Summer report

September 14, 2017

I investigated how to simulate soft elastic object effectively using only coarse meshes. We will break this investigation into 2 parts in this report

1. How to quickly fit the material property of an object to a FEM simulation through dynamics using DAC ([**?**] Section 3).
2. Investigate FEM locking/numerical stiffening on coarse mesh. In particular, I looked at the cause and how it affects the simulation in static and dynamic.

1 Fitting Young's modulus using DAC

The material properties of elastic objects specified by the manufacturer are often inaccurate (due to variation in fabrication process, I assume), and calibration is needed for each object. Fitting material properties through tracking is not an easy task. but DAC is a simple algorithm and work surprisingly well in the cases considered in [**?**]. DAC consists of the following steps:

1. Track the oscillation of a simple deformation mode of the object
2. Solve the harmonic inverse problem [**?**] from the trajectory in 1. and extract the dominating frequency (frequency with largest amplitude), ω_l
3. Create a mesh for the FEM simulation, with initial Young's modulus $Y_0 = 1Pa$. Other material parameters (Poisson ratio, density, etc) are not fitted here, so will be set to some physical values and stay unchanged after DAC. For example Poisson ratio can be set to $P = 0.48$, and the density ρ is set to the value specified by the manufacturer. Also the material model can is arbitrary (linear, Neo-hookean etc)
4. Find the lowest the eigenvalue problem

   $$Kx = \lambda_l Mx,$$

   where K and M are the stiffness matrix and mass matrix from the FEM simulation[1] 3., with the same boundary conditions from the simple deformation in 1..

[1] If the model chosen in 3. is nonlinear, K must be chosen at a equilibrium state to ensure positive definiteness.

5. Rescale the Young's modulus by 2. and 4.

$$Y = Y_0 \frac{\omega_l^2}{\lambda_l}$$

This fitting is based on the fact that the eigenvalues of the system scale linearly with Young's modulus. After DAC, the Young's modulus can be used for simulating the same object *with the same mesh*. The next section of this writeup investigate the mesh-dependent stiffness, which prevent us from using DAC on the same object with different mesh resolutions.

2 Mesh-dependent stiffness

Following the fact that Young's modulus scales linearly with the eigenvalue, we observed that different mesh resolution of same object would have different eigenvalues[2], meaning the apparent stiffness of the object is different, and the modes would oscillate at different frequencies.

2.1 A 2D Example

First start from an polygon shape object (from Distmesh [**?**]) with 2 level of refinements (See Fig 1, 2, 3). Solve the general eigenvalue problem with Neumann B.C. for both mesh

$$Kx = \lambda Mx$$

and eliminate the null space corresponding to the rigid motion, we plot the lowest eigenmode in Fig 4 and Fig 5. These eigenmodes look similar on the object, but the eigenvalues are very different, $\lambda = 2.35$ for the low resolution mesh, and $\lambda = 1.99$ for the high resolution mesh, meaning simulating on the low resolution mesh would make the object look stiffer and oscillate faster.

2.2 Theoretical background

This discrepancy comes from the discretization error from FEM, so it is safe to assume that the material will look softer and softer as we refine the mesh. This is indeed the case, as indicated in [**?**] (lemma 3), where they showed that the approximate the general eigenvalue problem of an elliptical B.V.P. with Ritz approximation/Galerkin method will always lead to larger eigenvalues. Intuitively, this comes from the fact that we are searching in a less inclusive function space as we coarsen the mesh in the weak form of the elliptical B.V.P.

[2] As mesh gets finer and finer, the number of eigenvalue increases, and the new eigenvalues corresponds to the high oscillatory modes which damp out quickly. Here we are talking about lower eigenvalues, which correspond to the dominating dynamic modes with larger amplitudes, and should ideally stay the same across different mesh resolutions.

