MA 214: Numerical Analysis Notes

Aryaman Maithani

2020-02-28 12:28:21+05:30
Until: Lecture 12

Disclaimer

This is just a collection of formulae/algorithms compiled together.
In the case of algorithms, I explain the procedure concisely. However, do not take this as a substitute for lecture slides as I don't go into the theory at all.
Also, I've modified some things compared to the lecture slides wherever I felt it was an error. It also is possible that I've made a typo of my own. So, be warned.
Sometimes, I also change the order in the notes compared to how it was taught in slides if I feel like that's more efficient.

---

1 Interpolation

1. **Lagrange Polynomials**
   Let $x_0,\ x_1,\ \ldots,\ x_n$ be $n+1$ distinct points in $[a,b]$. Let $f : [a,b] \to \mathbb{R}$ be a function whose value is known at those aforementioned points.
   We want to construct a polynomial $p(x)$ of degree $\leq n$ such that $p(x_i) = f(x_i)$ for all $i \in \{0,\ldots,n\}$.
   Towards this end, we define the polynomials $I_k(x)$ for $k \in \{0,\ldots,n\}$ in the following manner:

$$I_k(x) := \prod_{i=0, i\neq k}^{n} \frac{x - x_i}{x_k - x_i}.$$

   (Intuitive understanding: I_k is a degree n polynomial such that $I_k(x_j) = 0$ if $k \neq j$ and $I_k(x_k) = 1$.)
   Now, define $p(x)$ as follows:

$$p(x) := \sum_{i=0}^{n} f(x_i) I_i(x)$$

2. **Newton's form**
   Let $x_0, x_1, \ldots, x_n$ be $n+1$ distinct points in $[a,b]$. Let $f : [a,b] \to \mathbb{R}$ be a function whose value is known at those aforementioned points.
   We want to construct a polynomial $P_n(x)$ of degree $\leq n$ such that $p(x_i) = f(x_i)$ for all $i \in \{0,\ldots,n\}$.

   We define the divided differences (recursively) as follows:

$$\begin{aligned} f[x_0] &:= f(x_0) \\ f[x_0, x_1, \ldots, x_k] &:= \frac{f[x_1,\ldots,x_k] - f[x_0,\ldots,x_{k-1}]}{x_k - x_0} \qquad \text{for all } 1 < k \leq n \end{aligned}$$

   With this in place, the desired polynomial $P_n(x)$ is (not so) simply:

$$\begin{aligned} P_n(x) := f[x_0] &+ f[x_0,x_1](x - x_0) \\ &+ f[x_0,x_1,x_2](x-x_0)(x-x_1) \\ &+ \ldots \\ &\ \ \vdots \\ &+ f[x_0,x_1,\ldots,x_n](x-x_0)(x-x_1)\cdots(x-x_{n-1}) \end{aligned}$$

   Remarks. Note that $x - x_n$ does not appear in the last term.
   Note that given $P_n(x)$, it is simple to construct $P_{n+1}(x)$.

3. **Osculatory Interpolation**
   This is essentially the same as the previous case.
   I'll state the problem in the form I think is the simplest. (Any other form can be reduced to this.)
   Suppose we are given $k+1$ distinct points $x_0, \dots, x_k$ in $[a,b]$ and a function $f : [a,b] \to \mathbb{R}$ which is sufficiently differentiable.
   Suppose we are given the following values:

$$\begin{gathered} f^{(0)}(x_0), f^{(1)}(x_0), \dots, f^{(m_1-1)}(x_0) \\ f^{(0)}(x_1), f^{(1)}(x_1), \dots, f^{(m_2-1)}(x_1) \\ \vdots \\ f^{(0)}(x_k), f^{(1)}(x_k), \dots, f^{(m_k-1)}(x_k) \end{gathered}$$

   (Notation: $f^{(0)}(x) = f(x)$ and $f^{(n)}(x)$ is the n^{th} derivative.)
   Thus, we are given $m_1 + m_2 + \cdots m_k =: n+1$ data. As usual, we now want to compute a polynomial $P_n(x)$ that agrees with f at all the data. (That is, all the given derivatives must also be same.) As it goes without saying, $P_n(x)$ must have degree $\le n$.

