WI4205 - 2022/23
Applied Finite Elements
Assignment 1.2

Deadline - 23:59, April 13, 2023

Instructions and assessment criteria to keep in mind:

- A submission for Assignment 1.2 is **required for passing** WI4205.
- This is an **individual/group assignment**. You can work by yourself, but if you wish you can also work in groups of maximum two and submit a joint report.
- The reports need to be **typed in LaTeX or Word**.
- The **deadline** for uploading your solutions to Brightspace is 23:59, April 13, 2023.
- Grading of late submissions will be done as per the **grading protocol posted on Brightspace**; make sure you are familiar with it.
- Provide **clear and motivated answers** to the questions. No/reduced points will be awarded if your solutions are unaccompanied by explanations.

An Implementation of the 1D Finite Element Method (14 points)

Let $\Omega = (0, L) \subset \mathbb{R}$, and let Δ denote a mesh on Ω constructed by choosing points $0 = x_0 < x_1 < \cdots < x_{m-1} < x_m = L$. Define the i-th element of the mesh, denoted Ω_i, as follows,

$$\Omega_i := \begin{cases} [x_{i-1}, x_i) \ , & i = 1, \dots, m-1 \ , \\ [x_{i-1}, x_i] \ , & i = m \ . \end{cases}$$

On this mesh, and given degree $p \geq 0$ and smoothness $-1 \leq k \leq p-1$, define the finite element space $\mathcal{F}(p, k; \Delta)$ as

$$\mathcal{F}(p,k;\Delta) := \Bigg\{ f : \overline{\Omega} \to \mathbb{R} \ \Big| \quad f|_{\Omega_i} \in \mathcal{P}_p \ , \ i = 1, \dots, m \ , \\ \frac{d^r f}{dx^r}(x_i^-) = \frac{d^r f}{dx^r}(x_i^+) \ , \ i = 1, \dots, m-1 \ , \ r = 0, \dots, k \Bigg\} \ .$$

Recall from class that the dimension of this space is $n := m(p+1) - (m-1)(k+1)$. Finally, the number of quadrature points will be denoted by n_q.

In this assignment, you will implement the finite element method into a computer code that can utilize the above finite element spaces to solve problems on Ω. You are asked to do this in the modular fashion explained in class, and the following are the necessary steps. The following is important information to keep in mind:

- You are free to use your programming language of choice. However, the scripts accompanying the assignment are only in MATLAB and Python. (To use them in your programming language, you can either reimplement them or you can run them for each case in MATLAB/Python, store the data in CSV files, and read them in your code.)

- When necessary, in all of the following, the size of input/output matrices in the MATLAB scripts is denoted in red as $[a, b]$, and in the Python scripts is denoted in blue as $[a, b]$.
- **The equations use one-based indexing for all variables. For MATLAB, this is equivalent to what you will see in your code; for Python you will need to shift by one.**

Reference element: quadrature scheme and basis

You **are given** a function :

```
function create_ref_data($n_q$, $p$, integrate_flag)
    ⋮
return ref_data
```

where data_kind is a string that is either equal to 'plot' or 'integrate'. It returns a structure ref_data with the following fields:

ref_data

├ deg	$[1, 1]$ / $[1,]$
├ reference_element	$[1, 2]$ / $[2,]$
├ evaluation_points	$[q, 1]$ / $[q,]$
├ quadrature_weights	$[q, 1]$ / $[q,]$
├ reference_basis	$[p + 1, q]$ / $[p + 1, q]$
└ reference_basis_derivatives	$[p + 1, q]$ / $[p + 1, q]$

where deg $= p$; reference_element $= [0, 1]$, the i-th entry of evaluation_points contains the i-th evaluation point $\tilde{\xi}_i \in [0, 1]$ where the reference basis functions have been evaluated; the i-th entry of quadrature_weights contains the i-th quadrature weight if integrate_flag is True (else it is empty); the (i, j)-th entry of reference_basis contains the i-th reference basis function, $\tilde{N}_i$, evaluated at the j-th evaluation point (i.e., $\tilde{N}_i(\tilde{\xi}_j)$); and the (i, j)-th entry of reference_basis_derivatives contains the derivative of the i-th reference basis function $\tilde{N}_i$ evaluated at the j-th evaluation point (i.e., $d\tilde{N}_i(\tilde{\xi}_j)/d\xi$).

