Assignment

Math3101/5305 Term 3 2020

Due: 5PM Monday 23 November

Instructions. Upload your source files and a pdf with your written answers to Moodle. You will need to use matrix assembly techniques from Labs 2–4, but in the simpler case of piecewise-linear (rather than piecewise-quadratic) finite elements, and with no fixed nodes. Also, the radial geometry leads to a factor ρ^2 in the integrands that define the entries of the stiffness matrix, mass matrix and load vector.

As a rough guide, my typed solutions for the written questions (Q1 and Q11) come to 2 pages, the code for my functions totals about 160 lines, and the code for my scripts totals about 200 lines. Comments should be minimal, since you may assume that anyone reading your code has already studied the text of the assignment.

Marking scheme. Marks will be awarded as follows, giving a total of 55 for Math3101, and 75 for Math5305.

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
14	6	3	4	2	8	4	6	8	4	8	8

1 Dimensional reduction of a 3D heat equation

Suppose that the region $\Omega \subseteq \mathbb{R}^3$ is a body with thermal conductivity κ, so that the temperature $u = u(\vec{x}, t)$ in the region satisfies the heat equation,

$$u_t - \nabla \cdot \big(\kappa(\vec{x})\nabla u\big) = f(\vec{x}, t) \quad \text{for } \vec{x} \in \Omega \text{ and } 0 \le t \le T, \tag{1}$$

where as usual $f(\vec{x}, t)$ is the density of heat sources. The body is initially at thermal equilibrium with its environment at a constant temperature u_0, implying the initial condition

$$u = u_0 \quad \text{for } \vec{x} \in \Omega \text{ when } t = 0. \tag{2}$$

We also assume a boundary condition of the form

$$-\vec{n} \cdot \big(\kappa(\vec{x})\nabla u\big) = \alpha(u - u_0) \quad \text{for } \vec{x} \in \partial\Omega \text{ and } 0 \le t \le T, \tag{3}$$

where $\vec{n}$ denotes the outward unit normal to Ω and α is a positive constant. In words: the normal component of the heat flux leaving Ω is proportional to the temperature difference $u - u_0$ (so heat leaves more quickly the hotter Ω is compared to its environment). The weak form of the initial-boundary value problem (1)–(3) is

$$\int_\Omega \big(u_t v + \kappa \nabla u \cdot \nabla v\big) + \int_{\partial\Omega} \alpha u v = \int_\Omega f v + \int_{\partial\Omega} \alpha u_0 v \tag{4}$$

for all test functions v, with $u = u_0$ when $t = 0$. Also, the average temperature of the ball is given by

$$\bar{u}(t) = \frac{1}{\mathrm{Vol}(\Omega)} \int_\Omega u(\cdot, t) \quad \text{for } 0 \le t \le T. \tag{5}$$

We will assume for simplicity that Ω is a ball with radius R, centered at the origin, and that the thermal conductivity κ and the heat source f are radially symmetric. Thus, $\kappa = \kappa(\rho)$ and $f = f(\rho, t)$ depend on $\vec{x}$ only through the radial variable $\rho = |\vec{x}|$. It follows that the temperature $u = u(\rho, t)$ is also radially symmetric. For any radially symmetric function $g = g(\rho)$, we have

$$\nabla g = g'(\rho)\,\vec{e}_\rho, \qquad \int_\Omega g = \int_0^R g(\rho) 4\pi\rho^2\,d\rho, \qquad \int_{\partial\Omega} g = 4\pi R^2 g(R),$$

where $\vec{e}_\rho = \rho^{-1}\vec{x}$ is the unit vector in the radial direction. Hence, the weak formulation (4) simplifies to

$$\int_0^R \big(u_t v + \kappa u_\rho v_\rho\big)\rho^2\,d\rho + \alpha R^2 u(R,t)v(R) = \int_0^R f v \rho^2\,d\rho + \alpha R^2 u_0 v(R) \tag{6}$$

for all (radially symmetric) test functions $v = v(\rho)$. The formula for the average temperature (5) simplifies to

