SIERE: a hybrid semi-implicit exponential integrator for efficiently simulating stiff deformable objects

YU JU (EDWIN) CHEN, University of British Columbia
SEUNG HEON SHEEN, University of British Columbia
URI M. ASCHER, University of British Columbia
DINESH K. PAI, University of British Columbia and Vital Mechanics Research

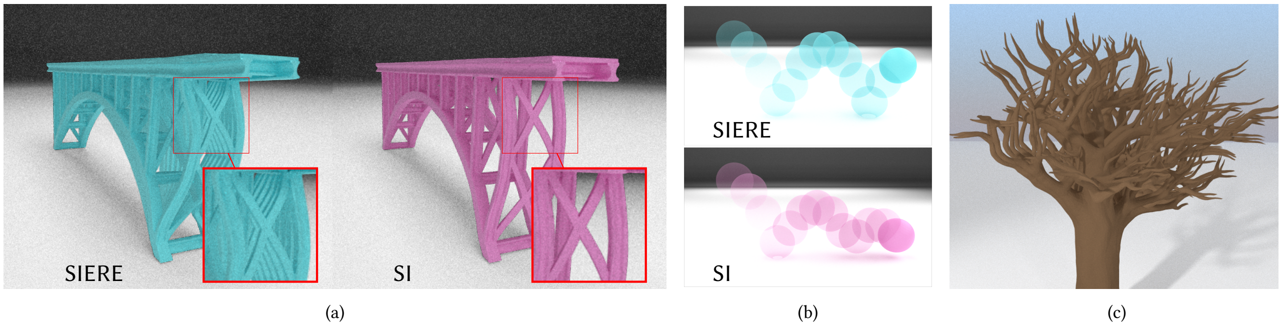


Fig. 1. We propose a new time integrator for elastodynamic simulations that can efficiently handle very stiff and nonlinear constitutive material models while maintaining good stability and avoiding unwanted visual artifacts. No solution of nonlinear algebraic equations is needed. A stiff bridge (a) modeled with nearly 100K tetrahedra exhibits both slow oscillations that are visually very important and a huge number of high frequency oscillations. The semi-implicit backward Euler (SI) integrator can excessively damp out the important scale motions. Our method, SIERE, can integrate such systems effectively. (b) SIERE (cyan) also has better damping behavior than SI (pink) upon collision. (c) SIERE can capture salient features of complex objects efficiently.

Physics-based simulation methods for deformable objects suffer limitations due to the conflicting requirements that are placed on them. The work horse semi-implicit backward Euler method (SI) is very stable and inexpensive, but it is also a blunt instrument. It applies heavy damping, that depends on the time step, to all solution modes and not just to high frequency ones, and as such makes simulations less lively, potentially missing important animation details. At the other end of the scale, exponential methods (ERE) are known to deliver good approximations to all modes, but they get prohibitively expensive and less stable for very stiff material. In this paper we devise a hybrid, semi-implicit method called SIERE that allows the previous methods SI and ERE to each perform what they are good at. To do this we employ at each time step a partial spectral decomposition which picks the lower, leading modes, applying ERE in the corresponding subspace. The rest is handled (i.e., effectively damped out) by SI. No solution of nonlinear algebraic equations is required throughout the algorithm. We show that the resulting method produces simulations that are visually as good as those of the exponential method at a computational price that does not increase with stiffness, while displaying stability and damping with respect to the high frequency modes. Furthermore, the phenomenon of occasional divergence of SI is avoided.

CCS Concepts: • **Computing methodologies** → **Physics simulation**.

Additional Key Words and Phrases: Physically-based simulation, deformable object, time integration, stiffness, nonlinear constitutive material

ACM Reference Format:
Yu Ju (Edwin) Chen, Seung Heon Sheen, Uri M. Ascher, and Dinesh K. Pai. 2020. SIERE: a hybrid semi-implicit exponential integrator for efficiently simulating stiff deformable objects. *ACM Trans. Graph.* 1, 1 (July 2020), 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Physics-based simulations of deformable objects are ubiquitous in computer graphics today. They arise in various applications including animation, robotics, control and fabrication. For almost two decades the semi-implicit backward-Euler (SI) method of Baraff and Witkin [1998], has been widely employed. This method allows for stable simulations even when large time steps are used for efficiency reasons, and it is very stable when incorporating contacts and collisions due to its heavy damping and error localization properties. Moreover, numerical damping is in agreement with the observation that our visual system does not detect high frequency vibrations, even when objects with large Young's modulus are simulated. However, recent years have also seen growing concerns that heavy numerical damping may be unsuitable for many purposes, because it can only be controlled via the time step size, so it does not distinguish phenomena related to material heterogeneity and more complex damping forces [Chen et al. 2018; Chung and Hulbert 1993].

Authors' addresses: Yu Ju (Edwin) Chen, University of British Columbia, aq978063@cs.ubc.ca; Seung Heon Sheen, University of British Columbia, shsheen@cs.ubc.ca; Uri M. Ascher, University of British Columbia, ascher@cs.ubc.ca; Dinesh K. Pai, University of British Columbia and Vital Mechanics Research, pai@cs.ubc.ca.

0730-0301/2020/7-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Use of the SI method in applications such as control and fabrication has also been considered debatable [Chen et al. 2017]. A further concern is that SI has long been known to occasionally diverge wildly where the fully implicit backward Euler (BE) still yields acceptable results. This phenomenon, demonstrated in Sections 4.1 and 4.2, is due to divergence in low frequency modes.

Exponential time integration methods have been introduced relatively recently. These methods are attractive because, although non-conservative, they produce only little damping and approximate the entire modal spectrum acceptably well. These methods, like SI and unlike the full BE and various conservative methods, avoid the need of solving nonlinear algebraic equations at each time step [Chen et al. 2018; Michels et al. 2017]. However, their performance cost becomes prohibitive when the simulated system of differential equations in time is stiff, which happens for large Young's modulus values or upon using a fine spatial discretization. See Figure 2 and Table 2.

In this paper we combine the exponential Rosenbrock Euler (ERE) integrator described in [Chen et al. 2018] with the SI method in a manner that allows each to concentrate on what it is good at and improve upon the other method's deficiencies. The result is a semi-implicit integrator that outperforms both its predecessors, especially for stiff problems: it produces animations that are similar to the exponential integrator's at a computational cost that does not increase as the simulated object gets stiffer.

The key idea is to consider, at each time step, the appropriately transformed assembly of forces in the equations of motion as a weighted sum of many high frequency modes that are dealt with well by the heavily damping SI method, and a few low-to-medium frequency modes (which are the ones that contribute most to our visual perception [Chen et al. 2019]) that are dealt with efficiently by ERE. No nonlinear algebraic equations arise, even in the presence of nonlinear elastic forces. To do this we employ at each time step a partial spectral decomposition which computes the lower, leading modes, and apply ERE in the corresponding subspace. The rest is handled (i.e., damped out) by SI. We call the resulting method SIERE. To summarize, the advantages of the new method are:

- It produces lively animations of a similar quality to that of the exponential method ERE; the results are better than SI both in staying closer to the simulated energy manifold and in avoiding over-smoothing of artifacts such as material heterogeneity and secondary motion.
- It handles contacts and collisions, inheriting this property from SI.
- The computational cost of SIERE, unlike that of ERE, remains a small multiple of SI cost as the stiffness increases.
- As the exponential part of the method is performed only in a small spectrally-decomposed subspace, the significant cost of exponentiating a large matrix is avoided altogether.
- Potential wild divergence of SI (which arises mostly when relatively soft material stiffens under large deformation) is avoided by SIERE.

We demonstrate the performance and utility of SIERE on several challenging examples.

2 CONTEXT AND RELATED WORK

Equations of motion: A common approach for discretizing an elastodynamic system is to first apply a finite element method (FEM) in space, already at the variational level. This spatial discretization results in a large ODE system in time t, written in standard notation as

$$M\ddot{\mathbf{q}} = \mathbf{f}_{\text{tot}}(\mathbf{q}, \mathbf{v}), \tag{1a}$$

where the unknowns $\mathbf{q} = \mathbf{q}(t)$ are nodal displacements in the FEM mesh with corresponding nodal velocities $\mathbf{v}(t) = \dot{\mathbf{q}}(t)$. The mass matrix M is symmetric positive definite and sparse. The total force is further written as

$$\mathbf{f}_{\text{tot}} = \mathbf{f}_{\text{els}}(\mathbf{q}) + \mathbf{f}_{\text{dmp}}(\mathbf{q}, \mathbf{v}) + \mathbf{f}_{\text{ext}}(\mathbf{q}). \tag{1b}$$

A similar-looking problem arises upon using a mass-spring system [Baraff and Witkin 1998; Boxerman and Ascher 2004], though the spectrum of the corresponding stiffness matrix is less challenging to split in that case.

