2022 Research Interim Report

Qichen Eric Dai

University of Toronto
Engineering Science 2T3 + PEY
Aerospace Option

July 21, 2022

Mathematical Foundations

Mathematical Foundations

Variational Methods

find a function u such that it satisfies

$$a(u, v) = l(v) \quad \forall v \in \mathcal{V}$$

where $a(u, v)$ is the bilinear form, $l(v)$ is the linear form of the variational problem. $\mathcal{V}$ is the function space.

Finite Element Method

After triangulating our domain into multiple segments, we want to find a piecewise continuous function $u_h \in \mathcal{V}_h$ that satisfies

$$a(u_h, v) = l(v) \quad \forall v \in \mathcal{V}_h$$

Where $\mathcal{V}_h$ is a space of piecewise linear functions defined over our triangulation of the domain.

Finite Element Method Formulation

We can represent u_h and v using Lagrange basis:

$$u_h = \hat{u}_{h,j}\phi_j$$
$$v = \hat{v}_i\phi_i$$

where $\hat{u}_h$ and $\hat{v}_h$ are the function values at the nodes of our triangulations, and ϕ_i is the i-th piecewise Lagrange basis polynomial.

Finite Element Method Formulation:

The problem became: find $\hat{u}_h$ such that

$$a(\phi_j, \phi_i)\hat{u}_h = l(\phi_i) \quad \forall i = 1, 2, \ldots, N$$

Visualization of Lagrange Polynomials

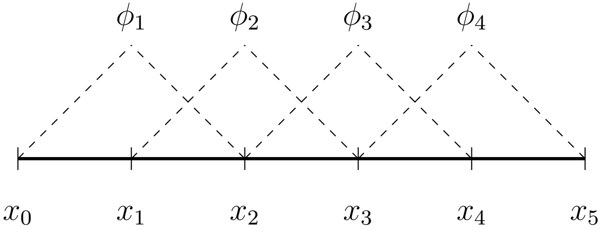

Figure: Piecewise linear lagrange basis polynomials with 4 degrees of freedom

Vectorization of Problem

find $\hat{u}_h$ such that

$$a(\phi_j, \phi_i)\hat{u}_h = l(\phi_i) \quad \forall i = 1, 2, \ldots, N$$

$$\rightarrow \begin{bmatrix} a(\phi_1, \phi_1) & \ldots & a(\phi_N, \phi_1) \\ \vdots & \ddots & \vdots \\ a(\phi_1, \phi_N) & \ldots & a(\phi_N, \phi_N) \end{bmatrix} \begin{bmatrix} \hat{u}_{h,1} \\ \vdots \\ \hat{u}_{h,N} \end{bmatrix} = \begin{bmatrix} l(\phi_1) \\ \vdots \\ l(\phi_N) \end{bmatrix}$$

$$\rightarrow \mathbf{A}\mathbf{u} = \mathbf{f}$$

Question 1 - Linear FEM

Question 1 - Linear FEM

Sample Problem Statement

Linear reaction-diffusion

We have a physical domain $\Omega \equiv (0,1) \in \mathbb{R}^1$ and a parameter domain $\mathcal{D} \equiv (10^{-4}, 10^{-1}) \subset \mathbb{R}^1$. Reaction-diffusion system is given by the following problem statement:
Given $\mu \in \mathcal{D}$, find $u(\mu) : \Omega \to \mathcal{R}$ such that:

$$
\begin{aligned}
-\mu \Delta u(\mu) + u(\mu) &= 1 \quad \text{in } \Omega \\
u(\mu) &= 0 \quad \text{on } \partial\Omega
\end{aligned}
$$

