

Laporan Tugas Pemrograman B - Kelompok 28

Nama: Alfonsus Tanara Gultom **NPM:** 2306267126

Abstrak

Tugas pemrograman ini berfokus pada penerapan metode numerik Aturan Trapesium Komposit untuk menyelesaikan studi kasus estimasi jarak tempuh berdasarkan data kecepatan diskrit. Sebuah aplikasi dikembangkan menggunakan bahasa pemrograman C++ untuk mengimplementasikan metode tersebut. Data input berupa vektor waktu dan kecepatan yang bersesuaian. Program menghitung integral numerik dari fungsi kecepatan terhadap waktu untuk memperoleh estimasi total jarak yang ditempuh. Hasil perhitungan menunjukkan bahwa Aturan Trapesium Komposit dapat memberikan aproksimasi jarak tempuh yang berguna dari data kecepatan diskrit. Laporan ini mendokumentasikan latar belakang, metode yang digunakan, data, implementasi program, hasil, dan analisisnya, serta menyertakan kode sumber yang terdokumentasi dalam repositori GitHub.

Pendahuluan

Dalam berbagai disiplin ilmu teknik, pengukuran dan analisis data pergerakan suatu objek merupakan hal yang fundamental. Salah satu parameter kunci dalam analisis pergerakan adalah jarak total yang ditempuh oleh objek tersebut selama interval waktu tertentu. Secara matematis, jarak tempuh dapat diperoleh dengan mengintegrasikan fungsi kecepatan objek terhadap waktu. Namun, dalam banyak aplikasi praktis, data kecepatan seringkali tidak tersedia dalam bentuk fungsi kontinu, melainkan sebagai serangkaian titik data diskrit yang diperoleh dari pengukuran pada interval waktu tertentu.

Ketika dihadapkan pada data diskrit, metode analitik untuk integrasi tidak dapat langsung diterapkan. Di sinilah metode numerik memainkan peran penting. Metode numerik menyediakan pendekatan untuk mengaproksimasi nilai integral dari suatu fungsi berdasarkan titik-titik data yang diketahui. Salah satu metode integrasi numerik yang paling dasar dan banyak digunakan adalah Aturan Trapesium (Trapezoidal Rule).

Tugas pemrograman ini bertujuan untuk menerapkan Aturan Trapesium Komposit, sebuah pengembangan dari Aturan Trapesium dasar, untuk mengestimasi jarak tempuh suatu objek berdasarkan data kecepatan diskrit. Sebuah program komputer akan dikembangkan menggunakan bahasa pemrograman C++ untuk mengimplementasikan algoritma Aturan Trapesium Komposit. Studi kasus yang diangkat adalah menghitung estimasi jarak tempuh dari sekumpulan data pasangan waktu dan kecepatan.

Laporan ini akan membahas latar belakang teoritis mengenai integrasi numerik dan Aturan Trapesium, menjelaskan data yang digunakan sebagai input, memaparkan implementasi algoritma dalam program C++, menyajikan hasil eksperimen, serta melakukan analisis terhadap hasil yang diperoleh. Selain itu, laporan ini juga menyertakan tautan ke repositori GitHub yang berisi kode sumber program yang dikembangkan.

Studi Literatur

Integrasi numerik adalah teknik fundamental dalam analisis numerik yang digunakan untuk mengaproksimasi nilai integral tentu dari suatu fungsi, terutama ketika solusi analitik sulit atau tidak mungkin diperoleh, atau ketika fungsi hanya diketahui pada titik-titik diskrit. Berbagai metode telah dikembangkan untuk tujuan ini, mulai dari metode sederhana hingga yang lebih kompleks dengan tingkat akurasi yang lebih tinggi [1].

Salah satu metode integrasi numerik yang paling dasar dan intuitif adalah Aturan Trapesium (Trapezoidal Rule). Metode ini bekerja dengan mengaproksimasi area di bawah kurva fungsi sebagai serangkaian trapesium. Untuk interval tunggal $[a, b]$, integral $\int_a^b f(x) dx$ diaproksimasi sebagai luas trapesium dengan alas $(b-a)$ dan sisi sejajar $f(a)$ dan $f(b)$ [1].

