# Report
# Numerical Simulation of Beam Propagation including Error Convergence Analysis

Prepared by: Nicolae Cadin
Supervised by: Ramon Springer

15-05-2018

# Contents

# 1 Introduction

Beam propagation in media is of high interest for scientists and also for industrial purposes. In this report beam propagation will be simulated, problems will be discussed. The purpose of the project is to compare numerical method and analytical solution of beam propagation. Convergence of used numerical method is to be shown. For the simplicity simulation of beam propagation is done in free space, linear medium, at zero incident angle, so to fit the scope of mini-project, later on can be extended for more sophisticated cases.

## 1.1 Analytical solution

To better analytically understand propagation of beam in space we should solve Helmholz Equation.

$$(\nabla^2 + k^2)E = 0$$

This equation generally holds for electromagnetic wave, hence for electric and magnetic field components, for simplicity we chose electric field $E$. Here $k$ is wave number and is equal to $2\pi n/\lambda$, where $\lambda$ is wavelength, $n$ is refractive index of medium (here we assume it to be equal to 1). As assumption we consider that our wave propagates in $z$ direction and electric field can be represented in next form.

$$E(x, y, z) = u(x, y, z)e^{-ik_o z}$$

$u$ is complex function which describes non-plane part of the beam. Substituting our ansatz into Helmholtz equation in cartesian coordinates and doing some simplifications we get.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} - 2ik_o\frac{\partial u}{\partial z} + (k^2 - k_o^2)u = 0$$

Using paraxial approximation we see that third term is smaller than other, therefore can be neglected, as result we get following equation.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - 2ik_o\frac{\partial u}{\partial z} + (k^2 - k_o^2)u = 0 \tag{1}$$

Solving this differential equation we find $u$ hence we find $E$. We assume that $k = k_o$, therefore solution for one dimensional case, where wave propagates in z-direction and oscillates in xz plane is,

$$E(x, z) = E_o\left(\frac{w_o}{w(z)}\right)^{\frac{1}{2}} exp\left(\frac{x^2}{w(z)^2}\right) exp\Big(-i\Big(k_o z + k_o\frac{x^2}{2R(z)} - \phi(z)\Big)\Big) \tag{2}$$

where $w(z)$ is beam radius, $w_o = w(0)$ is beam waist, $E_o$ is electric field amplitude, $R(z)$ is beam curvature, $\phi$ is Gouy phase. Their explicit mathematical formula can be found in the Table 1. This electric field formula will be later used as reference to numerical solution. Intensity can be calculated based on electric field, $I = \frac{\epsilon_o c n_o}{2}|E_z|^2$.

| Beam Radius | Beam Curvature | Gouy phase |
|---|---|---|
| $w(z) = w_o\sqrt{1+\left(\frac{z\lambda}{\pi w_o^2}\right)^2}$ | $R = z\left[1+\left(\frac{\pi w_o^2}{z\lambda}\right)^2\right]$ | $\phi(z) = arctan\left(\frac{z\lambda}{\pi w_o^2}\right)$ |

Table 1: Auxiliary formulas to equation 2.

## 1.2 Numerical solution

Equation 1 for one dimensional case can be rewritten in the following form.

$$2ik_o\frac{\partial u}{\partial z} = \frac{\partial^2 u}{\partial x^2} + (k^2 - k_o^2)u \tag{3}$$

To simulate light beam propagation Beam Propagation Method (BPM) is used. BMP is descritization of formula (3), formula should be descritized for z-component and x-component. For beggining let's do descitization only for z component, our equation will get following form.

$$2ik_o\frac{u^{l+1}-u^l}{\Delta z} = \frac{\partial^2 u^l}{\partial x^2} + (k^2 - k_o^2)u^l$$

Index $l$ denotes the order of grid point in z-direction. This finite difference scheme is known as **forward difference**. However after discretization of x-component numerical instability can be seen. There is another alternative where order of grid point is taken one step in advance, this method is called **backward difference**

$$2ik_o\frac{u^{l+1}-u^l}{\Delta z} = \frac{\partial^2 u^{l+1}}{\partial x^2} + (k^2 - k_o^2)u^{l+1}$$

Neither this scheme shows good stability, however the combination of these two solves instability problem. These method is called **Crank-Nicolson** method.

