Nonlinear Elasticity for Mesh Deformation with High-Order Discontinuous Galerkin Methods for the Navier-Stokes Equations on Deforming Domains

Bradley Froehle and Per-Olof Persson

Abstract We present a numerical framework for simulation of the compressible Navier-Stokes equations on problems with deforming domains where the boundary motion is prescribed by moving meshes. Our goal is a high-order accurate, efficient, robust, and general purpose simulation tool. To obtain this, we use a discontinuous Galerkin space discretization, diagonally implicit Runge-Kutta time integrators, and fully unstructured meshes of triangles and tetrahedra. To handle the moving boundaries, a mapping function is produced by first deforming the mesh using a neo-Hookean elasticity model and a high-order continuous Galerkin FEM method. The resulting nonlinear equations are solved using Newton's method and a robust homotopy approach. From the deformed mesh, we compute grid velocities and deformations that are consistent with the time integration scheme. These are used in a mapping-based arbitrary Lagrangian-Eulerian formulation, with numerically computed mapping Jacobians which satisfy the geometric conservation law. We demonstrate our methods on a number of problems, ranging from model problems that confirm the high-order accuracy to the flow in domains with complex deformations.

1 Introduction

Over the last decade, high-order accurate methods such as discontinuous Galerkin (DG) methods [3, 6] have become increasingly popular for computational fluid dynamics simulations [15]. One of the main reasons for this popularity is that the schemes produce stable discretizations of conservation laws on fully unstructured meshes of tetrahedral elements, with arbitrary orders of accuracy. More recently, they have also been applied to problems with moving boundaries and deforming domains [11], for applications such as flapping flight simulations [16].

A popular technique for handling the deforming domains is the Arbitrary Lagrangian Eulerian (ALE) method [14, 4, 8], which allows for a deforming grid by using a discretization which accounts for the grid motion. While usually formulated in a moving grid framework, in [11] it was demonstrated how these schemes can

Bradley Froehle · Per-Olof Persson
Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720-3840, USA,
e-mail: brad.froehle@gmail.com, persson@berkeley.edu

be used in a DG setting with a mapping-based formulation and a fixed reference domain, to easily obtain high-order accuracy in both space and time.

For complex geometries and deformations, this domain mapping has to be solved for numerically using some type of mesh deformation scheme. In this work, we show how to do this using a quasi-static nonlinear elasticity approach, similar to the one used for high-order curved mesh generation in [13]. We show how to use the resulting deformed meshes in a DG-based ALE scheme, and how to derive discretely consistent grid velocities for diagonally implicit Runge-Kutta methods. Using a non-trivial test problem we can demonstrate optimal order convergence. We also show that a lower-order element-wise mapping is preferable to a full isoparametric mapping, which is convenient in the case of rigid body motions. Finally we show how the scheme has been applied to two complex flapping flight applications.

2 Governing Equations

The fluid flow is governed by the compressible Navier-Stokes equations, which can be written in conservation form as:

$$\frac{\partial}{\partial t}(\rho) + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \tag{1}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j + p\delta_{ij}) = \frac{\partial}{\partial x_j}\tau_{ij} \qquad \text{for } i = 1,2,3 \tag{2}$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_j}(\rho u_j E + u_j p) = \frac{\partial}{\partial x_j}(-q_j + u_i\tau_{ij}) \tag{3}$$

where the conserved variables are the fluid density ρ, momentum in the j-th spatial coordinate direction ρu_j, and total energy ρE. The viscous stress tensor and heat flux are given by

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_j}\delta_{ij}\right) \quad \text{and} \quad q_j = -\frac{\mu}{\text{Pr}}\frac{\partial}{\partial x_j}\left(E + \frac{p}{\rho} - \frac{1}{2}u_k u_k\right). \tag{4}$$

Here, μ is the viscosity coefficient and $\text{Pr} = 0.72$ is the Prandtl number which we assume to be constant. For an ideal gas, the pressure p has the form $p = (\gamma-1)\rho\,(E - u_k u_k/2)$, where γ is the adiabatic gas constant. We write the system of conservation laws (1)-(3) in vector form as

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla\cdot\boldsymbol{f}(\boldsymbol{u},\nabla\boldsymbol{u}) = \boldsymbol{0}, \tag{5}$$

where $\boldsymbol{u} = [\rho, \rho u_1, \rho u_2, \rho u_3, \rho E]$ is the vector of conserved quantities and $\boldsymbol{f}$ is the corresponding flux function. In our examples we impose two types of boundary conditions – free-stream conditions and adiabatic no-slip wall conditions.

