Dynamic Open-Source Velocity Prediction Program (VPP) for large Merchant Sailing Vessels

Sergio Perez, USMMA Marine Engineering Department

Before building a new sailing vessel with an unique design: how fast will it go?
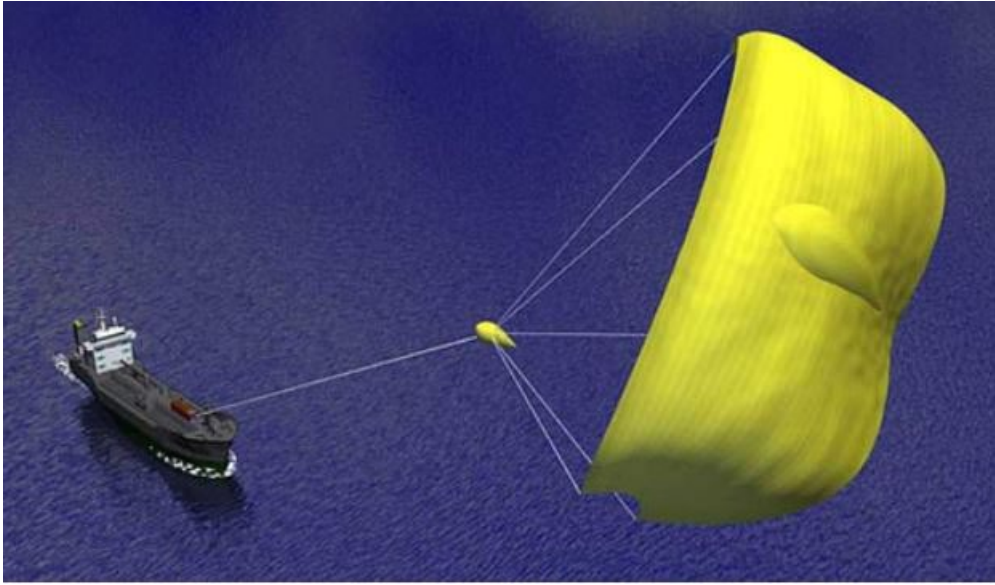
Image from gcaptain.com

Image from interestingengineering.com

Image from tradewindnews.com

Image from gcaptain.com

Solar Sailor

Goal: development of an open- source tool (VPP or Velocity Prediction Program) for predicting the speed of large sailing cargo-carriers.

