Supplementary Information Figures S1-S5

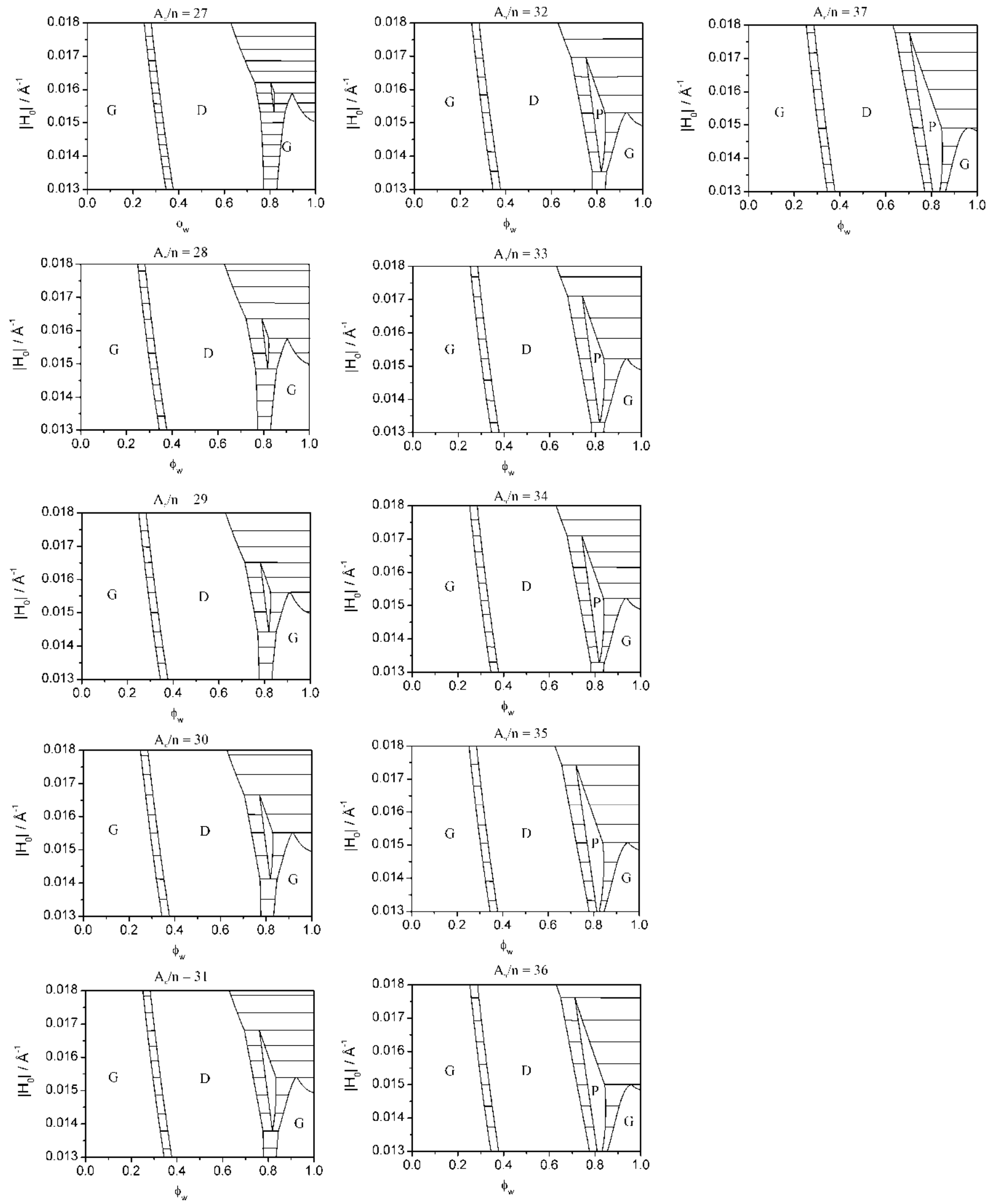


Figure S1. Partial theoretical phase diagrams, where the magnitude of H_0 ($|H_0|$) is acting as a first order proxy for temperature, for which A_n/v = 0.06 Å^{-1}, v_n/v = 0.74, κ_G/κ = -0.75 and λ/κ = 0.0005 Å^{-2}, with A_n/n varying between 27 Å^2 and 37 Å^2.

