# BLENDED FINITE ELEMENT METHOD AND ITS CONVERGENCE FOR THREE-DIMENSIONAL IMAGE RECONSTRUCTION USING $L^2$-GRADIENT FLOW*

GUOLIANG XU† AND CHONG CHEN‡

**Abstract.** The gradient-flow-based explicit and semi-implicit finite element methods proposed in our earlier papers have been applied to solve various variational models for image reconstruction in cryo-electron microscopy and X-ray computed tomography, respectively. In this paper, we develop a gradient-flow-based blended finite element method for solving the variational model of three-dimensional image reconstruction. The method can be regarded as a linear combination of the explicit and semi-implicit schemes, which combines the advantages of both. The computational cost of each iteration of the method is less than that of the semi-implicit situation. In addition, the convergence rate of the method is faster because the temporal step-size can be larger than that of the explicit scheme. Furthermore, the convergence analysis for the blended finite element method is presented. The numerical results also show that the method is more efficient than the explicit and semi-implicit methods.

**Key words.** Three-dimensional image reconstruction, blended finite element method, convergence analysis, variational model, cryo-electron tomography, X-ray computed tomography.

**AMS subject classifications.** 65K05, 65R32, 92C55.

## 1. Introduction

The reconstruction problem in computed tomography, or some other application fields such as cryo-electron microscopy single-particle analysis or cryo-electron tomography, is to produce a two-dimensional (2D) or three-dimensional (3D) image from a large number of line-integral projections from different views (see [7, 14, 15, 20, 22]). Recently, the iterative reconstruction algorithms have been much investigated because they often yield better image quality than the classical methods for sparse-view and (or) high-noise detected data (see [28, 24, 5, 21, 32]). The iterative tomographic reconstruction algorithms have been mainly used to solve two different types of models in biomedical imaging. The first one includes the various optimization models based on the total variation (TV) norm, Tikhonov norm, or other modified ones, which start from the discrete formulation directly (see [17, 24, 30, 22]). The second one, however, includes the variational models based on the TV regularization term, Tikhonov regularization term, etc. that begin with modeling from the view of continuous functional and then variating, finally followed by discretization (see [5, 21, 32]). It is worth noting that the TV of the reconstructed image (see [26]) has been considered frequently in the optimization-based or variational models as a result of the sparsity on the gradient of the piecewise-constant image because of the introduction of compressed sensing theory (see [2, 10]). In this article, our main focus is to analyze the advantages and

†LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, P.O. Box 2719, Beijing 100190, P.R. China (xuguo@lsec.cc.ac.cn).
‡Corresponding author. LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, P.O. Box 2719, Beijing 100190, P.R. China (chench@lsec.cc.ac.cn).

the disadvantages of our proposed gradient-flow-based finite element methods and to further develop a new one.

In [21], we presented an iterative algorithm for reconstructing a 3D density function from a set of 2D electron microscopy images. By minimizing an energy functional consisting of a fidelity term and a regularization term, an $L^2$-gradient flow was derived. The flow was integrated by an explicit finite element method. The method is compared with weighted back-projection [25], the algebraic reconstruction technique [18], the simultaneous iterative reconstructive technique [16], etc. The numerical results show that the $L^2$-gradient flow method achieves a better resolution than the other classical methods. Furthermore, the theoretical analysis of the convergence of the iterative method has been considered in [6]. In addition, we also developed the gradient-flow-based semi-implicit finite element method and its convergence analysis for image reconstruction in [5]. The experimental results demonstrate that this method has more desirable performance compared with other reconstruction methods in solving a number of challenging reconstruction problems.

In this paper, we present a gradient-flow-based blended finite element method, which includes the explicit and semi-implicit finite element methods as its special cases, for solving the same variational model. An approximately optimal temporal step-size is determined which makes the blended scheme more efficient than the explicit one. The amount of calculation in each iteration of the method is less than that of the semi-implicit situation. Moreover, the convergence analysis for the blended finite element method is also presented. The numerical results show that the method is more efficient than the explicit and semi-implicit methods.

The rest of this paper is organized as follows. Section 2 sketches an overview of the mathematical background knowledge on image reconstruction. In Section 3, we first introduce a new computational method, then describe the detailed derivation of our algorithm, and finally give the details of numerical computing. The theoretical analysis of our numerical method is given in Section 4. In Section 5, several numerical results are presented. Finally, Section 6 concludes the paper.

## 2. Mathematical preliminaries

The material of this section serves as the mathematical basis for the rest of the paper. For a more detailed derivation, we refer the interested readers to [19, 22, 12]. Let $f$ be a function defined on $\mathbb{R}^3$, where $\mathbb{R}^3$ is the 3D real space consisting of 3-tuples of real numbers, usually denoted by single letters, $\mathbf{x} = (x_1, x_2, x_3)^T$, $\mathbf{y} = (y_1, y_2, y_3)^T$, etc. The inner product and norm in $\mathbb{R}^3$ are defined by $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{y} = \sum_1^3 x_i y_i$ and $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$, respectively. In addition, the gradient of $f(\mathbf{x})$ is denoted as $\nabla f = (f_{x_1}, f_{x_2}, f_{x_3})^T$. The unit sphere in $\mathbb{R}^3$ is denoted by $S^2$. Let $X$ be a real Banach space with norm $\|\cdot\|_X$. Then the space $L^p((0, T_0); X)$ is defined as the collection of all measurable functions $u : [0, T_0] \to X$ with

$$\|u\|_{L^p((0,T_0);X)} := \left( \int_0^{T_0} \|u(t)\|_X^p \, \mathrm{d}t \right)^{\frac{1}{p}} < \infty,$$

for $1 \le p < \infty$, denoted as $\|u\|_{L^p(X)}$ for short, and

$$\|u\|_{L^\infty((0,T_0);X)} := \operatorname*{ess\,sup}_{0 \le t \le T_0} \|u(t)\|_X < \infty,$$

abbreviated as $\|u\|_{L^\infty(X)}$.

For a given $\theta \in S^2$, the X-ray transform of a function $f \in L^2(\mathbb{R}^3)$ in the direction $\theta$ is defined as

$$\mathbf{P}_\theta f(\mathbf{y}) = \int_{\mathbb{R}^1} f(s\theta + \mathbf{y}) \mathrm{d}s, \quad \mathbf{y} \in \theta^\perp, \tag{2.1}$$

where $\theta^\perp$ is the hyperplane passing through the origin and orthogonal to $\theta$. From (2.1), we can see that the X-ray transform is the integral of $f \in L^2(\mathbb{R}^3)$ over the straight line through the point $\mathbf{y} \in \theta^\perp$ along the direction $\theta \in S^2$.

## 3. Reconstruction algorithm

In this paper, we concentrate on 3D image reconstruction from the parallel projections at different views inasmuch as the reconstruction method we propose can be straightforwardly generalized to other projection geometries and higher dimensions.

**3.1. Reconstruction model.** Let $f(\mathbf{x}) : \Omega \subset \mathbb{R}^3 \to \mathbb{R}$ represent an unknown density function of a biomedical object, which has a bounded support in a cube $\Omega$, namely,

$$\mathrm{supp}(f) \subseteq \Omega. \tag{3.1}$$

Let $BV(\Omega)$ denote the bounded variation space. For the definition and properties of $BV(\Omega)$, we refer the interested readers to [11, 1]. We want to find a function $f \in BV(\Omega)$ such that the following energy functional is minimized:

$$E(f) = E_1(f) + \lambda E_2(f), \tag{3.2}$$

where $E_1(f)$ is represented as the fidelity term, $E_2(f)$ stands for the regularized term obtained from a certain maximum a posterior estimation or *priori* information, and $\lambda \geq 0$ is a parameter which balances the effects of the fidelity term and the regularized one. In this work, $E_1(f)$ and $E_2(f)$ are given as follows:

$$E_1(f) = \frac{1}{2} \sum_{i=1}^p \int_{\mathbb{R}^2} \left( \mathbf{P}_{\theta_i} f(\mathbf{y}) - g_i(\mathbf{y}) \right)^2 \mathrm{d}\mathbf{y}, \tag{3.3}$$

$$E_2(f) = \int_{\mathbb{R}^3} \phi(\|\nabla f\|) \mathrm{d}\mathbf{x}, \tag{3.4}$$

where $\theta_i \in S^2$ is the given $i$th projection direction, $g_i(\mathbf{y}) = g(\theta_i, \mathbf{y})$ is the corresponding $i$th measured data. The way on how to choose the potential function $\phi$ can be found in [1, 3]. $\phi$ is the engine to remove interfered artifacts as well as to preserve geometric features.

From a theoretical point of view, the model (3.2) is well-posed. We present this result in the following theorem.

THEOREM 3.1. *If $\phi$ is a convex, nondecreasing function from $\mathbb{R}^+$ to $\mathbb{R}^+$ with $\lim_{s \to +\infty} \phi(s) = +\infty$, there exists two constants $c > 0$ and $b \geq 0$ such that $cs - b \leq \phi(s) \leq cs + b$, $\forall s \geq 0$, and $E_1(f)$ and $E_2(f)$ are defined as (3.3) and (3.4) respectively, then the minimization problem*

$$\min_{f \in BV(\Omega)} \left[ E_1(f) + \lambda E_2(f) \right] \tag{3.5}$$

*admits a unique solution.*

The proof of existence and uniqueness of a solution for the minimization problem is similar to that of [31]. Hence, we do not present the proof because there is no essential difference.

To derive the reconstruction equations, we first need to variate the regularized model. Using formulas (3.2), (3.3), and (3.4) and then variating $E(f)$, we have

$$\delta(E(f),h) = \sum_{i=1}^{p} \int_{\mathbb{R}^2} ((\mathbf{P}_{\theta_i} f)(\mathbf{y}) - g_i(\mathbf{y}))(\mathbf{P}_{\theta_i} h)(\mathbf{y}) \mathrm{d}\mathbf{y}$$
$$+ \lambda \int_{\mathbb{R}^3} \frac{\phi'(\|\nabla f\|)\nabla f^T \nabla h}{\|\nabla f\|} \mathrm{d}\mathbf{x}, \tag{3.6}$$

where $\forall h \in C^1(\mathbb{R}^3)$. Using the relation (see [22])

$$\int_{\theta_i^\perp} \mathbf{P}_{\theta_i} f(\mathbf{y}) g(\mathbf{y}) \mathrm{d}\mathbf{y} = \int_{\mathbb{R}^3} f(\mathbf{x}) \mathbf{P}_{\theta_i}^* g(\mathbf{x}) \mathrm{d}\mathbf{x},$$

and the Green formula, we obtain the following Euler-Lagrange equation:

$$\sum_{i=1}^{p} \mathbf{P}_{\theta_i}^* (\mathbf{P}_{\theta_i} f - g_i) - \lambda \operatorname{div} \left[ \frac{\phi'(\|\nabla f\|)\nabla f}{\|\nabla f\|} \right] = 0, \tag{3.7}$$

where $\mathbf{P}_{\theta_i}^* g(\mathbf{x}) = g_i(E_{\theta_i}\mathbf{x})$ with $E_{\theta_i}$ the orthogonal projection on $\theta_i^\perp$.

