

LAPORAN METODE BEDA HINGGA
Persamaan Gelombang 1 Dimensi dengan
Sumber Gelombang



Disusun Oleh :

Ivan Hartana
6161901119

Program Studi Matematika
Fakultas Teknologi Informasi dan Sains
Universitas Katolik Parahyangan
Bandung
2021

Daftar Isi

1	PENDAHULUAN	1
1.1	Latar Belakang	1
1.2	Rumusan Masalah	2
1.3	Tujuan	2
2	LANDASAN TEORI	3
2.1	Persamaan Diferensial Biasa (<i>Ordinary Differential Equation</i>)	3
2.2	Metode Beda Hingga (<i>Finite Difference Method</i>)	3
2.2.1	Beda Maju (<i>Forward Difference</i>)	4
2.2.2	Beda Mundur (<i>Backward Difference</i>)	4
2.2.3	Beda Pusat atau Beda Tengah (<i>Centered Difference</i>)	4
2.3	Persamaan Diferensial Parsial (<i>Partial Differential Equations</i>)	4
2.4	Persamaan Gelombang Satu Dimensi dengan Sebuah Sumber Gelombang (<i>One-Dimensional Waves Equation with A Source</i>)	7
3	PEMBAHASAN	9
3.1	Persamaan Diferensial Biasa	9
3.1.1	Metode Analitik	9
3.1.2	Metode Numerik	11
3.1.3	Perbandingan Metode Analitik dan Metode Numerik Persamaan Diferensial Biasa	20
3.2	Persamaan Diferensial Parsial	22
3.2.1	Metode Analitik	22
3.2.2	Metode Numerik	22
3.2.3	Perbandingan Metode Analitik dan Metode Numerik Persamaan Diferensial Parsial	23
4	Penutup	26
4.1	Kesimpulan	26
4.2	Saran	26
5	Daftar Pustaka	27
6	Lampiran	28

1 PENDAHULUAN

1.1 Latar Belakang

Dalam penyelesaian berbagai permasalahan matematis, salah satu metode yang digunakan untuk menyelesaikan permasalahan tersebut adalah dengan metode analitik. Namun terkadang proses penyelesaian tersebut cukup rumit dan tidak efisien dalam waktu pengerjaannya. Maka metode lain yang dapat digunakan dalam penyelesaian permasalahan ini yaitu metode numerik.

Metode numerik adalah teknik penyelesaian permasalahan matematis dengan pengaplikasian operasi aritmatik.^[7] Berkebalikan dengan metode analitik, metode ini dapat digunakan untuk menghampiri atau menyelesaikan hampir semua jenis permasalahan dengan kekurangan solusi dari metode numerik ini berupa hampiran sehingga akan terdapat galat dalam solusi metode numerik.^[7] Selain itu, karena sifat metode numerik yang mengaplikasikan operasi aritmatik yang bisa berulang-ulang maka dengan menggunakan bantuan program untuk dilakukan *looping*, solusi hampiran dapat dicari dengan cepat.^[7]

Salah satu metode numerik yang dapat digunakan dalam penyelesaian persamaan diferensial adalah metode beda hingga. Metode beda hingga sering digunakan dalam menyelesaikan Persamaan Diferensial Biasa (*Ordinary Differential Equations*) atau Persamaan Diferensial Parsial (*Partial Differential Equations*) dengan cara menghampiri solusi persamaan diferensial menggunakan beda atau selisih. Metode beda hingga ini dilakukan dengan cara iterasi atau secara rekursif sehingga terbentuk hubungan rekursi-nya.

Dalam laporan ini, metode analitik dan metode numerik beda hingga akan diimplementasikan dalam bahasa pemrograman *python* untuk menyelesaikan Persamaan Diferensial Biasa dan Persamaan Diferensial Parsial dimana metode numerik akan menggunakan perintah perulangan (*looping*). Kita akan melihat bagaimana aplikasi bahasa pemrograman *python* untuk masing-masing metode pada penyelesaian Persamaan Diferensial Biasa dan menggunakan permasalahan Persamaan Diferensial Parsial untuk kedua metode dengan dibandingkan hasil solusi. Selain itu, kita membandingkan hasil dari solusi persamaan menggunakan metode numerik (metode beda hingga) dengan solusi eksak persamaan yang terdapat pada halaman 71 buku *Partial Differential Equations: An Introduction* (edisi yang kedua) yang ditulis oleh Walter A. Strauss. Kita memilih topik yang berjudul "Persamaan Gelombang Satu Dimensi dengan Sebuah Sumber Gelombang" (*One-Dimensional Waves Equation with A Source*) untuk proyek ini.

1.2 Rumusan Masalah

1. Bagaimana penerapan metode analitik dan numerik dalam menyelesaikan persamaan gelombang.
2. Bagaimana perbandingan hasil persamaan gelombang 1 dimensi dengan solusi eksak halaman 71 buku *Partial Differential Equations: An Introduction* (edisi yang kedua) yang ditulis oleh Walter A. Strauss dengan topik "Persamaan Gelombang Satu Dimensi dengan Sebuah Sumber Gelombang".
3. Bagaimana membuat kode program dalam *python* mengenai metode yang digunakan dalam menyelesaikan permasalahan yang diberikan.

1.3 Tujuan

1. Dapat mengetahui penerapan metode analitik dan metode numerik dalam menyelesaikan persamaan gelombang.
2. Dapat membandingkan hasil persamaan gelombang 1 dimensi dengan solusi eksak halaman 71 buku *Partial Differential Equations: An Introduction* (edisi yang kedua) yang ditulis oleh Walter A. Strauss dengan topik "Persamaan Gelombang Satu Dimensi dengan Sebuah Sumber Gelombang".
3. Dapat membuat kode program dalam *python* mengenai metode yang digunakan dalam menyelesaikan permasalahan yang diberikan.

2 LANDASAN TEORI

2.1 Persamaan Diferensial Biasa (*Ordinary Differential Equation*)

Persamaan diferensial biasa merupakan persamaan yang melibatkan turunan biasa. Turunan biasa di sini merupakan turunan dari suatu fungsi satu variabel (peubah). Selanjutnya, **orde** dari persamaan diferensial biasa adalah derajat tertinggi dari turunan yang ada di suatu persamaan. Bentuk umum persamaan diferensial biasa adalah

$$F(t, y, y', y'', y''', \dots, y^{(n)}) = 0. \quad (1)$$

Orde dari persamaan diferensial untuk persamaan (1) adalah n . Lalu, bentuk umum persamaan diferensial biasa (PDB) **linear** orde- n adalah

$$a_n(t)y^{(n)} + a_{n-1}(t)y^{(n-1)} + \dots + a_2(t)y'' + a_1(t)y' + a_0(t)y = g(t). \quad (2)$$