The elastic and damping forces are written as

$$\mathbf{f}_{\text{els}}(\mathbf{q}(t)) = -\frac{\partial}{\partial \mathbf{q}} W(\mathbf{q}(t)), \ \mathbf{f}_{\text{dmp}}(\mathbf{v}(t)) = -D\mathbf{v}(t),$$

where $W(\mathbf{q}(t))$ is the elastic potential of the corresponding hyperelasticity model. We further define the tangent stiffness matrix $K = -\frac{\partial}{\partial \mathbf{q}} \mathbf{f}_{\text{els}}(\mathbf{q}(t))$. This matrix is constant and symmetric positive definite if $\mathbf{f}_{\text{els}}$ is linear, but for nonlinear elastic forces it depends on the unknown $\mathbf{q}(t)$ and may occasionally become indefinite. The damping matrix $D = D(\mathbf{q}(t), \mathbf{v}(t))$ is symmetric nonnegative definite at all times. See, for instance, [Barbic and James 2005; Chen et al. 2017, 2018; Ciarlet 1988; Sifakis and Barbic 2012].

The ODE system in Eq. (1) can be written as a first-order system

$$\dot{\mathbf{u}}(t) \equiv \begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}D \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \mathbf{v} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{g} \end{bmatrix} \equiv \mathbf{F}(\mathbf{u}). \tag{2}$$

Let us denote the (often large) length of $\mathbf{u}$ by n.

Numerical damping: Eq. (2) must be discretized before simulation, but as stated above, the popular SI method and even fully implicit integrators such as BE and higher order backward differentiation formulae (BDF) introduce significant, step-size dependent, artificial damping. All BDF methods collocate the ODE system only at the unknown time level [Ascher 2008]. This typically yields a potentially heavy damping of high frequency modes that has often been observed in practice.

To get an indication of this we extend in an Appendix the artificial damping analysis of [Chen et al. 2018] that was applied to the backward Euler (BE) method, which is the one-step BDF, to the two-step method BDF2. In particular, see Figures 18 and 19.

For linear problems, the methods SI and BE coincide. Further, in the stiff context a semi-implicit method such as SI cannot have an order of accuracy higher than 1. Replacing a nonlinear solver by such a linearization in BDF2 therefore reduces the resulting method's order of accuracy from 2 to 1.

Exponential integrators: Several exponential integration schemes have been proposed in the computer graphics literature [Chen et al. 2018; Michels et al. 2017; Michels and Mueller 2016; Michels et al.

2015, 2014]. In principle they all involve calculations of the action of the matrix exponential of $M^{-1}K$ on various vectors. If the tangent stiffness matrix is known to be always positive definite, especially in case it is constant, then efficient methods using its square root matrix may be devised. However, for nonlinear forces that appear in practice there is the need to exponentiate at every time step, since K depends on $\mathbf{q}(t)$, and there is no such guarantee of positive definiteness. The ERE method of [Chen et al. 2018] is particularly suitable for nonlinear forces in such challenging situations, and its nominal order is commensurate with that of the BE method, so we proceed with it.

Stepping from time t, where the approximate solution u is known, to the next time $t_+ = t + h$, the ERE method for the ODE Eq. (2) at each time step is given by

$$\mathbf{u}_+ = \mathbf{u} + h\phi_1(hJ)\mathbf{F}(\mathbf{u}),$$

where J is the Jacobian matrix of $\mathbf{F}$, and $\phi_1(Z) = Z^{-1}(\exp(Z) - I)$ for a given square matrix Z (with I the identity of the same size).

The explicit formation of the exponential matrix appearing in ϕ_1 is out of the question for large matrices J such as we have for Eq. (2), as this matrix is full even though K, M and D are all sparse. Researchers resort instead to Krylov subspace methods for approximating the product of $\exp(hJ)$ with vectors $\mathbf{F}$ [Al-Mohy and Higham 2011; Moler and Van Loan 2003; Niesen and Wright 2012]. However, as shown in Figure 2 and Table 2, these methods become expensive for stiff problems, thus limiting the utility of exponential integrators in the context of deformable object simulations [Chen et al. 2018; Michels et al. 2017]. The root of this problem is in the fact that the required number of Krylov vectors needed in order to fully resolve the error for a stiff system as in Eq. (2) with a wide spectrum matrix and large $\mathbf{F}(\mathbf{u})$ can be very large. One might consider choosing a smaller time step to assist the matrix function evaluation for stiff objects; however, in physics-based simulations to stay competitive we expect to be able to keep employing large time steps independent of the material parameters. The method described in Section 3 alleviates this difficulty by applying the exponential integrator only in a suitable subspace, and the exponential matrix evaluation at each time step is simplified due to diagonalization.

Additive methods: If $\mathbf{F}(\mathbf{u})$ can be written for each time instance t as a sum of two parts

$$\mathbf{F}(\mathbf{u}) = \mathbf{H}(\mathbf{u}) + \mathbf{G}(\mathbf{u}), \tag{3}$$

then it is possible to write $\mathbf{u}(t) = \mathbf{x}(t) + \mathbf{y}(t)$, where $\mathbf{x}$ and $\mathbf{y}$ satisfy

$$\dot{\mathbf{x}} = \mathbf{H}(\mathbf{u}) = \mathbf{H}(\mathbf{x} + \mathbf{y}), \tag{4a}$$

$$\dot{\mathbf{y}} = \mathbf{G}(\mathbf{u}) = \mathbf{G}(\mathbf{x} + \mathbf{y}). \tag{4b}$$

We can therefore apply one integration scheme to Eq. (4a) and another one to Eq. (4b), add them up and obtain a combined integration scheme for Eq. (2) [Ascher 2008; Luan et al. 2017].

A popular class of additive methods is the IMEX class, where an implicit discretization scheme is applied to Eq. (4a) and an explicit one to Eq. (4b) [Ascher et al. 1995; Boxerman and Ascher 2004; Eberhardt et al. 2000]. This makes sense if the first term in Eq. (3) is stiff, while the second is not. Our additive method is similar in spirit to IMEX, although strictly speaking neither of its components is a fully implicit solver and both rely on solving a system of linear equations at each time step. The key is in determining a suitable splitting $\mathbf{G}$ and $\mathbf{H}$, described in the next section. Our spectral splitting allows us to achieve results in the graphics context that in general cannot be realized by the other additive methods cited above.[1]

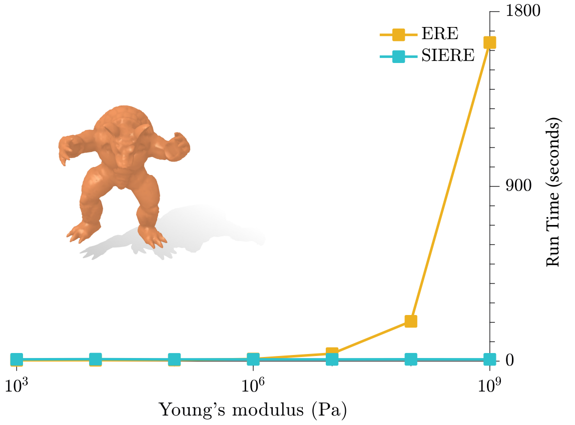


Fig. 2. Exponential integrators (e.g., ERE) can better preserve the oscillations, but at a prohibitive cost as stiffness parameter and/or system size increases. By contrast, the cost of our method (SIERE) does not grow significantly with stiffness.

If a fully implicit method is employed, as is required for large-step conservative methods (see, e.g., [Chen et al. 2017; Kharevych et al. 2006]) and for higher order BDF methods such as BDF2, then a nontrivial nonlinear system of algebraic equations must be solved at each time step; see [Bouaziz et al. 2014; Dinev et al. 2018a,b; Li et al. 2019; Su et al. 2013] and further references therein. The difficulty in solving such algebraic equations can increase when simulating objects with very stiff components, as in several of our examples; some of the efforts to design robust nonlinear solvers give rise to rather elaborate schemes and packages. Note further that in general energy projection methods cannot be symplectic or even reversible. None of these methods are claimed to be faster or simpler than SI per step. SIERE elegantly avoids this significant and time consuming issue by using an inexpensive semi-implicit method throughout, resulting in a practical, fast and powerful method that is derived from solid, clear principles and can be easily incorporated to fit one's specific needs and code.

3 METHOD

An additive method: Assuming that we can write the equations of motion in the form Eq. (3), we consider first the additive method

$$\mathbf{u}_+ = \mathbf{u} + h\mathbf{H}(\mathbf{u}_+) + h\phi_1(hJ_G)\mathbf{G}(\mathbf{u}),$$

[1] It is noteworthy that in the larger context of highly oscillatory dynamical systems in science there are significant challenges that remain unresolved even after decades of intense work. A famous example is the general *molecular dynamics* setting. In fact, researchers in that context appear to have stopped designing large-step integration methods. Any general additive method is highly unlikely to measure up to such a challenge, because no adequate separation of scales is automatically implied. Fortunately, in the graphics context of simulating deformable objects there is a very large class of problems where the observed motion is determined by the first few modes, and this is what we capitalize upon here.

where J_G is the Jacobian of the yet unspecified $\mathbf{G}(\mathbf{u})$. This combines the ERE method with the backward Euler (BE) method. One advantage of this is that calculating the ERE term first provides a "warm start" for the ensuing solution of the BE part.

