1) THE MATLAB SCRIPT BEGINS BY SETTING THE VALUES OF THOSE ARRAY VARIABLES (INITIAL SHELL CASE)

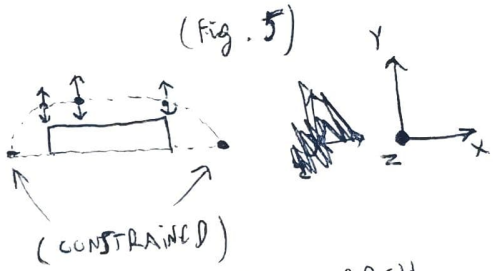
2) CREATE 1ST GENERATION BY RANDOMLY MUTATING THE GENES OF THIS SHELL

a) NOTE FOR THE FIRST GA VERSION, I SUGGEST WORKING ONLY ON 1 COORDINATE OF THE PTS.

→ WE THUS SIMPLIFY WORK TO ONLY 3 PTS. PER CURVE (AS THE 1ST & LAST PTS. SHOULD REMAIN CONSTRAINED)

(Fig. 5)

i.e.,



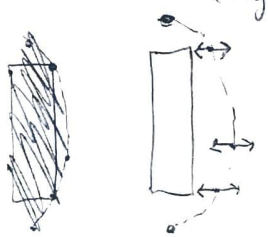
(CONSTRAINED)

b) ACTUALLY, THIS WILL BE A DIFFERENT DIMENSION FOR EACH OF THE CURVES

i) TOP CURVE REMAINS ON X-Y PLANE, THUS ONLY "Y" VARIED (Fig. 5)

ii) BOTTOM CURVE IS ON Z-X PLANE, THUS ONLY "Z" VARIED

(Fig. 6)



(TOP VIEW)

iii) MIDDLE CURVE IS ON THE PLANE DEFINED BY THE "DIAGONAL" EXTRMA OF THE CHASSIS

(Fig. 7)

(SO Y & Z WILL BE VARIED SIMULTANEOUSLY TO MAKE SURE THE PTS. LIE ON THIS LINE:

(Fig. 8)

CHROMOSOMES ≡ CURVES

GENE ≡ POINT ON ONE OF THE CURVES
(CONSISTS OF (X, Y, Z) coordinate vector)

INITIAL CASE: ALL GENES KNOWN
(SPECIFIED BASED ON EXTREMUM PTS. OF INTERIOR CHASSIS)

e.g. (Fig. 1)

(SIDE VIEW)

- I SUGGEST STARTING WITH JUST THESE 5 CURVES (Fig. 2)

(FRONT VIEW)

(THIS SHOULD ENSURE THAT THE SHELL DOES NOT INTERSECT THE RECTILINEAR CHASSIS)

- FURTHERMORE, WE CAN EXPLOIT SYMMETRY TO ONLY WORK ON 3 CURVES (Fig. 3)

- I SUGGEST WE START WITH 5 PTS. AS ON FIG. 1 (ON EACH CURVE)

(LATER WE CAN ADD MORE, IN PARTICULAR IN THE TOP SEGMENT)

e.g.  (Fig. 4)

- SO EACH CHROMOSOME IS A 5×3 MATRIX

$$\begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ \vdots & \vdots & \vdots \\ X_5 & Y_5 & Z_5 \end{bmatrix} \begin{matrix} (\text{POINT } 1) \\ (\text{POINT } 2) \\ \vdots \\ (\text{POINT } 5) \end{matrix}$$

- AND EACH "SHELL" HAS 3 OF THESE MATRICES IN TOTAL

(SHOULD CONCATENATE THEM, OR STORE AS SEPARATE VARIABLES?)

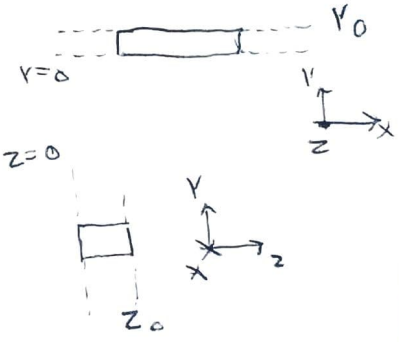
c) SO NOW BOUNDS SPECIFIED & MUTATION HAPPENS WITHIN THEM

CURV1_YBOUND = [Y_0, ...]

CURV 2_Y BOUND = [Y_0, ...]

CURV 3_ZBOUND = [Z_0, ...]