**3.2. Numerical computing.** Equation (3.7) is a typical integro-differential equation. The equation is highly nonlinear so that the Fourier analysis method is useless here. The frequently used method resorts to a gradient flow, i.e., converting the elliptic differential equation to a time-dependent parabolic one in the domain $[0, T_0] \times \Omega$ with a given $T_0 \gg 0$. Hence, we obtain the following gradient flow, i.e., a parabolic differential equation

$$\begin{cases} \frac{\partial f}{\partial t} = \lambda \operatorname{div} \left[ \frac{\phi'(\|\nabla f\|)\nabla f}{\|\nabla f\|} \right] - \sum_{i=1}^{p} \mathbf{P}_{\theta_i}^* (\mathbf{P}_{\theta_i} f - g_i), & \text{in } \Omega_{T_0}, \\ f = 0, & \text{on } \partial\Omega_{T_0}, \end{cases} \tag{3.8}$$

with a given initial condition $f_0 = f(\mathbf{x}, 0)$, where $\Omega_{T_0} := (0, T_0] \times \Omega$ and $\partial\Omega_{T_0} := (0, T_0] \times \partial\Omega$. When the above gradient flow achieves its steady state solution, we obtain the solution of the Euler-Lagrange equation (3.7). Therefore, in what follows our problem is

$$\int_{\Omega} \left[ \frac{\partial f}{\partial t} h + \frac{\lambda \phi'(\|\nabla f\|)(\nabla f)^T \nabla h}{\|\nabla f\|} \right] \mathrm{d}\mathbf{x} + \sum_{i=1}^{p} \int_{\mathbb{R}^2} (\mathbf{P}_{\theta_i} f - g_i) \mathbf{P}_{\theta_i} h \mathrm{d}\mathbf{y} = 0, \tag{3.9}$$

with a given initial value $f_0 = f(\mathbf{x}, 0)$. To preserve the geometric features of the reconstructed image, we need to choose the regularization function $\phi(s)$. In this paper, we choose

$$\phi(s) = \sqrt{s^2 + \epsilon^2}, \tag{3.10}$$

and use a blended finite element method to solve (3.9), where $\epsilon$ is a fixed small positive constant (we take it as $10^{-5}$). From a geometric point of view, the above regularizer is able to remove the artifacts while preserving well the geometric features (see [9, 23, 4]),

which can be seen as a regularization potential of modified TV (see [26]). Let $\mathcal{T}_h$ be a voxel mesh of $\Omega$ with mesh size $h \in (0,1)$. Let

$$V^h = \text{span}\{\phi_0, \phi_1, \dots, \phi_N\}$$

be the finite element space, where $\phi_i \in C^1(\mathbb{R}^3)$ are the basis functions with compact support $\Omega_i$ in $\Omega$. We assume that $\partial\Omega_i$ is regular (Lipschitz continuous). In this paper, we use a cubic B-spline function in the tensor product form defined on the uniform mesh $\mathcal{T}_h$, that is, $\phi_i$ is defined as

$$\phi_{\zeta(n+1)^2+\beta(n+1)+\gamma}(\mathbf{x}) = N_\zeta(x)N_\beta(y)N_\gamma(z),$$

with $\zeta = 0,1,\dots,n$, $\beta = 0,1,\dots,n$, $\gamma = 0,1,\dots,n$, where $N_\zeta$ is the one-dimensional cubic B-spline basis function defined on the knots $[-2+\zeta, -1+\zeta, \zeta, 1+\zeta, 2+\zeta]$. It is easy to see that

$$N_\zeta(x) = N_0(x-\zeta), \tag{3.11}$$

so $\Omega_i$ is a hypercube. Let

$$\Omega = \cup_{i=0}^N \Omega_i, \quad N = (n+1)^3 - 1.$$

Then

$$f(\mathbf{x}) = \sum_{i=0}^N f_i \phi_i(\mathbf{x})$$

has compact support $\Omega$. Let $\{t_m\}_{m=0}^{m_0}$ be a partition of $[0, T_0]$ with mesh sizes

$$\tau_m = t_m - t_{m-1} \in (0,1), \quad \tau = \max_{1 \le m \le m_0} \tau_m.$$

In the following, we use the notation

$$d_t f^m := \frac{f^m - f^{m-1}}{\tau_m}$$

for the operator $d_t$. Then the blended finite element discretization for the gradient flow (3.9) is given as follows: Find $F^m \in V^h$, for $m = 1,2,\dots,m_0$, such that

$$\int_{\mathbb{R}^3} \left[ d_t F^m v_h + \alpha \frac{\lambda \phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|} (\nabla F^m)^T \nabla v_h \right] d\mathbf{x} + \alpha \sum_{i=1}^p \int_{\mathbb{R}^2} (\mathbf{P}_{\theta_i} F^m - g_i) \mathbf{P}_{\theta_i} v_h d\mathbf{y}$$

$$= (\alpha - 1) \left[ \int_{\mathbb{R}^3} \frac{\lambda \phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|} (\nabla F^{m-1})^T \nabla v_h d\mathbf{x} + \sum_{i=1}^p \int_{\mathbb{R}^2} (\mathbf{P}_{\theta_i} F^{m-1} - g_i) \mathbf{P}_{\theta_i} v_h d\mathbf{y} \right], \tag{3.12}$$

for $\forall v_h \in V^h$, with some initial value $F^0 \in V^h$ that approximates $f_0$, where $\alpha \in [0,1]$.

Representing $F^m(\mathbf{x})$ as $\sum_{k=0}^N f_k^{(m)} \phi_k(\mathbf{x})$ and taking the test function $v_h(\mathbf{x}) = \phi_j(\mathbf{x})$, we can rewrite (3.12) as the following linear system:

$$\sum_{k=0}^N [m_{jk} + \alpha \tau_m (q_{jk} + r_{jk})] f_k^{(m)}$$

$$= \sum_{k=0}^{N} [m_{jk} - (1-\alpha)\tau_m(q_{jk}+r_{jk})]f_k^{(m-1)} + \tau_m b_j, \qquad (3.13)$$

for $j=0,\ldots,N$, where

$$m_{jk} = \int_{\mathbb{R}^3} \phi_j(\mathbf{x})\phi_k(\mathbf{x})\mathrm{d}\mathbf{x}, \qquad (3.14)$$

$$q_{jk} = \lambda \int_{\mathbb{R}^3} \left[ \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|}(\nabla\phi_k)^T\nabla\phi_j \right]\mathrm{d}\mathbf{x}, \qquad (3.15)$$

$$r_{jk} = \sum_{i=1}^{p} \int_{\mathbb{R}^2} (\mathbf{P}_{\theta_i}\phi_k)(\mathbf{P}_{\theta_i}\phi_j)\mathrm{d}\mathbf{y}, \qquad (3.16)$$

$$b_j = \sum_{i=1}^{p} \int_{\mathbb{R}^2} g_i(\mathbf{y})(\mathbf{P}_{\theta_i}\phi_j)(\mathbf{y})\mathrm{d}\mathbf{y}. \qquad (3.17)$$

Note that because the basis functions are locally supported, the coefficient matrices

$$M := \{m_{jk}\}_{j,k=0}^{N}, \quad Q := \{q_{jk}\}_{j,k=0}^{N}$$

of the system are sparse. However, the matrix $R := \{r_{jk}\}_{j,k=0}^{N}$ is not sparse. Also note that the matrices $M$, $R$, and $B = [b_0,\ldots,b_N]^T$ do not depend on $F^{m-1}$, but $Q$ does. Hence, $M$, $R$, and $B$ can be previously computed. In matrix form, equation (3.13) can be rewritten as

$$[M + \alpha\tau_m(Q+R)]X^{(m)} = [M - (1-\alpha)\tau_m(Q+R)]X^{(m-1)} + \tau_m B, \qquad (3.18)$$

where

$$X^{(m)} = [f_0^{(m)},\ldots,f_N^{(m)}]^T, \quad m=1,2,\ldots.$$

System (3.18) is usually large, and solving it directly is impractical. An iterative method, such as the GMRES method (see [27]), can be employed. However, the computational cost is huge because a great number of matrix-vector multiplications for each iteration are required.

Obviously, when $\alpha=0$ and $\alpha=1$, (3.18) becomes the explicit and semi-implicit schemes, respectively. Furthermore, equation (3.18) can be regarded as a convex combination of the following explicit and semi-implicit schemes:

$$MX^{(m)} = [M - \tau_m(Q+R)]X^{(m-1)} + \tau_m B, \qquad (3.19)$$
$$[M + \tau_m(Q+R)]X^{(m)} = MX^{(m-1)} + \tau_m B, \qquad (3.20)$$

with combination coefficients $1-\alpha$ and $\alpha$. Let

$$X^{(m)} = X^{(m-1)} + \tau_m \sum_{i=0}^{\infty} \tau_m^i Y_i, \qquad (3.21)$$

with

$$Y_i = [y_0^{(i)}, y_1^{(i)}, \ldots, y_N^{(i)}]^T.$$

Then substituting $X^{(m)}$ into (3.18), we have

$$[M + \alpha\tau_m(Q+R)]\sum_{i=0}^{\infty}\tau_m^i Y_i = B - (Q+R)X^{(m-1)}.$$

Comparing the coefficient of $\tau_m^i$, we obtain

$$Y_0 = M^{-1}[B - (Q+R)X^{(m-1)}], \qquad (3.22)$$
$$Y_i = -\alpha M^{-1}(Q+R)Y_{i-1}, \quad i = 1,2,\dots.$$

Notice that by choosing an appropriate value of $\alpha$, we can make $\|\alpha M^{-1}(Q+R)\| < 1$. Then the power series (3.21) is convergent. Let

$$y_i(\mathbf{x}) = \sum_{j=0}^{N} y_j^{(i)}\phi_j(\mathbf{x}), \quad i = 0,1,\dots.$$

Then

$$F^m(\mathbf{x}) = F^{m-1}(\mathbf{x}) + \tau_m\sum_{i=0}^{\infty}\tau_m^i y_i(\mathbf{x}).$$

Substituting $F^m(\mathbf{x})$ into (3.2) and putting

$$e(\tau_m) := E\left(F^{m-1}(\mathbf{x}) + \tau_m\sum_{i=0}^{\infty}\tau_m^i y_i(\mathbf{x})\right),$$

we can represent $e'(\tau_m)$ as a power series with respect to $\tau_m$,

$$e'(\tau_m) = \sum_{i=0}^{\infty}e_i\tau_m^i.$$

In our implementation, expansion (3.21) is truncated as

$$X^{(m)} = X^{(m-1)} + \tau_m Y_0 + \tau_m^2 Y_1. \qquad (3.23)$$

From

$$e'(\tau_m) = 0,$$

we obtain a cubic equation (ignoring the higher order term $O(\tau_m^4)$ yielded from the regularizer)

$$e_0 + e_1\tau_m + e_2\tau_m^2 + e_3\tau_m^3 = 0. \qquad (3.24)$$

Then we take $\tau_m$ as the minimal positive root of the equation. To satisfy the convergence conditions of the iterative method, the $\tau_m$ need to be adjusted to verify the restriction (4.2) (see Remark 4.2 for details). Having $\tau_m$, $Y_0$, and $Y_1$, $X^{(m)}$ can be approximated as (3.23). The coefficients $e_i$ in equation (3.24) are given as follows:

$$e_0 = \sum_{i=1}^{p}\int_{\mathbb{R}^2}(\mathbf{P}_{\theta_i}F^{m-1} - g_i)(\mathbf{P}_{\theta_i}y_0)\mathrm{d}\mathbf{x} + \lambda r'(0), \qquad (3.25)$$

$$e_1 = \sum_{i=1}^{p} \int_{\mathbb{R}^2} (\mathbf{P}_{\theta_i} y_0)^2 + 2(\mathbf{P}_{\theta_i} F^{m-1} - g_i)(\mathbf{P}_{\theta_i} y_1) \mathrm{d}\mathbf{x} + \lambda r''(0), \qquad (3.26)$$

$$e_2 = \sum_{i=1}^{p} \int_{\mathbb{R}^2} 3(\mathbf{P}_{\theta_i} y_0)(\mathbf{P}_{\theta_i} y_1) \mathrm{d}\mathbf{x} + \frac{\lambda}{2} r'''(0), \qquad (3.27)$$

$$e_3 = \sum_{i=1}^{p} \int_{\mathbb{R}^2} 2(\mathbf{P}_{\theta_i} y_1)^2 \mathrm{d}\mathbf{x} + \frac{\lambda}{6} r''''(0), \qquad (3.28)$$

where $r'$, $r''$, ... denote the derivatives of $r(\tau)$ with respect to $\tau$, and

$$r(\tau) = \int_{\mathbb{R}^3} \phi(g(\tau)) \mathrm{d}\mathbf{x}, \quad g(\tau) = \|\nabla F^{m-1} + \tau \nabla y_0 + \tau^2 \nabla y_1\|.$$

