# Declaration of Integrity for online assessments

By signing this statement, I hereby declare that the submitted exam

Course code:

Course name:

was produced independently by me, without external help.

By signing this statement, I explicitly declare that I am aware of the fraud sanctions as stated in the Education and Examination Regulations (EERs) of Maastricht University.

Place:

Date:

First and last name:

Study programme and year:

ID number:

Signature

# DKE Numerical Mathematics 2019/2020
# Exam Questions

**Programme:** Bachelor Data Science and Knowledge Engineering

**Course codes:** KEN1540

**Examiners:** Dr. P.J. Collins, Dr. G. Stamoulis

**Date/time:** Wednesday June 3$^{\text{rd}}$, 2020, 13.00-16.00h

**Format:** Closed book exam

**Allowed aides:** Pens, simple (non-programmable) calculator from the DKE-list of allowed calculators, formula sheet (provided).

**Instructions to students:**

- The exam consists of 7 questions on 1 pages (excluding the 1 cover page(s)).
- Fill in your name and student ID number on the top-right corner of each side of paper you submit.
- Ensure that you properly motivate your answers.
- Do not use red pens, and write in a readable way. Answers that cannot be read easily cannot be graded and may therefore lower your grade.
- You are not allowed to have a communication device within your reach, nor to wear or use a watch.
- If you think a question is ambiguous, or even erroneous, and you cannot ask during the exam to clarify this, explain this in detail in with your answer to the question.
- If you have not registered for the exam, your answers will not be graded, and thus handled as invalid.
- **Good luck!**

1. (12 points) Use two steps of the secant method to estimate the root of $f(x) = x^3 - x - 1$ in $[1, 2]$, starting at $p_0 = 2$ and $p_1 = 1$. Compute $f(p_3)$, and give the best bracket for the root you have found, an estimate of the error, and the residual.

2. (12 points) Use three steps of Ralston's method to estimate the solution of the initial value problem $\dot{y} = y - t/y$, $y(1) = 1.400$ up to time $t = 2$. You should aim to compute to solution using Ralston's method to an accuracy of at least 3 decimal places, and use sufficient precison in your working to do this.

   Compare your answer with the exact solution, which has $y(2) = 2.4287786654\,(10\text{ dp})$. What would you expect the absolute error to be if you were to use Ralston's method 12 steps?

3. (12 points) Use divided differences to compute the polynomial interpolating the following data:

| $x_i$ | 1.0 | 1.5 | 0.5 | 2.0 |
|---|---|---|---|---|
| $y_i$ | 0.761 | 0.477 | 0.620 | 0.169 |

   Estimate the value of $y$ when $x = 0.8$. Provide a bound on the error of your estimate if the data comes from a function $f$ satisfying $f'''(x) \leq 4$ for $x \in [0, 3]$

4. (12 points) Use the trapezoid rule with $n = 6$ partitions to approximate

$$\int_{0.2}^{1.4} \frac{1}{1 + x^3}\, dx.$$

   Use the trapezoid rule error estimate to estimate the error on each of the intervals $[0.2 : 0.6]$, $[0.6 : 1.0]$ and $[1.0 : 1.4]$. What is your estimate of the total error? On which inteval should you subdivide in order to reduce the error the most?

5. (12 points) Compute the first mode $s_1$ of the discrete Fourier transform for the following $n = 6$ data points over a time interval of length $T = 3$:

| $t_j$ | 0.0 | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 |
|---|---|---|---|---|---|---|
| $y_j$ | 0.017 | 0.620 | 0.761 | 0.477 | 0.169 | 0.038 |

   Hence estimate the value of $y$ when $t = 0.8$.