Figure 1: Polygon from Distmesh [**?**]

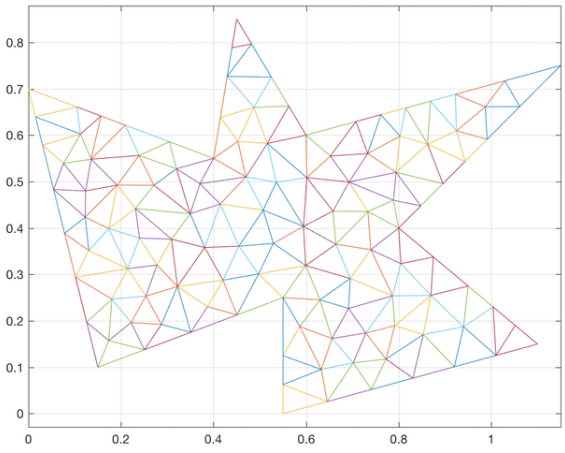

Figure 2: Coarse mesh

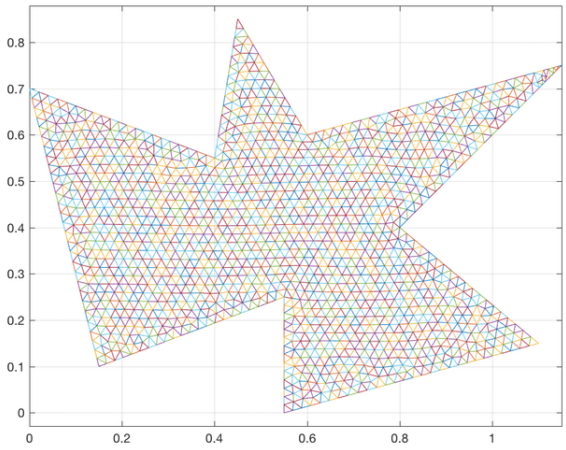

Figure 3: Fine mesh

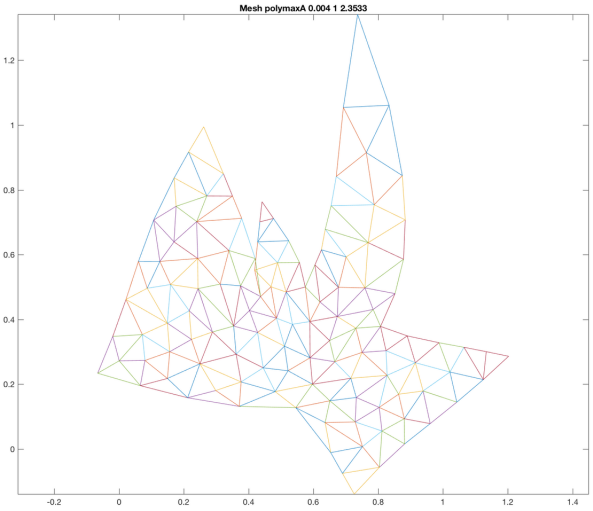

Figure 4: Coarse mesh first mode with eigenvalue = 2.3533

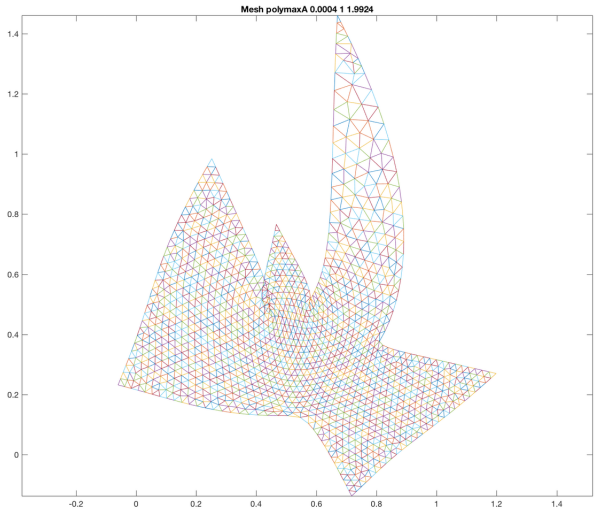

Figure 5: Fine mesh first mode with eigenvalue = 1.99

2.3 A 1D example

In this subsection we discuss a 1D example in more detail and demonstrate DAC and it's drawback.