Question 1 - Linear FEM

Weak Form Formulation

To find the weak form, we adhere to the given example in the AER1418 note, multiplying the ODE by a test function v and integrate on both side:

$$\int_\Omega v(-\mu\Delta u + u)dx = \int_\Omega v dx$$

We integrate the first term by part:

$$-\int v\mu\Delta u dx = -\left[\cancelto{0}{\mu v\frac{\mathrm{d}u}{\mathrm{d}x}\bigg|_0^1} - \int \mu\frac{\mathrm{d}v}{\mathrm{d}x}\frac{\mathrm{d}u}{\mathrm{d}x}dx\right]$$

$$= \mu\int\frac{\mathrm{d}v}{\mathrm{d}x}\frac{\mathrm{d}u}{\mathrm{d}x}dx$$

Weak Form Formulation

Weak form of reaction - diffusion equation:

$$\boxed{\therefore \mu \int \frac{\mathrm{d}v}{\mathrm{d}x}\frac{\mathrm{d}u}{\mathrm{d}x} dx + \int uv dx = \int v dx}$$

Biliner Form:

$$a(u, v) = \mu \int \frac{\mathrm{d}v}{\mathrm{d}x}\frac{\mathrm{d}u}{\mathrm{d}x} dx + \int uv dx$$

Linear Form:

$$l(v) = \int v dx$$

Question 1 - Linear FEM

Well-Posedness of Problem

Lax - Milgram Theorem: Given a Hilbert Space $\mathcal{V}$, a continuous, coercive bilinear form $a : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$, and a continuous linear functional $l \in \mathcal{V}'$, there exists a unique $u \in \mathcal{V}$ such that

$$a(u, v) = l(v) \quad \forall v \in \mathcal{V}$$

where $\mathcal{V} = H^1(\Omega)$

Can prove:

- a is coercive and continuous;
- l is also continuous.

Question 1 - Linear FEM

- Problem Statement
- Weak Form Formulation
- Well-Posedness
- 1D FEM Solver

1D FEM Solver

Weak Form Rewritten:

$$a(\phi_j, \phi_i)\hat{u}_{h,j} = l(\phi_i) \quad \forall i = 1, 2, \ldots, N$$

$$\rightarrow \left(\mu \int \frac{\mathrm{d}\phi_i}{\mathrm{d}x}\frac{\mathrm{d}\phi_j}{\mathrm{d}x}dx + \int \phi_i\phi_j dx\right)\hat{u}_{h,j} = \int \phi_i dx$$

We can compute the integrals using Gauss quadrature

Gauss Quadrature

We wish to approximate the integral with a finite summation

$$\int_\Omega f(x)dx \approx w_i f(x_i)$$

w_i-s are the integration weights and the x_i-s are the quadrature points. The weights and quadrature points are predefined in tables.

N-point Gauss quadrature rule can integrate $2N - 1$ degree polynomial exactly.

Integration Matrices

For a domain with K degrees of freedom, we can apply a total of N-points Gauss Quadrature rule over the entire domain. Can define:

Weight Vector $\mathbf{w} : \mathbb{R}^N$ contains the quadrature weights to all quadrature points in the domain.
Basis Matrix $\mathbf{\Phi} : \mathbb{R}^{N \times K}$ where its entry ϕ_{ij} is the function value of the j-th basis function evaluated at the i-th quadrature point.
Derivative Basis Matrix $\mathbf{\Phi}' : \mathbb{R}^{N \times K}$ where its entry ϕ'_{ij} is the derivative value of the j-th basis function evaluated at the i-th quadrature point.

Integrating Functions

Integrating our solution over the domain became:

$$\begin{aligned}\int_\Omega u(x;\mu)dx &\approx w_i u(x_i)\\ &= w_i \phi_{ij}\hat{u}_{h,j}\\ &= \mathbf{w}^\intercal \mathbf{\Phi}\mathbf{u}\end{aligned}$$

The stiffness matrix $\mathbf{A}_h$ is:

$$\begin{aligned}\mathbf{A} &= \mu\mathbf{\Phi}'^\intercal \mathbf{W}\mathbf{\Phi}' + \mathbf{\Phi}^\intercal\mathbf{W}\mathbf{\Phi}\\ &= \mu\mathbf{A}_1 + \mathbf{A}_2\end{aligned}$$

