An efficient numerical algorithm for the $L^2$ optimal transport problem with periodic densities

Louis-Philippe Saumier*, Martial Agueh and Boualem Khouider

Department of Mathematics and Statistics, University of Victoria, PO BOX 3060 STN, Victoria, BC, Canada V8W 3R4

*Corresponding author: lsaumier@uvic.ca agueh@uvic.ca khouider@uvic.ca

[Received on 26 June 2012; revised on 21 February 2013; accepted on 29 May 2013]

We present an extension of the numerical method of Loeper and Rapetti (2005, Numerical solution of the Monge–Ampère equation by a Newton’s algorithm. C.R. Acad. Sci. Paris, I, 319–324) for the Monge–Ampère equation to non-uniform target densities and adopt it to solve the optimal transport problem with quadratic cost. The method employs a damped Newton algorithm to solve the Monge–Ampère equation. We show that the algorithm converges for sufficiently large damping coefficients, for the case where the source and target densities are sufficiently smooth, periodic and bounded away from zero. At each Newton iteration, we solve a non-constant coefficient linear partial differential equation. To improve the efficiency of the procedure, we use an analytically preconditioned fast Fourier transform method coupled with GMRES (Strain, J. (1994) Fast spectrally-accurate solution of variable-coefficients elliptic problems. Proc. Amer. Math. Sci., 122, 843–850) to solve this equation, as opposed to a more straightforward approach based on a second-order finite-difference discretization combined with biconjugate gradient used in the original LOEPER and RAPETTI paper. Finally, we present some numerical experiments in image processing to demonstrate the efficiency of the method.

Keywords: Monge–Ampère equation; optimal transport; numerical solution; Newton’s method; periodic boundary conditions; image processing.

1. Introduction

The optimal transport (OT) problem, also known as the Monge–Kantorovich problem, originated from a famous engineering problem by Monge (1781) for which Kantorovich (2006) produced a more tractable relaxed formulation. This problem deals with the optimal way of allocating resources from one site to another while keeping the cost of transportation minimal. Formally, if $\mu$ is a probability measure modelling the distribution of material in the source domain $X \subset \mathbb{R}^d$ and $\nu$ is another probability measure modelling the structure of the target domain $Y \subset \mathbb{R}^d$, the Monge–Kantorovich problem consists in finding the OT plan $T$ in

$$\inf_{T: X \rightarrow Y} \left\{ \int_X c(x - T(x)) \, d\mu(x); \ T#\mu = \nu \right\},$$

where $c(x - y)$ denotes the cost of transporting a unit mass of material from a position $x \in X$ to a location $y \in Y$, and $T#\mu = \nu$ means that $\nu(B) = \mu(T^{-1}(B))$ for all Borel sets $Y \subset \mathbb{R}^n$, that is, the quantity of material supplied in a region $B$ of $Y$ coincides with the total amount of material transported from the region $T^{-1}(B)$ of $X$ via the transport plan $T$. When the cost function is quadratic, i.e. $c(x - y) = |x - y|^2/2$, the corresponding OT problem is known as the $L^2$ OT problem. This particular case has attracted many researchers in the past few decades, and a lot of interesting theoretical results have been obtained along with several applications in science and engineering, such as meteorology, fluid
dynamics and mathematical economics. We refer to the recent monographs of Villani (2003, 2009) for an account on these developments. One of the most important results concerns the form of the solution to the $L^2$ OT problem. Brenier (1991) showed that if both the source and target measures are absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^d$, $d\mu/dx = f(x)$, $d\nu/dx = g(x)$, then the $L^2$ OT problem has a unique invertible solution $\tilde{T} (\mu)$ a.e. that is characterized by the gradient of a convex function, $\tilde{T} = \nabla \Psi$. Moreover, if $f$ and $g$ are sufficiently regular (in a sense to be specified later), it is proved that $\Psi$ is of class $C^2$ and satisfies the Monge–Ampère equation

$$g(\nabla \Psi(x)) \det (D^2 \Psi(x)) = f(x)$$

(Delanoë, 1991; Caffarelli, 1992, 1996; Urbas, 1997). Therefore, for smooth source and target probability densities $f$ and $g$, a convex solution $\Psi$ to Monge–Ampère equation (2) provides the optimal solution $\tilde{T} = \nabla \Psi$ to the $L^2$ OT problem.

In this paper, we are interested in the numerical resolution of the $L^2$ OT problem. Concerning this issue, several numerical methods have been developed in the past few years. Some of these methods exploit alternative formulations of the OT problem (1). For example, in Benamou & Brenier (2000), the authors recast the transport problem into a fluid dynamics framework, which eliminates the non-linearity of the constraint but introduces an additional time variable. In Haber et al. (2010), the authors introduce a penalty term to the objective functional (1) which exploits the fact that the optimal solution has to be curl-free, and then solve the resulting constrained optimization problem with an SQP-type algorithm. A gradient flow approach for the computation of the OT map has also been explored in Angenent et al. (2003). Another method presented in Chartrand et al. (2009) uses a gradient descent on the dual of the OT problem to obtain the numerical solution.

As previously mentioned, one can recover the solution of the $L^2$ OT problem by finding a convex solution of Monge–Ampère equation (2). Thanks in part to OT, this equation has lately been the subject of a lot of attention in the scientific computing community. In the case of Dirichlet boundary conditions, a variety of approaches have been explored. Indeed, Dean & Glowinski (2006) introduced two methods to solve the Monge–Ampère equation in dimension 2: one based on a combination of non-linear least-squares and operator-splitting and another based on an augmented Lagrangian resolution of a saddle-point problem. In Benamou et al. (2010), the authors also presented two methods to achieve this, a standard finite-difference discretization and an iterative procedure involving the solution of a Poisson equation. In addition, Feng & Neilan (2011) presented Galerkin methods for approximating viscosity solutions of the Monge–Ampère equation, whereas Brenner et al. (2011) developed $C^0$ penalty methods for the 2D equation. Note that in all the four cases, the methods introduced are only designed to handle uniform target densities.

Another type of boundary condition for the Monge–Ampère equation that arises naturally in the context of optimal transportation is the second boundary condition, also known as OT boundary condition, which states that the optimal map $\tilde{T}$ must satisfy $\tilde{T}(X) = Y$. Some very robust algorithms have been recently designed in Benamou et al. (2012a,b) and Froese (2012) to tackle this problem. These methods rely on clever reformulations of the boundary condition (to either a Hamilton–Jacobi equation on the boundary or a sequence of Neumann boundary conditions), and the usage of a Newton-type algorithm. They successfully recover viscosity solutions of the OT problem for general domains and mass densities.

It is also worth mentioning that different approaches to solve the Monge–Ampère equation have been used in other situations. For example, in the context of grid registration, Sulman et al. (2011) computed the solution as a steady state of a parabolic Monge–Ampère equation and Finn et al. (2008)
used a multigrid preconditioned Newton–Krylov algorithm. The list of numerical methods included above is therefore not exhaustive, and more references can be found within the papers cited.

Our goal in this work is to present an efficient numerical method specifically designed to solve the OT problem in the case where the mass densities are periodic. While some of the methods presented earlier can handle periodic mass densities, none of them has been built specifically to take advantage of the periodicity. If \( f \) and \( g \) are periodic, it has been shown that the optimal map takes the form \( \tilde{T}(x) = x + \nabla u(x) \), where \( u \) is also periodic (Cordero-Erausquin, 1999). This insight has been exploited by Loeper & Rapetti (2005) to design a damped Newton method to solve the Monge–Ampère equation in the particular case where the target density \( g \) is uniform (\( g = 1 \)). They also provided a proof of convergence of their method (before discretization) under some appropriate regularity assumptions on the initial density. However, their assumption on the final density \( g \) is too restrictive and therefore strongly limits the potential applications of their result, from an OT point of view.

Here, we propose to extend their method to the more general case where the final density \( g \) is arbitrary (but sufficiently regular and bounded away from zero). As in Loeper & Rapetti (2005), the main idea is to approximate the fully non-linear partial differential equation (PDE) (2) by a sequence of linear elliptic PDE’s via a damped Newton algorithm. However, we make the algorithm more efficient by presenting several numerical improvements. More precisely, we compare two different discretization methods to solve the linear PDE’s approximating (2). The first one is similar to the one used in Loeper & Rapetti (2005) and consists in a standard second-order finite-difference discretization combined with a biconjugate gradient (BICG) solver. The second one, designed to take advantage of the periodicity, is an analytically preconditioned fast Fourier transform (FFT) implementation combined with a generalized minimal residual (GMRES) solver (Strain, 1994). The FFT algorithm provides what appears to be a globally stable \( O(P \log P) \) method. In addition to this FFT speed up compared with Loeper & Rapetti (2005), we use for both implementations fourth-order centred differences to approximate the first- and second-order derivatives of \( u_n \), the approximation of \( u \), at every step.

