

Trabalho 3 - Relatividade Numérica

Nuno Teixeira - 75494

Instituto Superior Técnico -Tópicos Avançados de Física Computacional

5 de Novembro de 2018

1 Notas sobre o Programa

Este programa está desenhado para gerar os dados relativos à intensidade de uma onda ψ , no tempo e no espaço, de acordo com as secções (2) e (3). A partir destes dados gerar dois tipos de outputs usando a biblioteca de gráficos da GNU, "Gnuplot". Em relação ao output, ou é indicado para gerar um gráfico 3D com a posição tempo e intensidade da onda, ou um filme de curta metragem, "Output_gif.gif", em que é mostrado explicitamente a variação da onda consoante o tempo da simulação.

O programa que gerou os dados relativos à intensidade da onda foi feito em C++ e está dividido em duas partes: a implementação, que está contida no ficheiro "S3_onda.h", e a interface que está incluída em "S3_onda.cpp". Na implementação estão contidas as funções de criação de grelha uniforme, para criar os eixos x e t ; derivação por diferenças finitas com uma precisão de 2ª ordem contemplando as diferentes condições fronteira de acordo com a secção (3.2); a função Runge-Kutta de 4ª ordem como explicito na secção (3.1) assim como outras funções auxiliares de soma de arrays, escrita para ficheiros, etc. Em relação à interface, esta corresponde genericamente ao algoritmo para evoluir a equação de onda no tempo, e é onde o programa começa e portanto onde colocamos os parametros de input:

- N_x - O número de pontos espaciais para interpolar a onda.
- x_{inf} - Limite inferior da caixa.
- x_{sup} - Limite superior da caixa.
- N_t - O número de iterações temporais.
- t_f - O tempo final da simulação.
- σ - Desvio médio da onda
- x_0 - Posição inicial média da onda.
- Condições fronteira - Que correspondem às opções da função derivação por diferenças finitas. Existem três alternativas, "none" onde não são aplicadas condições fronteira, "periodicas" para aplicar condições fronteira periódicas e "reflectoras" para aplicar condições fronteira reflectoras.

Para gerar os dados, executamos os seguintes comandos no terminal:

```
$ g++ S3_onda.cpp -o S3_onda
$ ./ S3_onda
```

A partir daqui se queremos gerar um "splot" fazemos,

```
$ gnuplot Output_splot.plt
```

Para gerar um ficheiro gif com a evolução da onda,

```
$ gnuplot Output_gif.gp
$ xdg-open Output_gif.gif
```

Elaborei um script em bash que executa todos estes comandos no terminal, sendo apenas necessário descomentar/comentar as linhas referentes ao output desejado/indesejado. A sua utilização é dada apenas pela linha,

```
$ . script.sh
```

2 Introdução

O intuito deste trabalho é resolver a equação de Maxwell da propagação de uma onda ψ num tempo finito t_f estando limitada por uma caixa de $[x_{inf}, x_{sup}]$.

A propagação da onda no tempo e no espaço, segue a seguinte equação:

$$\partial_{tt}\psi = \partial_{xx}\psi, \quad (1)$$

tendo,

$$\psi_0(x) = e^{-\frac{(x-x_0)^2}{2\sigma^2}}, \quad (2)$$

como condição inicial. Podemos então recorrer à resolução do sistema segundo duas variáveis auxiliares, ϕ e π :

$$\phi = \partial_x \psi, \quad (3)$$

$$\pi = \partial_t \psi, \quad (4)$$

$$\partial_t \psi_0 = \lambda \partial_x \psi_0. \quad (5)$$

com λ entre 0, 1 e -1. O sistema a integrar no tempo fica:

$$\begin{bmatrix} \partial_t \phi \\ \partial_t \pi \\ \partial_t \psi \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \partial_x \phi \\ \partial_x \pi \\ \pi \end{bmatrix}. \quad (6)$$

Na resolução da Eq. (1), torna-se útil perceber a dinâmica segundo os modos left e right, que chamaremos, l e r . Estes modos empurram a onda, na próxima iteração temporal, ou para esquerda se estivermos a evoluir a equação left ou para a direita, se estivermos a evoluir a equação right. Tornando o sistema diagonal, pode-se provar que as Eqs. (6) assumem os seguintes valores próprios,

$$r = \phi - \pi, \quad (7)$$

$$l = \phi + \pi, \quad (8)$$

com,

$$\begin{bmatrix} \partial_t l \\ \partial_t r \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \partial_x l \\ \partial_x r \end{bmatrix}. \quad (9)$$

3 Resolução Numérica do Sistema

Para resolver as equações do sistema de Eqs. (6), foi usado o Método das Linhas (Method of Lines). Este método consiste em discretizar o espaço e aplicar diferenças finitas no que toca às derivadas espaciais, e de seguida aplicar uma integração Runge-Kutta de quarta ordem para obter a evolução do sistema espacial e temporal.