$$2ik_o\frac{u^{l+1}-u^l}{\Delta z} = (1-\alpha)\frac{\partial^2 u^l}{\partial x^2} + (1-\alpha)(k^2 - k_o^2)u^l + \alpha\frac{\partial^2 u^{l+1}}{\partial x^2} + \alpha(k^2 - k_o^2)u^{l+1}$$

$\alpha$ here shows interest of which difference we want higher contribution, usually is set to 1/2. Then Crank-Nicolson scheme has next form.

$$2ik_o\frac{u^{l+1}-u^l}{\Delta z} = \frac{1}{2}\left(\frac{\partial^2 u^l}{\partial x^2} + (k^2 - k_o^2)u^l\right) + \frac{1}{2}\left(\frac{\partial^2 u^{l+1}}{\partial x^2} + (k^2 - k_o^2)u^{l+1}\right)$$

Now discretization of x-component can be performed.

$$2ik_o\frac{u_j^{l+1}-u_j^l}{\Delta z} = \frac{1}{2}\left(\frac{u_{j-1}^l - 2u_j^l + u_{j+1}^l}{\Delta x^2} + (k^2 - k_o^2)u_j^l\right) + \frac{1}{2}\left(\frac{u_{j-1}^{l+1} - 2u_j^{l+1} + u_{j+1}^{l+1}}{\Delta x^2} + (k^2 - k_o^2)u_j^{l+1}\right)$$

$j$ index denotes the order of grid point in x-direction. Such discretization can be rewritten and later solved using triagonal matrix algorithm or so called **Thomas algorithm**.

$$2ik_o\frac{u^{l+1}-u^l}{\Delta z} = \frac{1}{2}\left(L_h u^{l+1} + L_h u^l\right)$$

$L_h$ is spacial discretization operator for one dimensional case and is numerically defined in Thomas algorithm as

$$L = \frac{1}{\Delta x^2}\begin{bmatrix} -2+(k^2-k_o^2)\Delta x^2 & 1 & 0 & \dots & 0 \\ 1 & -2+(k^2-k_o^2)\Delta x^2 & 1 & \dots & 0 \\ 0 & 1 & -2+(k^2-k_o^2)\Delta x^2 & & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & & -2+(k^2-k_o^2)\Delta x^2 \end{bmatrix}$$

From here doing simple mathematical rearrangement we can write previous equation in the following form.

$$u^{l+1} = \left(I - \frac{\Delta z}{4ik_o}L_h\right)^{-1}\left(I + \frac{\Delta z}{4ik_o}L_h\right)u^l$$

Here $I$ is identity matrix, which has the same size as triagonal matrix $L$. We can notice that this is iterative process, in our particular case we instantiate $u^{l=0}$ at $z = 0$ to Gaussian function, in another words $u(x, z = 0) = E_o exp\big(-\frac{x^2}{w_o^2}\big)$.

### 1.2.1 Absorbing Boundary Condition

Implementing numerical method, wave approaching boundary will be reflected, this happens due to the nature of numerical algorithm. Physically this can be interpreted as insertion of mirror at the boundary. Reflection at the boundaries can be prevented introducing absorption, such method is called absorbing boundary condition(ABC). One such way is to add a complex part to refractive index, refractive index becomes $\tilde{n} = n + i\kappa$. $\kappa$ is absorption index, it should increase gradually slowly, fast increase can again cause reflection. Absorption index is designed to have parabolic increase and can be expressed in following way.

$$\kappa(x) = \kappa_{max}\left(\frac{x - l/2 + \delta}{\delta}\right)^2, x\left[\frac{l}{2}, \frac{l}{2} - \delta\right];$$

$$\kappa(x) = \kappa_{max}\left(\frac{l/2 + \delta - x}{\delta}\right)^2, x\left[-\frac{l}{2} + \delta, -\frac{l}{2}\right];$$

Here, $\kappa_{max}$ is maximum value of absorption index to which absorption index as function of x can increase, $l$ is the width of our computational domain, in our case will be corresponding to range of x, $\delta$ is a part of total width where absorption will be introduced, geometry of introduced absorption can be seen in Figure 1. It is important to notice that ABC do not change along z-direction. The function $\kappa(x)$ is show in Figure 1.

