Introduction to Parton Showers

Stefan Höche

SLAC National Accelerator Laboratory

School and Workshop on pQCD @ West Lake

Hangzhou, 03/27/2018

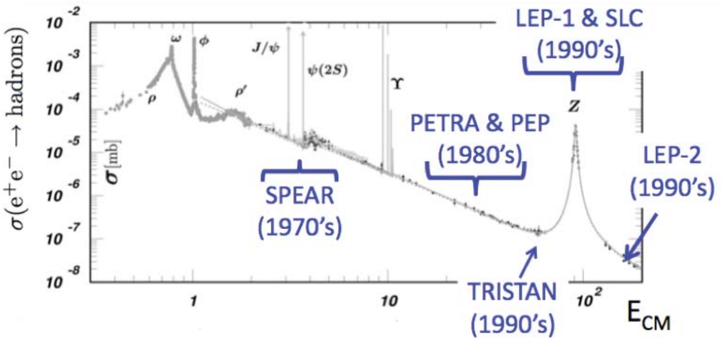
Suggested reading

- R. K. Ellis, W. J. Stirling, B. R. Webber
  QCD and Collider Physics
  Cambridge University Press, 2003
- R. D. Field
  Applications of Perturbative QCD
  Addison-Wesley, 1995
- T. Sjöstrand, S. Mrenna, P. Z. Skands
  PYTHIA 6.4 Physics and Manual
  JHEP 05 (2006) 026
- L. Dixon, F. Petriello (Editors)
  Journeys Through the Precision Frontier
  Proceedings of TASI 2014, World Scientific, 2015

Outline of lectures

- Introduction
  - Historical context
  - Collider observables
  - Event generators
- Parton showers
  - Leading-order formalism
  - Assessment of formal precision
- Combination with fixed-order calculations
  - Matching to NLO calculations
  - LO-Merging of multiplicities
  - Combination of matched results

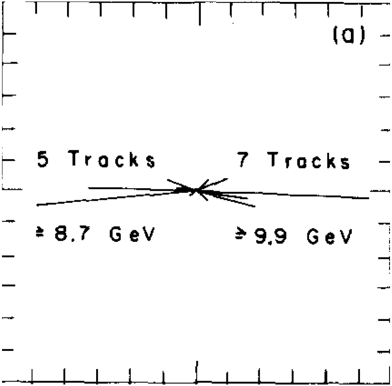
QCD in e^+e^- annihilation

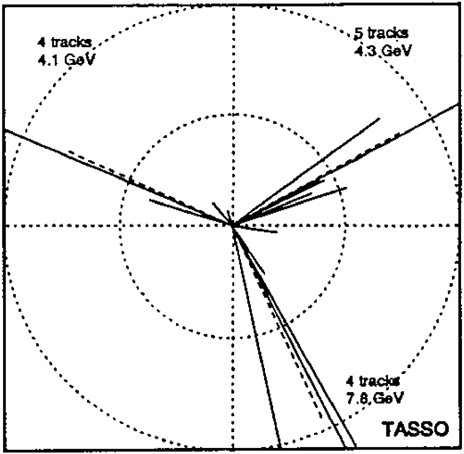


- SPEAR (SLAC): Discovery of quark jets
- PETRA (DESY) & PEP (SLAC): First high energy (>10 GeV) jets
  Discovery of gluon jets (PETRA) & pioneering QCD studies
- LEP (CERN) & SLC (SLAC): Large energies → more reliable
  QCD calculations, smaller hadronization uncertainties
  Large data samples → precision tests of QCD

Discovery of the gluon

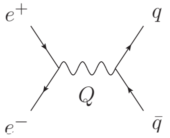
[TASSO] PLB86(1979)243 & Proc. Neutrino '79, Vol.1, p.113



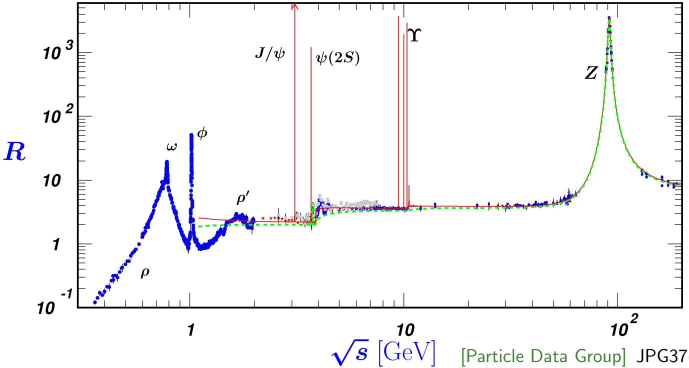


- Gluon discovery at the PETRA collider at DESY
- Typical three-jet event (right) vs. two-jet event (left)

Basic process for $e^+e^- \to$hadrons



- Prediction for $e^+e^- \to q\bar{q}$ at leading perturbative order differs from $e^+e^- \to \mu^+\mu^-$ only by quark charges
- Define ratio $R = \dfrac{\sigma_{e^+e^-\to\text{hadrons}}}{\sigma_{e^+e^-\to\mu^+\mu^-}} \quad \xrightarrow{\text{LO}} \quad \sum_i e_{q,i}^2$



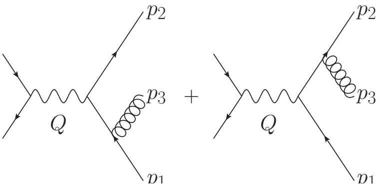
[Particle Data Group] JPG37(2010)075021

- Kinematic variables $x_i = \frac{2p_i \cdot Q}{Q^2}$

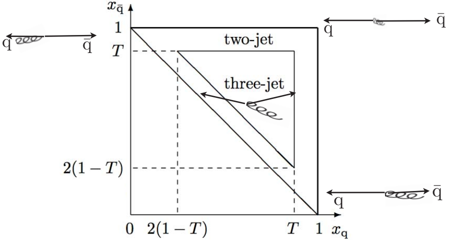
  $\to x_i < 1, \quad x_1 + x_2 + x_3 = 2$

- Differential cross section

$$\frac{\mathrm{d}^2\sigma}{\mathrm{d}x_1\mathrm{d}x_2} = \sigma_0 \frac{\alpha_s}{\pi} C_F \frac{x_1^2 + x_2^2}{(1-x_1)(1-x_2)}$$