Where $\mathbf{W} = \text{diag}(\mathbf{w})$

And the load vector $\mathbf{f}$ is

$$\mathbf{f} = \mathbf{\Phi}^\intercal\mathbf{w}$$

Finite Element Result

Solving the linear equation, we can get the following results in MATLAB:

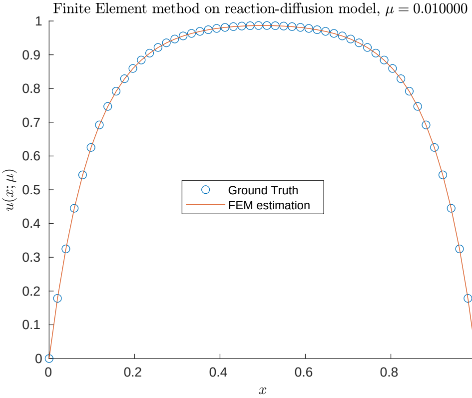

Figure: Finite element method result on reaction-diffusion equation with $\mu = 0.01$, using linear basis functions

Finite Element Result

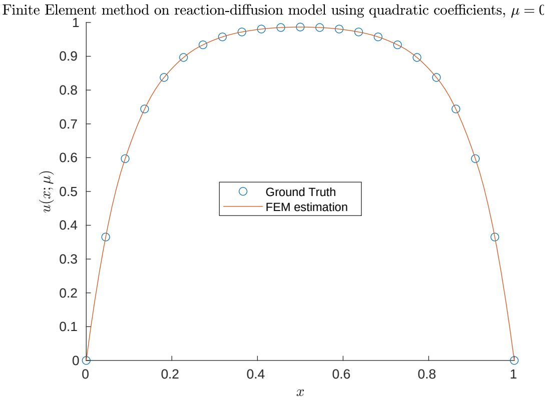

Figure: Finite element method result on reaction-diffusion equation with $\mu = 0.01$, using quadratic basis functions

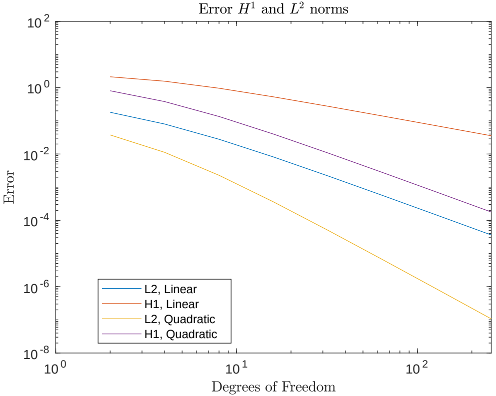

Figure: Finite Element error convergence rate

Functional Evaluation

Now we can compute our approximation solution u_h, we wish to approximate a desired value $s \equiv \int_\Omega 1 - u(x;\mu)\,dx$. Using our integration matrix, we can get:

$$\begin{aligned} s &= \int_0^1 1 - u(x;\mu)dx \\ &= \int_0^1 1dx - \int_0^1 u(x;\mu)dx \\ &= 1 - \mathbf{w}^\intercal \mathbf{\Phi} \mathbf{u} \end{aligned}$$

Question 2 - Optimization

- Problem Statement
- Gradient-Free optimization
- Adjoint-Based Method
- Gradient-Based Optimization

Question 2 - Optimization

- Problem Statement
- Gradient-Free optimization
- Adjoint-Based Method
- Gradient-Based Optimization

We wish to inversely find the μ_{true} value that gives us the desired s_{true} value. The objective is to

$$\min\ (s_{true} - s)^2$$
$$s = \int_\Omega (1 - u(x;\mu))\ dx$$

Question 2 - Optimization

- Problem Statement
- Gradient-Free optimization
- Adjoint-Based Method
- Gradient-Based Optimization

