Vietnam Journal of Mathematics (2023) 51:113–126
https://doi.org/10.1007/s10013-022-00578-z

ORIGINAL ARTICLE

Solving Cubic Matrix Equations Arising in Conservative Dynamics

Michele Benzi[1] · **Milo Viviani**[2]

Received: 23 November 2021 / Accepted: 18 April 2022 / Published online: 11 October 2022

Abstract
In this paper we consider the spatial semi-discretization of conservative PDEs. Such finite dimensional approximations of infinite dimensional dynamical systems can be described as flows in suitable matrix spaces, which in turn leads to the need to solve polynomial matrix equations, a classical and important topic both in theoretical and in applied mathematics. Solving numerically these equations is challenging due to the presence of several conservation laws which our finite models incorporate and which must be retained while integrating the equations of motion. In the last thirty years, the theory of geometric integration has provided a variety of techniques to tackle this problem. These numerical methods require solving both direct and inverse problems in matrix spaces. We present three algorithms to solve a cubic matrix equation arising in the geometric integration of isospectral flows. This type of ODEs includes finite models of ideal hydrodynamics, plasma dynamics, and spin particles, which we use as test problems for our algorithms.

Keywords Cubic matrix equations · Lie–Poisson integrator · Euler equations · Plasma vortices · Spin systems

Mathematics Subject Classification (2010) 37M15 · 65F45 · 65P10

1 Introduction

The numerical solution of polynomial matrix equations is a well studied and active field of research [2, 3, 8]. Its relevance clearly goes beyond pure mathematics and the development

Dedicated to Alfio Quarteroni for his 70th Birthday.

✉ Milo Viviani
milo.viviani@sns.it

Michele Benzi
michele.benzi@sns.it

[1] Scuola Normale Superiore, Piazza dei Cavalieri, 7, Pisa, 56126, Italy

[2] CRM Ennio De Giorgi - Collegio Puteano, Scuola Normale Superiore, Piazza dei Cavalieri, 3, Pisa, 56126, Italy

of efficient algorithms is crucial in several areas of computational science. Linear matrix equations have been studied since the 19th Century, beginning with Sylvester [9]. Linear matrix equations have a variety of different formulations, and according to the specific structure, different techniques can be used to solve them [3, 8]. One degree higher, we find quadratic matrix equations. For such problems the theory is more intricate and various numerical issues may appear [2]. Among the quadratic matrix equations, one of the most studied is the continuous-time algebraic Riccati equation (CARE) [2], which will appear in Section 3.

In this paper, we study the numerical solution of the cubic matrix equation

$$(I - h\mathcal{L}X)X(I + h\mathcal{L}X) = Y, \tag{1}$$

where X is unknown and Y is given in $\mathbb{M}(N, \mathbb{R})$ or $\mathbb{M}(N, \mathbb{C})$ (the spaces of $N \times N$ real or complex matrices), $\mathcal{L} \neq 0$ is a linear operator acting on matrices and $h > 0$. Equation (1) appears in the geometric integration of matrix flows of the form:

$$\begin{aligned} \dot{Y} &= [\mathcal{L}Y, Y], \\ Y(0) &= Y_0, \end{aligned} \tag{2}$$

where $Y = Y(t)$ is a curve in some subspace of the space $\mathbb{M}(N, \mathbb{R})$ or $\mathbb{M}(N, \mathbb{C})$ and the square brackets denote the commutator of two matrices: $[A, B] = AB - BA$. The flow of (2) is isospectral, which means that the eigenvalues of $Y(t)$ do not depend on t. Furthermore, when $\mathcal{L}$ is self-adjoint with respect to the pairing $\langle A, B\rangle = \mathrm{Tr}(AB)$, (2) is Hamiltonian (another term is "Lie-Poisson"), with Hamiltonian function given by

$$H(Y) = \frac{1}{2}\mathrm{Tr}(Y\mathcal{L}Y). \tag{3}$$

A discrete approximation of the solution of (2) is determined, for $h > 0$ sufficiently small, by the implicit-explicit iteration defined in [10] as:

$$\begin{aligned} (I - h\mathcal{L}X_n)X_n(I + h\mathcal{L}X_n) &:= Y_n, \\ (I + h\mathcal{L}X_n)X_n(I - h\mathcal{L}X_n) &=: Y_{n+1}. \end{aligned} \tag{4}$$

In this scheme, Y_n denotes the approximate solution at time t_n. This scheme preserves the spectrum of Y_0 and nearly conserves the Hamiltonian (3), indeed it is a Lie–Poisson integrator (see [10]). We observe that whenever Y belongs to a quadratic matrix Lie algebra

$$\mathfrak{g}_J = \{Y \in \mathbb{M}(N, \mathbb{C}) \,|\, YJ + JY^* = 0\}$$

for some fixed $J \in \mathbb{M}(N, \mathbb{C})$, and $\mathcal{L} : \mathfrak{g}_J \to \mathfrak{g}_J$, (1) admits solutions in $\mathfrak{g}_J$. Indeed, the left-hand side of (1) is the differential of the inverse of the Cayley transform, which is know to preserve quadratic Lie algebras [4]. Of particular interest for applications to PDEs is the case of $J = I$, for which $\mathfrak{g}_J = \mathfrak{u}(N)$, the Lie algebra of the skew-Hermitian matrices. In Section 4, we consider the Lie algebra $\mathfrak{su}(N)$, which consists of skew-Hermitian matrices with zero trace.

