0. Latar Kiraan Berkomputer

0.0 Komputer
0.1 Pengiraan berkomputer
0.2 Kaedah berangka asas

- Tugasan 1

0.0 Komputer

0.0.0 Mesin pengira

Komputer ialah mesin yang membuat pengiraan. Secara amnya, ada banyak jenis mesin pengira, yang berbentuk mekanikan, elektronik, dan lain-lain, namun bila ‘komputer’ disebut, ia merujuk kepada mesin berbentuk tertentu. Ia dicirikan oleh senibina tertentu.

0.0.1 Senibina

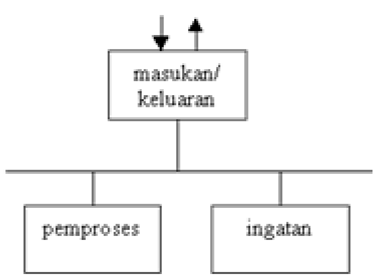
Suatu komputer lazim mempunyai ciri-ciri berikut:

- digitan (lawan analog atau hibrid) – sesuatu talian bawa hanya maklumat dedua (‘0’ dan ‘1’). ‘0’ diwakili oleh voltan rendah, misalnya kurang daripada 0.5 V, dan ‘1’ oleh voltan tinggi, misalnya lebih tinggi daripada 1 V. Ada selang voltan di mana nilai dedua tak terdefinisi; ini supaya tiada ralat atau kekaburan akibat ketaktepatan ukuran voltan. Sistem analog mewakili sesuatu nilai it uterus dengan nilai sebenar voltan, jadi ia terdedah kepada ralat dan hereotan. Sistem hibrid menggunakan campuran kaedah digitan dan analog.

- pemprosesan berjujukan. Data di atas talian diproses satu persatu.

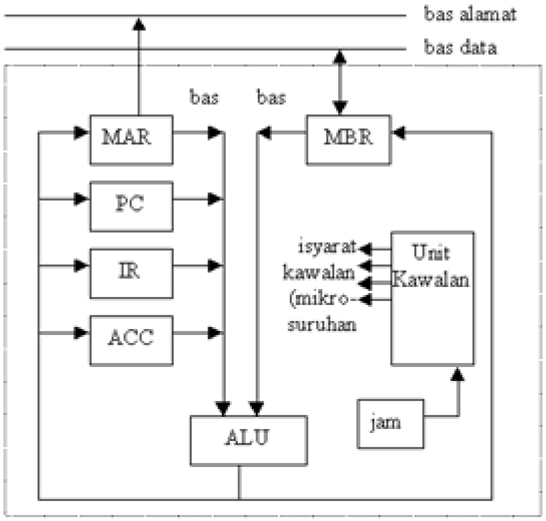
- konsep aturcara tersimpan. Perjalanan pemprosesan dikawal supaya jujukan tertentu proses-proses diajalankan. Dalam komputer lazim, perjalanan proses ini boleh ditentukan oleh suatu ‘aturcara’, dan aturcara ini disimpan dalam ingatannya. Bahkan aturcara ini boleh dimodifikasi dan diubahsuai.

- penggunaan bas kongsian – bas data, bas alamat (lokasi data), bas kawalan. Peralihan maklumat dilakukan menerusi talian dikongsipakai. Pehak penghantar dan pehak penerima ditentukan oleh isyarat-isyarat kawalan pada sesuatu masa itu. Rajah berikut menskemakan sambungan bas di antara peranti-peranti.



Senibina lazim komputer ini digelar senibina von Neumann.

Hati komputer ialah pemproses pusatnya. Peranti inilah yang mengawal telatah peranti-peranti lain pada sesuatu masa itu, dan disinilah operasi ke atas data dijalankan. Di bawah dipaparkan senibina pemproses pusat yang lazim, secara berskema.



MAR: alatdaftar alamat ingatan
PC: pembilang aturcara
IR: alatdaftar suruhan
ACC: akumulator
MBR: alatdaftar penimbal ingatan
ALU: unit aritmetik dan logik

Bas alamat ialah bas kongsian yang membawa maklumat lokasi dalam peranti ingatan bagi data yang sedang diteliti, sementara bas data adalah bas kongsian yang membawa data berkenaan. Pemproses pusat juga punyai bas dalaman.

Alatdaftar alamat ingatan (MAR) dalam pemproses pusat menyimpan lokasi ingatan yang mengandungi data yang diteliti, dalam bentuk nombor dedua. Kandungan alatdaftar ini ditulis terus bas alamat (kandungan alatdaftar menentukan keadaan talian bas). Kandungan bas alamat disimpan sementara dalam alatdaftar penimbal ingatan (MBR) dalam pemproses. Data ini dikomunikasi ke peranti lain menerusi bas data ini, dan kawalan daripada pemproses menentukan samada pernati berkenaan (termasuk pemproses sendiri) menulis atau membaca daripada bas ini (iaitu arah komunikasi).

Perjalanan pemproses adalah menurut suatu kitar. Arahan diambil satu persatu daripada aturcara tersimpan di dalam ingatan. Kedudukan arahan ini ditentukan kandungan MAR pada masa berkenaan, yang memberitahu ingatan data dari lokasi mana perlu diambil. Kod arahan, yang merupakan data yang berada dalam lokasi ini, dibaca dari ingatan dan ditulis kepada MBR menerusi bas data. Kemudian kandungan MBR disalin ke dalam alatdaftar suruhan (IR) dalam pemproses. Setiap suruhan kemudiannya diterjemahkan kepada jujukan mikrosuruhan-mikrosuruhan tertentu. Mikrosuruhan diaktifkan mengikut masa tertentu, yang dikawal denyutan jam dalaman pemproses, diberikan oleh suatu Unit Kawalan, yang bolehjadi berbentuk matriks (suruhan apa, lawan denyut jam). Mikrokawalan boleh juga menggerakkan kawalan ke peranti lain.