   *Note:* The discrete Fourier transform is given by

$$s_m(t) = \frac{a_0}{2} + \sum_{k=1}^{m} a_1 \cos(2\pi kt/T) + b_1 \sin(2\pi kt/T)$$

   where

$$a_k = \frac{2}{n} \sum_{i=0}^{n-1} y_j \cos\left(\tfrac{2\pi k}{T} t_j\right) = \frac{2}{n} \sum_{i=0}^{n-1} y_j \cos(2\pi jk/n), \quad b_k = \frac{2}{n} \sum_{i=0}^{n-1} y_j \cos(2\pi jk/n)\,.$$

6. (12 points) Let

$$A = \begin{pmatrix} 7 & 3 \\ 3 & 2 \end{pmatrix}, \quad \mu^{(0)} = 2, \quad \mathbf{x}^{(0)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

   Use two steps of the inverse power method to estimate the smallest eigenvalue of $A$ and its corresponding eigenvector.

7. (8 points) In a compartmental model of an infection disease, individuals are either susceptible (S), exposed (E), infectious (I) or recovered (R). The progress of the disease is modelled by

$$\dot{S} = -(\alpha E + \beta I)S/N, \quad \dot{E} = (\alpha E + \beta I)S/N - (\gamma + \delta)E, \quad \dot{I} = \gamma E - \epsilon I, \quad R = \delta E + \epsilon I.$$

   Assume initally 1 exposed individual in a population of $N = 17\,000\,000$.

   Show how to solve this system of differential equations in Matlab, including writing the code you would use.

# DKE KEN1540 Numerical Mathematics 2020 Formula Sheet

## Algebraic equations

**Secant method**

$$p_{n+1} = p_n - \frac{p_n - p_{n-1}}{f(p_n) - f(p_{n-1})} f(p_n)$$

**Newton's method** $p_{n+1} = p_n - f(p_n)/f'(p_n)$

## Polynomial Interpolation

**Lagrange Form** Interpolating polynomial

$$p(x) = \sum_{i=0}^{n} y_i l_i(x) \text{ where } l_i(x) = \prod_{\substack{j \neq i \\ j=0}}^{n} \left( \frac{x - x_j}{x_i - x_j} \right)$$

**Divided Differences** Recurrence relation

$$f[x_i, \dots, x_j] = \frac{f[x_{i+1}, \dots, x_j] - f[x_i, \dots, x_{j-1}]}{x_j - x_i}$$

**Theorem (Divided differences and derivatives)** If $f^{(n)}$ is continous on $[a, b]$ and $x_0, \dots, x_n$ are distinct points in $[a, b]$, there exists $\xi \in [a, b]$ with

$$f[x_0, \dots, x_n] = f^{(n)}(\xi)/n!$$

**Nested Form** Let $a_k = f[x_0, \dots, x_k]$. Then

$$p(x) = a_0 + (x - x_0)\big(a_1 + (x - x_1)\big(a_2 + \cdots + (x - x_{n-2})\big(a_{n-1} + (x - x_{n-1})a_n\big)\big)\big)$$

**Chebyshev nodes** $x_k = \frac{a+b}{2} - \frac{b-a}{2} \cos\big(\frac{2k+1}{2(n+1)}\pi\big)$ for $k = 0, \dots, n$.

**Theorem (Error of polynomial interpolation)** If $p$ is the degree-$n$ polynomial of interpolating $f$ at the $n+1$ distinct nodes $x_0, x_1, \dots, x_n$ in $[a, b]$, and $f^{(n+1)}$ is continuous, then for each $x$ in $[a, b]$, there is a $\xi$ in $(a, b)$ for which

$$f(x) - p(x) = \frac{1}{(n+1)!} f^{n+1}(\xi) \prod_{i=0}^{n} (x - x_i)$$

**Theorem (Maximum error of polynomial interpolation)** If $p$ is the interpolating polynomial of $f$ with $n+1$ equally-spaced nodes, then for any $x \in [a, b]$,

$$|f(x) - p(x)| \leq \frac{(b-a)^{n+1}}{4n^{n+1}(n+1)} \max_{\xi \in [a,b]} |f^{(n+1)}(\xi)|$$

If $p$ is the interpolating polynomial of $f$ with $n+1$ Chebyshev nodes, then for any $x \in [a, b]$,

$$|f(x) - p(x)| \leq \frac{(b-a)^{n+1}}{2^{2n+1}(n+1)!} \max_{\xi \in [a,b]} |f^{(n+1)}(\xi)|$$