Remark 1 It is not hard to check that the following equalities hold:

$$\begin{aligned} Y_{n+1} &= Y_n + h\left[\mathcal{L}\left(\frac{Y_{n+1} + Y_n}{2}\right), \frac{Y_{n+1} + Y_n}{2}\right] + \mathcal{O}(h^2), \\ X_{n+1} &= X_n + \frac{h}{2}([\mathcal{L}X_{n+1}, X_{n+1}] + [\mathcal{L}X_n, X_n]) + \mathcal{O}(h^2). \end{aligned} \tag{5}$$

Hence, up to a term of order $\mathcal{O}(h^2)$, Y_n evolves via the midpoint scheme, whereas X_n via the trapezoidal scheme. It is known that the midpoint and the trapezoidal method are *conjugate*

symplectic [4]. Hence, we have that X_n evolves accordingly to a scheme $\phi_h^T : X_n \mapsto X_{n+1}$ which is *conjugate isospectral* to the scheme $\phi_h^M : Y_n \mapsto Y_{n+1}$ as defined in (4), i.e. there exists an invertible map χ_h such that:

$$\phi_h^T = \chi_h^{-1} \circ \phi_h^M \circ \chi_h.$$

The map is clearly given by the left-hand side of the first equation in (4), which we denote as $\chi_h = \phi_{h/2}^{EE}$. We observe that $\chi_h = I + \mathcal{O}(h)$. Hence, if Y_n evolves on a compact set, then X_n evolves on a compact set for any $n \geq 0$ [4, VI.8]. We illustrate this relationship in Fig. 1 below.

The need for an efficient solver for (1) can be understood by the fact that conservative PDEs, like the vorticity equation of fluid dynamics [11, 12] or the drift-Alfvén model for a quasineutral plasma [6], admit a spatial discretization in $\mathfrak{su}(N)$, for $N = 1, 2, \ldots$. The crucial aspect of these finite-dimensional models is that they retain the conservation laws of the original equations. In order to retain these features, the resulting semi-discretized equations can be integrated in time using the scheme (4). Clearly, to get good spatial accuracy, N has to be quite large (at least 10^3). Moreover, the need for an efficient solver for (1) can be necessary for spin systems with many interacting particles. In this case, the equation (2) are posed in the product Lie algebra $\mathfrak{su}(2)^N$, where N is the number of particles. We stress that in a typical simulation, hundreds of these cubic matrix equations have to be solved to high accuracy. This paper is devoted to devise an efficient way to solve (1). First we prove existence and uniqueness of the solution for (1), for h sufficiently small. Then, we propose and investigate three possible algorithms to solve (1), which intrinsically preserve the quadratic matrix Lie algebras. First, in Section 3.1 we consider an explicit fixed point iteration scheme, whose convergence follows from the existence and uniqueness result. Then, in Section 3.2, we consider a linear scheme again based on a fixed point iteration which requires the solution of a linear matrix equation. Again the existence and uniqueness result guarantees the convergence of the scheme. We will see in Section 3.4 that a suitable inexact Newton method applied to (1) is also convergent, at least locally. We will see in Section 3.3 that the third scheme, based on the Riccati equation, is not practical, due to the non uniqueness of the solution. In the last section, we show the results of numerical experiments aimed at assessing the efficiency of the different schemes for various linear operators $\mathcal{L}$, which correspond to different physical models.

We mention in passing that a different cubic matrix equation has been studied in [1].

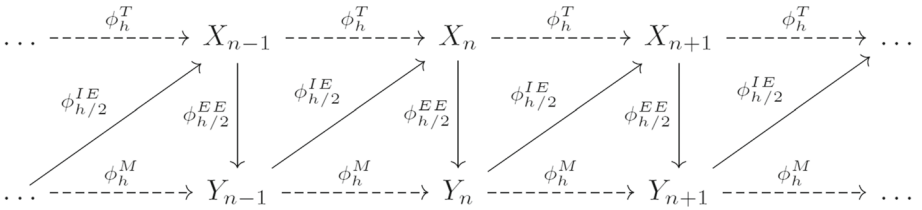

Fig. 1 Illustration of the schemes (4-5). Here ϕ_h^M denotes the map in the first line of (5), ϕ_h^T the map in the second line of (5), $\phi_{h/2}^{IE}$ the map in the first line of (4) and $\phi_{h/2}^{EE}$ the map in the second line of (4). Those correspond up to $\mathcal{O}(h^2)$ terms respectively to the implicit midpoint, trapezoidal, implicit Euler, explicit Euler schemes

2 Existence and Uniqueness

In this section, we show the existence and uniqueness of the solution for the equation (1), when the time-step h is sufficiently small.

Theorem 1 *Given $Y \in \mathbb{M}(N, \mathbb{C})$, (1) has a unique solution for sufficiently small $h > 0$ in some neighbourhood of Y. Furthermore, when (1) takes place in $\mathfrak{su}(N)$, the solution is unique in some neighbourhood of $Y \neq 0$ for any $h < \frac{1}{3\|\mathcal{L}\|_{op}\|Y\|}$.*

