SPLINT integration with kinematic cut

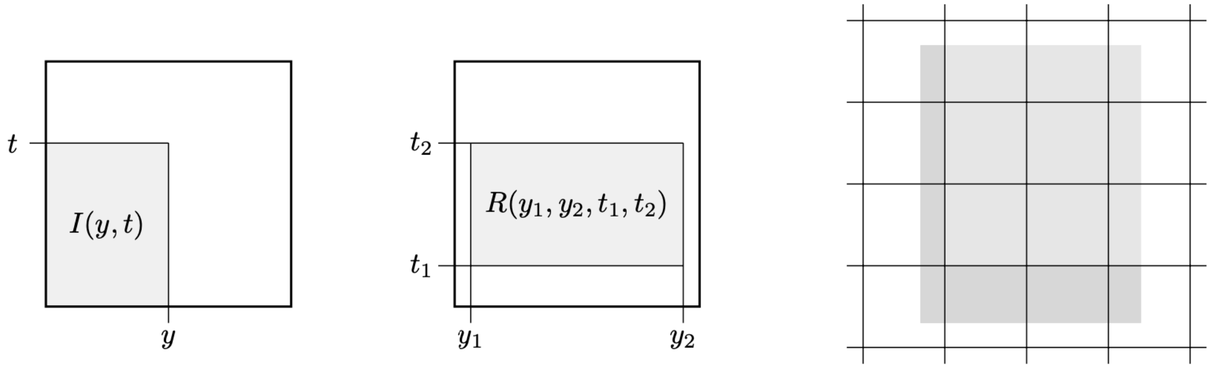
Michiel Botje

BAT meeting 20-09-2021

SPLINT 2-dimensional integration

- Spline in $y = -\log x$ and $t = \log \mu^2$ introduces Jacobian $e^{-y} e^{t}$ in the integral

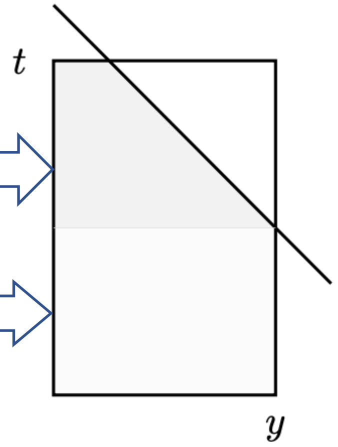
- Splint has very fast routines to integrate over rectangles



- Can now also handle kinematic limit

SPLINT integration over y
n-point Gauss quadrature over t

SPLINT rectangle integration



Quite some mathematical detail: see Appendix A of the SPLINT write-up

New version of SPLINT

- The 2-dim filling routines have a $\sqrt{s}$ cut `rsc` that is a filling cut
- Need to enter the proper value rs of $\sqrt{s}$ in the 2-dimensional integration routine

```
val   = dsp_IntS2(ia, x1, x2, q1, q2, rs, np)
```