- VPP's have been around since the early 1970's.
    - Woodward (MARAD, 1975) predicted performance of 15, 30 and 45,000 CDWT sailing vessels.
    - Kerwin handicapped sailboat races (1975).
    - Larsson, Principles of Yacht Design, 4th ed 2014
- Useful for evaluating innovative designs
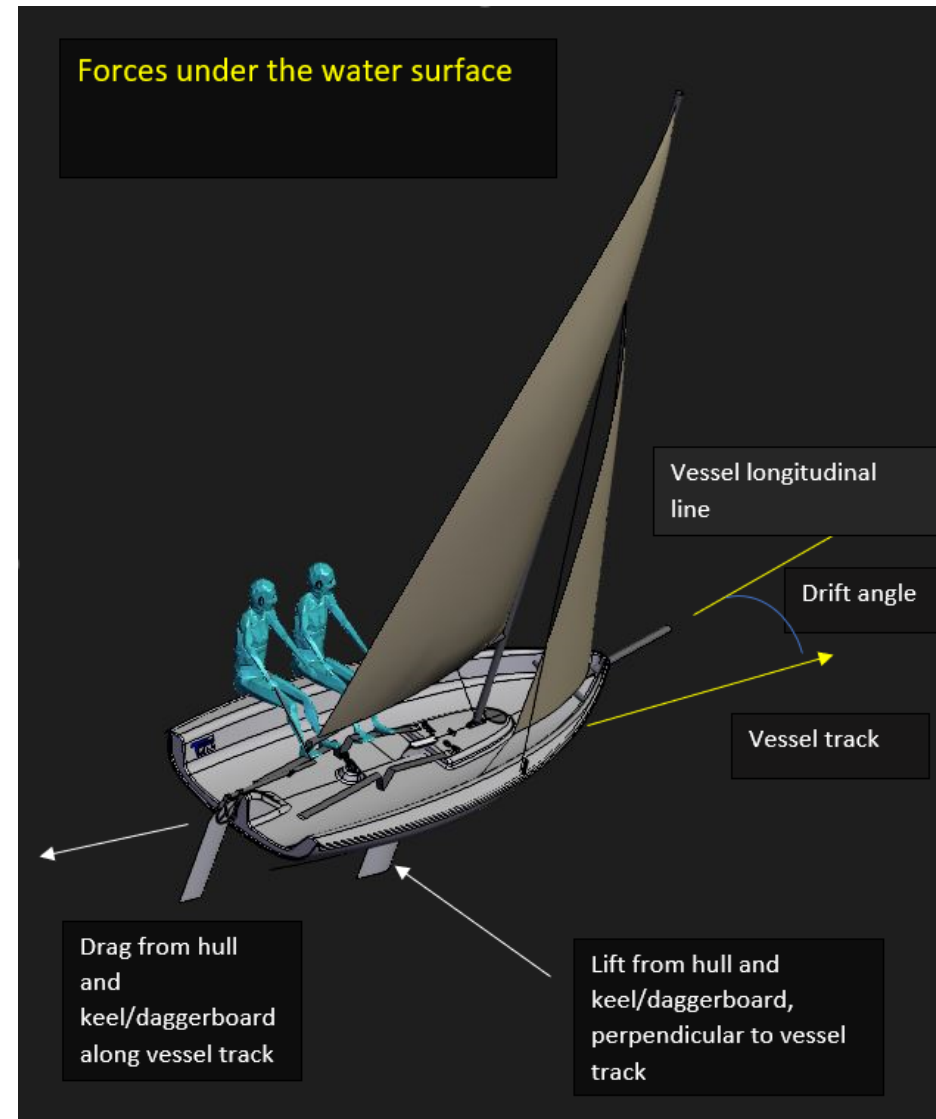- Few open-source VPP's exist. Lack of documentation. Companies guard their intellectual property.
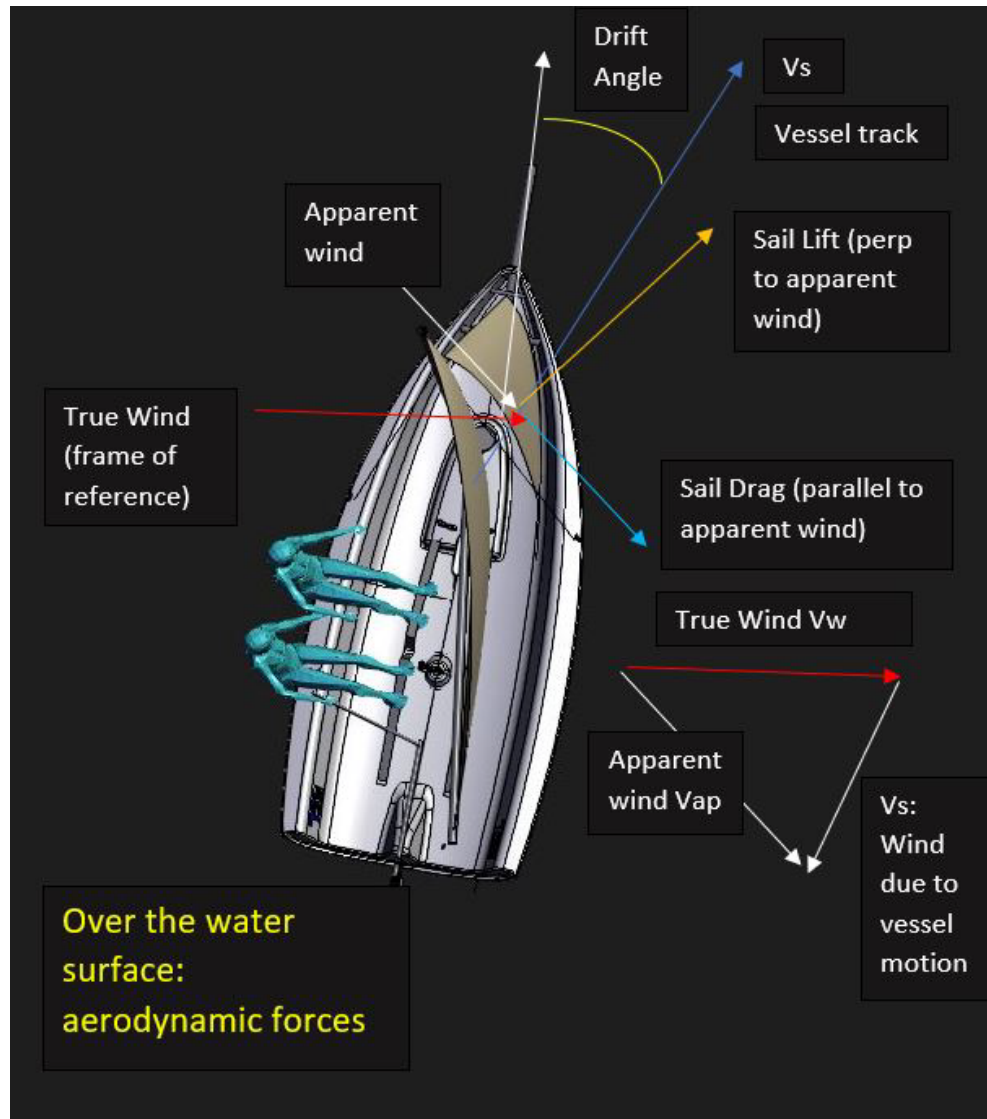
What data are needed to use a sailing vessel VPP?

