

1 File description

Download the Matlab files from http://rezaabedi.com/wp-content/uploads/Courses/DG/DG_1D_PDEs.zip. The Matlab files and what parts are to be completed for the 3 series of assignments are:

- Solving config files(s).
 - scriptSolvePDE solves 1 config file (used for early stages of code development and observing the solution for 1 element size set-up and model parameters).
 - MAIN_PDEComparison: Solves one problem with a series of model parameters (*e.g.*, ϵ, σ , *etc.*) and element sizes (domain size / 1, /2, /4, *etc.*).

The file (function) Solve1D_PDE.m is internally used in these two files.

- Problem loads and exact solution:
 - PDE1DBCData.m: In the input config file the entry flags is followed by two entries. The first one is the left edge flag and second one the right one. The values are flags are:
 - * 0: Essential BC (*e.g.*, T for thermal equation and \mathbf{v} for elastodynamics).
 - * 1: Natural BC (*e.g.*, $\mathbf{q}\cdot\mathbf{n}$ for thermal equation and traction $\mathbf{s}\cdot\mathbf{n}$ for elastodynamics).
 - * 2: Periodic boundary. The flag must be used for left and right boundaries.
 Examples are flags 0 0 for two sides essential, 0 1 for left essential / right natural (*e.g.*, thermal problem done in class), 2 2 for a periodic boundary condition. Examples can be found in the config files.
 - PDE1DSourceData.m: A vector of coordinates (xs) is sent, and source term (Q) is computed for these coordinates. For all problems considered Q is zero, except for config_DG.Sine where for a domain $x \in [0, 1]$, $Q = \sin(\pi x)$ is applied for the elliptic problem (3a) and boundary conditions $u(0) = u(1) = 0$.
 - PDE1DICData.m: Specifies initial conditions $u_0(x) = u(x, t = 0)$ for parabolic, and $u_0(x), \dot{u}_0(x) = \dot{u}(x, t = 0)$ for hyperbolic PDE: [T0, Tdot0] = PDE1DICData(xs, loadCase, PDEtype).
 - PDEgetExactSolution.m: Exact solution (value u , time derivative \dot{u} (and $\kappa u, q$)) are obtained for the problems considered. The functions in getDelError.m and l2ErrorDG.m are internally computed in the code to compute the L2 error of the DG solution with respect to the exact solution,

$$L_2(u) = \sqrt{\int_0^L (u^h(x) - u(x))^2 dx} \quad (1)$$

- PDE1DFEM.m: This is the class for reading, solving, and reporting the solution for 1D elliptic, parabolic, and hyperbolic PDEs. All the functions in this class are implemented. Some parts are intentionally deleted, so that as a part of the assignment the students will complete these functions. The list of incomplete functions is:

1. ComputedG_KM: function for computing stiffness (conductivity) and mass (capacity) matrices. For hyperbolic PDEs, damping matrix \mathbf{C} is also assembled. The latter is only assembled for parabolic and hyperbolic PDEs. In this function, stiffness (and damping for hyperbolic PDE) are computed from the following contributions:
 - (a) Interior of element matrices k_i^e (and c_i^e): need to be computed. Depending on the PDEtype (explained below), some parts are intentionally incomplete.

- (b) Interior interface matrices $k_r^{e^-e^+}$ (and $c_r^{e^-e^+}$): The stiffness (and damping) at interior interfaces of the domain Γ are computed for the interface between elements e^- and e^+ .
- (c) Essential boundary condition k_u^e (and c_u^e): The stiffness (and damping) at essential boundary conditions are computed.

In addition Mass matrix (capacity for thermal parabolic equation) is computed in this function. Since, this involves an integral only inside the element, only m_i^e needs to be computed and assembled. This matrix has the same form for parabolic and hyperbolic PDEs (not applicable to elliptic ones).

Also at the end of this function, eigenvalues and eigenvectors of \mathbf{K} are computed after `if (obj.computeKEigen)` statement. Do not remove this part. Finally after `% matrices needed for time marching of parabolic and hyperbolic` PDE various matrices needed in time marching of parabolic and hyperbolic PDEs are computed. Again, do not remove these calculations. You can use them under item 3 (time marching).

2. `ComputeDG_F`: This function computes the force vector on the right hand side of the semi-discrete matrix equation. The sources of \mathbf{F} are:
 - (a) F_r : The force from the source terms. This part is not removed from the function implementation and is assembled from the element level contributions.
 - (b) F_u and F_f : Forces from essential and natural boundary conditions have different forms for elliptic, parabolic, and hyperbolic PDEs and left ($x = 0$) and right ($x = L$) boundaries. Certain implementations are remained as samples.
3. System solution `ComputeDG_Sln`: This function has three modes for elliptic, parabolic, and hyperbolic PDEs. They require the solution to,

$$\mathbf{KA} = \mathbf{F} \quad \text{Elliptic PDE} \quad \text{ComputeDG_Sln_Elliptic} \quad (2a)$$

$$\mathbf{M}\dot{\mathbf{A}} + \mathbf{KA} = \mathbf{F} \quad \text{Parabolic PDE} \quad \text{ComputeDG_Sln_Parabolic} \quad (2b)$$

$$\mathbf{M}\ddot{\mathbf{A}} + \mathbf{C}\dot{\mathbf{A}} + \mathbf{KA} = \mathbf{F} \quad \text{Hyperbolic PDE} \quad \text{ComputeDG_Sln_Hyperbolic} \quad (2c)$$

The solution to the elliptic problem (2a) is already complete. But through these series of assignments, you'll complete the implementation for forward Euler time marching schemes in for (2b) and (2c).

