Applied Mathematics and Computation 219 (2013) 6182–6196

Contents lists available at SciVerse ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Operator-splitting finite element algorithms for computations of high-dimensional parabolic problems

Sashikumaar Ganesan [a,*], Lutz Tobiska [b]

[a] Numerical Mathematics and Scientific Computing, Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore 560 012, India
[b] Institute of Analysis and Numerical Mathematics, Otto-von-Guericke University, PF 4120, D-39016 Magdeburg, Germany

ARTICLE INFO

Keywords:
Operator-splitting method
Finite element method
Parabolic equations
High-dimensional problems

ABSTRACT

An operator-splitting finite element method for solving high-dimensional parabolic equations is presented. The stability and the error estimates are derived for the proposed numerical scheme. Furthermore, two variants of fully-practical operator-splitting finite element algorithms based on the quadrature points and the nodal points, respectively, are presented. Both the quadrature and the nodal point based operator-splitting algorithms are validated using a three-dimensional (3D) test problem. The numerical results obtained with the full 3D computations and the operator-split 2D + 1D computations are found to be in a good agreement with the analytical solution. Further, the optimal order of convergence is obtained in both variants of the operator-splitting algorithms.

1. Introduction

The numerical solution of partial differential equations in many mathematical models depends not only on time and space but also on some other properties of the considered problem. For instance, the population balance equation (PBE) in a population balance system with one internal coordinate depends on the time, the physical space and a property of the particles, *e.g.*, the size of the particles. Since the PBE contains also derivatives with respect to the properties of the particles, it is posed on a high-dimensional domain in comparison to all other equations in the population balance system [4,12,13,15]. Another more challenging example is the FENE Fokker–Planck equation modeling polymeric fluids [14]. In fact, for a flow domain contained in $\mathbb{R}^d$, the polymer configuration domain for a beadspring chain polymer model consisting of $M+1$ beads and M springs, the configuration space domain is contained in $\mathbb{R}^{Md}$ and therefore the Fokker–Planck equation is posed on a domain in $\mathbb{R}^{d+Md}$. Even in the simplest case $M=1$ the Fokker–Planck equation has to be solved on a domain in $\mathbb{R}^{2d}$, i.e., four dimensions when $d=2$ and six dimensions when $d=3$.

The numerical solution of partial differential equations on high-dimensional domains are more challenging due to higher storage requirements and computational complexity. To overcome these challenges the sparse grid method [3,9,10] can be used. In the sparse grid method, a high-dimensional equation can be solved by constructing a tensor product sparse grid space using an one-dimensional multilevel basis for each coordinate direction. Similarly, in the space-time sparse grid method, the sparse grid spaces are constructed by a tensor product of a d-dimensional basis in space and an one-dimensional basis in time.

[*] Corresponding author.
E-mail addresses: sashi@serc.iisc.in (S. Ganesan), tobiska@ovgu.de (L. Tobiska).
URLs: http://www.serc.iisc.ernet.in/~sashi/ (S. Ganesan), http://www-ian.math.uni-magdeburg.de/home/tobiska (L. Tobiska).

0096-3003/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.amc.2012.12.027

Another popular method for solving high-dimensional equations is the operator-splitting method developed in [5,6] to replace a parabolic differential equation in two space dimensions by solving two one-dimensional problems. This basic idea can be extended to split a high-dimensional equation into a system of low-dimensional equations and solve each low-dimensional equation separately. In most of the previous studies the finite difference method is used to solve the operator-split low-dimensional equations, for example, see [11,18,20] and the references therein. Applying the operator-splitting in the context of finite difference method is simple and it can easily be extended to high-dimensional problems. However, it is well-known that finite difference methods are more difficult to apply for problems with complex domains. The operator-splitting has also been used in the finite volume approximation of the population balance equation [11,16,17,19,22].

Recently, the operator-splitting has been used in the tailored finite element and spectral method to solve the high-dimensional FENE Fokker–Planck equation in [14]. An advantage of solving a system of low-dimensional equations separately is that we can use tailored numerical methods to solve low-dimensional equations in the system of different type. For example, in [7] the standard Galerkin finite element method and the Streamline Upwind Petrov Galerkin (SUPG) finite element method has been combined whereas in [1] the discontinuous Galerkin method and the local projection type stabilization has been considered.

In this paper, we present an operator-splitting scheme for finite element approximation of high-dimensional parabolic problems in all spaces. Moreover, to use the operator-splitting in the finite element computations, we present two new algorithms based on the quadrature points and the nodal points, respectively. These algorithms are fully practical and easy to implement. The operator-splitting for the high-dimensional equations is not new, however to the best of our knowledge, operator-splitting algorithms in the context of finite element approximation in all space dimensions has not been proposed before in the literature.

The paper is organized as follows. In the next Section, we derive the standard discrete and algebraic forms of a parabolic problem. Then, we present a finite element analysis of the operator-split parabolic problem in Section 3. After that, in Section 4, the two variants of operator-splitting algorithms in the context of finite element method are described in detail. Finally, numerical results are presented in Section 5 to validate the proposed operator-splitting algorithms.

2. Equations on a high-dimensional domain

Let $\Omega := \Omega_X \times \Omega_L$, where $\Omega_X \subset \mathbb{R}^p$ and $\Omega_L \subset \mathbb{R}^q$ be the considered high-dimensional $(d = p + q)$ domain. Suppose the equation of interest to be solved is

$$\begin{aligned} &\frac{\partial u}{\partial t} + A_x u + A_L u = 0 \quad \text{in } (0,T] \times \Omega, \\ &u = u_D \quad \text{on } (0,T] \times \partial\Omega, \\ &u(0,x,\ell) = u_0, \end{aligned} \tag{1}$$

where the operators A_x and A_L represent physical phenomena (convection, diffusion, reaction etc.) in Ω_X and Ω_L, respectively. To solve the Eq. (1) in Ω, the standard finite element method can be used when the dimension of the problem is small enough, say $d \leqslant 3$. However, when $d > 3$ or as mentioned in the introduction, when d is higher than the dimension of the other equations in the system, the numerical solution of (1) with the conventional methods become more expensive. In such cases, we can use the operator-splitting method provided that the high-dimensional domain can be defined as a Cartesian product of low-dimensional subdomains. The basic idea is to split the operators in (1) based on their dependency on the subdomains, and solve a system of low-dimensional (say, less than or equal to 3) problems in subdomains separately. To solve the system of low-dimensional problems, we can use the standard finite element method. However, special algorithms have to be implemented in order to apply the idea of operator-splitting in finite element computations. As already mentioned in the introduction, we can use tailored numerical methods to solve the low-dimensional equations in the system. Since the present study emphasizes on developing algorithms for the implementation of the operator-splitting method, we restrict our representation to a simple model problem and on the standard Galerkin finite element method applied to all equations in the collection.

2.1. Model equation

For the methodological development of operator-splitting algorithms for solving high-dimensional problems in the context of the finite element method, let us consider the following parabolic problem:

$$\begin{aligned} &\frac{\partial u}{\partial t} - \Delta u = f \quad \text{in } (0,T] \times \Omega, \\ &u = 0 \quad \text{on } (0,T] \times \partial\Omega, \\ &u(0,x,\ell) = u_0. \end{aligned} \tag{2}$$

As mentioned above, we assume that the domain Ω can be defined as a Cartesian product of two subdomains, *i.e.*, $\Omega := \Omega_X \times \Omega_L$, and let for simplicity $\Omega_X = (0,1)^2,\ \Omega_L = (0,1)$. Here, $u(t,x,\ell)$ is the unknown, t is the time in the given time interval $[0,T], x = (\mathcal{X}_1, \mathcal{X}_2) \in \Omega_X,\ \ell \in \Omega_L, f$ is a given source, and u_0 is an initial value.

2.2. Weak formulation

Let $(\cdot,\cdot)$ and $||\cdot||$ be the L^2-inner product and norm over $\Omega := \Omega_X \times \Omega_L$:

$$(v,q) := \int_{\Omega_X\times\Omega_L} v(x,\ell)q(x,\ell), \quad ||v||^2 = (v,v) \quad \forall v,q \in L^2(\Omega).$$

With these definitions, the weak form of (2) reads: For given $u_0 \in L^2(\Omega)$ and $f \in L^2((0,T]\times\Omega)$, find $u \in L^2(0,T;H_0^1(\Omega))$ with$u' \in L^2(0,T;H^{-1}(\Omega))$ such that

$$\begin{aligned}&\frac{d}{dt}(u(t),v) + (\nabla u(t),\nabla v) = (f(t),v) \quad \forall v \in H_0^1(\Omega),\\&(u(0),v) = (u_0,v) \quad \forall v \in L^2(\Omega),\end{aligned} \tag{3}$$

where $f(t) := f(t,\cdot)$. Note that $u \in L^2(0,T;H_0^1(\Omega))$ and $u' \in L^2(0,T;H^{-1}(\Omega))$ implies that $t\mapsto u(t)$ as a mapping from [0,T] in $L^2(\Omega)$ is continuous. Thus, $u(0)$ is well defined.