Furthermore, if we want to avoid solving nonlinear equations then we can perform a single Newton iteration for $\mathbf{H}$ at each time step, replacing BE by SI. Thus, approximating $\mathbf{H}(\mathbf{u}_+) \approx \mathbf{H}(\mathbf{u}) + J_H(\mathbf{u}_+ - \mathbf{u})$, we obtain our proposed SIERE method as

$$\mathbf{u}_+ = \mathbf{u} + (I - hJ_H)^{-1}(h\mathbf{H}(\mathbf{u}) + h\phi_1(hJ_G)\mathbf{G}(\mathbf{u})). \tag{5}$$

Let us note in passing that, since this stable additive method consists of a combination of two first order methods, it converges like $O(h)$ as the step size h shrinks to 0. Our practical focus, however, remains on large time steps.

3.1 Model reduction and subspace splitting

Next, we define the splitting $\mathbf{G}$ and $\mathbf{H}$, crucial to the success of our method. The idea is to apply ERE in the subspace of the first s modes ($s \ll n$: typically, $5 \leq s \leq 20$) and project it back to the original full space. In the bridge example of Figure 1, $n \approx 100,000$, so this is a rather large reduction.

Then we use SI on the remaining unevaluated part, as per Eq. (5). We use mass-PCA to find our reduced space. That is, considering at the beginning of each time step a solution mode of the form $\mathbf{q}(t) = \mathbf{w}\exp(\imath\sqrt{\lambda}t)$ for the ODE $M\ddot{\mathbf{q}} + K\mathbf{q} = \mathbf{0}$, we solve the generalized eigenvalue problem

$$K\mathbf{w} = \lambda M\mathbf{w}$$

for the s smallest eigenvalues λ and their corresponding eigenvectors $\mathbf{w}$ (dominant modes). In MATLAB, for instance, this can be achieved by calling the function `eigs`. In our implementation, we use the C++ Spectra library [Qiu 2019]. Denote this partial spectral decomposition by

$$KU_s = MU_s\Lambda_s, \tag{6}$$

where the "long and skinny" U_s is $n \times s$, has the first s eigenvectors $\mathbf{w}$ as its columns, and the small Λ_s is a diagonal $s \times s$ matrix with the eigenvalues $\lambda_1, ..., \lambda_s$ on the diagonal. Notice that both K and M are large sparse symmetric matrices. In addition, M is positive definite, so U_s has *M-orthogonal* columns:

$$U_s^T M U_s = I_s, \quad U_s^T K U_s = \Lambda_s. \tag{7}$$

Next, we write Eq. (2) in the split form Eq. (3), with the splitting $\mathbf{H}$ and $\mathbf{G}$ defined based on the partial spectral decomposition Eq. (6). We define at each time step

$$\begin{aligned} \mathbf{G}(\mathbf{u}) &= \begin{bmatrix} \mathbf{v}_G \\ M^{-1}\mathbf{f}_G \end{bmatrix}, \quad \mathbf{H}(\mathbf{u}) = \begin{bmatrix} \mathbf{v}_H \\ M^{-1}\mathbf{f}_H \end{bmatrix}, \\ \mathbf{v}_G &= U_s U_s^T M\mathbf{v}, \quad \mathbf{v}_H = \mathbf{v} - \mathbf{v}_G, \\ \mathbf{f}_G &= MU_s U_s^T \mathbf{f}, \quad \mathbf{f}_H = \mathbf{f} - \mathbf{f}_G. \end{aligned} \tag{8}$$

We also need the Jacobian matrices

$$J_G = \begin{bmatrix} 0 & U_s U_s^T M \\ -U_s U_s^T K U_s U_s^T M & 0 \end{bmatrix}, \quad \text{and} \tag{9a}$$

$$J_H = \begin{bmatrix} 0 & I \\ -M^{-1}K & 0 \end{bmatrix} - J_G. \tag{9b}$$

Notice that the ERE expression, $h\phi_1(hJ_G)\mathbf{G}(\mathbf{u})$, can be evaluated in the subspace first, and then projected back to the original space.

The additive method defined by inserting Eqs. (8) and (9) into Eq. (5) has three advantages:

1. At each time step, the majority of the update comes from ERE in the dominating modes. Thus it is less affected by artificial damping from SI.
2. The computation load of ERE is greatly reduced, because the stiff part is handled by SI (or BE for that matter). Furthermore, the evaluation of the exponential function in the subspace has only marginal cost since the crucial matrix involved has been diagonalized.
3. The "warm start" for SI makes its result closer to that of BE.

ERE update in the subspace: To evaluate the update in the subspace of dimension s we rewrite Eq. (5) as

$$\begin{aligned} \mathbf{u}_+ &= \mathbf{u} + (I - hJ_H)^{-1}\left(h\mathbf{H}(\mathbf{u})\right. \\ &+ \left. h\begin{bmatrix} U_s & 0 \\ 0 & U_s \end{bmatrix} \phi_1(hJ_G^r)\mathbf{G}^r(\mathbf{u})\right), \end{aligned} \tag{10}$$

where

$$\begin{aligned} J_G^r &= \begin{bmatrix} 0 & I \\ -U_s^T K U_s & 0 \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\Lambda_s & 0 \end{bmatrix}, \\ \mathbf{G}^r(\mathbf{u}) &= \begin{bmatrix} U_s^T M\mathbf{v} \\ U_s^T\mathbf{f} \end{bmatrix}. \end{aligned} \tag{11}$$

The evaluation of the action of the matrix function ϕ_1 involves only matrices of size $2s \times 2s$. Furthermore, the matrix function ϕ_1 can be evaluated directly through the eigenpairs of J_G^r

$$\left\{\imath\sqrt{\lambda_l}, \quad \begin{bmatrix} \mathbf{e}_l \\ \imath\sqrt{\lambda_l}\mathbf{e}_l \end{bmatrix}\right\}, \quad \left\{-\imath\sqrt{\lambda_l}, \quad \begin{bmatrix} -\mathbf{e}_l \\ \imath\sqrt{\lambda_l}\mathbf{e}_l \end{bmatrix}\right\}, \quad l = 1, ..., s, \tag{12}$$

with $\mathbf{e}_l$ being the l^{th} column of the identity matrix.

The large $n \times n$ linear system solved in Eq. (10) is not sparse due to the fill-in introduced by the small subspace projection. Specifically, the off-diagonal blocks of the Jacobian matrix J_G defined in Eq. (9a) are not sparse. If not treated carefully, solving the linear system in Eq. (5) and Eq. (10) can be prohibitively costly. Fortunately, this modification matrix has the low rank s. We can write

$$J_G = Y_1 Z_1^T + Y_2 Z_2^T,$$

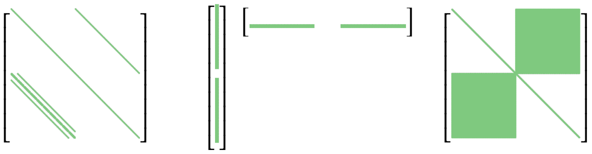
where

$$\begin{aligned} Y_1 &= \begin{bmatrix} U_s \\ 0 \end{bmatrix}, \quad Z_1 = \begin{bmatrix} 0 \\ MU_s \end{bmatrix}, \\ Y_2 &= \begin{bmatrix} 0 \\ -U_s U_s^T K U_s \end{bmatrix}, \quad Z_2 = \begin{bmatrix} MU_s \\ 0 \end{bmatrix}. \end{aligned}$$

The linear system in Eq. (10) becomes

$$I - hJ_H = (I - hJ) + hY_1 Z1^T + hY_2 Z_2^T, \tag{13}$$

where the four matrices Y_i and Z_i are all "long and skinny" like U_s, while the matrix J is square and large, but very sparse. Figure 3 illustrates this situation. For the linear system to be solved in Eq. (10) we may employ an iterative method such as conjugate gradient, whereby the matrix-vector products involving J or $Y_i Z_i^T$ are all straightforward to carry out efficiently. However, we have often found out that a direct solution method is more appropriate for



(a) $I - hJ$ (b) Y_i and Z_i^T (c) $I - hJ_H$

Fig. 3. The matrix $I - hJ_H$ in the linear system eq. (10) is not sparse (c). Fortunately, by Eq. (13) the fill-in to the original sparse matrix $I - hJ$ (a) has low rank (b) allowing us to use the SMW formula Eq. (14).

these linear equations in our context. In our implementation we use `pardiso` [De Coninck et al. 2016; Kourounis et al. 2018; Verbosio et al. 2017]. For this we can employ the formula of Sherman, Morrison and Woodbury (SMW) [Nocedal and Wright 2006], given by

$$(A + YZ^T)^{-1} = A^{-1} - A^{-1}Y(I + Z^T A^{-1}Y)^{-1}Z^T A^{-1}, \tag{14}$$

to solve the linear system in Eq. (10). In our specific notation we set at each time step $A = I - hJ_H$ in Eq. (14), and apply the formula twice: once for $Y = Y_1$, $Z = Z_1$, and once for $Y = Y_2$, $Z = Z_2$. Note that the matrices $I + Z^T A^{-1} Y$ in Eq. (14) are only $2s \times 2s$, and this results in an efficient implementation, so long as the subspace dimension s remains small.

