PERTEMUAN 8

REPRESENTASI BILANGAN, OPERASI ARITMATIKA, *ERROR* DAN PROPAGASI *ERROR*

TUJUAN PRAKTIKUM

Mahasiswa mampu memahami dan menjelaskan representasi bilangan, operasi aritmatika, *error* dan propagasi *error* pada program R.

TEORI PENUNJANG

Representasi Bilangan pada Komputer

Sistem bilangan yang sering digunakan dalam perhitungan sehari-hari adalah sistem desimal. Dalam representasi desimal, bilangan dinyatakan dalam bentuk

$$a_n a_{n-1} \ldots a_1 a_0 . a_{-1} a_{-2} \ldots a_{-m}$$

$$= a_n .10^n + a_{n-1} .10^{n-1} + \ldots + a_1 .10^1 + a_0 .10^0 + a_{-1} .10^{-1} + a_{-2} .10^{-2} + \ldots + a_{-m} .10^{-m}$$

dimana setiap koefisien (digit) adalah salah satu dari 0, 1, 2, 3, … , 9. Kebanyakan komputer melakukan aritmatika menggunakan sistem bilangan biner daripada menggunakan bilangan desimal. Dalam sistem bilangan biner, bilangan integer direpresentasikan dalam bentuk

$$a_n 2^n + a_{n-1} 2^{n-1} + \ldots + a_1 2^1 + a_0 .2^0$$

di mana koefisien-koefisien adalah digit biner atau bit, yaitu 0 atau 1. Bentuk tersebut sering digunakan untuk merepresentasikan integer positif dalam komputer.

Bilangan *Floating Point*

Selain sistem bilangan biner, komputer menggunakan sistem floating-point untuk representasi bilangan. Bilangan floating-point digunakan untuk aplikasi *scientific* dan rekayasa. Sistem bilangan biner umumnya digunakan untuk merepresentasikan bilangan integer. Sedangkan sistem floating-point digunakan untuk merepresentasikan bilangan real. Cara baku untuk menyatakan bilangan real yang dinamakan notasi *scientific*, adalah dengan menggeser titik desimal dan memberikan pangkat yang sesuai pada basis bilangan. Notasi scientific adalah sistem floating-point desimal. Sebagai contoh :

$0.000739 = 7.39 \, x \, 10^{-4}$

$65.7891 = 6.57891 \times 10$

$88000000 = 8.8 \, x \, 10^{7}$

Untuk sistem bilangan berbasis β, sebuah bilangan *x* dapat ditulis sebagai berikut

$$x = \pm f \times \beta^{\pm E} \qquad (1)$$

dimana E adalah integer tak negatif dan *f* adalah bilangan real. Bentuk (1) adalah representasi floating-point basis β untuk *x* dengan pangkat ±E dan fraksi atau mantisa, F. Bentuk (1) dapat dituliskan dalam bentuk lain yaitu

$$x = \pm(f\beta) \times \beta^{\pm E-1} \qquad (2)$$

untuk memperoleh representasi yang unik, komputer menggunakan format floating-point yang dinormalisasi, dimana x dinyatakan dalam bentuk (1) dengan

$$\frac{1}{\beta} \le f < 1 \qquad (3)$$

Dalam representasi floating-point yang dinormalisasi, bilangan x dituliskan sebagai

$$x = \pm(0.f_1 f_2 \ldots f_k) \times \beta^{\pm E} \qquad (4)$$

dimana basis β adalah integer yang memenuhi

$$1 \le f_1 < \beta \text{ dan } 0 \le f_i < \beta \qquad (5)$$

$$0.f_1 f_2 \ldots f_k = f_1\beta^{-1} + f_2\beta^{-2} + \ldots + f_k\beta^{-k} \qquad (6)$$

Dalam komputer tertentu,representasi floating-point dari bilangan biner *x* dinyatakan dalam bentuk

$$x = \delta.\bar{x}.2^{e} \qquad (7)$$

dimana δ = +1 atau –1, *e* adalah integer, dan $\bar{x}$ adalah fraksi biner yang memenuhi

$$(.1)_2 \le \bar{x} < 1 \qquad (8)$$

Sebagai contoh, jika $x = (1101.10111)_2$, maka δ = +1 , *e* = 4 = $(100)_2$ dan $\bar{x}$ = $(.110110111)_2$.