Pengambilan suruhan ini merupakan proses asas dalam kitaran pemproses. Kitar suruhan diberikan berikut:

Kitar suruhan:

ulang
- MAR $\leftarrow$ PC
- PC $\leftarrow$ PC + 1
- MBR $\leftarrow$ (MAR)
- IR $\leftarrow$ MBR
- Unit Kawalan: mikrosuruhan menurut IR

(MAR $\leftarrow$ PC ... IR $\leftarrow$ MBR) peringkat ambil

(Unit Kawalan: mikrosuruhan menurut IR) peringkat nyahkod dan laksana

Jadi, aturcara yang terstor dalam ingatan diambil pemproses, dinyahkod dan dilaksana. Kod-kod suruhan dedua ini mengkodkan aturcara, dan dipanggil kod mesin. Bahasa himpunan, bahasa aturcara komputer yang termudah (peringkat terendah) biasanya boleh diterjemah satu-ke-satu ke bahasa mesin, dan sebaliknya.

Tak semua kandungan ingatan mengadungi kod suruhan. Ada juga yang mengkodkan data tertentu. Bahkan ada suruhan yang memerlukan data tertentu, bolehjadi nilai integer sebenar, atau nilai alamat sesuatu lokasi. Misalnya, operasi storkan (iaitu salin kandungan akumulator ACC kepada lokasi dalam ingatan) atau bebankan (ingatan ke pemproses), operasi-operasi aritmetik dan logik (dikendalikan unit aritmetik dan logik ALU), dan operasi lompat dan lompat bersyarat (kawalan dialihkan kepada kod suruhan di suatu alamat lain dalam ingatan).

Sebagai contoh, baris aturcara dalam bahasa himpunan yang berbentuk

```
LDA 1BH
```

yang menyuruh bebankan akumulator pemproses dengan kandungan alamat 001BH (1B dalam perenambelasan) atau dedua 0000 0000 0001 1100 iaitu persepuluhan 28, mungkin dalam bahasa mesinnya (bergantung kepada jenis/jenama pemproses)

```
3A 1B
```

dalam heks (perenambelasan), iaitu 3A mewakili ‘LDA’ dan 1B ialah data diperlukan, iaitu alamat sumber. Katakan suhan ini berada di alamat 0010 (katakan sistem 8 bit, iaitu satu lokasi boleh diisi dengan 8 digit dedua, atau 2 digit perenambelasan) :

000F	
0010	3A
0011	1B
0012	

dan di lokasi 001B pula,

001A	
001B	1F
001C	

Bila pengira aturcara PC mencapai 0010 dan isinya ditulis kepada MAR (PC automatis ditokok menjadi 0011), kandungan MAR ini pula ditulis ke bas alamat sementara dibaca oleh ingatan, yang menulis kandungan lokasi yang diisyaratkan it uke atas bas data, yang dibaca ke atas MBR. Iaiut, kandungan yang ditunjukkan MAR di alihkan ke MBR. Kawalan-kawalan ini diorkestrakan oleh unit kawalan pemproses pusat. MBR, yang kini mengandungi kod suruhan yang bakal dilaksnanan, iaitu ‘3A’, menyalinkannya kepada IR. Suruhan dalam IR itu dinyahkodkan kepada jujukan mikrosuruhan berikut:

MAR $\leftarrow$ MAR + 1	alamat seteursnya utk data diperlukan
PC $\leftarrow$ PC + 1	offset PC
MBR $\leftarrow$ (MAR)	ambil data ini
ACC $\leftarrow$ MBR	bebankan akumulator

Maka selepas ini, PC mengandungi ‘12’, iaitu alamat suruhan seterusnya, dan ACC mengandungi ‘1F’ (MAR kini mengandungi ‘11’ sementara MBR ‘1F’)

Senibina pemproses sebegini adalah pusat kepada senibina aturcara terstor von Neumann.

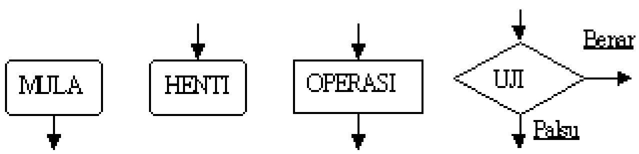
0.0.2 Aturcara

Aturcara menentukan telatah mesin komputer pada masa tertentu itu. Telatah yang ditentukan itu, untuk memperolehi hasil tertentu, disebut algoritma.

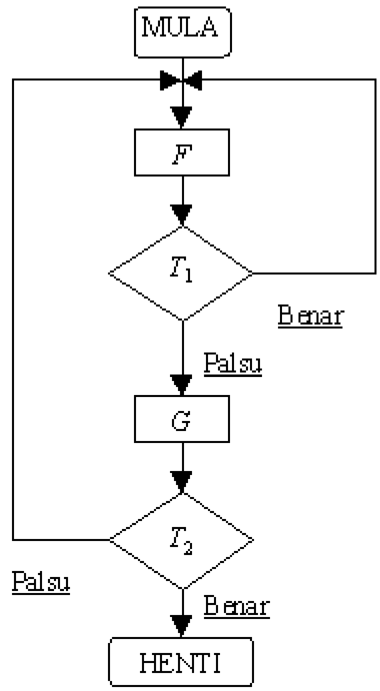
Secara formal, aturcara ialah suatu set berstruktur suruhan-suruhan yang membolehkan suatu ‘mesin’ mengenakan satu persatu operasi-operasi dan ujian-ujian asas yang tertentu dalam cara berketentuan ke atas data awal, sehingga data tersebut telah diolah ke dalam bentuk yang dikehendaki. Mesin bakal dibedah kemudian.

Ada pelbagai model untuk aturcara secara formal. Di sini kita lihat tiga contoh: aturcara cartaliran, aturcara sementara, dan aturcara tatacara.

