Appendix A. Corrosion-model code

The following self-contained Python reproduces both correlations, the route profile and the Monte Carlo, so the results are fully auditable rather than available only on request.

```
import numpy as np; log10=np.log10
def a_coef(P,T): return 10**(P*(0.0031 - 1.4/T)) # fugacity coeff, P<=250 bar
def dwm(TC,pco2,P):
 T=TC+273.15; return 10**(5.8 - 1710/T + 0.67*log10(pco2*a_coef(P,T)))
Kt={5:0.42,15:1.59,20:4.762,40:8.927,60:10.695,80:9.949,90:6.25,120:7.77,150:5.203}
def fpH(pH): return 2.0676-0.2309*pH if pH<4.6 else 4.342-1.051*pH+0.0708*pH**2
Re=150*2.17*0.66/1.5e-5; fr=0.001375*(1+(2e4*(5e-5/0.66)+1e6/Re)**(1/3))
S=0.5*150*fr*2.17**2 # wall shear stress, Pa
def nor_at(Ti,f,pH):
 n=0.36 if Ti==5 else 0.62
 return Kt[Ti]*f**n*(S/19.)**(0.146+0.0324*log10(f))*fpH(pH)
def norsok(TC,pco2,P,pH=4.2): # interpolate RATES between tabulated T
 T=TC+273.15; f=pco2*a_coef(P,T); Tt=sorted(Kt)
 if TC<=5: return nor_at(5,f,pH)
 if TC>=150: return nor_at(150,f,pH)
 lo=max(x for x in Tt if x<=TC); hi=min(x for x in Tt if x>=TC)
 if lo==hi: return nor_at(lo,f,pH)
 return nor_at(lo,f,pH)+(nor_at(hi,f,pH)-nor_at(lo,f,pH))*(TC-lo)/(hi-lo)
# route: P 120->95 bar, T 60->6 C, 4 mol% CO2, local water-wetting peaks
```