The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

G14CAM Computational Applied Mathematics

— Coursework 2 (20%) —

Choice P (Parabolic PDEs with a financial flavour)

Hand-in deadline: ***6pm, Wednesday, 10 May 2017***

Instructions:

- Your coursework submission should simply contain your answers to the numbered questions below.
- You may use the **LaTeX template** (download on Moodle) to type your coursework.
- Upload **2 files** on Moodle: a PDF of your coursework submission and
  a ZIP containing all the computer codes you used to do your coursework.

Rules:

- You are allowed to work together, but you must write your own report.
- You must program by yourself all code, except for the codes that are made available for sharing through Moodle.
- If you use shared code, you must declare it in your report.
- Only certain subroutines and sub-functions will be allowed for sharing (suggestions are welcome).
- For this coursework, each student may again share up to three of his/her new subroutines. Email these to the lecturer for approval. This can be done any time before **6pm Wednesday, 3 May**.
- Just for fun: A ranking list is kept to determine at the end of the semester the top-3 students whose shared code have been used most often by others!

This coursework contributes **20%** towards the overall grade for the module.

A total of **100** marks are available for this coursework.

Coursework aim: In this coursework you will develop computer code to solve partial differential equations (and inequalities). You have a *choice* of two different sets of 5 problems:

- **Choice E (Elliptic PDEs in 2D domains)**
  (1D Elliptic PDE, Convection–Diffusion–Reaction (CDR) in a square and in a triangle, 1D Elliptic Ineq., CDR Ineq.)
- **Choice P (Parabolic PDEs with a financial flavour)**
  (1D Elliptic PDE, Heat Eq., Black–Scholes Eq., 1D Elliptic Ineq., Black–Scholes Ineq.)

Important note I: Most problems within a set have the following *generic questions*:

Q-0 **(a) PDE Discretisation**

- Introduce an appropriate set of grid-points. Obtain the system of equations (or inequalities). Do this by deriving them yourself, or by stating a relevant result in any of the reading materials.
- Write down the corresponding matrix equation(s) (or linear complementary system(s)).

(b) Algorithm Implementation Write a computer code to solve the discretised problem with the indicated method.

- Provide a print-out of your code in your report.

(c) Code Verification Carry out a verification of your implementation by comparing computed approximations with the indicated exact solution:

- Plot a sufficiently-coarse (but not too coarse) approximation and the exact solution in one figure. (The difference between both should be visible.) In case of a 2D domain, plot the approximation and solution versus x for $y = 0.25$, 0.5 and 0.75. In case of a time-dependent problem, plot the approximation and solution versus x for $t = 0$, $T/2$ and T.
- Study the convergence of the method: Define an appropriate norm. Provide a table with the norm of errors for a sequence of appropriate mesh-sizes (and corresponding time-step sizes in case of a time-dependent PDE). Plot the errors versus the mesh-widths (using double-log axes). Comment on the observed rates of convergence.

Important note II: Problems with a star ($\star$) are tedious, but not directly needed to continue further.

Choice P (Parabolic PDEs with a financial flavour)

P-1 1-D Elliptic PDE

First consider the 1D elliptic PDE for $u : (0,1) \to \mathbb{R}$ subject to Dirichlet BCs (boundary conditions):

$$-u'' = f \qquad \text{for } x \in (0,1) \tag{1a}$$
$$u(0) = \alpha \tag{1b}$$
$$u(1) = \beta \tag{1c}$$

with $f : (0,1) \to \mathbb{R}$, $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$.

(a) PDE Discretisation [See **Q-0(a)**] Obtain the discretisation for a second-order difference scheme as explained in Chapter 2 in [LeVeque, 2007]. **[5 marks]**

(b) Algorithm Implementation [See **Q-0(b)**] **[5 marks]**

(c) Code Verification [See **Q-0(c)**] Verify against the exact solution

$$u_{\text{exact}}(x) = x - \sin(\pi x)\,, \tag{2}$$

using the data:[1]

	α	β	$f(x)$
Case 1:	0	1	$-u''_{\text{exact}}(x)$

[5 marks]

(d⋆) Problem Variation Repeat **(c)** but now using

$$u_{\text{exact}}(x) = x\sqrt{x}\ . \tag{3}$$

[5 marks]

[1] The verification technique, where a given exact solution is used to generate the data in the problem, is referred to as the *manufactured-solution technique*.

