=20$	$N=30$	$N=40$	$N=50$	$N=60$
$t = 0.02$					
0.1	0.2904	0.2813	0.2845	0.2830	0.2830
0.3	0.7437	0.7488	0.7500	0.7482	0.7483
0.5	0.9473	0.9464	0.9443	0.9445	0.9446
0.7	0.9275	0.9251	0.9261	0.9277	0.9276
0.9	0.4924	0.4960	0.4966	0.4973	0.4972
$t = 0.1$					
0.1	0.06495	0.06405	0.06432	0.06410	0.06410
0.3	0.1953	0.1957	0.1959	0.1956	0.1956
0.5	0.2971	0.2969	0.2968	0.2968	0.2968
0.7	0.3044	0.3038	0.3040	0.3043	0.3043
0.9	0.1424	0.1430	0.1432	0.1434	0.1434

Table 3b. Comparison of UNIT code solutions with different integration schemes for two-dimensional Burgers' equation and evaluation of error estimation procedure

x	GITT ^a	Error estimate ^b	GITT ^c	Error estimate ^d	<i>NDSolve</i> ^e
$t = 0.02$					
0.1	0.2830	-4.46×10^{-4}	0.2827	-4.48×10^{-4}	0.2837
0.3	0.7482	-3.79×10^{-4}	0.7480	-3.85×10^{-4}	0.7490
0.5	0.9445	1.81×10^{-5}	0.9446	1.89×10^{-5}	0.9450
0.7	0.9277	2.69×10^{-4}	0.9276	2.72×10^{-4}	0.9274
0.9	0.4973	3.23×10^{-4}	0.4977	3.26×10^{-4}	0.4968
$t = 0.1$					
0.1	0.06410	-6.65×10^{-5}	0.06407	-6.49×10^{-5}	0.06434
0.3	0.1956	-3.78×10^{-5}	0.1955	-3.71×10^{-5}	0.1957
0.5	0.2968	4.37×10^{-7}	0.2966	4.54×10^{-7}	0.2968
0.7	0.3043	2.35×10^{-5}	0.3042	2.30×10^{-5}	0.3043
0.9	0.1434	3.16×10^{-5}	0.1434	3.09×10^{-5}	0.1434

^aSemi-analytical integration ($N = 50$ and $M = \{28, 28\}$).

^{b,d}estimate for 3 terms residue.

^cGaussian quadrature ($N = 50$ and $M = \{28, 28\}$); and

^e*Mathematica 7*.

Figures 2a–2c depict the graphical comparison of the present UNIT code solutions for the homogeneous problem at different times, and, respectively, for the three-, two- and one-dimensional situations, against those obtained with the *NDSolve* routine. A very good graphical agreement between the results can be observed, including the more computationally intensive three-dimensional case.

Now, in order to illustrate and compare the implemented automatic progressive linear filtering and the Laplacian polynomial filter for multi-dimensional applications, we investigate the second application presented in section 7, given by the nonlinear Burgers' equation with non-homogeneous boundary conditions, presented

Table 4. Convergence of UNIT code solution for two-dimensional Burgers' equation (GITT with fixed truncation order, $N = 50$, and different number of sub-regions (M) in semi-analytical integration)

X	M				
	{16, 16}	{20, 20}	{24, 24}	{28, 28}	Default {32, 28}
$t = 0.02$					
0.1	0.2834	0.2832	0.2831	0.2830	0.2829
0.3	0.7486	0.7484	0.7483	0.7482	0.7482
0.5	0.9444	0.9445	0.9445	0.9445	0.9445
0.7	0.9279	0.9278	0.9277	0.9277	0.9276
0.9	0.4964	0.4969	0.4971	0.4973	0.4974
$t = 0.1$					
0.1	0.06418	0.06414	0.06412	0.06410	0.06410
0.3	0.1958	0.1957	0.1956	0.1956	0.1956
0.5	0.2971	0.2969	0.2968	0.2968	0.2968
0.7	0.3044	0.3043	0.3043	0.3043	0.3043
0.9	0.1433	0.1433	0.1433	0.1434	0.1434