3.1 Runge-Kutta de 4ª ordem

Para uma expressão genérica a uma dimensão

$$\dot{y} = f(t, y), \quad (10)$$

usando como espaçamento temporal entre iterações dt e número total de iterações N_t ,

$$dt = t_f / (N_t - 1), \quad (11)$$

a próxima iteração dada pelo método de Runge-Kutta de quarta ordem [2],

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \quad (12)$$

Com,

$$k_1 = dt f(t, y_n), \quad (13)$$

$$k_2 = dt f\left(t + \frac{dt}{2}, y_n + \frac{k_1}{2}\right), \quad (14)$$

$$k_3 = dt f\left(t + \frac{dt}{2}, y_n + \frac{k_2}{2}\right), \quad (15)$$

$$k_4 = dt f(t + dt, y_n + k_3). \quad (16)$$

3.1.1 Aplicação de Runge-Kutta de 4ª Ordem ao Problema

Foi aplicado um algoritmo de integração Runge-Kutta 2D tal como [1], visto que o nosso sistema depende de duas variáveis, ϕ e π . Para o caso específico de ϕ , temos, de acordo com 12,

$$\partial_t \phi = \partial_x \pi, \quad (17)$$

$$p_1 = dt f(t, \phi_n, \pi_n) = dt \partial_x \pi_n, \quad (18)$$

$$p_2 = dt f\left(t + \frac{dt}{2}, \phi_n + \frac{p_1}{2}, \pi_n + \frac{p_1}{2}\right) = dt \partial_x \left(\pi_n + \frac{p_1}{2}\right), \quad (19)$$

$$p_3 = dt f\left(t + \frac{dt}{2}, \phi_n + \frac{p_2}{2}, \pi_n + \frac{p_2}{2}\right) = dt \partial_x \left(\pi_n + \frac{p_2}{2}\right), \quad (20)$$

$$p_4 = dt f(t + dt, \phi_n + p_3, \pi_n + p_3) = dt \partial_x (\pi_n + p_3), \quad (21)$$

$$\phi_{n+1} = \phi_n + \frac{1}{6}(p_1 + 2p_2 + 2p_3 + p_4). \quad (22)$$

Do mesmo modo para π ,

$$\partial_t \pi = \partial_x \phi, \quad (23)$$

$$q_1 = dt f(t, \phi_n, \pi_n) = dt \partial_x \phi_n, \quad (24)$$

$$q_2 = dt f\left(t + \frac{dt}{2}, \phi_n + \frac{q_1}{2}, \pi_n + \frac{q_1}{2}\right) = dt \partial_x \left(\phi_n + \frac{q_1}{2}\right), \quad (25)$$

$$q_3 = dt f \left(t + \frac{dt}{2}, \phi_n + \frac{q_2}{2}, \pi_n + \frac{k_2}{2} \right) = dt \partial_x \left(\phi_n + \frac{q_2}{2} \right), \quad (26)$$

$$q_4 = dt f(t + dt, \phi_n + q_3, \pi_n + q_3) = dt \partial_x (\phi_n + k_3), \quad (27)$$

$$\pi_{n+1} = \pi_n + \frac{1}{6}(q_1 + 2q_2 + 2q_3 + q_4). \quad (28)$$

Em relação a ψ , a integração é passiva, na medida em que é uma equação espectadora no que toca às restantes, mas também é aplicar o método de integração de Runge-Kutta para determinar a próxima iteração.

$$\partial_t \psi = \pi, \quad (29)$$

$$s_1 = dt f(t, \phi_n, \pi_n) = dt \pi_n, \quad (30)$$

$$s_2 = dt f \left(t + \frac{dt}{2}, \pi_n + \frac{s_1}{2}, \pi_n + \frac{s_1}{2} \right) = dt \left(\phi_n + \frac{s_1}{2} \right), \quad (31)$$

$$s_3 = dt f \left(t + \frac{dt}{2}, \pi_n + \frac{s_2}{2}, \pi_n + \frac{s_2}{2} \right) = dt \left(\phi_n + \frac{s_2}{2} \right), \quad (32)$$

$$s_4 = dt f(t + dt, \pi_n + s_3, \pi_n + s_3) = dt (\phi_n + s_3), \quad (33)$$

$$\psi_{n+1} = \psi_n + \frac{1}{6}(q_1 + 2q_2 + 2q_3 + q_4). \quad (34)$$

3.2 Derivação Numérica sobre a Ausência de Condições Fronteira

Relativamente às derivadas espaciais, concretamente, foi usado um método de diferenças finitas com uma precisão de segunda ordem. Este método, sobre a ausência de condições fronteira, consiste em calcular a derivada do primeiro da ponto da grelha com a informação dos dois próximos pontos, a derivada do último ponto com a informação dos dois pontos anteriores e as restantes derivadas são por diferenças centrais. Podemos então escrever, para um array f com N_x pontos ordenados segundo uma grelha a representar o domínio da caixa, com o primeiro índice a começar em 0, na posição x_{inf} e o último a acabar em $N_x - 1$ na posição x_{sup} :