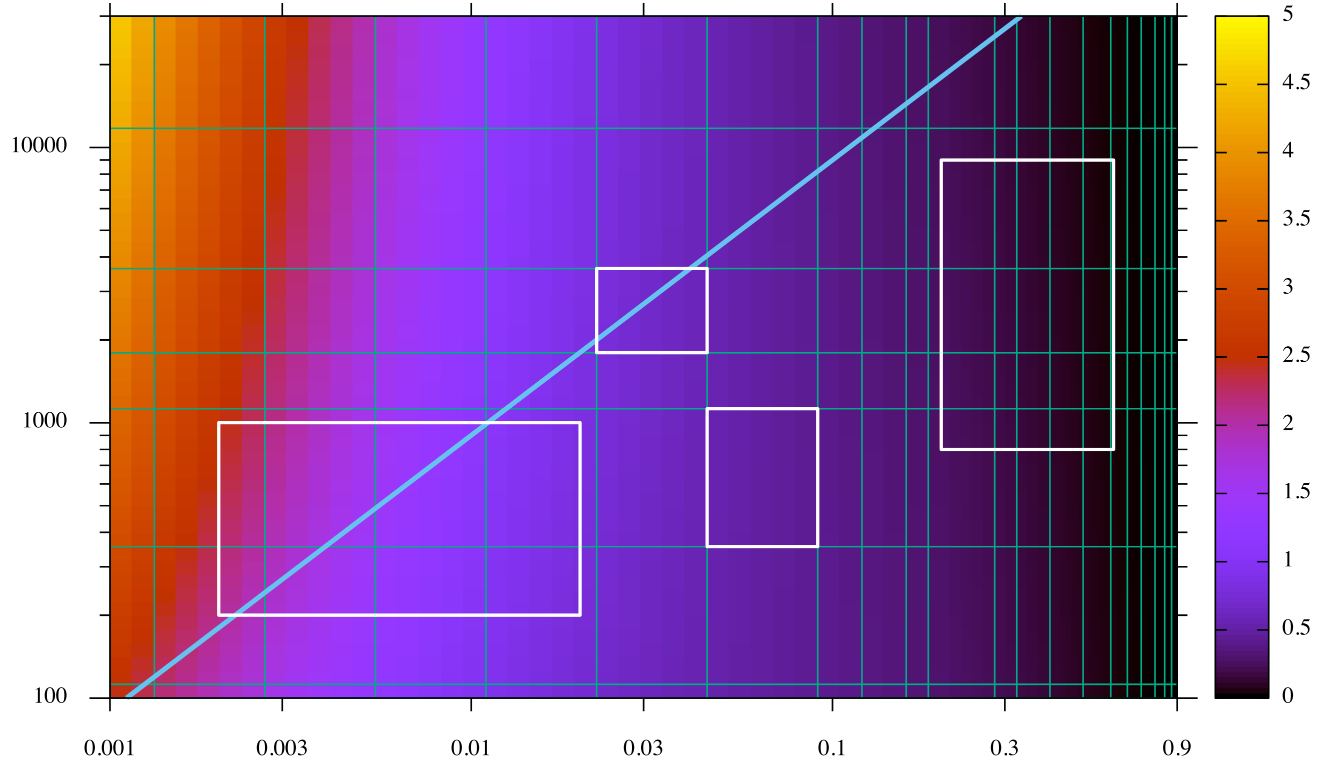
- The argument `np` = 2, 3, 4 sets the number of Gauss quadrature points
- For `np` > 4 the MBUTIL/CERNLIB adaptive n-point routine `dmb_gauss` is used
- For `np` < 2 the kinematic limit is ignored in node-bins crossed by the cut
- Another change in the new SPLINT version is

```
rsmax = dsp_RsMax(ia) → dsp_RsMax(ia, rsc)
```

- New version is included in QCDNUM-17-01-83
- Remark: SPLINT disk files will be obsolete so you must write these again

Check of the 2-dimensional integration

- Extend a 1-dim n-point Gauss quadrature routine to 2-dim and compare to SPLINT
- Integrate F_2 over single node-bins or arbitrary rectangles, with and without crossing the kinematic limit
- Relative difference $\lesssim\ 10^{-9}$ in all cases
- SPLINT is a factor 300 faster than Gauss



Integrate the cross-section

- Integration of 429 cross-section bins takes 0.8 msec
- Of the 429 bins 28 cross the kinematic limit of $\sqrt{s} = 300$ GeV
- Integrate these bins with 2,3,4-point quadrature and compare to n-point Gauss quadrature
- 2-point Gauss is still very good but not a real CPU time saver
- Best is to set np = 5 and use the adaptive n-point Gauss routine

np	$\Delta I/I$
2	4×10^{-7}
3	8×10^{-8}
4	7×10^{-10}

Here is the complete timing summary →

	n_x	n_q	t [ms]
Evolution	100	50	3.6
6 Stf splines	22	7	2.9
Xsec spline	100	25	2.2
Integration			0.8

Simple exercise: how much momentum sum do we see?

- Compute $$\frac{1}{\mu_2^2 - \mu_1^2} \int_{\mu_1^2}^{\mu_2^2} \int_{x_1}^{x_2} (xg + x\Sigma)\, \mathrm{d}x \mathrm{d}\mu^2$$

- Make spline and integrate over the full x - μ^2 range

- The answer is 0.89

We’re done!