We also extend the convergence results of Loeper & Rapetti (2005) to this more general context. To prove this theoretical convergence of the solutions of the linear elliptic PDE’s to the actual convex solution of Monge–Ampère equation (2), we exploit interior \textit{a priori} estimates on the classical solutions to the Monge–Ampère equation. As far as we know, no global estimates that we could use are available for this equation, but as we consider a periodic setting, we can take advantage of the local estimates.

The paper is organized as follows. In Section 2, we present the problem together with some background results that will be used later. In Section 3, we introduce the damped Newton algorithm for the Monge–Ampère equation in the general case of the \( L^2 \) OT problem and discuss its convergence. In Section 4, we propose two different methods for discretizing the linearized Monge–Ampère equation, and then test these implementations on three examples in Section 5. One of these examples is taken from medical imaging, namely, the detection of the multiple sclerosis (MS) disease in a brain magnetic resonance imaging (MRI) scan. Finally, we conclude with some remarks and comparisons with existing algorithms in Section 6.

2. Problem setting

In what follows, \( d \geq 1 \) is an integer, and we denote by \( e_i \) the \( i \)-th-canonical unit vector of \( \mathbb{R}^d \). A function \( \zeta : \mathbb{R}^d \to \mathbb{R} \) is said to be \( 1 \)-periodic if \( \zeta(x + e_i) = \zeta(x) \) for all \( x \in \mathbb{R}^d \) and \( i \in \{1, \ldots, d\} \). Note that for such a function, its values on the subset \( \Omega := [0, 1]^d \) of \( \mathbb{R}^d \) are sufficient to define its entire values on the whole space \( \mathbb{R}^d \). Based on this remark, we will identify in the sequel 1-periodic functions on \( \mathbb{R}^d \) with their restrictions on \( \Omega = [0, 1]^d \). Now, let \( \mu \) and \( \nu \) be two probability measures absolutely continuous
with respect to the Lebesgue measure on \(\mathbb{R}^d\), and assume that their respective densities \(f\) and \(g\) are 1-periodic. Then, \(f, g : \Omega \to \mathbb{R}\), and the \(L^2\) OT problem with these densities reads as

\[
\inf_{T : \Omega \to \Omega} \left\{ \int_{\Omega} |x - T(x)|^2 \, dx \mid T_\# \mu = \nu, \ D \mu(x) = f(x) \, dx, \ D \nu(x) = g(x) \, dx \right\}.
\] (3)

Moreover, the unique solution \(\tilde{T} = \nabla \Psi\) to this problem (where \(\Psi : \Omega \to \mathbb{R}\) is convex) satisfies Monge–Ampère equation (2) on \(\Omega\). The regularity and boundary conditions corresponding to this Monge–Ampère equation are given by the following theorem due to Cordero-Erausquin (1999).

**Theorem 2.1** Assume that \(\mu, \nu, f\) and \(g\) are defined as above and let \(m_f, m_g, M_f\) and \(M_g\) denote the infima and suprema of \(f\) and \(g\), respectively. Then there exists a convex function \(\Psi : \Omega \to \Omega\) that pushes \(\mu\) forward to \(\nu\) (i.e. \((\nabla \Psi)^\# \mu = \nu\)) such that \(\nabla \Psi\) is additive in the sense that \(\nabla \Psi(x + p) = \nabla \Psi(x) + p\) for almost every \(x \in \mathbb{R}^d\) and for all \(p \in \mathbb{Z}^d\). Moreover, \(\nabla \Psi\) is unique and invertible (\(\mu\) a.e.), and its inverse \(\nabla \Psi\) satisfies \((\nabla \Psi)^\# \nu = \mu\). In addition, if \(f\) and \(g\) are of class \(C^\alpha(\Omega)\) with \(\alpha > 0\) and if \(m_g, M_f > 0\), then \(\Psi \in C^{2,\beta}(\Omega)\) for some \(0 < \beta < \alpha\) and it is a convex solution of Monge–Ampère equation (2).

Note that since \(\nabla \Psi\) is additive, it can be written as \(\nabla \Psi(x) = x + \nabla u(x)\), where \(u\) is a 1-periodic function. Thus, we assume \(\Psi(x) = |x|^2/2 + u(x)\) with \(\nabla u(x + p) = \nabla u(x)\) for all \(p \in \mathbb{Z}^d\). So by using this change of function, \(\Psi(x) = |x|^2/2 + u(x)\), in Monge–Ampère equation (2), we see that the corresponding equation in \(u\) satisfies a periodic boundary condition on \(\Omega\). This justifies the introduction of this change of function in Section 3 to rewrite (2). In fact, the periodic boundary conditions will allow us to use interior a priori estimates for classical solutions of the Monge–Ampère equation on the whole domain in order to prove the convergence of our algorithm (see Section 3.2). We also infer from this theorem that if \(f, g \in C^\alpha(\Omega)\), then \(\Psi \in C^{2,\beta}(\Omega)\) is the unique (up to a constant) convex solution of Monge–Ampère equation (2) on \(\Omega\). Finally, classical bootstraping arguments from the theory of elliptic regularity can be used to prove that if \(f, g \in C^{k,\alpha}(\Omega)\), then \(\Psi \in C^{k+2,\beta}(\Omega)\).

3. The damped Newton algorithm

3.1 Derivation of the algorithm

Loeper & Rapetti (2005) presented a numerical method based on Newton’s algorithm to solve the equation

\[ \det(D^2 \Psi) = f(x), \]

in a periodic setting. This equation can be associated with the OT problem in the case where the target measure \(\nu\) has a uniform density, i.e. \(g = 1\). Here, we propose to extend this algorithm and the underlying analysis to the general case of an arbitrary smooth 1-periodic density \(g\). Motivated by the remark made in Section 2, we follow Loeper & Rapetti (2005) and introduce the change of function \(\Psi(x) = |x|^2/2 + u(x)\) to rewrite Monge–Ampère equation (2) in the equivalent form

\[ M(u) := g(x + \nabla u(x)) \det(I + D^2 u(x)) = f(x), \] (4)

where \(I\) denotes the \(d \times d\) identity matrix. Therefore, we will solve (4) for a 1-periodic solution \(u\) such that \(|x|^2/2 + u\) is convex on \(\Omega = [0, 1]^d\). Since we want to develop an algorithm based on Newton’s method, we first linearize (4). Using the formula for the derivative of the determinant (Loeper & Rapetti,
Numerical Algorithm for $L^2$ Optimal Transport, we have
\[
\det(I + D^2(u + s\theta)) = \det(I + D^2u) + s \text{Tr}(\text{Adj}(I + D^2u)D^2\theta) + O(s^2),
\]
where $\text{Adj}(A) = \det(A) \cdot A^{-1}$. Also, from the usual Taylor expansion, we have
\[
g(x + \nabla(u + s\theta)) = g(x + \nabla u) + s \nabla g(x + \nabla u) \cdot \nabla \theta + O(s^2).
\]
Multiplying the latter two expressions, we obtain that the derivative in direction $\theta$ of the right-hand side of (4), denoted by $D_u M \cdot \theta$, is given by
\[
D_u M \cdot \theta := g(x + \nabla u) \text{Tr}(\text{Adj}(I + D^2u)D^2\theta) + \det(I + D^2u)\nabla g(x + \nabla u) \cdot \nabla \theta.
\]
For simplicity, we will denote from now on the linear Monge–Ampère operator by $L(u_n)$ (so that $L(u_n)\theta_n := D_u M \cdot \theta_n$). With this linearization at hand, we can now present the damped Newton algorithm that we will use to solve (4).

\[\text{Damped Newton algorithm}\]

\[
\begin{align*}
\text{With } u_0 \text{ given, loop over } n \in \mathbb{N}, \\
\text{Compute } f_n &= g(x + \nabla u_n) \det(I + D^2u_n), \\
\text{Solve the linearized Monge–Ampère equation } L(u_n)\theta_n &= \frac{1}{\tau}(f - f_n), \\
\text{Update the solution : } u_{n+1} &= u_n + \theta_n.
\end{align*}
\]

The factor $1/\tau$ ($\tau \geq 1$) in the algorithm is used as a step-size parameter to help preventing the method from diverging by taking a step that goes ‘too far’. As we will see below, the value of $\tau$ is crucial for the proof of convergence of the algorithm. Indeed, we will show that if we start with a constant initial guess for the Newton method, then there is a $\tau$ such that the method converges (under some additional conditions on the densities). Furthermore, by modifying some results presented in Gilbarg & Trudinger (2001), it is possible to prove that a second-order linear strictly elliptic PDE with periodic boundary conditions has a unique solution up to a constant if its zeroth-order coefficient is 0. The linearized Monge–Ampère equation at step $n$ falls into this category through the setting of the algorithm. To fix the value of that constant, we select the solution satisfying $\int_{\Omega} u \, dx = 0$. This is guaranteed by choosing a $\theta_n$ that satisfies this condition for every $n$. In addition, note that $f_n$ remains a probability density for every $n$ (see (10)). However, due to numerical errors, this property might be lost after discretization. We thus introduce a renormalization step after the computation of $f_n$ in (6). More details will be presented in Section 4.