The deformable domains are handled through an Arbitrary Lagrangian Eulerian (ALE) formulation. A point $\boldsymbol{X}$ in a fixed reference domain V is mapped to $\boldsymbol{x}(\boldsymbol{X},t)$ in a time-varying domain $v(t)$. The deformation gradient $\boldsymbol{G}$, mapping (or mesh) velocity $\boldsymbol{\nu}$, and mapping Jacobian g are defined as

$$\boldsymbol{G} = \nabla_{\boldsymbol{X}} \boldsymbol{x}, \qquad \boldsymbol{\nu} = \frac{\partial \boldsymbol{x}}{\partial t}, \qquad g = \det \boldsymbol{G} \tag{6}$$

The system (5) in the physical domain $(\boldsymbol{x},t)$ can then be rewritten as a system of conservation laws in the reference domain $(\boldsymbol{X},t)$

$$\frac{\partial \boldsymbol{U}}{\partial t} + \nabla_{\boldsymbol{X}} \cdot \boldsymbol{F}(\boldsymbol{U}, \nabla_{\boldsymbol{X}} \boldsymbol{U}) = \boldsymbol{0} \tag{7}$$

where the conserved quantities in reference space are $\boldsymbol{U} = g\boldsymbol{u}$ with the fluxes $\boldsymbol{F} = g\boldsymbol{G}^{-1}\boldsymbol{f} - \boldsymbol{u}\boldsymbol{G}^{-1}\boldsymbol{\nu}$, and the gradient of the solution is given by

$$\nabla \boldsymbol{u} = (\nabla_{\boldsymbol{X}}(g^{-1}\boldsymbol{U}))\boldsymbol{G}^{-T} = (g^{-1}\nabla_{\boldsymbol{X}}\boldsymbol{U} - \boldsymbol{U}\nabla_{\boldsymbol{X}}(g^{-1}))\boldsymbol{G}^{-T}. \tag{8}$$

For more details, including a convenient method for satisfying the Geometric Conservation Law (GCL) by introducing an additional set of ODEs, see [11].

3 Numerical Schemes

Discretization of the Navier-Stokes equations

Our 3DG flow solver is based on the high-order Discontinuous Galerkin (DG) method with tetrahedral mesh elements and nodal basis functions. For simplicity, we change the notation and use lower-case symbols for the solution $\boldsymbol{u}$, and we omit the subscripts on the derivative operators. We also split the fluxes into an inviscid component $\boldsymbol{F}^i(\boldsymbol{u})$ and a viscous component $\boldsymbol{F}^v(\boldsymbol{u}, \nabla \boldsymbol{u})$. The ALE system (7) can then be written in a split form as

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{F}^i(\boldsymbol{u}) - \nabla \cdot \boldsymbol{F}^v(\boldsymbol{u}, \boldsymbol{q}) = \boldsymbol{0}, \tag{9}$$

$$\nabla \boldsymbol{u} = \boldsymbol{q}. \tag{10}$$

Next, we introduce a computational mesh $\mathscr{T}_h = \{K\}$ of the reference domain Ω, and the finite element spaces $\mathscr{V}_h^p$ and Σ_h^p:

$$\mathscr{V}_h^p = \{\boldsymbol{v} \in [L^2(\Omega)]^5 \mid \boldsymbol{v}|_K \in [\mathscr{P}_p(K)]^5 \ \forall K \in \mathscr{T}_h\}, \tag{11}$$

$$\Sigma_h^p = \{\boldsymbol{\tau} \in [L^2(\Omega)]^{5\times 3} \mid \boldsymbol{\tau}|_K \in [\mathscr{P}_p(K)]^{5\times 3} \ \forall K \in \mathscr{T}_h\}, \tag{12}$$

where $\mathscr{P}_p(K)$ is the space of polynomial functions of degree at most $p \geq 1$ on K, and 3 and 5 refer to the dimension and number of solution components of the Navier-

Stokes equations in three dimensions. We multiply the system of equations (9)-(10) by test functions $\boldsymbol{v}, \boldsymbol{\tau}$ and integrate by parts. Our semi-discrete DG formulation is then expressed as: find $\boldsymbol{u}_h \in \mathscr{V}_h^p$ and $\boldsymbol{q}_h \in \Sigma_h^p$ such that for all $K \in \mathscr{T}_h$, we have

$$\int_K \frac{\partial \boldsymbol{u}_h}{\partial t} \cdot \boldsymbol{v}\, dx + \int_K \left(\boldsymbol{F}^i(\boldsymbol{u}_h) - \boldsymbol{F}^v(\boldsymbol{u}_h, \boldsymbol{q}_h) \right) : \nabla \boldsymbol{v}\, dx - \oint_{\partial K} \left(\widehat{\boldsymbol{F}^i(\boldsymbol{u}_h)} - \widehat{\boldsymbol{F}^v(\boldsymbol{u}_h, \boldsymbol{q}_h)} \right) \cdot \boldsymbol{v}\, ds = \boldsymbol{0}, \forall \boldsymbol{v} \in [\mathscr{P}_p(K)]^5 \quad (13)$$

$$\int_K \boldsymbol{q}_h : \boldsymbol{\tau}\, dx + \int_K \boldsymbol{u}_h \cdot (\nabla \cdot \boldsymbol{\tau})\, dx - \oint_{\partial K} (\hat{\boldsymbol{u}}_h \otimes \boldsymbol{n}) : \boldsymbol{\tau}\, ds = \boldsymbol{0}, \forall \boldsymbol{\tau} \in [\mathscr{P}_p(K)]^{5\times 3} \quad (14)$$