   To do this, we list the above points as follows:

$$\underbrace{x_0, x_0, \dots x_0}_{m_1}, \underbrace{x_1, x_1, \dots, x_1}_{m_2}, \dots, \underbrace{x_k, x_k, \dots, x_k}_{m_k}.$$

   Now, we just apply the above (Newton's) formula with the following modification in the definition of the divided difference:

$$f[\underbrace{x_i, x_i, \dots, x_i}_{p+1 \text{ times}}] := \frac{f^{(p)}(x_i)}{p!}.$$

4. **Richardson Extrapolation**
   Suppose that for sufficiently small $h \neq 0$, we have the formula:

$$M = N_1(h) + k_1 h + k_2 h^2 + k_3 h^3 + \cdots,$$

   for some constants $k_1, k_2, k_3, \dots$.
   Define the following:

$$N_j(h) := N_{j-1}(h/2) + \frac{N_{j-1}(h/2) - N_{j-1}(h)}{2^{j-1} - 1} \quad \text{for } j \ge 2.$$

   Choose some h *sufficiently small* (whatever that means). Then, $N_j(h)$ keeps becoming a better approximation of M as j increases.
   We create a table of h and $N_j(h)$ as follows:

h	$N_1(h)$	$N_2(h)$	$N_3(h)$	$N_4(h)$
h	$N_1(h)$			
$h/2$	$N_1(h/2)$	$N_2(h)$		
$h/4$	$N_1(h/4)$	$N_2(h/2)$	$N_3(h)$	
$h/8$	$N_1(h/8)$	$N_2(h/4)$	$N_3(h/2)$	$N_4(h)$

   $N_4(h)$ will be a good approximation, then.
   (Look at slide 15 of Lecture 7 for an example.)

   Special case
   Sometimes, we may have the following scenario:

$$M = N_1(h) + k_2 h^2 + k_4 h^4 + \cdots.$$

   In this case, we define:

$$N_j(h) := N_{j-1}(h/2) + \frac{N_{j-1}(h/2) - N_{j-1}(h)}{4^{j-1} - 1} \quad \text{for } j \ge 2.$$

   Then, we do the remaining stuff as before.

2 Numerical Integration

$$I = \int_a^b f(x)\mathrm{d}x$$

1. **Rectangle Rule**

$$I \approx (b-a)f(a)$$

$$E^R = f'(\eta)\frac{(b-a)^2}{2}, \text{ for some } \eta \in [a,b]$$

2. **Midpoint Rule**

$$I \approx (b-a)f\left(\frac{a+b}{2}\right)$$

$$E^M = \frac{f''(\eta)}{24}(b-a)^3, \text{ for some } \eta \in [a,b]$$

3. **Trapezoidal Rule**

$$I \approx \frac{1}{2}(b-a)[f(a)+f(b)]$$

$$E^T = -f''(\eta)\frac{(b-a)^3}{12}, \text{ for some } \eta \in [a,b]$$

4. **Corrected Trapezoidal**

$$I \approx \frac{1}{2}(b-a)[f(a)+f(b)] + \frac{(b-a)^2}{12}(f'(a)-f'(b))$$

$$E^{CT} = f^{(4)}(\eta)\frac{(b-a)^5}{720}, \text{ for some } \eta \in [a,b]$$

5. **Composite Trapezoidal**

$$I \approx \frac{h}{2}\left[f(x_0) + 2\sum_{i=1}^{N-1} f(x_i) + f(x_N)\right]$$

$$E_C^T = -f''(\xi)\frac{h^2(b-a)}{12}, \text{ for some } \xi \in [a,b]$$

Here, $Nh = b-a$ and $x_i = a+ih$.

6. **Simpson's Rule**

$$I \approx \frac{b-a}{6}\left\{f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right\}$$

$$E^S = -\frac{1}{90}f^{(4)}(\eta)\left(\frac{b-a}{2}\right)^5, \text{ for some } \eta \in [a,b]$$

7. **Composite Simpson's**

$$I \approx \frac{h}{6}\left[f(x_0) + 2\sum_{i=1}^{N-1} f(x_i) + 4\sum_{i=1}^{N} f\left(x_{i-1}+\frac{h}{2}\right) + f(x_N)\right]$$

$$E_C^S = -f^{(4)}(\xi)\frac{(h/2)^4(b-a)}{180}, \text{ for some } \xi \in [a,b]$$

Here, $Nh = b-a$ and $x_i = a+ih$.

