

# Exponential Rosenbrock-Euler Integrators for Elastodynamic Simulation

Yu Ju Chen, Uri M. Ascher, Dinesh K. Pai

Department of Computer Science, University of British Columbia, Vancouver, Canada

**Abstract**—High quality simulations of the dynamics of soft flexible objects can be rather costly, because the assembly of internal forces through an often nonlinear stiffness at each time step is expensive. Many standard implicit integrators introduce significant, time-step dependent artificial damping. Here we propose and demonstrate the effectiveness of an exponential Rosenbrock-Euler (ERE) method which avoids discretization-dependent artificial damping. The method is relatively inexpensive and works well with the large time steps used in computer graphics. It retains correct qualitative behaviour even in challenging circumstances involving non-convex elastic energies. Our integrator is designed to handle and perform well even in the important cases where the symmetric stiffness matrix is not positive definite at all times. Thus we are able to address a wider range of practical situations than other related solvers. We show that our system performs efficiently for a wide range of soft materials.

**Index Terms**—computer graphics, physically based animation, deformable models, time integration, exponential integrators



## 1 INTRODUCTION

Simulating the motion of soft objects is ubiquitous in computer graphics and robotics, and industrial applications abound. Examples include simulating the dynamics of cloth, skin, and other soft tissues of the human body, especially of the face.

Such simulations often require fine scale discretization and complex nonlinear materials to capture detailed qualitative behaviors. In a traditional scientific computing setting, such problems may be tackled using small and adaptively selected time steps to obtain high accuracy, but applications in computer graphics have their own unique constraints. For instance, the output is required only at large fixed time steps (typically at frequencies of 33-90 Hz). Qualitative behaviors are more important than pointwise accuracy, since the underlying material properties are highly uncertain and often set by artists. Stability and efficiency of simulation are paramount, being essential for productivity in most applications.

In computer graphics, stability is typically achieved using implicit or semi-implicit integrators, following the influential work of [4]. However, this comes at the cost of excessive numerical damping that depends on the discretization step size; examples of such methods include backward Euler, semi-implicit methods [4], and projective dynamics [6], [30]. Fully implicit methods also require expensive solution of nonlinear equations.

We opt instead for a carefully chosen exponential integrator for simulating general elastodynamics systems. Our integrator is designed to perform well even in cases where the symmetric stiffness matrix is not positive definite at every time point of interest. Thus, we are able to address a wide range of physical phenomena, including some that are not covered by other recent work on exponential integrators, for instance, in [21], [34], [36].

Following a (potentially co-rotated) finite element discretization in space, a system of ODEs in time is described in Section 3, where we allow the nonlinear stiffness matrix to not always be positive definite (corresponding to a non-convex energy). We then derive

a mathematical expression for the artificial numerical damping of several implicit and semi-implicit methods by relating their solution to that of a “ghost ODE” that is highly damped in the case that a large step size is used. In Section 4 we then derive our exponential Rosenbrock-Euler (ERE) algorithm and discuss its implementation and stability in the general highly oscillatory case. Performance comparisons follow in Section 5 in which the robustness and superiority of our method against explicit Runge-Kutta, implicit Euler and midpoint, and semi-implicit methods is demonstrated. Our results are obtained in the context of simulation and animation of soft elastodynamics motion. In particular, the ERE algorithm shines because the cost of assembling the stiffness matrix (or force differential) at a point in time is far larger than that of evaluating the action of a matrix exponential.

In summary, the highlights in this paper are:

- We extend the use of exponential integrators to general elastodynamics. More specifically, our method can handle any nonlinear elasticity model with non-convex potential energy. As with previous work, stability is not theoretically guaranteed in all cases, but it performs well for a range of practical applications.
- We apply a simple analysis to study the artificial damping behaviour of some popular simulation methods.
- Our algorithm recovers the dynamical response of soft material more faithfully than the semi-implicit Euler and other methods that introduce artificial damping.
- The integrator does not require nonlinear iterations, although a product of a matrix exponential and a vector is calculated at each time step; furthermore, for a wide array of soft materials the computational cost is comparable to the semi-implicit Euler method in large scale simulations.

## 2 RELATED WORK

*Integrators for stiff elastodynamics:* The equations of elastodynamics describe the traveling elastic waves inside a deforming object. This class of problems is said to be *stiff* because oscillating waves with speeds on different and widely varying time scales arise. Such problems are numerically challenging because an appropriate algorithm must handle highly oscillating waves, even though the deforming motion of a soft object such as cloth appears visibly smooth due to damping. To be efficient, an algorithm must be able to stably use large steps, commensurate with the visible motion but not with the internal high frequency modes.

Currently, the most popular integrator for elastodynamics simulation in computer graphics is a semi-implicit (SI) method proposed for cloth simulation in [4]. This method (which is equivalent to backward Euler (BE) employing only one Newton iteration at each time step [3]) requires the solution of one linear system per step. In addition to being fast, the SI method also exhibits reasonable large step stability (see also [43]). However, this stability comes from the numerical damping inherent to the method, making it hard to capture the correct dynamic response. Such numerical damping depends on the step size rather than on material properties, and it does not act uniformly on all waves: this is discussed further in Section 3.2 below.

Other implicit time integrators have also been used for elastodynamics, including BDF2 [18], implicit/explicit methods [8], [12] and variational integrators [25]. Unlike the SI method, these methods require the solution of a system of nonlinear equations at every time step, which is typically solved by some inexact Newton's method employing a Krylov subspace method for the sub-iterations. If the nonlinear equations are solved at each time step to sufficient accuracy, then these implicit Newton-Krylov methods are often more robust and accurate. Hence they are typically used when the cost of time stepping is not the major concern [26].

A branch of implicit integrators commonly used in the graphics community can be derived from the discretization for the Lagrangian corresponding to an IMEX method (e.g., backward Euler for elastic forces and forward Euler for external forces) [6], [14], [30], [31], [33], [38]. While being implicit, all these algorithms avoid the expensive Newton-Krylov methods and accelerate the nonlinear solve by using superior optimization techniques. This series of methods also applies to more general and complicated material models. However, as material stiffness increases, the performance level of all the techniques mentioned above drops significantly. Specifically, as material stiffens, these methods require many iterations, and the simulated material will appear softer if the optimization process stops prematurely.

*Exponential integrators:* Exponential integrators offer an attractive alternative to the methods mentioned above for the class of problems considered in this paper. This is because they allow taking much larger steps than explicit Runge-Kutta (RK) without introducing artificial damping that depends on the step size. Moreover, they can be tuned to perform effectively for a stiffness range that includes many practically important applications. A review of such methods can be found in [21]. In [20], [32] exponential integrators are compared to Newton-Krylov methods for various stiff systems. In the computer graphics context, such methods were applied to elastodynamics in [34], [35], [36].

The technique that the above mentioned articles use requires that the stiffness matrix be symmetric positive definite (SPD) at all relevant times. This enables usage of a corresponding square root of the stiffness matrix, and efficient methods for highly oscillatory ODEs based on this have been developed and improved over several decades [13], [15], [17], [19]. However, practical situations in which the energy is not convex are common in flexible body simulations [11]. In this paper, we extend the use of exponential integrators to a general hyperelasticity model of soft materials, without the convexity restriction. This is further discussed in the next two sections.

## 3 ELASTODYNAMICS

In this section we briefly describe the ODE system for elastodynamics that arises from a spatial discretization of the partial differential equations in time and space, often considered in variational form, that describe hyperelastic material models. For more details, please see [5], [7], [42]. We then derive for the simplest scalar case an explicit expression for the damping effect of the semi-implicit (SI), backward Euler (BE), and other discretizations when a large step size is employed.

### 3.1 The system of ODEs

Upon discretization of spatial variables, typically using a finite element method (FEM) with appropriate mass lumping, an elastodynamic system can be written as a second-order ODE system in time  $t$ , expressing Newton's second law of motion:

$$\begin{aligned} M\ddot{\mathbf{q}}(t) &= \mathbf{f}_{\text{tot}}(\mathbf{q}, \mathbf{v}) \\ &\equiv \mathbf{f}_{\text{els}}(\mathbf{q}) + \mathbf{f}_{\text{dmp}}(\mathbf{q}, \mathbf{v}) + \mathbf{f}_{\text{ext}}(\mathbf{q}). \end{aligned} \quad (1)$$

The unknowns  $\mathbf{q} = \mathbf{q}(t)$  are nodal displacements in a spatial FEM discretization or a mass-spring system, and the corresponding velocities are  $\mathbf{v}(t) = \dot{\mathbf{q}}(t)$ . The elastic and damping forces are written as

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

where  $W(\mathbf{q}(t))$  is the elastic potential of the corresponding hyperelasticity model. For example,  $W(\mathbf{q}(t))$  is quadratic in the linear elasticity model, and it is quartic in the linear FEM model with StVK material. In other models such as co-rotated FEM with linear material, neo-Hookean material, or artificially designed material [45], nonlinear potentials are required to describe the desired physical behavior.