Rounding* dan *Chopping

Jika bilangan *x* dalam (7) tidak memiliki mantissa $\bar{x}$ yang sesuai dengan banyaknya n bit yang tersedia, maka $\bar{x}$ harus dipendekan. Pemendekan ini dilakukan dalam dua cara. Cari yang paling sederhana adalah memotong (*truncate*) $\bar{x}$ menjadi n bit, digit sisanya diabaikan. Cara yang kedua adalah membulatkan (*round*) $\bar{x}$ menjadi n digit, berdasarkan ukuran dari bagian $\bar{x}$ yang mengikuti digit n. Jika digit ke n+1 adalah nol, potong (*chop*) $\bar{x}$ menjadi n digit, selainnya *chop* $\bar{x}$ ke n digit dan tambahkan 1 ke digit terakhir.

Error

Error dalam kuantitas terhitung didefinisikan sebagai

error = nilai sebenarnya – nilai pendekatan

Error absolut didefinisikan sebagai

$$E_t = |\,nilai\ sebenarnya - nilai\ pendekatan\,|$$

error relative adalah ukuran dari *error* terhadap nilai sebenarnya, dinyatakan sebagai

Error relative absolut dinyatakan

$$\varepsilon_t = \left| \frac{nilai\ sebenarnya - nilai\ \ pendekatan}{nilai\ sebenarnya} \right| \times 100$$

Sumber *Error*

Error biasanya terjadi dalam proses penyelesaian masalah matimatika yang dilakukan melalui prosedur komputasi. Berikut ini adalah penyebab munculmya *error*.

1. Pemodelan matematika untuk mempresentasikan masalah nyata.
2. Data fisik yang mengandung *error* observasi.
3. *Error* mesin yang terjadi ketika kalkulasi menggunakan kalkulator atau komputer. Sebagai contoh adalah *error rounding* atau *error chopping*. *Error* ini tidak dapat dihindari ketika menggunakan aritmatika floating-point.
4. *Error* pendekatan matematika.

Loss of Significance Error

Perhatikan evaluasi dari

$$f(x) = \sqrt{x+2}\ - \sqrt{x} \qquad (9)$$

untuk nilai x yang menaik. Hasil dengan menggunakan kalkulator desimal 6 digit ditunjukkan dalam Tabel 1

Tabel 1 Nilai f(x) pada persamaan (9) menggunakan kalkulator 6 digit

x	Computed $f(x)$	True $f(x)$
1	0.732050	0.732051
10	0.301820	0.301824
100	0.099500	0.099505
1000	0.031600	0.031607

Sebagaimana x naik, terdapat lebih sedikit digit akurasi dalam nilai terhitung dari $f(x)$ .Perhatikan untuk $x = 100$, pada kalkulator

$$\sqrt{100} = 10.0000 \qquad \sqrt{102} = 10.0995$$

Nilai yang pertama adalah nilai eksak, dan yang kedua adalah nilai yang dibulatkan ke 6 digit signifikan untuk akurasi. Kemudian,

$$\sqrt{x+2} - \sqrt{x} = \sqrt{102} - \sqrt{100} = 0.0995 \qquad (10)$$

Nilai sebenarnya haruslah 0.099501. Perhitungan dalam (10) dikatakan memiliki *loss-of-significance error*. Tiga digit akurasi dalam $\sqrt{x+2} = \sqrt{102}$ dihilangkan oleh pengurangan dari digit yang sesuai dalam $\sqrt{x} = \sqrt{100}$. Untuk $f(x)$ (9), *loss of significance error* dapat dihindari dengan memformulasikan kembali bentuk $f(x)$. Perhatikan bahwa fungsi (9) dapat dinyatakan sebagai berikut :

$$f(x) = x\frac{\sqrt{x+2}-\sqrt{x}}{1}.\frac{\sqrt{x+2}+\sqrt{x}}{\sqrt{x+2}+\sqrt{x}} = \frac{x}{\sqrt{x+2}+\sqrt{x}} \quad ...(11)$$

Dengan menggunakan kalkulator dengan 6 digit desimal, persamaan (11) memberikan

$$f(100) = 0.099505$$

yang merupakan jawaban sebenarnya untuk 6 digit..