3.2 Time stepping cost vs subspace dimension

How does performance deteriorate as the subspace dimension s is increased for a fixed system size n? To enable discussion, let us restrict consideration to the range $0 < s < \sqrt{n}$, because our method as described makes no sense for larger s, and not just in terms of sparsity.[2] In this range, it is not difficult to see theoretically, by examining the various algebraic operations just described per time step, that the computation time for carrying out Eqs. (8)-(14) increases at worst linearly with s.

The theoretical cost of the partial spectral decomposition Eq. (6) involving a generalized eigenvalue problem is a more delicate matter; see [Saad 2011]. For our range of s it can grow slightly faster than linearly: see Figure 4. However, in our setting the cost of other tasks dominates. Indeed, the two major expenses at a given time step are the assembly of the sparse stiffness matrix K (hence J), and the solution of linear algebraic systems involving $I - hJ$, both of which are independent of s. In fact, by Eqs. (13)-(14) the latter is required $2s + 4$ times.

Beyond the general trend, precise runtimes depend heavily on the problem's characteristics. Figure 4 displays performance of the partial spectral decomposition using Matlab's sparse linear algebra packages for a very simple stiffness matrix in 2D. The results agree with the theoretical expectation. For a more quantitative feeling regarding the entire algorithm, we have observed for some of the very stiff examples described in the next section, which give a nonlinear solver much grief, that for $s = 50$ there is roughly a runtime cost increase by a factor of 5 compared to that of SI.

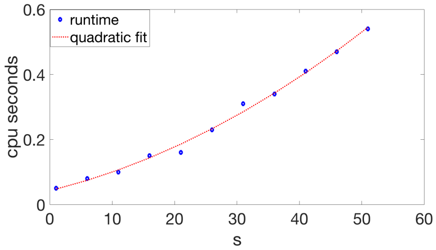


Fig. 4. Illustrating the cost of the partial spectral decomposition for a very simple stiffness matrix with $n = 20{,}000$ as s grows.

In Algorithm 1 we collect the pieces just derived.

ALGORITHM 1: SIERE step from t to $t + h$

```
Input: $\mathbf{u} = \begin{bmatrix}\mathbf{q}\\ \mathbf{v}\end{bmatrix}$, $\mathbf{f}$, $K$, $M$, $h$, $s$
;                                   ▷ $K$, $M$ symmetric, $M$ positive definite
begin
    Solve (6) for $U_s$, $\Lambda_s$
    Construct $\mathbf{H}$ ;                                            ▷ (8)
    Construct $J_G^r$, $\mathbf{G}^r$ ;                                 ▷ (11)
    Evaluate matrix function $\phi_1(hJ_G^r)$ ;                         ▷ use (12)
    $\delta \leftarrow h\mathbf{H} + h\begin{bmatrix}U_s & 0\\ 0 & U_s\end{bmatrix}\phi_1(hJ_G^r)\mathbf{G}^r$
    Construct $Y_1$, $Z_1$, $Y_2$, $Z_2$ ;                              ▷ (13)
    $\mathbf{x}_0 \leftarrow (I - hJ)^{-1}\delta$
    $\mathbf{x}_1 \leftarrow \mathbf{x}_0$ updated with $Y_1$, $Z_1$ ;  ▷ first (14)
    $\mathbf{x}_2 \leftarrow \mathbf{x}_1$ updated with $Y_2$, $Z_2$ ;  ▷ second (14)
    $\mathbf{u}_+ \leftarrow \mathbf{u} + \mathbf{x}_2$ ;               ▷ (10)
end
Output: $\begin{bmatrix}\mathbf{q}_+\\ \mathbf{v}_+\end{bmatrix} = \mathbf{u}_+$
```

4 RESULTS

Here we compare the performance of the new method SIERE against SI and ERE on several examples. Our purpose is to show that SIERE produces simulation results that are similar to those of ERE (and thus do not suffer from the excessive damping of SI), with an implementation cost that does not increase when the stiffness increases (unlike that of ERE). Furthermore, we show examples where SIERE overcomes occasional divergence and instability phenomena arising in SI and ERE, as well as the fully implicit BE. Below in Sections 4.1 and 4.2 we begin with the latter, as the examples used also allow us to investigate the effect of selecting the dimension s of the ERE update subspace. Later on we fix s and concentrate on the other aspects, comparing SIERE and SI simulation quality on larger and more geometrically complex objects where ERE performance is inadequate.

A list of meshes used and their sizes are summarized in Table 1. In Table 2 we list the CPU time for using the integrator for 100 frames. In our simulations we used a wide range of Young's moduli $\hat{Y}$ and a uniform Poisson's ratio $\nu = 0.45$. We also profiled the cost of the

[2] Indeed, let us explicitly discourage usage of our method for problems that require the subspace dimension s to be comparable in size to the ODE system size n. Fortunately, there are many deformable objects where s can be rather small, as demonstrated in Section 4.

Table 1. List of meshes used in representative experiments.

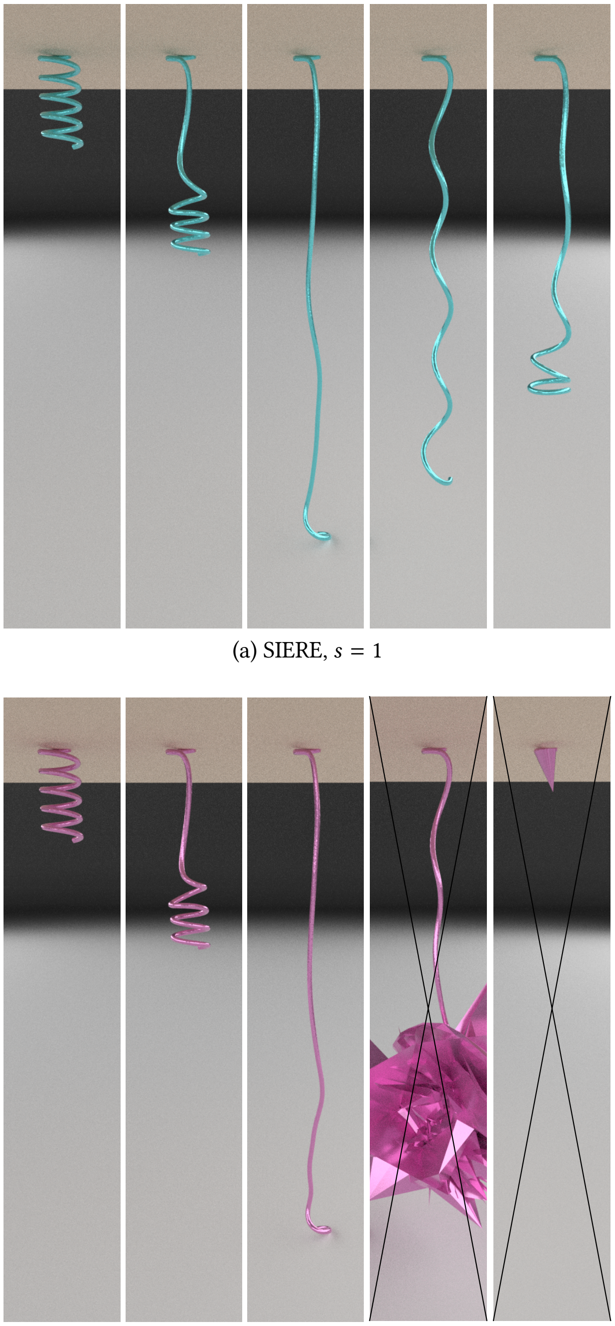
Mesh ID	#Vertices	#Tetrahedrons
Ball	1,018	4,637
Cuboid	1,245	4,624
Honeycomb	3,642	9,850
Moebius Ball	5,829	16,857
Rope	12,791	47,279
Eiffel Tower	16,027	69,271
Tree	24,533	79,217
Bridge	30,133	99,455
Human Body	40,540	189,021

Table 2. CPU Times (in seconds) for different methods, run on a core i7 machine with our C++ implementation. The reported stiffness is the peak stiffness of the object in Pa. The symbol ⊗ indicates unstable simulation, or divergence, the relevant method's timing becoming irrelevant. The symbol × indicates that the simulation took far more than 20 times that of SIERE and thus we stopped the program before the simulation ended. All the SIERE results are for $s = 5$, except the rope example where $s = 1$ and the human body example where $s = 6$.

Mesh	Stiffness	SI	SIERE	ERE
Ball	$1e8$	44	97	36
	$1e10$	42	96	112
Cuboid	$1e5$	⊗	110	⊗
Honeycomb	$1e8$	85	154	537
	$1e10$	84	156	5,840
Moebius Ball	$1e8$	162	223	5,780
	$1e10$	154	260	×
Rope	$1e6$	⊗	651	×
Eiffel Tower	$3e7$	631	2,315	×
Tree	$5e7$	⊗	2,690	×
Bridge	$1e10$	1,375	4,585	×
Human Body	$1.6e4$	5,209	9,672	79,367

linear algebra calculations for SIERE, and found that under 20% of the runtime was due to partial eigendecomposition in Eq. (6), while the rest came from solving the linear system with low-rank update in Eq. (10). Notice that after the eigendecomposition, the exponential matrix evaluation is automatically completed from Eq. (12) at no extra cost.