2.3.1 Continuous case

Consider the 1D wave equation with Dirichlet BC over an interval $x \in [0, L]$

$$\begin{aligned} u_{tt} &= c^2 \nabla^2 u \\ u(t, x = 0) &= u(t, x = L) = 0 \end{aligned} \tag{1}$$

Look for the solution of the form

$$u(t, x) = e^{i\sqrt{\lambda}t} U(x)$$

is equivalent to solve the eigenvalue problem for the operator

$$\mathbb{L}U = -c^2 \nabla^2 U = \lambda U \tag{2}$$

The analytical solution for the eigenvalues:

$$\lambda_i = \frac{(ic\pi)^2}{L^2} \ , i \in \mathbb{N} \tag{3}$$

For example, with $L = 1$, $i = 1$ (the lowest eigenvalue) we have $\lambda_1 = \pi^2 c^2 \approx 9.86c^2$, and $\sqrt{\lambda_1}$ is the frequency (in time) of the dominating dynamic mode Semi-discretization

It is common to do semi-discretization on (1) in space, and step the resulting ODE in time to simulate the dynamics. This semi-discretization in space is highly related to the eigenvalue problem (2). In a sense, (2) split the spatial variable from the time variable in (1), just as how the semi-discretization split the spatial variable from the time variable. However, it is not clear

1. how the discretization error from semi-discretizing in space can pollute the overall time trajectory (dynamic behavior),
2. and how does this error affect the ODE time stepping error

Here we will mainly focus on enum(1.).

2.3.2 Finite difference

The operator in (2) is the negative Laplacian scaled. For finite difference in 1D with Dirichlet BCs the Laplacian matrix and eigenvalues are well-known. That is, using central differences on a uiform grid the FD matrix for (2)

$$\begin{aligned} A_h\mathbf{u} &= \lambda_h\mathbf{u} \\ A_h &= \frac{-L^2c^2}{h^2}\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \end{aligned}$$

where h is the interval size. The eigenvalues of this matrix is also well-known

$$\lambda_{h,i} = \frac{2c^2L^2}{h^2}\left(1-\cos(\frac{ih\pi}{L})\right),\ h\in\mathbb{Z}^+,\ i\in[1,\frac{L}{h}] \tag{4}$$

Notice that the eigenvalue from (4) approach the eigenvalues (3) from below as $h\to 0$. At first this seems contradictory to what we mentioned in 2.2 and the lemma in [?]. However, this is not the case, since finite difference discretization is not based on the weak form.

2.3.3 Finite element

Another way to solve the eigenvalue problem (2) is using finite element. First convert (2) to the weak form

$$-c^2\int V\nabla^2U dx \quad = c^2\int \nabla U\nabla V dx \quad -c^2\int \nabla\cdot(V\nabla U)dx$$

and divergence theorem says the second term on the RHS is 0 with Dirichlet BCs, so (2) is

$$c^2\int \nabla U\nabla V dx = \lambda\int VU dx \tag{5}$$

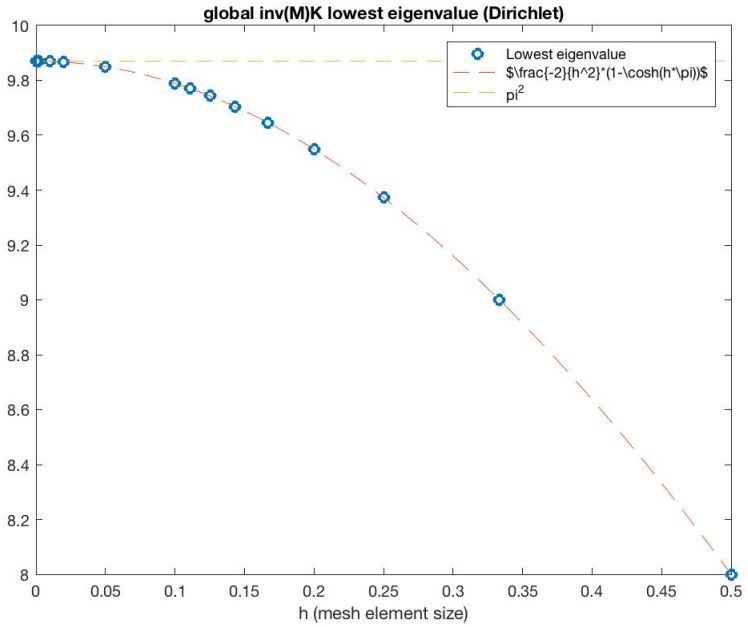
global inv(M)K lowest eigenvalue (Dirichlet)
Lowest eigenvalue
$\frac{-2}{h^2}*(1-\cosh(h*\pi))$
pi^2
10
9.8
9.6
9.4
9.2
9
8.8
8.6
8.4
8.2
8
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
h (mesh element size)

