INDIAN STATISTICAL INSTITUTE

Second Semester Examination : 2015–16

Course : Bachelor of Statistics (Hons.)

Subject : Numerical Analysis : BStat-I

Date : 4 May 2016 | Maximum Marks : 100 | Duration : 3 Hours

Attempt any four (4) from the first eight (8) problems. In addition, attempt the bonus problem.

Problem 1 [25]

A. Describe the *fixed-point* and *floating-point* representations of numbers in a computer system. Describe the standard IEEE format for storing a `float`, as in C? What is the largest and the smallest (positive) number that can be represented using the floating-point format $1.\square\square\square \times 2^E$, where $-2 \leq E \leq 2$, and the empty boxes contain binary digits 0 or 1? [15]

B. Suppose that you want to compute the matrix-vector product $\mathbf{Av}$. If the matrix is $\mathbf{A}$ fixed and error-free, but the vector $\mathbf{v}$ is an input prone to errors, describe the *relative forward error* and the *relative backward error* of this numerical computation. Hence compute the *relative condition number* of the matrix-vector product, and provide a bound for the same. [10]

Problem 2 [25]

Given a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, the *simple LU decomposition* of $\mathbf{A}$ may be defined as $\mathbf{A} = \mathbf{LU}$, where $\mathbf{L} \in \mathbb{R}^{n\times n}$ is a lower-triangular matrix, and $\mathbf{R} \in \mathbb{R}^{n\times n}$ is an upper-triangular matrix.

A. Describe an algorithm that may decompose $\mathbf{A}$ into $\mathbf{L}$ and $\mathbf{U}$ through successive left multiplication by lower triangular matrices, such that $\mathbf{L}_n \cdots \mathbf{L}_2\mathbf{L}_1\mathbf{A} = \mathbf{U}$, and $\mathbf{L} = (\mathbf{L}_n \cdots \mathbf{L}_2\mathbf{L}_1)^{-1}$. [10]

B. Is it always possible to obtain such a *simple LU decomposition* of $\mathbf{A} \in \mathbb{R}^{n\times n}$? If so, provide a complete justification that your proposed algorithm (in Part A) achieves a *simple LU decomposition* for any matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$. If not, state a case where the *simple LU decomposition* may not be feasible, and modify the proposed algorithm (in Part A) to suit such a case? [15]

Problem 3 **[25]**

A. Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and a vector $\mathbf{b} \in \mathbb{R}^m$, propose a (linear algebraic) numerical method to minimize $||\mathbf{Ax} - \mathbf{b}||_2$, where $\mathbf{x} \in \mathbb{R}^n$. Modify this method to minimize $||\mathbf{Ax} - \mathbf{b}||_2^2 + \lambda ||\mathbf{x}||_2^2$, where $\mathbf{x} \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$ is a given constant. [15]

B. What is the output of the following algorithm, if it terminates? Justify your answer. [10]

> **Input** : Matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ and threshold $\epsilon > 0$
>
> Choose $\mathbf{v}^{(0)} \in \mathbb{R}^n$ at random, with $||\mathbf{v}^{(0)}||_2 = 1$
>
> Compute $r^{(0)} = (\mathbf{v}^{(0)})^T \mathbf{A}\ \mathbf{v}^{(0)}$
>
> **for** $k = 1, 2, 3, \ldots$ **do** :
>
>   Solve $(\mathbf{A} - r^{(k-1)} \mathbf{I}_{n \times n})\ \mathbf{w} = \mathbf{v}^{(k-1)}$, for $\mathbf{w}$
>
>   Normalize $\mathbf{v}^{(k)} = \mathbf{w} / ||\mathbf{w}||_2$
>
>   Compute $r^{(k)} = (\mathbf{v}^{(k)})^T \mathbf{A}\ \mathbf{v}^{(k)}$
>
>   **if** $||r^{(k)} - r^{(k-1)}||_2 < \epsilon$: **break**
>
> **return** $r^{(k)}$ and $\mathbf{v}^{(k)}$

Problem 4 **[25]**

Given n points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ in $\mathbb{R}^2$, there exists a univariate polynomial $p(x)$, of degree at most $(n-1)$, passing through all the points. That is, $y_i = p(x_i)$ for all $i = 1, 2, \ldots, n$.

A. Suppose that $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$, expressed in the monomial basis. Describe an algorithm to construct $p(x)$, that is, to construct the monomial basis coefficients $a_0, a_1, \ldots, a_{n-1}$, given the n points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. [10]

B. Suppose that $p(x) = b_0 + b_1(x - x_1) + b_2(x - x_1)(x - x_2) + \cdots + b_{n-1}(x - x_1)(x - x_2) \cdots (x - x_n)$, expressed in the Newton basis. Describe an algorithm to construct $p(x)$, that is, to construct the coefficients $b_0, b_1, \ldots, b_{n-1}$, given the n points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. [15]

Problem 5 **[25]**

Given a function $f : \mathbb{R} \to \mathbb{R}$, one may find a root of the equation $f(x) = 0$ by converting the problem into a fixed-point problem $g(x) = x$, so that any fixed-point of $g(x)$ is a solution to $f(x) = 0$.