$$\bar{u}(t) = \frac{3}{R^3}\int_0^R u(\rho,t)\rho^2\,d\rho. \tag{7}$$

2 Dimensionless variables

To reduce the number of parameters in the problem, we introduce dimensionless[1] quantities

$$\rho_* = \frac{\rho}{R}, \qquad t_* = \frac{\alpha}{R}t, \qquad u_*(\rho_*, t_*) = \frac{u(\rho,t)}{u_0}, \qquad v_*(\rho_*, t_*) = \frac{v(\rho,t)}{u_0},$$
$$\kappa_*(\rho_*) = \frac{\kappa(\rho)}{\alpha R}, \qquad f_*(\rho_*, t_*) = \frac{R}{\alpha u_0} f(\rho,t), \qquad T_* = \frac{\alpha}{R}T, \qquad \bar{u}_*(t_*) = \frac{\bar{u}(t)}{u_0}.$$

It can be shown that (6) implies

$$\int_0^1 \big((u_*)_{t_*} v_* + \kappa_* (u_*)_{\rho_*}(v_*)_{\rho_*}\big)(\rho_*)^2\,d\rho_* + u_*(1,t_*)v_*(1) = \int_0^1 f_* v_* (\rho_*)^2\,d\rho_* + v_*(1) \tag{8}$$

for $0 \le t \le T_*$ and all $v_*(\rho_*) = v(\rho)/u_0$, and that (7) implies

$$\bar{u}_*(t_*) = 3\int_0^1 u_*(\rho_*, t_*)(\rho_*)^2\,d\rho_* \quad \text{for } 0 \le t_* \le T_*. \tag{9}$$

As is customary, we drop the $*$ subscripts, writing

$$\int_0^1 \big(u_t v + \kappa u_\rho v_\rho\big)\rho^2\,d\rho + u(1,t)v(1) = \int_0^1 f v \rho^2\,d\rho + v(1) \quad \text{for } 0 \le t \le T, \tag{10}$$

and

$$\bar{u}(t) = 3\int_0^1 u(\rho,t)\rho^2\,d\rho \quad \text{for } 0 \le t \le T. \tag{11}$$

[1]Here, "dimensionless" means "without physical units". The problem still has one space dimension and one time dimension.

3 Discretisation in space and time

Let V_h denote the space of continuous, piecewise-linear functions with respect to a chosen grid

$$0 = \rho_0 < \rho_1 < \cdots < \rho_P = 1.$$

Since the boundary conditions do not fix any values of the solution, the trial set and test space both coincide with V_h, so the finite element solution $u_h : [0, T] \to S_h = V_h$ is given by

$$\int_0^1 \big((u_h)_t v + \kappa (u_h)_\rho v_\rho\big)\rho^2\,d\rho + u_h(1,t)v(1) = \int_0^1 f v \rho^2\,d\rho + v(1) \quad \text{for all } v \in T_h = V_h. \tag{12}$$

Our numerical approximation to the average temperature is then defined in the obvious way:

$$\bar{u}_h(t) = 3\int_0^1 u_h(\rho,t)\rho^2\,d\rho \quad \text{for } 0 \le t \le T.$$

Define the linear shape functions for the pth element $[\rho_{p-1}, \rho_p]$ by

$$\psi_1^{\langle p\rangle}(\rho) = \frac{\rho_p - \rho}{h_p} \quad \text{and} \quad \psi_2^{\langle p\rangle}(\rho) = \frac{\rho - \rho_{p-1}}{h_p} \quad \text{for } \rho_{p-1} \le \rho \le \rho_p \text{ and } 1 \le p \le P,$$

where $h_p = \rho_p - \rho_{p-1}$. The 2×2 element stiffness matrix is

$$\boldsymbol{A}^{\langle p\rangle} = \begin{bmatrix} a_{11}^{\langle p\rangle} & a_{12}^{\langle p\rangle} \\ a_{21}^{\langle p\rangle} & a_{22}^{\langle p\rangle} \end{bmatrix} \quad \text{where} \quad a_{jk}^{\langle p\rangle} = \int_{\rho_{p-1}}^{\rho_p} \kappa(\rho)\frac{d\psi_k^{\langle p\rangle}}{d\rho}\frac{d\psi_j^{\langle p\rangle}}{d\rho}\,\rho^2\,d\rho, \tag{13}$$