3.2 Proof of convergence

To prove the (theoretical) convergence of our algorithm (6), we follow the arguments in Loeper & Rapetti (2005), but we introduce several new arguments to deal with the non-uniform final density $g$. In particular, we rely on three a priori estimates for the solution of the Monge–Ampère equation. The first one is derived by Liu et al. (2010) and goes as follows: if $\lambda \leq f/g \leq \Lambda$ for some positive constants
Table 1 Quantities involved in the bounds on \(1/(1+k)\) for \(C = \Lambda/\lambda\)

<table>
<thead>
<tr>
<th>(d)</th>
<th>(w_d)</th>
<th>Rounded (K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>64C</td>
</tr>
<tr>
<td>2</td>
<td>(\pi)</td>
<td>4550C</td>
</tr>
<tr>
<td>3</td>
<td>(4\pi/3)</td>
<td>140039C</td>
</tr>
<tr>
<td>4</td>
<td>(\pi^2/2)</td>
<td>2709370C</td>
</tr>
</tbody>
</table>

\(\lambda, \Lambda,\) and if \(f \in C^\alpha(\Omega)\) for some \(\alpha \in (0, 1)\), then a convex solution of (2) satisfies

\[
\|\Psi\|_{C^{2,\alpha}(\Omega_{r_1})} \leq R_1 \left[ 1 + \frac{1}{\alpha(1-\alpha)} \left\| \frac{f(x)}{|g(\nabla \Psi(x))|} \right\|_{C^\alpha(\Omega)} \right],
\]

(7)

where \(\Omega_{r_1} = \{ x \in \Omega : \text{dist}(x, \partial \Omega) > r_1 \}\) and \(R_1\) is a constant that depends only on \(d, r_1, \lambda, \Lambda\) and \(\Omega\). The second one, discovered by Caffarelli (Villani, 2009) and expressed by Forzani & Maldonado (2004), presents a bound on the Hölder component of \(\nabla \Psi\). It states that if \(f\) and \(g\) are as in the previous estimate, there exists some constant \(k\) such that

\[
\frac{|\nabla \Psi(z) - \nabla \Psi(y)|}{|z - y|^{1/(1+k)}} \leq R_y \left( \frac{K}{m(\psi_y^*, 0, R_y)} \right)^{1/(1+k)} \left( \frac{M(\Psi, y, r_2)}{r_2} \right)^{1/(1+k)},
\]

(8)

for \(|z - y| \leq r_2\), where \(\psi_y(x) = \Psi(x + y) - \Psi(y) - \nabla \Psi(y) \cdot x\),

\[R_y = \max \left\{ 1, \left( \frac{KM(\Psi, y, r_2)}{m(\psi_y^*, 0, 1)} \right)^{k/(1+k)} \right\},\]

and \(M(\Psi, y, r_2), m(\Psi, y, r_2)\) denote, respectively, the maximum and minimum of \(\psi_y(z - y)\) taken over the points \(z\) such that \(|z - y| = r_2\). Since this estimate does not hold for all \(k\), one might wonder for which values it is actually valid. In Forzani & Maldonado (2004), it is shown that it holds for

\[k = 2K(K + 1), \quad K = \frac{2^{3d+2}w_dw_{d-1}\Lambda}{d^{-3/2}\lambda} \quad \text{and} \quad w_m = \frac{\pi^{m/2}}{\Gamma(m/2 + 1)}.
\]

Here \(w_m\) is the volume of the \(m\)-dimensional unit ball. To give the reader an idea of these values, a few of them are presented in Table 1.

Finally, the third estimate controls the growth of the second derivatives of \(\Psi\) with respect to boundary values, provided \(f/g \in C^2(\Omega)\) and \(\Psi \in C^4(\Omega)\) (Gilbarg & Trudinger, 2001; Trudinger & Wang, 2006):

\[
\sup_{\Omega} |D^2 \Psi| \leq R_3 \left( 1 + \sup_{\partial \Omega} |D^2 \Psi| \right),
\]

(9)

where \(R_3\) depends only on \(\Omega, d, f/g\) and on the sup of \(\Psi + \nabla \Psi\) in \(\Omega\). We can now state and prove the theorem on the convergence of algorithm (6). Note that the arguments of the proof are similar to the ones in Loeper & Rapetti (2005), but the fact that the target density \(g\) is now non-uniform introduces some
new difficulties that are worthwhile exposing. In addition, the proof provides important information that can be used to gain some intuition about the performance of the algorithm in practice.

**Theorem 3.1** Assume that $\Omega = [0, 1]^d$ and let $f, g$ be two positive 1-periodic probability densities bounded away from 0. Assume that the initial guess $u_0$ for Newton algorithm (6) is constant. If $f \in C^{2, \alpha}(\Omega)$ and $g \in C^{3, \alpha}(\Omega)$ for any $0 < \alpha < 1$, then there exists $\tau \geq 1$ such that $(u_n)$ converges in $C^{4, \beta}(\Omega)$, for any $0 < \beta < \alpha$, to the unique—up to a constant—solution $u$ of Monge–Ampère equation (4). Moreover, $\tau$ depends only on $\alpha, d, \|f\|_{C^{2, \alpha}(\Omega)}, \|g\|_{C^{3, \alpha}(\Omega)}$ and $M_f, M_g, m_f, m_g$, which are defined as in Theorem 2.1.

**Proof.** First we note that due to the additivity of the transport map, by applying the change of variable $y = \tilde{T}_n(x) = x + \nabla u_n(x)$, we can prove that for all $n$,

$$\int_{\tilde{T}_n(\Omega)} g(y) \, dy = \int_{\Omega} g(\tilde{T}_n(x)) \det(D\tilde{T}_n(x)) \, dx = \int_{\Omega} f_n \, dx = 1,$$

i.e. at every step, we are solving the OT problem sending $f_n$ to $g$. Moreover, unless otherwise stated, we only need to assume that $f \in C^{\alpha}(\Omega)$ and $g \in C^{2, \alpha}(\Omega)$. The main steps of the proof consist in showing by induction that the following claims hold for all $n$:

1. $I + D^2 u_n$ and $\text{Adj}(I + D^2 u_n)$ are $C^{\alpha}(\Omega)$ smooth, uniformly positive-definite (u.p.d.) matrices.
2. $f / C_1 \leq f_n \leq C_1 f$, where $C_1$ is independent of $n$.
3. $\|f - f_n\|_{C^i(\Omega)} \leq C_2$, where $C_2$ is independent of $n$.

We say that a matrix $A$ is $C^{\alpha}(\Omega)$ smooth if all of its coefficients are in $C^{\alpha}(\Omega)$ and that it is u.p.d. if there exists a constant $k > 0$ such that $\xi^T A \xi \geq k|\xi|^2$ for all $\xi \in \Omega$. It is also worth mentioning that the statement in (1) actually implies that $\Psi_n = |x|^2 / 2 + u_n$ is uniformly convex and that $L(u_n)$ is a strictly elliptic linear operator.

