AugerBot Calculations

```
Quit
```

Trial 1: 9/19 - 9/26

Modified Francisco Calculations : 10/18 (Do not use!)

Plotting Fx to find U which Balances Forces: 11/1

Plotting Fx to find U which Balances Forces 2 : 11/9

Inside Equation for Thrust: 11/16

Backtracking: 11/27

Francisco' s with Chen' s Coefficients: 11/30 (CORRECT ONES)

```
Quit;
```

Francisco With Chen' s Coefficients

In[1]:=
```
(*Parameters from paper*)
r = 5/1000; w = 10.4; n = 1;

(*LP poppy Fourier coefficients*)
A00 = 0.051; A10 = 0.047; B11 = 0.053; B01 = 0.083;
Bn11 = 0.020; C11 = -0.026; C01 = 0.057; Cn11 = 0; D10 = 0.025;
```

In[3]:=
```
β = (Pi/2) - ϕ; (*ϕ is symbolic, radians*)
αz = Bn11 * Sin[2 * Pi * (-β/Pi)] + A00 * Cos[2 * Pi * 0] +
   B01 * Sin[2 * Pi * 0] + A10 * Cos[2 * Pi * (β/Pi)] + B11 * Sin[2 * Pi * (β/Pi)];
(*Vertical stress per unit depth, N/m^3*)
αx = Cn11 * Cos[2 * Pi * (-β/Pi)] + C01 * Cos[2 * Pi * 0] +
   D10 * Sin[2 * Pi * (β/Pi)] + C11 * Cos[2 * Pi * (β/Pi)];
(*Horizontal stress per unit depth, N/m^3*)

(*Friction coefficients, expressed in terms of ϕ*)
d = 0.05;(*Depth robot buried, 50mm*)
Cn = αx * d; (*N/m^2*)
Ct = αz * d;

Cn /. ϕ → 0
Ct /. ϕ → 0
```

Out[9]= 0.00415

Out[10]= 0.0002

In[11]:=
```
(*Eq 2 - 10/21/17 | Parametrized f Integrated only wrt dθ*)
FranChen[u_] := (2 * Pi * n * r / Cos[ϕ]) *
   ((Cn - Ct) * r * w * Sin[ϕ] * Cos[ϕ] - u * (Ct * Sin[ϕ]^2 + Cn * Cos[ϕ]^2)) /
    Sqrt[u^2 + (r * w)^2];
```

Calculating U/Rw when Fx = 0 for Many ϕ Cases

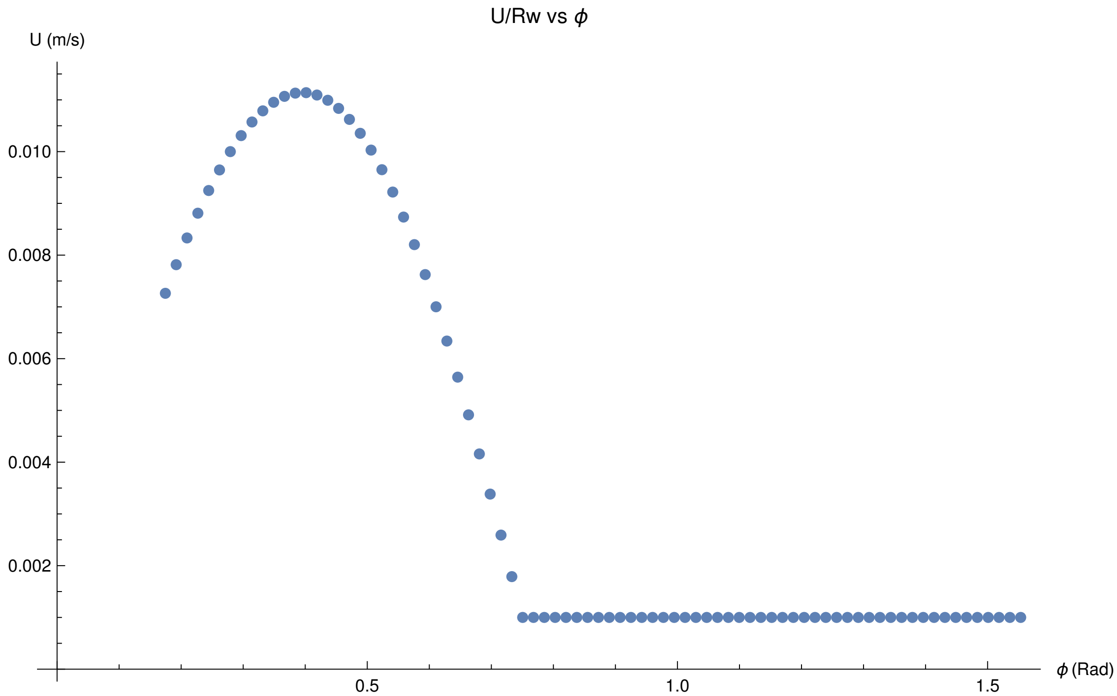
In[12]:=
```
ϕ = 10 * Pi/180 // N; (*Local inclination, radians*)
ϕstore = {};
ustore = {};
Fstore = {};
FMaxstore = {}; (*FxIn when u = 0.001*)
```