2.3. Finite element spaces for operator-splitting method

Let $H^m(\Omega_X)$ and $H^m(\Omega_L), m \geqslant 0$ be the usual Sobolev spaces containing L^2-functions with weak derivatives of order m in L^2. Then, we define

$$H^{m,m}(\Omega) := H^m(\Omega_X;H^m(\Omega_L)) \cap H^m(\Omega_L;H^m(\Omega_X)), \tag{4}$$

with associated norms and seminorms

$$\|v\|^2_{m,m} := \sum_{|\beta|\leqslant m,}\sum_{|\alpha|\leqslant m}\|\partial_\ell^\beta\partial_x^\alpha v\|^2_{L^2(\Omega)}, \quad |v|^2_{m,m} := \sum_{|\beta|=m,}\sum_{|\alpha|=m}\|\partial_\ell^\beta\partial_x^\alpha v\|^2_{L^2(\Omega)}.$$

The space $H^{1,1}(\Omega)$ is slightly more regular than the standard space $H^1(\Omega)$, in particular the mixed partial derivatives of second order are bounded in $L^2(\Omega)$.

Suppose $V_h \subset H_0^1(\Omega_X)$ and $W_h \subset H_0^1(\Omega_L)$ are conforming finite element spaces with basis functions $\phi_h := \phi_i, i = 1,2,\ldots,\mathcal{M}$ and $\psi_h := \psi_k, k = 1,2,\ldots,\mathcal{N}$, respectively such that

$$V_h = \text{span}\{\phi_i\}, \quad W_h = \text{span}\{\psi_k\}. \tag{5}$$

Now,

$$V_h \otimes W_h = \left\{\xi_h : \xi_h = \sum_{i=1}^{\mathcal{M}}\sum_{k=1}^{\mathcal{N}}\xi_{i,k}\phi_i\psi_k; \xi_{i,k} \in \mathbb{R}\right\} \subset H^{1,1}(\Omega).$$

Further, we define the finite element ansatz and test functions as

$$\begin{aligned}&u_h(t,x,\ell) = \sum_{j=1}^{\mathcal{M}}\sum_{l=1}^{\mathcal{N}}u_{j,l}(t)\phi_j(x)\psi_l(\ell), \quad v_h = \sum_{i=1}^{\mathcal{M}}\sum_{k=1}^{\mathcal{N}}v_{i,k}\phi_i\psi_k,\\&\nabla_x u_h = \sum_{j=1}^{\mathcal{M}}\sum_{l=1}^{\mathcal{N}}u_{j,l}(\nabla_x\phi_j)\psi_l, \quad \nabla_\ell u_h = \sum_{j=1}^{\mathcal{M}}\sum_{l=1}^{\mathcal{N}}u_{j,l}\phi_j(\nabla_\ell\psi_l),\end{aligned} \tag{6}$$

where ∇_x and ∇_ℓ are the gradient operators in Ω_X and Ω_L, respectively. Using the above definitions, the semi-discrete (in space) form of (3) can be written as: For given u_0 and $f(t)$, find $u_h(t) \in V_h \otimes W_h$ such that

$$\begin{aligned}&\frac{d}{dt}(u_h(t),v_h) + (\nabla u_h(t),\nabla v_h) = (f(t),v_h) \quad \forall v_h \in V_h\otimes W_h,\\&(u_h(0),v_h) = (u_0,v_h) \quad \forall v_h \in V_h\otimes W_h.\end{aligned} \tag{7}$$

2.4. Temporal discretization

Let $0 = t^0 < t^1 < \ldots < t^N = \mathrm{T}$ be a uniform decomposition of the considered time interval $[0,\mathrm{T}]$, and $\delta t = t^n - t^{n-1}, 1 \leqslant n \leqslant N$, be the time step size. The general form of the θ-scheme for the semi-discrete form (7) in the interval (t^{n-1},t^n) can be written as

$$\left(\frac{u_h^n - u_h^{n-1}}{\delta t}, v_h\right) + (1-\theta)(\nabla u_h^{n-1},\nabla v_h) + \theta(\nabla u_h^n,\nabla v_h) = (f^{n-1+\theta},v_h), \tag{8}$$

where $f^{n-1+\theta} = (1-\theta)f(t^{n-1}) + \theta f(t^n)$. To rewrite the discrete form into an algebraic form, we define the mass and stiffness matrices and the load vector as follows

$$M := M_x \otimes M_\ell, \quad A := A_x \otimes M_\ell + M_x \otimes A_\ell, \quad F^{n-1+\theta}_{i,k} := \int_\Omega f^{n-1+\theta}\phi_i\psi_k, \tag{9}$$

where

$$\begin{aligned} M_{x_{ij}} &= \int_{\Omega_X} \phi_i\phi_j, \quad A_{x_{ij}} = \int_{\Omega_X} \nabla_x\phi_i \cdot \nabla_x\phi_j, \\ M_{\ell_{kl}} &= \int_{\Omega_L} \psi_k\psi_l, \quad A_{\ell_{kl}} = \int_{\Omega_L} \nabla_\ell\psi_k \cdot \nabla_\ell\psi_l, \end{aligned} \tag{10}$$

where $\otimes$ denotes the Kronecker product of two matrices. For example, the $\mathcal{M}\mathcal{N} \times \mathcal{M}\mathcal{N}$ entries in the mass matrix are given by

$$M = M_x \otimes M_\ell := \begin{bmatrix} [\Phi_{1,1}]_{k,l} & \cdot & \cdot & \cdot & [\Phi_{1,\mathcal{M}}]_{k,l} \\ \cdot & \cdot & & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & & \cdot & \cdot \\ [\Phi_{\mathcal{M},1}]_{k,l} & \cdot & \cdot & \cdot & [\Phi_{\mathcal{M},\mathcal{M}}]_{k,l} \end{bmatrix}_{\mathcal{M}\mathcal{N}\times\mathcal{M}\mathcal{N}},$$

where the entries in the $\mathcal{N} \times \mathcal{N}$ block matrix $[\Phi_{i,j}]_{k,l}$, $1 \leqslant k,l \leqslant \mathcal{N}$ are evaluated by

$$[\Phi_{i,j}]_{k,l} := \int_{\Omega_X} \phi_i\phi_j \int_{\Omega_L} \psi_k\psi_\ell.$$

2.4.1. Backward Euler scheme

Choosing $\theta = 1$ in (8), we get the backward Euler scheme in the time interval (t^{n-1}, t^n)

$$\left(\frac{u_h^n - u_h^{n-1}}{\delta t}, v_h\right) + (\nabla u_h^n, \nabla v_h) = (f^n, v_h). \tag{11}$$

Substituting the definitions of the finite element functions (6), we have

$$\begin{aligned} \int_\Omega u_h^n v_h &= \sum_{j=1}^{\mathcal{M}}\sum_{l=1}^{\mathcal{N}}\sum_{i=1}^{\mathcal{M}}\sum_{k=1}^{\mathcal{N}} \int_\Omega u_{j,l}^n \phi_j\psi_l\phi_i\psi_k, \\ \int_\Omega \nabla u_h^n \cdot \nabla v_h &= \sum_{j=1}^{\mathcal{M}}\sum_{l=1}^{\mathcal{N}}\sum_{i=1}^{\mathcal{M}}\sum_{k=1}^{\mathcal{N}} \int_\Omega u_{j,l}^n \big((\nabla_x\phi_j)\psi_l \cdot (\nabla_x\phi_i)\psi_k + \phi_j(\nabla_\ell\psi_l)\phi_i \cdot (\nabla_\ell\psi_k)\big). \end{aligned}$$

Applying these representations in (11) and employing the definition of the matrices (10), the algebraic form of the backward Euler scheme reads:

$$\{(M_x \otimes M_\ell) + \delta t((A_x \otimes M_\ell) + (M_x \otimes A_\ell))\}\overline{U}^n = \delta t F^n + (M_x \otimes M_\ell)\overline{U}^{n-1}. \tag{12}$$

Here, $\overline{U}^n = \mathrm{vec}(U^n)$ is the vectorization of the solution matrix $U^n = [u_{j,l}^n]_{\mathcal{M}\times\mathcal{N}}$. Using the Kronecker product definitions (9), we get the standard representation in the algebraic form:

$$(M + \delta t A)\overline{U}^n = \delta t F^n + M\overline{U}^{n-1}. \tag{13}$$

For the backward Euler scheme, we can derive an error estimate with first order accurate in time.

Theorem 1 (See Theorem 1.5 in [21]). *Let u and u_h^n be the solutions of (2) and (11), respectively. Further, let $V_h \otimes W_h$ be a finite element space of order r, and the initial solution u_0 be approximated by $u_h(0)$ of order r. Then, we have for $n \geqslant 0$*

$$||u_h^n - u(t^n)|| \leqslant Ch^r\left(||u_0||_r + \int_0^{t^n} ||u_t||_r ds\right) + \delta t \int_0^{t^n} ||u_{tt}|| ds,$$

where u_t and u_{tt} are the first and second derivatives of u with respective to time t.

