Numerical Analysis Midterm Cheat sheet

1. Computational Errors

True Error $E_T = True\ value - Approximation$

True Percent Relative Error

$$\varepsilon_t = \left|\frac{True\ value - Approximation}{True\ value}\right| \times 100\%$$

Approximate Relative Error

$$\varepsilon_a = \left|\frac{Current\ approx. - Previous\ approx.}{Current\ approx.}\right| \times 100\%$$

Error Tolerance $\varepsilon_s = \left(0.5 \times 10^{(2-n)}\right)\%$

$$n = least\ number\ of\ correct\ significant\ digits$$

Maclurin Series Expansions

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

$$\left\{\sin x = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots\right\} \left\{\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots\right\}$$

Taylor Series

$$f(x_{i+1}) \cong f(x_i) + \frac{f^{`}(x_i)}{1!}(x_{i+1} - x_i) + \frac{f^{``}(x_i)}{2!}(x_{i+1} - x_i)^2 + \cdots + \frac{f^{(n)}(x_i)}{n!}(x_{i+1} - x_i)^n + R_n$$

Remainder Term $(n+1 \rightarrow \infty)$

$$R_n = \frac{f^{(n+1)}(\varepsilon)}{(n+1)!}h^{(n+1)} \quad where \quad h = (x_{i+1} - x_i)$$

Series tests $\left\{\int_1^{\infty} U_n \frac{=finite\ conv.}{=\pm\infty\ div.}\right\}$

$\left\{Partial\ sum: \lim_{n\to\infty} S_N \frac{=finite\ conv.}{=\pm\infty\ div.}\right\}$ $\left\{\lim_{n\to\infty} U_n \frac{\neq 0\ Div}{=0\ Fail}\right\}$

$\left\{Alternating\ series\ \sum_{n=1}^{\infty}(-1)^n b_n : \lim_{n\to\infty} b_n \frac{\neq 0\ Div}{=0\ Conv.}\right\}$

Error Propagation

Assuming $\tilde{x}$ is an approximation of x

Estimate of the error of the function:

$$\Delta f(\tilde{x}) = |f(x) - f(\tilde{x})|$$

$$\Delta f(\tilde{x}) = \left|f^{`}(\tilde{x})\right|(x - \tilde{x})$$

Estimate of the error of x: $\Delta\tilde{x} = |x - \tilde{x}|$

Absolute Error

$$|x - fl(x)| \leq \beta^{e-n}, in\ chopping$$
$$|x - fl(x)| \leq \frac{1}{2}\beta^{e-n}, in\ rounding$$

Relative Error

$$\left|\frac{x - fl(x)}{x}\right| \leq \beta^{1-n}, in\ chopping$$
$$\left|\frac{x - fl(x)}{x}\right| \leq \frac{1}{2}\beta^{1-n}, in\ rounding$$

- $fl(x) - rounded\ or\ chopped\ value\ of\ x$
- $\beta - base$
- $n - \#\ of\ digits\ in\ the\ mantissa$

Propagation of Errors:

$Total\ error = propagated\ error + rounding\ error$

- $\swarrow$ - operation between x_T and y_T
- $\swarrow^*$- corresponding operation carried out by computer

$$(x_T \swarrow y_T) - (x_A \swarrow^* y_A) =$$
$$((x_T \swarrow y_T) - (x_A \swarrow y_A)) + ((x_A \swarrow y_A) - (x_A \swarrow^* y_A))$$

$$E_{x+y} = (x_T + y_T) - (x_A + y_A) = E_x + E_y$$

$$E_{x-y} = (x_T - y_T) - (x_A - y_A) = E_x - E_y$$

$$\varepsilon_{xy} = \varepsilon_x + \varepsilon_y \qquad \varepsilon_{x/y} = \varepsilon_x - \varepsilon_y$$

$$E - True\ error \qquad \varepsilon - Relative\ error$$

Floating Point Representation

$$\pm(0.a_1a_2a_3 \ldots a_n)_\beta \times \beta^e, with\ a_1 \neq 0$$

- $\pm(0.a_1a_2a_3 \ldots a_n)_\beta$ – mantissa (fractional part)
- $e - exponent\ (characteristic)$
- $\beta - base\ (radix)$

