Supporting information for

Hydrate stability in the H_2S–H_2O system – visual observations and measurements in a high–pressure optical cell and thermodynamic models

Jiyue Sun[1,3,‡], Yang Xin[1,3,‡], I–Ming Chou[1], Rui Sun[2], Lei Jiang[1,],*

[1]CAS Key Laboratory of Experimental Study under Deep–sea Extreme Conditions, Institute of Deep-sea Science and Engineering, Chinese Academy of Sciences, Sanya, Hainan 572000, China

[2]Department of geology, Northwest University, Xi'an 710069, China

[3]University of Chinese Academy of Sciences, Beijing 100049, China

*Corresponding author: Lei Jiang

Tel: +86 18689571979

E-mail: jl@idsse.ac.cn

[‡]Yang Xin is the co-first author

Table of Contents

1. General information

The details of the materials and suppliers are listed in Table S1.

Table S1. Suppliers and material purities.

Materials	Suppliers	Index
Distilled deionized water	Sichuan Ulupure Technology Co., Ltd	18.25 MΩ•cm
Hydrogen sulfide	Wuhan Newradar Special Gas Co., Ltd	99.99% (mass fraction)
Mercury	Shanghai Macklin Biochemical Co., Ltd	99.9% (mass fraction)
Fused silica capillary tube	Polymicro Technologie Co. Ltd	TSP600300 TSP150075
Valve	HIP Co. Ltd	HIP 15-15AF1
High pressure line	HIP Co. Ltd	HIP 60-9H2
Linkam stage	Linkam Co. Ltd	CAP 500
Pressure gauge	Oemga Co. Ltd	DPG400-5K PSI DPG400-10K PSI
Pressure pump	HIP Co. Ltd	17-4H1150HT#G23117
Microscope	OLYMPUS	BX-51

2. Sample Loading Procedure

Fig. S1 is a schematic diagram of HPOC. Fig. S2 is a schematic diagram that shows the sample loading system. The sample loading procedures are as follows:

(1) *Evacuating and loading H_2S in the pressure line*: [a] Close the valves V1, V2, V5, V6, and V10, and open V3, V4, V7, and V8. [b] Open V9, evacuate the pressure line using a vacuum pump (vii in Fig. S2), and monitor the pressure with a pressure gage (v in Fig. S2). [c] Close V9, and open V1 to inject H_2S gas into the pressure line from a gas cylinder (iv in Fig. S2). [d] Close V1, open V9, and evacuate the pressure line with the vacuum pump. [e] Repeat steps [c] and [d] several times to clean the pressure line with H_2S gas.

(2) *Loading of gas sample into HPOC*: [a] Connect one end of a thin fused silica capillary tube (75 μm ID, 150 μm OD, and ~30 cm length) to a 5–mL syringe loaded with distilled deionized water, and insert the other end into HPOC until it touches the very end of the enclosed end; [b] deliver water from the syringe into HPOC until it fills up the cell and outflows from the open end; [c] fix the close end of HPOC (ii in Fig. S2) on the microscope stage (i in Fig. S2) so that the sample can be clearly observed under the microscope; [d] withdraw the thin tube until the open end is approximately 3 cm from the enclosed end of HPOC; [e] disconnect the other end of the thin tube from the syringe and connect it to valve V5 (Fig. S2); [f] slowly open V5 and deliver H_2S from the pressure line into HPOC through the thin tube and push part of the water in HPOC out; [g] withdraw the thin tube from HPOC and insert another thin tube, which was connected to a 5–mL syringe loaded with Hg, into HPOC until the open end is approximately 1 mm from the enclosed end of HPOC; [h] deliver the section of approximately 25 mm of Hg into HPOC and push part of the water in HPOC out; [i] insert the thin tube that was connected to V5 into HPOC in the water section near the

open end, and slowly open V5 to expel air in the thin tube with the H_2S sample; [j] close V5, push the open end of the thin tube through the Hg section in HPOC, and deliver H_2S into HPOC until an approximately 200 mm long section of H_2S in HPOC between the water and Hg is reached (Fig. S3); [k] pull out the thin tube, connect the open end of HPOC to the pressure line shown in Fig. 2 (in manuscript file), and insert HPOC into the Linkam CAP 500 heating–cooling stage; [l] adjust the position of the sample under the microscope using a three-dimensional translation stage; and [m] pressurize the sample with water against Hg in HPOC using a pressure generator, and measure the pressure with a pressure transducer.

