QCD multi-jet calculation on MadGraph

Yoshitaro Takaesu (Sokendai)

In collaboration with K. Hagiwara (KEK)

What is MadGraph?

→ Automatic Amplitude Generator

Input

```
# Begin MODEL
mssm
# End    MODEL
```

```
pp>(go>uul~)(go>ddl~)
QCD=99                # Mc
QED=99                 # M
end_coup              # Er
```

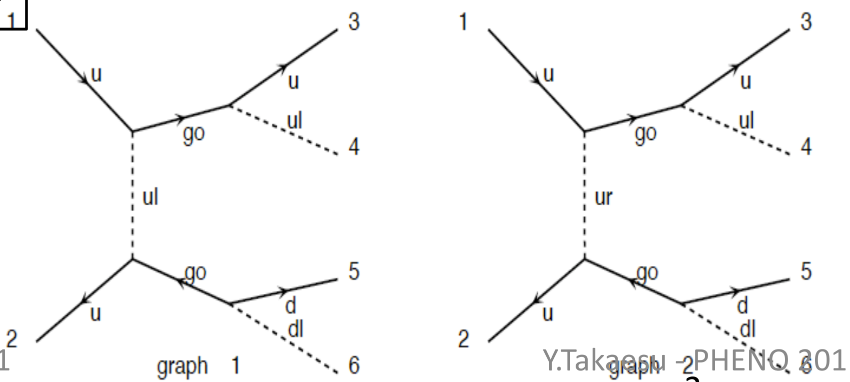
Model + process

MG

テキ

Output

Diagram



HELAS Code

```
CALL VXXXXX(P(0,1   ),ZERO ,NHEL(1   ),-1*IC(1   ),W(1,1   ))
CALL VXXXXX(P(0,2   ),ZERO ,NHEL(2   ),-1*IC(2   ),W(1,2   ))
CALL OXXXXX(P(0,3   ),ZERO ,NHEL(3   ),+1*IC(3   ),W(1,3   ))
CALL SXXXXX(P(0,4   ),+1*IC(4   ),W(1,4   ))
CALL IXXXXX(P(0,5   ),ZERO ,NHEL(5   ),-1*IC(5   ),W(1,5   ))
CALL SXXXXX(P(0,6   ),+1*IC(6   ),W(1,6   ))
CALL FSOXXX(W(1,3   ),W(1,4   ),GQLGOP ,MGO     ,WGO     ,W(1,
&    7   ))
CALL FVOXXX(W(1,7   ),W(1,2   ),GGI ,MGO     ,WGO     ,W(1,8   ))
```

