

Demo

May 25, 2024

```
[ ]: import math
```

Data form AC

```
[ ]: mmo = 0.87
weight = 248.7
alt = 300
ci = 80
```

Get reference data

```
[ ]: alts = [250, 270, 290, 310, 330, 350, 370, 390, 410, 430]
weights = [140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260, 270, 280, 290]

lowerAlt = [_alt for _alt in alts if _alt <= alt][-1]
upperAlt = [_alt for _alt in alts if _alt >= alt][0]
ratioAlt = (alt - lowerAlt) / 20.0
print(lowerAlt, alt, upperAlt, ratioAlt)

lowerWeight = [_weight for _weight in weights if _weight <= weight][-1]
upperWeight = [_weight for _weight in weights if _weight >= weight][0]
ratioWeight = (weight - lowerWeight) / 10.0
print(lowerWeight, weight, upperWeight, ratioWeight)
```

290 300 310 0.5

240 248.7 250 0.8699999999999999

Using Curves directly

```
[ ]: matrix = {}
file = open(f"curve_params_{mmo}.data", 'r')
lines = file.readlines()
file.close()

for i in range(0, len(lines)):
    line = lines[i].split(";")
    for j in range(0, len(weights)):
        try:
            matrix[alts[i]][weights[j]] = line[j].strip().split(",")
```

```

except KeyError:
    matrix[alts[i]] = {}
    matrix[alts[i]][weights[j]] = line[j].strip().split(",")
except IndexError:
    pass

```

```

[ ]: def func_root(a, b, c, CI):
    return float(a) * math.sqrt(CI) + float(b) * CI + float(c)

lowerAltLowerWeightMach = round(func_root(*matrix[lowerAlt][lowerWeight], ci),
    ↪3)
lowerAltUpperWeightMach = round(func_root(*matrix[lowerAlt][upperWeight], ci),
    ↪3)
upperAltLowerWeightMach = round(func_root(*matrix[upperAlt][lowerWeight], ci),
    ↪3)
upperAltUpperWeightMach = round(func_root(*matrix[upperAlt][upperWeight], ci),
    ↪3)
print(lowerAltLowerWeightMach, lowerAltUpperWeightMach,
    ↪upperAltLowerWeightMach, upperAltUpperWeightMach)

ratioedLowerAltMach = round(lowerAltLowerWeightMach + (lowerAltUpperWeightMach
    ↪- lowerAltLowerWeightMach) * ratioWeight, 3)
ratioedUpperAltMach = round(upperAltLowerWeightMach + (upperAltUpperWeightMach
    ↪- upperAltLowerWeightMach) * ratioWeight, 3)
ratioedMach = round(ratioedLowerAltMach + (ratioedUpperAltMach -
    ↪ratioedLowerAltMach) * ratioAlt, 3)
print(ratioedLowerAltMach, ratioedUpperAltMach, ratioedMach)

```

0.821 0.821 0.823 0.824

0.821 0.824 0.823

Using LUTs

```

[ ]: lowerAltMatrix = {}
    upperAltMatrix = {}

    file = open(f"ci2mach_{lowerAlt}_{mmo}.data")
    lines = file.readlines()

    for i in range(0, len(lines)):
        line = lines[i].strip().split(",")
        lowerAltMatrix[weights[i]] = line

    file = open(f"ci2mach_{upperAlt}_{mmo}.data")
    lines = file.readlines()

    for i in range(0, len(lines)):

```

```

line = lines[i].strip().split(",")
upperAltMatrix[weights[i]] = line

```

```

[ ]: lowerAltLowerWeightMach = float(lowerAltMatrix[lowerWeight][ci])
lowerAltUpperWeightMach = float(lowerAltMatrix[upperWeight][ci])
upperAltLowerWeightMach = float(upperAltMatrix[lowerWeight][ci])
upperAltUpperWeightMach = float(upperAltMatrix[upperWeight][ci])
print(lowerAltLowerWeightMach, lowerAltUpperWeightMach,
      ↪upperAltLowerWeightMach, upperAltUpperWeightMach)

ratioedLowerAltMach = round(lowerAltLowerWeightMach + (lowerAltUpperWeightMach
      ↪- lowerAltLowerWeightMach) * ratioWeight, 3)
ratioedUpperAltMach = round(upperAltLowerWeightMach + (upperAltUpperWeightMach
      ↪- upperAltLowerWeightMach) * ratioWeight, 3)
ratioedMach = round(ratioedLowerAltMach + (ratioedUpperAltMach -
      ↪ratioedLowerAltMach) * ratioAlt, 3)
print(ratioedLowerAltMach, ratioedUpperAltMach, ratioedMach)

```

```

0.821 0.821 0.823 0.824
0.821 0.824 0.823

```