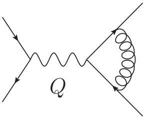


- Divergent as
  - $x_1 \to 1$ $(p_3 \parallel p_1)$
  - $x_2 \to 1$ $(p_3 \parallel p_2)$
  - $(x_1, x_2) \to (1, 1)$ $(x_3 \to 0)$



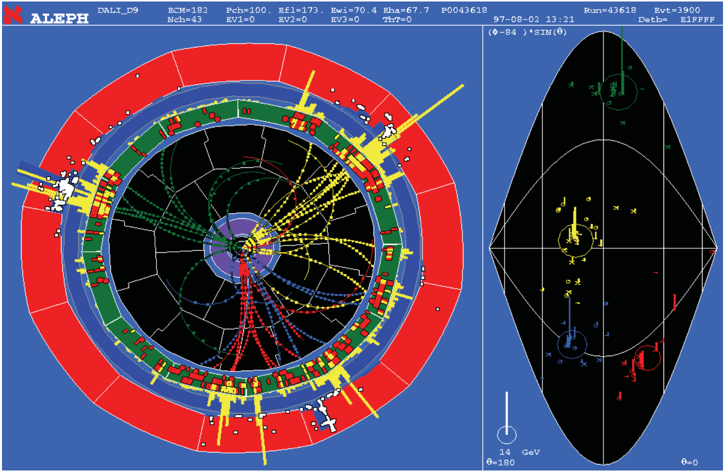
- Divergences canceled by virtual correction
  Total correction to $e^+e^- \to$ hadrons:

$$\sigma^{\mathrm{NLO}} = \sigma_0 \left(1 + \frac{3}{4} C_F \frac{\alpha_s}{\pi}\right)$$



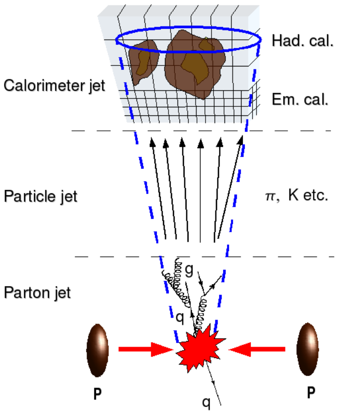
High-energy colliders and jets

[ALEPH]



Jet algorithms

SLAC



- Identify hadronic activity in experiment with partonic activity in pQCD theory

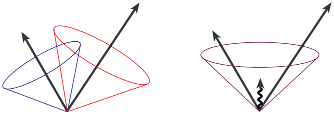
$\Rightarrow$ Requirements

- Applicable both to data and theory
  - partons
  - stable particles
  - measured objects (calorimeter objects, tracks, etc.)
- Gives close relationship between jets constructed from any of the above
- Independent of the details of the detector, e.g. calorimeter granularity

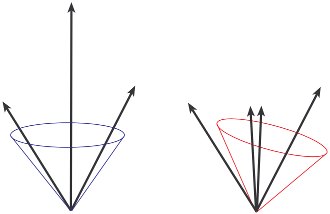
Jet algorithms

Further requirements from QCD

- Infrared safety → no change when adding a soft particle

Counterexample: 

- Collinear safety → no change when substituting particle
  with two collinear particles

Counterexample: 

Jet algorithms

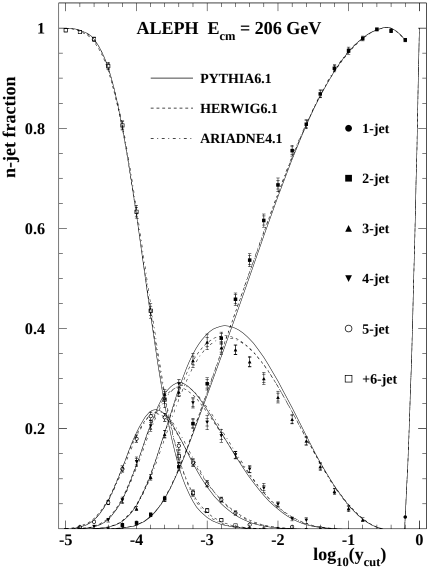
SLAC

- Most widely used jet algorithms today of sequential recombination type
- Example: Durham algorithm
  1. Start with a list of preclusters
  2. For each pair of preclusters calculate

$$y_{ij} = \frac{2}{E_{cm}^2} \min\left\{E_i^2, E_j^2\right\} (1 - \cos\theta_{ij}) \approx \frac{k_T^2}{E_{cm}^2}$$

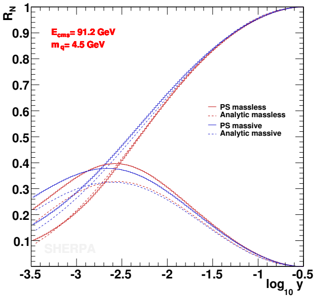
  3. Identify $y_{kl} = \min\{\, y_{ij} \,\}$
  4. If $y_{kl} < y_{\rm cut}$, define all preclusters as jets and stop
     else merge preclusters k and l and continue at step 1
- Ambiguities:
  - Distance measure y_{ij} (e.g. Jade algorithm $y_{ij} \to 2p_ip_j/E_{cm}^2$)
  - Recombination scheme (e.g. four-momentum addition $p_{kl} = p_k + p_l$)
  - Resolution criterion $y_{\rm cut}$
- For hadron collider algorithms, see [Salam] arXiv:0906.1833

Jets in $e^+e^- \rightarrow$hadrons



[ALEPH] CERN-EP-2003-084

- Can compute n-jet rate in coherent branching formalism
  [Catani,Olsson,Turnock,Webber] PLB269(1991)432
- Alternatively simulate with MC event generators



Event shape variables

- Shape variables characterize event as a whole
- Thrust (introduced 1978 at PETRA)

  $$T = \max_{\vec{n}} \frac{\sum_i |\vec{p}_i \cdot \vec{n}|}{\sum_j |\vec{p}_j|}$$

  - $T \to 1$ – back-to-back event
  - $T \to 1/2$ – spherically symmetric event

  Vector for which maximum is obtained $\to$ thrust axis $\vec{n}_T$

- Jet broadening

  $$B_i = \frac{\sum_{k \in H_i} |\vec{p}_k \times \vec{n}_T|}{2 \sum_j |\vec{p}_j|}$$

  Computed for two hemispheres w.r.t. $\vec{n}_T$, then

  - $B_W = \max(B_1, B_2)$ – Wide jet broadening
  - $B_N = \min(B_1, B_2)$ – Narrow jet broadening