Pemodelan aturcara yang paling langsung dan mudah ialah dengan menggunakan carta aliran. Aturcara yang diwakilkan menggunakan carta aliran dinamakan aturcara cartaliran. Ada empat komponen asas carta aliran, seperti berikut:



Suatu carta aliran merupakan gabungan komponen komponen ini, yang masing-masing menandakan permulaan aturcara, akhir aturcara, sesuatu operasi dalam aturcara, dan sesuatu ujian dalam aturcara. Contoh suatu aturcara cartaliran adalah berikut ini:



Di sini, F, G, … merupakan pengecam operasi sementara T_1, T_2, … merupakan pengecam ujian. Aturcara ini sepadan dengan jujukan suruhan-suruhan terlabel berikut:

1: **do** *F* **then goto** 2
2: **if** T_1 **then goto** 1 **else goto** 3
3: **do** *G* **then goto** 4
4: **if** T_2 **then goto** 5 **else goto** 1

di mana makna-makna berikut terpakai:

do – lakukan   **then goto** – maka/kemudian ke
if – jika benar   **else goto** – jika tidak pergi ke

Dua bentuk suruhan terlabel dibenarkan:

l: **do** *F* **then goto** *l* '
l: **if** *T* **then goto** *l* ' **else goto** *l* "

di mana F adalah operasi, T ujian, dan l, l', l'' adalah label-label kedudukan dalam carta aliran. Secara definisi, suatu cartaliran P ialah set terhad suruhan-suruhan terlabel dengan ciri

$\forall$ suruhan $i,j \in P$ jika $\lambda(i) = \lambda(j)$ maka $i = j$

di mana $\lambda(i)$ adalah label bagi i. Label awal adalah misalnya 1, dan label penamat merupakan l yang muncul dalam P yang mana tiada i dalam P di mana $l = \lambda(i)$.

Aturcara sementara pula merupakan aturcara yang berpusatkan terma sementara, atau ‘while’. Aturcara dibina berdasarkan komposisi dua (sub)aturcara,

jika P dan Q aturcara, $P;Q$ juga aturcara
(jalankan P kemudian Q)

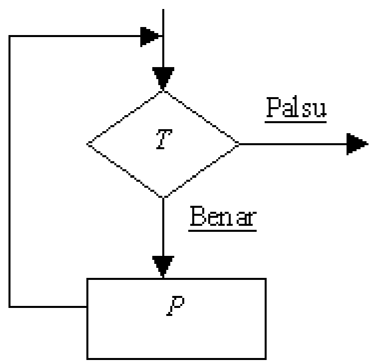
pembentukan kenyataan bersyarat daripada dua (sub)aturcara pilihan,

jika P dan Q aturcara, (if T then P else Q) ialah aturcara
(jalankan P jika T benar atau Q jika T palsu)

dan pembentukan kenyataan 'while',

jika P aturcara, while T do (P) ialah aturcara
(berulang uji T dan lakukan P, sehingga ujian T berhasil palsu)

Aturcara nol, iaitu tak buat apa-apa, diwakili I (jadi $I;P = P;I = P$ bagi apa-apa P). Kenyataan while T do (P) setara dengan until $\neg T$ do (P), dan dibina darpada kenyataan 'if', seperti diberi cartaliran berikut:



Maka setiap aturcara sementara boleh diterjemahkan kepada aturcara cartaliran yang sepadan.

Aturcara tatacara pula berbentuk:

E **where** R_1 **is** E_1, R_2 **is** $E_2, \ldots, R_n$ **is** E_n.

E merupakan 'ungkapan awal' yang mendefinisikan aturcara, dengan pengecam-pengecam tatacara R_i di dalamnya, setiapnya didefiniskan ungkapan E_i (yang seterusnya mengandungi tatacara-tatacara terdefinisi ungkapan masing-masing). Ungkapan bolehjadi pengecam tatacara, atau pengecam operasi, atau ungkapan nol I, atau komposisi ungkapan-ungkapan $D;E$, atau bentuk 'if' (**if** T **then** D **else** E). Satu contoh ialah:

$R_1;R_2$ **where**
R_1 **is** F;(**if** T **then** R_1 **else** $G;R_2$),
R_2 **is** (**if** T **then** I **else** $F;R_1$)

Bagi setiap aturcara cartaliran, aturcara tatacara yang sepadan boleh dituliskan.

Yang relevan dalam pembinaan aturcara ialah, kesetaraan di antara aturcara berlainan, ketamatan dalam membuat pengiraan, dan kebetulan hasil pengiraan menurut apa yang dikehendaki. Untuk pengkajian hal-hal ini, proses pengkomputeran diarah aturcara boleh diformalkan.

0.0.3 Pengkomputeran

Untuk menjalankan pengkomputeran seperti diarahkan aturcara, suatu pelantar pengkomputeran diperlukan, di atas mana operasi dan ujian mendapat makna. Pelantar ini dipanggil mesin, dan aturcara berjalan atas mesin memberikan komputeran.

Mesin memberikan semua maklumat yang tiada dalam aturcara supaya komputeran boleh dijalankan. Ia menentukan makna-makna pengecam operasi, ujian, masukan dan keluaran maklumat. Definisi sesuatu mesin mempunyai aklumat tentang struktur ingatan dan bagaimana masukan mencorak ingatan mesin (fungsi masukan) dan keluaran menterjemah ingatan mesin (fungsi keluaran). Definisi mesin digunakan juga termasuk penentuan setiap operasi, bagaimana ia masing-masing mengolah ingatan mesin, dan penentuan setiap ujian yang ada, apakah pula fungsi kebenaran masing-masing, yang bergantung kepada kandungan ingatan mesin.