Persamaan diferensial yang tidak berbentuk seperti persamaan (2) merupakan persamaan diferensial biasa non-linear. Jenis persamaan diferensial berikutnya ada persamaan diferensial homogen (*homogeneous*). Apabila pada persamaan (2), $g(t) = 0$ maka bentuk umum persamaan diferensial homogen adalah ^[5]

$$y^{(n)} + p_1(t)y^{(n-1)} + \dots + p_{n-1}(t)y' + p_n(t)y = 0. \quad (3)$$

Ada juga jenis persamaan diferensial yang lain yaitu persamaan diferensial nonhomogen (*nonhomogeneous*). Apabila pada persamaan (2), $g(t) \neq 0$ maka bentuk umum persamaan diferensial nonhomogen adalah ^[5]

$$y^{(n)} + p_1(t)y^{(n-1)} + \dots + p_n(t)y = g(t). \quad (4)$$

2.2 Metode Beda Hingga (*Finite Difference Method*)

Metode beda hingga adalah suatu metode numerik untuk menghampiri solusi dari persamaan diferensial biasa atau parsial dengan menghampiri turunan pada persamaan diferensial. Misalkan suatu fungsi $u(x)$ dari 1 variabel dan partisi interval yaitu Δx , maka penulisan nilai $u(j\Delta x)$ untuk $x = j\Delta x$ dengan suatu angka u_j yaitu:^[5]

$$u_j \sim u(j\Delta x).$$

Kemudian, 3 standar aproksimasi untuk turunan pertama $\frac{\partial u}{\partial x}(j\Delta x)$ dibagi ke 3 sub-bagian berikut:

2.2.1 Beda Maju (*Forward Difference*)

$$\text{The Forward Difference Method : } \frac{u_{j+1} - u_j}{\Delta x}$$

2.2.2 Beda Mundur (*Backward Difference*)

$$\text{The Backward Difference Method : } \frac{u_j - u_{j-1}}{\Delta x}$$

2.2.3 Beda Pusat atau Beda Tengah (*Centered Difference*)

$$\text{The Centered Difference Method : } \frac{u_{j+1} - u_{j-1}}{2\Delta x}$$

Aproksimasi paling sederhana dengan menggunakan turunan kedua:

$$\text{Centered Second Difference : } u''(j\Delta x) \approx \frac{u_{j+1} - 2u_j + u_{j-1}}{(\Delta x)^2}$$

2.3 Persamaan Diferensial Parsial (*Partial Differential Equations*)

Persamaan diferensial parsial merupakan persamaan yang melibatkan operator turunan parsial (∂) sehingga ada bentuk turunan parsial (contoh: $\partial u / \partial x$ bisa ditulis sebagai u_x). Persamaan diferensial parsial adalah persamaan diferensial bagi fungsi peubah banyak (fungsi yang bergantung pada lebih dari satu variabel bebas/*independen*). Orde dari persamaan diferensial parsial adalah turunan parsial tertinggi yang muncul pada persamaan diferensial parsial tersebut. Sekarang, lihat bentuk paling umum dari persamaan diferensial parsial orde satu untuk fungsi dengan dua peubah adalah $u(x, y)$.

$$F(x, y, u(x, y), u_x(x, y), u_y(x, y)) = F(x, y, u, u_x, u_y) = 0. \quad (5)$$

Selanjutnya, bentuk paling umum dari persamaan diferensial parsial orde dua bagi fungsi dengan dua peubah $u(x, y)$ adalah

$$F(x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}) = 0. \quad (6)$$

Solusi dari baik persamaan diferensial parsial (7) maupun persamaan diferensial parsial (8) adalah sebuah fungsi $u(x, y)$ yang memenuhi persamaan diferensial parsial tersebut untuk suatu daerah di bidang- xy . Beberapa contoh persamaan diferensial parsial adalah^[4]

1. Persamaan Gelombang/*Wave Equation*.

$$u_{tt} = c^2 u_{xx}$$

2. Persamaan Panas/Persamaan Difusi/*Heat Equation/Diffusion Equation*.

$$u_t = k u_{xx}$$

3. Persamaan Transport.

$$u_x + u_t = 0$$

4. Persamaan Transport.

$$u_x + t u_t = 0$$

5. Persamaan Laplace.

$$u_{xx} + u_{yy} = 0$$

6. Persamaan Poisson.

$$u_{xx} + u_{yy} = f(x, y)$$

7. Persamaan Burgers, merupakan bentuk khusus persamaan Buckley-Leverett.

$$u_x + u u_t = 0$$

8. Mekanika Kuantum.

$$u_t - i u_{xx} = 0$$

dengan $i = \sqrt{-1}$.

Selanjutnya, ada klasifikasi persamaan diferensial parsial. Perhatikan bentuk umum persamaan diferensial parsial berikut ini.

$$A u_{xx} + 2B u_{xy} + C u_{yy} = F(x, y, u, u_x, u_y). \quad (7)$$

Berdasarkan nilai A , B , dan C , ada tiga tipe persamaan diferensial parsial:

1. Jika $AC - B^2 > 0$ maka persamaan diferensial parsial bertipe **eliptik** (*elliptic*).

2. Jika $\boxed{AC - B^2 = 0}$ maka persamaan diferensial parsial bertipe **parabolik** (*parabolic*).
3. Jika $\boxed{AC - B^2 < 0}$ maka persamaan diferensial parsial bertipe **hiperbolik** (*hyperbolic*).

Berikut ini adalah beberapa contoh menentukan tipe dari persamaan diferensial parsial:^[4]

1. Persamaan Laplace.

$$\boxed{u_{xx} + u_{yy} = 0}$$

Kita punya nilai $A = 1$, $B = 0$, dan $C = 1$. Maka, $AC - B^2 = 1 \cdot 1 - 0^2 = 1 - 0 = 1 > 0$ sehingga **persamaan Laplace** merupakan persamaan diferensial parsial (PDP) bertipe **eliptik**.

2. Persamaan Panas/Persamaan Difusi.

$$\boxed{u_y = ku_{xx}}$$

Kita memiliki nilai $A = k$ dan $B = C = 0$. Lalu, $AC - B^2 = k \cdot 0 - 0^2 = 0 - 0 = 0$ sehingga persamaan panas/persamaan difusi merupakan persamaan diferensial parsial (PDP) bertipe **parabolik**.

3. Persamaan Gelombang.

$$\boxed{u_{yy} = c^2 u_{xx}}$$

Pertama, kita bisa tulis ulang persamaan menjadi

$$c^2 u_{xx} - u_{yy} = 0.$$

Sekarang, kita mempunyai nilai $A = c^2$, $B = 0$, dan $C = -1$. Maka, $AC - B^2 = c^2(-1) - 0^2 = -c^2 - 0 = -c^2 < 0$ (asumsikan $c \neq 0$.) sehingga persamaan gelombang merupakan persamaan diferensial parsial (PDP) bertipe **hiperbolik**.

2.4 Persamaan Gelombang Satu Dimensi dengan Sebuah Sumber Gelombang (*One-Dimensional Waves Equation with A Source*)

Bentuk umum dari persamaan gelombang satu dimensi dengan satu sumber gelombang adalah

$$u_{tt} - c^2 u_{xx} = f(x, t) \quad (8)$$

dimana sumber gelombang $f(x, t)$ dapat dianggap sebagai suatu gaya luar (*external force*) yang bekerja pada sebuah senar bergetar dengan panjang tak hingga. Kemudian, syarat awal (*initial conditions*) untuk persamaan (3) adalah

$$\begin{aligned} u(x, 0) &= \phi(x) \\ u_t(x, 0) &= \psi(x) \end{aligned} \quad (9)$$

