Calculation of the Numerical Solution of Two-dimensional Helmholtz Equation

G. Hegedűs, M. Kuczmann

Laboratory of Electromagnetic Fields, Department of Telecommunications
"Széchenyi István" University, Egyetem tér 1, H-9026 Győr, Hungary
Pannon University, Deák F. u. 16, H-8360, Keszthely, Hungary
Phone: +3683545372, fax: +3683545373
e-mail: hegedus@georgikon.hu

Abstract: Many physical phenomena in acoustics, optics and electromagnetic wave theory are governed by the scalar wave equation. In the frequency-domain, the wave equation is the so called Helmholtz equation. In many cases, a theoretical numerical solution can be obtained for this equation by using finite differences with Sommerfeld boundary condition, resulting in a system of linear equations to be solved. The Sommerfeld boundary condition is used to solve uniquely the Helmholtz equation. However, in practice great difficulties are caused by the above method's great demand on operative storing capacity and calculation time. In the following contribution, a method for directly solving a linear equation system with a five off-diagonal matrix is presented. We show, that for this method, the number of computational steps and the memory requirement can be significantly reduced, and the possibilities for parallelization are also analyzed.

Keywords: Helmholtz equation, Sommerfeld boundary condition, finite difference method, sparse matrix

1. Introduction

Let $u = u(x, y)$ be the complex valued wave function on the region Ω, satisfying the Helmholtz equation [13][12]

$$\Delta u + k^2 u = 0\,, \qquad (1)$$

where

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\,, \qquad (2)$$

and k is the wave number, $k = \frac{2\pi}{\lambda}$, with λ being the wavelength. Let the shape of Ω be a rectangle, and the wave propagate in the Ω plane. A Sommerfeld boundary condition [1] is applied, i.e.,

$$\frac{\partial u}{\partial n} - iku = 0 \tag{3}$$

on the subset $\Gamma = \partial\Omega \setminus \Gamma'$, where $\partial\Omega$ is the boundary of domain, on the set Γ' the values of u are known. Here $\boldsymbol{n}$ denotes the unit normal vector of $\partial\Omega$, and i means the imaginary unit. Let the examined domain Ω be covered with an equidistant grid of spacing d, centered in a certain grid point with coordinates (x, y). Applying this choice, the discretized wave function is given only in the grid points as $u(x, y) = u(pd, qd) = u_{pq}$ [14], with $0 \le p \le a \quad 0 \le q \le b$, $p, q \in \mathrm{N}$. The discretization scheme is illustrated in Figure 1. The aim of the work is to determine the values of u in these points according to the prescribed boundary conditions.

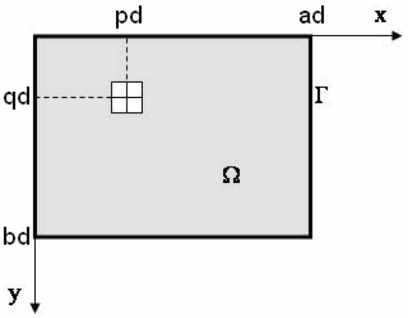

Figure 1. The studied domain Ω with the boundary $\Gamma = \partial\Omega$, and the applied grid centered in the gray point.

The discretization, together with the finite difference approximation, results in a system of linear equations. The system matrix is a large but sparse matrix with complex values. In order to obtain a sufficiently accurate numerical solution, the number of grid points per wavelength should be sufficiently large. As a result, the linear system becomes extremely large.

In this work, a method is presented to generate a solution of the problem. Efforts were made to place as much valuable (non-zero) data into the memory as possible and to apply the fastest possible operations.

The linear equation system describing the studied wave-range is composed of matrices with five non-zero off-diagonals, which can be transformed into matrices containing five valuable lines. This is, however, still too large to be kept in the memory simultaneously. We can achieve further memory size decrease by applying a sliding working-window in which the data transfer is minimized for the optimized operation. Within the work-window a direct procedure was used based on the Gaussian elimination. Further decrement in the necessary storing capacity can be achieved by dividing the domain.

Considering everything that depends on the capacity of the operating memory, we can achieve a good calculation capacity if the wave-range is optimally selected within the memory and the applied variables are ideally organized.

The effectiveness of the presented method is investigated by a numerical example of the beam propagation in a homogeneous medium.

