# Geometric Integration for High Fidelity Visual Computing Applications

**Dominik L. Michels**
High Fidelity Algorithmics Group, Computer Science Department, Stanford University,
353 Serra Mall, MC 9515, Stanford, US-CA 94305
michels@cs.stanford.edu

**Abstract**. To be able to take into account a multitude of physical effects, high fidelity simulations are nowadays of growing interest for analyzing and synthesizing visual data. In contrast to most numerical simulations in engineering, local accuracy is secondary to the global visual plausibility. Global accuracy can be achieved by preserving the geometric nature and physical quantities of the simulated systems for which reason geometric integration algorithms like symplectic methods are often considered as a natural choice. Additionally, if the underlying phenomena behaves numerically stiff, a non-geometric nature comes into play requiring for strategies to capture different timescales accurately.
In this contribution, a hybrid semi-analytical, semi-numerical Gautschi-type exponential integrator for modeling and design applications is presented. It is based on the idea to handle strong forces through analytical expressions to allow for long-term stability in stiff cases. By using an appropriate set of analytical filter functions, this explicit scheme is symplectic as well as time-reversible. It is further parallelizable exploiting the power of up-to-date hardware. To demonstrate its applicability in the field of visual computing, various examples including collision scenarios and molecular modeling are presented.

**Acknowledgments**. This work has been supported by the Max Planck Center for Visual Computing and Communication.

## Scope of this Report

With the increasing demand for realism in synthesized visual data comes the need for hight fidelity simulations, which usually involve the numerical integration of the underlying differential equations. This requires integrators which are able to solve large complex systems in milliseconds to analyze or synthesize visual data efficiently or even to reach interactivity for applications with user interaction. Global accuracy should be achieved, so that the preservation of geometric and physical quantities is eligible. Moreover, the integrators should be parallelizable to exploit the improvements in computational power of the recent years.
In this workshop contribution, we study a class of geometric integration methods for solving stiff Newton equations of motion: The semi-analytical, semi-numerical Gautschi-type exponential integrator. We show the construction of explicit, symplectic, and time-reversible schemes of this kind and discuss appropriate parallelization techniques using rational Krylov subspace projection methods in order to reach applicability in the field of visual computing.
This report provides a presentation of the studied parallel exponential integrator framework. The general concept of exponential integration was initially introduced based on the observation that traditional integrators do not exploit analytical solutions to linear differential equations, cf. [7, 8]. In our construction, we make used of analytical trigonometric expressions introduced in [4] and a trapezoidal rule used originally in [2]. This is extended with additional so-called filter functions to achieve symmetry and symplecticity, cf. [3].

Exponential integration were originally introduced in the area of visual computing in [9]. Recent work on rational Krylov subspace-based approximations of matrix functions can be found in [5] and references therein.

**Gautschi-type Exponential Integrators**

We provide a brief derivation of a symplectic, and time-reversible Gautschi-type exponential integrator for the numerical treatment of $N$-dimensional systems of second-order ordinary differential equations of the form

$$\boldsymbol{M}\ddot{\boldsymbol{x}}(t) + \boldsymbol{D}\dot{\boldsymbol{x}}(t) + \boldsymbol{K}\boldsymbol{x}(t) + \boldsymbol{\Lambda}(\boldsymbol{x}(t)) = \boldsymbol{0}, \tag{1}$$

which are also known as semi-linear equations of motion with system configuration $\boldsymbol{x} : \mathbb{R}_{\geq t_0} \to \mathbb{R}^N$, mass matrix $\boldsymbol{M} \in \mathrm{GL}_N(\mathbb{R})$, damping matrix $\boldsymbol{D} \in \mathbb{R}^{N\times N}$, symmetric stiffness matrix $\boldsymbol{K} \in \mathbb{R}^{N\times N}$, and non-linearity $\boldsymbol{\Lambda} : \mathbb{R}^N \to \mathbb{R}^N$.
Using the modified coordinates $\boldsymbol{\chi}(t) := \sqrt{\boldsymbol{M}}\boldsymbol{x}(t)$, Eq. (1) can be written in the equivalent form

$$\ddot{\boldsymbol{\chi}}(t) + \boldsymbol{A}\boldsymbol{\chi}(t) + \boldsymbol{\Psi}\left(\boldsymbol{\chi}, \dot{\boldsymbol{\chi}}\right) = 0 \tag{2}$$

with a symmetric system matrix $\boldsymbol{A} := \sqrt{\boldsymbol{M}}^{-1}\boldsymbol{K}\sqrt{\boldsymbol{M}}^{-1}$ and a modified non-linearity

$$\boldsymbol{\Psi}\left(\boldsymbol{\chi}(t), \dot{\boldsymbol{\chi}}(t)\right) := \sqrt{\boldsymbol{M}}^{-1}\boldsymbol{\Lambda}\left(\sqrt{\boldsymbol{M}}^{-1}\boldsymbol{\chi}(t)\right) + \sqrt{\boldsymbol{M}}^{-1}\boldsymbol{D}\sqrt{\boldsymbol{M}}^{-1}\dot{\boldsymbol{\chi}}(t).$$

