Introduction to Gridap.jl

Wenjie Yao

November 20, 2020

Massachusetts Institute of Technology

Introduction to FEM

Numerical Methods for Solving PDEs

There are many numerical methods used to solve a partial differential equation (PDE):

$$\mathcal{L}u(x) = f(x)$$

where $\mathcal{L}$ is a partial differential operator and f acts as a source term

- Finite difference method (FDM): very easy to implement
- Finite element method (FEM): can handle complicated geometries
- Boundary element method (BEM): only involves boundary (surface) values
- Finite volume method (FVM): “conservative”, often used in fluid dynamics
- ...

Deriving the Weak Form for FEM

- The key part in FEM is to derive the weak form of the original problem
- Poisson Equation

$$\begin{aligned} -\Delta u &= f \text{ in } \Omega \\ u &= g \text{ on } \Gamma \end{aligned}$$

- Galerkin's method: if u is a solution to the equations above, then for any function $v \in V$ that is continuous in Ω and zero at boundary Γ:

$$-\int_\Omega v \Delta u \mathrm{d}\Omega = \int_\Omega v f \mathrm{d}\Omega$$

- Integral by part:

$$\int_\Omega \nabla v \nabla u \mathrm{d}\Omega - \int_\Gamma v \frac{\partial u}{\partial n} \mathrm{d}\Gamma = \int_\Omega v f \mathrm{d}\Omega$$

Deriving the Weak Form for FEM

- Since $v = 0$ at Γ:

$$\int_\Omega \nabla v \nabla u \mathrm{d}\Omega = \int_\Omega v f \mathrm{d}\Omega$$

- We denote $a(u, v) = \int_\Omega \nabla v \nabla u \mathrm{d}\Omega$ as the bilinear term and $b(v) = \int_\Omega v f \mathrm{d}\Omega$ as the linear term.

Weak form

Find $u \in U$ (trial space with given boundary condition) such that for any $v \in V$ (test space with zero Dirichlet boundary condition):

$$a(u, v) = b(v)$$

Conformity of FE Function Spaces

The choices of the finite element basis function should be restricted

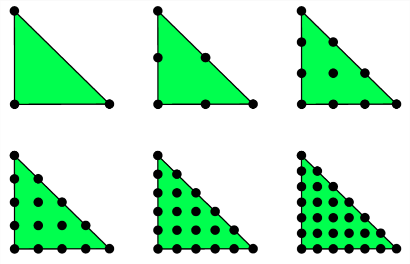
- Continuous space (C0): f continuous
- Hilbert space (L2): $\int_{-\infty}^{\infty} f(x)^2 dx < \infty$
- Sobolev space (Hm): $f \in L2, \frac{\partial^2 f}{\partial x^n} \in L2, n \leq m$
- H-div space (Hdiv): $f \in L2, \nabla \cdot f \in L2$
- H-curl space (Hcurl): $f \in L2, \nabla \times f \in L2$

For example, the FE function space should be $H1$ for the Poisson equation.

Typical FE Basis Function Types

ReferenceFE	Function Space	Conforming?
Lagrange	H1	Yes
Discontinuous Lagrange	L2	Yes
Nedelec	Hcurl	Yes
Raviart–Thomas	Hdiv	Yes
Hermit	H2	No
Crouzeix–Raviart	H1	No

Lagrange (polynomials) of order 1-6 for triangular mesh



From Weak Form to Matrix Form

- From the weak form

$$a(u, v) = b(v)$$

- Suppose we have the FE basis function

$$u = \sum_j u_j \hat{u}_j, v = \sum_i v_i \hat{v}_i$$

- Substitute into the weak form

$$a(\sum_j u_j \hat{u}_j, \sum_i v_i \hat{v}_i) = b(\sum_i v_i \hat{v}_i)$$

- Since a and b are bilinear and linear:

$$\sum_i v_i \left[\sum_j u_j a(\hat{u}_j, \hat{v}_i) - b(\hat{v}_i) \right] = 0$$

From Weak Form to Matrix Form

- For the equation above to hold for any value of v, we then have the matrix form

$$\mathbf{Au} = \mathbf{b}$$

  where $A_{ij} = a(\hat{u}_j, \hat{v}_i)$ and $b_i = b(\hat{v}_i)$
- Note that this $\mathbf{u}$ solved here is only a vector of coefficients, it might not have any physical meaning, the actual field should be $u(x) = \sum_i u_i \hat{u}_i$.
- In the case of first-order Lagrange, $\mathbf{u}$ denotes the nodal field values.
- Recall that $a(u, v)$ and $b(v)$ are some integral over the domain Ω, those integrals can then be computed approximately via Gauss quadrature for each element and then sum-up.