2. The difference equations and the boundary equations

Inside the domain, the studied point is elements of an equidistant grid [11] which can be seen in Figure 2.

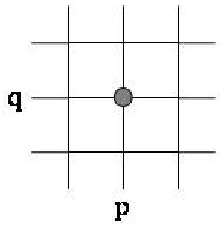

Figure 2. The equidistant grid for finite differences method.

The equation (1) can be approximated by the 5-point difference scheme [3][11]:

$$u_{p-1q}+u_{pq-1}+u_{p+1q}+u_{pq+1}-(4-k^2h^2)u_{pq}=0\,. \tag{4}$$

Two types of boundary points can be defined, where the values u are unknown. Figure 3. A) shows the $x=ad$ side points, except for the edges. Calculating with grid points on the side [7][8]:

$$(4-2ikh-k^2h^2)u_{pq}-2u_{p-1q}-u_{pq-1}-u_{pq+1}=0\,. \tag{5}$$

The same procedure can be followed on the other corners.

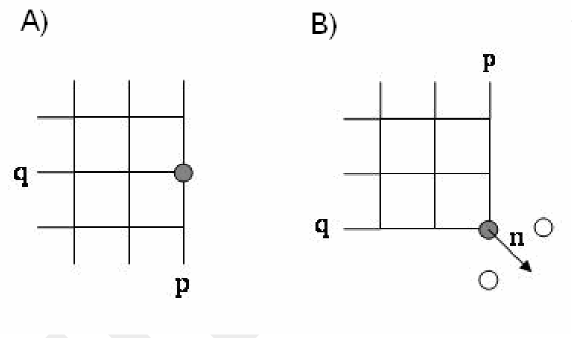

Figure 3. The boundary points of type A) side, B) corner.

Applying the $n = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$ condition and second-order accuracy Taylor series expansion, on the $x = ad \quad y = bd$ corner of the domain: [4]

$$\frac{\partial u_{pq}}{\partial n} = \left(\frac{\partial u_{pq}}{\partial x}, \frac{\partial u_{pq}}{\partial y}\right) n = \frac{\sqrt{2}}{2h}\left(2u_{pq} - u_{p-1q} - u_{pq-1} + \frac{1}{2}\frac{\partial^2 u_{pq}}{\partial x^2}h^2 + \frac{1}{2}\frac{\partial^2 u_{pq}}{\partial y^2}h^2\right). \qquad (6)$$

Based on (4) it yields on the $x = ad \quad y = bd$ corner :

$$0 = \left(2 - \sqrt{2}ikh - \frac{1}{2}k^2h^2\right)u_{pq} - u_{p-1q} - u_{pq-1}, \qquad (7)$$

which can be applied for the other corners as well.

3. Numerical solution

3.1. Optimal buffering in the operating memory

Applying the conditions (4), (5) and (7), a 5 diagonal homogenous linear set of equations is received containing the equation of $(a+1)(b+1)$ [2]. Figure 4.A) represents the extended matrix of simultaneous equations in the case of $a = b = 5$. The black places indicate zero values, while the white ones mean some complex values different from zero.

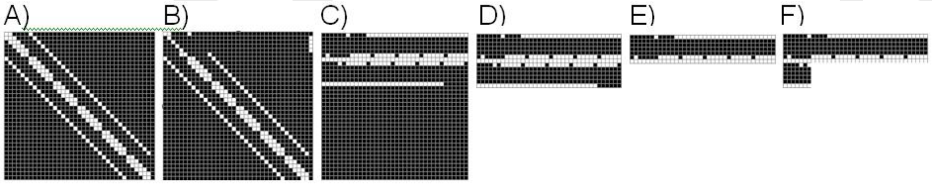

Figure 4. Storage layouts

Considering Dirichlet boundary points, inhomogeneous system of equation is resulted in that is illustrated the Figure 4.B). The Helmholtz equation with special preliminary conditions can be represented by a large system of equations with a sparse matrix. The size of the matrix is too large compared to the stored information [6]. The necessary storing capacity can be significantly reduced in the following way. The last column can be detached, and stored in a separate vector. The valuable diagonal dots of the system with coordinates (x, y) can be transformed into a row formation according to

$$(x, y) \mapsto (x - [(y/d - a - 1) \mod ((a+1)(b+1))]d, y). \qquad (8)$$

Figure 4.C) demonstrates the state after the row transformation.