Table 5a. Convergence of UNIT code solution for one-dimensional Burgers' equation ($N=20$ to 35 terms and $M=150$ in semi-analytical integration)

x	$N=20$	$N=25$	$N=30$	$N=35$
$t=0.02$				
0.1	0.2867	0.2867	0.2867	0.2867
0.3	0.7577	0.7577	0.7577	0.7577
0.5	0.9567	0.9567	0.9567	0.9567
0.7	0.9394	0.9394	0.9394	0.9394
0.9	0.5038	0.5039	0.5039	0.5039
$t=0.1$				
0.1	0.08283	0.08282	0.08281	0.08281
0.3	0.2542	0.2542	0.2542	0.2542
0.5	0.3923	0.3923	0.3923	0.3923
0.7	0.4137	0.4138	0.4138	0.4138
0.9	0.2004	0.2004	0.2004	0.2004

in problem formulation (15). Since the progressive filtering procedure is based on successively filtering the problem in each spatial direction, the order in which each direction is filtered and then transformed must be chosen. First, the two-dimensional version of problem (15) is investigated. Thus two possibilities arise: the progressive filtering procedure can start in the x direction, or it can start in the y direction, whose boundary conditions are actually already homogeneous in the original formulation. In order to evaluate the performance of the progressive filtering, both possibilities are investigated. In order to provide reference results for this investigation, and also to illustrate the flexibility of the UNIT code, the user provided filtering option has been employed and a Laplacian polynomial filtering solution for problem (15), that

Table 5b. Comparison of UNIT code solutions with different integration schemes for one-dimensional Burgers' equation and evaluation of error estimation procedure

X	GITT ^a	Error estimate ^b	GITT ^c	Error estimate ^d	<i>NDSolve</i> ^e
$t=0.02$					
0.1	0.2867	1.48×10^{-6}	0.2867	1.47×10^{-6}	0.2867
0.3	0.7577	-6.75×10^{-6}	0.7577	-6.74×10^{-6}	0.7576
0.5	0.9567	5.54×10^{-7}	0.9567	5.49×10^{-7}	0.9567
0.7	0.9394	1.22×10^{-5}	0.9394	1.22×10^{-5}	0.9395
0.9	0.5039	-4.49×10^{-5}	0.5039	-4.49×10^{-5}	0.5040
$t=0.1$					
0.1	0.08281	5.15×10^{-6}	0.08281	5.13×10^{-6}	0.08297
0.3	0.2542	-5.86×10^{-6}	0.2542	-5.85×10^{-6}	0.2543
0.5	0.3923	8.94×10^{-7}	0.3923	8.92×10^{-7}	0.3923
0.7	0.4138	1.04×10^{-5}	0.4138	1.04×10^{-5}	0.4138
0.9	0.2004	-4.48×10^{-5}	0.2004	-4.48×10^{-5}	0.2006

^aSemi-analytical integration ($N=35$ and $M=180$).^{b,d}estimate for 3 terms residue.^cGaussian quadrature ($N=35$ and $M=180$); and^e*Mathematica 7*.

Table 6. Convergence of UNIT code solution for one-dimensional Burgers' equation (GITT with fixed truncation order, $N=35$, and different number of sub-regions (M) in semi-analytical integration)

x	M			
	60	100	140	180
$t=0.02$				
0.1	0.2867	0.2867	0.2867	0.2867
0.3	0.7577	0.7577	0.7577	0.7577
0.5	0.9567	0.9567	0.9567	0.9567
0.7	0.9394	0.9394	0.9394	0.9394
0.9	0.5038	0.5039	0.5039	0.5039
$t=0.1$				
0.1	0.08282	0.08282	0.08281	0.08281
0.3	0.2542	0.2542	0.2542	0.2542
0.5	0.3923	0.3923	0.3923	0.3923
0.7	0.4138	0.4138	0.4138	0.4138
0.9	0.2004	0.2004	0.2004	0.2004

is able to homogenize all the four boundary conditions simultaneously, is implemented. The simplest choice is given by the following polynomial filtering.