### 1.2.2 Error convergence

In order to assess the accuracy of the numerical method, error is calculated. Error is a difference between numerical and analytical electric field and can be calculated using different norms. There is fixed relation between number of grid point and total error, if one increases the number of grid points, error is expected to decrease. If number of grid points is increased $2^n$ times, error

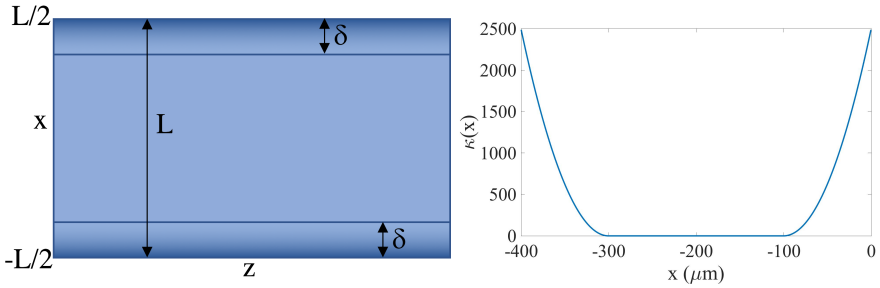



Figure 1: Absorption boundary, figure on the right show the geometry of absorbed boundary, appropriate $\delta$ can be chosen during simulation time, figure on the left shows function absorption index $\kappa\ (x)$ and its change along x-direction.

should decrease $2^{-n}$ times, $n$ is integer number which also tells about order of error convergence.

$$||Error||_2 = \frac{1}{Number\ of\ grid\ points}\sqrt{\sum_{i=1}^{n}(E_{z\ numerical} - E_{z\ analytic})^2}$$

$$||Error||_\infty = (E_{z\ numerical} - E_{z\ analytic})_{maximum}$$

$$||Error||_{at\ particular\ point} = Error\ at\ any\ particular\ physical\ point\ is\ recorded$$

# 2 Results

Numerical algorithm and analytical solution were both implemented and compared, results can be seen in Figure 2. In first row depicts Crank-Nicolson numerical method, second row shows analytical solution, first and second columns show real component of electric field and intensity of light consequently. Parameters of laser beam in our case are: z-meshsize = x-meshsize = 1 $\mu m$, wavelength ($\lambda$) = 20 $\mu m$, waist ($w_o$) = 30 $\mu m$, refractive index ($n_o$) = 1.

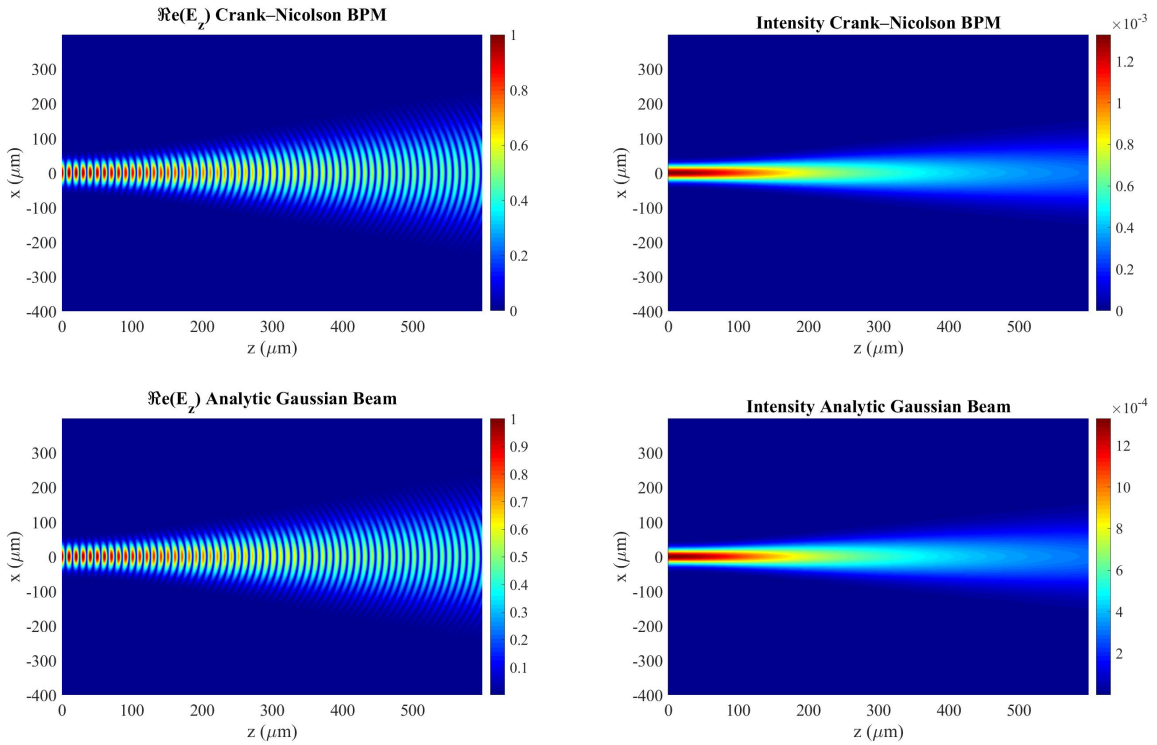



Figure 2: Comparison of two methods.

