

# 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 # ↑
end_coup # Er
```

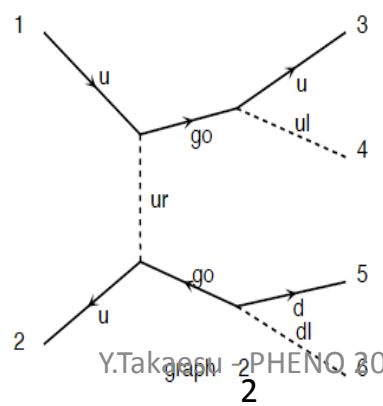
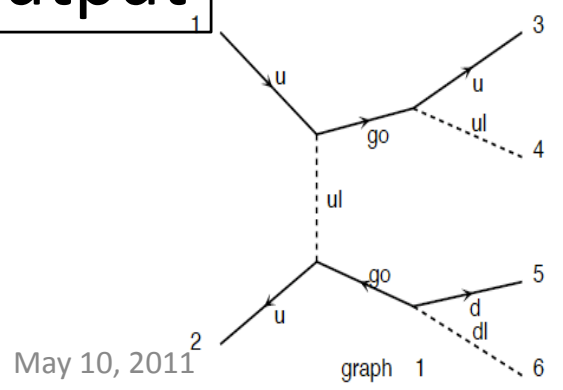
Model + process

MG

Output

Diagram

HELAS Code



```
CALL VXXXX(P(0,1 ),ZERO ,NH(1 ),-1*IC(1 ),W(1,1 ))
CALL VXXXX(P(0,2 ),ZERO ,NH(2 ),-1*IC(2 ),W(1,2 ))
CALL OXXXX(P(0,3 ),ZERO ,NH(3 ),+1*IC(3 ),W(1,3 ))
CALL SXXXX(P(0,4 ),+1*IC(4 ),W(1,4 ))
CALL IXXXX(P(0,5 ),ZERO ,NH(5 ),-1*IC(5 ),W(1,5 ))
CALL SXXXX(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 ))
```

May 10, 2011

Y.Takahashi - PHENIX 2011 &

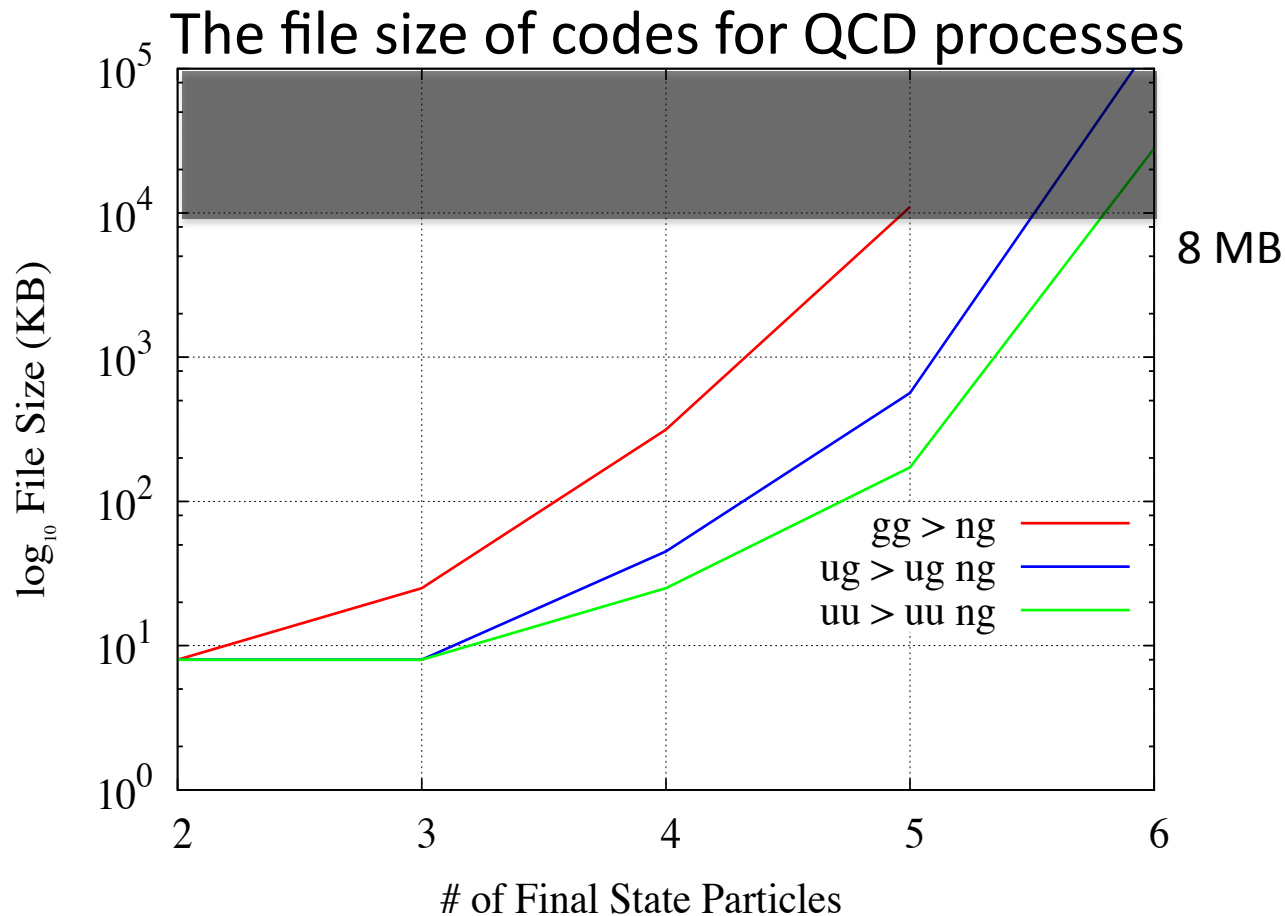
テ

**MG is very useful and popular**

**BUT ...**

# Current Limits of MG

Generated Codes  $> 8\text{MB}$  cannot be compiled.



~~$gg \rightarrow 5g$~~   
 ~~$qg \rightarrow q5g$~~   
 ~~$qq \rightarrow qq4g$~~

**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.

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

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

