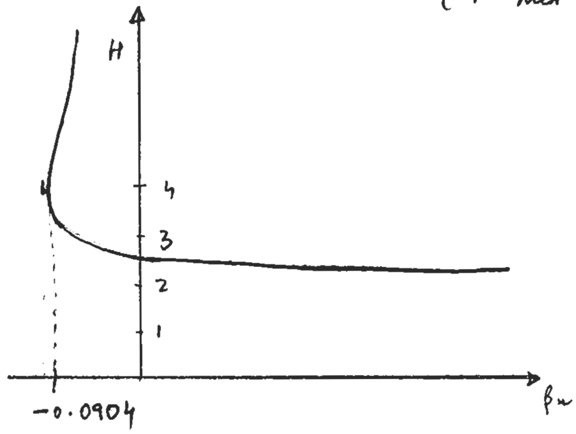
1,2> β_u is minimum when $H=4$ $(\beta_{u_{min}} = -0.0904)$



No solution exists for $\beta_{u_{spec}} < \beta_{u_{min}}$, so Newton method oscillates back and forth without converging

3> $$U_w = -\frac{\partial \psi}{\partial x}\Big|_{y=0} = -\frac{\partial}{\partial \xi}(MF) + \frac{\eta}{\Delta}\cdot\cancelto{0}{\frac{d\Delta}{d\xi}}\frac{\partial}{\partial \eta}(MF) = -M\frac{\partial F}{\partial \xi} - F\frac{dM}{d\xi} = -\frac{M}{\xi}\beta_M F$$

$\frac{\partial F}{\partial \xi} = 0 \rightarrow$ similar flow

$$V_w \equiv \frac{\xi}{M} V_{wall} = \frac{\xi}{\Delta}\frac{V_{wall}}{U_e} = -\beta_M F(0) = -\left(\frac{\beta_u + 1}{2}\right) F(0)$$

Modified b.Cs:

a> $U_w = U_{w\,spec} \Rightarrow R_{BC_2} \equiv U_1 - U_{w\,spec} = 0$

b> $V_w = V_{w\,spec} \Rightarrow R_{BC_1} \equiv \frac{\beta_u + 1}{2} F(0) - V_{w\,spec} = 0$

3a> Solving for $H(\beta_u)$ for a range of U_w produces $U_w = 0.415$ when $\beta_u = -0.18$

3b> Solving for $H(\beta_u)$ for a range of V_w produces $V_w = -0.345$ when $\beta_u = -0.18$.

A direct approach would be to make U_w and V_w global variables. Augment the right hand side with another column vector $\{\partial \vec{R}/\partial U_w\}$

new column

$$\Rightarrow \left[\frac{\partial \vec{R}}{\partial(F,U,S)}\right]\begin{Bmatrix}\delta F\\ \delta U\\ \delta S\end{Bmatrix} = -\{\vec{R}\} - \delta U_w\left\{\frac{\partial \vec{R}}{\partial U_w}\right\}, \quad \left\{\frac{\partial \vec{R}}{\partial U_w}\right\} = \begin{Bmatrix}-1\\ 0\\ \vdots\\ \vdots\end{Bmatrix} \leftarrow R_{BC2}$$

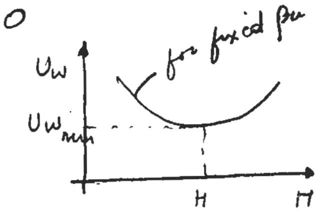
$$\left[\partial R_p/\partial(F,U,S)\right]\{\delta U_w\} = \{-R_p\} - \delta U_w\{\partial R_p/\partial U_w\}, \quad \{\partial R_p/\partial U_w\} = \{0\}$$

new variable

Solve for δF_i, δU_i, δS_i, ∂U_w and update

Since $\beta_u = f(U_w, H) = -0.18$, $\quad \delta\beta_u = \frac{\partial \beta_u}{\partial H}dH + \frac{\partial \beta_u}{\partial U_w}dU_w = 0$

$$\Rightarrow \frac{dU_w}{dH} = \frac{-\partial\beta_u/\partial H}{\partial\beta_u/\partial U_w}$$



which is 0 when U_w is a minimum, since $\partial\beta_u/\partial H = 0$.
Same approach can be applied to calculate V_w as a global variable