For the backward Euler (BE) nonlinear solver we simply employed Newton's method at each time step, declaring convergence when the residual's norm was below 10^{-6}. Failure was declared if there was no convergence after 20 iterations. In our code we also have the option of cutting the step size by 2 and repeating, which is a form of numerical continuation [Deuflhard 2011], but we opted not to use this or any other more sophisticated techniques, as such is not our focus here.



(a) SIERE, $s = 1$

(b) SI

Fig. 5. An uncoiling rope modelled with soft neo-Hookean material $\hat{Y} = 1e6\,Pa$. (a) SIERE (with subspace dimension $s = 1$) stays stable throughout the simulation, whereas (b) SI becomes unstable (diverges) when the rope stiffens at the end of the uncoiling process. This demonstrates the superior stability of SIERE.

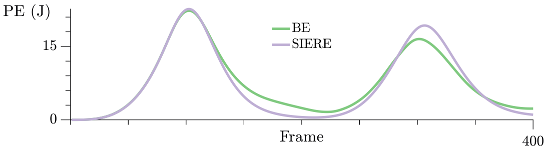


Fig. 6. Energy curves comparing SIERE and BE over 400 frames for the uncoiling rope.

4.1 Uncoiling rope

We start with an example to demonstrate the superior stability of SIERE arising from the "warm start" provided by ERE. In this example we simulate the process of uncoiling a soft homogeneous neo-Hookean rope (12,791 vertices and 47,279 tetrahedrons) with Young's modulus $\hat{Y} = 1e6\,Pa$ and using $h = 0.01s$ for the time step size in seconds. The magnitude of deformation increases as the rope uncoils, and so does the stiffness.

When the rope is fully uncoiled, the deformation is localized at the very tip of the rope, leading to a challenging numerical system. BE performed adequately here. However, SI quickly became unstable since one Newton iteration for BE could not adequately reduce the residual for the system, and this iteration divergence gave rise to a blowup. On the other hand, the semi-implicit system arising from SIERE is stable. To emphasize the point that the selection of s is based on geometry rather than stiffness considerations, we chose the smallest subspace dimension $s = 1$ for this example: the exponential integrator still provides enough input to stabilize the simulation; see Figure 5.

In addition, we also observed that SIERE is able to uncoil each loop independently. This means that SIERE can simulate local deformation very well, unlike traditional linear subspace methods where global deformation artifacts appear for small subspace dimensions [James and Pai 2002]. SIERE does not have this issue because higher energy modes are not discarded.

Our BE implementation required 1257s for 400 frames, roughly double SIERE's runtime. The resulting energy curves are plotted in Figure 6. Notice that SIERE is not only faster, but also more dynamic. It can regain the elastic potential energy after each oscillation.

4.2 Untwisting cuboid

In this example we simulate the untwisting of a relatively soft elastic cuboid for a homogeneous neo-Hookean material with $\hat{Y} = 1e5\,Pa$. See Figure 7. We have used the time step size $h = 0.01s$.

Note that the starting configuration here is challenging, since we begin the simulation where the stiffness matrix K is not positive definite and with the cuboid under large deformation.

Both SI (see Figure 7(a)) and ERE became unstable after a few steps, if for different reasons (namely, SI because of divergence of the BE approximation, and ERE because of insufficient smoothing), while SIERE stayed stable.

Importantly, this example further demonstrates that it is unnecessary to employ a large subspace dimension s for SIERE to capture a complex deformation. We changed the subspace dimension s for SIERE and plotted the elastic potential energy for $s = 5, 6$ and 10; see Figures 7(b-d). Figure 7(e) shows that choosing a larger dimension s can improve the energy behaviour. However, a small $s = 6$ suffices to capture this rather complex twisting simulation, and the difference in potential energy between $s = 6$ and $s = 20$ is small. Notice also that $s = 5$ already makes for a lively simulation, although $s = 6$ visibly improves upon it. Please watch our supplementary video for a better grasp of this demonstration as well as the following one.

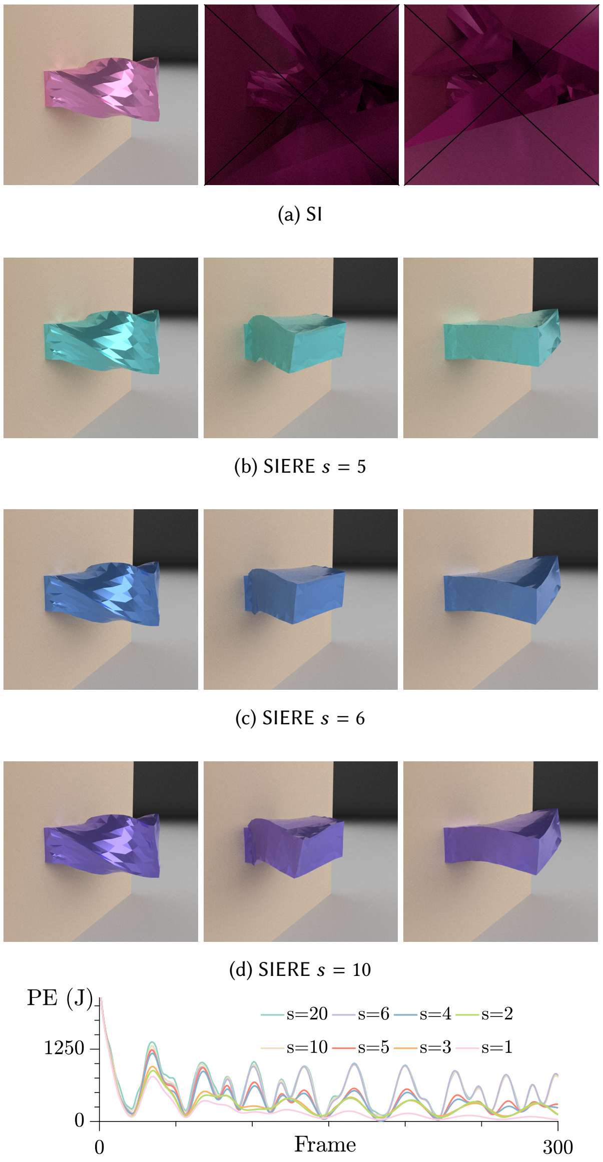
ARAP energy. To demonstrate the versatility of our method, we applied SIERE to the same cuboid example with ARAP energy [Chao et al. 2010; Sorkine and Alexa 2007] instead of neo-Hookean. Using the same setup and simulating for 150 frames, SIERE required 87 CPU seconds and BE cost 129 seconds. SI failed here. Figure 8 shows the corresponding energy curves comparing SIERE and BE. Again, we see that SIERE is superior to BE with lower computational cost and better energy behaviour.

4.3 Ball movement

For the present example, as well as all the following ones in Sections 4.5–4.7, we have fixed the subspace dimension at $s = 5$, having verified for each example that the results using $s = 10$ and $s = 20$ are not significantly different visually. This allows us to concentrate on other issues and show cost comparisons in Table 2 as the problem size gets larger.

4.3.1 Hanging ball. In the present simulation ERE still performs well, simulating a stiff elastic ball under a uniform force field, with a fixed top. We use the neo-Hookean material and time step size $h = 0.005s$. The ball has radius 10cm with a hard shell of thickness 2cm with peak Young's modulus $\hat{Y} = 1e8\,Pa$ and inner core with $\hat{Y} = 1e6\,Pa$. This setting gives rise to a stiffness matrix K with rather large condition numbers. At such high stiffness, the ball would not deform under gravity, so we add a uniform downward pulling force ($20 \times g$) to simulate greater deformation. See Figure 9 for energy plots of simulations with the three methods. We also experimented with the damping behaviour of SI, SIERE, and ERE. The runtimes are recorded in Table 2, while the mesh size is in Table 1. Notice that SIERE has as good an energy behaviour as ERE, whereas the uncontrolled damping of SI makes its resulting simulation look dull. In particular, the total energy plot in Figure 9 indicates that SI damps out gravitation potential very fast. For a small system like this one, ERE is fast as expected. In fact, the first row of Table 2 indicates that the additional subspace decomposition cost in SIERE can be higher than the additional cost of evaluating ϕ_1 action in ERE. We then increased the stiffness of the system by a factor of $1e2$ and re-simulated the scene, recording timing data in the second row of Table 2. Notice that the cost for both SI and SIERE remained constant, but that for ERE increased by a factor of 3. We also tried to reduce the damping error from SI by reducing the step size by a factor of 5; however, at such stiffness level, the uncontrolled damping is still large and no clear damping reduction was observed upon significantly reducing the step size.

4.3.2 Ball bouncing. In this example we drop the same ball as described above and simulate collision with the floor using simple penalty methods. The results are described visually in Figure 10 using both screenshots and energy plots.



(a) SI

(b) SIERE $s = 5$

(c) SIERE $s = 6$

(d) SIERE $s = 10$

(e) Potential Energy

Fig. 7. Simulation of a cuboid untwisting with SI and different subspace dimensions s for SIERE: (a) SI failed in this challenging simulation; (b-d) SIERE results with different s values; (e) elastic potential Energy for these SIERE simulations and more. In this example, $s = 6$ can already capture the complex twisting deformation. While smaller s could not capture the energy trajectory fully, at $s = 5$ the simulation already looks energetic and lively.