$$F(x, y) = ax^2 + by^2 + cxy + dx + ey \tag{18a}$$

Where the coefficients are readily computed by solving a system of algebraic equations stemming from substituting Eq. (18a) into the following system.

$$\frac{\partial^2 F(x, y)}{\partial x^2} + \frac{\partial^2 F(x, y)}{\partial y^2} = 0 \tag{18b}$$

$$F(0, y) = 1; \quad F(1, y) = 0 \tag{18c, 18d}$$

$$\left. \frac{\partial F(x, y)}{\partial y} \right|_{y=0} = 0; \quad F(x, 1) = 0 \tag{18e, 18f}$$

Figures 3a and 3b depict the graphical comparison of the UNIT code solutions with progressive filtering at different times, along x (with $y=0.5$) and along y (with $x=0.5$), respectively, obtained with $N=65$ terms selected with the simple squared eigenvalues reordering criterion, and $M = \{20, 20\}$ points in the integrations via Gaussian quadrature, against those obtained with the polynomial filtering and $N=90$ terms selected with the combined reordering scheme and $M = \{20, 20\}$ points in the integrations via Gaussian quadrature. Both possibilities for the progressive filtering, either starting with the direction x or with the direction y , are presented. These figures confirm the overall good adherence of the most direct UNIT code solutions, with automatic progressive filtering and the simplest reordering scheme, to the reference values, also obtained via UNIT code, but with an user provided filtering option and more robust reordering scheme. For this case, it is also possible to see slightly improved results achieved when the progressive filtering procedure starts