Gradient-Free Optimization

```
>> grad_free
                                                        First-order
 Iteration  Func-count       f(x)          Step-size     optimality
     0           2        0.0121151                         0.864
     1          10      0.000223879       0.0295014         0.186
     2          12      0.000140069               1         0.134
     3          14      9.40704e-07               1        0.0114
     4          16      4.31609e-09               1      0.000774
     5          18      9.55124e-14               1      4.16e-06
     6          20      1.91625e-15               1      1.51e-09

Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>
Elapsed time is 0.424427 seconds.
mu = 0.0269833, true mu = 0.0269850
```

Figure: Gradient-free optimization result

Question 2 - Optimization

- Problem Statement
- Gradient-Free optimization
- Adjoint-Based Method
- Gradient-Based Optimization

Adjoint-Based Method

For an ODE/PDE constrained optimization problem, we wish to:

$$\text{Minimize } J(u) = q(u(\mu); \mu)$$
$$\text{Subject to } R(u(\mu); \mu) = 0$$

We can compute the Jacobian of the functional: $\dfrac{\partial J}{\partial \mu} = \dfrac{\partial q}{\partial u}\dfrac{\partial u}{\partial \mu} + \dfrac{\partial q}{\partial \mu}$

Since $\dfrac{\partial R}{\partial u}\dfrac{\partial u}{\partial \mu} + \dfrac{\partial R}{\partial \mu} = 0$ from the residual constraint, we can rewrite

$$\frac{\partial u}{\partial \mu} = -\left(\frac{\partial R}{\partial u}\right)^{-1} \frac{\partial R}{\partial \mu}$$

Adjoint-Based Method

Thus we can write our Jacobian to be:

$$
\begin{aligned}
\frac{\partial J}{\partial \mu} &= -\frac{\partial q}{\partial u}\left(\frac{\partial R}{\partial u}\right)^{-1}\frac{\partial R}{\partial \mu} + \frac{\partial q}{\partial \mu} \\
&= -\psi^{\mathsf{T}}\frac{\partial R}{\partial \mu} + \frac{\partial q}{\partial \mu}
\end{aligned}
$$

where ψ is the adjoint that satisfies

$$
\left(\frac{\partial R}{\partial u}\right)^{\mathsf{T}}\psi = \left(\frac{\partial q}{\partial u}\right)^{\mathsf{T}}
$$

Adjoint-Based Method

Our question can be rewritten as:

$$\text{Minimize } J(\mathbf{u}) = (s_{true} - s)^2 = (s_{true} - 1 + \mathbf{w}^\intercal \mathbf{\Phi} \mathbf{u})^2$$

$$\text{Subject to } \mathbf{R}(\mathbf{u}) = \mathbf{A}\mathbf{u} - \mathbf{f} = (\mu \mathbf{A}_1 + \mathbf{A}_2)\, \mathbf{u} - \mathbf{f} = 0$$

Using the adjoint-based method formulation, we have

$$\frac{\partial q}{\partial u} = 2(s_{true} - 1 + \mathbf{w}^\intercal \mathbf{\Phi} \mathbf{u}) \mathbf{w}^\intercal \mathbf{\Phi}$$

$$\frac{\partial q}{\partial \mu} = 0$$

$$\frac{\partial R}{\partial u} = \mathbf{A}$$

$$\frac{\partial R}{\partial \mu} = \mathbf{A}_1 \mathbf{u}$$

Question 2 - Optimization

- Problem Statement
- Gradient-Free optimization
- Adjoint-Based Method
- Gradient-Based Optimization

