Monotonicity recovering and accuracy preserving optimization methods for postprocessing finite element solutions

Oleg Burdakov, Ivan Kapyrin and Yuri Vassilevski

Monotonicity recovering and accuracy preserving optimization methods for postprocessing finite element solutions

Oleg Burdakov[a,1], Ivan Kapyrin[b] and Yuri Vassilevski[b]

[a] *Department of Mathematics, Linköping University, Linköping, Sweden*

[b] *Institute of Numerical Mathematics, Russian Academy of Sciences, Moscow, Russia*

Abstract

We suggest here a least-change correction to available finite element (FE) solution. This postprocessing procedure is aimed at recovering the monotonicity and some other important properties that may not be exhibited by the FE solution. Although our approach is presented for FEs, it admits natural extension to other numerical schemes, such as finite differences and finite volumes. For the postprocessing, a priori information about the monotonicity is assumed to be available, either for the whole domain or for a subdomain where the lost monotonicity is to be recovered. The obvious requirement is that such information is to be obtained without involving the exact solution, e.g., from expected symmetries of this solution.

The postprocessing is based on solving a monotonic regression problem with some extra constraints. One of them is a linear equality-type constraint that models the conservativity requirement. The other ones are box-type constraints, and they originate from the discrete maximum principle. The resulting postprocessing problem is a large scale quadratic optimization problem. It is proved that the postprocessed FE solution preserves the accuracy of the discrete FE approximation.

We introduce an algorithm for solving the postprocessing problem. It can be viewed as a dual ascent method based on the Lagrangian relaxation of the equality constraint. We justify theoretically its correctness. Its efficiency is demonstrated by the presented results of numerical experiments.

Keywords: Finite element solution, Accuracy analysis, Constrained monotonic regression, Large scale quadratic optimization, Lagrangian relaxation, Dual ascent method

1 Introduction

In this paper, we consider the following constrained monotonic regression problem. Given: a vector of positive weights $w \in R^n$, a vector $\bar{u} \in R^n$, a monotonicity establishing matrix

[1]Corresponding author. *E-mail address:* oleg.burdakov@liu.se

$M \in R^{l\times n}$, scalars $a < b$ and m. Find $u^* \in R^n$ that solves the least-distance problem:

$$\begin{array}{ll} \min & \frac{1}{2}\|u-\bar{u}\|_w^2 \\ u \in R^n & \text{subject to:} \\ & Mu \geq 0, \qquad \text{(M)} \\ & ae \leq u \leq be, \quad \text{(B)} \\ & w^T u = m. \qquad \text{(C)} \end{array} \tag{1}$$

Here the notations $\|a\|_w^2 = \sum_{i=1}^n w_i a_i^2$ and $e = (1, 1, \ldots, 1)^T \in R^n$ are used. The matrix M has the structure of an arc-node incidence matrix [1]. This means that each row of M is composed of zeros, but two elements, one of them being equal to $+1$ and the other one to -1. Therefore, each constraint in (M) is of the form $u_i \geq u_j$, and it establishes a monotonicity relation between components i and j of the vector u.

Problem (1) with the only constraint (M), referred further to as (1M), is a classical monotonic regression problem [2, 37]. There exist efficient methods for solving this problem of moderate [5, 33, 36] and very large [7, 8, 9, 10] sizes. To the best of our knowledge, there are no efficient methods to solving large scale constrained monotonic regression problems of the form (1). The conventional constrained optimization methods [19, 20, 35], quadratic optimization methods [17] and constrained least squares methods [20, 39] require unacceptably too long computational time to solve such large scale problems. The main aim of this paper is to develop optimization methods for solving problem (1) which would be efficient in the large scale case, and also to apply them to postprocessing finite element (FE) solutions.

Our interest to studying problem (1) is motivated by the following reasons.

FE discretizations have become conventional in the computational and engineering communities due to their theoretical basis and technological capabilities. In addition to good approximation properties such as the second order accuracy, there may be other requirements to the FE solution. In many practical cases, it is desirable that the maximum principle and mass conservation are inherited by the resulting discrete systems. Even for the FE discretization of a model advection-diffusion equation, an accurate discretization method that satisfies the discrete maximum principle (DMP) is hard to develop.

In the case of the diffusion equation with full and anisotropic diffusion tensor, numerical solutions often demonstrate nonphysical behavior: they may be negative in large regions where the continuous solution is strictly positive, or have unwanted spurious oscillations. In advection dominated advection-diffusion problems, a continuous solution may have internal shock and exponential or parabolic boundary layers. The thickness of these features is small compared to mesh size and hence the layers cannot be resolved properly. Most of numerical schemes either smear out these layers excessively or violate the DMP. The design of advanced discretization schemes which eliminate or significantly reduce these disadvantages is the field of extensive research for more than three decades.

Already in 70's, P.Ciarlet and P.Raviart [15] presented the theoretical analysis of sufficient mesh conditions that provides the DMP for piecewise-linear finite element approximations of the diffusion equation. Later, the validity of DMP has been shown for weaker mesh conditions [23, 38].

The most popular approach to the stabilization of FE methods for advection-diffusion equations was proposed by Brooks and Hughes in [6] and is referred to as the streamline upwind Petrov-Galerkin (SUPG) method. However, the spurious oscillations along sharp layers may still appear in the numerical solution stabilized with SUPG. They are caused by the fact that the SUPG method is not a monotone method. A review of several modifications and improvements for SUPG method is presented in [21]. Such modifications aimed at improving the FE methods that satisfy the DMP, at least in some model cases, are called *spurious oscillations at layers diminishing* (SOLD) methods. We wish also to mention here the algebraic flux correction approach [24] to the design of monotone FE discretization methods. It was noticed in [4, 21] that *nonlinear* approximations is the key ingredient and the price which has to be paid to construct monotone and at least the second order accurate discretization. To guarantee solution monotonicity for arbitrary meshes, a number of *nonlinear* methods have been proposed in both FE [12, 32] and finite volume [4, 16, 22, 25, 26, 28, 29, 30, 34, 40, 41] frameworks.

We present here a procedure for postprocessing non-monotone FE solution which produces a corrected solution satisfying the monotonicity, conservativity and DMP requirements. It also preserves the order of accuracy. It is necessary to emphasize that this will allow for using already implemented numerical schemes which produce unwanted non-monotonic features in the discrete solution. Moreover, our approach can naturally be extended to the case of postprocessing solutions produced by other numerical schemes, such as finite differences and finite volumes.

The postprocessing procedure is based on finding a least-change correction to the available FE solution determined by the vector of the corresponding FE coefficients $\bar{u}$. The values of $\bar{u}_i$ and u_i are assumed to be associated with the i-th FE basis function, and w_i be the corresponding weight of this function in the sense of its integral over the support. If to consider an advection-diffusion process, $\bar{u}_i$ and u_i can have the meaning of concentrations, original and corrected, respectively. If m is the total mass involved in the advection-diffusion process, then equation (C) can be viewed as the mass conservativity requirement. The box-type constraints $a \leq u_i \leq b$ in (B) may originate from the DMP. In (M), the inequality constraints of the type $u_i \geq u_j$ establish a local monotonic relation for the corresponding couple of basis functions i and j, typically, with adjacent supports.

We do realize that our approach does not offer a universal tool, because it requires an a priori knowledge about local monotonicity relations presented by the matrix M. In some problems, like those used in our numerical experiments, the monotonicity relations can be obtained by analyzing the equations and boundary conditions that define the problems (see Section 4). In such problems, it is possible to find locally, without invoking the exact solution, a set of directions along which the solution, e.g. the concentration, can not increase. If two adjacent basis functions, say i and j, lie along a direction from the mentioned local set, this implies a monotonic relation of the form $u_i \geq u_j$. If FE solutions are expected to be symmetric in some sense, they may naturally suggest such local sets of directions.

It should be mentioned that if constraint (M) is absent in (1), i.e. no monotonicity relations are provided, then the postprocessing performs the least-change correction of the FE solution that assures the mass conservativity and DMP.

The paper is organized as follows. In Section 2, we study postprocessing problem (1). In particular, we consider how to establish the monotonicity relations and how to define the constraints (M) in (1). Then we prove that the postprocessed FE solution preserves the accuracy of the discrete FE approximation. In the same section, it is shown also how to use the solution to problem (1M) for obtaining the solution to problem (1M,B). The postprocessing algorithm aimed at solving large scale constrained monotonic regression problems of the form (1) is introduced in Section 3. It is based on the Lagrangian relaxation of constraint (C) in (1), which implies solving a finite sequence of problems of the form (1M,B). We show how to avoid solving them by modifying properly the solution to problem (1M). The results of our numerical experiments are presented and discussed in Section 4.

2 Postprocessing problem

In the FE method [14], a function $\Phi : R^d \to R^1$ of the form

$$\Phi(\chi) = \sum_{i=1}^{n} u_i \varphi_i(\chi) \tag{2}$$

is produced for approximating the exact solution $\Phi^*(\chi)$ to an original problem to be solved, e.g. partial differential equations, integral equations etc. Here $\{\varphi_i(\chi)\}_{i=1}^n$ are, for instance, P_0 or P_1 FE basis functions, and $\{u_i\}_{i=1}^n$ are the corresponding scalar coefficients.

In our postprocessing problem, a discrete FE solution is supposed to be available. It is determined by a vector of coefficients $\bar{u}$ which produces an approximate solution of the form (2). This approximate solution may not retain some important properties of the exact solution. In this paper, we focus on the following requirements for the approximate solution:

- local monotonicity, which means that the approximate solution should not increase locally along certain directions (modelled in (1) by constraint (M));
- narrowness, e.g. within the bounds determined by the maximum principle [13] (modelled in (1) by constraint (B));
- conservativity (modelled in (1) by constraint (C)).

Our postprocessing problem (1) is aimed at recovering the listed properties by finding a least-change correction to the coefficients $\bar{u}$. The vector of corrected coefficients u^* is required to satisfy the constraints that model these properties.

2.1 Monotonicity constraints

Consider the procedure of generating monotonicity constraints (M). It exploits the following relation between the monotonicity of the coefficients u_i and the monotonicity of the function $\Phi(\chi)$. Assume, for the moment, that the exact solution $\Phi^*(\chi)$ is known to be a non-increasing function of χ, i.e.

$$\chi' \leq \chi'' \quad \Longrightarrow \quad \Phi^*(\chi') \geq \Phi^*(\chi'').$$

Here $\chi' \leq \chi''$ is a component-wise inequality. Let $\chi_i \in R^d$ denote the point of collocation of the i-th FE basis function. For the P_0 FE basis function, χ_i is a cell barycenter, while for the P_1 FE basis function, χ_i is a mesh node.

We assume that the FEs possess the following property.