and the 2×2 element mass matrix is

$$\boldsymbol{M}^{\langle p\rangle} = \begin{bmatrix} m_{11}^{\langle p\rangle} & m_{12}^{\langle p\rangle} \\ m_{21}^{\langle p\rangle} & m_{22}^{\langle p\rangle} \end{bmatrix} \quad \text{where} \quad m_{jk}^{\langle p\rangle} = \int_{\rho_{p-1}}^{\rho_p} \psi_k^{\langle p\rangle}(\rho)\,\psi_j^{\langle p\rangle}(\rho)\,\rho^2\,d\rho. \tag{14}$$

Likewise, the element load vector is

$$\boldsymbol{f}^{\langle p\rangle}(t) = \begin{bmatrix} f_1^{\langle p\rangle}(t) \\ f_2^{\langle p\rangle}(t) \end{bmatrix} \quad \text{where} \quad f_j^{\langle p\rangle}(t) = \int_{\rho_{p-1}}^{\rho_p} f(\rho,t)\psi_j^{\langle p\rangle}(\rho)\,\rho^2\,d\rho. \tag{15}$$

To evaluate these integrals, we introduce a quadrature rule on the reference element $[0, 1]$,

$$\int_0^1 g(y)\,d\rho \approx \sum_{\ell=1}^{J} w_\ell g(y_\ell), \tag{16}$$

so that, via the substitution $\rho = \rho_{p-1} + y h_p$, the corresponding quadrature rule on the pth element is

$$\int_{\rho_{p-1}}^{\rho_p} g(\rho)\,d\rho \approx h_p \sum_{\ell=1}^{J} w_\ell g(\rho_\ell^{\langle p\rangle}) \quad \text{where} \quad \rho_\ell^{\langle p\rangle} = \rho_{p-1} + y_\ell h_p.$$

Thus,

$$a_{jk}^{\langle p\rangle} \approx \frac{(-1)^{j+k}}{h_p}\sum_{\ell=1}^{J} w_\ell \kappa(\rho_\ell^{\langle p\rangle})(\rho_\ell^{\langle p\rangle})^2 \quad \text{and} \quad m_{jk}^{\langle p\rangle} \approx h_p \sum_{\ell=1}^{J} w_\ell \psi_k^{\langle p\rangle}(\rho_\ell^{\langle p\rangle})\psi_j^{\langle p\rangle}(\rho_\ell^{\langle p\rangle})(\rho_\ell^{\langle p\rangle})^2,$$

with

$$f_j^{\langle p\rangle}(t) \approx h_p \sum_{\ell=1}^{J} w_\ell f(\rho_\ell^{\langle p\rangle}, t)\psi_j^{\langle p\rangle}(\rho)(\rho_\ell^{\langle p\rangle})^2.$$

Since there are no fixed nodes, the obvious global numbering of the nodes is

$$\mathsf{n}_p = \rho_{p-1} \quad \text{for } 1 \le p \le P+1,$$

giving the $2 \times P$ connectivity matrix

$$\boldsymbol{C} = [C_{jp}] = \begin{bmatrix} 1 & 2 & 3 & \cdots & P \\ 2 & 3 & 4 & \cdots & P+1 \end{bmatrix}, \tag{17}$$

which we use to assemble the global stiffness matrix $\boldsymbol{A}$, mass matrix $\boldsymbol{M}$ and load vector $\boldsymbol{f}$. Furthermore, denoting the nodal values of the finite element solution by

$$U_r(t) = u_h(\mathsf{n}_r, t) = u_h(\mathsf{n}_j^{\langle p\rangle}, t) = U_j^{\langle p\rangle}(t) \quad \text{if } r = C_{jp},$$

we have

$$u_h(\rho, t) = U_1^{\langle p\rangle}(t)\psi_1^{\langle p\rangle}(\rho) + U_2^{\langle p\rangle}(t)\psi_2^{\langle p\rangle}(\rho) \quad \text{for } \rho \in [\rho_{p-1}, \rho_p],$$

with

$$u_h(R, t) = U_2^{\langle P\rangle} = U_{P+1} \quad \text{and} \quad v(R) = V_2^{\langle P\rangle} = V_{P+1}. \tag{18}$$