Note that for $u_0$ constant, we have $f_0 = g$. Next, let

$$C_1 = \max\left(\frac{M_f}{m_g}, \frac{M_g}{m_f}\right) \quad \text{and} \quad C_2 = \|f\|_{C^i(\Omega)} + \|g\|_{C^i(\Omega)}.$$  

Then, it is easy to see that all the claims (1), (2) and (3) hold for $n = 0$. Let us assume that they hold for a certain $n \in \mathbb{N}$ and prove them for $n + 1$. For now, we suppose that the step-size parameter could vary with $n$. We shall prove later that we can actually take it to be constant without affecting any result. Let $\theta_n$ be the unique solution of $L(u_n)\theta_n = (f - f_n)/\tau$ such that $\int_{\Omega} \theta_n \, dx = 0$. According to the results of Gilbarg & Trudinger (2001) (modified for the periodic case), there exists a constant $k_{\theta_n}$ such that

$$\|\theta_n\|_{C^\alpha(\Omega)} \leq \frac{k_{\theta_n}}{\tau} \|f - f_n\|_{C^\alpha(\Omega)} \leq \frac{k_{\theta_n} C_2}{\tau}, \quad i = 1, 2.$$
Because \( u_{n+1} = u_n + \theta_n \), we deduce that \( I + D^2 u_{n+1} \) and then \( \text{Adj}(I + D^2 u_{n+1}) \) are \( C^a(\Omega) \) smooth. Now, since \( I + D^2 u_n \) is u.p.d., by assumption we get:

\[
\xi^\top (I + D^2 u_n) \xi = \xi^\top (I + D^2 u_n) \xi + \xi^\top (D^2 \theta_n) \xi \\
\geq K_1 |\xi|^2 - \frac{k_0}{\tau} \|f - f_n\|_{C^a(\Omega)} \sum_{j=1}^d \xi_j \xi_j \\
\geq K_1 |\xi|^2 - \frac{k_0}{2\tau} \|f - f_n\|_{C^a(\Omega)} \sum_{j=1}^d (\xi_j^2 + \xi_j^2) \\
= \left[ K_1 - \frac{k_0 d}{\tau} \|f - f_n\|_{C^a(\Omega)} \right] |\xi|^2 \\
\geq K_2 |\xi|^2,
\]

for \( \tau \) large enough, where \( K_2 \) is a positive constant. Hence, \( I + D^2 u_{n+1} \) is a u.p.d. matrix. Next, inspired by the Taylor expansions previously shown, we write \( f_{n+1} \) in terms of \( f_n \) as follows:

\[
f_{n+1} = g(x + \nabla u_{n+1}) \det(I + D^2 u_{n+1}) \\
= g(x + \nabla u_n) \det(I + D^2 u_n) + L_n \theta_n + r_n \\
= f_n + \frac{f - f_n}{\tau} + r_n. \tag{12}
\]

Now we bound the residual \( r_n \). It is easy to see that an explicit formula for \( r_n \) can be obtained from the second-order terms of the Taylor expansion of the Monge–Ampère operator, and it consists of a sum of products of at least two first or second derivatives of \( \theta_n \) with \( g \) and its derivatives evaluated at \( \nabla \Psi_n \) and with second derivatives of \( \Psi_n \). By (11), we know that we can bound the \( C^a(\Omega) \) norm of the second derivatives of \( \theta_n \) by a constant times \( \|f - f_n\|_{C^a(\Omega)}/\tau \). In addition, since \( g \in C^{2,a}(\Omega) \), the Hölder norm of \( g \) and its first and second derivatives are all uniformly bounded. We then deduce that

\[
\|r_n\|_{C^a(\Omega)} \leq \frac{k_{r_n}}{\tau^2} \|f - f_n\|_{C^a(\Omega)}^2, \tag{13}
\]

where \( k_{r_n} \) could potentially depend on the Hölder norms of the first and second derivatives of \( \Psi_n \). Next, by selecting \( \tau \geq 2k_{r_n} C_2 \), (12), (13) and (3) imply

\[
\|f - f_{n+1}\|_{C^a(\Omega)} \leq \left( 1 - \frac{1}{\tau} \right) \|f - f_n\|_{C^a(\Omega)} + \frac{k_{r_n}}{\tau^2} \|f - f_n\|_{C^a(\Omega)}^2 \\
\leq \|f - f_n\|_{C^a(\Omega)} \left( 1 - \frac{1}{\tau} + \frac{k_{r_n} C_2}{\tau^2} \right) \\
= \left( 1 - \frac{1}{2\tau} \right) \|f - f_n\|_{C^a(\Omega)}.
\]

This shows that bound (3) is preserved for \( \tau \) large enough. In addition, it shows that we can take the step-size \( \tau \) such that the sequence of bounds \( K_2 \) created recursively will converge to a constant strictly
greater than 0. Let us now verify bound (2). If we take \( \tau \geq k_r C_2^2/|m_f(1 - 1/C_1)| \), from all the previous results and hypothesis we get

\[
f - f_{n+1} \leq \frac{\tau - 1}{\tau} (f - f_n) + \frac{k_r}{\tau^2} \|f - f_n\|^2_{C^2(\Omega)}
\]

\[
\leq \frac{\tau - 1}{\tau} f \left( 1 - \frac{1}{C_1} \right) + \frac{k_r C_2^2}{\tau^2}
\]

\[
= f \left( 1 - \frac{1}{C_1} \right),
\]

from which we deduce that \( f/C_1 \leq f_{n+1} \). Following a similar approach with a step-size \( \tau \geq k_r C_2^2/|m_f(C_1 - 1)| \), we obtain the other part of (2). Then, we go back and finish the proof of the first statement. Knowing that \( I + D^2u_{n+1} \) is u.p.d., we see that \( \det(I + D^2u_{n+1}) > 0 \) and therefore \( I + D^2u_{n+1} \) is invertible. We can prove that its inverse is also a u.p.d. matrix. Indeed, if \( \xi = (I + D^2u_{n+1})y \in \Omega \), we have

\[
\xi^\top (I + D^2u_{n+1})^{-1} \xi = [(I + D^2u_{n+1})y]^\top (I + D^2u_{n+1})^{-1}((I + D^2u_{n+1})y)]
\]

\[
= y^\top (I + D^2u_{n+1})y \geq K_3 |y|^2 = K_3 |(I + D^2u_{n+1})^{-1} \xi|^2.
\]

Using the inequality \( |AB| \leq |A| |B| \) with \( A = I + D^2u_{n+1} \) and \( B = (I + D^2u_{n+1})^{-1} \xi \), we obtain \( |\xi| \leq |I + D^2u_{n+1}| |(I + D^2u_{n+1})^{-1} \xi| \). Next, motivated by the equivalence of norms, we use the bounds we derived previously to get

\[
|I + D^2u_{n+1}| \leq d \max_{i,j} \{ |(I + D^2u_{n+1})_{ij}| \}
\]

\[
\leq d (1 + \|u_n\|_{C^2(\Omega)} + \|\theta_n\|_{C^2(\Omega)}) \leq K_4,
\]

where \( K_4 \) is a positive constant. This yields the claim:

\[
\xi^\top (I + D^2u_{n+1})^{-1} \xi \geq \frac{K_3 |\xi|^2}{|I + D^2u_{n+1}|^2} \geq \frac{K_3}{K_4} |\xi|^2 = K_5 |\xi|^2, \quad K_5 > 0.
\]

We now use these statements to show that \( L_{n+1} \) is a strictly elliptic operator,

\[
g(x + \nabla u_{n+1}) \sum_{i,j=1}^d \text{Adj}(I + D^2u_{n+1})_{ij} \xi_i \xi_j \geq f_{n+1} K_3 |\xi|^2
\]

\[
\geq \frac{f}{C_1} K_3 |\xi|^2 \geq K_6 |\xi|^2.
\]

Note that by removing \( g \) from the previous inequalities, we get that \( \text{Adj}(I + D^2u_{n+1}) \) is a u.p.d. matrix, which completes the proof of (1). Now, we show that the step-size \( \tau \) can be taken constant, as claimed before. Indeed, (1) gives \( \Psi_n \in C^{2,d}(\Omega) \) by construction while (2) yields \( f_n \in C^d(\Omega) \) and

\[
0 < \frac{m_f}{C_1 M_g} \leq \frac{f_n(x)}{g(x + \nabla u_n)} \leq \frac{C_1 M_f}{m_g}.
\]
Therefore, all the conditions to estimate (7) are satisfied at every step. Using inequalities on Hölder norms, we find

\[ \left\| \frac{f_n}{g(\nabla \Psi_n)} \right\|_{C^\alpha(\Omega)} \leq \left\| f_n \right\|_{C^\alpha(\Omega)} \left( \frac{1}{g} \right) \left( 1 + \| \nabla \Psi_n \|_{C^{\alpha}(\Omega)} \right). \]