Assumption. *If $\Phi(\chi)$ is a non-increasing function of χ, then for each pair of basis functions, i and j, the coefficients u_i and u_j are such that $u_i \geq u_j$ whenever $\chi_i \leq \chi_j$ component-wise.*

The P_0 and P_1 FE basis functions meet this requirement due to their property that $\Phi(\chi_i) = u_i$. Generalization of Assumption to higher order finite elements should take into account that there may be several basis functions contributing to $\Phi(\chi_i)$.

Note that Assumption gives only a necessary condition for the monotonicity of the function $\Phi(\chi)$. Obviously, the monotonicity of the function values $\Phi(\chi_i)$ in a discrete set of points does not necessarily imply that the function is monotonic. Our postprocessing makes the corrected function values monotonic, and it does help in practice to recover the lost monotonicity of the function as well, despite the fact that, in general, this is not guaranteed.

The monotonic relation that we have just considered is a kind of global monotonicity. To turn to modeling a local monotonicity, we observe that for establishing the relation $u_i \geq u_j$ we actually required that the vector $\chi_j - \chi_i$ belongs to the positive orthant of R^d which is a convex cone of all vectors with non-negative components.

In the case of local monotonicity, it may be known for the original problem that its solution $\Phi^*(\chi)$ does not increase locally around χ_i along any vector from an a priori available convex cone K_i. Here, *locally* means in a neighborhood Ω_i of χ_i, where Ω_i is, for instance, the union of the supports of basis function i and its adjacent basis functions. Formally, this means that

$$\Phi^*(\chi_i) \geq \Phi^*(\chi_i + sp)$$

for any $p \in K_i$ and for any positive scalar s such that $\chi_i + sp \in \Omega_i$. Let FE j be adjacent to FE i and $\chi_j - \chi_i \in K_i$. Then the considered local monotonicity can be modelled by the inequality $u_i \geq u_j$.

Of course, such cones may not coincide with the positive orthant and they may depend on the location of the corresponding FEs. This is well illustrated in Problem 1 considered in Section 4.

Note that problems may have, in practice, some subdomains in which it is difficult to draw a priori any conclusion about local monotonicity. In such subdomains, the monotonicity establishing cones K_i should be empty.

The constraints in (M) are aimed on recovering the expected monotonicity. Therefore, these constraints should mainly originate from the subdomains, where the FE approximation produced by $\bar{u}$ violates the monotonicity. In the subdomains, where the monotonicity is safely preserved, it is reasonable to skip generating redundant constraints for (M). This approach decreases the total number of constraints in (M) and eases the solution process of postprocessing problem (1).

2.2 Accuracy analysis

In the accuracy analysis, we do not require that all the cases of violated monotonicity are corrected by the postprocessing algorithm. Our estimates hold even if (M) consists of only one constraint, or even if it has no constraints at all. The most important is that each monotonicity constraint in (M) is consistent with the monotonicity of $\Phi^*(\chi)$ in the sense that inequality $u_i - u_j \geq 0$ can belong to constraints (M) only if $\Phi^*(\chi_i) - \Phi^*(\chi_j) \geq 0$. This requirement refers to M, and it can be written as $M\Phi^* \geq 0$, where Φ^* denotes the vector with the components $\Phi^*(\chi_1), \ldots, \Phi^*(\chi_n)$.

In this sub-section, we assume that the basis functions are such that

$$\varphi_i(\chi_j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases}$$

holds for all $1 \leq i, j \leq n$. Under this assumption, (2) implies $\Phi(\chi_i) = u_i$, which is typical, e.g., for the P_0 and P_1 basis functions.

Then the L_2 norm of the error of finite element solution may be estimated in terms of the weighted Euclidean norm $\|\bar{u} - \Phi^*\|_w$, where the weight w_i is the representative volume associated with i-th FE basis function. In the case of P_0 finite element, w_i equals to the ith cell volume, whereas in the case of P_1 finite element, w_i is the volume of a cell of the dual mesh associated with the ith node.

The next result implies that the postprocessed FE solution retains the same order of accuracy in L_2-norm as the original FE solution.

Theorem 1 *Let the vector of collocated exact solution Φ^* satisfy conditions (M), (B), (C) for given matrix M, vector w and scalars α, β, m. Suppose that vector $\bar{u}$ defines a finite element approximation of Φ^*, and vector u^* defines the postprocessed finite element solution, i.e. it solves problem (1) for the same data M, w, α, β, m. Then*

$$\|u^* - \Phi^*\|_w \leq \|\bar{u} - \Phi^*\|_w. \tag{3}$$

Proof. The feasible set of points in R^n, i.e. those satisfying conditions (M), (B) and (C), is a convex set. The postprocessed solution u^* is the projection of $\bar{u}$ onto this set, which implies that

$$(\Phi^* - u^*, \bar{u} - u^*)_w \leq 0,$$

where

$$(u', u'')_w = \sum_{i=1}^{n} w_i u'_i u''_i.$$

Then

$$\begin{aligned} \|\bar{u} - \Phi^*\|_w^2 &= \|\bar{u} - u^* + u^* - \Phi^*\|_w^2 = \|\bar{u} - u^*\|_w^2 + \|u^* - \Phi^*\|_w^2 - 2(\Phi^* - u^*, \bar{u} - u^*)_w \\ &\geq \|\bar{u} - u^*\|_w^2 + \|u^* - \Phi^*\|_w^2 \geq \|u^* - \Phi^*\|_w^2. \end{aligned}$$

This proves (3). ∎

Let u_M denote the solution to monotonic regression problem (1M). Like in the proof of Theorem 1, we can show that

$$\|u^* - \Phi^*\|_w \leq \|u_M - \Phi^*\|_w \leq \|\bar{u} - \Phi^*\|_w. \tag{4}$$

This relation between the three errors is illustrated by the results of our numerical experiments presented in Section 4 for a problem whose exact solution is known.

2.3 Features of the postprocessing problem

We assume that the feasible set in problem (1) is not empty. Since it is a polyhedron and the objective function is the squared weighted Euclidean distance from the point $\bar{u}$ to this polyhedron, the optimal solution u^* exists and is unique.

Clearly, the solution u_M to monotonic regression problem (1M) may violate the constraints (B) and (C). But if the discrete FE solution is such that

$$w^T\bar{u} = m, \tag{5}$$

this assures that (C) holds for u_M [2, 37]. In this case, u_M solves problem (1M,C), however the constraint (B) may be violated.

Let u_{MB} denote the solution to box-constrained monotonic regression problem (1M,B). If (5) holds, this does not necessarily imply that u_{MB} satisfies constraint (C).

In Theorem 2, we will show that u_{MB} can be easily obtained from u_M. The solution u_{MB} plays an important role in our postprocessing algorithm, in which u_{MB} is successively modified until finally the modified solution satisfies (C). This assures that the resulting modified solution solves postprocessing problem (1).

Let π be the projection operation defined for a scalar v as follows

$$\pi(v) = \begin{cases} a, & \text{if } v \leq a, \\ b, & \text{if } v \geq b, \\ v, & \text{if } a < v < b. \end{cases} \tag{6}$$

If u is a vector and $\pi(u)$ is defined component-wise as above, then $\pi(u)$ is the orthogonal projection of the point u on the box determined by constraints (B).

The next result presents a relation between u_M and u_{MB}.

Theorem 2 *Let u_M solve problem* (1M). *Then $u_{MB} = \pi(u_M)$ solves problem* (1M,B).

Proof. Consider the Lagrangian function for problem (1M). It is defined as

$$L_M(u, \lambda) = \frac{1}{2}\|u - \bar{u}\|_w^2 - \lambda^T M u,$$

where the vector of the Lagrange multipliers λ has the same number of components as the number of constraints in (M).

Since the constraints in (1M) are linear, there exist Lagrange multipliers λ_M such that the following Karush-Kuhn-Tucker (KKT) conditions [3] hold for u_M:

$$\begin{array}{ll} \nabla_u L_M(u_M, \lambda_M) = 0, & \text{(a)} \\ M u_M \geq 0, & \text{(b)} \\ \lambda_M^T M u_M = 0, & \text{(c)} \\ \lambda_M \geq 0. & \text{(d)} \end{array} \qquad (7)$$

With the use of the same Lagrangian function, we can present the KKT conditions for problem (1M,B) in the following form [3]:

$$\begin{array}{ll} \left.\begin{array}{ll} \nabla_{u_i} L_M(u_{MB}, \lambda_{MB}) \geq 0, & \text{if } (u_{MB})_i = a, \\ \nabla_{u_i} L_M(u_{MB}, \lambda_{MB}) \leq 0, & \text{if } (u_{MB})_i = b, \\ \nabla_{u_i} L_M(u_{MB}, \lambda_{MB}) = 0, & \text{if } a < (u_{MB})_i < b, \end{array}\right\} \quad i = 1, 2, \ldots, n, & \text{(a)} \\ M u_{MB} \geq 0, & \text{(b)} \\ \lambda_{MB}^T M u_{MB} = 0, & \text{(c)} \\ \lambda_{MB} \geq 0, & \text{(d)} \\ ae \leq u_{MB} \leq be, & \text{(e)} \end{array} \qquad (8)$$

They are necessary and sufficient optimality conditions for (1M,B), because it is a convex quadratic programming problem.

We will prove that KKT conditions (8) hold for $u_{MB} = \pi(u_M)$ and $\lambda_{MB} = \lambda_M$.

Indeed, conditions (8d) and (8e) obviously hold.

If $(u_M)_i - (u_M)_j \geq 0$, then $\pi((u_M)_i) - \pi((u_M)_j) \geq 0$. This implies (8b).

Let $(\lambda_M)_{ij}$ stand for the Lagrange multiplier associated in (7) with the constraint $(u_M)_i - (u_M)_j \geq 0$. Then the complementarity conditions, which follow from (7), can be written as

$$(\lambda_M)_{ij} \cdot [(u_M)_i - (u_M)_j] = 0.$$

Hence, as it can be easily verified, the equality

$$(\lambda_M)_{ij} \cdot [\pi((u_M)_i) - \pi((u_M)_j)] = 0$$

holds for each constraint in (M). This proves (8c).

We observe that

$$\nabla_u L_M(u_{MB}, \lambda_{MB}) = \nabla_u L_M(u_M, \lambda_M) + \operatorname{diag}(w)\Delta u, \qquad (9)$$

where $\Delta u = u_{MB} - u_M$. By the definition of the projection operation $\pi(\cdot)$, we have

$$\left.\begin{array}{ll} (\Delta u)_i \geq 0, & \text{if } (u_{MB})_i = a, \\ (\Delta u)_i \leq 0, & \text{if } (u_{MB})_i = b, \\ (\Delta u)_i = 0, & \text{if } a < (u_{MB})_i < b, \end{array}\right\} \quad i = 1, 2, \ldots, n, \qquad (10)$$

Due to relations (7a) and (10) along with the inequality $w \geq 0$, equation (9) gives (8a).

