Exam 2 Equation Sheet

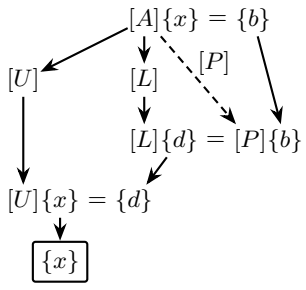
Absolute: E — Approximate: a
Relative (percent): ε — True: t
Specified: s

function GAUSS_PIVOT($[A]$, $\{b\}$)
  for $k \leftarrow 1$ **to** $n-1$ **do**
    $p \leftarrow \text{argmax}_{i \geq k} |a_{ik}|$
    swap rows k and p of $[A]$ and $\{b\}$
    for $i \leftarrow k+1$ **to** n **do**
      $f \leftarrow a_{ik}/a_{kk}$
      for $j \leftarrow k$ **to** n **do**
        $a_{ij} \leftarrow a_{ij} - f\, a_{kj}$
      $b_i \leftarrow b_i - f\, b_k$
  $x_n \leftarrow b_n/a_{nn}$
  for $i \leftarrow n-1$ **down to** 1 **do**
    $x_i \leftarrow \left(b_i - \sum_{j=i+1}^{n} a_{ij}\, x_j\right)/a_{ii}$
  return $\{x\}$

$$\frac{2}{3}n^3 + O(n^2) \text{ flops}$$

$$\sum_{i=k}^{m} a\, f(i) = a \sum_{i=k}^{m} f(i)$$
$$\sum_{i=k}^{m} [f(i) + g(i)] = \sum_{i=k}^{m} f(i) + \sum_{i=k}^{m} g(i)$$
$$\sum_{i=1}^{m} 1 = m \qquad \sum_{i=k}^{m} 1 = m - k + 1$$
$$\sum_{i=1}^{m} i = \frac{m(m+1)}{2}$$
$$\sum_{i=1}^{m} i^2 = \frac{m(m+1)(2m+1)}{6}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ f_{21} & 1 & 0 \\ f_{31} & f_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} = [P][A]$$



$$\text{cond}[A] = \|[A]\|\, \|[A]^{-1}\|$$

$$\frac{\|\{\Delta x\}\|}{\|\{x\}\|} \leq \text{cond}[A]\, \frac{\|[\Delta A]\|}{\|[A]\|}$$

$$\frac{\|\{\Delta x\}\|}{\|\{x\}\|} \leq \text{cond}[A]\, \varepsilon$$

2.22×10^{-16} $\qquad$ 1.19×10^{-7}

$E_a = \frac{\Delta x}{n-1}$ $\qquad$ $E_a = \frac{\Delta x}{2^n}$

$n = \frac{\Delta x}{E_s} + 1$ $\qquad$ $n = \log_2\left(\frac{\Delta x}{E_s}\right)$

function NEWTON(f, f', x_0, E_s)
  $x \leftarrow x_0$
  for $k \leftarrow 1$ **to** maxit **do**
    $\Delta x \leftarrow -f(x)/f'(x)$
    $x \leftarrow x + \Delta x$
    if $|\Delta x| < E_s$ **then return** x
  report failure

$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$

$x_{i+1} = x_i - \frac{f(x_i)\,(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)}$

$\varepsilon_a = \left|\frac{x_{i+1} - x_i}{x_{i+1}}\right| \times 100\%$

$[J](\{x_i\})\,\{\Delta x\} = -\{f\}(\{x_i\})$
$\{x_{i+1}\} = \{x_i\} + \{\Delta x\}$

$E_{a,i+1} = \frac{E_{a,i}}{2}$ $\qquad$ $E_{t,i+1} \approx \frac{-f''(x_r)}{2f'(x_r)} E_{t,i}^2$

$$[J] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

Forward: $f'(x) \approx \frac{f(x+h) - f(x)}{h}$ $\qquad$ $h \approx \sqrt{\varepsilon}\, x$

Central: $f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$

$\frac{\partial \{f\}}{\partial x_j} = \frac{\{f\}(\{x\} + h\{e_j\}) - \{f\}(\{x\})}{h}$

No unique $[A]\{x\} = \{b\}$ solution $\iff$ $\det[A] = 0$ $\iff$ $[A]$ is singular $\iff$ $[A]$ not full rank $\iff$ $[A^{-1}]$ does not exist.

