The Peridynamics problem

We look at the Peridynamics problem

$$\ddot{u} = \int_{H_x} k(u' - u, x' - x)dx' + b(x) \tag{1}$$

where H_x is a ball of radius δ centered at x. We can show that the kernels

$$k(u' - u, x' - x) = \frac{2(u' - u)}{\delta^2 |x' - x|}, \qquad k(u' - u, x' - x) = \frac{3(u' - u)}{\delta^3 \Pi |x' - x|}, \tag{2}$$

converge to the Laplacian operator in 1-D and 2-D respectively (as $\delta \to 0$). This is not the only possible choice for the PD kernel, but it is a good starting point.

The Finite Element discretization of the PD equation is

$$\int_\Omega \ddot{u}\Phi_k(x)dx = \frac{1}{2}\int_\Omega \int_{H_x} k(u' - u, x' - x)\left(\Phi_k(x') - \Phi_k(x)\right)dx'dx + \int_\Omega \Phi_k(x)b(x), \tag{3}$$

where $\Phi_k(x)$ are the Finite Element test functions. The solution $u(t, x)$ is approximated by the basis functions

$$u(t, x) = \sum_i c_i(t)\Phi_i(x). \tag{4}$$

Boundary conditions

If Dirichlet boundary conditions are specified, we extend the domain by a region of size δ and we add boundary elements. The solution is known in the boundary elements, hence the test functions and $\Phi_k(x)$ may be only a subset of all the basis functions $\Phi_i(x)$. If we have Neumann boundary conditions (as in the classical elasticity case), then we specify those as a force acting on a layer of size δ inside the domain. That is $b(t) = \frac{D}{\delta}$ on a layer of depth δ, we can show that this will converge to the classical elasticity PDE with Neumann condition $u|_{\partial\Omega} = D$.

The discrete system

For a linear kernel, the final discretization of the PD problem becomes

$$M\ddot{u}(t) = Au(t) + B(t) + G(t), \tag{5}$$

where M is the classical mass matrix, A is the discretization of the PD operator, $B(t)$ is the external forcing term (and Neumann boundary conditions) and $G(t)$ is the Dirichlet boundary term. $u(t)$ is the vector of the unknown variables, this is a bit ambiguous notation, but the code has only discrete $u(t)$ hence there was no need to distinguish between the continuous solution $u(t, x)$ and $u(t) \in R^N$.

For non-linear kernels, we have the discretization

$$M\ddot{u}(t) = F(u(t)) + B(t) + G(t), \tag{6}$$

where $F(\cdot)$ is the non-linear discretization of the PD operator.

Time Integration

We have a second order time dependent problem. We use central difference to discretize the derivative on the left and we get the second order two-step explicit scheme:

$$u^{n+1} = \Delta t^2 M^{-1} \left(A u^n + B + G \right) + 2u^n - u^{n-1}. \tag{7}$$

In general the scheme is stable so long as $|\Delta t^2 \lambda_{max}| < 1$, where λ_{max} is the largest (in absolute value) eigenvalue of A. I think the eigenvalues of the PD kernels above were of the order $O(\frac{1}{\delta \Delta x})$, but I may be wrong.

The non-linear system simply replaces the matrix vector product Au^n with $F(u^n)$ and the stability condition depends on the eigenvalues of the linearization of $F(\cdot)$.

Steady State

The steady state problem is

$$Au = -B - G. \tag{8}$$

Note that the solution only exists if we specify Dirichlet boundary conditions somewhere in the domain.

General Quadrature Consideration

Consider the integral

$$\frac{1}{2} \int_{\Omega} \int_{H_x} k(u' - u, x' - x) \left(\Phi_k(x') - \Phi_k(x) \right) dx' dx. \tag{9}$$

We approximate the integral with two quadratures, outer and inner. The outer quadrature is defined on an elements E_j contained in the support of $\Phi_k(x)$. Thus the outer integral is approximated as

$$\frac{1}{2} \sum_j \int_{E_j} \int_{H_x} k(u' - u, x' - x) \left(\Phi_k(x') - \Phi_k(x) \right) dx' dx \tag{10}$$

$$= \frac{1}{2} \sum_j \sum_m w_m^{o,j} \int_{H_{x_m^{o,j}}} k(u' - u(x_m^{o,j}), x' - x_m^{o,j}) \left(\Phi_k(x') - \Phi_k(x_m^{o,j}) \right) dx', \tag{11}$$

where $w_m^{0,j}$ are the m weights associated with the outer quadrature over element E_j and $x_m^{o,j}$ are the corresponding points. For every point $x_m^{o,j}$ we need an inner quadrature defined over $H_{x_m^{o,j}}$ that evaluates the inner integral

$$\frac{1}{2} \sum_j \sum_m w_m^{o,j} \sum_n w_n^{i,j,m} k(u'(x_n^{i,j,m}) - u(x_m^{o,j}), x_n^{i,j,m} - x_m^{o,j}) \left(\Phi_k(x_n^{i,j,m}) - \Phi_k(x_m^{o,j}) \right), \tag{12}$$

where the weights $w_n^{i,j,m}$ are associated with $H_{x_m^{o,j}}$ and $x_n^{i,j,m}$ are the corresponding points. For simplicity we will only consider the quadrature over one element, the outer most sum will simply add the integrals over all elements where $\Phi_k(x)$ has support.

The proper choice of the quadrature rule depends on the type of basis functions that we use.