- C-Parameter

  Linearized momentum tensor

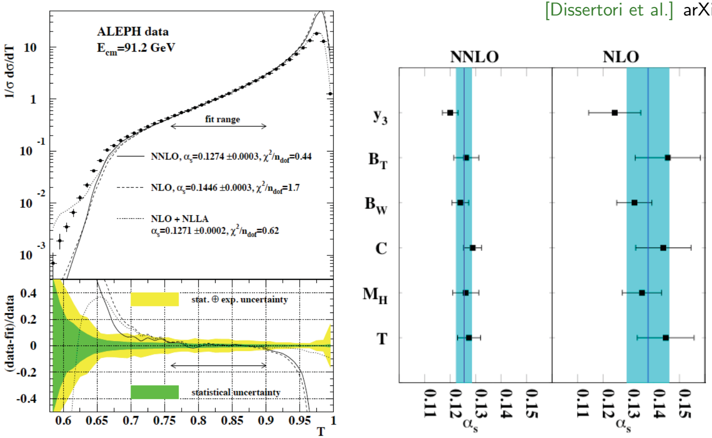
  $$\Theta^{\alpha\beta} = \frac{1}{\sum_j |\vec{p}_j|} \sum_i \frac{p_i^\alpha p_i^\beta}{|\vec{p}_i|} ,$$

  Eigenvalues λ_i define $C = 3(\lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_3\lambda_1)$

Application of event shape variables

- Discovery of quark and gluon jets – Sphericity & Oblateness
- Measurement of strong coupling constant – T, C, B, M_H, jet rates

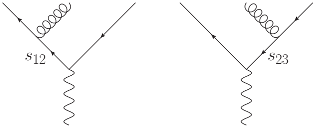
[Dissertori et al.] arXiv:0906.3436



Parton evolution

- Consider $e^+e^- \to 3$ partons

$$\frac{1}{\sigma_{2\to 2}}\frac{\mathrm{d}\sigma_{2\to 3}}{\mathrm{d}\cos\theta\mathrm{d}z} \sim C_F\frac{\alpha_s}{2\pi}\frac{2}{\sin^2\theta}\frac{1+(1-z)^2}{z}$$



θ - angle of gluon emission
z - fractional energy of gluon

- Divergent in
  - Collinear limit: $\theta \to 0, \pi$
  - Soft limit: $z \to 0$
- Separate into two independent jets

$$\frac{2\mathrm{d}\cos\theta}{\sin^2\theta} = \frac{\mathrm{d}\cos\theta}{1-\cos\theta} + \frac{\mathrm{d}\cos\theta}{1+\cos\theta} = \frac{\mathrm{d}\cos\theta}{1-\cos\theta} + \frac{\mathrm{d}\cos\bar\theta}{1-\cos\bar\theta} \approx \frac{\mathrm{d}\theta^2}{\theta^2} + \frac{\mathrm{d}\bar\theta^2}{\bar\theta^2}$$

- Independent evolution with θ

$$\mathrm{d}\sigma_3 \sim \sigma_2 \sum_{\text{jets}} C_F \frac{\alpha_s}{2\pi}\frac{\mathrm{d}\theta^2}{\theta^2}\mathrm{d}z\,\frac{1+(1-z)^2}{z}$$

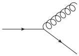
Parton evolution

SLAC

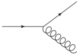
- Same equation for any variable with same limiting behavior
  - Transverse momentum $\quad k_T^2 = z^2(1-z)^2\theta^2 E^2$
  - Virtuality $\quad t = z(1-z)\theta^2 E^2$
- Call this the "evolution variable"

$$\frac{\mathrm{d}\theta^2}{\theta^2} = \frac{\mathrm{d}k_T^2}{k_T^2} = \frac{\mathrm{d}t}{t} \qquad \leftrightarrow \qquad \text{collinear divergence}$$

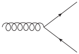
- Absorb z-dependence into flavor-dependent splitting kernel $P_{ab}(z)$



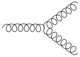
$$= C_F \frac{1+z^2}{1-z}$$



$$= C_F \frac{1+(1-z)^2}{z}$$



$$= T_R \left[z^2 + (1-z)^2 \right]$$



$$= C_A \left[\frac{z}{1-z} + \frac{1-z}{z} + z(1-z) \right]$$

- Branching equation emerges, but so far only pQCD, no hadrons

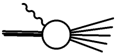
$$\mathrm{d}\sigma_{n+1} \sim \sigma_n \sum_{\text{jets}} \frac{\mathrm{d}t}{t} \mathrm{d}z \, \frac{\alpha_s}{2\pi} \, \frac{1}{2} P_{ab}(z)$$

The DGLAP equation

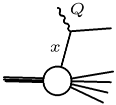
[Altarelli,Parisi] NPB126(1977)298

- Hadronic cross section factorizes into perturbative & non-perturbative piece

$$\sigma = \sum_{a=q,g} \int \mathrm{d}x\, f_a(x,\mu_F^2)\hat{\sigma}_a(\mu_F^2)$$

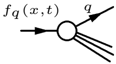


$= \sum_a$

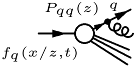


- Evolution from previous slide turns into evolution equation for $f_a(x,\mu_F^2)$
- $f_a(x,\mu_F^2)$ cannot be predicted as a function of x, but dependence on μ_F^2 can be computed order by order in pQCD due to invariance of σ under change of μ_F
- DGLAP equation $\leftrightarrow$ renormalization group equation

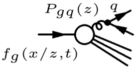
$\frac{\mathrm{d}}{\mathrm{d}\log(t/\mu^2)}$



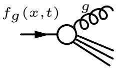
$= \int_x^1 \frac{\mathrm{d}z}{z}\frac{\alpha_s}{2\pi}$



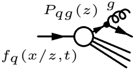
$+ \int_x^1 \frac{\mathrm{d}z}{z}\frac{\alpha_s}{2\pi}$



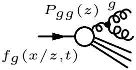
$\frac{\mathrm{d}}{\mathrm{d}\log(t/\mu^2)}$