Dengan diberi definisi mesin, pengkomputeran adalah jujukan gabungan-gabungan unsur aturcara dengan keadaan ingatan mesin yang sepadan. Misalnya, aturcara *P* dengan label-label $l_i \in L$, komputeran *P* atas mesin *M* dengan keadaan-keadaan ingatan $v_i \in V$, ditulis *mP*, ialah jujukan unsur-unsur $L \cdot V$,

$$(l_1, v_1), (l_2, v_2), \ldots, (l_j, v_j), \ldots, (l_n, v_n).$$

Pengkomputeran menamat jika l_n adalah label penamat. Secara formal, pengkomputeran *P* atas *M*,

fungsi $mP: X \to Y$

$$\begin{array}{ll} \text{utk } x \in X & (X \text{ set masukan}) \\ \quad v_1 = I_M(x) & (I_M \text{ fungsi masukan bagi } M) \\ \quad mP(x) = O_M(v_n) & (O_M \text{ fungsi keluaran bagi } M) \end{array}$$

Dengan demikan, dua aturcara *P* setara *Q* atas *M* jika

$$mP(x) = mQ(x), \quad \forall x \in X$$

dan *P* dan *Q* setara secara kuat,

$$P \equiv Q \quad \text{jika} \quad mP = mQ \quad \forall M$$

Dan, *P* tamat atas *M* jika $mP(x)$ terdefinisi, $\forall x \in X$. Dari segi kebetulan aturcara pula,

P <u>separa betul</u> terhadap φ dan ψ (atas *M*)

jika $\psi(x, mP(x))$ benar $\forall x \in X$

utk $\varphi(x)$ benar dan $mP(x)$ terdefinisi

(perlu tamat hanya untuk yang memenuhi syarat)

sementara

P <u>sepenuhnya betul</u> terhadap φ dan ψ (atas *M*)

jika $mP(x)$ terdefinisi

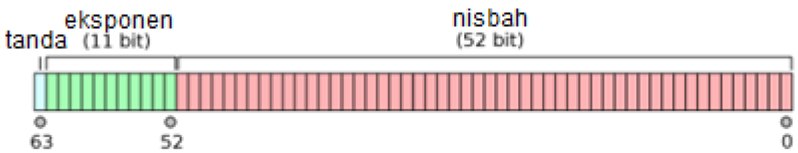
dan $\psi(x, mP(x))$ benar $\forall x \in X$ utk $\varphi(x)$ benar

di mana φ merupakan syarat-syarat tentang apa yang diinginkan, dan ψ merupakan. apa yang didapati. *P* sepenuhnya betul bila *P* tamat utk semua kes, dan separa betul jika penamatan tidak disyaratkan bagi kes-kes yang tidak memenuhi syarat.

0.0.4 Kekangan perkakasan

Perkakasan yang digunakan memberi kekangan ke atas pengkomputeran yang dibuat.

Isu petama ialah tentang perwakilan nombor di atas mesit dedua. Nombor-nombor perlu diwakilkan pada akhirnya dalam bentuk digit-digit dedua atau bit. Saiz dan ketepatan nombor yang dapat diwakilkan juga bergantung kepada saiz data yangd digunakan sistem. Misalnya, bagi sistem 64 bit, nilai integer bertanda terbesar yang boleh diwakilkan ialah 2^{63} bagi penterjemahan terus dedua. Bagi nombor titik apungan, pewakilan nombor nyata, yang ditulis $s \times 10^{e}$, s nisbah (0 hingga 1) dan e eksponen. Bagi sistem 64 bit, biasanya 11 bit dirizab untuk eksponen e dan 52 bit untuk nisbah s:



Ini bermakna bilangan digit terhad untuk s, yang membawa kepada ketepatan terhad, dan saiz terhad untuk e, yang mengahadkan saiz nombor yang boleh diwakili.

Satu pertimbangan lagi ialah kelajuan sistem. Mungkin sistem sekarang boleh melakukan 10^{10} operasi/saat. Dalam masa terhad diberikan, hanya bilangan tertentu operasi dapat dijalankan.

Saiz ingatan sistem yang terhad juga menjadi kekangan. Ini bermakna ia hanya dapat tangani masalah dengan saiz terhad. Misalnya, dalam simulasi bahan, perlu dipertimbangkan bahawa bilangan zarah dalam sampel munasabah cecair atau plasma ialah $N >> 10^{6}$ (lebih kepada nombor Avogadro, 10^{23}) dan satu megabait hanya boleh menyimpan 10^{6} data. Juga, bilangan salingtindak di antara zarah-zarah ini, yang perlu dikira menerusi operasi aritmetik, ialah ½ $N(N$-1).

Walaupun kita ada suatu sistem yang cukup pantas untuk membuat bilangan operasi yang banyak dengan cepat, tenaga yang digunakan, dan yang dilepaskan, perlu diambilkira, dalam era dunia pemanasan sejagat yang ada sekarang. Apa pun, kos tenaga, dan kesediaan tenaga boleh mengekang banyaknya pengiraan yang boleh dibuat.

0.1 Pengiraan berkomputer

Senibina komputer cocok untuk dilakukan penilaian fungsi, ulangan dan keputusan logik.

Cara kiraan fungsi boleh didefinisikan dalam sesuatu aturcara, maka seterusnya nilai fungsi ini boleh dinilaikan untuk apa-apa argumen. Penilaian fungsi ini boleh dilakukan lagi dan lagi untuk pelbagai argumen, tanpa perlu menuliskannya lagi secara tersurat.

Penilisan aturcara juga membolehkan ulangan dibuat tanpa kelesuan. Kiraan atau operasi tertentu boleh diulangkan beribu atau juta kali, bergantung kepada kelajuan sistem.

Perjalanan aturcara juga mempunyai kebolehan membuat keputusan logik. Dengan itu, perjalanan aturcara boleh ditentukan oleh nilai logik sesuatu fungsi, membolehkan pelakuan terkawal.

0.2 Kaedah berangka asas

Senibina komputer cocok untuk perjalanan beberapa jenis aturcara yang boleh digunakan untuk mengira penghampiran dan lain-lain menerusi apa yang dikatakan kaedah berangka.