To complete the description we need to specify the numerical fluxes for all element boundaries ∂K. The inviscid fluxes $\widehat{\boldsymbol{F}^i(\boldsymbol{u}_h)}$ are computed using a standard approximate Riemann solver and the modification for our ALE formulation described in [11]. For the viscous fluxes $\widehat{\boldsymbol{F}_h^v}$, $\hat{\boldsymbol{u}}_h$, we use a formulation based on the Compact DG (CDG) method [10]. At a boundary face, we impose either far field or no-slip conditions weakly through the fluxes.

Using a standard finite element procedure, we obtain the semi-discrete form of our equations:

$$M \frac{d\bar{\boldsymbol{u}}}{dt} = \bar{\boldsymbol{r}}(\bar{\boldsymbol{u}}), \quad (15)$$

for discrete solution vector $\bar{\boldsymbol{u}}$, mass matrix $\boldsymbol{M}$, and residual function $\bar{\boldsymbol{r}}(\bar{\boldsymbol{u}})$. We integrate this system of ODEs in time using Diagonally Implicit Runge-Kutta (DIRK) methods [1], where the solution is advanced from time t_n to t_{n+1} by:

$$M\bar{\boldsymbol{k}}_i = \bar{\boldsymbol{r}} \left(t_n + c_i \Delta t, \bar{\boldsymbol{u}}_n + \Delta t \sum_{j=1}^{s} a_{ij} \bar{\boldsymbol{k}}_j \right), \quad i = 1, \ldots, s \quad (16)$$

$$\bar{\boldsymbol{u}}_{n+1} = \bar{\boldsymbol{u}}_n + \Delta t \sum_{j=1}^{s} b_j \bar{\boldsymbol{k}}_j. \quad (17)$$

We consider a variety of DIRK schemes, but in particular the 2- and 3-stage L-stable schemes presented in [1]. Note that the implicit scheme requires inversion of matrices of the form $M - a_{ii}\Delta t d\bar{\boldsymbol{r}}/d\bar{\boldsymbol{u}}$. This is accomplished by using a preconditioned parallel Newton-Krylov solver, see [12] for details.

Computation of gradients and mesh velocities

The ALE equations (7) require the mesh deformation gradient $\boldsymbol{G}$, which is computed as the gradient of the mesh position $\boldsymbol{x}$. The $\nabla_X g^{-1}$ term is computed as

$$\nabla_X g^{-1} = \frac{-1}{g^2} \nabla_X g = \frac{-1}{g^2} \nabla_X \det \boldsymbol{G} \quad (18)$$

where the gradient of $\det \boldsymbol{G}$ is computed component-wise using the formula

$$\frac{d\det(\boldsymbol{G})}{dX_i} = \det(\boldsymbol{G})\,\mathrm{tr}\left(\boldsymbol{G}^{-1}\frac{d\boldsymbol{G}}{dX_i}\right) \tag{19}$$

with $d\boldsymbol{G}/dX_i$ computed numerically.

Next we consider the computation of the mesh velocity $\boldsymbol{\nu} = \partial \boldsymbol{x}/\partial t$. Depending on the specifics of the problem there are a few different ways in which the mesh velocities may be calculated. In the simplest case the mesh motion may be given as an analytic function of time, in which case we may simply take the derivative to compute the mesh velocity. For example, if the mesh position is given by an interpolation of a deformation of the boundary using radial basis functions [2], it is often natural to use the same interpolation process to interpolate boundary deformation velocities into mesh velocities.

However, if only numerical values of the mesh position are available we must resort to a numerical differentiation procedure to compute the mesh velocity. It is desirable to use a definition which uses specific details of the time integrator used for the time integration. We say a method of calculating mesh velocities is consistent if, when integrated using the numerical method they recover the numerical mesh positions. This was done in, for example, [9] for several explicit multistep and Runge-Kutta methods. Here we show an extension of this idea to the case of diagonally implicit Runge-Kutta methods.

Given the mesh position $\boldsymbol{x}_i$ at stages $i = 1,\ldots,s$, we say the mesh velocities $\boldsymbol{\nu}_i$ at stages $i = 1,\ldots,s$ are *stage consistent* if

$$\boldsymbol{x}_i = \boldsymbol{x}_0 + \Delta t \sum_{j=1}^{s} a_{ij}\boldsymbol{\nu}_j, \quad i = 1,\ldots,s \tag{20}$$

where $\boldsymbol{x}_0$ is the initial mesh position and a_{ij} are the Runge-Kutta coefficients.