In[17]:=
```
(*Finding U intercepts*)
While[ϕ < 90 * Pi / 180,
 (*Print statements*)
 (*Print["Let ϕ = ",ϕ*180/Pi, " deg"];*)
 (*Print@
   Plot[{FranChen[u],0},{u,0,0.2}, PlotLabel→"Inner Fx vs. Helix Velocity U/Rw",
    AxesLabel→{"U (m/s)","Fx (N/m^2)"}, PlotRange→All];*)

 (*Finding U intercept: Newton-Raphson Method*)
 guess = 0.001; (*Reset initial guess*)
 grad = D[FranChen[u], u];
 While[FranChen[guess] > 10^-6, (*Keep iterating until FxIn ≃ 0*)
  gradEval = grad /. u → guess; (*Find FxIn'(guess)*)
  guess = guess - FranChen[guess] / gradEval (*u_{i+1} = u_i - FxIn(u_i)/FxIn'(u_i)*)
 ];
 uint = guess; (*U/Rw intercept found*)

 (*Storing data in arrays*)
 ϕstore = Join[ϕstore, {ϕ}]; (*Storing ϕ in Radians*)
 FMaxstore = Join[FMaxstore, {FranChen[10^-3]}]; (*FxIn(0.001)*)
 ustore = Join[ustore, {uint}]; (*U found when FxIn < 10^-6*)
 Fstore = Join[Fstore, {FranChen[uint]}]; (*FxIn val at u-intercept*)

 ϕ = ϕ + (1 * Pi / 180) (*Increment by 1 deg*)
]
```

Plotting Francisco - Chen Results

In[18]:=
```
ϕFCcol = {ϕstore}ᵀ; uFCcol = {ustore}ᵀ;

(*Helix Translation Speed vs. Local Incline Angle ϕ*)
dataPlot = Join[ϕFCcol, uFCcol, 2];
ListPlot[dataPlot, PlotLabel → "U/Rw vs ϕ",
 AxesLabel → {"ϕ (Rad)", "U (m/s)"}, ImageSize → Large]
```

Out[20]=



Using Chen's Fourier coefficients does result in a maximum speed!!! Only phi values between 0 and 45 ° produce net forward thrust results!