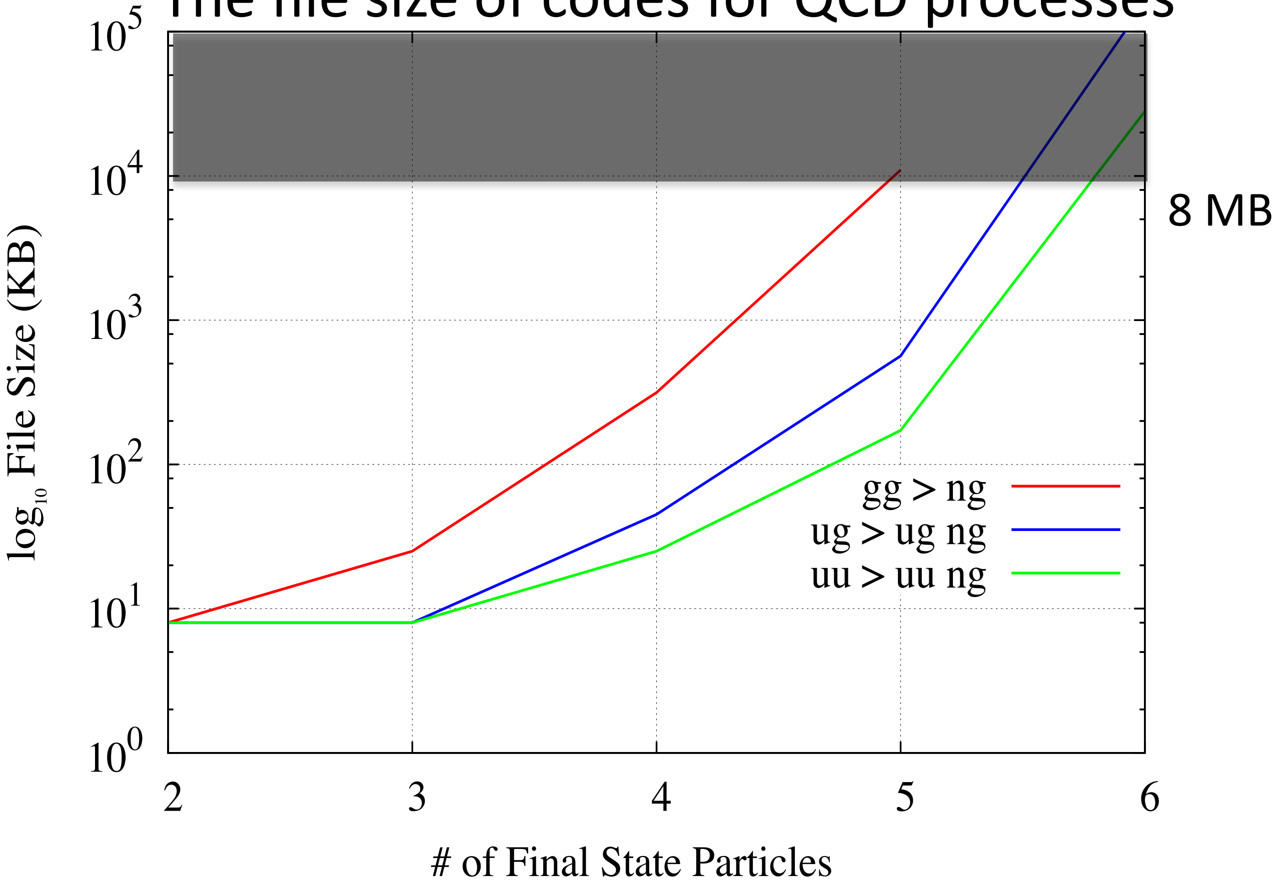
MG is very useful and popular

BUT …

Current Limits of MG

Generated Codes > 8MB cannot be compiled.

The file size of codes for QCD processes



$$gg \to 5g$$
$$qg \to q5g$$
$$qq \to qq4g$$

What makes the codes to be

LARGE ?

The structure of the code

The size of Color Matrix and

Dual Amps get Large

as # of final particles grows.

ex) $Tr[T^a T^b T^c T^d]$

Color Factor **dual amp.**

$$\mathcal{M} = \sum_i C_i A_i$$

Dual Amp.