- Vessel drag and lift coefficients as function of drift and heel angles (vessel drifts downwind at an angle to its track through water, and vessel heels (rolls) due to wind)

- Sail drag and lift coefficients as function of vessel angle to wind and sail angle of attack

How to get data?

- Towing tank and wind tunnel tests
- CFD (Computational Fluid Dynamics)
  - RANS
  - LES
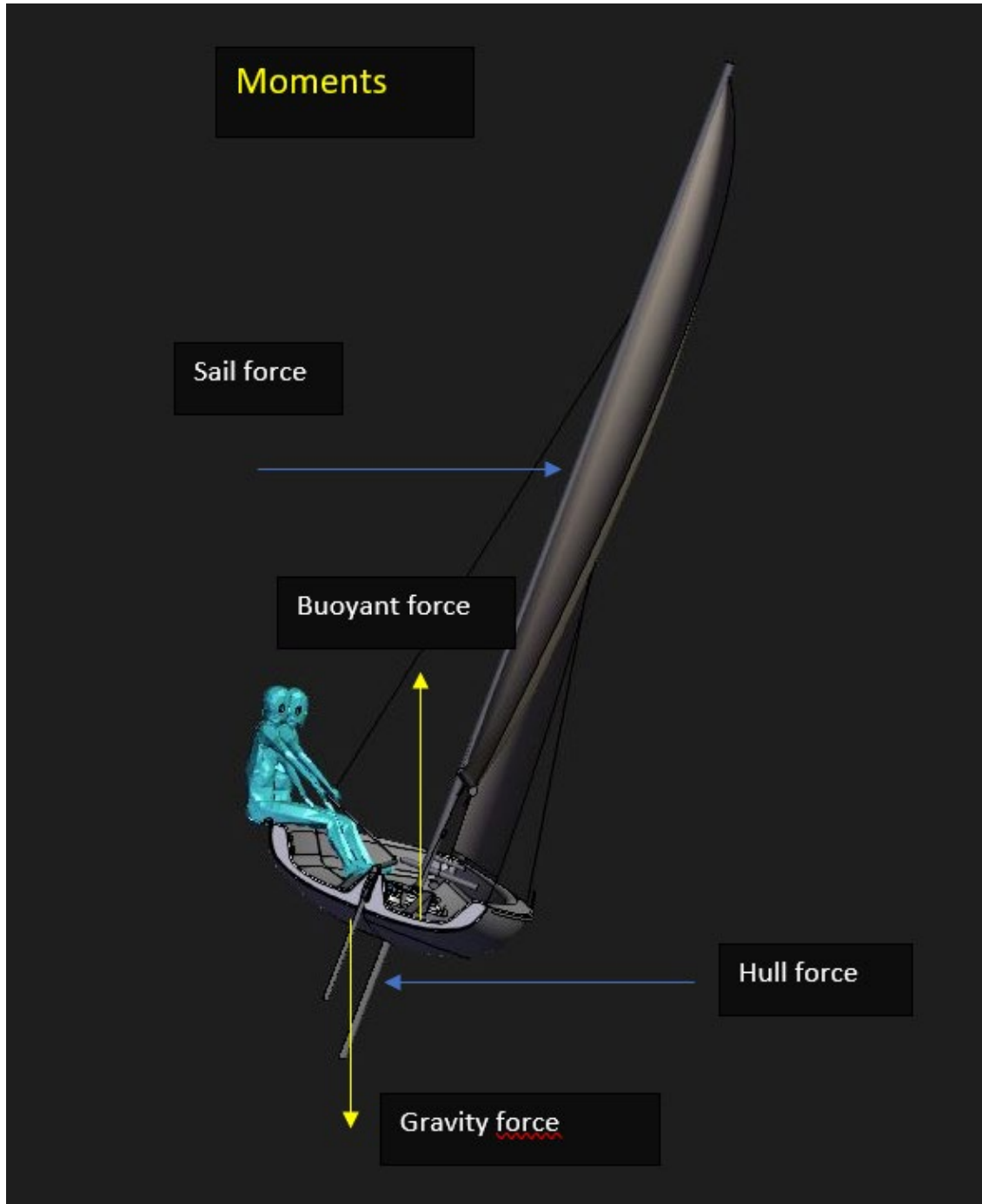 - DNS