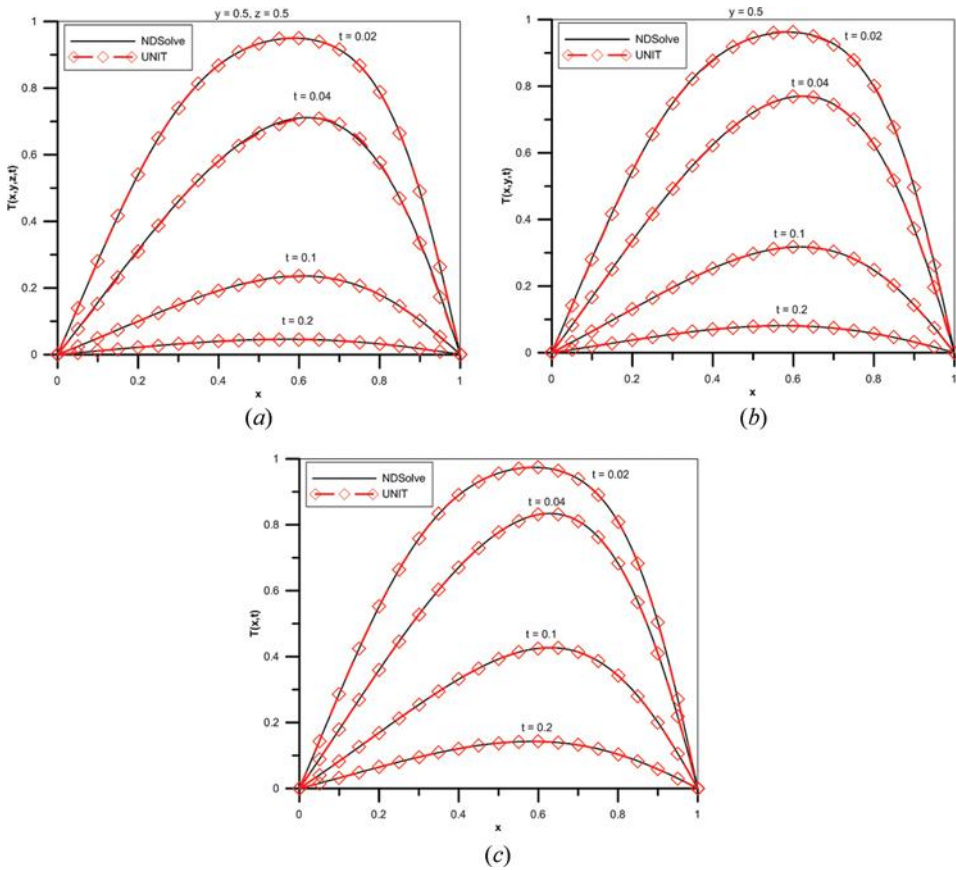


Figure 2. Comparison between the UNIT code and the *NDSolve* routine solutions with homogeneous boundary conditions. (a) three-dimensional formulation; (b) two-dimensional formulation; and (c) one-dimensional formulation (color figure available online).

with the direction having originally homogeneous boundary conditions, being more noticeable for increasing y values, especially near $y = 1$ (Figure 3b).

Finally, we investigate problem (15) in its full three-dimensional version employing the automatic progressive filtering option and also an user provided polynomial filtering solution in order to provide reference results, which has been chosen as follows.

$$F(x, y, z) = ax^4 + by^4 + cz^2 + dxyz + exy + fxz + gyz \tag{19a}$$

Where the coefficients are computed to satisfy the following.

$$\frac{\partial^2 F(x, y, z)}{\partial x^2} + \frac{\partial^2 F(x, y, z)}{\partial y^2} + \frac{\partial^2 F(x, y, z)}{\partial z^2} - F(x, y, z) = 0 \tag{19b}$$

$$F(0, y, z) = 1; \quad F(1, y, z) = 0 \tag{19c, 19d}$$

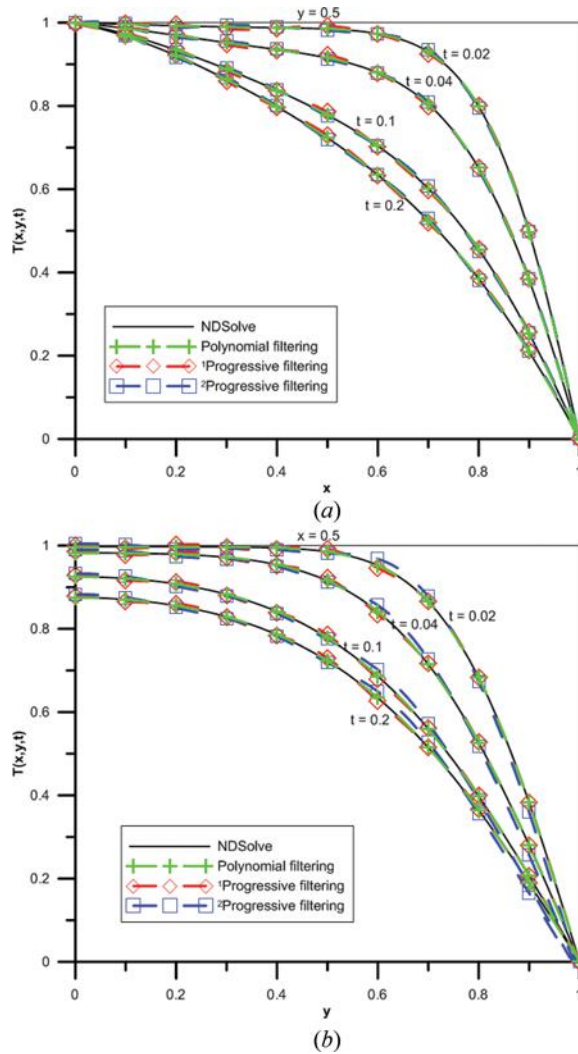


Figure 3. Comparison between the UNIT code solutions with automatic progressive filtering and user provided polynomial filtering for the two-dimensional formulation with nonhomogeneous boundary conditions. (1) automatic progressive filtering starting with the x direction; (2) automatic progressive filtering starting with the y direction: (a) along x ; and (b) along y (color figure available online).