## 2.1 Absorption Boundary Condition

In this section we show the results of applied ABC. In Figure 3 the necessity of applying ABC can be seen. In the figure on the left no ABC is applied, as result reflection occurs on the boundary on the top and bottom, because of reflection interference effect can be observed. In the figure on the right ABC is applied. Gradual absorption of electric wave can be observed, also wave doesn't reach the boundary.

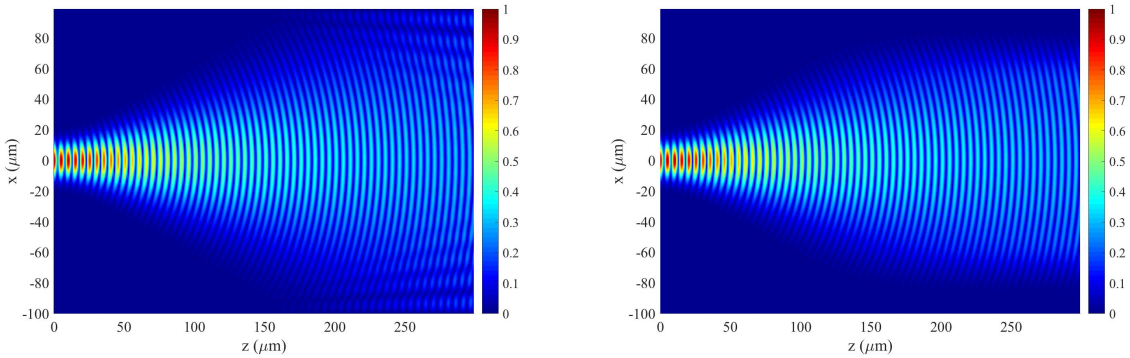
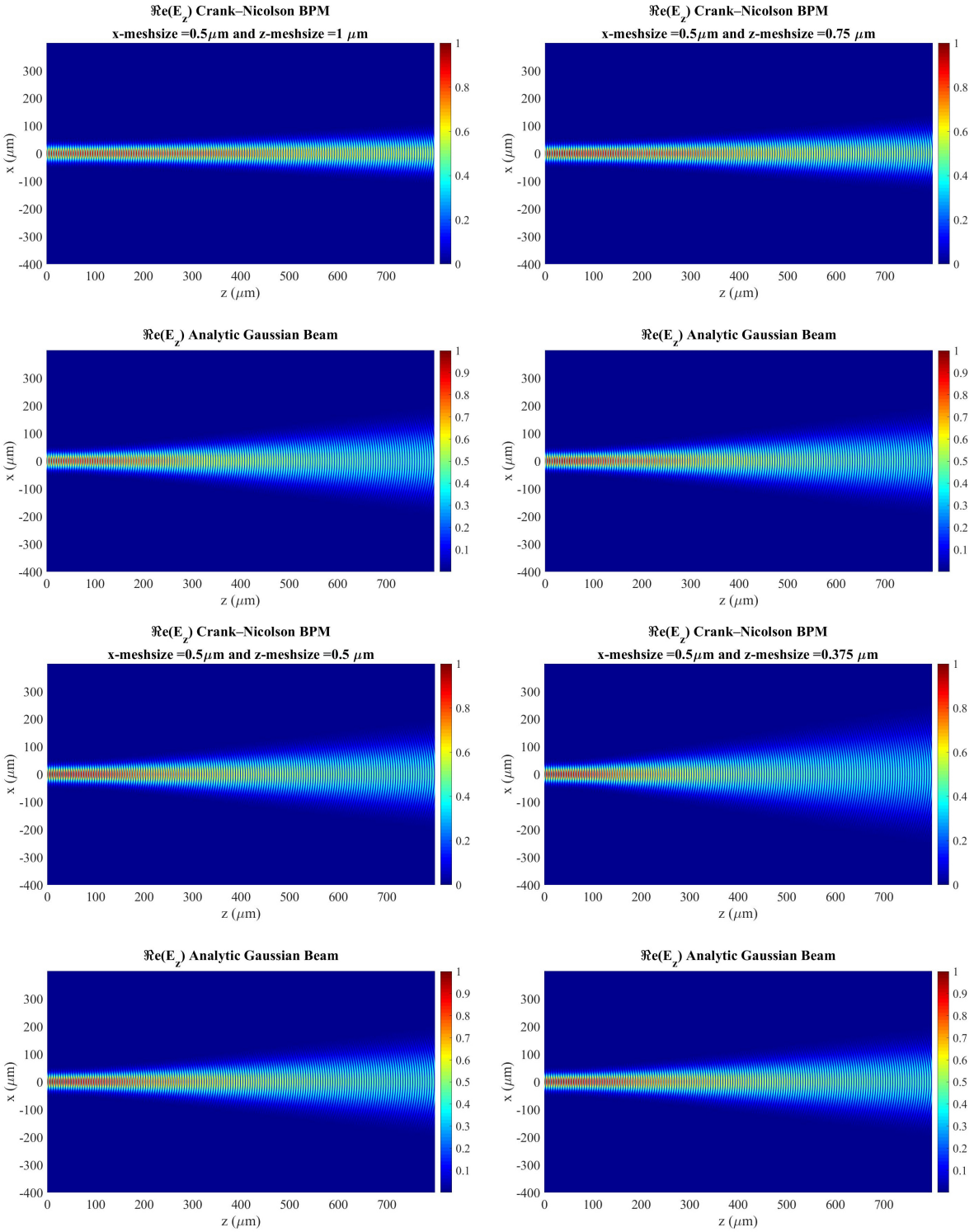