Proof We rewrite (1) as a fixed point problem $F_h(X) = X$, for

$$F_h(X) = Y + h[\mathcal{L}X, X] + h^2(\mathcal{L}X)X(\mathcal{L}X). \tag{6}$$

We first show the existence of a solution for the fixed point problem $F_h(X) = X$. Let us introduce the following Cauchy problem:

$$\begin{aligned} &X'(h) = G_h^{-1}(X(h))\left[\tfrac{\partial F_h}{\partial h}(X(h))\right], \\ &X(0) = Y, \end{aligned} \tag{7}$$

where

$$G_h(X) = I - \frac{\partial F_h}{\partial X},$$

which is invertible for h sufficiently small, being $\|\frac{\partial F_h}{\partial X}\|_{op} = \mathcal{O}(h)$. Hence, the Cauchy problem (7) has solution $X(h)$ for $\|\frac{\partial F_h}{\partial X}\|_{op} < 1$, being the right-hand side continuous. This ensures the existence of a solution for the fixed point problem $F_h(X) = X$, because of the equality

$$\frac{d}{dh}[F_h(X(h))] = \frac{\partial F_h}{\partial h}(X(h)) + \frac{\partial F_h}{\partial X}[X'(h)].$$

In order to get uniqueness for the fixed point problem $F_h(X) = X$, we show that there exists a neighbourhood of a fixed point X containing Y in which F_h is a contraction. Let us calculate

$$\begin{aligned} F_h(X) - F_h(Z) &= h([\mathcal{L}X, X] - [\mathcal{L}Z, Z]) + h^2((\mathcal{L}X)X(\mathcal{L}X) - (\mathcal{L}Z)Z(\mathcal{L}Z)) \\ &= h([\mathcal{L}(X - Z), X] + [\mathcal{L}Z, X - Z]) \\ &\quad + h^2((\mathcal{L}X)(X - Z)(\mathcal{L}X) + (\mathcal{L}(X-Z))Z(\mathcal{L}X) + (\mathcal{L}Z)Z(\mathcal{L}(X-Z))). \end{aligned}$$

Hence

$$\begin{aligned} \|F_h(X) - F_h(Z)\| \leq\ & h\|\mathcal{L}\|_{op}(\|X\| + \|Z\|)\|X - Z\| \\ & + h^2\|\mathcal{L}\|_{op}^2(\|X\|^2 + \|X\|\|Z\| + \|Z\|^2)\|X - Z\|. \end{aligned}$$

Therefore, given a fixed point X and $0 < \varepsilon < 1$, in the neighbourhood of $(0, X)$

$$\begin{aligned} \mathfrak{U}_\varepsilon =\ & (0, X) + \{(h, Z) \mid h\|\mathcal{L}\|_{op}(\|X\| + \|X + Z\|) \\ & + h^2\|\mathcal{L}\|_{op}^2(\|X\|^2 + \|X\|\|X + Z\| + \|X + Z\|^2) < 1 - \varepsilon\}, \end{aligned} \tag{8}$$

we find $\{h\} \times B_r(X) \subset \mathfrak{U}_\varepsilon$, with $h, r > 0$ determined by (8), such that $F_h : B_r(X) \to B_r(X)$ is a contraction and in $\overline{B_r(X)}$ we can apply the Banach–Caccioppoli theorem and get a unique solution to the fixed point problem $F_h(X) = X$. Taking $X + Z = Y$, we get the neighbourhood of Y in which we have a unique solution for (1).

Let us now assume that (1) takes place in $\mathfrak{su}(N)$. Then

$$\begin{aligned}\|Y\|^2 &= \|(I - h\mathcal{L}X)X(I + h\mathcal{L}X)\|^2 \\ &= \mathrm{Tr}((I - h\mathcal{L}X)X(I + h\mathcal{L}X)((I - h\mathcal{L}X)X(I + h\mathcal{L}X))^*) \\ &= \mathrm{Tr}(X(I - (h\mathcal{L}X)^2)(I - (h\mathcal{L}X)^2)X^*).\end{aligned}$$

The matrix $(I-(h\mathcal{L}X)^2)(I-(h\mathcal{L}X)^2)$ is symmetric positive definite for any X. Indeed, the eigenvalues of $-(h\mathcal{L}X)^2$ are real non negative, since $\mathcal{L}X \in \mathfrak{su}(N)$. Therefore, $\|X\| \leq \|Y\|$. Hence, replacing $\|X\|$, $\|X + Z\|$ with $\|Y\|$ in (8) and letting $\varepsilon \to 0$, we can find, for $Y \neq 0$, the following bound for h:

$$h < \frac{1}{3\|\mathcal{L}\|_{op}\|Y\|}.$$

□

3 Numerical Schemes

In this section, we present three possible numerical schemes to solve (1).

3.1 Explicit Fixed Point

Theorem 1 gives a first scheme to solve (1):

$$X_{k+1} := F_h(X_k) = Y + h[\mathcal{L}X_k, X_k] + h^2(\mathcal{L}X_k)X_k(\mathcal{L}X_k) \tag{9}$$

for $k = 1, 2, \ldots$. From Theorem 1 we have that F_h is a contraction for h sufficiently small. Hence, the fixed point iteration has a unique solution for h small. When we look for solutions in $\mathfrak{su}(N)$, we can take any $h < \frac{1}{3\|\mathcal{L}\|_{op}\|Y\|}$. The resulting Algorithm 1 is given below. We observe that the cost per iteration is $\mathcal{O}(N^3)$.

Require: $Y \in \mathfrak{su}(N)$; $\mathcal{L} : \mathfrak{su}(N) \to \mathfrak{su}(N)$

$tol \leftarrow 10^{-10}$
$err \leftarrow 1$
$X_0 \leftarrow Y$
while $err > tol$ **do**
  $X_1 \leftarrow F_h(X_0)$
  $err \leftarrow \|X_1 - X_0\|$
  $X_0 \leftarrow X_1$
end while

Algorithm 1 Linear scheme in $\mathfrak{su}(N)$.