At this point, the only remaining challenge is to bound \( \| \nabla \Psi_n \|_{C^{\alpha}(\Omega)} \). It can be achieved through the second estimate (8). Since \( \nabla \Psi_n \) is the transport map moving \( f_n \) to \( g \), we can refer to Theorem 2.1 to deduce that \( \nabla \Psi_n \) is invertible and thus \( \nabla \Psi_n(\Omega) \subseteq \Omega \), which in turn yields \( \Psi_n(\Omega) \subseteq [0, \sqrt{d}] \) when \( \Omega = [0, 1]^d \). Therefore, we see that the maximum terms \( M(\Psi, y, r_2) \) are going to be uniformly bounded and that the only problem could come from the minimum terms \( m(\psi_n^*, 0, a), a = 1 \) or \( r_1 \). Using ideas from convex analysis (Hiriart-Urruty & Lemaréchal, 1996), we can show that since \( \Psi_n \) is uniformly convex for every \( n \), we have \( m(\psi_n^*, 0, a) = \min \psi_n^*(z) \) where the minimum is taken on the sphere \( |z| = a, a \geq 1 \) (with the periodicity we can increase the size of \( \Omega \) to include it inside and still have a uniform bound on \( \Psi_n \) and \( \nabla \Psi_n \)). Furthermore, \( \nabla \psi_n^*(z) = 0 \) if and only if \( z = 0, \nabla \psi_n^* \) is strictly monotonically increasing because \( \nabla \psi_n \) is and \( \nabla \psi_n^{-1} = \nabla \psi_n^* \). We see that the only possible breakdown happens when \( \nabla \psi_n^* \) converges to a function that is zero up to \( |z| = a \). This means \( |\nabla \psi_n| = |\nabla \psi_n(x + y) - \nabla \psi_n(y)| \to \infty \) as \( |x| \to 0 \) and \( n \to \infty \), for any \( y \). Observe now that if we increase the regularity of the densities to \( f \in C^{2,\alpha}(\Omega), g \in C^{3,\alpha}(\Omega) \), we get \( f_n \in C^{2,\alpha}(\Omega) \) at every step. This tells us that \( \theta_n \in C^{4,\alpha}(\Omega) \) (Gilbarg & Trudinger, 2001) and thus \( \Psi_n \in C^{4,\alpha}(\Omega) \). Therefore, we can apply estimate (9) and rule out this potential breakdown case. We obtain that the \( C^{2,\alpha}(\Omega) \) norm of \( \psi_n \) is uniformly bounded and thus by the additivity of that function in a periodic setting, the same conclusion holds for its \( C^{2,\alpha}(\Omega) \) norm. Hence, we deduce that it is also the case for \( k_n \) and then \( k_0 \). From this, we get that we can select a \( \tau \geq 1 \) constant such that the three statements hold for all \( n \in \mathbb{N} \) by induction. Moreover, the sequence \( (u_n)_{n \in \mathbb{N}} \) is uniformly bounded in \( C^{2,\alpha}(\Omega) \), and thus equicontinuous. By the Ascoli–Arzela theorem, it converges uniformly in \( C^{2,\beta}(\Omega) \) for \( 0 < \beta < \alpha \) to the solution \( u \) of (4), which is unique since we impose \( \int_{\Omega} \psi \, dx = 0 \). Finally, due to the fact that the initial and final densities are actually \( C^{2,\alpha}(\Omega) \), we know that this solution will be in \( C^{4,\beta}(\Omega) \).

\[ \square \]

3.3 Remarks on the proof

This proof by induction provides precious information concerning the properties of the iterates created by our method. First, since \( I + D^2 u_n \) is u.p.d. at every step, we realize that the sequence of functions \( \Psi_n \) is actually one of uniformly convex functions. Recall that Monge–Ampère equation (2) is elliptic only when we restrict it to the space of convex functions. Therefore, the algorithm enforces this by approximating the convex solution of the Monge–Ampère equation by a sequence of uniformly convex functions. In addition, this guarantees that the linearized equation is strictly elliptic and thus has a unique solution (once we fix the constant). Furthermore, as in Loeper & Rapetti (2005), we can obtain estimates on the speed of convergence of the method. Indeed, assuming that \( \tau \geq 2 k_{\text{res}} C_2 \), we got

\[ \| f - f_{n+1} \|_{C^\alpha(\Omega)} \leq \left( 1 - \frac{1}{2\tau} \right) \| f - f_n \|_{C^\alpha(\Omega)}, \]

which tells us that \( (f_n) \) converges to \( f \), following a geometric convergence with a rate of at least \( 1 - 1/2\tau \). When it comes to the step-size parameter \( \tau \), it would be very useful to know \( a \text{ priori} \) which value to select in order to make the algorithm converge. Such an estimate is unfortunately hard to acquire since some of the constants used through interior bounds are obtained via rather indirect arguments.
However, we observe from lower bounds on $\tau$ used in the proof, i.e.

$$\tau \geq \frac{k_{\text{res}}C_2^2}{m_f(1 - 1/C_1)}, \quad \tau \geq \frac{k_{\text{res}}C_2^2}{(C_1 - 1)m_f} \quad \text{or} \quad \tau \geq 2k_{\text{res}}C_2,$$

that the minimum value required on the step-size parameter to achieve convergence could potentially be large when $m_f$ is close to 0 or when either $\|f\|_{C^\alpha(\Omega)}$ or $\|g\|_{C^\alpha(\Omega)}$ is large. Through the numerous numerical experiments we conducted, we realized that $\tau$ seems to behave according to both conditions. Therefore, knowing a priori that $f$ could get close to 0, we can react accordingly by either increasing the value of the step-size parameter or by modifying the representation of the densities (which is possible in some applications). Finally, even if our proof only guarantees convergence when the update $\theta_n$ is the solution of (5), in practice we can get good results by replacing it by the solution of $g(x + \nabla u_n) \text{Tr}(\text{Adj}(\mathcal{I} + D^2 u_n)D^2\theta_n) = (f - f_n)/\tau$, or sometimes by an even simpler equation.

4. Numerical discretization

We present here a 2D implementation of the Newton algorithm (6).1 We consider a uniform $N \times N$ grid with a space-step $h = 1/N$ where we identify $x_i = 0$ with $x_i = 1$ ($i = 1, 2$) by the periodicity. We employ fourth-order centred finite differences for the discretization of the first and second derivatives of $u_n$. If we have an explicit formula for the target density $g$, then we can compute the compositions $g(x + \nabla u_n)$ and $\nabla g(x + \nabla u_n)$ directly. However, it is not always the case, especially when we deal with discrete data as in the examples of image processing in Section 5. In such circumstances, our experiments have shown us that using a closest neighbour interpolation to approximate the value of $x + \nabla u_n(x)$ is sufficient to obtain good results.

It is easy to see that the most important step for the efficiency of the method is the resolution of the linearized Monge–Ampère equation

$$L(u_n)\theta_n = \frac{(f - f_n)}{\tau}, \quad (14)$$

where $L(u_n)$ is defined as in (5). Indeed if we take $P$ to be the number of points on the grid ($P = N^2$ in 2D), as every other step can be done in $O(P)$ operations, the computational complexity of the whole method is dictated by the resolution of this linear PDE. Therefore, we will introduce below two methods for solving this equation for $\theta_n$. Note that for both of these methods, the same fourth-order accurate finite differences are used for the derivatives of $u_n$, the difference being the way $\theta_n$ is approximated. We will refer to the first method, which consists in discretizing (14) by using standard second-order accurate finite differences and in solving the resulting linear system with the BICG algorithm, as the finite-difference implementation. The other method, which uses instead a combination of the FFT and GMRES algorithm, will be referred to as the Fourier transforms implementation. Note that by using fourth-order accurate finite differences for $u_n$ in both cases, we improve considerably the accuracy of the results compared with Loeper & Rapetti (2005), where second-order differences are used to approximate these terms, but at the same time we do not decrease the efficiency of the whole algorithm whose complexity is dominated by the resolution of the linear PDE.

As pointed out in Section 3.1, another salient point is that even though in theory $f_n$ has a total mass of 1 at every step, it is not necessarily the case in the numerical experiments, due to discretization errors. However, we need the right-hand side of the linearized Monge–Ampère equation to integrate to 0 on

---

1 The MATLAB code can be obtained from the authors upon request.
the whole domain. To deal with this, we introduce a normalization step right after computing \( f_n \) in the implemented algorithm, taking

\[
\tilde{f}_n = f_n - \frac{1}{N^2} \sum_{i,j=0}^{N-1} f_{n,i,j} + 1,
\]

instead of \( f_n \) and thus translating it at every step.

