Specification of move data

Explain the move data implemented using Gauss code in VirtualLink class.

Reference:

Muramatsu, K., Suzuki, S., & Taguchi, K. (2024). On Matveev-Piergallini moves for branched spines. *arXiv preprint* arXiv:2405.18743.

URL: https://arxiv.org/abs/2405.18743

MP moves

16 types of MP moves are classified into types [A,B,C,D] and [1,2,3,4].

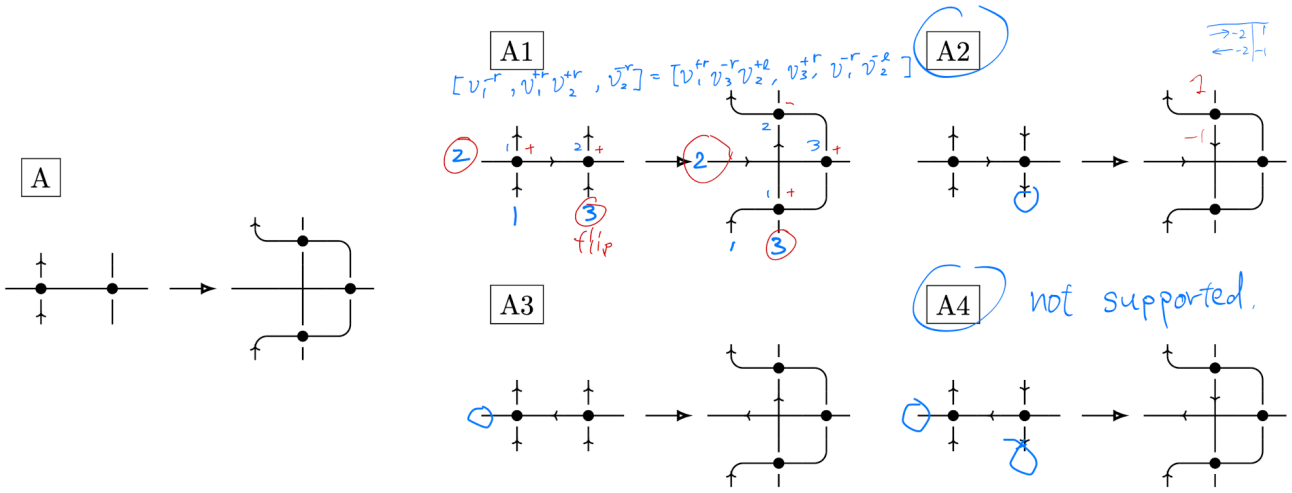

Figure 2.18. MP moves of type A.

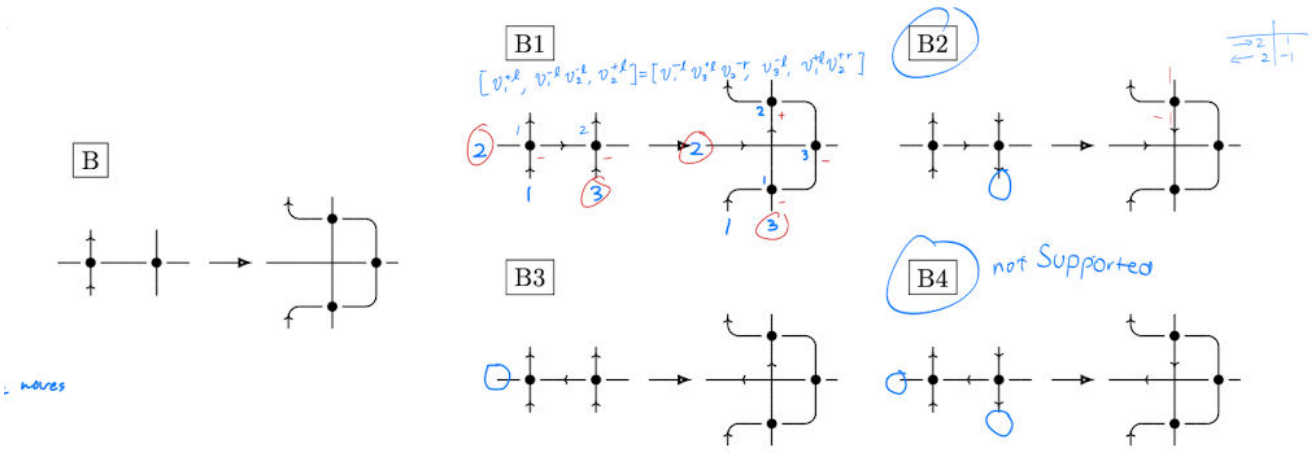

Figure 2.19. MP moves of type B.