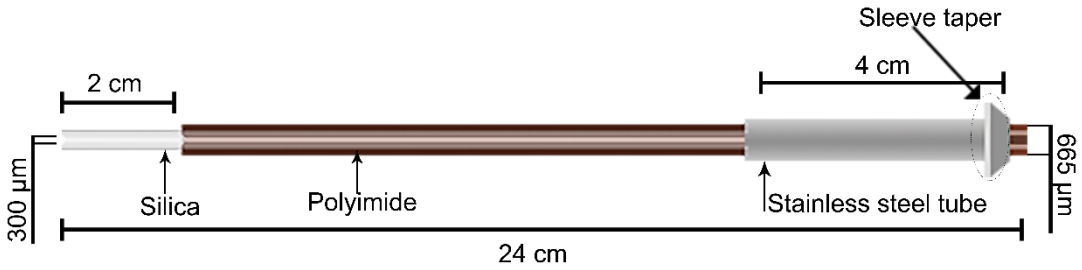
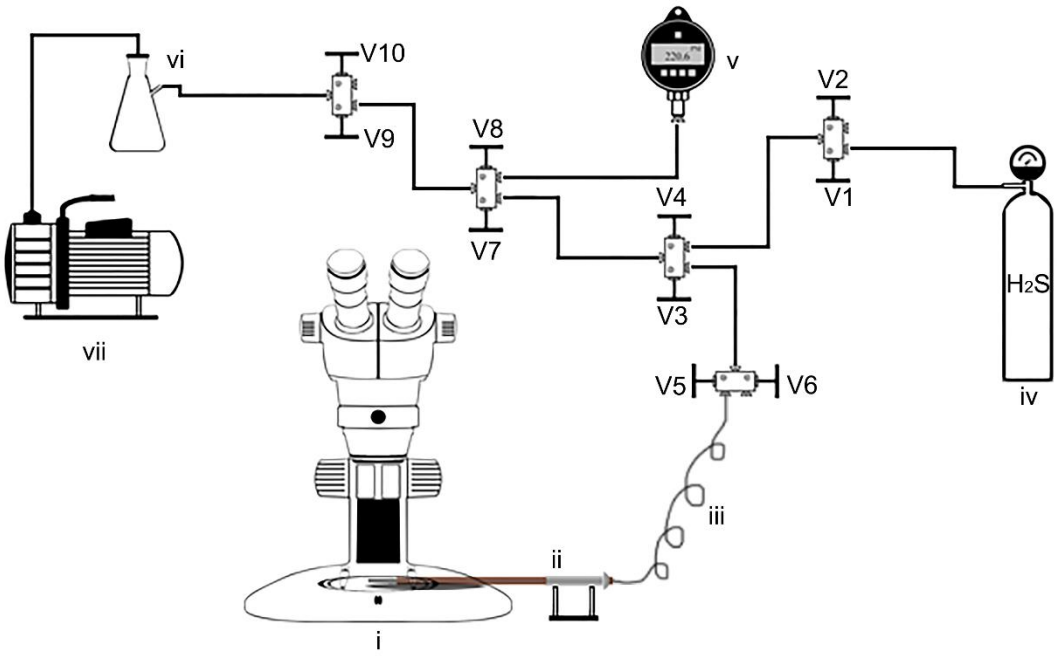


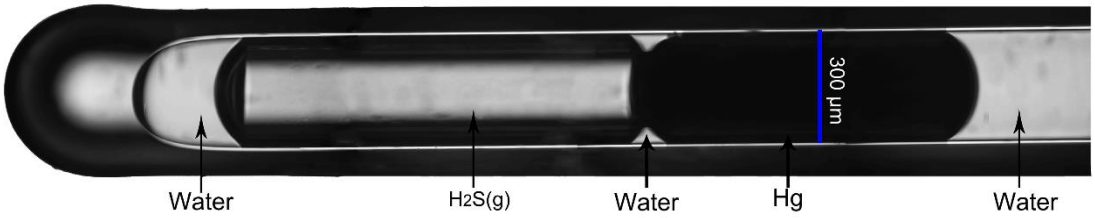
Figure S1. The diagram of the High Pressure Optical (HPOC)



67

68 **Figure S2.** Apparatus diagram for loading H_2S gas in the HPOC

69



70

71 **Figure S3.** The sample observed in HPOC with a microscope

72

3. Experimental data of literatures

The experimental data measured in this study as shown in Table 3 and 4. In addition, the experimental measured in previous as follows (Table S2-Table S7):

Author name(s): Scheffer, F. E. C.
Article title: On the determination of three phase pressures in the system hydrogen sulphide + water
https://www.dwc.knaw.nl/
Year: 1911
Notably, the units of temperature and pressure reported by Scheffer (1911) data are °F and atm.

Table S2. Experimental data reported by Scheffer[11].

Temperature(K)	Pressure(MPa)	Improved model	[a]AAD%	CSMHyd	[b]AAD%
289.5	0.53	0.54		0.54	
291.1	0.63	0.63		0.63	
292.9	0.76	0.75		0.77	
294.2	0.86	0.86		0.88	
296.5	1.07	1.09	0.8	1.13	3.4
298.0	1.29	1.29		1.33	
298.8	1.40	1.41		1.46	
300.6	1.71	1.73		1.81	
301.7	1.95	1.97		2.07	

[a] AAD calculated from this improved model.
[b] AAD calculated from CSMHYD.

Author name(s): Bond, D. C. and Russell, N. B.
Article title: Effect of Antifreeze Agents on the Formation of Hydrogen Sulphide Hydrate
https://www.onepetro.org/
Year: 1949
Notably, the units of temperature and pressure reported by Bond and Russell (1949) data are ℃ and psi.

Table S3. Experimental data reported by Bond and Russell[12].

Temperature(K)	Pressure(MPa)	Improved model	[a]AAD%	CSMHyd	[b]AAD%
283.2	0.310	0.287	5.7	0.281	9.2
291.2	0.710	0.634		0.641	
299.7	1.496	1.558		1.622	
302.4	2.240	2.229		6.173	

[a] AAD calculated from this improved model.
[b] AAD calculated from CSMHYD.

Author name(s): Selleck, F. T., Carmichael, L. T. and Sage, B. H.
Dataset title: Phase Behavior in the Hydrogen Sulfide-Water System
Data repository:
Version: https://pubs.acs.org/doi/abs/10.1021/ie50513a064
Year: 1952
Notably, the units of temperature and pressure reported by Selleck et al. (1952) data are °F and psi.