To solve Eq. (2), we consider its homogeneous version (i.e. $\boldsymbol{\Psi} = \boldsymbol{0}$), whose analytical solution is simply given by

$$\boldsymbol{\chi}(t) = \mathsf{cos}\left(t\sqrt{\boldsymbol{A}}\right)\boldsymbol{\chi}(0)$$

for initial configurations with $\dot{\boldsymbol{\chi}}(t_0) = 0$ (w.l.o.g. we also set $t_0 := 0$). By applying higher-dimensional Chebyshev polynomials defined by $\mathsf{T}_n(\cdot) = \mathsf{cos}\left(n\,\mathsf{acos}(\cdot)\right)$[1], we obtain the two-step method

$$\boldsymbol{\chi}_{n+1} = \mathsf{T}_{n+1}\left(\mathsf{cos}\left(\Delta t\sqrt{\boldsymbol{A}}\right)\right)\boldsymbol{\chi}(0), \tag{3}$$

in which $\boldsymbol{\chi}_n$ denotes $\boldsymbol{\chi}(n\Delta t)$.
We evaluate Eq. (3) and see that $\boldsymbol{\chi}_{n+1} = 2\,\mathsf{cos}\left(\Delta t\boldsymbol{A}\right)\boldsymbol{\chi}_n - \boldsymbol{\chi}_{n-1}$ is a representation of the analytical solution of the homogeneous system. In order to handle the inhomogeneity as well we add a numerical solution of the inhomogeneity to this analytical solution of the homogeneous part. Integrating $\boldsymbol{\Psi}$ twice using a rectangular approximation leads to the integrator

$$\begin{aligned}\boldsymbol{\chi}_{n+1} = 2\mathsf{cos}\left(\Delta t\sqrt{\boldsymbol{A}}\right)\boldsymbol{\chi}_n - \boldsymbol{\chi}_{n-1} - \Delta t^2\boldsymbol{M}^{-1}\mathsf{sinc}^2\left(\Delta t\sqrt{\boldsymbol{A}}\right)\\ \cdot\,\boldsymbol{\Psi}\left(\mathsf{sinc}\left(\Delta t\sqrt{\boldsymbol{A}}\right)\boldsymbol{\chi}_n, \frac{1}{\Delta t}\mathsf{sinc}\left(\Delta t\sqrt{\boldsymbol{A}}\right)\left(\boldsymbol{\chi}_n - \boldsymbol{\chi}_{n-1}\right)\right).\end{aligned} \tag{4}$$

[1] In this context, $\mathsf{T}_n$ denotes the multi-dimensional generalization of the first kind Chebyshev polynomial of order $n$ defined recursively by the relation

$$\mathsf{T}_{n+1}(\boldsymbol{\mathcal{A}}) := \begin{cases} 2\boldsymbol{\mathcal{A}}\mathsf{T}_n(\boldsymbol{\mathcal{A}}) - \mathsf{T}_{n-1}(\boldsymbol{\mathcal{A}}), & \text{if } n \in \mathbb{N}^{\geq 2} \\ \boldsymbol{\mathcal{A}}, & \text{if } n = 1 \\ \boldsymbol{1}, & \text{if } n = 0 \end{cases}$$

for square matrix arguments $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{N\times N}$. Its equivalence to the definition $\mathsf{T}_n(\cdot) = \mathsf{cos}\left(n\,\mathsf{acos}(\cdot)\right)$ can easily be proved using the relation $\mathsf{cos}\left(\boldsymbol{\mathcal{A}}\right)\mathsf{cos}\left(\boldsymbol{\mathcal{B}}\right) = 1/2\,\left(\mathsf{cos}\left(\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}}\right) + \mathsf{cos}\left(\boldsymbol{\mathcal{A}} + \boldsymbol{\mathcal{B}}\right)\right)$ for $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \in \mathbb{R}^{N\times N}$.