Introduction to Gridap.jl

About Gridap.jl

- Gridap provides a set of tools for the grid-based approximation of PDEs written in the Julia programming language.
- `https://github.com/gridap/Gridap.jl`
- Install Gridap by typing `] add Gridap` in Julia REPL.
- Advantage: All written in Julia, open source, free
- Shortage: Under development, missing features such as periodic boundary support

Using Gridap.jl–Discrete Model

- Create model from a mesh file: (`.msh` file from GMSH or other mesh files)

Code

$$model = DiscreteModelFromFile("\,file\_name")$$

- Create model using built-in functions: (currently only support a Cartesian rectangular geometry)

Code

$$model = CartesianDiscreteModel(domain, cells)$$

Using Gridap.jl–FE Spaces

- In Gridap, the test FE function space is defined by

Code

$$
\begin{aligned}
V = \mathit{TestFESpace}(\quad & \mathit{reffe} && =: \mathit{Lagrangian}, \\
& \mathit{order} && = 1, \\
& \mathit{valuetype} && = \mathit{Float64}, \\
& \mathit{model} && = \mathit{model}, \\
& \mathit{conformity} && =: \mathit{H1}, \\
& \mathit{dirichlet\_tags} && = \mathit{boudnary\_tags})
\end{aligned}
$$

- The trial function space is then

Code

$$
U = \mathit{TrialFESpace}(V, \mathit{dirichlet\_values})
$$

Using Gridap.jl–Numerical Integration

- Generate triangulation and quadrature from model (for the whole domain Ω)

Code

$$
\begin{aligned}
trian &= Triangulation(model) \\
degree &= 2 \\
quad &= CellQuadrature(trian, degree)
\end{aligned}
$$

- If a boundary integral is needed (e.g. Neumann boundary conditions):

Code

$$
\begin{aligned}
btrian &= BoundaryTriangulation(model, boundary\_tags) \\
bquad &= CellQuadrature(btrian, degree)
\end{aligned}
$$

Using Gridap.jl–Weak Form

- In Gridap, the weak form can be written in a very nice and neat way with symbolic formulas
- Weak form for the Poisson equation:

Code

$$
\begin{aligned}
f(x) &= 1.0 \\
a(u, v) &= \nabla(v) \odot \nabla(u) \\
b\_\Omega(v) &= v * f \\
t\_\Omega &= AffineFETerm(a, b\_\Omega, trian, quad)
\end{aligned}
$$

Using Gridap.jl–Assemling

- Assemble the Gridap operator and solve

Code

$$
\begin{aligned}
op &= AffineFEOperator(U, V, t\_\Omega) \\
A &= get\_matrix(op) \\
b &= get\_vector(op) \\
uvec &= A \backslash b
\end{aligned}
$$

- For complex numbers, one needs to define

Code

$$
\begin{aligned}
op \quad = AffineFEOperator(\quad & SparseMatrixCSC\{ComplexF64, Int\}, \\
& Vector\{ComplexF64\}, \\
& U, V, t\_\Omega)
\end{aligned}
$$

Using Gridap.jl–Analysis

- After you obtain the field vector **u**, you can generate $u(x)$ with the expansion on the FE function space U:

Code

$$uh = FEFunction(U, uvec)$$

- You can view the fields by writing it to a `.vtk` file and view it via ParaView:

Code

$$writevtk(trian, "results", cellfields = ["uh" => uh])$$

Using Gridap.jl

To sum up, the key steps using Gridap.jl are

1. Create model (`model`) from mesh file or built-in function;
2. Define test FE space (`V`) and trial space (`U`);
3. Generate interpolation spaces (`trian`) and cell quadrature (`quad`) used for integration;
4. Define weak form: bilinear term `a(u,v)` and linear term `b(v)`;
5. Assemble the matrix `A` and vector `b`, then solve `u=A\b`;
6. Result analysis.

Example 1–Helmholtz Equation

Helmholtz Equation

- Consider a 2D scalar Helmholtz equation (TM-polarized wave equation for solving H_z):

$$\left[-\nabla \cdot \frac{1}{\varepsilon} \cdot \nabla - k^2 \mu\right] u = f$$

  where $k = 2\pi/\lambda$ is the wave number and $f = \frac{\partial}{\partial x}\left(\frac{1}{\varepsilon} J_y\right) - \frac{\partial}{\partial y}\left(\frac{1}{\varepsilon} J_x\right)$ is a source term.

- For simplicity, consider vacuum with $\varepsilon = \mu = 1$, the weak form is then

$$\begin{aligned} a(u, v) &= \int_\Omega (\nabla v \cdot \nabla u - k^2 v u) \mathrm{d}\Omega \\ b(v) &= \int_\Omega v f \mathrm{d}\Omega \end{aligned}$$

Computational Cell

- Consider a square domain $L \times L$ and a point source at the center $f(x) = \delta(x)$
- Perfectly matched layers (PMLs) at the sides with additional thickness d_{pml}

$$\frac{\partial}{\partial x} \to \frac{1}{1 - \mathrm{i}\frac{\sigma_x(x)}{\omega}} \frac{\partial}{\partial x} = \frac{1}{s_x} \frac{\partial}{\partial x}$$

  or

$$\nabla \to \Lambda \cdot \nabla$$

  where

$$\Lambda = \begin{bmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & \frac{1}{s_z} \end{bmatrix}$$