$$dx = \frac{x_{sup} - x_{inf}}{N_x - 1}, \quad (35)$$

$$\frac{df}{dx}[0] = \frac{-3f[0] + 4f[1] - 1f[2]}{2dx}, \quad (36)$$

$$\frac{df}{dx}[N_x - 1] = \frac{3f[N_x - 1] - 4f[N_x - 2] + 1f[N_x - 3]}{2dx}, \quad (37)$$

para i entre 1 e $N_x - 2$

$$\frac{df}{dx}[i] = \frac{f[i + 1] - f[i - 1]}{2dx}. \quad (38)$$

3.3 Condições Fronteira Periódicas

Para condições fronteira periódicas, queremos ligar o último ponto com o primeiro, portanto neste caso as expressões são muito parecidas com o caso anterior, à excepção que no primeiro ponto são calculadas diferenças finitas centrais com o último ponto de f , ao passo que no último ponto são calculadas diferenças finitas com o ponto 0. Deste modo:

$$\frac{df}{dx}[0] = \frac{f[1] - f[N_x - 1]}{2 dx}, \quad (39)$$

$$\frac{df}{dx}[N_x - 1] = \frac{f[0] - f[N_x - 2]}{2 dx}, \quad (40)$$

$$\frac{df}{dx}[i] = \frac{f[i + 1] - f[i - 1]}{2 dx}. \quad (41)$$

3.3.1 Condições Fronteira Reflectoras e de Saída

Devido à natureza das equações left e right, após a onda colidir com a parede, com condições fronteira reflectoras, esta reflecte. Assumindo que a parede situada no índice 0 tem um índice de reflexão R_0 e que a parede situada no ponto $N_x - 1$ tem um índice de reflexão de R_1 , isto significa que,

$$r[0] = R_0 l[0], \quad (42)$$

e que,

$$l[N_x - 1] = R_1 r[N_x - 1]. \quad (43)$$

Após aplicarmos a dinâmica do sistema de Eqs. 9, ficamos com as seguintes expressões para aplicar no Runge-Kutta,

$$\partial_t \phi|_{x=x_{inf}} = \frac{1 + R_0}{2} \frac{\partial(\phi + \pi)}{\partial x} \Big|_{x=x_{inf}}, \quad (44)$$

$$\partial_t \pi|_{x=x_{inf}} = \frac{1 - R_0}{2} \frac{\partial(\phi + \pi)}{\partial x} \Big|_{x=x_{inf}}, \quad (45)$$

$$\partial_t \phi|_{x=x_{sup}} = -\frac{1 + R_1}{2} \frac{\partial(\phi - \pi)}{\partial x} \Big|_{x=x_{sup}}, \quad (46)$$

$$\partial_t \pi|_{x=x_{sup}} = \frac{1 - R_1}{2} \frac{\partial(\phi - \pi)}{\partial x} \Big|_{x=x_{sup}}. \quad (47)$$

4 Resultados

Devido à condição inicial, Eq. (5), sabemos que pelas Eqs. (7) e (8) que,

$$r_0 = \frac{1 - \lambda}{2} \partial_x \psi_0(x), \quad (48)$$

$$l_0 = \frac{1 + \lambda}{2} \partial_x \psi_0(x). \quad (49)$$

Deste modo, caso $\lambda = 1$, $r_0 = 0$ e só temos propagação da parte left, caso $\lambda = -1$, $l_0 = 0$ e temos apenas propagação da parte right, caso $\lambda = 0$ temos uma sobreposição dos dois modos em que cada um possui metade da intensidade total.

Executando o programa com os seguintes parametros

- $N_x = 100$
- $N_t = 4000$
- $x_{inf} = -100$
- $x_{sup} = 100$
- $\sigma = 0.122$
- $t_f = 400$
- $R_0 = 1$
- $R_1 = 1$
- $dt/dx = 0.0495124$

Obtive os resultados e de seguida foi gerado o output num gráfico 3D em que num dos eixos está marcada a posição, noutro o tempo e no eixo z, está representada a intensidade da onda.