Apabila ruas kiri para persamaan (3) bernilai nol, maka akan menjadi bentuk persamaan gelombang yaitu:

$$u_{tt} = c^2 u_{xx} \quad (10)$$

Di berbagai domain yang sudah dipelajari seperti domain tak-hingga (*infinite*) dan domain setengah tak-hingga (*half-infinite*). Solusi umum untuk persamaan (5) yang sering ditemui adalah **solusi D'Alembert**. **Solusi D'Alembert** merupakan solusi umum untuk persamaan gelombang pada domain tak-hingga. Bentuk solusi umum dari persamaan gelombang pada domain tak-hingga (yaitu **solusi D'Alembert**) adalah

$$u(x, t) = \frac{1}{2}[\phi(x + ct) + \phi(x - ct)] + \frac{1}{2c} \int_{x-ct}^{x+ct} \psi(s) ds \quad (11)$$

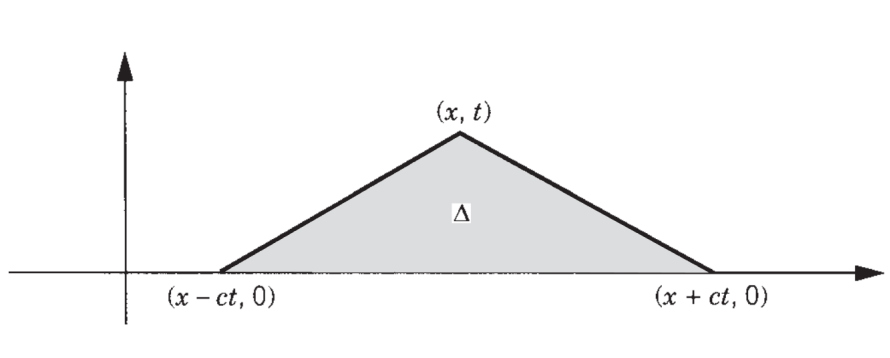
Sekarang, kita perhatikan kembali persamaan (13). Karena

$$\mathcal{L} = \partial_t^2 - c^2 \partial_x^2$$

merupakan operator linear, maka solusi eksak-nya akan berupa penjumlahan dari tiga suku, yaitu satu suku untuk fungsi ϕ , satu suku untuk fungsi ψ , dan satu suku untuk fungsi f . Jadi, solusi eksak-nya akan berupa solusi yang unik atau solusi tunggal yaitu

$$u(x, t) = \frac{1}{2}[\phi(x + ct) + \phi(x - ct)] + \frac{1}{2c} \int_{x-ct}^{x+ct} \psi(s) ds + \frac{1}{2c} \iint_{\Delta} f \quad (12)$$

dimana Δ merupakan segitiga karakteristik.



Gambar 2.4: Segitiga Karateristik.

Daerah berbentuk segitiga yang diarsir yaitu daerah segitiga karakteristik Δ disebut domain ketergantungan. Daerah segitiga karakteristik di atas dibatasi oleh sepasang garis karakteristik (*characteristic lines*) yang melalui titik (x, t) . Kemudian, interval $(x - ct, x + ct)$ disebut dengan interval ketergantungan dari titik (x, t) pada $t = 0$. *Double integral* pada persamaan (7) sama dengan integral lipat dua sehingga kita bisa menuliskan

$$\iint_{\Delta} f = \int_0^t \int_{x-ct}^{x+ct} f(y, s) dy ds \quad (13)$$

Akibatnya, bentuk solusi umum untuk persamaan gelombang satu dimensi dengan satu sumber gelombang adalah

$$u(x, t) = \frac{1}{2}[\phi(x + ct) + \phi(x - ct)] + \frac{1}{2c} \int_{x-ct}^{x+ct} \psi(s) ds + \frac{1}{2c} \int_0^t \int_{x-ct}^{x+ct} f(y, s) dy ds$$

3 PEMBAHASAN

3.1 Persamaan Diferensial Biasa

3.1.1 Metode Analitik

Berikut akan diselesaikan permasalahan roket dimana fungsi y adalah ketinggian roket dan akan digunakan metode beda hingga untuk menyelesaikan persamaan diferensial

$$\frac{d^2y}{dt^2} = -g,$$

dengan kondisi batas $y(0) = 0$ dan $y(5) = 50$ beserta $g = 9.8m/s^2$. Akan dicari kecepatan awal roket diluncur dengan menggunakan metode analitik. Dari persamaan diferensial biasa, dapat ditulis ulang menjadi

$$y''(t) = -9.8$$

Bentuk $y''(t)$ dalam kasus ini biasa disebut sebagai percepatan. Kita dapat mengintegrasikan bentuk percepatan $y''(t)$ menjadi bentuk kecepatan $y'(t)$ yang kemudian dapat diintegrasikan menjadi bentuk jarak $y(t)$. Maka dapat dicari hasil integral sebagai berikut:

- Persamaan kecepatan

$$\begin{aligned}\int y''(t) dt &= -9.8 \\ y'(t) &= -9.8t + c_1\end{aligned}$$

- Persamaan jarak

$$\begin{aligned}\int y'(t) dt &= -9.8t + c_1 \\ y(t) &= -4.9t^2 + c_1t + c_2.\end{aligned}$$

Selanjutnya akan dicari nilai konstanta variabel c_1 dan c_2 . Kita dapat mencari nilai tersebut dengan mensubstitusi kondisi batas $y(0) = 0$ dan $y(5) = 50$ ke persamaan jarak yaitu sebagai berikut:

- $y(0)$

$$\begin{aligned}-4.9(0) + c_1(0) + c_2 &= 0 \\ 0 + 0 + c_2 &= 0 \\ c_2 &= 0\end{aligned}$$

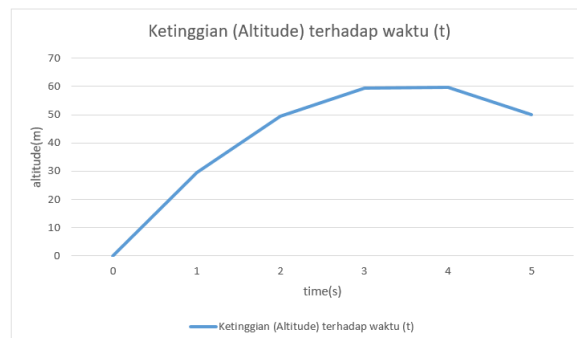
- $y(5)$

$$\begin{aligned} -4.9(25) + c_1(5) + c_2 &= 50 \\ -122.5 + 5c_1 + 0 &= 50 \\ 5c_1 &= 172.5 \\ c_1 &= 34.5. \end{aligned}$$

Setelah mendapatkan nilai konstanta variabel c_1 dan c_2 , kita dapat substitusi konstanta tersebut untuk mencari jarak ($y(t)$) dan kecepatan awal ($y'(0)$) yaitu

$$y(t) = -4.9t^2 + 34.5t,$$

sehingga dapat dibuat grafik ketinggian roket pada waktu $t = 0, 1, 2, 3, 4, 5$ sebagai berikut:



Gambar 3.1.1: Ketinggian (*altitude*) terhadap waktu (t).

Untuk kecepatan awal kita peroleh dengan

$$\begin{aligned} y'(t) &= -9.8t + 34.5 \\ y'(0) &= -9.8(0) + 34.5 \\ y'(0) &= 34.5. \end{aligned}$$

Jadi dengan metode analitik, kita mendapatkan kecepatan awal sebesar 34.5 m/s .