$$\left. \frac{\partial F(x, y, z)}{\partial y} \right|_{y=0} = 0; \quad F(x, 1, z) = 0 \tag{19e, 19f}$$

$$\left. \frac{\partial F(x, y, z)}{\partial z} \right|_{z=0} = 0; \quad F(x, y, 1) = 0 \tag{19g, 19h}$$

Figures 4a–4c bring the comparison between the solutions obtained with the UNIT code using the automatic progressive filtering option with those obtained

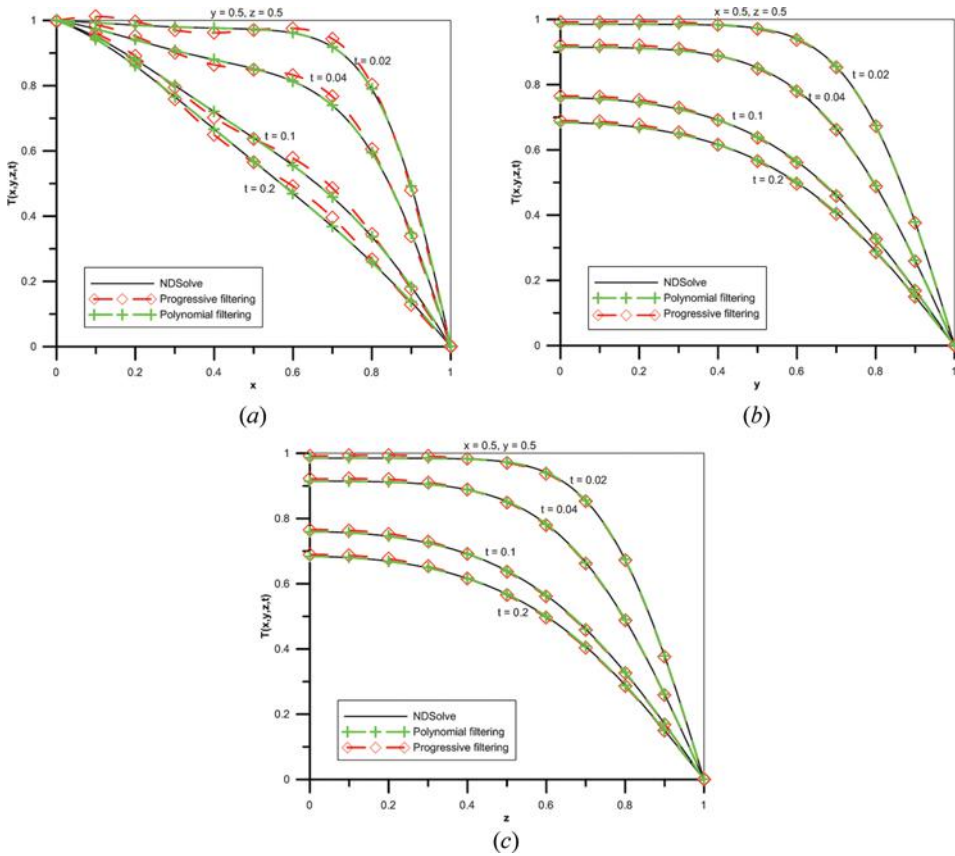


Figure 4. Comparison between the UNIT code solutions with automatic progressive filtering and user provided polynomial filtering for the three-dimensional formulation with nonhomogeneous boundary conditions: (a) along x ; (b) along y ; and (c) along z (color figure available online).