We further split  $\mathbf{f}_{\text{els}}$  at each time instance  $t$  as:

$$\mathbf{f}_{\text{els}}(\mathbf{q}) = -K\mathbf{q} + \tilde{\mathbf{f}}_{\text{els}}(\mathbf{q}),$$

where  $K = \frac{\partial}{\partial \mathbf{q}} \mathbf{f}_{\text{els}}(\mathbf{q}(t))$  is the tangent stiffness matrix at time  $t$ , and  $\tilde{\mathbf{f}}_{\text{els}}$  is the remaining non-linear part. We will assume, as is often the case in practice, that  $M^{-1}K\mathbf{q}$  dominates all other forces in  $M^{-1}\mathbf{f}_{\text{els}}$  and  $M^{-1}\mathbf{f}_{\text{ext}}$ , i.e., that the linearized elastic force dominates.

The damping matrix  $D = D(\mathbf{q}(t), \mathbf{v}(t))$  is symmetric nonnegative definite at all times.

To solve Eq. (1), we first define the state vector  $\mathbf{u} = (\mathbf{q}, \mathbf{v})^T$  and rewrite the ODE as a first-order system

$$\dot{\mathbf{u}}(t) = \begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ M^{-1}\mathbf{f}_{\text{tot}}(\mathbf{q}, \mathbf{v}) \end{bmatrix} \stackrel{\text{def}}{=} \mathbf{b}(\mathbf{u}(t)),$$

which we further write as

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

$$\mathbf{g} = \mathbf{g}(\mathbf{q}(t)) = M^{-1}(\tilde{\mathbf{f}}_{\text{els}} + \mathbf{f}_{\text{ext}}).$$

*Stiff and oscillatory ODE system:* When  $\kappa = \|M^{-1}K(\mathbf{q}(t))\| \gg 1$ , the system (2) is considered *stiff*; and if the real parts of the large eigenvalues of  $K$  dominate, then the modes of the ODE (2) may oscillate with frequencies on very different time scales.<sup>1</sup> This is usually the case in elastodynamics because  $\kappa$  becomes large when

- the material stiffens under large deformation, and/or
- the simulation involves elastic waves traveling at speeds on different scales.

Upon using classical explicit integrators such as explicit RK, step sizes on the order of  $\kappa^{-1}$  are required for stability. As we further show in Section 3.2, classical integrators such as BE or SI add to each mode uncontrolled numerical damping depending on the step size. Conservative implicit integrators such as the midpoint method do not suffer from artificial damping; however, nontrivial nonlinear systems of algebraic equations must be solved at each large time step, which is well-known to be a potential show-stopper because the known solution at the start of the current step may not be a good guess for the sought solution at the next time level. To faithfully reproduce the dynamical behaviour of the underlying physical model while taking large time steps, we therefore introduce in Section 4 exponential integrators in order to numerically simulate Eq. (2).

*Non-convex potential energy:* The matrix function  $M^{-1}K(\mathbf{q}(t))$  can be viewed as being smooth in  $t$  away from “discrete events” (such as collisions). If  $M^{-1}K$  is SPD everywhere, then for each  $t$  there is a square root branch that gives an SPD matrix function  $\Omega(\mathbf{q}(t))$  that is smooth where  $M^{-1}K$  is smooth and  $M^{-1}K(\mathbf{q}(t)) = \Omega(\mathbf{q}(t))^2$ . However, to capture realistic behaviors of deformable objects, such as orientation preservation and material frame-indifference, it is essential to use non-convex elastic potential functions  $W(\mathbf{q})$  (see Ch. 4.8 in [11]). In fact, hyperelasticity models that have convex  $W(\mathbf{q})$  have undesirable artifacts. This non-convexity in  $W(\mathbf{q})$  makes the symmetric stiffness matrix  $K$  indefinite under some configurations, which excludes existence of  $\Omega(\mathbf{q}(t))$ . This in turn poses numerical challenges for exponential integrators, since the underlying assumption in [21], [34] and references therein is violated. For a non-convex potential energy one could attempt to find a smooth scalar function  $\mu(t) \geq 0$  such that  $\hat{K}(\mathbf{q}(t)) = M^{-1}K(\mathbf{q}(t)) + \mu(t)I$  is SPD, add  $-\mu\mathbf{q}$  to  $M^{-1}\tilde{\mathbf{f}}_{\text{els}}$ , and proceed to find the square root matrix function

1. More specifically, if  $M^{-1}K$  is SPD, there is an SPD square root matrix  $(M^{-1}K)^{1/2}$ . The eigenvalues of the matrix in (2) with  $D = 0$  are then equal to  $\pm i\lambda$  for each eigenvalue  $\lambda > 0$  of  $(M^{-1}K)^{1/2}$ . If  $M^{-1}K$  is not SPD, then it is harder to be specific, but damping often helps to diffuse the issue in applications.

of  $\hat{K}$  at each relevant  $t$  in a smooth way. This however could be a tall order, and moreover, the shift of the eigenvalues would also decrease accuracy of the exponential integration method since the shift term  $-\mu\mathbf{q}$  is no longer integrated exactly. In [34], [35], [36], the authors picked the popular co-rotated spring model  $\mathbf{f}_{\text{spring}}(\mathbf{q}) = RK(R^T\mathbf{q}(t) - \mathbf{q}(0))$ , for which the non-convex rotational part can be separated, and the exponential treatment was only applied to the SPD stiffness matrix  $K$ .<sup>2</sup> However, we want to work with more general models where such separation is not available. For these reasons we avoid taking this route despite the availability of good additional methods for the SPD case. Instead, we utilize a Rosenbrock-type exponential integrator [22].

### 3.2 Artificial damping

A pronounced damping effect of the BE and SI methods is often observed in practice. Here we encapsulate it in an explicit formula for the simplest case. We then apply a similar analysis to other methods that have been favoured in computer graphics practice.

Thus, we consider the scalar constant-coefficient ODE

$$\ddot{q} + d\dot{q} + \omega^2 q = 0, \quad (3)$$

where  $d \geq 0$  is a damping parameter, and  $\omega > d/2$  is a real-valued frequency. Setting  $q(t) = \exp(\lambda t)$  in Eq. (3) we obtain the quadratic equation  $\lambda^2 + d\lambda + \omega^2 = 0$ , so

$$\lambda = \frac{1}{2}[-d \pm i\sqrt{4\omega^2 - d^2}]. \quad (4a)$$

In particular, for the undamped case  $d = 0$ , the modes are  $\exp(\lambda t)$  with  $\lambda = \pm i\omega$ , i.e., these are oscillatory, undamped Fourier-type modes. Furthermore, for both eigenvalues in Eq. (4a) we have

$$|\exp(\lambda t)| = \exp(\text{Re}(\lambda)t) = \exp\left(-\frac{d}{2}t\right). \quad (4b)$$

*Backward Euler damping of the undamped ODE:* The BE and SI methods coincide here, and using a formula for inverting a  $2 \times 2$  matrix, the method with step size  $\tau$  at time  $t_n = n\tau$  can be written as

$$\begin{pmatrix} q_{n+1} \\ v_{n+1} \end{pmatrix} = \frac{1}{1 + \tau d + \tau^2 \omega^2} \begin{pmatrix} 1 + \tau d & \tau \\ -\tau \omega^2 & 1 \end{pmatrix} \begin{pmatrix} q_n \\ v_n \end{pmatrix} \equiv T_d \begin{pmatrix} q_n \\ v_n \end{pmatrix}. \quad (5)$$

Continuing with the undamped ODE, we set  $d = 0$  in Eq. (5). The two eigenvalues  $\mu$  of the propagator matrix  $T_0$  are  $\mu = 1/(1 \pm i\tau\omega)$ . The spectral radius of  $T_0$  is therefore  $\rho = \max|\mu| = 1/\sqrt{1 + (\tau\omega)^2}$ . We note that  $\rho < 1$  decreases (thus the damping effect of the numerical method increases) *monotonically* as the product  $\tau\omega$  of the method’s step size and the model’s frequency is increased.

Next, comparing the damping effect of SI/BE for the undamped ODE problem to that of a similar damped problem Eq. (3) with  $d = d^{BE} > 0$ , we equate  $\exp(-\frac{d^{BE}}{2}n\tau) = \rho^n = (1 + (\tau\omega)^2)^{-n/2}$ . Taking the natural logarithm and cancelling  $-n/2$  we see that the effect of the BE method is to introduce the artificial damping level

$$d^{BE} = \frac{1}{\tau} \ln(1 + (\tau\omega)^2). \quad (6)$$

2. The matrix  $K$  is constant for linear springs. For quadratic or cubic springs,  $K$  is not constant but it remains SPD. We refer to Algorithms 2 and 3 in [36] for more details.

Figure 1 shows  $d^{BE}/\omega$  as a function of  $\tau\omega$  for any nonnegative  $\tau$  and  $\omega$ . Note, in particular, that  $d^{BE} < 2\omega$  for any  $\tau\omega > 0$ .

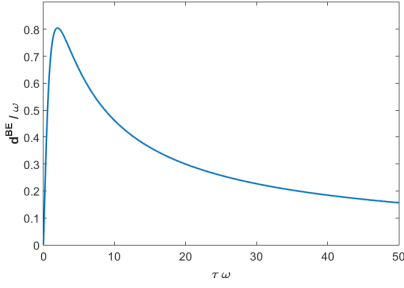


Figure 1: The damping coefficient curve  $d^{BE}/\omega$ , see Eq. (6), as a function of the product  $\tau\omega$  of the method's step size and the model's frequency.