Propagasi *Error*

Jika kalkulasi dilakukan dengan bilangan yang mengandung *error*, maka solusi yang dihasilkan akan dipengaruhi *error* tersebut. Untuk operasi aritmatika biasa, misalkan x_A dan y_A menyatakan bilangan-bilangan yang digunakan dalam kalkulasi, dan misalkan x_T dan y_T adalah nilai sebenarnya kita ingin membatasi

$$\mathrm{E} = (x_T\ \omega y_T) - (x_A\ \omega\, y_A) \quad ...(12)$$

dimana ω menyatakan salah satu dari operasi " $+$ ", " $-$ ", " . " atau " \ ". *Error* E dinamakan *propagated error* (*error* yang berpropagasi).

LAPORAN PENDAHULUAN

1. Jelaskan secara singkat bagaimana representasi bilangan integer dan real dalam komputer.
2. Apa yang dimaksud dengan *error*, *error* absolut, *error* absolut relatif, *loss of significance error*, dan propagasi *error*.
3. Jelaskan secara singkat sumber-sumber *error* dan berikan contoh kasusnya.

MATERI PRAKTIKUM

1. Representasi bilangan integer dan real dalam R.
2. Buatlah fungsi sederhana menggunakan R untuk menghitung error absolut (beri nama fungsi tersebut dengan error_absolut) dan *error relative* absolut (beri nama fungsi tersebut dengan error_relative).
3. Diberikan fungsi pada persamaan (9). Untuk fungsi tersebut:
   a. Buatlah fungsi dalam R untuk persamaan tersebut, beri nama fungsi tersebut dengan fungsi9
   b. Buatlah Tabel 1 dalam R menggunakan fungsi9 yang telah dibuat pada materi 3a.

c. Hitunglah nilai fungsi pada persamaan (9) untuk x = 100 dan x = 101, dengan perhitungan menggunakan 6 digit signifikan.

d. Nilai sebenarnya dari fungsi pada persamaan (9) adalah 0.0498756. Dengan menggunakan fungsi error_absolut dan error_relative pada materi no 2, hitunglah *error absolut* dan *error relative absolut* untuk hasil yang diperoleh dari materi 3c.

e. Jelaskan secara singkat mengapa *loss of significance error* terjadi dalam perhitungan ini.

4. Lakukan langkah-langkah dalam materi praktikum no 3 untuk fungsi pada persamaan (11).

5. Diketahui δx *random error* untuk variabel *x*. Jika *z* = *f*(*x*) untuk suatu fungsi *f*(), maka *error* dalam fungsi *f*(*x*) dihitung sebagai berikut:

$$\delta z = |f'(x)|\delta x \qquad (13)$$

a. Buatlah fungsi dalam R untuk menghitung persamaan (13), beri nama fungsi tersebut dengan propagasi_error. Gunakan Deriv yang tersedia dalam R untuk menghitung turunan fungsi.

b. Dengan menggunakan fungsi yang dibuat pada soal 5a, hitung propagasi *error* dalam menghitung volume kubus yang memiliki sisi 3 cm dengan kesalahan pengukuran ± 0.02 cm.

DAFTAR PUSTAKA

1. Victor A. Bloomfield. 2014. *Using R for Numerical Analysis in Science and Engineering*. 1 edition. Chapman and Hall/CRC
2. Steven C. 2011. *Applied Numerical Methods W/MATLAB: for Engineers & Scientists*. 3 edition. McGraw-Hill