3.2 Linear Scheme

Equation (1) can be decomposed into two coupled matrix equations. This splitting induces the following scheme:

$$\begin{aligned} P_k &:= \mathcal{L}X_k, \\ (I - hP_k)X_{k+1}(I + hP_k) &:= Y, \end{aligned} \tag{10}$$

for $k = 1, 2, \ldots$. Note that the second matrix equation in (10) is linear in the unknown matrix X_{k+1}. It is straightforward to check that $X_{k+1} = S(X_k)$, for $S_h(X) = (I - h\mathcal{L}X)^{-1}Y(I + h\mathcal{L}X)^{-1}$. Theorem 1 gives existence and uniqueness of a solution $\overline{X}$ for h sufficiently small for (1). Hence, since S_h is analytic in the set $\{(h, X) | \|h\mathcal{L}X\| < 1\}$, we can conclude that the fixed point iteration $X_{k+1} = S_h(X_k)$ converges to $\overline{X}$, when X_0 is taken in a closed neighbourhood of $\overline{X}$ in which S_h is a contraction.

When $P \in \mathfrak{su}(N)$, we have $(I - hP)^* = (I + hP)$. Hence, it is enough to calculate the LU-factorization for $(I - hP)$ to have the one for $(I + hP)$. The resulting Algorithm 2 is given below. We observe that the cost per iteration is $\mathcal{O}(N^3)$.

Require: $Y \in \mathfrak{su}(N)$; $\mathcal{L} : \mathfrak{su}(N) \to \mathfrak{su}(N)$

$tol \leftarrow 10^{-10}$
$err \leftarrow 1$
$X_0 \leftarrow Y$
while $err > tol$ **do**
  $P \leftarrow \mathcal{L}X_0$
  $[L, U] = lu\_factorization(I - hP)$
  $X_1 \leftarrow U^{-1}L^{-1}Y(L^{-1})^*(U^{-1})^*$
  $err \leftarrow \|X_1 - X_0\|$
  $X_0 \leftarrow X_1$
end while

Algorithm 2 Linear scheme in $\mathfrak{su}(N)$.

3.3 Quadratic Scheme

Similarly, we can consider the same decomposition of the previous section for (1), but reversing the roles of the known and unknown variables. This splitting induces the following scheme:

$$\begin{aligned} (I - hP_k)X_k(I + hP_k) &:= Y, \\ \mathcal{L}X_{k+1} &:= P_k, \end{aligned} \tag{11}$$

where the unknown in the first equation is P_k. The first quadratic equation can be put in the form

$$h^2PXP + h[P, X] + Y - X = 0, \tag{12}$$

which is a type of CARE.

Consider the case when (12) is posed in $\mathfrak{su}(N)$. Then we have two main issues concerning its solvability. On the one hand, defining $Z := (I - hP)$, we see that the first equation can be written as:

$$ZXZ^* = Y. \tag{13}$$

Therefore, in order for (13) to have solution, X and Y must be congruent. Hence, X_0 must be defined via some congruence transformation of Y. On the other hand, the following proposition shows that (13) admits infinitely many solutions. Let $I_{p,N-p}$ denote the diagonal matrix with the first p entries equal to 1 and the remaining $N-p$ entries equal to -1, where $0 \leq p \leq N$. We denote by $U(p, N-p)$ the Lie group of matrices that leave the bilinear form $b(x, y) = x^* I_{p,N-p} y$ invariant.

Proposition 2 *Let $A, B \in \mathfrak{su}(N)$ be non-singular, with signature matrix equal to $I_{p,N-p}$. Then, the equation*

$$ZAZ^* = B \tag{14}$$

has solution $Z \in GL(N, \mathbb{C})$ (the Lie group of invertible complex matrices) if and only if $Z = CUD$, for some $U \in U(p, N-p)$ and C, D non-singular such that $B = CI_{p,N-p}C^*$ and $DAD^* = I_{p,N-p}$.

Proof Let A, B, C, D be as in the hypotheses. Then, we can rewrite (14) as

$$C^{-1}ZD^{-1}I_{p,N-p}(C^{-1}ZD^{-1})^* = I_{p,N-p}.$$

Therefore, $C^{-1}ZD^{-1}$ is an element of $U(p, N-p)$. On the other hand, for any $U \in U(p, N-p)$, we have that $Z = CUD$ is a solution of (14). Hence, all the solutions of (14) have this form and are parametrized by $U \in U(p, N-p)$. □

In our particular situation, we are interested in solutions of the form $Z = I + P$, for P skew-Hermitian. For instance, taking $A, B \in \mathfrak{su}(N)$ diagonal such that $\mathrm{i}(A-B) \geq 0$ and $\mathrm{i}A < 0$, any P diagonal skew-hermitian (i.e., purely imaginary) such that $P^2 = BA^{-1} - I$ is a solution. Hence, for generic A, B as above, we get 2^N solutions, making the iteration (11) not well-defined.