3c) $Re = 10^6 = \frac{U_\infty c}{\nu}$, $\quad u_{wall} = U_{wall}\cdot U_e \approx U_{wall}\cdot U_\infty = 0.415\,U_\infty$ (large, 40% of U_∞)

$$v_{wall} = V\,U_e\,\frac{\Delta}{c} \approx -0.345\,U_\infty\sqrt{\frac{\nu}{U_\infty c}} = -0.000345\,U_\infty$$

$\therefore$ V_w is more reasonable
$\Rightarrow$ suction is more feasible / practical

4) Boundary layer suction ($v_w \neq 0$, $u_w = 0$) is applied on the upper surface of an airfoil to suppress separation

$$u_e(x) = U_\infty (x/c)^{-0.09}$$

a) For similarity, V_w = constant

$$= \frac{v_w\,\xi}{m} \quad \text{or} \quad v_w = V_w \cdot \frac{m}{\xi} = \text{Const.}\,\xi^{(\beta_m-1)/2}$$

On the airfoil $\quad v_w(x) = \text{Const.}\left(\frac{x}{c}\right)^{\frac{(\beta_u - 1)}{2}} = \text{const}\left(\frac{x}{c}\right)^{-0.545}$ //

4b) $\beta_u = -0.09$, $Re_c = 10^6$

$$\frac{\theta}{c}\Big|_{t.e} = \theta_1 / \sqrt{Re_c} \quad , \quad C_{D_{prof}} = 2\left[\left(\frac{\theta}{c}\right)_u + (\theta/c)_L\right]_{t.e}$$

$$= \frac{2}{\sqrt{Re_c}}\left[\theta_{1u}(V_w) + \theta_{1L}\right] \quad , \quad \theta_{1L} = 0.587 \;\; (a = \beta_u = 0.09)$$

On the upper surface

$$\frac{V_w(x)}{U_\infty} = \frac{V_w}{\sqrt{Re_c}} \cdot (x/c)^{-1/2} \cdot \left(\frac{u_e}{U_\infty}\right)^{1/2}$$

$$= \frac{V_w}{\sqrt{Re_c}} (x/c)^{-1/2} (x/c)^{\beta_u/2} = \frac{V_w}{\sqrt{Re_c}} (x/c)^{-\frac{1}{2}(\beta_u - 1)}$$

Define suction coefficient

$$C_S = \int_0^c \frac{\rho V_w \, dx}{\rho U_\infty c} = \int_0^1 \left(\frac{V_w}{U_\infty}\right) d(x/c) = \frac{V_w}{\sqrt{Re_c}} \left(\frac{\beta_u + 1}{2}\right)$$

The power required to pump the suctioned boundary layer fluid back to free stream stagnation pressure

$$P_{suction} = \int_0^c \frac{\Delta P_t}{\rho} \cdot d\dot{m}_{suction} = \int_0^c \left(\frac{P_{t\infty} - P_e}{\rho}\right) d\dot{m}_{suction}$$

$$P_{t\infty} - P_{te} = \frac{1}{2}\rho u_e^2$$

$$\therefore P_{suction} = \int_0^c \frac{1}{2} u_e^2 \, d\dot{m}_{suction} \qquad (>0)$$

$$C_{P_{suction}} = \frac{P_{suction}}{\frac{1}{2}\rho U_\infty^3 \cdot c} = \int_0^1 \left(\frac{u_e}{U_\infty}\right)^2 \left(\frac{|V_w|}{U_\infty}\right) \cdot d(x/c) \qquad (>0)$$

$$= \int_0^1 (x/c)^{2\beta_u} \cdot \frac{|V_w|}{\sqrt{Re_c}} \cdot (x/c)^{\frac{1}{2}(\beta_u - 1)} d(x/c)$$

$$= \frac{|V_w|}{\sqrt{Re_c}} \int_0^1 (x/c)^{\frac{5\beta_u - 1}{2}} d(x/c) = \frac{|V_w|}{\sqrt{Re_c}} \cdot \left(\frac{5\beta_u + 1}{2}\right)$$

$$\therefore \quad C_{P\,suction} = |C_a| \left(\frac{5\beta_u + 1}{\beta_u + 1} \right)$$