Remark 1. Choosing $\theta = 0.5$ in (8), we get the Crank–Nicolson scheme in the time interval (t^{n-1}, t^n) as

$$\left(\frac{u_h^n - u_h^{n-1}}{\delta t}, v_h\right) + \frac{1}{2}(\nabla u_h^n, \nabla v_h) = (f^{n-1/2}, v_h) - \frac{1}{2}(\nabla u_h^{n-1}, \nabla v_h). \tag{14}$$

The algebraic form of the Crank–Nicolson scheme can be written as:

$$\left\{(M_x\otimes M_\ell)+\frac{\delta t}{2}((A_x\otimes M_\ell)+(M_x\otimes A_\ell))\right\}\overline{U}^n=\delta tF^{n-1/2}+\left\{(M_x\otimes M_\ell)-\frac{\delta t}{2}((A_x\otimes M_\ell)+(M_x\otimes A_\ell))\right\}\overline{U}^{n-1}. \tag{15}$$

For the Crank–Nicolson scheme, the following error estimate holds.

Theorem 2 (See Theorem 1.6 in [21]). *Let u and u_h^n be the solutions of (2) and (14), respectively. Further, let $V_h\otimes W_h$ be a finite element space of order r, and the initial solution u_0 be approximated by $u_h(0)$ of order r. Then, we have for $n\geqslant 0$*

$$\|u_h^n-u(t^n)\|\leqslant Ch^r\left(\|u_0\|_r+\int_0^{t^n}\|u_t\|_r ds\right)+C(\delta t)^2\int_0^{t^n}(\|u_{ttt}\|+\|\Delta u_{tt}\|)ds,$$

where u_{ttt} is the third derivative of u respective to time t.

3. Operator-splitting method

In the previous section, we derived the standard Galerkin discrete form of the considered problem (2). In this section, we first split the continuous problem (2) into two subproblems based on the dependency of the Laplace operator on the spatial directions. After that we derive a discrete form for the operator-split equations.

3.1. Operator-split equations

To apply the operator-splitting method for the considered model problem (2), we denote

$$\Delta=\Delta_x+\Delta_\ell,\quad \Delta_x=\frac{\partial^2}{\partial\mathcal{X}_1^2}+\frac{\partial^2}{\partial\mathcal{X}_2^2},\quad \Delta_\ell=\frac{\partial^2}{\partial\ell^2}.$$

Using this notation in (2), we get

$$\begin{aligned}&\frac{\partial u}{\partial t}-\Delta_x u-\Delta_\ell u=f\quad\text{in } (0,T]\times\Omega_X\times\Omega_L,\\ &u=0\quad\text{on } (0,T]\times\partial\Omega,\\ &u(0,x,\ell)=u_0.\end{aligned} \tag{16}$$

After applying the temporal discretization, we split the semi-discrete form of the above problem (16) into two subproblems, (one in L-direction and another in X-direction) in the interval (t^{n-1},t^n) as follows:

Step 1 (L-direction)
Find $\hat{u}(t,x,\ell)$ for all $x\in\Omega_X$ such that

$$\begin{aligned}&\frac{\partial\hat{u}}{\partial t}-\Delta_\ell\hat{u}=f\quad\text{in } (t^{n-1},t^n)\times\Omega_L,\\ &\hat{u}=0\quad\text{on } (t^{n-1},t^n)\times\partial\Omega_L,\\ &\hat{u}(t^{n-1},x,\ell)=u^{n-1}.\end{aligned} \tag{17}$$

Step 2 (X-direction)
Find $\tilde{u}(t,x,\ell)$ for all $\ell\in\Omega_L$ such that

$$\begin{aligned}&\frac{\partial\tilde{u}}{\partial t}-\Delta_x\tilde{u}=0\quad\text{in } (t^{n-1},t^n)\times\Omega_X,\\ &\tilde{u}=0\quad\text{on } (t^{n-1},t^n)\times\partial\Omega_X,\\ &\tilde{u}(t^{n-1},x,\ell)=\hat{u}(t^n,x,\ell).\end{aligned} \tag{18}$$

Here, we used the Lie's operator-splitting method for the Eq. (16), for other variants of operator-splitting methods we refer to [8].

Remark 2. For brevity, the diffusion coefficient values in (16) are assumed to be one. Nevertheless, different diffusion coefficients ϵ_x and ϵ_ℓ in X- and L-directions, respectively, can be used, and it will result in an $\epsilon_x\epsilon_\ell$ factor in the mixed derivative term of the bilinear form (23). However, in practical applications the values of ϵ_x and ϵ_ℓ will always be less than one, and therefore we expect a reduction of the operator-splitting error.

Remark 3. The source term f in (16) is included in the L-direction Eq. (17). Nevertheless, when the contribution of the source term is significant to the diffusive phenomena in Ω_X, it is natural to include it in the X-direction Eq. (18). Another possibility is to split the right-hand side into the convex linear combination $f = \alpha f + (1-\alpha)f,\ \alpha \in [0,1]$, and to include the first part in the L-direction equation and the second part in the X-direction equation, respectively.

Remark 4. Note that, we have Dirichlet boundary condition on the entire boundary $\partial\Omega$, thus in the L-direction, it is enough to solve the Eq. (17) for the inner points of the X-direction space, *i.e.*, for all $x \in \Omega_X$. Similarly, in the X-direction, it is enough to solve the Eq. (18) for the inner points of L-direction space, *i.e.*, for all $\ell \in \Omega_L$.

3.2. Discrete form of the operator-split equations

Let $x_j \in \Omega_X, j = 1,\dots,N\_XP$, and $\ell_l \in \Omega_L, l = 1,\dots,N\_LP$ be the Cartesian coordinates which are necessary to evaluate nodal functionals of the finite element spaces V_h and W_h, respectively. Further, we define the finite element functions $\hat{u}_h(x_j,\cdot) \in W_h$ and $\tilde{u}_h(\cdot,\ell_l) \in V_h$ as

$$\hat{u}_h(x_j,\ell) := \sum_{l=1}^{N\_LP} \hat{u}_{j,l}\psi_l(\ell), \quad \tilde{u}_h(x,\ell_l) := \sum_{j=1}^{N\_XP} \tilde{u}_{j,l}\phi_j(x).$$

Using these definitions in the spatial discretization, the discrete forms of (17) and (18) in the time interval (t^{n-1}, t^n) read:

Step 1 (L-direction)

For given $\hat{u}_h^{n-1}(x_j,\ell) = u_h^{n-1}(x_j,\ell)$ and $f^{n-1+\theta}$, find $\hat{u}_h^n(x_j,\ell) \in W_h$ such that for all $\psi_h \in W_h$ and for all $j = 1,\dots,N\_XP$,

$$\left(\frac{\hat{u}_h^n - \hat{u}_h^{n-1}}{\delta t}, \psi_h\right)_{\Omega_L} + (1-\theta)(\nabla_\ell \hat{u}_h^{n-1}, \nabla_\ell \psi_h)_{\Omega_L} + \theta(\nabla_\ell \hat{u}_h^n, \nabla_\ell \psi_h)_{\Omega_L} = (f^{n-1+\theta}, \psi_h)_{\Omega_L}. \tag{19}$$

Step 2 (X-direction)

For the given $\tilde{u}_h^{n-1}(x,\ell_l) = \hat{u}_h^n(x,\ell_l)$, find $\tilde{u}_h^n(x,\ell_l) \in V_h$ such that for all $\phi_h \in V_h$ and for all $l = 1,\dots,N\_LP$,

$$\left(\frac{\tilde{u}_h^n - \tilde{u}_h^{n-1}}{\delta t}, \phi_h\right)_{\Omega_X} + (1-\theta)(\nabla_x \tilde{u}_h^{n-1}, \nabla_x \phi_h)_{\Omega_X} + \theta(\nabla_x \tilde{u}_h^n, \nabla_x \phi_h)_{\Omega_X} = 0. \tag{20}$$

Finally, we retrieve the global discrete solution

$$u_h^n(x,\ell) = \sum_{j=1}^{N\_XP}\sum_{l=1}^{N\_LP} u_{j,l}^n \phi_j(x)\psi_l(\ell), \tag{21}$$

by setting $u_{j,l}^n = \tilde{u}_{j,l}^n$, and use it as the initial solution for (19) in the next time step.

In the above Eq. (19), the discrete nodal points x_j can also be the quadrature points, which are necessary to evaluate integrals over Ω_X. It gives another variant of the discrete form of the operator-split Eqs. (17) and (18), see Section 4.

3.3. Analysis of the operator-splitting finite element method

In order to show the role of the additional regularity condition in the operator-splitting finite element method, we derive an error estimate of the operator-split backward Euler discrete equation. Furthermore, we show that the approximation order of the operator-splitting finite element method is same as the classical finite element method provided the additional regularity condition is satisfied. To illustrate these, we first derive the associated one-step discrete form of the operator-split Eqs. (19) and (20) in the following.