In the case when A is of full rank, i.e., a fully implicit Runge-Kutta or diagonally implicit Runge-Kutta method, some algebraic manipulation allows us to write the stage mesh velocity as a linear combination of the mesh positions

$$\boldsymbol{\nu}_i = \sum_{j=1}^{s} (A^{-1})_{ij}\frac{\boldsymbol{x}_j - \boldsymbol{x_0}}{\Delta t}, \quad i = 1,\ldots,s. \tag{21}$$

For a diagonally implicit Runge-Kutta method A^{-1} is lower triangular so each stage mesh velocity may be calculated using only mesh positions from that and previous stages. This preserves an obvious time dependency relationship and may be desirable, especially in cases when the stage mesh position is calculated on-the-fly from current stage variables as in the case of a fluid-structure interaction problem [5].

If the first stage of the Runge-Kutta scheme is explicit, say in an ESDIRK method, the coefficient matrix A will not be invertible and thus a different approach is required. In fact, it is clear that the stage mesh velocities $\boldsymbol{\nu}_i$ are not even uniquely defined in terms of the stage mesh positions $\boldsymbol{x}_i$. In this case it is natural to require

an initial mesh velocity $\boldsymbol{\nu}_0$. The first (explicit) stage mesh velocity $\boldsymbol{\nu}_1$ is set to this value and mesh velocities at later stages are then uniquely given by eq. (20). The ESDIRK schemes we have considered all have the first same as last property, that is, the final stage coefficients are the same as the weights, and so it is natural to use the mesh velocity at the final stage $\boldsymbol{\nu}_s$ of one timestep as the initial mesh velocity in the following timestep.

4 Mesh Deformation

For the mesh deformation, we use a quasi-static hyperelastic neo-Hookean formulation [7]. The deformation is given by a mapping $\boldsymbol{x}(\boldsymbol{X})$ which maps a point $\boldsymbol{X}$ in the unstretched reference configuration Ω to its location $\boldsymbol{x}$ in the deformed configuration. We differentiate $\boldsymbol{x}$ with respect to space to obtain the *deformation gradient tensor* $\boldsymbol{G}$ as $\boldsymbol{G} = \nabla_{\boldsymbol{X}} \boldsymbol{x}(\boldsymbol{X})$. The governing equations are then given by

$$-\nabla \cdot \boldsymbol{P}(\boldsymbol{G}) = \boldsymbol{b} \qquad \text{in } \Omega, \tag{22}$$

$$\boldsymbol{x} = \boldsymbol{x}_D \qquad \text{on } \Gamma, \tag{23}$$

where $\boldsymbol{P}$ is the first Piola-Kirchhoff stress tensor and $\boldsymbol{b}$ is an external body force per unit reference volume, which we typically assume is zero. On the boundary of the domain $\Gamma = \partial\Omega$ we have assumed Dirichlet boundary conditions, i.e., specified material positions $\boldsymbol{x}_D$.

In this work we use a compressible neo-Hookean material model, with first Piola-Kirchhoff stress tensor given by [7]

$$\boldsymbol{P}(\boldsymbol{G}) = \frac{\partial W}{\partial \boldsymbol{G}} = \mu J^{-2/3} \left(\boldsymbol{G} - \frac{1}{3} \operatorname{tr}(\boldsymbol{G}\boldsymbol{G}^T) \boldsymbol{G}^{-T} \right) + \kappa (J-1) J \boldsymbol{G}^{-T}, \tag{24}$$

where the constants μ and κ are the shear and bulk modulus of the material. For two-dimensional problems we use a plane strain formulation in which we treat the stretching in the third dimension as constant.

To develop a finite element formulation for (22)-(23), we define the space of continuous piecewise polynomials of degree p:

$$\mathscr{V}_h^p = \left\{ \boldsymbol{v} \in [\mathscr{C}_0(\Omega)]^3 \;\middle|\; \boldsymbol{v}|_K \in [\mathscr{P}_p(K)]^3 \; \forall K \in \mathscr{T}_h \right\}, \tag{25}$$

where the domain Ω is divided into elements $\mathscr{T}_h = \{K\}$, and $\mathscr{P}_p(K)$ is the space of polynomial functions of degree at most $p \geq 1$ on K. Furthermore, we define the subspaces of functions in $\mathscr{V}_h^p$ that satisfy the non-homogeneous as well as the homogeneous Dirichlet boundary conditions:

$$\mathscr{V}_{h,D}^p = \left\{ \boldsymbol{v} \in \mathscr{V}_h^p \;\middle|\; \boldsymbol{v}|_{\partial V} = \boldsymbol{x}_D^p \right\}, \tag{26}$$

$$\mathscr{V}_{h,0}^p = \left\{ \boldsymbol{v} \in \mathscr{V}_h^p \;\middle|\; \boldsymbol{v}|_{\partial V} = 0 \right\}. \tag{27}$$