P-2 Heat Equation

Consider the parabolic PDE (heat equation) for $u(t,x) \in \mathbb{R}$ subject to Dirichlet BCs and an initial condition:

$$u_t - a u_{xx} = 0 \qquad \text{for } t \in (0,T]\,,\ x \in (0,1) \tag{4a}$$
$$u(t,0) = g_0 \tag{4b}$$
$$u(t,1) = g_1 \tag{4c}$$
$$u(0,x) = u_0(x) \tag{4d}$$

with $a > 0$ (heat-conduction coefficient), $T > 0$ and $u_0 : (0,1) \to \mathbb{R}$.

The solution $u(t,x)$ represents the temperature, at time t and position x, in a one-dimensional piece of conductive material.

(a) PDE Discretisation [See **Q-0(a)**] Obtain the discretisation for the implicit method (backward difference in time, centred in space) as explained in Chapter 9 in [Epperson, 2013]. **[5 marks]**

(b) Algorithm Implementation [See **Q-0(b)**] **[5 marks]**

(c) Code Verification [See **Q-0(c)**] Verify against the exact solution

$$u_{\text{exact}}(t,x) = \mathrm{e}^{-4t}\sin(2\pi x) + x\,, \tag{5}$$

using the data:

	a	T	g_0	g_1	$u_0(x)$
Case 2:	π^{-2}	1	0	1	$\sin(2\pi x) + x$

Hint: In studying the convergence, take $\Delta t = Ch^2$ for some chosen C (with h the mesh width and Δt the time-step size). Measure errors in the max-norm at the final time T, in other words,

$$\max_i |e_i^N|\,, \tag{6}$$

with N such that $N\Delta t = T$. **[5 marks]**

P-3 **Black–Scholes Equation (European option)**

Consider the parabolic PDE (Black–Scholes equation) for $v(t,x) \in \mathbb{R}$ subject to time-dependent Dirichlet BCs and an initial condition:

$$\frac{\partial v}{\partial t} - \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 v}{\partial x^2} - r\,x\,\frac{\partial v}{\partial x} + r\,v = 0 \qquad \text{for } t \in (0,T]\,,\ x \in (0,R) \tag{7a}$$

$$v(t,0) = f_0(t) \tag{7b}$$

$$v(t,R) = f_R(t) \tag{7c}$$

$$v(0,x) = g(x) \tag{7d}$$

with $\sigma > 0$ (constant volatility), $r > 0$ (interest rate), $T > 0$, $R > 0$, $f_0 : (0,T] \to \mathbb{R}$, $f_R : (0,T] \to \mathbb{R}$, and $g : \mathbb{R} \to \mathbb{R}$ (pay-off function).

The solution $v(t,x)$ represents the value of a European option with maturity time T, at time-to-maturity t and spot price x.[2]

(a⋆) PDE Discretisation [See **Q-0(a)**] Use the following *implicit* finite-difference scheme for (7a)–(7d): Consider the PDE (7a) at an arbitrary point $(x,t) = (x_i, t_{n+1})$, then replace $\frac{\partial v}{\partial t}$ by $\frac{v_i^{n+1} - v_i^n}{\Delta t}$, replace $\frac{\partial^2 v}{\partial x^2}$ by the standard second-order central difference approximation at t_{n+1}, replace $\frac{\partial v}{\partial x}$ by $\frac{v_{i+1}^{n+1} - v_i^{n+1}}{h}$, and replace v by v_i^{n+1}. **[5 marks]**

