Theory and implementation

The resolution of the viscous flow on the curved shell will be done in an updated Lagrangian manner. At each time step, we consider the current geometrical state of the shell as the reference surface. After computing the shell velocity, we update the geometry accordingly and consider this new geometry as the reference surface for the next time step.

As discussed in the paper, Naghdi-type shell theories (shearable models) are easier to implement that Koiter-type theories. Shearing energy will be here artificially penalized in order to approximate the thin shell model derived in the paper.

Discrete shell geometry

The discrete shell reference surface, denoted by $\boldsymbol{\varphi}_0$, is assumed to be made of an assembly of flat triangular elements and $\boldsymbol{d}_0 = \boldsymbol{n}_0$ denotes the reference shell director which we assume to initially coincide with the reference shell normal. The reference metric tensor $a = \nabla\boldsymbol{\varphi}_0^T\nabla\boldsymbol{\varphi}_0$, and curvature tensor $b = -\frac{1}{2}(\nabla\boldsymbol{\varphi}_0^T\nabla\boldsymbol{n}_0 + \nabla\boldsymbol{n}_0^T\nabla\boldsymbol{\varphi}_0)$.

Following [Hale et al., 2018], we assume the shell director to be unstretchable ($\|\boldsymbol{d}\| = 1$) so that we parametrize the unit director $\boldsymbol{d}$ by two angles (β_1, β_2), as

$$\boldsymbol{d}(\beta_1, \beta_2) = (\cos\beta_1\sin\beta_2, -\sin\beta_1, \cos\beta_1\cos\beta_2)$$

Note that such a simple parametrization of the the director prevent us from working with a whole sphere due to singularities arising at the poles. In the numerical simulations, we benefit from the cell symmetry and work with only a portion of a sphere with appropriate boundary conditions.

The strain rate tensor $\Delta_{\alpha\beta}$ and rate of change of curvature tensor $\Omega_{\alpha\beta}$ reads:

$$\boldsymbol{\Delta} = \frac{1}{2}(\nabla\boldsymbol{\varphi}_0^T\nabla\boldsymbol{U} + \nabla\boldsymbol{U}^T\nabla\boldsymbol{\varphi}_0),$$

and

$$\boldsymbol{\Omega} = -\frac{1}{2}\left(\nabla\boldsymbol{U}^T\nabla\boldsymbol{d}_0 + \nabla\boldsymbol{\varphi}_0^T\nabla\boldsymbol{\omega} + \nabla\boldsymbol{d}_0^T\nabla\boldsymbol{U} + \nabla\boldsymbol{\omega}^T\nabla\boldsymbol{\varphi}_0\right)$$

where $\boldsymbol{\omega} = d\boldsymbol{d}/dt$ is the rate of change of director. In addition, we must also consider the rate of shear strains,

$$\boldsymbol{\gamma} = \nabla\boldsymbol{\varphi}^T\boldsymbol{\omega} + \nabla\boldsymbol{U}^T\boldsymbol{d}_0$$

When $\boldsymbol{\gamma} = 0$, we recover that $\nabla\boldsymbol{\varphi}^T\boldsymbol{d} = \nabla\boldsymbol{\varphi}_0^T\boldsymbol{n}_0 = 0$, hence $\boldsymbol{d} = \boldsymbol{n}$, that is an unshearable kinematics.

Rayleigh potential and constitutive equations

To approximately enforce a thin shell kinematics, we penalize the shear strain rate term in the Rayleigh potential by adding an additional dissipation contribution

$$\phi(\boldsymbol{U},\boldsymbol{d}) = N^{\alpha\beta}\Delta_{\alpha\beta} + M^{\alpha\beta}\Omega_{\alpha\beta} + k_{s,\text{pen}}\mu T a^{\alpha\beta}\gamma_\alpha\gamma_\beta$$

where $k_{s,\text{pen}}$ is a sufficiently large penalization constant.

where the internal membrane forces and bending moments are given by:

$$N^{\alpha\beta} = 4\mu T\left(\mathcal{A}^{\alpha\beta\mu\delta}\Delta_{\mu\delta} + \frac{1}{2}a^{\alpha\beta}\left(k_d - \frac{v_p}{T}\right)\right) + T\mathcal{Q}^{\alpha\beta}\zeta$$

and

$$M^{\alpha\beta} = \mu\frac{T^3}{3}\mathcal{A}^{\alpha\beta\mu\delta}\Omega_{\mu\delta}$$

where:

$$\mathcal{A}^{\alpha\beta\mu\delta} = \frac{1}{4}(a^{\alpha\mu}a^{\beta\delta} + a^{\alpha\delta}a^{\beta\mu}) + \frac{1}{2}a^{\alpha\beta}a^{\mu\delta}$$