Figure 3: Propagation of electric field. On the left implementation of numerical method without ABC, on the right with ABC. Parameters used: z-meshsize = x-meshsize = 1 $\mu m$, wavelength ($\lambda$) = 10 $\mu m$, waist ($w_o$) = 10 $\mu m$, refractive index ($n_o$) = 1. For ABC simulation $\kappa_{max} = 0.0013$, $\delta = 50$ $\mu m$, $l = 200$ $\mu m$.

## 2.2 X and Z direction meshsize relation

In this section we made a research on how x-meshsize and z-meshsize relation affect simulation. In Figure 4 change of z-meshsize changes the result of numerical approach. At the point where z-meshsize is equal to x-meshsize numerical method has the most similarity to analytical solution. As result, we can conclude that once one changes x-meshsize, z-meshsize should be changed too and furthermore z-meshsize should be equal to x-meshsize. In the Figure below parameters are: x-meshsize = 0.5 $\mu m$ is kept constant, wavelength ($\lambda$) = 10 $\mu m$, waist ($w_o$) = 25 $\mu m$, refractive index ($n_o$) = 1, z-meshsize = 1, 0.75, 0.5, 0.375, 0.25 $\mu m$.


ℜe(E_z) Crank–Nicolson BPM
x-meshsize =0.5μm and z-meshsize =1 μm
ℜe(E_z) Crank–Nicolson BPM
x-meshsize =0.5μm and z-meshsize =0.75 μm
ℜe(E_z) Analytic Gaussian Beam
ℜe(E_z) Analytic Gaussian Beam
ℜe(E_z) Crank–Nicolson BPM
x-meshsize =0.5μm and z-meshsize =0.5 μm
ℜe(E_z) Crank–Nicolson BPM
x-meshsize =0.5μm and z-meshsize =0.375 μm
ℜe(E_z) Analytic Gaussian Beam
ℜe(E_z) Analytic Gaussian Beam
x (μm)
z (μm)