Only the first $2a+3$ rows of the matrix are kept, it is unnecessary to reserve space for the others (Figure 4.D). After this reduction, extra care needs to be taken in order not to step out of the reduced $(2a+3)\times(a+1)(b+1)$ matrix during the elimination of the coefficients. During the Gaussian elimination modified considering (8), the coefficients under the row $(a+2)$ can be zeroed in increasing column-index (Procedure I), then the coefficients above row $(a+2)$ in decreasing column-index can also be eliminated in the same way.

A further decrease in storing capacity can be obtained if the line number $(a+3)$ and the last row are stored in two separate vectors, and afterwards, rows after line number $(a+2)$ are left out. The size of the resulting matrix is $(a+2)\times(a+1)(b+1)$, which can be seen in Figure 4.E).

For Procedure I, a sliding matrix of size $(a+1)\times(a+2)$ can be used, which is filled by values from separate vectors for an iteration step. This sliding matrix also stores the transitional values of the elimination process under the „transformed main diagonal", as it can be seen in Figure 4.F).

When moving the sliding matrix one step to the right, the new incoming column can be written to the place of the outgoing column, thus there is no need to rewrite the whole matrix. The columns can be referred to with the modulo-index $(a+2)$.

With the first procedure, the range above the row $(a+2)$ gets saturated with transitional values, which is to be eliminated with Procedure II.

3.2. The required storing capacity

It becomes obvious that, basically the distance of the two side-diagonals determines the size of the storage demand according to the above method. On the other hand, this distance depends on the values of the border dimensions *a* and *b* in the studied range.

Further decrement in the necessary storing capacity can be achieved by dividing the domain Ω. The question is, what shape and size is practical for the resulting sub-domains. According to Figure 4.E), in case of 16 byte storage of the complex values, the necessary storage capacity in byte units is

$$S(a,b)=16\big((a+2)(a+1)(b+1)+(a+2)(a+1)+2(a+1)(b+1)\big). \tag{9}$$

The area of the domain Ω was

$$A(a,b)=\big((a+1)(b+1)d\big)^2. \tag{10}$$

For a given area *A* the necessary storage capacity can be reduced by decreasing parameter *a* as it can be derived from the following expression

$$S(a,A)=16\left(\frac{\sqrt{A}}{d}(a+4)+(a+2)(a+1)\right). \tag{11}$$

Unfortunately parameter *a* can not be decreased arbitrarily because of the distortion of the result. According to the above considerations, it is effective to divide the system parallel to axis *y*, thus generating *n* congruent sub-domains [3]. Compared with the case of (9), the simultaneous storage of these data needs less capacity by a factor

$$\mu(a,n)=\frac{n\left(\left(\frac{a}{n}+4\right)\left(\frac{a}{n}+1\right)(b+1)+\left(\frac{a}{n}+2\right)\left(\frac{a}{n}+1\right)\right)}{(a+4)(a+1)(b+1)+(a+2)(a+1)}<\frac{n\left(\frac{a}{n}+4\right)\left(\frac{a}{n}+1\right)}{(a+2)(a+1)}<<1\ . \tag{12}$$

Although even according to (12) a considerable memory load decrement can be achieved, it is not necessary to store the data of the sub-domains simultaneously, the procession of their data is possible separately.

The memory need of a sub-domain in case of $a=b$ is

$$S(a,n)=16\left(\frac{a}{n}+1\right)\left(\left(\frac{a}{n}\right)^2+6\frac{a}{n}+6\right). \tag{13}$$

According to (13), the value of S increases strongly with the augmentation of a, but choosing the right number of sub-domains n, it can be divided into computationally manageable sub-problems. This experience can be derived from Figure 5.

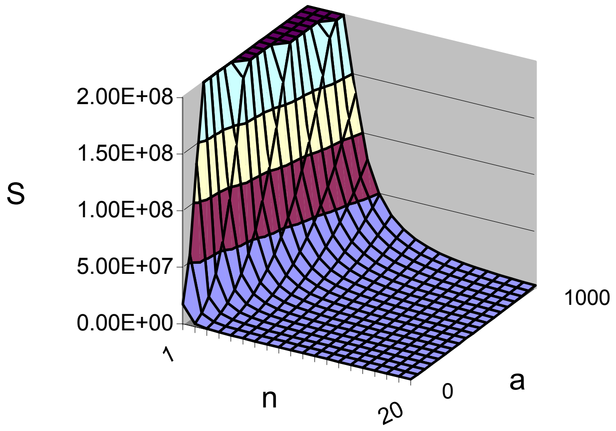

Figure 5. The necessary storing capacity as a function of the linear domain size a and the number of sub-domains n.