3.1.2 Metode Numerik

Berikut akan diselesaikan permasalahan roket dimana fungsi y adalah ketinggian roket dan akan digunakan metode beda hingga untuk menyelesaikan persamaan diferensial

$$\frac{d^2y}{dt^2} = -g,$$

dengan kondisi batas (*boundary conditions*) $y(0) = 0$ dan $y(5) = 50$ beserta $g = 9.8m/s^2$. Akan dicari kecepatan awal roket diluncur dengan menggunakan metode numerik. Kita pilih nilai n sehingga interval $0 \leq t \leq 5$ akan dibagi menjadi n subinterval yaitu dipilih $n = 10$ dengan titik awal $a = 0$ dan titik akhir $b = 5$, maka lebar setiap subinterval dapat dicari sebagai berikut:

$$\begin{aligned}h &= \frac{b - a}{n} \\h &= \frac{5 - 0}{10} \\h &= 0.5.\end{aligned}$$

Kita akan menggunakan aproksimasi turunan menggunakan metode beda hingga pusat atau beda hingga tengah untuk turunan kedua untuk interval $0 \leq t \leq 5$

$$\frac{d^2y(t)}{dt^2} \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}.$$

Aproksimasi tersebut akan disubstitusi ke persamaan diferensial sehingga diperoleh

$$\begin{aligned}\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} &= -g \\y_{i+1} - 2y_i + y_{i-1} &= h^2(-g),\end{aligned}$$

dengan $i = 1, 2, \dots, n - 1$, $y_0 = 0$, dan $y_{10} = 50$. Kemudian kita akan mencari kecepatan awal yaitu $y'(0)$. Kita punya aproksimasi turunan beda hingga tengah untuk turunan pertama yaitu

$$\frac{dy}{dx} = \frac{y_{i+1} - y_{i-1}}{2h}.$$

Maka,

$$\begin{aligned}y'(0) &= \frac{y_{0+1} - y_{0-1}}{2h} \\y'(0) &= \frac{y_1 - y_{-1}}{2h}.\end{aligned}$$

Namun nilai y_{-1} belum diketahui. Kita dapat menggunakan aproksimasi turunan kedua beda hingga pusat $\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = -g$ dengan $i = 0$ dan akan dicari nilai y_1 dari solusi persamaan diferensial untuk dicari nilai y_{-1} . Selanjutnya, kita akan mencari nilai dari $y_1, y_2, y_3, \dots, y_9$. Kita memiliki beberapa persamaan berikut ini.

- $y_0 = 0$,
- $y_{10} = 50$, dan
- $y_{i-1} - 2y_i + y_{i+1} = -gh^2$, dengan $i = 1, 2, 3, \dots, n - 1$.

Kita menggunakan nilai $g = 9,8 \text{ m/s}^2$ serta $h = 0,5$ sehingga $-gh^2 = -2,45$. Karena $n = 10$, maka $i = 1, 2, 3, \dots, 9$ sehingga kita bisa menuliskan sistem persamaan di bawah ini

- Untuk $i = 1, y_0 - 2y_1 + y_2 = -2,45$
- Untuk $i = 2, y_1 - 2y_2 + y_3 = -2,45$
- Untuk $i = 3, y_2 - 2y_3 + y_4 = -2,45$
- Untuk $i = 4, y_3 - 2y_4 + y_5 = -2,45$
- Untuk $i = 5, y_4 - 2y_5 + y_6 = -2,45$
- Untuk $i = 6, y_5 - 2y_6 + y_7 = -2,45$
- Untuk $i = 7, y_6 - 2y_7 + y_8 = -2,45$
- Untuk $i = 8, y_7 - 2y_8 + y_9 = -2,45$
- Untuk $i = 9, y_8 - 2y_9 + y_{10} = -2,45$.

Dari persamaan tersebut akan diubah menjadi sebuah perkalian dalam bentuk matriks, sehingga kita dapat menuliskannya menjadi sebagai berikut:

$$\begin{bmatrix} 1 & 0 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & & 1 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_9 \\ y_{10} \end{bmatrix} = \begin{bmatrix} 0 \\ -2.45 \\ \vdots \\ -2.45 \\ 50 \end{bmatrix}$$

Dari perkalian matriks tersebut, akan dicari solusi y untuk masing waktu t dengan menggunakan operasi baris elementer (OBE) metode *Gauss-Jordan* sehingga terbentuk matriks eselon baris tereduksi. Maka akan dibentuk matriks awal sebagai berikut

	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	$-gh^2$
1	1	0	0	0	0	0	0	0	0	0	0	0
2	1	-2	1	0	0	0	0	0	0	0	0	-2.45
3	0	1	-2	1	0	0	0	0	0	0	0	-2.45
4	0	0	1	-2	1	0	0	0	0	0	0	-2.45
5	0	0	0	1	-2	1	0	0	0	0	0	-2.45
6	0	0	0	0	1	-2	1	0	0	0	0	-2.45
7	0	0	0	0	0	1	-2	1	0	0	0	-2.45
8	0	0	0	0	0	0	1	-2	1	0	0	-2.45
9	0	0	0	0	0	0	0	1	-2	1	0	-2.45
10	0	0	0	0	0	0	0	0	1	-2	1	-2.45
11	0	0	0	0	0	0	0	0	0	0	1	50

Akan dipilih y_0 pada persamaan baris 1 sebagai pivot pada kolom pertama dan akan dieliminasi angka selain pivot menjadi 0 sehingga didapat tabel berikut

	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	$-gh^2$
1	1	0	0	0	0	0	0	0	0	0	0	0
2	0	-2	1	0	0	0	0	0	0	0	0	-2.45
3	0	1	-2	1	0	0	0	0	0	0	0	-2.45
4	0	0	1	-2	1	0	0	0	0	0	0	-2.45
5	0	0	0	1	-2	1	0	0	0	0	0	-2.45
6	0	0	0	0	1	-2	1	0	0	0	0	-2.45
7	0	0	0	0	0	1	-2	1	0	0	0	-2.45
8	0	0	0	0	0	0	1	-2	1	0	0	-2.45
9	0	0	0	0	0	0	0	1	-2	1	0	-2.45
10	0	0	0	0	0	0	0	0	1	-2	1	-2.45
11	0	0	0	0	0	0	0	0	0	0	1	50

Akan dipilih y_1 pada persamaan baris 3 sebagai pivot pada kolom kedua kemudian akan dipindah persamaan baris 2 dan persamaan baris 3 dan akan dieliminasi angka selain pivot menjadi 0 sehingga didapat tabel berikut

	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	$-gh^2$
1	1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	-2	1	0	0	0	0	0	0	0	-2.45
3	0	0	-3	2	0	0	0	0	0	0	0	-7.35
4	0	0	1	-2	1	0	0	0	0	0	0	-2.45
5	0	0	0	1	-2	1	0	0	0	0	0	-2.45
6	0	0	0	0	1	-2	1	0	0	0	0	-2.45
7	0	0	0	0	0	1	-2	1	0	0	0	-2.45
8	0	0	0	0	0	0	1	-2	1	0	0	-2.45
9	0	0	0	0	0	0	0	1	-2	1	0	-2.45
10	0	0	0	0	0	0	0	0	1	-2	1	-2.45
11	0	0	0	0	0	0	0	0	0	0	1	50

Akan dipilih y_2 pada persamaan baris 4 sebagai pivot pada kolom ketiga kemudian akan dipindah persamaan baris 3 dan persamaan baris 4 dan akan dieliminasi angka selain pivot menjadi 0 sehingga didapat tabel berikut