4.1 Condições Fronteira Periódicas

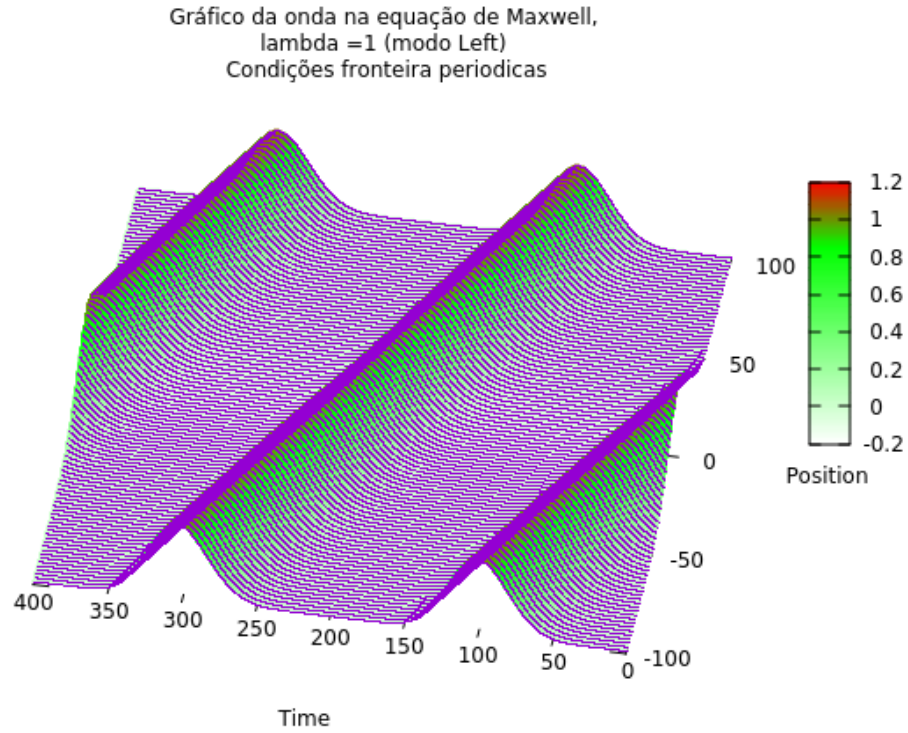


Figura 1: Este gráfico exemplifica a propagação do modo left, $\lambda = 1$, com condições fronteira periódicas. Como a figura mostra, quando este modo atinge a parede, na iteração seguinte, a onda é desviada do comprimento da caixa, $2 \times (x_{sup} - x_{inf})$. No geral, a intensidade da onda mantém-se constante porque apenas estamos a propagar um dos modos.

Gráfico da onda na equação de Maxwell,
 $\lambda = -1$ (modo right)
 Condições fronteira periódicas

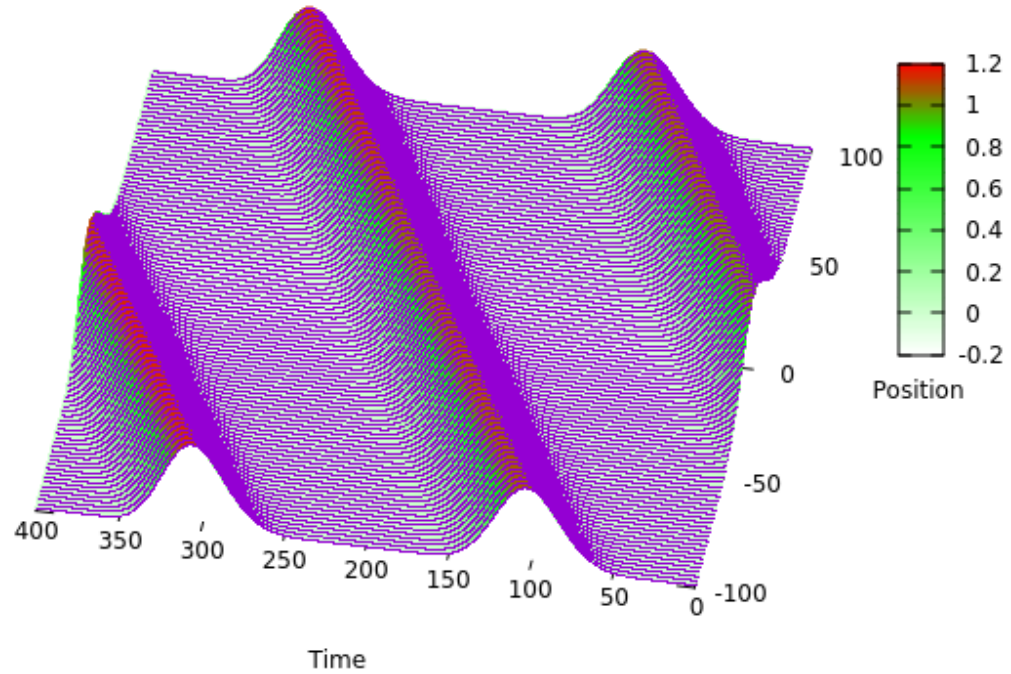


Figura 2: Gráfico relativo a $\lambda = -1$ com condições fronteira periódicas, propagação do modo right. Esta figura é semelhante ao da Fig. (4.1), na medida em que existe uma simetria na propagação do modo.