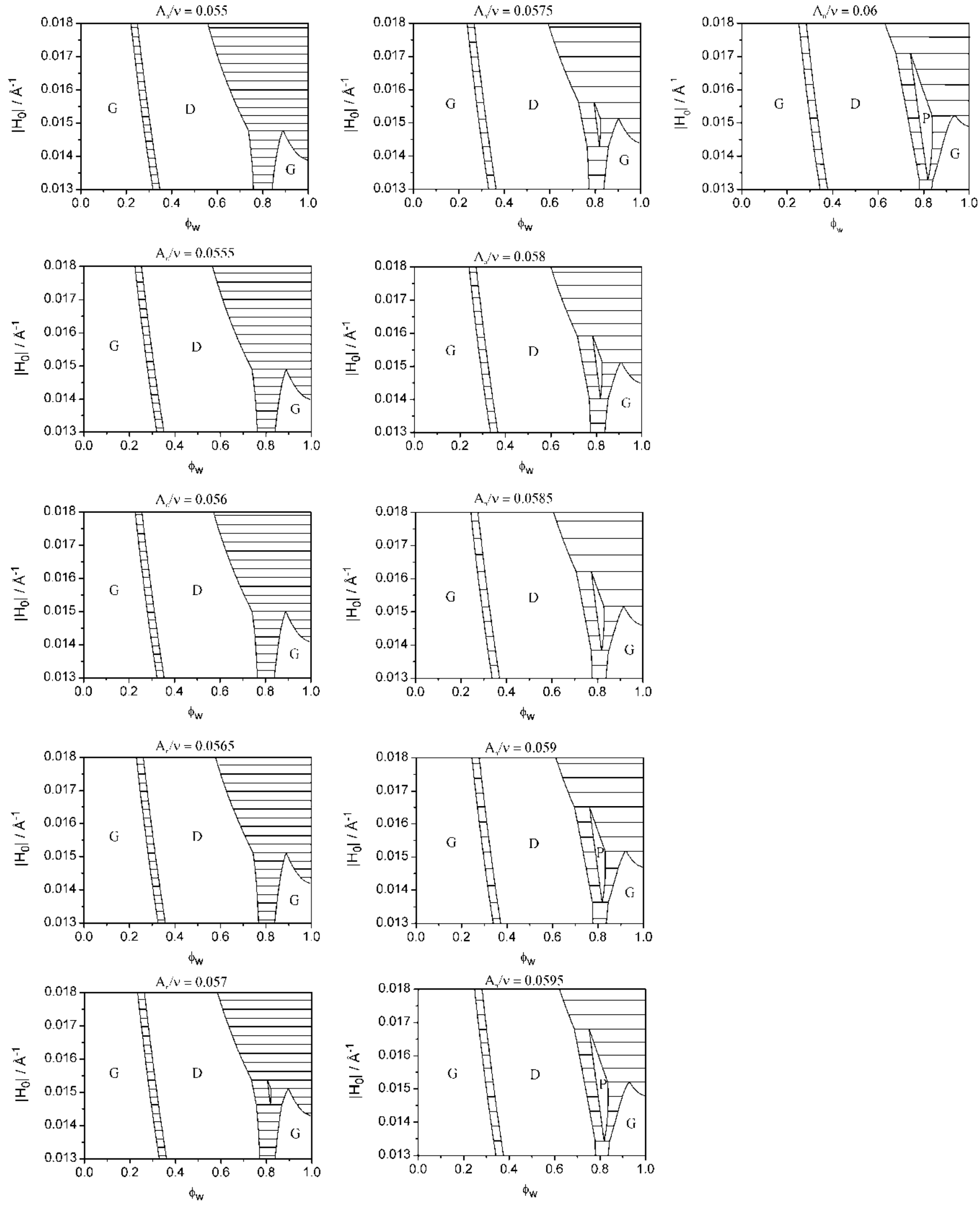


Figure S2. Partial theoretical phase diagrams, where $|H_0|$ is acting as a first order proxy for temperature, for which A_n/n = 33 Å^2, v_n/v = 0.74, κ_G/κ = -0.75 and λ/κ = 0.0005 Å^{-2}, with A_n/v varying between 0.055 Å^{-1} and 0.06 Å^{-1}.

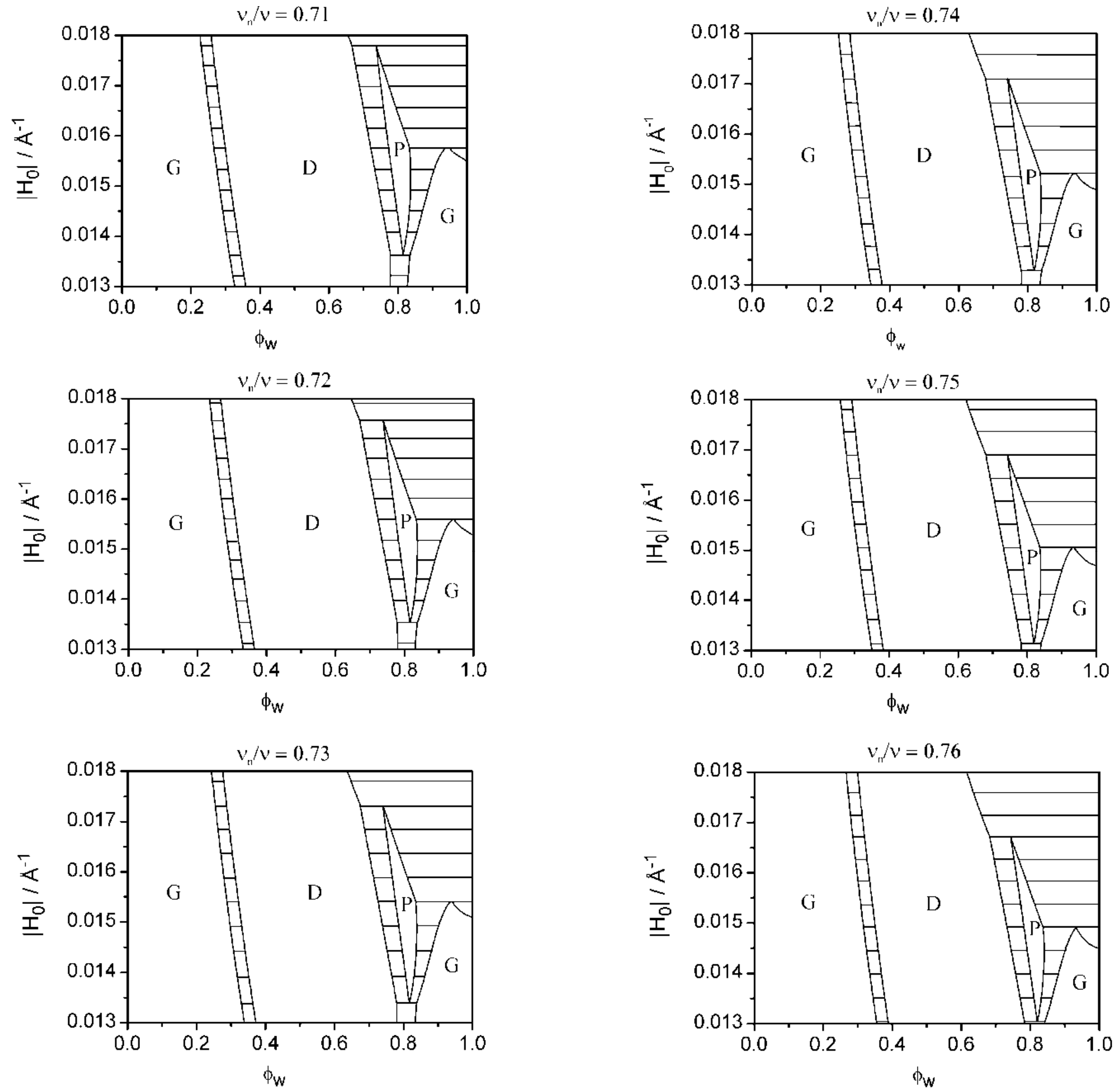
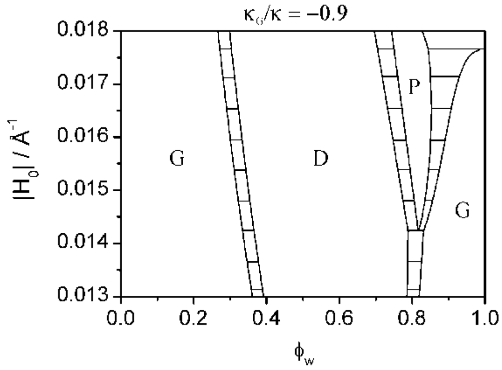
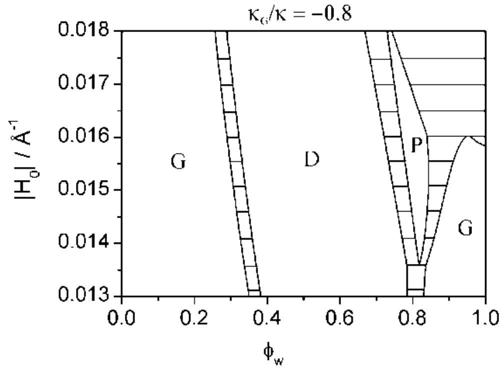
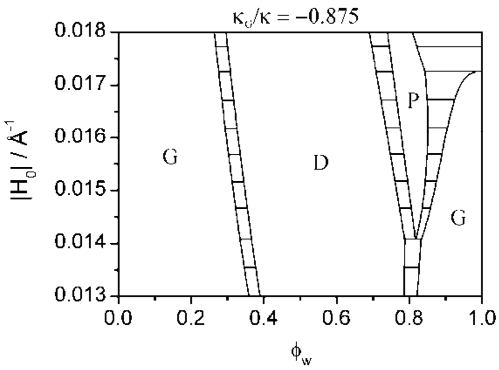
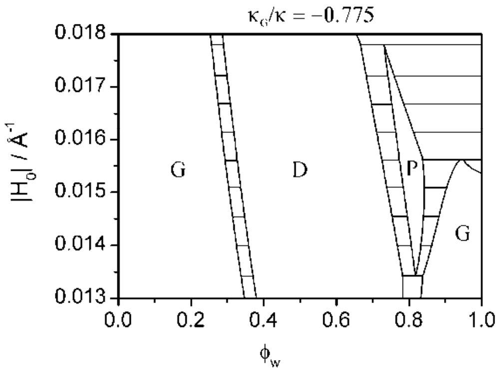