In terms of the nodal basis functions $\chi_r \in V_h$ satisfying $\chi_r(\mathsf{n}_s) = \delta_{rs}$,

$$u_h(\rho, t) = \sum_{s=1}^{P+1} U_s(t)\chi_s(\rho), \qquad f_r(t) = \int_0^1 f(\rho, t)\chi_r(\rho)\rho^2\, d\rho,$$

$$a_{rs} = \int_0^1 \kappa(\rho)\, \frac{d\chi_s}{d\rho}\, \frac{d\chi_r}{d\rho}\, \rho^2\, d\rho, \qquad m_{rs} = \int_0^1 \chi_s(\rho)\chi_r(\rho)\rho^2\, d\rho.$$

In the usual way, we form the nodal vector

$$\boldsymbol{U}(t) = [U_1(t), U_2(t), \ldots, U_{P+1}(t)]^\top$$

and arrive at the system of linear ODEs

$$\boldsymbol{M}\, \frac{d\boldsymbol{U}}{dt} + (\boldsymbol{A} + \boldsymbol{e}_{P+1}\boldsymbol{e}_{P+1}^\top)\boldsymbol{U} = \boldsymbol{f}(t_n) + \boldsymbol{e}_{P+1},$$

where $\boldsymbol{e}_r$ denotes the rth standard basis vector in $\mathbb{R}^{P+1}$; thus, $\boldsymbol{e}_r\boldsymbol{e}_s^\top$ is the $(P+1) \times (P+1)$ matrix with all zero entries except for a 1 in the rs position.

For the time discretisation, we choose a positive integer N, let

$$t_n = n\,\Delta t \quad \text{for } 0 \le n \le N, \text{ where } \Delta t = \frac{T}{N},$$

and seek

$$U_r^n \approx U_r(t_n) = u_h(\mathsf{n}_r, t_n)$$

using the backward Euler method:

$$\boldsymbol{M}\, \frac{\boldsymbol{U}^n - \boldsymbol{U}^{n-1}}{\Delta t} + (\boldsymbol{A} + \boldsymbol{e}_{P+1}\boldsymbol{e}_{P+1}^\top)\boldsymbol{U}^n = \boldsymbol{f}^n + \boldsymbol{e}_{P+1}. \tag{19}$$

Hence, at the nth time step we must solve the linear system

$$\big(\boldsymbol{M} + \Delta t\,(\boldsymbol{A} + \boldsymbol{e}_{P+1}\boldsymbol{e}_{P+1}^\top)\big)\boldsymbol{U}^n = \boldsymbol{M}\boldsymbol{U}^{n-1} + \Delta t\,\big(\boldsymbol{f}^n + \boldsymbol{e}_{P+1}\big),$$

giving the discrete-time finite element solution

$$U^n(\rho) = \sum_{s=1}^{P+1} U_s^n \chi_s(\rho) \approx u_h(\rho, t_n) \approx u(\rho, t_n). \tag{20}$$

We can then obtain an approximation for the average temperature at time t_n,

$$\bar{U}^n = 3\int_0^1 U^n(\rho)\rho^2\, d\rho \approx \bar{u}_h(t_n) \approx \bar{u}(t_n). \tag{21}$$

Figure 1: Spatial profiles: heating up (left) and cooling down (right).