A. Describe an algorithm to solve the fixed-point problem $g(x) = x$, that is, an algorithm that finds out a fixed point of a given function $g(x)$. Discuss the convergence of this algorithm. [10]

B. How does the Newton-Raphson method for finding a root of the equation $f(x) = 0$ choose $g(x)$? Describe the Newton-Raphson method, and discuss the convergence of this algorithm. [15]

Problem 6 **[25]**

Given a function $f : \mathbb{R} \to \mathbb{R}$, one may want to estimate the values of the derivatives $f'(x)$ or $f''(x)$ at a specific point $x = x_0$ by evaluating the function $f(x)$ at the neighborhood of $x = x_0$.

A. From the definition of derivative, suggest an estimate for $f'(x_0)$ in terms of the functional values $f(x_0)$ and $f(x_0 + h)$, where h is adequately small. Comment on the accuracy of this estimate. Provide an *improved* estimate for $f'(x_0)$ when you are allowed to use three values of the function, $f(x_0)$, $f(x_0 + h)$ and $f(x_0 - h)$. Comment on the accuracy of this improved estimate. [15]

B. Extend the same logic (as in Part A) to suggest an estimate for $f''(x_0)$ in terms of the values of $f(x)$ at the neighborhood of $x = x_0$. How many functional values would you require, and what would be the accuracy of your estimate? [10]

Problem 7 **[25]**

Given a function $f : \mathbb{R} \to \mathbb{R}$, one may find the value of $\int_a^b f(x)dx$ by sub-dividing the interval $[a, b]$ into adequately small sub-intervals, and applying a quadrature rule on each small interval.

A. Suppose that $h = (b - a)/k$, and the small subintervals are $[x_i, x_{i+1}]$, where $x_i = a + (i - 1) \times h$ for $i = 1, 2, \ldots, k$. Describe mid-point rule and trapezoidal rule for computing $\int_a^b f(x)dx$. [10]

B. Suppose that $h = (b - a)/k$, and the small subintervals are $[x_i, x_{i+1}]$, where $x_i = a + (i - 1) \times h$ for $i = 1, 2, \ldots, k$. Describe Simpson's rule for computing $\int_a^b f(x)dx$. Compare the accuracy of Simpson's rule with that of Mid-Point rule and Trapezoidal rule (as in Part A). [15]

Problem 8 **[25]**

Suppose that we want to solve the first-order ordinary differential equation $y'(t) = F(t, y)$, subject to the initial value $y(0) = 1$, where the relation between t, y, y' is given by $F(t, y) = t + 2t \cdot y(t)$. To solve this problem, one may consider an equally-spaced time sequence $t_0 = 0, t_1, t_2, \ldots, t_k, t_{k+1}, \ldots$, with $t_{k+1} = t_k + h$ for all $k \geq 0$, and provide a recurrence relation to compute $y(t_{k+1})$ given $y(t_k)$.

A. Describe the *forward* and *backward* Euler methods to compute $y(t_{k+1})$ given $y(t_k)$ in case of the ODE $\{y'(t) = t + 2t \cdot y(t);\ y(0) = 1\}$. Which method is more efficient, and why? [10]

B. Describe Runge-Kutta method for solving the ODE $\{y'(t) = t + 2t \cdot y(t);\ y(0) = 1\}$. Compare the accuracy of Runge-Kutta method with that of the Euler methods for solving ODEs. [15]

—— *End of the segment for regular problems. Please turn over for the Bonus problem.* ——

Bonus Problem **[10]**

Suppose that G is an undirected graph consisting of 6 vertices $\{A, B, C, D, E, F\}$, and an unknown number of edges. You are provided with neither the adjacency matrix of the graph, nor with the list of edges in it. However, you know that the following algorithm was executed step-by-step on the graph G, and the corresponding output is as follows.

Algorithm

1. Construct the 6×6 adjacency matrix $\mathbf{A}$ of the graph G.
2. Construct a 6×6 matrix $\mathbf{D}$ that has the degrees of the six vertices in G as its diagonal elements. All other elements of $\mathbf{D}$ are zero.
3. Construct a 6×6 matrix $\mathbf{L}$ as follows: $\mathbf{L} = \mathbf{D} - \mathbf{A}$.
4. Compute the eigenvalues and eigenvectors of the matrix $\mathbf{L}$.

Output

```
Eigenvalues of L

4.561553e+00  3.000000e+00  3.000000e+00  3.000000e+00  4.384472e-01  5.313210e-16

Eigenvectors of L (in columns)

    [v1]          [v2]           [v3]          [v4]         [v5]        [v6]
 0.6571923   0.066054535   0.000000e+00   0.5735592  -0.2609565   0.4082483
 0.1845241  -0.734936838   2.807975e-02  -0.2059434   0.4647051   0.4082483
 0.1845241   0.668882304  -2.807975e-02  -0.3676157   0.4647051   0.4082483
-0.1845241  -0.060922638  -7.065490e-01  -0.2835670  -0.4647051   0.4082483
-0.6571923   0.066054535   7.979728e-17   0.5735592   0.2609565   0.4082483
-0.1845241  -0.005131897   7.065490e-01  -0.2899922  -0.4647051   0.4082483
```

Interpret the output eigenvalues and eigenvectors of $\mathbf{L}$ to take an educated guess about the structure of the graph G, and draw a regular vertex-edge layout of the graph to depict your guess.

Good luck! ☺