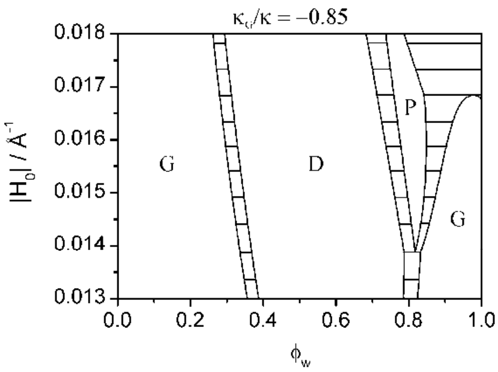
Figure S3. Partial theoretical phase diagrams, where $|H_0|$ is acting as a first order proxy for temperature, for which A_n/n = 33 Å^2, A_n/v = 0.06 Å^{-1}, κ_G/κ = -0.75 and λ/κ = 0.0005 Å^{-2}, with v_n/v varying between 0.71 and 0.76.

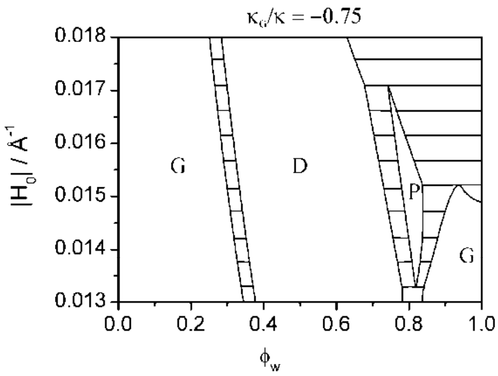

κG/κ = −0.9
|H0| / Å-1
φw
G
D
P
G


κG/κ = −0.8
|H0| / Å-1
φw
G
D
P
G


κG/κ = −0.875
|H0| / Å-1
φw
G
D
P
G


κG/κ = −0.775
|H0| / Å-1
φw
G
D
P
G


κG/κ = −0.85
|H0| / Å-1
φw
G
D
P
G


κG/κ = −0.75
|H0| / Å-1
φw
G
D
P
G

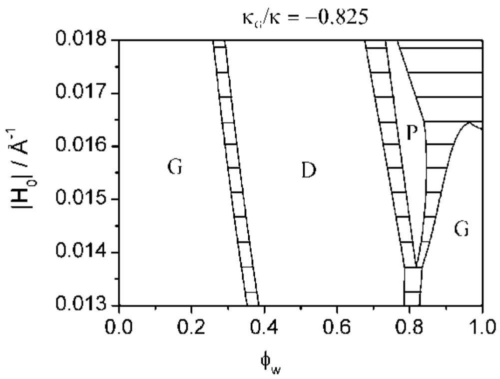

κG/κ = −0.825
|H0| / Å-1
φw
G
D
P
G

Figure S4. Partial theoretical phase diagrams, where $|H_0|$ is acting as a first order proxy for temperature, for which A_n/n = 33 Å^2, A_n/v = 0.06 Å^{-1}, v_n/v = 0.74 and λ/κ = 0.0005 Å^{-2}, with κ_G/κ varying between -0.9 and -0.75.