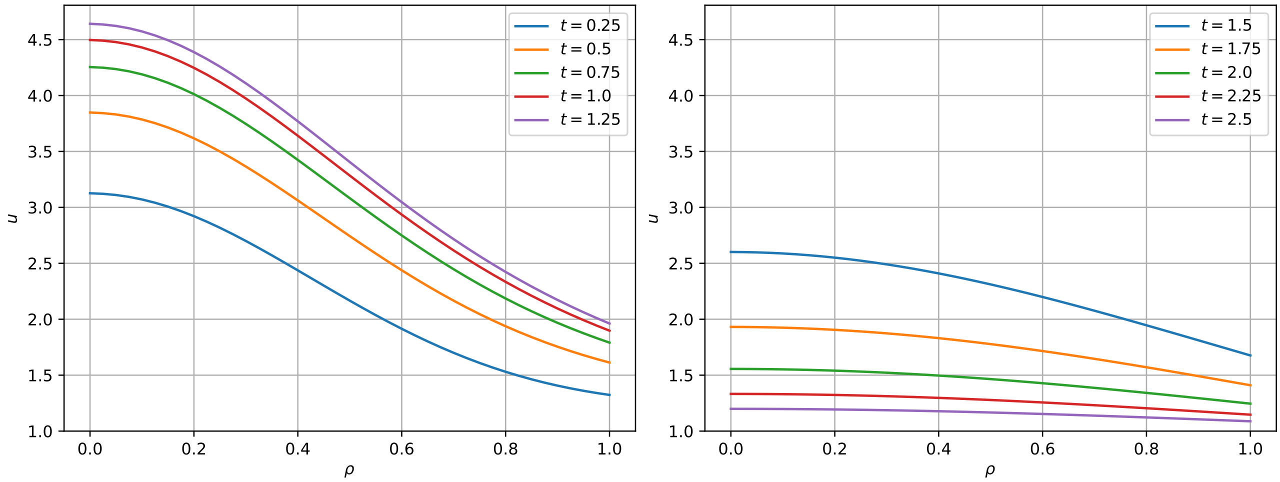
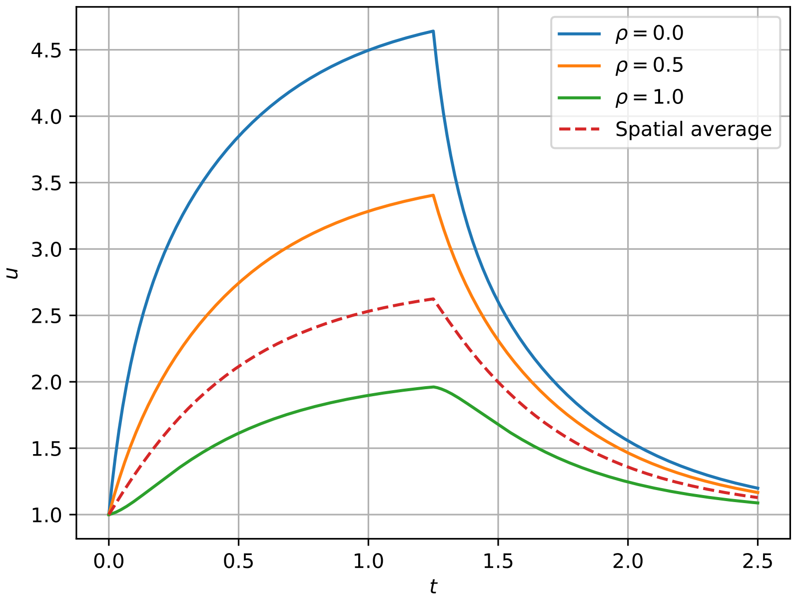


Figure 2: Time profiles. Notice the rapid drop in temperature at $t = 1{\cdot}25$ when the heat source is switched off.



4 Questions for both Math3101 and Math5305

1.

(a) Use the first Green identity to derive the weak formulation (4) from the PDE (1) and boundary conditions (3).

(b) Explain how the dimensionless equations (8) and (9) follow from their dimensioned counterparts (6) and (7).

(c) Show that $\bar{U}^n = 3\,\mathbf{1}^\top \boldsymbol{M}\boldsymbol{U}^n$, where $\mathbf{1}$ denotes the column vector in $\mathbb{R}^{P+1}$ having every component equal to 1.