Using the chain rule for the derivatives of a composite function, we easily obtain

$$r' = \int_{\mathbb{R}^3} \phi' g' \mathrm{d}\mathbf{x},$$

$$r'' = \int_{\mathbb{R}^3} (\phi''(g')^2 + \phi' g'') \mathrm{d}\mathbf{x},$$

$$r''' = \int_{\mathbb{R}^3} (\phi'''(g')^3 + 3\phi'' g' g'' + \phi' g''') \mathrm{d}\mathbf{x},$$

$$r'''' = \int_{\mathbb{R}^3} (\phi''''(g')^4 + 6\phi'''(g')^2 g'' + 3\phi''(g'')^2 + 4\phi'' g' g''' + \phi' g'''') \mathrm{d}\mathbf{x}.$$

According to

$$g' g = (\nabla F^{m-1} + \tau \nabla y_0 + \tau^2 \nabla y_1)^T (\nabla y_0 + 2\tau \nabla y_1),$$

the higher order derivative of $g$ can be progressively computed using the following formulas:

$$g'' g + (g')^2 = (\nabla y_0 + 2\tau \nabla y_1)^T (\nabla y_0 + 2\tau \nabla y_1) + 2(\nabla F^{m-1} + \tau \nabla y_0 + \tau^2 \nabla y_1)^T \nabla y_1,$$
$$g''' g + 3g'' g' = 6(\nabla y_0 + 2\tau \nabla y_1)^T \nabla y_1,$$
$$g'''' g + 4g''' g' + 3(g'')^2 = 12(\nabla y_1)^T \nabla y_1.$$

LEMMA 3.2. *Assume $Y_0 \neq \mathbf{0}$, $e_0$, and $e_1$ are defined by (3.25) and (3.26), respectively. Then $e_0 < 0$ and $e_1 \geq 0$.*

*Proof.* Using the notations (3.15)–(3.17) and formula (3.22), $e_0$ can be translated into the equation

$$\begin{aligned} e_0 &= Y_0^T R X^{(m-1)} - Y_0^T B + Y_0^T Q X^{(m-1)} \\ &= Y_0^T [(Q+R) X^{(m-1)} - B] \\ &= -Y_0^T M Y_0. \end{aligned} \qquad (3.29)$$

Because $M$ is a positive definite Gram matrix, $e_0 = -Y_0^T M Y_0 < 0$. Combining notations (3.15)–(3.17), we obtain

$$e_1 = Y_0^T R Y_0 + 2(Y_1^T R X^{(m-1)} - Y_1^T B) + Y_0^T Q Y_0 + 2Y_1^T Q X^{(m-1)}$$
$$+ \lambda \int_{\mathbb{R}^3} \left[ \frac{\phi''(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|^2} - \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|^3} \right] [(\nabla F^{m-1})^T \nabla y_0]^2 \mathrm{d}\mathbf{x}. \quad (3.30)$$

Then by (3.22)–(3.23) and the symmetry of $Q$, $R$, and $M$, the first four terms on the right-hand side of (3.30) become

$$Y_0^T(Q+R)Y_0 + 2Y_1^T[(Q+R)X^{(m-1)} - B] = Y_0^T(Q+R)Y_0 - 2Y_1^T MY_0$$
$$= (1+2\alpha)Y_0^T(Q+R)Y_0. \qquad (3.31)$$

Because $\phi(s) = \sqrt{\epsilon^2 + s^2}$, we immediately obtain

$$\phi'(s) = \frac{s}{(\epsilon^2 + s^2)^{1/2}}, \quad \phi''(s) = \frac{\epsilon^2}{(\epsilon^2 + s^2)^{3/2}}. \qquad (3.32)$$

Then, in the last term on the right-hand side of (3.30), we have

$$\frac{\phi''(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|^2} - \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|^3} = -\frac{1}{(\epsilon^2 + \|\nabla F^{m-1}\|^2)^{3/2}}. \qquad (3.33)$$

By (3.15), (3.33), and the inequality $((\nabla F^{m-1})^T \nabla y_0)^2 \le \|\nabla F^{m-1}\|^2 \|\nabla y_0\|^2$, the last term on the right-hand side of (3.30) becomes

$$-\lambda \int_{\mathbb{R}^3} \frac{[(\nabla F^{m-1})^T \nabla y_0]^2}{(\epsilon^2 + \|\nabla F^{m-1}\|^2)^{3/2}} d\mathbf{x} \ge -\lambda \int_{\mathbb{R}^3} \frac{\|\nabla F^{m-1}\|^2 \|\nabla y_0\|^2}{(\epsilon^2 + \|\nabla F^{m-1}\|^2)^{3/2}} d\mathbf{x}$$
$$\ge -\lambda \int_{\mathbb{R}^3} \frac{\|\nabla y_0\|^2}{\sqrt{\epsilon^2 + \|\nabla F^{m-1}\|^2}} d\mathbf{x}$$
$$= -\lambda \int_{\mathbb{R}^3} \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|} \|\nabla y_0\|^2 d\mathbf{x}$$
$$= -Y_0^T QY_0. \qquad (3.34)$$

Therefore, using (3.30), (3.31), and (3.34), we easily obtain

$$e_1 \ge (1+2\alpha)Y_0^T RY_0 + 2\alpha Y_0^T QY_0, \quad \alpha \in [0,1].$$

As $Q$ is positive definite and $R$ is at least semi-positive definite, hence $e_1 \ge 0$. $\qquad \square$

REMARK 3.1.    Because $e_0 < 0$, $e(\tau)$ is a decreasing function around zero. If there exists a minimal positive root of equation (3.24), we then choose $\tau_m$ as this root. If equation (3.24) has no positive root, we can choose any positive number as $\tau_m$ (in our implementation, we choose 0.1 as $\tau_m$ in this case).

Now we summarize the iteration scheme as the following algorithm.

**Algorithm 3.1** *Blended Finite Element Method*

1. Set $m = 0$, set the initial B-spline coefficients $X^{(0)} = \mathbf{0}$.
2. Compute $M = \{m_{jk}\}$ and $B = \{b_j\}$ using (3.14) and (3.17), respectively.
3. Compute $Q = \{q_{jk}\}$ using (3.15).
4. Compute $Y_0$ and $Y_1$ using (3.22) and (3.23), respectively.
5. Compute $e_0$, $e_1$, $e_2$, and $e_3$ using (3.25)–(3.28) and then compute $\tau_m$ using (3.24).
6. Compute $X^{(m)}$ using (3.23).
7. Check the stopping condition. If it is satisfied, stop the iteration. Otherwise, set $m$ to be $m+1$ and return to step 3.

REMARK 3.2.    If we take $Y_1 = \mathbf{0}$ in (3.23), then the obtained $X^{(m)}$ is the same as the one obtained from the explicit finite element method for the given $\tau_m$. Hence our method can be regarded as a correction of the explicit finite element method by adding a second-order term $\tau_m^2 Y_1$. Even higher order terms can be computed from (3.21). But because $\tau_m$ is small in general, these terms are negligible.

**3.3. Computation of $RX$.**    In the above algorithm, we need to compute the multiplication of $R$ and $X = [x_0, \ldots, x_N]^T$. In general, matrix $R$ is not sparse. Hence if $N$ is large, for instance $N = 512^3$, the required space for storing the matrix $R$, which is $N^2 = 512^6$, may be beyond the capacity of the used computer. Hence, it is impractical to compute the multiplication of $R$ and $X$ directly. What we suggest is to represent $RX$ as follows:

$$RX = \sum_{i=1}^p \int_{\mathbb{R}^2} \sum_{k=0}^N x_k (\mathbf{P}_{\theta_i} \phi_k)(\mathbf{y})(\mathbf{P}_{\theta_i} \phi_j)(\mathbf{y}) \mathrm{d}\mathbf{y}.$$

Hence, we first compute

$$X_i(\mathbf{y}) := \sum_{k=0}^N x_k (\mathbf{P}_{\theta_i} \phi_k)(\mathbf{y}), \quad i = 1, \ldots, p, \tag{3.35}$$

and then compute

$$y_{ij} := \int_{\mathbb{R}^2} X_i(\mathbf{y})(\mathbf{P}_{\theta_i} \phi_j)(\mathbf{y}) \mathrm{d}\mathbf{y}, \quad i = 1, \ldots, p, \ \ j = 0, \ldots, N. \tag{3.36}$$

Finally, $RX$ is computed as

$$RX = \left[ \sum_{i=1}^p y_{i0}, \ldots, \sum_{i=1}^p y_{iN} \right]^T. \tag{3.37}$$

**3.4. Algorithm details and analysis of computational complexity.**    We present the algorithm details and analyze the computational complexity for Algorithm 3.1 from step 2 to step 6. The computational cost of the first and last steps are relatively small.

1. Matrix $M$ is sparse and its elements have closed-form representations. Utilizing the tensor product form of the basis functions, we only need to store and invert an $n \times n$ matrix. Hence, the computational cost of the inversion of $M$ is $O(n^3)$. Let $p$ be the number of projections. Then using the translation property (3.11) of the basis functions, we know that $\mathbf{P}_{\theta_i} \phi_j$ can be computed from $\mathbf{P}_{\theta_i} \phi_0$. Because $\mathbf{P}_{\theta_i} \phi_j$ has compact support, the computational cost of $b_j$ is $O(p)$. Hence, the computational complexity of $B$ is $O(pn^3)$. All these computations in step 2 are out of the $m$-iteration loop in Algorithm 3.1. They can be previously computed.

2. The cost of computing $Q$ in step 3 is of the same order as computing $M$: $O(n^3)$. Because $q_{jk}$ depends on $F^{m-1}$, it needs to be recomputed in each of the $m$ iterations.

3. To compute $Y_0$ and $Y_1$ in step 4, we need to first compute $QX^{(m-1)}$ and $RX^{(m-1)}$. Because $Q$ is a sparse matrix, the cost of computing $QX^{(m-1)}$ is $O(n^3)$. $RX^{(m-1)}$ is computed using (3.35)–(3.37). The cost of computing

each $X_i(\mathbf{y})$ using (3.35) is $O(n^3)$. For computing all $X_i(\mathbf{y})$, $i=1,\ldots,p$, the cost is $O(pn^3)$. Because $\mathbf{P}_{\theta_i}\phi_j$ is locally supported, the cost of computing $y_{ij}$ is $O(1)$. Hence the total cost of computing $\{y_{ij}\}$ is $O(pn^3)$. Finally, computing $RX$ using (3.37) requires $O(pn^3)$ arithmetic operations. Adding these together, the total cost of computing $RX$ is $O(pn^3)$. The cost of computing the multiplication $M^{-1}$ and a vector is $O(n^3)$ using the property that $M^{-1}$ is approximated by a band matrix. In summary, $Y_0$ can be computed with the complexity $O(pn^3)$. After $Y_0$ is computed, $Y_1$ is similarly computed via (3.23). Again, the cost is $O(pn^3)$.

4. Now we consider the computations of $e_0,\ldots,e_3$ in step 5. The cost of computing all the $\mathbf{P}_{\theta_i}y$ is $O(pn^3)$. The cost of computing $\nabla F^{m-1}$ and $\nabla y$ is $O(n^3)$. Hence the order of computing $e_0,\ldots,e_3$ using (3.25) and (3.28) is $O(pn^3)$.

5. The cost of computing $X^{(m)}$ in step 6 using (3.23) is $O(n^3)$. Hence, for each $m$-iteration, the total cost is on the order of $O(pn^3)$.