For instance, if  $\tau = 1/\omega$ , which is close to where the curve in Figure 1 attains maximum, then  $d^{BE} = \omega \ln(2)$ . The BE method with this time step applied to the highly oscillatory, undamped ODE  $\ddot{q} + \omega^2 q = 0$ , is then best related in terms of damping to the exact solution (mode) of the *ghost ODE*

$$\ddot{q} + \omega \ln(2)\dot{q} + \omega^2 q = 0.$$

Figure 2 demonstrates this effect for initial values  $q_0 = 0$ ,  $v_0 = 1$ .

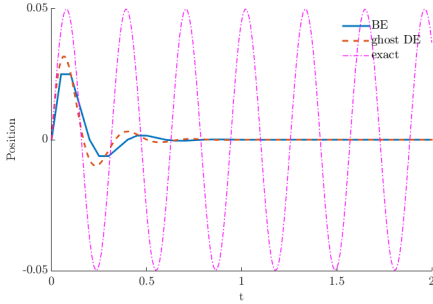


Figure 2: The backward Euler solution approximates the solution of the ghost ODE with damping coefficient given by Eq. (6) rather than the true, undamped one. Calculated positions are displayed for  $\omega = 20$ ,  $\tau = 1/\omega = .05$ .

Another instance is the “large step”  $\tau = 1$  with  $\omega \gg 1$ . An example is depicted in Figure 3, where through the first time step the BE solution skips many oscillations and yet lands on the damped ghost ODE curve.

*Conservative methods:* The undamped ODE with  $d = 0$  in Eqs. (3)-(4) is a simple instance of a Hamiltonian system. A symplectic discretization method such as implicit midpoint (IM) would therefore not allow any artificial damping. Indeed it can be easily verified directly that for this method the spectral radius of the propagator  $T_0$  is  $\rho = 1$  for any frequency and step size, and thus  $\ln(\rho) = 0$ , yielding  $d^{IM} = 0$ .

The same conclusion holds also for other conservative methods such as [23].

These methods are particularly attractive for quality simulations and using step sizes in the range of  $\tau\omega_{\max} = \mathcal{O}(1)$ . For much

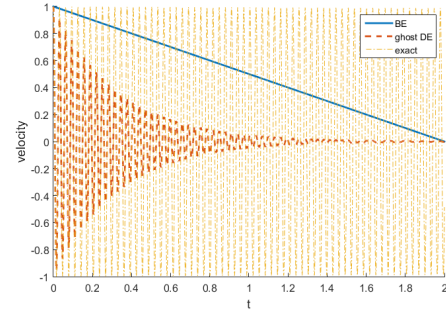


Figure 3: Same as Figure 2 but for  $\omega = 200$ ,  $\tau = 1$ .

larger steps, as mentioned earlier, there are typically difficulties in solving the nonlinear equations for an ODE system such as described in Section 3.1. Also, in such circumstances a solution trajectory can potentially be completely inaccurate despite being obtained stably.

*Generalized  $\alpha$  and other compromises:* It is popular in practice to seek methods that have some features of both BE- and IM-like methods. A weighted average of BE and IM comes to mind, but more popular in mechanical engineering has been the generalized  $\alpha$  method [10], [27]. It is a Newmark method, and it can be viewed as a second order accurate method discretizing a la RK the differential-algebraic representation of Eq. (1),  $\dot{\mathbf{q}} = \mathbf{v}$ ,  $\dot{\mathbf{v}} = \mathbf{a}$ ,  $M\mathbf{a} = \mathbf{f}_{\text{tot}}(\mathbf{q}, \mathbf{v})$ . Thus the dependent unknown vector functions in time are  $\mathbf{q}(t)$ ,  $\mathbf{v}(t)$  and  $\mathbf{a}(t)$ . The method has a knob,  $0 \leq r \leq 1$ , that can be tuned for a given frequency  $\omega$  and step size  $\tau$  to achieve a measured artificial damping. Applied to our test equation with  $d = 0$  this method reads

$$\begin{pmatrix} q_{n+1} \\ v_{n+1} \\ a_{n+1} \end{pmatrix} = T_0 \begin{pmatrix} q_n \\ v_n \\ a_n \end{pmatrix},$$

$$T_0 = \begin{pmatrix} 1 & 0 & -\tau^2\beta \\ 0 & 1 & -\tau\gamma \\ (1-\alpha_f)\omega^2 & 0 & (1-\alpha_m) \end{pmatrix}^{-1} \begin{pmatrix} 1 & \tau & \frac{\tau^2}{2}(1-2\beta) \\ 0 & 1 & \tau(1-\gamma) \\ -\alpha_f\omega^2 & 0 & -\alpha_m \end{pmatrix},$$

where the parameters are given in terms of  $r$  as  $\alpha_m = (2r - 1)/(1+r)$ ,  $\alpha_f = r/(1+r)$ ,  $\alpha = \alpha_m + \alpha_f$ ,  $\beta = (1-\alpha)^2/4$ ,  $\gamma = 1/2 - \alpha$ .

Calculating the log spectral radius  $\ln(\rho(T_0))$  for various values of  $\tau\omega$  we obtain the curves displayed in Figure 4.

Note, however, that despite the added flexibility as compared to BE, the general tradeoff of damping and conservatism is done with generalized  $\alpha$  on similar grounds, since one value of  $r$  must be somehow suitable for the entire wide range of relevant frequencies.

Furthermore, the general problem of simulating elastodynamics has of course far more complexity than the simple ODE of Eq. (3) can capture. Exponential integrators, discussed next, in fact solve the simple model ODE exactly, which still does not make them perfect.

## 4 EXPONENTIAL INTEGRATORS

Here we briefly describe exponential integrators as needed for our purposes. Much more is discussed in [21] and [37].

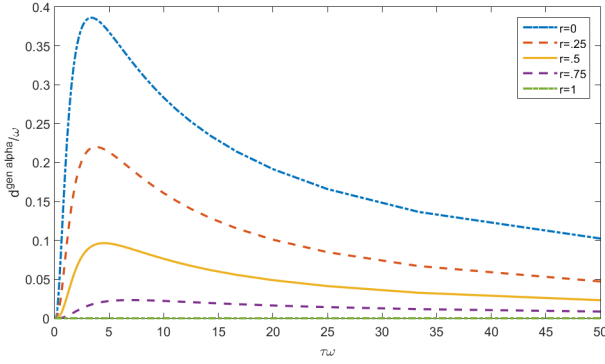


Figure 4: The damping coefficient curve  $d^{gen\ alpha}/\omega$ , as a function of  $\tau\omega$  for the generalized  $\alpha$  method.

*Variation-of-constants formula:* Consider the autonomous ODE

$$\dot{\mathbf{u}}(t) = \mathbf{b}(\mathbf{u}(t)), \quad \mathbf{u}(0) = \mathbf{u}_0. \quad (7)$$

Exponential integrators are based on splitting  $\mathbf{b}(\mathbf{u})$  as

$$\mathbf{b}(\mathbf{u}) = J\mathbf{u} + \mathbf{c}(\mathbf{u}), \quad \text{where } J = \frac{\partial \mathbf{b}}{\partial \mathbf{u}}(\mathbf{u}_0),$$

and employing the variation-of-constants formula

$$\mathbf{u}(t) = \exp(tJ)\mathbf{u}(0) + \int_0^t \exp((t-s)J)\mathbf{c}(\mathbf{u}(s))ds. \quad (8)$$

By approximating the integral in Eq. (8), we can obtain various exponential integrators.

*Exponential Rosenbrock-Euler (ERE) Method:* Let us write Eq. (7) at  $\mathbf{u}_n = \mathbf{u}(t_n)$  as

$$\begin{aligned} \dot{\mathbf{u}}(t) &= J_n \mathbf{u}(t) + \mathbf{c}_n(\mathbf{u}(t)), \\ J_n &= \frac{\partial \mathbf{b}}{\partial \mathbf{u}}(\mathbf{u}_n), \quad \mathbf{c}_n(\mathbf{u}) = \mathbf{b}(\mathbf{u}_n) - J_n \mathbf{u}(t). \end{aligned} \quad (9)$$

We then use Eq. (8) to integrate Eq. (9) from  $t_n$  to  $t_{n+1} = t_n + \tau$ , and restart the linearization process at  $t_{n+1}$ . The simplest method of this form can be constructed by fixing  $J_n$  and  $\mathbf{c}_n(\mathbf{u}(s))$  at  $s = t_n$ , enabling exact integration in Eq. (8) and thus leading to the exponential Rosenbrock-Euler (ERE) method

$$\begin{aligned} \mathbf{u}_{n+1} &= \exp(\tau J_n) \mathbf{u}_n + \tau \phi_1(\tau J_n) \mathbf{c}_n(\mathbf{u}_n) \\ &= \mathbf{u}_n + \tau \phi_1(\tau J_n) \mathbf{b}(\mathbf{u}_n) \end{aligned} \quad (10)$$

with  $\phi_1(z) = z^{-1}(\exp(z) - 1)$ .

Henceforth we will use this method (and not higher order Rosenbrock<sup>3</sup>) because we are interested in a qualitatively correct, inexpensive integrator and are less focused on very small pointwise errors (unlike [34], for instance). On the other hand, we have found it necessary, for the applications considered here, to refresh  $J_n$  and re-evaluate its exponential at each time step. Furthermore, we have considered and discarded use of conservative average vector field (AVF) methods [44] in the present context, because they also lead to potentially difficult large-step nonlinear systems. In our context, where assembling all the forces to form  $\mathbf{b}(\mathbf{u})$  is the major expense at a given time step, the use of ERE allows us to concentrate only on the evaluation of the exponential matrix function times a vector.