2 PDE description

In 3 series of assignments the following PDEs are solved for 1D spatial PDEs:

$$-\nabla \cdot \kappa \nabla u = Q \quad \Rightarrow \quad -(\kappa u, x), x = Q \text{ in 1D} \quad \text{Elliptic (PDEtype = 0)} \quad (3a)$$

$$c\dot{u} - \nabla \cdot \kappa \nabla u = Q \quad \Rightarrow \quad c\dot{u} - (\kappa u, x), x = Q \text{ in 1D} \quad \text{Parabolic (PDEtype = 1)} \quad (3b)$$

$$c\ddot{u} + d\dot{u} - \nabla \cdot \kappa \nabla u = Q \quad \Rightarrow \quad c\ddot{u} + d\dot{u} - (\kappa u, x), x = Q \text{ in 1D} \quad \text{Hyperbolic (PDEtype = 2)} \quad (3c)$$

where κ plays the role of conductivity in (3b) and (3a) if thermal problem is considered or similarly elastic stiffness in (3c) and (3a) if elastodynamics is considered. Parameter c is volumetric thermal capacity in (3b) and mass density in (3c) for the sample problems discussed. Finally, d in (3c) is the damping coefficient for elastodynamics. Clearly, these coefficients have similar meaning and effect for other 1D PDEs.

3 Config file and problem description

The following four config files are provided for the test problems considered. The first entry in these files is `PDEtype`, which as described in (3), values 0, 1, and 2 correspond to elliptic, parabolic, and hyperbolic

PDEs, respectively. These PDEs solve the problems (3) for $x \in [0, L]$ for $Q = 0, c = 1, d = 0, \kappa = 1$, unless otherwise provides below.

The **loadCase** is a number in these files that defines IC, BC, source term (Q), and the exact solution.

The list of config files is:

1. **loadCase = 0**, config_DG_LeftEssential0_RightNatural1.txt: Solves PDEs (3a) for $u(x = 0) = 0$ and $-\kappa u, x(x = L) = 1$ (BC flags [0 1], BC values [0 1]). Domain length is $L = 1$. The exact solutions are $u(x) = -x$ and $u(x, t) = \min(1 - x - t, 0)$ for PDEmode = 0 and 2, respectively; cf. PDEgetExactSolution.m.
2. **loadCase = 1**, config_DG_Sine.txt: Solves the elliptic problem (3a) for $L = 1, Q(x) = \sin(\pi x)$, and homogeneous essential boundary condition on both ends (flags 0 0 and BC values data = [0 0] in PDE1DBCData.m. The exact solution for this problem is $u(x) = 1/\pi^2 \sin(\pi x)$.
3. **loadCase = 3**, config_DG_PeriodicSine.txt: Solves a PDE on a domain with $L = 2\pi$ and initial conditions $u_0(x) = u(x, 0) = \sin(x)$ (and $\dot{u}_0(x) = \dot{u}(x, 0) = 0$ for PDEtype == 2) for transient PDEs (PDEtype $\neq 0$). Periodic boundary conditions are used, that is $u(L) = u(0)$. This corresponds to flags 2 2 in the config file. The exact solutions are $u(x, t) = e^{-t} \sin(x)$ (PDEtype == 1) and $u(x, t) = \cos(t) \sin(x)$ (PDEtype == 2).
4. **loadCase = 4**, config_DG_Hyperbolic2regions.txt: This problem is set up for a hyperbolic PDE. Its exact solution is provided in the function [u, uDot, ux, CuDot, kappaux] = getStepWaveExactSolution. It corresponds to the problem with left and right material properties, $\kappa_l = 1, c_l = 1, \kappa_r = 0.1$, and $c_r = 1$, zero damping and material interface at $x = 0.5$. A natural boundary condition of $-\kappa u_x(x = 0) = -1$ and essential boundary $u(x = L = 1) = 0$ are used at the two ends of the domain. The propagation of the wave from the left boundary and reflection / transmission at the material interface makes this a challenging test case. Same problem is solved for 1D electromagnetics problem in “R. Abedi, S. Mudaliar. An Asynchronous Spacetime Discontinuous Galerkin Finite Element Method for Time Domain Electromagnetics, Journal of Computational Physics, , 351:121144, 2017.” where κ and c are replaced by inverse of electric permittivity and magnetic permeability.