REMARK 3.3. The above analysis shows that the computational complexity of the blended scheme is $O(pn^3)$ for one iteration. This is of the same order as that of the explicit finite element method presented in [21]. However, because the blended scheme requires fewer iterations in general than the explicit scheme, the presented method is more efficient.
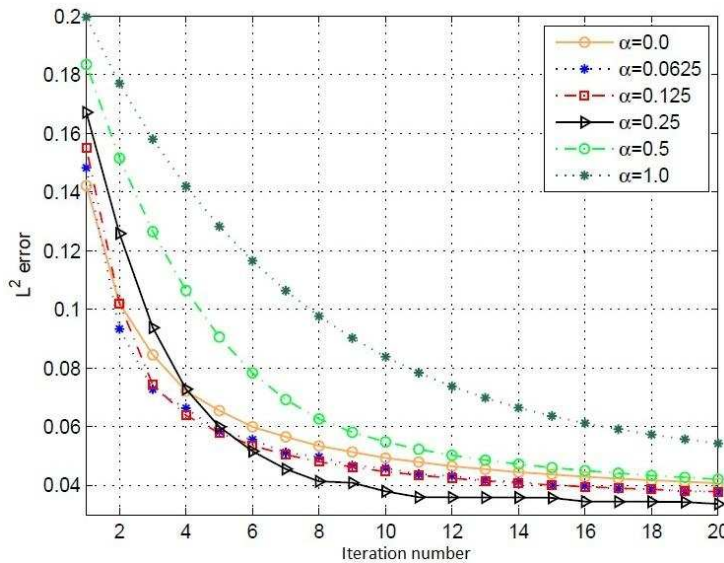


FIG. 3.1. *Curves of $L^2$-error for different values of $\alpha \in [0,1]$.*

**3.5. Selection of $\alpha$.** For a given $\tau_m$, a smaller $\alpha \geq 0$ makes the convergence of the series (3.21) faster. However, the smaller $\alpha \geq 0$ makes the iteration scheme (3.18) closer to the explicit scheme. On the other hand, when $\alpha$ approaches 1, the scheme (3.18) is implicit which could support the use of larger $\tau_m$. However, the convergence of the power series expansion becomes slow and the truncation of the power series may not be an accurate approximation of the inverse of the matrix $I+\alpha\tau_m M^{-1}(Q+R)$.

Therefore, using too large a value of $\alpha$ (close to one) or too small a value of $\alpha$ (close to zero) are both not ideal choices.

Based on an experiment for phantom data, we take $\alpha = 0.25$. The experiment is conducted as follows. First we construct sphere data, serving as the exact data. Then we compute projection images using some given directions from the volume data. Finally we reconstruct volume data using the reconstruction algorithm to be tested. Figure 3.1 shows the curves of the $L^2$-error between the exact volume data and the reconstructed volume data. For different values of $\alpha$ in (3.18), the $L^2$-errors are plotted with respect to the number of iterations. It is easy to see that $\alpha = 0.25$ yields the best result.

## 4. Convergence of blended finite element discretization

In this section, we give the convergence analysis of the finite element discretization for the blended scheme.

THEOREM 4.1.  *Given $\alpha \in [0,1]$, assume that $f_0 \in L^2(\mathbb{R}^3)$ has support $\Omega$, and $g_i \in L^2(\mathbb{R}^2)$. Then $\{F^m\}$ derived from the blended scheme*

$$X^{(m)} = X^{(m-1)} + \tau_m [M + \tau_m \alpha (R+Q)]^{-1} [B - (R+Q)X^{(m-1)}], \qquad (4.1)$$

*with the constraint*

$$0 < \tau_m \le \frac{\|X^{(m)} - X^{(m-1)}\|_M^2}{2(1-\alpha)[\|X^{(m)} - X^{(m-1)}\|_R^2 + \|X^{(m)} - X^{(m-1)}\|_Q^2]}, \qquad (4.2)$$

*satisfies*

$$\sum_{m=1}^{l} \left[ \frac{\tau_m}{2} \|d_t F^m\|_{L^2(\mathbb{R}^3)}^2 + \frac{\tau_m^2}{2} \sum_{i=1}^{p} \|d_t (\mathbf{P}_{\theta_i} F^m - g_i)\|_{L^2(\mathbb{R}^2)}^2 \right] + E(F^l) \le E(F^0). \quad (4.3)$$

*Proof.*     To verify (4.3), taking the testing function $v_h$ in (3.12) as $d_t F^m$, we obtain

$$\|d_t F^m\|_{L^2(\mathbb{R}^3)}^2 + \int_{\mathbb{R}^3} \lambda \left[ \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|} (\alpha \nabla F^m + (1-\alpha)\nabla F^{m-1})^T \nabla d_t F^m \right] d\mathbf{x}$$

$$+ \sum_{i=1}^{p} \int_{\mathbb{R}^2} \left[ (\alpha \mathbf{P}_{\theta_i} F^m + (1-\alpha)\mathbf{P}_{\theta_i} F^{m-1}) - g_i \right] d_t \mathbf{P}_{\theta_i} F^m d\mathbf{y} = 0. \quad (4.4)$$

Considering the last term on the left-hand side of equation (4.4), we have

$$\int_{\mathbb{R}^2} \left[ (\alpha \mathbf{P}_{\theta_i} F^m + (1-\alpha)\mathbf{P}_{\theta_i} F^{m-1}) - g_i \right] d_t \mathbf{P}_{\theta_i} F^m d\mathbf{y}$$

$$= \int_{\mathbb{R}^2} \left[ \alpha(\mathbf{P}_{\theta_i} F^m - \mathbf{P}_{\theta_i} F^{m-1}) + (\mathbf{P}_{\theta_i} F^{m-1} - g_i) \right] d_t \mathbf{P}_{\theta_i} F^m d\mathbf{y}$$

$$= \frac{d_t \|\mathbf{P}_{\theta_i} F^m - g_i\|_{L^2(\mathbb{R}^2)}^2}{2} + \frac{(2\alpha - 1)\tau_m \|d_t(\mathbf{P}_{\theta_i} F^m - g_i)\|_{L^2(\mathbb{R}^2)}^2}{2}, \qquad (4.5)$$

and similarly

$$[\alpha \nabla F^m + (1-\alpha)\nabla F^{m-1}]^T \nabla d_t F^m = \frac{d_t \|\nabla F^m\|^2}{2} + \frac{(2\alpha-1)\tau_m \|\nabla d_t F^m\|^2}{2}. \qquad (4.6)$$

Hence, combining equations (4.5) and (4.6), equation (4.4) becomes

$$
\|d_t F^m\|_{L^2(\mathbb{R}^3)}^2 + \frac{\sum_{i=1}^p d_t \|\mathbf{P}_{\theta_i} F^m - g_i\|_{L^2(\mathbb{R}^2)}^2}{2}
$$
$$
+ \frac{(2\alpha-1)\tau_m \sum_{i=1}^p \|d_t(\mathbf{P}_{\theta_i} F^m - g_i)\|_{L^2(\mathbb{R}^2)}^2}{2}
$$
$$
+ \frac{\lambda}{2} \int_{\mathbb{R}^3} \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|} (d_t\|\nabla F^m\|^2 + \tau_m\|\nabla d_t F^m\|^2)\mathrm{d}\mathbf{x}
$$
$$
+ \frac{\lambda}{2} \int_{\mathbb{R}^3} \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|} (2\alpha-2)\tau_m\|\nabla d_t F^m\|^2 \mathrm{d}\mathbf{x} = 0. \tag{4.7}
$$

Because

$$
d_t\|\nabla F^m\|^2 = \frac{2\|\nabla F^{m-1}\|(\|\nabla F^m\| - \|\nabla F^{m-1}\|) + (\|\nabla F^m\| - \|\nabla F^{m-1}\|)^2}{\tau_m},
$$

the fourth term on the left-hand side of (4.7) becomes

$$
\frac{1}{2} \int_{\mathbb{R}^3} \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|} d_t\|\nabla F^m\|^2 \mathrm{d}\mathbf{x}
$$
$$
= \frac{1}{\tau_m} \int_{\mathbb{R}^3} \phi'(\|\nabla F^{m-1}\|)(\|\nabla F^m\| - \|\nabla F^{m-1}\|)\mathrm{d}\mathbf{x}
$$
$$
+ \frac{1}{2\tau_m} \int_{\mathbb{R}^3} \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|} (\|\nabla F^m\| - \|\nabla F^{m-1}\|)^2 \mathrm{d}\mathbf{x}. \tag{4.8}
$$

Using the Cauchy inequality $(\nabla F^m)^T \nabla F^{m-1} \leq \|\nabla F^m\|\|\nabla F^{m-1}\|$, we have

$$
\|\nabla d_t F^m\|^2 \geq \frac{(\|\nabla F^m\| - \|\nabla F^{m-1}\|)^2}{\tau_m^2}.
$$

Then by $\phi'(s) \geq 0$, we obtain

$$
\frac{1}{2} \int_{\mathbb{R}^3} \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|} \tau_m\|\nabla d_t F^m\|^2 \mathrm{d}\mathbf{x}
$$
$$
\geq \frac{1}{2\tau_m} \int_{\mathbb{R}^3} \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|} (\|\nabla F^m\| - \|\nabla F^{m-1}\|)^2 \mathrm{d}\mathbf{x}. \tag{4.9}
$$

Substituting (4.8) and (4.9) into (4.7), we obtain

$$
\|d_t F^m\|_{L^2(\mathbb{R}^3)}^2 + \frac{\sum_{i=1}^p d_t \|\mathbf{P}_{\theta_i} F^m - g_i\|_{L^2(\mathbb{R}^2)}^2}{2}
$$
$$
+ \frac{(2\alpha-1)\tau_m \sum_{i=1}^p \|d_t(\mathbf{P}_{\theta_i} F^m - g_i)\|_{L^2(\mathbb{R}^2)}^2}{2}
$$
$$
+ \frac{\lambda}{\tau_m} \int_{\mathbb{R}^3} \phi'(\|\nabla F^{m-1}\|)(\|\nabla F^m\| - \|\nabla F^{m-1}\|)\mathrm{d}\mathbf{x}
$$
$$
+ \frac{\lambda}{\tau_m} \int_{\mathbb{R}^3} \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|} (\|\nabla F^m\| - \|\nabla F^{m-1}\|)^2 \mathrm{d}\mathbf{x}
$$
$$
+ \frac{\lambda}{2} \int_{\mathbb{R}^3} \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|} (2\alpha-2)\tau_m\|\nabla d_t F^m\|^2 \mathrm{d}\mathbf{x} \leq 0. \tag{4.10}
$$

We can show that (see [5] for details of this derivation)

$$\int_{\mathbb{R}^3} \phi'(\|\nabla F^{m-1}\|)(\|\nabla F^m\| - \|\nabla F^{m-1}\|)\mathrm{d}\mathbf{x}$$
$$+ \int_{\mathbb{R}^3} \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|}(\|\nabla F^m\| - \|\nabla F^{m-1}\|)^2 \mathrm{d}\mathbf{x}$$
$$\geq \int_{\mathbb{R}^3} \phi'(\|\nabla F^m\|)(\|\nabla F^m\| - \|\nabla F^{m-1}\|)\mathrm{d}\mathbf{x}. \tag{4.11}$$

Using the convexity of $\phi(s)$, the term on the right-hand side of (4.11) is bounded by

$$\int_{\mathbb{R}^3} \phi'(\|\nabla F^m\|)(\|\nabla F^m\| - \|\nabla F^{m-1}\|)\mathrm{d}\mathbf{x} \geq \tau_m d_t \int_{\mathbb{R}^3} \phi(\|\nabla F^m\|)\mathrm{d}\mathbf{x}. \tag{4.12}$$

Substituting (4.12) into (4.11), and then substituting the resulting inequality into (4.10) after multiplying it by $\tau_m$, we obtain

$$\tau_m \|d_t F^m\|_{L^2(\mathbb{R}^3)}^2 + \frac{\sum_{i=1}^p \tau_m d_t \|\mathbf{P}_{\theta_i} F^m - g_i\|_{L^2(\mathbb{R}^2)}^2}{2}$$
$$+ \frac{(2\alpha - 1)\tau_m^2 \sum_{i=1}^p \|d_t(\mathbf{P}_{\theta_i} F^m - g_i)\|_{L^2(\mathbb{R}^2)}^2}{2}$$
$$+ \lambda \tau_m d_t \int_{\mathbb{R}^3} \phi(\|\nabla F^m\|)\mathrm{d}\mathbf{x}$$
$$+ \frac{\lambda}{2} \int_{\mathbb{R}^3} \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|}(2\alpha - 2)\tau_m^2 \|\nabla d_t F^m\|^2 \mathrm{d}\mathbf{x} \leq 0. \tag{4.13}$$