Table S4. Experimental data reported by Selleck et al.[13]

Temperature(K)	Pressure(MPa)	Improved model	[a]AAD%	CSMHyd	[b]AAD%
		H–Lw–H_2S (*g*)			
272.8	0.093	0.103		0.096	
277.6	0.157	0.165		0.158	
283.2	0.281	0.286		0.281	
285.2	0.345	0.349		0.345	
288.8	0.499	0.499		0.500	
291.9	0.690	0.681		0.690	
294.3	0.890	0.871	2.0	0.890	1.6
295.8	1.035	1.018		1.044	
298.5	1.379	1.362		1.411	
299.9	1.597	1.593		1.661	
300.5	1.724	1.708		1.783	
302.1	2.069	2.068		2.172	
302.7	2.239	2.229		-	
		H–Ice–H_2S (*g*)			
250.5	0.034	0.039		0.036	
255.4	0.044	0.049		0.045	
258.2	0.050	0.056		0.051	
261.0	0.057	0.063		0.058	
263.8	0.064	0.071	10.5	0.066	2.5
265.4	0.069	0.076		0.070	
266.5	0.072	0.079		0.074	
269.3	0.081	0.089		0.083	
272.1	0.090	0.099		0.093	
272.8	0.093	0.102		0.096	
		H–Lw–H_2S (*l*)			
302.7	2.239	2.229		6.173	
303.2	7.828	7.886		11.550	
303.8	14.193	14.723	2.0	18.462	49.5
304.3	20.959	20.696		24.527	
304.9	27.848	28.185		32.191	
305.4	35.076	34.699		38.898	

[a] AAD calculated from this improved model.
[b] AAD calculated from CSMHYD.

Author name(s): Carroll
Article title: Phase behaviour in the system water-hydrogen sulphide
Data repository: National Library of Canada
https://era.library.ualberta.ca/
Year: 1990
Notably, the unit of temperature reported by Carroll (1990) data is ℃.

Table S5. Experimental data reported by Carroll[14].

Temperature(K)	Pressure(MPa)	Improved model	[a]AAD%	CSMHyd	[b]AAD%
298.7	1.610	1.392		1.446	
298.8	1.620	1.408		1.461	
299.0	1.710	1.439		1.495	
299.1	1.680	1.456		1.512	
299.3	1.700	1.489		1.547	
299.5	1.700	1.523		1.584	
299.8	1.750	1.575	11.7	1.641	8.0
299.9	1.770	1.593		1.661	
300.1	1.810	1.630		1.700	
300.3	1.850	1.668		1.741	
300.4	1.870	1.688		1.762	
300.8	1.970	1.768		1.849	
300.9	2.070	1.789		1.871	

[a] AAD calculated from this improved model.
[b] AAD calculated from CSMHYD.

Author name(s): Mohammadi, A. H. and Richon, D.
Article title: Equilibrium Data of Carbonyl Sulfide and Hydrogen Sulfide Clathrate Hydrates
https://pubs.acs.org/doi/10.1021/je900209y
Year: 2009

Table S6. Experimental data reported by Mohammadi and Richon[15].

Temperature(K)	Pressure(MPa)	Improved	[a]AAD%	CSMHyd	[b]AAD%
277.7	0.164	0.166		0.160	
279.0	0.188	0.189		0.183	
280.5	0.218	0.219		0.213	
281.7	0.247	0.247		0.241	
283.1	0.283	0.284		0.278	
284.6	0.332	0.329		0.325	
286.1	0.380	0.381		0.378	
288.6	0.514	0.489		0.490	
290.0	0.578	0.562	1.7	0.566	2.5
291.3	0.672	0.641		0.648	
292.5	0.764	0.724		0.735	
294.0	0.865	0.844		0.861	
295.3	0.988	0.966		0.990	
296.6	1.131	1.108		1.141	
297.5	1.237	1.221		1.261	
298.8	1.425	1.408		1.461	
300.4	1.692	1.688		1.762	
301.3	1.861	1.876		1.965	

[a] AAD calculated from this improved model.
[b] AAD calculated from CSMHYD.

Author name(s): Ward, Z. T., Deering, C. E., Marriott, R. A., Sum, A. K., Sloan, E. D. and Koh, C. A.
Article title: Phase Equilibrium Data and Model Comparisons for H_2S Hydrates
https://pubs.acs.org/doi/10.1021/je500657f
Year: 2014

Table S7. Experimental data reported by Ward et al.[16]