8. **Gaussian Quadrature**
Let $Q_{n+1}(x)$ denote the $(n+1)^{\text{th}}$ Legendre polynomial.
Let $r_0, \ldots, r_{n+1}$ be its roots. (These will be distinct, symmetric about the origin and will lie in the interval $[-1, 1]$.)
For each $i \in \{0, \ldots, n\}$, we define c_i as follows:

$$c_i := \int_{-1}^{1} \left(\prod_{k=0, k\neq i}^{n} \frac{x - x_k}{x_i - x_k} \right) \mathrm{d}x.$$

Then, we have

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \sum_{i=0}^{n} f(r_i)c_i.$$

Moreover, if f is a polynomial of degree $\leq 2n+1$, then the above is "approximation" is exact.

Standard values:
$n = 0 : Q_1(x) = x$ and $x_0 = 0$. $c_0 = 2$.
$n = 1 : Q_2(x) = x^2 - \frac{1}{3}$ and $x_0 = -\frac{1}{\sqrt{3}},\ x_1 = \frac{1}{\sqrt{3}}$. $c_0 = c_1 = 1$.
$n = 2 : Q_3(x) = x^3 - \frac{3}{5}x$ and $x_0 = -\sqrt{\frac{3}{5}},\ x_1 = 0,\ x_2 = \sqrt{\frac{3}{5}}$. $c_0 = c_2 = 5/9,\ c_1 = 8/9$.

9. **Improper integrals using Taylor series method**
Suppose we have $f(x) = \frac{g(x)}{(x-a)^p}$ for some $0 < p < 1$ and are asked to calculate $I = \int_a^b f(x)\mathrm{d}x$.
For the sake of simplicity, I assume $a = 0$ and $b = 1$.
Let $P_4(x)$ denote the fourth Taylor polynomial of g around a. (In this case 0.)
Now, compute $I_1 = \int_0^1 \frac{P_4(x)}{x^p}\mathrm{d}x$. This can be integrated exactly. (Why?)
Now, we approximate $I - I_1$.
Define

$$G(x) := \begin{cases} \frac{g(x) - P_4(x)}{x^p} & \text{if } 0 < x \leq 1 \\ 0 & \text{if } x = 0 \end{cases}$$

Then, approximate $I_2 = \int_0^1 G(x)\mathrm{d}x$ using the composite Simpson's rule.
Then, $I = I_1 + I_2$.

For the case of $a = 0$, $b = 1$ and $N = 2$ for the composite Simpson's part, we get that
$I_2 \approx \frac{1}{12}[2G(0.5) + 4G(0.25) + 4G(0.75) + G(1)]$.

That is, finally:

$$I \approx \int_0^1 \frac{P_4(x)}{x^p}\mathrm{d}x + \frac{1}{12}[2G(0.5) + 4G(0.25) + 4G(0.75) + G(1)].$$

10. **Adaptive Quadrature**
Let $I = \int_a^b f(x)\mathrm{d}x$ be the integral that we want to approximate.
Suppose that ϵ is the accuracy to which we want I. That is, we want a number P such that $|P - I| < \epsilon$.
Here is what you do:
Subdivide $[a, b]$ into N intervals: $[x_0, x_1]$, $[x_1, x_2]$, $\ldots$, $[x_{n-1}, x_n]$. (Naturally, $a = x_0 \leq x_1 \leq \ldots \leq x_n = b$.)
Now, for each subinterval, compute the following values:

$$S_i = \frac{h}{6}\left(f(x_i) + 4f\left(x_i + \frac{h}{2}\right) + f\left(x_{i+1}\right)\right), \text{ and}$$

$$\overline{S_i} = \frac{h}{12}\left(f(x_i) + 4f\left(x_i + \frac{h}{2}\right) + 2f\left(x_i + \frac{h}{2}\right) + 4f\left(x_i + \frac{3h}{4}\right) + f(x_{i+1})\right).$$

Now, calculate $E_i = \frac{1}{15}|\overline{S_i} - S_i|$.

Now, if $E_i \leq \frac{x_i - x_{i-1}}{b-a}\epsilon$, then move on to the next interval.

Otherwise, subdivide again to better approximate $\int_{x_{i-1}}^{x_i} f(x)\mathrm{d}x$.