We can see that the Riccati equation (12) has a non-uniqueness issue also in the following way. The equation can be split into two orthogonal components, one parallel to X and one orthogonal to it with respect to the Frobenius inner product:

$$\begin{aligned} h^2\Pi_X(PXP) + \Pi_X Y - X &= 0, \\ h^2\Pi_X^{\perp}(PXP) + h[P, X] + \Pi_X^{\perp} Y &= 0, \end{aligned}$$

where Π_X is the orthogonal projection onto $\mathrm{stab}_X := \{A \in \mathbb{M}(N, \mathbb{C}) \text{ s.t. } [A, X] = 0\}$. If we write $P_{\parallel} := \Pi_X P$ and $P_{\perp} := \Pi_X^{\perp} P$, we get:

$$\begin{aligned} h^2 P_{\parallel} X P_{\parallel} + h^2 \Pi_X(P_{\perp} X P_{\perp}) + \Pi_X Y - X &= 0, \\ h^2\Pi_X^{\perp}(P_{\perp} X P_{\perp}) + h^2 P_{\perp} X P_{\parallel} + h^2 P_{\parallel} X P_{\perp} + h[P_{\perp}, X] + \Pi_X^{\perp} Y &= 0. \end{aligned}$$

If these equations have a solution $(P_{\parallel}, P_{\perp})$, then we also have a solution $(-P_{\parallel}, P'_{\perp})$, where $P'_{\perp} = P_{\perp} + \mathcal{O}(h^2)$. In $\mathfrak{su}(2) \cong \mathbb{R}^3$ this is easily seen, since the above scheme reads [10]:

$$\begin{aligned} h^2 p_{\parallel}(x \cdot p_{\parallel}) + \Pi_x y - x &= 0, \\ h^2 p_{\perp}(x \cdot p_{\parallel}) + h p_{\perp} \times x + \Pi_x^{\perp} y &= 0, \end{aligned}$$

where $\times$ denotes the vector product and the matrices in $\mathfrak{su}(2)$ have been represented as vectors in $\mathbb{R}^3$, via the standard isomorphism. Hence, we have the solutions:

$$\begin{aligned} p_{\parallel} &= \pm\sqrt{\frac{\|x\| - \|\Pi_x y\|}{h^2\|x\|}}\,\frac{x}{\|x\|}, \\ R p_{\perp} &= -\Pi_x^{\perp} y, \end{aligned}$$

where $R \in \mathbb{M}(3, \mathbb{R})$ is such that $Rp_\perp = h^2 p_\perp (x \cdot p_\parallel) + h p_\perp \times x$. Hence, the ambiguity of the sign in $p_\parallel$ causes a non-uniqueness of solution.

3.4 Cubic Scheme

Newton's method can be directly applied to solve (1). A practical implementation is obtained rewriting (1) in the following way:

$$F(X) = X - h[\mathcal{L}X, X] - h^2(\mathcal{L}X)X(\mathcal{L}X) - Y = 0.$$

The Jacobian of F applied to a matrix Z is given by:

$$\begin{aligned} DF(X)[Z] \;=\; & Z - h([\mathcal{L}Z, X] + [\mathcal{L}X, Z]) \\ & -h^2((\mathcal{L}Z))X(\mathcal{L}X) + (\mathcal{L}X)Z(\mathcal{L}X) + (\mathcal{L}X)X(\mathcal{L}Z)). \end{aligned} \tag{15}$$

Remark 2 Here we consider an inexact Newton approach. Hence, in order to apply the Newton's method, we consider some approximation for $DF(X)^{-1}$. We notice that $DF(X) = I - h(\mathcal{B}_1 + h\mathcal{B}_2)$, for $\mathcal{B}_1 = [\mathcal{L}\cdot, X] + [\mathcal{L}X, \cdot]$ and $\mathcal{B}_2 = (\mathcal{L}\cdot))X(\mathcal{L}X) + (\mathcal{L}X) \cdot (\mathcal{L}X) + (\mathcal{L}X)X(\mathcal{L}\cdot)$. Hence, we get the following third order approximation of $DF(X)^{-1}$:

$$DF(X)^{-1} = I + h\mathcal{B}_1 + h^2(\mathcal{B}_1^2 + \mathcal{B}_2) + \mathcal{O}(h^3).$$

At least four reasonable approximations of $DF(X)^{-1}$ can be chosen:

1. $DF(X)^{-1} \approx I + h\mathcal{B}_1$,
2. $DF(X)^{-1} \approx I + h\mathcal{B}_1 + h^2\mathcal{B}_2$,
3. $DF(X)^{-1} \approx I + h\mathcal{B}_1 + h^2\mathcal{B}_1^2$,
4. $DF(X)^{-1} \approx I + h\mathcal{B}_1 + h^2(\mathcal{B}_1^2 + \mathcal{B}_2)$.

We have found out that, among those, the second one in general performs better. Indeed, the first one might have convergence issues for large h, and even for large matrices the performances are at most comparable to the second one. The third one and the fourth one do not perform better than the second one, because the norm of $\mathcal{B}_1^2$ is in general much smaller than the one of $\mathcal{B}_2$. Hence, the fourth one is computationally more expensive than the second one, without any gain in convergence, whereas the third one has a slower convergence than the second one.

Then, we consider the following approximation for the inverse of the Jacobian evaluated in $F(X)$:

$$\begin{aligned} DF(X)^{-1}[F(X)] \;\approx\; & \widetilde{DF}(X)[F(X)] \\ :=\; & F(X) + h([\mathcal{L}F(X), X] + [\mathcal{L}X, F(X)]) \\ & +h^2((\mathcal{L}F(X))X(\mathcal{L}X) + (\mathcal{L}X)F(X)(\mathcal{L}X) + (\mathcal{L}X)X(\mathcal{L}F(X))). \end{aligned}$$

This approximation leads to the inexact Newton scheme (Algorithm 3) described below.