Thus, the optimality conditions hold for $\pi(u_M)$. This finally proves that $u_{MB} = \pi(u_M)$ solves problem (1M,B). $\square$

3 Postprocessing algorithm

Our postprocessing algorithm is based on the Lagrangian relaxation [3] of the linear equality constraint (C). Consider the corresponding Lagrangian function

$$L_C(u,\mu) = \frac{1}{2}\|u-\bar{u}\|_w^2 + \mu(m - w^T u), \tag{11}$$

where the scalar μ is the Lagrange multiplier.

The corresponding Lagrangian dual function $\varphi(\mu)$ is given by

$$\begin{array}{rll} \varphi(\mu) = & \min & L_C(u,\mu) \\ & u \in R^n & \text{subject to:} \\ & & Mu \geq 0, \\ & & ae \leq u \leq be. \end{array} \tag{12}$$

As with all Lagrangian dual functions, $\varphi(\mu)$ is a concave function of μ [3].

Since $L_C(u,\mu)$ is a strictly convex function of u, and since the constraints in (12) are linear, the solution to the minimization problem in (12) is unique for any μ. We denote it by $u(\mu)$. Then

$$\varphi(\mu) = L_C(u(\mu),\mu) = \frac{1}{2}\|u(\mu)-\bar{u}\|_w^2 + \mu(m - w^T u(\mu)).$$

Consider the dual problem

$$\max_{\mu \in R^1} \varphi(\mu). \tag{13}$$

We denote its optimal solution by μ^*.

Due to the uniqueness of $u(\mu)$, the dual function $\varphi(\mu)$ is continuously differentiable, and its derivative has the form (see, e.g., [3]):

$$\varphi'(\mu) = m - w^T u(\mu). \tag{14}$$

From the necessary and sufficient optimality conditions for problem (13), we have $\varphi'(\mu^*) = 0$. Thus, the solution μ^* can be obtained by solving the scalar equation

$$m - w^T u(\mu) = 0 \tag{15}$$

in μ (see Fig. 1), where the function $u(\mu)$ is implicitly given by (12). The duality theory [3, Theorem 6.5.1] implies that $u(\mu^*)$ solves our postprocessing problem (1). Our postprocessing algorithm is based on solving equation (15), which we shall refer to as *the main equation.*

3.1 Properties of the main equation

Since the function $\varphi(\mu)$ is concave, its derivative $\varphi'(\mu)$ is a monotonically non-increasing function of μ. Due to (14), the same refers to the left-hand side function of the main equation. It will be explained below why $m - w^T u(\mu)$ is even strictly decreasing with μ in a large neighborhood of μ^*.

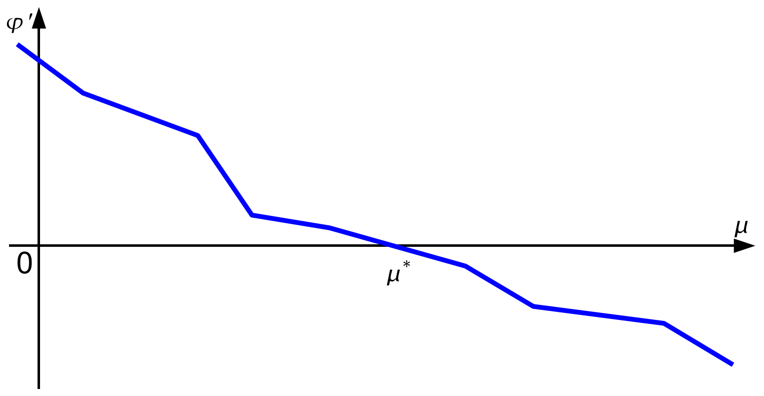

Figure 1: Derivative of the dual function

This property, of course, eases the process of solving the main equation, but it still looks necessary to generate $u(\mu)$ by repeatedly solving problem (12) for each iterate μ.

For $\mu = 0$, problem (12) is equivalent to (1M,B). By Theorem 2, $u(0)$ can be easily obtained from the solution to the standard monotonic regression problem (1M) by virtue of a simple orthogonal projection (6) of u_M onto the box defined by (B) in (1).

We will show that, for getting $u(\mu)$, it is sufficient to solve problem (12) only once, namely for $\mu = 0$. We will show also how to obtain $u(\mu)$, in an efficient way, from u_M for any given μ without resolving problem (12). The key observation here is the following.

The Lagrangian function $L_C(u, \mu)$ defined by (11) can be presented equivalently as

$$L_C(u, \mu) = \frac{1}{2}\|u - \bar{u}_\mu\|_w^2 + q(\mu), \tag{16}$$

where

$$\bar{u}_\mu = \bar{u} + \mu e,$$

and

$$q(\mu) = \mu(m - w^T\bar{u}) - \frac{1}{2}\mu^2 w^T e.$$

This can be verified by the direct substitution of $\bar{u}_\mu$ and $q(\mu)$ into (16).

Since $q(\mu)$ does not depend on u, formula (16) allows for rewriting problem (12) equivalently as

$$\begin{array}{rll} \varphi(\mu) = q(\mu) + & \min & \frac{1}{2}\|u - \bar{u}_\mu\|_w^2 \\ & u \in R^n & \text{subject to:} \\ & & Mu \geq 0, \\ & & ae \leq u \leq be. \end{array} \tag{17}$$

This presentation of the dual function enables establishing a useful relation between $u(\mu)$ and u_M, which can be formulated as follows.

Theorem 3 *Let u_M solve problem* (1M). *Then*

$$u(\mu) = \pi(u_M + \mu e) \tag{18}$$

solves the minimization problem in (17) for any value of μ.

The proof of this theorem immediately follows from Theorem 2 combined with the following result.

Lemma 4 *Let u_M solve problem* (1M). *Then $u_M + \mu e$ solves the problem*

$$\begin{array}{ll} \min & \frac{1}{2}\|u - \bar{u}_\mu\|_w^2 \\ u \in R^n & \text{subject to:} \\ & Mu \geq 0. \end{array} \tag{19}$$

Proof. Consider the new variables $v = u - \mu e$. Note that $M\mu e = 0$. Then, the substitution of $u = v + \mu e$ into (19) gives an equivalent problem, which has the same form as (1M) with the only difference that the variables v are used instead of u. Thus, $v = u_M$ solves the monotonic regression problem for the new variables. This proves that $u_M + \mu e$ solves problem (19). $\square$

In formula (18), the operation of adding the scalar μ to each component of the vector u_M is used. We shall call it the *μ-shift* of u_M. Theorem 3 allows us to avoid direct solving problem (12) whenever it is necessary to find $u(\mu)$. Formula (18) offers the following computationally efficient way of finding $u(\mu)$.

First, we solve problem (1M). Then for any given μ, the value of $u(\mu)$ is obtained by the μ-shift of u_M with subsequent orthogonal projection of $u_M + \mu e$ by formula (6) onto the box defined by (B) in (1). It is necessary to emphasize that the major computational burden of finding $u(\mu)$ is associated with problem (1M) which is to be solved only once.

In the next subsection, the further reduction of the computational burden will be based on the observation that in the process of solving equation (15), it is not necessary to construct explicitly the vector $u(\mu)$. It is sufficient to manipulate with the value of the scalar product $w^T u(\mu)$ instead.

In what follows, we assume that the components of the vector u_M are sorted in increasing order, i.e. $(u_M)_i \leq (u_M)_{i+1}$. Obviously, both the μ-shift and the subsequent orthogonal projection (6) retain the same increasing order of the components.

We shall use the notations:

$$\begin{aligned} \alpha_-(\mu) &= \max\{i : \ (u_M)_i + \mu < a, \ 1 \leq i \leq n\}, \\ \alpha_+(\mu) &= \min\{i : \ (u_M)_i + \mu > a, \ 1 \leq i \leq n\}, \\ \beta_-(\mu) &= \max\{i : \ (u_M)_i + \mu < b, \ 1 \leq i \leq n\}, \\ \beta_+(\mu) &= \min\{i : \ (u_M)_i + \mu > b, \ 1 \leq i \leq n\}. \end{aligned} \tag{20}$$

These indices are presented in the example shown in Fig. 2. For simplicity, we consider only those values of μ for which these indices exist. Note that if $\alpha_-(\mu) + 1 < \alpha_+(\mu)$, than $(u_M)_i + \mu = a$ for all i such that $\alpha_-(\mu) < i < \alpha_+(\mu)$. Similarly, if $\beta_-(\mu) + 1 < \beta_+(\mu)$, than $(u_M)_i + \mu = b$ for all i such that $\beta_-(\mu) < i < \beta_+(\mu)$. Denote also

$$W(\mu) = \sum_{i=\alpha_+(\mu)}^{\beta_-(\mu)} w_i.$$

For a given μ, consider how the left-hand side of the main equation depends on the perturbed value $\mu + \Delta\mu$. We assume that the perturbation $\Delta\mu \in [\Delta\mu_-(\mu), \Delta\mu_+(\mu)]$,

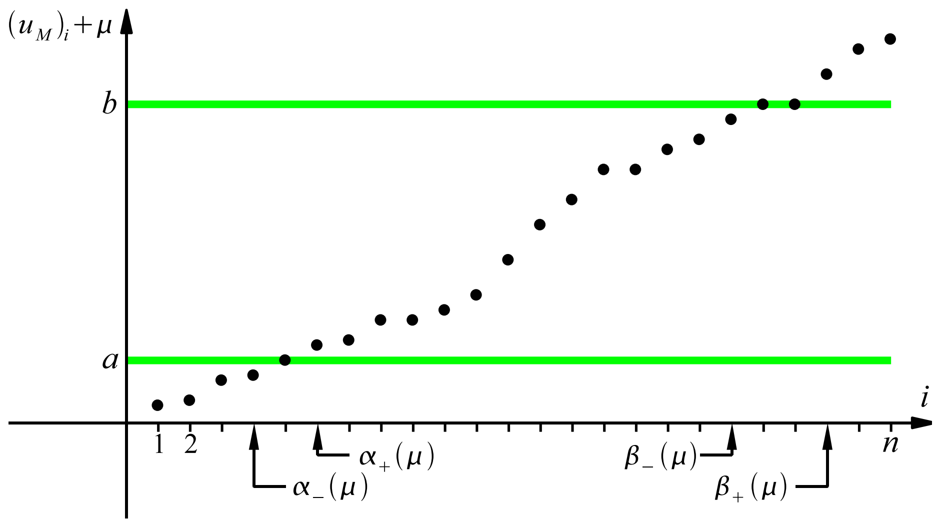

Figure 2: The μ-shift of u_M with the components $(u_M)_i$ sorted in increasing order.

where the scalars $\Delta\mu_-(\mu) < 0$ and $\Delta\mu_+(\mu) > 0$ are defined by the formulas:

$$\Delta\mu_-(\mu) = \max\{a - (u_M)_{\alpha_+(\mu)},\ b - (u_M)_{\beta_+(\mu)}\} - \mu,$$

$$\Delta\mu_+(\mu) = \min\{a - (u_M)_{\alpha_-(\mu)},\ b - (u_M)_{\beta_-(\mu)}\} - \mu.$$

It can be easily verified that, for any $\Delta\mu$ of the mentioned interval, the equality

$$\varphi'(\mu + \Delta\mu) = \varphi'(\mu) + \Delta\mu \cdot \begin{cases} \varphi''_-(\mu), & \text{if } \Delta\mu \in [\Delta\mu_-(\mu), 0], \\ \varphi''_+(\mu), & \text{if } \Delta\mu \in [0, \Delta\mu_+(\mu)] \end{cases} \tag{21}$$

holds, where the notations

$$\varphi''_-(\mu) = -W(\mu) - \sum_{i=\beta_-(\mu)+1}^{\beta_+(\mu)-1} w_i, \qquad \varphi''_+(\mu) = -W(\mu) - \sum_{i=\alpha_-(\mu)+1}^{\alpha_+(\mu)-1} w_i \tag{22}$$

are used. Equality (21) implies that $\varphi''_-(\mu)$ and $\varphi''_+(\mu)$ are, respectively, the left and right derivatives of $\varphi'(\mu)$.