Gráfico da onda na equação de Maxwell,
 $\lambda = 0$ (Sobreposição dos dois modos)
 Condições fronteira periódicas

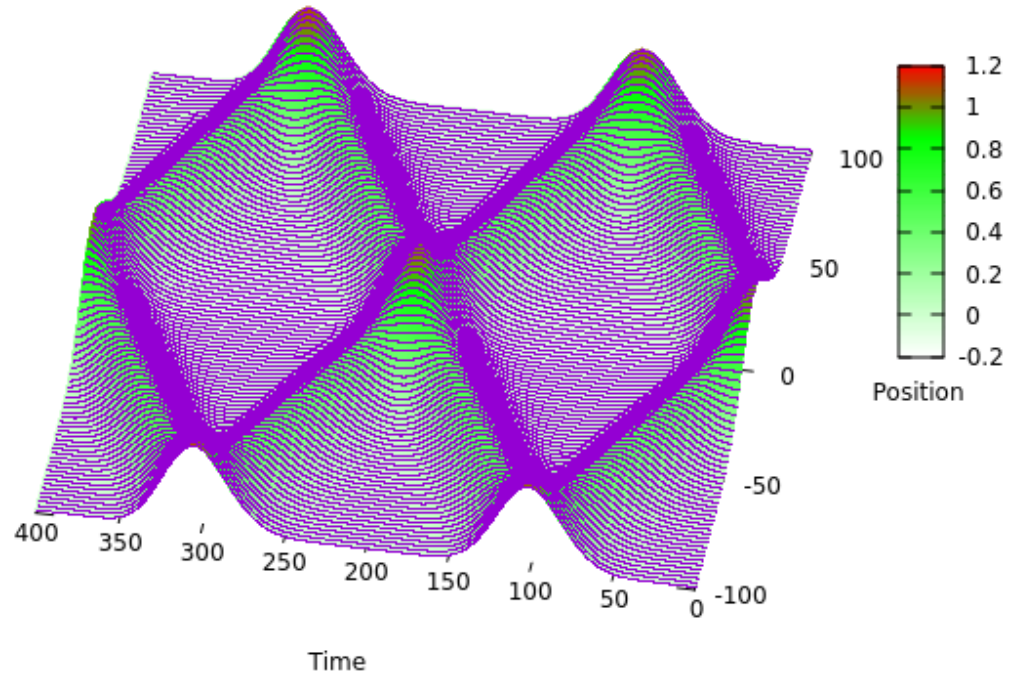


Figura 3: Gráfico relativo a $\lambda = 0$ com condições fronteira periódicas. Neste caso existe uma propagação dos dois modos em simultâneo, as duas ondas somam-se quando se interceptam, voltando a reaver a intensidade inicial. Quando um dos modos passa num dos limites da caixa, este aparece no outro continuando a propagar-se no tempo para a esquerda caso escolhamos o modo left ou para a direita caso escolhamos o modo right.