0.2.1 Pembezaan

Pembeza merupakan operasi ke atas fungsi selanjar, yang menghasilkan keluaran fungsi selanjar juga. Siri Taylor memberi kembangan nilai suatu fungsi pada kedudukan berhampiran:

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2!}f''(x_0) + \frac{h^3}{3!}f'''(x_0) + \cdots$$

Jadi,

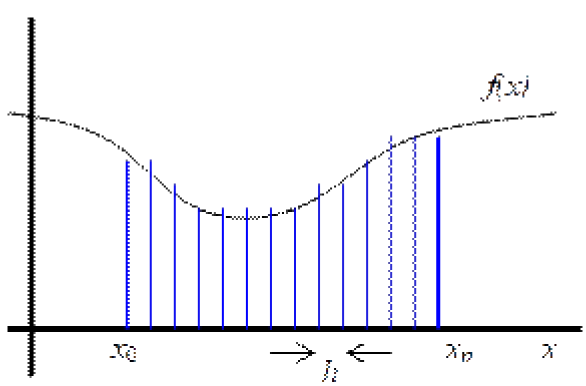
$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} + O(h)$$

Dengan itu, pembezaan *f* dinilaikan pada x_0, boleh dihampirkan sebagai $(f(x_0 + h) - f(x_0))/h$ dengan ralat peringkat pertama $O(h)$, menerusi kiraan aritmetik yang membabitkan penilaian fungsi.

0.2.2 Kamiran

Banyak masalah membabitkan kamiran, misalnya penyelesaian sesetengah persamaan pembezaan. Ada bentuk fungsi yang boleh dikamirkan secara analisisan, namun banyak yang tidak. Kamiran berangka ialah kaedah penghampiran yang digunakan untuk mengamir fungsi-fungsi begini, dengan didiskretkan pembolehubah kamiran, dan mengubah kamiran menjadi hasiltambah luas bahagian-bahagian diskret. Lebih kecil kuantum pendiskretan, lebih tepat penghampiran, namun memerlukan lebih banyak pengiraan. Namun operasi ulangan seperti ini boleh ditangani mudah oleh komputer.

Ringkasnya, nilai kamiran dihampirkan dengan didiskretkan pembolehubah kamiran, kemudian dihasiltambah luas ‘papan’-‘papan’diskret tadi. Petua trapezium, misalnya, menganggapkan setiap papan sebagai satu trapezium:



Luas trapezium ke *i* ialah

$$\frac{h}{2}(f_{i-1}+f_i)$$

di mana $f_i = f(x_i)$, memerikan anggaran kamiran:

$$\int_{x=x_0}^{x=x_n} f(x)dx \cong \sum_{i=1,n} \frac{h}{2}(f_{i-1}+f_i) = \frac{h}{2}(f_0 + f_n + 2(f_1 + f_2 + \ldots + f_{n-1}))$$

Di sini, bahagian lengkung di hujung papan dianggarkan sebagai segmen garis lurus, maka anggaran ini hanya tepat kepada $O(h)$, memeberi ralat $O(h^2)$.

Petua Simpson merupakan anggaran yanglebih tepat, di mana lengkung hujung papan dipadankan sebagai parabola. Ini memerlukan 3 titik (iaitu 2 papan berjiranan digabung). $2n$ papan diperlukan. Luas gabungan papan-papan ke $2i$-1 dan $2i$ ialah

$$\frac{h}{3}(f_{2i-2} + 4f_{2i-1} + f_{2i})$$

Maka anggaran kamiran ialah

$$\int_{x=x_0}^{x=x_n} f(x)dx \cong \sum_{i=1,n} \frac{h}{3}(f_{2i-2} + 4f_{2i-1} + f_{2i}) = \frac{h}{3}(f_0 + f_{2n} + 4(f_1 + f_3 + \ldots + f_{2n-1}) + 2(f_2 + f_4 + \ldots + f_{2n-2}))$$

Dengan bahagian lengkung dianggarkan sebagai segmen parabola, ini tepat kpd $O(h^2)$, ataupun ia punyai ralat $O(h^4)$.

Pengiraan kamiran menerusi petua trapezium dan petua Simpson ini boleh dibuat oleh komputer, dengan menggunakan gelung ulangan untuk membuat hasiltambah, dan penilaian nilai fungsi melalui panggilan fungsi.

Kamiran Romberg ialah kaedah yang menyeluruhkan petua trapezium dan petua Simpson, di mana penghampiran berturutan dibuat. Pertimbangkan petua trapezium untuk $n=2^k$ papan yg memberikan anggaran $I \approx I_0^{(k)}$ dengan ralat $O(h^2)$. Jadi,

$$I = I_0^{(k)} + \mu_0 h^2$$

dan, bila dihiris dua papan-papan ini, iaitu utk 2^{k+1} papan (dengan lebar $h/2$),

$$I = I_0^{(k+1)} + \mu_0 (h/2)^2$$

Menggantikan untuk μ_0 dalam persamaan-persamaan memberikan I, yang kita tulis

$$I_1^{(k)} = (1/3)(4\, I_0^{(k+1)} - I_0^{(k)}).$$

Ini sebenarnya ialah petua Simpson, dengan ralat $O(h^4)$. Gunakan kaedah sama, kini untuk petua Simpson, kita tuliskan,

$$I = I_1^{(k)} + \mu_1 h^4$$

dan

$$I = I_1^{(k+1)} + \mu_1 (h/2)^4$$

Menggantikan untuk μ_1 kini memberikan

$$I_2^{(k)} = (1/15)(16\, I_1^{(k+1)} - I_1^{(k)})$$

dengan ralat $O(h^6)$. Secara amnya, secara menyeluruh,

$$I_n^{(k)} = (4^n\, I_{n-1}^{(k+1)} - I_{n-1}^{(k)})/(4^n-1)$$

dgn ralat $O(h^{2n+2})$. Dalam kaedah Romberg, kita hanya perlu mengira $I_0^{(k)}$ bagi berlainan k, kemudiannya kita boleh kira $I_1^{(k)}$ dan seterusnya, dengan ketepatan yang bertambah.