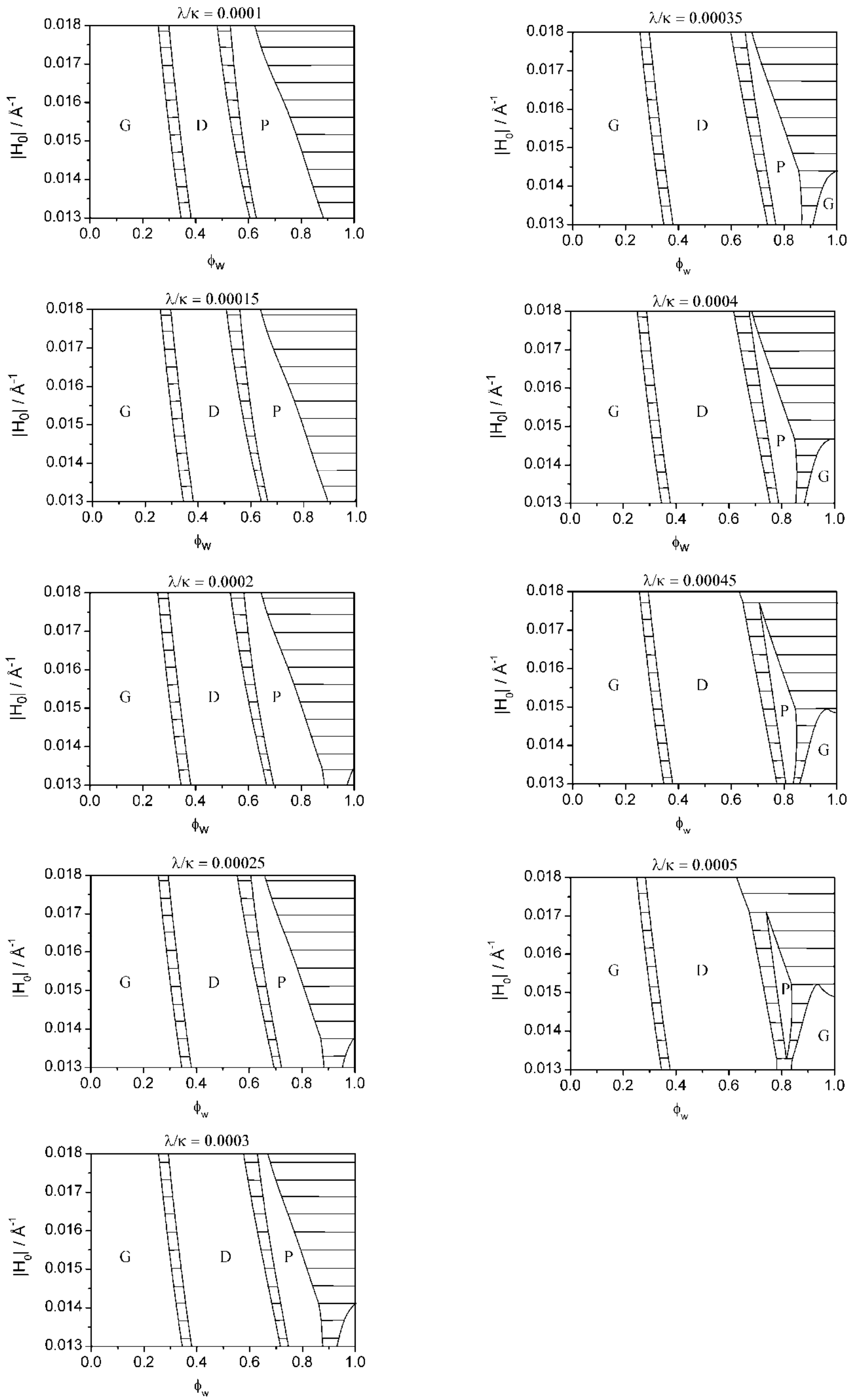

λ/κ = 0.0001
λ/κ = 0.00035
λ/κ = 0.00015
λ/κ = 0.0004
λ/κ = 0.0002
λ/κ = 0.00045
λ/κ = 0.00025
λ/κ = 0.0005
λ/κ = 0.0003
|H0| / Å-1
φw
G
D
P
0.013
0.014
0.015
0.016
0.017
0.018
0.0
0.2
0.4
0.6
0.8
1.0

Figure S5. Partial theoretical phase diagrams, where $|H_0|$ is acting as a first order proxy for temperature, for which A_n/n = 33 Å^2, A_n/v = 0.06 Å^{-1}, v_n/v = 0.74 and κ_G/κ = -0.75, with λ/κ varying between 0.0001 Å^{-2} and 0.0005 Å^{-2}.

