International Conference on Analysis and Geometry

August 30 – September 3, 1999, Novosibirsk, Russia

Abstracts

AGRANOVSKY, MARK L., *Bar Ilan University, Israel*;
email: agranovs@macs.cs.biu.ac.il

Geometry of Stationary Sets for the Wave Equation in R^n

Characterization of *nodal sets* is one of classical problem in analysis. Nodal sets are the zero sets of eigenfunctions of the Laplace operator or, equivalently, sets of zeros (stationary points) of time-harmonic solutions to the wave equation. This problem has been studied by many authors (Courant-Hilbert, C. Fefferman, Donelli and many others). An important property is that the energy is preserved in domains bounded by nodal surfaces. While analytic structure of nodal sets is well studied, much less is known about their geometry. On the other hand, for initial data with sufficiently fast decay at infinity, the solutions have infinite Laplace spectrum and therefore the sets of time-invariant zeros (stationary sets) are intersections of infinite families of nodal sets. It appears that in this case the stationary sets have a nice and quite transparent geometry. E. T. Quinto and the author (1994) completely described the geometry of stationary sets for the wave equation in the plane (membrane equation) with compactly supported initial distribution. The case of higher dimensions is more difficult and only partial results are known. C. Berenstein, P. Kuchment and the author (1996) proved that stationary sets cannot have bounded closed component if the initial distribution vanishes at infinity is sufficiently fast. Very recently E. T. Quinto and the author gave the complete description of stationary sets for the wave equation with finitely supported initial distributions (finite number of point sources), in the space of $\mathbb{R}^n$ of arbitrary dimension. These and related results will be presented and connections of the problem to the Radon transform on real quadrics, uniqueness theorems for Schrödinger equation and divisibility problem for harmonic polynomials will be explained.