## Differential Equations

**Euler's method** Local error $O(h^2)$. $w_{i+1} = w_i + h_i f(t_i, w_i)$.

**Ralston's 2nd-order method** Local error $O(h^3)$.
$w_{i+1} = w_i + \frac{1}{4} h_i \big(f(t_i, w_i) + 3f(t_i + \frac{2}{3}h_i, w_i + \frac{2}{3}h_i f(t_i, w_i))\big)$.
$k_{i,1} = h_i f(t_i, w_i)$; $k_{i,2} = h_i f(t_i + \frac{2}{3}h_i, w_i + \frac{2}{3}k_{i,1})$; $w_{i+1} = w_i + \frac{1}{4}(k_{i,1} + 3k_{i,2})$.

**Kutta's 3rd-order method** Local error $O(h^4)$.
$k_{i,1} = h_i f(t_i, w_i)$; $k_{i,2} = h_i f(t_i + \frac{1}{2}h_i, w_i + \frac{1}{2}k_{i,1})$; $k_{i,3} = h_i f(t_i + h_i, w_i - k_{i,1} + 2k_{i,2})$;
$w_{i+1} = w_i + \frac{1}{6}(k_{i,1} + 4k_{i,2} + k_{i,3})$.

**Classical 4th-order Runge-Kutta method** Local error $O(h^5)$.
$k_{i,1} = h_i f(t_i, w_i)$; $k_{i,2} = h_i f(t_i + \frac{1}{2}h_i, w_i + \frac{1}{2}k_{i,1})$;
$k_{i,3} = h_i f(t_i + \frac{1}{2}h_i, w_i + \frac{1}{2}k_{i,2})$; $k_{i,4} = h_i f(t_i + h_i, w_i + k_{i,3})$;
$w_{i+1} = w_i + \frac{1}{6}(k_{i,1} + 2k_{i,2} + 2k_{i,3} + k_{i,4})$.

**2-stage Adams-Bashforth method** Local error $O(h^3)$.
$w_{i+1} = w_i + (h/2)\big(3f(t_i, w_i) - f(t_{i-1}, w_{i-1})\big)$.

**3-stage Adams-Bashforth method** Local error $O(h^4)$.
$w_{i+1} = w_i + (h/12)\big(23f(t_i, w_i) - 16f(t_{i-1}, w_{i-1}) + 5f(t_{i-2}, w_{i-2})\big)$.

**2-stage Adams-Moulton method** Local error $O(h^4)$.
$w_{i+1} = w_i + (h/12)\big(5f(t_{i+1}, w_{i+1}) + 8f(t_i, w_i) - f(t_{i-1}, w_{i-1})\big)$.

**Backward Euler method** $w_{i+1} = w_i + h_i f(t_{i+1}, w_{i+1})$.

**2-stage backward-difference method** Local error $O(h^4)$.
$w_{i+1} = \frac{4}{3}w_i - \frac{1}{3}w_{i-1} + \frac{2}{3}hf(t_{i+1}, w_{i+1})$.

**Bogacki-Shampine adaptive method** Local error $O(h^4)$.
$k_{i,1} = hf(t_i, w_i)$; $k_{i,2} = hf(t_i + \frac{1}{2}h, w_i + \frac{1}{2}k_{i,1})$; $k_{i,3} = hf(t_i + \frac{3}{4}h, w_i + \frac{3}{4}k_{i,2})$;
$w_{i+1} = w_i + \frac{2}{9}k_{i,1} + \frac{3}{9}k_{i,2} + \frac{4}{9}k_{i,3}$;
$k_{i,4} = hf(t_i + h, w_{i+1})$; $\hat{w}_{i+1} = w_i + \frac{7}{24}k_{i,1} + \frac{1}{4}k_{i,2} + \frac{1}{3}k_{i,3} + \frac{1}{8}k_{i,4}$.
Set $q = (\epsilon h / |w_{i+1} - \hat{w}_{i+1}|)^{1/2}$. Step size estimate $qh$.