3. Higher order methods require additional matrix exponential evaluations in the internal stages at each time step, similar to RK methods. See, e.g., [21] for more detail.

## 4.1 Implementation of ERE

The expression Eq. (10) seems computationally attractive as it only requires one matrix function evaluation. However, the evaluation of the function  $\phi_1(z)$  suffers from numerical instability due to cancellation error when  $z \approx 0$  [24], [39]. In elastodynamics simulations, this could happen when the elastic waves are traveling on different time scales. To avoid computing  $\phi_1$ , we follow [2], [40] and rewrite Eq. (10) as

$$\begin{aligned} \mathbf{u}_{n+1} &= [I_N \quad 0_{N \times 1}] \exp(\tau A) \tilde{\mathbf{u}}_n, \quad \text{where} \quad (11) \\ A &= \begin{bmatrix} J_n & \mathbf{c}_n(\mathbf{u}_n) \\ 0_{1 \times N} & 0 \end{bmatrix}, \quad \tilde{\mathbf{u}}_n = \begin{bmatrix} \mathbf{u}_n \\ 1 \end{bmatrix}, \end{aligned}$$

which only involves one product of a matrix exponential of the slightly larger matrix with a vector.

### 4.1.1 Computing the action of the matrix exponential

When computing the product  $\exp(\tau A)\mathbf{u}$  for a large but sparse matrix  $A$  as in Eq. (11), it is important to avoid forming the full matrix exponential explicitly. Some of the most efficient algorithms implement substepping (or scaling) of the form

$$\exp(\tau A)\mathbf{u} = \exp(\delta t_1 A) \exp(\delta t_2 A) \cdots \exp(\delta t_s A) \mathbf{u}, \quad \sum_{i=1}^s \delta t_i = \tau \quad (12)$$

using one of the following methods:

- truncated Taylor series [2],
- Krylov subspace methods [41],
- Leja approximations [9].

The number of stages  $s$  or the substeps  $\delta t_i$  are chosen to optimize the cost and accuracy based on desired tolerance and error estimation. See [37] for more detail and further justification. The costs of all three methods grow with the matrix norm  $\|A\|$ , which in our context is majorized by the system stiffness-density ratio  $\|M^{-1}K\|$ . For the present type of application, we found that the Krylov subspace based algorithm from [41] performed best, so we have subsequently used their Matlab toolbox for all the calculations involving matrix exponentials reported here. At each substep  $\delta t_i$ , this algorithm uses the Arnoldi process to project the exponential operator onto a small Krylov subspace, where a small matrix exponential is calculated, and projects the result back. Further details of the algorithm are omitted here, as these are not contributions of the present paper and may be found in [41].

Let us denote the material stiffness-density ratio

$$k/m = \text{Young's modulus/density},$$

measured in  $(m/sec)^2$ . This quantity relates to the natural frequency of the material at the rest shape. Figure 5 shows the relationship between  $k/m$  and the cost of computing the action of the matrix exponential with the Matlab package `expv` from `expokit` by [41]. We make the following observations:

- Substepping of the form Eq. (12) is equivalent to time-stepping in a linear ODE with system matrix  $A$ . Hence, the growth in number of substeps  $s$  and ERE cost is similar to what is obtained from the stringent time step requirement for classical explicit methods for ODE systems with fast travelling waves. However, the cost of

substepping is cheap (e.g., involving an Arnoldi iteration and exponentiation of a small matrix when using a Krylov subspace), whereas small time steps with explicit methods are expensive (due to force calculations through FEM assembly from per element contribution). More comparisons with explicit methods are discussed in Section 5.2.

- ERE can become costly when  $\|M^{-1}K\|$  is very large, so it is only suitable for materials that are not too stiff. On the other hand, when  $\|M^{-1}K\|$  is large, regular implicit methods (e.g., backward Euler or implicit midpoint) encounter numerical difficulty from solving nonlinear systems at each time step. In particular, one can implement a more robust version of Newton's method by mixing it with gradient descent or damping the step [30], [33], but such implementation might increase cost due to loss of quadratic convergence and involves picking ad hoc parameters.
- On a positive note, according to experimental measurements from [1], [28], [29], there is a wide range of soft tissues of much current interest that have the material properties in the range where ERE is efficient. In addition, soft engineering materials that exhibit highly dynamical behavior can also be simulated effectively by ERE.

*Stability of exponential integrators:* For ERE to be stable, a necessary condition is that all the eigenvalues of the Jacobian matrix  $J$  must have negative real part at every time step [21], which corresponds to the damping in the system. The elastodynamic system that is being simulated has this property since elastic objects are naturally damped due to internal friction. In our implementation we have introduced a minimal amount of damping using the Rayleigh damping model

$$f_{\text{dmp}}(\mathbf{q}, \mathbf{v}) = (\alpha M + \beta K_0)\mathbf{v}, \quad (13)$$

(i.e., the damping matrix is  $D = \alpha M + \beta K_0$ ), where  $K_0$  is the (SPD) stiffness matrix at the initial state and  $\alpha, \beta > 0$ . In our experience, although the simple Rayleigh damping model introduced in Eq. (13) does not always guarantee stability, it works for a wide range of examples. In particular, the stability of ERE suffers with large  $\|M^{-1}K\|$ , and under large deformation, when  $\mathbf{g}(\mathbf{q}(t))$  in Eq. (2) grows larger. In the next section we show that ERE is stable for a wide range of soft materials that arise in computer graphics.

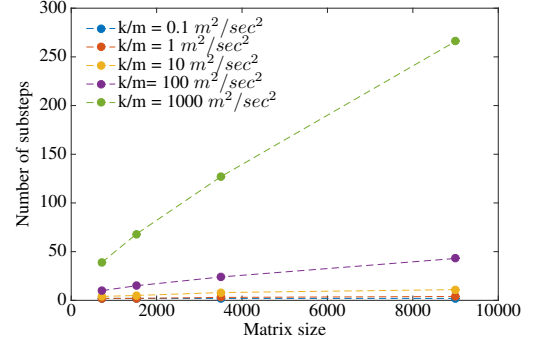
## 5 RESULTS

In this section we demonstrate the performance of ERE on (i) a linear mass-spring system with linear density  $0.1\text{kg/m}$ ; and (ii) a nonlinear neo-Hookean material model with density  $1000\text{kg/m}^3$  and Poisson ratio  $0.48$ , discretized using linear displacement tetrahedral elements. In both cases we employ the Rayleigh damping model (13) with  $\alpha = 0.01$  and  $\beta = 0.01$ .<sup>4</sup>

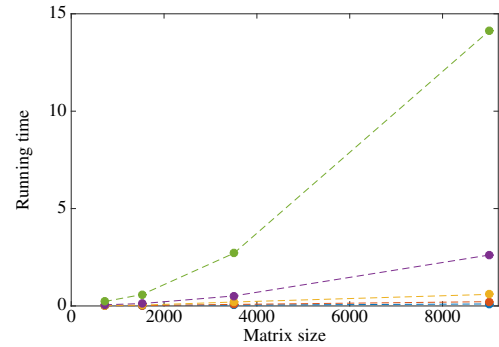
For each simulation we choose a range of linear stiffness  $K_l$  or Young's modulus  $Y^5$  and step size  $\tau$  to demonstrate performance.

4. Note that our linear mass-spring system is only linear in stretch, and a nonlinear system arises because of rotation, in contrast to the mass-spring system in [36].

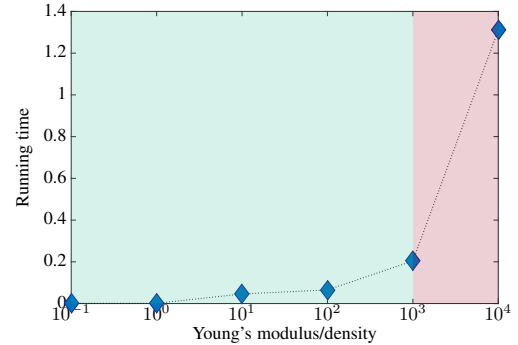
5. Since the material is nonlinear, the stiffness parameters are chosen for the rest state only.



(a) Number of substeps  $s$  vs. matrix size  $N$  with different stiffness-density ratio  $k/m$  for neo-Hookean material under small strain.



(b) Running time of  $\text{expv}$  vs. matrix size  $N$  with different  $k/m$  values for neo-Hookean material under small strain.



(c) Running time of  $\text{expv}$  vs.  $k/m$  for neo-Hookean material under small strain for a system of size 738.

Figure 5: Cost of calculating the matrix exponential action for ERE on a soft ball with neo-Hookean material under small strain at step size  $\tau = \frac{1}{30}$ . The matrix exponential actions are evaluated using  $\text{expv}$  from  $\text{expokit}$  [41]. (a) Number of substeps  $s$  grows with the stiffness-density ratio  $k/m$  (lines with different color) and matrix size  $N$  (x-axis). (b)  $s$  grows (roughly) linearly with  $N$ , leading to super-linear growth in total runtime. (c) The running time for computing the action of the matrix exponential depends on  $k/m$ . Nevertheless, soft tissues and soft engineering materials of much current interest can be simulated effectively by ERE.