T **(K)**	***P*** **(MPa)**	**Improved model**	[a]**AAD%**	**CSMHyd**	[b]**AAD%**
273.68	0.1078	0.1121		0.106	
274.18	0.1120	0.1178		0.111	
274.68	0.1176	0.1238		0.117	
275.17	0.1237	0.1299		0.123	
275.67	0.1323	0.1365		0.130	
276.16	0.1372	0.1433		0.136	
276.66	0.1463	0.1506		0.144	
277.17	0.1516	0.1583		0.151	
277.65	0.1614	0.1661		0.159	
278.15	0.1673	0.1745		0.168	
278.65	0.1774	0.1833		0.176	
279.15	0.1849	0.1925		0.186	
279.66	0.1958	0.2024		0.196	
280.16	0.2042	0.2126		0.206	
280.67	0.2159	0.2235		0.217	
281.15	0.2259	0.2343		0.228	
281.65	0.2384	0.2462		0.240	
282.15	0.2498	0.2586		0.253	
282.65	0.2645	0.2716		0.266	
283.13	0.2766	0.2847		0.279	
283.63	0.2914	0.2991		0.294	
284.13	0.3049	0.3142		0.309	
284.63	0.3218	0.3301	2.2	0.326	1.9
285.12	0.3379	0.3464		0.342	
285.61	0.3560	0.3635		0.360	
286.11	0.3725	0.3819		0.379	
286.61	0.3929	0.4013		0.390	
287.10	0.4123	0.4213		0.419	
287.61	0.4319	0.4431		0.442	
288.11	0.4536	0.4656		0.465	
288.60	0.4777	0.4889		0.490	
289.10	0.5028	0.5139		0.516	
289.61	0.5289	0.5407		0.543	
290.09	0.5600	0.5673		0.571	
290.58	0.5854	0.5959		0.601	
291.08	0.6198	0.6267		0.633	
291.59	0.6502	0.6598		0.668	
292.07	0.6814	0.6927		0.702	
292.58	0.7227	0.7296		0.741	
293.07	0.7618	0.7670		0.780	
293.57	0.8023	0.8073		0.823	
294.09	0.8460	0.8515		0.869	
294.58	0.8918	0.8961		0.916	
295.06	0.9400	0.9420		0.965	
295.58	0.9909	0.9947		1.020	

296.07	1.0454	1.0475	1.077
296.58	1.1033	1.1058	1.139
297.07	1.1668	1.1654	1.202
297.54	1.2359	1.2260	1.267
298.04	1.3058	1.2946	1.340
298.55	1.3809	1.3693	1.420
299.03	1.4594	1.4443	1.500
299.52	1.5475	1.5261	1.588
300.03	1.6382	1.6174	1.686
300.52	1.7374	1.7115	1.788
301.01	1.8446	1.8127	1.897
301.53	1.9598	1.9228	2.022

146 [a] AAD calculated from this improved model.
147 [b] AAD calculated from CSMHYD.
148
149

150

4. Appendix A

According to Duan et al.,[22, 23] the solubility of H_2S (m_{H_2S}) is expressed as follows:

$$m_{H_2S} = \ln y_{H_2S} P + \ln \varphi_{H_2S}(T,P,y) - \ln \gamma_{H_2S}(T,P,m) - \frac{\mu_{H_2S}^{l(0)}}{RT} \quad (A.1)$$

where y_{H_2S} is the mole fraction of H_2S in the vapor phase, φ_{H_2S} is the fugacity coefficient of H_2S, γ_{H_2S} is the activity coefficient of H_2S in the liquid phase, and $\mu_{H_2S}^{l(0)}$ represents the chemical potential in a hypothetical ideal solution of unity solution molality. For the gas–H_2O binary system, γ_{H_2S} is approximated as unity; y_{H_2S} is approximately calculated from

$$y_{H_2S} = \frac{P - P_{H_2O}}{P} \quad (A.2)$$

where P_{H_2O} is the saturation vapor pressure of water.

$\mu_{H_2S}^{l(0)}$ is given by

$$\begin{aligned} \frac{\mu_{H_2S}^{l(0)}}{RT} &= c_1 + c_2 T + c_3/T + c_4 T^2 + c_5/(680 - T) \\ &\quad + c_6 P + c_7 P/(680 - T) + c_8 P^2/T \end{aligned} \quad (A.3)$$

where $c_1 - c_8$ are the interaction parameters, as shown in Table 6.

The EOS for H_2S is expressed as follows:

$$\begin{aligned} Z = \frac{P_r V_r}{T_r} &= 1 + \frac{a_1 + a_2/T_r^2 + a_3/T_r^3}{V_r} + \frac{a_4 + a_5/T_r^2 + a_6/T_r^3}{V_r^2} \\ &\quad + \frac{a_7 + a_8/T_r^2 + a_9/T_r^3}{V_r^4} + \frac{a_{10} + a_{11}/T_r^2 + a_{12}/T_r^3}{V_r^5} \\ &\quad + \frac{a_{13}}{T_r^3 V_r^2}\left(a_{14} + \frac{a_{15}}{V_r^2}\right)\exp\left(-\frac{a_{15}}{V_r^2}\right) \end{aligned} \quad (A.4)$$

where $a_1 - a_{15}$ are the regression coefficients, which are given in Table 6. P_r, V_r and T_r represent reduced pressure, reduced volume, and reduced temperature, respectively

$$P_r = \frac{P}{P_c} \tag{A.5}$$

$$T_r = \frac{T}{T_c} \tag{A.6}$$

$$V_r = \frac{V}{V_c} \tag{A.7}$$

where T_c and P_c are the critical pressure and temperature of H_2S, respectively. $T_c = 373.6$ K and $P_c = 9.008$ MPa. V_c is the mole volume of H_2S, which is expressed as follows:

$$V_c = \frac{RT_c}{P_c} \tag{A.8}$$

$\ln \varphi_{H_2S}(T,P,y)$ is expressed as follows:

$$\begin{aligned} \ln \varphi_{H_2S}(T,P) = {} & Z - 1 - \ln Z + \frac{a_1 + a_2/T_r^2 + a_3/T_r^3}{V_r} + \frac{a_4 + a_5/T_r^2 + a_6/T_r^3}{2V_r^2} \\ & + \frac{a_7 + a_8/T_r^2 + a_9/T_r^3}{4V_r^4} + \frac{a_{10} + a_{11}/T_r^2 + a_{12}/T_r^3}{5V_r^5} \\ & + \frac{a_{13}}{2T_r^3 a_{15}}\left[a_{14} + 1 - \left(a_{14} + 1 + \frac{a_{15}}{V_r^2}\right)\right]\exp\left(-\frac{a_{15}}{V_r^2}\right) \end{aligned} \tag{A.9}$$