when using the user provided polynomial filtering (19), showing T profiles along x , y , and z coordinates, respectively. The progressive filtering solution is obtained with $N=75$ terms, $M=\{8, 8, 10\}$ points in the integrations via Gaussian quadratures, and employing the simplest reordering scheme with the squared eigenvalues criterion. For the reference results, with the polynomial filtering solution, $N=94$ terms were selected via the combined reordering scheme and $M=\{8, 8, 10\}$ points were employed in the integrations via Gaussian quadratures. In all the graphs one can observe an excellent adherence of the curves obtained when employing the automatic progressive filtering option with the reference solution with user provided boundary conditions filter. These results illustrate the robustness of the UNIT code in tackling nonhomogeneous boundary conditions also for non-linear multidimensional formulations, and confirms the automatic progressive filtering option as an alternative tool either for users with no experience to propose a specific filtering solution or for users interested in the most practical solution path for a given problem.

9. CONCLUSION

This article presented a unified automatic algorithm for the hybrid numerical-analytical solution of multi-dimensional convection-diffusion problems based on the Generalized Integral Transform Technique. The UNIT algorithm, as it has been previously coined in one-dimensional situations [17], was implemented using mixed symbolic-numerical computation under the *Mathematica* framework. The algorithm and relevant features in its computational implementation are discussed, and application results were subsequently presented. Three specific aspects in the algorithm are more closely analyzed in the present work. First, the alternative automatic coefficients integration schemes, a semi-analytical procedure and a Gaussian quadrature scheme, with automatic selection of the number of integration sub-regions or points based on the oscillatory behavior of the eigenfunction of highest frequency; second, the automatic reordering schemes for multidimensional situations, either the traditional squared eigenvalues reordering criterion, or a more conservative combined reordering scheme that adds extra terms due to the estimated importance of the initial conditions and source terms on the transformed potentials decay; and third, the automatic filtering of nonhomogeneous boundary conditions in multidimensional problems, here described as a progressive filtering strategy, and critically compared to a user provided Laplacian polynomial filter for the boundary conditions.

A three-dimensional nonlinear Burgers' equation formulation with homogeneous boundary conditions is first adopted to test the algorithm. Then, a second test case, the three-dimensional nonlinear Burgers' equation formulation with non-homogeneous boundary conditions is addressed, in order to particularly compare the filtering schemes for multidimensional problems. As a basis for comparison, the solutions are here obtained with the UNIT code in its most crude and automatic usage form, i.e., with automatic semi-analytical or non-adaptive Gaussian integration schemes and with reordering based on the sum of the squared eigenvalues. Although convergence rates, accuracy and computational costs may always be optimized with this same approach, for instance via specific filtering using information regarding the equations source terms, analytical integration of initial conditions and coefficients, eigenvalue problems with convective terms, local instantaneous filtering and dynamic reordering, the current examples herein presented were intended to test the algorithm under the most adverse conditions due to the simplest user choices, so as to demonstrate the robustness of the UNIT implementation. Nevertheless, even under these most direct usage choices, results were reported for all three levels of formulation, from three to one-dimensional, confirming that the approach is able to achieve accurate results in all cases, with higher accuracy in the one dimensional case for a fixed truncation order in the eigenfunction expansions. The computational costs are not significantly different for the three levels of formulation at the same truncation order, and the more significant computing effort increase is in the generation of the transformed source terms through the approximate integration schemes for the multidimensional situations. Research should now proceed towards the implementation of additional refinements and usage options to the basic code structure so as to progressively offer a more complete development platform for hybrid integral transforms methods.