Require: $Y \in \mathfrak{su}(N)$; $\mathcal{L} : \mathfrak{su}(N) \to \mathfrak{su}(N)$

$tol \leftarrow 10^{-10}$
$err \leftarrow 1$
$X_0 \leftarrow Y$
while $err > tol$ **do**
  $X_1 \leftarrow X_0 - \widetilde{DF}(X_0)[F(X_0)]$
  $err \leftarrow \|X_1 - X_0\|$
  $X_0 \leftarrow X_1$
end while

Algorithm 3 Inexact Newton scheme.

We observe that the Inexact Newton method above has its main computational cost in the evaluation of the approximated Jacobian (15), due to the several matrix-matrix multiplications required. Hence, for large N, the lower complexity of the scheme defined in Section 3.2 makes it more advantageous than the Inexact Newton's one, in terms of cost per iteration.

4 Numerical Experiments

In this section, we test our algorithms on three concrete examples arising from the numerical solution of spatially semi-discretized conservative PDEs: the incompressible Euler equations, the Drift-Alfvén plasma model, and the Heisenberg spin chain. To integrate in time the equations of motion, we apply the numerical scheme (4). For each equation, we test the performances of the schemes defined in Sections 3.2 and 3.4. Here we briefly summarize our findings. For large time-step h, the scheme of Section 3.2 is more efficient when solving (1) for large matrices. This makes the linear scheme more suitable for solving the Euler equations or the Drift-Alfvén plasma model. Analogously, we observe that for spin-systems, for large time-step and many particles, the scheme of Section 3.2 is faster and has better convergence properties than the one of Section 3.4.

We observe that for both the Euler equations and Drift-Alfvén plasma model, the number of iterations tends to decrease with increasing N. This is due to the fact that we are normalizing the initial values of the vorticity and the fact that we are absorbing into the time-step the factor $N^{3/2}$ which should multiply the matrix bracket in order to have spatial convergence of the right-hand side of (17) and (19) (see [7]). Indeed, the same phenomenon is not observed for the Heisenberg spin chain, where the spatial discretization is kept fixed while the number of particles is increased.

Another observation is that both in the Euler equations and in the Drift-Alfvén plasma model, the small number of Newton iterations for large N prevents any benefit from combining in series two different algorithms, i.e., using a few steps of a fixed point iteration to get a good initial guess for the inexact Newton scheme. Analogously, we have not observed any improvement in the convergence speed using a mixed scheme for the Heisenberg spin chain.

The simulations are run in Matlab2020a on a Dell laptop, processor Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz, RAM 16.0 GB. For each simulation, a tolerance of 10^{-10} has been used as stopping criterion of the iteration. The results in the tables have been obtained

as the average of 10 runs of the respective algorithm for solving (1), each run with respect to a different randomly generated Y. The CPU time is measured in seconds.

4.1 2D Euler Equations

The 2D Euler equations on a compact surface $S \subset \mathbb{R}^3$ can be expressed in the vorticity formulation as:

$$\begin{aligned} \dot{\omega} &= \{\psi, \omega\}, \\ \Delta\psi &= \omega, \end{aligned} \tag{16}$$

where ω is the vorticity field, ψ is the stream function, the curly brackets denote the Poisson brackets, and Δ is the Laplace–Beltrami operator on S. Let us fix $S = \mathbb{S}^2$, the 2-sphere. Equations (16) admit a spatial discretization called *consistent truncation* (see [11, 12]), which takes the form:

$$\begin{aligned} \dot{W} &= [P, W], \\ \Delta_N P &= W, \end{aligned} \tag{17}$$

where $P, W \in \mathfrak{su}(N)$, for $N = 1, 2, \ldots$ and for a suitable operator $\Delta_N : \mathfrak{su}(N) \to \mathfrak{su}(N)$. Since Δ_N can be chosen to be invertible, we get that (17) are of the form (2), with $\mathcal{L}W = \Delta_N^{-1} W$. Equations (17) are also a Lie–Poisson system, hence the scheme (4) is well-suited to retain its qualitative properties [4]. Clearly, in order to get a good approximation of the equations (16), we have to take N large (at least around 10^3, see [7]).

In Table 1 we show the performances of the different schemes proposed in the previous section. We notice that for large N the linear scheme performs somewhat better than the inexact Newton one in terms of CPU time. However, the most efficient scheme is the explicit fixed point.

4.2 Drift-Alfvén Plasma Model

The Drift-Alfvén plasma model [6] can be formulated in terms of the so called generalized vorticities $\omega_\pm$, ω_0. Although these are not directly physical quantities, they are a linear combinations of the generalized parallel momentum and the plasma density. In particular,

Table 1 Solution of the Euler equation on the sphere. CPU time and number of iterations of the two proposed schemes, for time-step $h = 0.5$ and normalized randomly generated initial values

N	Explicit fixed point		Linear scheme		Newton	
	Iter	CPU time	Iter	CPU time	Iter	CPU time
3	12	0.0008	11	0.0009	6.9	0.0020
5	9.5	0.0006	8.9	0.0005	5.5	0.0014
9	9.9	0.0006	8.3	0.0008	5.8	0.0016
17	11	0.0010	8.7	0.0021	6.1	0.0025
33	7.3	0.0022	6.5	0.0027	4.4	0.0037
65	7.8	0.0058	6.8	0.0053	4.8	0.0090
129	6.2	0.0125	5.6	0.0151	3.9	0.0229
257	7.2	0.0447	6.4	0.0571	4.4	0.0792
513	6.3	0.2020	5.8	0.2613	4.1	0.3836
1025	7.3	1.6191	6.3	1.9520	4.4	3.3740

neglecting the third order non-linearities, and absorbing the parameters in the time variable, we get the following equations:

$$\begin{aligned} \dot{\omega}_\pm &= \{\Phi_\pm, \omega_\pm\}, \\ \dot{\omega}_0 &= \{\Phi, \omega_0\}, \end{aligned}$$

$$\begin{aligned} \Phi_\pm &= \Phi \pm \frac{1}{\lambda}\Psi, \\ \Delta\Phi &= \omega_+ + \omega_- + \omega_0, \\ \Delta\Psi - \frac{1}{\lambda^2}\Psi &= \frac{1}{\lambda}\omega_+ - \frac{1}{\lambda}\omega_-, \end{aligned} \tag{18}$$

where λ is the ratio between the electron inertial skin depth and the ion sound gyroradius [6]. Analogously to the Euler equations, equations (18) have a matrix representation in $\mathfrak{su}(N) \times \mathfrak{su}(N) \times \mathfrak{su}(N)$, for any $N \geq 1$. If we assume $\omega_0 = 0$ (which physically means excluding the electrostatic drift vortices), we get:

$$\dot{\omega}_\pm = \{\Phi_\pm, \omega_\pm\},$$

$$\begin{aligned} \Phi_\pm &= \Phi \pm \frac{1}{\lambda}\Psi, \\ \Delta\Phi &= \omega_+ + \omega_-, \\ \Delta\Psi - \frac{1}{\lambda^2}\Psi &= \frac{1}{\lambda}\omega_+ - \frac{1}{\lambda}\omega_-. \end{aligned}$$

With the same notation of the previous section, we have the matrix equations:

$$\dot{W}_\pm = [F_\pm, W_\pm],$$

$$\begin{aligned} F_\pm &= F \pm \frac{1}{\lambda}P, \\ \Delta_N F &= W_+ + W_-, \\ \Delta_N P - \frac{1}{\lambda^2}P &= \frac{1}{\lambda}W_+ - \frac{1}{\lambda}W_-. \end{aligned} \tag{19}$$

Equations (19) can be cast in the form (2) in $\mathfrak{su}(N) \oplus \mathfrak{su}(N)$, for $\mathcal{L} : \mathfrak{su}(N) \oplus \mathfrak{su}(N) \to \mathfrak{su}(N) \oplus \mathfrak{su}(N)$, defined by

$$\mathcal{L}(W_+, W_-) = \left(\Delta_N^{-1}(W_+ + W_-), \quad \frac{1}{\lambda}\left(\Delta_N - \frac{1}{\lambda^2}\right)^{-1}(W_+ - W_-) \right).$$

In Table 2, the performances of the three algorithms applied component-wise for the Drift-Alfvén model are reported. Analogously to the Euler equations, the linear scheme performs a bit better than the inexact Newton scheme, especially for large matrices. As in the previous example, the most efficient scheme is the explicit fixed point.

4.3 Heisenberg Spin Chain

The Heisenberg spin chain is a conservative model of spin particle dynamics. This model arises from the spatial discretization of the Landau–Lifshitz–Gilbert Hamiltonian PDE [5]:

$$\partial_t \sigma = \sigma \times \partial_{xx}\sigma, \tag{20}$$

Table 2 Solution of the Drift-Alfvén model. CPU time and number of iterations of the two proposed schemes, for time-step $h = 0.5$, $\lambda = 5$, and normalized randomly generated initial values

N	Explicit fixed point		Linear scheme		Newton	
	Iter	CPU time	Iter	CPU time	Iter	CPU time
3	12	0.0013	11	0.0014	6.8	0.0024
5	9.6	0.0017	8.6	0.0020	5.6	0.0042
9	11	0.0023	9.2	0.0036	6.8	0.0045
17	8.4	0.0035	7.7	0.0040	5.1	0.0055
33	6	0.0055	5.6	0.0055	3.7	0.0081
65	6.3	0.0121	5.5	0.0157	3.9	0.0223
129	6.4	0.0502	5.9	0.0714	4	0.1019
257	5.6	0.2388	5.2	0.3207	3.7	0.4853
513	5.8	1.3479	5.2	1.6521	3.6	2.6889
1025	7	10.3985	6.1	12.2737	4.3	21.5882

where $\sigma : \mathbb{R} \times \mathbb{S}^1 \to \mathbb{S}^2$ is a closed smooth curve. Each value taken by σ in $\mathbb{S}^2$ represents a spin of an infinitesimal particle. We notice that unlike the previous examples of hyperbolic PDEs, (20) is a parabolic PDE. In Tables 3 and 4, we see that this requires a smaller time-step in order to have a comparable number of iterations of the two schemes as for the previous two examples.

Discretizing $\sigma \approx \{s_i\}_{i=1}^{N+1}$ on an evenly spaced grid with step size Δx of $\mathbb{S}^1$, $\{x_i\}_{i=1}^{N+1}$, with the conditions $s_{N+1} = s_1$ and $x_{N+1} = x_1$, we obtain

$$\partial_{xx}\sigma(x_i) \approx \frac{s_{i-1} - 2s_i + s_{i+1}}{\Delta x^2}.$$

Table 3 Solution of Heisenberg spin chain model. CPU time and number of iterations of the two proposed schemes, for time-step $h = 0.5$ and normalized randomly generated initial values

N	Explicit fixed point		Linear scheme		Newton	
	Iter	CPU time	Iter	CPU time	Iter	CPU time
3	62	0.0022	24	0.0013	31	0.0041
5	151	0.0039	23	0.0011	71	0.0071
9	NC	NC	24	0.0014	NC	NC
17	NC	NC	26	0.0018	NC	NC
33	NC	NC	28	0.0028	NC	NC
65	NC	NC	31	0.0047	NC	NC
129	NC	NC	30	0.0063	NC	NC
257	NC	NC	31	0.0121	NC	NC
513	NC	NC	32	0.0210	NC	NC
1025	NC	NC	33	0.0386	NC	NC