	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	$-gh^2$
1	1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	-3	2	0	0	0	0	0	0	-7.35
3	0	0	1	-2	1	0	0	0	0	0	0	-2.45
4	0	0	0	-4	3	0	0	0	0	0	0	-14.7
5	0	0	0	1	-2	1	0	0	0	0	0	-2.45
6	0	0	0	0	1	-2	1	0	0	0	0	-2.45
7	0	0	0	0	0	1	-2	1	0	0	0	-2.45
8	0	0	0	0	0	0	1	-2	1	0	0	-2.45
9	0	0	0	0	0	0	0	1	-2	1	0	-2.45
10	0	0	0	0	0	0	0	0	1	-2	1	-2.45
11	0	0	0	0	0	0	0	0	0	0	1	50

Akan dipilih y_3 pada persamaan baris 5 sebagai pivot pada kolom keempat kemudian akan dipindah persamaan baris 4 dan persamaan baris 5 dan akan dieliminasi angka selain pivot menjadi 0 sehingga didapat tabel berikut

	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	$-gh^2$
1	1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	-4	3	0	0	0	0	0	-14.7
3	0	0	1	0	-3	2	0	0	0	0	0	-7.35
4	0	0	0	1	-2	1	0	0	0	0	0	-2.45
5	0	0	0	0	-5	4	0	0	0	0	0	-24.5
6	0	0	0	0	1	-2	1	0	0	0	0	-2.45
7	0	0	0	0	0	1	-2	1	0	0	0	-2.45
8	0	0	0	0	0	0	1	-2	1	0	0	-2.45
9	0	0	0	0	0	0	0	1	-2	1	0	-2.45
10	0	0	0	0	0	0	0	0	1	-2	1	-2.45
11	0	0	0	0	0	0	0	0	0	0	1	50

Akan dipilih y_4 pada persamaan baris 6 sebagai pivot pada kolom kelima kemudian akan dipindah persamaan baris 5 dan persamaan baris 6 dan akan dieliminasi angka selain pivot menjadi 0 sehingga didapat tabel berikut

	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	$-gh^2$
1	1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	-5	4	0	0	0	0	-24.5
3	0	0	1	0	0	-4	3	0	0	0	0	-14.7
4	0	0	0	1	0	-3	2	0	0	0	0	-7.35
5	0	0	0	0	1	-2	1	0	0	0	0	-2.45
6	0	0	0	0	0	-6	5	0	0	0	0	-36.75
7	0	0	0	0	0	1	-2	1	0	0	0	-2.45
8	0	0	0	0	0	0	1	-2	1	0	0	-2.45
9	0	0	0	0	0	0	0	1	-2	1	0	-2.45
10	0	0	0	0	0	0	0	0	1	-2	1	-2.45
11	0	0	0	0	0	0	0	0	0	0	1	50

Akan dipilih y_5 pada persamaan baris 7 sebagai pivot pada kolom keenam kemudian akan dipindah persamaan baris 6 dan persamaan baris 7 dan akan dieliminasi angka selain pivot menjadi 0 sehingga didapat tabel berikut

	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	$-gh^2$
1	1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	-6	5	0	0	0	-36.75
3	0	0	1	0	0	0	-5	4	0	0	0	-24.5
4	0	0	0	1	0	0	-4	3	0	0	0	-14.7
5	0	0	0	0	1	0	-3	2	0	0	0	-7.35
6	0	0	0	0	0	1	-2	1	0	0	0	-2.45
7	0	0	0	0	0	0	-7	6	0	0	0	-51.45
8	0	0	0	0	0	0	1	-2	1	0	0	-2.45
9	0	0	0	0	0	0	0	1	-2	1	0	-2.45
10	0	0	0	0	0	0	0	0	1	-2	1	-2.45
11	0	0	0	0	0	0	0	0	0	0	1	50

Akan dipilih y_6 pada persamaan baris 8 sebagai pivot pada kolom ketujuh kemudian akan dipindah persamaan baris 7 dan persamaan baris 8 dan akan dieliminasi angka selain pivot menjadi 0 sehingga didapat tabel berikut

	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	$-gh^2$
1	1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	-7	6	0	0	-51.45
3	0	0	1	0	0	0	0	-6	5	0	0	-36.75
4	0	0	0	1	0	0	0	-5	4	0	0	-24.5
5	0	0	0	0	1	0	0	-4	3	0	0	-14.7
6	0	0	0	0	0	1	0	-3	2	0	0	-7.35
7	0	0	0	0	0	0	1	-2	1	0	0	-2.45
8	0	0	0	0	0	0	0	-8	7	0	0	-68.6
9	0	0	0	0	0	0	0	1	-2	1	0	-2.45
10	0	0	0	0	0	0	0	0	1	-2	1	-2.45
11	0	0	0	0	0	0	0	0	0	0	1	50

Akan dipilih y_7 pada persamaan baris 9 sebagai pivot pada kolom kedelapan kemudian akan dipindah persamaan baris 8 dan persamaan baris 9 dan akan dieliminasi angka selain pivot menjadi 0 sehingga didapat tabel berikut

	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	$-gh^2$
1	1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	-8	7	0	-68.6
3	0	0	1	0	0	0	0	0	-7	6	0	-51.45
4	0	0	0	1	0	0	0	0	-6	5	0	-36.75
5	0	0	0	0	1	0	0	0	-5	4	0	-24.5
6	0	0	0	0	0	1	0	0	-4	3	0	-14.7
7	0	0	0	0	0	0	1	0	-3	2	0	-7.35
8	0	0	0	0	0	0	0	1	-2	1	0	-2.45
9	0	0	0	0	0	0	0	0	-9	8	0	-88.2
10	0	0	0	0	0	0	0	0	1	-2	1	-2.45
11	0	0	0	0	0	0	0	0	0	0	1	50

Akan dipilih y_8 pada persamaan baris 10 sebagai pivot pada kolom kesembilan kemudian akan dipindah persamaan baris 9 dan persamaan baris 10 dan akan dieliminasi angka selain pivot menjadi 0 sehingga didapat tabel berikut

	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	$-gh^2$
1	1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	-9	8	-88.2
3	0	0	1	0	0	0	0	0	0	-8	7	-68.6
4	0	0	0	1	0	0	0	0	0	-7	6	-51.45
5	0	0	0	0	1	0	0	0	0	-6	5	-36.75
6	0	0	0	0	0	1	0	0	0	-5	4	-24.5
7	0	0	0	0	0	0	1	0	0	-4	3	-14.7
8	0	0	0	0	0	0	0	1	0	-3	2	-7.35
9	0	0	0	0	0	0	0	0	1	-2	1	-2.45
10	0	0	0	0	0	0	0	0	0	-10	9	-110.25
11	0	0	0	0	0	0	0	0	0	0	1	50

Akan dipilih y_9 pada persamaan baris 10 sebagai pivot pada kolom kesepuluh dan akan dieliminasi angka selain pivot menjadi 0 sehingga didapat tabel berikut

	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	$-gh^2$
1	1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	-0.1	11.025
3	0	0	1	0	0	0	0	0	0	0	-0.2	19.6
4	0	0	0	1	0	0	0	0	0	0	-0.3	25.725
5	0	0	0	0	1	0	0	0	0	0	-0.4	29.4
6	0	0	0	0	0	1	0	0	0	0	-0.5	30.625
7	0	0	0	0	0	0	1	0	0	0	-0.6	29.4
8	0	0	0	0	0	0	0	1	0	0	-0.7	25.725
9	0	0	0	0	0	0	0	0	1	0	-0.8	19.6
10	0	0	0	0	0	0	0	0	0	1	-0.9	11.025
11	0	0	0	0	0	0	0	0	0	0	1	50