The left-hand side of above inequality can be written as the sum of the following two parts:

$$P_1 = \frac{\tau_m}{2} \|d_t F^m\|_{L^2(\mathbb{R}^3)}^2 + \frac{\sum_{i=1}^p \tau_m d_t \|\mathbf{P}_{\theta_i} F^m - g_i\|_{L^2(\mathbb{R}^2)}^2}{2} + \lambda \tau_m d_t \int_{\mathbb{R}^3} \phi(\|\nabla F^m\|)\mathrm{d}\mathbf{x},$$

$$P_2 = \frac{\tau_m}{2} \|d_t F^m\|_{L^2(\mathbb{R}^3)}^2 + \frac{(2\alpha - 1)\tau_m^2 \sum_{i=1}^p \|d_t(\mathbf{P}_{\theta_i} F^m - g_i)\|_{L^2(\mathbb{R}^2)}^2}{2}$$
$$+ \frac{\lambda}{2} \int_{\mathbb{R}^3} \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|}(2\alpha - 2)\tau_m^2 \|\nabla d_t F^m\|^2 \mathrm{d}\mathbf{x}.$$

Because

$$\tau_m^2 \|d_t F^m\|_{L^2(\mathbb{R}^3)}^2 = \|X^m - X^{m-1}\|_M^2,$$

$$\tau_m^2 \sum_{i=1}^p \|d_t(\mathbf{P}_{\theta_i} F^m - g_i)\|_{L^2(\mathbb{R}^2)}^2 = \|X^m - X^{m-1}\|_R^2,$$

$$\tau_m^2 \int_{\mathbb{R}^3} \frac{\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|} \|\nabla d_t F^m\|^2 \mathrm{d}\mathbf{x} = \|X^m - X^{m-1}\|_Q^2,$$

then under the constraint (4.2), we have

$$2P_2 \geq \tau_m^2 \sum_{i=1}^p \|d_t(\mathbf{P}_{\theta_i} F^m - g_i)\|_{L^2(\mathbb{R}^2)}^2.$$

Therefore, (4.13) becomes

$$\frac{\tau_m}{2}\|d_t F^m\|^2_{L^2(\mathbb{R}^3)} + \frac{\tau_m^2 \sum_{i=1}^p \|d_t(\mathbf{P}_{\theta_i} F^m - g_i)\|^2_{L^2(\mathbb{R}^2)}}{2}$$
$$+ \frac{\sum_{i=1}^p \tau_m d_t \|\mathbf{P}_{\theta_i} F^m - g_i\|^2_{L^2(\mathbb{R}^2)}}{2} + \lambda \tau_m d_t \int_{\mathbb{R}^3} \phi(\|\nabla F^m\|) \mathrm{d}\mathbf{x} \le 0. \quad (4.14)$$

Applying the summation operation $\sum_{m=1}^l$ to the above inequality, we get (4.3). $\quad \square$

REMARK 4.1.    From Theorem 4.1, we have the following conclusions.

1. If $\alpha = 0$, the iterative scheme (3.12) is explicit. Hence, the inequality (4.2) becomes

$$0 < \tau_m \le \frac{\|X^{(m)} - X^{(m-1)}\|^2_M}{2\|X^{(m)} - X^{(m-1)}\|^2_R + 2\|X^{(m)} - X^{(m-1)}\|^2_Q}, \quad (4.15)$$

where

$$X^{(m)} = X^{(m-1)} + \tau_m M^{-1}[B - (R+Q)X^{(m-1)}].$$

From the numerical point of view, the temporal step-size should be finite for the convergence of the explicit iterative scheme which is consistent with the theoretical constraint (4.15).

2. If $\alpha = 1$, the iterative scheme (3.12) is semi-implicit. The inequality (4.2) becomes

$$0 < \tau_m \le +\infty. \quad (4.16)$$

Hence, the choice of the temporal step-size is arbitrary which is consistent with the result of the semi-implicit iterative scheme in [5].

3. If $\alpha \in (0,1)$, the iterative scheme (3.12) is a blended iterative scheme. As a result of the equivalence of norms in the finite element space, there exist positive parameters $a$ and $b$ such that

$$a\|v\|_R \le \|v\|_M, \quad b\|v\|_Q \le \|v\|_M.$$

Then

$$\frac{\|X^{(m)} - X^{(m-1)}\|^2_M}{(2-2\alpha)\|X^{(m)} - X^{(m-1)}\|^2_R + (2-2\alpha)\|X^{(m)} - X^{(m-1)}\|^2_Q}$$
$$\ge \frac{a^2 b^2}{2(1-\alpha)(a^2 + b^2)}. \quad (4.17)$$

Hence, if

$$\tau_m \le \frac{a^2 b^2}{2(1-\alpha)(a^2 + b^2)},$$

(4.2) is satisfied. Therefore, the constraint on $\tau_m$ is nonempty.

REMARK 4.2.    For the case $\alpha \in (0,1)$, using (3.23), we can estimate a bound $\tau_m^{(1)}$ for $\tau_m$ by solving a cubic equation. Let

$$\tau_m^{(1)} = \min\left\{\tau_m > 0 : \tau_m = \frac{\|X^{(m)} - X^{(m-1)}\|_M^2}{2(1-\alpha)[\|X^{(m)} - X^{(m-1)}\|_R^2 + \|X^{(m)} - X^{(m-1)}\|_Q^2]}\right\}.$$

Then we require that $\tau_m \leq \tau_m^{(1)}$. We can prove that there exists a positive constant $c$ with $\tau_m^{(1)} \geq c$.

Based on Theorem 4.1, we can establish the convergence result. Given $\alpha \in [0,1]$, let $\{F^m\}_{m=1}^{m_0}$ be the finite element solutions with $\tau_m$ satisfying constraint (4.2). In what follows we give its constant and linear interpolation in the temporal direction (see [13]):

$$\overline{F}^{h,\tau}(\mathbf{x},t) := F^{m-1}(\mathbf{x}), \quad \forall t \in [t_{m-1}, t_m), \tag{4.18}$$

$$\overline{\overline{F}}^{h,\tau}(\mathbf{x},t) := \frac{t - t_{m-1}}{\tau_m}F^m(\mathbf{x}) + \frac{t_m - t}{\tau_m}F^{m-1}(\mathbf{x}), \quad \forall t \in [t_{m-1}, t_m], \tag{4.19}$$

for $1 \leq m \leq m_0$, where $m_0$ is defined by

$$m_0 = \arg\min\left\{l \geq 1 : \sum_{m=1}^{l}\tau_m \geq T_0\right\},$$

and $T_0$ is a given positive number. We assume $\tau_{m_0}$ is truncated such that

$$\sum_{m=1}^{m_0}\tau_m = T_0.$$

Obviously, $\overline{F}^{h,\tau}$ is continuous with respect to $\mathbf{x}$ but discontinuous in $t$. However, $\overline{\overline{F}}^{h,\tau}$ is continuous with respect to both $\mathbf{x}$ and $t$.

THEOREM 4.2.    *Assume that $f_0 \in L^2(\mathbb{R}^3)$ with support $\Omega$, $g_i \in L^2(\mathbb{R}^2)$, and there exists a unique solution $f \in L^\infty((0,T_0); BV(\Omega)) \cap H^1((0,T_0); L^2(\mathbb{R}^3))$ of the gradient flow (3.8). Then under the following constraint of the initial value $f_0$,*

$$\lim_{h \to 0}\|f_0 - F^0\|_{L^2(\mathbb{R}^3)} = 0, \tag{4.20}$$

*the following results are valid:*

$$\lim_{h,\tau \to 0}\|f - \overline{\overline{F}}^{h,\tau}\|_{L^\infty((0,T_0); L^p(\mathbb{R}^3))} = 0, \tag{4.21}$$

$$\lim_{h,\tau \to 0}\|f - \overline{F}^{h,\tau}\|_{L^\infty((0,T_0); L^p(\mathbb{R}^3))} = 0, \tag{4.22}$$

*for any $p \in [1, \frac{n}{n-1})$ $(n = 3)$.*

*Proof.*    To show (4.21) and (4.22), we first notice that (4.3) implies the following (uniform in both $h$ and $k$) estimates:

$$\|\overline{\overline{F}}_t^{h,\tau}\|_{L^2(L^2(\mathbb{R}^3))} = \left(\sum_{m=1}^{m_0}\tau_m\|d_t F^m\|_{L^2(\mathbb{R}^3)}^2\right)^{\frac{1}{2}} \leq C, \tag{4.23}$$

$$\|\mathbf{P}_{\theta_i}\overline{F}^{h,\tau}\|_{L^\infty(L^2(\mathbb{R}^2))} \leq \|\mathbf{P}_{\theta_i}\overline{\overline{F}}^{h,\tau}\|_{L^\infty(L^2(\mathbb{R}^2))} = \max_{0\leq m\leq m_0}\|\mathbf{P}_{\theta_i}F^m\|_{L^2(\mathbb{R}^2)}$$

$$\leq \max_{0\leq m\leq m_0}\|\mathbf{P}_{\theta_i}F^m - g_i\|_{L^2(\mathbb{R}^2)} + \|g_i\|_{L^2(\mathbb{R}^2)} \leq C, \qquad (4.24)$$

$$\|\nabla\overline{F}^{h,\tau}\|_{L^\infty(L^1(\mathbb{R}^3))} \leq \|\nabla\overline{\overline{F}}^{h,\tau}\|_{L^\infty(L^1(\mathbb{R}^3))} = \max_{0\leq m\leq m_0}\|\nabla F^m\|_{L^1(\mathbb{R}^3)}$$

$$\leq \max_{0\leq m\leq m_0}\int_{\mathbb{R}^3}\phi(\|\nabla F^m\|)\mathrm{d}\mathbf{x} \leq C, \qquad (4.25)$$

$$\sum_{m=1}^{m_0}\sum_{i=1}^p\|\mathbf{P}_{\theta_i}F^m - \mathbf{P}_{\theta_i}F^{m-1}\|_{L^2(\mathbb{R}^2)}^2$$

$$=\sum_{m=1}^{m_0}\tau_m^2\sum_{i=1}^p\|d_t(\mathbf{P}_{\theta_i}F^m - g_i)\|_{L^2(\mathbb{R}^2)}^2 \leq C, \quad \text{if} \quad \lambda\neq 0, \ (4.26)$$

where $C$ is a constant. Then taking the test function $v_h$ in (3.12) as $F^m$, we have

$$\int_{\mathbb{R}^3}\left[(d_tF^m)^T F^m + \frac{\lambda\phi'(\|\nabla F^{m-1}\|)}{\|\nabla F^{m-1}\|}(\alpha\nabla F^m + (1-\alpha)\nabla F^{m-1})^T\nabla F^m\right]\mathrm{d}\mathbf{x}$$

$$+\sum_{i=1}^p\int_{\mathbb{R}^2}[(\alpha\mathbf{P}_{\theta_i}F^m + (1-\alpha)\mathbf{P}_{\theta_i}F^{m-1}) - g_i]\mathbf{P}_{\theta_i}F^m\mathrm{d}\mathbf{y} = 0. \ (4.27)$$

For the first term in the integration of (4.27), we have

$$(d_tF^m)^T F^m = \frac{F^m - F^{m-1}}{\tau_m}F^m = \frac{d_t|F^m|^2}{2} + \frac{\tau_m|d_tF^m|^2}{2}. \qquad (4.28)$$

Hence, using the matrix notations, equation (4.27) becomes

$$\frac{d_t\|F^m\|_{L^2(\mathbb{R}^2)}^2}{2} + \frac{\tau_m\|d_tF^m\|_{L^2(\mathbb{R}^2)}^2}{2} + \alpha X^{(m)T}QX^{(m)} + (1-\alpha)X^{(m)T}QX^{(m-1)}$$

$$+ \alpha X^{(m)T}RX^{(m)} + (1-\alpha)X^{(m)T}RX^{(m-1)} - X^{(m)T}B$$

$$=\frac{d_t\|F^m\|_{L^2(\mathbb{R}^2)}^2}{2} + \frac{\tau_m\|d_tF^m\|_{L^2(\mathbb{R}^2)}^2}{2}$$

$$+X^{(m)T}[\alpha(R+Q)X^{(m)} + (1-\alpha)(R+Q)X^{(m-1)} - B]$$

$$=0. \qquad (4.29)$$

Note that

$$(X^{(m)})^T[\alpha(R+Q)X^{(m)} - \alpha B]$$

$$=\alpha\left[\|X^{(m)}\|_Q^2 + \frac{1}{2}\sum_{i=1}^p\|\mathbf{P}_{\theta_i}F^m - g_i\|_{L^2(\mathbb{R}^2)}^2 + \|\mathbf{P}_{\theta_i}F^m\|_{L^2(\mathbb{R}^2)}^2 - \|g_i\|_{L^2(\mathbb{R}^2)}^2\right]. \ (4.30)$$