Since high external frequencies tend to produce numerical instabilities, we have added a low-pass $\mathsf{sinc}$-filter in both integration steps ($\mathsf{sinc}^2$) and filter the argument of the inhomogeneity in a similar way ($\mathsf{sinc}$). This choice ($\mathsf{sinc}^2, \mathsf{sinc}$) for the filter functions correspond to a symplectic integration scheme, cf. [6]. Moreover, Eq. (4) is time-reversible.
This kind of integrators are usually denoted as *Gautschi-type exponential integrators*, since Walter Gautschi already developed a semi-analytical integrator in the early 1960s (cf. [4]), which corresponds to the presented algorithm if we change the filter function pair to $\left(\mathsf{sinc}^2\left(1/2\,\cdot\right), \mathbf{1}\right)$. The term *exponential* was proposed in [8] since such a construction for a semi-linear first-order differential equation system naturally leads to the use of a matrix exponential describing the analytical solution of the homogeneous part.

**Rational Krylov Subspace Projection**

To implement the Gautschi-type exponential integrator efficiently, a fast computation of the matrix functions $\mathbf{f}(\cdot) \in \left\{\mathsf{cos}(\Delta t\sqrt{\cdot}), \mathsf{sinc}(\Delta t\sqrt{\cdot}), \mathsf{sinc}^2(\Delta t\sqrt{\cdot})\right\}$ is needed. According to [11], the key idea is to prevent the direct computation of the matrix functions $\mathbf{f}(\boldsymbol{A})$ and to instead polynomially approximate the actions $\mathbf{f}(\boldsymbol{A})\boldsymbol{b}$ on an operand vector $\boldsymbol{b}$ after projecting on a lower $N \gg m$-dimensional Krylov subspace

$$\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{b}) := \mathsf{span}\left(\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \ldots, \boldsymbol{A}^{m-1}\boldsymbol{b}\right)$$

generated by the matrix $\boldsymbol{A} \in \mathbb{C}^{N\times N}$ and the vector $\boldsymbol{b} \in \mathbb{C}^N$. Like in a power iteration, $\boldsymbol{A}^{m-1}\boldsymbol{b}$ approximates the dominant eigenvector. Hence the basis vectors $\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \ldots, \boldsymbol{A}^{m-1}\boldsymbol{b}$ are adequate approximations of the eigenvectors corresponding to the $m$-largest eigenvalues. Using an Arnoldi iteration, a basis of $\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{b})$ can be determined, cf. [10]. The projection of $\mathbf{f}(\boldsymbol{A})$ onto this basis is then approximated by projecting the argument $\boldsymbol{A}$ itself onto this basis. Therefore the initial problem of the evaluation of a function with $N \times N$ range has been reduced to the evaluation of a function with $m \times m$ range, where $m \ll N$ is the length of the Krylov basis. The remaining low-dimensional function itself can then be evaluated efficiently, e.g. via an eigendecomposition. This can trivially be parallelized by parallelizing the elementary matrix and vector operations as noted in [9].

In contrast to the classical (polynomial) Krylov subspace approximation described so far, in a so-called rational Krylov sub-projection, the action $\mathbf{f}(\boldsymbol{A})\boldsymbol{b} \approx \mathbf{r}_m(\boldsymbol{A})\boldsymbol{b}$ is approximated with a rational function of the form $\mathbf{r}_m = \mathbf{p}_{m-1}/\mathbf{q}_{m-1}$ in which the denominator polynomial $\mathbf{q}_{m-1}$ of order $m-1$ is given by $\mathbf{q}_{m-1}(\boldsymbol{A}) = \prod_{i=1}^{m-1}(\mathbf{1} - \boldsymbol{A}/\zeta_i)$ with poles $\zeta_1, \ldots, \zeta_{m-1} \in \mathbb{C} \cup \{\infty\}$ unequal from the eigenvalues of $\boldsymbol{A}$. The $m$-dimensional rational Krylov subspace is then defined by

$$\mathcal{Q}_m(\boldsymbol{A}, \boldsymbol{b}) := \mathbf{q}_{m-1}^{-1}(\boldsymbol{A})\,\mathsf{span}\left(\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \ldots, \boldsymbol{A}^{m-1}\boldsymbol{b}\right).$$

A basis of $\mathcal{Q}_m(\boldsymbol{A}, \boldsymbol{b})$ can easily be determined using a slightly modified version of the Arnoldi iteration, the so-called rational Arnoldi iteration. If all poles are set to infinity, $\mathcal{Q}_m(\boldsymbol{A}, \boldsymbol{b})$ reduces to the rational Krylov subspace $\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{b})$ and the rational Arnoldi iteration to the classical one. For other intelligent choices for the poles, it offers the option to achieve results of equal quality with a significantly smaller dimension $m$ compared to the classical Krylov subspace method. In general, to find these optimal poles, the following nonlinear optimization problem has to be solved:

$$\text{Find } \mathbf{q}_{m-1}(\cdot) = \prod_{i=1}^{m-1}(\mathbf{1} - \cdot/\zeta_i) \text{ such that } \max_{\Delta t\in[\Delta t_i, \Delta t_f]} \mathsf{Error}\left(\mathbf{f}(\cdot), \mathbf{q}_{m-1}\right) \text{ is minimal.}$$

As described in [1], this can be simplified and finally solved using a global search on a coarse discretization over a given interval $[\Delta t_i, \Delta t_f]$.