### 4.1 A finite-difference implementation for approximating \( \theta_n \)

We begin by presenting an implementation of the resolution of linearized Monge–Ampère equation (14) through standard finite differences to approximate \( \theta_n \). This choice is motivated by the fact that it is the method chosen by Loeper & Rapetti (2005) for their corresponding algorithm. In this case, we use centred finite differences of second-order for the derivatives of \( \theta_n \). Since the linear PDE has a unique solution only up to a constant, the linear system \( Ax = b \) corresponding to its discretization has one free parameter that we need to fix to turn the matrix into an invertible one. A possible strategy to achieve this is to create a new system \( \hat{A}x = \hat{b} \) by adding the extra equation \( \sum x_i = 0 \), which corresponds to selecting \( \theta_n \) such that \( \int_\Omega \theta_n dx = 0 \). Note that this new matrix has full rank, but it is not square. We take that extra line, add it to all the other lines of \( \hat{A} \) and then delete it to get a square system, \( \tilde{A}x = b \). The next lemma shows conditions under which the resolution of \( \tilde{A}x = b \) will produce a valid solution for the system \( Ax = b \). For the sake of notation, consider the new equation to be stored in the first line of \( \hat{A} \).

**Lemma 4.1** Let \( \hat{A} \) and \( \tilde{A} \) be as defined above, i.e. \( \hat{A} \) is a \((P + 1) \times P\) matrix with rank \( P = N^2 \), and there exist real numbers \( \alpha_1, \alpha_2, \ldots, \alpha_{P+1} \) not all zero such that

\[\alpha_1 L_1 + \alpha_2 L_2 + \cdots + \alpha_{P+1} L_{P+1} = 0, \]

where \( L_i \) is the \( i \)th line of \( \hat{A} \). If \( \alpha_2 + \cdots + \alpha_{P+1} \neq \alpha_1 \), then \( \tilde{A} \) has rank \( P \).

The proof is a straightforward use of matrix algebra and therefore not reported here for brevity. This lemma does not hold if the condition \( \alpha_2 + \cdots + \alpha_{P+1} \neq \alpha_1 \) is not satisfied. Take, for example, a matrix \( \hat{A} \) such that its second line is equal to the negative of its first line and all its other lines are linearly independent. Then \( \hat{A} \) has rank \( P - 1 \). Nonetheless, due to the structure of our problem, this is not going to happen. Unfortunately, this strategy has the downside of somewhat destroying the sparsity of the matrix. One way to avoid this would be to equivalently fix only the value of \( \theta \) at a given point and then use the same strategy. This would preserve most of the sparsity of the matrix.

Next, to actually solve the system \( Ax = b \), we employ the BICG iterative method. This choice can be justified by the fact that we are dealing with a (potentially sparse) matrix that is not symmetric nor positive definite, the BICG procedure being specifically designed to deal with these conditions (Saad, 2003). One feature of this method is that, provided the method does not break down before, the sequence of approximate solutions it produces is guaranteed to converge to the solution of the linear system in a maximum of \( P \) steps, which yields a computational complexity of at worst \( O(P^2) \). However, as we shall see later, in practice it can be much smaller than that.

### 4.2 A fourier transforms implementation for approximating \( \theta_n \)

One should realize that given our boundary conditions, the first implementation we employed to solve linearized Monge–Ampère equation (14) for \( \theta_n \) might not be the best method. Indeed, there exist much cheaper ways to solve a linear second-order strictly elliptic equation with such boundary conditions. The one we are going to explore here is due to Strain (1994) and requires only \( O(P \log P) \) operations...
through the use of the FFT algorithm. It consists in rewriting the problem as the system

\[
\begin{align*}
L(u_n)\bar{L}(u_n)^{-1}\sigma(x) &= h(x), \\
\bar{L}(u_n)\theta(x) &= \sigma(x),
\end{align*}
\]  

(16)

where \(\bar{L}(u_n)\) is the averaged \(L(u_n)\) in the sense that its coefficients are the integral over \(\Omega\) of the coefficients of \(L(u_n)\). We then expand \(\sigma\) in Fourier series by taking

\[
\sigma(x) = \sum_{k \neq 0 \in \mathbb{Z}^d} \hat{\sigma}(k) e^{2\pi i k \cdot x} \quad \text{and} \quad \hat{\sigma}(k) = \int_{\Omega} \sigma(x) e^{-2\pi i k \cdot x} \, dx,
\]

where \(i\) represents \(\sqrt{-1}\) and \(\hat{\sigma}(k)\) is the usual Fourier coefficient. Using this expansion in the first part of (16) yields the formula

\[
L(u_n)\bar{L}(u_n)^{-1}\sigma(x) = \sum_{i,j=1}^{d} a_{ij}(x) \sum_{k \neq 0 \in \mathbb{Z}^d} 2\pi i k_i 2\pi i k_j \hat{\rho}(k) \hat{\sigma}(k) e^{2\pi i k \cdot x}
\]

\[
+ \sum_{i=1}^{d} b_i(x) \sum_{k \neq 0 \in \mathbb{Z}^d} 2\pi i k_i \hat{\rho}(k) \hat{\sigma}(k) e^{2\pi i k \cdot x}
\]

\[
= \sum_{i,j=1}^{d} a_{ij}(x)\alpha_{ij}(x) + \sum_{i=1}^{d} b_i(x)\beta_i(x),
\]

where

\[
\hat{\rho}(k) = \begin{cases} 
1 & \text{if the sum is not 0,} \\
0 & \text{otherwise.} 
\end{cases}
\]  

(17)

For the discretized problem, knowing the value of \(\sigma\), we can compute \(\hat{\sigma}\) with one application of the FFT algorithm and then compute \(\alpha_{ij}\) and \(\beta_i\) with \(d(d + 1)\) applications of the inverse FFT algorithm to be able to get the value of \(L(u_n)\bar{L}(u_n)^{-1}\sigma(x)\) in \(\mathcal{O}(P \log P)\) operations. Therefore, we can use an iterative method to solve the first equation of system (16) at a cost of \(\mathcal{O}(P \log P)\) operations per iteration. As in Strain (1994), we use the GMRES method. Just like BICG, it is an efficient way of solving a linear system of equations where the matrix is non-symmetric and non-positive definite (Saad, 2003). Moreover, GMRES does not use projections on the Krylov subspace generated by the transposed matrix. This makes it easier to code for the particular setting we are dealing with since we do not form \(A\) directly; we reference it instead through the result of its product with a given vector \(\sigma\). Strain observed that the number of GMRES iterations required did not vary with \(P\), which yields a global complexity of \(\mathcal{O}(P \log P)\). Note that for better performances, we actually employ like the author the restarted GMRES(m) method. After computing \(\sigma(x)\), we need to solve \(\bar{L}\theta(x) = \sigma(x)\). This can be easily achieved since we already know the value of \(\bar{L}(u_n)^{-1}\sigma(x)\). More specifically, we have

\[
\theta(x) = \sum_{k \neq 0 \in \mathbb{Z}^d} \hat{\rho}(k)\hat{\sigma}(k) e^{2\pi i k \cdot x},
\]
i.e. it requires only one other application of the (inverse) FFT algorithm to obtain \( \theta \), which is taken as the update \( \theta_n \). One of the advantages of this method is that it is spectrally accurate, i.e. the error decreases faster than any power of the grid size as the space-step size goes to 0. We can also prove that the convergence rate for the GMRES algorithm is independent of the grid size. For more details, one should consult the original paper by Strain (1994). In the actual discretization of this method, we truncate the sums in the usual way by varying \( |k| \) from \(-N/2\) to \(N/2\). We compute the averages of the operator’s coefficients with Simpson’s numerical integration formula. Finally, the discrete linear system still has a solution unique only up to a constant and we can use the same strategy as in the previous case to fix it.