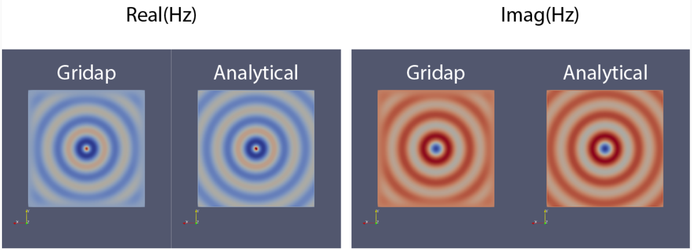
- PML is not actually a boundary condition (you still need 0 Dirichlet boundary condition at the end of PML), it acts like a absorbing material

Compare to Analytical Result

- The analytical expression for a magnetic dipole in 2D is

$$H(r) = -\frac{\mathrm{i}}{4}\mathrm{Hankel}^{(2)}(0, kr)$$

- The Gridap results show an excellent match except at diverging point (center) and PML



Adjoint Method in Gridap

Adjoint Method for Optimization Problem

- Optimization Problem:

$$\begin{aligned} \text{maximize} \quad & g(\mathbf{u}) \\ \text{constraint} \quad & \mathbf{A}(\mathbf{p})\mathbf{u} = \mathbf{b} \end{aligned}$$

  where **p** is some design parameters.

- Adjoint Method: the derivative to design parameters are

$$\frac{\mathrm{d}g}{\mathrm{d}\mathbf{p}} = -\lambda^T \frac{\partial A}{\partial \mathbf{p}} \mathbf{u}$$

  with the adjoint equation:

$$A^T \lambda = \left(\frac{\partial g}{\partial \mathbf{u}}\right)^T$$

Adjoint Method in Gridap

- Suppose we have a weak form

$$a(p, u, v) = \int_\Omega [\xi(p) a_1(u, v) + a_2(u, v)] \, \mathrm{d}\Omega$$

with $p(x) = \sum_k p_k \hat{p}_k(x)$ and $\hat{p}(x) \in P$

- The matrix $\frac{\partial A}{\partial \mathbf{p}}$ is then

$$\begin{aligned}
\left(\frac{\partial A}{\partial p_k}\right)_{ij} &= \int_\Omega \frac{\partial \xi(p)}{\partial p_k} a_1(\hat{u}_j, \hat{v}_i) \mathrm{d}\Omega \\
&= \int_\Omega \frac{\partial \xi}{\partial p} \frac{\partial(\sum_k p_k \hat{p}_k)}{\partial p_k} a_1(\hat{u}_j, \hat{v}_i) \mathrm{d}\Omega \\
&= \int_\Omega \frac{\partial \xi}{\partial p} a_1(\hat{u}_j, \hat{v}_i) \hat{p}_k \mathrm{d}\Omega
\end{aligned}$$

Adjoint Method in Gridap

- Now consider the derivative

$$
\begin{aligned}
\frac{\mathrm{d}g}{\mathrm{d}p_k} &= -\lambda^T \frac{\partial A}{\partial p_k}\mathbf{u} \\
&= -\sum_{ij} \lambda_i \left[\int_\Omega \frac{\partial \xi}{\partial p} a_1(\hat{u}_j, \hat{v}_i)\hat{p}_k \mathrm{d}\Omega\right] u_j \\
&= -\int_\Omega \frac{\partial \xi}{\partial p} a_1(\sum_j u_j \hat{u}_j, \sum_i \lambda_i \hat{v}_i)\hat{p}_k \mathrm{d}\Omega \\
&= -\int_\Omega \frac{\partial \xi}{\partial p} a_1(u(x), \lambda(x))\hat{p}_k \mathrm{d}\Omega \\
&= \int_\Omega dG(x)\hat{p}_k \mathrm{d}\Omega
\end{aligned}
$$

where $dG(x) = -\frac{\partial \xi}{\partial p} a_1(u(x), \lambda(x))$

Adjoint Method in Gridap

- Compare to the source term

$$
\begin{aligned}
b_i &= b(\hat{v}_i) \\
&= \int_\Omega f(x)\hat{v}_i \mathrm{d}\Omega
\end{aligned}
$$

- We can see that the derivative $\frac{\mathrm{d}g}{\mathrm{d}p_k}$ is equivalent to a source term with the function $f(x) \to dG(x)$

Code

$$
\begin{aligned}
dG(p,u,v,dp) &= -dxidp(p) * a\_1(u,v) * dp \\
uh &= FEFunction(U, uvec) \\
ph &= FEFunction(P, pvec) \\
\lambda h &= FEFunction(V, \lambda vec) \\
t &= FESource((dp)-> dG(ph, uh, \lambda h, dp), \\
& \quad trian, quad)
\end{aligned}
$$

Questions?