Contoh:

$$\int_1^2 \frac{dx}{x} = \ln 2 = 0.693147181$$

2^k	$I_0^{(k)}$ (Trapezium)	$I_1^{(k)}$ (Simpson)	$I_2^{(k)}$	$I_3^{(k)}$	$I_4^{(k)}$
1	0.750000000				
2	0.708333333	0.694444444			
4	0.697023809	0.693253967	0.693174603		
8	0.694121851	0.693154532	0.693147901	0.693147479	
16	0.693391202	0.693147652	0.693147193	0.693147182	0.693147181

Pengiraan $I_1^{(k)}$ dan seterusnya boleh dibuat oleh komputer secara lelaran atau rekursi.

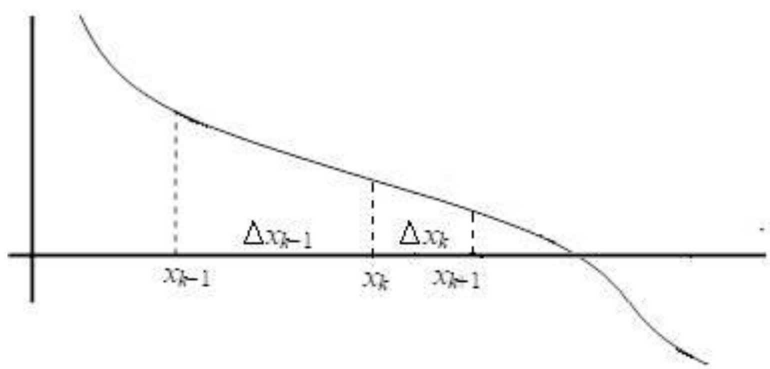
0.2.3 Punca

Pencarian punca membabitkan penyelesaian $f(x) = 0$ untuk x.

Carian punca boleh dibuat secara lelaran mengikut algoritma berikut:

> Baiki anggaran $x \to x + \Delta x$ ke arah $f(x)$ mengurang, dengan tekaan rambang Δx.
> Baiki lagi, dengan $\Delta x \to \Delta x/2$ misalnya. Ubah tanda Δx kalau f berubah tanda.
> Ulang seterusnya.

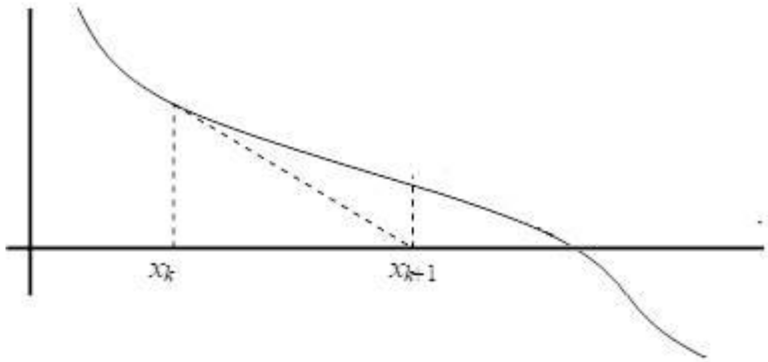
Ini digambarkan dalam rajah di bawah:



Dalam kaedah Newton, lengkung $f(x)$ dan tangennya, digunakan untuk memilih Δx, berturutan:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

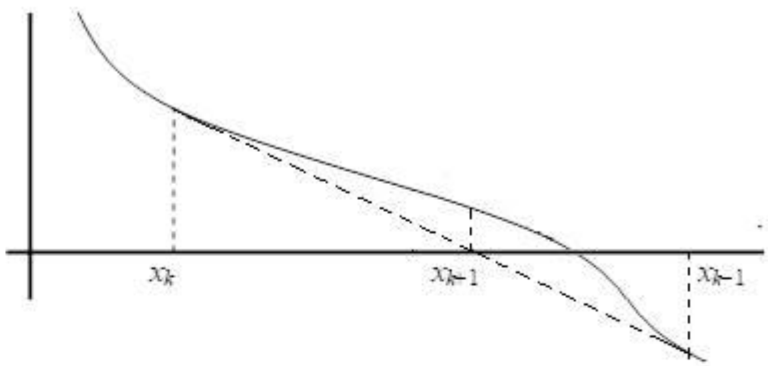
Ini digambarkan seperti berikut:



Kaedah sekan pula tidak memerlu kiraan kecerunan. Kita gunakan

$$x_{k+1} = x_k - f(x_k)\frac{(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}$$

iaitu kita meneka punca mmenerusi anggaran garis lurus di antara dua tekaan sebelumnya,



Untuk kedua-dua kaedah ini, tekaan punca dibuat berulang-ulang: penyelesaian dibuat komputer secara lelaran. Perlu ditentukan bahawa proses ini menumpu.

Tulis dalam bentuk $x = F(x)$. Untuk penyelesaian, dilelarkan $x_{n+1} = F(x_n)$ sehingga menumpu. Proses menumpu jika $|F'(x)| < 1$.