Untuk meningkatkan akurasi, interval integrasi $[a, b]$ dapat dibagi menjadi beberapa subinterval yang lebih kecil, dan Aturan Trapesium diterapkan pada setiap subinterval. Pendekatan ini dikenal sebagai Aturan Trapesium Komposit. Semakin banyak subinterval yang digunakan (atau semakin kecil lebar subinterval), semakin baik aproksimasi integralnya, dengan asumsi fungsi berperilaku baik [1].

Penerapan Aturan Trapesium Komposit sangat relevan dalam kasus di mana data diperoleh secara eksperimental atau melalui sensor pada interval waktu diskrit. Contoh klasik adalah estimasi jarak tempuh dari data kecepatan. Jika $v(t)$ adalah kecepatan sebagai fungsi waktu, maka jarak tempuh D dari waktu $t=a$ hingga $t=b$ diberikan oleh integral $D = \int_a^b v(t) dt$. Dengan data kecepatan diskrit (t_i, v_i) , Aturan Trapesium Komposit dapat digunakan untuk mengaproksimasi nilai integral ini [1]. Buku "Numerical Methods for Engineers" oleh Chapra dan Canale [1] menyediakan penjelasan mendalam mengenai metode ini dan aplikasinya dalam konteks rekayasa.

Penjelasan Data Yang Digunakan

Untuk mendemonstrasikan aplikasi metode Aturan Trapesium Komposit dalam estimasi jarak tempuh, digunakan satu set data hipotetis yang merepresentasikan pengukuran kecepatan suatu objek pada titik-titik waktu tertentu. Data ini didefinisikan langsung di dalam kode sumber program C++ sebagai dua buah vektor: `time` untuk menyimpan titik waktu pengukuran (dalam detik) dan `velocity` untuk menyimpan kecepatan yang terukur pada waktu tersebut (dalam meter per detik).

Data yang digunakan dalam contoh implementasi adalah sebagai berikut:

- **Vektor Waktu (t):** {0.0, 1.0, 2.5, 3.0, 4.8, 5.0, 6.2} detik
- **Vektor Kecepatan (v):** {0.0, 5.0, 7.0, 6.5, 8.0, 8.5, 9.0} meter/detik

Tabel berikut menyajikan data tersebut dalam format yang lebih terstruktur:

Waktu (t) (detik)	Kecepatan (v) (m/s)
0.0	0.0
1.0	5.0
2.5	7.0
3.0	6.5
4.8	8.0
5.0	8.5
6.2	9.0

Data ini sengaja dipilih dengan interval waktu yang tidak seragam ($t[i+1] - t[i]$ tidak konstan) untuk menunjukkan fleksibilitas metode Aturan Trapesium Komposit yang tidak mensyaratkan interval data yang sama. Vektor waktu harus dipastikan terurut menaik, yang merupakan syarat agar perhitungan lebar setiap trapesium ($h = t[i+1] - t[i]$) menghasilkan nilai positif dan perhitungan integral berjalan dengan benar. Program C++ yang dikembangkan juga menyertakan validasi untuk memastikan kondisi ini terpenuhi.

Penjelasan Metode Yang Digunakan

Metode numerik yang diimplementasikan dalam program C++ untuk mengestimasi jarak tempuh adalah **Aturan Trapesium Komposit (Composite Trapezoidal Rule)**. Metode ini dipilih karena kesederhanaannya, kemudahan implementasinya, dan kemampuannya untuk menangani data dengan interval yang tidak seragam, seperti yang sering ditemui dalam pengukuran dunia nyata.