Figure 6:

In Galerkin method we choose the same linear hat function as the function space for U and V. The resulting element stiffness matrix is

$$K_{e,h} = \frac{Lc^2}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

and element mass matrix from the RHS

$$M_{e,h} = \frac{\lambda h}{6L} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

After assembly and eliminate the boundary variables from Dirichlet BCs, we have the system

$$\begin{aligned} K_h \mathbf{u} &= \lambda M_h \mathbf{u} \qquad (6) \\ K_h &= \frac{-Lc^2}{h} \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \\ M_h &= \frac{h}{6L} \begin{bmatrix} 4 & 2 & & & \\ 2 & 4 & 2 & & \\ & \ddots & \ddots & \ddots & \\ & & 2 & 4 & 2 \\ & & & 2 & 4 \end{bmatrix} \end{aligned}$$

The analytical solution to the generalized eigenvalue problem (6) is not known yet, but it seems like it is following

$$\lambda_{h,i} = -\frac{2c^2L^2}{h^2} \left(1 - \cosh(\frac{ih\pi}{L})\right), \ h \in \mathbb{Z}^+, \ i \in [1, \frac{L}{h}]. \qquad (7)$$

Notice that (7) approach (3) from above as $h \to 0$.

2.3.4 Story so far

The simple analysis above shows that, the eigenvalues to (3), which represents the dynamic property of the solution, is affected by the mesh resolution used in the semi-discretization. The effect is not huge numerically ($\mathcal{O}(h^2)$), but the resulting dynamic simulation may look very different. This is a simplest aspect of enum(11.). A weird/interesting discovery on the path was that changing from FD to FE reversed the h-dependence of the eigenvalues. (Figure 1 is like a mirrored inverse of Figure 2, observed by Danny)

2.3.5 DAC fix

DAC provides a simple to match the dynamic property of the original continuous equation (1). That is, we can scale the eigenvalue corresponding to the

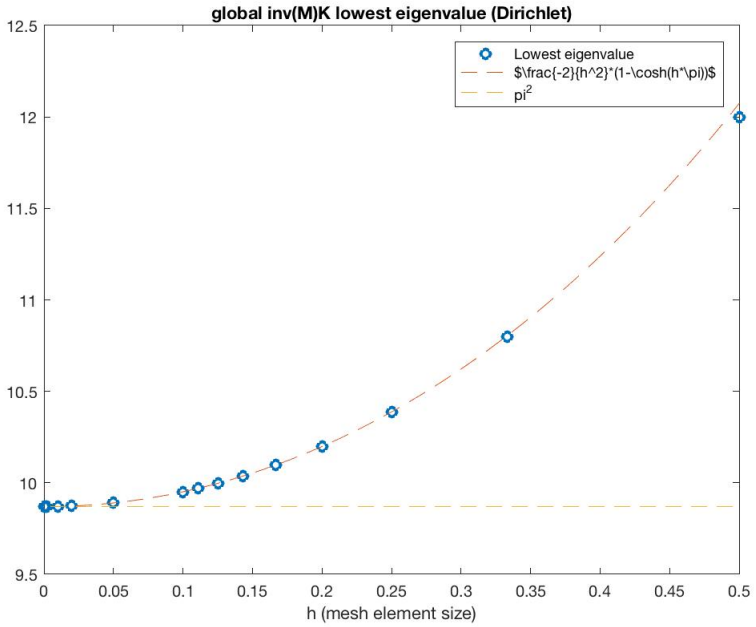
global inv(M)K lowest eigenvalue (Dirichlet)
Lowest eigenvalue
$\frac{-2}{h^2}*(1-\cosh(h*\pi))$
pi^2
12.5
12
11.5
11
10.5
10
9.5
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
h (mesh element size)

Figure 7:

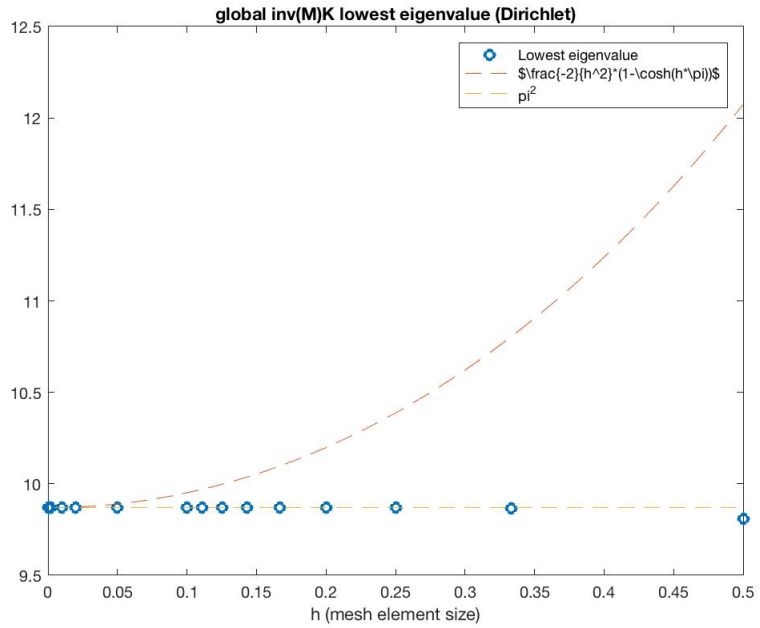

Figure 8:

dominating mode (the lowest eigenvalue). If we do that, we can rescale the stiffness matrix by $\frac{\lambda_1}{\lambda_{h,1}}$. The resulting plot for the lowest eigenvalue is in Figure 3. The difference between the method in [**?**] and what I used here is that, the rescaling factor is derived from an analytical form of the lowest eigenvalue (the dominating mode), which is not always avaible. To handle more complicated objects, [**?**] used a vision-based technique to extract the dominating mode of motion, as described in Section 1.

2.3.6 No free lunch

However, there's no free lunch, and this simple scaling won't fix other eigenvalues. For example, the second eigenvalue will be scaled to another number, but not its continuous counterpart, as shown in Figure 4.

2.4 2D static example

Here is another demonstration in the static setting

In Figure 10 and 11, the yellow triangle is the rest shape. In Figure 10 we deformed the top node and have the lower 2 fixed, changing it into the blue triangle. The red arrows are the force required to hold the blue triangle in this deformed shape. If we refine the triangle, as shown in Figure 11, and perform

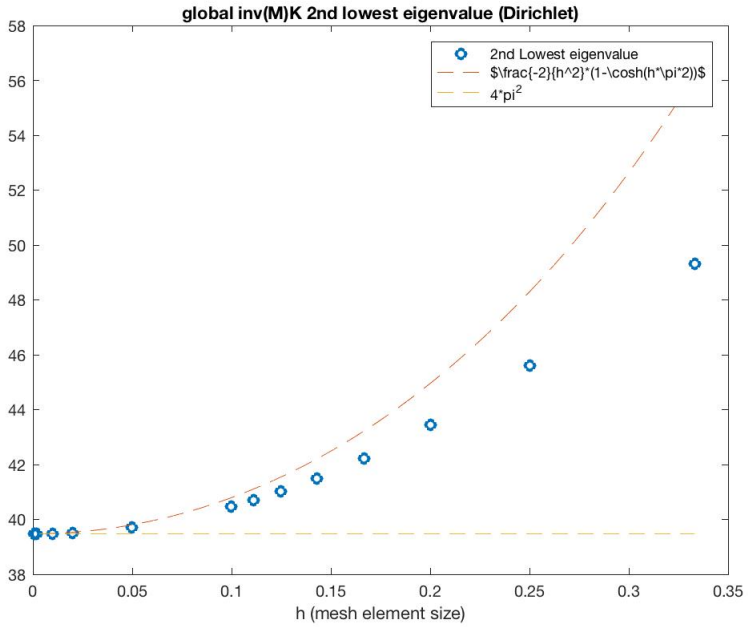
global inv(M)K 2nd lowest eigenvalue (Dirichlet)
2nd Lowest eigenvalue
$\frac{-2}{h^2}*(1-\cosh(h*\pi*2))$
4*pi^2
58
56
54
52
50
48
46
44
42
40
38
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
h (mesh element size)