Finally, sum up all the $\overline{S_i}$s and that's the answer. That is,

$$I \approx P = \sum_{i=1}^{n} \overline{S_i}.$$

11. **Romberg Integration**
Essentially the baby of composite Trapezoidal rule and Richardson extrapolation.
Suppose we want to calculate $\int_a^b f(x)\mathrm{d}x$.
Let N be a power of 2.
$T_N := \frac{h}{2}\left[f(x_0) + 2\sum_{i=1}^{N-1} f(a+ih) + f(x_N)\right]$, where $Nh = b-a$.
Note that T_N can be computed using $T_{N/2}$ (assuming $N \neq 2^0$) as:

$$T_N = \frac{T_{N/2}}{2} + h\sum_{i=1}^{N/2} f\left(a + (2i-1)h\right).$$

In the above, we have $2Nh = b-a$. (That is, it's not the same as the previous h.) Now for $m \geq 1$, we define:

$$T_N^m = T_N^{m-1} + \frac{T_N^{m-1} - T_{N/2}^{m-1}}{4^m - 1}.$$

(Where T_N^0 is just T_N.)
(Also, for some reason, T_N' has been used instead of T_N^1.)
Note that $\frac{N}{2^m}$ must be an integer for T_N^m to be defined. We create a table as follows:

N	T_N	T_N'	T_N^2	T_N^3
1	T_1			
2	T_2	T_2^1		
4	T_4	T_4^1	T_4^2	
8	T_8	T_8^1	T_8^2	T_8^3

T_8^3 will be a good approximation, then.
(Look at slide 25 of Lecture 7 for an example.)

Remark. It can be shown that $I = T_N + c_2h^2 + c_4h^4 + \cdots$. This is why we used the special case formula of 1. 4.

3 Numerical Differentiation

1.
$$f'(a) \approx \frac{f(a+h) - f(a)}{h}$$
$$E(f) = -\frac{1}{2}hf''(\eta) \quad \text{for some } \eta \in [a, a+h].$$

2. **Central Difference Formula**
$$f'(a) \approx \frac{f(a+h) - f(a-h)}{2h}$$
$$E(f) = -\frac{1}{6}h^2 f^{(3)}(\eta) \quad \text{for some } \eta \in [a-h, a+h].$$
Note that this is an $O(h^2)$ approximation. Thus, we can use the special case of §1. 4. for better accuracy.

3.
$$f'(a) \approx \frac{-3f(a) + 4f(a+h) - f(a+2h)}{2h}$$

$$E(f) = \frac{1}{3}h^2 f^{(3)}(\eta) \quad \text{for some } \eta \in [a, a+2h].$$

Formula 2 is always the better one whenever applicable. At end points, formula 3 is better than formula 1.

4. **Central difference for second derivative**

$$f''(x_0) = \frac{f(x_0+h) - 2f(x_0) + f(x_0-h)}{h^2} - \frac{h^2}{12}f^{(4)}(\xi),$$

for some $\xi \in (x_0 - h, x_0 + h)$.

5. **Solving boundary-value problems in ODE**
Suppose that we want to solve the following (linear) ODE:

$$y''(x) + f(x)y'(x) + g(x)y = q(x)$$

in the interval $[a, b]$ such that we know $y(a) = \alpha$, and $y(b) = \beta$.

Set $h := \frac{b-a}{N}$ for some $N \in \mathbb{N}$ and $x_i = a + ih$ for $h \in \{0, 1, \ldots, N\}$.
Using central difference approximation, we set up $N - 1$ linear equations as follows:

$$\frac{y_{i-1} - 2y_0 + y_i}{h^2} + f(x_i)\frac{y_{i+1} - y_{i-1}}{2h} + g(x_i)(y_i) = q(x_i)$$

$$i = 1, 2, \ldots, N-1$$

The above equations can be rearranged as:

$$\left(1 - \frac{hf_i}{2}\right) y_{i-1} + (-2 + h^2 g_i)y_i + \left(1 + \frac{hf_i}{2}\right) y_{i+1} = h^2 q_i,$$

for $i = 1, \ldots, N-1$; where $f_i = f(x_i)$ and so on.

4 Solution of non-linear equations

Let f be a continuous function on $[a_0, b_0]$ such that $f(a_0)f(b_0) < 0$ in all these cases. We want to find a root of f in $[a_0, b_0]$. (Existence in implied.)

