

MG/ME v5 ?

Do we really need it ?

- Pros:
 - More and more peripheral tools, less and less space to implement them
 - Drastic optimizations may be possible (conserved quantities/flow for MG, lessons from Gridpack for ME)
 - Real move to parallel computing (no central disk, grid, ...)
 - Get rid of the last SM oriented stuffs (generic HELAS routines, ...)
 - Code is now hard to maintain, need experts for even small modifications, dangerous in the future (leaving people, ...)
 - Rationalization and simplification
 - GNU
 - P.H.Y.S.I.C.S: make a long list of ambitious things here!
 - “if you don’t move, you loose” (but we are not Apple neither...)
- Cons:
 - T.I.M.E: but it may become a huge Pro once done regarding the current situation
 - risk to loose confidence
 - Too “polished” image, ie loose confidence because too much time invested in coding and not enough in physics
 - ...

What do we really need ?

- OPEN interface (many way to interpret this)
- Overall speed improvement by re-thinking crucial routines but also by centralizing and optimizing administrative routines
- Get rid of a long list of annoying things
- Listen to users, especially the new ones!

Milestones from v4 to v5

- Check MadWeight and MadOnia are now well merged (v4.4)
- “Translate” v4 to gfortran (not yet optimization, need to compile w/o warnings but not necessarily with -Wall !)
 1. MadGraph: only ASSIGN statements still to fix
 2. HELAS
 3. MadEvent
 4. Tests (!!!) + speed comparison
 5. External tools: SM and MSSM calculators, DECAY, MadAnalysis
- Develop new “peripheral” tools in Python (interest: learn the language, build confidence, speed tests, build modules of standard routines and associated “test suites”)
 - USRMOD2 (work in progress, finish next week ?)
 - A new tool somewhere between MadAnalysis and Root to work interactively with LHE files? (“hobby” project)
 - ... ?

- Optimization + “cleaning” + rewriting of existing of ME and HELAS gfortran version (focusing only on actual computation routines, not I/O). Not necessary for MG ? (full switch to Python expected, time not “so” important for diagram generation)
- Starting from wanted new features, define structure (object, module, ...) of MG and ME.
- Two options:
 - step-by-step: first Python version of MG producing MEv4 compatible files, make a release, and only then modify ME. Advantage: easy to check -> confidence. Disadvantage: takes more time, current structure will affect new one
 - everything at the same time: allows to define a really new structure, but will require a lot of coding before real life tests (test suites may help a lot here)
 - at this time, only command line anyway (a la Root interface ?)!
- Adapt/rewrite web interface