Mathematica Code:

```
Off[General::spell]
Off[General::spell1]

μc := Apdivn*(2*(Hp - H0)^2 + κGdivκ*Kn)
Θ := (vpdivv)*(1 - ϕ)
suma := (σa*(Θ)^0) + (σb*(Θ)^2) + (σc*(Θ)^4) + (σd*(Θ)^6)
secsum := (ζa*(Θ^1)) + (ζb*(Θ^3)) + (ζc*(Θ^5)) + (ζd*(Θ^7)) + (ζe*(Θ^9))
a := (2/Apdivv)*(1/(1 - ϕ))* suma
Hp := - secsum / a
Sn := (Apdivv/2)*((a^3)*(1 - ϕ))
Kn := 2*π*χ/ Sn

diamond := 0.000337618(1 - ϕ)^2 + 0.0000337858(1 - ϕ)^4 - 0.000489167(1 - ϕ)^6 + 0.00216436(1 - ϕ)^8
primitive := 0.000210286(1 - ϕ)^2 + 0.00258955(1 - ϕ)^4 - 0.0166996(1 - ϕ)^6 + 0.125123(1 - ϕ)^8
gyroid := 0.00010488(1 - ϕ)^2 + 0.000356135(1 - ϕ)^4 - 0.00228924(1 - ϕ)^6 + 0.00450621(1 - ϕ)^8
varprimfapc := primitive*a^2 /. σa -> 2.3451 /. σb -> -1.1713 /. σc -> -0.22227 /. σd -> -2.4952
vardiafapc := diamond*a^2 /. σa -> 1.9189 /. σb -> -0.8893 /. σc -> -0.061856 /. σd -> -0.65474
vargyrfapc := gyroid*a^2 /. σa -> 3.0915 /. σb -> -1.3317 /. σc -> -0.19974 /. σd -> -0.80113
μfp := λdivκ*varprimfapc
μfd := λdivκ*vardiafapc
μfg := λdivκ*vargyrfapc

da := a /. σa -> 1.9189 /. σb -> -0.8893 /. σc -> -0.061856 /. σd -> -0.65474 /. ζa -> 1.8379 /. ζb -> -0.17345 /. ζc -> 16.508 /. ζd -> -62.547 /. ζe -> 76.638 /. χ -> -2
pa := a /. σa -> 2.3451 /. σb -> -1.1713 /. σc -> -0.22227 /. σd -> -2.4952 /. ζa -> 3.0458 /. ζb -> -14.256 /. ζc -> 237.3 /. ζd -> -1365.6 /. ζe -> 2719.7 /. χ -> -4
ga := a /. σa -> 3.0915 /. σb -> -1.3317 /. σc -> -0.19974 /. σd -> -0.80113 /. ζa -> 2.7953 /. ζb -> -9.9778 /. ζc -> 59.478 /. ζd -> -111.96 /. ζe -> 74.685 /. χ -> -8

dμc := μc /. σa -> 1.9189 /. σb -> -0.8893 /. σc -> -0.061856 /. σd -> -0.65474 /. ζa -> 1.8379 /. ζb -> -0.17345 /. ζc -> 16.508 /. ζd -> -62.547 /. ζe -> 76.638 /. χ -> -2
pμc := μc /. σa -> 2.3451 /. σb -> -1.1713 /. σc -> -0.22227 /. σd -> -2.4952 /. ζa -> 3.0458 /. ζb -> -14.256 /. ζc -> 237.3 /. ζd -> -1365.6 /. ζe -> 2719.7 /. χ -> -4
gμc := μc /. σa -> 3.0915 /. σb -> -1.3317 /. σc -> -0.19974 /. σd -> -0.80113 /. ζa -> 2.7953 /. ζb -> -9.9778 /. ζc -> 59.478 /. ζd -> -111.96 /. ζe -> 74.685 /. χ -> -8
μptot := μfp + pμc
μdtot := μfd + dμc
μgtot := μfg + gμc
```

The phase diagram can be determined from free energy graphs by obtaining the common tangent lines between different phases - where the common tangent touches the curve becomes the point at

which the phases mix in each case. In order to determine the common tangent line, one must equate both the gradients and the y-intercepts and then solve the simultaneous equation. The excess water point can be determined by finding the minimum of each of the free energy curves by differentiating and equating to zero (to find a turning point). These can then be compared to each other to find the one which lies at the lowest energy.
Please change the OpenWrite and Import addresses to more suitable ones of your preference.

```
phasematrix = OpenWrite["C:\phasematrix_t", FormatType -> OutputForm];
yphasematrix = OpenWrite["C:\yphasematrix_t", FormatType -> OutputForm, PageWidth -> 200];
```