$= \sum_{i=1}^{2n_f}\int_x^1 \frac{\mathrm{d}z}{z}\frac{\alpha_s}{2\pi}$



$+ \int_x^1 \frac{\mathrm{d}z}{z}\frac{\alpha_s}{2\pi}$



Properties of DGLAP kernels

- At leading order, splitting functions are probability densities
  They obey a special symmetry relation ($\varepsilon > 0$)

$$\sum_{b=q,g} \int_0^{1-\varepsilon} \mathrm{d}\zeta\, \zeta\, P_{qb}(\zeta) = \int_\varepsilon^{1-\varepsilon} \mathrm{d}\zeta\, P_{qq}(\zeta) + \mathcal{O}(\varepsilon)$$

$$\sum_{b=q,g} \int_0^{1-\varepsilon} \mathrm{d}\zeta\, \zeta\, P_{gb}(\zeta) = \int_\varepsilon^{1-\varepsilon} \mathrm{d}\zeta \left[\frac{1}{2} P_{gg}(\zeta) + n_f\, P_{gq}(\zeta) \right] + \mathcal{O}(\varepsilon)$$

  Can thus replace $1/2 \to z$ in branching equations
- Physical sum rules must hold at any order

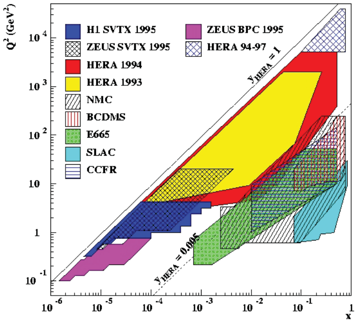
$$\int_0^1 \mathrm{d}\zeta\, \hat{P}_{qq}(\zeta) = 0 \qquad \rightarrow \qquad \text{flavor sum rule}$$

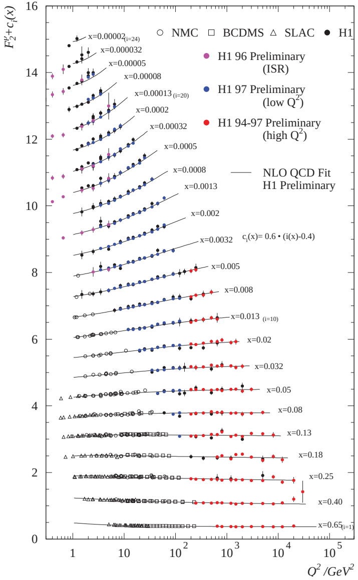
$$\sum_{c=q,g} \int_0^1 \mathrm{d}\zeta\, \zeta\, \hat{P}_{ac}(\zeta) = 0 \qquad \rightarrow \qquad \text{momentum sum rule}$$

  $\rightarrow$ defines regularized DGLAP splitting functions $\hat{P}_{ab}$ as

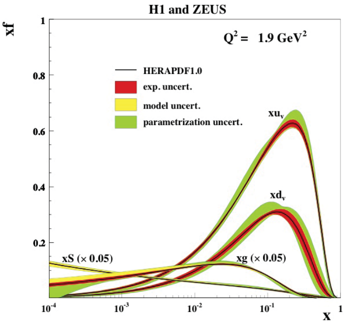
$$\hat{P}_{ab}(z) = \lim_{\varepsilon \to 0} \left[P_{ab}(z)\Theta(1-\varepsilon-z) - \delta_{ab} \frac{\Theta(z-1+\varepsilon)}{\varepsilon} \sum_{c=q,g} \int_0^{1-\varepsilon} d\zeta\, \zeta\, P_{ac}(\zeta) \right]$$

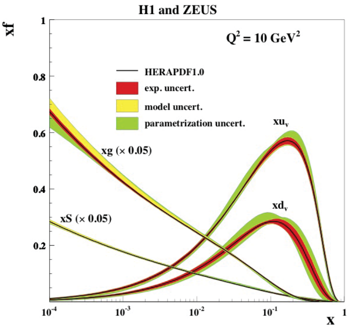
PDF measurements



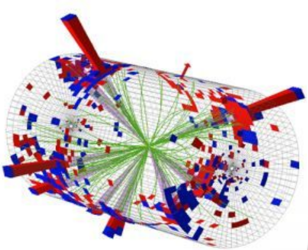


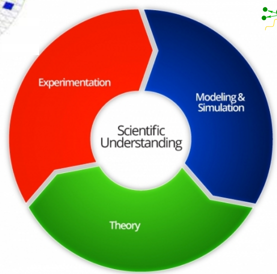
PDF measurements

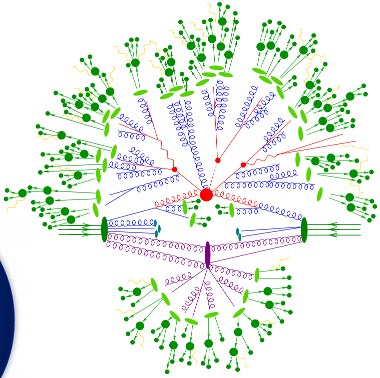




How event generators fit in



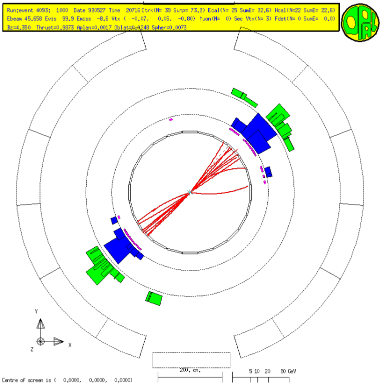


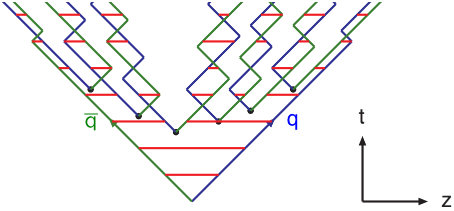


$$\mathcal{L} = -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} + i\bar{\psi}\not{D}\psi + h.c.$$

Event generators in 1978

[Andersson,Gustafson,Ingelman,Sjöstrand] Phys.Rept.97(1983)31





- Lund string model: ∼ like rubber band that is pulled apart and breaks into pieces, or like a magnet broken into smaller pieces.
- Complete description of 2-jet events in e^+e^- →hadrons