Lemma 3. *The associated one-step backward Euler discrete form of the operator-split discrete Eqs.* (19) *and* (20) *is*

$$\left(\frac{u_h^n - u_h^{n-1}}{\delta t}, v_h\right) + a_{OS}(u_h^n, v_h) = (f^n, v_h), \tag{22}$$

where the operator-split bilinear form is given by

$$a_{OS}(u_h^n, v_h) = \int_\Omega \nabla u_h^n \cdot \nabla v_h + \delta t \int_\Omega \nabla_x \nabla_l u_h^n : \nabla_x \nabla_l v_h. \tag{23}$$

Proof. Taking $\theta = 1$ in (19) and (20), we get the algebraic form of the Eq. (19) as

$$(M_\ell + \delta t A_\ell)\widehat{U}^n = \delta t F_\ell^n + M_\ell U^{n-1} \tag{24}$$

and the algebraic form of the Eq. (20) as

$$(M_x + \delta t A_x)\widetilde{U}^n = M_x(\widehat{U}^n)^{\mathrm{T}}. \tag{25}$$

Here,

$$\widehat{U}^n = (U^n)^{\mathrm{T}}, \quad \widetilde{U}^n = U^n, \quad F_\ell^n := \int_{\Omega_L} f^n \psi_h,$$

and we recall the definition of the solution matrix $U^n = [u_{j,l}^n]_{\mathcal{M}\times\mathcal{N}}$. Now, multiply (24) by $M_x \otimes \mathbb{I}$, and (25) by $\mathbb{I} \otimes (M_\ell + \delta t A_\ell)$, we get

$$\{(M_x \otimes M_\ell) + \delta t(M_x \otimes A_\ell)\}\overline{U}^n = \delta t(M_x \otimes F_\ell^n) + (M_x \otimes M_\ell)\overline{U}^{n-1}$$

and

$$\Big\{(M_x \otimes M_\ell) + \delta t\{(A_x \otimes M_\ell) + (M_x \otimes A_\ell)\} + (\delta t)^2(A_x \otimes A_\ell)\Big\}\overline{U}^n = \{(M_x \otimes M_\ell) + \delta t(M_x \otimes A_\ell)\}\overline{U}^n,$$

respectively. Equating the above equations, we get

$$\Big\{(M_x \otimes M_\ell) + \delta t\{(A_x \otimes M_\ell) + (M_x \otimes A_\ell)\} + (\delta t)^2(A_x \otimes A_\ell)\Big\}\overline{U}^n = \delta t(M_x \otimes F_\ell^n) + (M_x \otimes M_\ell)\overline{U}^{n-1}. \tag{26}$$

Using (9) and (21), the above Eq. (26) can be written as

$$\Big\{M + \delta t A + (\delta t)^2(A_x \otimes A_\ell)\Big\}\overline{U}^n = \delta t(M_x \otimes F_\ell^n) + M\overline{U}^{n-1} \tag{27}$$

For the cross matrix term in (27), we have

$$\begin{aligned}(A_x \otimes A_\ell)\overline{U}^n &:= \sum_{j=1}^{\mathcal{M}}\sum_{l=1}^{\mathcal{N}}\sum_{i=1}^{\mathcal{M}}\sum_{k=1}^{\mathcal{N}} \int_\Omega u_{j,l}^n \nabla_x\phi_j \cdot \nabla_x\phi_i : \nabla_\ell\psi_l \cdot \nabla_\ell\psi_k = \sum_{j=1}^{\mathcal{M}}\sum_{l=1}^{\mathcal{N}}\sum_{i=1}^{\mathcal{M}}\sum_{k=1}^{\mathcal{N}} \int_\Omega \nabla_x(u_{j,l}^n\phi_j\nabla_\ell\psi_l) : \nabla_x(\phi_i\nabla_\ell\psi_k) \\ &= \sum_{j=1}^{\mathcal{M}}\sum_{l=1}^{\mathcal{N}}\sum_{i=1}^{\mathcal{M}}\sum_{k=1}^{\mathcal{N}} \int_\Omega \nabla_x(\nabla_\ell(u_{j,l}^n\phi_j\psi_l)) : \nabla_x(\nabla_\ell(\phi_i\psi_k)) = \int_\Omega \nabla_x\nabla_\ell u_h^n : \nabla_x\nabla_\ell v_h.\end{aligned} \tag{28}$$

Next, to show $M_x \otimes F_\ell^n = F^n$ in (27), we write their discrete form to get

$$M_x \otimes F_\ell^n := \int_{\Omega_X}\int_{\Omega_L} \sum_{i=1}^{\mathcal{M}}\sum_{j=1}^{\mathcal{M}} \phi_i\phi_j \sum_{k=1}^{\mathcal{N}} f^n\psi_k. \tag{29}$$

Further, each equation in the algebraic system (27) is obtained by applying summation to the ansatz indices j and l. Thus, applying summations to the indices j and l in (29), the right hand side vector $rhs_{i,k}$, for $i = 1,\ldots,\mathcal{M}$, and $k = 1,\ldots,\mathcal{N}$ becomes

$$rhs_{i,k} = \int_{\Omega_X\times\Omega_L} f^n\phi_i\psi_k\sum_{j=1}^{\mathcal{M}}\phi_j = \int_\Omega f^n\phi_i\psi_k = F_{i,k}^n \tag{30}$$

see (9). Thus, we have $M_x \otimes F_\ell^n = F^n$. Substituting the bilinear form of the cross term(28) and the right hand side vector (30) in (27) we get the desired result. □

Remark 5. Note the additional mixed derivative term in the discrete Eq. (22) due to the application of operator-splitting. This error term scales with the time step δt.

Next, we derive an a priori error estimate for the operator-splitting backward Euler scheme (22). Let us first introduce the approximation properties of the finite element spaces V_h and W_h, (cf. Theorem 4.8.12 and Corollary 4.8.15 in [2]).

(**A1.**) Approximation property of V_h: We assume that there is an interpolation operator $I_X \in \mathcal{L}(H_0^1(\Omega_X); V_h)$ such that for all $1 \leqslant s \leqslant r+1$

$$\begin{aligned}&\|I_X u\|_{H^s(\Omega_X)} \leqslant C\|u\|_{H^s(\Omega_X)}, u \in H_0^1(\Omega_X)\cap H^s(\Omega_X),\\ &\|u - I_X u\|_{L^2(\Omega_X)} + h\|u - I_X u\|_{H^1(\Omega_X)} \leqslant Ch^s|u|_{H^s(\Omega_X)}, u \in H_0^1(\Omega_X)\cap H^s(\Omega_X).\end{aligned}$$

(**A2.**) Approximation property of W_h: We assume that there is an interpolation operator $I_L \in \mathcal{L}(H_0^1(\Omega_L); W_h)$ such that for all $1 \leqslant s \leqslant r+1$

$$\begin{aligned}&\|I_L u\|_{H^s(\Omega_L)} \leqslant C\|u\|_{H^s(\Omega_L)}, \quad u \in H_0^1(\Omega_L)\cap H^s(\Omega_L),\\ &\|u - I_L u\|_{L^2(\Omega_L)} + h\|u - I_L u\|_{H^1(\Omega_L)} \leqslant Ch^s|u|_{H^s(\Omega_L)} u \in H_0^1(\Omega_L)\cap H^s(\Omega_L).\end{aligned}$$

Here, $\mathcal{L}(X;Y)$ denotes the set of linear and continuous mappings from X into Y. Now, we define the interpolation operator $I_h \in \mathcal{L}(H_0^{1,1}(\Omega) \cap H^{r+1,r+1}(\Omega), V_h \otimes W_h)$ by

$$I_h : I_X I_L = I_L I_X.$$

For the error estimate we decompose the error into two parts, the first part measures the projection error whereas the second part measures the error of the solution to the interpolation. Define

$$e_h^n := u(t^n) - u_h^n = (u(t^n) - I_h u(t^n)) + (I_h u(t^n) - u_h^n) =: \eta^n + \xi^n.$$

Theorem 4. *Let the solution u of (2) be smooth enough and the approximation properties (A1) and (A2) be satisfied. Then, the error estimate*

$$\|e_h^n\|^2 + \sum_{k=1}^{n} \delta t \|\nabla e_h^k\|^2 \leqslant C_u\left(h^{2r} + \delta t^2\right), \quad n = 1, \ldots, N$$

holds true where C_u is a constant depending on certain norms of the solution specified within the proof of the Theorem.

Proof. For estimating the error $\xi^n \in V_h \otimes W_h$, we take $\xi^n = u(t^n) - u_h^n - \eta^n$, use (22) with $v_h = \xi^n$, and (3) for $t = t^n$ and $v = \xi^n$ to obtain

$$\begin{aligned}
&\left(\frac{\xi^n - \xi^{n-1}}{\delta t}, \xi^n\right) + a_{OS}(\xi^n, \xi^n) = \left(\frac{u(t^n) - u(t^{n-1})}{\delta t}, \xi^n\right) + a_{OS}(u(t^n), \xi^n) - (f^n, \xi^n) - \left(\frac{\eta^n - \eta^{n-1}}{\delta t}, \xi^n\right) \\
&- a_{OS}(\eta^n, \xi^n = \left(\frac{u(t^n) - u(t^{n-1})}{\delta t} - \frac{\partial u(t^n)}{\partial t}, \xi^n\right) + \delta t(\nabla_x \nabla_\ell u(t^n), \nabla_x \nabla_\ell \xi^n) - \left(\frac{\eta^n - \eta^{n-1}}{\delta t}, \xi^n\right) \\
&- (\nabla \eta^n, \nabla \xi^n) - \delta t(\nabla_x \nabla_\ell \eta^n, \nabla_x \nabla_\ell \xi^n).
\end{aligned} \tag{31}$$

From $\xi^n|_{\partial\Omega_x} = 0$ it follows that $\nabla_\ell \xi^n|_{\partial\Omega_x} = 0$, thus with $\xi^n|_{\partial\Omega_\ell} = 0$ we have by integration by parts

$$(\nabla_x \nabla_\ell u(t^n), \nabla_x \nabla_\ell \xi^n) = (\Delta_x \Delta_\ell u(t^n), \xi^n).$$