Akan dipilih y_{10} pada persamaan baris 11 sebagai pivot pada kolom kesepuluh dan akan dieliminasi angka selain pivot menjadi 0 sehingga didapat tabel berikut

	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	$-gh^2$
1	1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	-0.1	11.025
3	0	0	1	0	0	0	0	0	0	0	-0.2	19.6
4	0	0	0	1	0	0	0	0	0	0	-0.3	25.725
5	0	0	0	0	1	0	0	0	0	0	-0.4	29.4
6	0	0	0	0	0	1	0	0	0	0	-0.5	30.625
7	0	0	0	0	0	0	1	0	0	0	-0.6	29.4
8	0	0	0	0	0	0	0	1	0	0	-0.7	25.725
9	0	0	0	0	0	0	0	0	1	0	-0.8	19.6
10	0	0	0	0	0	0	0	0	0	1	-0.9	11.025
11	0	0	0	0	0	0	0	0	0	0	1	50

Setelah operasi baris elementer (OBE) metode *Gauss-Jordan*, maka pada matriks eselon baris tereduksi terakhir, dapat dilihat semua nilai y pada waktu $i = 0, 1, 2, \dots, 10$ yaitu dapat dilihat sebagai berikut

- $y_0 = 0$.
- $y_1 = 16,025$.
- $y_2 = 29,6$.
- $y_3 = 40,725$.

- $y_4 = 49,4$.
- $y_5 = 55,625$.
- $y_6 = 59,4$.
- $y_7 = 60,725$.
- $y_8 = 59,6$.
- $y_9 = 56,025$.
- $y_{10} = 50$.

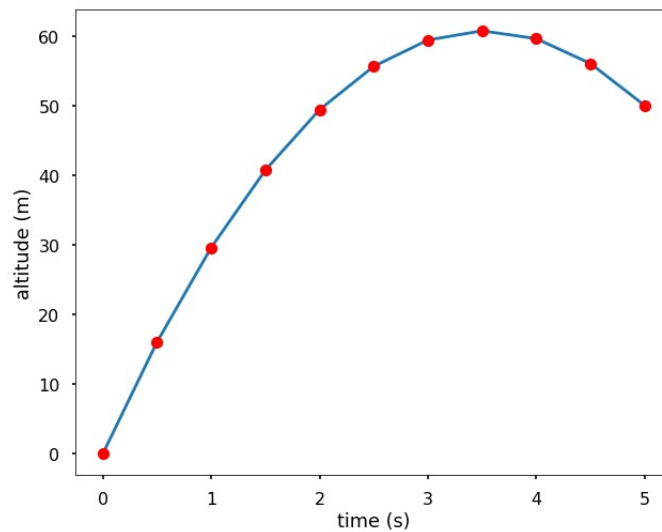
Kemudian, kita akan substitusi nilai y_0 dan y_1 ke aproksimasi persamaan turunan beda hingga turunan kedua sebelumnya untuk dicari nilai y_{-1} sehingga

$$\begin{aligned}y_{-1} &= (-g)h^2 + 2y_0 - y_1 \\y_{-1} &= (-9.8)(0.5)^2 + 2(0) - 16.025 \\y_{-1} &= -18.4749\end{aligned}$$

sehingga nilai kecepatan awal $y'(0)$ adalah

$$\begin{aligned}y'(0) &= \frac{16.025 - (-18.4749)}{2(0.5)} \\y'(0) &= 34.5\end{aligned}$$

Selain itu, kita juga dapat menggunakan program koding *python* dengan algoritma beda hingga untuk membentuk grafik trayektori ketinggian dan mendapat solusi kecepatan awal $y'(0)$ yaitu sebagai berikut:



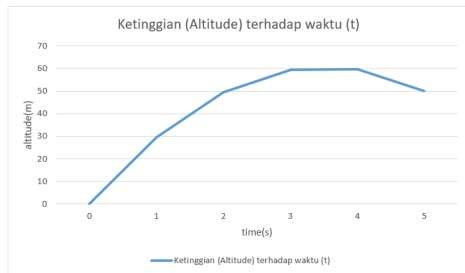
Gambar 3.1.2: Ketinggian (*altitude*) terhadap waktu (*t*).

Kecepatan Awal dari proses peluncuran adalah 34.5

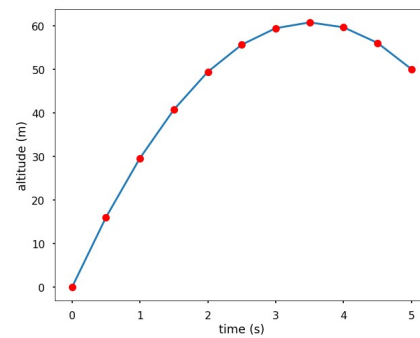
Gambar 3.1.3: Hasil kecepatan awal.

3.1.3 Perbandingan Metode Analitik dan Metode Numerik Persamaan Diferensial Biasa

Berikut ini merupakan grafik perbandingan metode analitik dan metode numerik pada persamaan diferensial biasa dimana grafik dengan metode analitik dibuat dengan mensubstitusi $t = 0, 1, 2, 3, 4, 5$ ke persamaan jarak untuk dihitung ketinggian secara manual yang kemudian diplot dengan fitur *line chart* pada aplikasi *Word* sedangkan grafik dengan metode numerik merupakan hasil kodingan python dengan metode beda hingga:



Gambar 1: Grafik Metode Analitik



Gambar 2: Grafik Metode Numerik

Perbandingan Grafik ketinggian

Jadi kedua grafik memiliki trayektori yang sama dengan ketinggian pada waktu t terakhir yaitu detik ke-5 akan berakhir di ketinggian 50 m. Tetapi dapat dilihat trayektori pada grafik analitik tidak sehalus pada grafik numerik karena hanya menggunakan 6 titik t dimana perlu dihitung manual satu per satu ketinggian pada setiap titik t dan akan menggunakan waktu yang lama sedangkan pada grafik numerik menggunakan algoritma yang sudah ada dengan bantuan program koding *python* sehingga gambar grafik akan lebih bagus dan lebih cepat untuk didapat.

Metode analitik akan memerlukan waktu yang lama dalam menghitung jarak dan kecepatan pada berbagai macam titik t yang berbeda dimana pada metode numerik dengan algoritma yang sudah ada dengan bantuan program *python* dapat dicari solusi dengan cepat baik jarak atau kecepatan pada banyak titik t berbeda.