Event generators in 1978

[Andersson,Gustafson,Ingelman,Sjöstrand] Phys.Rept.97(1983)31

```fortran
      SUBROUTINE JETGEN(N)
      COMMON /JET/ K(100,2), P(100,5)
      COMMON /PAR/ PUD, PS1, SIGMA, CX2, EBEG, WFIN, IFLBEG
      COMMON /DATA1/ MESO(9,2), CMIX(6,2), PMAS(19)
      IFLSGN=(10-IFLBEG)/5
      W=2.*EBEG
      I=0
      IPD=0
C 1 FLAVOUR AND PT FOR FIRST QUARK
      IFL1=IABS(IFLBEG)
      PT1=SIGMA*SQRT(-ALOG(RANF(0)))
      PHI1=6.2832*RANF(0)
      PX1=PT1*COS(PHI1)
      PY1=PT1*SIN(PHI1)
  100 I=I+1
C 2 FLAVOUR AND PT FOR NEXT ANTIQUARK
      IFL2=1+INT(RANF(0)/PUD)
      PT2=SIGMA*SQRT(-ALOG(RANF(0)))
      PHI2=6.2832*RANF(0)
      PX2=PT2*COS(PHI2)
      PY2=PT2*SIN(PHI2)
C 3 MESON FORMED, SPIN ADDED AND FLAVOUR MIXED
      K(I,1)=MESO(3*(IFL1-1)+IFL2,IFLSGN)
      ISPIN=INT(PS1+RANF(0))
      K(I,2)=1+9*ISPIN+K(I,1)
      IF(K(I,1).LE.6) GOTO 110
      TMIX=RANF(0)
      KM=K(I,1)-6+3*ISPIN
      K(I,2)=8+9*ISPIN+INT(TMIX+CMIX(KM,1))+INT(TMIX+CMIX(KM,2))
C 4 MESON MASS FROM TABLE, PT FROM CONSTITUENTS
  110 P(I,5)=PMAS(K(I,2))
      P(I,1)=PX1+PX2
      P(I,2)=PY1+PY2
      PMTS=P(I,1)**2+P(I,2)**2+P(I,5)**2
C 5 RANDOM CHOICE OF X=(E+PZ)MESON/(E+PZ)AVAILABLE GIVES E AND PZ
      X=RANF(0)
      IF(RANF(0).LT.CX2) X=1.-X**(1./3.)
      P(I,3)=(X*W-PMTS/(X*W))/2.
      P(I,4)=(X*W+PMTS/(X*W))/2.
C 6 IF UNSTABLE, DECAY CHAIN INTO STABLE PARTICLES
  120 IPD=IPD+1
      IF(K(IPD,2).GE.8) CALL DECAY(IPD,I)
      IF(IPD.LT.I.AND.I.LE.96) GOTO 120
C 7 FLAVOUR AND PT OF QUARK FORMED IN PAIR WITH ANTIQUARK ABOVE
      IFL1=IFL2
      PX1=-PX2
      PY1=-PY2
C 8 IF ENOUGH E+PZ LEFT, GO TO 2
      W=(1.-X)*W
      IF(W.GT.WFIN.AND.I.LE.95) GOTO 100
      N=I
      RETURN
      END
```

```fortran
      SUBROUTINE LIST(N)
      COMMON /JET/ K(100,2), P(100,5)
      COMMON /DATA3/ CHA1(9), CHA2(19), CHA3(2)
      WRITE(6,110)
      DO 100 I=1,N
      IF(K(I,1).GT.0) C1=CHA1(K(I,1))
      IF(K(I,1).LE.0) IC1=-K(I,1)
      C2=CHA2(K(I,2))
      C3=CHA3((47-K(I,2))/20)
      IF(K(I,1).GT.0) WRITE(6,120) I, C1, C2, C3, (P(I,J), J=1,5)
  100 IF(K(I,1).LE.0) WRITE(6,130) I, IC1, C2, C3, (P(I,J), J=1,5)
      RETURN
  110 FORMAT(////T11,'I',T17,'ORI',T24,'PART',T32,'STAB',
     &T44,'PX',T56,'PY',T68,'PZ',T80,'E',T92,'M'/)
  120 FORMAT(10X,I2,4X,A2,1X,2(4X,A4),5(4X,F8.1))
  130 FORMAT(10X,I2,4X,1X,I2,2(4X,A4),5(4X,F8.1))
      END
```

```fortran
      SUBROUTINE DECAY(IPD,I)
      COMMON /JET/ K(100,2), P(100,5)
      COMMON /DATA1/ MESO(9,2), CMIX(6,2), PMAS(19)
      COMMON /DATA2/ IDCO(12), CBR(29), KDP(29,3)
      DIMENSION U(3), BE(3)
C 1 DECAY CHANNEL CHOICE, GIVES DECAY PRODUCTS
      TBR=RANF(0)
      IDC=IDCO(K(IPD,2)-7)
  100 IDC=IDC+1
      IF(TBR.GT.CBR(IDC)) GOTO 100
      ND=(59+KDP(IDC,3))/20
      DO 110 I1=I+1,I+ND
      K(I1,1)=-IPD
      K(I1,2)=KDP(IDC,I1-I)
  110 P(I1,5)=PMAS(K(I1,2))
C 2 IN THREE-PARTICLE DECAY CHOICE OF INVARIANT MASS OF PRODUCTS 2+3
      IF(ND.EQ.2) GOTO 130
      SA=(P(IPD,5)+P(I+1,5))**2
      SB=(P(IPD,5)-P(I+1,5))**2
      SC=(P(I+2,5)+P(I+3,5))**2
      SD=(P(I+2,5)-P(I+3,5))**2
      TDU=(SA-SD)*(SB-SC)/(4.*SQRT(SB*SC))
      IF(K(IPD,2).GE.11) TDU=SQRT(SB*SC)*TDU**3
  120 SX=SC+(SB-SC)*RANF(0)
      TDF=SQRT((SX-SA)*(SX-SB)*(SX-SC)*(SX-SD))/SX
      IF(K(IPD,2).GE.11) TDF=SX*TDF**3
      IF(RANF(0)*TDU.GT.TDF) GOTO 120
      P(100,5)=SQRT(SX)
C 3 TWO-PARTICLE DECAY IN CM, TWICE TO SIMULATE THREE-PARTICLE DECAY
  130 DO 160 IL=1,ND-1
      IO=(IL-1)*100-(IL-2)*IPD
      I1=I+IL
      I2=(ND-IL-1)*100-(ND-IL-2)*(I+IL+1)
      PA=SQRT((P(IO,5)**2-(P(I1,5)+P(I2,5))**2)*
     &(P(IO,5)**2-(P(I1,5)-P(I2,5))**2))/(2.*P(IO,5))
  140 U(3)=2.*RANF(0)-1.
      PHI=6.2832*RANF(0)
      U(1)=SQRT(1.-U(3)**2)*COS(PHI)
      U(2)=SQRT(1.-U(3)**2)*SIN(PHI)
      TDA=1.-(U(1)*P(IO,1)+U(2)*P(IO,2)+U(3)*P(IO,3))**2/
     &(P(IO,1)**2+P(IO,2)**2+P(IO,3)**2)
      IF(K(IPD,2).GE.11.AND.IL.EQ.2.AND.RANF(0).GT.TDA) GOTO 140
      DO 150 J=1,3
      P(I1,J)=PA*U(J)
  150 P(I2,J)=-PA*U(J)
      P(I1,4)=SQRT(PA**2+P(I1,5)**2)
  160 P(I2,4)=SQRT(PA**2+P(I2,5)**2)
C 4 DECAY PRODUCTS LORENTZ TRANSFORMED TO LAB SYSTEM
      DO 190 IL=ND-1,1,-1
      IO=(IL-1)*100-(IL-2)*IPD
      DO 170 J=1,3
  170 BE(J)=P(IO,J)/P(IO,4)
      GA=P(IO,4)/P(IO,5)
      DO 190 I1=I+IL,I+ND
      BEP=BE(1)*P(I1,1)+BE(2)*P(I1,2)+BE(3)*P(I1,3)
      DO 180 J=1,3
  180 P(I1,J)=P(I1,J)+GA*(GA/(1.+GA)*BEP+P(I1,4))*BE(J)
  190 P(I1,4)=GA*(P(I1,4)+BEP)
      I=I+ND
      RETURN
      END
```