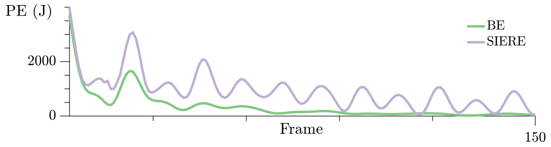


Fig. 8. Energy curves comparing SIERE and BE over 150 frames for the cuboid simulation.

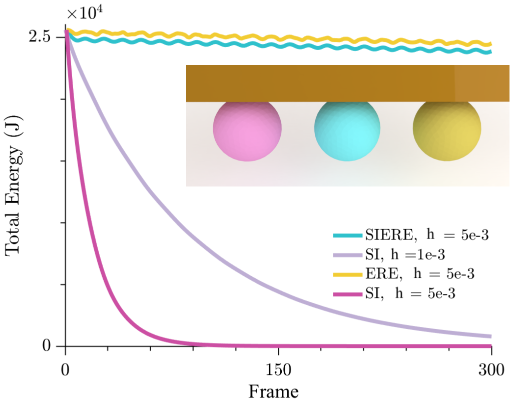


Fig. 9. Energy plot from the simulation of a stiff elastic ball. While SI displays a typical decline in energy, both ERE and SIERE are close to conserving it.

We make two essential observations, which again are easier to appreciate by watching the supplementary video. The first is that, comparing to the SIERE animation (blue), in the SI animation (purple) the ball loses height faster than it should and is less dynamic. The second important observation, is that SIERE captures secondary motion that SI does not. This can also be seen in the potential energy plots in Figure 10(c).

4.4 Human body dynamics

To demonstrate the applicability of our method to more complex real-world examples with constraints, we simulate the soft tissue dynamics of a human body. The body is modeled using a tetrahedral mesh of the soft tissue layer, with 40K vertices and 189K tets, which is attached to an “inner body” surface by constraining 6.2K vertices of the mesh. The body is initially in its rest shape and simulated with gravity, and subjected to an initial pulse of upwards force to emulate the body’s impact with the ground. The body is modeled employing a neo-Hookean material with $\hat{Y} = 1.6e4\, Pa$, integrated using time step size $h = 0.008s$. Although the material properties were chosen to be close to that of human soft tissues, and therefore softer than in the other examples, we found that simulations for this model can be very unstable, and the standard SI failed without additional damping. To handle the unstable high-frequency modes, we added the Rayleigh damping term $.005K\mathbf{v}$ to $\mathbf{f}_{\text{dmp}}(\mathbf{q}, \mathbf{v})$ in Eq. (1b). A total of 500 frames were simulated, where the initial force lasted for 50 frames.

When using SI, the low-frequency motion of the soft tissue is damped out very quickly, and after around the 150th frame virtually no dynamics is visible. However, when SIERE with $s = 6$ is used, the dominating oscillation of the soft tissue is captured by the exponential part of the integrator and remains sufficiently undamped to be visible even until the last frames. The results are shown in Figure 11, visualized for magnitude of per-vertex velocity to more

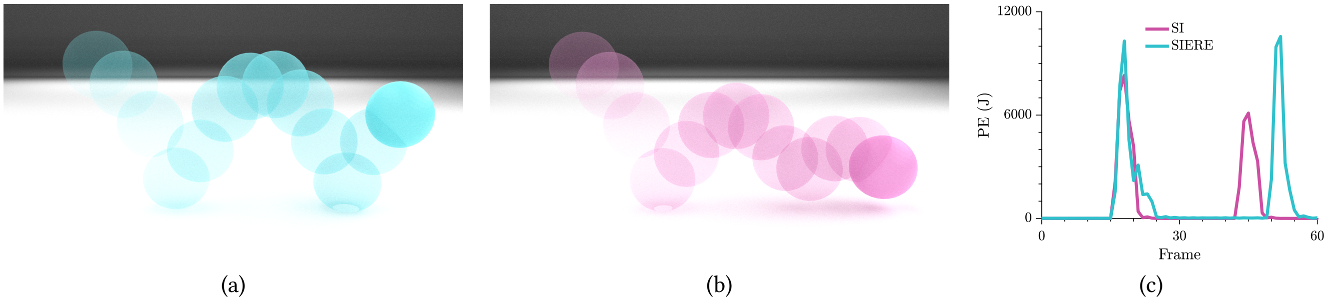


Fig. 10. A soft ball bounces after colliding with the floor: (a) SIERE ball movement with ($s = 5$) can simulate the secondary motion with more dynamics; (b) SI is more lethargic, as it does not recover the ball height after collision well and it loses the secondary motion; (c) corresponding energy plots reflect both observations. Notice that SIERE preserves the energy level after two impacts.

clearly demonstrate the difference in dynamics between the two methods. The runtimes are listed in Table 2: SIERE took about 85% longer than SI, where around 29% of the runtime of SIERE was due to the eigendecomposition. Comparing against SIERE with $s = 6$, using $s = 60$ instead took about twice longer (19,382 CPU seconds), and ERE took over eight times longer. The total potential energy of the system is plotted in Figure 12, showing the clear advantage of SIERE in terms of damping behaviour. For a better appreciation of the soft tissue dynamics, please watch our supplementary video.

4.5 Honeycomb structure

In this example we simulate a honeycomb sheet, a popular engineering structure that is strong and stiff in one direction but soft in the other two. We hang up the sheet and observe the ensuing oscillation. Again, we use neo-Hookean material. We experiment with soft core at $\hat{Y} = 1e6\,Pa$ and stiff edge $\hat{Y} = 1e8\,Pa$, and integrate using $h = 0.01s$. Note the significant heterogeneous jump.

Plots of energy behaviour are recorded in Figure 13 and runtime results are in Table 2. Notice that, even though the stiffness setting is similar to that in Section 4.3, ERE is already much more expensive than both SI and SIERE at this system size. When we further increase the stiffness by factor $1e2$ a similar trend can be observed more emphatically: ERE becomes prohibitively expensive, whereas SI and SIERE have a nearly constant cost at different stiffness levels. Finally, observe that SIERE has very good energy behaviour for the dominating motions: damping out only the high frequency regime it achieves excellent stability similar to SI.

4.6 Moebius ball

Next we apply SIERE to simulate an exotic object, a Moebius ball, to demonstrate that wild shapes of the deforming object won't be the limiting factor of SIERE; see Figure 14. In this example we used nonlinear StVK material. We set a stiff $\hat{Y} = 1e10\,Pa$ material for the core and $\hat{Y} = 1e8\,Pa$ for the thin sheet spiral, and stepped forward with $h = 0.02s$. ERE is prohibitively expensive to use in this situation and we terminated the program before its simulation ended.

4.7 Large scale structures

To further demonstrate the capacity of SIERE, we applied it to some large scale examples. In the first of these, we simulated a toy Eiffel tower mesh swinging under strong wind; see Figure 15. We used homogeneous StVK material with $\hat{Y} = 3e7\,Pa$. In the second larger scale example we simulated a multiscale system of a bridge using ARAP energy; see Figure 16. We used stiffness parameter $\hat{Y} = 1e10\,Pa$ at the stiffest part, and 1e6 at the softest part. In both cases we simulated using a large step size $h = 0.1s$.

In both meshes, we observe similar features of SIERE and reach the same conclusion: SIERE can bypass the efficiency barrier of exponential integrators and remain cheap regardless of system size and material stiffness. Furthermore, SIERE achieves excellent damping behaviour and stability by selectively damping out the high frequency oscillations.

4.7.1 Tree under strong wind. In our last example, we simulated a tree swaying in strong and varying wind using homogeneous neo-Hookean material with $\hat{Y} = 5e7\,Pa$. The time step size was $h = 0.01s$. The tree experiences large deformations and the branches have a complex structure with nearly 25,000 vertices; see Figure 17. SIERE can simulate a detailed motion for the branches and create a lively physical simulation with a small subspace dimension ($s = 5$). On the other hand, SI diverged for this simulation due to the large external force.

5 CONCLUSION

The quest of simulating, at a fair price, deformable structures that involve a high degree of stiffness in a manner that does not introduce over-damping is ubiquitous in computer graphics applications. Here we achieve this using an astonishingly simple approach. The key idea in this paper is to separate at each time step high frequency modes from the others, then employ for each regime a suitable method, and finally combine the results in an additive method that involves a low rank correction. Along the way we obtain a method that requires no solution of nonlinear algebraic equations and that is less prone to divergence and instabilities than both of its components. We have demonstrated the efficacy of our method, and we urge the reader to watch our supplementary video since animations tell the story far better than stills.