Applying the Huygens-Fresnel principle to the elementary domains, these sub-domains shall be treated as the starting objects of waves and the wave propagation between the sub-domains have to be ensured. In order to guarantee the proper wave propagation, domain Ω should not be divided into disjoint parts, but into overlapping regions. All of the two adjacent overlapping sets of points along the boundary are common. The sub-domains, containing known values of u at their boundary, can be calculated, applying Sommerfeld boundary condition in boundary points which are contained unknown values of u . Then the neighbouring sub-domains can receive the wave propagation data from the overlapping regions, i.e., through the u_{pq} values received from their neighbouring domains and through the relation (4) applicable as the continuation of the two common sets of points. In practice two grid lines of overlap in the division of Ω can ensure sufficient wave propagation.

The number of the possible starting threads in the parallel calculations is the number of the domains, where boundary values *u* is known. After the solution of one sub-domain two neighbouring sub-domains receive boundary values. Starting two new threads with each of these new boundary stripes the calculation can be carried out, the direction of the propagation remains the same. The gain from the parallel computing algorithm depends on the initial conditions, i.e., on the number of possible parallel treads and the position of their beginning sub-domain within the system. Generally only the following can be stated. For *n* sub-domains, even if the number of available computing units is sufficiently large, the necessary solution time of the parallel computation is at least n^{-2} times the sequential computing time. In the next section the duration of the calculation of a rather simple system is analyzed.

3.3. Simulation results

In Figure 6 the solution of the same problem with two Dirichlet boundary points can be seen for different subdivisions of the whole domain.

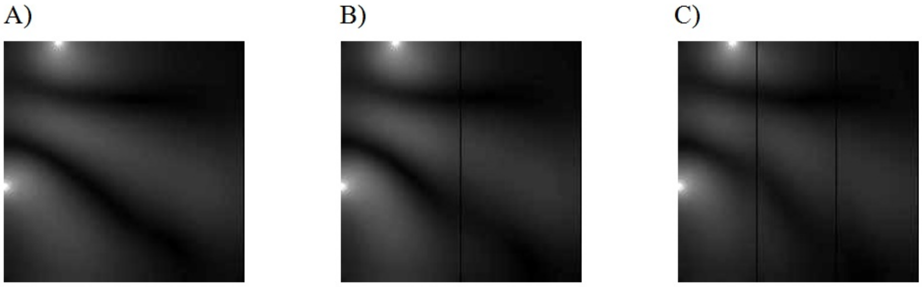

Figure 6. Subplots A), B), C) give the wave-space intensities corresponding to n=1, n=2, n=3 respectively, showing the overlapping regions. The applied data are the following. Grid size a=b=200, wavelength λ=0.0003m, d=0.000003m,

$$\Gamma' = \{(p = 45, q = 0, value = 1), (p = 0, q = 120, value = 1)\}$$

Calculation of error of solution with partition the domain compared to calculation of undivided domain is shown the following result. The relative error of the solution in Figure 6. B) according to norm $\| \, \|_1$ is 0.14, whereas according to norm $\| \, \|_\infty$ the relative error is 0.13 compared to the values of the non-divided solution in Figure 6. A). The relative error of the calculated solution in Figure 6. C) is 0.29 according to norm $\| \, \|_1$, while for norm $\| \, \|_\infty$ 0.22.

By compact storage of the matrix for determining the solution of the wave-space linear system of equations, not only the operative storing capacity load is decreased, but the necessary calculation time shortened, as well.