Here, $\boldsymbol{x}_D^p$ is a suitable projection of $\boldsymbol{x}_D$ onto the space of piecewise polynomials of order p defined over ∂V. By multiplying (22) by an arbitrary test function $\boldsymbol{z} \in \mathscr{V}_{h,0}^p$, integrating over the domain V, and integrating by parts, we obtain our finite element formulation: find $\boldsymbol{x}_h \in \mathscr{V}_{h,D}^p$ such that for all $\boldsymbol{z} \in \mathscr{V}_{h,0}^p$,

$$\int_V \boldsymbol{P}(\boldsymbol{G}(\boldsymbol{x}_h)) : \nabla \boldsymbol{z}\, dV = \int_V \boldsymbol{b} \cdot \boldsymbol{z}\, dV. \tag{28}$$

This system of equations is generated using standard finite element techniques. Using nodal basis functions, the computed elemental residuals are assembled into a global discrete system of equations $\bar{\boldsymbol{r}}(\bar{\boldsymbol{x}}) = \boldsymbol{0}$. We solve this system using a standard Newton method, which involves the Jacobian matrix $\boldsymbol{K} = \partial \bar{\boldsymbol{r}} / \partial \bar{\boldsymbol{x}}$ which is evaluated for each element and assembled into a global matrix. The prescribed displacement at the boundary nodes is imposed by elimination of the corresponding variables from the system of equations. The linear systems that arise are solved using a direct sparse solver. For problems with complex deformations, we use the simple homotopy approach described in [13] to obtain global convergence.

5 Results

Deformed mesh quality

As a test problem to demonstrate the quality of the nonlinear elasticity based mesh deformation, we consider a square with a smaller square removed from the center:

$$\Omega = [0.0, 1.0]^2 \setminus [0.4, 0.6]^2. \tag{29}$$

The domain is triangulated in a structured fashion using isoparametric elements of polynomial degree 2. We fix the outer boundary of the domain and rotate the inner boundary about the center, $[0.5, 0.5]^T$, by an angle θ. Clearly, for increasing θ any deformation strategy will eventually fail and produce invalid elements. However, for moderate angles this is a good test case for comparing different methods.

We first perform the mesh deformation using the commonly used radial basis function interpolation [2]. Here we seek an interpolant giving the deformed mesh position $\boldsymbol{x}$ as function of the position $\boldsymbol{X}$ in the reference mesh, of the form

$$\boldsymbol{x}(\boldsymbol{X}) = \sum_{j=1}^{n} \boldsymbol{\alpha}_j \phi_j(\|\boldsymbol{X} - \boldsymbol{X}_j\|_2 / r_j) + \boldsymbol{p}(\boldsymbol{X}) \tag{30}$$

where $\boldsymbol{X}_j$ are a set of control points, ϕ_j radial basis functions, r_j characteristic radii, and $\boldsymbol{p}$ a linear polynomial. The coefficients $\boldsymbol{\alpha}_j$ and coefficients of the polynomial $\boldsymbol{p}$ are found by imposing the value of $\boldsymbol{x}$ at the control points $\boldsymbol{X}_j$, and additionally requiring that the function preserves polynomial deformations of degree less than or equal to the degree of $\boldsymbol{p}$. We solve the resulting linear system using a direct solver.

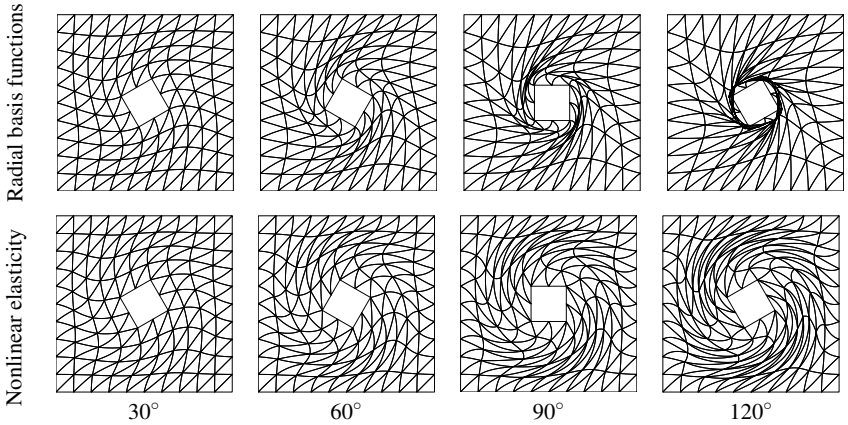

Fig. 1 Mesh deformation using radial basis function interpolation (top) and using the quasi-static nonlinear elasticity method (bottom).