Testing Auger Model with Correct Chen Coefficients

Parameters

For Helix

In[1]:=
```
(*Current param: R = 1.8cm, n = 3.5*)
R = 0.018; (*Screw radius, m*)
n = 1; (*Number of helix turns*)
```

For Material

In[3]:=
```
(*LP poppy Fourier coefficients*)
A00 = 0.051; A10 = 0.047; B11 = 0.053; B01 = 0.083;
Bn11 = 0.020; C11 = -0.026; C01 = 0.057; Cn11 = 0; D10 = 0.025;

β = (Pi/2) - ϕ; (*ϕ is symbolic, radians*)
αz = Bn11 * Sin[2 * Pi * (-β/Pi)] + A00 * Cos[2 * Pi * 0] +
   B01 * Sin[2 * Pi * 0] + A10 * Cos[2 * Pi * (β/Pi)] + B11 * Sin[2 * Pi * (β/Pi)];
(*Vertical stress per unit depth, N/m^3*)
αx = Cn11 * Cos[2 * Pi * (-β/Pi)] + C01 * Cos[2 * Pi * 0] +
   D10 * Sin[2 * Pi * (β/Pi)] + C11 * Cos[2 * Pi * (β/Pi)];
(*Horizontal stress per unit depth, N/m^3*)

d = 0.05;(*Depth robot buried, m*)

(*Friction coefficients, expressed in terms of ϕ*)
Cn = αx * d; (*N/m^2*)
Ct = αz * d;
```

For Motor

In[10]:=
```
w = 2 * 1000 * (2 * Pi) / 3584;(*Angular velocity with 12V source, rad/s*)
```

- (2 ticks/ms)*(1000 ms/s)*(2*Pi rad/rev)*(1 rev/3584 ticks)

Horizontal Thrust Inner Equation

In[11]:=
```
FxIn[U_] := (2 * Pi * n / Cos[ϕ]) *
   (((Cn - Ct) * w * Sin[ϕ] * Cos[ϕ]) * (((R / (2 * w^2)) * Sqrt[(R * w)^2 + U^2]) +
        ((U^2 / (2 * w^3)) * (Log[U] - Log[R * w + Sqrt[(R * w)^2 + U^2]]))) -
     (U * (Ct * Sin[ϕ]^2 + Cn * Cos[ϕ]^2) * (Sqrt[(R * w)^2 + U^2] - U) / w^2));
```

ϕ input must be in radians

Calculating U/Rw when Fx = 0 for Many ϕ Cases

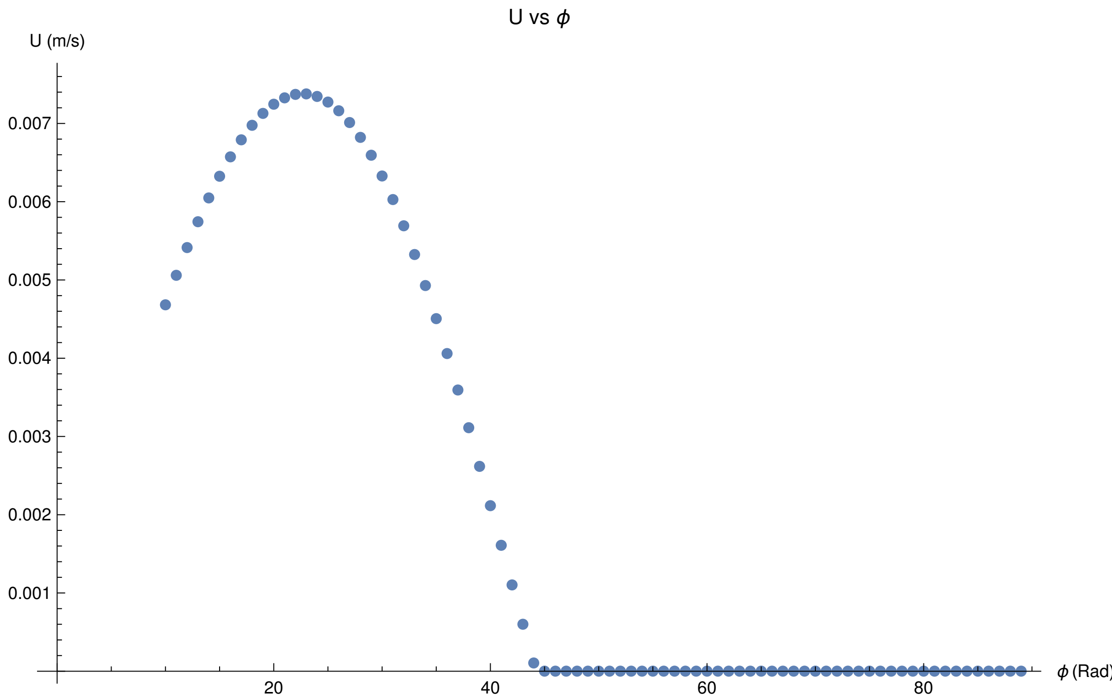
In[43]:=
```
ϕ = 10 * Pi / 180 // N; (*Local inclination, radians*)
ϕstore = {};
Ustore = {};
Fstore = {};
FMaxstore = {}; (*FxIn when u = 0.001*)
```