```

C
C   COLOR DATA
C
C   DATA DENOM(1)/6/
C   DATA (CF(I, 1),I= 1, 6) / 19, -2, -2, -2, -2, 4/
C   1 Tr(1,2,3,4)
C   DATA DENOM(2)/6/
C   DATA (CF(I, 2),I= 1, 6) / -2, 19, -2, 4, -2, -2/
C   .
C   .
C   1 Tr(1,4,2,3)
C   DATA DENOM(6)/6/
C   DATA (CF(I, 6),I= 1, 6) / 4, -2, -2, -2, -2, 19/
C   1 Tr(1,4,3,2)
C
C
C   BEGIN CODE
C
C   -----
C   CALL VXXXXX(P(0,1),ZERO,NHEL(1),-1*IC(1),W(1,1))
C   CALL VXXXXX(P(0,2),ZERO,NHEL(2),-1*IC(2),W(1,2))
C   CALL VXXXXX(P(0,3),ZERO,NHEL(3),+1*IC(3),W(1,3))
C   CALL VXXXXX(P(0,4),ZERO,NHEL(4),+1*IC(4),W(1,4))
C   Amplitude(s) for diagram number 1
C   CALL VVVV4_0(W(1,1),W(1,2),W(1,3),W(1,4),GC_6,AMP(1))
C   CALL VVVV3_0(W(1,1),W(1,2),W(1,3),W(1,4),GC_6,AMP(2))
C   .
C   .
C   CALL VVV1_0(W(1,2),W(1,4),W(1,6),GC_4,AMP(5))
C   CALL VVV1_1(W(1,1),W(1,4),GC_4,ZERO, ZERO, W(1,7))
C   Amplitude(s) for diagram number 4
C   CALL VVV1_0(W(1,2),W(1,3),W(1,7),GC_4,AMP(6))
C   JAMP(1)=+2*(-AMP(3)+AMP(1)-AMP(4)+AMP(6))
C   JAMP(2)=+2*(+AMP(3)+AMP(2)+AMP(4)+AMP(5))
C   JAMP(3)=+2*(-AMP(1)-AMP(2)-AMP(5)-AMP(6))
C   JAMP(4)=+2*(+AMP(3)+AMP(2)+AMP(4)+AMP(5))
C   JAMP(5)=+2*(-AMP(1)-AMP(2)-AMP(5)-AMP(6))
C   JAMP(6)=+2*(-AMP(3)+AMP(1)-AMP(4)+AMP(6))
C
C   MATRIX = 0.D0
C   DO I = 1, NCOLOR
C     ZTEMP = (0.D0,0.D0)
C     DO J = 1, NCOLOR
C       ZTEMP = ZTEMP + CF(J,I)*JAMP(J)
C     ENDDO
C     MATRIX = MATRIX+ZTEMP*DCONJG(JAMP(I))/DENOM(I)
C   ENDDO
  
```

# Overcome the limit

Color Matrix