3.2 Persamaan Diferensial Parsial

3.2.1 Metode Analitik

Kita akan menyelesaikan

$$u_{tt} = c^2 u_{xx}, \text{ dengan } u(x, 0) = \cos(2\pi x) \text{ dan } u_t(x, 0) = 0$$

Solusi persamaan gelombang diatas dapat dicari dengan menggunakan solusi umum dari persamaan gelombang pada domain tak hingga yaitu solusi *D'Alembert* yang sudah dinyatakan pada persamaan (13) di landasan teori dimana

$$\phi(x) = \cos(2\pi x)$$

$$\psi(x) = 0$$

sehingga dapat dicari solusi umum persamaan gelombang

$$u(x, t) = \frac{1}{2}[\cos(2\pi((x + ct))) + \cos(2\pi((x - ct)))] + \frac{1}{2c} \int_{x-ct}^{x+ct} 0 \, ds$$

yang kemudian didapat solusi analitik untuk persamaan gelombang kita sebagai berikut:

$$u(x, t) = \frac{1}{2}[\cos(2\pi((x + ct))) + \cos(2\pi((x - ct)))]$$

3.2.2 Metode Numerik

Metode numerik persamaan diferensial parsial

$$u_{tt} = c^2 u_{xx}, \text{ dengan } u(x, 0) = \cos(2\pi x) \text{ dan } u_t(x, 0) = 0$$

dengan $c = 1$ dan suatu kondisi batas periodik

$$u(x, t) = u(x + 1, t).$$

Kita akan menggunakan metode beda pusat untuk melakukan perhitungan. Substitusikan rumus metode beda pusat orde dua u_{tt} dan u_{xx} ke persamaan sehingga didapatkan

$$\frac{U_j^{n+1} - 2u_j^n + u_j^{n-1}}{k^2} = c^2 \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2}$$

dan diketahui sebuah notasi $r = c \frac{k}{h}$ sehingga persamaan sebelumnya dapat ditulis sebagai berikut

$$U_j^{n+1} = 2u_j^n - u_j^{n-1} + r^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$$

kemudian, dengan $n = 0$, akan dicari u_j^{n-1} dengan menggunakan kondisi Neumann,

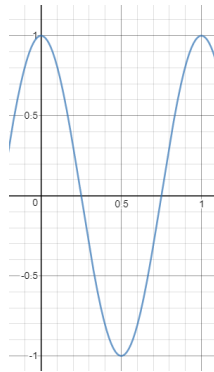
$$\begin{aligned} u_t(x, 0) &= \frac{u_j^1 - u_j^{-1}}{2k} \\ &= 0. \end{aligned}$$

Sehingga didapatkan $u_j^{-1} = u_j^1$. Substitusi ini ke dalam diskritisasi yang kemudian diberikan $n = 0$

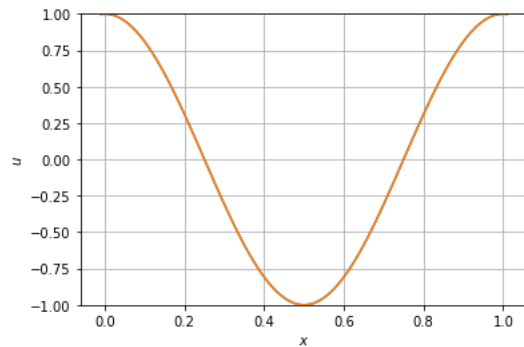
$$u_j^1 = u_j^0 + \frac{1}{2}r^2(u_{j+1}^0 - 2u_j^0 + u_{j-1}^0)$$

3.2.3 Perbandingan Metode Analitik dan Metode Numerik Persamaan Diferensial Parsial

Berikut ini merupakan grafik perbandingan metode analitik dan metode numerik pada persamaan diferensial parsial dimana grafik dengan metode analitik dibuat dengan bantuan <https://www.transum.org/> menggunakan hasil solusi analitik pdp sedangkan grafik dengan metode numerik merupakan hasil kodingan *python* dengan metode beda hingga:

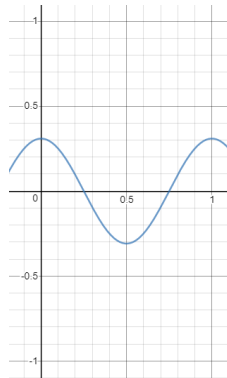


Gambar 3: Grafik Metode Analitik

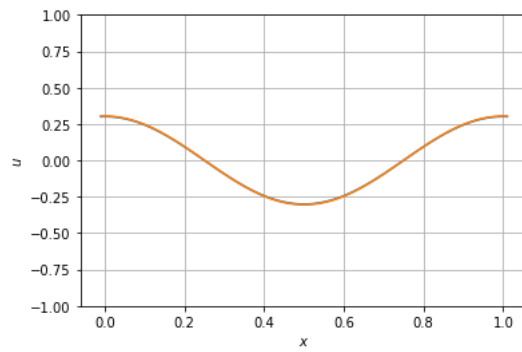


Gambar 4: Grafik Metode Numerik

Perbandingan Grafik pada saat $t = 0$

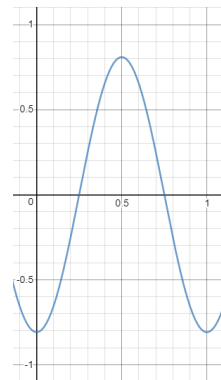


Gambar 5: Grafik Metode Analitik

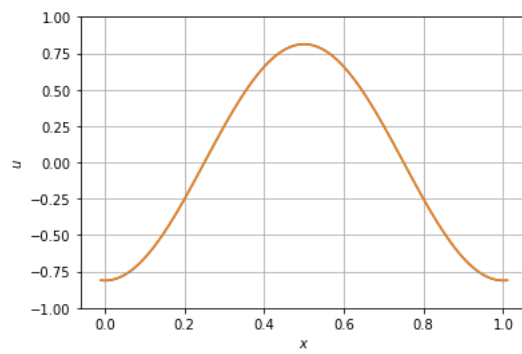


Gambar 6: Grafik Metode Numerik

Perbandingan Grafik pada saat $t = 0.2$

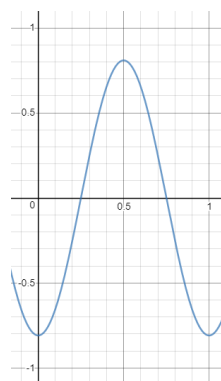


Gambar 7: Grafik Metode Analitik

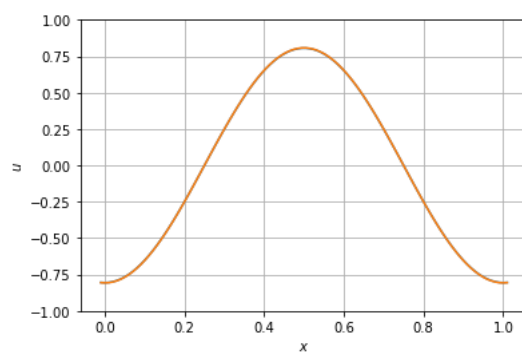


Gambar 8: Grafik Metode Numerik

Perbandingan Grafik pada saat $t = 0.4$

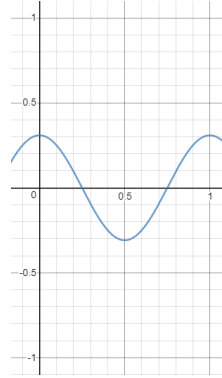


Gambar 9: Grafik Metode Analitik

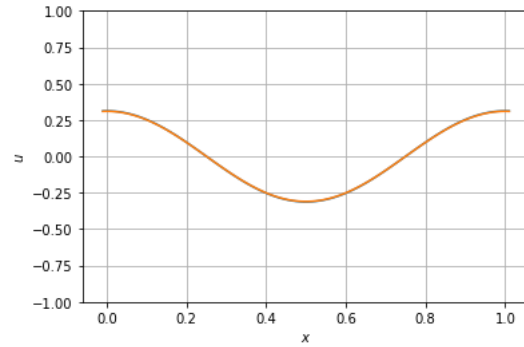


Gambar 10: Grafik Metode Numerik

Perbandingan Grafik pada saat $t = 0.6$

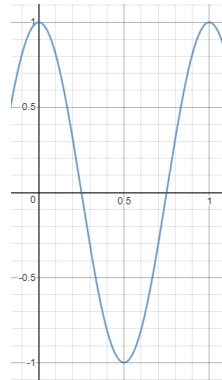


Gambar 11: Grafik Metode Analitik

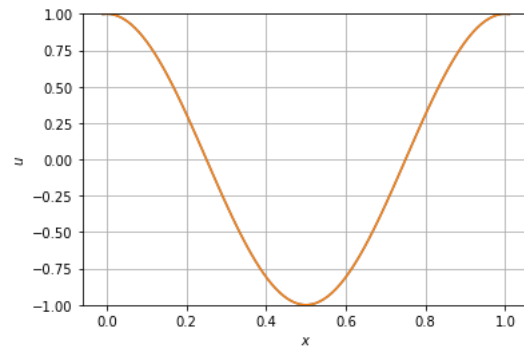


Gambar 12: Grafik Metode Numerik

Perbandingan Grafik pada saat $t = 0.8$



Gambar 13: Grafik Metode Analitik



Gambar 14: Grafik Metode Numerik

Perbandingan Grafik pada saat $t = 1$

Jadi, dapat dilihat dari perbandingan grafik metode analitik dan metode numerik sebelumnya, kedua grafik menunjukkan hasil yang sama pada berbagai macam periode t . Hal yang membedakan kedua metode yaitu pada hasil solusi nya dimana metode analitik mempunyai solusi eksak $\frac{1}{2}(\cos(2\pi(x - t)) + \cos(2\pi(x + t)))$ sedangkan solusi metode numerik adalah suatu hampiran $u_j^1 = u_j^0 + \frac{1}{2}r^2(u_{j+1}^0 - 2u_j^0 + u_{j-1}^0)$.

4 Penutup

4.1 Kesimpulan

Dari laporan yang telah saya buat, saya dapat menyimpulkan bahwa dalam menyelesaikan kasus persamaan diferensial biasa maupun persamaan diferensial parsial menggunakan metode analitik dan numerik, memiliki hasil grafik yang sama. Dalam penyelesaian metode analitik memang memerlukan waktu yang lebih lama untuk memperoleh hasilnya namun dapat memberikan solusi nilai eksak, sedangkan metode numerik hanya memberikan suatu hasil aproksimasi dengan algoritma yang sudah ada menggunakan bantuan program.

4.2 Saran

Berdasarkan kesimpulan tersebut, saya dapat memberikan saran untuk menyelesaikan suatu persamaan diferensial dengan menggunakan metode numerik, karena hasil grafiknya dapat dikatakan cukup sama dibandingkan metode analitik walaupun hanya berupa sebuah hampiran dan dapat mempersingkat waktu pengerjaan.

5 Daftar Pustaka

Pustaka

- [1] Boyce, William E. dan Richard C. DiPrima. (2012). *Elementary Differential Equations* 10th edition, John Wiley.
- [2] Johansson, Robert. (2019). *Numerical Python : Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib-Apress*.
- [3] Kong, Qingkai, Timmy Siau, dan Alexandre M. Bayen. (2020). *Python Programming and Numerical Methods : A Guide for Engineers and Scientists*.
- [4] Kreyszig, Erwin, Herbert Kreyszig, dan Edward J. Norminton. (2011). *Advanced Engineering Mathematics* 10th edition, John Wiley.
- [5] Strauss, W.A. (2008). *Partial Differential Equation: An Introduction* 2nd Edition, John Wiley.
- [6] Chapra, Steven dan Raymond P. Canale. (2021). *Numerical Methods for Engineers* 8th edition, McGraw-Hill.
- [7] Rosidi, Mohammad. (2019). *Metode Numerik Menggunakan R Untuk Teknik Lingkungan*.

6 Lampiran

Kode Program untuk Metode Beda Hingga PDB

```
import numpy as np
import matplotlib.pyplot as plt
plt.style.use("seaborn-poster")

n = 10
h = (5-0)/n

A = np.zeros((n+1, n+1))
A[0,0] = 1
A[n,n] = 1
for i in range (1,n):
    A[i,i-1] = 1
    A[i,i] = -2
    A[i,i+1] = 1
print("Matriks A=",A)

B = np.zeros(n+1)
B[1:-1] = -9.8*h**2
B[-1] = 50
print("Matriks B=",B)

#Menghitung persamaan linear
y = np.linalg.solve(A, B)
t = np.linspace(0,5,11)

#Membuat plot gambar
plt.figure(figsize=(10,8))
plt.plot(t,y)
plt.plot([0,0.5,1,1.5],[0,16.025,29.6,40.725],"ro")
plt.plot([2,2.5,3,3.5],[49.4,55.625,59.4,60.725],"ro")
plt.plot([4,4.5,5],[59.6,56.025,50],"ro")
plt.xlabel("time_(s)")
plt.ylabel("altitude_(m)")
plt.show()
```

```

#Mencari Kecepatan Awal
y_n1 = -9.8*h**2 + 2*y[0] - y[1]
Solusi = (y[1] - y_n1) / (2*h)

print ("Kecepatan Awal dari proses peluncuran adalah", Solusi)

```

Kode Program untuk Metode Beda Hingga PDP

```

import numpy as np
import matplotlib.pyplot as plt

class wave_eq:
    def __init__(self, init_m, init_n):
        self.M = init_m
        self.N = init_n
        self.x_grid = np.linspace(0, 1, self.M)
        dx = self.x_grid[1] - self.x_grid[0]
        self.x_grid = np.hstack((-dx, self.x_grid,
        self.x_grid[-1] + dx))
        self.t_grid = np.linspace(0, 1, self.N + 2)
        self.solution = []

    def forward_euler_solver(self, g):
        M = self.M
        N = self.N

        h = self.x_grid[1] - self.x_grid[0]
        k = self.t_grid[1] - self.t_grid[0]

        r = k/h
        u = np.zeros(shape = (M+2, N+2))

        u[:, 0] = g(self.x_grid)

        print(N, M, len(self.t_grid))
        for i in range(N+1):
            print('time_step_{}/{}'.format(i+1, N+1))

```

```

        if (i == 0):
            u[1:-1,i+1] = u[1:-1,i] + (r**2) *
                (u[2:,i] - 2*u[1:-1,i] + u[:-2,i])
        else:
            u[1:-1,i+1] = 2*u[1:-1,i] -
                u[1:-1,i-1] + (r**2)*(u[2:,i] -
                    2*u[1:-1,i] + u[:-2,i])

        u[0,i+1] = u[-3,i+1]
        u[-1,i+1] = u[2,i+1]

    self.solution = u

def init_c_g(x):
    return np.cos(2*np.pi*x)

def analytical_sol(t,x):
    return 0.5 * ( np.cos(2*np.pi*(x-t)) +
        np.cos(2*np.pi*(x+t)) )

obj = wave_eq(init_m=100, init_n=1000)
obj.forward_euler_solver(g=init_c_g)

nt = obj.solution.shape[1]
ylim = [np.min(obj.solution), np.max(obj.solution)]

for i in range(1,nt,200):
    plt.figure()
    plt.plot(obj.x_grid, obj.solution[:,i])
    plt.plot(obj.x_grid, analytical_sol(obj.t_grid[i],
        obj.x_grid))
    plt.ylim(ylim)
    plt.grid()
    plt.xlabel(r'$x$')
    plt.ylabel(r'$u$')

```