Precision physic at colliders: introducing reSolve

Francesco Coradeschi*
(Department of Applied Mathematics and Theoretical Physics
University of Cambridge)

IPPP Durham - April 21, 2018

* In collaboration with T. Cridge (DAMTP)

What is reSolve ?

- reSolve is a MC code which aims to add analytic transverse momentum resummation to generic processes
- Generic doesn't mean quite any – limited by the formalism: means (for now at least) no colour in the final state
- Work in progress, however first trial version already available: → https://github.com/fkhorad/reSolve (many intended features still missing[1])

[1] Still young (< 1 year) & small (2 of us)!

What is reSolve ?

- reSolve is a spiritual successor to the family of codes DYqT, DYres, HqT, Hres, 2gres
- It shares (almost) no lines of code with its predecessors: completely reimagined and reimplemented in C++
- Two updates already in the pipeline: one in the next few days + another on a "soonish" timescale

What is reSolve ?

- Underlying philosophy: modular, transparent, easily customisable code
- Set it up in way which allows (relatively) easy interfacing with other codes
- Build it with parallelisation already in mind (more on this later)

On q_T resummation

- . . . surely does not need a lengthy introduction in this context
- Long story short, logarithmic contributions of the form

$$\sim \log(q_T^2/M^2)^n \alpha_s^m$$

  appear to all orders in perturbation theory for $q_T \ll M$.
- They are associated with multiple, (relatively) soft radiation emission and have to be included to get a realistic description of the q_T spectrum of F

On q_T resummation

- A standard way to take this into account is through Parton Showers. reSolve uses instead an analytic approach, based on [Catani,Cieri,De Florian,Ferrera,Grazzini('14)]
- History of the formalism is long – dates back to [Dokshitzer,Diakonov,Troian('78)] , [Parisi,Petronzio('79)] , [Kodaira,Trentadue('82)] , [Collins,Soper,Sterman('85)] – much work happened between now and then
- I'll jump straight to the final master formula and highlight some of its characteristics, relevant to describe what reSolve can potentially do

q_T resummation: Master Formula

- As already specified, F is a generic non-coloured final state

$$\frac{d\sigma^{F(res)}(s,q_T,M,y,\Omega)}{d^2q_T dM^2 dy d\Omega} = \frac{M^2}{s}\int \frac{d^2b}{(2\pi)^2} e^{ib\cdot q_T} S_c(M,b) \int_{x_1}^1 \frac{dz_1}{z_1}\int_{x_2}^1 \frac{dz_2}{z_2} d\tilde{\sigma}_{c\bar{c}}^{F;h_1h_2\lambda_1\lambda_2}$$
$$\cdot C_{ca_1}^{h_1\lambda_1}(z_1,\alpha_s(b_0^2/b^2)) C_{\bar{c}a_2}^{h_2\lambda_2}(z_2,\alpha_s(b_0^2/b^2)) f_{a_1/h_1}(x_1/z_1,b_0^2/b^2) f_{a_2/h_2}(x_2/z_2,b_0^2/b^2)$$

with

$$S_c = \exp\left[-\int_{\frac{b_0^2}{b^2}}^{M^2}\frac{dq^2}{q^2}\left(A_c(\alpha_s(q^2))\log(M^2/q^2)+B_c(\alpha_s(q^2))\right)\right]$$

$$d\tilde{\sigma}_{c\bar{c}}^{F;h_1h_2\lambda_1\lambda_2} = d\hat{\sigma}_{c\bar{c}}^{F(0)} H_c^{F;h_1h_2\lambda_1\lambda_2}; \qquad x_{1,2} = \frac{M}{\sqrt{\hat{s}}}e^{\pm y}$$

$$K_{c,\dots}(\alpha_s) = \sum_n \left(\frac{\alpha_s}{\pi}\right)^n K_{c,\dots}^{(n)}$$

q_T resummation: Master Formula

- As already specified, F is a generic non-coloured final state

$$\frac{d\sigma^{F(res)}(s,q_T,M,y,\Omega)}{d^2q_T dM^2 dy d\Omega} = \frac{M^2}{s}\int \frac{d^2b}{(2\pi)^2} e^{ib\cdot q_T} S_c(M,b) \int_{x_1}^1 \frac{dz_1}{z_1}\int_{x_2}^1 \frac{dz_2}{z_2} d\tilde{\sigma}_{c\bar{c}}^{F;h_1h_2\lambda_1\lambda_2}$$
$$\cdot C_{ca_1}^{h_1\lambda_1}(z_1,\alpha_s(b_0^2/b^2)) C_{\bar{c}a_2}^{h_2\lambda_2}(z_2,\alpha_s(b_0^2/b^2)) f_{a_1/h_1}(x_1/z_1,b_0^2/b^2) f_{a_2/h_2}(x_2/z_2,b_0^2/b^2)$$