Contoh: mahu cari punca bagi $x^3=9$.
Persamaan lelaran $x_{n+1} = 9x_n^{-2}$ tidak menumpu
$x_{n+1} = 3\ x_n^{-1/2}$ menumpu
kepada nilai 2.080

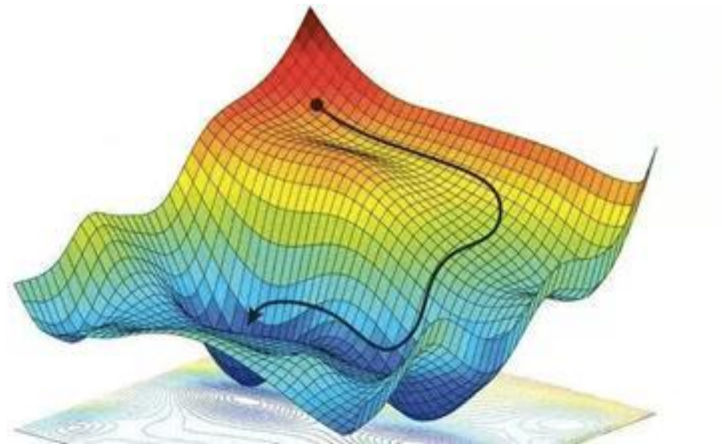
0.2.4 Ekstremum

Pencarian ekstremum ialah penentian **x** di mana $f(\mathbf{x})$ maksimum/minimum. Ini ialah serupa dengan pencarian punca fungsi pembezaan. Bezakan $f(\mathbf{x}) \rightarrow f\,'(\mathbf{x})$, dan kemudian cari punca $f\,'(\mathbf{x})$.

Satu kaedah pencarian ekstremum ialah kaedah panjat bukit/turun cerun. Turun cerun ialah apabila tekaan titik ekstremum diubah dalam arah menurut kecerunan $f(\mathbf{x})$ dan begitulah sebaliknya. Tekaan untuk titik ekstremum diubah dengan $\Delta\mathbf{x}$, dengan

$$\Delta\mathbf{x} = \pm\alpha\boldsymbol{\nabla}f$$

Ini diulang sehingga semua arah tak benarkan, iaitu apabila $\Delta\mathbf{x}$ dalam semua arah tidak lagi punyai tanda yang asal. Kadar perubahan, a boleh berubah. Nilai yang kecil menyebabkan kadar penumpuan yang perlahan, sementara kadar yang besar boleh menyebabkan tekaan 'terlajak'. Penggambaran pencarian minium menggunakan kaedah turun bukit adalah seperti berikut:



Selain cara tekaan yang tertentu, suatu kaedah berkebarangkalian boleh digunakan. Perubahan $\Delta\mathbf{x}$ dijana menurut kebarangkalian tertentu, misalnya kebarangkalian Boltzmann dengan mengiaskan ∇f sebagai keupayaan. Jadi $\Delta\mathbf{x}$ berlaku dengan kebarangkalian

$$\propto e^{-\Delta E/kT}$$

dengan

$$\Delta E = \Delta\mathbf{x} \cdot \boldsymbol{\nabla} f \ .$$

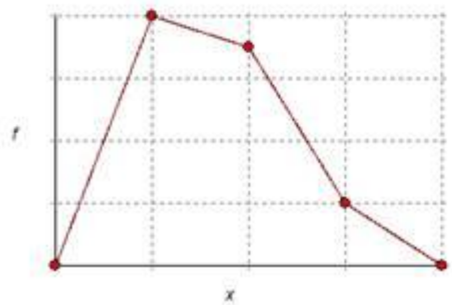
Kebarangkalian bergantung kepada nilai 'suhu' T yang boleh dikawal. Suhu yang kecil hampir melarang perubahan berlwanan turun cerun, namun bila suhu bernilai besar, ada kebarangkalian perubahan dalam $\mathbf{x}$ yang melawan cerun. 'Sepuhlindap' tersimulasi ialah meniru proses sepuhlindap bahan di mana bahan dipanaskan dan kemudian disejukkan secara perlahan supaya aturan terbentuk. Suhu awal yang tinggi membenarkan pelepasan daripada tertangkap dalam minimum tempatan untuk mencari minimum sejagat.

0.2.5 Penentudalaman

Penentudalaman ialah tabiran maklumat tempatan, diberi data diskret/tak lengkap. Misalnya, mahu dianggarkan titik data antara dua titik yang diberikan.

Katakan dikehendaki anggaran $f(x)$, diberi $f_i \equiv f(x_i)$ untuk beberapa titik x_i.

Kaedah termudah ialah penentudalaman linear, di mana dipadankan garislurus kepada dua titik berjujukan, seperti berikut:



Anggaran garislurus ini diberikan

$$f(x) = f_i + \frac{(x-x_i)}{(x_{i+1}-x_i)}(f_{i+1} - f_i) + \varepsilon$$

di antara x_i dan x_{i+1}, dengan ralat ε.

Anggaran yang lebih baik ialah penentudalaman Lagrange, di mana padankan parabola kepada tiga titik berjujukan. Jika dibuat sedemikian, didapati:

$$f(x) = \frac{x-x_{i+1}}{x_i-x_{i+1}} f_i + \frac{x-x_i}{x_{i+1}-x_i} f_{i+1} + \varepsilon = \sum_{k=i}^{i+1} f_k P_k^{(1)}(x) + \varepsilon$$

di mana

$$P_k^{(1)}(x) = \frac{x-x_j}{x_k-x_j}$$

dengan $j \neq k$. Ini boleh diseluruhkan, memberi ungkapan am berikut untuk lengkung tertib (n-1) melalui n titik,

$$f(x) = \sum_{k=1}^{n} f_k P_k^{(n-1)}(x) + \varepsilon$$

dengan

$$P_k^{(n-1)}(x) = \prod_{j\neq k}^{n} \frac{x-x_j}{x_k-x_j}.$$

Dengan kaedah Aitken pula, penentudalaman Lagrange (cekap) dilakukan menerusi jujukan penentudalaman linear:

Gunakan

$$f_{i\ldots j} = \frac{x-x_j}{x_i-x_j} f_{i\ldots(j-1)} + \frac{x-x_i}{x_j-x_i} f_{(i+1)\ldots j}$$

Kira f_1, f_2, …., f_n.
Dengan itu kira f_{12}, f_{23}, …, $f_{(n-1)n}$.
Kemudian f_{123}, f_{234}, … .
Seterusnya,
akhirnya perolehi $f_{123\ldots n}$.