Multiplying (31) with $2\delta t$ and using the identity $2a(a-b) = a^2 - b^2 + (a-b)^2$ for the first term on the left hand side, we get

$$\begin{aligned}
\|\xi^n\|^2 + \|\xi^n - \xi^{n-1}\|^2 + 2\delta t\|\nabla \xi^n\|^2 + 2\delta t^2 \|\nabla_x \nabla_\ell \xi^n\|^2 \leqslant \|\xi^{n-1}\|^2 + 2\delta t|(S^n, \xi^n)| + 2\delta t|(\nabla \eta^n, \nabla \xi^n)| \\
+ 2\delta t^2 |(\nabla_x \nabla_\ell \eta^n, \nabla_x \nabla_\ell \xi^n)|,
\end{aligned}$$

where

$$S^n := \frac{u(t^n) - u(t^{n-1})}{\delta t} - \frac{\partial u(t^n)}{\partial t} - \frac{\eta^n - \eta^{n-1}}{\delta t} + \delta t \Delta_x \Delta_\ell u(t^n).$$

Using Cauchy–Schwarz's and Young's inequality, we get

$$\begin{aligned}
&2\delta t|(S^n, \xi^n)| \leqslant \delta t\|S^n\|^2 + \delta t\|\xi^n\|^2, \\
&2\delta t|(\nabla \eta^n, \nabla \xi^n)| \leqslant \delta t\|\nabla \eta^n\|^2 + \delta t\|\nabla \xi^n\|^2, \\
&2\delta t^2 |(\nabla_x \nabla_\ell \eta^n, \nabla_x \nabla_\ell \xi^n)| \leqslant \frac{\delta t^2}{2} \|\nabla_x \nabla_\ell \eta^n\|^2 + 2\delta t^2 \|\nabla_x \nabla_\ell \xi^n\|^2.
\end{aligned}$$

Applying Taylor's theorem with integral remainder for the S^n term, we have

$$\delta t\|S^n\|^2 \leqslant C\left(\delta t^2 \int_{t^{n-1}}^{t^n} \left\|\frac{\partial^2 u(s)}{\partial s^2}\right\|^2 ds + \int_{t^{n-1}}^{t^n} \left\|\frac{\partial \eta(s)}{\partial s}\right\|^2 ds + \delta t^3 \|\Delta_x \Delta_\ell u(t^n)\|^2\right).$$

Collecting all estimates above we end up with

$$\begin{aligned}
(1-\delta t)\|\xi^n\|^2 + \delta t\|\nabla \xi^n\|^2 \leqslant \|\xi^{n-1}\|^2 \\
+ C\left(\delta t^2 \int_{t^{n-1}}^{t^n} \left\|\frac{\partial^2 u(s)}{\partial s^2}\right\|^2 ds + \delta t^3 \|\Delta_x \Delta_\ell u(t^n)\|^2 + \int_{t^{n-1}}^{t^n} \left\|\frac{\partial \eta(s)}{\partial s}\right\|^2 ds + \delta t\|\nabla \eta^n\|^2 + \delta t^2 \|\nabla_x \nabla_\ell \eta^n\|^2\right).
\end{aligned}$$

Divide the inequality by $(1 - \delta t)$, use $1 \leqslant 1/(1 - \delta t) \leqslant 1 + 2\delta t \leqslant 2$ for $\delta t \leqslant 1/2$, and sum over $n = 1, \ldots, N$ to get

$$\|\xi^N\|^2+\sum_{n=1}^{N}\delta t\|\nabla\xi^n\|^2\leqslant\|\xi^0\|^2+2\sum_{n=1}^{N}\delta t\|\xi^{n-1}\|^2+C\delta t^2\left(\int_0^T\left\|\frac{\partial^2u(s)}{\partial s^2}\right\|^2ds+\sum_{n=1}^{N}\delta t\|\Delta_x\Delta_\ell u(t^n)\|^2\right)$$
$$+C\left(\int_0^T\left\|\frac{\partial\eta(s)}{\partial s}\right\|^2ds+\sum_{n=1}^{N}\delta t\|\nabla\eta^n\|^2+\delta t\sum_{n=1}^{N}\delta t\|\nabla_x\nabla_\ell\eta^n\|^2\right).$$

Finally, we apply a discrete form of Gronwall's Lemma resulting in

$$\|\xi^N\|^2+\sum_{n=1}^{N}\delta t\|\nabla\xi^n\|^2\leqslant C\exp(2T)$$
$$\left(\|\xi^0\|^2+\delta t^2\left\|\frac{\partial^2u}{\partial s^2}\right\|^2_{L^2(0,T;L^2(\Omega))}+\delta t^2\|\Delta_x\Delta_\ell u\|^2_{l^2(0,T;L^2(\Omega))}+\int_0^T\left\|\frac{\partial\eta(s)}{\partial s}\right\|^2ds+\sum_{n=1}^{N}\delta t\|\nabla\eta^n\|^2+\delta t\sum_{n=1}^{N}\delta t\|\nabla_x\nabla_\ell\eta^n\|^2\right).\tag{32}$$

It remains to estimate the interpolation error for which we have the $L^2(\Omega)$-bound:

$$\|\eta\|=\|u-I_XI_Lu\|\leqslant\|u-I_Xu\|+\|I_Xu-I_LI_Xu\|\leqslant\left(\int_{\Omega_L}\|u-I_Xu\|^2_{L^2(\Omega_X)}\right)^{1/2}+\left(\int_{\Omega_X}\|I_Xu-I_LI_Xu\|^2_{L^2(\Omega_L)}\right)^{1/2}$$
$$\leqslant Ch^r\left(\int_{\Omega_L}|u|^2_{H^r(\Omega_X)}\right)^{1/2}+Ch^r\left(\int_{\Omega_X}\|I_Xu\|^2_{H^r(\Omega_L)}\right)^{1/2}\leqslant Ch^r\left(\|u\|_{L^2(\Omega_L;H^r(\Omega_X))}+\|u\|_{L^2(\Omega_X;H^r(\Omega_L))}\right).$$

Similarly we get the following bounds for the derivatives

$$\left\|\frac{\partial\eta}{\partial s}\right\|\leqslant Ch^r\left(\left\|\frac{\partial u}{\partial t}\right\|_{L^2(\Omega_L;H^r(\Omega_X))}+\left\|\frac{\partial u}{\partial t}\right\|_{L^2(\Omega_X;H^r(\Omega_L))}\right),$$
$$\|\nabla_x\eta\|\leqslant\|\nabla_xu-\nabla_xI_Xu\|+\|\nabla_xI_Xu-I_L\nabla_xI_Xu\|\leqslant Ch^r\left(\|u\|_{L^2(\Omega_L;H^{r+1}(\Omega_X))}+\|u\|_{H^1(\Omega_X;H^r(\Omega_L))}\right),$$
$$\|\nabla_\ell\eta\|\leqslant\|\nabla_\ell u-\nabla_\ell I_Lu\|+\|\nabla_\ell I_Lu-I_X\nabla_\ell I_Lu\|\leqslant Ch^r\left(\|u\|_{L^2(\Omega_X;H^{r+1}(\Omega_L))}+\|u\|_{H^1(\Omega_L;H^r(\Omega_X))}\right),$$
$$\|\nabla_x\nabla_\ell\eta\|\leqslant\|\nabla_x\nabla_\ell u-\nabla_xI_X\nabla_\ell u\|+\|\nabla_\ell\nabla_xI_Xu-\nabla_\ell I_L\nabla_xI_Xu\|\leqslant Ch^r\left(\|u\|_{H^1(\Omega_L;H^{r+1}(\Omega_X))}+\|u\|_{H^1(\Omega_X;H^{r+1}(\Omega_L))}\right).$$

Taking into consideration that $\|\nabla\eta\|^2=\|\nabla_x\eta\|^2+\|\nabla_\ell\eta\|^2$, substituting the estimates above, and using $\|\xi^0\|\leqslant\|\eta^0\|$, we get the statement of the theorem. □

Remark 6. From the estimate (32), it is clear that we need a slightly higher regularity condition, that is, the mixed derivative $\Delta_x\Delta_\ell u(t^n)$ has to be bounded. Thus, it requires that the source f should be more regular than in $L^2((0,T]\times\Omega)$, and this restricts the class of problems for which the operator-splitting can be used. However, we can derive profit by using operator-splitting for high-dimensional problems when we have higher regularity for f, see the numerical results in Section 5.

4. Implementations of the operator-splitting method

In this section, we discuss the following two variants of implementations of the operator-splitting method in the context of the finite element method.

1. Quadrature point based operator-splitting method.
2. Nodal point based operator-splitting method.

4.1. Quadrature point based operator-splitting method

The basic idea in the quadrature point based operator-splitting method is to solve (19) at every quadrature point of Ω_X, and then solve (20) at every nodal point of W_h.