Remark: All the above quantities are defined on the reference element $\tilde{\Omega} = [0, 1]$. In particular, you are provided a special reference basis that forms a *partition of unity*, i.e., for any $\xi \in \tilde{\Omega}$,

$$\sum_{i=1}^{p+1} \tilde{N}_i(\xi) = 1 \ .$$

Remark: When the input argument integrate is equal to True, the reference basis functions are computed at the quadrature points for a n_q-point Gauss-Legendre quadrature scheme. Otherwise, the reference basis functions are computed on n_q points uniformly distributed on the reference element.

Mesh

You **need to implement** a function:

```
function create_fe_mesh($[x_0,\ x_1, \ldots,\ x_m]$)
    ⋮
return mesh
```

which returns a structure mesh with the following fields:

mesh

├ m	$[1, 1]$ / $[1,]$

└ elements	$[m, 2]$ / $[2, m]$

where nel is equal to the number of elements and each row/column of elements contains the boundary points of Ω_i in increasing order.

Reference element to mesh element maps

You **need to implement** a function:

```
function create_parametric_map(mesh)
  ⋮
return param_map
```

which returns a structure param_map with the following fields:

param_map

├ map	
├ map_derivatives	$[m, 1]$ / $[m,]$
└ imap_derivatives	$[m, 1]$ / $[m,]$

where map is a function that accepts arguments (ξ, x_{i-1}, x_i) and returns $x_{i-1} + \xi(x_i - x_{i-1})$. Note that this is the linear map from the reference element $\tilde{\Omega}$ to $\overline{\Omega}_i$,

$$\phi_i \ : \ \tilde{\Omega} \ni \xi \mapsto \phi_i(\xi) := x_{i-1} + \xi(x_i - x_{i-1}) \ .$$

The i-th entry of map_derivatives contains a constant which is equal to $\partial\phi_i/\partial\xi$, and i-th entry of imap_derivatives contains a constant which is equal to $\partial\phi_i^{-1}/\partial x$, where ϕ_i^{-1} denotes the inverse map from $\overline{\Omega}_i$ to $\tilde{\Omega}$.

Finite element space

You **are given** a function:

```
function create_fe_space(p, k, mesh)
  ⋮
return space
```

which returns a structure space with the following fields:

space

├ n	$[1, 1]$ / $[1,]$
├ supported_bases	$[m, p+1]$ / $[m, p+1]$
└ extraction_coefficients	$[p+1, p+1, m]$ / $[m, p+1, p+1]$

where the i-th row of support_bases contains the indices of the $(p+1)$ finite element basis functions that are non-zero on Ω_i, and the i-th slice of `extraction_coefficients` is a square matrix $[e_{jki}]_{j,k=1}^{p+1}$ containing coefficients $e_{jki} \in \mathbb{R}$.

In other words, if the the i-th row of `element_in_support` is equal to $[j_1, j_2, \dots, j_{p+1}]$, then we have

$$N_{j_s}\big|_{\Omega_i}(x) = \sum_{k=1}^{p+1} e_{ski}\tilde{N}_{k,i}(x) := \sum_{k=1}^{p+1} e_{ski}\tilde{N}_k(\phi_i^{-1}(x)) \ , \quad s = 1, \dots, p+1 \ . \tag{1}$$

Note that you are provided with a special choice of finite element basis functions such that $\sum_{j=1}^{p+1} e_{jki} = 1$ for any $i = 1, \dots, m$, and for any $k = 1, \dots, p+1$.

Finally, observe that for a finite element function $u_h = \sum_{i=1}^{n} u_i N_i$ for $u_i \in \mathbb{R}$, you can write

$$u_h\big|_{\Omega_i} = \sum_{s=1}^{p+1} u_{j_s} N_{j_s}\big|_{\Omega_i} \ .$$

For the rest of this assignment, assume that the problem domain is $\Omega = [0, 2]$, i.e., $L = 2$, but make sure your implementation is more general so that we can utilize the same code for a future assignment where L may take on a different value.

(2 points) **The finite element space**
For this section, use the data provided in `fe_space.zip`.
Remark: Note that the provide plots are made by evaluating the functions at 20 uniformly spaced points on each Ω_i (including the boundary points of Ω_i).

A. (1 point) Provide pseudo-codes for your implementations of create_mesh and create_parametric_map. *In your pseudocode, reference the structures and fields as defined in the assignment.*

~~B~~. (0 points: Do not show these plots, this task is here so that you can verify that your implementation is behaving correctly.)
Choose $m = 4$, $p = 1$, $k = 0$, $x_i = iL/m$. Plot the values and first derivatives of the first, fourth and p-th finite element basis functions on Ω, and verify that they correspond to the provided plots in `0.png`.