Color Matrix

$$\sum_{color} |\mathcal{M}|^2 = \sum_{ij} A_i \Big(\sum_{color} C_i C_j^* \Big) A_j^*$$

```
C
C      COLOR DATA
C
       DATA DENOM(1)/6/
       DATA (CF(I,  1),I=  1,  6) /   19,   -2,   -2,   -2,   -2,    4/
C      1 Tr(1,2,3,4)
       DATA DENOM(2)/6/
       DATA (CF(I,  2),I=  1,  6) /   -2,   19,   -2,    4,   -2,   -2/
                    .
                    .
                    .
C      1 Tr(1,4,2,3)
       DATA DENOM(6)/6/
       DATA (CF(I,  6),I=  1,  6) /    4,   -2,   -2,   -2,   -2,   19/
C      1 Tr(1,4,3,2)
C
C      BEGIN CODE
C      ----------
       CALL VXXXXX(P(0,1),ZERO,NHEL(1),-1*IC(1),W(1,1))
       CALL VXXXXX(P(0,2),ZERO,NHEL(2),-1*IC(2),W(1,2))
       CALL VXXXXX(P(0,3),ZERO,NHEL(3),+1*IC(3),W(1,3))
       CALL VXXXXX(P(0,4),ZERO,NHEL(4),+1*IC(4),W(1,4))
C      Amplitude(s) for diagram number 1
       CALL VVVV4_0(W(1,1),W(1,2),W(1,3),W(1,4),GC_6,AMP(1))
       CALL VVVV3_0(W(1,1),W(1,2),W(1,3),W(1,4),GC_6,AMP(2))
                    .
                    .
                    .
       CALL VVV1_0(W(1,2),W(1,4),W(1,6),GC_4,AMP(5))
       CALL VVV1_1(W(1,1),W(1,4),GC_4,ZERO, ZERO, W(1,7))
C      Amplitude(s) for diagram number 4
       CALL VVV1_0(W(1,2),W(1,3),W(1,7),GC_4,AMP(6))
       JAMP(1)=+2*(-AMP(3)+AMP(1)-AMP(4)+AMP(6))
       JAMP(2)=+2*(+AMP(3)+AMP(2)+AMP(4)+AMP(5))
       JAMP(3)=+2*(-AMP(1)-AMP(2)-AMP(5)-AMP(6))
       JAMP(4)=+2*(+AMP(3)+AMP(2)+AMP(4)+AMP(5))
       JAMP(5)=+2*(-AMP(1)-AMP(2)-AMP(5)-AMP(6))
       JAMP(6)=+2*(-AMP(3)+AMP(1)-AMP(4)+AMP(6))

       MATRIX = 0.D0
       DO I = 1, NCOLOR
         ZTEMP = (0.D0,0.D0)
         DO J = 1, NCOLOR
           ZTEMP = ZTEMP + CF(J,I)*JAMP(J)
         ENDDO
         MATRIX = MATRIX+ZTEMP*DCONJG(JAMP(I))/DENOM(I)
       ENDDO
```

Overcome the limit

Color Matrix

```
C
C     COLOR DATA
C
      DATA DENOM(1)/6/
      DATA (CF(I,  1),I=  1,  6) /   19,   -2,   -2,   -2,   -2,    4/
C     1 Tr(1,2,3,4)
      DATA DENOM(2)/6/
      DATA (CF(I,  2),I=  1,  6) /   -2,   19,   -2,    4,   -2,   -2/
                          .
                          .
                          .
C     1 Tr(1,4,2,3)
      DATA DENOM(6)/6/
      DATA (CF(I,  6),I=  1,  6) /    4,   -2,   -2,   -2,   -2,   19/
C     1 Tr(1,4,3,2)
C     ----------
C     BEGIN CODE
C
```

Dual Amp.

```
      CALL VXXXXX(P(0,1),ZERO,NHEL(1),-1*IC(1),W(1,1))
      CALL VXXXXX(P(0,2),ZERO,NHEL(2),-1*IC(2),W(1,2))
      CALL VXXXXX(P(0,3),ZERO,NHEL(3),+1*IC(3),W(1,3))
      CALL VXXXXX(P(0,4),ZERO,NHEL(4),+1*IC(4),W(1,4))
C     Amplitude(s) for diagram number 1
      CALL VVVV4_0(W(1,1),W(1,2),W(1,3),W(1,4),GC_6,AMP(1))
      CALL VVVV3_0(W(1,1),W(1,2),W(1,3),W(1,4),GC_6,AMP(2))
                          .
                          .
                          .
      CALL VVV1_0(W(1,2),W(1,4),W(1,6),GC_4,AMP(5))
      CALL VVV1_1(W(1,1),W(1,4),GC_4,ZERO, ZERO, W(1,7))
C     Amplitude(s) for diagram number 4
      CALL VVV1_0(W(1,2),W(1,3),W(1,7),GC_4,AMP(6))
      JAMP(1)=+2*(-AMP(3)+AMP(1)-AMP(4)+AMP(6))
      JAMP(2)=+2*(+AMP(3)+AMP(2)+AMP(4)+AMP(5))
      JAMP(3)=+2*(-AMP(1)-AMP(2)-AMP(5)-AMP(6))
      JAMP(4)=+2*(+AMP(3)+AMP(2)+AMP(4)+AMP(5))
      JAMP(5)=+2*(-AMP(1)-AMP(2)-AMP(5)-AMP(6))
      JAMP(6)=+2*(-AMP(3)+AMP(1)-AMP(4)+AMP(6))
```

```
      MATRIX = 0.D0
      DO I = 1, NCOLOR
        ZTEMP = (0.D0,0.D0)
        DO J = 1, NCOLOR
          ZTEMP = ZTEMP + CF(J,I)*JAMP(J)
        ENDDO
        MATRIX = MATRIX+ZTEMP*DCONJG(JAMP(I))/DENOM(I)
      ENDDO
```