Figure 9:

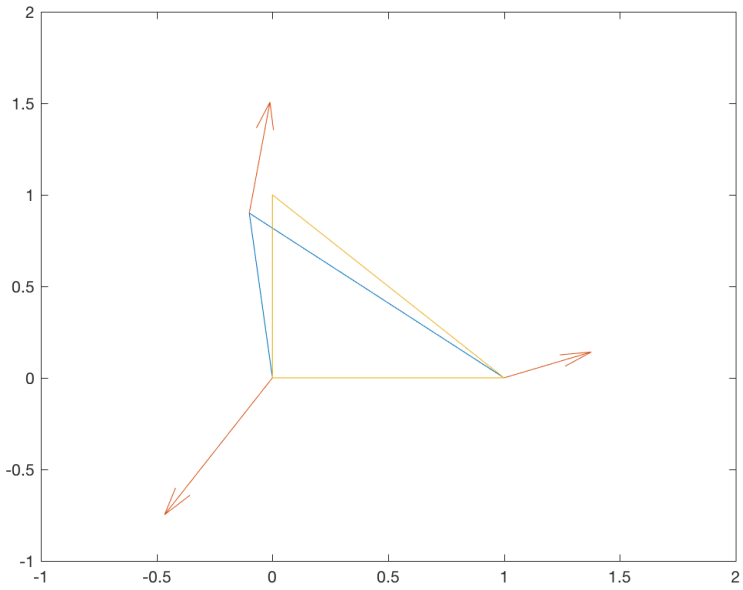

Figure 10: Force required to move a single triangle element

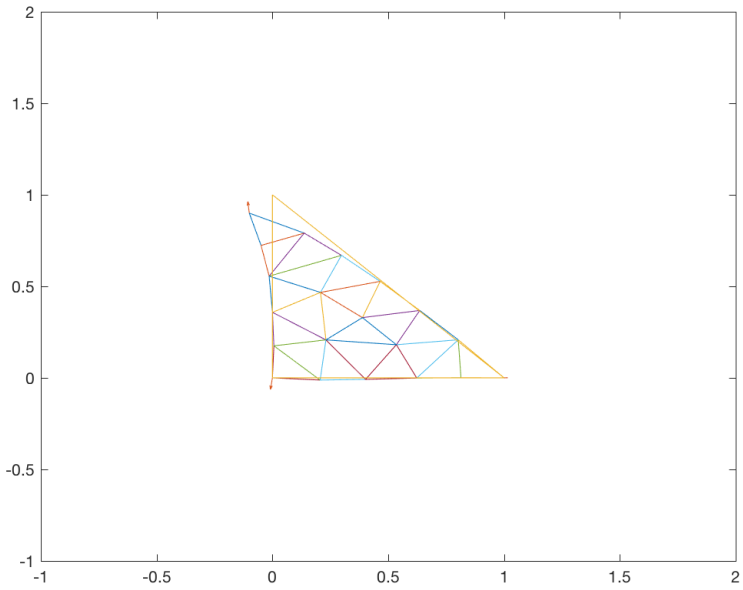

Figure 11: Force required to achieve the same deformation on a refined mesh

the same operation, that is, move the top node while keeping the bottom two corners fixed (all other nodes are free), the force required at each nodes to keep the deformed triangle is much less.

2.5 Locking/Numerical Stiffening

The mesh-dependent stiffness is caused by the fact the function space spanned by the bases functions is not rich enough, which is also the same cause for the locking effect discussed in engineering literature: when the Poisson ratio is high, it takes force up to an order of magnitude larger to bend linear elements to the same amount of deformation on a coarse mesh. Thus it is worth trying different ways people mitigate locking. However, we should keep in mind that we have a different application:

- In engineering, the force magnitude is usually much larger, and the focus is usually the displacement
- For dynamic simulation in graphics, we also care about oscillation frequency, since collision is highly sensitive to the trajectory.

2.6 Discontinuous Galerkin

Discontinuous Galerkin has been used to solve this problem in [**?**], so the next step is to study this paper.