Brief review of forces on a sailing vessel:

Drift Angle

Vs

Vessel track

Apparent wind

Sail Lift (perp to apparent wind)

True Wind (frame of reference)

Sail Drag (parallel to apparent wind)

True Wind Vw

Apparent wind Vap

Vs: Wind due to vessel motion

Over the water surface:
aerodynamic forces

Forces under the water surface

Vessel longitudinal line

Drift angle

Vessel track

Drag from hull and keel/daggerboard along vessel track

Lift from hull and keel/daggerboard, perpendicular to vessel track

TWO TYPES OF VPP's

Equilibrium: Vessel assumed moving at constant speed. All forces assumed balanced.
- In simplest form:
    - Sum of forces along vessel track = 0
    - Sum of forces perpendicular to track = 0
    - Sum of moments along ship longitudinal axis = 0
- Result is a set of simultaneous equations. Solved by computer using iteration (intelligent guessing).
- Can be difficult to solve if a good initial guess of vessel velocity is not available.
- The most frequently used type of VPP.

Dynamic: Forces on vessel are not in equilibrium.
- Vessel stationary at time = 0 and allowed to accelerate from force of wind. Forces act for a small time (time step).
- The acceleration in x and y directions are calculated, and the new velocity and direction are calculated at the end of the time step by Sum of forces = m * a
- Process is repeated (time-marching) until equilibrium of forces is reached, resulting in the steady velocity and track of the vessel
- Does not require solving equations simultaneously. Easier to program.
- Can be used to solve for unsteady performance as well as steady speed.
- Can require very small time steps to be accurate, which might take longer to run than equilibrium methods. Not too much of a factor with today's computers.

Dynamic open-source VPP for merchant vessels:

Assumptions used:
- Angle of heel of vessel is less than 10 degrees
- Vessel and sail characteristics (lift and drag) are unaffected by heel
- Vessel is balanced in yaw

Other information:
- Written in Python language
- 4th order Runge-Kutta method used for marching forward in time
- Sail lift and drag information must be input into the program, in the form of Coefficients of Lift and Drag (Cl and Cd).
- Hull lift and drag information must be input into the program in the form of Cl and Cd as a function of drift angle.
- Vessel information must be input: GM, mass, length, CG location, sail CP
- A time step is selected by the user after some initial trials: running a simulation with smaller and smaller time steps until there is little or no difference in the solution.
- A drift angle and wind speed are selected by the program user. The program then solves for vessel velocity and resulting orientation into the wind.
- The program automatically cycles through a series of sail angle of attacks (some efficient, some not) for each apparent wind angle to the vessel. Many solutions are possible, and the maximum is selected to ascertain the fastest speed of the vessel.

