LINFO1113 - Cheat Sheet - 2019-2020

- Numbers and Error propagation
  - Float32: $(-1)^{b_{31}} \cdot (\mathbf{1}.b_{22} \dots b_0)_2 \cdot 2^{(b_{30} \dots b_{23})_2 - 127}$
  - $fl(x) = x(1 + \varepsilon)$ with $|\varepsilon| \leq 2^{-p}$ with p=24 (53) for float32 (float64)
  - $e = |\tilde{x} - x| = |fl(x) - x| \leq 2^{-p\,2t}$; $\varepsilon = \frac{|\tilde{x} - x|}{|x|} \leq |fl(x) - x| \leq 2^{-p}$
- Linear Systems ($\boldsymbol{Ax} = \boldsymbol{b}$)
  - Doolitle: $\boldsymbol{A} = \boldsymbol{LU}$ with $L_{ii} = 1$
  - Choleski: $\boldsymbol{A} = \boldsymbol{LU}$ with $\boldsymbol{L} = \boldsymbol{U^T}$; $\boldsymbol{L_{jj}} = \sqrt{\boldsymbol{A_{jj}} - \sum_{k=1}^{j-1} \boldsymbol{L_{jk}^2}}$; $\boldsymbol{L_{ij}} = \left(\boldsymbol{A_{ij}} - \sum_{k=1}^{j-1} \boldsymbol{L_{ik} L_{jk}}\right) / \boldsymbol{L_{jj}}$
  - Jacobi: $\boldsymbol{X_{k+1}} \coloneqq \boldsymbol{D^{-1}}(\boldsymbol{b} - \boldsymbol{RX_k})$
- Curve fitting and interpolation
  - Lagrange: $\mathrm{P_n(x)} = \sum_i \mathrm{y_i}\, \mathrm{l_i(x)}$ with $\mathrm{l_i(x)} = \prod_{j \neq i} \frac{(\mathrm{x - x_j})}{(\mathrm{x_i - x_j})}$
  - Newton: $\mathrm{P_k(x)} = a_{n-k} + (x - x_{n-k}) P_{k-1}(x)$ with $a_0 = y_0$; $a_1 = \nabla y_1$; $a_2 = \nabla^2 y_2$; …
  - Neville: $\mathrm{P_k}[x_i, \dots, x_{i+k}] = \{(x - x_{i+k}) \mathrm{P_{k-1}}[x_i, \dots, x_{i+k-1}] + (x_i - x) \mathrm{P_{k-1}}[x_{i+1}, \dots, x_{i+k}]\} / (x - x_{i+k})$
  - Cubic spine: $k_{i-1} + 4k_i + k_{i+1} = \frac{6}{h^2}(y_{i-1} - 2y_i + y_{i+1})$
    $f_{i,i+1}(x) = \frac{k_i}{6}\left[\frac{(x - x_{i+1})^3}{-h} - (x - x_{i+1})(-h)\right] - \frac{k_{i+1}}{6}\left[\frac{(x - x_i)^3}{-h} - (x - x_i)(-h)\right] + \frac{y_i(x - x_{i+1}) - y_{i+1}(x - x_i)}{-h}$
  - Least-Square: $S = \sum (y_i - f(x_i))^2$; $\sigma = \sqrt{S/(n-m)}$
    In the case: $\mathbf{Y} = f(\mathbf{x}) = \sum_{i=0}^{m} a_m x_m = \mathbf{XA}$ → S = $\mathbf{Y^T Y} + \mathbf{A^T X^T X A} - \mathbf{2 A^T X^T Y}$ → $\mathbf{X^T X A} = \mathbf{X^T Y}$
- Root Search
  - RegulaFalsi /Secant: $x_3 = x_2 - f_2 \frac{(x_2 - x_1)}{(f_2 - f_1)}$ Rider: $x_4 = x_3 \pm (x_3 - x_1) \frac{f_3}{\sqrt{f_3^2 - f_1 f_2}}$
  - Newton-Raphson: $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$ or for a system: $\boldsymbol{x} \leftarrow (\boldsymbol{x} + \mathbf{\Delta x})$ with $\mathbf{J}(\boldsymbol{x}) \mathbf{\Delta x} = -\boldsymbol{f}(\boldsymbol{x})$
