0.0.1 Comparison of finite-difference and spectral methods

Summary:
This practice-track section compares two very different methods of numerical beam propagation in radially symmetric geometry

- Double-coding is an effective way to discover not only bugs, but also algorithmic problems in simulation codes. Here we compare results from *different algorithms*, which is a very strong test. It is practically impossible for the two methods used here to be implemented incorrectly yet giving the same numerical results.
- The physical context chosen is once again that of the Poisson‘s bright spot. This problem is a useful stress test for virtually all aspects of the given algorithm (resolution and numerical dispersion issues, stepping procedure, absorbing boundary guard implementation, etc)
- This exercise is an opportunity for the reader to hone intuition that often guides important design decisions in scientific computation practice. In particular, here one can get a good sense for how DHT-based and Crank-Nicolson based BPM compare in terms of numerical effort required to obtain similar-quality data.

The context chosen for this is the simulation of the Poisson‘s bright spot. The reason it is a good stress test for a beam propagation algorithm were appreciated in the earlier exercises. Here, we take advantage of the fact that a very simple geometry results in richly structured beam solutions which are ideally suitable for comparison of beam propagation methods.

Very often, especially at a beginning of a project that deals with development of simulation software, one has to ask: "Which method is better?" Of course, "better" means nothing without specification of what one needs to achieve. The proper question to ask would be, which method, and in our case we compare DHT-BPM versus the Crank-Nicolson based radial finite-difference method, is more suitable to produce a solution of the given problem (radial intensity profile behind the circular obstacle, in this case) with a given accuracy? This is the point of this practice track package.

Perhaps a good point to start from is to try to guess which of the two approaches will be faster to produce an accurate numerical solution. The answer may not be obvious. On one hand, DHT is well suited to this task since it is a spectral algorithm and it is free of numerical dispersion issues. On the other hand, Crank-Nicolson method has a more favorable complexity scaling; with the effort being proportional to the number N of grid points populating the computational domain. In theory it must outperform the DHT algorithm, complexity of which scales with the number of grid points squared. However, the outcome will depend on whether or not the N is sufficiently large for the complexity scaling to determine the answer...

Besides the verification of the correctness of the two codes, their perfomance comparison is an important aspect of this exercise. To compare apples to apples, one should select the problem in a way that actually make the computation speed comparison meaningfull:

- To minimize the number of possible issues affecting the quality of the simulated beams, no boundary guard will be used. This also means that long propagation distances can not be simulated (Why?).

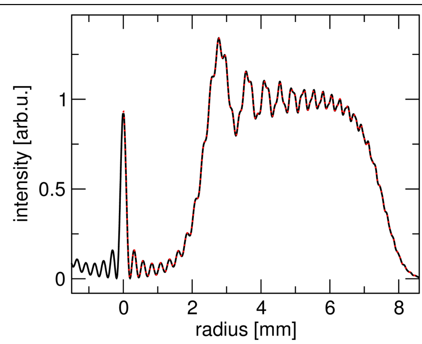
- To minimize the effects of high spatial frequencies (which behave in very different ways in the two method being compared), the initial condition is set up with a smooth screen edge. This is potentially dangerous, and one should check that important features of the solution are not affected.
- The Crank-Nicolson method is paraxial, while DHT-BPM can simulate both paraxial and non-paraxial regimes. To make the comparison quantitative, the paraxial propagator will be used in the DHT method.

Task 1:

Set up two simulations, one using DHT-BPM and the other working with CN-BPM algorithm, to simulate the radial intensity profile behind an opaque circular obstacle illuminated by a collimated super-gaussian beam. Assume that the simulation results must represent the field with some reasonable sampling density up to certain maximal propagation distance. This means that the DHT method must be executed not in a single jump, but as a series of shorter propagation steps. Adjust this to what you deem to be a reasonable sampling along z, and compare running times in the two methods. Demonstrate that the calculated intensity profile agree with each other.

Solution:

Two instructor's solution programs have been set up in this directory that are implementing DHT-BPM (*p2DHT.m* and supporting sources), and C-N BPM (*cn-bpm-radial.cc*), respectively. Parameters in both are hard-coded such that a particular scenario is simulated with good quality, and a nice agreement between the results is demonstrated between the two methods. The following picture illustrates that the radial intensity of the propagated beams agree very well:



Numerical comparison and verification of two different numerical methods. Initially collimated super-gaussian beam diffracts around a circular screen, and the resulting radial intensity profile is simulated. The red dotted curve corresponds to the Crank-Nicolson method, and the black to the DHT-BPM (with data extended symmetrically to negative radii). On the scale of this figure, the agreement is perfect as one should expect.

Task 2:

Explore the effects of non-physical parameter variations departing from the instructor's solution template. The goal is to get a feeling about what is a still acceptable grid resolution, an appropriate integration step, and how do they affect running times, etc. This is an exploratory, open-ended exercise left to the reader.

Task 3:

Next, modify the smooth-edge initial condition (or fuzzy screen edge) in the Crank-Nicolson program to one with a sharp edge. Observe the consequences and make a few attempts to eliminate or minimize numerical artifacts that are caused in the solution by the sharp initial condition. Discuss the origin of these artifacts, and think about how would the *implicit* Euler method behave in these circumstances.