By the formulas in (22), the points μ in which $\varphi'(\mu)$ is not differentiable, i.e. $\varphi''_-(\mu) \neq \varphi''_+(\mu)$, are among those for which there exists, at least one, i such that either $\mu = a - (u_M)_i$ or $\mu = b - (u_M)_i$. They correspond to the jog points on the graph in Fig. 1. For all other values of μ, the left-hand side of the main equation is differentiable, and its derivative is equal to $W(\mu)$.

We have $W(\mu) > 0$ for all values of μ such that the set $\{i :\ a < (u_M)_i + \mu < b\}$ is not empty. This typically holds in the problems of practical interest within a wide interval around the optimal value μ^*. On this interval, the left-hand side of the main equation is strictly monotonically decreasing with μ.

3.2 Implementation issues

Here we present an algorithm for solving the main equation (15). Its main feature is that it avoids explicit calculation of the vector $u(\mu)$, except $u(0)$ which, as noticed above, can be easily calculated by projecting u_M on the box.

Since the sign of $\varphi^{'}(0) = m - w^T u(0)$ determines the sign of μ_*, one can decide whether to increase μ if $\varphi^{'}(0) > 0$, or to decrease it if $\varphi^{'}(0) < 0$. Depending on this, the μ-shifted components of u_M move with μ up or down in Fig. 2.

It is typical for the solution to monotonic regression problem (1M) that a few components of u_M have the same value. In the case of the FE problem, this means that some of the adjacent cells in the areas, where the monotonicity is violated, may have the same value of $(u_M)_i$. It can be easily verified that, if two components of u_M are of the same value then the values of these components in $\pi(u_M + \mu e)$ remain equal for any μ.

We take this observation into account in our algorithm, and we assume that the components of u_M with the same value are represented in the new vector u_M by only one component of this value. The weight corresponding to the new component is assumed to be equal to the sum of weights w_i of the components which are represented by the new component. For the sake of simplicity, we shall continue using the same notations u_M and w for the new vectors. Furthermore, n will also denote the number of components of the new vectors, although these vectors may be shorter. The further assumption that the components of the new (*shortened*) vector u_M are sorted in increasing order implies that $(u_M)_i < (u_M)_{i+1}$.

The basic idea is the following. Suppose that $\varphi^{'}(0) > 0$. Then starting from $\mu = 0$, the value of μ is increased, step by step, by adding $\Delta\mu_+(\mu)$. As the result, every new value of μ is equal to

$$\min\{a - (u_M)_{\alpha-(\mu)},\ b - (u_M)_{\beta-(\mu)}\}.$$

The value of $\varphi^{'}(\mu)$ is updated by formula (21). The values of μ generated in this way correspond to the jog points. This procedure terminates when $\varphi^{'}(\mu)$ becomes negative. The root μ^* belongs to the interval between the final value of μ and the previous one. Since the function $\varphi^{'}(\mu)$ is linear on this interval, one Newton step gives $\mu^* = \mu - \varphi^{'}(\mu)/\varphi^{''}_{-}(\mu)$.

The outlined idea is formally presented by Algorithm 1. The input parameters of this algorithm are a, b, m, shortened u_M and w. It returns μ^*.

Algorithm 1.

```
$\mu \leftarrow 0$
Compute $\alpha_-$, $\alpha_+$, $\beta_-$ and $\beta_+$ by (20) for $\mu = 0$
$\varphi' \leftarrow m - a\sum_{i=1}^{\alpha_-} w_i - \sum_{i=\alpha_-+1}^{\beta_-}(u_M)_i w_i - b\sum_{i=\beta_-+1}^{n} w_i$
**if** $\varphi' = 0$ **then**
    $\mu^* \leftarrow 0$ and stop
**if** $\varphi' > 0$ **then**
    $\alpha \leftarrow \alpha_-,\ \beta \leftarrow \beta_-,\ W \leftarrow \sum_{i=\alpha+1}^{\beta} w_i$
    **while** $\varphi' > 0$ **do**
        $\mu_a \leftarrow a - (u_M)_\alpha,\ \mu_b \leftarrow b - (u_M)_\beta$
        **if** $\mu_a < \mu_b$ **then**
            $\Delta\mu \leftarrow \mu_a - \mu,\ \mu \leftarrow \mu_a,\ \Delta W \leftarrow w_\alpha,\ \alpha \leftarrow \alpha - 1$
        **else**
            $\Delta\mu \leftarrow \mu_b - \mu,\ \mu \leftarrow \mu_b,\ \Delta W \leftarrow -w_\beta,\ \beta \leftarrow \beta - 1$
        $\varphi'' \leftarrow -W,\ \varphi' \leftarrow \varphi' + \varphi''\Delta\mu,\ W \leftarrow W + \Delta W$
**if** $\varphi' < 0$ **then**
    $\alpha \leftarrow \alpha_+,\ \beta \leftarrow \beta_+,\ W \leftarrow \sum_{i=\alpha}^{\beta-1} w_i$
    **while** $\varphi' < 0$ **do**
        $\mu_a \leftarrow a - (u_M)_\alpha,\ \mu_b \leftarrow b - (u_M)_\beta$
        **if** $\mu_a < \mu_b$ **then**
            $\Delta\mu \leftarrow \mu_b - \mu,\ \mu \leftarrow \mu_b,\ \Delta W \leftarrow w_\beta,\ \beta \leftarrow \beta + 1$
        **else**
            $\Delta\mu \leftarrow \mu_a - \mu,\ \mu \leftarrow \mu_a,\ \Delta W \leftarrow -w_\alpha,\ \alpha \leftarrow \alpha + 1$
        $\varphi'' \leftarrow -W,\ \varphi' \leftarrow \varphi' + \varphi''\Delta\mu,\ W \leftarrow W + \Delta W$
$\mu^* \leftarrow \mu - \varphi'/\varphi''$
```

So far, it was assumed that the indices in (20) exist for all values of μ generated by Algorithm 1. For the case when α_- or β_+ may not exist, we recommend to introduce two extra components:

$$(u_M)_0 = (u_M)_1 - 2(b-a) \quad \text{and} \quad (u_M)_{n+1} = (u_M)_n + 2(b-a)$$

with $w_0 = w_{n+1} = 0$. Any upper estimate for $|\mu^*|$ can be used here instead of $b - a$.

The value of μ^* returned by Algorithm 1 is used for producing the solution vector $u^* = \pi(u_M + \mu^* e)$ by formula (18). The projection can be easily obtained with the use of the final values of indices α and β generated by Algorithm 1 and with consideration of whether $\mu_a < \mu_b$ or not. This information facilitates the identification of those components of u^* which are equal to a or b. The rest of the components are simple shifts of the form $(u_M)_i + \mu^*$.

Note that for running Algorithm 1, it is not necessary to sort all components of u_M in increasing order. Suppose that, in the case of $\varphi'(0) > 0$, the optimal shift μ^* is known to be bounded above by $\bar{\mu}$, i.e. $\mu^* < \bar{\mu}$. Then it is sufficient to sort in increasing order only two subsets of components of u_M, namely, those for which $a - \bar{\mu} < (u_M)_i < a$ and those for which $b - \bar{\mu} < (u_M)_i < b$. The other components, if shifted by Algorithm 1, would not cross the boundaries defined by a and b.

It is necessary to emphasize that the computational burden of Algorithm 1 grows with n not faster than linearly. For producing a sufficiently accurate solution to monotonic regression problem (1M), i.e. an approximation to u_M, we suggest to use the GPAV algorithm [7, 8, 9, 10, 11]. Although the worst case complexity of the GPAV algorithm is estimated as $O(n^2)$, its computational time grows in practice just a bit faster than linearly with n. Moreover, the solution u_M produced by this algorithm is in a form close to the shortened version required for running Algorithm 1.

All these considerations of the computational burden allow us to conclude that the postprocessing of FE solution is a relatively inexpensive procedure. In our numerical experiments, the postprocessing time was significantly shorter than the time of producing the FE solution, and this difference was growing with n.

4 Numerical experiments

For each of the two problems presented below, we produce a FE solution that is subsequently postprocessed. This solution gives the value of m in constraint (C) of problem (1). At the first stage of the postprocessing, the GPAV algorithm [7, 8, 9, 10, 11] is applied to solving problem (1M) for producing a *monotonicity recovering solution.* At the final stage, this monotone solution is used by Algorithm 1, as described in Section 3, for producing a *postprocessed solution* that solves problem (1) comprising all the constraints.