(b⋆) Algorithm Implementation [See **Q-0(b)**] **[5 marks]**

(c⋆) Code Verification I [See **Q-0(c)**] Verify against the exact solution

$$v_{\text{exact}}(t,x) = K\mathrm{e}^{-rt}\,\Phi\big(-d_-(t,x)\big) - x\,\Phi\big(-d_+(t,x)\big)\,, \tag{11}$$

where

$$d_\pm(t,x) := \frac{1}{\sigma\sqrt{t}}\Big(\ln\big(x/K\big) + \big(r \pm \tfrac{\sigma^2}{2}\big)t\Big) \tag{12}$$

and $\Phi(\cdot)$ is the cumulative distribution function[3] of the standard normal distribution:

$$\Phi(d) := \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{d} e^{-z^2/2}\,\mathrm{d}z\,, \tag{13}$$

using the data:

[2] In particular, v is related to V by $v(T-t,x) = V(t,x)$, where $V(t,x)$ is the value of the European option in *forward* time t, and which satisfies

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 V}{\partial x^2} + r\,x\,\frac{\partial V}{\partial x} - r\,V = 0\,, \tag{8}$$

subject to the *terminal* condition $V(T,x) = g(x)$. The value of the European option is given by the (discounted) expected pay-off at maturity:

$$V(t,x) = \mathrm{e}^{-r(T-t)}\,\mathbb{E}\Big[g\big(S(T)\big)\,\Big|\,S(t) = x\Big]\,, \tag{9}$$

where $S(t) = S_0\,\mathrm{e}^{(r-\sigma^2/2)t + \sigma W(t)}$ is the stock price according to the stochastic differential equation:

$$\mathrm{d}S = rS\,\mathrm{d}t + \sigma S\,\mathrm{d}W\,. \tag{10}$$

The connection between (9) and (8) is given by the Feynman–Kac formula.

[3] It is recommended to use a built-in function for this Gaussian cumulative distribution function. For example,

	r	σ	T	R	$f_0(t)$	$f_R(t)$	K
Case 3(i):	0	0.5	5	300	$K\,\mathrm{e}^{-rt}$	$v_{\text{exact}}(t,R)$	100

and[4]

$$g(x) = \begin{cases} K - x & \text{if } x < K\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{14}$$

[5 marks]

(d⋆) Code Verification II Repeat Question **(c)** but now for the data:

	r	σ	T	R	$f_0(t)$	$f_R(t)$	K
Case 3(ii):	0.1	0.1	5	300	$K\,\mathrm{e}^{-rt}$	$v_{\text{exact}}(t,R)$	100

[5 marks]

(e⋆) Scheme Variation

- Explain how your results in **(d⋆)** would be different if in the scheme you replace $\frac{\partial v}{\partial x}$ by $\frac{v_{i+1}^{n+1} - v_{i-1}^{n+1}}{2h}$.
- Similarly, explain how they would be different if you replace $\frac{\partial v}{\partial x}$ by $\frac{v_{i}^{n+1} - v_{i-1}^{n+1}}{h}$. **[5 marks]**

$\Phi(x)$ is related to built-in functions `erf` and `erfc`:

$$\Phi(x) = \frac{1}{2}\mathtt{erfc}\big(-\frac{x}{\sqrt{2}}\big) = \frac{1}{2}\Big(1 - \mathtt{erf}\big(-\frac{x}{\sqrt{2}}\big)\Big)$$

[4] The pay-off in (14) is typical for a *put* option with strike price K.

P-4 1-D Elliptic Inequality

Consider now the 1D elliptic *inequality* problem for $u : (0,1) \to \mathbb{R}$ subject to the Dirichlet BCs:

$$\left.\begin{aligned} -u'' &\le f \\ u &\le \psi \\ (-u'' - f)(u - \psi) &= 0 \end{aligned}\right\} \quad \text{for } x \in (0,1) \tag{15}$$

$$u(0) = 0 \tag{16}$$

$$u(1) = 0 \tag{17}$$

with $f : (0,1) \to \mathbb{R}$ and $\psi : (0,1) \to \mathbb{R}$ (obstacle function).

(a) PDE Discretisation [See **Q-0(a)**] Obtain the discretisation for a second-order difference scheme as explained in Lecture 9 notes. **[5 marks]**

(b) Algorithm Implementation [See **Q-0(b)**] Implement the Projected SOR method with relaxation parameter ω. **[5 marks]**

(c) Code Verification Instead of the usual code verification with a complete exact solution, consider the following data

	$f(x)$	$\psi(x)$	ω
Case 4:	$50/3$	1	1.8

• Take a mesh-width $h = 1/16$. Plot the first 8 iterations of the Projected SOR method ($\omega = 1.8$), assuming an initial guess $\underset{\sim}{\boldsymbol{u}}^{(0)} = \underset{\sim}{\boldsymbol{0}}$.