We emphasize that ERE is not limited by the above systems: it can be applied to the dynamic system of any constitutive model semi-discretized in the form of Eq. (1).

All reported times  $t$  and step sizes  $\tau$  are in terms of seconds (sec). For reading clarity we occasionally omit this unit in the sequel.

### 5.1 Cost of ERE under large deformation

As mentioned in Section 4.1.1, the number of substeps  $s$  in evaluating the matrix exponential inside ERE depends on  $\|M^{-1}K\|$ , which depends on material properties. Under large deformation with a nonlinear material,  $\|M^{-1}K\|$  can also increase due to stiffening, and thus plague the performance of ERE. In the present example, we simulated a neo-Hookean cylinder under both compression and stretching (Figure 6) and recorded the number of substeps  $s$  used by `expv` within ERE in Table 1. We observe that  $s$  increases under stiffening only by less than a factor of 3 at 50% compression and by less than a factor of 2 at 100% stretch. For materials such as soft tissue and organs, the region of elastic deformation is less than 20%, meaning the cost of ERE will not change much for practical use.

	$k/m$	$s_{2\%}$	$s_{50\%}$	$s_{100\%}$
compression	$10^3$	2	3	
	$10^4$	5	7	
	$10^5$	12	23	
	$10^6$	48	123	
stretch	$10^3$	2	2	2
	$10^4$	4	4	5
	$10^5$	12	12	13
	$10^6$	47	51	58

Table 1: Number of substeps  $s$  used by `expv` in ERE for compressing and stretching a neo-Hookean cylinder (2383 tetrahedra) with different  $k/m$  at step size  $\tau = 1/30$  at 2%, 50% strain along the cylinder axis. For stretching, 100% stretch is also tested.

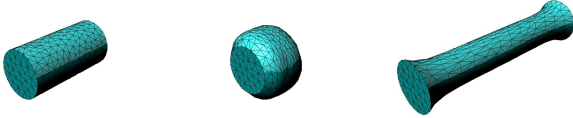


Figure 6: Compressing and stretching a neo-Hookean cylinder (2383 tetrahedrons)

### 5.2 Comparison with RK4

For soft material that is not extremely stiff, explicit methods such as the classical fourth order RK, denoted RK4, can be stable and efficient because the time step constraint is less stringent. To see that ERE compares favorably for soft materials nonetheless, we first test ERE against RK. In this comparison, we simulate one meter of elastic rope as 50 springs in sequence for 2 seconds and list the stable step size  $\tau_{rk}$  and the total running time  $t_{rk}^c$  for RK4 with different stiffness-density ratios. Results are collected in Table 2.<sup>6</sup> While ERE introduces with the listed step size some positional error  $\epsilon_{ere}$ , the simulation results are qualitatively correct

6. The solutions from RK4 are used as “ground truth” to calculate the error for ERE, since RK4 is a fourth order method that is necessarily evaluated using a much smaller time step.

and the method remains stable for all stiffness parameter values at step size  $\tau_{ere} = 1/60$  and has shorter total running time  $t_{ere}^c$ . The results demonstrate that RK4 is much more expensive (twice to 30 times). This is because calculating the force model (mass-spring and Rayleigh damping in this case) for small time steps is expensive, whereas the sub-stepping for the matrix exponential in ERE is much cheaper.

$K_l$	$\tau_{ere}$	$t_{ere}^c$	$\max \ \epsilon_{ere}\ $	$\tau_{rk}$	$t_{rk}^c$
$10^0$	1/60	5.2278	0.0720	1/180	9.7558
$10^1$	1/60	4.9717	0.0210	1/420	19.0327
$10^2$	1/60	6.1674	0.0511	1/1080	41.7449
$10^3$	1/60	6.6823	0.1424	1/4680	179.1033

Table 2: ERE and RK4 step sizes  $\tau$  and total running time  $t^c$  (measured in CPU seconds) for simulating an elastic rope with different linear stiffness  $K_l(N/m)$  for 2 seconds. Positional error  $\epsilon$  is measured in meters.

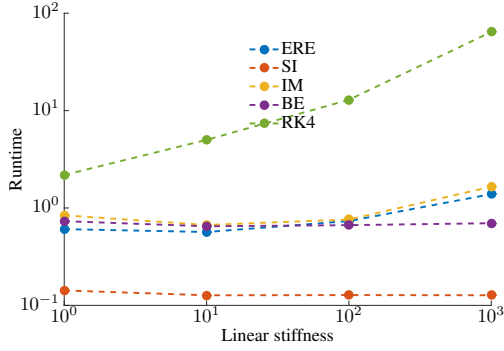
### 5.3 Comparison with semi-implicit method and implicit methods

*Solving nonlinear equations:* In the following examples, we used Newton’s method for the nonlinear equations arising in backward Euler (BE) and implicit midpoint (IM) integrators. The step is successful if the iteration’s residual is less than  $\epsilon_n = 10^{-6}$ . If the residual does not decrease after 3 iterations, then we locally reduce the time step by half and repeat the procedure over two half-steps. This is a standard procedure in numerical ODE practice (see, e.g., [3]) that is a simple instance of a numerical continuation method, using smaller steps to approximate the path from  $t_n$  to  $t_n + \tau$ . Specifically, it leverages the physical structure of the problem because by reducing the time step the starting guess  $\mathbf{u}_{n+1/2}$  is often closer to the desired  $\mathbf{u}_{n+1}$ . This procedure still guarantees quadratic convergence for small substep problems, whereas implementing linear search and Hessian fix [33] could reduce the convergence speed.

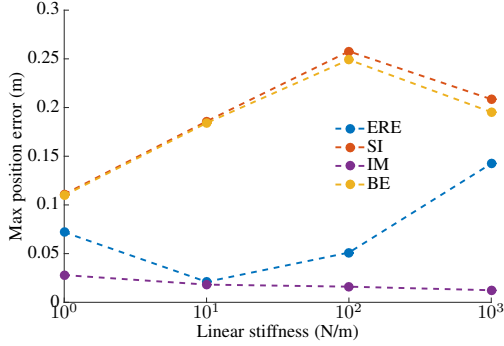
It is also possible to apply the semi-implicit approximation (i.e., a single Newton iteration) to IM rather than to BE. The cost per step is then the same as SI. However, we have observed that stability of the implicit midpoint method can be highly effected under this early termination. A similar effect was observed in [31]. Hence we do not consider this variant further.

*1D Mass-spring systems:* We first use the same rope simulation as described in Section 5.2 to compare the performance of ERE against the semi-implicit (SI) method and BE method (which both introduce artificial damping), as well as the IM method (which does not); see Figure 7. All the methods were run at step size 1/60 and remained stable for all stiffness parameters tested. Figure 7 shows that ERE is cheaper compared to the fully implicit methods (BE and IM) while having good accuracy (Figure 7b). Figure 7c shows energy profiles of different methods for the simulation with linear stiffness  $K_l = 100N/m$ . ERE is clearly better than BE and SI because less artificial damping is introduced.

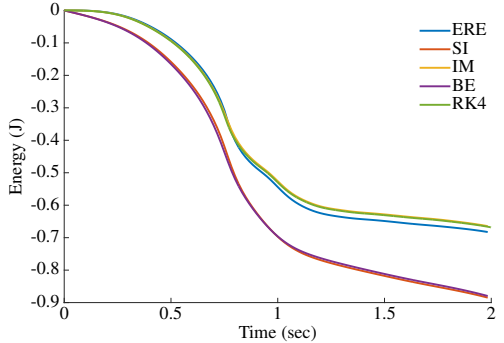
*2D Mass-spring systems:* In this example we simulate a piece of cloth with 121 particles ( $11 \times 11$ ) and simple stretching and shearing springs. The cloth is dropped with four corners fixed, and then quickly shaken at  $t = 1$ ; see Figure 9. In Figure 8 we compare ERE with SI, BE, and IM, all with step size  $\tau = 1/60$ , in terms of running time and maximum position error (RK4 with small time



(a) Running time of each integrator for the rope simulation with different stiffness parameter.



(b) ERE gives up some point-wise accuracy; however, it is close to that of IM and performs significantly better than BE and SI. The latter also introduce more artificial damping.



(c) The energy profile of each integrator in the rope simulation with  $K_l = 100\text{N/m}$ . Both SI and BE damp out the energy significantly, while the energy profile of ERE follows closely those of IM and the small-step RK4.

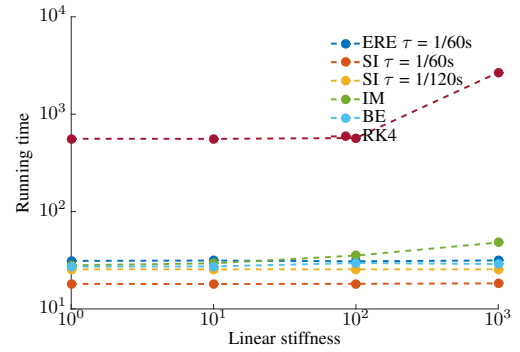
Figure 7: These plots compare ERE with SI, BE, IM, and RK4 over 2 seconds of the rope simulation. Compared are the (a) running time, (b) position error, and (c) energy profile.

step serves as ground truth). Figure 9 depicts the frames at  $t = 1.7$ , after the shaking event at  $t = 1$ . The cloth looks more damped with SI and BE, whereas the ground truth is more responsive and dynamic. The ERE solution is only slightly damped, and it remains more faithful to the true solution. Once the SI step size is reduced down to  $\tau_{si} = 1/120$ , the simulation error and visual impressions are similar to ERE with step size  $\tau_{ere} = 1/60$ . This is in keeping with the fact that the damping introduced by SI reduces for a smaller step size. However, reducing the step size also increases the total running time (Figure 8a).