Problem 1 [27]. Given a positive scalar T, a domain $\Omega = (0; +\infty) \times (-\infty; +\infty)$, a diagonal diffusion tensor $\mathcal{D}$ and a divergence-free vector field $\vec{b}$, find the solute concentration $C(x, y, t)$ that solves the two-dimensional non-stationary advection-diffusion problem:

$$\frac{\partial C}{\partial t} - \nabla \mathcal{D} \nabla C + \vec{b} \nabla C = 0 \qquad \text{in } \Omega \times [0; T]$$

with the following initial and boundary conditions. Initial conditions:

$$C(x, y, 0) = 0 \qquad \text{in } \Omega.$$

Neumann boundary conditions:

$$\begin{aligned} \left.\frac{\partial C}{\partial x}\right|_{x \to +\infty} &= 0 \qquad y \in (-\infty; +\infty),\ t > 0, \\ \left.\frac{\partial C}{\partial y}\right|_{y \to \pm\infty} &= 0 \qquad x \in (0; +\infty),\ t > 0. \end{aligned}$$

Non-homogeneous Dirichlet boundary conditions:

$$C(0, y, t) = \begin{cases} 1, & \text{if } |y| < 0.25, \\ 0, & \text{otherwise.} \end{cases}$$

This problem is widely used for testing numerical algorithms. Its exact solution is given, e.g., in [27]. The one presented by Fig. 3 (top left) corresponds to $t = 0.5$, the

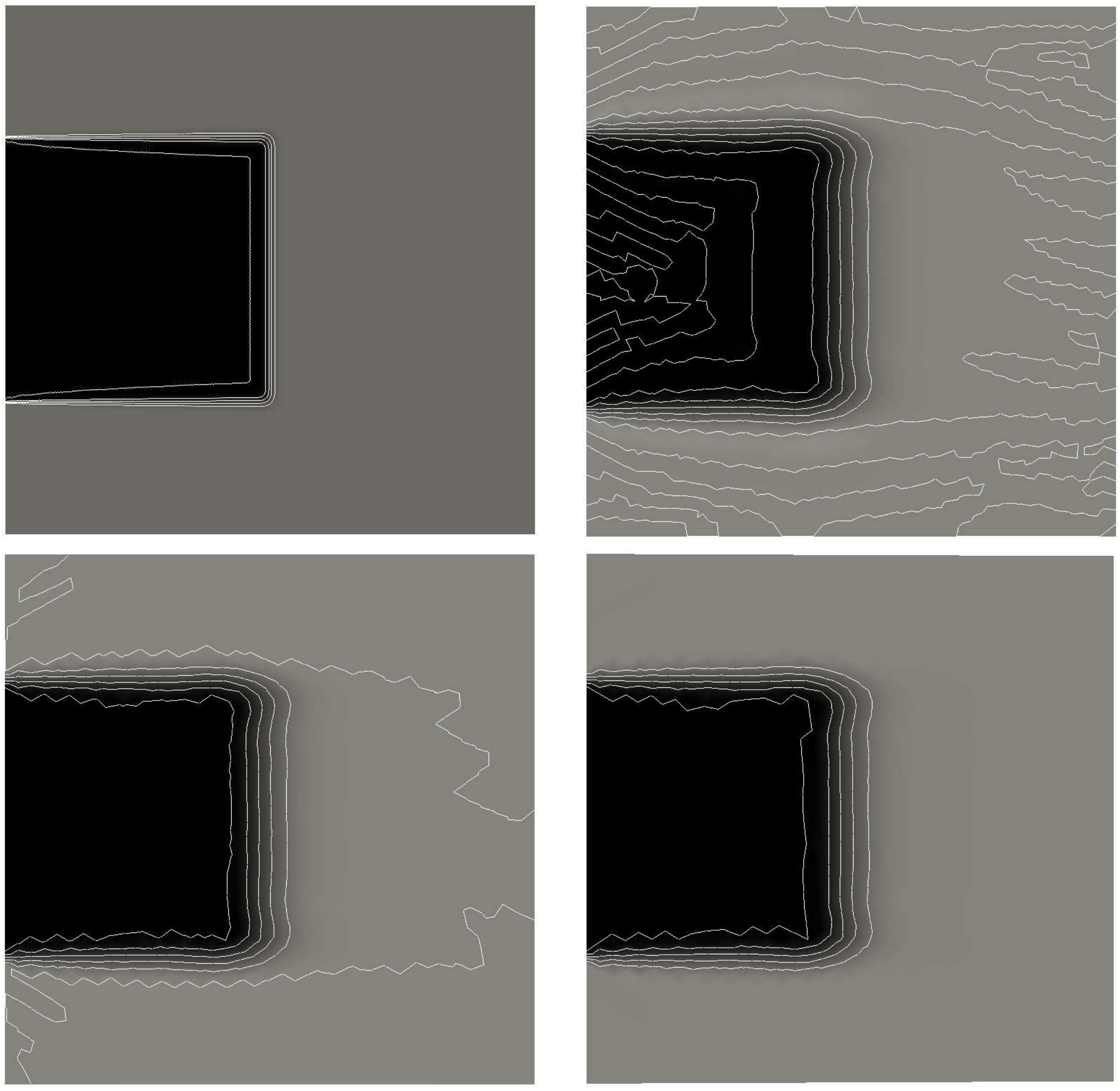

Figure 3: Solutions to Problem 1: exact (top left), SUPG for $\Delta t = 0.02$ (top right), monotonicity recovering (bottom left), postprocessed for $0 \leq C \leq 1$ (bottom right).

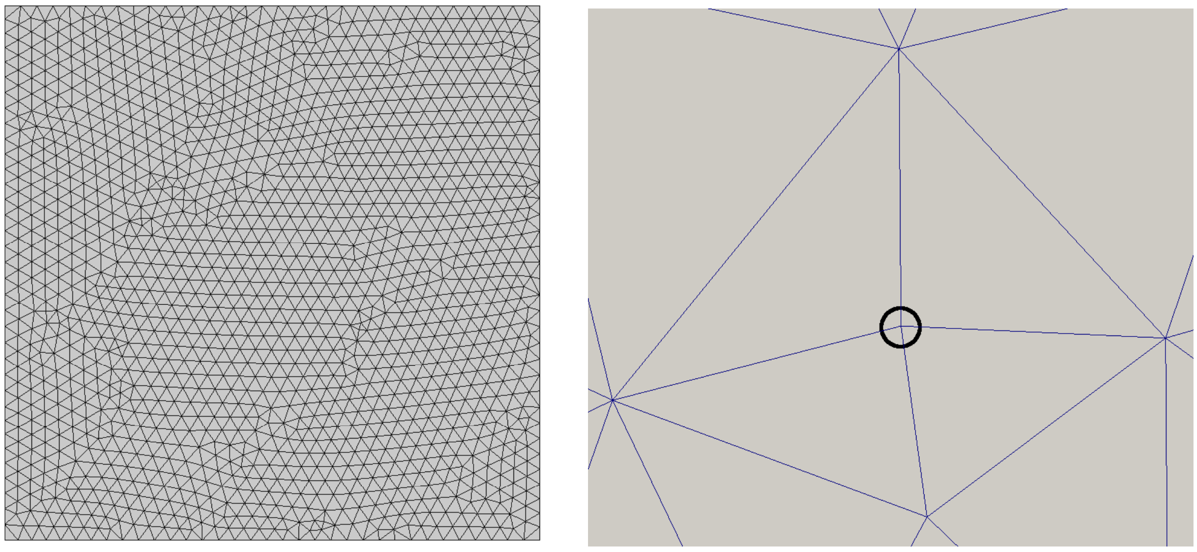

Figure 4: The computational grid in Problem 1 (left) and a node from Ω_1 whose monotonicity cone $\{(x,y)^T : \; x \geq 0, y \geq 0\}$ contains no adjacent nodes (right).

simplest advection field $\vec{b} = (1,0)$ and the scalar diffusion tensor

$$\mathcal{D} = \begin{pmatrix} 10^{-4} & 0 \\ 0 & 10^{-4} \end{pmatrix}.$$

In Problem 1, we denote $\chi = (x,y)^T$. The features of the equations and boundary conditions that define this problem allow for drawing some useful conclusions about the monotonicity of its solution for any fixed value of $t \in (0;T]$, without invoking the exact solution.

A simple analysis of the problem formulation shows that the solution $C(x,y,t)$ is monotonically non-increasing in x and y in the subdomain $\Omega_1 = [0;+\infty) \times [0;+\infty)$. This means that

$$C(x',y',t) \geq C(x'',y'',t) \tag{23}$$

for all $\chi' = (x',y')^T \in \Omega_1$ and $\chi'' = (x'',y'')^T \in \Omega_1$ such that $\chi' \leq \chi''$.

In the subdomain $\Omega_2 = [0;+\infty) \times (-\infty;0]$, the solution $C(x,y,t)$ is non-increasing in x and non-decreasing in y. This implies that inequality (23) holds for all $\chi', \chi'' \in \Omega_2$ such that $x' \leq x''$ and $y' \geq y''$.

Thus, depending on the location χ_i of node i, the convex cone of the directions of local decrease (non-increase) is defined as follows:

$$K_i = \begin{cases} \{(x,y)^T : \; x \geq 0, y \geq 0\}, & \text{if } \chi_i \in \Omega_1, \\ \{(x,y)^T : \; x \geq 0, y \leq 0\}, & \text{if } \chi_i \in \Omega_2. \end{cases}$$

This a priori available information about the local monotonicity can be used, completely or partially, for generating constraints of the form (M). In our simplest procedure it is used partially. Namely, for each node i of the mesh and for each of its adjacent nodes

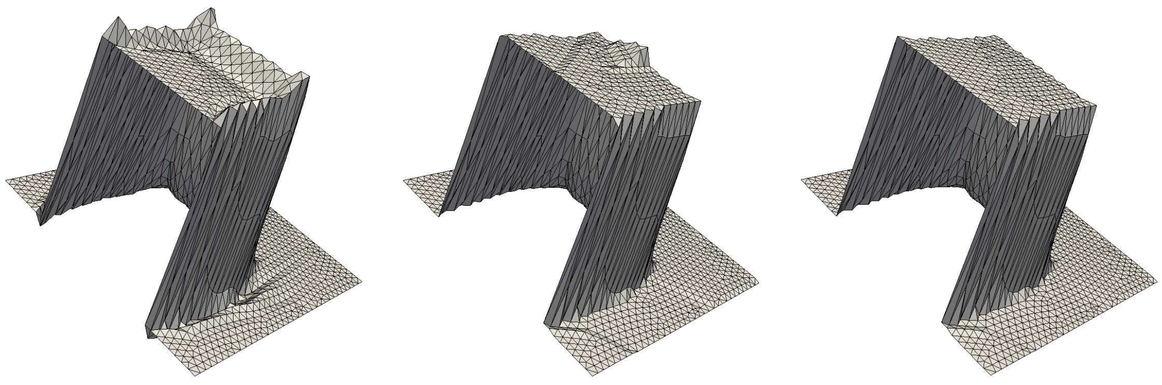

Figure 5: SUPG solution to Problem 1 for $\Delta t = 0.02$ (left) and its postprocessed versions: with the non-negativity constraint $C \geq 0$ (center) and with the maximum principle constraint $0 \leq C \leq 1$ (right).

j, we check if $\chi_j - \chi_i \in K_i$. If so, we add the inequality $u_i - u_j \geq 0$ to those already generated. The computational burden of this procedure grows linearly with the number of nodes. We also consider below its extension that allows for taking into account the local monotonicity more completely.

