Bergen, Norway

BNAM 2010

May 10-12

Behind Norstøy – Technology of a national tool for road traffic noise mapping

Rolf Tore Randeberg
Department of Acoustics, SINTEF, S. P. Andersens vei 15B, NO-7031 Trondheim, Norway, rolfr@sintef.no

Herold Olsen
Department of Acoustics, SINTEF, S. P. Andersens vei 15B, NO-7031 Trondheim, Norway, heroldo@sintef.no

The third generation of *NorCalc*, a tool for road traffic noise calculation was released in 2009. This work has been commissioned by the Norwegian Public Roads Administration. In the development of the software, several approaches have been taken to reduce the total calculation time. This includes utilizing free CPU resources on networked computers, exclusion of non-significant sources and simplified calculation of some receiver points.

1 Introduction

The development of a new tool for calculation of road traffic noise, *NorStøy*, has for several years been commissioned by The Norwegian Public Roads Administration (NPRA). The tool consists of a user interface (*Noise*) and a calculation model (*NorCalc*), the latter developed by SINTEF. The tool is suited for both large scale mappings as required by the EU (European noise directive, 2002/49/EC), and the assessment of noise levels around individual buildings.

NorCalc's calculation kernel, which calculates the point-to-point ground attenuation, is a Fortran implementation of the Nord 2000 method [1, 2]. The Nord 2000 method has a very high computational power consumption. Hence, there is a need for procedures to achieve a reasonable balance between calculation time and precision, when the method is used for large scale mapping.

The first generation of the calculation model, NorCalc 1, was released in 2007. The model requires the following input to compute noise levels:

- **Topography:** primarily a terrain elevation model. It can also include descriptions of artificial objects such as buildings, screens, mounds, roads, etc. Optionally, a file to define various ground type regions can be included.
- **Source description:** defines source quantities, time distributions, vehicle class distributions, etc. for the sources which are to be included in the calculation.
- **Task description:** describes the region (grid) to be calculated. An optional file can be used to define the location and height of facade points to calculate noise levels in front of building facades.

In the first version of the model, the result of the calculation was a grid of noise levels, and noise levels for the facade points (if defined). Noise contours were created by the user interface program, Noise, based on the grid of noise values.

The next generation, NorCalc 2 (released 2008), had a completely rewritten internal topography model, based on constrained triangulation of the input topography files [3, 4, 5]. Noise contours were now generated by NorCalc, not Noise, and the internal workload was improved (as described in 2.1).

The latest generation of the calculation model, NorCalc 3 (released 2009), supports the requirements of the new software for calculation of indoor noise, *Støybygg 3*, which was released in March 2010. Facade points are now created automatically for specified buildings. 3D-visualization of the internal topography model is also supported.

This paper presents the procedures that had to be implemented to achieve a reasonable balance between calculation time and precision during large scale mapping.

2 Procedures for reduction of calculation time

Reduction of the calculation time requirements of any noise computation software can be achieved through three fundamental approaches:

- Increase available computational power
- Reduce the number of point-to-point calculations by reducing the number of included noise sources or calculated receiver points
- Reduce the point-to-point calculation time

The latter approach was taken prior to the development of NorCalc, by implementing a Fortran version of the Nord 2000 method (which previously was available as MATLAB source code). The first and the second approaches have been used in NorCalc to further reduce the computation time without sacrificing precision of the noise levels.

2.1 Increase computational power

Increased computational power is achieved by splitting the total calculation into smaller tasks, and distributing these subtasks to available computers in the network. The computers can be dedicated servers or ordinary office workstations. The distributed subtasks are calculated with low priority to prevent interference with ordinary use of the workstations.

Since the first version, the NorCalc software suite has included four separate applications designed to facilitate the divide-and-conquer approach:

- **Main program:** responsible for reading input files, splitting the task into subtasks, collecting subtasks and generating accumulated results
- **Calculation cell:** responsible for calculation of a subtask received from the main program
- **Status monitor:** a background service responsible for managing the distribution of subtasks to other computers
- **Status monitor user interface:** allows user access to configure the status monitor and gives access to current status of the calculation(s)

After the initial processing of input files and task splitting, the main program queries the local status monitor for free calculation cells, and distributes the subtasks sequentially to the available cells. The local status monitor queries the status monitors on a number of given remote computers. These monitors create instances of calculation cells if necessary, and report their address back to the local status monitor, and subsequently the main program.

In the first version of NorCalc, the main program was responsible for performing significance testing of sources (see section 2.2). This had some negative side effects, such as high bandwidth requirements (as the subtasks were sent repeatedly back and forth between the main program and the calculations cells), and high CPU requirements on the local computer. Since version 2 however, the significance testing is handled by the calculation cells. Hence, except for processing of input files, most of the workload is now handled by the calculation cells.

2.2 Reduce number of included noise sources

The reduction of included sources is carried out in several steps. During the task splitting, the number of sources (roads) included in a subtask is limited to roads closer than 500 m to the subtask boundary. The roads are not cropped at this point, therefore it is possible that parts of a road are more than 500 m away from the subtask. When the subtask is received by the calculation cell, the roads are cut into straight segments corresponding to the coordinates defining the road center line. Segments which are too far away from the subtask boundary are excluded from the calculation. These steps affect all receiver points in the subtask.