0.2.6 Penghampiran

Diberi data diskret atau tak lengkap, perlu ditakbirkan maklumat sejagat. Misalnya, diberi (f_i,x_i), diinginkan penghampiran kepada $f(x)$ yang memberikan titik-titik ini.

Kaedah yang paling popular ialah kaedah kuasadua terkecil. Ingin didapatkan $f(x)$ yang paling hampir kepada (f_i,x_i). Bagi pendekatan mudah, $f(x)$ boleh dipilih sebagai polinomial tertib m (dengan $m < n$, bilangan titik yang diberi):

$$p_m(x) = \sum_{k=0}^{m} a_k x^k$$

Pilih a_k yang meminimumkan

$$\Delta[a_k] = \sum_{i=1}^{n} [p_m(x_i) - f_i]^2$$

menerusi kaedah peminimuman. Ini memberikan penghampiran kuasadua terkecil polinomial tertib m, $p_m(x)$ untuk $f(x)$.

0.2.7 Kiraan Fungsi, Nombor Rawak

Suatu kecekapan komputer ialah dalam mengira fungsi. Setelah fungsi didefinisikan, panggilan fungsi ini dapat menilaikannya untuk pelbagai argumen, berkali-kali.

Panggilan fungsi boleh juga digunakan secara lelaran atau rekursi. Ini di mana kiraan sebelumnya digunakan. Misalnya, kiraan factorial:

```
fakt(x) = x*fakt(x-1)
```

Pengiraan fungsi bolehjadi rumit atau tiada punyai kaedah langsung. Satu cara untuk pengiraan fungsi seperti ini ialah dengan mengembangkannya sebagai siri. Misalnya, fungsi sinus diberikan sebagai siri Taylor adalah seperti berikut:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

Jika sebutan dalam siri itu makin mengecil, nilai fungsi itu dapat dianggarkan dengan menghasiltambahkan terma-terma siri sehingga had tertentu. Yakni, siri boleh dipangkas jika ia menumpu. Ralat penilaian sebegini ialah pada peringkat nilai terma yang seterusnya. Pengiraan terma-terma individu juga seperti panggilan fungsi yang diulang-ulangkan.

Satu panggilan fungsi yang istimewa ialah permintaan nombor rawak. Fungsi nombor rawak biasanya suatu fungsi terbina dalam ang ada disediakan. Biasanya hasil panggilan bukan memberikan nombor rawak tulen, tetapi dijana nombor pseudorawak yang seragam. Nombor rawak dijana menurut turutan secara lelaran. Daripada suatu benih yang menjadi mulaan turutan, nombor-nombor lain dijana. Benih atau mulaan yang sama memberikan turutan nombor-nombor sama selepas itu.

Kita mahukan kala panjang, iaitu jujukan panjang sebelum turutan dimulai semula setelah nombor yang pernah dijana, dijana sekali lagi.

Kita juga mahu kerawakan, iaitu korelasi (n titik),

$$\langle x_{i_1}, x_{i_2}, \ldots, x_{i_n} \rangle$$

yang rendah. Plot x_{i+1} lawan x_i seharusnya seragam.

Juga, untuk penggunaan praktik, pengiraan harus laju.

0.3 Kriteria pengiraan berangka

Pengiraan berangka membabitkan penghampiran. Kaedah pengiraan berangka perlu dipertimbangkan berdasarkan kriteria tertentu. Yang utamanya, perlu dilihat dari segi;

- Kekonsistenan
  Persamaan pembezaan dianggarkan menerusi pendiskretan ruang dan/atau masa, menjadi persamaan perbezaan, dengan kuantum pendiskretan h. Dalam had $h \to 0$, persamaan perbezaan harus kembali menjadi persamaan pembezaan asal.

- Ketepatan
  Ketaktepatan boleh terakibat daripada pendiskretan, dan lain-lain sumber termasuk:
  - Pembundaran nombor nyata
  - Ralat pangkasan, yang datang daripada pangkasan siri dalam penganggaran. Ini bergantung kepada skema perbezaan.
  - Keadaan tidak sihat di mana ada situasi nilai dibahagi dengan suatu nombor kecil, dan situasi tolakan dua nombor besar yang hampir sama nilainya

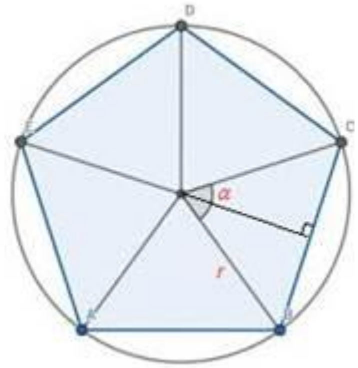
- Kestabilan
  Walau pun ralat pada suatu langkah pengiraan itu kecil, ralat mungkin berkumpul atau bertumpuk, dan membesar.
  Jika $\varepsilon(n+1) = g\varepsilon(n)$ di mana $\varepsilon(n)$ ialah ralat pada langkah masa ke-n, maka g ialah faktor gandaan. Untuk kestabilan, perlukan $|\varepsilon(n+1)| < |\varepsilon(n)|$ bila $n \to \infty$, iaitu $|g| \geq 1$.

- Kecekapan
  Kiraan yang dibuat perlulah laju. Pengiraan lebih mudah lebih baik, termasuk tidak membabitkan pengiraan fungsi-fungsi kompleks.

Tugasan 1

1. Guna kaedah turun cerun untuk mencari minimum bagi $f(x) = x^2 - 4x$.
   Anda boleh gunakan kod ditulis dalam bahasa Python.

2. Guna kaedah Achimedes untuk menganggarkan nilai π.



Hampirkan perimeter poligon biru sebagai perimeter bulatan iaitu $2\pi r$.

Kira
perimeter poligon menggunakan trigonometri, mengaitkannya dengan r, yang begantung kepada bilangan sisi poligon n.

Plot anggaran untuk p lawan n.