In[48]:=
```
While[ϕ < 90 * Pi / 180,
 (*Print statements*)
 (*Print["Let ϕ = ",ϕ*180/Pi, " deg"];*)
 (*Print@Plot[{FxIn[U],0},{U,0,0.1}, PlotLabel→"Inner Fx vs. Helix Velocity U/Rw",
    AxesLabel→{"U (m/s)","Fx (N/m^2)"}, PlotRange→All];*)

 (*Finding U intercept: Newton-Raphson Method*)
 guess = 10^-6; (*Reset initial guess*)
 grad = D[FxIn[U], U];

 (*Print[FxIn[guess]];*)

 While[FxIn[guess] > 10^-12, (*Keep iterating until FxIn ≃ 0*)
  gradEval = grad /. U → guess; (*Find FxIn'(guess)*)
  guess = guess - FxIn[guess] / gradEval (*u_{i+1} = u_i - FxIn(u_i)/FxIn'(u_i)*)
 ];
 Uint = guess; (*U intercept found*)

 (*Storing data in arrays*)
 ϕstore = Join[ϕstore, {ϕ}]; (*Storing ϕ in Radians*)
 FMaxstore = Join[FMaxstore, {FxIn[10^-3]}]; (*FxIn(0.001)*)
 Ustore = Join[Ustore, {Uint}]; (*U found when FxIn < 10^-6*)
 Fstore = Join[Fstore, {FxIn[Uint]}]; (*FxIn val at U-intercept*)

 ϕ = ϕ + (1 * Pi / 180) (*Increment by 1 deg*)
]
```

Analysis

In[49]:=
```
ϕcol = {ϕstore}ᵀ;
Ucol = {Ustore}ᵀ;
Fcol = {Fstore}ᵀ;
Fmaxcol = {FMaxstore}ᵀ;

(*Helix Translation Speed vs. Local Incline Angle ϕ*)
dataPlot = Join[ϕcol * 180 / Pi, Ucol, 2];
ListPlot[dataPlot, PlotLabel → "U vs ϕ",
 AxesLabel → {"ϕ (Rad)", "U (m/s)"}, ImageSize → Large]
```

