Discussion Topics for the Hackers & Developers Session at the LAMMPS User Workshop 2013

New physics features

- charge equilibration: alternate schemes (SplitQ, ACKS2), extrapolation for faster convergence
- linked rigid bonds for TraPPE force field (cf. p-LINCS from gromacs)
- dummy atoms (e.g. to support SHAKE with linear molecules, see gromacs)
- polarizable point dipoles (cf. Tangney/Scandolo JCP paper)
- generalized Born implicit solvent model (GBIS, GBSA)

Input/output, pre-/post-processing features

- HDF5 file support for restart and trajectory
- native readers for .psf (CHARMM/NAMD) or .parmtop (Amber) or .tpr (Gromacs) or …
- changes to the data file format to include additional and optional (e.g. style) information
- consolidate converter tools to interface to different codes (which ones?) into one package
- VMD GUI plugin frontend for running simple calculations (cf. NAMD GUI plugin in VMD)
- embedded movie maker (as extension to dump image), ray-tracer support

Improved efficiency and accuracy

- use Kokkos for SIMD/GPU/Threads
- structs for storage of parameters instead of individual arrays, especially for non-bonded
- explicit SIMD support via compiler vectorization including: SIMD friendly storage of x-,y-,z- data, SIMD friendly neighborlist (cf. paper by Páll / Hess in CPC), mixed precision SIMD force kernels, vectorizable inline math for potentials using exp()/log()/pow() (vdt++, fastermath)
- store coordinates with domain offsets for more accurate position data with large systems
- better load-balancing, e.g. bias the result of fix balance using feedback from timer class

Build system

- adopt cmake or automake; stop copying sources, but include via variables and vpath
- full support for building multiple targets including libraries from one source tree
- better support for developers using IDEs (eclipse, netbeans, visual c++, …)
- support dynamically loadable styles through a plugin system (incremental updates for binary distributions, executable only contains executable code that is used)

Documentation, training and outreach

- collect and edit available tutorials; adapt tutorials for general MD to use LAMMPS
- bundle the above with introductions to relevant theory and publish as a collaborative book
- improvements to developer's guide, inline documentation a la doxygen or sphinx
- coding style guide; recommendations for portability and efficiency
- organize developer training (e.g. combined with HPC programming, example ICTP)
- alternate ways of user support, especially help for beginners in MD and LAMMPS

Project management and software engineering

- how to make it more attractive for people to contribute to LAMMPS
- how to reduce the growing redundancy and keep the project maintainable
- automated regression/unit testing; use bug reporting / issue tracking tool
- organize developer hack-a-thon (aka "sprint meeting") to work on topics of general concern or implement features requiring significant work (= multiple people)