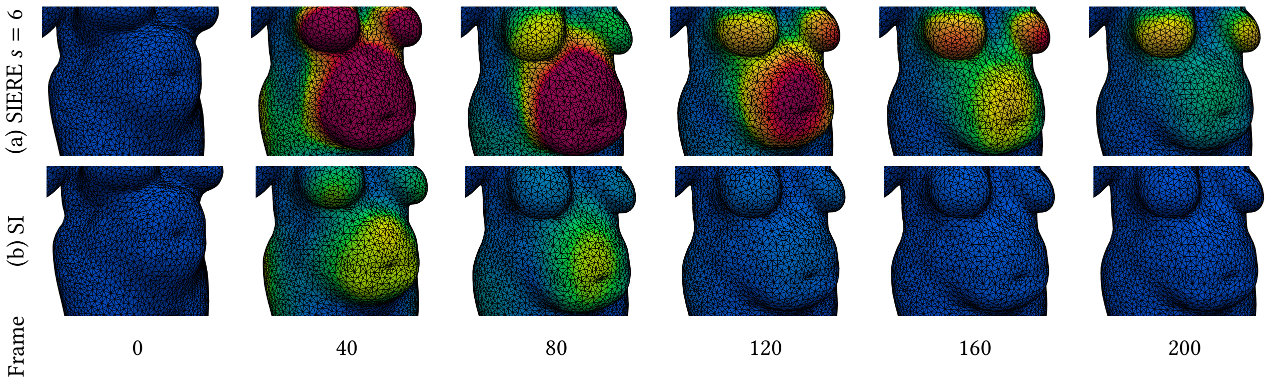


Fig. 11. A soft human body under gravity is subjected to an initial pulse of upwards force: (a) SIERE with subspace dinmension $s = 6$ captures the dominating motion of the soft tissue, and visible oscillations persist long after the initial force is turned off; (b) SI damps out the dynamics very quickly and no visible effects can be observed in the later frames. The six frames in (a) and (b) are visualized for the magnitude of per-vertex velocity and correspond to the simulation at the initial, 40th, 80th, 120th, 160th and 200th frame.

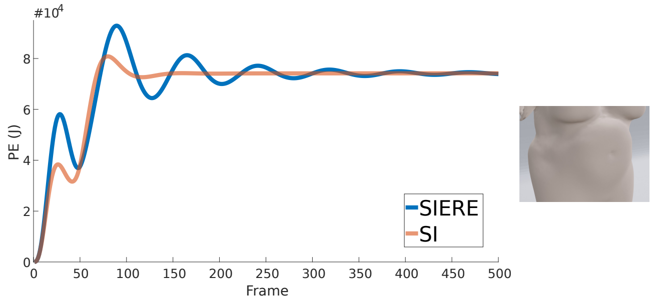


Fig. 12. Plot of the elastic energy potential from the human body simulation in Figure 11. Virtually all oscillations are damped out after around the 150th frame for SI, while SIERE is able to preserve the oscillations relatively undamped even until the last frames.

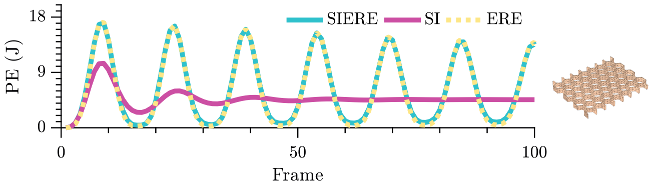


Fig. 13. Simulation of a honeycomb sheet: energy plot.

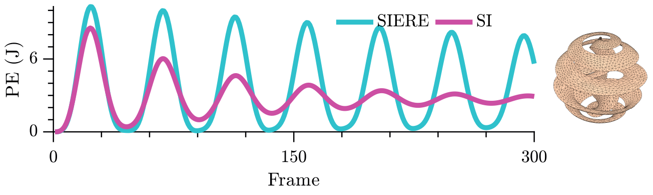


Fig. 14. Simulation of a Moebius ball: energy plot.

Limitations. Our paper does not render the SI method obsolete: there are many situations where SI performs adequately, and in

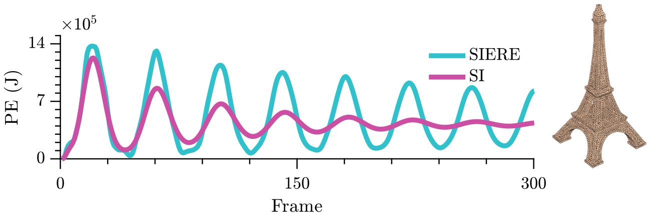


Fig. 15. Simulation of an Eiffell Tower mesh: energy plot.

those cases it is the cheapest alternative (see also [Tournier et al. 2015]).

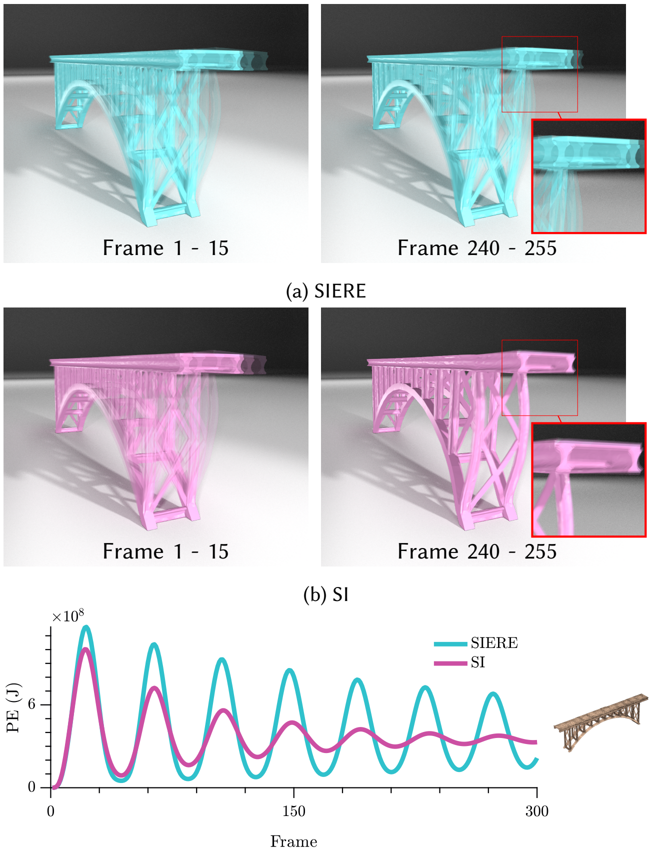
At the other end, we note that our method is not structure-preserving as are suitable methods that allow no artificial damping at all [Chen et al. 2017]. We do claim that there is a significant range of applications in between these two extremes, however, where SIERE outperforms both.

For SIERE to succeed, we must assume that a moderate value of the subspace dimension s suffices. In particular, we must have $s \ll n$, where n is the ODE system size, for the method to make computational sense. While this is the case with many visual simulations as we have shown, there are applications (such as sound) where other methods are more suitable.

Future. Additive hybrid methods such as ours are not necessarily limited to first order accuracy, and we believe that higher order methods of this sort can be constructed. Their utility relative to SIERE in difficult nonlinear scenarios would require a future project, though. Another interesting extension is to simulations involving mass-spring systems for stiff cloth. Yet another very interesting project would be the extension of SIERE to constraint-based simulations.

ACKNOWLEDGMENTS

The authors' research was funded in part by NSERC Discovery grants, NSERC PGSD, Canada Research Chairs Program, and Vital Mechanics Research.



(a) SIERE

(b) SI

(c) Potential Energy

Fig. 16. Simulation of a bridge deforming under strong wind: (a) SIERE remains dynamic with visible oscillations after 200 frames; whereas (b) oscillations under SI are heavily damped. A potential energy plot (c) underscores this observation.

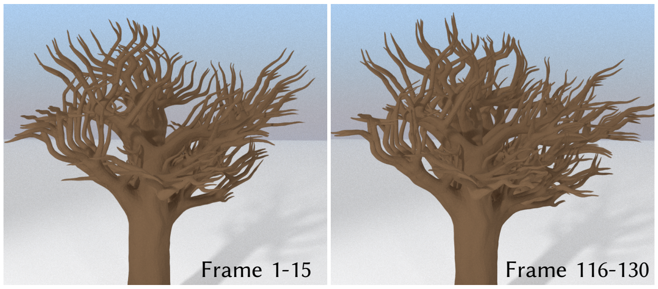


Fig. 17. Simulation of a tree swaying in strong wind.

REFERENCES

Awad H Al-Mohy and Nicholas J Higham. 2011. Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM journal on scientific computing* 33, 2 (2011), 488–511.

Uri Ascher. 2008. *Numerical Methods for Evolutionary Differential Equations*. Vol. 5. SIAM.

Uri Ascher, Steven Ruuth, and Brian Wetton. 1995. Implicit-explicit methods for time-dependent partial differential equations. *SIAM journal on numerical analysis* 32, 3 (1995), 797–823.

David Baraff and Andrew Witkin. 1998. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 43–54.

Jernej Barbic and Doug James. 2005. Real-Time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Trans. Graphics* 24(4) (2005), 982–990.

Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective dynamics: fusing constraint projections for fast simulation. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 154.

Eddy Boxerman and Uri Ascher. 2004. Decomposing cloth. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 153–161.

Isaac Chao, Ulrich Pinkall, Patrick Sanan, and Peter Schröder. 2010. A simple geometric model for elastic deformations. *ACM Trans. Graph. (TOG)* 29, 4 (2010), 38.

