

NAME

kh – compute the vapor-liquid constant in water or heavy water.

LIBRARY

iapws library (*libiapws*, *-liapws*)

SYNOPSIS

```

Fortran      pure subroutine kd(T,gas,heavywater,k)
              real(dp), intent(in), contiguous :: T(:)
              character(len=*), intent(in) :: gas
              integer(int32), intent(in) :: heavywater
              real(dp), intent(out), contiguous :: k(:)

C            void iapws_g704_kd(double *T,char *gas,int  heavywater,double *k,int  size_gas,size_t
              size_T);

Python       kd(T:np.ndarray,gas:str,heavywater:bool=False)->Union[np.ndarray,float]:

```

DESCRIPTION

Compute the vapor-liquid constant kD for a given temperature T in water or heavy water.

Arguments:

<i>T</i> (:)	Temperature in K.
<i>gas</i>	Gas.
<i>heavywater</i>	Flag if D2O (1) is used or H2O (0).
<i>k</i> (:)	Vapor-liquid constant in MPa.

RETURN VALUE

k is filled with NaN if gas is not found.

EXAMPLES

Fortran:

```

use iapws
real(dp) :: T(1), k(1)
T(1) = 25.0_dp + 273.15_dp
call kd(T, "O2", 0, k)

```

C:

```

#include "iapws.h"
double T = 25.0 + 273.15;
char *gas = "O2";
double k;
iapws_g704_kd(&T, gas, 0, &k, strlen(gas), 1);

```

Python:

```

import numpy as np
import pyiapws

```

```
T = np.asarray([25.0])
k = pyiapws.kd(T+273.15, "02", False)
print(k)
```

SEE ALSO

***iapws*(1) *iapws_lib*(3) *iapws_psat*(3) *iapws_Tsat*(3) *iapws_kh*(3) *iapws_kd*(3) *iapws_kw*(3)
iapws_Tc(3) *iapws_pc*(3) *ciaaw*(3), *codata*(3),**