**Parallelization of Gautschi-type Exponential Integrators**

The output of the (rational) Arnoldi iteration is usually given by a matrix $\boldsymbol{V}_m$, which contains the basis of $\mathcal{Q}_m(\boldsymbol{A}, \boldsymbol{b})$ and a Hessenberg matrix $\boldsymbol{H}_m$, cf. [10]. According to [5], this can be used to compute $\boldsymbol{w} = \mathbf{f}\,(t\boldsymbol{A})\,\boldsymbol{b}$ using the approximation

$$\boldsymbol{w} \approx \boldsymbol{V}_m\, \mathbf{f}(\boldsymbol{H}_m \boldsymbol{K}_m^{-1})\, \boldsymbol{V}_m^* \boldsymbol{b}, \tag{5}$$

in which the shifted Hessenberg matrix $\boldsymbol{K}_m$ is given by $\boldsymbol{K}_m = \mathbf{1}_m + \boldsymbol{H}_m\, \mathsf{diag}(\zeta_1^{-1}, \ldots, \zeta_p^{-1})$.
In terms of computation time, the most expensive part is the determination of $\boldsymbol{V}_m$ and $\boldsymbol{H}_m$. In contrast to the classical Arnoldi iteration, the rational formulation requires solutions of linear systems of the form $(\mathbf{1} - \boldsymbol{A}/\zeta_j)\boldsymbol{x}_j = \boldsymbol{A}\boldsymbol{y}_j$ to evaluate the denominator polynomial $\mathbf{q}_{m-1}$. In this context, $\boldsymbol{y}_j = \sum_{i=1}^{j} u_{i,j}\boldsymbol{v}_i$ denotes the initial basis vector of the current iteration given as a linear combination of previous computed basis vectors $\boldsymbol{v}_i$. Therefore, the parallelization setup is defined by the elements of $\boldsymbol{U}_m := (u_{i,j})_{i,j\in\{1,\ldots,m\}}$. E.g. in the case of a diagonal structure of $\boldsymbol{U}_m$, it leads to a sequential algorithm, whereas a structure, in which only the first row is unequal to zero, allows to compute all basis vectors in parallel.
We make use of a setup, in which $\boldsymbol{U}_m$ is separated in $p^2$ identical blocks with $p|m$. Only the first row of each diagonal block in unequal to zero, so that it allows for the parallel computation of $p$ basis vectors. We use $p$ pairwise different poles $(\zeta_1, \ldots, \zeta_p)$ and execute the parallel computation for $m/p$ times using the same set of poles to determine the Krylov basis. The matrix functions in Eq. (4) are then computed using Eq. (5).

# References


[1] R.-U. Börner, O. G. Ernst, and S. Güttel. Three-Dimensional Transient Electromagnetic Modeling Using Rational Krylov Methods, 2014.

[2] P. Deuflhard. A study of extrapolation methods based on multistep schemes without parasitic solutions. *Journal of Applied Mathematics and Physics*, 30:177–189, 1979.

[3] B. García-Archilla, J. M. Sanz-Serna, and R. D. Skeel. Long-time-step methods for oscillatory differential equations. *SIAM Journal on Scientific Computing*, 20:930–963, 1999.

[4] W. Gautschi. Numerical integration of ordinary differential equations based on trigonometric polynomials. *Numerische Mathematik*, 3:381–397, 1961.

[5] S. Güttel. Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection. *GAMM-Mitteilungen*, 36(1):8–31, 2013.

[6] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, volume 31 of *Springer Series in Computational Mathematics*. Springer, 2006.

[7] J. Hersch. Contribution à la méthode des équations aux différences. *Zeitschrift für Angewandte Mathematik und Physik*, 9:129–180, 1958.

[8] M. Hochbruck, C. Lubich, and H. Selfhofer. Exponential integrators for large systems of differential equations. *SIAM Journal on Scientific Computing*, 19(5):1552–1574, 1998.

[9] D. L. Michels, G. A. Sobottka, and A. G. Weber. Exponential integrators for stiff elastodynamic problems. *ACM Transactions on Graphics*, 33(1):7:1–7:20, 2 2014.

[10] Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 29:209–228, 1992.

[11] R. B. Sidje. EXPOKIT: A software package for computing matrix exponentials. *ACM Transactions on Mathematical Software*, 24(1):130–156, 1998.