We need to initialise the starting values first, then increment them as:
H0 varies as the spontaneous radius of curvature rato of 0.2A/K
An/n increases 0.17%/K
An/v increases 0.09%/K
vn/v does not change
Kg/K scales as (An/n)^-2.5
lambda/K scales as (An/n)^7.5
Here we have set our initial R0 value as 35, but this can be changed.

```
Do[
Clear[phaserec];
gtopsecboundl = 10;
gtodsecboundl = 10;
dtopsecboundl = 10;
dtogsecboundl = 10;
ptodsecboundl = 10;
ptogsecboundl = 10;
Apdivn = 33  + 33*(0.0017*k);
vpdiv  = 0.76;
Apdivv = 0.057  + 0.057*(0.0009*k);
κGdivκ = -0.75*((Apdivn/33)^(-2.5));
λdivκ = 0.00035*((Apdivn/33)^(7.5));
T = k;
R0 = (35 - 0.2  k);
H0 = -1/2/R0;
diffp = D[μptot, φ];
diffd = D[μdtot, φ];
diffg = D[μgtot, φ];
pmin =  FindRoot[diffp == 0,  {φ, 0.1}, MaxIterations  ->  200];
dmin = FindRoot[diffd ==  0,  {φ, 0.1}, MaxIterations  ->  200];
gmin = FindRoot[diffg ==  0,  {φ, 0.1}, MaxIterations  -> 200];
yμptot = μptot /. φ -> al;
yμdtot = μdtot /. φ -> b;
yμgtot = μgtot /. φ -> c;
diffpal = diffp /. φ -> al;
diffdb = diffd /. φ -> b;
diffgc = diffg /. φ ->  c;
yintp = yμptot - (diffpal*al);
yintd = yμdtot - (diffdb*b);
yintg = yμgtot - (diffgc*c);
```

```
record[1] = T;
record[2] = φ/. pmin[[1]];
record[3] = φ/. dmin[[1]];
record[4] = φ/. gmin[[1]];
phaserec[1] = T;
ypmin = μptot /. φ-> record[2];
ydmin = μdtot /. φ-> record[3];
ygmin = μgtot /. φ-> record[4];
If[ypmin < ydmin && ypmin < ygmin,
phaserec[2] = record[2];
If[record[2] > 0.999,
phaserec[2] = 1]];
If[ydmin < ypmin && ydmin < ygmin,
phaserec[2] = record[3];
If[record[3] > 0.999,
phaserec[2] = 1]];
If[ygmin < ypmin && ygmin < ydmin,
phaserec[2] = record[4];
If[record[4] > 0.999,
phaserec[2] = 1]];
player = Array[record, 4];
t = 1;
n = 0;
Do[
If[μgtot < μdtot, w = 1];
If[μdtot < μgtot, w = 3];
If[t ≠ w && t == 1 && w == 3,
n = n + 1;
m = (3*(n - 1));
record[m + 5] = t;
record[m + 6] = w;
record[m + 7] = φ;
];
If[t ≠ w && t == 3 && w == 1,
n = n + 1;
n = n + 1;
m = (3*(n - 1));
record[m + 5] = t;
record[m + 6] = w;
record[m + 7] = φ;
];
t = w;
Null, {φ, 0.226, 0.5, 0.001}]
Do[
If[μgtot < μdtot && μgtot < μptot, w = 1];
If[μptot < μgtot && μptot < μdtot, w = 2];
If[μdtot < μgtot && μdtot < μptot, w = 3];
If[t ≠ w && t == 1 && w == 2,
n = n + 1;
m = (3*(n - 1));
```

```
record[m + 5] = t;
record[m + 6] = w;
record[m + 7] = φ;
];
If[t ≠ w && t == 2 && w == 1,
n = n + 1;
m = (3*(n - 1));
record[m + 5] = t;
record[m + 6] = w;
record[m + 7] = φ;
];
If[t ≠ w && t == 1 && w == 3,
n = n + 1;
m = (3*(n - 1));
record[m + 5] = t;
record[m + 6] = w;
record[m + 7] = φ;
];
If[t ≠ w && t == 3 && w == 1,
n = n + 1;
m = (3*(n - 1));
record[m + 5] = t;
record[m + 6] = w;
record[m + 7] = φ;
];
If[t ≠ w && t == 2 && w == 3,
n = n + 1;
m = (3*(n - 1));
record[m + 5] = t;
record[m + 6] = w;
record[m + 7] = φ;
];
If[t ≠ w && t == 3 && w == 2,
n = n + 1;
m = (3*(n - 1));
record[m + 5] = t;
record[m + 6] = w;
record[m + 7] = φ;
];
t = w;
Null, {φ, 0.5, 0.999, 0.001}]
j = 0;
Do[
phaserec[i] = 1;
Null, {i, 3, (2  n) + 2}];
Do[
m = (3*(i - 1));
If[record[m + 5] == 1 && record[m + 6] == 3 && record[m + 7] < phaserec[2],
gtod = FindRoot[{diffgc == diffdb, yintg == yintd}, {c, record[m + 7] - 0.05, 0.112, 1}, {b, record[m + 7]
+ 0.05, 0.226, 1}, MaxIterations -> 200];
```

```
gtodboundl = c /. gtod[[1]];
gtodboundr = b /. gtod[[2]];
ygtodboundl = yμgtot /. c -> gtodboundl;
ygtodboundr = yμdtot /. b -> gtodboundr;
slopegtod = (ygtodboundr - ygtodboundl)/(gtodboundr - gtodboundl);
If[record[m + 5] == 1 && record[m + 6] == 3 && record[m + 8] == 3 && record[m + 9] == 2 &&
record[m + 10] < phaserec[2],
gtopsec = FindRoot[{diffpal == diffgc, yintp == yintg}, {c, record[m + 7], 0.112, 1}, {al, record[m + 10],
0.5, 1}, MaxIterations -> 200];
gtopsecboundl = c /. gtopsec[[1]];
gtopsecboundr = al /. gtopsec[[2]];
If[gtopsecboundl > 0.112 && gtopsecboundl < 1 && gtopsecboundr > 0.5 && gtopsecboundr < 1,
ygtopsecboundl = yμgtot /. c -> gtopsecboundl;
ygtopsecboundr = yμptot /. al -> gtopsecboundr;
slopegtopsec = (ygtopsecboundr - ygtopsecboundl)/(gtopsecboundr - gtopsecboundl);
If[slopegtopsec < slopegtod,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = gtopsecboundl;
phaserec[(2 (j - 1)) + 4] = gtopsecboundr;
];
];
];
If[record[m + 5] == 1 && record[m + 6] == 3 && record[m + 2] == 2 && record[m + 3] == 1 &&
record[m + 6] < phaserec[2] && phaserec[(2 (j - 1)) + 3] == ptodsecboundl,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = ptodsecboundl;
phaserec[(2 (j - 1)) + 4] = ptodsecboundr;
];
If[ygtodboundl > ygtodboundr && phaserec[(2 (j - 1)) + 3] ≠ gtopsecboundl && phaserec[(2 (j - 1)) +
3] ≠ ptodsecboundl,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = gtodboundl;