## Differentiation

**Two-point forward difference** $f'(x) = (f(x+h) - f(x))/h - f''(\xi)h/2$.

**Three-point centred difference** $f'(x) = (f(x+h) - f(x-h))/2h - f'''(\xi)h^2/6$.

**Three-point forward difference**
$f'(x) = (-3f(x) + 4f(x+h) - f(x+2h))/2h + f'''(\xi)h^2/3$.

**Five-point centred difference**
$f'(x) = \big(f(x-2h) - 8f(x-h) + 8f(x+h) + f(x+2h)\big)/12h + f^{(5)}(\xi)h^4/30$.

**Second derivative** $f''(x) = \big(f(x+h) - 2f(x) + f(x-h)\big)/h^2 - f^{(4)}(\xi)\, h^2/12$.

# Integration

**Midpoint rule** $M_n(f;a,b) = h\big(f(x_{1/2}) + f(x_{1+1/2}) + \cdots + f(x_{n-1/2})\big)$
Error $\int_a^b f(x)dx - M(f;P) = (b-a)\,h^2\,f^{(2)}(\xi)/24$.

**Trapezoid rule** $T_n(f;a,b) = h\big(\frac{1}{2}f(x_0) + f(x_1) + \cdots + f(x_{n-1}) + \frac{1}{2}f(x_n)\big)$.
Error $\int_a^b f(x)dx - T(f;P) = -(b-a)\,h^2\,f^{(2)}(\xi)/12$.

**Simpson's rule**
$S_n(f;a,b) = h\big(f(x_0) + 4f(x_1) + 2f(x_2) + \cdots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)\big)/3$.
Error $\int_a^b f(x)dx - S(f;P) = -(b-a)\,h^4\,f^{(4)}(\xi)/180$.

**Romburg integration** $R_{k,0} = T_{2^k}$; $\quad R_{k,j} = R_{k,j-1} + (R_{k,j-1} - R_{k-1,j-1})/(4^j - 1)$.

**Adaptive trapezoid rule** Error estimate
$\big|\int_a^b f(x)\,dx - T_2(f;a,b)\big| \lessapprox \frac{1}{3}\big|T_2(f;a,b) - T_1(f;a,b)\big| = \frac{b-a}{12}\big|f(a) - 2f(\frac{a+b}{2}) + f(b)\big|$.

# Least-Squares Approximation

**Linear least squares** $g(x) = ax + b$ fitting data $(x_1, y_1), \ldots, (x_m, y_m)$ where
$$a = (\overline{XY} - \overline{X}\cdot\overline{Y})/(\overline{X^2} - \overline{X}^2); \quad b = \overline{Y} - a\,\overline{X}.$$

**Nonlinear least squares** If $g(x) = \sum_{i=0}^n c_i\,\phi_i(x)$ then for $i = 0, \ldots, n$:
$$\textstyle\sum_{j=0}^n c_j \sum_{k=1}^m \phi_i(x_k)\,\phi_j(x_k) = \sum_{k=1}^m \phi_i(x_k)\,y_k.$$

**Continuous least-squares** $g(x) = \sum_{i=0}^n c_i\phi_i(x)$
$$\textstyle\sum_{j=0}^n c_j \int_a^b \phi_i(x)\,\phi_j(x)\,dx = \int_a^b \phi_i(x)\,f(x)\,dx.$$

**Orthogonal basis** $\int_a^b w(x)\,\phi_i(x)\,\phi_j(x)\,dx = 0$ for $i \neq j$.

**Least-squares approximation in orthogonal basis** $g(x) = \sum_{i=0}^n c_i\phi_i(x)$
with $c_i = (\int_a^b w(x)\,\phi_i(x)\,f(x)\,dx)/(\int_a^b w(x)\,(\phi_i(x))^2\,dx)$

**Square error for orthogonal functions** With $\alpha_k = \int_a^b w(x)\phi_k(x)^2\,dx$,
$$\textstyle\int_a^b w(x)(f(x) - g(x))^2\,dx = \int_a^b w(x)f(x)^2\,dx - \sum_{k=0}^n \alpha_k c_k^2.$$