(WHERE



(AND CURV2_ZBOUND IS NOT NEEDED AS IT IS SPECIFIED TO MATCH THE EQN. OF THE LINE ON FIG. 8)

(≡ Z COORD. OF EACH PT. ON CURVE 2 CALCULATED BASED ON THE SPECIFIED Y COORD.)

FOR i=1 TO G % "G" IS THE SIZE OF EACH GENERATION

CURV1_NEW = CURV1 % MAYBE WILL NEED TO APPEND TO A BIG ARRAY INSTEAD?

(.....)

FOR j = 2 to 4

CURV1_NEW [j, 2] = RANDOM (CURV1_Y BOUND, DECIMAL)

CURV 2_ NEW [j, 2] = RANDOM (CURV2_YBOUND, DECIMAL)

CURV 2_ NEW [j, 3] = a · CURV 2_NEW [j, 2] + b

CURV3_NEW [j, 3] = RANDOM (CURV 3_ZBOUND, DECIMAL)

3) EVALUATE FITNESS

FOR i = 1 TO G

a) SAVE COORDS TO TXT/CSV FILE (& SET PARITY BIT TO 0)

b) RUN SOLIDWORKS MACRO (WHICH TAKES THE COORDS. & UPDATES THE MODEL & THEN SETS THE PARITY BIT TO 1 (TO SIGNIFY IT IS DONE))

c) RUN COMSOL CFD USING LIVELINK (WHICH SAVES RESULTS TO ANOTHER CSV/TXT FILE) → (& SET ITS PARITY BIT TO 1?)

d) READ CSV/TXT FILE & APPEND TO ARRAY WITH RESULTS IN MATLAB → (& SET ITS PARITY BIT TO 0?)

4) RANK THE RESULTS (SORT), TAKE "P" (EVEN NO.) BEST TO BE THE PARENTS

a) NEXT GENERATION WILL CONSIST OF:

i) "P" PARENTS (PASSED ON WITHOUT CHANGING)

ACTUALLY NOT THIS (WILL BE STORED IN MEMORY INSTEAD)

ii) "P" CHILDREN CREATED VIA CROSSOVER + MUTATION

iii) "P" CHILDREN ... VIA CROSSOVER ALONE

iv) "G-3P" CHILDREN VIA MUTATION (WITH A DOUBLE MUTATION FREQUENCY)

b) NOTE: IMO iii) is redundant when we already have ii) BUT WE CAN TRY & SEE WHAT WE GET

c) ALSO, FOR iv) COULD USE SOME OF THE NON-BEST RESULTS, BUT THIS MIGHT LEAD TO SLOWER CONVERGENCE

5) CROSSOVER DONE VIA:

a) C_RATE = ... % CROSSOVER FREQUENCY, SET TO BE A FRACTION WITH AN INVERSE THAT IS AN INTEGER (e.g. $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{16}$ ...)

```
FOR i = 2 TO 4
    RAND = RANDOM (1/C_RATE, INTEGER)
    IF RAND == 1
        TEMP = FIRST_CURVE[i]
        FIRST_CURVE[i] = SECOND_CURVE[i]
        SECOND_CURVE[i] = FIRST_CURVE[i]
```

b) LATER CAN DO MORE ELABORATE CROSSOVER SCHEMES & TEST WHICH LEADS TO FASTER CONVERGENCE OF THE GA

6) MUTATION DONE VIA:

M_RATE = ... % MUTATION FREQUENCY (SET SO THAT INVERSE IS AN INTEGER, AS FOR THE C_RATE)

FOR i = 2 to 4

RAND = RANDOM (1/M_RATE , INTEGER)

IF RAND == 1

CURVE [i, 2] = RANDOM (CURVE_BOUND , DECIMAL)

(OR 3, IF "2" IS VARIED)

7) ONCE NEW GENERATION IS READY, REPEAT STEPS 3) & 4) (& THEN ALSO 5) & 6) TO GET NEW GENERATION)

a) TERMINATE WHEN CONVERGENCE CRITERIA HAS BEEN REACHED

b) FOR STARTERS, I THINK I'D ONLY DO ONE CONDITION (NUMBER OF GENERATIONS)

i) STORE THE BEST RESULTS & THEN WE CAN PLOT TO SEE HOW IT CONVERGES OVER TIME

ii) & THUS WE CAN TEST DIFFERENT SETTINGS OF THE GA UNTIL WE ARE HAPPY WITH IT

iii) LATER WE CAN ALSO SET A CONDITION BASED ON THE RATE OF CHANGE BETWEEN SUBSEQUENT GENERATIONS' DRAG VALUES