Let $x_j\in\Omega_X,\ j=1,\dots,N\_XP$ be the Cartesian coordinates of all quadrature points which are necessary to evaluate all integrals in (20). For simplicity, let us assume that all quadrature points are inside the cells, see Fig. 1. Next, let $\ell_l\in\Omega_L,\ l=1,\dots,N\_LP$ be the Cartesian coordinates of all nodal points of L-direction finite element space W_h. To explain the algorithm, let us assume that all nodal functionals of W_h are defined by point values. Further, to store the solutions in L-and X-directions we define two-dimensional arrays $U_Q[N\_XP][N\_LP]$ and its transpose $U_Q^{\mathrm{Tr}}[N\_LP][N\_XP]$, respectively.

After gathering all these information at the beginning, we first solve N_XP number of L-direction Eq. (19), that is, we solve (19) for each quadrature point $x_j,\ j=1,\dots,N\_XP$. Then, we transpose the L-direction solution array U_Q to get U_Q^{Tr}, which is needed in the solution of X-direction Eq. (20) in Step 2.

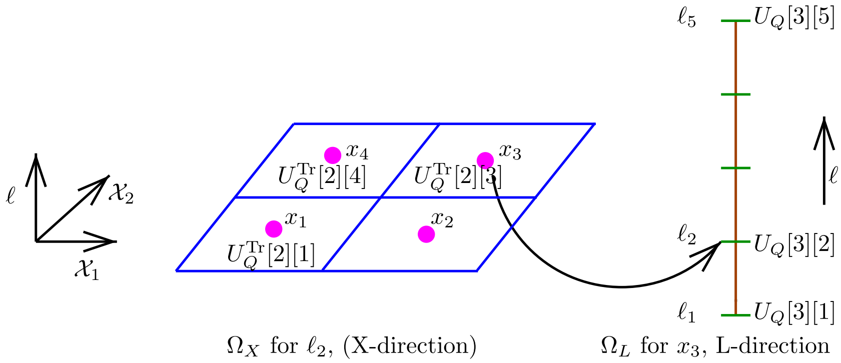
ℓ
$\mathcal{X}_2$
$\mathcal{X}_1$
x_4
$U_Q^{\mathrm{Tr}}[2][4]$
$U_Q^{\mathrm{Tr}}[2][3]$
x_3
x_1
$U_Q^{\mathrm{Tr}}[2][1]$
x_2
ℓ_5
$U_Q[3][5]$
ℓ
ℓ_2
$U_Q[3][2]$
ℓ_1
$U_Q[3][1]$
Ω_X for ℓ_2, (X-direction)
Ω_L for x_3, L-direction

Fig. 1. Representation of quadrature points $x_j \in \Omega_X$ for the nodal point $\ell_2 \in \Omega_L$ (left), and the nodal points $\ell_l \in \Omega_L$ of W_h for the quadrature point $x_3 \in \Omega_X$ (right).

In Step 2, we solve N_LP number of X-direction Eq. (20), that is, we solve (20) for each nodal point ℓ_l, $l = 1, \ldots, N\_LP$ of W_h. In this step, the key point is the assembling of the right-hand side of the X-direction equations from the L-direction solution U_Q^{Tr}. For example at $t = t^n$ in the backward Euler scheme, *i.e.*, for $\theta = 1$ in (20), we have to assemble $M_x \widetilde{U}^{n-1}$ on the right hand side. However, we do not have $\widetilde{U}^{n-1}$ explicitly, instead we have only $U_Q^{n,\mathrm{Tr}}$, which is the transpose of U_Q^n. Therefore, we cannot assemble $M_x \widetilde{U}^{n-1}$ by the classical matrix multiplication. Thus, we provide the following algorithm. We have

$$M_x[\tilde{u}_{j,l}^{n-1}]_{\mathcal{M}\times 1} := \int_{\Omega_X} \tilde{u}_h^{n-1}(x,\ell_l)\phi_h = \sum_{m=1}^{N\_Cells} \int_{K_m} \tilde{u}_h^{n-1}(x,\ell_l)\phi_h, \quad l = 1,\ldots,\mathcal{N}.$$

Then, for each l, the ith component $(i = 1,\ldots,\mathcal{M})$ of the right-hand side vector $M_x[\tilde{u}_{j,l}^{n-1}]$ can be evaluated as

$$rhs[i] = \sum_{m=1}^{N\_Cells} \int_{K_m} \sum_{j=1}^{\mathcal{M}} \tilde{u}_{j,l}^{n-1}\phi_j\phi_i = \sum_{m=1}^{N\_Cells} \sum_{k=(m-1)N\_KQP}^{m*N\_KQP} \omega_k \underbrace{\sum\nolimits_{j=1}^{\mathcal{M}} \tilde{u}_{j,l}^{n-1}\phi_j(x_k)}_{U_Q^{n,\mathrm{Tr}}[l][k]} \phi_i(x_k) = \sum_{m=1}^{N\_Cells} \sum_{k=(m-1)N\_KQP}^{m*N\_KQP} \omega_k U_Q^{n,\mathrm{Tr}}[l][k]\phi_i(x_k). \tag{33}$$

Here, N_KQP is the number of Gaussian quadrature points in a cell. In addition, the $A_x \widetilde{U}^{n-1}$ matrix has to be assembled on the right hand side when the Crank–Nicolson scheme is used in (19) and (20). However, it is not straightforward to assemble this term as like $M_x \widetilde{U}^{n-1}$. Actually from Step 1, at each quadrature point x_j, we only have

$$\sum_{j=1}^{\mathcal{M}} \tilde{u}_{j,l}^{n-1}\phi_j(x_k) = U_Q^{n,\mathrm{Tr}}[l][k]$$

from the L-direction. However, to assemble $A_x \widetilde{U}^{n-1}$, we need

$$\sum_{j=1}^{\mathcal{M}} \tilde{u}_{j,l}^{n-1}\nabla_x\phi_j(x_k).$$

To overcome this challenge, for each l, we calculate the nodal functionals $\tilde{u}_{j,l}^{n-1}$ from the L-direction solution $U_Q^{n,\mathrm{Tr}}[l][k]$ using the $L_2 -$ *projection*. That is, in each cell $K_m \in \mathcal{T}_h$, $m = 1,\ldots,N\_Cells$, we solve

$$\int_{K_m} \tilde{u}_{j,l}^{n-1}\phi_j\phi_i = \sum_{k=(m-1)N\_KQP}^{m*N\_KQP} \omega_k \underbrace{\sum\nolimits_{j=1}^{\mathcal{M}} \tilde{u}_{j,l}^{n-1}\phi_j(x_k)}_{U_Q^{n,\mathrm{Tr}}[l][k]} \phi_i(x_k)$$

to obtain $\tilde{u}_{j,l}^{n-1}$. Since the above $L_2 -$ *projection* is local, the degrees of freedoms of V_h on the vertices, edges and faces will get different values from all cells associated with them. Thus, in our implementation, we take the average of all these values arising from all associated cells. Finally, we assemble $A_x \widetilde{U}^{n-1}$ by the classical matrix multiplication.

Overall, the calculation of the right hand side in the X-direction Eq. (20) (Step 2) from the L-direction finite element solution is not straightforward in the quadrature point based operator-splitting method. Further, a special technique such as $L^2 -$ *projection* as proposed above is needed when the right hand side of (20) contains the derivative of the L-direction finite element solution. An advantage is that the L-direction Eq. (20) are independent each other, and thus (20) can be solved in parallel without communication when the quadrature points x_j of Ω_X are inside the cells.

4.2. Nodal point based operator-splitting method

In the nodal point based operator-splitting method, we solve the L-direction Eq. (19) at each nodal point of the X-direction finite element space V_h. Thus, immediately after the solution of Step 1, we have the nodal values, and hence we assemble the right-hand side of (20) by the classical matrix multiplication.

As in Section 3.2, let $x_j \in \Omega_X,\ j = 1, \ldots, N\_XP$, and $\ell_l \in \Omega_L,\ l = 1, \ldots, N\_LP$ be the Cartesian coordinates which are necessary to evaluate nodal functionals of the finite element spaces V_h and W_h, respectively. To explain the algorithm, let us consider Q_1 and P_1 finite element spaces in X- and L-directions, respectively. Further, assume that the nodal points are the vertices of the computational mesh, see Fig. 2. Next, we define two-dimensional arrays, $U_N[N\_XP][N\_LP]$ and its transpose $U_N^{\rm Tr}[N\_LP][N\_XP]$ to store the solutions of L- and X-directions, respectively.

Now, we first solve N_XP number of L-direction Eq. (19), that is, we solve (19) for each nodal point $x_j,\ j = 1, \ldots, N\_XP$, of V_h. Then, we transpose the solution U_N to get $U_N^{\rm Tr}$, which is needed in the right-hand side of X-direction Eq. (20) in Step 2.

In Step 2, we solve N_LP number of X-direction Eq. (20), that is, we solve (20) for each nodal point $\ell_l,\ l = 1, \ldots, N\_LP$ of L-direction finite element space W_h. Since the nodal functionals of V_h are defined by point values, at $t = t^n$ we get $\widetilde{U}^{n-1} = U_N^{n,\rm Tr}$ directly. Even if we define the nodal functionals of V_h using some quadrature formulas, we can easily evaluate $\widetilde{U}^n$ from $U_N^{n,\rm Tr}$. Also, note that in the nodal point based operator-splitting algorithm we do not need any special method or L^2- *projection* to assemble the right-hand side of X-direction equation from the L-direction update.