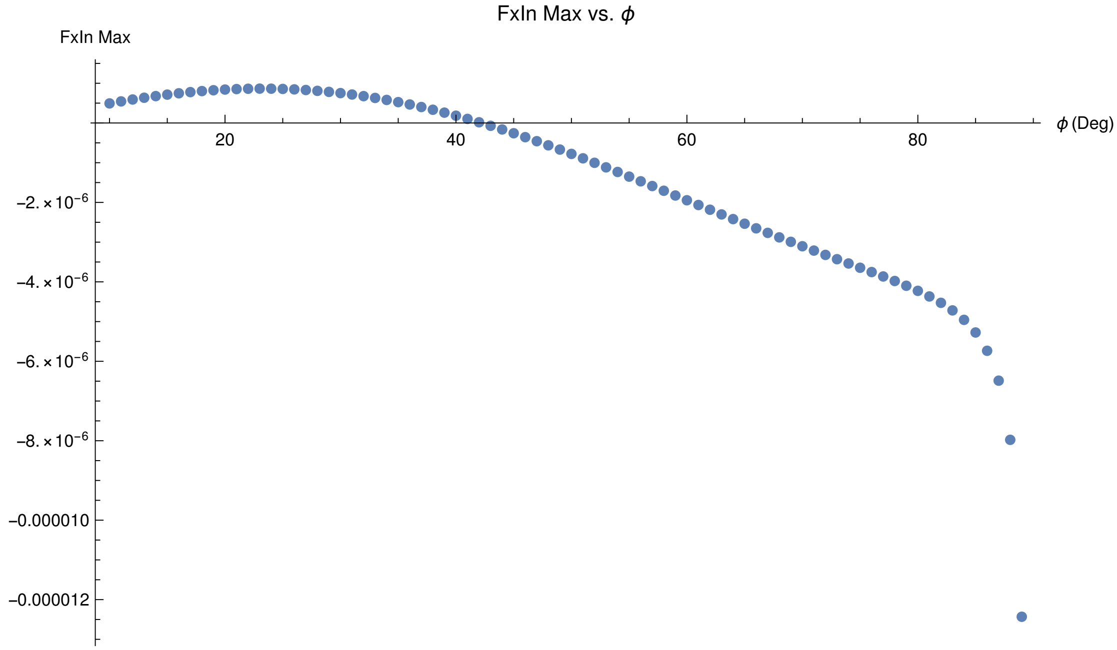
Out[51]=



With Chen's coefficients, the dimensional Fx equation does yield a U maximum speed :) Same deal with Francisco's model, where net forward thrust only exists for $0 < \phi < 40°$

In[52]:=
```
(*FxIn Max vs. Local Incline Angle ϕ*)
dataMax = Join[ϕcol * 180 / Pi, Fmaxcol, 2];
ListPlot[dataMax, PlotLabel → "FxIn Max vs. ϕ",
 AxesLabel → {"ϕ (Deg)", "FxIn Max"}, ImageSize → Large, PlotRange → Full]
```

Out[53]=



In[54]:=
```
(*Checking Data Values: ϕ is left col, U/Rw is right col*)
dataTable = Join[ϕcol * 180 / Pi, Ucol, Fcol, 2];
Grid[Join[{{"ϕ (Deg)", "U", "FxIn[U]"}}, dataTable, 1]]
```

ϕ (Deg)	U	FxIn[U]
10.	0.00468258	1.8561×10^{-15}
11.	0.00505958	3.17729×10^{-15}
12.	0.00541394	5.0654×10^{-15}
13.	0.00574407	7.59502×10^{-15}
14.	0.0060484	1.07902×10^{-14}
15.	0.00632535	1.46078×10^{-14}
16.	0.00657334	1.89275×10^{-14}
17.	0.00679084	2.35511×10^{-14}
18.	0.00697635	2.82127×10^{-14}
19.	0.00712844	3.25989×10^{-14}
20.	0.00724577	3.63793×10^{-14}
21.	0.00732711	3.92417×10^{-14}
22.	0.00737137	4.09297×10^{-14}
23.	0.00737762	4.12758×10^{-14}
24.	0.00734515	4.0226×10^{-14}
25.	0.00727345	3.78512×10^{-14}
26.	0.00716228	3.4342×10^{-14}
27.	0.00701169	2.99876×10^{-14}
28.	0.00682204	2.51412×10^{-14}
29.	0.00659404	2.0176×10^{-14}