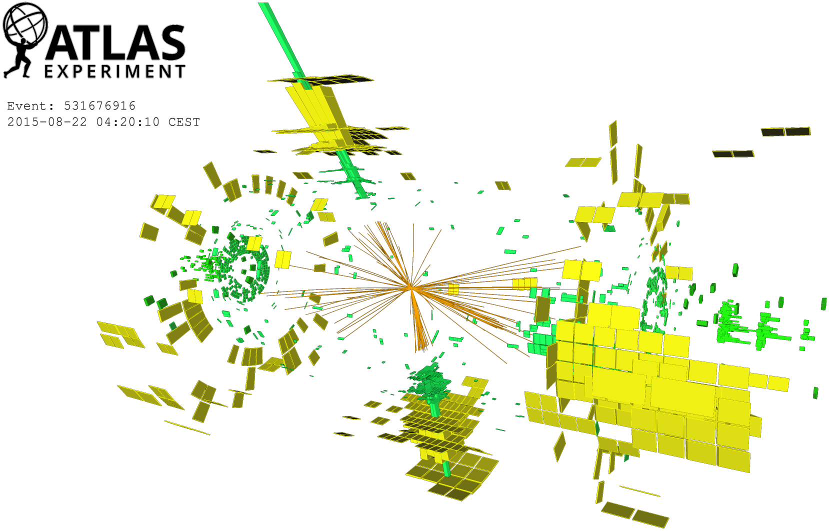
```fortran
      SUBROUTINE EDIT(N)
      COMMON /JET/ K(100,2), P(100,5)
      COMMON /EDPAR/ ITHROW, PZMIN, PMIN, THETA, PHI, BETA(3)
      REAL ROT(3,3), PR(3)
C 1 THROW AWAY NEUTRALS OR UNSTABLE OR WITH TOO LOW PZ OR P
      I1=0
      DO 110 I=1,N
      IF(ITHROW.GE.1.AND.K(I,2).GE.8) GOTO 110
      IF(ITHROW.GE.2.AND.K(I,2).GE.6) GOTO 110
      IF(ITHROW.GE.3.AND.K(I,2).EQ.1) GOTO 110
      IF(P(I,3).LT.PZMIN.OR.P(I,4)**2-P(I,5)**2.LT.PMIN**2) GOTO 110
      I1=I1+1
      K(I1,1)=IDIM(K(I,1),0)
      K(I1,2)=K(I,2)
      DO 100 J=1,5
  100 P(I1,J)=P(I,J)
  110 CONTINUE
      N=I1
C 2 ROTATE TO GIVE JET PRODUCED IN DIRECTION THETA, PHI
      IF(THETA.LT.1E-4) GOTO 140
      ROT(1,1)=COS(THETA)*COS(PHI)
      ROT(1,2)=-SIN(PHI)
      ROT(1,3)=SIN(THETA)*COS(PHI)
      ROT(2,1)=COS(THETA)*SIN(PHI)
      ROT(2,2)=COS(PHI)
      ROT(2,3)=SIN(THETA)*SIN(PHI)
      ROT(3,1)=-SIN(THETA)
      ROT(3,2)=0.
      ROT(3,3)=COS(THETA)
      DO 130 I=1,N
      DO 120 J=1,3
  120 PR(J)=P(I,J)
      DO 130 J=1,3
  130 P(I,J)=ROT(J,1)*PR(1)+ROT(J,2)*PR(2)+ROT(J,3)*PR(3)
C 3 OVERALL LORENTZ BOOST GIVEN BY BETA VECTOR
  140 IF(BETA(1)**2+BETA(2)**2+BETA(3)**2.LT.1E-8) RETURN
      GA=1./SQRT(1.-BETA(1)**2-BETA(2)**2-BETA(3)**2)
      DO 160 I=1,N
      BEP=BETA(1)*P(I,1)+BETA(2)*P(I,2)+BETA(3)*P(I,3)
      DO 150 J=1,3
  150 P(I,J)=P(I,J)+GA*(GA/(1.+GA)*BEP+P(I,4))*BETA(J)
  160 P(I,4)=GA*(P(I,4)+BEP)
      RETURN
      END
```

```fortran
      BLOCK DATA
      COMMON /PAR/ PUD, PS1, SIGMA, CX2, EBEG, WFIN, IFLBEG
      COMMON /EDPAR/ ITHROW, PZMIN, PMIN, THETA, PHI, BETA(3)
      COMMON /DATA1/ MESO(9,2), CMIX(6,2), PMAS(19)
      COMMON /DATA2/ IDCO(12), CBR(29), KDP(29,3)
      COMMON /DATA3/ CHA1(9), CHA2(19), CHA3(2)
      DATA PUD/0.4/, PS1/0.5/, SIGMA/350./, CX2/0.77/,
     &EBEG/10000./, WFIN/100./, IFLBEG/1/
      DATA ITHROW/1/, PZMIN/0./, PMIN/0./, THETA,PHI,BETA/5*0./
      DATA MESO/7,1,3,2,8,5,4,6,9,7,2,4,1,8,6,3,5,9/
      DATA CMIX/2*0.5,1.,2*0.5,1.,2*0.25,0.5,2*0.,1./
      DATA PMAS/0.,2*139.6,2*493.7,2*497.7,135.,548.8,957.6,
     &2*765.9,2*892.2,2*896.3,770.2,782.6,1019.6/
      DATA IDCO/0,1,6,11,12,13,15,17,19,21,22,25/
      DATA CBR/1.,0.381,0.681,0.918,0.969,1.,0.426,0.662,0.959,
     &0.980,1.,1.,1.,0.667,1.,0.667,1.,0.667,1.,0.667,1.,1.,
     &0.899,0.987,1.,0.486,0.837,0.984,1./
      DATA KDP/1,1,8,2,1,1,2,8,1,1,1,2,3,6,4,7,5,4,6,5,7,2,2,
     &1,2,4,6,2,1,1,1,8,3,2,1,3,8,17,18,1,8,8,2,8,3,8,3,8,2,8,
     &3,3,8,3,5,7,3,9,0,0,8,8,3,8,9,9,14*0,8,4*0,8,0/
      DATA CHA1/'UD','DU','US','SU','DS','SD','UU','DD','SS'/
      DATA CHA2/'GAMM','PI+','PI-','K+','K-','K0','KB0','PI0','ETA',
     &'ETAP','RHO+','RHO-','K*+','K*-','K*0','KB*0','RHO0','OMEG','PHI'/
      DATA CHA3/'    ','STAB'/
      END
```