**Generating orthogonal polynomials** $\phi_k(x) = (x - B_k)\phi_{k-1}(x) - C_k\phi_{k-2}(x)$
$$\text{with } B_k = \frac{\int_a^b x\,w(x)\,(\phi_{k-1}(x))^2\,dx}{\int_a^b w(x)\,(\phi_{k-1}(x))^2\,dx}, \quad C_k = \frac{\int_a^b x\,w(x)\,\phi_{k-1}(x)\,\phi_{k-2}(x)\,dx}{\int_a^b w(x)\,(\phi_{k-2}(x))^2\,dx}.$$

**Legendre polynomials**
$$P_0(x) = 1; \quad P_1(x) = x; \quad P_k(x) = (2 - 1/k)xP_{k-1}(x) - (1 - 1/k)P_{k-2}(x).$$
$$\textstyle\int_{-1}^1 P_i(x)\,P_j(x)\,dx = 0,\ i \neq j; \quad \int_{-1}^{+1} P_k^2(x)\,dx = \frac{2}{2k+1}.$$

**Chebyshev polynomials**
$$T_0(x) = 1;\ T_1(x) = x;\ T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x); \quad T_k(x) = \cos(k\,\mathrm{acos}(x)).$$
$$\int_{-1}^1 \frac{T_i(x)\,T_j(x)}{\sqrt{1-x^2}}\,dx = 0,\ i \neq j; \quad \int_{-1}^1 \frac{T_i(x)^2}{\sqrt{1-x^2}} = \begin{cases}\pi, & i = 0,\\ \pi/2, & i \neq 0\end{cases}$$

**Fourier series**
$$f(x) \approx \frac{a_0}{2} + \sum_{k=1}^n a_k\cos(kx) + b_k\sin(kx);$$
$$a_k = \frac{1}{\pi}\int_{-\pi}^{\pi}\cos(kx)f(x)\,dx; \quad b_k = \frac{1}{\pi}\int_{-\pi}^{\pi}\sin(kx)f(x)\,dx.$$

**Discrete Fourier transform** If $x_j = (j/m - 1)\pi$ for $j = 0, 1, \ldots, 2m-1$, then
$$a_k = \frac{1}{m}\sum_{j=0}^{2m-1} f(x_j)\cos(kx_j), \quad b_k = \frac{1}{m}\sum_{j=0}^{2m-1} f(x_j)\sin(kx_j).$$

# Linear Algebraic Equations

**Conditioning** $K(A) = ||A|| \cdot ||A^{-1}||$;
$$||\mathbf{x} - \tilde{\mathbf{x}}|| \leq ||A^{-1}|| \cdot ||\mathbf{b} - A\tilde{\mathbf{x}}||; \qquad \frac{||\mathbf{x} - \tilde{\mathbf{x}}||}{||\mathbf{x}||} \leq K(A)\,\frac{||\mathbf{b} - A\tilde{\mathbf{x}}||}{||\mathbf{b}||}$$

**LU factorization** $PA = LU$; $A\mathbf{x} = \mathbf{b}$ if $L\mathbf{y} = P\mathbf{b}$, $U\mathbf{x} = \mathbf{y}$.

**Cholesky factorization** $S = U^T\,U = L\,L^T$.

**Jacobi method** $x_i^{(n+1)} = \big(b_i - \sum_{j\neq i} a_{i,j}x_j^{(n)}\big)/a_{i,i} = \mathbf{x}^{(n)} - D^{-1}\big(A\mathbf{x}^{(n)} - \mathbf{b}\big)$

**Gauss-Seidel method** $x_i^{(n+1)} = \big(b_i - \sum_{j<i} a_{i,j}x_j^{(n+1)} - \sum_{j>i} a_{i,j}x_j^{(n)}\big)/a_{i,i}$

**Successive over-relaxation**
$x_i^{(n+1)} = (1-\omega)x_i^{(n)} + \omega\big(b_i - \sum_{j<i} a_{i,j}x_j^{(n+1)} - \sum_{j>i} a_{i,j}x_j^{(n)}\big)/a_{i,i}$