4.2 Condições Fronteira Reflectoras

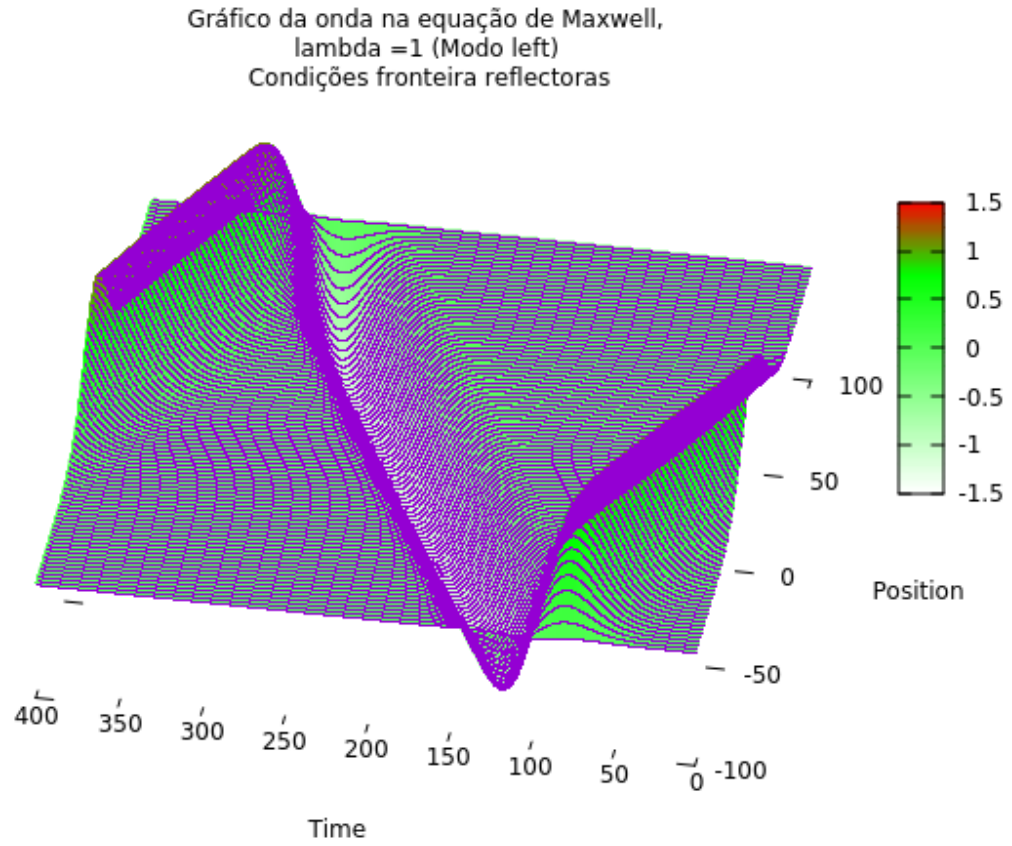


Figura 4: Este gráfico exemplifica a propagação inicial do modo left, $\lambda = 1$, com condições fronteira reflectoras. Como a figura mostra, quando este modo atinge a parede, na iteração seguinte, a onda é reflectida, mudando a amplitude de $+1$ para -1 e o modo de propagação de left para right. No geral, a intensidade da onda alterna o sinal mas em módulo mantém-se constante porque apenas estamos a propagar um dos modos.

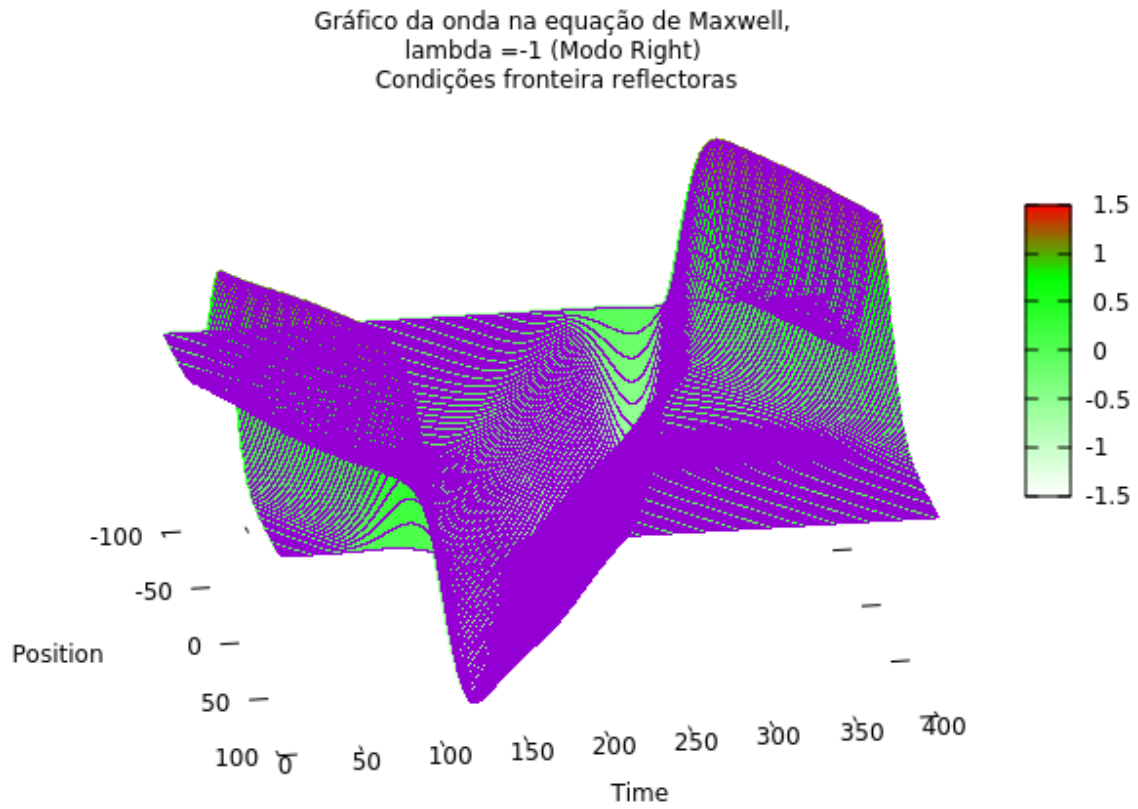


Figura 5: Gráfico relativo a $\lambda = -1$ com condições fronteira reflectoras, propagação inicial do modo right. Esta figura é semelhante ao da Fig. (4.1), na medida em que existe uma simetria na propagação do modo.

Gráfico da onda na equação de Maxwell,
 $\lambda = 0$ (Sobreposição de modos)
 Condições fronteira reflectoras