Overcome the limit

Remove Color Matrix

Calculate it on the fly

Dual Amp.

```
C     ----------
C     BEGIN CODE
C
      CALL VXXXXX(P(0,1),ZERO,NHEL(1),-1*IC(1),W(1,1))
      CALL VXXXXX(P(0,2),ZERO,NHEL(2),-1*IC(2),W(1,2))
      CALL VXXXXX(P(0,3),ZERO,NHEL(3),+1*IC(3),W(1,3))
      CALL VXXXXX(P(0,4),ZERO,NHEL(4),+1*IC(4),W(1,4))
C     Amplitude(s) for diagram number 1
      CALL VVVV4_0(W(1,1),W(1,2),W(1,3),W(1,4),GC_6,AMP(1))
      CALL VVVV3_0(W(1,1),W(1,2),W(1,3),W(1,4),GC_6,AMP(2))
                        .
                        .
                        .
      CALL VVV1_0(W(1,2),W(1,4),W(1,6),GC_4,AMP(5))
      CALL VVV1_1(W(1,1),W(1,4),GC_4,ZERO, ZERO, W(1,7))
C     Amplitude(s) for diagram number 4
      CALL VVV1_0(W(1,2),W(1,3),W(1,7),GC_4,AMP(6))
      JAMP(1)=+2*(-AMP(3)+AMP(1)-AMP(4)+AMP(6))
      JAMP(2)=+2*(+AMP(3)+AMP(2)+AMP(4)+AMP(5))
      JAMP(3)=+2*(-AMP(1)-AMP(2)-AMP(5)-AMP(6))
      JAMP(4)=+2*(+AMP(3)+AMP(2)+AMP(4)+AMP(5))
      JAMP(5)=+2*(-AMP(1)-AMP(2)-AMP(5)-AMP(6))
      JAMP(6)=+2*(-AMP(3)+AMP(1)-AMP(4)+AMP(6))
```

```
      MATRIX = 0.D0
      DO I = 1, NCOLOR
        ZTEMP = (0.D0,0.D0)
        DO J = 1, NCOLOR
          ZTEMP = ZTEMP + CF(J,I)*JAMP(J)
        ENDDO
        MATRIX = MATRIX+ZTEMP*DCONJG(JAMP(I))/DENOM(I)
      ENDDO
```