2. Write a function

```
Ap, Mp = elt_matrices(elt, kappa, w, y)
```

that returns the element stiffness matrix (13) and the element mass matrix (14). The argument `elt` is an array of length two holding the end points of the pth element $[\rho_{p-1}, \rho_p]$, `kappa` is a function that evaluates $\kappa(x)$, and `w` and `y` are arrays holding the weights w_ℓ and points y_ℓ of the quadrature rule (16).

3. Write a function

```
fp = elt_load_vector(elt, f, w, y)
```

that returns the element load vector (15).

4. If we choose $[\rho_{p-1}, \rho_p] = [0{\cdot}5, 1{\cdot}0]$, $\kappa(\rho) = e^{-\rho}$ and $f(\rho) = \sin(\pi\rho)$, then

$$\boldsymbol{A}^{\langle p\rangle} = (13e^{-1/2} - 20e^{-1}) \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{M}^{\langle p\rangle} = \frac{1}{480} \begin{bmatrix} 32 & 23 \\ 23 & 62 \end{bmatrix},$$

with

$$\boldsymbol{f}^{\langle p\rangle} = \frac{1}{2\pi^4} \begin{bmatrix} -\pi^2 + 16\pi - 24 \\ 2\pi^3 - \pi^2 - 20\pi + 24 \end{bmatrix}.$$

(You do not have to derive these formulae.) Use this example in a test script `test_elt_matvec` to check your code from Q2 and Q3. For the quadrature rule (16), your script should read the weights (first column) and points (second column) provided in the file `Gauss8.txt`.

5. Write a function

```
node, t, C = grid_points(P, N, T)
```

that returns the vectors `node` $= [\mathtt{n}_1, \mathtt{n}_2, \ldots, \mathtt{n}_{P+1}]$ and `t` $= [t_0, t_1, \ldots, t_N]$, and the connectivity matrix `C` $= [c_{jp}]$ from (17). For simplicity, use equally spaced nodes.

6. Write a function

```
A, M = assemble_matrices(node, C, kappa, w, y)
```

that assembles the global stiffness matrix `A` and the global mass matrix `M`. Use an appropriate sparse matrix format. Write another function

```
f_vec = assemble_load_vector(node, C, f, w, y)
```

that assembles the global load vector `f_vec`.

7. Write a script `test_assemble` that computes `A`, `M` and `f_vec` in the case $P = 5$, $\kappa(x) = 1$ and $f(x) = 1$. You may assume that in this case

$$\boldsymbol{A} = \frac{1}{15}\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 8 & -7 & 0 & 0 & 0 \\ 0 & -7 & 26 & -19 & 0 & 0 \\ 0 & 0 & -19 & 56 & -37 & 0 \\ 0 & 0 & 0 & -37 & 98 & -61 \\ 0 & 0 & 0 & 0 & -61 & 61 \end{bmatrix}$$

with

$$\boldsymbol{M} = \frac{1}{7500}\begin{bmatrix} 2 & 3 & 0 & 0 & 0 & 0 \\ 3 & 44 & 23 & 0 & 0 & 0 \\ 0 & 23 & 164 & 63 & 0 & 0 \\ 0 & 0 & 63 & 364 & 123 & 0 \\ 0 & 0 & 0 & 123 & 644 & 203 \\ 0 & 0 & 0 & 0 & 203 & 452 \end{bmatrix} \quad \text{and} \quad \boldsymbol{f} = \frac{1}{1500}\begin{bmatrix} 1 \\ 14 \\ 50 \\ 110 \\ 194 \\ 131 \end{bmatrix}.$$

8. Write a function

```
U, Ubar, node, t = backward_Euler(P, N, T, kappa, f, w, y)
```

that computes U_r^n and $\bar{U}^n$ for $0 \le n \le N$ and $1 \le r \le P + 1$.

