MATH 661.FA23 Final Examination

Instructions. Solve the problems for your course track. Formulate your answers clearly and cogently. Sketch out an approach on scratch paper first. Concisely transcribe the approach to the answer you turn in, followed by appropriate calculations and conclusions, within allotted time. Use concise, complete English sentences in the description of your approach. Each question is meant to be completely answered and transcribed from proof to final copy within thirty minutes. Concentrate foremost on clear exposition of the concept underlying your approach. Sign each page thus acknowledging honor code rules. Do not turn in scratch work.

Grading. $4\,\text{problems} \times 5\,\text{points}/\text{problem} = 20\,\text{points}$. $\text{Midterm}\,n = \max\,(\text{Midterm score}, 2 \times \text{problem}\,n)$

Preamble. The course objectives set out in the syllabus are reproduced below, along with a formulary. Problems probe *understanding* of the course concepts, rather than rote memorization.

"There are just four mathematics ideas that are considered in this course:

Approximation. The notion of replacing some complicated mathematical object by one that is simpler to compute. In succession, the course presents the approximation of numbers, functions, and operators. The main focus is on numerical approximation, but computational analytical approximation is also presented.

Linear combination. An approach to constructing complex objects by scaling and addition of simpler objects. Note the link to approximation, in that "simpler to compute" is interpreted as scaling and addition.

Nonlinear combination. An approach to constructing complex objects by function composition, successive nonlinear transformation of simpler objects.

Limits. An approach to constructing complex objects by a sequence of approximations.

Comparable simplicity is encountered in computational ideas:

Memory management. Transfer and organization of data on a computer.

Repetition. Multiple execution of a task. Two repetition types are encountered:

Iteration. A portion of code that is repeatedly executed, typically within a loop.

Recursion. A portion of code organized as a function that calls itself.

Condition testing. Carrying out decisions based on data."

Formulary. Newton method to solve $f(x) = 0$: $x_{n+1} = x_n - f(x_n) / f'(x_n)$.

Newton form of interpolating polynomial: $p_n(t) = [y_0] + [y_1, y_0](t - x_0) + \cdots + [y_n, \cdots, y_0](t - x_0)...(t - x_{n-1})$

Divided differences: $[y_k] = y_k$, $[y_{k+m}, ..., y_k] = ([y_{k+m}, ..., y_{k+1}] - [y_{k+m-1}, ..., y_k]) / (x_{k+m} - x_k)$

Finite difference operators: $E^k f(x) = f(x + kh)$, $\Delta f(x) = (E - I) f(x)$, $\nabla f(x) = (I - E^{-1}) f(x)$, $\delta f(x) = (E^{1/2} - E^{-1/2}) f(x)$

Binomial expansion: $(x + y)^n = \sum_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) x^{n-k} y^k$, $\left(\begin{array}{c} n \\ k \end{array} \right) = \dfrac{n!}{k!\,(n-k)!}$, $n \in \mathbb{N}$

Generalized binomial expansion: $(x + y)^\alpha = \sum_{k=0}^{\infty} \left(\begin{array}{c} \alpha \\ k \end{array} \right) x^{\alpha-k} y^k$, $\alpha \in \mathbb{R}$

Interpolation operator: $x_k = x_0 + kh$, $p_n(x_0 + \alpha h) = (I + \triangle)^a y_0$

Householder reflector $\boldsymbol{H} = \boldsymbol{I} - 2\dfrac{\boldsymbol{v}\boldsymbol{v}^*}{\boldsymbol{v}^*\boldsymbol{v}}$

1 Track 1

1. Consider $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m \times m}$ symmetric. Determine if the following are also symmetric.

   a) $\boldsymbol{A}^2 - \boldsymbol{B}^2$

   b) $(\boldsymbol{A} + \boldsymbol{B})(\boldsymbol{A} - \boldsymbol{B})$

   c) $\boldsymbol{A}\boldsymbol{B}\boldsymbol{A}$

   d) $\boldsymbol{A}\boldsymbol{B}\boldsymbol{A}\boldsymbol{B}$

2. Construct a spline interpolant of data $\mathcal{D} = \{(x_i, y_i), i = 0, 1, .., n\}$ using piecewise trigonometric functions $s_i: [x_{i-1}, x_i] \to \mathbb{R}$, $i = 1, 2, .., n$

$$s_i(t) = a_i + b_i \cos(t) + c_i \sin(t).$$

3. Find a quadrature formula

$$\int_{-1}^{1} f(x)\,\mathrm{d}x \cong c(f(x_0) + f(x_1) + f(x_2))$$

   that is exact for all quadratic polynomials.