Some specific entries of the config files are:

- **PDEtype**: As mentioned options 0, 1, and 2 correspond to elliptic, parabolic, and hyperbolic PDEs.
- **eps, sigma, beta, gamma** ($\epsilon, \sigma, \beta, \gamma$): Options for the star values for elliptic and parabolic PDEs as described in the class. For the assignments β and γ will be zero.
- **hyperStartOption** (0 average flux, 1 Riemann flux) and **hyper1FLambdaScalingOn** are options for hyperbolic PDEs. The value of one for the latter means all terms in the weighted residual are dimensionally consistent as this is discussed in class through parameter λ .
- Domain geometry, material properties, and number of elements within each region are provided in next lines. **numberOfElements** has number of elements in each region, and **LsPerRegion, CsPerRegion, kappasPerRegion**, and **dampingsPerRegion** have the values of length, c , κ , and d in each of these regions; cf. (3).
- **loadCase**: As described in config files above, this number is used in IC/BC/source term/exact solution functions to extract the desired values for a given problem.
- **flags**: two integers provide boundary condition types for the left and right (first and second entries) boundaries. 0, 1, and 2 correspond to essential, natural, and periodic boundaries as described above.

- **finalTime**: For transient PDEs (parabolic and hyperbolic) it provides the final time.
- **stabilityDeltFactor**: For transient PDEs, this is a number that is multiplied by base stability time step as described in the class.

4 Comparison of various options for star values

The function “function MAIN_PDEComparison(configName, resMin, resMax) ” can be run to compare the effect of parameters of star option (ϵ, σ for elliptic / parabolic; Average vs. Riemann, and λ choice for hyperbolic PDEs). The element size goes from $2^{-\text{resMin}}$ to $2^{-\text{resMax}}$ with powers of 2 increase in each region of the domain. For the assignment, you need to run this function with the config file name provided.

Elliptic PDEs:

Homework 3: Part 1:

Complete `ComputeDG_KM` and `ComputeDG_F` functions in `PDE1DFEM` class. Then run “`config_DG_Sine.txt`” and send me your `PDE1DFEM.m` file along with two generated files `config_DG_Sine.u.png` and `config_DG_Sine.DuDx.png` and the output file `config_DG_Sine.out` after `scriptSolvePDE.m` is ran with `configNameWOExt = 'config_DG_Sine';` is chosen as the active config file (last entry). Note, no other file must be changed.

Homework 3: Part 2 (Be **brief** in your answers. Try to answer all questions in less than two pages (but be specific and to the point answering the question asked)).

Run `MAIN_PDEComparison.m` with `config_DG_LeftEssential0_RightNatural1.txt` and `config_DG_Sine.txt` as last entries each time. These runs are simulated for $\epsilon = -1, 0, 1$ and $\sigma = 0$ or 1 for each of the three ϵ options (a total of 6 cases). Based on your observations on the results of the two config files answer the following questions:

1. Among the choices $\epsilon = 0, -1, 1$ which one does not provide a result no matter what for even $\sigma = 1$?
2. For $\epsilon = 1, \sigma = 0$ comment on which config files provide results (nonsingular matrix) and which ones do not. Also, if for any of the config files stiffness is singular (observed with getting no results in plots for DG curves), does changing $\sigma = 1$ eliminate this problem?
3. Briefly (less than 2 sentence–3 lines) describe the solutions for $\epsilon = -1$ with $\sigma = 0$ and $\sigma = 1$.
4. Referring to `log2_L2_error.png` for the results generated by `config_DG_Sine.txt`, provide the order of convergence of $\epsilon = 1, \sigma = 1$.
5. Relate the choices $\epsilon = -1$ ($\sigma = 0$ or $\sigma \neq 0$), $\sigma = 1$ ($\sigma = 0$ or $\sigma \neq 0$) and $\epsilon = 0$ to the methods described on page 5 of the book “Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations”, by Riviere. Be specific in terms of the names of these interior penalty methods (if no specific name, mention author and name). Relate the properties discussed on that page to behaviors observed in these runs (note that only elliptic problems are considered in this homework assignment).
6. Show that $k_{ij}^{e_- e_+} = k_{ji}^{e_+ e_-}$ arising from the global weak statement at an interface between elements e_- and e_+ on interior boundaries in the domain Γ . In the class, we derived this expression for $k_{ij}^{e_- e_+}$. Discuss the effect of the choice of $\epsilon = 0, -1, 1$ on the symmetry or skew-symmetry of this matrix.
7. Coercivity Q1: Discuss for each choice of epsilon $\epsilon = -1, 1, 0$ all the eigenvalues of global stiffness K will be nonnegative ($B(u^h, u^h) \geq 0$).
8. Coercivity Q2: Refer to `max_Re_lambda.png` generated by running `config_DG_LeftEssential0_RightNatural1.txt` and comment on the minimum eigenvalue of global stiffness K .