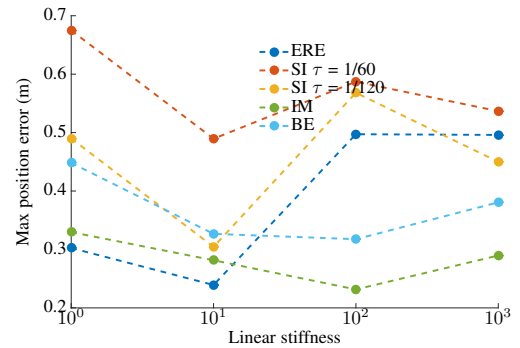
In the next example we make the same cloth mesh with linear stiffness  $K_l = 100\text{N/m}$  collide with a sphere. The collision is resolved by projecting each particle onto the closest point on the sphere. Figure 10 depicts the state of the cloth after the collision. The cloth responds more dynamically in the simulation using ERE and IM. Both SI and BE introduce much more damping into the system, as shown in Figure 11. Table 3 displays the total simulation running time for each integrator with step size  $\tau = 1/60$ . ERE and SI are much cheaper than BE and IM, since no additional Newton iterations are required.

In Figure 12 we changed half of the cloth to a slightly softer material  $K_l = 20\text{N/m}$  and the other half to a stiffer material  $K_l = 200\text{N/m}$ . In this mixed material example, both SI and BE are seen to introduce non-uniform damping, relative to the stiffness, and fail to capture the state of the cloth even qualitatively, whereas ERE looks qualitatively similar to IM.

*Character Cloth:* In this example we simulate a cape ( $11 \times 11$ ) attached to a dancing character, with step size  $\tau = 1/60$  (Figures 13 and 14). From these examples we conclude that ERE is an efficient explicit method for problems such as cloth simulation that does not introduce too much damping and responds more dynamically to user inputs and collision events.



(a) Running time for simulating 5 seconds of cloth modeled by a 2D Mass-spring system.



(b) Max position errors for each integrator during the simulation.

Figure 8: These plots compare ERE to SI, BE, and IM over 5 seconds of cloth modeled by a mass-spring system. SI with  $\tau = 1/120$  is added to demonstrate the  $\tau$ -dependent damping effect of this method.

*3D Volumetric Bar:* Next we use a neo-Hookean material model to simulate elastic deformation of 3D volumetric bars. For the first volumetric simulation, we simulate a twisted soft elastic bar with



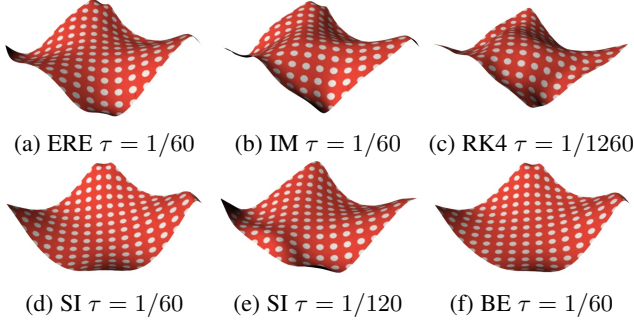


Figure 9: Response of the cloth with ( $K_l = 100N/m$ ) after the shaking event using different integrators.

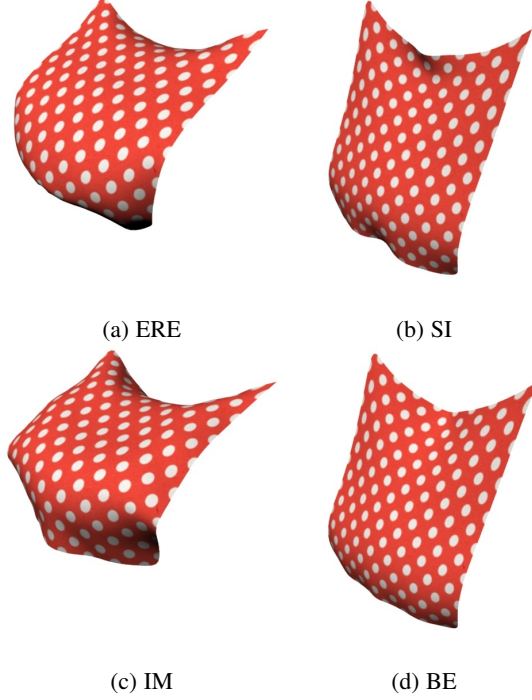


Figure 10: Response of the cloth ( $K_l = 100N/m$ ) after colliding with a sphere using different integrators.

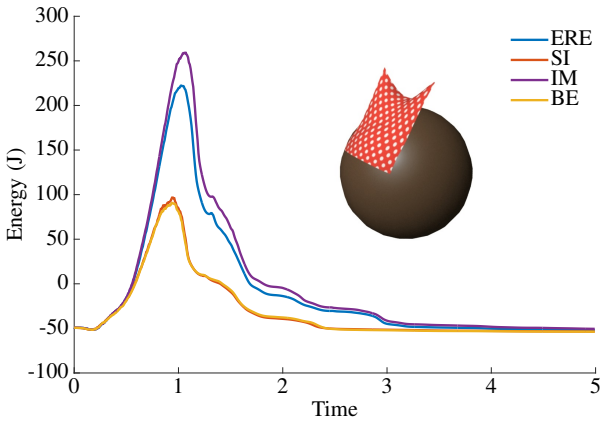


Figure 11: Energy profile of each method in the simulation with cloth collision.

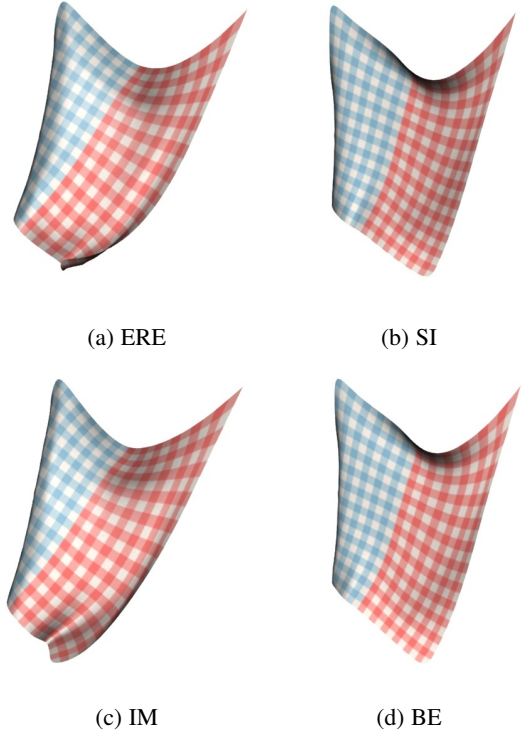


Figure 12: Response of the cloth with mixed stiffness,  $K_l = 20N/m$  (red) and  $K_l = 200N/m$  (blue), after colliding with a sphere. The highly damping integrators miss the correct qualitative behaviour.

Integrator	Total running time
ERE	8.4187
SI	6.8089
BE	18.9087
IM	29.8337

Table 3: Running time of each method for simulating the cloth collision scene for 5 seconds.

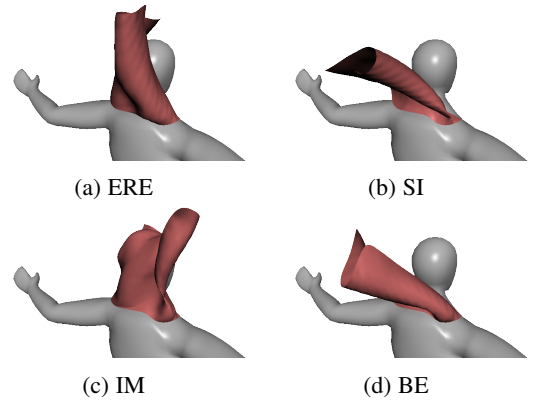


Figure 13: Simulating the cape motion of a dancing character using different integrators ( $K_l = 50N/m$ ).

Young's modulus values  $Y$  ranging from  $10Pa$  to  $10kPa$ , see Figure 15.

In our second volumetric example we simulate elastic bars, with Young's modulus ranging from  $10kPa$  to  $50kPa$ , dropping with

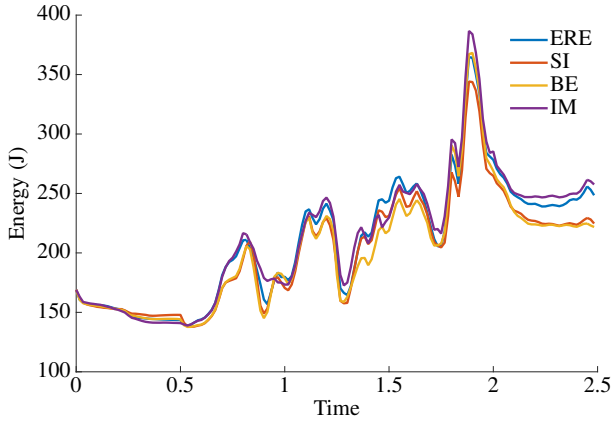


Figure 14: Energy profiles of the four integrators for the example of a cape attached to a dancer.

one end fixed; see Figure 16. Similarly, we plot the running time of each method in Figure 16a, and the energy profile of the bar with Young's modulus  $10kPa$  in Figure 16b. We observe that the running time for ERE increased as we stiffen the material, as predicted in Figure 5.