The operation steps demand corresponding to the case given in Figure 4.F) with the constraint $a = b$ is

$$M(a) \approx ((a+1)^2 - 1)a^2 + ((a+1)^2 - 1)(a+1) + (a+1)^2 = (a+1)^2(a^2 + a + 1) + a \, . \quad (14)$$

After dividing the domain Ω to *n* sub-domains, the necessary operation steps for one part can be reduced to

$$M_0(a,n) \approx \left(\frac{a}{n}+1\right)^2 \left(\left(\frac{a}{n}\right)^2 + \frac{a}{n} + 1\right) + \frac{a}{n}. \tag{15}$$

If *m* parallel computational threads can be started, the wave propagation has to be followed in all the remaining sub-domains, thus the total number of operation steps is

$$M_m(a,n) \approx nm\left(\left(\frac{a}{n}+1\right)^2 \left(\left(\frac{a}{n}\right)^2 + \frac{a}{n} + 1\right) + \frac{a}{n}\right) \tag{16}$$

Comparing *M* and M_m of equations (14) and (16) yields to an operation-saving factor, which relates the complete calculation within the domain to the calculation performed in the undivided Ω domain as

$$\vartheta_m(a,n) \approx \frac{M_m}{M} = \frac{nm\left(\left(\frac{a}{n}+1\right)^2 \left(\left(\frac{a}{n}\right)^2 + \frac{a}{n} + 1\right) + \frac{a}{n}\right)}{(a+1)^2\left(a^2+a+1\right)+a} < 1 \qquad (1 < n). \tag{17}$$

By dividing the problem into sub-domain problems, the necessary real calculation time compared to that of the undivided problem depends on the value of ϑ_m, but of course, it is also affected by the programming technique.

As an example, let $a = b$ hold, and $t_2(a,n)$ denote the necessary computation time of the space with two Dirichlet boundary points in the case of *n* subdomains with the data visualized in Figure 6. Thus the time-saving factor $D1$ according to this calculation is

$$D1(n) = \frac{t_2(a,n)}{t_2(a,1)}. \tag{18}$$

On the basis of measurements, according to relation (18), the values $D1$ are given in Tab. 1. At a fixed grid size *a* let us introduce another factor $D2$ in order to facilitate the exploration of the relation between $D1$ and ϑ_2 as

$$D2(n) = c(n)\vartheta_2(a,n). \tag{19}$$

Expression *c*(*n*) can be determined the following way. Condition

$$D1(n) \approx D2(n) \tag{20}$$

has to be satisfied, in order to make $D1$ and ϑ_2 comparable. Based on the experiments, condition (2) is ensured by relation

$$c(n) = n. \tag{21}$$

The value $D2$ calculated according to condition (21) can be seen in Table 1., while the realization of (20) is illustrated in Figure 7.

Table 1. The values D1 and D2 for various numbers of sub-domains n in case of c(n)=n.

n	2	3	4	5	6	7	8	9	10
D1	0,501	0,230	0,132	0,092	0,064	0,050	0,040	0,032	0,028
D2	0,508	0,229	0,131	0,085	0,06	0,045	0,035	0,028	0,023

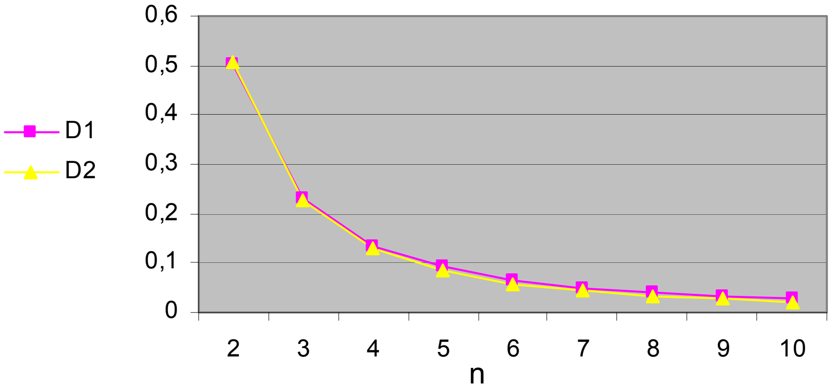

Figure 7. The values D1 and D2 for various numbers of sub-domains n in case of $c(n) = n$.

4. Conclusion

With the development of hardware and software technology, together with increasing calculation capacities and memory-optimization, the direct method can also be used successfully in the case of small-sized electromagnetic wave-ranges of relatively long wavelength. For reducing the required storing capacity, a special matrix reduction method was introduced in this paper, using sliding matrices. For aiding parallel computing, a wave space dividing method was also introduced and tested with small overlaps ensuring the wave front transmission between the space parts with reasonable results.