Prinsip dasar Aturan Trapesium adalah mengaproksimasi integral dari suatu fungsi $f(x)$ pada interval $[a, b]$ dengan menghitung luas sebuah trapesium yang dibentuk oleh titik $(a, f(a))$, $(b, f(b))$, $(a, 0)$, dan $(b, 0)$. Luas trapesium ini diberikan oleh:

$$\int [a,b] f(x) dx \approx (b - a) * [f(a) + f(b)] / 2$$

Dalam Aturan Trapesium Komposit, interval total integrasi $[t_0, t_{n-1}]$ (dimana n adalah jumlah titik data - 1) dibagi menjadi n subinterval $[t_i, t_{i+1}]$. Aturan Trapesium kemudian diterapkan pada setiap subinterval ini, dan hasilnya dijumlahkan untuk mendapatkan aproksimasi integral total. Untuk data diskrit (t_i, v_i) dengan interval waktu yang tidak harus sama, rumus Aturan Trapesium Komposit menjadi:

$$\text{Jarak} \approx \int [t_0, t_{n-1}] v(t) dt \approx \sum_{i=0}^{n-2} \left[(v_i + v_{i+1}) / 2 \right] * (t_{i+1} - t_i)$$

Dimana: * t_i adalah waktu pada titik data ke- i * v_i adalah kecepatan pada titik data ke- i * n adalah jumlah subinterval (jumlah titik data - 1) * $(t_{i+1} - t_i)$ adalah lebar subinterval ke- i * $(v_i + v_{i+1}) / 2$ adalah rata-rata kecepatan pada subinterval ke- i

Implementasi dalam program C++ dilakukan melalui fungsi `trapezoidal_composite`. Fungsi ini menerima dua vektor, `t` (waktu) dan `v` (kecepatan), sebagai input. Fungsi tersebut melakukan iterasi dari $i = 0$ hingga $n-2$ (dimana n adalah jumlah total titik data). Dalam setiap iterasi, fungsi menghitung lebar interval $h = t[i+1] - t[i]$ dan luas trapesium untuk segmen tersebut menggunakan $(v[i] + v[i+1]) / 2.0 * h$. Luas segmen ini kemudian ditambahkan ke variabel `integral` yang mengakumulasi total estimasi jarak tempuh.

Program juga menyertakan pemeriksaan validitas input data, seperti memastikan ukuran vektor waktu dan kecepatan sama, terdapat minimal dua titik data untuk membentuk setidaknya satu trapesium, dan memastikan vektor waktu terurut secara menaik ($t_{i+1} > t_i$) untuk menghindari lebar interval negatif atau nol.

Diskusi dan Analisa Hasil Experimen

Program C++ yang mengimplementasikan Aturan Trapesium Komposit dijalankan menggunakan data waktu dan kecepatan yang telah dijelaskan pada bagian sebelumnya. Eksekusi program menghasilkan output berupa estimasi total jarak yang ditempuh oleh objek selama interval waktu 0.0 detik hingga 6.2 detik.

Berdasarkan data input: * Waktu (t): {0.0, 1.0, 2.5, 3.0, 4.8, 5.0, 6.2} *
Kecepatan (v): {0.0, 5.0, 7.0, 6.5, 8.0, 8.5, 9.0}

Program menghitung luas trapesium untuk setiap subinterval: 1. Interval [0.0, 1.0]: Luas = $(1.0 - 0.0) * (0.0 + 5.0) / 2 = 2.5$ 2. Interval [1.0, 2.5]: Luas = $(2.5 - 1.0) * (5.0 + 7.0) / 2 = 1.5 * 6.0 = 9.0$ 3. Interval [2.5, 3.0]: Luas = $(3.0 - 2.5) * (7.0 + 6.5) / 2 = 0.5 * 6.75 = 3.375$ 4. Interval [3.0, 4.8]: Luas = $(4.8 - 3.0) * (6.5 + 8.0) / 2 = 1.8 * 7.25 = 13.05$ 5. Interval [4.8, 5.0]: Luas = $(5.0 - 4.8) * (8.0 + 8.5) / 2 = 0.2 * 8.25 = 1.65$ 6. Interval [5.0, 6.2]: Luas = $(6.2 - 5.0) * (8.5 + 9.0) / 2 = 1.2 * 8.75 = 10.5$