≈ 200 punched cards

Fortran code

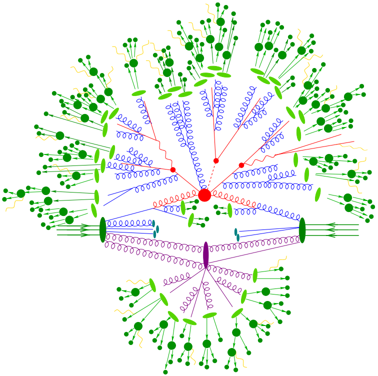
Experimental situation in 2016



Event generators in 2016

Need to cover large dynamic range

- Short distance interactions
  - Signal process
  - Radiative corrections
- Long-distance interactions
  - Hadronization
  - Particle decays



Divide and Conquer

- Quantity of interest: Total interaction rate
- Convolution of short & long distance physics

$$\sigma_{p_1 p_2 \to X} = \sum_{i,j \in \{q,g\}} \int \mathrm{d}x_1 \mathrm{d}x_2 \underbrace{f_{p_1,i}(x_1, \mu_F^2) f_{p_2,j}(x_2, \mu_F^2)}_{\text{long distance}} \underbrace{\hat{\sigma}_{ij \to X}(x_1 x_2, \mu_F^2)}_{\text{short distance}}$$

General-purpose event generators for LHC physics

[Buckley et al.] arXiv:1101.2599

Herwig

- Originated in coherent shower studies → angular ordered PS
- Front-runner in development of Mc@Nlo and Powheg
- Simple in-house ME generator & spin-correlated decay chains
- Original framework for cluster fragmentation

Pythia

- Originated in hadronization studies → Lund string
- Leading in development of multiple interaction models
- Pragmatic attitude to ME generation → external tools
- Extensive PS development and earliest ME⊕PS matching

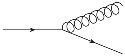
Sherpa

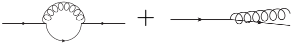
- Started with PS generator Apacic++ & ME generator Amegic++
- Current MPI model and hadronization pragmatic add-ons
- Leading in development of automated ME⊕PS merging
- Automated framework for NLO calculations and Mc@Nlo

Radiative corrections as a branching process

[Marchesini,Webber] NPB238(1984)1, [Sjöstrand] PLB157(1985)321

- Make two well motivated assumptions
  - Parton branching can occur in two ways

 - observed

 - unobserved

  - Evolution conserves probability
- The consequence is Poisson statistics
  - Let the decay probability be λ
  - Assume indistinguishable particles $\rightarrow$ naive probability for n emissions

$$P_{\text{naive}}(n, \lambda) = \frac{\lambda^n}{n!}$$

  - Probability conservation (i.e. unitarity) implies a no-emission probability

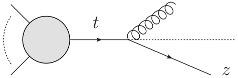
$$P(n, \lambda) = \frac{\lambda^n}{n!} \exp\{-\lambda\} \qquad \longrightarrow \qquad \sum_{n=0}^{\infty} P(n, \lambda) = 1$$

- In the context of parton showers $\Delta = \exp\{-\lambda\}$ is called Sudakov factor

Radiative corrections as a branching process

- Decay probability for parton state in collinear limit

$$\lambda \to \frac{1}{\sigma_n}\int_t^{Q^2} \mathrm{d}\bar{t}\,\frac{\mathrm{d}\sigma_{n+1}}{\mathrm{d}\bar{t}} \approx \sum_{\text{jets}}\int_t^{Q^2}\frac{\mathrm{d}\bar{t}}{\bar{t}}\int \mathrm{d}z\,\frac{\alpha_s}{2\pi}P(z)$$



Parameter t identified with evolution "time"

- Splitting function $P(z)$ spin & color dependent

$$P_{qq}(z) = C_F\left[\frac{2}{1-z} - (1+z)\right] \qquad P_{gq}(z) = T_R\left[\,z^2 + (1-z)^2\,\right]$$

$$P_{gg}(z) = C_A\left[\frac{2}{1-z} - 2 + z(1-z)\right] + (z \leftrightarrow 1-z)$$

- Matching to soft limit will requires some care, because full soft emission probability present in all collinear sectors

$$\frac{1}{t}\,\frac{2}{1-z} \xrightarrow{z\to 1} \frac{p_i p_k}{(p_i q)(q p_k)}$$

Soft double counting problem [Marchesini,Webber] NPB310(1988)461

- Let us first see how to compute the Poissonian in practice

Monte-Carlo methods: Basic integration

SLAC

- Pseudo-random number generators produce uniform numbers
- The probability to draw a point in $[x, x + \Delta x]$ is Δx hence we can compute integrals as expectation values

  $$I = \int_0^1 \mathrm{d}x = \frac{1}{N}\sum_{i=1}^{N} 1 = \langle 1 \rangle = 1$$

  N - Number of MC events (points)
- The statistical uncertainty on this integral is