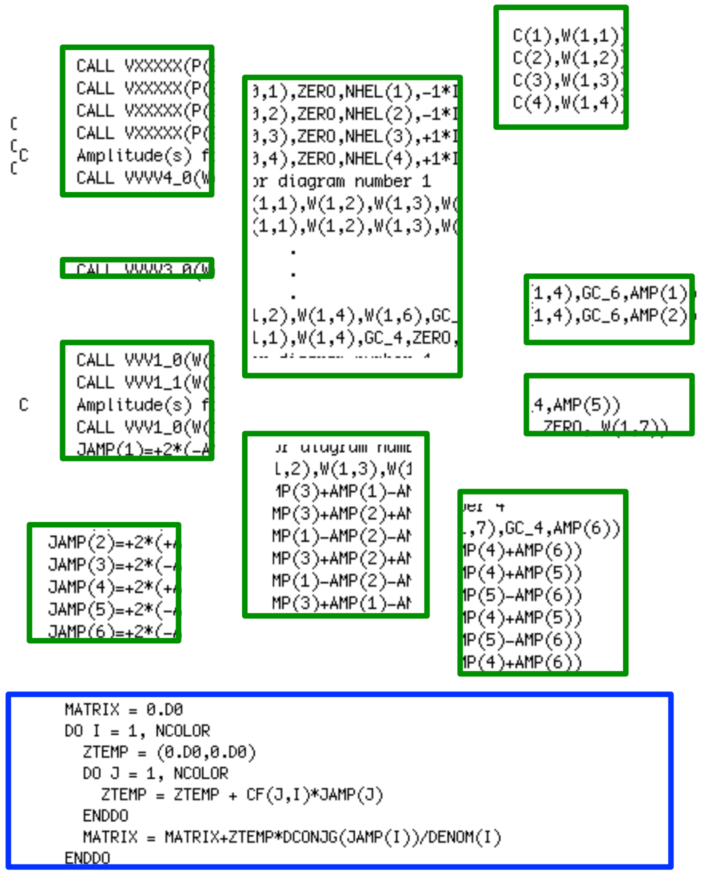
Overcome the limit

Remove Color Matrix

Calculate it on the fly

Divide the subroutine part into small pieces

(1 Source file for 1 Dual Amp)



Overcome the limit

Remove Color Matrix

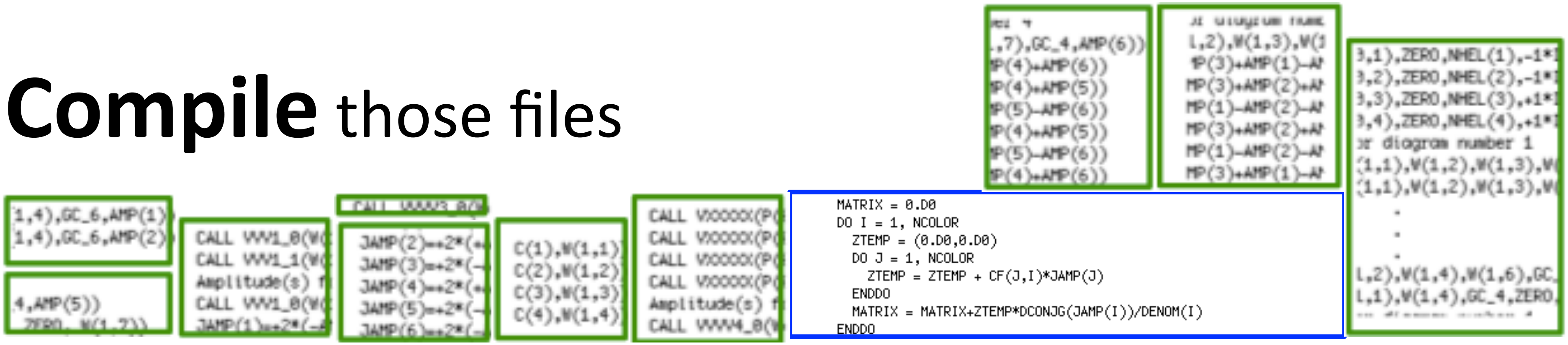
Calculate it on the fly

Divide the subroutine

part into small pieces

(1 Source file for 1 Dual Amp)