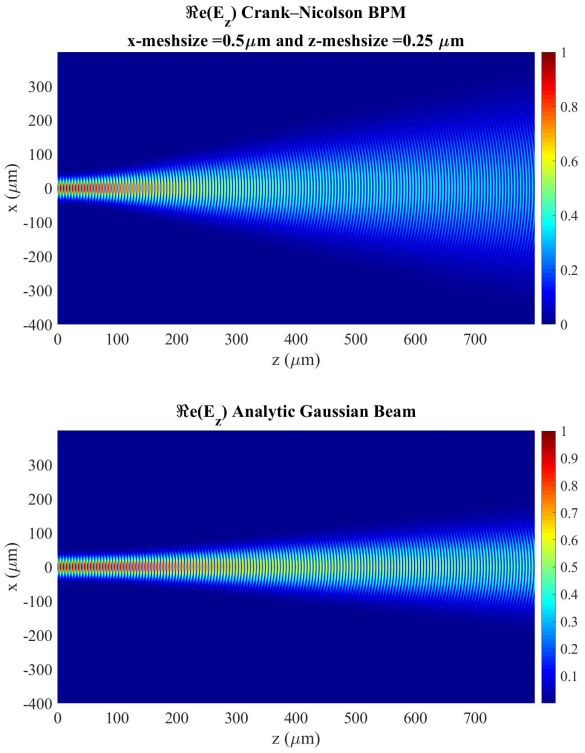



Figure 4: Numerically computed electric field at different relation of x-meshize and y-meshsize compared to analytical solution.

## 2.3 Error convergence

In this section error convergence is calculated as described in section 1.2.2, plots are presented in Figure 5 and results in Table 2. From table we see that if number of grid points increases twice, $L_2$ norm error decreases thrice , $L_\infty$ increases twice and so on. From Figure we see that of error has steady behavior in logarithmic scale. This convergence is reached at following parameters: wavelength ($\lambda$) = 10 $\mu m$, waist ($w_o$) = 25 $\mu m$, refractive index ($n_o$) = 1, z-meshsize = x-meshsize = $\{2, \sqrt{2}, 1, 0.5\sqrt{2}, 0.5, 0.25\sqrt{2}, 0.25\}$ $\mu m$.

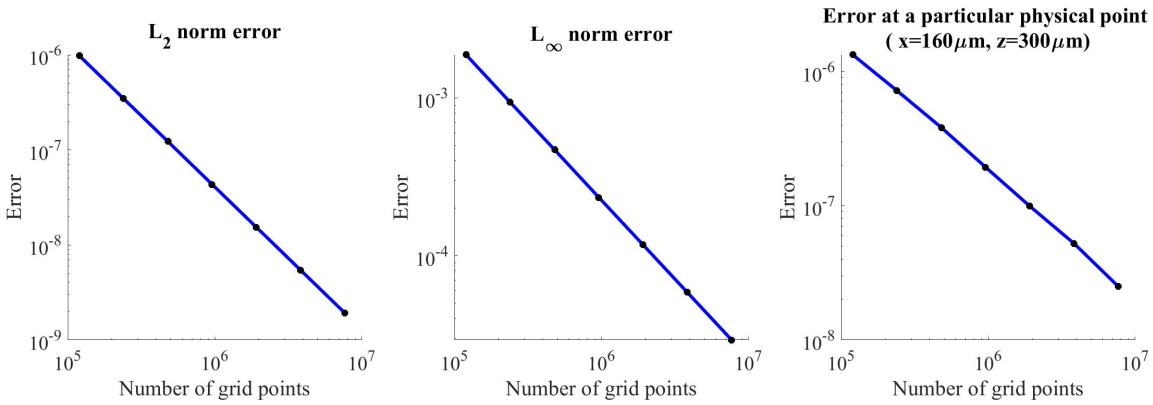



Figure 5: $L_2$, $L_\infty$ errors and error at a particular point.

| **Number of Grid points** | 1 | 2 | 4 | 8 | 16 | 32 | 64 | increase **2x** |
|---|---|---|---|---|---|---|---|---|
| $L_2$**norm error** | 1 | 0.3546 | 0.1254 | 0.0443 | 0.0156 | 0.0055 | 0.0019 | decrease **3x** |
| $L_\infty$**norm error** | 1 | 0.5001 | 0.2503 | 0.1250 | 0.0625 | 0.0312 | 0.0156 | decrease **2x** |
| **Error at a point** | 1 | 0.5410 | 0.2874 | 0.1457 | 0.0745 | 0.0393 | 0.0188 | decrease **2x** |

Table 2: An example table.

# 3 Conlusion

In this report Crank-Nicolson algorithm to simulate Gaussian beam propagation is shown. Results show high resemblance between analytic solution and beam propagation method. In the case when beam propagates in the direction of upper or lower boundary, absorption boundary condition method should be used to compensate the reflection. However in order to keep physical domain same, computational domain should be increased.

