Finite element method for ODEs

Yihang Chen, 1700010780

December 29, 2019

Contents

1 Settings

In this report, we will solve the following ODE by the finite element method (FEM).

$$\begin{cases} -u'' + u = f, \quad x \in (0,1) \\ u(0) = 0, \quad u'(1) + u(1) = g \end{cases} \tag{1}$$

2 FEM form

Define $\mathbb{V} = \{f \in \mathbb{H}^1(0,1) | f(0) = 0\}$, the variation form can be formalized as: Find $u \in \mathbb{V}$, for all $\phi \in \mathbb{V}$

$$\int_0^1 (u\phi + u'\phi')\ \mathrm{d}x + u(1)\phi(1) = \int_0^1 f\phi\ \mathrm{d}x + g\phi(1) \tag{2}$$

Then for the finite element space $\mathbb{V}_n \subset \mathbb{V}$, suppose $0 = x_0 < x_1 < \cdots < x_{n-1} < x_n = 1$. We construct base function $\lambda_i,\ \forall 1 \le i \le n$ below.

For $j = 1, 2, \dots, n-1$.

$$\lambda_j(x) = \begin{cases} \frac{x_j - x}{x_j - x_{j-1}}, & x \in [x_{j-1}, x_j] \\ \frac{x - x_{j-1}}{x_j - x_{j-1}}, & x \in [x_{j-1}, x_j] \\ 0, & \text{others} \end{cases} \tag{3}$$

For $j = n$,

$$\lambda_n(x) = \begin{cases} \frac{x_n - x}{x_n - x_{n-1}}, & x \in [x_{n-1}, x_n] \\ 0, & \text{others} \end{cases} \tag{4}$$

Assume the finite element solution is $u_h = \sum_{i=1}^n u_i \lambda_i(x)$, then by setting $\phi = \lambda_i$ in Equation 2, we get

$$\sum_{i=1}^n u_i \int_0^1 \lambda_i \lambda_j + \lambda_i' \lambda_j' \, \mathrm{d}x = \int_0^1 f \lambda_j \, \mathrm{d}x, \quad \forall j = 1, 2, \dots, n-1 \tag{5}$$

For $j = n$,

$$u_n + \sum_{i=1}^n u_i \int_0^1 \lambda_i \lambda_n + \lambda_i' \lambda_n' \, \mathrm{d}x = \int_0^1 f \lambda_n \, \mathrm{d}x + g \tag{6}$$

In the following, $h_j = x_j - x_{j-1}, I_j = [x_{j-1}, x_j]$. We can explicitly calculate that, for $j = 1.2. \dots, n-1$,

$$\int_0^1 \lambda_i \lambda_j \, \mathrm{d}x = \begin{cases} \frac{h_j}{6}, & i = j-1 \\ \frac{h_j + h_{j+1}}{3}, & i = j \\ \frac{h_{j+1}}{6}, & i = j+1 \end{cases} \tag{7}$$

$$\int_0^1 \lambda_i' \lambda_j' \, \mathrm{d}x = \begin{cases} -\frac{1}{h_i} & i = j-1 \\ \frac{1}{h_j} + \frac{1}{h_{j+1}}, & i = j \\ -\frac{1}{h_{j+1}} & i = j+1 \end{cases} \tag{8}$$

$$\int_0^1 f \lambda_j \, \mathrm{d}x \approx \frac{h_j f(x_{j-1}) + 2h_j f(x_j) + 2h_{j+1} f(x_i) + h_{j+1} f(x_{j+1})}{6} \tag{9}$$

For $j = n$,

$$\int_0^1 \lambda_i \lambda_n \, \mathrm{d}x = \begin{cases} \frac{h_n}{6}, & i = n-1 \\ \frac{h_n}{3}, & i = n \end{cases} \tag{10}$$

$$\int_0^1 \lambda_i' \lambda_n' \, \mathrm{d}x = \begin{cases} -\frac{1}{h_n}, & i = n-1 \\ \frac{1}{h_n}, & i = n \end{cases} \tag{11}$$

$$\int_0^1 f \lambda_n \, \mathrm{d}x \approx \frac{h_n f(x_n)}{2} \quad \text{(middle point)} \tag{12}$$

Then we can convert the Equation 5, 6 to a linear equation in the form

$$A_n u = F_n \tag{13}$$

The explicit form of A_n will be obvious in the code fem.m, hence, we do not write it down here. Besides, the uniqueness of the solution can be directly derived from the fact that A_n is diagonal dominant.