5. Thermodynamic program

```
program thermodynamic
      implicit none
!-----The program to solve the formation pressure of h2s hydrate
!-----Using Sloan(1998) Model
!-----duh is computed by using the method of Van der Waals and Platteeuw(1959)
!-----The Langmuir constant is computed using Kihara Potential(Sloan, 1998)
!-----duw is computed by using the formula of Sloan(1998)
!-----h2s fugacity coefficient is computed using Duan-92 EOS(1992)
      double precision no,terr,err,i,Pcal,Pexp,T      !
      !implicit double precision (a-h,o-z)
      open(21,file="data-hlv-h2s.dat")
      open(22,file="hydrate-h2s.out")
      go to 101
      print*,'Calculation of the dissociation pressure of H2S hydrate'
      read(21,*)no
      terr=0.
      do 10 i=1,no
         read(21,*)T,Pexp
         call hlvh2s(T,Pcal)
         err=100*(Pcal-Pexp)/Pexp
         terr=terr+abs(err)
         write(22,15)T,Pexp,Pcal,Pcal-Pexp,err
10     continue
15     format(1x,5d12.5)
         terr=terr/(1.*no)
      write(22,*)'AAD=',terr
      go to 102
```

```
199
200     101 print*,'Please enter Temperature(K)'
201         read*,T
202         call hlvh2s(T,Pcal)
203         go to 101
204     102 end
205
206             subroutine hlvh2s(T,Pcal)
207     !-----unit: T/K, P/bar
208             implicit double precision (a-h,o-z)
209             double                                                  precision
210     eps,T,P,du,du1,du2,dv,duh,duh1,duh2,duw,duw1,duw2,p1,p2,P0,n,Pcal,thi1,thi2
211         double precision ch2s(2,2)
212         common /b/ch2s
213         eps=2.e-5
214         call pdiss(T,P0)
215         print*,p0
216         call Cmlh2s(T,Ch2s)
217         print*,'-------The Langmuir constants of h2s--------'
218         print*,'c1s=',ch2s(1,1),'    c1l=',ch2s(1,2)
219         print*,'c2s=',ch2s(2,1),'    c2l=',ch2s(2,2)
220         call duh72h2s(T,P0,duh)
221         call duw98h2s(T,P0,duw)
222         print*,p0,'    duh=',duh,'    duw=',duw
223         P1=0.7*P0
224         P2=1.3*P0
225         n=0
226     301 n=n+1
227             P=(P1+P2)/2.
228         call duh72h2s(T,P,duh)
```

```
229         call duw98h2s(T,P,duw)
230         call duh72h2s(T,P1,duh1)
231         call duw98h2s(T,P1,duw1)
232         call duh72h2s(T,P2,duh2)
233         call duw98h2s(T,P2,duw2)
234         du=duh-duw
235         du1=duh1-duw1
236         du2=duh2-duw2
237         print*,'n=',n
238         print*,'du=',du,'    du1=',du1,'    du2=',du2
239         if(du1.lt.0.and.du2.lt.0) then
240            P2=P2*1.3
241            go to 301
242         end if
243         if(du1.gt.0.and.du2.gt.0) then
244            P1=P1*0.7
245            go to 301
246         end if
247         if(du*du1.le.0) then
248            P1=P1
249            P2=P
250         else
251            P1=P
252            P2=P2
253         end if
254         if(abs((P2-P1)/P).le.eps) go to 302
255         if(n.ge.20) then
256            go to 302
257         else
```

```
258             go to 301
259         end if
260   302    Pcal=P
261             print*,'T=',T,'    Pcal=',Pcal,'    duh=',duh,'    duw=',duw
262             call foccupy(T,P,thi1,thi2)
263         print*,'thi1=',thi1,'    thi2=',thi2
264         return
265         end
266
267         subroutine pdiss(T,P)
268         implicit double precision (a-h,o-z)
269         if (T.lt.302.65) then
270            P=exp(31.65-9335/T)*10.
271         else
272            P=10*(-3252+10.751*T)
273         end if
274         return
275         end
276
277              subroutine foccupy(T,P,thi1,thi2)
278         double precision T,P,Ch2s(2,2),fih2s,thi1,thi2
279         common /b/ch2s
280         v1s=1./23.
281         v1l=3./23.
282            duhh2s=0.
283            call solfih2s(T,P,fih2s)
284            fh2s=fih2s*P*1e5
285            thi1=ch2s(1,1)*fh2s/(1+ch2s(1,1)*fh2s)
286            thi2=ch2s(1,2)*fh2s/(1+ch2s(1,2)*fh2s)
```

```
287             duhh2s=v1s*log(1+ch2s(1,1)*fh2s)+v1l*log(1+ch2s(1,2)*fh2s)
288             duhh2s=duhh2s*8.314*T
289         return
290         end
291
292         subroutine duh72h2s(T,P,duhh2s)
293     !-----this program is to calculate deltaUh of h2s. The Langmuir constants of guest gases
294     !-----of clathrate hydrate by using the empirical formula of Parrish & Prausnitz(1972)-
295     ---
296     !-----C*(1,1): Langmuir constant of gas in small cavity of Structure I'
297     !-----C*(1,2): Langmuir constant of gas in large cavity of Structure I'