Growth ($n \to \infty$): $f(n) = O(g(n))$ if $\exists\, c, n_0 > 0$ such that $0 \leq f(n) \leq c\, g(n)$ $\forall n \geq n_0$
$cf(n) = O(f(n))$ $\forall c > 0$; if $|f(n)|$ is bounded, $f(n) = O(1)$
$O(n^a) + O(n^b) = O\left(n^{\max(a,b)}\right)$; $O(n^a) \cdot O(n^b) = O\left(n^{a+b}\right)$
$O(1) \subsetneq O(\log\log n) \subsetneq O(\log n) \subsetneq O(n) \subsetneq O(n^2) \subsetneq O(n^3) \subsetneq O(n^k) \subsetneq O(a^n) \subsetneq O(n!)$, $k > 3$, $a > 1$

Convergence ($h \to 0$): $E(h) = O(g(h))$ if $\exists\, c, h_0 > 0$ such that $|E(h)| \leq c\, g(h)$ $\forall\, 0 < h \leq h_0$
$c\, h^p = O(h^p)$ $\forall c > 0$; if $E(h) = O(1)$, the method does not converge
$O(h^a) + O(h^b) = O\left(h^{\min(a,b)}\right)$; $O(h^a) \cdot O(h^b) = O\left(h^{a+b}\right)$
$O(h^p) \subsetneq O(h^4) \subsetneq O(h^3) \subsetneq O(h^2) \subsetneq O(h) \subsetneq O(1)$, $p > 4$
If $E(h) = O(h^p)$, log-log plot of $|E|$ vs. h has slope $p \approx \frac{\log(E(h_1)/E(h_2))}{\log(h_1/h_2)}$

$\dot{x} = a\,x \quad \Rightarrow \quad x(t) = Ce^{at}$

$m\,\ddot{x} + c\,\dot{x} + k\,x = 0 \quad \Rightarrow \quad m\lambda^2 + c\lambda + k = 0$

$\lambda_{1,2} = \frac{-c \pm \sqrt{c^2 - 4mk}}{2m}$

Real $\lambda_1 \neq \lambda_2$: $x(t) = C_1 e^{\lambda_1 t} + C_2 e^{\lambda_2 t}$

Repeated $\lambda_1 = \lambda_2 = \lambda$: $x(t) = (C_1 + C_2\,t)\,e^{\lambda t}$

Complex $\lambda = \sigma \pm i\omega$: $x(t) = e^{\sigma t}\,(C_1 \sin \omega t + C_2 \cos \omega t)$

$m\,\ddot{x} + c\,\dot{x} + k\,x = 0$ with $v \equiv \dot{x}$: $\quad \dot{\vec{x}} = f(t, \vec{x})$

$$\vec{x} = \begin{bmatrix} x \\ v \end{bmatrix} \qquad f(t, \vec{x}) = \begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ -\frac{c}{m} v - \frac{k}{m} x \end{bmatrix}$$

function GENERIC_RUNGE_KUTTA(f, t_0, x_0, h, N)
  $t \leftarrow t_0$; $\quad x \leftarrow x_0$
  for $i \leftarrow 0$ **to** $N - 1$ **do**
    $\phi \leftarrow$ slope estimate from f, t_i, x_i, h
    $x_{i+1} \leftarrow x_i + \phi\,h$
    $t_{i+1} \leftarrow t_i + h$
  return $t_0, \ldots, t_N$ and $x_0, \ldots, x_N$

	ϕ	Local	Global
Euler	k_1	$O(h^2)$	$O(h)$
Heun's (no iteration)	$\frac{1}{2}(k_1 + k_2)$	$O(h^3)$	$O(h^2)$
Midpoint	k_2	$O(h^3)$	$O(h^2)$
Ralston's	$\frac{1}{3}k_1 + \frac{2}{3}k_2$	$O(h^3)$	$O(h^2)$
Classical RK4	$\frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$	$O(h^5)$	$O(h^4)$

$k_1 = f(t_i, x_i)$

Heun's: $k_2 = f(t_i + h, x_i + k_1 h)$

Midpoint: $k_2 = f(t_i + \frac{h}{2}, x_i + \frac{1}{2}k_1 h)$

Ralston's: $k_2 = f(t_i + \frac{3}{4}h, x_i + \frac{3}{4}k_1 h)$

RK4:

$k_2 = f(t_i + \frac{h}{2}, x_i + \frac{1}{2}k_1 h) \qquad k_3 = f(t_i + \frac{h}{2}, x_i + \frac{1}{2}k_2 h)$

$k_4 = f(t_i + h, x_i + k_3 h)$

$x_{i+1} = x_i + f(t_{i+1}, x_{i+1})\,h$

Unconditionally stable for $\dot{x} = -ax$, $a > 0$

A numerical integrator is *unstable* for a problem and step size if errors grow exponentially for a problem with a bounded solution.