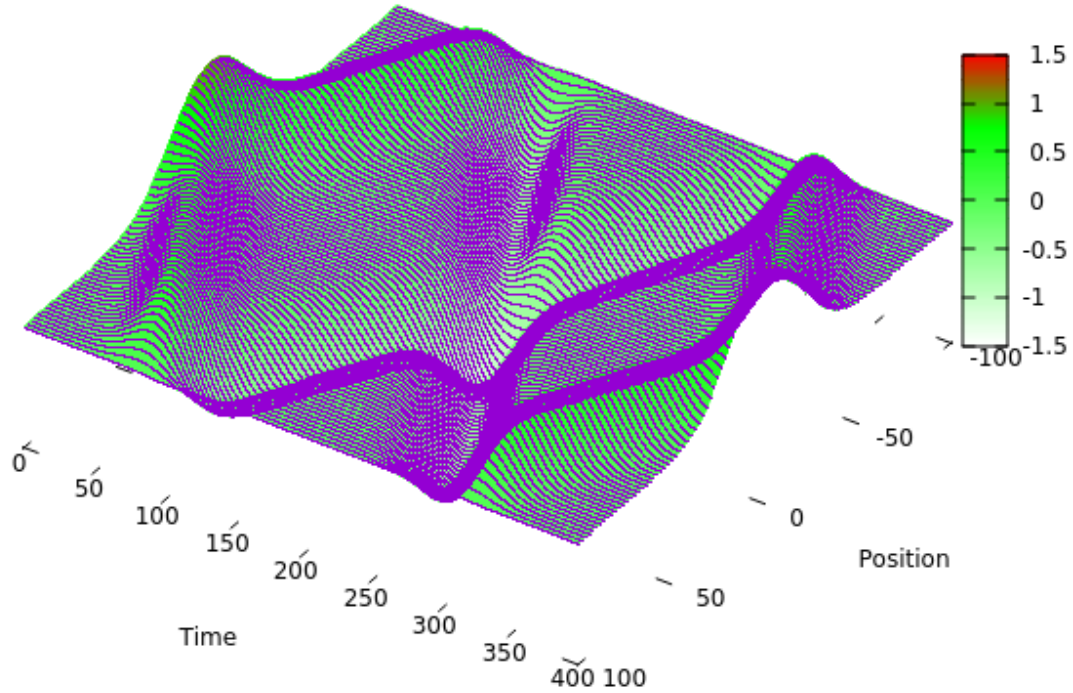


Figura 6: Gráfico relativo a $\lambda = 0$ com condições fronteira reflectoras. Neste caso existe uma propagação dos dois modos em simultâneo, as duas ondas somam-se quando se interceptam, voltando a reaver a intensidade inicial ou o valor da intensidade inicial multiplicado por -1. Quando um dos modos passa num dos limites da caixa, este reflecte alterando o sinal e o modo de propagação.

5 Análise da Estabilidade

Este código permite-nos simular para certos valores de dt e dx a evolução no tempo de uma onda. Contudo, por usarmos um método de diferenças finitas de segunda ordem e no método de Runge-Kutta estarmos a aplicar a mesma formula recursivamente (o que quer dizer que o erro acumula), por vezes o nosso código torna-se instável. Isto acontece na medida em que a onda começa a interferir com ela própria quando, nalgum momento da simulação, a instabilidade da onda cria outra uma pequena onda criando uma ressonância no sistema e propagando erros por toda a simulação, tornando o código inválido no domínio em estudo.

Em parte, isto acontece porque existem poucos pontos a interpolar a gaussiana e as variações da derivada espacial tornam-se grandes face ao comprimento do intervalo, especialmente nas caudas e o no cume que são as zonas onde costumam a aparecer instabilidades. Uma forma de contornar este problema é adequar a largura da gaussiana, σ de acordo com o comprimento da caixa. Isto faz com que a interpolação da gaussiana seja adequada e haja uma suavização das derivadas espaciais, reduzindo uma possível fonte de erros na simulação.

Outra das razões é pelo espaçamento temporal, dt ser demasiado grande face ao espaçamento temporal dx , aumentando a fonte de erro do Runge-Kutta e faz com que rapidamente os valores da função de onda se tornam arbitrariamente grandes como consequência.

Foi feito um estudo dos limites de divergência deste código sobre dois aspectos da simulação: um deles o coeciente máximo dt/dx tal que a simulação não apresentasse defeitos na visualização e o $t_f/\Delta x$, com Δx o comprimento da caixa $x_{sup} - x_{inf}$, para termos uma medida normalizada da duração da simulação consoante um certo valor de σ . Com comprimento= 2, $N_x = 251$, $\lambda = 0$, $x_0 = 0$, $\sigma = 0.13$ e com as fronteiras na condição de reflexão total, $R_1 = R_0 = 1$ foi registado o tempo limite T_f o qual a simulação correu sem instabilidades consoante os valores de N_t . Os resultados deste estudo encontram-se na Tabela (5).

Δx	T_f	N_x	N_t	$T_f/\Delta x$	dt/dx
2	0.82	251	301	0.41	0.3417
2	1.1	251	601	0.55	0.2292
2	1.42	251	1001	0.71	0.1775
2	2	251	2001	1	0.1250
2	3.1	251	5001	1.55	0.0775
2	4.4	251	10001	2.2	0.0550
2	6.3	251	20001	3.15	0.0394
2	7.8	251	30001	3.9	0.0325
2	8.7	251	40001	4.35	0.0272
2	10	251	50001	5	0.0250

Tabela 1: Registo dos vários valores T_f face aos diferentes valores de N_t . Existe uma relação inversa entre $T_f/\Delta x$ e o dt/dx máximo ao qual a simulação foi bem sucedida.