The postprocessing was applied to a FE solution to Problem 1 for $t = 0.5$. For producing this solution, the unstructured quasi-uniform mesh shown in Fig. 4 (left) was generated by the software *Ani2D* [31], and then the streamline upwinding Petrov-Galerkin (SUPG) method was applied for two different time steps, $\Delta t = 0.02$ and $\Delta t = 0.005$. Fig. 3 (top right) shows 2D plot of the resulting numerical solution for $\Delta t = 0.02$ represented by a piecewise-linear function. The white lines are concentration isolines corresponding to the range $[0; 1]$ with the step 0.2. One can observe unwanted features, like negative concentrations and numerous spurious oscillations presented by the isolines $C = 0$ and $C = 1$. The exact solution is, of course, free of such features (see Fig. 3 (top left)).

The solution to problem (1M) depicted in Fig. 3 (bottom left) shows that if nothing more than an incomplete monotonicity is invoked, the monotonicity recovering stage of the postprocessing allows for eliminating most of the oscillations, i.e. it performs a monotonic smoothing. The final stage of the postprocessing that additionally involves constraints (B) and (C) is aimed at eliminating the remaining oscillations as shown in Fig. 3 (bottom right) for $0 \leq C \leq 1$ in (B). The shift performed in the postprocessing was of a relatively small value, $\mu^* = 0.0046$.

We will study now how the incompleteness of the information used in monotonicity constraints (M) and box constraints (B) affects the quality of the resulting monotonicity recovering and postprocessed solutions.

The completeness of information about the bounds in (B) is of no less importance for the postprocessing than the completeness of the exploited monotonicity. This is illustrated in Fig. 5 showing how the quality of the postprocessed solution deteriorates when the incomplete information about the bounds, $C \geq 0$, is used instead of the complete one, $0 \leq C \leq 1$. The same figure shows the ability of the complete information used in

Table 1: Problem 1: solution minima and maxima excluding the boundary.

	SUPG for $\Delta t = 0.02$	monotonicity recovering	postprocessed ($C \geq 0$)	postprocessed ($0 \leq C \leq 1$)
min	-8.71E-2	-1.00E-2	0.00E+0	0.00E+0
max	1.17E+0	1.11E+0	1.10E+0	1.00E+0

(B) to repair the incompletely recovered monotonicity, cf. (center) and (right) in Fig. 5. The violations of the maximum principle are summarized for the generated solutions in Table 1. Note that the monotonicity recovering stage reduces the most severe violation of the non-negativity condition by one order of magnitude.

The quality of monotonicity recovering depends on how completely the monotonicity relations are presented by constraints (M). As it is apparent from Figs. 3 (bottom left) and 5 (center), the monotonicity was restored incompletely. This is caused by the incomplete use of the a priori available information about the monotonicity.

The complete use implies, in particular, that $u_j - u_i \geq 0$ must hold for *all* nodes i and j such that

$$\chi_j - \chi_i \in K_i, \tag{24}$$

but not only for those adjacent, like in the simplest procedure. In some cases, the use of only adjacent nodes guarantees complete monotonicity recovering. An example is the case of the regular grid used for solving Problem 2 below. This is not the case in Problem 1 because of the unstructured mesh. Some nodes i of this mesh, like the one in Fig. 4 (right), have no adjacent nodes j for which (24) holds. For such 'orphaned' nodes, the value of u_i does not impose any upper bound on u_j for the distant nodes j satisfying (24). This may result in an incomplete monotonicity recovering. There exist orphaned nodes of another type. Such nodes j have no adjacent nodes i for which (24) holds.

The deficiency of monotonic relations in (M) caused by the orphaned nodes can be compensated by extending the set of nodes for which (24) is verified from adjacent to more distant couples. To emphasize the importance of using more complete information about the monotonicity, we conduct the following experiments.

For generating monotonicity relations (M), we now verify (24) for j from two layers of i's neighbours. The resulting postprocessed solution for $C \geq 0$ is presented in Fig. 6 (left). One can see that, in comparison with the one layer case in Fig. 5 (center), the crest on the top looks more localized (see the corresponding zooms on Fig. 7), and the bottom part looks better smoothed.

The presence of the mentioned crest indicates that the monotonicity was still recovered incompletely. This is due to the remaining orphaned nodes located in the vicinity of the line $y = 0$ that is the interface between the subdomains Ω_1 and Ω_2. To eliminate the crest, we exploit the fact that the solution does not increase with x along that line. Based on this a priori available information about the local monotonicity, we generate extra constraints of the form (M) as follows. We introduce additional monotonicity relations for the cells crossed by the line $y = 0$, even though the central nodes of such couples of cells may belong to different subdomains. Note that this interface line crosses cell i if at

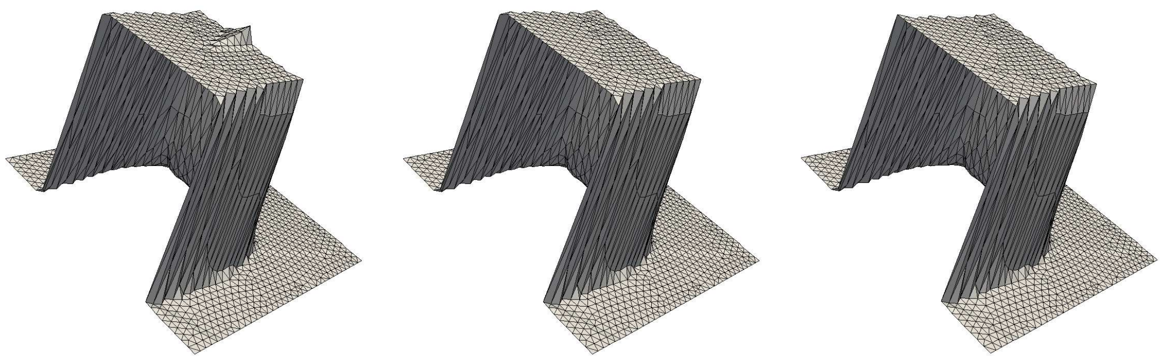
Figure 6: Solutions to Problem 1 with two layers of neighbours. Monotonicity recovering solution without supplementary monotonicity constraints along the line $y = 0$ (left). Solutions with such constraints: monotonicity recovering (center) and postprocessed with $0 \leq C \leq 1$ (right).

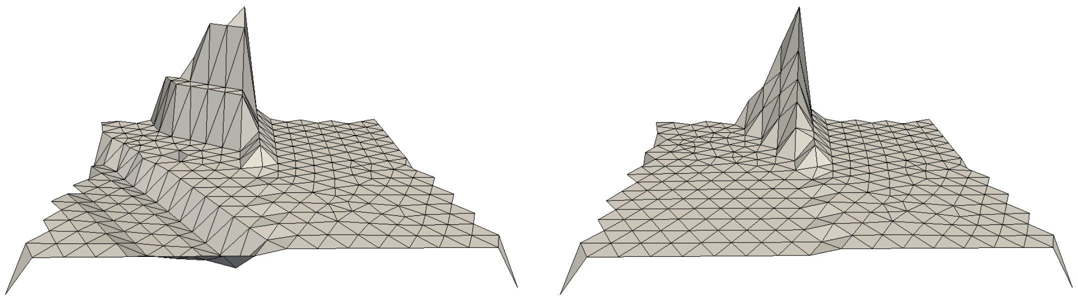
Figure 7: Zoomed and scaled (3 times) areas of $C \geq 0.95$ of the monotonized solutions obtained using one layer (left) and two layers (right) of adjacent cells.

least one of the nodes adjacent to i belongs to the subdomain that is different from the one containing i. We call them *interface* cell and node. In our implementation, if two interface nodes i and j are such that $x_i < x_j$ and j belongs to the two layer neighbourhood of i, then we add the inequality $u_i - u_j \geq 0$ to those already generated. The resulting solution presented in Fig. 6 (center) demonstrates substantial improvement in the monotonicity recovering in comparison with the other solution Fig. 6 (left). Some further smoothing is performed at the final stage of the postprocesing, see Fig. 6 (right). In this solution the spurious oscillations are removed more completely than in the postprocessed solution that is based on the less complete information about the monotonicity, cf. Figs. 6 (right) and 5 (right) in the areas close to $C = 0$.

Note that in the case of $\Delta t = 0.02$ the major part of the spurious oscillations is removed by the postprocessing based on the narrowest set of the considered monotonicity relations (one layer). For the smaller time step $\Delta t = 0.005$, the result is different: only the use of the widest set of the relations (two layers and supplementary constraints along $y = 0$) allows to get rid of the oscillations. The original SUPG solution for $\Delta t = 0.005$

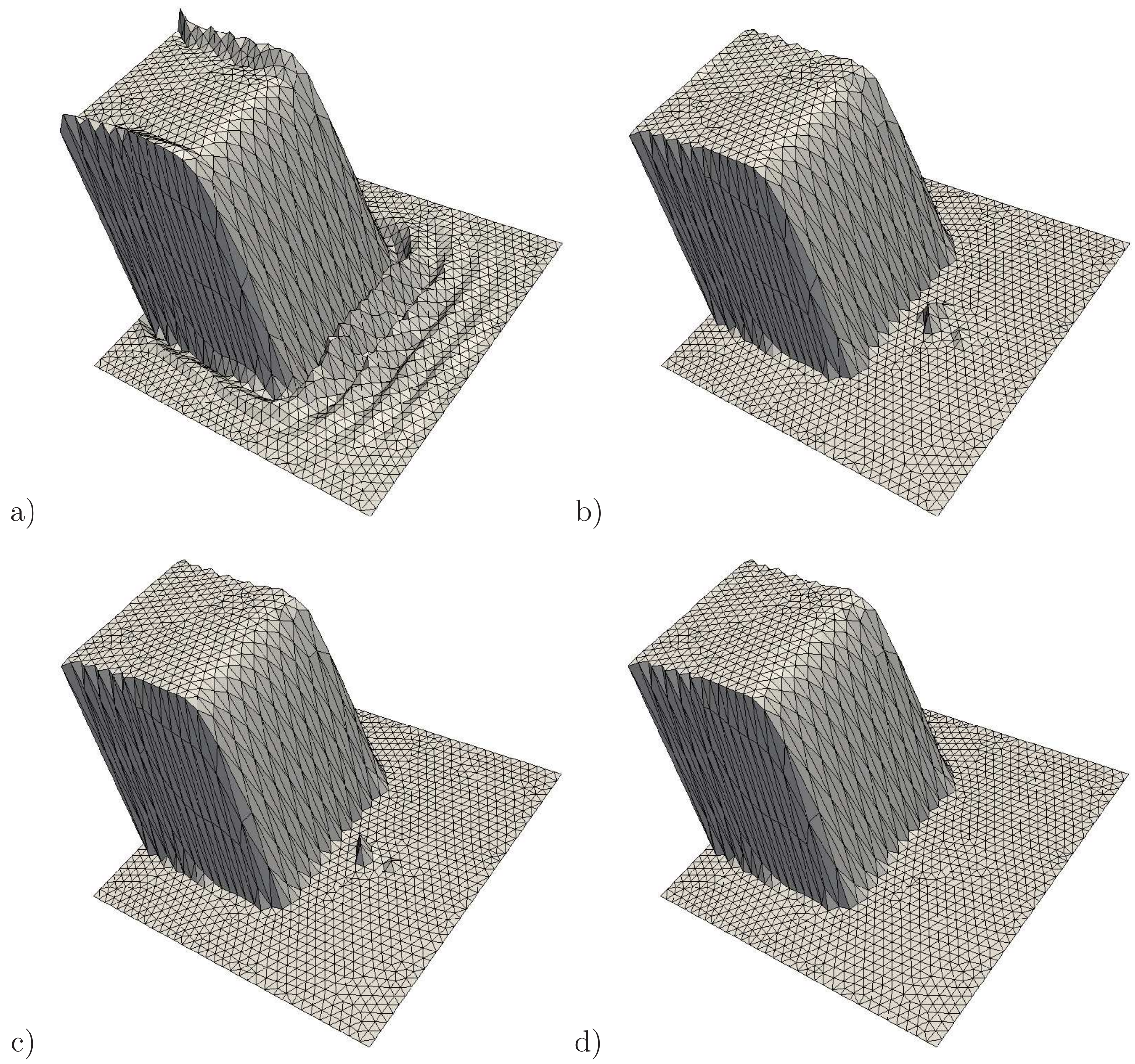

Figure 8: a) SUPG solution to Problem 1 for $\Delta t = 0.005$; and its postprocessed versions: b) with one layer of neighbours; c) with two layers of neighbours; d) with two layers of neighbours and supplementary monotonicity constraints along the line $y = 0$.