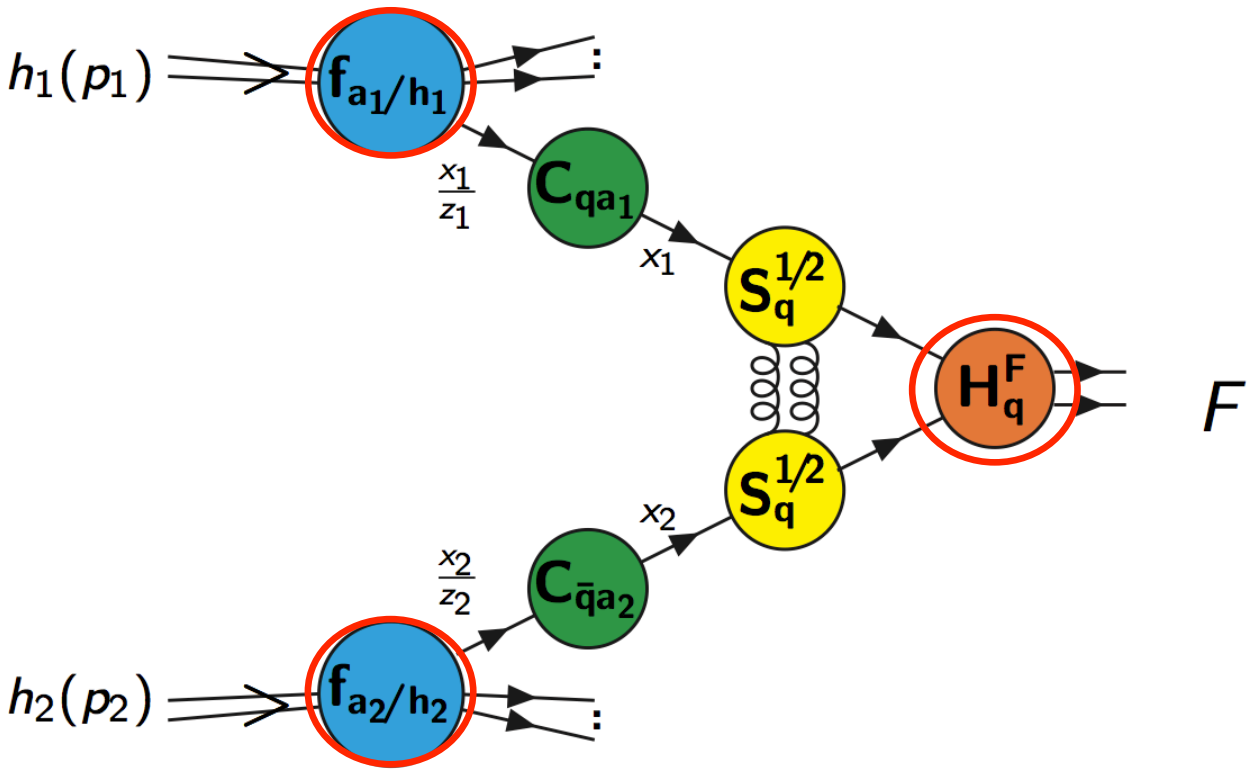
with

$$S_c = \exp\left[-\int_{\frac{b_0^2}{b^2}}^{M^2}\frac{dq^2}{q^2}\left(A_c(\alpha_s(q^2))\log(M^2/q^2)+B_c(\alpha_s(q^2))\right)\right]$$

$$d\tilde{\sigma}_{c\bar{c}}^{F;h_1h_2\lambda_1\lambda_2} = d\hat{\sigma}_{c\bar{c}}^{F(0)} H_c^{F;h_1h_2\lambda_1\lambda_2}; \qquad x_{1,2} = \frac{M}{\sqrt{\hat{s}}}e^{\pm y}$$

$$K_{c,\dots}(\alpha_s) = \sum_n \left(\frac{\alpha_s}{\pi}\right)^n K_{c,\dots}^{(n)}$$

q_T resummation: Master Formula



q_T resummation: Master Formula

- As highlighted, most factors in master formula are universal – which really begs for a general numerical implementation
- Exactly as standard factorisation, the formula is inclusive in unresolved radiation, but fully differential in all F internal variables
- The formula is a small q_T approximation: it formally holds up to $\mathcal{O}(q_T^2/M^2)$ corrections. A matching to fixed order can be defined in a general way in order to handle the full q_T spectrum ($+$ any other observable) in a single formalism.