Beyond the studied discretization method, there are many other possibilities to solve the above problems. These solutions can be characterized by the number of computational steps and the necessary memory capacity for the sufficiently accurate results, which, at the end, determine the necessary computing time. The following estimations are based on the case of square grids with $a=b$. The method considered in Section 3, is based on stripped Gaussian elimination, its necessary computational capacity according to (14) is $O(a^4)$. The filling does not impact the whole matrix, but only approximately a^3 elements. Applying LU decomposition, a quicker solution can be achieved with significantly larger storage need. The conjugate gradient method is much more favourable both in computational (less than $O(a^4)$) and in storage needs, but it demands a symmetric positive definite matrix, which has to be pre-conditioned for an efficient convergence. The special shape of the studied domain makes it possible to apply

FFT (Fast Fourier Transform), which presents a quick solution with a computational requirement of O(a^2log$a$) steps, and its storage need is O(a^2log$a$). Wavelet based differential equation solving methods of order a exist [5], but their application range is limited mostly to elliptic differential equations and Schrödinger type eigenvalue equations [9], and their straightforward representation of the kinetic energy can lead to systematic errors [10], which results in slower convergence. The boundary element method's storage capacity demand is approximately the same as that of the finite differences method, but its algorithmical complexity is O(a^3). The best solution seems to be the multigrid method, since its computational need is O(a^2), and memory demand is about the same as in the conjugate gradient method. More accurate and effective solution can be achieved by unevenly meshed multigrid method.

The main advantage of the method presented in this article is the simple algorithm which can be easily applied even if no complex, efficient program is available, and the development time has to be minimal. It can also play a role in the design of the uneven grid multigrid method by giving a rough scale solution of the problem as a starting point.

References

[1] Arnold, S.: *Mathematische Theorie der Diffraction*, Math. Ann., Vol. 47, pp. 317–374, (1896).

[2] Buzbee, B. L., Dorr, F. W., George, J. A., and Golub, G. H.: *The direct solution of the discrete Poisson equation on irregular regions*, SIAM J. Numer. Anal., Vol. 8, pp. 722-730, (1971).

[3] Choi, C.T.M., Webb, J.P.: *The wave-envelope method and absorbing boundary conditions*, IEEE Transactions on Magnetics, Vol. 33, Issue 2, pp.1420 – 1423 (1997).

[4] Claudio M.: A *Reference Discretiazion Strategy for the Numerical Solution of Physical Field Problems,* Advances In Imaging and Electron Physics, Vol. 2. (2002).

[5] Dahmen, W.: *Wavelet methods for PDEs – some recent developments*, J. Comput. App. Math., Vol.128, pp. 133-185, (2001)

[6] Duff, I. S., Erisman, A. M., and Reid, J. K.: *Direct Methods for Sparse Matrices*, Clarendon, Oxford (1986).

[7] Iványi, A.: *Folytonos és diszkrét szimulációk az elektrodinamikában (Continuous and discrete simulations in electrodynamics)*, Akadémiai Kiadó, Budapest, pp. 180-240, (2003).

[8] Lois C. M., Romeo F. S., David E. K., Hafiz M. A.: *Additive Schwarz Methods with Nonreflecting Boundary Condition of Helmholtz Problems*, Contemporary Mathematics, Vol. 218, pp. 349-353, (1998).

[9] Nagy, Sz., Pipek, J.: *A wavelet-based adaptive method for determining eigenstates of electronic systems*, Theor. Chem. Acc., Vol. 125, pp. 471-479, (2010).

[10] Pipek, J., Nagy, Sz.: *Artifacts of grid-based electron structure calculations*, Chem. Phys. Lett., Vol. 464, pp. 103-106, (2008).

[11] Shashkov, M.: *Conservative Finite-Difference Methods on General Grids*, CRC Press, Boca Raton, FL (1996).

[12] Shashkov, M., Steinberg, S.: *Support-operator finite-difference algorithms for general elliptic problems*. J. Comput. Phys., Vol. 118, pp. 13 1-15 1, (1995).

[13] Simonyi, K., Fodor, Gy.: *Electrodynamics*, Tankönyvkiadó, Budapest Vol. 2, pp. 290–364, (1967).

[14] Strand, B.: *Summation by parts for finite difference approximation for dldx.* J. Comput. Phys. Vol. 110, pp.47-67, (1994).