Gradient-Based Optimization Result

Using the formulation from adjoint-based method, we can construct a gradient-based optimization scheme. The resulting runtime is shown below:

```
>> optim_grad
    0.0943



                                        Norm of      First-order
 Iteration           f(x)               step          optimality   CG-iterations
     0            0.0274198                             1.48
     1           0.00623217          0.0230088          0.504             1
     2          0.000660485          0.0187058          0.131             1
     3          1.43326e-05         0.00906815         0.0174             1
     4          9.63375e-09         0.00161849       0.000443             1
     5           4.6507e-15         4.3433e-05       3.08e-07             1

Local minimum possible.

fminunc stopped because the final change in function value relative to
its initial value is less than the value of the function tolerance.

<stopping criteria details>
Elapsed time is 0.044588 seconds.
mu = 0.0942773, true mu = 0.0942794
```

Figure: Gradient-based optimization result using adjoint-based method

Comparison

Method	iterations	runtime(s)
Gradient-Free	6	0.424427
Gradient-Based	5	0.04588

Table: Runtime comparison between gradient-based and gradient-free optimization

Question 3 - Optimization Constrained By Non-Linear PDE

Question 3 - Optimization Constrained By Non-Linear PDE

Problem Statement

Given $\mu \in \mathcal{D}$, find $u(x;\mu)$ such that

$$\begin{aligned} -\nabla \cdot (\mu_i \nabla u(\mu)) + u(\mu)^3 &= 1 \quad \text{in } \Omega_i \ \forall i = 1, ..., P \\ u(\mu) &= 0 \quad \text{on } \partial\Omega \end{aligned}$$

And

$$\begin{aligned} \min \quad & (s_{true} - s)^2 \\ \text{subject to} \quad & -\nabla \cdot (\mu_i \nabla u(\mu)) + u(\mu)^3 - 1 = 0 \\ \text{where} \quad & s = \int (1 - u(x;\mu)) \, dx \end{aligned}$$

Question 3 - Optimization Constrained By Non-Linear PDE

Weak Form

The weak form of the equation can be derived by multiplying test function v :

$$\int v(-\mu\Delta u + u^3)\; dx = \int v\; dx$$

$$\rightarrow \mu \int \nabla u \nabla v\; dx + \int u^3 v\; dx = \int v\; dx$$

Unfortunately the weak form is non-linear, thus must be solved using root-finding algorithms like Newton's Method.

Question 3 - Optimization Constrained By Non-Linear PDE

Residual

Substituting $u = u_{h,j}\phi_j$ and $v = v_i\phi_i$, we can define residual vector $\mathbf{R}$ such that:

$$R_i = \mu \int \frac{\mathrm{d}\phi_j}{\mathrm{d}x}\frac{\mathrm{d}\phi_i}{\mathrm{d}x}\hat{u}_{h,j}\ dx + \int (\hat{u}_{h,k}\phi_k)^3\phi_i\ dx - \int \phi_i\ dx$$

$$\mathbf{R} = \mu\mathbf{A}_1\mathbf{u} + \left[(\mathbf{\Phi}\mathbf{u})^{\circ 3} \circ \mathbf{\Phi}\right]^{\mathsf{T}}\mathbf{w} - \mathbf{f}$$

where $\circ$ is the Hadamard product.

Jacobian

We can also define the Jacobian of the residual with respect to $\mathbf{u}$

$$\mathcal{J}_{ij} = \frac{\partial R_i}{\partial u_j} = \mu \int \nabla \phi_j \cdot \nabla \phi_i \; dx + \int 3(\hat{u}_{h,k} \phi_k)^3 \phi_i \phi_j \; dx$$

$$\boldsymbol{\mathcal{J}} = \mu \mathbf{A}_1 + \mathbf{\Phi}^\intercal \mathbf{Z} \mathbf{\Phi}$$

Where $\mathbf{Z} = \text{diag}(3\mathbf{w} \circ (\mathbf{\Phi u})^{\circ 2})$

Newton's Method

We can use Newton's Methods to numerically find the root of the residual:

$$\mathbf{u}^{i+1} = \mathbf{u}^{i} + \delta\mathbf{u}$$

where $\boldsymbol{\mathcal{J}}\delta\mathbf{u} = -\mathbf{R}$

```
>> demo
1 2.17640516e-02
2 2.24634793e-03
3 1.90450115e-05
4 1.38134265e-09
```