Using the iterative scheme (4.1), we obtain

$$(X^{(m)})^T[(R+Q)X^{(m-1)} - B]$$

$$=\left(X^{(m-1)} + \tau_m[M + \tau_m\alpha(R+Q)]^{-1}(B - (R+Q)X^{(m-1)})\right)^T[(R+Q)X^{(m-1)} - B]$$

$$=X^{(m-1)T}[(R+Q)X^{(m-1)} - B] - \tau_m\|(R+Q)X^{(m-1)} - B\|_{[M+\tau_m\alpha(R+Q)]^{-1}}^2$$

$$
\begin{aligned}
&= \|X^{(m-1)}\|_Q^2 + \frac{\sum_{i=1}^{p} \|\mathbf{P}_{\theta_i} F^{m-1} - g_i\|_{L^2(\mathbb{R}^2)}^2 + \|\mathbf{P}_{\theta_i} F^{m-1}\|_{L^2(\mathbb{R}^2)}^2 - \|g_i\|_{L^2(\mathbb{R}^2)}^2}{2} \\
&\quad - \tau_m \|(R+Q)X^{(m-1)} - B\|_{[M+\tau_m \alpha(R+Q)]^{-1}}^2 \\
&= \|X^{(m-1)}\|_Q^2 + \frac{\sum_{i=1}^{p} \|\mathbf{P}_{\theta_i} F^{m-1} - g_i\|_{L^2(\mathbb{R}^2)}^2 + \|\mathbf{P}_{\theta_i} F^{m-1}\|_{L^2(\mathbb{R}^2)}^2 - \|g_i\|_{L^2(\mathbb{R}^2)}^2}{2} \\
&\quad - \tau_m^{-1} \|X^{(m)} - X^{(m-1)}\|_{M+\tau_m \alpha(R+Q)}^2.
\end{aligned}
\tag{4.31}
$$

Using (4.30) and (4.31), we can write equation (4.29) as

$$
\begin{aligned}
&\frac{d_t \|F^m\|_{L^2(\mathbb{R}^2)}^2}{2} + \frac{\tau_m \|d_t F^m\|_{L^2(\mathbb{R}^2)}^2}{2} \\
&\quad + X^{(m)^T} [\alpha(R+Q)X^{(m)} + (1-\alpha)(R+Q)X^{(m-1)} - B] \\
&= \frac{d_t \|F^m\|_{L^2(\mathbb{R}^2)}^2}{2} + \frac{\tau_m \|d_t F^m\|_{L^2(\mathbb{R}^2)}^2}{2} \\
&\quad + \alpha \left[ \|X^{(m)}\|_Q^2 + \frac{\sum_{i=1}^{p} \|\mathbf{P}_{\theta_i} F^m - g_i\|_{L^2(\mathbb{R}^2)}^2 + \|\mathbf{P}_{\theta_i} F^m\|_{L^2(\mathbb{R}^2)}^2 - \|g_i\|_{L^2(\mathbb{R}^2)}^2}{2} \right] \\
&\quad + (1-\alpha) \left[ \frac{\sum_{i=1}^{p} \|\mathbf{P}_{\theta_i} F^{m-1} - g_i\|_{L^2(\mathbb{R}^2)}^2 + \|\mathbf{P}_{\theta_i} F^{m-1}\|_{L^2(\mathbb{R}^2)}^2 - \|g_i\|_{L^2(\mathbb{R}^2)}^2}{2} \right. \\
&\qquad\qquad \left. + \|X^{(m-1)}\|_Q^2 - \tau_m^{-1} \|X^{(m)} - X^{(m-1)}\|_{M+\tau_m \alpha(R+Q)}^2 \right] \\
&= 0.
\end{aligned}
\tag{4.32}
$$

Ignoring the nonnegative terms

$$
\|X^{(m)}\|_Q^2, \quad \|\mathbf{P}_{\theta_i} F^m - g_i\|_{L^2(\mathbb{R}^2)}^2, \quad \|\mathbf{P}_{\theta_i} F^m\|_{L^2(\mathbb{R}^2)}^2,
$$

and

$$
\|X^{(m-1)}\|_Q^2, \quad \|\mathbf{P}_{\theta_i} F^{m-1} - g_i\|_{L^2(\mathbb{R}^2)}^2, \quad \|\mathbf{P}_{\theta_i} F^{m-1}\|_{L^2(\mathbb{R}^2)}^2,
$$

in (4.32), we obtain

$$
\begin{aligned}
&\frac{d_t \|F^m\|_{L^2(\mathbb{R}^2)}^2}{2} + \frac{\tau_m \|d_t F^m\|_{L^2(\mathbb{R}^2)}^2}{2} - (1-\alpha)\tau_m^{-1} \|X^{(m)} - X^{(m-1)}\|_{M+\tau_m \alpha(R+Q)}^2 \\
&= \frac{d_t \|F^m\|_{L^2(\mathbb{R}^2)}^2}{2} + (\alpha - \frac{1}{2})\tau_m^{-1} \|X^{(m)} - X^{(m-1)}\|_M^2 - \alpha(1-\alpha)\|X^{(m)} - X^{(m-1)}\|_{R+Q}^2 \\
&\leq \sum_{i=1}^{p} \|g_i\|_{L^2(\mathbb{R}^2)}^2.
\end{aligned}
\tag{4.33}
$$

By (4.33), we have

$$
\begin{aligned}
&\frac{d_t \|F^m\|_{L^2(\Omega)}^2}{2} + \alpha \tau_m \|d_t F^m\|_{L^2(\Omega)}^2 - \alpha(1-\alpha)\|f^m - f^{m-1}\|_{R+\lambda Q^{m-1}}^2 \\
&\leq \sum_{i=1}^{p} \|g_i\|_{L^2(\mathbb{R}^2)}^2 + \frac{\tau_m \|d_t F^m\|_{L^2(\Omega)}^2}{2}.
\end{aligned}
\tag{4.34}
$$

Using the constrained condition (4.2), we have

$$\alpha\tau_m\|d_tF^m\|_{L^2(\Omega)}^2 - \alpha(1-\alpha)\|f^m - f^{m-1}\|_{R+\lambda Q^{m-1}}^2 \geq 0.$$

Hence by the inequality above and (4.34), we obtain

$$\frac{d_t\|F^m\|_{L^2(\Omega)}^2}{2} \leq \sum_{i=1}^{p}\|g_i\|_{L^2(\mathbb{R}^2)}^2 + \frac{\tau_m\|d_tF^m\|_{L^2(\Omega)}^2}{2}. \tag{4.35}$$

Conducting operations $2\sum_{m=1}^{l}\tau_m$ on the two sides of (4.35), we have

$$\|F^l\|_{L^2(\Omega)}^2 \leq 2\sum_{m=1}^{l}\tau_m\sum_{i=1}^{p}\|g_i\|_{L^2(\mathbb{R}^2)}^2 + \|F^0\|_{L^2(\Omega)}^2 + \sum_{m=1}^{l}\tau_m^2\|d_tF^m\|_{L^2(\Omega)}^2, \tag{4.36}$$

for $\forall 1 \leq l \leq m_0$. According to the conclusion (4.3) of Theorem 4.1, there exists a positive constant $C$ satisfying the following inequality:

$$\sum_{m=1}^{m_0}\tau_m\|d_tF^m\|_{L^2(\Omega)}^2 \leq C.$$

Assuming that $0 < \tau_m < 1$, we immediately obtain

$$\sum_{m=1}^{m_0}\tau_m^2\|d_tF^m\|_{L^2(\Omega)}^2 \leq \sum_{m=1}^{m_0}\tau_m\|d_tF^m\|_{L^2(\Omega)}^2 \leq C. \tag{4.37}$$

Thus using (4.36) and (4.37), we have

$$\|F^l\|_{L^2(\Omega)}^2 \leq 2\sum_{m=1}^{l}\tau_m\sum_{i=1}^{p}\|g_i\|_{L^2(\mathbb{R}^2)}^2 + \|F^0\|_{L^2(\Omega)}^2 + C, \quad \forall 1 \leq l \leq m_0. \tag{4.38}$$

Utilizing the known condition $f_0 \in L^2(\mathbb{R}^3)$, $g_i \in L^2(\mathbb{R}^2)$, and (4.20), we have

$$2\sum_{m=1}^{l}\tau_m\sum_{i=1}^{p}\|g_i\|_{L^2(\mathbb{R}^2)}^2 + \|F^0\|_{L^2(\Omega)}^2 \leq 2T_0\sum_{i=1}^{p}\|g_i\|_{L^2(\mathbb{R}^2)}^2 + \|F^0\|_{L^2(\Omega)}^2 \leq C. \tag{4.39}$$

Combining (4.38) and (4.39), we obtain

$$\|F^l\|_{L^2(\Omega)}^2 \leq C, \quad \forall 1 \leq l \leq m_0. \tag{4.40}$$

As a result of (4.40), we have

$$\|\overline{F}^{h,\tau}\|_{L^\infty(L^2(\mathbb{R}^3))} \leq \|\overline{\overline{F}}^{h,\tau}\|_{L^\infty(L^2(\mathbb{R}^3))} = \max_{0 \leq m \leq m_0}\|F^m\|_{L^2(\mathbb{R}^3)} \leq C. \tag{4.41}$$

Then based on (4.23), (4.25), (4.37), and (4.41), there exists a convergent subsequence of $\{\overline{\overline{F}}^{h,\tau}\}$ (denoted by the same notation) (see [29, 13]) and a function $f \in L^\infty((0,T_0);BV(\Omega)) \cap H^1((0,T_0);L^2(\mathbb{R}^3))$ such that as $h,k \to 0$,

$$\begin{aligned}
\overline{\overline{F}}^{h,\tau} \longrightarrow f \quad &weakly\star \quad in \quad L^\infty((0,T_0);L^2(\mathbb{R}^3)), \\
&weakly \quad in \quad L^2((0,T_0);L^2(\mathbb{R}^3)), \\
&strongly \quad in \quad L^p(\Omega), 1 \leq p < \frac{n}{n-1}, \ for \ a.e. \ t \in [0,T_0],
\end{aligned} \tag{4.42}$$

and

$$\overline{\overline{F}}_t^{h,\tau} \longrightarrow f_t \quad weakly \ \ in \ \ L^2((0,T_0);L^2(\mathbb{R}^3)). \tag{4.43}$$

Here we have used the fact that $BV(\Omega)$ is compactly embedded in $L^p(\Omega)$ for $1 \leq p < \frac{n}{n-1}$. Notice that the assumption on $F^0$ implies that $f(0) = f_0$. As the authors have done in [13, 5], we can further obtain that the whole sequence $\overline{\overline{F}}_t^{h,\tau}$ converges to $f$. Therefore, the proof of (4.21) is completed.