Como é possível ver na Tabela (5), existe uma relação inversa entre $T_f/\Delta x$ e dt/dx . Podemos determinar um modelo que emula esta suposição,

$$y = A/x, \quad (50)$$

em que x neste caso é dado por T_f/Comp e y por dt/dx . Tendo acesso a $y(1) = 0.125$ através da Tabela (5), determinamos $A = 0.125$. A Fig. (5) ilustra a relação entre as colunas 5 e 6 da Tabela (5).

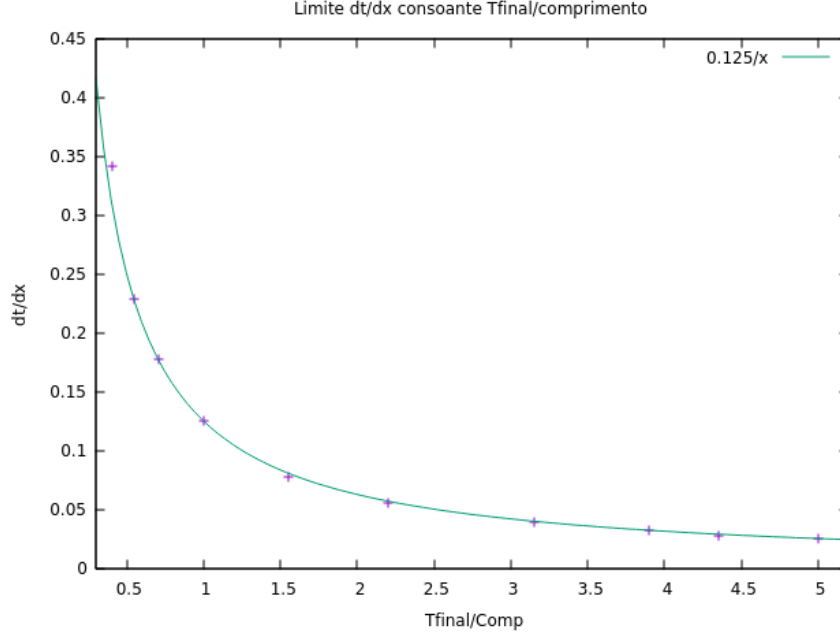


Figura 7: Relação entre $T_f/\Delta x$ e $\max(dt/dx)$. Como a figura ilustra, o acordo com uma relação inversa entre as duas quantidades é quase perfeito.

Dado o nosso modelo de variação de dt/dx consoante o tempo relativo ao comprimento da caixa, que podemos considerar como um período, para $N_t = 100001$, com o auxílio da expressão Eq. (50) determinou-se $T_f = 14.113$ como o tempo necessário termos uma simulação estável nas mesmas condições do caso anterior. A ilustração desta simulação encontra-se na Fig. (8).

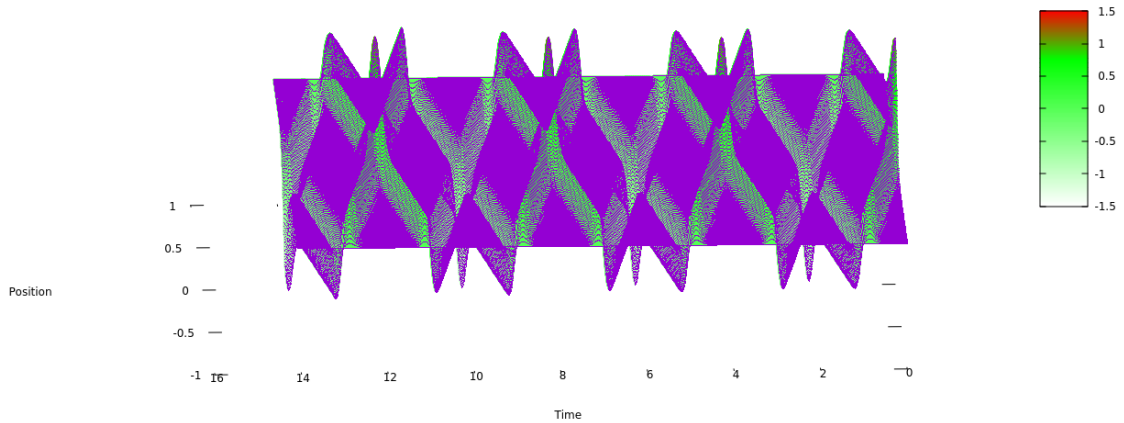


Figura 8: Simulação com os seguintes parametros comprimento= 2, $N_x = 251$, $N_t = 100001$, $\lambda = 0$, $x_0 = 0$, $\sigma = 0.13$ e com as fronteiras na condição de reflexão total, $R_1 = R_0 = 1$. Não ocorreram instabilidades nesta simulação, a simulação correu de acordo com o esperado.

Referências

- [1] <http://wps.prenhall.com/wps/media/objects/884/905485/chapt4/proj4.3A/proj4-3A.pdf>
- [2] https://en.wikipedia.org/wiki/Runge-Kutta_methods