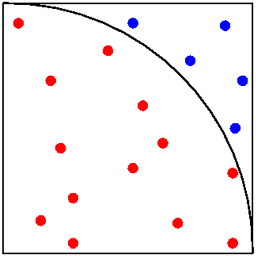
  $$\sigma_I = \sqrt{\frac{\langle 1^2 \rangle - \langle 1 \rangle^2}{N-1}} = 0\,, \qquad \text{if} \qquad N > 1$$

- Repeating this with an unknown function $f(x)$ and arbitrary integration range reveals the power of the method: MC error scales as $1/\sqrt{N}$, independent of number of dimensions

  $$I = \int_a^b \mathrm{d}x\, f(x) = \frac{b-a}{N}\sum_{i=1}^{N} f(x) = [\,b-a\,]\langle f \rangle$$

  $$\sigma_I = [\,b-a\,]\sqrt{\frac{\langle f^2 \rangle - \langle f \rangle^2}{N-1}}$$

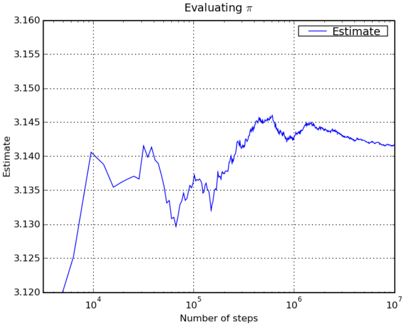
Monte-Carlo methods: Basic hit-or-miss



$$\frac{\text{Hits}}{\text{Misses} + \text{Hits}} \rightarrow \frac{\pi}{4}$$

Throw random points (x,y),
with x, y in [0,1]

For hits: $(x^2 + y^2) < r^2 = 1$



Monte-Carlo methods: Importance sampling

- So far we used uniformly distributed random numbers
- Assume we want points following the distribution $g(x)$ and that $g(x)$ has a known primitive $G(x) = \int^x \mathrm{d}x' g(x')$
- Probability of producing point in $[x, x + \mathrm{d}x]$ should be $g(x)\,\mathrm{d}x$
- This can be achieved by solving the following equation for x

$$\int_a^x \mathrm{d}x'\, g(x') = R \int_a^b \mathrm{d}x'\, g(x')$$

where R is a uniform random number in $[0, 1]$

$$x = G^{-1}\Big[G(a) + R\left(G(b) - G(a)\right)\Big]$$

- In many cases we can approximate the unknown integral of a function $f(x)$ with some known function $g(x)$ such that primitive $G(x)$ is known
- This amounts to a variable transformation

$$I = \int_a^b \mathrm{d}x\, g(x)\, \frac{f(x)}{g(x)} = \int_{G(a)}^{G(b)} \mathrm{d}G(x)\, w(x) \quad \text{where} \quad w(x) = \frac{f(x)}{g(x)}$$

- Integral and error estimate are

$$I = \big[G(b) - G(a)\big]\langle w\rangle \qquad\qquad \sigma = \big[G(b) - G(a)\big]\sqrt{\frac{\langle w^2\rangle - \langle w\rangle^2}{N-1}}$$

  N - Number of MC events (points)
- Note: I is independent of $g(x)$, but σ is not
  $\rightarrow$ suitable choice of $g(x)$ can be used to minimize error

Monte-Carlo methods: Poisson distributions

- Assume nuclear decay process described by $g(x)$
- Nucleus can decay only if it has not decayed already
  Must account for survival probability $\leftrightarrow$ Poisson distribution

$$\mathcal{G}(x) = g(x)\Delta(x,b) \qquad \text{where} \qquad \Delta(x,b) = \exp\left\{-\int_x^b \mathrm{d}x'\, g(x')\right\}$$

- If $G(x)$ is known, then we also know the integral of $\mathcal{G}(x)$

$$\int_x^b \mathrm{d}x' \mathcal{G}(x') = \int_x^b \mathrm{d}x' \, \frac{\mathrm{d}\Delta(x',b)}{\mathrm{d}x'} = 1 - \Delta(x,b)$$

- Can generate events by requiring $1 - \Delta(x,b) = 1 - R$

$$x = G^{-1}\Big[G(b) + \log R\Big]$$

SLAC

- Hit-or-miss method for Poisson distributions $\to$ veto algorithm
  - Generate event according to $\mathcal{G}(x)$
  - Accept with $w(x) = f(x)/g(x)$
  - If rejected, continue starting from x
- Probability for immediate acceptance

$$\frac{f(x)}{g(x)}\,g(x)\exp\left\{-\int_x^b \mathrm{d}x'\,g(x')\right\}$$

- Probability for acceptance after one rejection

$$\frac{f(x)}{g(x)}\,g(x)\int_x^b \mathrm{d}x_1 \exp\left\{-\int_x^{x_1} \mathrm{d}x'\,g(x')\right\}\left(1-\frac{f(x_1)}{g(x_1)}\right)g(x_1)\exp\left\{-\int_{x_1}^b \mathrm{d}x'\,g(x')\right\}$$

- For n intermediate rejections we obtain n nested integrals $\int_x^b \int_{x_1}^b \ldots \int_{x_{n-1}}^b$
- Disentangling yields $1/n!$ and summing over all possible rejections gives

$$f(x)\exp\left\{-\int_x^b \mathrm{d}x'\,g(x')\right\}\sum_{n=0}^{\infty}\frac{1}{n!}\left[\int_x^b \mathrm{d}x'\left[g(x')-f(x')\right]\right]^n = f(x)\exp\left\{-\int_x^b \mathrm{d}x'\,f(x')\right\}$$

Monte-Carlo method for parton showers

- Start with set of n partons at scale t', which evolve collectively
  Sudakovs factorize, schematically

$$\Delta(t,t') = \prod_{i=1}^{n} \Delta_i(t,t')\;, \qquad\qquad \Delta_i(t,t') = \prod_{j=q,g} \Delta_{i\to j}(t,t')$$

- Find new scale t where next branching occurs using veto algorithm
  - Generate t using overestimate $\alpha_s^{\max} P_{ab}^{\max}(z)$
  - Determine "winner" parton i and select new flavor j
  - Select splitting variable according to overestimate
  - Accept point with weight $\alpha_s(k_T^2) P_{ab}(z) / \alpha_s^{\max} P_{ab}^{\max}(z)$
- Construct splitting kinematics and update event record
- Continue until t falls below an IR cutoff