Vessel Modeled: 525 ft long, 15,000 CDWT bulk carrier
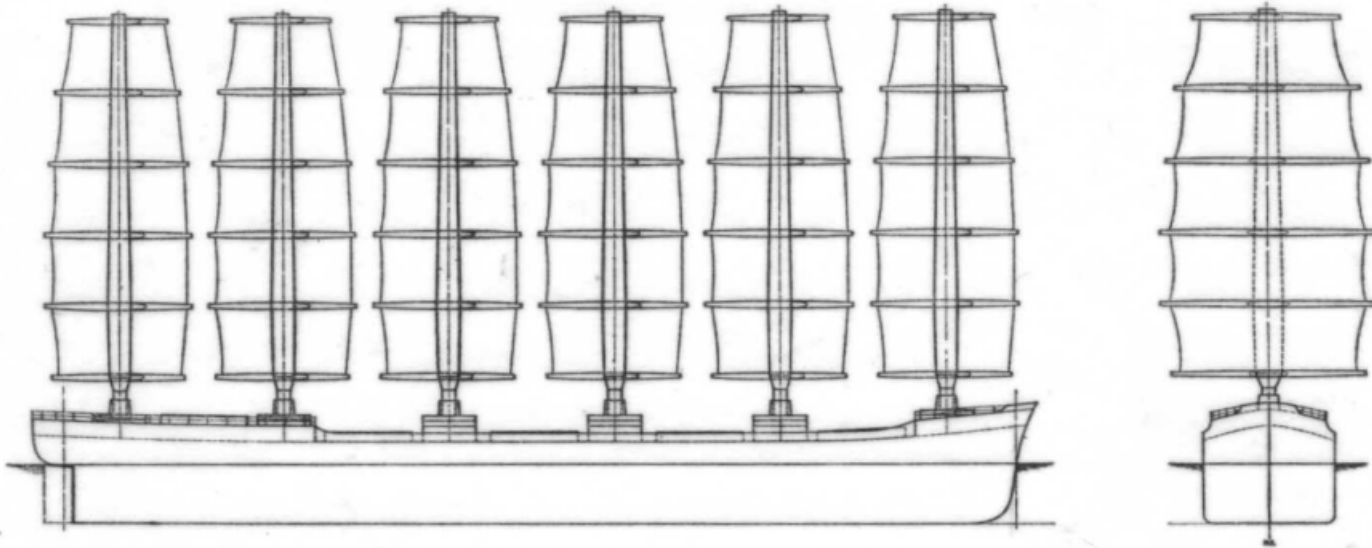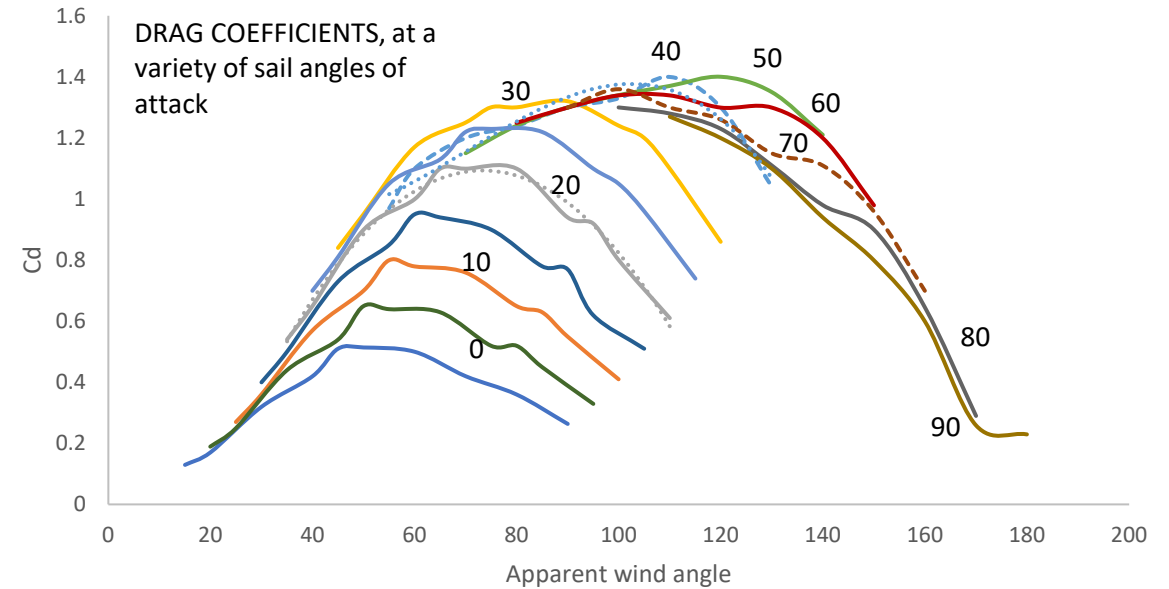