```

C
C   COLOR DATA
C
C   DATA DENOM(1)/6/
C   DATA (CF(I, 1),I= 1, 6) / 19,  -2,  -2,  -2,  -2,  4/
C   1 Tr(1,2,3,4)
C   DATA DENOM(2)/6/
C   DATA (CF(I, 2),I= 1, 6) /  -2, 19,  -2,  4,  -2,  -2/
C
C
C
C   DATA DENOM(3)/6/
C   DATA (CF(I, 3),I= 1, 6) /  -2, 19,  -2,  4,  -2,  -2/
C
C   DATA DENOM(4)/6/
C   DATA (CF(I, 4),I= 1, 6) /  -2, 19,  -2,  4,  -2,  -2/
C
C   DATA DENOM(5)/6/
C   DATA (CF(I, 5),I= 1, 6) /  -2, 19,  -2,  4,  -2,  -2/
C
C   DATA DENOM(6)/6/
C   DATA (CF(I, 6),I= 1, 6) /  4,  -2,  -2,  -2,  -2,  19/
C   1 Tr(1,4,3,2)

```

Dual Amp.

```

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

```

```

MATRIX = 0.00
DO I = 1, NCOLOR
  ZTEMP = (0.00,0.00)
  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 = 0.00  
DO I = 1, NCOLOR  
  ZTEMP = (0.00,0.00)  
  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)

```
CALL VXXXXX(P...
CALL VXXXXX(P...
CALL VXXXXX(P...
CALL VXXXXX(P...
Amplitude(s) f...
CALL VVVV4_0(W...

CALL VVVV3_0(W...

CALL VVV1_0(W...
CALL VVV1_1(W...
Amplitude(s) f...
CALL VVV1_0(W...
JAMP(1)=+2*(-A...

JAMP(2)=+2*(+...
JAMP(3)=+2*(-...
JAMP(4)=+2*(+...
JAMP(5)=+2*(-...
JAMP(6)=+2*(-...

C(1),W(1,1)
C(2),W(1,2)
C(3),W(1,3)
C(4),W(1,4)

3,1),ZERO,NHEL(1),-1*1
3,2),ZERO,NHEL(2),-1*1
3,3),ZERO,NHEL(3),+1*1
3,4),ZERO,NHEL(4),+1*1
or diagram number 1
(1,1),W(1,2),W(1,3),W(
(1,1),W(1,2),W(1,3),W(
:
L,2),W(1,4),W(1,6),GC_
L,1),W(1,4),GC_4,ZERO.

1,4),GC_6,AMP(1)
1,4),GC_6,AMP(2)

4,AMP(5))
ZERO W(1,7))

JE diagram NUM
L,2),W(1,3),W(1
IP(3)+AMP(1)-AN
MP(3)+AMP(2)+AN
MP(1)-AMP(2)-AN
MP(3)+AMP(2)+AN
MP(1)-AMP(2)-AN
MP(3)+AMP(1)-AN

JE 4
,7),GC_4,AMP(6))
IP(4)+AMP(6))
IP(4)+AMP(5))
IP(5)-AMP(6))
IP(4)+AMP(5))
IP(5)-AMP(6))
IP(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)

**Compile** those files

```
1,4),GC_6,AMP(1)
1,4),GC_6,AMP(2)
4,AMP(5))
ZERO, W(1,2))

CALL WWW_8(W
CALL WWW_1(W
Amplitude(s) f
CALL WWW_8(W
JAMP(1)=2*(-

CALL WWW_8(W
JAMP(2)=2*(+
JAMP(3)=2*(-
JAMP(4)=2*(+
JAMP(5)=2*(-
JAMP(6)=2*(-

C(1),W(1,1)
C(2),W(1,2)
C(3),W(1,3)
C(4),W(1,4)

CALL W0000(P
CALL W0000(P
CALL W0000(P
CALL W0000(P
Amplitude(s) f
CALL WWW_8(W

MATRIX = 0.00
DO I = 1, NCOLOR
  ZTEMP = (0.00,0.00)
  DO J = 1, NCOLOR
    ZTEMP = ZTEMP + CF(J,I)*JAMP(J)
  ENDDO
  MATRIX = MATRIX+ZTEMP*DCONJG(JAMP(I))/DENOM(I)
ENDDO