- Numerical differentiation
  - Be able to derivate first/second (non-)central approximation for f', f'' and associated error
  - Richardson: IF G = g(h) + E(h) with $E(h) = ch^p$ and $h_2 = h_1/2$ then $G = \frac{2^p g(h_1/2) - g(h_1)}{2^p - 1}$
- Numerical Integration
  - Newton-Cotes: $I \approx \sum_{i=0}^{n} A_i f(x_i)$ with $A_i = \int_a^b l_i(x) dx$, trapzoid (n=1), simpson 1/3 (n=2) 3/8 (n=3)
  - Composite Trapezoidal: I = $\frac{h}{2}[f(x_0) + 2f(x_1) + \dots + 2f(x_{n-1}) + f(x_n)]$
  - Recursive Trapezoidal: $I_k = \frac{1}{2} I_{k-1} + \frac{H}{2^{k-1}} \sum_{i=1}^{2^{k-2}} f\left(a + \frac{(2i-1)H}{2^{k-1}}\right)$
  - Simpson 1/3: $\frac{h}{3}[f(x_i) + 4f(x_{i+1}) + f(x_{i+2})]$; Simpson 3/8: $\frac{3h}{8}[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)]$
  - Romberg: $R_{i,j} = \frac{4^{j-1} R_{i,j-1} - R_{i-1,j-1}}{4^{j-1} - 1}$
- Initial Value Problem
  - Euler: $\boldsymbol{y}(x + h) \approx \boldsymbol{y}(x) + \boldsymbol{y}'(x) h$
  - Modified Euler (RK2): $\boldsymbol{y}(x + h) \approx \boldsymbol{y}(x) + h\boldsymbol{F}\left(x + \frac{h}{2}, \boldsymbol{y} + \frac{h}{2} \boldsymbol{F}(x, \boldsymbol{y})\right)$
  - RK4: $\boldsymbol{y}(x + h) \approx \boldsymbol{y}(x) + \frac{1}{6}(\boldsymbol{K}_0 + 2\boldsymbol{K}_1 + 2\boldsymbol{K}_2 + \boldsymbol{K_3})$ with
    $\boldsymbol{K_0} = h\boldsymbol{F}(x, \boldsymbol{y})$; $\boldsymbol{K_1} = h\boldsymbol{F}\left(x + \frac{h}{2}, \boldsymbol{y} + \frac{\boldsymbol{K_0}}{\mathbf{2}}\right)$; $\boldsymbol{K_2} = h\boldsymbol{F}\left(x + \frac{h}{2}, \boldsymbol{y} + \frac{\boldsymbol{K_1}}{\mathbf{2}}\right)$; $\boldsymbol{K_3} = h\boldsymbol{F}(x + h, \boldsymbol{y} + \boldsymbol{K_2})$
- Optimization
  - Unconstrained: $F^*(\boldsymbol{x}) = F(\boldsymbol{x}) + \lambda\, P(\boldsymbol{x})$ with $P(\boldsymbol{x}) = \sum_{\boldsymbol{i}}^{\boldsymbol{M}} (g_i(\boldsymbol{x}))^{\mathbf{2}} + \sum_{\boldsymbol{j}}^{\boldsymbol{N}} \max(0, h_j(\boldsymbol{x}))^{\mathbf{2}}$
  - Golden Search: $x_1 = b - Rh$; $x_2 = a + Rh$; $R = \frac{-1 + \sqrt{5}}{2}$
  - Gradient Descent: $\boldsymbol{x} \leftarrow \boldsymbol{x} - \eta\, \boldsymbol{\nabla} f(\boldsymbol{x})$