4. Newton's method to solve $f(x) = 0$ uses data $\mathcal{D}_n = \{(x_n, f_n, f_n')\}$ to construct a linear approximant g, and the next iterate x_{n+1} is the solution of $g(x) = 0$. Construct an analogous method based upon data $\mathcal{D} = \{(x_{n-1}, f_{n-1}, f_{n-1}'), (x_n, f_n, f_n')\}$, and establish its order of convergence ($f_n = f(x_n)$, $f_n' = f'(x_n)$).

2 Track 2

1. For $\boldsymbol{A}=[A_{ij}], \boldsymbol{B}=[B_{ij}], \boldsymbol{C}=[C_{ij}]\in\mathbb{R}^{m\times m}, m=2$, the Strassen algorithm computes $\boldsymbol{C}=\boldsymbol{A}\boldsymbol{B}$ as

   ◦ $$P_1=(A_{11}+A_{22})(B_{11}+B_{22}), P_2=(A_{21}+A_{22})B_{11}, P_3=A_{11}(B_{12}-B_{22}), P_4=A_{22}(B_{21}-B_{11}),$$
   $$P_5=(A_{11}+A_{12})B_{22}, P_6=(A_{21}-A_{11})(B_{11}+B_{12}), P_7=(A_{12}-A_{22})(B_{21}+B_{22}),$$
   $$C_{11}=P_1+P_4-P_5+P_7, C_{12}=P_3+P_5, C_{21}=P_2+P_4, C_{22}=P_1+P_3-P_2+P_6.$$

   a) Compare the computational complexity of the Strassen algorithm with that of standard matrix multiplication
   $$C_{ij}=\sum_{k=1}^{m} A_{ik}B_{kj}.$$
   Separately count multiplications and additions.

   b) Consider $m=2n$ and block-structured matrices $\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}$, e.g., $A_{ij}\in\mathbb{R}^{n\times n}$. Compare the computational complexity of the Strassen algorithm with that of standard matrix multiplication in this case.

   c) Consider $m=2^p$ and compare the computational complexities of Strassen and standard matrix multiplication.

   d) In two-decimal-digit floating point arithmetic ($\mathbb{F}_2$) the matrices
   $$\boldsymbol{A}=\boldsymbol{B}=\begin{bmatrix} .99 & .0010 \\ .0010 & .99 \end{bmatrix},$$
   are exactly represented. The Strassen algorithm produces $\tilde{C}_{12}=0.0$. Compute C_{12} in $\mathbb{F}_2$ using standard multiplication and identify the source of any discrepancy with respect to $\tilde{C}_{12}$.

2. Consider the cubic spline interpolant $S(t)$ of $f:[-1,1]\to\mathbb{R}$, constructed from data $\mathcal{D}=\{(x_i, y_i=f(x_i)), i=0,1,..,n\}$.

   a) Define the mathematical problem $\mathcal{P}$ of constructing $S(t)$.

   b) Determine the condition number κ of problem $\mathcal{P}$.

   c) Can $\kappa=1$? Under what conditions on f and choice of sampling points x_i.

3. Construct a Gauss quadrature for integrals of form
   $$\int_0^1 e^{-t\boldsymbol{P}}\boldsymbol{f}(t)\,\mathrm{d}t,$$
   that is exact for cubic $\boldsymbol{f}(t)$, where $\boldsymbol{P}\in\mathbb{R}^{m\times m}$ is a projection matrix.

4. Construct a second-order accurate scheme to solve the integro-differential equation
   $$my''+cy'+ky=\int_0^t R(t-\tau)\,y(\tau)\,\mathrm{d}\tau,$$
   that describes the motion of a point mass m subject to drag $-cv=-cy'$, elastic force $-ky$, and internal relaxation $R*y$, starting from initial conditions $y(0)=y_0,\ y'(0)=v_0$.