Table 2: Errors with respect to the collocated exact solution to Problem 1.

monotonicity relations	solution	error value	
		$\Delta t = 0.005$	$\Delta t = 0.02$
no relations	SUPG	$9.12E-2$	$9.90E-2$
one layer	monotonicity recovering	$8.47E-2$	$9.67E-2$
	postprocessed	$8.06E-2$	$9.63E-2$
two layers	monotonicity recovering	$8.43E-2$	$9.65E-2$
	postprocessed	$8.05E-2$	$9.62E-2$
two layers & line $y=0$	monotonicity recovering	$8.42E-2$	$9.64E-2$
	postprocessed	$8.05E-2$	$9.30E-2$

and its postprocessed versions obtained for different monotonicity relations are shown in Fig. 8.

Table 2 presents errors with respect to the collocated exact solution Φ^*, namely: $\|\bar{u}-\Phi^*\|_w$ for the SUPG, $\|u_M-\Phi^*\|_w$ for the monotonicity recovering stage, and $\|u^*-\Phi^*\|_w$ for the final stage of the postprocessing with $0 \leq C \leq 1$. The numerical values of these errors are in a good agreement with inequalities (4), whatever the monotonicity relations are used. One can observe also the decrease of the errors $\|u_M-\Phi^*\|_w$ and $\|u^*-\Phi^*\|_w$ when the set of the monotonicity relations enlarges. The numerically demonstrated validity of inequalities (4) shows how the postprocessing improves the accuracy with respect to the exact solution. This improvement means that the postprocessing preserves the approximation accuracy.

In the case of Problem 1, we saw that unstructured grids and mesh skewness may require extra efforts for establishing monotonicity relations. Regular meshes, like the one in the next problem, make this process easier.

Problem 2. Consider the following stationary diffusion problem:

$$-\nabla \mathcal{D} \nabla C = 0 \quad \text{in } \Omega \in R^2,$$

where

$$\mathcal{D} = QDQ^T,$$

$$Q = \begin{pmatrix} \cos\frac{\pi}{4} & \sin\frac{\pi}{4} \\ -\sin\frac{\pi}{4} & \cos\frac{\pi}{4} \end{pmatrix} \text{ and } D = \begin{pmatrix} 1 & 0 \\ 0 & 10^{-2} \end{pmatrix}.$$

The domain Ω is a square with a square hole: $\Omega = \Omega' \setminus \Omega''$, where $\Omega' = [-0.5, 0.5]^2$ and $\Omega'' = (-\frac{1}{18}, \frac{1}{18})^2$. Let $\Gamma_0 = \partial\Omega'$ and $\Gamma_1 = \partial\Omega''$ denote the external and internal boundaries of Ω, respectively, see Fig. 9. The boundary conditions are of the Dirichlet type:

$$C = 0 \text{ on } \Gamma_0,$$
$$C = 2 \text{ on } \Gamma_1.$$

Taking into account the symmetry of the problem, we introduce four cones:

$$\Omega_1 = \{(x,y)^T : -x \leq y \leq x\},$$

Table 3: Problem 2: solution minima and maxima excluding the boundary.

	FE	monotonicity recovering	postprocessed
min	-0.309	-0.027	0
max	1.83	1.83	1.67

$$\begin{aligned}\Omega_2 &= \{(x,y)^T : \; -y \le x \le y\},\\ \Omega_3 &= \{(x,y)^T : \; x \le y \le -x\},\\ \Omega_4 &= \{(x,y)^T : \; y \le x \le -y\}.\end{aligned}$$

They cover the domain Ω. Note that the lines $y = x$ and $y = -x$ that define the boundaries of the cones are parallel to the main axes of the diffusion tensor.

It can be easily derived from the formulation of Problem 2, without invoking its exact solution, that if χ' belongs to a subdomain $\Omega \cap \Omega_k$ for some $1 \le k \le 4$, then

$$C(\chi') \ge C(\chi''), \quad \forall \chi'' \in \Omega \cap \Omega_k \text{ such that } \chi'' - \chi' \in \Omega_k. \tag{25}$$

This means that the solution $C(x, y)$ is monotonically non-increasing along any of such directions $\chi'' - \chi'$. Relation (25) allows for defining in Problem 2, for any node χ_i, its monotonicity cone K_i as follows: if node $\chi_i \in \Omega_k$ for some $1 \le k \le 4$, then $K_i = \Omega_k$.

Like in the simplest procedure presented above for Problem 1, only the adjacent nodes are involved here in the construction of monotonicity constraints (M) for Problem 2. In contrast to Problem 1, the simplest procedure guarantees now the complete monotonicity recovering. This is assured by the 36×36 triangular grid of a regular structure used for solving Problem 2, see Fig. 9 (right). To show the completeness, consider any node $j' \in K_i$ which is not adjacent to node i. Due to the grid structure, there exists a node $j \in K_i$ adjacent to i and such that $j' \in K_j$. The spatial relation between these three nodes implies three monotonic relations of which $u_i - u_{j'} \ge 0$ is redundant, because it follows from the other two relations: $u_i - u_j \ge 0$ and $u_j - u_{j'} \ge 0$. Therefore, one can skip including $u_i - u_{j'} \ge 0$ in (M) without any loss of the monotonicity. The same refers to $u_j - u_{j'} \ge 0$ if nodes j and j' are not adjacent. Thus, any monotonicity relation $u_i - u_{j'} \ge 0$ between non-adjacent nodes, although not included in (M), is implicitly presented in (M) by a chain of constraints involving adjacent nodes only.

Fig. 10 (left) depicts the solution produced by the hybrid mixed finite element method (HMFEM). This figure shows spurious oscillations resulting in negative values of C, see Fig. 11 (left). The minimal value is close to -0.31, see Table 3. The same table presents the minimal and maximal values for all the produced solutions. The solution to problem (1M) recovers the monotonicity completely and removes the oscillations. However, it still contains negative values, see Fig. 11 (right). Table 3 shows that, at the monotonicity recovering stage, the most severe violation of the non-negativity condition is reduced by one order of magnitude. The postprocessed solution shown in Fig. 10 (right) is non-negative, meets the conservativity requirement and retains the recovered monotonicity.

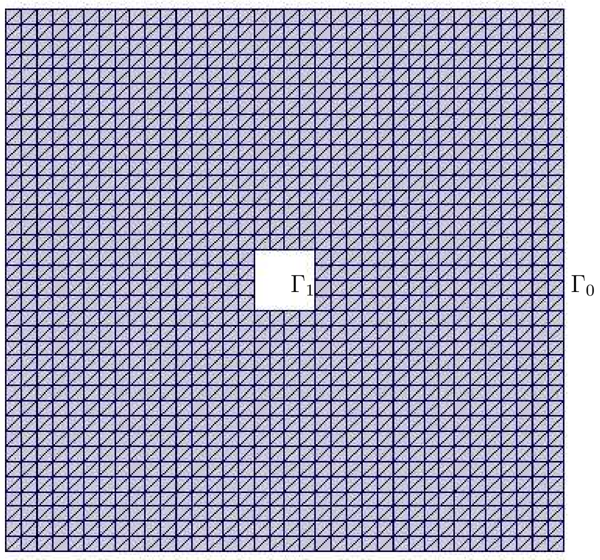

Figure 9: Computational domain and grid for Problem 2.

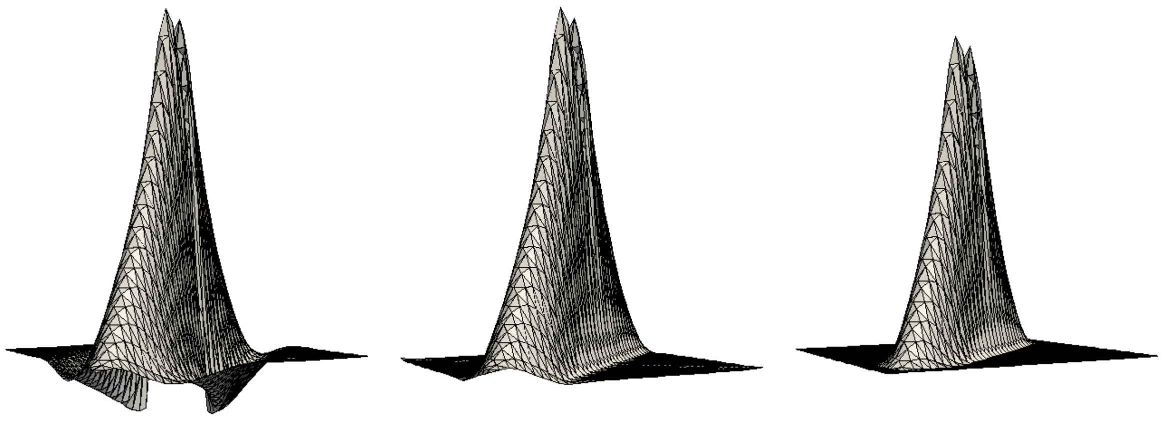

Figure 10: Solutions to Problem 2: HMFEM (left), monotonicity recovering (center), postprocessed (right).