In the final step, a significance test of the remaining segments is done for each receiver point. The noise contribution from each road segment is calculated using the fast, but lesser accurate, industrial noise method. A total noise level is calculated as the sum of all segment contributions. The segments are then sorted according to decreasing contribution. Contributions from the sorted segments are accumulated until the ratio of the accumulated sum and the total reaches a given tolerance, typically 0.2 dB. The noise level of the last included segment is termed $L_{\mathrm{lim.}}$ This limit level is subject to some adjustments:

- L_{lim} cannot be lower than 30 dB below the lowest noise level of interest (which typically is set to 50 dB)

- L_{lim} cannot be greater than 40 dB

These adjustments assure that the limit level is within an acceptable range. The adjusted L_{lim} is then used to test the significance of all segments; Segments contributing less than L_{lim} are excluded from the calculation (of the current receiver point). In average, the result is a 50–80 % reduction in the number of source segments.

It should be noted that during the significance test, the topography profiles of the point-to-point calculations are temporarily saved. The profiles corresponding to the significant segments are then reused during the subsequent Nord 2000 calculation. Thus, topography profiles are created only once per receiver point. The reuse of profiles gives a small, but nevertheless significant reduction of the total calculation time.

A final approach to reduce the number of calculations is to skip some of the point-to-point calculations for reflected sound. For each source point along the road, subsources are created for each of the road lanes, and for three separate source heights. For direct sound, each of these subsources is subject to ordinary point-to-point calculation. The subsources are then grouped according to similar contribution. For each group, the contribution from reflected sound is calculated *only once* (for one of the subsources) and the resulting ratio of reflected to direct contribution is applied to estimate the reflection contribution for the other subsources in the group.

2.3 Reduce number of calculated receiver points

As noted in the introduction, the internal topography is created by constrained triangulation of the input topography files. Similarly, the calculation points of the regular grid defined in the task specification is included in a separate triangulation model. This model includes additional points along constriction lines such as building outlines and screens defined in the topography, as shown in Figure 1.

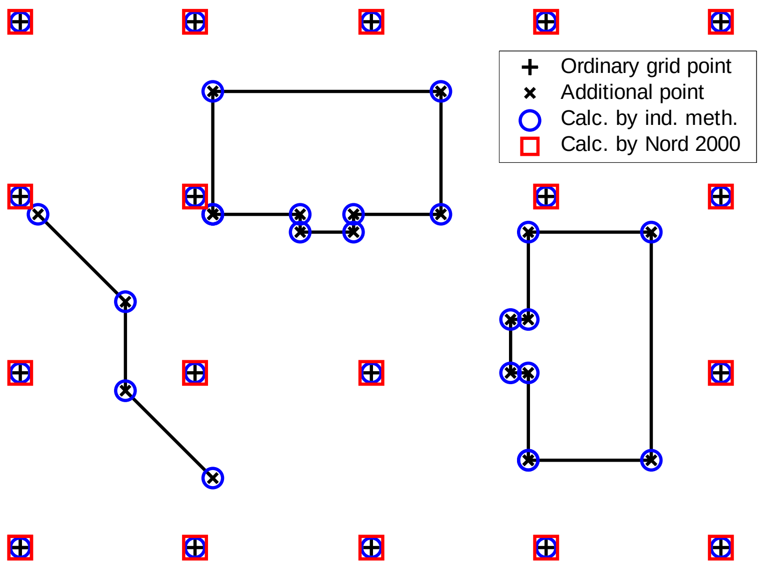

Figure 1. Receiver points including in noise contour generation. Ordinary grid points (*G*) are represented by '+', while additional points (*A*) representing buildings and screens are shown as 'x'. The *A* points are calculated by the industrial noise method only, represented by blue circles. The *G* points are calculated by both the industrial method and the Nord 2000 method, represented by red squares. The noise levels of the *A* points are then adjusted by a value determined from the difference between the methods at *G* points in the vicinity.

The purpose of this is to facilitate generation of noise contours constrained by buildings and screens, *i.e.* to avoid interpolation of noise levels across noise barriers. The drawback is that more receiver points must be calculated.

To minimize the calculation burden of the additional points (*A*), these points are calculated using the industrial noise method only (as part of the significance test discussed above). The ordinary grid points (*G*) are calculated as usual, using both the industrial noise method (per significance test) and Nord 2000. Since the *A* points are not calculated by the Nord 2000 method, corrections must be added to the *A* points to achieve noise levels consistent with the *G* points.

For a given *A* point, *G* points in the vicinity is located. For each of the *G* points, the difference between the noise levels of the industrial noise method and the Nord 2000 method is calculated. The correction applied to the *A* point is then taken as a distance-weighted average of these differences.

It should be emphasized that this simplified calculation of points along buildings and screens applies only to the generation of noise contours, *i.e.* the calculation of facade points around buildings is *not* affected.

Acknowledgements

The development of NorCalc has been funded by The Norwegian Public Roads Administration. Thanks to Ingunn Milford, Ida Solheim and Torunn Moltumyr for useful input and thorough testing through three generations of the software.

References

[1] B. Plovsing and J. Kragh, Nord2000. *Comprehensive Outdoor Sound Propagation Model. Part 1: Propagation in an Atmosphere without Significant Refraction*, DELTA Akustik & Vibration, Rapport AV 1849/00, revised Hørsholm 2006.

[2] B. Plovsing and J. Kragh, *Nord2000. Comprehensive Outdoor Sound Propagation Model. Part 2: Propagation in an Atmosphere with Refraction*, DELTA Akustik & Vibration, Rapport AV 1851/00, revised Hørsholm 2006.

[3] Wikipedia, *Delaunay triangulation*, http://en.wikipedia.org/wiki/Delaunay_triangulation, Accessed 2008

[4] S. Peterson, *Computing constrained Delaunay triangulations*, http://www.geom.uiuc.edu/~samuelp/del_project.html, Accessed 2008

[5] G. Leach, *Delaunay Triangulation [C source code]*, http://goanna.cs.rmit.edu.au/~gl/research/comp_geom/delaunay/delaunay.html, Accessed 2008