~~C~~. (0 points: Do not show these plots, this task is here so that you can verify that your implementation is behaving correctly.)
Choose $m = 4$, $p = 1$, $k = 0$, $x_i = i^2L/m^2$. Plot the values and first derivatives of the first, fourth and p-th finite element basis functions on Ω, and verify that they correspond to the provided plots in `1.png`.

~~D~~. (0 points: Do not show these plots, this task is here so that you can verify that your implementation is behaving correctly.)
Choose $m = 4$, $p = 2$, $k = 0$, $x_i = iL/m$. Plot the values and first derivatives of the first, fourth and p-th finite element basis functions on Ω, and verify that they correspond to the provided plots in `2.png`.

~~E~~. (0 points: Do not show these plots, this task is here so that you can verify that your implementation is behaving correctly.)
Choose $m = 4$, $p = 2$, $k = 0$, $x_i = i^2L/m^2$. Plot the values and first derivatives of the first, fourth and p-th finite element basis functions on Ω, and verify that they correspond to the provided plots in `3.png`.

~~F~~. (0.5 point) Choose $m = 5$, $p = 2$, $k = 1$, $x_i = i^2L/m^2$. Use the provided coefficients u_i in the file `coefficients.txt` and plot the finite element function u_h and its derivative $u_{h,x}$ on Ω,

$$u_h = \sum_{i=1}^{n} u_i N_i \ .$$

G. (0.5 point) Show that, regardless of the choices of p, k and Δ,

$$\sum_{i=1}^{n} N_i(x) = 1 \ ,$$

for any $x \in \Omega$. That is, our finite element basis functions form a *partition of unity.*

Weak problem

Consider a weak problem of interest: find $u_h \in \mathcal{S}_h$ such that

$$B(u_h, w_h) = L(w_h) \ , \quad \forall w_h \in \mathcal{W}_h \ ,$$

where B and L are appropriately defined bilinear and linear forms, respectively, and where essential boundary conditions are imposed on u_h,

$$\mathcal{S}_h := \{ f \in \mathcal{F}(p,k;\Delta) \mid f(0) = g_0 \ , \ f(L) = g_L \} \ ,$$
$$\mathcal{W}_h := \{ f \in \mathcal{F}(p,k;\Delta) \mid f(0) = 0 \ , \ f(L) = 0 \} \ .$$

Finite element assembly routine

Let the finite element solution be $u_h = \sum_{i=1}^{n} u_i N_i \in \mathcal{S}_h$, where $u_1 = g_0$ and $u_n = g_L$[1]. Then, the purpose of the finite element problem is to determine the rest of the coefficients u_i, $i = 2, \ldots, n-1$. Therefore, the purpose of the finite element assembly routine is to assemble the following matrix problem,

$$\underbrace{\begin{bmatrix} B(N_2,N_2) & B(N_2,N_3) & \cdots & B(N_2,N_{n-1}) \\ B(N_3,N_2) & B(N_3,N_3) & \cdots & B(N_3,N_{n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ B(N_{n-1},N_2) & B(N_{n-1},N_3) & \cdots & B(N_{n-1},N_{n-1}) \end{bmatrix}}_{=:\mathrm{A}} \begin{bmatrix} u_2 \\ u_3 \\ \vdots \\ u_{n-1} \end{bmatrix} = \underbrace{\begin{bmatrix} L(N_2) - g_0 B(N_2,N_1) - g_L B(N_2,N_n) \\ L(N_3) - g_0 B(N_3,N_1) - g_L B(N_3,N_n) \\ \vdots \\ L(N_{n-1}) - g_0 B(N_{n-1},N_1) - g_L B(N_{n-1},N_n) \end{bmatrix}}_{=:\mathrm{b}}$$

You **need to implement** a function:

```
function assemble_fe_problem(mesh, space, ref_data, param_map, problem_B, problem_L, [$g_0, g_L$])
    ⋮
return A, b
```

where, given B and L, `problem_B` and `problem_L` are anonymous functions that (you **need to implement** these) such that

$$B(u_h, w_h) = \int_\Omega \texttt{problem\_B}(x, u_h(x), u_{h,x}(x), w_h(x), w_{h,x}(x)) \, dx \ ,$$
$$L(w_h) = \int_\Omega \texttt{problem\_L}(x, w_h(x)) \, dx \ .$$

Remark: Follow the implementation introduced in class.