Table 4 Solution of Heisenberg spin chain model. CPU time and number of iterations of the two proposed schemes, for time-step $h = 0.1$ and normalized randomly generated initial values

N	Explicit fixed point		Linear scheme		Newton	
	Iter	CPU time	Iter	CPU time	Iter	CPU time
3	10	0.0008	9.2	0.0007	6.1	0.0014
5	13	0.0009	9.7	0.0008	7.3	0.0016
9	14	0.0011	9.9	0.0010	7.7	0.0016
17	14	0.0010	10	0.0013	7.6	0.0018
33	14	0.0011	10	0.0018	8.1	0.0021
65	15	0.0014	10	0.0019	8	0.0023
129	15	0.0019	10	0.0025	8	0.0031
257	15	0.0025	10	0.0042	8	0.0039
513	15	0.0046	11	0.0076	8.1	0.0058
1025	15	0.0093	11	0.0148	8.1	0.0136

Each spin vector s_i can be represented by a matrix S_i with unitary norm in $\mathfrak{su}(2) \cong \mathbb{R}^3$. The equations of motion are given by:

$$\partial_t S_i = \left[S_i, \frac{S_{i-1} + S_{i+1}}{\Delta x^2} \right],$$

for $i = 1, \dots N$. Hence, each spin interacts only with its neighbours (which explains the chain name). Hence, the matrices involved remain very sparse. A chain of N particles is an Hamiltonian system in $\mathfrak{su}(2)^N$, with Hamiltonian given by:

$$H(S_1, \dots, S_N) = \frac{1}{\Delta x^2} \sum_{i=1}^{N} \mathrm{Tr}(S_i^* S_{i+1}).$$

The operator $\mathcal{L} : \mathfrak{su}(2)^N \to \mathfrak{su}(2)^N$ is defined by

$$\mathcal{L}(S_1, S_2, \dots, S_N) = (S_N + S_2, S_1 + S_3, \dots, S_{N-1} + S_1).$$

In Tables 3 and 4, the results for the three algorithms applied component-wise are reported. In the numerical simulations, we set $\Delta x = 1$. We observe that both the explicit fixed point scheme and the inexact Newton method do not converge for spin-systems with many particles and large time-step $h = 0.5$. On the other hand, the linear scheme does converge for any $N \leq 2^{10} + 1$, making it more suitable for long time simulations. For smaller time-step $h = 0.1$ the three algorithms perform almost equally well.

5 Conclusions

In this paper we have proposed and investigated some iterative schemes for the solution of cubic matrix equations arising in the numerical solution of certain conservative PDEs by means of (Lie–Poisson) geometrical integrators. These types of schemes enable the preservation of important physical features of the original infinite-dimensional flows, which is generally not the case when more standard discretizations and numerical integrators are used. Both the fixed point iterations and the inexact Newton type scheme we have investigated tend to work well, but we found that the fixed point method which requires the

solution of a linear matrix equation is the most robust with respect to the time step and requires a comparable CPU time with the fully explicit scheme.

Acknowledgements The authors would like to thank the anonymous referees for their useful comments and suggestions. Special thanks to prof. Bruno Iannazzo for his observations on the manuscript and for pointing out reference [1].

References

1. Bankmann, D., Mehrmann, V., Nesterov, Y., Van Dooren, P.: Computation of the analytic center of the solution set of the linear matrix inequality arising in continuous- and discrete-time passivity analysis. Vietnam J. Math. **48**, 633–659 (2020)
2. Bini, D., Iannazzo, B., Meini, B.: Numerical Solution of Algebraic Riccati Equations. SIAM, Philadelphia (2012)
3. Gohberg, I., Lancaster, P., Rodman, L.: Invariant Subspaces of Matrices with Applications. SIAM, Philadelphia (2006)
4. Hairer, E., Lubich, C., Wanner, G.: Geometric Numerical Integration. Springer, Berlin Heidelberg (2006)
5. Lakshmanan, M.: The fascinating world of the Landau–Lifshitz–Gilbert equation: an overview. Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. **369**, 1280–1300 (2011)
6. Mentink, J.H., Bergmans, J., Kamp, L.P.J., Schep, T.J.: Dynamics of plasma vortices: the role of the electron skin depth. Phys. Plasmas **12**, 052311 (2005)
7. Modin, K., Viviani, M.: A Casimir preserving scheme for long-time simulation of spherical ideal hydrodynamics. J. Fluid Mech. **884**, 22 (2020)
8. Simoncini, V.: Computational methods for linear matrix equations. SIAM Rev. **58**, 377–441 (2016)
9. Sylvester, J.: Sur l'équations en matrices $px = xq$. C. R. Acad. Sci. Paris. **99**, 67–71 (1884)
10. Viviani, M.: A minimal-variable symplectic method for isospectral flows. BIT Num. Math. **60**, 741–758 (2020)
11. Zeitlin, V.: Finite-mode analogues of 2D ideal hydrodynamics: coadjoint orbits and local canonical structure. Phys. D **49**, 353–362 (1991)
12. Zeitlin, V.: Self-consistent finite-mode approximations for the hydrodynamics of an incompressible fluid on nonrotating and rotating spheres. Phys. Rev. Lett. **93**, 264501 (2004)