Out[55]=

30.	0.00632874	1.5439×10^{-14}
31.	0.00602757	1.12112×10^{-14}
32.	0.00569233	7.67893×10^{-15}
33.	0.00532521	4.92292×10^{-15}
34.	0.00492876	2.92488×10^{-15}
35.	0.00450587	1.58964×10^{-15}
36.	0.00405974	7.76558×10^{-16}
37.	0.00359388	3.32758×10^{-16}
38.	0.00311198	1.20726×10^{-16}
39.	0.00261794	3.51316×10^{-17}
40.	0.00211574	7.50228×10^{-18}
41.	0.00160943	9.99593×10^{-19}
42.	0.00110303	8.16426×10^{-13}
43.	0.000600491	8.02477×10^{-14}
44.	0.000105622	8.52882×10^{-17}
45.	$\frac{1}{1\,000\,000}$	-7.2164×10^{-8}
46.	$\frac{1}{1\,000\,000}$	-1.66505×10^{-7}
47.	$\frac{1}{1\,000\,000}$	-2.63455×10^{-7}
48.	$\frac{1}{1\,000\,000}$	-3.62789×10^{-7}
49.	$\frac{1}{1\,000\,000}$	-4.64271×10^{-7}
50.	$\frac{1}{1\,000\,000}$	-5.67661×10^{-7}
51.	$\frac{1}{1\,000\,000}$	-6.72712×10^{-7}
52.	$\frac{1}{1\,000\,000}$	-7.79171×10^{-7}
53.	$\frac{1}{1\,000\,000}$	-8.86782×10^{-7}
54.	$\frac{1}{1\,000\,000}$	-9.95281×10^{-7}
55.	$\frac{1}{1\,000\,000}$	-1.10441×10^{-6}
56.	$\frac{1}{1\,000\,000}$	-1.21389×10^{-6}
57.	$\frac{1}{1\,000\,000}$	-1.32345×10^{-6}
58.	$\frac{1}{1\,000\,000}$	-1.43284×10^{-6}
59.	$\frac{1}{1\,000\,000}$	-1.54176×10^{-6}
60.	$\frac{1}{1\,000\,000}$	-1.64996×10^{-6}
61.	$\frac{1}{1\,000\,000}$	-1.75716×10^{-6}
62.	$\frac{1}{1\,000\,000}$	-1.8631×10^{-6}
63.	$\frac{1}{1\,000\,000}$	-1.9675×10^{-6}
64.	$\frac{1}{1\,000\,000}$	-2.07011×10^{-6}
65.	$\frac{1}{1\,000\,000}$	-2.17066×10^{-6}
66.	$\frac{1}{1\,000\,000}$	-2.2689×10^{-6}
67.	$\frac{1}{1\,000\,000}$	-2.36458×10^{-6}

68.	$\frac{1}{1000000}$	-2.45746×10^{-6}
69.	$\frac{1}{1000000}$	-2.5473×10^{-6}
70.	$\frac{1}{1000000}$	-2.63387×10^{-6}
71.	$\frac{1}{1000000}$	-2.71694×10^{-6}
72.	$\frac{1}{1000000}$	-2.79631×10^{-6}
73.	$\frac{1}{1000000}$	-2.87177×10^{-6}
74.	$\frac{1}{1000000}$	-2.94313×10^{-6}
75.	$\frac{1}{1000000}$	-3.01019×10^{-6}
76.	$\frac{1}{1000000}$	-3.0728×10^{-6}
77.	$\frac{1}{1000000}$	-3.13078×10^{-6}
78.	$\frac{1}{1000000}$	-3.18398×10^{-6}
79.	$\frac{1}{1000000}$	-3.23228×10^{-6}
80.	$\frac{1}{1000000}$	-3.27554×10^{-6}
81.	$\frac{1}{1000000}$	-3.31365×10^{-6}
82.	$\frac{1}{1000000}$	-3.34652×10^{-6}
83.	$\frac{1}{1000000}$	-3.37407×10^{-6}
84.	$\frac{1}{1000000}$	-3.39624×10^{-6}
85.	$\frac{1}{1000000}$	-3.41301×10^{-6}
86.	$\frac{1}{1000000}$	-3.42439×10^{-6}
87.	$\frac{1}{1000000}$	-3.4305×10^{-6}
88.	$\frac{1}{1000000}$	-3.43177×10^{-6}
89.	$\frac{1}{1000000}$	-3.43047×10^{-6}

$\phi = 23°$ yields the highest value for U (m/s).