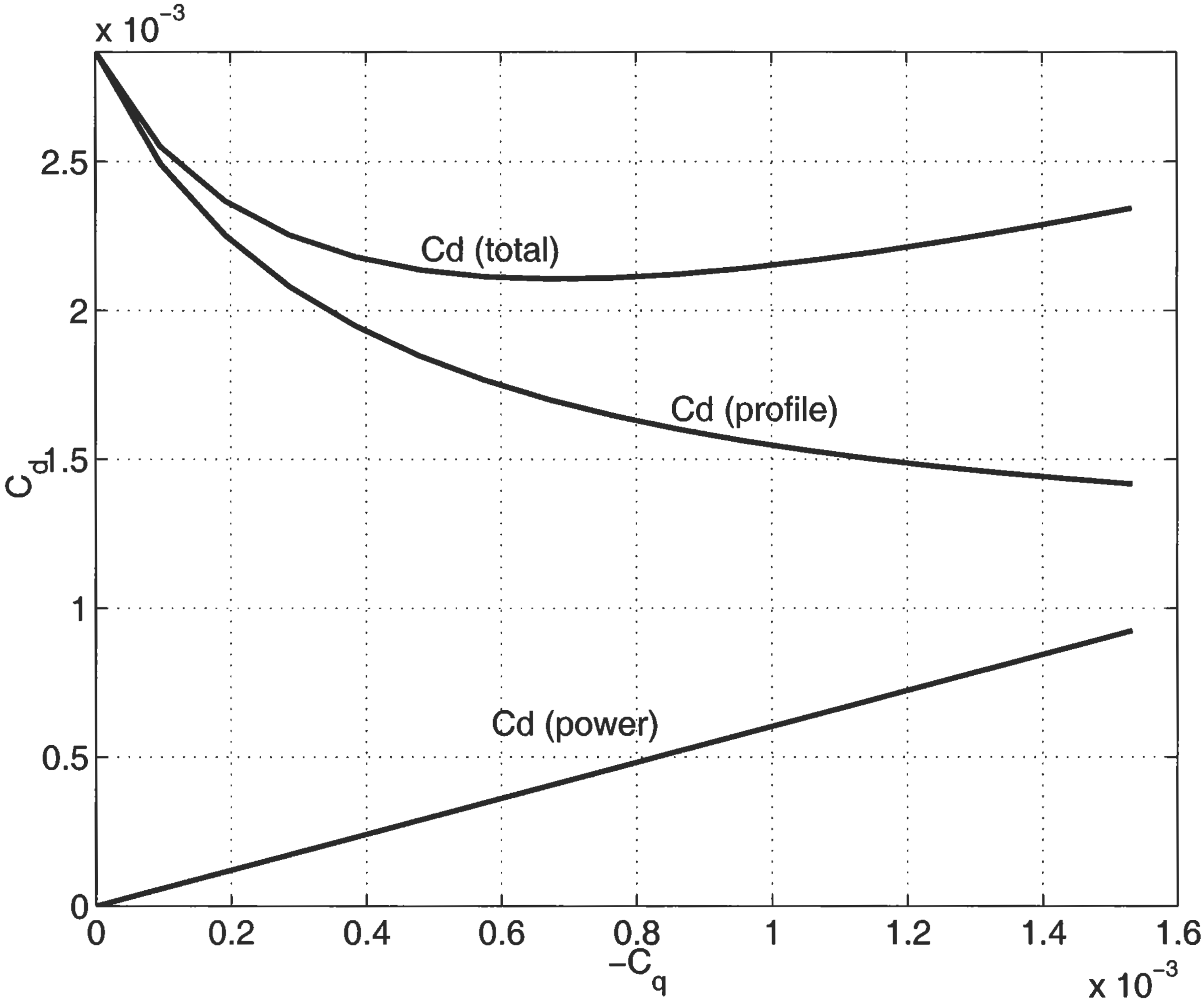
$$\beta_u = -0.09 \implies C_{P\,suction} = |C_a| \, 0.604.$$

The total drag is then

$$D = D_{prof} + \frac{P_{suction}}{U_\infty}$$

$$C_D = C_{D\,prof} + C_{P\,suction} \;//$$

Min. C_D occurs at $C_Q \approx 0.7 \times 10^{-3}$ (see attached plot)