Desai Chen, David I. W. Levin, Wojciech Matusik, and Danny M. Kaufman. 2017. Dynamics-aware Numerical Coarsening for Fabrication Design. *ACM Trans. Graph.* 36, 4, Article 84 (July 2017), 15 pages. https://doi.org/10.1145/3072959.3073669

Yu Ju Chen, Uri Ascher, and Dinesh K Pai. 2018. Exponential Rosenbrock-Euler Integrators for Elastodynamic Simulation. *IEEE Transactions on Visualization and Computer Graphics* 24, 10 (2018), 2702–2713.

Yu Ju (Edwin) Chen, David I. W. Levin, Danny M. Kaufman, Uri M. Ascher, and Dinesh K. Pai. 2019. EigenFit for Consistent Elastodynamic Simulation Across Mesh Resolution. *Proceedings SCA* (2019). doi.org/10.1145/3309486.3340248.

Jintai Chung and GM Hulbert. 1993. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. *Journal of applied mechanics* 60, 2 (1993), 371–375.

Philippe G Ciarlet. 1988. *Three-dimensional elasticity*. Vol. 20. Elsevier.

Arne De Coninck, Bernard De Baets, Drosos Kourounis, Fabio Verbosio, Olaf Schenk, Steven Maenhout, and Jan Fostier. 2016. Needles: Toward Large-Scale Genomic Prediction with Marker-by-Environment Interaction. 203, 1 (2016), 543–555. https://doi.org/10.1534/genetics.115.179887 arXiv:http://www.genetics.org/content/203/1/543.full.pdf

Peter Deuflhard. 2011. *Newton Methods for Nonlinear Problems*. Springer.

Dimitar Dinev, Tiantian Liu, and Ladislav Kavan. 2018a. Stabilizing Integrators for Real-Time Physics. *ACM Trans. Graphics* 37(1) (2018).

Dimitar Dinev, Tiantian Liu, Jing Li, Bernhard Thomaszewski, and Ladislav Kavan. 2018b. FEPR: Fast Energy Projection for Real-Time Simulation of Deformable Objects. *ACM Trans. Graphics* 37(4) (2018).

Bernhard Eberhardt, Olaf Etzmuß, and Michael Hauth. 2000. *Implicit-explicit schemes for fast animation with particle systems*. Springer.

Doug L James and Dinesh K Pai. 2002. DyRT: dynamic response textures for real time deformation simulation with graphics hardware. In *ACM Transactions on Graphics (TOG)*, Vol. 21. ACM, 582–585.

L. Kharevych, W. Wei, Y. Tong, E. Kanso, J. E. Marsden, P. Schröder, and M. Desbrun. 2006. Geometric, Variational Integrators for Computer Animation. *Proc. SCA* (2006).

D. Kourounis, A. Fuchs, and O. Schenk. 2018. Towards the Next Generation of Multiperiod Optimal Power Flow Solvers. *IEEE Transactions on Power Systems* PP, 99 (2018), 1–10. https://doi.org/10.1109/TPWRS.2017.2789187

Minchen Li, Ming Gao, Timothy Langlois, Chenfanfu Jiang, and Danny M. Kaufman. 2019. Decomposed Optimization Time Integrator for Large-Step Elastodynamics. *ACM Transactions on Graphics* 38, 4 (2019).

Vu Thai Luan, Mayya Tokman, and Greg Rainwater. 2017. Preconditioned implicit-exponential integrators (IMEXP) for stiff PDEs. *J. Computational Physics* 335, 15 (2017), 846–864.

Dominik L Michels, Vu Thai Luan, and Mayya Tokman. 2017. A stiffly accurate integrator for elastodynamic problems. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 116.

Dominik L. Michels and J. Paul T. Mueller. 2016. Discrete Computational Mechanics for Stiff Phenomena. In *SIGGRAPH ASIA 2016 Courses*. ACM, 13:1–13:9.

Dominik L. Michels, J. Paul T. Mueller, and Gerrit A. Sobottka. 2015. A Physically Based Approach to the Accurate Simulation of Stiff Fibers and Stiff Fiber Meshes. *Computers & Graphics* 53B (Dec. 2015), 136–146.

Dominik L. Michels, Gerrit Sobottka, and Andreas Weber. 2014. Exponential Integrators for Stiff Elastodynamic Problems. *ACM Trans. Graph.* 33, 1 (Jan. 2014). https://doi.org/10.1145/2508462

Cleve Moler and Charles Van Loan. 2003. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Rev.* 45, 1 (2003), 3–49.

Jitse Niesen and Will M. Wright. 2012. Algorithm 919: A Krylov subspace algorithm for evaluating the ϕ-functions appearing in exponential integrators. *ACM Trans. Math. Software (TOMS)* 38, 3 (2012), article 22.

Jorge Nocedal and Stephen Wright. 2006. *Numerical Optimization*. New York: Springer. 2nd Ed.

Yixuan Qiu. 2015-2019. Spectra: C++ Library For Large Scale Eigenvalue Problems. https://spectralib.org/

Yussef Saad. 2011. *Numerical Methods for Laege Eigenvalue Problems*. SIAM. 2nd Ed.

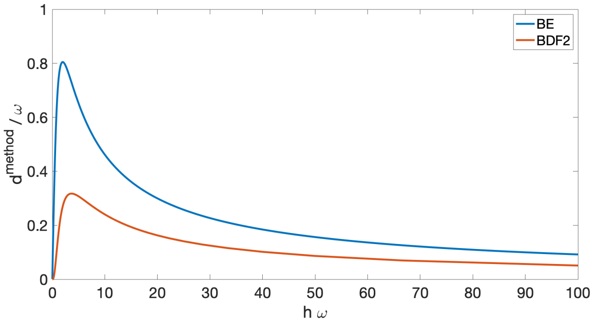


Fig. 18. Using BDF methods such as BE and BDF2 introduces significant artificial damping that depends on the time step.

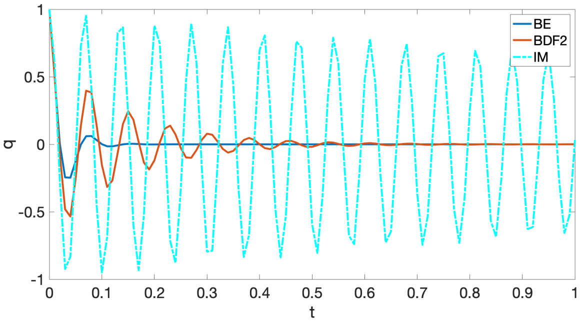


Fig. 19. Solutions of the scalar test equation $\ddot{q} + d\dot{q} + \omega^2 q = 0$ for $\omega = 100$, $d = 1$, using BE, BDF2 and implicit midpoint (IM) with step size h=0.01. Both BDF methods produce significant damping. See also Figures 6, 8-9, 13-15, and 16(c).

Eftychios Sifakis and Jernej Barbic. 2012. FEM simulation of 3D deformable solids: a practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*. ACM, 20.

Olga Sorkine and Marc Alexa. 2007. As-Rigid-As-Possible Surface Modeling. *Eurographics Symposium on Geometry Processing* 4 (2007), 109–116.

Jonathan Su, Rahul Sheth, and Ronald Fedkiw. 2013. Energy conservation for the simulation of deformable bodies. *IEEE Visualization and Computer Graphics* 19(2) (2013).

M. Tournier, M. Nesme, B. Gilles, and F. Faure. 2015. Stable constrained dynamics. *ACM Trans. Graphics* 34(4) (2015).

Fabio Verbosio, Arne De Coninck, Drosos Kourounis, and Olaf Schenk. 2017. Enhancing the scalability of selected inversion factorization algorithms in genomic prediction. *Journal of Computational Science* 22, Supplement C (2017), 99 – 108. https://doi.org/10.1016/j.jocs.2017.08.013

APPENDIX

The solution $q(t)$ of the scalar, constant coefficient ODE

$$\ddot{q} + \omega^2 q = 0$$

oscillates undamped with frequency ω. But applying BDF with step size h gives a numerical solution that more closely approximates the modified ODE

$$\ddot{q} + d^{\text{method}}\dot{q} + \omega^2 q = 0,$$

where the damping coefficient $d^{\text{method}} > 0$ depends on the step size h and on ω. Straightforward linear algebra gives a value such that d^{method}/ω depends only on the product $h\omega$ [Chen et al. 2018] .

Figure 18 shows the curves obtained for BE and BDF2, which are both popular methods for animating deformable objects. Figure 19 further depicts solution curves for particular, typical values of ω and h. As mentioned before, such artificial damping can lead to significant undesirable artifacts in the simulation when $h\omega$ is not small. On the other hand, the conservative implicit midpoint (IM) method introduces no artificial damping.

Note further that SI corresponds to applying one Newton iteration to solve the nonlinear algebraic equations arising from BE at each time step. Hence SI is the same as BE for linear problems. Further, we note in passing that an exponential method such as ERE reproduces the exact solution of this simple test problem. Although this ERE accuracy does not extend to general elastodynamics, the energy plots presented in the figures throughout Section 4 strongly indicate that the simple analysis in this appendix is indicative of the more general behaviour of BE and BDF2.