There are many choices of radial basis functions, but on the recommendation of [2] we use a C^2 compactly supported function

$$\phi(r) = \begin{cases} (1-r)^4(4r+1) & \text{if } 0 \le r \le 1 \\ 0 & \text{if } 1 \le r. \end{cases} \tag{31}$$

with characteristic radius 1, which gave the best results for several different RBF interpolants and radii examined. The resulting deformed mesh for rotations of 30°, 60°, 90°, and 120° are shown in Fig. 1 (top). Here we see that this mesh deformation method does a very good job with the small deformation (30°), but has some difficulty with larger deformations. In particular, some elements have already inverted (i.e., the determinant of the local Jacobian mapping is negative) by 90°.

In addition one can easily show that a non-inverting deformation of a 180° rotation of the inner square is not possible using this technique for any choice of radial basis function interpolant. To see this, recall that the deformed position of any node depends linearly on the positions of the boundary nodes. Since a $+180°$ and $-180°$ rotation of the inner square would lead to the same locations of the boundary nodes, the RBF interpolant is unable to distinguish between these two cases. In particular, a curve connecting the left outer boundary to the left inner boundary in the undeformed mesh would have to pass both under the square in the $+180°$ rotation and under the square in the $-180°$ rotation, which is not possible.

In the bottom plots of Fig. 1 we repeat the same experiment, this time using the nonlinear elasticity deformation method. Here we set $\nu = 0.40$ and a spatially varying E according to

$$E(x) = 1 + \frac{100}{1 + (d(x)/d_0)^2} \tag{32}$$

where $d_0 = 0.05$ and $d(x) = \max\{0.0, \min(\text{dist}(x,\Gamma_{in}) - d_0, \text{dist}(x,\Gamma_{out}) + 2d_0)\}$. Here, Γ_{in} and Γ_{out} are the inner and outer boundaries. This expression for E was chosen to cause more deformation to occur in the intermediate region between the inner and the outer boundaries, which is desirable. As the figure shows, the resulting mesh still has not inverted, even at a rotation of 120°, although the element quality does become quite poor for the larger rotations. These results are significantly better than what we could achieve even with the best possible parameters for the RBF deformation.

Because the deformation equations are nonlinear, the system may exhibit multiple solutions for a given configuration of the boundary. In particular, the zero that we find is going to be dependent upon the initial approximation in the Newton solver. In particular this means that we are in principle able to construct deformed meshes corresponding to $+180°$ and $-180°$ rotations of the inner boundary using essentially a homotopy of intermediate rotations.

Convergence test, expanding pressure wave

To study the accuracy of the Arbitrary Lagrangian-Eulerian formulation, we consider a case with a specified analytic mesh deformation and compare the spatial convergence for several deformation strategies. As a non-trivial test problem, we consider a viscous flow problem with a small Gaussian perturbation in the density and the pressure of an otherwise constant state.

As the domain we choose $\Omega = [0,1]^2$ with far-field boundary conditions on the left, bottom, and right walls and an adiabatic no-slip condition on the top wall. The momentum is initialized as $\rho\boldsymbol{u} = 0$, and the spatially varying initial density and pressure are $\rho = \rho_\infty \varphi(x)$ and $p = p_\infty \varphi(x)$, respectively, where

$$\varphi(x) = 1 + d_0 \exp(\|x - x_0\|_2^2 / r_0^2) \tag{33}$$

and the non-dimensionalized far-field density $\rho_\infty = 1$. The far-field pressure p_∞ is calculated using the non-dimensionalized sound speed $a_\infty = 5$. The perturbation parameters where chosen as $d_0 = 0.1$, $r_0 = 0.1$, and $x_0 = [0.5, 0.7]^T$.

The fluid is modeled using the compressible Navier-Stokes equations (1)–(3), with dynamic viscosity $\mu = 1/1000$. The background mesh was deformed using an analytic mapping

$$x(X,Y,t) = X + A\sin(2\pi X)\sin(2\pi Y)\sin(2\pi f t), \tag{34}$$

$$y(X,Y,t) = Y + A\sin(2\pi X)\sin(2\pi Y)\sin(4\pi f t), \tag{35}$$

with amplitude $A = 0.05$ and frequency $f = 20$.

For the time-integration, we use an explicit RK4 scheme with a sufficiently small Δt so that the spatial errors are dominating. We integrate until a final time of $T = 1/20$, which is one entire period of the mesh deformation so that the mesh starts in an undeformed configuration at time $t = 0$ and returns to an undeformed configuration