$Q_{\alpha\beta}$ is the nematic tensor and:

$$\mathcal{Q}^{\alpha\beta} = a^{\alpha\mu}a^{\beta\delta}Q_{\mu\delta} - a^{\alpha\beta}Q_{33}$$

Finite-element discretization

At each time step, we must solve for the shell mid-surface velocity $\boldsymbol{U}$ and the new director $\boldsymbol{d}$. In fact, we consider the rate of change $\dot{\boldsymbol{\beta}}$ of the β-parameters to be the main unknowns from which we also have

$$\boldsymbol{\omega} = \begin{bmatrix} -\sin\beta_{01}\sin\beta_{02} & \cos\beta_{01}\cos\beta_{02} \\ -\cos\beta_{01} & 0 \\ -\sin\beta_{01}\cos\beta_{02} & -\cos\beta_{01}\sin\beta_{02} \end{bmatrix}\dot{\boldsymbol{\beta}} = \boldsymbol{Q}(\boldsymbol{\beta}_0)\dot{\boldsymbol{\beta}}$$

As a result, we have a 5-dof system to solve at each time step: three velocity components: $\boldsymbol{U} = (U_1, U_2, U_3)$ and two angular rates of change: $\dot{\boldsymbol{\beta}} = (\dot{\beta}_1, \dot{\beta}_2)$. In the present work, we do not follow the MITC discretization discussed in to remove shear locking but rather use a much simpler finite-element discretization, namely:

- a quadratic continuous (P^2-Lagrange) interpolation for the velocity field $\boldsymbol{U}$
- a linear interpolation with continuity enforced at the triangle mid-sides (Crouzeix-Raviart element) for the angular rates $\dot{\boldsymbol{\beta}}$

Such a discretization choice is extremely simple and seems free of any shear locking effects as shown in [Campello et al., 2003] for nonlinear shells and [Bleyer, 2021] in the case of plate and shell limit analysis theory.

The thickness field T is discretized using a continuous linear (P^1-Lagrange) interpolation. The corresponding definitions are implemented in the `set_functions_space` method of the `ActiveShell` class.

System evolution

When considering the system evolution between time t_n and t_{n+1}, the resulting linear system for the current velocity and rate of change of director field is solved when considering the shell thickness to be fixed to its previous value i.e. $T = T_n$

$$\boldsymbol{U}_{n+1}, \dot{\boldsymbol{\beta}}_{n+1} = \arg\min_{\boldsymbol{U}, \dot{\boldsymbol{\beta}}} \mathcal{R}(\boldsymbol{U}, \dot{\boldsymbol{\beta}}; T_n)$$

where

$$\mathcal{R}(\boldsymbol{U}, \dot{\boldsymbol{\beta}}; T_n) = \int_S \phi(\boldsymbol{U}, \dot{\boldsymbol{\beta}}; T_n)\, \mathrm{d}A$$

is the system total Rayleighian.

Here, we assume no external loading of the shell, such that all shell deformations are powered by internally generated active stress. Note that, due to the linear expressions of the strain rates and the quadratic form of the dissipation potential ϕ, the resulting system is in fact linear.

Updating the geometry

We then update the geometry in an asynchronous and Lagrangian manner, by solving the discretized thickness evolution equation first, then updating the midsurface position and the director using as follows:

$$\begin{aligned} \frac{T_{n+1} - T_n}{\Delta t} &= -T_{n+1} a^{\alpha\beta} \Delta_{\alpha\beta, n+1} - T_{n+1} k_d + v_p (1 - H_n T_{n+1}) \\ \boldsymbol{d}_{n+1} &= \boldsymbol{d}(\boldsymbol{\beta}_n + \Delta t \dot{\boldsymbol{\beta}}_{n+1}) \\ \boldsymbol{x}_{n+1} &= \boldsymbol{x}_n + \Delta t\, \boldsymbol{U}_{n+1} = \end{aligned}$$