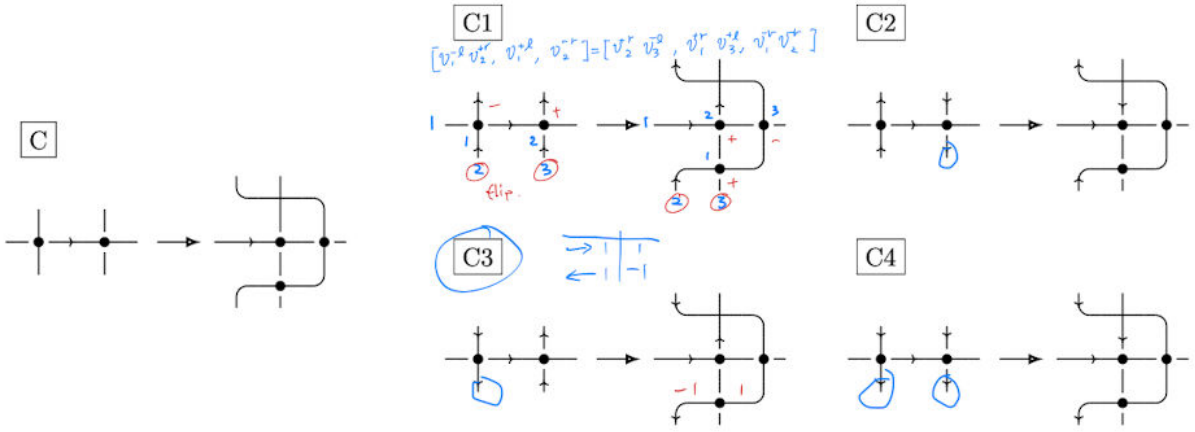

Figure 2.20. MP moves of type C.

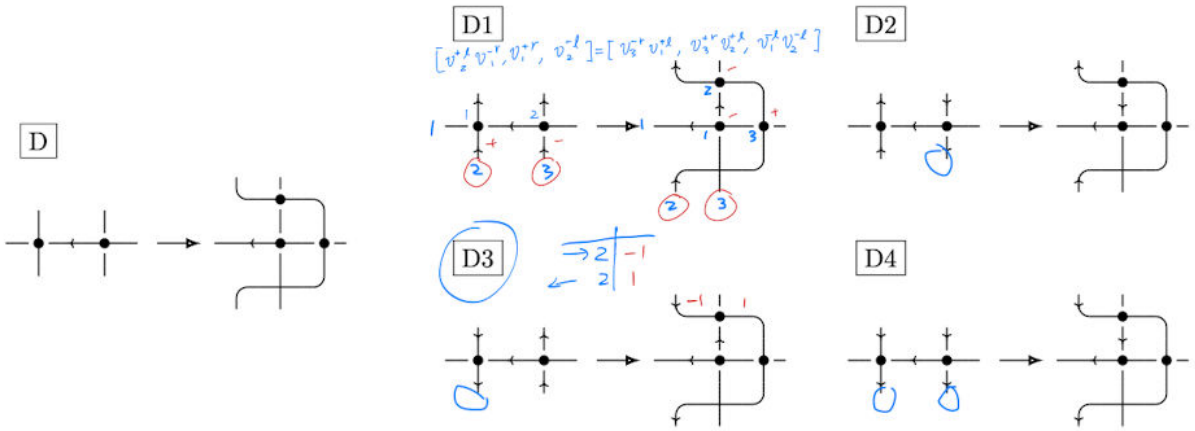

Figure 2.21. MP moves of type D.