Sails: Prolls Dynarig Design (1967), 88,000 sq. ft
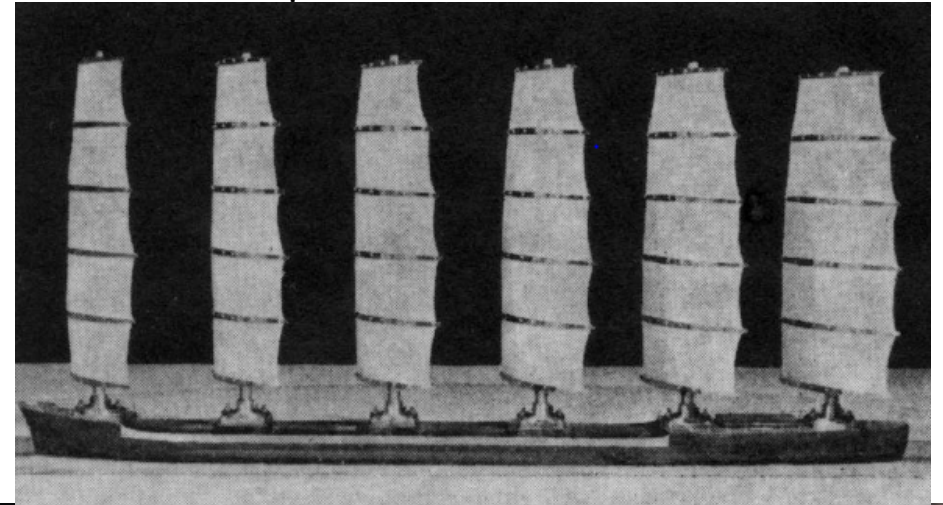
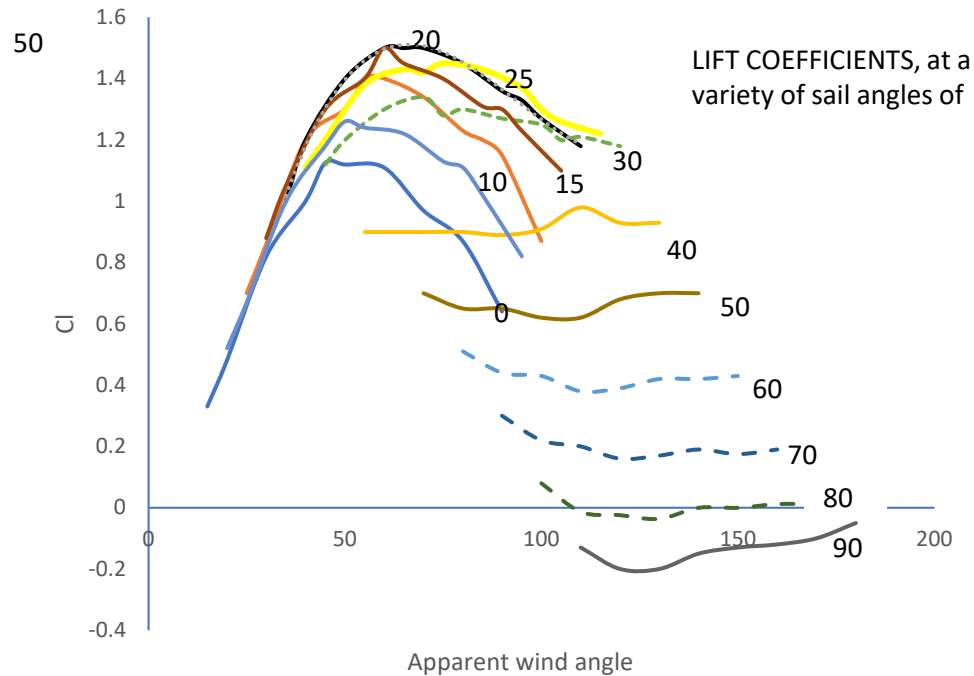