3 Error estimation

3.1 Prior estimation

We can easily derive that $\forall \phi \in C_0^\infty(0,1)$,

$$-\int_0^1 u'\phi' \, \mathrm{d}x = \int_0^1 (f-u)\phi \, \mathrm{d}x \tag{14}$$

Besides, $u \in \mathbb{H}^1(0.1)$, and $\mathbb{H}^1(0,1) \hookrightarrow \mathbb{C}(0,1)$. Thus, $f-u \in \mathbb{C}(0,1)$. Owing to the definition of weak derivative, u'' is continuous. Hence, $u \in \mathbb{H}^2(0,1)$.
Assume the finite element solution is u_h, and $\Omega = (0,1)$, the regularity condition implies the estimation

$$\|u - u_h\|_{1,\Omega} \leq Ch|u|_{2,\Omega} \tag{15}$$

Assuming some regularity condition in the Aubin-Nische lemma, we got the estimation

$$\|u - u_h\|_{0,\Omega} \leq Ch^2|u|_{2,\Omega} \tag{16}$$

3.2 Posterior estimation

In the adaptive grid, we use posterior error estimator to decide which interval to refine. Specifically, we use

$$\eta_i = h_i\|f_{I_i} - u_h\|_{0,2,I_i} \quad \text{where } f_{I_i} = \frac{\int_{I_i} f \, \mathrm{d}x}{h_i} \tag{17}$$

4 Numerical test

We consider two different real solutions:

- Case 1: $u(x) = \sin(10\pi x)$
- Case 2: $u(x) = \exp(-10(x-0.5)^2) - \exp(-5/2)$

$f(x), g$ will be defined accordingly.

In numerically calculating the error, we directly use

```
norm(Err, inf)
```

to calculate the l^∞ norm, but we use

```
err = sqrt(sum(Err.^2.*h));
```

to approximate the l^2 norm

$$\left(\int_0^1 (u_h - u)^2 \, \mathrm{d}x\right)^{1/2} \quad , \quad \text{and} \quad \left(\int_0^1 (u_h' - u')^2 \, \mathrm{d}x\right)^{1/2} \tag{18}$$

The l^∞ error and l^2 error of the function value is $\mathcal{O}(n^{-2})$, and the l^∞ error and l^2 error of the derivative is $\mathcal{O}(n^{-1})$, which is aligned with the theoretical results.

4.1 Grid type

Uniform grid In the case, $x_i = i/n$.

Random grid We generate uniformly random variables in (0,1) as $\hat{h}_i$, and scale them to h_i so that $\sum_i h_i = 1$.

Perturbed grid For size n grid, we first generate a uniform grid, and then perturb the grid point. In other words, $x_i \in [(i-\epsilon)/n, (i+\epsilon)/n$. Practically, we set $\epsilon = 0.1$

Adaptive grid We use the posterior estimator 17. Assume

$$\eta = \max_i \eta_i \tag{19}$$

If $\eta_i > \alpha\eta$, then we add the middle point of I_i into the grid.

4.2 Non-adaptive grid

We set $n = 8000, 8200, 8400, \ldots, 10000$ for case 1, and $4000, 4200, \ldots, 6000$ for case 2. To reproduce the results, please execute

```
convergence(choice, grid, err_type, derivative)
```

where

- choice = 1 or 2 representing case 1 or case 2
- grid = 'uniform' or "random" or "perturbed".
- err_type = 'inf' or '2' represent l^∞ norm or l^2 norm
- derivative = 0 or 1 representing calculating the value of the error or the derivative.

The error figure is in the folder "./Figure collections/perturbed", "./Figure collections/uniform", "./Figure collections/random", with file names transparent to understand.

Table 1: Uniform grid, value

n	8000	8200	8400	8600	8800	9000	9200	9400	9600	9800	10000
log l^∞, case 1	-9.8570	-9.9063	-9.9546	-10.0016	-10.0476	-10.0925	-10.1365	-10.1795	-10.2216	-10.2628	-10.3032
log l^2 case 1	-10.4650	-10.5144	-10.5626	-10.6096	-10.6556	-10.7005	-10.7445	-10.7875	-10.8296	-10.8709	-10.9113

Table 2: Uniform grid, value

n	4000	4200	4400	4600	4800	5000	5200	5400	5600	5800	6000
log l^∞, case 2	-15.3014	-15.4006	-15.4922	-15.5796	-15.6660	-15.7490	-15.8222	-15.8997	-15.9729	-16.0512	-16.1158
log l^2 case 2	-16.0421	-16.1419	-16.2331	-16.3199	-16.4068	-16.4903	-16.5615	-16.6397	-16.7131	-16.7944	-16.8580