Central (second derivative): $f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$

$$\nabla f(\vec{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \qquad H(\vec{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

$$\underset{x \in D}{\text{minimize}} f(x)$$

For a C^2 function: FONC: $f'(x^*) = 0$, SONC: $f''(x^*) \geq 0$

FONC: $\nabla f(\vec{x}^*) = \vec{0}$, SONC: $H(\vec{x}^*)$ is positive semi-definite

A symmetric matrix H is **positive definite** if $\vec{x}^T H \vec{x} > 0 \;\; \forall \vec{x} \neq \vec{0}$, or equivalently all eigenvalues > 0.

H is **positive semi-definite** if $\vec{x}^T H \vec{x} \geq 0 \;\; \forall \vec{x} \neq \vec{0}$, or equivalently all eigenvalues ≥ 0.

function GRADIENT_DESCENT(f, ∇f, $\vec{x}_0$, α, E_s)
  $k \leftarrow 0$
  while $\|\nabla f(\vec{x}_k)\| > E_s$ **do**
    $\vec{x}_{k+1} \leftarrow \vec{x}_k - \alpha \nabla f(\vec{x}_k)$
    $k \leftarrow k + 1$
  return $\vec{x}^* = \vec{x}_k$

x^* is a global minimum if $f(x^*) \leq f(x) \quad \forall x \in D$

x^* is a local minimum if $f(x^*) \leq f(x) \quad \forall x$ in a neighborhood of x^* within D

x^* is a strong local minimum if $f(x^*) < f(x) \quad \forall x \neq x^*$ in a neighborhood of x^* within D

`abs(x)` $|x|$ `sign(x)` -1, 0, or 1

`norm(x)` $\|x\|_2$ (2-norm of vector or matrix)

`sqrt(x)`, `exp(x)`, `log2(x)`

`[m, i] = max(abs([4,-6,3]))` value & index: `m=6`, `i=2`

`n = length(v)`; `[m, n] = size(A)`

`zeros(m, n)`, `ones(m, n)`, `eye(n)`

`A(i, :)`, `A(:, j)`, `A(i, j:n)`, `&&`, `||`, `~=`, `==`, `~`

`A'` transpose; `A*x`; `x = A\b`

`inv(A)`, `det(A)`, `cond(A)`

`eps`, `Inf`, `NaN`, `isnan(x)`, `isinf(x)`

`@(x) x^2 - 3` defines an anonymous function that computes $x^2 - 3$

`@sin` creates a handle for the function `sin`

`[t, x] = any_lab_3_integrator(dynamics, tspan, x0, h)`: `dynamics(t, x)` returns $\dot{\vec{x}}$, `tspan = [t0 tf]`, `t` is a column of times, `x(i,:)` is the state at `t(i)`

`[t, x] = ode45(dynamics, tspan, x0)` adaptive explicit RK; `ode23` same interface, implements the BS23 adaptive step algorithm discussed in class; `ode23s` same interface, implicit, for stiff systems

`[x, info] = any_lab_2_solver(f, fprime, interval, atol, maxit)`: `f(x)` and `fprime(x)` are function handles, `interval` is a two-element vector, `x` is a root estimate within `atol` of a true root, and some methods ignore or reinterpret arguments

`incremental_search`, `bisection`, `newton_raphson` (starts at the midpoint of `interval`), `secant` (the two ends of `interval` are the two starting points)

`[x, info] = newton_sys(f, J, x0, atol, maxit)`: `f(x)` returns $\vec{f}(\vec{x})$ as a column, `J(x)` returns the Jacobian, e.g. `@(x) numjac(f, x, h)`

`J = numjac(f, x, h)` forward-difference Jacobian with increment `h`