2. System of equations

Forward Elimination:

$$row_i = (row_i) - (m_{ik} * row_k)$$
$$m_{ik} = a_{ik} / a_{kk}$$
$$where\ row_k\ is\ the\ pivot\ coeffiecient\ row$$

Tridiagonal Systems:

$$\begin{bmatrix} f_1 & g_1 & & \\ e_2 & f_2 & g_2 & \\ & e_3 & f_3 & g_3 \\ & & e_4 & f_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}$$

For LU: for(k, 2→n) $\{\boldsymbol{e_k = e_k / f_{k-1}} \quad \boldsymbol{f_k = f_k - e_k g_{k-1}}\}$

Forward substitution: $\boldsymbol{d_1 = r_1}$

for(k, 2→n) $\{\boldsymbol{d_k = r_k - e_k d_{k-1}}\}$

Backward substitution: $\boldsymbol{x_n = d_n / f_n}$

for(k, n-1→1) $\{\boldsymbol{x_k = (d_k - g_k x_{k+1}) / f_k}\}$

Cholesky decomposition (LU for symmetric arrays):

$$\boldsymbol{l_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2}\ ,\ \ l_{ki} = \frac{a_{ki} - \sum_{j=1}^{i-1} l_{ij} * l_{kj}}{l_{ii}}} \quad \text{←for(i, 1, k-1)}$$

Jacobi method: ***intial guess***: $\boldsymbol{x_1^k,\ x_2^k,\ x_3^k}$

$$\boldsymbol{x_1^k = \frac{b_1 - a_{12}x_{2\ old}^{k-1} - a_{13}x_{3\ old}^{k-1}}{a_{11}}} \qquad \boldsymbol{x_2^k = \frac{(b_2 - a_{21}x_{1\ old}^{k-1} - a_{23}x_{3\ old}^{k-1})}{a_{22}}}$$

$$\boldsymbol{x_3^k = \frac{(b_3 - a_{31}x_{1\ old}^{k-1} - a_{32}x_{2\ old}^{k-1})}{a_{33}}}$$

Gauss-Seidel method: as Jacobi but each new x is used in the next formula

3. Curve fitting

Standard deviation $S_y = \sqrt{\frac{\sum(y_i - \bar{y})^2}{n-1}}$

Variance $v = \frac{\sum(y_i - \bar{y})^2}{n-1}$

Coefficient of variation $c.v. = \frac{S_y}{\bar{y}} * 100\%$

Residual $\varepsilon_i = y_i - f(x_i)$

Sum of residuals squared

$$S_r = \sum_{i=1}^{n}(\varepsilon_i)^2 = \sum_{i=1}^{n}(y_i - a_1 x_i - a_0)^2$$

Linear regression for $y = a_1 x + a_0$ $\{\boldsymbol{a_0 = \bar{y} - a_1 \bar{x}}\}$

$\{a_1 = \frac{n\sum y_i x_i - \sum y_i \sum x_i}{n\sum({x_i}^2) - (\sum x_i)^2}\}$ $\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum({x_i}^2) \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i x_i \end{bmatrix}$

Goodness of a fit $r^2 = \frac{S_t - S_r}{S_t}$

$\{S_r - \text{Sum of residuals squared}\}$ $\{S_t - \sum(y_i - \bar{y})^2\}$

$\{Perfect\ fit: \ r = 1, \ S_r = 0\}$
$\{No\ improvement: \ r = 0, \ S_r = S_t\}$

$\{r - correlation\ cofficient\}$ $\{r^2 - cofficient\ of\ determination\}$

Regression for exponential equations

$y = \alpha e^{\beta x}$ $\ln(y) = \ln(\alpha) + \beta x \leftarrow$ *continue as linear regression*

Regression for power equations

$y = a x^b$ $\log(y) = \log(a) + b\log(x) \leftarrow$ *continue as linear regression*

Regression for saturation growth rate equations

$y = a\frac{x}{b+x}$ $\frac{1}{y} = \frac{1}{a} + \frac{b}{ax} \leftarrow$ *continue as linear regression*