5. Numerical tests

5.1 A theoretical example

Our goal here is to observe and compare the behaviour of the two implementations presented in Sections 4.1 and 4.2. Starting with a known \( u \) and a known \( g \), we compute the corresponding right-hand side \( f \) with (4), and then we run the algorithm to obtain \( u_n \). We consider functions of the form

\[
    u(x_1, x_2) = \frac{1}{k} \cos(2\pi \gamma x_1) \sin(2\pi \gamma x_2),
\]

\[
    g(x_1, x_2) = 1 + \alpha \cos(2\pi \rho x_1) \cos(2\pi \rho x_2).
\]

For the first implementation, we select for the BICG algorithm a tolerance of \( 10^{-4} \) and a maximum number of 1000 iterations per Newton step. For the Fourier transforms implementation, we take the same tolerance with a restarting threshold of \( m = 10 \) inner iterations for the GMRES algorithm. In both cases, a value of \( \tau = 1 \) was enough to achieve convergence. The errors \( \|u - u_n\|_\ell^2 \) and \( \|f - f_n\|_\ell^2 \) are plotted in Fig. 1 as functions of the Newton iterations for both implementations and for various grid sizes ranging from \( 16 \times 16 \) to \( 256 \times 256 \) grid points. We see that in both cases the error gets smaller as we increase the grid size. In particular, for this value of \( \tau \), after the first four iterations, where \( \|u - u_n\|_\ell^2 \) settles down very quickly, the convergence of \( \|f - f_n\|_\ell^2 \) follows a linear slope with a convergence rate slightly higher than a half. The estimated ratio is actually about 0.45 in the FFT case and is about 0.33 in the finite-difference case, so the convergence is faster in this latter case for this final stage. Computing the observed order of accuracy from the errors between \( u \) and \( u_n \), we get from smaller to bigger grid sizes, 4.3521, 4.0035, 3.9965 and 3.9990. This confirms that the fourth-order is consistent with the order of the finite-difference scheme used to compute derivatives of \( u_n \).

In order to investigate whether we can decrease the computing time without losing too much precision on the results, we try to run the experiment again with a tolerance \( 10^{-1} \) (see Fig. 2). Due to the looser tolerance employed, the results are more erratic for the finite-difference implementation, but overall still very good. Figure 2(c) shows the 3D plot of \( u - u_n \) for \( N = 128 \) in the Fourier transform case to get an idea of the distribution of the errors. As we can see, they seem evenly distributed on the whole domain. Figure 2(d) depicts what happens when we vary the value of the step-size parameter \( \tau \). The results behave according to our expectations, with a slower convergence for a bigger \( \tau \). Note that for this new tolerance, the computational cost of one iteration is now much less and the global computing time decreases significantly in both cases. We can quantify this by looking at Table 2. Observe that the BICG algorithm required less operations than the worst case scenario \( O(P^2) \). This being said, we still realize at first glance that the FFT method is much faster than the finite-difference method. The
Fig. 1. Error behaviour for $u_0$ and $f_n$ for a tolerance of $10^{-4}$. (a) The $\|u - u_n\|_2$ error for the Fourier transform implementation on a semilog plot. (b) The $\|u - u_n\|_2$ error for the finite-difference implementation on a semilog plot. (c) The $\|f - \tilde{f}_n\|_2$ error for the Fourier transform implementation on a semilog plot. (d) The $\|f - \tilde{f}_n\|_2$ error for the finite-difference implementation on a semilog plot.

number of GMRES iterations per Newton iteration stayed nearly constant as we increased the grid size, which confirms the $O(P \log P)$ computational complexity.

Finally, in order to get an idea of the stability properties of both methods, we can measure the norm of the inverse of the matrix corresponding to the discretization of the linearized Monge–Ampère operator (Leveque, 2007). This can be achieved by computing the spectral radius of such matrix. In the finite-difference case, we could get this eigenvalue directly by first obtaining the inverse, and then computing the eigenvalues of the new matrix. We observed that for the current experiment, the spectral radius starts at about 10 for $N = 16$ and then grows almost linearly as we increase the grid size. Hence, the finite-difference method appears unstable. For the FFT case, since we do not possess an explicit representation of the matrix, we have to use an indirect method to compute the spectral radius. The one we used is the power method. For a matrix $A$, it starts with a vector $b_0$ and compute the iterates $b_{k+1} = Ab_k / \|Ab_k\|$. If $A$ has a dominant eigenvalue and if $b_0$ has a non-zero component in the direction of the
eigenvector associated with this largest eigenvalue, then the sequence \((b_k)\) converges to the eigenvector associated with the spectral radius of \(A\) (Golub & Van Loan, 1996). For the current experiment, we apply this technique to the inverse matrix produced at every Newton step \(n\) by the discretization of the linearized Monge–Ampère equation via the FFT implementation. More specifically, for a given \(n\), we start with a \(b_0\) randomly generated with components in \([0, 1]\). Then, using the method presented in Section 4.2, we compute the product \(A_n^{-1} b_k\) and then iterate with \(A_n^{-1} b_k / \|A_n^{-1} b_k\|_2\). Repeating this procedure several times with different random vectors \(b_0\), we observed that the power iteration always converges to a number close to 0.03 after only about \(k = 5\) iterations, for every \(n\) from 0 to 20 and for every \(N\) from 16 to 256. Thus, we conclude for this specific example that the spectral radius of the inverse matrix generated at every step of the Newton method is close to 0.03, which of course suggests that the FFT implementation is stable.

Fig. 2. Several examples of the results obtained with a tolerance of \(10^{-1}\). (a) The \(\|u - u_n\|_2\) error for the Fourier transform implementation on a semilog plot. (b) The \(\|u - u_n\|_2\) error for the finite-difference implementation on a semilog plot. (c) Surface plot of \(u - u_6\) for the Fourier transform implementation in the \(N = 128\) case. (d) The \(\|u - u_n\|_2\) error for the Fourier transform implementation with different values of \(\tau\) in the \(N = 128\) case on a semilog plot.
Table 2  Average number of BICG and GMRES iterations per Newton iteration and total computing time in seconds for the whole experiment (20 iterations) when the tolerance is set to $10^{-1}$. We used a MATLAB implementation on an Intel Xeon running at 2.33 GHZ. This is presented for all the different grid sizes.

<table>
<thead>
<tr>
<th>$N$</th>
<th>Average number of GMRES iterations</th>
<th>Total computing time for Fourier transforms</th>
<th>Average number of BICG iterations</th>
<th>Total computing time for finite differences</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>5.32</td>
<td>1.07</td>
<td>14.21</td>
<td>2.21</td>
</tr>
<tr>
<td>32</td>
<td>6.37</td>
<td>1.94</td>
<td>17.79</td>
<td>8.70</td>
</tr>
<tr>
<td>64</td>
<td>7.32</td>
<td>8.06</td>
<td>31.11</td>
<td>79.17</td>
</tr>
<tr>
<td>128</td>
<td>7.95</td>
<td>34.38</td>
<td>63.32</td>
<td>1221.10</td>
</tr>
<tr>
<td>256</td>
<td>8.05</td>
<td>145.07</td>
<td>134.63</td>
<td>34639.82</td>
</tr>
</tbody>
</table>

5.2 Application to medical imaging

Here, we test our algorithm on one of the various applications of OT. In the area of image processing, one of the most common tasks performed by practitioners is to determine a geometric correspondence between images taken from the same scene in order to compare them or to integrate the information they contain to obtain more meaningful data. One could think of pictures acquired at different times, with different types of equipment or from different viewpoints. This process falls into the category of what is referred to as image registration. There are two main types of image registration methods: the rigid ones, which involve translations or rotations, and the non-rigid ones, where some stretching of the image is required to map it to the other one. There has been attempts to use the OT theory as a non-rigid image registration technique (Haker et al., 2004; Rehman et al., 2009). Indeed, consider, for example, two greyscale images. We could think of them as representing a mass distribution of the amount of light ‘piled up’ at a given location. A bright pixel on that image would then represent a region with more mass, whereas a darker pixel would correspond to a region with less mass. Computing the optimal map between the two images and analysing the rate of change of that map would then reveal the best way (in terms of minimizing the transportation distance) of moving the mass from the first density to the second. In Rehman et al. (2009), the authors actually list a few advantages of the OT method for image registration. However, the method suffers from limitations due to the fact that in general, the OT map does not reflect the physical properties of the objects under study (Museyko et al., 2009). In any case, this context provides a good scenario for us to test our method on more applied examples.