REFERENCES

1. R. M. Cotta and M. N. Ozisik, Laminar Forced Convection in Ducts with Periodic Variation of Inlet Temperature, *Int. J. Heat Mass Transfer*, vol. 29, pp. 1495–1501, 1986.
2. R. M. Cotta, Hybrid Numerical-Analytical Approach to Nonlinear Diffusion Problems, *Numer. Heat Transfer B*, vol. 127, pp. 217–226, 1990.
3. R. Serfaty and R. M. Cotta, Hybrid Analysis of Transient Nonlinear Convection-Diffusion Problems, *Int. J. Numer. Meth. Heat Fluid Flow*, vol. 2, pp. 55–62, 1992.
4. R. M. Cotta, *Integral Transforms in Computational Heat and Fluid Flow*, CRC Press, Boca Raton, 1993.
5. R. M. Cotta, Benchmark Results in Computational Heat and Fluid Flow—The Integral Transform Method, *Int. J. Heat Mass Transfer* (invited paper), vol. 37, pp. 381–394, 1994.
6. R. M. Cotta and M. D. Mikhailov, *Heat Conduction: Lumped Analysis, Integral Transforms, Symbolic Computation*, Wiley-Interscience, Chichester, UK, 1997.
7. R. M. Cotta, (ed.), 1998. *The Integral Transform Method in Thermal and Fluids Sciences and Engineering*, Begell House, New York.
8. C. A. C. Santos, J. N. N. Quaresma, and J. A. Lima, *Benchmark Results for Convective Heat Transfer in Ducts—The Integral Transform Approach*, E-Papers, Rio de Janeiro, Brazil 2001.
9. R. M. Cotta and M. D. Mikhailov, Hybrid Methods and Symbolic Computations, In W. J. Minkowycz, E. M. Sparrow and J. Y. Murthy, (eds.), *Handbook of Numerical Heat Transfer*, 2nd ed., chap. 16, John Wiley, New York, 2006.
10. M. D. Mikhailov and M. N. Ozisik, *Unified Analysis and Solutions of Heat and Mass Diffusion*, John Wiley, New York, 1984; also, Dover Publications, 1994.
11. S. Wolfram, *The Mathematica Book*, Cambridge/Wolfram Media, Champaign, IL, 2005.
12. M. D. Mikhailov and R. M. Cotta, Integral Transform Method for Eigenvalue Problems, *Comm. Numer. Meth. Eng.*, vol. 10, pp. 827–835, 1994.
13. L. A. Sphaier and R. M. Cotta, Integral Transform Analysis of Multidimensional Eigenvalue Problems within Irregular Domains, *Numer. Heat Transfer B*, vol. 38, pp. 157–175, 2000.
14. L. A. Sphaier and R. M. Cotta, Analytical and Hybrid Solutions of Diffusion Problems within Arbitrarily Shaped Regions via Integral Transforms, *Computational Mechanics*, vol. 29, pp. 265–276, 2002.
15. C. P. Naveira-Cotta, R. M. Cotta, H. R. B. Orlande, and O. Fudym, Eigenfunction Expansions for Transient Diffusion in Heterogeneous Media, *Int. J. Heat Mass Transfer*, vol. 52, pp. 5029–5039, 2009.
16. R. M. Cotta and M. D. Mikhailov, Semi-Analytical Evaluation of Integrals for the Generalized Integral Transform Technique, *4th Workshop on Integral Transforms and Benchmark Problems*, Rio de Janeiro, Brazil, 23–25 August 2005.
17. L. A. Sphaier, R. M. Cotta, C. P. Naveira-Cotta, and J. N. N. Quaresma, The UNIT Algorithm for Solving One-Dimensional Convection-Diffusion Problems via Integral Transforms, *Int. Comm. in Heat and Mass Transfer*, vol. 38, pp. 565–571, 2011.
18. M. D. Mikhailov and R. M. Cotta, Ordering Rules for Double and Triple Eigenseries in the Solution of Multidimensional Heat and Fluid Flow Problems, *Int. Comm. Heat & Mass Transfer*, vol. 23, pp. 299–303, January 1996.
19. E. J. Correa, R. M. Cotta, and H. R. B. Orlande, On the Reduction of Computational Costs in Eigenfunction Expansions of Multidimensional Diffusion Problems, *Int. J. Numer. Meth. Heat & Fluid Flow*, vol. 7, no. 7, pp. 675–695, 1997.