Priority	Category	Name	Description
1	Documentation	DDA vs. FDTD	Mention in the manual and/or on the web site current state-of-the-art in DDA vs. FDTD
2	Usability feature	Input of fields	Implement reading of initial field of iterative solver and incident field from file.
2	Simulation feature	Calculation of Cabs	Study differerent formulations for Cabs, especially for very large refractive index. Also check, how cross sections should be calculated in the FCD framework.
2	Optimization	Optimize scattered fields	Furhter optimize calculation of the scattered field, according to the ideas of R. Scott Brock; in parallel mode distribute internal fields evenly between processors (MPI_ALLTOALLV).
2	Documentation	Adding new beam	Add a description in the appendix how to add new incident field option in the code. Do the same for adding command line options.
2	Documentation	Short name: ADDA	Change all references in the documentation and code from "Amsterdam DDA" to "ADDA". Add explanation why this is done, and send separate note to users with the next release.
3	Usability feature	Automatic testing	Implement automatic testing of releases with a wide range of command line options. This should decrease the chance of introducing new bugs into old parts/features of the code during further development.
3	Usability feature	Amplitude matrix	Implement saving of amplitude matrix (vs scattering angle) to file (together with Mueller matrix).
3	Usability feature	Formulation option	Add command line option to choose the whole formulation (like FCD, SO, LDR, etc.)
3	Usability feature	Checkpoints	Furhter development of the checkpoint system. Extend to calculations of two polarizations and orientational averaging.
3	Simulation feature	Near field	Convenient way to calculate near field in a number of points or on a grid around the particle. Possibly implement calculation of the B-field together with E-field.
3	Simulation feature	Incompl. orient. averaging	Implement calculation of scattering for the 2D grid of scattering angles (and asymmetry parameter) in combination with orientation averaging. This can be useful when orientation is not completely random.
3	Simulation feature	Continuous polarizability	Implement continuous polarizability (not discrete values, like now), change -read format accordingly. Possibly implement full tensor of polarizability. This will open way to things like Weighted Discretization.
3	Simulation feature	Polarization prescription	Implement all possible polarization prescription, at least those that are easy to do (DGF, LAK, a1-term, etc.). Think about other existing possibilities for -scat and -int options.
3	Simulation feature	Radiation forces	Completely rewrite calculation of radiation forces using FFT. Add its memory requirements to -prognose estimates, and perform other integration into the code.
3	Simulation feature	Convex Hull	Implement convex hull algorithm into shape generator to enable much wider variety of automatically generated shapes.
3	Optimization	Intel compiler	Try Intel C compiler 10.1, change Makefile if needed.
3	Optimization	Timing for MatVec product	Add timing (in standard timing regime) for MatVec function.
3	Optimization	Planning shape	Implement optimizations based on the analysis of the shape (position of real dipoles on the grid) to decrease tests and indirect indexing in MatVec, CalcField, etc.
3	Optimization	Optimize MatVec	Optimize arithmetic operations in the MatVec function, especially the middle one (Arithm3).
3	Optimization	Optimize MPI	Several possible ways to optimize MPI part of the code: 1) use one buffer for all MPI communications; 2) possibly use MPI_ALLTOALL and 'derived datatypes' for block_transpose; 3) use 'persistent communication requests' for repeated communications.
3	Optimization	Rewrite jagged	Rewrite implementation of the -jagged option, so everything is done in the end of make_particle(). Then everything else will be automatically compatible with it. Also optimize the main cycle in make_particle.

Priority	Category	Name	Description
3	Optimization	Integration for Csca and g	Use weighting function for integration of scattered fields to obtain Csca and g. Take into account at least the diffraction peak.
3	Optimization	Determine void dipoles	Analyze dipoles with refractive index equal to unity and make them void in the beginning of the program. Also find out why it fails now, probably somewhere there is division over zero.
3	Optimization	New iterative solvers	Implement new iterative solvers, e.g. 2-term QMR and CSYM. Should improve performance for very slow convergence.
3	Optimization	Residual	Check calculation of residual inside the iterative solvers. Maybe it would be more accurate to recalculate it at the end.
3	Documentation	Citeable manual	Implement citation of the manual independently of the code, through e.g. arxiv.org.
3	Documentation	Features	Add description of features (capabilities) in the manual and on the web site.
3	Documentation	Euler angles	Add specific definition (with figures) of the used notation for Euler angles in the manual.
3	Documentation	Surrounding medium	Explain the transformation of the radiation forces due to the surrounding medium in the manual.