Figure: Newton's method error shows quadratic convergence as expected

Result

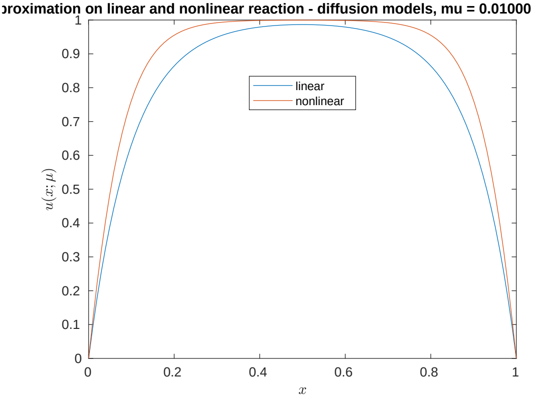

Figure: Finite element method result on non-linear reaction-diffusion equation with $\mu = 0.01$, using quadratic basis functions

Question 3 - Optimization Constrained By Non-Linear PDE

Adjoint-Based Optimization

We wish to do the same thing as question 2 to inversely find the μ value that gives the desired s value. Using adjoint optimization we can redefine our question to:

$$\text{Minimize} \quad J(\mathbf{u};\mu) = (s_{true} - s)^2 = (s_{true} - 1 + \mathbf{w}^\intercal \boldsymbol{\Phi}\mathbf{u})^2$$

$$\text{Subject to} \quad \mathbf{R} = \mu \mathbf{A}_1 \mathbf{u} + \left[(\boldsymbol{\Phi}\mathbf{u})^{\circ 3} \circ \boldsymbol{\Phi}\right]^\intercal \mathbf{w} - \mathbf{f}$$

The components of the adjoint-based method are:

$$\frac{\partial J}{\partial \mathbf{u}} = 2(s_{true} - 1 + \mathbf{w}^\intercal \boldsymbol{\Phi}\mathbf{u})\mathbf{w}^\intercal \boldsymbol{\Phi}$$

$$\frac{\partial J}{\partial \mu} = 0$$

$$\frac{\partial \mathbf{R}}{\partial \mathbf{u}} = \boldsymbol{\mathcal{J}}$$

$$\frac{\partial \mathbf{R}}{\partial \mu} = \mathbf{A}_1 \mathbf{u}$$

Result

```
>> optim_nonlin_gradfree
                                                          First-order
 Iteration  Func-count       f(x)          Step-size       optimality
     0           2       0.000500672                          0.234
     1           8       9.41675e-05          0.01            0.109
     2          10       4.76115e-06             1            0.0264
     3          12       4.05795e-08             1            0.0024
     4          14       1.65018e-11             1          4.89e-05
     5          16       2.72037e-15             1          9.25e-08

Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>
Elapsed time is 0.225611 seconds.
mu = 0.01340377, true mu = 0.01340378
```

Figure: Non-linear PDE constrained optimization result using gradient-free method

Result

```
>> optim_nonlin_adjoint

                                         Norm of      First-order
 Iteration             f(x)              step          optimality   CG-iterations
     0              0.0754556                              4.13
     1              0.0425862           0.0115499          2.02                1
     2              0.0150892            0.020641         0.853                1
     3             0.00197385           0.0253044          0.25                1
     4            3.37869e-05           0.0143498          0.03                1
     5            1.10081e-08          0.00222396      0.000535                1
     6            1.19818e-15         4.11175e-05      1.77e-07                1

Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>
Elapsed time is 0.073293 seconds.
mu = 0.08254883, true mu = 0.08254885
```

Figure: Non-linear PDE constrained optimization result using adjoint-based method

Result

Method	iterations	runtime(s)
Gradient-Free	5	0.225611
Gradient-Based	6	0.073923

Table: Runtime comparison between gradient-based and gradient-free optimization