Black and white images from Wagner, Wind tunnel tests for a six-masted sailing vessel by Prölss, 1967 :

Color image from Youtube, Maltese Falcon

Sail data from Wagner's paper (Wind tunnel tests for a six-masted sailing vessel by Prölss, 1967, translated from German in www.jmwe.org)

LIFT COEFFICIENTS, at a variety of sail angles of

DRAG COEFFICIENTS, at a variety of sail angles of attack

Each curve above was fit to a polynomial, for example, lift an drag coefficients as function of apparent wind (thetaap) at 10 degree sail angle of attack (LOTS OF CURVE FITTING!):

```
cl = 2.370286E-06* thetaap**3 - 8.721324E-04* thetaap**2 + 8.053825E-02* thetaap - 8.117323E-01
cd = 8.205687E-08* thetaap**4 - 1.928333E-05* thetaap**3 + 1.256630E-03* thetaap**2 - 1.094621E-02* thetaap + 2.100036E-02
```
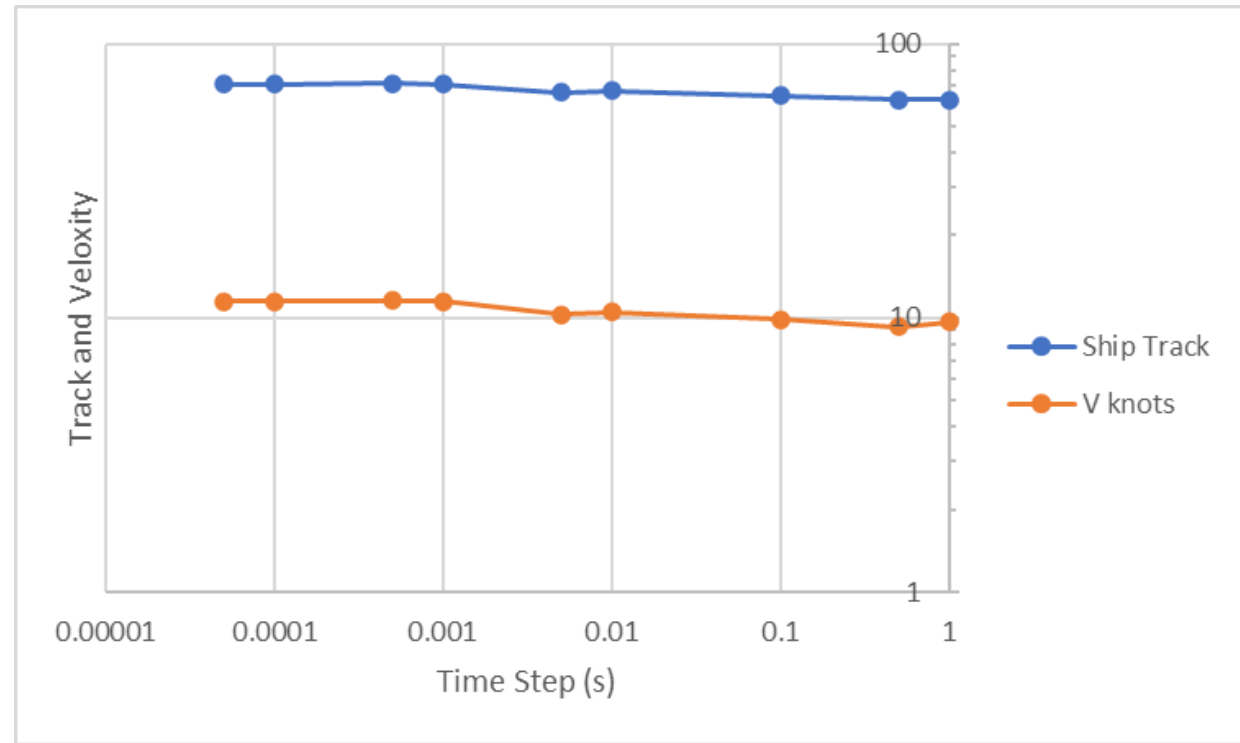
Same done for hull lift and drag forces (data from Wagner's Angled towing tests for a sailing vessel hull with and without bar keel and for the "Mariner", available in English at www.jmwe.org)

How big a time step to use?