5. Numerical experiments

For the validation we consider a simple problem with smooth solution and source term (see, Remark 6), and compare the numerical results with the analytical solution. Let $\Omega := \Omega_X \times \Omega_L$, with $\Omega_X = (0,1)^2$ and $\Omega_L = (0,1)$. Consider the problem (2) with $f = (3\pi^2 - 0.1)e^{-0.1t}\sin(\pi\mathcal{X}_1)\cos(\pi\mathcal{X}_2)\cos(\pi\ell)$. The initial and non-homogeneous boundary values are chosen such that the solution of (2) is $u = e^{-0.1t}\sin(\pi\mathcal{X}_1)\cos(\pi\mathcal{X}_2)\cos(\pi\ell)$. First, we performed an array of 3D computations for different levels of meshes and temporal discretizations. Further, we perform the convergence study for these algorithms. The L^2 error is computed by applying a quadrature rule on each cell of the decomposition of $\Omega \subset \mathbb{R}^3$. Further, to calculate the error in space and time we use

$$\ell^\infty(0,T;L^2(\Omega)) := \sup_{n=1,\ldots,N} \|u(t^n) - u_h(t^n)\|_{L^2(\Omega)},$$

$$\ell^2(0,T;L^2(\Omega)) := \left(\sum_{n=1}^{N} \delta t \|u(t^n) - u_h(t^n)\|^2_{L^2(\Omega)}\right)^{1/2}.$$

5.1. Standard 3D Computations with finite element method

To perform the standard (without dimensional splitting) 3D computations using finite element method, first we triangulate the 3D domain Ω using hexahedra. The computational meshes for successive levels are obtained by successively refining the initial coarse mesh uniformly. This results into 35937 degrees of freedom on level 5. Here, we compare the numerically computed solutions for different temporal discretizations with the analytical solution. Computations are performed up to level 5. To perform the convergence study, we consider the following cases

A1 : Q_1 in space and backward Euler for time with $\delta t \propto h^2$,
A2 : Q_1 in space and Crank–Nicolson for time with $\delta t \propto h$,

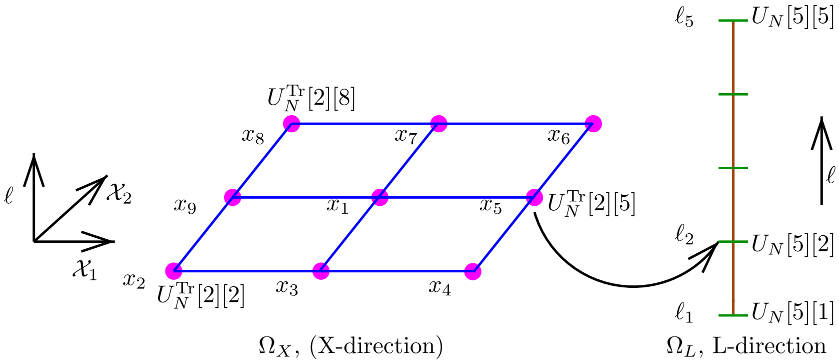

Fig. 2. Representation of Q_1 space nodal points x_i in Ω_X (left) at the second nodal point level ℓ_2 of L-direction finite element space W_h (right).

Table 1
Errors in $\ell^\infty(L^2)$ and $\ell^2(L^2)$ norms for the case A1 using standard 3D computations.

Level	h	δt	$\ell^\infty(L^2)\times 10^{-4}$	Order	$\ell^2(L^2)\times 10^{-4}$	Order
1	0.866025	0.75	1577.52		1728.91	
2	0.4330127	0.1875	477.545	1.72395	589.388	1.55257
3	0.2165064	0.046875	124.894	1.93493	155.772	1.91978
4	0.1082532	0.0117187	32.6134	1.93717	39.4336	1.98194
5	0.0541265	0.00292969	8.50921	1.93837	9.890197	1.99535

Table 2
Errors in $\ell^\infty(L^2)$ and $\ell^2(L^2)$ norms for the case A2 using standard 3D computations.

Level	h	δt	$\ell^\infty(L^2)\times 10^{-4}$	Order	$\ell^2(L^2)\times 10^{-4}$	Order
1	0.866025	0.866025	1559.33		1805.57	
2	0.4330127	0.4330127	477.363	1.70777	555.513	1.70056
3	0.2165064	0.2165064	128.247	1.89616	152.758	1.86257
4	0.1082532	0.1082532	32.1604	1.99557	38.6183	1.98389
5	0.0541265	0.0541265	7.87806	2.02937	9.7161	1.99084

Table 3
Interpolation error in L^2 norm for the 2D + 1D operator-splitting method.

Level	$h\times 10^{-2}$	$L^2\times 10^{-5}$	order
3	35.3553	3206.654	
4	17.6776	673.5361	2.2512
5	8.8388	139.0634	2.2760
6	4.4194	29.9046	2.2173
7	2.2097	6.7593	2.1454

The obtained numerical errors in both cases are presented in Table 1 and Table 2. The optimal order of convergence (two in L^2 norm) for the trilinear hexahedral elements (Q_1) finite element is obtained in both cases.

5.2. Computations with the operator-splitting method (2D + 1D)

To perform 2D + 1D computations with the operator-splitting method, we triangulate Ω_X and Ω_L using the quadrilaterals and subintervals, respectively. The successive mesh levels of the X- and L-direction equations are obtained by successively refining their respective initial coarse mesh uniformly. In the initial coarse mesh (level 2) of Ω_X and Ω_L, we have 16 quadrilaterals and 4 subintervals, respectively. Further, the Q_1 and P_1 finite elements are used to spatial discretization in all computations. The L^2 error is computed by applying quadrature rules in X- and L-direction, i.e.,

$$\|u\|^2_{L^2(\Omega)} = \int_\Omega u^2 = \int_{\Omega_X}\left(\int_{\Omega_L} u^2(x,\ell)d\ell\right)dx \approx \sum_{K_i^X}\sum_{m=1}^{N\_KXQP} w_m^x \int_{\Omega_L} u^2(x_m,\ell)d\ell \approx \sum_{K_i^X}\sum_{m=1}^{N\_KXQP} w_m^x \sum_{K_j^L}\sum_{l=1}^{N\_KLQP} w_l^\ell u^2(x_m,\ell_l),$$

where K_i^X and K_j^L are cells in Ω_X and Ω_L, respectively. Here, N_KXQP, w_m^x and N_KLQP, w_l^ℓ are the number of quadrature points, quadrature weights in each K_i^X and K_j^L, respectively.

5.2.1. Interpolation error

For verifying the implementation of the operator-splitting routines, we interpolate the initial solution $u_0 = \sin(\pi\mathcal{X}_1)\cos(\pi\mathcal{X}_2)\cos(\pi\ell)$ in the L-direction and compute the interpolation error in the L^2-norm. Table 3 shows optimal convergence of second order.

5.2.2. Quadrature point based operator-splitting method

To validate the quadrature point based implementation of operator-splitting method presented in the Section 4.1, we performed an array of computations for the considered test example with the backward Euler and Crank–Nicolson time discretizations up to the mesh level 6. In the fine mesh (level 6), we have 36864 number of quadrature points in X-direction finite element space V_h and 65 number of nodal points in the L-direction space W_h. Thus, we solved 65 times a 2D equation and 36864 times a 1D equation in the level 6 for the 3D problem. To perform the numerical study, the following cases are considered

Table 4
Errors in $\ell^\infty(L^2)$ and $\ell^2(L^2)$ norms for the case B1 using quadrature point based operator-split computations.

Level	$h \times 10^{-2}$	$\delta t \times 10^{-4}$	$\ell^\infty(L^2) \times 10^{-4}$	Order	$\ell^2(L^2) \times 10^{-4}$	Order
2	35.3553	1250	348.5833		452.4036	
3	17.6776	312.5	89.1769	1.9667	115.3489	1.9716
4	8.8388	78.125	22.3227	1.9981	28.82934	2.0003
5	4.4194	19.5312	5.5839	1.9991	7.208633	1.9997
6	2.2097	4.8828	1.3966	1.9993	1.653322	2.1243

Table 5
Errors in $\ell^\infty(L^2)$ and $\ell^2(L^2)$ norms for the case B2 using quadrature point based operator-split computations.

Level	$h \times 10^{-2}$	$\delta t \times 10^{-4}$	$\ell^\infty(L^2) \times 10^{-4}$	Order	$\ell^2(L^2) \times 10^{-4}$	Order
2	35.3553	35.3553	314.4611		433.9884	
3	17.6776	17.6776	75.7352	2.0538	104.4342	2.0550
4	8.8388	8.8388	18.11561	2.0637	23.9513	2.1244
5	4.4194	4.4194	4.15197	2.1253	5.3987	2.1494
6	2.2097	2.2097	0.94167	2.1404	1.1455	2.2365

Table 6
Errors in $\ell^\infty(L^2)$ and $\ell^2(L^2)$ norms for the case C1 using nodal point based operator-split computations.