(12 points) **The finite element problem**
For this section, use the data provided in `fe_problem.zip`.

H. (4 points) Provide pseudo-codes for your implementations of assemble_fe_problem. *In your pseudocode, reference the structures and fields as defined in the assignment.*

~~I.~~ (0 points: Do not show these results, this task is here so that you can verify that your implementation is behaving correctly.)
Choose $m = 4$, $p = 1$, $k = 0$, $n_q = 3$, $x_i = iL/m$, $g_0 = 0$, $g_L = 1$, $f = 1$, and

$$B(u_h, w_h) := \int_\Omega w_{h,x} u_{h,x} \, dx \ , \quad L(w_h) := \int_\Omega f w_h \, dx \ .$$

Verify that the computed A and b match the ones provided to you in `0.zip`.

[1] You can do this because the finite element functions provided to you are such that $N_1(0) = 1$, $N_n(L) = 1$, and $N_i(0) = N_i(L) = 0$ for $i = 2, \ldots, n-1$

J. (0 points: Do not show these results, this task is here so that you can verify that your implementation is behaving correctly.)
Choose $m = 4$, $p = 2$, $k = 0$, $n_q = 3$, $x_i = i^2 L/m^2$, $g_0 = 0$, $g_L = 1$, $f = 1$, and

$$B(u_h, w_h) := \int_\Omega w_{h,x} u_{h,x} \, dx \;, \quad L(w_h) := \int_\Omega f w_h \, dx \;.$$

Verify that the computed A and b match the ones provided to you in `1.zip`.

K. (1 point) Consider the matrix

$$\tilde{\mathrm{A}} = \begin{bmatrix} B(N_1,N_1) & B(N_1,N_2) & \cdots & B(N_1,N_n) \\ B(N_2,N_1) & B(N_2,N_2) & \cdots & B(N_2,N_n) \\ \vdots & \vdots & \ddots & \vdots \\ B(N_n,N_1) & B(N_n,N_2) & \cdots & B(N_n,N_n) \end{bmatrix} .$$

Show that the sum of entries in any column or any row of $\tilde{\mathrm{A}}$ must be equal to 0 for $B(u_h, w_h) := \int_\Omega u_{h,x} w_{h,x} \, dx$.
Remark: If you need it, you can use this fact to debug your assembly implementation, all you need is to run it for two additional rows and columns!

L. (0.5 point) Consider a bilinear form $B(u_h, w_h) := \int_\Omega u_{h,x} w_{h,x} \, dx$. Show that the corresponding A must be a symmetric matrix.
Remark: If you need it, you can use this fact to debug your assembly implementation!

M. (1.5 points) Consider a bilinear form $B(u_h, w_h) := \int_\Omega u_{h,x} w_{h,x} \, dx$ and consider the corresponding A. Show that the nullspace of A is trivial, i.e., only the zero vector is in the nullspace of A.
Remark: You can use this fact to debug your assembly implementation!

N. (2.5 points) Choose $m = 4$, $p = 2$, $k = 0$, $n_q = 3$, $x_i = iL/m$, $g_0 = 0$, $g_L = 1$, $f(x) = \pi^2 \sin(\pi x)$, and

$$B(u_h, w_h) := \int_\Omega w_{h,x} u_{h,x} \, dx \;, \quad L(w_h) := \int_\Omega f w_h \, dx \;.$$

- (1.5 points) Compute A, b and save them in text files `A.txt`, `b.txt` (use the same format as the data you are provided with). Submit the files in a zip file named `0.zip`.
- (1 point) Compute the finite element solution and plot its values and (piecewise-)derivatives in your report.

O. (2.5 points) Choose $m = 10$, $p = 2$, $k = 1$, $n_q = 3$, $x_i = iL/m$, $g_0 = 0$, $g_L = 1$, $f(x) = \pi^2 \sin(\pi^2 x) f_D(x)$ where

$$f_D(x) := \begin{cases} +1 \;, & x < 1 \;, \\ -1 \;, & x \geq 1 \;, \end{cases}$$

$$B(u_h, w_h) := \int_\Omega (1 - 0.4\cos(\pi x)) w_{h,x} u_{h,x} + w_h u_h \, dx \;, \quad L(w_h) := \int_\Omega f w_h \, dx \;.$$

- (1.5 points) Compute A, b and save them in text files `A.txt`, `b.txt` (use the same format as the data you are provided with). Submit the files in a zip file named `1.zip`.
- (1 point) Compute the finite element solution and plot its values and derivatives in your report.