phaserec[(2 (j - 1)) + 4] = gtodboundr;
];
];
If[record[m + 5] == 3 && record[m + 6] == 1 && record[m + 7] < phaserec[2],
dtog = FindRoot[{diffgc == diffdb, yintg == yintd}, {b, record[m + 7] - 0.05, 0.226, 1}, {c, record[m + 7]
+ 0.05, 0.112, 1}, MaxIterations -> 200];
dtogboundl = b /. dtog[[1]];
dtogboundr = c /. dtog[[2]];
ydtogboundl = yμdtot /. b -> dtogboundl;
ydtogboundr = yμgtot /. c -> dtogboundr;
slopedtog = (ydtogboundr - ydtogboundl)/(dtogboundr - dtogboundl);
If[record[m + 5] == 3 && record[m + 6] == 1 && record[m + 8] == 1 && record[m + 9] == 2 &&
record[m + 10] < phaserec[2],
dtopsec = FindRoot[{diffdb == diffpal, yintd == yintp}, {b, record[m + 7], 0.226, 1}, {al, record[m + 10],
0.5, 1}, MaxIterations -> 200];
dtopsecboundl = b /. dtopsec[[1]];
dtopsecboundr = al /. dtopsec[[2]];
If[dtopsecboundl > 0.226 && dtopsecboundl < 1 && dtopsecboundr > 0.5 && dtopsecboundr < 1,
```

```
ydtopsecboundl = yμdtot /. b -> dtopsecboundl;
ydtopsecboundr = yμptot /. al -> dtopsecboundr;
slopedtopsec = (ydtopsecboundr - ydtopsecboundl)/(dtopsecboundr - dtopsecboundl);
If[slopedtopsec < slopedtog,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = dtopsecboundl;
phaserec[(2 (j - 1)) + 4] = dtopsecboundr;
];
];
];
If[record[m + 5] == 3 && record[m + 6] == 1 && record[m + 2] == 2 && record[m + 3] == 3 &&
record[m + 6] < phaserec[2] && phaserec[(2 (j - 1)) + 3] == ptogsecboundl,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = ptogsecboundl;
phaserec[(2 (j - 1)) + 4] = ptogsecboundr;
];
If[ydtogboundl > ydtogboundr && phaserec[(2 (j - 1)) + 3] ≠ dtopsecboundl && phaserec[(2 (j - 1)) +
3] ≠ ptogsecboundl,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = dtogboundl;
phaserec[(2 (j - 1)) + 4] = dtogboundr;
];
];
If[record[m + 5] == 1 && record[m + 6] == 2 && record[m + 7] < phaserec[2],
gtop = FindRoot[{diffpal == diffgc, yintp == yintg}, {c, record[m + 7] - 0.05, 0.112, 1}, {al, record[m +
7] + 0.05, 0.5, 1}, MaxIterations -> 200];
gtopboundl = c /. gtop[[1]];
gtopboundr = al /. gtop[[2]];
ygtopboundl = yμgtot /. c -> gtopboundl;
ygtopboundr = yμptot /. al -> gtopboundr;
slopegtop = (ygtopboundr - ygtopboundl)/(gtopboundr - gtopboundl);
If[record[m + 5] == 1 && record[m + 6] == 2 && record[m + 8] == 2 && record[m + 9] == 3 &&
record[m + 10] < phaserec[2],
gtodsec = FindRoot[{diffgc == diffdb, yintg == yintd}, {c, record[m + 7], 0.112, 1}, {b, record[m + 10],
0.226, 1}, MaxIterations -> 200];
gtodsecboundl = c /. gtodsec[[1]];
gtodsecboundr = b /. gtodsec[[2]];
If[gtodsecboundl > 0.112 && gtodsecboundl < 1 && gtodsecboundr > 0.226 && gtodsecboundr < 1,
ygtodsecboundl = yμgtot /. c -> gtodsecboundl;
ygtodsecboundr = yμdtot /. b -> gtodsecboundr;
slopegtodsec = (ygtodsecboundr - ygtodsecboundl)/(gtodsecboundr - gtodsecboundl);
If[slopegtodsec < slopegtop,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = gtodsecboundl;
phaserec[(2 (j - 1)) + 4] = gtodsecboundr;
];
];
];
If[record[m + 5] == 1 && record[m + 6] == 2 && record[m + 2] == 3 && record[m + 3] == 1 &&
record[m + 6] < phaserec[2] && phaserec[(2 (j - 1)) + 3] == dtopsecboundl,
```

```
j = j + 1;
phaserec[(2 (j - 1)) + 3] = dtopsecboundl;
phaserec[(2 (j - 1)) + 4] = dtopsecboundr;
];
If[ygtopboundl > ygtopboundr && phaserec[(2 (j - 1)) + 3] ≠ gtodsecboundl && phaserec[(2 (j - 1)) +
3] ≠ dtopsecboundl,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = gtopboundl;
phaserec[(2 (j - 1)) + 4] = gtopboundr;
];
];
If[record[m + 5] == 2 && record[m + 6] == 1 && record[m + 7] < phaserec[2],
ptog = FindRoot[{diffpal == diffgc, yintp == yintg}, {al, record[m + 7] - 0.05, 0.5, 1}, {c, record[m + 7] +
0.05, 0.112, 1}, MaxIterations -> 200];
ptogboundl = al /. ptog[[1]];
ptogboundr = c /. ptog[[2]];
yptogboundl = yμptot /. al -> ptogboundl;
yptogboundr = yμgtot /. c -> ptogboundr;
slopeptog = (yptogboundr - yptogboundl)/(ptogboundr - ptogboundl);
If[record[m + 5] == 2 && record[m + 6] == 1 && record[m + 8] == 1 && record[m + 9] == 3 &&
record[m + 10] < phaserec[2],
ptodsec = FindRoot[{diffdb == diffpal, yintd == yintp}, {al, record[m + 7], 0.5, 1}, {b, record[m + 10], –
0.226, 1}, MaxIterations -> 200];
ptodsecboundl = al /. ptodsec[[1]];
ptodsecboundr = b /. ptodsec[[2]];
If[ptodsecboundl > 0.5 && ptodsecboundl < 1 && ptodsecboundr > 0.226 && ptodsecboundr < 1,
yptodsecboundl = yμptot /. al -> ptodsecboundl;
yptodsecboundr = yμdtot /. b -> ptodsecboundr;
slopeptodsec = (yptodsecboundr - yptodsecboundl)/(ptodsecboundr - ptodsecboundl);
If[slopeptodsec < slopeptog,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = ptodsecboundl;
phaserec[(2 (j - 1)) + 4] = ptodsecboundr;
];