Both these examples clearly demonstrate that our previous conclusions extend to a volumetric simulation, namely, that the BE and SI methods lose too much energy while the fully implicit methods are too slow.

*Stanford bunny*: In our last volumetric simulation we simulate the Stanford bunny ( $Y = 100kPa$ ,  $\tau = 1/180$ ) bouncing due to gravity. Figure 18 shows the energy profile for each integrator in a similar fashion as before. Similar conclusions are drawn from this figure, and we recommend watching the associated video clip at [www.cs.ubc.ca/~aq978063/ere.html](http://www.cs.ubc.ca/~aq978063/ere.html)

## 6 CONCLUSIONS AND FUTURE WORK

*Conclusions*: In this article we have proposed and demonstrated a low cost algorithm for producing qualitatively correct simulations of elastodynamic motion involving soft materials. Our algorithm works well, using large time steps, even when the stiffness matrix is not positive definite everywhere, and it produces simulations that do not suffer heavy step size-related damping. Our system derives its relative efficacy from the fact that in typical soft body simulations in computer graphics the major cost per time step is in the assembly of the forces rather than in evaluating products with matrix exponentials. The need of solving nontrivial nonlinear systems of algebraic equations at each time step is avoided as well. The performance of our system has been demonstrated in Section 5; please see also our animation video clip.

Currently, the majority of researchers employ the semi-implicit (SI) method of [4] for similar simulation purposes. This method is attractive for several reasons: it is very simple, constraints are naturally incorporated at the end of each step, and stability is typically not an issue (though not always [3]). However, it is also well-known that a significant artificial damping is introduced by this method, thus making an artist using such a simulation tool having to deal with implementation-dependent artifacts and introducing distortion in the resulting animation. In simple cases the obtained

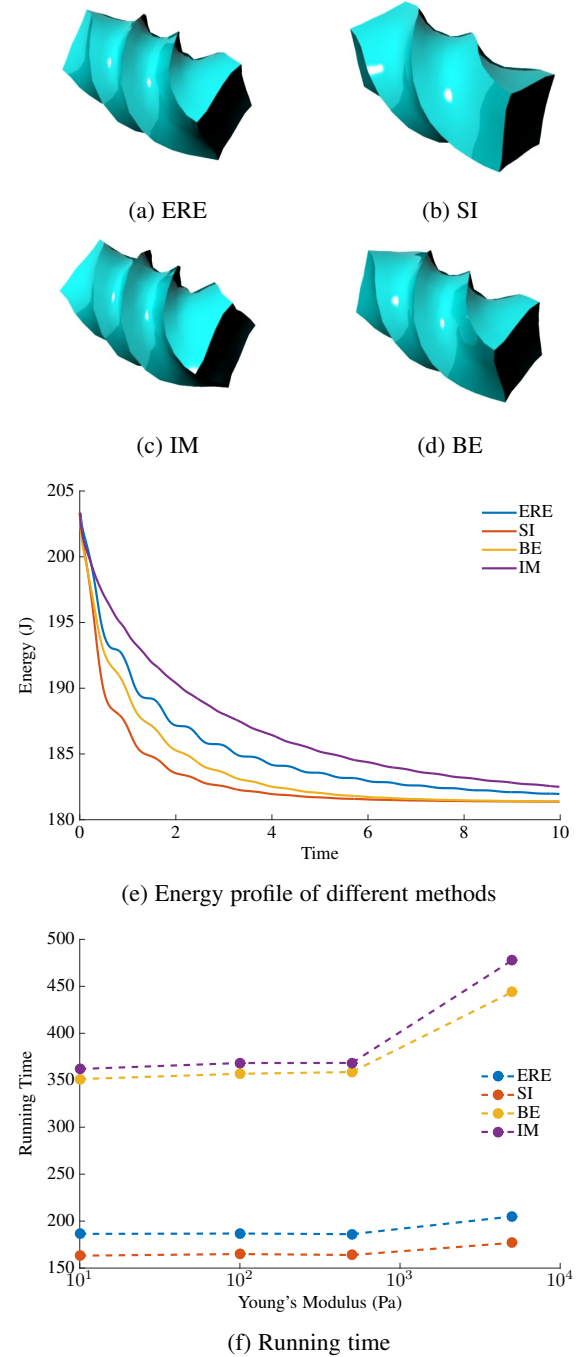
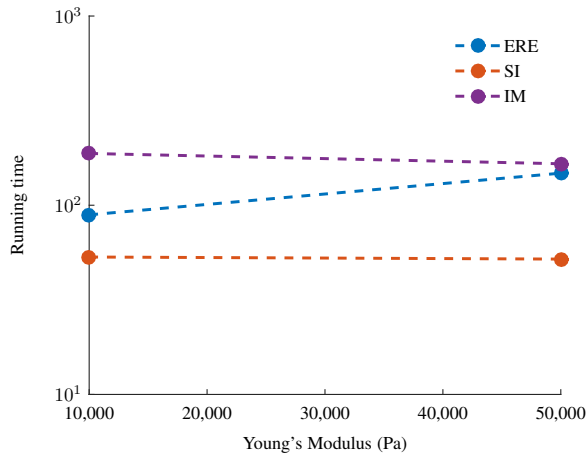


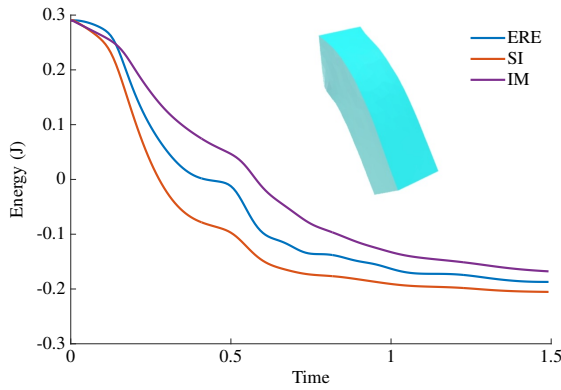
Figure 15: These plots compare ERE to SI, BE, and IM in the twisted bar simulation.

motion often looks realistic, but with a heavier damping than desired, and this effect is demonstrated analytically in Section 3.2. In the experiments of Section 5, ERE shows a clear advantage over SI for cloth simulation: the step size dependent damping introduced by SI could significantly change the cloth response to external force, whereas ERE keeps the solution qualitatively similar to that of the physical model, even when using large steps in time. In addition, since ERE is explicit, popular constraints for cloth motion [8], [16] can be easily imposed.

*Limitations and future work*: As demonstrated in Figure 5 and discussed more generally in Section 4, our algorithm loses efficacy



(a) Running time of each method for simulating a elastic bars with Young's modulus from 10kPa to 50kPa.



(b) Energy profile of different methods for the elastic bar with Young's modulus 10kPa.

Figure 16: These plots compare ERE to SI, BE, and IM in the simulation of a volumetric bar with one end fixed.

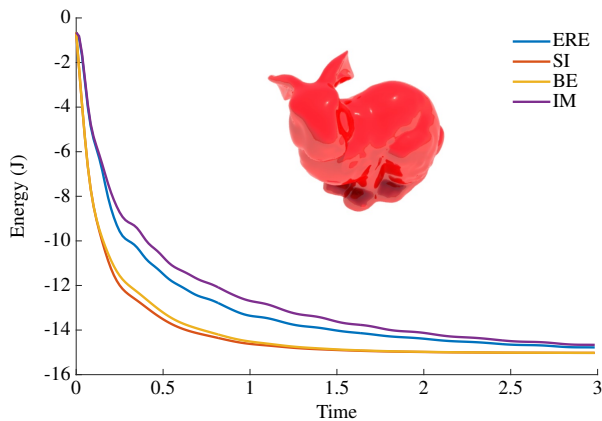


Figure 17: Energy profile of each method for the bunny with Young's modulus 100kPa.

Figure 18: This plot compare the energy in ERE to SI, BE, and IM in the simulation of the Stanford bunny.

when the material gets too stiff, although there is a comfortable practical range of applications where it wins.

Future work may include extension to higher order Rosenbrock-Euler methods and general-purpose constraint handling by step-end projection. We would also like to develop better physically-based damping models for elastodynamics to enhance the stability of ERE. There are also the interesting quest for cost-reducing dimensional reduction methods for the exponentiation, and possible extension of methods such as [13] to problems with non-convex energy.