Our first applied test is thus in the field of medical imagery. We consider the two MRI brain scans presented in Fig. 3. These images were taken from the BrainWeb simulated brain database at McGill university (Cocosco et al., 1997) and represent a slice of a healthy brain and a slice of the same brain where the MS disease is spreading. This nervous system disease damages the myelin sheets around the axons of the brain and leaves scars (or sclerosis) visible on an MRI. We chose MS as a test case since its actual detection process relies on neuro-imaging, which tries to identify the scars whose presence leaves traces similar to those of multiple tumours. We first renormalize the densities given by the greyscale distribution of pixels in the images by applying the translation presented in (15). We thus obtain two new densities, $f$ and $g$, on which we apply the algorithm. For the images considered here, these resulting densities also happen to be bounded away from zero, and thus no additional preprocessing step was required to make our method converge. Note that this might not always be the case and that we might sometimes need to modify the translated images before we can use them (so that they are bounded away
Fig. 3. The results of the MS detection experiment with the Fourier transform implementation. (a) Initial density $f$: MRI scan of a normal brain. (b) Final density $g$: MRI scan of the same brain with MS lesions. (c) Surface plot of $\text{div}(u_4)$. (d) Scan of the healthy brain on which was superposed the coloured (and filtered) contour plot of $\text{div}(u_4)$.

from zero). We also point out that the optimal map transporting the first image into the second one could be different from the one transporting the corresponding translated images ($f'$ and $g'$). However, as one can observe on the results in Fig. 3, we found that this is not a serious shortcoming for the application considered here. On the bottom panels of Fig. 3, we show the results reached after only four
Newton iterations with $\tau = 1$ and a tolerance of $10^{-2}$ for the Fourier transform implementation. We note that the corresponding $L^2$ norm of the error between $f$ and the density $f_4$ obtained after four Newton iterations is reduced to about 0.001. The 3D plot in Fig. 3(c) is characterized by very sharp spikes corresponding to variations in brightness between the two images where the scars are located. To get a better visual understanding of the results, we coloured the interior of the contour lines corresponding to the affected regions and we superposed this image to the MRI scan of the healthy brain (Fig. 3(d)). The number of GMRES iterations required per Newton iteration was very small and nearly constant (only one outer iteration and about six inner ones). Even if our code was not necessarily optimized in terms of computational speed, it only took about 30 s to compute these results on an Intel Xeon with 2.33 GHZ of RAM. Moreover, the spectral radius was still very close to 0.03 for the inverse discretization matrix, which again suggests stability. In addition to that change detection, the OT plan $\tilde{T} = x + \nabla u(x)$ actually provides information on the amount of variation from one MRI scan to the other. Indeed, we can define a metric between probability densities from the solution to the transport problem, the squared distance being

$$d^2(f, g) = \int |x - \tilde{T}(x)|^2 f(x) \, dx = \int |\nabla u(x)|^2 f(x) \, dx.$$

This can be used as a quantifier for the magnitude of change between the two images and thus help monitor the growth of the disease. In our experiment, we got a value of $4.14 \times 10^{-10}$. Note that when we compared it with other ones obtained from different numerical experiments on MRI scans with presence of MS, these numbers validated our visual estimates; the more the scars, the greater the number. Recall finally that even though we implemented the algorithm only in 2D, in theory it is valid in any dimension. Therefore, it could also be applicable on 3D datasets which would be much more realistic when it comes to analysing biological phenomena similar to the ones we present here.

5.3 A standard image processing example

As a final test case, we select an image processing example for which the two images are much more different from each other than in the previous case. We chose two famous pictures in the image processing community as initial and final densities, namely Lena and Tiffany. We first scale the images to $256 \times 256$ pixels and we renormalize them by applying the same translation as in Section 5.2. As in the previous case, this also has the additional (and not always guaranteed) effect of bounding the images away from zero sufficiently for our method to converge. The resulting images are presented in Fig. 4(a,f). We point out that we did not apply any smoothing to these images. Our goal is then to see whether or not we can obtain the optimal map that transforms the image in (f) into the image in (a). Note that this map could be different from the optimal map obtained if we enforced instead Dirichlet boundary conditions (for example) because our periodic boundary conditions allow mass to be moved across the boundary, to other periodic cells.

We select $\tau = 2$, $tol = 10^{-1}$ and run 20 iterations of the Newton algorithm with the Fourier transform implementation. The output is presented in Fig. 4. Yet again, the number of GMRES iterations stayed nearly constant (only about four inner iterations), which made the computing time very small, and the spectral radius of the inverse of the discretization matrix stayed close to 0.03. We see that $f_{20}$ and $f$ are almost visually identical, and thus, our algorithm performed well on this more difficult example. We also observed that the optimal map we obtained does indeed move mass across the boundary. This was not the case for our previous MRI brain scan example since the regions where mass is moving are far from the boundary. One should thus understand that the map we obtained for the Lena–Tiffany warp is optimal if we consider our domain to be the surface of a torus.
This time, our method did not converge for $\tau = 1$ (we had to use $\tau = 2$). This can be explained by the fact that the mass densities associated with the images are quite different from one another. In addition, even by translating the images, some values of the mass density functions were still very close to 0.
We repeated the same experiment with the finite-difference implementation and found that the \( \tau \) required jumped to 8 and we had to iterate about 60 times to get good results. We also tried to use a linear interpolation instead of a closest neighbour interpolation to compute the compositions with the Fourier transforms implementation. In this case, at the expense of a larger computational time, the maximum of \( f - f_n \) was significantly reduced compared with the one obtained with the closest neighbour approach, and thus the quality of the approximation was improved.

6. Concluding remarks

In conclusion, we saw that our algorithm presents a good way of computing the numerical solution of the \( L^2 \) OT problem in the case where the mass densities are smooth, periodic and bounded away from 0. The Fourier transform implementation makes it accurate, fast, stable and thus very efficient. In the context of image registration, the limitation to densities bounded away from 0 and to periodic boundary conditions did not seem to be a serious shortcoming for applying this algorithm to some practical examples, provided the densities do not vary much close to the boundary. Indeed, one has to keep in mind that the OT mapping we obtain is optimal for the specific geometry of the torus. We observed in the Lena–Tiffany warp that it can move mass across the boundary of the square domain considered. It is worth pointing out that the results of both a Lena–Tiffany warp experiment and an MRI brain scan warp experiment are also presented in Chartrand et al. (2009), with a different numerical method. We also saw that even if our method has the downside of having to choose a value of \( \tau \) without giving too much information on how to make this choice \textit{a priori}, when using the Fourier transform implementation, in all our experiments we never had to take a \( \tau \) bigger than 2 to get convergence. We also conducted more numerical experiments in Saumier (2010), such as taking the initial and final densities to be the periodic approximation of Gaussian distributions, and the performances were as good as the ones presented here.

One of the reasons why the Fourier transforms implementation was more effective than the finite-difference implementation is the preconditioning applied to the operator before discretization for the former method. Indeed, for the results presented in Section 5.1, no preconditioner was applied in the finite-difference implementation. We actually tried to apply different types of simple preconditioners to the linear system corresponding to the discretization of the operator with the finite-difference implementation. The results were better, but not as good as the ones obtained with the Fourier transforms implementation. In Finn et al. (2008), the authors used a method similar to the finite-difference implementation presented here to solve the Monge–Ampère equation (also involving a combination of the Newton and GMRES algorithms). By employing a multigrid preconditioner, they were able to obtain the numerical solution with computing times comparable with the ones we presented for our Fourier transforms method, for different boundary conditions.

We reiterate that the strength of the algorithm presented in this paper is that it is specifically built to deal with periodic mass densities and that one needs to be cautious when applying to situations with different geometries. In addition, several numerical methods present in the literature to solve the Monge–Ampère equation are limited to uniform target densities or to 2D problems. Ours can deal with a more general class of densities in any dimension. However, this class of densities could still be much larger: as mentioned in Section 1, some other numerical methods can handle vanishing densities with less stringent smoothness assumptions on arbitrary domains. Since our algorithm employs fourth-order centred finite differences to approximate the derivatives of \( u_n \), we do not expect it to work for weak solutions. We nevertheless saw that it was very efficient for the particular case of smooth enough, non-vanishing and periodic densities.
To pursue this work in the future, we would like to extend the method and corresponding convergence result to different types of boundary conditions. However, for this to happen, we would require global a priori estimates on the Hölder norm of the solution of the Monge–Ampère equation, which to the best of our knowledge are not yet available. Moreover, in order to improve its performance even more, it would be interesting to implement a version of the algorithm that would take advantage of modern parallel computing techniques.

Funding

This research was partially supported by Discovery grants and fellowships from the Natural Sciences and Engineering Research Council of Canada and by the University of Victoria.

References