4	Usability feature	Single polarization	Implement simulation for single given incident polarization (complex amplitude). It will cut down simulation time twice for certain applications.
4	Usability feature	Binary shapes	Implement binary format for shape files, which can save space tremendously for homogeneous particles.
4	Usability feature	Shape information	Save more information about the shape, when using -save_geom.
4	Simulation feature	Revise 'box' shape	Possibly revise the definition of box in 'box' and 'spherebox' shapes.
4	Simulation feature	New shapes	Implement general shapes like 'coated ellipsoid', 'boxsphere', 'spherebox' with a lot of input parameters.
4	Simulation feature	Spectrum	Implement convenient simulation of spectrum, either inside of ADDA, or with a separate script. Possibly optimize calculation by simultaneous solving for many wavelength values in one iterative sequence (should be especially relevant in case of Rayleigh particles).
4	Simulation feature	Rayleigh limit	Implement special Rayleigh mode, which enables some simplifications (like using CG iterative solver and calculating only total polarizability tensor). Use additional symmetries of the particle (rotation around y and x axes by 90 degrees).
4	Simulation feature	SO formulation	Further develop SO formulation, bring it out of the development phase.
4	Optimization	Shared-memory	Optimize ADDA for shared memory hardware (e.g. multi-core processors), using standard MPI functions.
4	Optimization	FFTW upgrade	Implement certain parts of the code using MPI functions in FFTW 3.2, either the whole 3D FFT transforms, or local and MPI transpose operations.
4	Optimization	Sphere quadrature	Implement orientation averaging using quadrature over sphere surface, according to ideas of Antti Penttila.
4	Optimization	Memory in prognose	Ensure then when -prognose option is used, minimum memory is allocated in make_particle().
4	Optimization	Accumulate alpha	Optimize accumulate function (parallel communication) when averaging over alpha angle. Do all communication at once and do not use unnecessary memory (Eplane <=> ampl_alpha).
4	Optimization	Overflow of clock	Try to avoid overflow of 'clock' function when run in a sequential mode.
4	Optimization	One polarization per orient.	Reformulate orientational averaging so that only one incident polarization per orientation is simulated.
5	Usability feature	Incident angles	Implement specifying incident propagation direction by two angles istead of Cartesian coordinates of the vector of incidence.
5	Usability feature	Accurate memory counting	Make memory accounting more accurate, include non-major contributions and take into account dynamic allocation/deallocation.

Priority	Category	Name	Description
5	Simulation feature	Multi-grid	Implement multi-grid DDA, according to the ideas of Vitezslav Karasek, for simulation of clusters of particles, which are at considerable distances from each other.
5	Simulation feature	Van der Mee & Hovenier	Implement Van der Mee & Hoveniers test on the Mueller matrix. Probably makes real sense only for the results of orientational averaging.
5	Optimization	Reduced D matrix	Optimize storage of D matrix to use less memory (at least in sequential mode).
5	Optimization	Optimize scattered fields	Further optimize scattered field calculation for large grid of angles, trying to use FFT-based ideas to transforming 3D array of polarizations to a sphere surface in spectral space.
5	Optimization	Fields from smaller dpl	Use internal fields from smaller dpl as an input for the iterative solver of a simulation with larger dpl. The general methodology is described somewhere in Numerical Recipes.
5	Optimization	Optimize two polarizations	Possibly use the internal field obtained for one incident polarization to cut down the number of iterations for the second.
5	Optimization	Reuse orientation	Reuse calculated internal fields from one orientation for the similar ones to speed up orientation averaging.
5	Optimization	Ci,Si	Optimize calculation of sine and cosine integrals, e.g. implement code from gsl.
5	Optimization	Optimize Mueller matrix	Optimize calculation of the Mueller matrix from the amplitude matrix.
5	Optimization	ReducedFFT and CS	Study the combination of reduced FFT and Complex Symmetric iterative solvers. Possibly implement automatic detection of some non-viable combinations.
5	Optimization	Accelerate D matrix	Optimize calculation of D matrix, especially relevant when FCD or SO is used.
5	Optimization	WKB initial field	Use internal field from WKB approximation as an input for the iterative solver.
5	Optimization	MatVec and 'her' option	Check that 'her' works fine in combination with not_reduced FFT. Maybe in this case matrix need to be duplicated in parallel mode.
5	Optimization	Linear algebra	Take a look on LAPACK, may be it can accelerate linear operations on large arrays.