];
];
If[record[m + 5] == 2 && record[m + 6] == 1 && record[m + 2] == 3 && record[m + 3] == 2 &&
record[m + 6] < phaserec[2] && phaserec[(2 (j - 1)) + 3] == dtogsecboundl,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = dtogsecboundl;
phaserec[(2 (j - 1)) + 4] = dtogsecboundr;
];
If[yptogboundl > yptogboundr && phaserec[(2 (j - 1)) + 3] ≠ ptodsecboundl && phaserec[(2 (j - 1)) +
3] ≠ dtogsecboundl,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = ptogboundl;
phaserec[(2 (j - 1)) + 4] = ptogboundr;
];
];
```

```
If[record[m + 5] == 3 && record[m + 6] == 2 && record[m + 7] < phaserec[2],
dtop = FindRoot[{diffdb == diffpal, yintd == yintp}, {b, record[m + 7] - 0.05, 0.226, 1}, {al, record[m +
7] + 0.05, 0.5, 1}, MaxIterations -> 200];
dtopboundl = b /. dtop[[1]];
dtopboundr = al /. dtop[[2]];
ydtopboundl = yμdtot /. b -> dtopboundl;
ydtopboundr = yμptot /. al -> dtopboundr;
slopedtop = (ydtopboundr - ydtopboundl)/(dtopboundr - dtopboundl);
If[record[m + 5] == 3 && record[m + 6] == 2 && record[m + 8] == 2 && record[m + 9] == 1 &&
record[m + 10] < phaserec[2],
dtogsec = FindRoot[{diffgc == diffdb, yintg == yintd}, {b, record[m + 7], 0.226, 1}, {c, record[m + 10],
0.112, 1}, MaxIterations -> 200];
dtogsecboundl = b /. dtogsec[[1]];
dtogsecboundr = c /. dtogsec[[2]];
If[dtogsecboundl > 0.226 && dtogsecboundl < 1 && dtogsecboundr > 0.112 && dtogsecboundr < 1,
ydtogsecboundl = yμdtot /. b -> dtogsecboundl;
ydtogsecboundr = yμgtot /. c -> dtogsecboundr;
slopedtogsec = (ydtogsecboundr - ydtogsecboundl)/(dtogsecboundr - dtogsecboundl);
If[slopedtogsec < slopedtop,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = dtogsecboundl;
phaserec[(2 (j - 1)) + 4] = dtogsecboundr;
];
];
];
If[record[m + 5] == 3 && record[m + 6] == 2 && record[m + 2] == 1 && record[m + 3] == 3 &&
record[m + 6] < phaserec[2] && phaserec[(2 (j - 1)) + 3] == gtopsecboundl,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = gtopsecboundl;
phaserec[(2 (j - 1)) + 4] = gtopsecboundr;
];
If[ydtopboundl > ydtopboundr && phaserec[(2 (j - 1)) + 3] ≠ dtogsecboundl && phaserec[(2 (j - 1)) +
3] ≠ gtopsecboundl,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = dtopboundl;
phaserec[(2 (j - 1)) + 4] = dtopboundr;
];
];
If[record[m + 5] == 2 && record[m + 6] == 3 && record[m + 7] < phaserec[2],
ptod = FindRoot[{diffdb == diffpal, yintd == yintp}, {al, record[m + 7] - 0.05, 0.5, 1}, {b, record[m + 7]
+ 0.05, 0.226, 1}, MaxIterations -> 200];
ptodboundl = al /. ptod[[1]];
ptodboundr = b /. ptod[[2]];
yptodboundl = yμptot /. al -> ptodboundl;
yptodboundr = yμdtot /. b -> ptodboundr;
slopeptod = (yptodboundr  - yptodboundl)/(ptodboundr - ptodboundl);
If[record[m + 5] == 2 && record[m + 6] == 3 && record[m + 8] == 3 && record[m + 9] == 1 &&
record[m + 10] < phaserec[2],
ptogsec = FindRoot[{diffpal == diffgc, yintp == yintg}, {al, record[m + 7], 0.5, 1}, {c, record[m + 10],
0.112, 1}, MaxIterations -> 200];
```

```
ptogsecboundl = al /. ptogsec[[1]];
ptogsecboundr = c /. ptogsec[[2]];
If[ptogsecboundl > 0.5 && ptogsecboundl < 1 && ptogsecboundr > 0.112 && ptogsecboundr < 1,
yptogsecboundl = yμptot /. al -> ptogsecboundl;
yptogsecboundr = yμgtot /. c -> ptogsecboundr;
slopeptogsec = (yptogsecboundr - yptogsecboundl)/(ptogsecboundr - ptogsecboundl);
If[slopeptogsec < slopeptod,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = ptogsecboundl;
phaserec[(2 (j - 1)) + 4] = ptogsecboundr;
];
];
];
If[record[m + 5] == 2 && record[m + 6] == 3 && record[m + 2] == 1 && record[m + 3] == 2 &&
record[m + 6] < phaserec[2] && phaserec[(2 (j - 1)) + 3] == gtodsecboundl,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = gtodsecboundl;
phaserec[(2 (j - 1)) + 4] = gtodsecboundr;
];
If[yptodboundl > yptodboundr && phaserec[(2 (j - 1)) + 3] ≠ ptogsecboundl && phaserec[(2 (j - 1)) +
3] ≠ gtodsecboundl,
j = j + 1;
phaserec[(2 (j - 1)) + 3] = ptodboundl;
phaserec[(2 (j - 1)) + 4] = ptodboundr;
];
];
Null, {i, 1, n}]
Do[
If[phaserec[i] > phaserec[2],
phaserec[i] = 1];
Null, {i, 3, (2 (j - 1)) + 4}];
Do[
Write[phasematrix, phaserec[i], "     ", phaserec[1]];
Null, {i, 2, (2 (j - 1)) + 4}];
totalst = ToString[phaserec[1]];
Do[
stpart = ToString[phaserec[i]];
totalst = StringJoin[totalst, "    ", stpart];
Null, {i, 2, (2 (j - 1)) + 4}];
Write[yphasematrix, totalst];
Null, {k, 0, 31, 0.5}]
  Close[phasematrix];
  Close[yphasematrix];
  phasedata = Import["C:\phasematrix_t", "Table"];
  phaseplot = ListPlot[phasedata, DefaultFont -> {"Times", 8}, FrameLabel -> {"φw", "Δ T / °C"},
FrameStyle -> AbsoluteThickness[0.5], DisplayFunction :> $DisplayFunction, Frame -> True,
PlotRange -> {{0, 0.995}, {0, 30}}, AxesOrigin -> {0, 0}];
  Null
```