Compile those files



Making the new Code

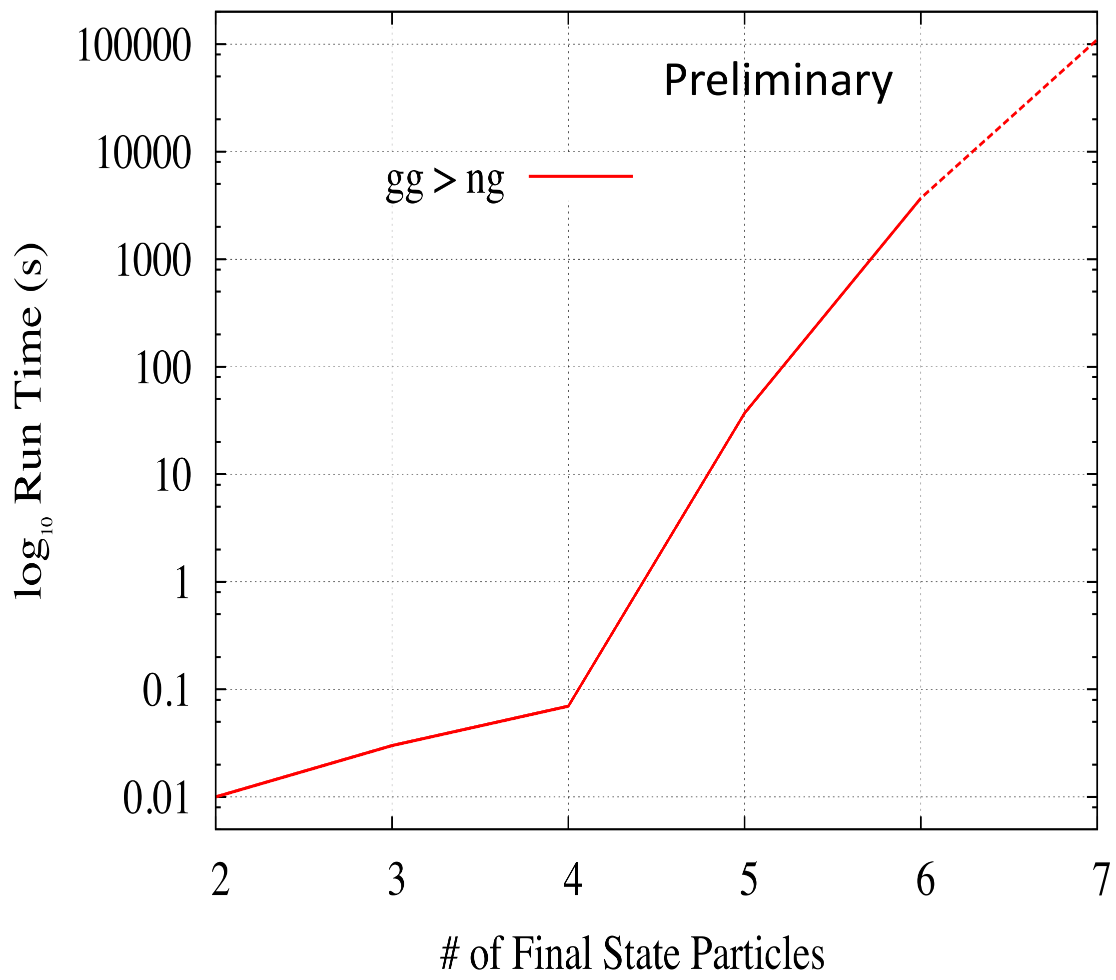
- Modified MG to generate small codes for each Dual Amp.

- Implemented the subroutine which calculate the color matrix from the information of Dual Amps

- Evaluated $\sum_{color} |\mathcal{M}|^2$ by using those codes and the subroutine

Example: gg -> ng

Runtime of the Code



Codes can be compiled and evaluated.

- $gg \rightarrow 5g$
- $gg \rightarrow 6g$
- $gg \rightarrow 7g$ (OK but slow..)

Conclusion

- Generated Codes by MG for multi-parton QCD process cannot be compiled for > 5 final states processes.
- Proposed a way to overcome the limitation by dividing and reducing a code into small files.
- Calculated multi-gluon processes (> 5g)

Future works

- Apply our method to processes with quarks:

  qg > q ng, qq > qq ng

- Phase space integration and event generation

- Speed up by using GPU

Dank u zee

MERCI BEAUCOUP!

감사합니다!

DANKE SCHÖN!

Thank you very much!

Grazie mille!

どうもありがとうございました！

谢谢