## REFERENCES

- [1] R. Akhtar, M. J. Sherratt, J. K. Cruickshank, and B. Derby, "Characterizing the elastic properties of tissues," *Materials Today*, vol. 14, no. 3, pp. 96–105, 2011.
- [2] A. H. Al-Mohy and N. J. Higham, "Computing the action of the matrix exponential, with an application to exponential integrators," *SIAM journal on scientific computing*, vol. 33, no. 2, pp. 488–511, 2011.
- [3] U. M. Ascher, *Numerical Methods for Evolutionary Differential Equations*. SIAM, 2008, vol. 5.
- [4] D. Baraff and A. Witkin, "Large steps in cloth simulation," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '98. New York, NY, USA: ACM, 1998, pp. 43–54.
- [5] J. Bonet and R. Wood, *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 2008.
- [6] S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. Pauly, "Projective dynamics: fusing constraint projections for fast simulation," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 154, 2014.
- [7] D. Braess, *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*. Cambridge University Press, 2007.
- [8] R. Bridson, S. Marino, and R. Fedkiw, "Simulation of clothing with folds and wrinkles," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2003, pp. 28–36.
- [9] M. Caliarì and A. Ostermann, "Implementation of exponential Rosenbrock-type integrators," *Applied Numerical Mathematics*, vol. 59, no. 3, pp. 568–581, 2009.
- [10] J. Chung and G. Hulbert, "A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized- $\alpha$  method," *J. Applied Mech.*, vol. 60, pp. 371–375, 1993.
- [11] P. G. Ciarlet, *Mathematical Elasticity: Three-Dimensional Elasticity*. Elsevier, 1993, vol. 1.
- [12] B. Eberhardt, O. Eitzmuß, and M. Hauth, *Implicit-explicit schemes for fast animation with particle systems*. Springer, 2000.
- [13] B. Garcia-Archilla, J. Sanz-Serna, and R. D. Skeel, "Long-time-step methods for oscillatory differential equations," *SIAM Journal on Scientific Computing*, vol. 20, no. 3, pp. 930–963, 1998.
- [14] T. F. Gast, C. Schroeder, A. Stomakhin, C. Jiang, and J. M. Teran, "Optimization integrator for large time steps," *IEEE Trans. Visualization and Computer Graphics*, vol. 21, no. 10, pp. 1103–1115, 2015.
- [15] W. Gautschi, "Numerical integration of ordinary differential equations based on trigonometric polynomials," *Numerische Mathematik*, vol. 3, no. 1, pp. 381–397, 1961.
- [16] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun, "Efficient simulation of inextensible cloth," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, vol. 26, no. 3, p. to appear, 2007.
- [17] V. Grimm and M. Hochbruck, "Error analysis of exponential integrators for oscillatory second-order differential equations," *Journal of Physics A: Mathematical and General*, vol. 39, no. 19, p. 5495, 2006.
- [18] M. Hauth and O. Eitzmuß, "A high performance solver for the animation of deformable objects using advanced numerical methods," in *Computer Graphics Forum*, vol. 20, no. 3. Wiley Online Library, 2001, pp. 319–328.

- [19] M. Hochbruck and C. Lubich, "A Gautschi-type method for oscillatory second-order differential equations," *Numerische Mathematik*, vol. 83, no. 3, pp. 403–426, 1999.
- [20] M. Hochbruck, C. Lubich, and H. Selhofer, "Exponential integrators for large systems of differential equations," *SIAM Journal on Scientific Computing*, vol. 19, no. 5, pp. 1552–1574, 1998.
- [21] M. Hochbruck and A. Ostermann, "Exponential integrators," *Acta Numerica*, vol. 19, pp. 209–286, 2010.
- [22] M. Hochbruck, A. Ostermann, and J. Schweitzer, "Exponential Rosenbrock-type methods," *SIAM Journal on Numerical Analysis*, vol. 47, no. 1, pp. 786–803, 2009.
- [23] C. Kane, J. Marsden, M. Ortiz, and M. West, "Variational integrators and the newmark algorithm for conservative and dissipative mechanical systems," *Int. J. Numer. Meth. Eng.*, vol. 49(10), pp. 1295–1325, 2000.
- [24] A.-K. Kassam and L. N. Trefethen, "Fourth-order time-stepping for stiff PDEs," *SIAM Journal on Scientific Computing*, vol. 26, no. 4, pp. 1214–1233, 2005.
- [25] L. Kharevych, W. Yang, Y. Tong, E. Kanso, J. E. Marsden, P. Schröder, and M. Desbrun, "Geometric, variational integrators for computer animation," in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2006, pp. 43–51.
- [26] D. A. Knoll and D. E. Keyes, "Jacobian-free Newton–Krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, vol. 193, no. 2, pp. 357–397, 2004.
- [27] M. Kobis and M. Arnold, "Convergence of generalized- $\alpha$  time integration for nonlinear systems with stiff potential forces," *Multibody Syst Dyn*, vol. 37, pp. 107–125, 2016.
- [28] B. C. W. Kot, Z. J. Zhang, A. W. C. Lee, V. Y. F. Leung, and S. N. Fu, "Elastic modulus of muscle and tendon with shear wave ultrasound elastography: variations with different technical settings," *PloS one*, vol. 7, no. 8, p. e44348, 2012.
- [29] X. Liang and S. A. Boppart, "Biomechanical properties of in vivo human skin from dynamic optical coherence elastography," *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 4, pp. 953–959, 2010.
- [30] T. Liu, A. W. Bargteil, J. F. O'Brien, and L. Kavan, "Fast simulation of mass-spring systems," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 214, 2013.
- [31] T. Liu, S. Bouaziz, and L. Kavan, "Quasi-newton methods for real-time simulation of hyperelastic materials," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 3, p. 23, 2017.
- [32] J. Loffeld and M. Tokman, "Comparative performance of exponential, implicit, and explicit integrators for stiff systems of ODEs," *Journal of Computational and Applied Mathematics*, vol. 241, pp. 45–67, 2013.
- [33] S. Martin, B. Thomaszewski, E. Grinspun, and M. Gross, "Example-based elastic materials," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4, p. 72, 2011.
- [34] D. L. Michels, V. T. Luan, and M. Tokman, "A stiffly accurate integrator for elastodynamic problems," *ACM Trans. Graph.*, vol. 36, no. 4, August 2017.
- [35] D. L. Michels, J. P. T. Mueller, and G. A. Sobottka, "A physically based approach to the accurate simulation of stiff fibers and stiff fiber meshes," *Computers & Graphics*, vol. 53B, pp. 136–146, Dec. 2015.
- [36] D. L. Michels, G. Sobottka, and A. Weber, "Exponential integrators for stiff elastodynamic problems," *ACM Trans. Graph.*, vol. 33, no. 1, Jan. 2014.
- [37] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM Rev.*, vol. 45, no. 1, pp. 3–49, 2003.
- [38] R. Narain, M. Overby, and G. E. Brown, "ADMM  $\supseteq$  projective dynamics: Fast simulation of general constitutive models," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '16. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2016, pp. 21–28. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2982818.2982822>
- [39] J. Niesen and W. M. Wright, "Algorithm 919: A Krylov subspace algorithm for evaluating the  $\phi$ -functions appearing in exponential integrators," *ACM Trans. Math. Software (TOMS)*, vol. 38, no. 3, p. article 22, 2012.
- [40] Y. Saad, "Analysis of some Krylov subspace approximations to the matrix exponential operator," *SIAM Journal on Numerical Analysis*, vol. 29, no. 1, pp. 209–228, 1992.
- [41] R. B. Sidje, "Expokit: a software package for computing matrix exponentials," *ACM Transactions on Mathematical Software (TOMS)*, vol. 24, no. 1, pp. 130–156, 1998.
- [42] E. Sifakis and J. Barbic, "FEM simulation of 3D deformable solids: a practitioner's guide to theory, discretization and model reduction," in *ACM SIGGRAPH 2012 Courses*. ACM, 2012, p. 20.
- [43] M. Tournier, M. Nesme, B. Gilles, and F. Faure, "Stable constrained dynamics," *ACM Trans. Graphics*, vol. 34(4), 2015.
- [44] B. Wang and X. Wu, "A new high precision energy-preserving integrator for system of oscillatory second-order differential equations," *Physics Letters A*, vol. 376, 2012.
- [45] H. Xu, F. Sin, Y. Zhu, and J. Barbič, "Nonlinear material design using principal stretches," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 75, 2015.



**Yu Ju Chen** is currently working towards his PhD degree at the University of British Columbia. He also received his BSc in Engineering Physics from the University of British Columbia.



**Uri M. Ascher** is Professor of Computer Science at the University of British Columbia, Vancouver, Canada. He obtained his BSc and MSc in Applied Mathematics from Tel-Aviv University in 1969 and 1971, and his PhD in Computer Science from the University of Minnesota in 1975. He is a SIAM Fellow and recipient of CAIMS Research Prize. The focus of his work is on the investigation, promotion and application of novel, efficient and reliable methods in scientific computation, and their application in computer graphics, imaging and other approximation problems involving differential equations with constraints and optimization.



**Dinesh K. Pai** Dinesh K. Pai is a Professor and Tier 1 Canada Research Chair in the Department of Computer Science at the University of British Columbia, where he directs the Sensorimotor Systems Laboratory. He is also a member of UBC's Institute for Computing, Information and Cognitive Systems (ICICS), the Institute of Applied Mathematics, the Graduate Program in Neuroscience, the ICORD International Collaboration on Repair Discoveries, and the Brain Research Centre. He received his Ph.D. from Cornell University, Ithaca, NY, and his B.Tech. degree from the Indian Institute of Technology, Madras. Pai has received several awards, including UBC's 2012 Killam Research Prize.