1. **Bisection Method**
Set $n = 0$ to start with.
Loop over the following:
Set $m = \frac{a_n + b_n}{2}$.
If $f(a_n)f(m) < 0$, then set $a_{n+1} = a_n$ and $b_{n+1} = m$.
Else, set $a_{n+1} = m$ and $b_{n+1} = b_n$.
Increase n by one.
We still have a root in $[a_n, b_n]$.

2. **Regula-falsi or false-position method**
Set $n = 0$ to start with.
Loop over the following:
Set $w = \frac{f(b_n)a_n - f(a_n)b_n}{f(b_n) - f(a_n)}$.
If $f(a_n)f(w) < 0$, then set $a_{n+1} = a_n$ and $b_{n+1} = w$.
Else, set $a_{n+1} = w$ and $b_{n+1} = b_n$.
Increase n by one.
We still have a root in $[a_n, b_n]$.

3. **Modified regula-falsi**
Set $n = 0$ and $w_0 = a_0$ to start with.
Loop over the following:

Set $F = f(a_n)$ and $G = f(b_n)$.
Set $w_{n+1} = \dfrac{Ga_n - Fb_n}{G - F}$.
If $f(a_n)f(w_{n+1}) \leq 0$, then set $a_{n+1} = a_n$ and $b_{n+1} = w_{n+1}$ and $G = f(w_{n+1})$.
Furthermore, if we also have $f(w_n)f(w_{n+1}) > 0$, set $F = \frac{F}{2}$.
Else, set $a_{n+1} = w_{n+1}$ and $b_{n+1} = b_n$ and $F = f(w_{n+1})$.
Furthermore, if we also have $f(w_n)f(w_{n+1}) > 0$, set $G = \frac{G}{2}$.
Increase n by one.
We still have a root in $[a_n, b_n]$.

4. **Secant method**
   Set $x_0 = a$, $x_1 = b$ and until satisfied, keep computing x_n given by

$$x_{n+1} = \frac{f(x_n)x_{n-1} - f(x_{n-1})x_n}{f(x_n) - f(x_{n-1})} \qquad \text{for } n \geq 1.$$

   Remark. This process will be forced to stop if we arrive at $f(x_n) = f(x_{n-1})$ at some point.

5 Iterative methods

1. **Newton's Method**
   You are given a function f which is continuously differentiable and you want to find its root. You are also given some x_0.
   Compute the following sequence recursively until satisfied:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad \text{for } n \geq 0.$$

2. **Fixed point iteration**
   Let I be a closed interval in $\mathbb{R}$. Let $f : I \to I$ be a differentiable function such that there exists some $K \in [0, 1)$ such that $|f'(x)| \leq K$ for all $x \in I$.
   Then, there is a unique $\xi \in I$ such that $f(\xi) = \xi$. To find this fixed point, choose any $x_0 \in I$ and define the sequence

$$x_n := f(x_{n-1}) \quad n \geq 1.$$

   Then, $x_n \to \xi$.

3. **Aitken's Δ^2 Process**
   Definition. Given a sequence (x_n), let $\Delta x_n := x_{n+1} - x_n$.
   Then, $\Delta^2 x_n = x_{n+2} - 2x_{n+1} + x_n$.

   Given a sequence $x_0, x_1, \ldots$ converging to ξ, calculate $\widehat{x_1}, \widehat{x_2}, \ldots$ by

$$\widehat{x_n} := x_{n+1} - \frac{(\Delta x_n)^2}{\Delta^2 x_{n-1}}.$$

   Then, $\widehat{x_n} \to \xi$.

   If the sequence $x_0, x_1, \ldots$ converges linearly to ξ, that is, if

$$\xi - x_{n+1} = K(\xi - x_n) + \theta(\xi - x_n), \qquad \text{for some } K \neq 0$$

   then $\widehat{x_n} = \xi + O(\xi - x_n)$, that is, $\dfrac{\widehat{x_n} - \xi}{x_n - \xi} \to 0$.

4. **Steffensen iteration**
   Let $g(x)$ be the function whose fixed point is desired. Let y_0 be some given point.
   Set $n = 0$ to start with.
   Loop over the following:
   Set $x_0 = y_n$.
   Set $x_1 = g(x_0)$, $x_2 = g(x_1)$.

Calculate Δx_1 and $\Delta^2 x_0$.