Discontinuous Galerkin Discretization

Suppose we use piece wise linear discontinuous function basis $\Phi_i(x)$. Then suppose that we select the outer quadrature points so that they are entirely contained inside E_j. Then for sufficiently small δ, $H_{x_n^{i,j,m}} \subset E_j$ and hence the discrete scheme will not include any interaction between the solution over E_j and the solution

over the adjacent elements. Thus the discrete scheme will be unstable. To avoid this issue, we should select the outer quadrature points so that at least some of them lie on the boundary of E_j.

In the 1-D case, when elements are simple intervals, we can select a quadrature that includes the two end points of the interval.

In the 2-D case, when elements are triangles, we can select the quadrature points on the mid points of the three edges.

One possible issues is that depending on the kernel, if the solution is discontinuous across the elements, then the resulting inner integral may be divergent. To avoid such behavior, we can introduce a small perturbation to the original kernel

$$k(u'-u, x'-x) = \frac{2(u'-u)}{\delta^2|x'-x|}, \quad \rightarrow \quad k_\epsilon(u'-u, x'-x) = \frac{2(u'-u)}{\delta^2|x'-x|^{1-\epsilon}}. \tag{13}$$

The result would be a very large integral, applying very large force at the point of the discontinuity trying to force the solution to become continuous. Note that in the peridynamics model, when we wish to model fractures, we need a special criteria for deciding when and how to break the bonds of the "particles". If we break the bond between two adjacent elements, then the kernel would be zero and the integral will converge.

Continuous Galerkin Discretization

Suppose we use piece wise linear continuous function basis $\Phi_i(x)$. The support of the test functions $\Phi_k(x)$ spans more then one element and hence we don't have to worry about disconnected elements. The main problem with continuous elements is that the internal integral

$$I(x) = \int_{H_x} k(u'-u, x'-x) \left(\Phi_k(x') - \Phi_k(x)\right) dx', \tag{14}$$

can have a drop in value near the boundary of the element. That is, the integrand can be constant except for region of size δ around the boundary. In such case, any quadrature that uses points at the boundary may cause instability. If we take the outer quadrature points to be entirely inside E_j, then we can show that when $\delta \rightarrow 0$, the PD discretization converges to the piece wise continuous linear discretization of the classical elasticity PDE. Gaussian quadratures are a good choice for outer quadrature in the case of continuous piece wise linear basis.

Mixed Discretization

For the most part of the domain, it is best to use continuous piece wise linear functions. Then we can take the size of the elements to be much larger than δ and the solution will behave close to the classical elasticity PDE. In the region where the solution has more erratic behavior, we need to refine the mesh and use elements no bigger than δ. Then we can switch to using discontinuous basis functions for this specific region only (with an appropriate outer quadrature).

Inner Quadrature

In the 1-D case, the domain of integration for the inner integral is an interval that intersects one or more elements E_j. One straight forward approach would be to integrate elements by elements over all of E_j (assuming that the kernel extends with zero beyond $H_{x_m^{o,j}}$). However, the kernel may have a jump discontinuity at the boundary of $H_{x_m^{o,j}}$ or the quadrature may simply fail to accurately integrate a function with support in only a small portion of the domain. Thus the safe way to do the inner integral is to take it elements by element, but integrate only over $E_j \cap H_{x_m^{o,j}}$.

In the 2-D case, we still need to consider regions of the form $E_j \cap H_{x_m^{o,j}}$, however, those can now have very irregular shape. Basically we need to find a way to integrate functions over all possible intersections of a disk with a triangle. In the 2-D code, this is done with the function "GeometryDynamicQuadrature-TriangleDiskIntersection". The algorithm first considers the intersections of the circle with the edges of the element. Each edge can be intersected zero, once or twice. By sorting the edges by number of intersections we have several general shapes ("210" means one edge is intersected twice, one edge is intersected once and one edge is not intersected). The resulting shape is then divided into a set of triangles and arcs. For the triangles, we can use simple Gaussian quadratures. For the arcs, we use a recursive rule that approximates each arc by a triangle and two arcs until the central angle associated with the arch drops below certain tolerance.

Try drawing a triangle and disk with 220 (the circle should intersect two of the sides of the triangle two times each), then you will see that regardless of how you orient the things, the resulting shape can always be divided into two triangles and two arcs. You will similar results for 200 (one arc) or 110. Sometimes you have to consider the vertexes of the elements as well, the shape may depend on whether or not one or two of the vertexes fall within the disk.

There are two special cases. When the quadrature point is inside the element E_j and δ is sufficiently small, $H_{x_m^{o,j}} \subset E_j$, then we simply do the integral using a disk quadrature.

Another special case (not included in the code), comes when we use a quadrature point on the edge of the element, then the resulting domain of integration can be a half-disk. Of course, this happens only when δ is sufficiently small so that $E_j \cap H_{x_m^{o,j}}$ is non-empty only for the two adjacent elements. Special quadrature can be created for such case. I have some code, but it is burried within an old implementation that doesn't work anymore, I can provide the quadrature code if needed.

Convergence Results

For a fixed mesh using piece wise continuous elements and if we let $\delta \to 0$, we can show that the steady state result converges to the discrete solution to the classical elasticity.

A problem occurs if we fix δ and we let the mesh size to go to zero. Then the solution does not converge. This can be caused from some problem in the analysis (we are not sure even if it should converge), problem with the discretization (something specific about the quadrature or basis functions) or a bug in the code.