298     !-----C*(2,1): Langmuir constant of gas in small cavity of Structure II'
299     !-----C*(2,2): Langmuir constant of gas in large cavity of Structure II'
300
301              double precision T,Ch2s(2,2),P,fih2s,fh2s,duhh2s
302         common /a/fih2s
303         common /b/ch2s
304         v1s=1./23.
305         v1l=3./23.
306             duhh2s=0.
307             call solfih2s(T,P,fih2s)
308     !       print*,'fih2s=',fih2s
309             fh2s=1e5*fih2s*P
310             duhh2s=v1s*log(1+ch2s(1,1)*fh2s)+v1l*log(1+ch2s(1,2)*fh2s)
311             duhh2s=duhh2s*8.314*T
312     !       print*,'duHh2s(structure I)=',duhh2s
313     !            write(22,25)T,P,duhh2s
314     25       format(1x,3d15.5)
315              return
316         end
```

```
317
318         subroutine duw98h2s(T,P,duw)
319     !-----This program is to calculate deltaUw of h2s by using the method of
320     !-----Sloan(1998)
321             implicit double precision (a-h,o-z)
322         T0=273.15
323         duw=0.
324             duw0=1264.
325     !   dcp0=38.12
326     !   a=-0.0336
327         if(T.le.272.8) then
328            dvw0=3.0
329            dhw0=1389.
330            dcp0=0.565
331            a=0.002
332         else
333            dvw0=3.0+1.598
334            dhw0=-4620.5
335            dcp0=-38.12
336            a=0.141
337         end if
338         sv=0.1*(P-1)*dvw0/8.314/T
339         sH=(1./T0-1./T)*(dhw0+0.5*a*T0**2-dcp0*T0)/8.314
340         sH=sH+log(T/T0)*(dcp0-a*T0)/8.314+0.5*a*(T-T0)/8.314
341         su=duw0/8.314/T0
342         gama=1.0
343         cnacl=0.
344         if(T.le.272.8) then
345            xw=1.
```

```
346         else
347           if(P.lt.40) then
348             call solubH2S(T,P,cnacl,fmH2S)
349           else
350             call solubH2S(T,40.d0,cnacl,fmH2S1)
351             fmh2s=fmh2s1+9.175d-4*(P-40.)
352           end if
353             xh2s=fmh2s/(55.5093+fmh2s)
354             print*,'xh2s, mH2S',xh2s,fmh2s
355             xw=1-xh2s
356         end if
357         duw=su-sH+sv-log(gama*xw)
358         duw=8.314*T*duw
359   !     print*,'du/RT=',su,'    sdh=',sH,'    sv=',sv,gama*xw,duw
360         return
361         end
362
363
364         subroutine solfih2s(T,P,fc)
365   !       The aim of this sub-program is to solve fugcity coefficient of h2s using
366   !       EOS of Duan(1992)
367             PARAMETER(limit=1.E-5)
368             implicit double precision (a-h,o-z)
369             DOUBLE PRECISION a(15)
370   !          DOUBLE PRECISION X,Y,Z
371             INTEGER NO
372   !-----paramater 43
373   !          data a/5.2386075E-2,-2.7463906E-1,-9.6760173E-2,1.3618104E-2,
374   !         &-8.8681753E-2,4.1176908E-2,3.6354018E-4,2.2719194E-003,
```

```
!        &-7.6962514E-4,-2.1948579E-5,-1.1707631E-4,4.0756926E-5,
!        &5.7582260E-2,6.000000E-2,1.d0/
!-----paramater 46
            data a/6.3370721E-2,-3.0858172E-1,-6.4755492E-2,2.4872387E-5,&
        &-4.6369149E-2,5.2259063E-2,1.1870632E-3,-1.0576579E-4,&
          &-4.9821065E-4,-1.0324415E-4,1.0983783E-4,-1.5221068E-5,&
          &3.5309394E-2,8.00000E-002,0.3693d0/
            COMMON B,C,D,E,F,beta,gama,Tr,Pr
        Tc=373.6
        Pc=90.08
        Vc=10.*8.314*Tc/Pc
            Tr=T/Tc
        Pr=P/Pc
            B=a(1)+a(2)/(Tr**2)+a(3)/(Tr**3)
            C=a(4)+a(5)/(Tr**2)+a(6)/(Tr**3)
            D=a(7)+a(8)/(Tr**2)+a(9)/(Tr**3)
            E=a(10)+a(11)/(Tr**2)+a(12)/(Tr**3)
            F=a(13)/(Tr**3)
            gama=a(14)
        beta=a(15)
!       write(*,*)'      B          ','      C          ','     D          ','     E        '
!         write(*,15)B*Vc,C*Vc**2,D*Vc**4,E*Vc**5
15      Format(1x,4d12.5)
!         PRINT*,B,C,D,E,F,GAMA
            if(P.lt.1) then
           x=15000./Vc/P
           y=50000./Vc/P
         go to 103
        end if
```

```
404             if(T.lt.373.6) then
405              call solps(T,Ps,Zl,Zv)
406              Vls=(10*Zl*8.314*T)/Ps
407              Vvs=(10*Zv*8.314*T)/Ps
408              if(T.le.373) then
409                 fk=1.0
410              else
411     !           fk=(1.0+0.0045*(T-368.15)/5.)
412                    fk=0.998
413     !           fk=1.0
414              end if
415              ps1=Ps*fk
416              print*,'ps=',ps
417              if(P.gt.Ps1)then
418                  X=0.08
419                     Y1=2.*Vls/vc
420     !                print*,x,y1,o(x),o(y1)
421     301              if(o(y1).lt.0) then
422                     y=y1