• It may take many iterations for the Projected SOR method to converge. Come up with a criterion to terminate the iterations, and implement this.

• For $h = 1/16$ compare the “converged” approximation to the unconstrained approximation of the standard elliptic model (i.e., take $\alpha = \beta = 0$ and $f(x) = 50/3$ in **P-1**). Explain why you think the results are correct.

• It is known that for the above data, the exact free boundary is at $x = \frac{1}{5}\sqrt{3}$ and $1 - \frac{1}{5}\sqrt{3}$. Given this information on the exact solution, come up with a verification of your implementation. **[10 marks]**

P-5 **Black–Scholes Inequality (American option)**

Consider finally the parabolic *inequality* problem (Black–Scholes model for American put option) for $v(t,x) \in \mathbb{R}$ subject to Dirichlet BCs and an initial condition:

$$\left.\begin{aligned} \frac{\partial v}{\partial t} - \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 v}{\partial x^2} - r\,x\,\frac{\partial v}{\partial x} + r\,v \geq 0 \\ v \geq g \\ \Big(\frac{\partial v}{\partial t} - \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 v}{\partial x^2} - r\,x\,\frac{\partial v}{\partial x} + r\,v\Big)\Big(v - g\Big) = 0 \end{aligned}\right\} \text{ for } t \in (0,T]\,,\ x \in (0,R) \tag{18a}$$

$$v(t,0) = K \tag{18b}$$

$$v(t,R) = 0 \tag{18c}$$

$$v(0,x) = g(x) \tag{18d}$$

where, as before,

$$g(x) = \begin{cases} K - x & \text{if } x < K\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{19}$$

The solution $v(t,x)$ represents the value of an American put option (with strike price K) at time-to-expiration t and spot price x.[5]

(a⋆) PDE Discretisation [See **Q-0(a)**] Obtain a finite-difference discretisation for (18a)–(18d) by extending your previous work.

- Explain how the Projected SOR method can be employed to solve the discrete problem. **[5 marks]**

(b⋆) Algorithm Implementation [See **Q-0(b)**] Implement the Projected SOR method with relaxation parameter ω for your discretisation. **[5 marks]**

(c⋆) Code Verification Consider the following data

	r	σ	T	R	K	ω
Case 5:	0.05	0.5	5	300	100	1.8

- Compare the American-option approximations with those obtained using the standard Black–Scholes equation for European options (i.e., take $f_0(t) = K$ and $f_R(t) = 0$ in **P-3**). Choose a sufficiently-coarse (but not too coarse) discretisation,

[5] In particular, v is related to V by $v(T-t,x) = V(t,x)$ where $V(t,x)$ is the value of the American option in *forward* time t. American options differ from European options in that they can be exercised at any point up to the expiry time T. The value of American options is therefore higher than European options because of this flexibility. In this case,

$$V(t,x) = \sup_{\tau \in \mathcal{T}_{t,T}} \mathbb{E}\Big[e^{-r(\tau - t)}\, g(S(\tau)) \Big| S(t) = x\Big] \tag{20}$$

The supremum (optimum) in (20) is attained at the so-called *stopping time*.

and plot both approximations in the same figure versus x for $t = 0$, $T/2$ and T. Explain why you believe your results are correct. **[5 marks]**

(d$\star$) Accurate Simulation For the data in Case 5, use your code with a sufficiently accurate discretisation to plot the free boundary $x_{\mathrm{fb}}(t)$ versus time. Which discretisation parameters (h, Δt) did you use? Explain how you obtained your plot.[6] **[5 marks]**

[6] The free boundary $x_{\mathrm{fb}}(t)$ represents the so-called *optimal stopping boundary*. This is the boundary between the *stopping domain*, at which it is optimal to exercise the option, and the *continuation domain*, at which it is optimal to not exercise the option.