q_T: matching and subtraction

- The matching strategy – which I won't describe in detail – essantially involves using a fixed-order expansion of the master formula as a counterterm to avoid double counting
- Nontrivial additional benefit of the matching is automatically defining a NNLO subtraction mechansim. This is nonlocal (and obviously only works for final states without colour), but is immediate to implement once you have resummation

back to reSolve – usage

- Installation as simple as typing "make" (only needs reasonably recent gcc and gfortran)
- Currently needs the legacy minuit library (sources+makefile provided with the code) for PDF fit
- Optionally uses the CUBA[Hahn('04)] library for integration – if desired, has to be downloaded and installed separately
- Main code is C++11-standard compliant – in principle fully portable (tested on several linux distros + MacOsX)

resolve – usage

- Input read from a simple flexibly-styled, customisable input card – most settings are rather obvious (couplings, masses, cuts, # of events. . .), all are documented
- Straightforward to add more parameters as needed
- Straightforward to add more processes (has to be done by hand, though – some templates are provided)

resolve – usage

- Basic output: of course the total cross-section, and distributions as desired (specify on input file)
- Main output are however LHA-style events, which can then be analysed with the user's favourite tool(s) – a simple built-in histogrammer tool is also provided

Code structure

- Code is organised in component modules, which are as far as possible independent
  - Main: read input, select process, call integrations
  - Histogrammer
  - Integral: integration routines
  - PDFfit: currently needed for resummation, kept separate
  - Process: one subfolder per final state
    - Diphoton: SM background only (no Higgs)
    - Drell-Yan
  - Resummation: the main module
  - Utility: some basic general-purpose routines

On universality

- The most important and "unique" part of the code, the resummation module makes full use of the universality of the formula
- It does not need to know any detail about the underlying process
- Can be compiled and used independently
- In principle, this part of the code to be interfaced with any matrix element provider and integrator/event generator

On parallelisation

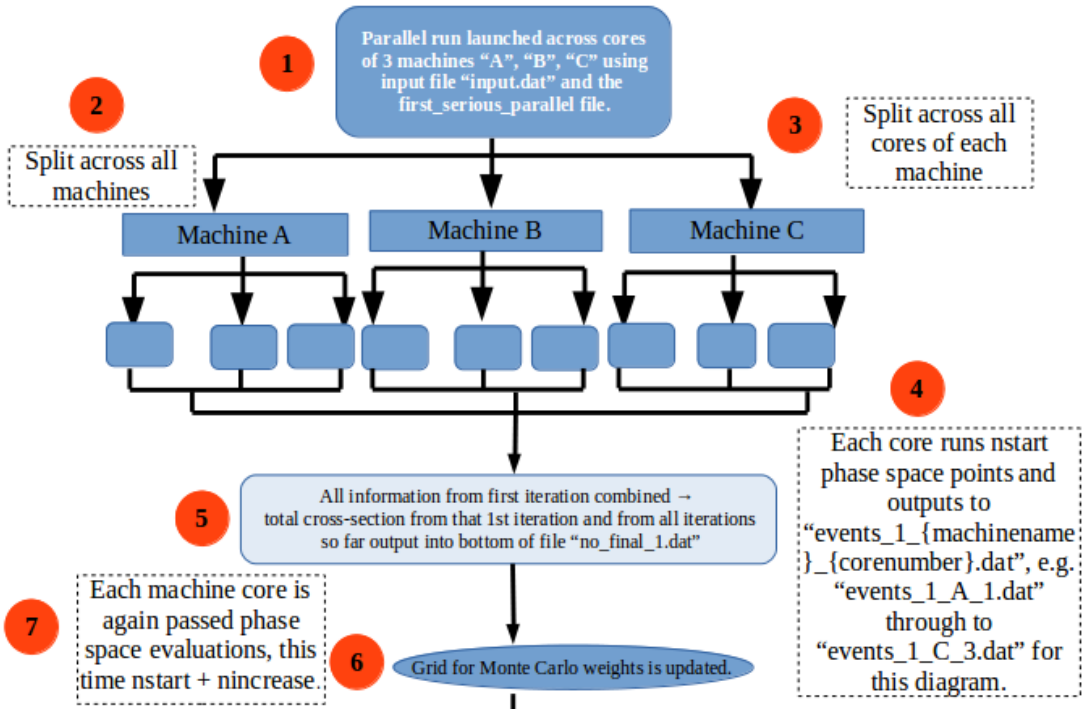
- I stated the code is set-up for parallelisation – what do I mean?
- ME are "agnostic" w.r.t. to the integrator: they only use clearly specified and local variables
- The code by default contains CUBA, which attempts to automatically parallelise on all available cores on the local machine (tested on Desktops with 4-8 cores)
- The code also has a dedicated Vegas implementation dubbed k_vegas – can also be used as a stand-alone integrator