Polynomial regression

$$\begin{bmatrix} n & \sum x_i & \cdots & \sum x_i^m \\ \sum x_i & \sum x_i^2 & \cdots & \sum x_i^{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum x_i^m & \sum x_i^{m+1} & \cdots & \sum x_i^{m+m} \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i x_i \\ \vdots \\ \sum y_i x_i^m \end{bmatrix}$$

Multiple Linear regression

$$\begin{bmatrix} n & \sum x_{1i} & \sum x_{2i} \\ \sum x_{1i} & \sum x_{1i}^2 & \sum x_{1i}x_{2i} \\ \sum x_{2i} & \sum x_{1i}x_{2i} & \sum x_{2i}^2 \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i x_{1i} \\ \sum y_i x_{2i} \end{bmatrix}$$

Linear interpolation

$$f(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

Quadratic interpolation

$f(x_2) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$

$\{b_0 = f(x_0)\}$ $\{b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}\}$

$\{b_2 = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}\}$

Newton's interpolating general form

$$\boldsymbol{f}[x_n, x_{n-1}, \ldots, x_1, x_0] = \frac{\boldsymbol{f}[x_n, x_{n-1}, \ldots, x_1] - \boldsymbol{f}[x_{n-1}, x_{n-2}, \ldots, x_0]}{x_n - x_0}$$

Newton's interpolating error

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)(x - x_1)\ldots(x - x_n) = f_{n+1}(x) - f_n(x)$$
$$\cong f[x_{n+1}, x_n, \ldots, x_1, x_0](x - x_0)(x - x_1)\ldots(x - x_n)$$

Lagrange's interpolating general form

$f_{n(x) = \sum_{i=0}^{n} L_i(x) f(x_i)}$ $L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$

$$f_2(x) = \frac{(x - x_1)}{(x_0 - x_1)}\frac{(x - x_2)}{(x_0 - x_2)}f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)}\frac{(x - x_2)}{(x_1 - x_2)}f(x_1) + \frac{(x - x_0)}{(x_2 - x_0)}\frac{(x - x_1)}{(x_2 - x_1)}f(x_2)$$

Coefficients of interpolating polynomials

$$\begin{bmatrix} 1 & x_0 & \ldots & x_0^n \\ 1 & x_1 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}$$

4. Eigenvalues and eigenvectors

$$\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \lambda_i = \boldsymbol{tr(A)}$$

$$\prod_{\boldsymbol{i=1}}^{\boldsymbol{n}} \lambda_i = \boldsymbol{det(A)}$$

Power method $\boldsymbol{intial\ guess: z^{(1)}}$ $\boldsymbol{w^{(1)} = Az^{(1)}}$

$\boldsymbol{\lambda_k^{(1)} = w_k^{(1)}}$ $\boldsymbol{where: w_k^{(1)} = max\left(w^{(1)}\right)}$ $\boldsymbol{z^{(2)} = \frac{w^{(1)}}{\lambda_k^{(1)}}}$

$\boldsymbol{Continue\ until}$: $\boldsymbol{Norm = \|Ax - \lambda x\|\ to\ be < tol.}$ **Or given iterations**

Orthogonal diagonalization:

$$\boldsymbol{P = [p_1\ p_2 \ldots p_n]}$$

First find eigenvectors, if some eigenvalues are found 1 time only then $p = \frac{v}{\|v\|}$

Else $p_n = \frac{w_n}{\|w_n\|}$ where $w_1 = v_1$

$$w_n = v_n - \frac{v_n \ . \ w_{n-1}}{w_{n-1} \ . \ w_{n-1}} w_{n-1}$$

$the\ (.)\ is\ the\ dot\ product\ of\ the\ 2\ vectors$

Order of convergence

$Assume\ f(h) = p(h) + O(h^n)\ and\ g(h) = q(h) + O(h^m), r = \min[m, n]$

$\{f(h) + g(h) = p(h) + q(h) + O(h^r)\}$, $\{f(h)g(h) = p(h)q(h) + O(h^r)\}$

$if\ p(h)\ is\ a\ taylor\ expansion\ of\ v\ terms, then\ O(h^n) = O(h^{v+1})$