**Conjugate gradient method** $\mathbf{v}^{(0)} = \mathbf{x}^{(0)} = \mathbf{0}$, $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$;
$\mathbf{v}^{(k+1)} = \mathbf{r}^{(k)} + \big(\langle\mathbf{r}^{(k)}, \mathbf{r}^{(k)}\rangle/\langle\mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)}\rangle\big)\mathbf{v}^{(k)}$, $\quad t^{(k)} = \langle\mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)}\rangle/\langle\mathbf{v}^{(k)}, A, \mathbf{v}^{(k)}\rangle$,
$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + t^{(k)}\mathbf{v}^{(k)}$, $\quad \mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} - t_k A\mathbf{v}^{(k)}$.

# Orthogonality and Eigenvalues

**Gram-Schmidt orthogonalisation**
$r_{i,i} = ||\mathbf{v}_i||_2$, $r_{j,i} = \mathbf{u}_j \cdot \mathbf{x}_i$ $(j < i)$, $\mathbf{v}_i = \mathbf{x}_i - \sum_{j=1}^{i-1} r_{j,i}\mathbf{u}_j$, $\mathbf{u}_i = \mathbf{v}_i/r_{i,i}$.

**Power method** $\mathbf{y}^{(n)} = A\mathbf{x}^{(n)}$, $\mathbf{x}^{(n+1)} = \mathbf{y}^{(n)}/||\mathbf{y}^{(n)}||$; $\quad \mathbf{y}^{(n)}/\mathbf{x}^{(n)} \to \lambda_{\max}$.

**Inverse power method** $\mathbf{y}^{(n)} = (A - \mu I)^{-1}\mathbf{x}^{(n)}$, $\mathbf{x}^{(n+1)} = \mathbf{y}^{(n)}/||\mathbf{y}^{(n)}||$.

**Householder matrix** $H = I - 2\mathbf{v}\mathbf{v}^T/\mathbf{v}^T\mathbf{v}$; $\quad H = I - 2\mathbf{u}\mathbf{u}^T$ if $||\mathbf{u}||_2 = 1$.

**Givens rotation ($\mathbf{2} \times \mathbf{2}$)** $G = \begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix} = \begin{pmatrix}a & -b\\ b & a\end{pmatrix}$ where $a^2 + b^2 = 1$.

**Upper Hessenburg** If $\alpha = \big(\sum_{i=2}^n a_{i,1}^2\big)^{1/2}$, $v_1 = 0$, $v_2 = a_{2,1} \pm \alpha$, $v_i = a_{i,1}$ $(i \geq 3)$,
$H = I - 2\mathbf{v}\mathbf{v}^T/\mathbf{v}^T\mathbf{v}$, $A' = H^T AH$. Then $A'_{i,1} = 0$ for $i \geq 3$.

## SHORT PROTOCOL – FOR EXAM SESSION

1. You have two hours to make the exam from the moment the invigilator announces the start. During this time, you are not allowed to leave the room or communicate with the invigilator. In case of issues – like a failing internet connection – you are allowed to continue your exam, and the invigilator will not interrupt you.

2. Write down your solutions using pen and paper, taking care to number the pages on the top left corner and leaving space for your student ID on the top right corner.

3. When the invigilator announces the end of the exam (or: when you have raised your hand >30 minutes before the end of the exam, and received permission from the invigilator), switch your desktop/laptop screen sharing to video sharing.

4. Upload your exam solutions and signed declaration of integrity, taking care to:

    a. Place your student ID on each page
    b. Upload all documents in one single batch
    c. Check that the upload was successful

5. In case you experience technical issues during uploading, you are allowed to submit your exam by email. Notify your invigilator and send your files to: **DKE-PROCTORED@maastrichtuniversity.nl.** Indicate your full name, the name of the exam, and put your invigilator in cc!

6. Wait until the invigilator announces the end of the exam. If called on by the invigilator, present the requested page of your exam to the camera.

7. You are obliged to keep the physical (paper) copy of your exam for 2 years. During this period, we may ask you to show it.