Table 3: Uniform grid, derivative

n	8000	8200	8400	8600	8800	9000	9200	9400	9600	9800	10000
$\log l^\infty$, case 1	-2.7847	-2.8094	-2.8336	-2.8571	-2.8801	-2.9026	-2.9246	-2.9461	-2.9672	-2.9878	-3.0081
$\log l^2$ case 1	-3.1323	-3.1570	-3.1811	-3.2046	-3.2276	-3.2501	-3.2721	-3.2936	-3.3146	-3.3352	-3.3555

Table 4: Uniform grid, derivative

n	4000	4200	4400	4600	4800	5000	5200	5400	5600	5800	6000
$\log l^\infty$, case 2	-5.9914	-6.0402	-6.0867	-6.1312	-6.1737	-6.2145	-6.2538	-6.2915	-6.3279	-6.3630	-6.3969
$\log l^2$ case 2	-6.6250	-6.6738	-6.7203	-6.7648	-6.8074	-6.8482	-6.8874	-6.9251	-6.9615	-6.9966	-7.0305

Table 5: Random grid, value

n	8000	8200	8400	8600	8800	9000	9200	9400	9600	9800	10000
$\log l^\infty$, case 1	-9.6758	-8.9533	-9.2742	-9.7289	-9.7965	-8.9342	-8.9246	-10.4259	-9.4119	-9.2822	-10.2778
$\log l^2$ case 1	-10.6134	-10.6844	-9.9841	-10.7182	-10.9876	-11.0380	-9.9820	-10.6699	-11.3024	-10.0695	-9.9954

Table 6: random grid, value

n	4000	4200	4400	4600	4800	5000	5200	5400	5600	5800	6000
$\log l^\infty$, case 2	-15.5131	-14.5794	-15.1713	-14.8216	-14.6269	-14.7332	-15.8504	-14.9044	-15.5822	-16.1597	-16.0817
$\log l^2$ case 2	-15.9560	-15.2143	-15.5980	-16.2123	-16.2883	-16.4203	-16.4518	-16.1496	-16.2878	-16.6784	-16.5455

Table 7: random grid, derivative

n	8000	8200	8400	8600	8800	9000	9200	9400	9600	9800	10000
$\log l^\infty$, case 1	-2.0927	-2.1159	-2.1436	-2.1632	-2.1989	-2.2175	-2.2365	-2.2460	-2.2783	-2.2982	-2.3174
$\log l^2$ case 1	-2.7846	-2.8147	-2.8418	-2.8606	-2.8828	-2.9108	-2.9339	-2.9465	-2.9704	-2.9847	-3.0039

Table 8: Random grid, derivative

n	4000	4200	4400	4600	4800	5000	5200	5400	5600	5800	6000
$\log l^\infty$, case 2	-5.3110	-5.3386	-5.3965	-5.4585	-5.5000	-5.5315	-5.5756	-5.6017	-5.6371	-5.6616	-5.7100
$\log l^2$ case 2	-6.2781	-6.3314	-6.3701	-6.4154	-6.4709	-6.5018	-6.5372	-6.5798	-6.6198	-6.6437	-6.6914

Table 9: Perturbed grid, value

n	8000	8200	8400	8600	8800	9000	9200	9400	9600	9800	10000
$\log l^\infty$, case 1	-9.8035	-9.9746	-9.9120	-10.0734	-10.1683	-10.0087	-10.1196	-10.1306	-10.2760	-10.1970	-10.2150
$\log l^2$ case 1	-10.4114	-10.5823	-10.5204	-10.6814	-10.7760	-10.6171	-10.7276	-10.7385	-10.8835	-10.8049	-10.8230

Table 10: Perturbed grid, value

n	4000	4200	4400	4600	4800	5000	5200	5400	5600	5800	6000
log l^∞, case 2	-15.2434	-15.4639	-15.3856	-15.5474	-15.7140	-15.6741	-15.7971	-15.9024	-15.9312	-15.9178	-15.9988
log l^2 case 2	-15.9850	-16.2006	-16.1279	-16.2888	-16.4519	-16.4163	-16.5385	-16.6427	-16.6729	-16.6602	-16.7413

Table 11: Perturbed grid, derivative

n	8000	8200	8400	8600	8800	9000	9200	9400	9600	9800	10000
log l^∞, case 1	-2.6154	-2.6339	-2.6604	-2.6870	-2.7083	-2.7249	-2.7499	-2.7682	-2.7911	-2.8196	-2.8331
log l^2 case 1	-3.1223	-3.1469	-3.1710	-3.1947	-3.2178	-3.2403	-3.2621	-3.2835	-3.3046	-3.3253	-3.3456