Solution:

Instructor‘s solution has been placed in the sub-directory *Task3*. This first figure shows result comparison between the base run (which uses a smooth initial condition) , and a simulation starting from an initial beam inident on a circular screen with a sharp edge.

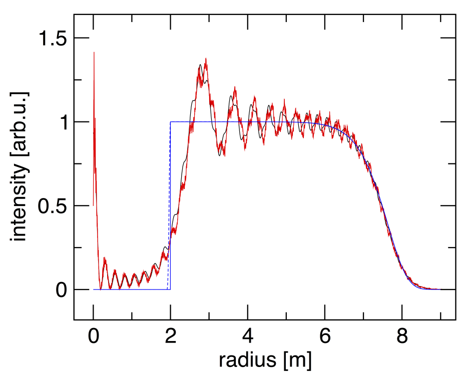
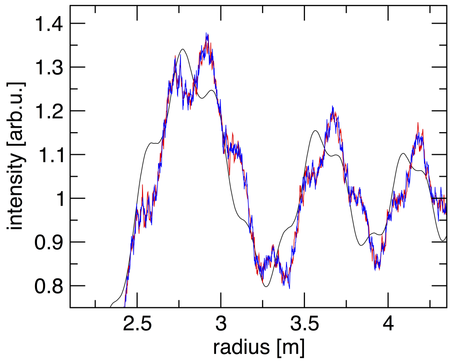


Illustration of the influence the high transverse wavenumbers, present in the initial condition, have on the propagated beam. The initial condition for the modified simulation run is shown as solid blue line. The base-run initial condition is shown in dashed and is barely distinguishable from the former on the scale of this fugure. Note that the only difference between the two is the sharp edge at $r=2$ mm. The black and red lines represent the propagated beam intensity profieles for the fuzzy and sharp obstacle edge.

There are two kinds of differences between the base run and the new simulation with an initial condition with a sharp screen edge. The first is a “phase shift” between the beam intensity modulations. One can assume that this likely physical and correct, reflecting what is a slightly different effective size of the screen. The second difference is the appearance of high spatial frequencies in the new solution.

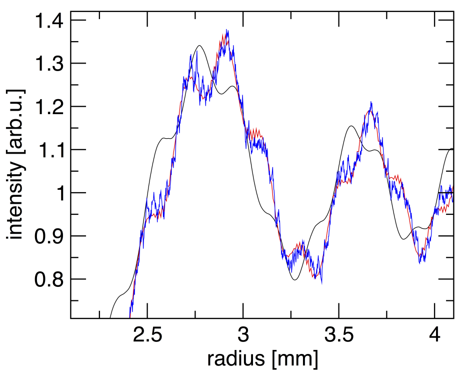
The waves with high spatial wavenumbers are naturally present in the initial condition with a sharp feature, while they are much more damped in the situation with a fuzzy screen. However, while the C-N method does not amplify them, and also does not damp these waves, these waves do have wrong dispersion properties. The consequence o this is that they diffract “out of sync” with other components of the beam’s spatial spectrum. This is why even after a short propagation distance they appear as a “noise” superimposed on the background of a smooth solution.

The question is if it is possible to improve things by selecting better or coarser spatial grid resolution or perhaps shorter integration step. Let us look at the effect of a shorter integration step first:



Comparison of two simulations (red and green lines) with different integration steps. It is hardly possible to tell which represents the run with the finer integration step. This means that the integration step is not the parameter that "causes" the artifacts. The black curve shows the result of the base run for the reference.

Clearly, for this particular regime, decreasing Δz did not help; the strength of the noise is roughly the same. So, in the following picture we try a run with a coarser spatial grid:



The solution shown in red is the one obtained on a coarser grid. It is evident that the amplitude of the unwanted noise diminished, though the artifacts were not eliminated completely.

Comparison with the previous "noisy" solution shows that that high spatial frequency artifacts are somewhat less visible, but the result is certainly far from optimal. Decreasing spatial resolution even further is not good either, as the sampling of physical features present in the beam profile is already quite coarse. Moreover, it should not be difficult to realize that the only reason for less noise is that the waves with highest transverse wavenumbers have been removed from the picture. The behavior of other waves actually did not improve at all. This is therefore not really a viable way to deal with the problem.

In summary, the take-home lesson concerning the C-N approach is this:

- Spatially sharp features in the initial condition act a source of numerical noise which does not go away during simulations
- Decreasing the integration step does not completely eliminate the problem
- These artifacts are less severe on coarser grids, but can not be easily removed.

The numerical experimentation with the simulation in this package suggests that there is something "wrong" with the numerical waves that have large transverse wavenumbers, or equivalently, only encompass a small number of grid points per transverse wavelength. Of course, this issue should be familiar to us from the exploration of the Maxwell-solver properties in the first Chapter; in any finite-difference based approach, the extreme waves with fastest oscillation can not propagate properly. In fact, the waves with maximal possible wavenumber on a discrete grid exhibit plus-minus oscillations and as such they cannot distinguish between right and left propagation direction and that is why they do not propagate at all. Once they are created within the simulation domain, they will remain there. The remedy is to ensure that too high spatial frequencies are not generated in the first place. In most situations it is possible. In this example, one should work with a fuzzy edge, such such that the blurring is much smaller than one Fresnel zone width, *and* use very high spatial grid resolution so that the screen edge appears smooth on the scale of the grid spacing. Obviously, the penaltyis a singnificntly longer computation.