Total estimasi jarak tempuh adalah jumlah dari luas semua trapesium: Jarak Total $\approx 2.5 + 9.0 + 3.375 + 13.05 + 1.65 + 10.5 = \mathbf{40.075 \text{ meter}}$

Program C++ yang dijalankan akan menghasilkan nilai ini. Hasil ini merupakan aproksimasi dari integral fungsi kecepatan terhadap waktu. Akurasi dari metode Aturan Trapesium bergantung pada beberapa faktor, termasuk seberapa "halus" fungsi kecepatan yang sebenarnya dan lebar dari subinterval (Δt_i). Semakin kecil lebar subinterval (semakin banyak titik data), umumnya akurasi akan semakin meningkat. Dalam kasus ini, karena kita hanya memiliki data diskrit, hasil 40.075 meter adalah estimasi terbaik yang bisa didapatkan menggunakan metode Aturan Trapesium dengan data yang tersedia.

Kelebihan metode ini adalah kemudahannya dalam implementasi dan kemampuannya menangani interval waktu yang tidak seragam. Namun, perlu diingat bahwa Aturan Trapesium mengasumsikan bahwa fungsi berperilaku linear di antara titik-titik data. Jika fungsi kecepatan sebenarnya memiliki lengkungan yang signifikan di antara titik pengukuran, metode ini mungkin menghasilkan error yang cukup besar dibandingkan metode orde lebih tinggi seperti Aturan Simpson (yang memerlukan interval seragam atau perlakuan khusus). Untuk studi kasus ini, dengan asumsi data kecepatan yang diberikan cukup representatif, hasil estimasi jarak dianggap memadai.

Kesimpulan

Tugas pemrograman ini telah berhasil mengimplementasikan metode numerik Aturan Trapesium Komposit menggunakan bahasa C++ untuk mengestimasi jarak tempuh

berdasarkan data kecepatan diskrit. Program yang dikembangkan mampu menerima input data waktu dan kecepatan dengan interval yang tidak seragam dan menghasilkan estimasi jarak tempuh sebesar 40.075 meter untuk data contoh yang diberikan.

Penerapan Aturan Trapesium Komposit terbukti menjadi pendekatan yang praktis dan mudah diimplementasikan untuk kasus di mana fungsi kecepatan tidak diketahui secara analitik, melainkan hanya berupa titik-titik data. Meskipun memiliki keterbatasan dalam akurasi dibandingkan metode orde tinggi jika fungsi asli sangat non-linear di antara titik data, metode ini memberikan hasil estimasi yang berguna dan memadai untuk banyak aplikasi rekayasa dasar.

Pengembangan program ini tidak hanya memberikan solusi untuk studi kasus yang spesifik tetapi juga memperkuat pemahaman mengenai konsep dasar integrasi numerik dan penerapannya dalam penyelesaian masalah rekayasa praktis menggunakan pemrograman komputer.

Link Github

Kode sumber lengkap untuk program C++ yang dijelaskan dalam laporan ini tersedia di repositori GitHub publik berikut:

<https://github.com/alfons07/Tugas-Pemrograman-B-AlfonsusTanara.git>

Referensi

[1] S. C. Chapra and R. P. Canale, Numerical Methods for Engineers, 6th ed. New York: McGraw-Hill, 2010.

[2] P. J. Davis and P. Rabinowitz, Methods of Numerical Integration, 2nd ed. New York: Academic Press, 1984. [3] K. E. Atkinson, An Introduction to Numerical Analysis, 2nd ed. New York: John Wiley & Sons, 1989. [4] R. L. Burden and J. D. Faires, Numerical Analysis, 9th ed. Boston, MA: Brooks/Cole, Cengage Learning, 2011. [5] K. Hussain, K. S. Nisar, and F. Ismail, "An improved Trapezoidal rule for numerical integration," J. Phys.: Conf. Ser., vol. 2090, no. 1, p. 012104, Dec. 2021. doi: 10.1088/1742-6596/2090/1/012104. [6] "Trapezoidal rule," Wikipedia, The Free Encyclopedia. Wikimedia Foundation, Inc. Accessed: May 29, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Trapezoidal_rule