Level	$h \times 10^{-2}$	$\delta t \times 10^{-4}$	$\ell^\infty(L^2) \times 10^{-4}$	Order	$\ell^2(L^2) \times 10^{-4}$	Order
2	35.3553	1250	350.8710		456.4965	
3	17.6776	312.5	92.3602	1.9256	118.1260	1.9502
4	8.8388	78.125	23.8848	1.9511	29.95218	1.9795
5	4.4194	19.5312	6.0921	1.9710	7.537856	1.9904
6	2.2097	4.8828	1.5344	1.9892	1.888839	1.9966

Table 7
Errors in $\ell^\infty(L^2)$ and $\ell^2(L^2)$ norms for the case C2 using nodal point based operator-split computations.

Level	$h \times 10^{-2}$	$\delta t \times 10^{-4}$	$\ell^\infty(L^2) \times 10^{-4}$	Order	$\ell^2(L^2) \times 10^{-4}$	Order
2	35.3553	35.3553	324.3398		418.7031	
3	17.6776	17.6776	78.09492	2.0542	95.3343	2.13486
4	8.8388	8.8388	18.01345	2.1161	22.3156	2.09494
5	4.4194	4.4194	4.05971	2.1496	5.2474	2.08837
6	2.2097	2.2097	0.90164	2.1707	1.1464	2.19447

Table 8
Computational costs in seconds for one time step on different mesh levels.

Level	Full 3D	Quadrature based OS	Nodal based OS
3	0.04	0.03	0.007
4	0.36	0.2	0.05
5	7.7	1.4	0.4
6	–	11.6	3.4

B1 : $Q_1 \otimes P_1$ in space, and backward Euler for time with $\delta t \propto h^2$,
B2 : $Q_1 \otimes P_1$ in space, and Crank–Nicolson for time with $\delta t \propto h$.

The computed errors are presented in Tables 4 and 5 for the cases B1 and B2, respectively. The numerical error obtained with the Crank–Nicolson scheme is marginally less than the numerical error obtained with the backward Euler scheme. However, in both backward Euler and Crank–Nicolson schemes, we obtained the optimal order of convergence for the Q_1 and P_1 finite elements in the X- and L-direction, respectively, in both $\ell^\infty(L^2)$ and $\ell^2(L^2)$ norms. Interestingly, the numerical error obtained from 2D + 1D computation with the quadrature point based implementation of operator-splitting method is slightly better than the numerical error obtained from the 3D computation with the standard finite element method.

5.2.3. Nodal point based operator-splitting method

To validate the nodal point based implementation of the operator-splitting method presented in the Section 4.2, a set of computations is performed for the considered test example with the backward Euler and Crank–Nicolson time discretizations up to the mesh level 6. In the fine mesh (level 6), we have 4225 nodal points in X-direction finite element space V_h and 65 nodal points in the L-direction space W_h. Thus, we solved 65 times a 2D equation and 3969 times a 1D equation (excluding 256 Dirichlet boundary nodal points) in the level 6 of the 3D problem. To perform the numerical study, the following cases are consider

C1 : $Q_1 \otimes P_1$ in space and backward Euler for time with $\delta t \propto h^2$,
C2 : $Q_1 \otimes P_1$ in space and Crank–Nicolson for time with $\delta t \propto h$.

The numerical errors obtained in both cases are presented in the Table 6 and Table 7. As in the quadrature point based operator-splitting method, the numerical error obtained with the Crank–Nicolson scheme is marginally less than the numerical error obtained with the backward Euler scheme. Moreover, the optimal order of convergence is obtained in both cases. Even though, we solved less number of 1D equations in the nodal point based implementation compared to the quadrature point based implementation, we obtained a similar numerical errors in the nodal point based implementation. Thus, from the computational point of view the nodal point based implementation of the operator-splitting method is more efficient than the quadrature point based implementation. Further, transferring the solution from the L-direction to the X-direction is much simpler in the nodal point based implementation.

The following Table 8 gives some impression on the numerical costs for the solution of the considered examples when using a direct solver. We see that the nodal based operator-splitting approach is very efficient, in particular, on finer meshes. The expected gain by splitting a 4D problem into two 2D problems should be even higher.

6. Conclusion

An operator-splitting finite element method for solving high-dimensional parabolic problems is presented in this paper. For the backward Euler scheme, the equivalence up to a perturbation term of order $(\delta t)^2$ between the variational forms of the high-dimensional equation and the operator-split equations is shown. An a priori error estimate for the operator-splitting finite element method applied to a parabolic equation is presented. It is shown that the mixed partial derivatives of the solution has to be bounded in order to apply the operator-splitting method. Further, two variants of operator-splitting algorithms (i) quadrature point based operator-splitting algorithm, and (ii) nodal point based operator-splitting algorithm, in the context of the finite element method are presented in detail.

The proposed operator-splitting algorithms are validated using a 3D test problem, which is split into 2D and 1D subproblems. It is demonstrated that the numerical solutions obtained with these two variants of algorithms for the 2D + 1D subproblems are in a good agreement with the analytical solution. Further, the optimal order of convergence is obtained in both variants. Even though, the obtained numerical errors and the order of convergence are similar in both variants, the nodal point based operator-splitting algorithm is more efficient due to less number of nodal points in comparison with the number of quadrature points.

References

[1] N. Ahmed, G. Matthies, L. Tobiska, H. Xie, Finite element methods of an operator splitting applied to population balance equations, J. Comput. Appl. Math. 236 (2011) 1604–1621.
[2] S.C. Brenner, L.R. Scott, The Mathematical Theory of Finite Element Methods, Third ed., Springer, 2008.
[3] H.-J. Bungartz, M. Griebel, Sparse grids, Acta Numer. 13 (2004) 1–123.
[4] P. Chen, J. Sanyal, M. Dudukovic, CFD modeling of bubble columns flows: implementation of population balance, Chem. Eng. Sci. 59 (22–23) (2004) 5201–5207.
[5] J. Douglas, T. Dupont, Alternating-direction Galerkin methods on rectangles, in: B. Hubbard (Ed.), Numerical Solution of Partial Differential Equations-II, Academic Press, New York, 1971, pp. 133–214.
[6] J. Douglas Jr., On the numerical integration of $\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 = \partial u/\partial t$ by implicit methods, J. Soc. Indust. Appl. Math. 3 (1955) 42–65.
[7] S. Ganesan, An operator-splitting Galerkin/SUPG finite element method for population balance equations: stability and convergence, ESAIM: M2AN 46 (2012) 1447–1465.
[8] R. Glowinski, E.J. Dean, G. Guidoboni, D.H. Peaceman, H.H. Rachford, Applications of operator-splitting methods to the direct numerical simulation of particulate and free-surface flows and to the numerical solution of the two-dimensional elliptic Monge–Ampère equation, Japan J. Indust. Appl. Math. 25 (2008) 1–63.
[9] M. Griebel, D. Oeltz, A sparse grid space-time discretization scheme for parabolic problems, Computing 81 (2007) 1–34.
[10] M. Griebel, D. Oeltz, P. Vassilevski, Space-time approximation with sparse grids, SIAM J. Sci. Comput. 28 (2005) 701–727.
[11] R. Gunawan, I. Fusman, R.D. Braatz, High resolution algorithms for multidimensional population balance equations, AIChE Journal 50 (2004) 2738–2749.
[12] H. Hulburt, S. Katz, Some problems in particle technology: a statistical mechanical formulation, Chem. Eng. Sci. 19 (8) (1964) 555–574.
[13] V. John, M. Roland, T. Mitkova, K. Sundmacher, L.T.A. Voigt, Simulations of population balance systems with one internal coordinate using finite element methods, Chem. Eng. Sci. 64 (2009) 733–741.

[14] D.J. Knezevic, E. Süli, A heterogeneous alternating-direction method for a micro-macro dilute polymeric fluid model, ESAIM: M2AN 43 (2009) 1117–1156.

[15] G. Lian, S. Moore, L. Heeney, Population balance and computational fluid dynamics modelling of ice crystallisation in a scraped surface freezer, Chem. Eng. Sci. 61 (23) (2006) 7819–7826.

[16] D.L. Ma, D.K. Tafti, R.D. Braatz, High-resolution simulation of multidimensional crystal growth, Ind. Eng. Chem. Res. 41 (25) (2002) 6217–6223.

[17] A. Majumder, V. Kariwala, S. Ansumali, A. Rajendran, Fast high-resolution method for solving multidimensional population balances in crystallization, Ind. Eng. Chem. Res. 49 (8) (2010) 3862–3872.

[18] M.A. Pinto, C.D. Immanuel, F.J. Doyle III, A two-level discretisation algorithm for the efficient solution of higher-dimensional population balance models, Chem. Eng. Sci. 63 (5) (2008) 1304–1314.

[19] S. Qamar, S. Noor, M. Rehman, A. Seidel-Morgenstern, Numerical solution of a multi-dimensional batch crystallization model with fines dissolution, Comput. Chem. Eng. 35 (3) (2011) 412–422.

[20] D. Ramkrishna, Population Balances, Theory and Applications to Particulate Systems in Engineering, Academic Press, San Diego, 2000.

[21] V. Thomee, Galerkin Finite Element Methods for Parabolic Problems, third ed., Springer, 2008.

[22] X.Y. Woo, R.B.H. Tan, R.D. Braatz, Modeling and computational fluid dynamics-population balance equation-micromixing simulation of impinging jet crystallizers, Cryst. Growth Des. 9 (1) (2009) 156–164.