Now we prove (4.22). Using (4.3), we have

$$\sum_{m=1}^{m_0} \tau_m \|d_t F^m\|_{L^2(\mathbb{R}^3)}^2 \leq C.$$

According to above formula, it is easy to show that

$$
\begin{aligned}
\|\overline{\overline{F}}^{h,\tau} - \overline{F}^{h,\tau}\|_{L^2(L^2(\mathbb{R}^3))}^2 &= \int_0^{T_0} \|\overline{\overline{F}}^{h,\tau} - \overline{F}^{h,\tau}\|_{L^2(\mathbb{R}^3)}^2 \mathrm{d}t \\
&= \frac{1}{3} \sum_{m=1}^{m_0} \tau_m^3 \|d_t F^m\|_{L^2(\mathbb{R}^3)}^2 \\
&\leq \frac{\tau^2}{3} \sum_{m=1}^{m_0} \tau_m \|d_t F^m\|_{L^2(\mathbb{R}^3)}^2 \leq C\tau^2.
\end{aligned}
\tag{4.44}
$$

Using (4.21) and (4.44), we obtain (4.22). Therefore, for the blended scheme (3.12), the proof of convergence is completed. ☐

THEOREM 4.3.    *Under the conditions of Theorem 4.1, the sequence $\{E(F^m)\}_{m=0}^{\infty}$, obtained from the blended finite element scheme (4.1), satisfies*

$$E(F^m) < E(F^{m-1}), \quad m=1,2,\ldots. \tag{4.45}$$

*Furthermore, if $\tau_m$ satisfies*

$$c \leq \tau_m \leq \frac{\|X^{(m)} - X^{(m-1)}\|_M^2}{2(1-\alpha)[\|X^{(m)} - X^{(m-1)}\|_R^2 + \|X^{(m)} - X^{(m-1)}\|_Q^2]}, \tag{4.46}$$

*then we have*

$$\lim_{m \to \infty} F^m = F^*, \tag{4.47}$$

*where $F^*$ is the minimum solution of the minimization problem (3.5) in the space $V^h$, and c is a positive constant.*

*Proof.*    The results of this theorem can be easily obtained from the corresponding convergence analysis of the gradient-flow-based semi-implicit finite element method in [5]. It should be pointed out that $\tau_m$ satisfying the restriction (4.46) exists (see Remark 4.2). ☐

REMARK 4.3.    If $\alpha = 0$, the blended finite element method becomes the explicit scheme. The convergence result we obtained shows that the explicit scheme is convergent under the constraint (4.15) on the temporal step-size. It should be pointed out that in the convergence analysis for the blended finite element method, $X^{(m)}$ is computed from (4.1), while in Algorithm 3.1, $X^{(m)}$ is approximately computed by (3.23). Because the convergence of the blended scheme depends mainly on the temporal step-size as does that of the explicit scheme, if the temporal step-size is chosen properly, then the algorithm is convergent.

## 5. Numerical results

In this section, we present three numerical experiments. The aim of the first experiment is to show the effect of the regularizer for noisy projection images using a synthetic dataset. The second experiment is used to illustrate that our blended finite element method is convergent using a given function to be reconstructed. In the third experiment, we compare the efficiency of the blended finite element method with those of the explicit finite element method and the semi-implicit finite element method, using a real dataset.

**5.1. Effect of the regularizer.**     For noisy projection images, applying the regularizer in the reconstruction process is necessary. In this test, we first project a phantom (the widely used 3D Shepp–Logan phantom) of volume data (without noise) with size $129 \times 129 \times 129$ to produce a set of 2D projection images with size $129 \times 129$. The number of total projection images is 1296. The projection angles, $\theta_i$, are uniformly distributed around the spherical coordinates. The 3D image to be reconstructed has a size of $129 \times 129 \times 129$. Then these images are artificially polluted by adding the additive Gaussian white noise with noise levels of 6 dB and 12 dB, respectively, obtaining two sets of images. When the additive Gaussian white noise is added to the projections, the signal-to-noise ratio (SNR) of detected data in decibels is defined as

$$\text{SNR} = 10\log_{10} \frac{1}{VP\sigma^2} \sum_{v=0}^{V-1} \sum_{p=0}^{P-1} \left| g_v(y_p) \right|^2, \tag{5.1}$$

where $V$ is the number of the angles, $P$ is the number of projections from each angle and $\sigma^2$ is the variance of noise [8].
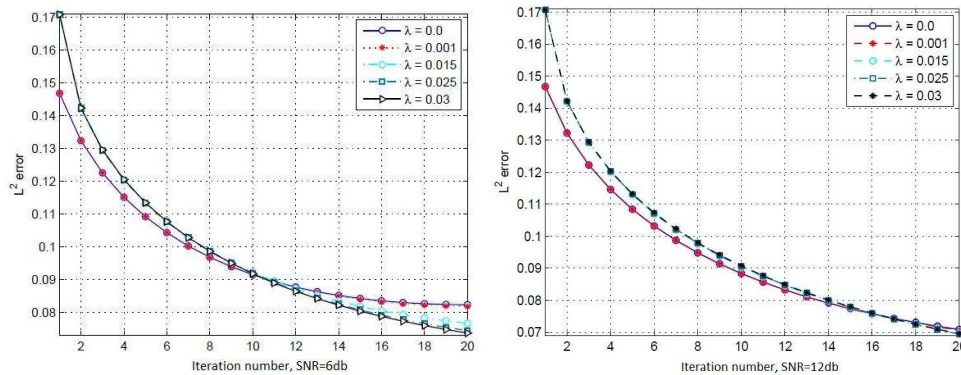


FIG. 5.1. *The graph plots of the $L^2$-errors of the exact volume data and the reconstructed volume data using different $\lambda$ for each set of the noisy images.*

Finally, we reconstruct the volume data via the noisy images using our blended finite element method. Figure 5.1 shows the $L^2$-errors between the exact volume data and the reconstructed volume data using different $\lambda$ for each set of noisy images. Here the $L^2$-error is defined as $[\sum_{i,j,k}^{N} (f_{ijk}^{\text{ex}} - f_{ijk}^{\text{re}})^2 / N^3]^{1/2}$, where $f^{\text{ex}}$ and $f^{\text{re}}$ denote the exact and the reconstructed volume data, respectively. $N$ is the dimension of the volume data in each direction. All the reconstructed volume data are the results of the blended finite element method after 20 iterations.

| SNR | $\lambda = 0$ | $\lambda = 0.001$ | $\lambda = 0.015$ | $\lambda = 0.025$ | $\lambda = 0.03$ |
|---|---|---|---|---|---|
| 6dB | 0.082 300 | 0.081 761 | 0.076 641 | 0.074 744 | 0.073 688 |
| 12dB | 0.070 841 | 0.070 587 | 0.069 594 | 0.069 406 | 0.069 477 |

TABLE 5.1. $L^2$-errors between the reconstructed functions and the exact function for different $\lambda$ and different SNR.

In table 5.1, we list the $L^2$-errors between the exact volume data and the reconstructed volume data. For the first set of images, $\lambda = 0.03$ leads to the best fitting, while for the second set of data, $\lambda = 0.025$ leads to the best reconstruction results. As shown in table 5.1, the detected data of 6 dB have more noise than those of 12 dB, and it makes sense that noisier data to be reconstructed should require a larger value of $\lambda$. Generally speaking, the ideal choice of the factor $\lambda$ depends on the noise level. Considering the fact that for the real data the exact volume data are unknown, how to choose a value of $\lambda$ that leads to the best reconstruction results in certain senses (for instance, in the $L^2$-sense) is a difficult problem. To further illustrate the effectiveness of the regularization term, we show, in figure 5.2 and figure 5.3, slices of reconstructed images after 20 iterations for comparison.
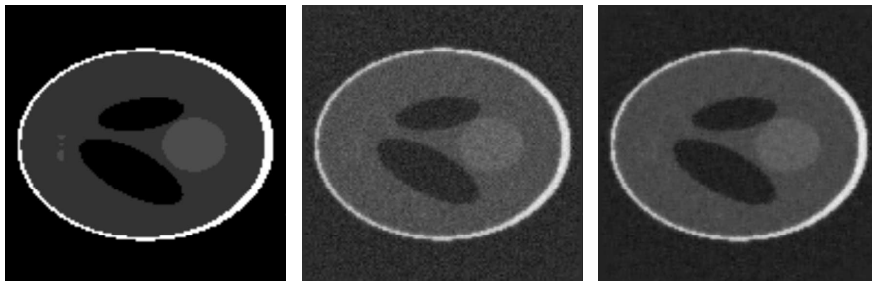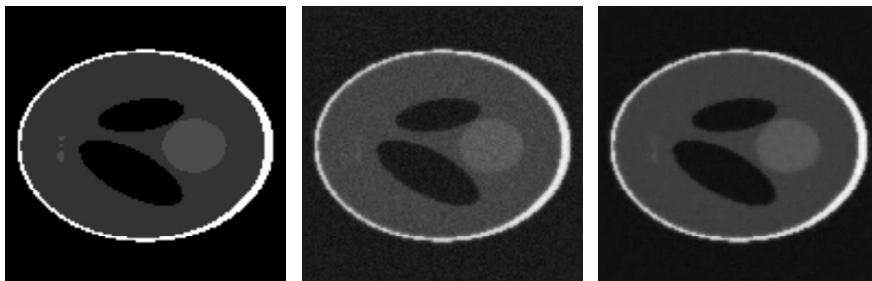


FIG. 5.2. SNR = 6dB, the central slices of volume data along z-axis. Left: The true Shepp-Logan phantom. Middle: The reconstructed volume data without regularization term ($\lambda = 0$). Right: The reconstructed volume data with regularization term ($\lambda = 0.03$).



FIG. 5.3. SNR = 12dB, the central slices of volume data along z-axis. Left: The true Shepp-Logan phantom. Middle: The reconstructed volume data without regularization term ($\lambda = 0$). Right: The reconstructed volume data with regularization term ($\lambda = 0.025$).

As shown in figure 5.2, the performance of the reconstruction when $\lambda = 0.03$ is better than that when $\lambda = 0$. In addition, figure 5.3 shows that the performance of the reconstruction when $\lambda = 0.025$ is better than that when $\lambda = 0$. In the following experiments, we do not address the effect of $\lambda$ and fix a value of $\lambda$. Of course, that may not be the best $\lambda$ for those data.

**5.2. Convergence test.** In these experiments, we are given a function $f$ on a domain $\Omega$. We first compute the projection images from a set of uniformly distributed projection directions around the spherical coordinates as the first experiment does. The size of these projection images is $143 \times 143$. Then we reconstruct $f$ using the projected data. The size of the reconstructed volume $f$ is $143 \times 143 \times 143$. Because the data are clean, we take a small $\lambda$ $(=0.001)$ to ensure that the reconstruction problem is well-posed. For increasing numbers of iterations and projection directions, we obtain a table of the reconstructed functions. The $L^2$-errors between the reconstructed functions and the exact function $f$ are computed. Table 5.2 lists these $L^2$-errors for iteration numbers 3, 6, 9, 12, 15, 18, and 21. The exact function to be projected is taken as a Gaussian map

$$f(\mathbf{x}) = \sum_{i=1}^{10} e^{-\sigma\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{r_i^2}-1\right)}, \tag{5.2}$$

where $\sigma = 0.125$, $\mathbf{x} \in \Omega = [-32,-8] \times [-26,-2] \times [-35,-11]$, $[\mathbf{x}_i, r_i]$ are taken as

$$[16.0, 27.0, 26.0, 20.0], \quad [19.2, 35.0, 32.0, 20.0], \quad [22.4, 43.0, 38.0, 20.0],$$
$$[25.6, 31.0, 31.0, 20.0], \quad [28.8, 39.0, 37.0, 24.0], \quad [32.0, 27.0, 30.0, 24.0],$$
$$[35.2, 35.0, 36.0, 24.0], \quad [38.4, 43.0, 29.0, 24.0], \quad [41.6, 31.0, 35.0, 28.0],$$
$$[44.8, 39.0, 28.0, 28.0].$$

|    | $3 \times 3$ | $6 \times 6$ | $12 \times 12$ | $24 \times 24$ | $48 \times 48$ | $96 \times 96$ |
|----|----------|----------|----------|----------|----------|----------|
| 3  | 0.060 928 | 0.039 551 | 0.034 565 | 0.034 424 | 0.034 423 | 0.034 422 |
| 6  | 0.055 311 | 0.023 134 | 0.014 116 | 0.013 738 | 0.013 731 | 0.013 730 |
| 9  | 0.053 429 | 0.017 421 | 0.007 177 | 0.006 572 | 0.006 550 | 0.006 540 |
| 12 | 0.052 398 | 0.014 749 | 0.004 348 | 0.003 651 | 0.003 600 | 0.003 573 |
| 15 | 0.051 702 | 0.013 295 | 0.003 065 | 0.002 446 | 0.002 357 | 0.002 305 |
| 18 | 0.051 178 | 0.012 400 | 0.002 431 | 0.001 987 | 0.001 868 | 0.001 793 |
| 21 | 0.050 755 | 0.011 788 | 0.002 096 | 0.001 832 | 0.001 695 | 0.001 604 |

TABLE 5.2. $L^2$-errors between the reconstructed functions and the exact function.