x 10^-3
2.5
2
1.5
1
0.5
0
Cd (total)
Cd (profile)
Cd (power)
C_d
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
-C_q
x 10^-3

```fortran
C       .
C       .
C
C
C---- wall BC equation matrix entries, put into 1,2 equations of i=1 block row
      I = 1
      R(1,1,I) = F(I) + VWALL*2.0/(1.0 + BU)
      A(1,1,I) = 1.0
      R(1,2,I) =      - VWALL*2.0/(1.0 + BU)**2
C
      R(2,1,I) = U(I) - UWALL
      A(2,2,I) = 1.0
C
C---- set up equations for each i..i+1 interval
      DO 12 I = 1, N-1
        BCON = 0.5*(1.0 + BU)
        DETA = ETA(I+1) - ETA(I)
C
C------ set S-equation matrix entries, put into 3rd equation of I block row
        R(3,1,I) = S(I+1) - S(I)                                ! Residual
     &           + BCON*0.5*DETA*(F(I+1)*S(I+1) + F(I)*S(I))
     &           + BU*DETA*(1.0 - 0.5*(U(I+1)**2 + U(I)**2))
C
        C(3,1,I) = BCON*0.5*DETA*       S(I+1)                  ! dR/dF(i+1)
        A(3,1,I) = BCON*0.5*DETA*                     S(I)      ! dR/dF(i)
C
        C(3,2,I) = BU*DETA*(    -      U(I+1)              )    ! dR/dU(i+1)
        A(3,2,I) = BU*DETA*(    -                 U(I)     )    ! dR/dU(i)
C
        C(3,3,I) = 1.0                                          ! dR/dS(i+1)
     &           + BCON*0.5*DETA* F(I+1)
        A(3,3,I) =        - 1.0                                 ! dR/dS(i)
     &           + BCON*0.5*DETA*                   F(I)
C
        R(3,2,I) =     0.25*DETA*(F(I+1)*S(I+1) + F(I)*S(I))    ! dR/dBetau
     &           +    DETA*(1.0 - 0.5*(U(I+1)**2 + U(I)**2))
C
C------ set F equation matrix entries, put into 1st equation of I+1 block row
        R(1,1,I+1) = F(I+1) - F(I) - 0.5*DETA*(U(I+1) + U(I))   ! Residual
        A(1,1,I+1) = 1.0                                        ! dR/dF(i+1)
        B(1,1,I+1) =        - 1.0                               ! dR/dF(i)
        A(1,2,I+1) =                 - 0.5*DETA                 ! dR/dU(i+1)
        B(1,2,I+1) =                 - 0.5*DETA                 ! dR/dU(i)
C
C------ set U equation matrix entries, put into 2nd equation of I+1 block row
        R(2,1,I+1) = U(I+1) - U(I) - 0.5*DETA*(S(I+1) + S(I))   ! Residual
        A(2,2,I+1) = 1.0                                        ! dR/dU(i+1)
        B(2,2,I+1) =        - 1.0                               ! dR/dU(i)
        A(2,3,I+1) =                 - 0.5*DETA                 ! dR/dS(i+1)
        B(2,3,I+1) =                 - 0.5*DETA                 ! dR/dS(i)
C
 12   CONTINUE
C
C---- edge BC equation matrix entries, put into 3rd equation of i=N block row
      I = N
      R(3,1,I) = U(I) - 1.0
      A(3,2,I) = 1.0
C
C       .
C       .
C       .
C
C
C
C
C
```

```
C
C
      RBETA = 0.
      RB_R1 = 0.
      RB_R2 = 0.
C
C---- accumulate  Rbeta,  dR/dU(i).r(i),  dR/dU(i).s(i)  from each interval
      DO 16 I = 1, N-1
        DETA = ETA(I+1) - ETA(I)
        UAV = 0.5*(U(I) + U(I+1))
C
C------ set RBETA increment, and its derivative with respect to UAV
        DRB     = (1.0-UAV)*DETA  -  HSPEC*(UAV - UAV**2 )*DETA
        DRB_UAV =     -     DETA  -  HSPEC*(1.0 - UAV*2.0)*DETA
C
C------ derivatives of RBETA with respect to U(i), U(i+1)
        DRB_UO = DRB_UAV*0.5
        DRB_UP = DRB_UAV*0.5
C
C------ accumulate RBETA, and RBETA changes resulting from R(.1.) and R(.2.)
        RBETA = RBETA + DRB
        RB_R1 = RB_R1 + DRB_UO*R(2,1,I) + DRB_UP*R(2,1,I+1)
        RB_R2 = RB_R2 + DRB_UO*R(2,2,I) + DRB_UP*R(2,2,I+1)
   16 CONTINUE
C
C---- require  RBETA  +  d(RBETA)  =  0
C-         or  RBETA  -  RB_R1  -  RB_R2*DBU  =  0
C
      DBU  =  (RBETA - RB_R1) / RB_R2
```