Table 12: Perturbed grid, derivative

n	4000	4200	4400	4600	4800	5000	5200	5400	5600	5800	6000
log l^∞, case 2	-5.8326	-5.8662	-5.9122	-5.9563	-6.0158	-6.0554	-6.0942	-6.1353	-6.1705	-6.2044	-6.2370
log l^2 case 2	-6.6153	-6.6641	-6.7105	-6.7549	-6.7974	-6.8382	-6.8774	-6.9152	-6.9516	-6.9867	-7.0206

4.3 Adaptive grid

We use initial grid of size 500, and refine it for 10 times. To reproduce the results, please execute

```
adapt_test(iter, alpha, choice, err_type, derivative)
```

where "iter" represents the number of derivative, and "alpha" specify the threshold. We consider two type of adaptive grid, i.e. $\alpha = 0.4, 0.5$. The error figures are in the folder "./adaptive"

Table 13: Adaptive grid, value

iter	1	2	3	4	5	6	7	8	9	10
log l^∞, case 1, $\alpha = 0.4$	-4.5951	-4.6077	-4.5989	-5.6982	-7.0844	-7.3446	-7.4366	-7.4722	-7.4844	-7.4879
log l^2 case 1, $\alpha = 0.4$	-5.2338	-5.1985	-5.2064	-5.2084	-6.3055	-7.6921	-7.9387	-8.0444	-8.0816	-8.0930
log l^∞, case 2, $\alpha = 0.4$	-10.8025	-11.0924	-11.3614	-11.5078	-12.8659	-12.9409	-14.3281	-14.2674	-15.0852	-15.8579
log l^2 case 2, $\alpha = 0.4$	-11.2756	-11.6056	-11.9250	-12.1004	-13.4535	-13.5442	-14.9415	-14.8039	-15.3328	-16.2643
log l^∞, case 1, $\alpha = 0.5$	-4.5994	-4.5892	-4.5970	-4.5989	-5.6982	-7.0844	-7.3288	-7.4337	-7.4705	-7.4818
log l^2 case 1, $\alpha = 0.5$	-5.2338	-5.1985	-5.2064	-5.2084	-6.3055	-7.6921	-7.9387	-8.0444	-8.0816	-8.0930
log l^∞, case 2, $\alpha = 0.5$	-10.6824	-10.8826	-11.1714	-11.3693	-11.4842	-12.8058	-12.8971	-12.9435	-14.3124	-14.3594
log l^2 case 2, $\alpha = 0.5$	-11.1107	-11.3308	-11.6947	-11.9348	-12.0704	-13.3824	-13.4928	-13.5474	-14.9193	-14.9831

Table 14: Adaptive grid, derivative

n	1	2	3	4	5	6	7	8	9	10
log l^∞, case 1, $\alpha = 0.4$	-0.6821	-0.6826	-0.6827	-0.6985	-1.3955	-2.0864	-2.7746	-3.4575	-4.0412	-4.5625
log l^2 case 1, $\alpha = 0.4$	-1.0072	-1.0498	-1.0530	-1.0530	-1.7461	-2.3990	-3.0700	-3.7566	-4.3842	-5.0194
log l^∞, case 2, $\alpha = 0.4$	-4.6032	-4.7246	-4.8520	-4.9821	-5.6282	-5.7014	-6.3672	-6.4058	-7.0851	-7.1027
log l^2 case 2, $\alpha = 0.4$	-4.9569	-5.2582	-5.4934	-5.8127	-6.2137	-6.6354	-6.9810	-7.4175	-7.8107	-8.1880
log l^∞, case 1, $\alpha = 0.4$	-4.9701	-5.3634	-5.7991	-6.3120	-6.8572	-7.3789	-7.9073	-8.4467	-8.9487	-9.5166
log l^2 case 1, $\alpha = 0.5$	-0.9725	-1.0434	-1.0519	-1.0530	-1.0530	-1.7461	-2.3616	-3.0126	-3.6366	-4.1123
log l^∞, case 2, $\alpha = 0.5$	-4.6030	-4.7202	-4.7483	-4.8590	-4.9638	-5.5757	-5.6546	-5.7063	-6.3580	-6.3934
log l^2 case 2, $\alpha = 0.5$	-4.9569	-5.2582	-5.4934	-5.8127	-6.2137	-6.6354	-6.9810	-7.4175	-7.8107	-8.1880

5 Summary

We can readily observe that under uniform grid, the convergence behavior satisfy the theoretical behavior. According to the figure in the folder ./random and ./perturbed, we find that the convergence behavior of the derivative is much better than the value, since Aubin-Nische lemma requires additional regularity.