423                  else
424                    Y1=y1*0.95
425                     if(y1.gt.x) then
426                        go to 301
427                     else
428                             y=y1/0.95
429                     end if
430                  end if
431              else
432                  x1=0.95*Vvs/vc
```

```
433                 y=150
434  302              if(o(x1).gt.0) then
435                  x=x1
436                 else
437                  x1=x1*1.02
438                  if(x1.lt.(1.3*Vvs/Vc)) then
439                     go to 302
440                  else
441                           x=x1
442                  end if
443                end if
444              end if
445         else
446              x=0.08
447              y=150
448         end if
449
450  103    NO=0
451              IF ((O(X)*O(Y)).GT.0) THEN
452                 print*,o(x),o(y)
453              PRINT*,'THE INITIAL X,Y IS ILL'
454                 fc=1.
455              Vcal=0.
456              GO TO 201
457              END IF
458  102    NO=NO+1
459              Z=(X+Y)/2.0
460  !           PRINT*,Z
461              IF((O(X)*O(Z)).LT.0) THEN
```

```
462               Y=Z
463             ELSE
464               X=Z
465             END IF
466             IF(NO.GT.20) GO TO 101
467             IF(ABS(Y-X).GT.LIMIT) GO TO 102
468     101     V=Z
469           Vcal=V*Vc
470     !          PRINT*,'V=',Vcal
471             Vr=V
472             Zh2s=1+B/Vr+C/Vr**2+D/Vr**4+E/Vr**5+(F/Vr**2)*(a(15)+a(14)/&
473            &Vr**2)*exp(-a(14)/Vr**2)
474     !          print*,'Z=',zh2s
475             alnz=log(Zh2s)
476             alnfi=B/Vr+C/2/Vr**2+D/4/Vr**4+E/5/Vr**5+(F/2./a(14))*(1+beta-&
477           &(1+beta+a(14)/Vr**2)*exp(-a(14)/Vr**2))
478             alnfi=alnfi+Zh2s-1-alnz
479                fc=exp(alnfi)
480     201     print*,'V, Z, phi',Vcal,zh2s,fc
481             return
482             END
483
484           double precision FUNCTION O(Vr)
485             implicit double precision (a-h,o-z)
486             COMMON B,C,D,E,F,beta,gama,Tr,Pr
487     !         print*,"T=",Tr*373.55,"P=",Pr*89.0
488
489     O=1+B/Vr+C/Vr**2+D/Vr**4+E/Vr**5+(F/(Vr**2))*(beta+gama/Vr**2)*&
490            &EXP(-gama/Vr**2)-Pr*Vr/Tr
491             END
```

```
492
493         subroutine solPs(T,Ps,Zl,Zv)
494             double precision T,Ps,Zl,Zv
495         Ps=exp(10.138249+3.8680356E-4*T-2162.0152/T)
496         if(T.le.300) then
497            Zl=-.18016569+2.6691167E-3*T-1.3493646E-5*T**2+2.3380698E-8*T**3
498         else if(T.lt.363) then
499            Zl=-8.1718266+8.0033730E-2*T-2.6267732E-4*T**2+2.9043672E-7*T**3
500         else
501            Zl=-824.83434+6.9796305*T-1.9687281E-2*T**2+1.8513753E-5*T**3
502         end if
503         if(T.le.358) then
504            Zv=3.6464702-2.8526053E-2*T+1.0573070E-4*T**2-1.4055174E-7*T**3
505         else
506            Zv=1008.1309-8.5313883*T+2.4093639E-2*T**2-2.2694674E-5*T**3
507         end if
508         return
509         end
510
511             subroutine solubH2S(T,P,cnacl,solH2S)
512             double precision T,P,cnacl,fc,ucal,Ps,vls
513             double precision y,c(8),d(5),solH2S,xh2s
514     !-----par51+61
515     !          data c/43.735962,-8.9458838E-2,-6217.2639,7.2080718E-5,
516     !     &-128.87113,2.4847634E-4,-9.6568159E-1,3.9325305E-3/
517     !          data d/8.6044430E-2,3.2742930E-5,-1.7784342,1.5291720E-5,
518     !     &-1.0671321E-2/
519     !-----par54+64
520             data c/42.564957,-8.6260377E-2,-6084.3775,6.8714437E-5,&
```

```
521         &-102.76849,8.4482895E-4,-1.0590768,3.5665902E-3/
522             data d/8.5004999E-2,3.5330378E-5,-1.5882605,1.1894926E-5,&
523         &-1.0832589E-2/
524         if(T.le.272.1) then
525            print*,'Error!    The input temperature is too low'
526            go to 101
527         end if
528             call Psh2o(T,Ps)
529     !        call vaporpressure(T,cNaCl,Ps,Vls)
530             if(P.lt.Ps) then
531            print*,'Error!    The input pressure is too low'
532            solh2s=0.
533            go to 101
534         end if
535            y=(P-Ps)/P
536     !           yH2S=y
537                call solfih2s(T,P,fc)
538     !           fc=fcH2S
539                call solu0(T,P,c,ucal)
540     !    print*,'uh2s(l0)',ucal
541         call solgama(T,P,cnacl,d,gama)
542            solH2S=log(y*P)+log(fc)-ucal-gama
543            solH2S=exp(solH2S)
544     !    xh2s=solH2S/(55.5093+solH2S)