Volume constraint

In some numerical illustrations, such as cell division, we want to ensure that the total volume V of the cell is conserved during the division. To do so, we enforce the following constraint,

$$\frac{dV}{dt} = \int_{\mathcal{S}} \boldsymbol{U} \cdot \boldsymbol{n}\, \mathrm{d}A = 0.$$

Enforcement of this constraint is achieved through the introduction of a Lagrange-multiplier which can be interpreted as the cell hydrostatic pressure P forming the system Lagrangian,

$$\mathcal{L}(\boldsymbol{U}, \dot{\boldsymbol{\beta}}, P; T_n) = \mathcal{R}(\boldsymbol{U}, \dot{\boldsymbol{\beta}}; T_n) + P \int_{\mathcal{S}} \boldsymbol{U} \cdot \boldsymbol{n}\, \mathrm{d}A$$

turning the minimization problem into a saddle-point problem

$$\boldsymbol{U}_{n+1}, \dot{\boldsymbol{\beta}}_{n+1}, P_{n+1} = \arg\max_{P} \arg\min_{\boldsymbol{U}, \dot{\boldsymbol{\beta}}} \mathcal{L}(\boldsymbol{U}, \dot{\boldsymbol{\beta}}, P; T_n).$$

In the FEniCS implementation, we use a `"Real"` function space to add this additional single scalar unknown P to the system degrees of freedom.

Note that the previous approach can also be easily extended to the case where the volume change rate is imposed to a given constant $\dot{V}_{\text{imp}}$

$$\frac{dV}{dt} = \int_{\mathcal{S}} \boldsymbol{U} \cdot \boldsymbol{n} \, \mathrm{d}A = \dot{V}_{\text{imp}}$$

by considering this new Lagrangian instead:

$$\mathcal{L}(\boldsymbol{U}, \dot{\boldsymbol{\beta}}, P; T_n) = \mathcal{R}(\boldsymbol{U}, \dot{\boldsymbol{\beta}}; T_n) + P \left(\int_{\mathcal{S}} \boldsymbol{U} \cdot \boldsymbol{n} \, \mathrm{d}A - \dot{V}_{\text{imp}} \right).$$

General steps of the implementation

We give here the main aspects of the implementation of the `ActiveShell` class.

At the beginning of each time step, the `initialize` method is called which fulfills different purposes by calling various methods:

- `set_shape` : defines the current position ϕ_0 based on the mesh coordinates at the beginning of the time step
- `set_local_frame` : computes the local basis vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{n}$ of the shell surface. Note that boundary conditions can be apply on the boundary to enforce the normal to lying in some specific symmetry plane (`boundary_conditions_n0` method)
- `boundary_conditions` : defines boundary conditions for the velocity and director fields
- `set_director` : computes the initial director angles $\boldsymbol{\beta}_0$ based on the initial normal, the initial director is initialized to the initial normal.
- `set_kinematics_and_fundamental_forms` : defines the current fundamental forms $\boldsymbol{a}_0$, $\boldsymbol{b}_0$, curvature H
- `set_kinematics` : defines the new unknown director $\boldsymbol{d}$ from $\boldsymbol{d}_0$ and the unknown angle rates $\dot{\boldsymbol{\beta}}$
- `set_energies` : defines the various constitutive relationships of the active viscous shell model and the corresponding dissipation potentials
- `set_total_energy` : aggregates all contributions into the global system lagragian, including volume constraints. Automatic differentiation is used to obtain the corresponding system jacobian

The `solve` method then solves the corresponding system for the new velocity and director angles rate.

The `evolution` method then advances in time by: * solving the thickness evolution equation (`set_thickness` method) * updating the mesh position with the mesh displacement $\boldsymbol{U}\Delta t$ (using FEniCS `ALE.move` method)

Finally, during the time stepping loop, mesh refinement is performed every `remeshing_frequency` time steps.

References

Hale, J. S., Brunetti, M., Bordas, S. P., & Maurini, C. (2018). Simple and extensible plate and shell finite element models through automatic code generation tools. Computers & Structures, 209, 163-181.

Campello, E. M. B., Pimenta, P. M., & Wriggers, P. (2003). A triangular finite shell element based on a fully nonlinear shell formulation. Computational mechanics, 31(6), 505-518.

Bleyer, J. (2021). A novel upper bound finite-element for the limit analysis of plates and shells. European Journal of Mechanics-A/Solids, 90, 104378.