On parallelisation

- k_vegas is just Vegas, but is set-up in order to be able to do partial iterations, save partial results for the integration grid, and recombine them at a later stage
- This allows for parallelisation on different nodes ($\rightarrow$ machines). An example simple bash script provided with the code uses this to parallelise a run over an arbitrary number of computers which are reachable from the central node via ssh.
- Of course, much room for improvement (but not so easy if one wants to stay system-independent)

On parallelisation

- Schema of parallelisation using k_vegas + script

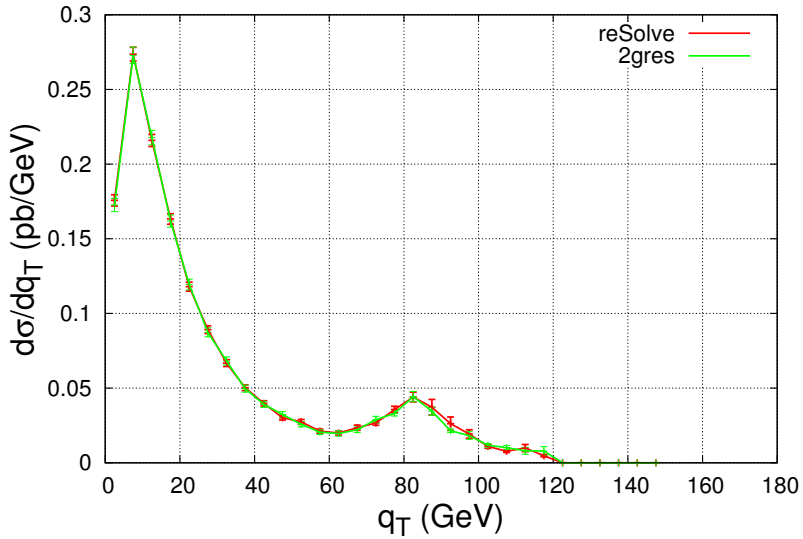


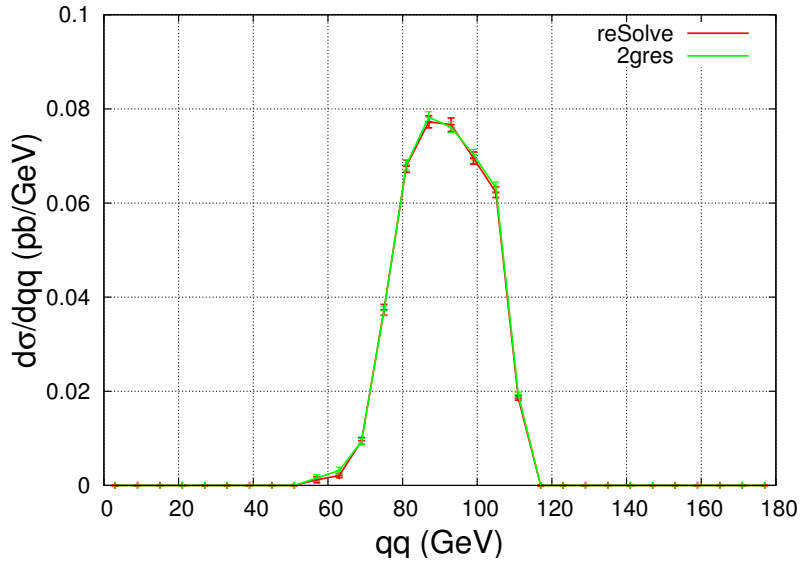
Current state & upgrades

- What the code does now: diphoton (SM background), no matching. It is however, the first public incarnation of 2gres (used for predictions in [Cieri,FC,De Florian('15)])
- What will definitely do very soon ($\sim$ days): Drell-Yan, easier (still non-automated) implementation of new processes (+ improvements not obvious from a user p.o.v.)
- What it will do "soonish": Higgs production, Higgs resonance-SM background interference in diphoton channel, matching, QED resummation