545     101     return
546         end
547
548         subroutine solu0(T,P,c,u)
549         double precision T,P,c(8),u
```

```
      u=c(1)+c(2)*T+c(3)/T+c(4)*T**2+c(5)/(680.-T)+c(6)*P+&
        &c(7)*P/(680.-T)+c(8)*P**2/T
           return
      end

      subroutine solgama(T,P,cnacl,d,gama)
      double precision T,P,cnacl,d(5)
!     ruda=d(1)+d(2)/T+d(3)*P*T+d(4)*P
      ruda=d(1)+d(2)*T+d(3)/T+d(4)*P
!       ruda=d(1)+d(2)*T+d(3)/T+d(4)*P*T+d(5)*P/(680-T)+d(6)*P
      cta=d(5)
      gama=2*cnacl*ruda+cnacl**2*cta
      return
      end

      subroutine psH2o(T,Ps)
      double precision T,Ps,c(5),t1
      data c/-38.640844,5.8948420,59.876516,26.654627,10.637097/
      Tc=647.29
      Pc=220.85
      t1=(T-Tc)/Tc
      ps=(Pc*T/Tc)*(1+c(1)*(-T1)**(1.9)+c(2)*t1+c(3)*t1**2+&
        &c(4)*t1**3+c(5)*t1**4)
      return
      end


         subroutine cmlh2s(T,cml)
      double precision T,cml(2,2)
```

```
579         real R(2,2),b(2,2)
580         PI=3.1415926
581         R(1,1)=3.975
582         R(1,2)=4.3
583         R(2,1)=3.91
584         R(2,2)=4.73
585
586         ah2s=0.36
587         a=0.
588             b(1,1)=R(1,1)-ah2s*2-0.0001
589             b(1,2)=R(1,2)-ah2s*2-0.0001
590             b(2,1)=R(2,1)-ah2s*2-0.0001
591             b(2,2)=R(2,2)-ah2s*2-0.0001
592             call simps(T,a,b(1,1),1,s1)
593             call simps(T,a,b(1,2),2,s2)
594             call simps(T,a,b(2,1),3,s3)
595             call simps(T,a,b(2,2),4,s4)
596             print*,'s1=',s1,'    s2=',s2,'    s3=',s3,'    s4=',s4
597         cml(1,1)=4*PI*s1*1.e-7/T/1.38
598         cml(1,2)=4*PI*s2*1.e-7/T/1.38
599         cml(2,1)=4*PI*s3*1.e-7/T/1.38
600         cml(2,2)=4*PI*s4*1.e-7/T/1.38
601     !   print*,'    c11=',cml(1,1),'    c12=',cml(1,2)
602     !        print*,'    c21=',cml(2,1),'    c22=',cml(2,2)
603         return
604             end
605     !----------------------------------------
606     !   The following is sinpson intergration
607             subroutine simps(T,a,b,jj,s)
```

```
608         double precision T
609         common /cml/jj0,T0
610         jj0=jj
611         T0=T
612             eps=1e-5
613             ck=1.
614     !       simpson integral method
615             h=(b-a)*0.5
616             fp=f(a)+f(b)
617             ci=f(a+h)
618             fs1=(4.0*ci+fp)*h/3.0
619             fp=2.0*ci+fp
620             n=1
621     1       n=n+n
622             x=a-0.5*h
623             ci=0.0
624             do 2 i=1,n
625                 x=x+h
626                 ci=f(x)+ci
627     2       continue
628             fs2=(4.0*ci+fp)*h/6.0
629             x=fs2-fs1
630             if(abs(fs2).ge.ck) then
631                x=x/fs2
632             end if
633             if(n.gt.2**15)then
634                s=fs2
635                print*,'x=',x,' n=',n
636                go to 11
```

```
637             end if
638             if(abs(x).ge.eps) then
639                fs1=fs2
640                fp=2.0*ci+fp
641                h=0.5*h
642                go to 1
643             else
644                s=fs2
645     !           print*,n
646             end if
647     11         end
648
649             function f(x)
650     !       double precision T
651             common /cml/jj,T
652             e=205.65
653             thigma=3.146
654             a=0.36
655             if (jj.eq.1) then
656                z=20.
657                R=3.975
658             else if(jj.eq.2) then
659                z=24.
660                R=4.3
661             else if(jj.eq.3) then
662                z=20.
663                R=3.91
664             else if(jj.eq.4) then
665                z=28.
```

```
666             R=4.73
667           end if
668     !     z=24.
669     !     R=4.33
670           if(x.ge.1e-3) then
671           f=x**2*exp(-2*z*e*(thigma**12*((1./(1.-x/R-a/R)**10-1./&
672           &(1.+x/R-a/R)**10)/10.+(a/R)*(1./(1.-x/R-a/R)**11-1./&
673           &(1.+x/R-a/R)**11)/11.)/(R**11*x)-thigma**6*((1./(1.-x/R-a/R)**4&
674             &-1./(1.+x/R-a/R)**4)/4.+(a/R)*(1./(1.-x/R-a/R)**5-1./&
675           &(1.+x/R-a/R)**5)/5.)/(R**5*x))/T)
676               else
677           f=x**2*exp(-2*z*e*(thigma**12*(2*R**10/(R-a)**11+(a/R)*2*R**11/&
678           &(R-a)**12)/R**11-thigma**6*(2*R**4/(R-a)**5+(a/R)*2*R**5/(R-a)**6)&
679             &/R**5)/T)
680           end if
681     !          f=100.*sin(10/x)/x**2
682               end
683     !---------------------------------------
```