## Appendix A: Implementation of Crank-Nicolson method

```
// INITIALIZATION

// x-coordinate which specifies our computational range in micrometers.
x0 = 400
// meshsize in x-deirection in micrometers.
x_mesh = 1
// vector which contains x values in micrometers from -x0 to x0 with
step x_mesh.
x = from -x0 to x0 with step x_mesh.
// number of grid points in x-direction.
Nx = length(x)

// z-coordinate which specifies our computational range in micrometers.
zend = 800
// meshsize in z-deirection in micrometers.
z_mesh = x_mesh
// vector which contains z values in micrometers from 0 to zend with
step z_mesh.
z = from 0 to zend with step z_mesh
// number of grid points in z-direction.
Nz = length(z)
// waist in micrometers.
wo = 30
// wavelength in micrometers.
lambda = 10
// refractive index of air in vacuum.
no = 1
// spacial frequency with no.
ko = 2*pi*no/lambda
// refractive index of medium.
n = 1
// spacial frequency with n.
k = 2*pi*n/lambda

// 2 dimensional matrix with Nx*Nz size, complete electric field, 2-dimensional
grid.
E = zeros(Nx,Nz)
// 2 dimensional matrix with Nx*Nz size, spacial component of electric
field, 2-dimensional grid.
u = zeros(Nx,Nz)
// 2 dimensional matrix with Nx*Nz size, periodic component of electric
field in z direction, 2-dimensional grid.
f = zeros(Nx,Nz)
// 2 dimensional matrix with Nx*Nx size, tridiagonal matrix or Thomas
algorithm (see section 1.2).
L = zeros(Nx,Nx)
```

```
// 2 dimensional matrix with Nx*Nx size, identity matrix used in Thomas
algorithm.
I = eye(Nx,Nx)

// set first colomn of u to gaussian.
u(first_colomn) = exp(-(x/wo)^2)

// CORE ALGORITHM

// L matrix computation.
for i from 1 to Nx
   L(i,i) = 1/x_mesh^2*(-2+(ko^2-k^2)*x_mesh^2)
   if i>1
      L(i,i-1)=1/x_mesh^2
      L(i-1,i)=1/x_mesh^2
   end if
end for

// Spacial component of electric field computation.
for i_colomn from 2 to Nz
   u(i_colomn) = MatrixInverse(I-x_mesh*L/(4*1j*ko))*...
                                (E+x_mesh*L/(4*1j*ko)))*u(i_colomn-1)
end for

// Periodic component of electric field in z direction computation.
for i_colomn from 1 to Nz
   f(i_colomn) = exp(-1j*ko*z(i_colomn))
end for

// Complete Electric field.
E = Elementwise_multiplication(u,f)
```

## Appendix B: Implementation of analytical method

```
//INITIALIZATION

// x-coordinate which specifies our computational range in micrometers.
x0 = 400
// meshsize in x-deirection in micrometers.
x_mesh = 1
// vector which contains x values in micrometers from -x0 to x0 with
step x_mesh.
x = from -x0 to x0 with step x_mesh.
// number of grid points in x-direction.
Nx = length(x)
// z-coordinate which specifies our computational range in micrometers.
zend = 800
// meshsize in z-deirection in micrometers.
z_mesh = x_mesh
// vector which contains z values in micrometers from 0 to zend with
step z_mesh.
z = from 0 to zend with step z_mesh
// number of grid points in z-direction.
Nz = length(z)
// waist in micrometers.
wo = 30
// width in micrometers.
w = 0
// curvature in micrometers.
r = 0
// Gouy phase.
gouy_phase = 0
// wavelength in micrometers.
lambda = 10
// Rayleigh length in micrometers.
zR = pi*wo^2/lambda;
// refractive index of air in vacuum.
no = 1
// spacial frequency with no.
ko = 2*pi*no/lambda
// Electic field.
E = zeros(Nx,Nz)
// Electic field amplitude.
Eo = 1
// 2 dimensional matrix with Nx*Nz size, spacial component of electric
field, 2-dimensional grid.
```

```
// CORE ALGORITHM

// Electric field.
for i from 1 to Nz
   w = wo*sqrt(1+(z(i)/zR).^2);
   r = z(i)+(zR^2/z(i));
   gouy_phase = atan(z(i)/zR);
   E(i_colomn) = Eo*(wo/w)^0.5.*exp(-(x^2)/w^2) ...
                       *exp(1j*(k*z(i)+k*(x^2)/(2*r)-gouy_phase))
end for
```