Set $y_{n+1} = x_2 - \dfrac{(\Delta x_1)^2}{\Delta^2 x_0}$.

Increase n by 1.

Note that we get a sequence $y_0, y_1, y_2, \ldots$. However, we only ever have x_0, x_1 and x_2.

Definition 1. *Let $x_0, x_1, x_2, \ldots$ be a sequence that converges to ξ and set $e_n = \xi - x_n$.*
If there exists a number P and a constant $C \neq 0$ such that

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^P} = C,$$

*then P is called the **order of convergence** and C is called **asymptotic error constant**.*

Examples.

1. **Fixed point iteration**
   ξ fixed point of $g : I \to I$ and $g'(\xi) \neq 0$.
   $P = 1$ and $C = |g'(\xi)|$.
2. **Newton's method**
   $\lim\limits_{n\to\infty} \dfrac{|e_{n+1}|}{|e_n|} = \dfrac{1}{2}\left|\dfrac{f''(\xi)}{f'(\xi)}\right|$.
   (If ξ is a double root, then $P = 1$.)
3. **Secant method**

   $$|e_{n+1}| = C|e_n||e_{n-1}|$$

   $P = \frac{1+\sqrt{5}}{2} = 1.618\ldots$.

   $$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^P} = \left|\frac{1}{2}\frac{f''(\xi)}{f'(\xi)}\right|^{1/P}, \quad \text{provided } f'(\xi) \neq 0.$$

Theorem 1. *Let $f : [a, b] \to \mathbb{R}$ be in $C^2[a, b]$ and let the following conditions be satisfied:*
1. *$f(a)f(b) < 0$,*
2. *$f'(x) \neq 0$, for all $x \in [a, b]$,*
3. *$f''(x)$ doesn't change sign in $[a, b]$ (might be zero at some points),*
4. $$\frac{|f(a)|}{|f'(a)|} \leq b - a \text{ and } \frac{|f(b)|}{|f'(b)|} \leq b - a.$$

Then, the Newton's method converges to the unique solution ξ of $f(x) = 0$ in $[a, b]$ for any choice $x_0 \in [a, b]$.

6 Solving systems of linear equations

6.1 LU Factorisation

We want solve $Ax = b$ where A is some known $n \times n$ matrix, b a known $n \times 1$ matrix and x is unknown.
Assumption: $Ax = b$ can be solved without any row interchange.
We define (finite) sequences of matrices $A^{(n)} = [a_{ij}^{(n)}]$ and $b^{(n)}$.

Define $A^{(1)} := A$. Let $m_{ji} := \dfrac{a_{ji}^{(i)}}{a_{ii}^{(i)}}$.

Define $M^{(1)}$ as

$$M^{(1)} := \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -m_{21} & 1 & 0 & \cdots & 0 \\ -m_{31} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -m_{n-1,1} & 0 & 0 & \cdots & 0 \\ -m_{n1} & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Thus, we can write $M^{(1)}A^{(1)}x = M^{(1)}b$.
Let $A^{(2)} := M^{(1)}A^{(1)}$ and $b^{(2)} = M^{(1)}b^{(1)}$.
Note that $A^{(2)}$'s first column will just have the top element non-zero and everything below will be zero.

We can similarly construct the later matrices that perform the row operations. In general, we have:

$$M^{(k)} := \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & \cdots & 0 \\ 0 & 0 & \cdots & -m_{k+1,k} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -m_{n,k} & \cdots & 1 \end{bmatrix},$$

along with

$$A^{(k+1)} = M^{(k)}A^{(k)} = M^{(k)} \cdots M^{(1)}A, \text{ and}$$
$$b^{(k+1)} = M^{(k)}b^{(k)} = M^{(k)} \cdots M^{(1)}b.$$

Finally, set $U = A^{(n)}$ and $L = [M^{(1)}]^{-1} \cdots \left[M^{(n-1)}\right]^{-1}$.
Then, we have

$$L = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ m_{21} & 1 & 0 & \cdots & 0 \\ m_{31} & m_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{n-1,1} & m_{n-1,2} & m_{n-1,3} & \cdots & 0 \\ m_{n1} & m_{n2} & m_{n3} & \cdots 1 & \end{bmatrix}.$$

Thus, we have $A = LU$. Now, set $y = Ux$. We solve $Ly = b$ for y. This is easy because L is lower triangular. Then, we solve $Ux = y$ for x.