Some validation plots

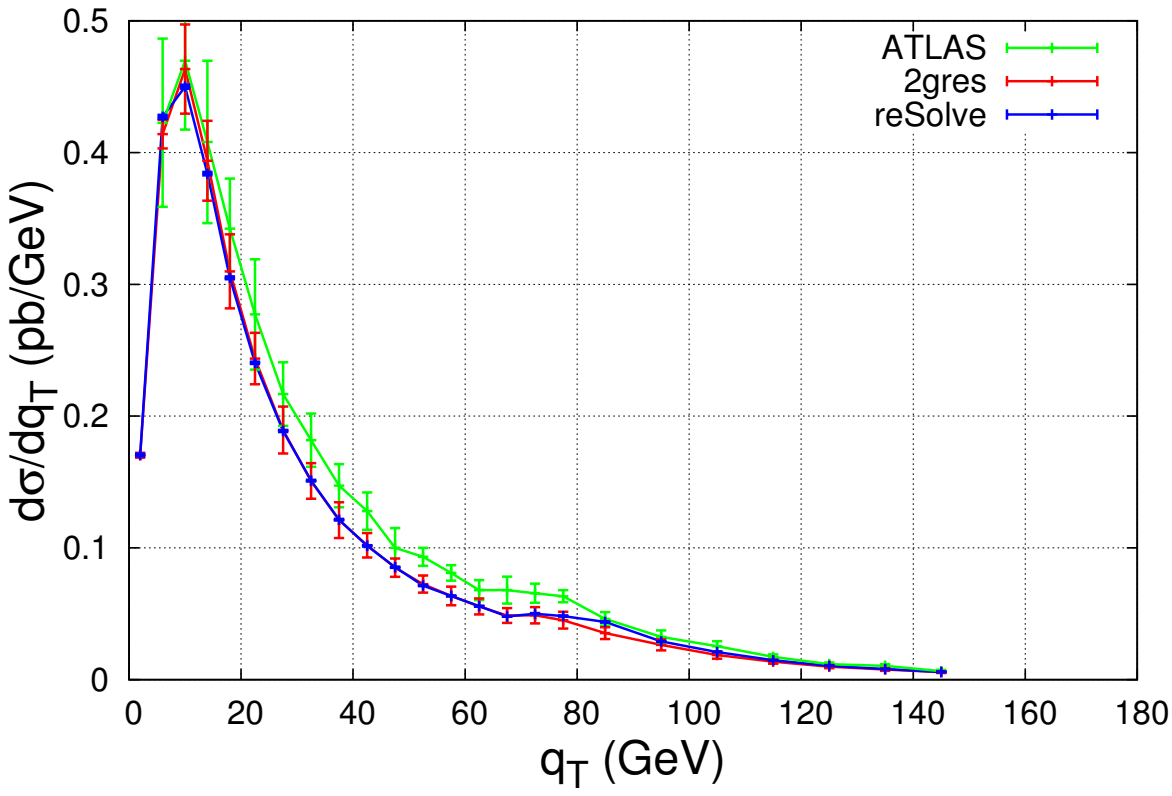
- Sample comparison with 2gres





Some validation plots

- Comparison with data (includes matching, done using 2gNNLO for the fixed-order part)



"The future" – potential directions

(essentially a personal wishlist)

- OBVIOUSLY: add more complete NNLL (or even N^3LL) processes (+ their f.o. for matching)
- Interface with existing ME generators: would allow automatic NLL+NLO, even for BSM – usual F class only, of course (easy)
- Interface in reverse order: produce a plug-in version to be used inside other MC codes

"The future" – potential directions

- Interface with existing NLO ME generators for semi-automatic NNLL+NNLO (not-so-easy)
- Extension to (at least some) Final States with colour (speculative)
- Improvements on qT subtraction (speculative)

Conclusions

- Precision calculations for processes at hadron colliders are one of the important challenges which lie ahead the theoretical particle physics community
- We must have the right tools to face the challenge: I hope `reSolve` will be a nice addition to the phenomenologist's (and possibily experimentalists's) toolbox