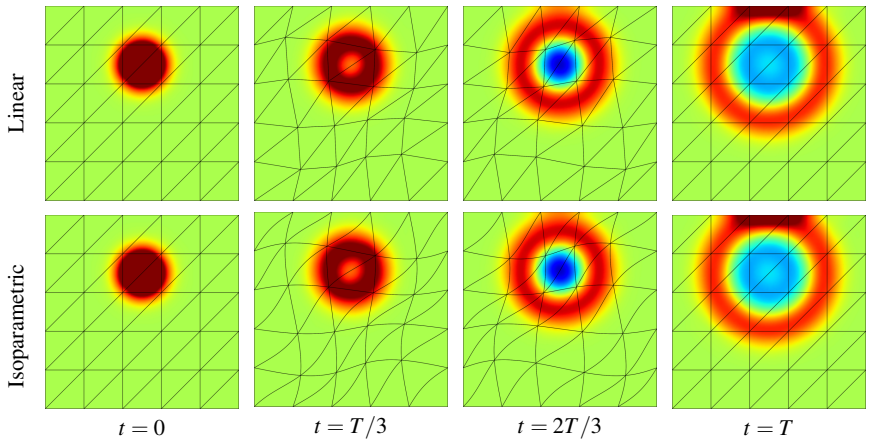

Fig. 2 An expanding pressure wave on a deforming mesh using a linear deformation (top) and an isoparametric deformation (bottom), for polynomial degrees $p = 5$. (Pressure).

at time $t = T$. This allows us to measure the accuracy of the ALE mapping by comparing the numerical solution of the problem at $t = T$ to one obtained on a non-deforming mesh.

The domain Ω is discretized using a regular grid of triangles with element size h, and we use polynomial degrees $p = 1$ through 5 within each element. Numerically the mesh deformation is represented on each element using either a linear $p = 1$ representation or an isoparametric representation. A time series of the solution on two meshes is shown in Fig. 2.

We observe that both deformation strategies are able to accurately capture the radiating pressure wave. Notice that when we represent the mesh deformation using $p = 1$ elements the resulting map $\boldsymbol{x}(\boldsymbol{X},t)$ is piecewise linear and hence the ALE formulation in Sec. 2 simplifies significantly as the deformation gradient $\boldsymbol{G}$ and mapping determinant g are both constant. This also simplifies the calculation of the viscous derivative as an entire term $\nabla_{\boldsymbol{X}}(g^{-1})$ vanishes. However, a $p = 1$ mesh deformation representation is likely not able to capture complicated boundary motions as accurately as the isoparametric $p = 5$ representation.

The relative accuracy of using a $p = 1$ deformation instead of an isoparametric can be discussed. We would expect the linear $p = 1$ mapping to produce slightly better results because it introduces less variations in the solution fields. This intuition is reflected in a numerical convergence plot which is shown in Fig. 3. Here we measure the error in the solution at $t = T$ in the discrete maximum norm for a non-deforming fixed mesh, a $p = 1$ deformation, and an isoparametric deformation ('Full P') for elements of order $p = 1$ through 5. In general we observe convergence orders at the expected $p+1$ rate for all the cases. For the lower p the difference in accuracy between the three methods is difficult to ascertain. However, for higher p there is a notable difference in accuracy between the three methods, with the fixed mesh being the most accurate and the isoparametric deformation being the least accurate.

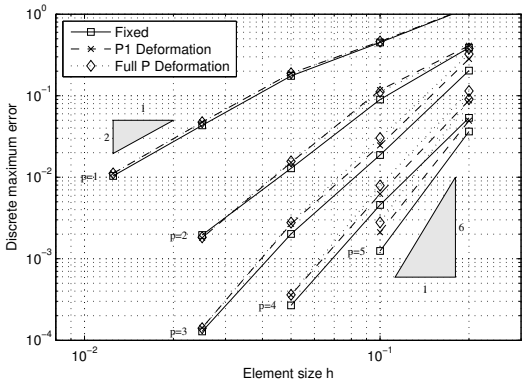

Fig. 3 Spatial convergence in the discrete maximum error at the final simulation time for an expanding pressure wave for meshes of polynomial degree $p = 1$ to 5. The deformation is either linear (P1) or isoparametric (Full P).

From this experiment we can generally recommend using a linear representation of the mesh deformation if possible. If not, the isoparametric deformation gives adequate results and is able to represent a much larger class of deformations. Mixed approaches should be feasible and represent a possible compromise.

Flapping wing applications

As two final examples, we show how our methods have been successfully applied to flapping flight problems. In Fig. 4, the simulation of a pair of flapping bat wings is shown. A representative surface mesh frame was chosen for the reference domain, and a high-quality tetrahedral mesh of the domain was generated (left plot). This mesh was then deformed for each subsequent time frame using the nonlinear elasticity approach (middle plot), and a preliminary simulation at a low Reynolds number was performed (right plot).

The second example is from [16], where several energetically optimal flapping wing designs were computed using a multi-fidelity approach. These designs were simulated using the high-fidelity DG framework presented here. Fig. 5 (top) shows the mesh deformation, and the bottom plots show flow fields from a sample design.