Check slide 27 of Lecture 11 for example.

LU factorisation requires $\theta(n^3/3)$ multiplication and division and $\theta(n^3/3)$ addition and subtraction. However, once the factorisation is done, it can be used again and again to solve $Ax = b$ for different values of b. The number of equations then taken to solve $Ax = b$ is $\theta(2n^2)$.

6.2 LPU Factorisation

In the previous case, we assumed that we aren't doing any row operations on A when doing the Gauss elimination. Now, we relax that condition.
Suppose, if possible, we have done some row exchanges while doing Gauss elimination on A. Construct the matrix P which is obtained upon doing those exact row exchanges on I. (Only the row exchanges.)
Suppose that the the final product of doing the Gauss elimination following the previous procedure gave us the upper triangular matrix as U and lower one as L.
Then, that means $PA = LU$. Thus, solving $Ax = b$ can first be reduced to $PAx = Pb = b'$. Now, converting PA to LU does not take any row exchange, so we're back in business. (Back to the previous case, that is.)
Check slide 15 of Lecture 12 for example.

6.3 Cholesky's Algorithm

Definition 2. *A matrix A is positive definite if:*

1. *$A = A^t$, and*
2. *$x^tAx > 0$ for all $x \neq 0$.*

Given any positive definite matrix A, we can write $A = LL^t$, where L is lower triangular. (And thus, L^T would be upper triangular.)
The algorithm to do so is as follows:
Let $L = (l_{ij})$. We now construct these entries.

`set` $l_{ii} := \sqrt{a_{11}}$.

`for` $j = 2, 3, \ldots, n$:
    `set` $l_{j1} := \frac{a_{j1}}{l_{11}}$.
`for` $i = 2, 3, \ldots, n-1$:

`set` $l_{ii} := \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}$

`for` $j = i+1, \ldots, n$:

`set` $l_{ji} := a_{ji} - \dfrac{\sum_{k=1}^{i-1} l_{jk} l_{ik}}{l_{ii}}$.

`set` $l_{nn} := \sqrt{a_{nn} - \sum_{k=1}^{n-1} l_{nk}^2}$.

Note that in the above, we've only defined l_{ij} for when $i \geq j$. The others are obviously 0.

Cholevsky's algorithm requires $\theta(n^3/6)$ multiplication and division and $\theta(n^3/6)$ addition and subtraction.

6.4 Scaled partial pivoting

Suppose we want to solve $Ax = b$. This could lead to round off errors (check slide 17 of Lecture 12). One possible way to combat this is to do the following:

First, define the scale factor s_i of row i as the maximum modulus of any element in the row. More precisely, $s_i := \max\limits_{1 \leq j \leq n} |a_{ij}|$.
(Note that we assume that A is invertible and thus, every s_i will be nonzero.)

Now, along the first column, we find the row which has the greatest ratio $\dfrac{|a_{k1}|}{s_k}$. Suppose this is achieved for row p, that is,

$$\frac{|a_{p1}|}{s_p} = \max_{1 \leq k \leq n} \frac{|a_{k1}|}{s_k}.$$

If $p \neq 1$, then we perform $R_1 \leftrightarrow R_p$.
(Now, when we refer to s_p, we will mean the scale factor of the new p, that is, the original s_1. Similarly, for s_1.)
Now, we make everything below a_{11} zero using the standard row operation.

Now, we repeat the same thing by going along column 2 and finding the row which has the greatest ratio $\dfrac{|a_{k2}|}{s_k}$. If the row is not 2, then we perform the interchange and go on.

Two things I would like to point out:
1. We never change the value of s_i except the interchanging that we do. In particular, after doing the row operations like $R_2 - m_{21} R_1$, I will **not** recalculate the value of s_2.
2. When considering the ratios $\dfrac{|a_{ki}|}{s_k}$ for column i, we will consider the a_{ki} that is currently present. That is, in this case, we will consider the values that we get after performing all the operations that have been performed until that point.

Consider the example given in slide 22 of Lecture 12. After doing $R_1 \leftrightarrow R_3, R_2 - \frac{4.01}{1.09} R_1$, and $R_3 - \frac{2.11}{1.09} R_1$, we *do not* recalculate s_i. (We do exchange the values.)
However, when considering the ratios, we consider the new matrix for taking a_{k2} obtained after the above subtractions.