From the table, we can see that the $L^2$-errors decrease as the numbers of projection directions and iterations increase. Hence, the convergence can be observed. In addition, as shown in figure 5.4, if the number of projections is large, the reconstructed image is almost the same as the true image. However, if the number of projections is small, the reconstructed function has artifacts and does not approximate well the exact function, although the algorithm is convergent.

**5.3. Efficiency test.** For the efficiency test, we use a set of real tomography images, named as 1FFK. This set has 5000 images with size $143 \times 143$. The size of the reconstructed 3D volume is $143 \times 143 \times 143$. These images are very noisy
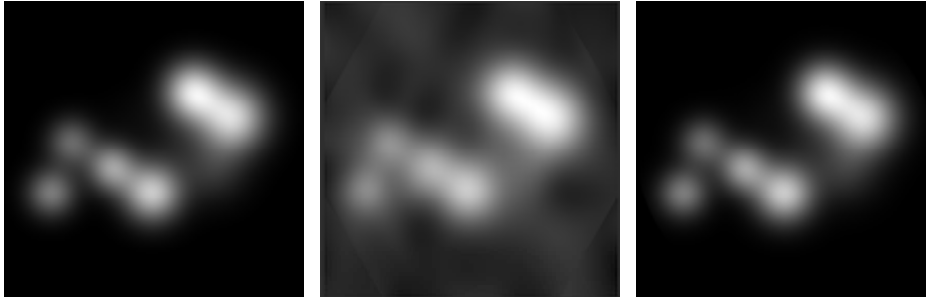
FIG. 5.4. *The central slices of volume data perpendicular to z-axis. Left: The exact image. Middle: The reconstructed image by $3 \times 3$ uniformly distributed projections with $\lambda = 0.001$. Right: The reconstructed image by $96 \times 96$ uniformly distributed projections with $\lambda = 0.001$.*

and we choose $\lambda = 0.05$. We compare the computational complexity of the blended finite element method with those of the explicit and the semi-implicit finite element methods.

To be fair in the comparison, we first set a time upper bound $T$. Then for the explicit and the blended finite element methods, we perform the reconstruction iteratively, and at the same time we compute $\tau = \sum_{s=1}^{K} \tau_s$, for a certain $K$, until $\tau \geq T$. For the semi-implicit finite element method, we iterate once with temporal step-size $T$, because the semi-implicit finite element method has no limitation on the temporal step-size. The resulting system (3.20) is solved by the GMRES iterative method, in which the control accuracy is set to be $10^{-5}$. Then we compute the required CPU time for each of the three cases.

Here all the numerical computations are conducted in parallel on a desktop equipped with an Intel(R) Xeon(R) CPU X5550 (2.76GHz) with 8 Cores. We set the time upper bound $T = 0.38$. For the blended finite element method, the temporal step-size is taken as $\tau_1 = 0.00606$ for the first iteration. The temporal step-size of the later iterations is taken as 0.02. Hence the number of iterations of the blended scheme is 20. The required CPU time is about 1.225 h for each iteration. The total time cost is 24.5 h. For the explicit scheme, the CPU time is about 0.4 h for each iteration. The explicit scheme is not convergent if the temporal step-size is taken as that of the blended scheme. In order to make the explicit scheme converge, its temporal step-size should be less than 0.0025. To arrive at $T = 0.38$, at least 152 iterations are required. Accordingly, the total time cost is at least 61 h. Therefore, the convergence rate of the blended scheme is much faster than that of the explicit scheme. On the other hand, using the semi-implicit finite element method, the required CPU time is about 26.25 h for one iteration with the temporal step-size taken as 0.38.

In addition, we use the slices and isocontours of reconstructed images to compare the performance of the blended scheme with those of the explicit and semi-implicit schemes. As shown in figure 5.5 and figure 5.6, the reconstruction results, with artifacts at the background, from the blend scheme and the explicit scheme are almost the same. Because the measured images are very noisy, these results are trustworthy. Comparing with these results, the semi-implicit scheme gives much smoother result. In fact, using one iteration for the semi-implicit scheme is not enough to obtain reasonable results because the scheme is not fully implicit. Some quantities are treated as explicit (using previous step data). Because the previous step data is the initial value

which is taken as zero, the quantities computed using previous data have no accuracy at all. Therefore, at least two iterations are needed. This increases the computational time to about 52.5 h.

In conclusion, not only does the blended scheme have better performance than the semi-implicit scheme, but the blended scheme requires less CPU time. Hence, the numerical comparisons indicate that the blended finite element method is numerically convergent and more efficient than the other two methods.
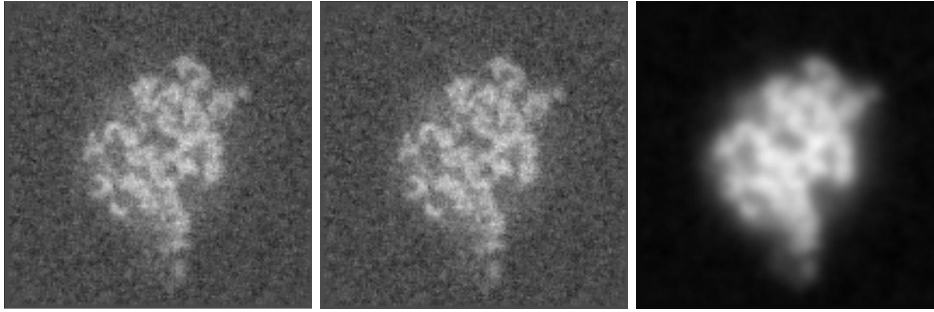


Fig. 5.5. *The central slices of reconstructed images perpendicular to z-axis. Left: The blended finite element method at* $T = 0.38$ *with 20 iterations. Middle: The explicit finite element method at* $T = 0.38$ *with 152 iterations. Right: The semi-implicit finite element method at* $T = 0.38$ *with one iteration.*
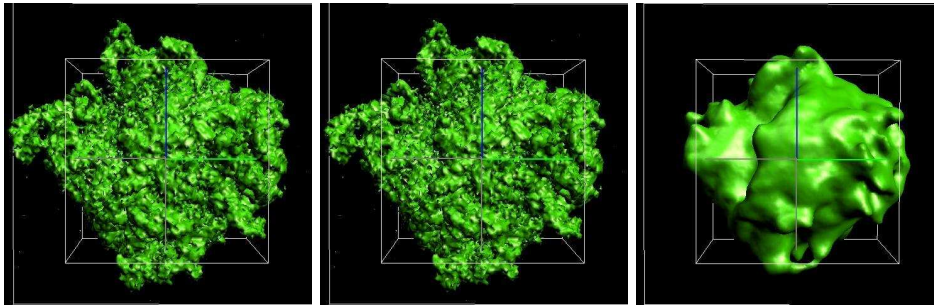


Fig. 5.6. *The middle-value isocontours of the reconstructed images. Left: The blended finite element method at* $T = 0.38$ *with 20 iterations. Middle: The explicit finite element method at* $T = 0.38$ *with 152 iterations. Right: The semi-implicit finite element method at* $T = 0.38$ *with one iteration.*

### 6. Conclusions

We have presented a gradient-flow-based blended finite element method which includes the explicit and semi-implicit schemes as its special cases. The proposed algorithm facilitates the optimal temporal step-size determination. The convergence analysis of the proposed algorithm has been conducted, demonstrating that the method is convergent. The numerical results illustrate that the presented method is also convergent and more efficient than both the explicit and semi-implicit schemes.

## REFERENCES

[1] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, Springer, Second Edition, 2000.

[2] E. Candès, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inf. The., 52(2), 489–509, 2006.

[3] P. Charbonnier, L. Blanc-Férand, G. Aubert, and M. Barlaud, *Globally deterministic edge-preserving regularized in compued imaging*, IEEE Trans. Image Proc., 6, 298–311, 1997.

[4] C. Chen and G. Xu, *A new reconstruction algorithm in tomography with geometric feature-preserving regularization*, in Proceedings of 2010 3rd International Conference on Biomedical Engineering and Informatics, Yantai, China, IEEE, 1, 41–45, 2010.

[5] C. Chen and G. Xu, *Gradient-flow-based semi-implicit finite-element method and its convergence analysis for image reconstruction*, Inv. Prob., 28(3), 035006, 2012.

[6] C. Chen and G. Xu, *Computational inversion of electron micrographs using $L^2$-gradient flows — convergence analysis*, Math. Meth. Appl. Sci., 36(18), 2492–2506, 2013.

[7] S. Deans, *The Radon Transform and Some of Its Applications*, John Wiley and Sons, New York, 1983.

[8] A. Delaney and Y. Bresler, *A fast and accurate fourier algorithm for iterative parallel-beam tomography*, IEEE Trans. Image Proc., 5(5), 740–753, 1996.

[9] A. Delaney and Y. Bresler, *Globally convergent edge-preserving regularized reconstruction: An application to limited-angle tomography*, IEEE Trans. Image Proc., 7, 204–221, 1998.

[10] D. Donoho, *Compressed sensing*, IEEE Trans. Inf. The., 52(4), 1289–1306, 2006.

[11] L. Evans and R. Gariepy, *Measure Theory and Fine Properties of Functions*, British Columbia, Canada: CNC Press, 1992.

[12] A. Faridani, *Introduction to the mathematics of computed tomography*, in Inside Out: Inverse Problems and Applications, G. Uhlmann (ed.), MSRI Publications, Cambridge University Press, 47, 1–46, 2003.

[13] X. Feng and A. Prohl, *Analysis of total varation flow and its finite element approximations*, ESIAM: Math. Mod. Num. Anal., 37, 533–556, 2003.

[14] J. Frank, *Electron Tomography: Methods for Three-Dimensional Visualization of Structures in the Cell*, Springer Verlag, 2006.

[15] J. Frank, *Three-Dimensional Electron Microscopy of Macromolecular Assemblies: Visualization of Biological Molecules in Their Native State*, Oxford University Press, New York, 2006.

[16] P. Gilbert, *Iterative methods for the three-dimensional reconstruction of an object from projections*, J. Theor. Biol., 36(1), 105–117, 1972.

[17] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sci., 2(2), 323–343, 2009.

[18] R. Gordon, R. Bender, and G. Herman, *Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography*, J. Theor. Biol., 29, 471–481, 1970.

[19] S. Helgason, *The Radon Transform*, Boston, USA: Birkhäuser, 1999.

[20] A. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*, SIAM, Philadelphia, 120, 2001.

[21] M. Li, G. Xu, C. Sorzano, F. Sun, and C. Bajaj, *Single-particle reconstruction using $L^2$-gradient flow*, J. Struct. Bio., 176, 259–267, 2011.

[22] F. Natterer, *The Mathematics of Computerized Tomography*, SIAM, Philadelphia, 2001.

[23] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative regularization method for total variation-based image restoration*, Multiscale Model. Simul., 4(2), 460–489, 2005.

[24] X. Pan, E. Sidky, and M. Vannier, *Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction?* Inv. Prob., 25, 123009, 2009.

[25] M. Radermacher, *Weighted back-projection methods*, J. Frank (ed.), Electron Tomography: Methods for Three-Dimensional Visualization of Structures in the Cell, Springer, New York, Second Edition, 8, 245–274, 2006.

[26] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D: Nonlin. Phen., 60, 259–268, 1992.

[27] Y. Saad and M. Schultz, *GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems*, SIAM J. Sci. Stat. Comput., 7(3), 856–869, 1986.

[28] E. Sidky, C. Kao, and X. Pan, *Accurate image reconstruction from few-views and limited-angle data in divergent-beam CT*, J. X-Ray Sci. Tech., 14(2), 119–139, 2006.

[29] J. Simon, *Compact sets in the space $L^p(0,T;B)$*, Ann. Mat. Pura. Appl., 4(146), 65–96, 1987.

[30] A. Tikhonov and V. Arsenin, *Solutions of Ill-Posed Problems*, V.H. Winston & Sons, Wash-

ington, D.C., 1977.

[31] L. Vese, *A study in the BV space of a denoising-deblurring variational problem*, App. Math. Opt., 44(2), 131–161, 2001.

[32] G. Xu, M. Li, A. Gopinath, and C. Bajaj, *Inversion of electron tomography images using $L^2$-gradient flows — computational methods*, J. Comput. Math., 29, 501–525, 2011.