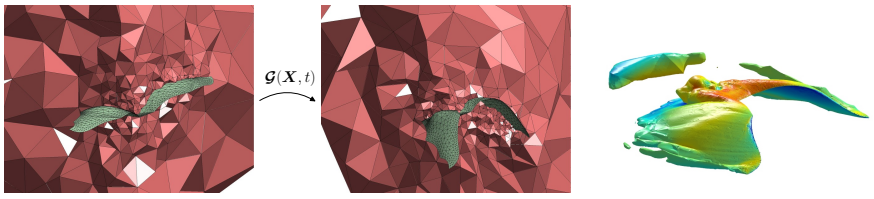

Fig. 4 A large deformation example of the flapping flight of a bat. The reference mesh (left) is deformed in time using the nonlinear elasticity approach which maintains the high-quality of the elements (middle). The right plot shows a sample solution field.

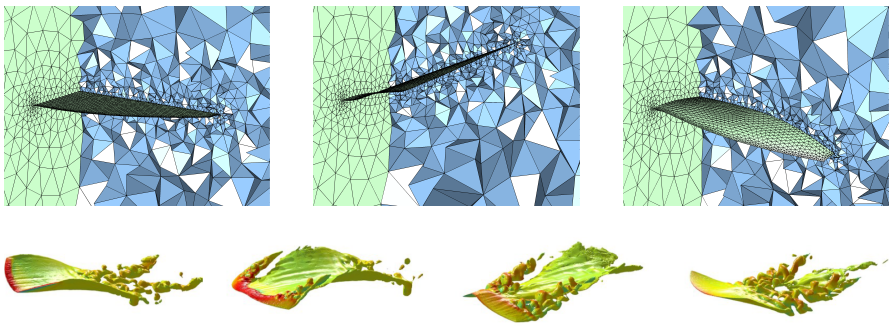

Fig. 5 High-order simulation of energetically optimal flapping wings (from [16]). The figures show a reference mesh, two deformed meshes, and some flow fields for a sample design.

References

1. Alexander, R.: Diagonally implicit Runge-Kutta methods for stiff O.D.E.'s. SIAM J. Numer. Anal. **14**(6), 1006–1021 (1977)
2. de Boer, A., van der Schoot, M.S., Bijl, H.: Mesh deformation based on radial basis function interpolation. Comput. Struct. **85**, 784–795 (2007)
3. Cockburn, B., Shu, C.W.: Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. J. Sci. Comput. **16**(3), 173–261 (2001)
4. Farhat, C., Geuzaine, P.: Design and analysis of robust ALE time-integrators for the solution of unsteady flow problems on moving grids. Comput. Methods Appl. Mech. Engrg. **193**(39-41), 4073–4095 (2004)
5. Froehle, B., Persson, P.O.: A high-order discontinuous Galerkin method for fluid-structure interaction with efficient implicit-explicit time stepping. J. Comput. Phys. **272**, 455–470 (2014)
6. Hesthaven, J.S., Warburton, T.: Nodal discontinuous Galerkin methods, *Texts in Applied Mathematics*, vol. 54. Springer, New York (2008). Algorithms, analysis, and applications
7. Holzapfel, G.A.: Nonlinear solid mechanics. John Wiley & Sons Ltd., Chichester (2000). A continuum approach for engineering
8. Lomtev, I., Kirby, R.M., Karniadakis, G.E.: A discontinuous Galerkin ALE method for compressible viscous flows in moving domains. J. Comput. Phys. **155**(1), 128–159 (1999)
9. Minoli, C.A.A., Kopriva, D.A.: Discontinuous Galerkin spectral element approximations on moving meshes. J. Comput. Phys. **230**(5), 1876–1902 (2011)
10. Peraire, J., Persson, P.O.: The compact discontinuous Galerkin (CDG) method for elliptic problems. SIAM J. Sci. Comput. **30**(4), 1806–1824 (2008)
11. Persson, P.O., Bonet, J., Peraire, J.: Discontinuous Galerkin solution of the Navier-Stokes equations on deformable domains. Comput. Methods Appl. Mech. Engrg. **198**(17–20), 1585–1595 (2009)
12. Persson, P.O., Peraire, J.: Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations. SIAM J. Sci. Comput. **30**(6), 2709–2733 (2008)
13. Persson, P.O., Peraire, J.: Curved mesh generation and mesh refinement using Lagrangian solid mechanics. In: 47th AIAA ASME, Orlando, Florida (2009). AIAA-2009-949
14. Venkatasubban, C.S.: A new finite element formulation for ALE (arbitrary Lagrangian Eulerian) compressible fluid mechanics. Internat. J. Engrg. Sci. **33**(12), 1743–1762 (1995)
15. Wang, Z., et al.: High-order CFD methods: current status and perspective. Internat. J. Numer. Methods Fluids. **72**(8), 811–845 (2013)
16. Willis, D.J., Persson, P.O.: Multiple-fidelity computational framework for the design of efficient flapping wings. AIAA J. **52**(12), 2840–2854 (2014)