Table 4: CPU time in seconds for HMFEM and the postprocessing.

number of unknowns	HMFEM time	postprocessing time
10240	0.37	0.03
40960	1.51	0.15
163840	7.20	0.50

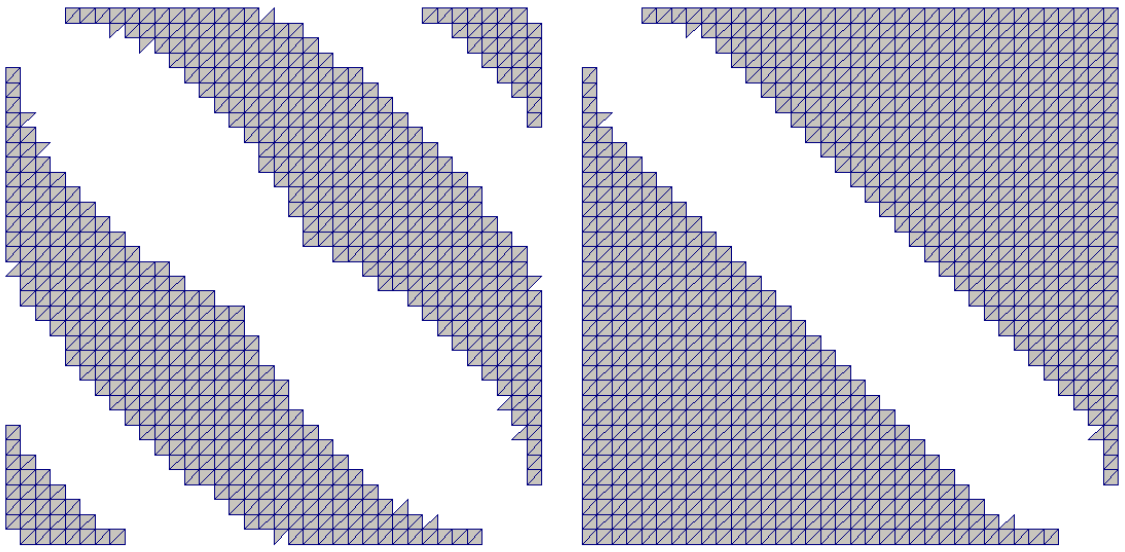

Figure 11: Location of negative concentrations in the solutions to Problem 2: HMFEM (left) and monotonicity recovering (right).

Table 4 enables comparison of the CPU time required for running the HMFEM solver and the postprocessing algorithm. It shows how the time scales with the number of unknowns. One can see that the postprocessing is much faster than HMFEM. Moreover, the postprocessing grows almost linearly with the number of unknowns, and it is approximately one order faster than HMFEM.

5 Conclusions

We have developed a computationally efficient procedure of postprocessing FE solutions. It is aimed at recovering the monotonicity that may be partially lost in the available FE solution. The postprocessed solution satisfies the maximum principle and the conservativity condition. As it has been shown, it also retains the accuracy of the discrete FE approximation. The quality of the recovered monotonicity depends on how completely constraints (M) model the monotonicity. Our examples show how some properties of the original problem, like a symmetry of the expected solution, can be exploited for constructing constraints (M), but we can not offer any universal procedure. Nevertheless, our monotonic smoothing can be applied in subdomains where it is still possible to establish monotonicity relations.

Acknowledgements

This work was partially supported by the Royal Swedish Academy of Sciences, the Russian Foundation of Basic Research (grant 11-01-00971), the Federal Program “Scientific and pedagogical staff of innovative Russia”.

References

[1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[2] R.E. Barlow, D.J. Bartholomew, J.M. Bremner, H.D. Brunk, Statistical Inference under Order Restrictions, Wiley, New York, 1972.

[3] M.S. Bazaraa, H.D. Sherali, C.M. Shetty, Nonlinear Programming: Theory and Algorithms, John Wiley, New York, 1993.

[4] E. Bertolazzi, G. Manzini, A second-order maximum principle preserving finite volume method for steady convection-diffusion problems, SINUM 43 (2005) 2172–2199.

[5] H. Block, S. Qian, A. Sampson, Structure algorithms for partially ordered isotonic regression, J. Comput. Graph. Stat. 3 (1994) 285–300.

[6] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, 32 Computer Methods in Applied Mechanics and Engineering (1982) 199–259.

[7] O. Burdakov, O. Sysoev, A. Grimvall, M. Hussian, An $O(n^2)$ algorithm for isotonic regression problems, in: G. Di Pillo and M. Roma (Eds.), Large Scale Nonlinear Optimization, Springer-Verlag, 2006, pp. 25–33.

[8] O. Burdakov, A. Grimvall, O. Sysoev, Data preordering in generalized PAV algorithm for monotonic regression, J. Comput. Math. 24 (2006) 771–790.

[9] O. Burdakov, A. Grimvall, O. Sysoev, Generalized PAV algorithm with block refinement for partially ordered monotonic regression, in: A. Feelders and R. Potharst (Eds.), Proc. of the Workshop on Learning Monotone Models from Data at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 2009, pp. 23–37.

[10] O. Sysoev, O. Burdakov, A. Grimvall, A segmentation-based algorithm for large-scale partially ordered monotonic regression, Computational Statistics and Data Analysis 55 (2011) 2463-2476.

[11] O. Burdakov, A. Grimvall, O. Sysoev, MATLAB algorithms for solving partially ordered monotonic regression problems.
http://www.mai.liu.se/~olbur/SOFTWARE/

[12] E. Burman, A. Ern, Discrete maximum principle for Galerkin approximations of the Laplace operator on arbitrary meshes, Compt. Rend. Mathematique 338 (2004) 641–646.

[13] P.G. Ciarlet, Discrete Maximum Principle for Finite-Difference Operators, Aequationes Mathematicae 4 (1970) 338–352.

[14] P.G. Ciarlet, The Finite Element Method for Elliptic Problems, North Holland, New York, 1978.

[15] P. G. Ciarlet, P.-A. Raviart, Maximum principle and uniform convergence for the finite element method, Comput. Methods Appl. Mech. Eng. 2 (1973) 17–31.

[16] A. Danilov, Yu. Vassilevski, A monotone nonlinear finite volume method for diffusion equations on conformal polyhedral meshes, Russian J. Numer. Anal. Math. Modelling, 24 (2009) 207–227.

[17] Z. Dostál, Optimal Quadratic Programming Algorithms: With Applications to Variational Inequalities, Springer, 2009.

[18] F. Gao, Y. Yuan, D. Yang, An upwind finite-volume element scheme and its maximum-principle-preserving property for nonlinear convection-diffusion problem, Int. J. Numer. Meth. Fl. 56 (2008) 2301–2320.

[19] Ph.E. Gill, W. Murray, M.A. Saunders, SNOPT: An SQP algorithm for large-scale constrained optimization, SIAM Review 47 (2005) 99–131.

[20] Ph.E. Gill, W. Murray, M.H. Wright, Practical Optimization. Academic Press, London, 1981.

[21] V. John, P. Knobloch, On spurious oscillations at layers diminishing (SOLD) methods for convection-diffusion equations: Part I — A review, Comput. Methods Appl. Mech. Eng. 196 (2007) 2197–2215.

[22] I. Kapyrin, A family of monotone methods for the numerical solution of three-dimensional diffusion problems on unstructured tetrahedral meshes, Dokl. Math. 76 (2007) 734–738.

[23] S. Korotov, M. Křížek, P. Neittaanmäki, Weakened acute type condition for tetrahedral triangulations and the discrete maximum principle, Math. Comp. 70 (2001) 107–119.

[24] D. Kuzmin, On the design of general-purpose flux limiters for finite element schemes. I. Scalar convection, J. Comput. Phys. 219 (2006) 513–531.

[25] C. Le Potier, Schema volumes finis monotone pour des operateurs de diffusion fortement anisotropes sur des maillages de triangle non structures, Comp. R. Mathematique 341 (2005) 787–792.

[26] C. Le Potier, Finite volume scheme satisfying maxcimum and minimum principles for anisotropic diffusion operators. in: R. Eymard, J.-M- Hedard (Eds.), Finite Volumes for Complex Applications V, Wiley-ISTE, 2008, pp. 103–118.

[27] F.J. Leij, J.H. Dane, Analytical solutions of the one-dimensional advection equation and two- or three-dimensional dispersion equation, Water Resour Res. 26 (1990) 1475-1482.

[28] K. Lipnikov, D. Svyatskiy, M. Shashkov, Yu. Vassilevski, Monotone finite volume schemes for diffusion equations on unstructured trangular ans shape-regular polygonal meshes, J. Comput. Phys. 227 (2007) 492–512.

[29] K. Lipnikov, D. Svyatskiy, Y. Vassilevski, Interpolation-free monotone finite volume method for diffusion equations on polygonal meshes, J. Comput. Phys. 228 (2009) 703–716.

[30] K. Lipnikov, D. Svyatskiy, Y. Vassilevski, Interpolation-free monotone finite volume method for diffusion equations on polygonal meshes, J. Comput. Phys. 229 (2010) 4017–4032.

[31] K. Lipnikov, Y. Vassilevski, Ani2D — a 2D generator of anisotropic meshes, http://sourceforge.net/projects/ani2d/

[32] R. Liska, M. Shashkov, Enforcing the discrete maximum principle for linear finite element solutions of second-order elliptic problems, Commun. Comput. Phys. 3 (2008) 852–877.

[33] W.L. Maxwell, J.A. Muchstadt, Establishing consistent and realistic reorder intervals in production-distribution systems, Oper. Res. 33 (1985) 1316–1341.

[34] K. Nikitin, Yu. Vassilevski, A monotone nonlinear finite volume method for advection-diffusion equations on unstructured polyhedral meshes in 3D, Russian J. Numer. Anal. Math. Modelling, 25 (2010) 335–358.

[35] J. Nocedal, S.J. Wright, Numerical Optimization (2nd ed.), Springer, 2006.

[36] R. Roundy, A 98% effective lot-sizing rule for a multiproduct multistage production/inventory system, Math. Oper. Res. 11 (1986) 699–727.

[37] T. Robertson, F.T. Wright, R.L. Dykstra, Order Restricted Statistical Inference, Wiley, New York, 1988.

[38] V.R. Santos, On the strong maximum principle for some piecewise linear finite element approximate problems of nonpositive type, J. Fac. Sci. Univ. Tokyo Sect. IA Math. 29 (1982) 473–491.

[39] K. Schittkowski, Numerical Data Fitting in Dynamical Systems - A Practical Introduction with Applications and Software, Kluwer Academic Publishers, Dotrecht, 2002.

[40] Yu. Vassilevski, I. Kapyrin, Two splitting schemes for nonstationary convection-diffusion problems on tetrahedral meshes, Comp. Math. Math. Phys. 48 (2008) 1349–1366.

[41] G. Yuan, Z. Sheng, Monotone finite volume schemes for diffusion equationson polygonal meshes, J. Comp. Phys. 227 (2008) 6288–6312.