- Results shown for drift angle of 17 degrees

Time step of 0.001 s seem appropriate

Calculation time:
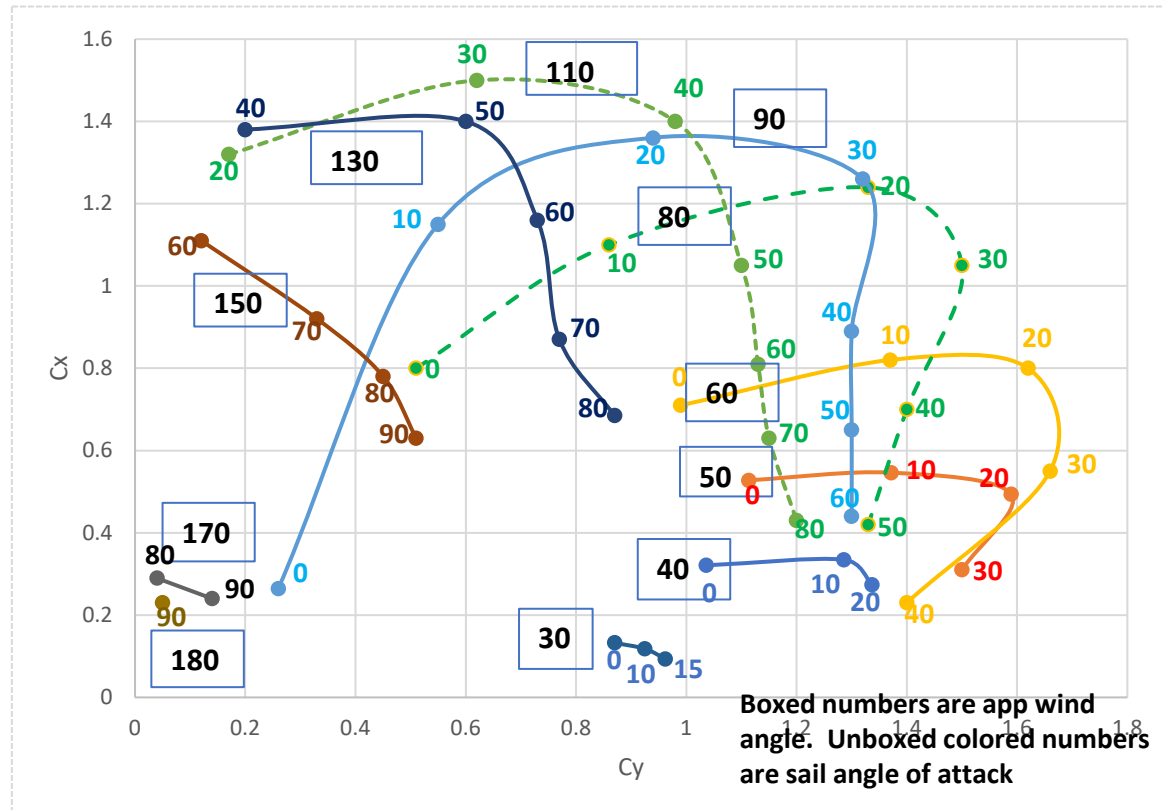About 2-3 minutes per angle of attack, about twelve angles of attack tested max.

Best angle at each heading into wind found by trial and error.

Results (Beaufort = 7)

BF 7

Useful advice/conclusions:

1) Make a plot as below. Cx is force coefficient in direction of vessel track, Cy is perpendicular to the vessel track. Shows where to look for highest vessel speed.

2) Use programming language with a15 digit precision:

- 1+ 0.00000000000001 = 1.000000000000001 ?

- Most modern languages no problem
- Originally written in BASIC language

3) When making curve fits to Cl and Cd data, use as small exponents as possible in curve fit.

4) 1st order Runge-Kutta (Euler's Method) resulted in oscillations of results. Need greater accuracy of 4th order.

5) Cultivate a mindset that promotes calm and harmony