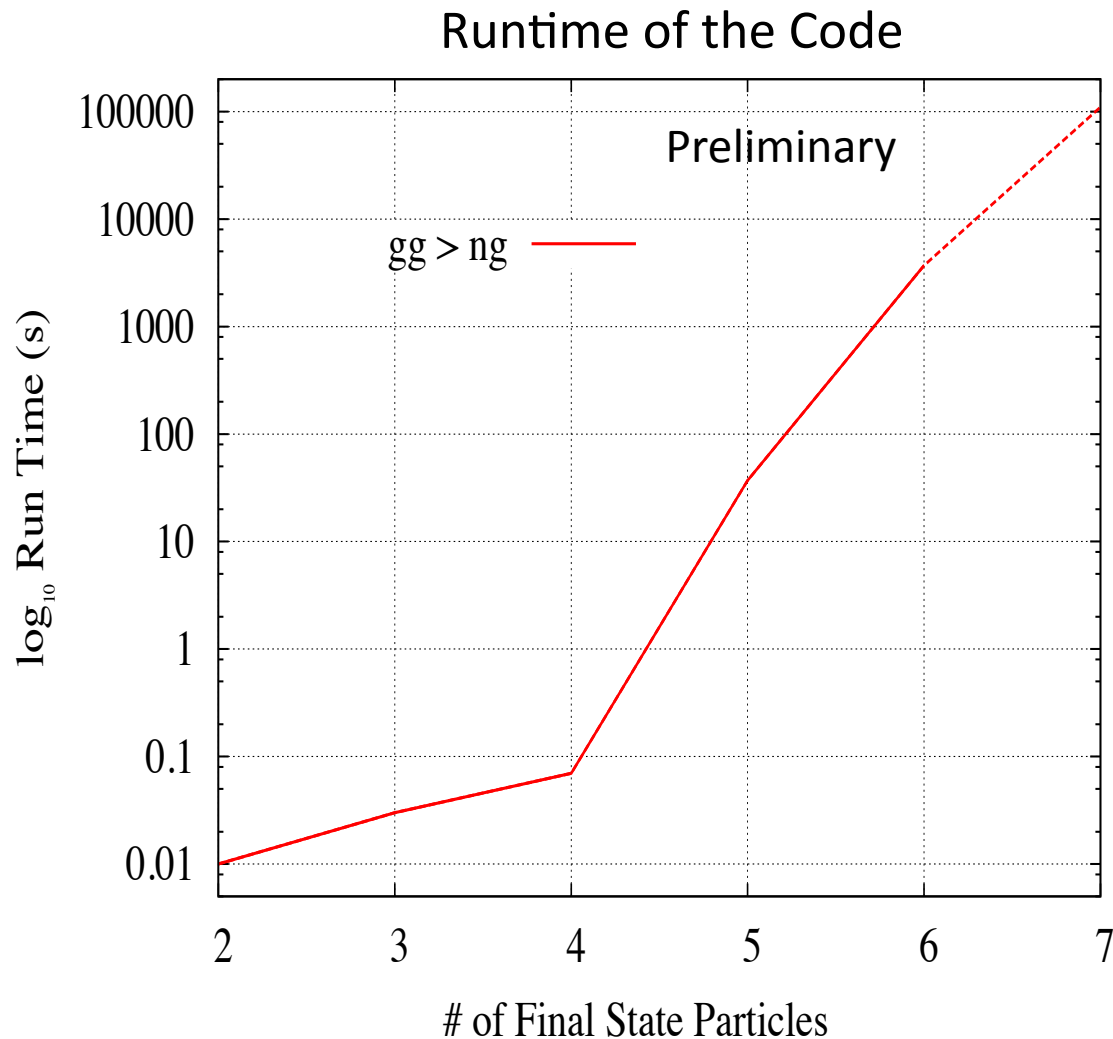
1,2),W(1,3),V(1
P(3)+AMP(1)-A
MP(3)+AMP(2)+A
MP(1)-AMP(2)-A
MP(3)+AMP(2)+A
MP(1)-AMP(2)-A
MP(3)+AMP(1)-A

3,1),ZERO,NHEL(1),-1*
3,2),ZERO,NHEL(2),-1*
3,3),ZERO,NHEL(3),+1*
3,4),ZERO,NHEL(4),+1*
or diagram number 1
(1,1),V(1,2),W(1,3),V
(1,1),V(1,2),W(1,3),V
.
.
.
1,2),V(1,4),W(1,6),GC
1,1),V(1,4),GC_4,ZERO
... ..
```

# 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: $gg \rightarrow ng$



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:  
 $qg \rightarrow qng, qq \rightarrow qqng$
- Phase space integration and event generation
- Speed up by using GPU

Dank u ze

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