`gc` : Gauss Code of strand {[array of signed vertices]}, `ori` : orientations, signatures of crossings

T_MP_L : left hand side of MP move (2 vertices), T_MP_R : right hand side of MP move (3 vertices)

LHS

```
load("MPmoveData.mat")
disp(T_MP_L(:,["param","gc","ori"]))
```

```
    param                 gc                    ori
    _____    ____________________________    _______
    "A1"     {[  -1]}    {[  1 2]}    {[-2]}     1      1
    "B1"     {[   1]}    {[-1 -2]}    {[ 2]}    -1     -1
    "C1"     {[-1 2]}    {[    1]}    {[-2]}    -1      1
    "D1"     {[2 -1]}    {[    1]}    {[-2]}     1     -1
    "A2"     {[  -1]}    {[  1 2]}    {[-2]}     1     -1
    "B2"     {[   1]}    {[-1 -2]}    {[ 2]}    -1      1
    "C2"     {[-1 2]}    {[    1]}    {[-2]}    -1     -1
    "D2"     {[2 -1]}    {[    1]}    {[-2]}     1      1
    "A3"     {[  -1]}    {[  2 1]}    {[-2]}    -1     -1
    "B3"     {[   1]}    {[-2 -1]}    {[ 2]}     1      1
    "C3"     {[-1 2]}    {[    1]}    {[-2]}     1      1
    "D3"     {[2 -1]}    {[    1]}    {[-2]}    -1     -1
    "A4"     {[  -1]}    {[  2 1]}    {[-2]}    -1      1
    "B4"     {[   1]}    {[-2 -1]}    {[ 2]}     1     -1
    "C4"     {[-1 2]}    {[    1]}    {[-2]}     1     -1
    "D4"     {[2 -1]}    {[    1]}    {[-2]}    -1      1
```

RHS

```
disp(T_MP_R(:,["param","gc","ori"]))
```

```
    param                   gc                      ori
    _____    ________________________________    _____________
    "A1"     {[ 1 -3 2]}    {[  3]}    {[-1 -2]}     1    -1     1
    "B1"     {[-1 3 -2]}    {[ -3]}    {[  1 2]}    -1     1    -1
    "C1"     {[   2 -3]}    {[1 3]}    {[-1 -2]}     1     1    -1
    "D1"     {[   -3 1]}    {[3 2]}    {[-1 -2]}    -1    -1     1
    "A2"     {[ 1 -3 2]}    {[  3]}    {[-2 -1]}    -1     1     1
    "B2"     {[-1 3 -2]}    {[ -3]}    {[  2 1]}     1    -1    -1
    "C2"     {[   2 -3]}    {[1 3]}    {[-2 -1]}    -1    -1    -1
    "D2"     {[   -3 1]}    {[3 2]}    {[-2 -1]}     1     1     1
    "A3"     {[ 1 -3 2]}    {[  3]}    {[-1 -2]}     1    -1    -1
    "B3"     {[-1 3 -2]}    {[ -3]}    {[  1 2]}    -1     1     1
    "C3"     {[   2 -3]}    {[3 1]}    {[-1 -2]}    -1     1     1
    "D3"     {[   -3 1]}    {[2 3]}    {[-1 -2]}    -1     1    -1
    "A4"     {[ 1 -3 2]}    {[  3]}    {[-2 -1]}    -1     1    -1
    "B4"     {[-1 3 -2]}    {[ -3]}    {[  2 1]}     1    -1     1
    "C4"     {[   2 -3]}    {[3 1]}    {[-2 -1]}     1    -1     1
    "D4"     {[   -3 1]}    {[2 3]}    {[-2 -1]}     1    -1    -1
```

PS move

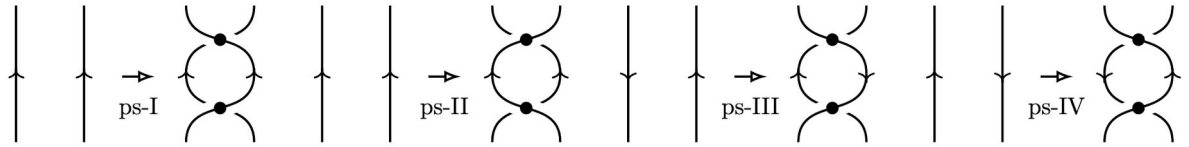

Figure 2.15. Pure sliding move ps-I – ps-IV.

```
load("PSmoveData.mat")
disp(T_PS_R(:,["param","gc","ori"]))
```

```
    param            gc              ori
    _____    _________________    ________
     "1"     {[1 2]}    {[-1 -2]}     1    -1
     "2"     {[1 2]}    {[-1 -2]}    -1     1
     "3"     {[1 2]}    {[-2 -1]}     1    -1
     "4"     {[1 2]}    {[-2 -1]}    -1     1
```

```
disp(T_PS_L(:,["gc","ori"]))
```

```
              gc                    ori
    ____________________________    _________
    {0×0 double}    {0×0 double}    1×0 double
```