9. Write a script `profiles` that plots numerical approximations to the temperature profile at times $t = 0{\cdot}25, 0{\cdot}50, \ldots, 2{\cdot}50$ in the case

$$T = 2{\cdot}50, \qquad \kappa(\rho) = 0{\cdot}5, \qquad f(\rho, t) = \begin{cases} 20e^{-4\rho^2} & \text{if } 0 \le t \le T/2, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, a heat source concentrated near the centre of the ball is switched on at $t = 0$ and switched off at $t = T/2$. Plot the the heating-up phase separately from the cooling-down phase, as shown in fig. 1. Create another plot showing how the average temperature and the temperatures at $\rho = 0$, $\rho = 1/2$ and $\rho = 1$ vary with time, as shown in fig. 2.

5 Questions for Math5305 only

10. The backward difference formula of order 2 is given by

$$\frac{3u(t) - 4u(t - \Delta t) + u(t - 2\,\Delta t)}{2\,\Delta t} = u'(t) + O(\Delta t^2),$$

and results in the BDF2 scheme

$$\boldsymbol{M}\,\frac{3\boldsymbol{U}^n - 4\boldsymbol{U}^{n-1} + \boldsymbol{U}^{n-2}}{2\,\Delta t} + (\boldsymbol{A} + \boldsymbol{e}_{P+1}\boldsymbol{e}_{P+1}^\top)\boldsymbol{U}^n = \boldsymbol{f}^n + \boldsymbol{e}_{P+1} \quad \text{for } 2 \le n \le N.$$

Write a function

```
U, Ubar, node, t = BDF2(P, N, T, kappa, w, y)
```

that computes U_r^n and $\bar{U}^n$ for $0 \le n \le N$ and $1 \le r \le P + 1$. Use the backward Euler method for the first step (to compute U_r^1 and $\bar{U}^1$) followed by the BDF2 scheme for $n \ge 2$.

11. Suppose that for some exponents $\mu = \mu(t_n) > 0$ and $\nu = \nu(t_n) > 0$,

$$\bar{U}^n = \bar{u}(t_n) + C_1\,\Delta t^{\mu} + C_2\,h^{\nu} + (\text{higher order terms in } \Delta t \text{ and } h),$$

and write $U^n = U^n(P, N)$, $\Delta t = \Delta t(N)$ and $h = h(P)$ to show the dependence on N and P.

(a) Given a sufficiently large number P and a sequence N_1, N_2, N_3, … such that $N_{j+1} = 2N_j$, show that

$$\mu(T/4) \approx \log_2\left(\frac{V_{j-2} - V_{j-1}}{V_{j-1} - V_j}\right) \quad \text{where} \quad V_j = \bar{U}^{N_j/4}(P, N_j),$$

assuming N_1 is divisible by 4 (so $N_j/4$ is always a whole number).

(b) Given a sequence P_1, P_2, P_3, … such that $P_{j+1} = 2P_j$ and a sufficiently large number N (divisible by 4), show that

$$\nu(T/4) \approx \log_2\left(\frac{W_{j-2} - W_{j-1}}{W_{j-1} - W_j}\right) \quad \text{where} \quad W_j = \bar{U}^{N/4}(P_j, N).$$

12. Write a script `convergence` that uses these formulae to estimate $\mu(T/4)$ and $\nu(T/4)$ for the implicit Euler and BDF2 schemes.

Matlab notes. As in Lab 7, store U_p^n in `U(p+1,n+1)` and use `chol` to compute the Cholesky factorization needed for the linear solve at each time step. The command

```
load Gauss8.txt
```

will read the data from the file and use it to create a matrix `Gauss8` from which you can extract the weights and points for the quadrature rule.

Python notes. As in Lab 7, store U_p^n in `U[n,p]` and use `scipy.sparse.linalg.factorized` to create a `solve` function that you use to solve the linear system at each time step. The commands

```
from numpy import loadtxt
Gauss8 = loadtxt('Gauss8.txt')
```

will read the data from the file and use it to create an array `Gauss8` from which you can extract the weights and points for the quadrature rule. If you prefer, you can keep your source code in a Jupyter notebook instead of in several Python source files (so you upload just a single `.ipynb` file instead of several `.py` files). You could also include your answers to Q1 and Q11 in the notebook if you wish, instead of using a separate pdf.