



Universiteit
Leiden
The Netherlands

Exploring structure dependencies of gas-surface interactions with curved single crystals

Auras, S.V.

Citation

Auras, S. V. (2021, March 11). *Exploring structure dependencies of gas-surface interactions with curved single crystals*. Retrieved from <https://hdl.handle.net/1887/3151627>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/3151627>

Note: To cite this publication please use the final published version (if applicable).

Cover Page



Universiteit Leiden



The handle <https://hdl.handle.net/1887/3151627> holds various files of this Leiden University dissertation.

Author: Auras, S.V.

Title: Exploring structure dependencies of gas-surface interactions with curved single crystals

Issue Date: 2021-03-11

A

SUPPLEMENTARY INFORMATION FOR CHAPTER 3: SURFACE STRUCTURE CHARACTERIZATION OF A CURVED Pt CRYSTAL WITH HIGHLY KINKED STEPS

A.1. HISTOGRAMS OF FACET DISTRIBUTIONS IN KINKED STEPS

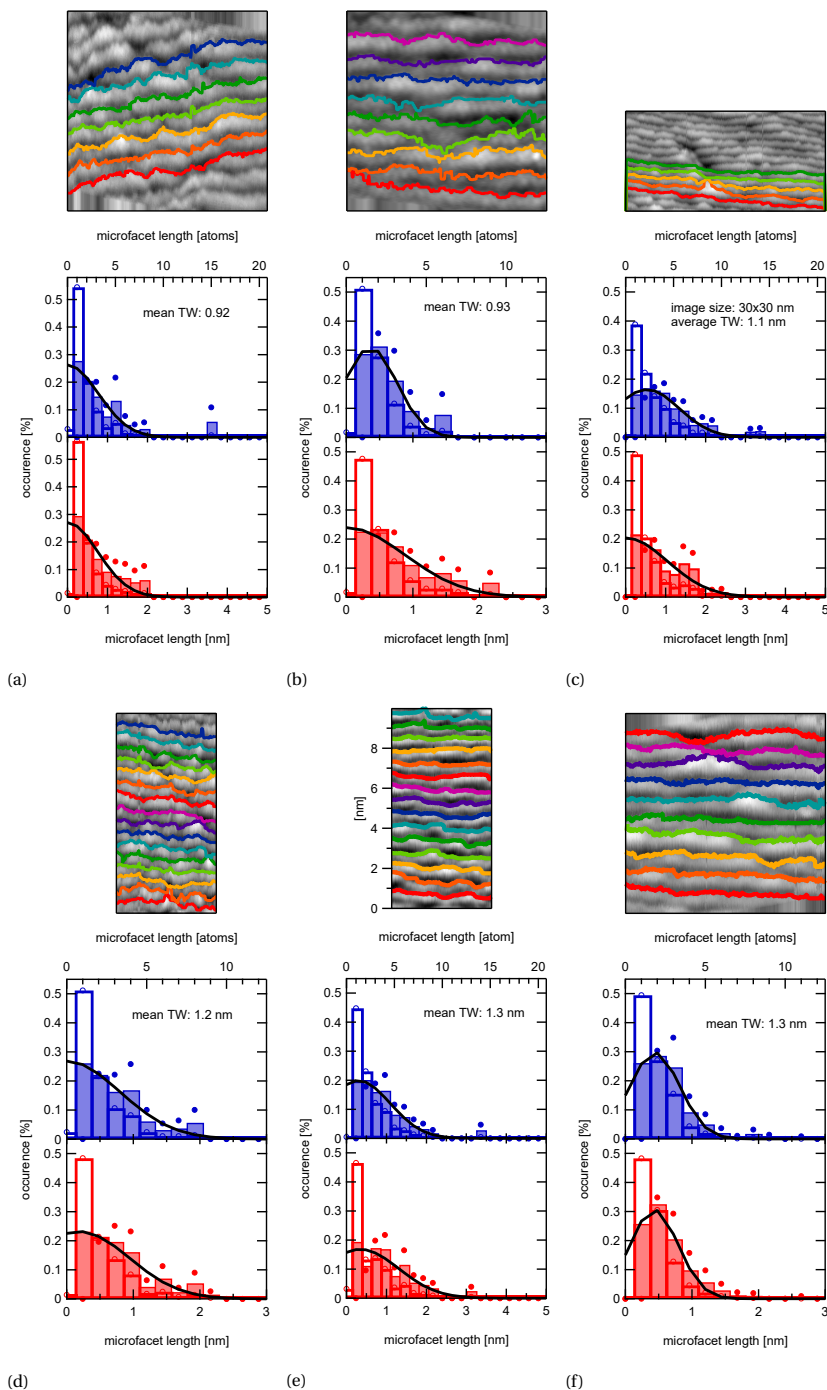


Figure A.1: a)–m) STM images and distributions of segment lengths along the kinked step edges. Kink densities and Gaussian fits are used in figure 3.9.

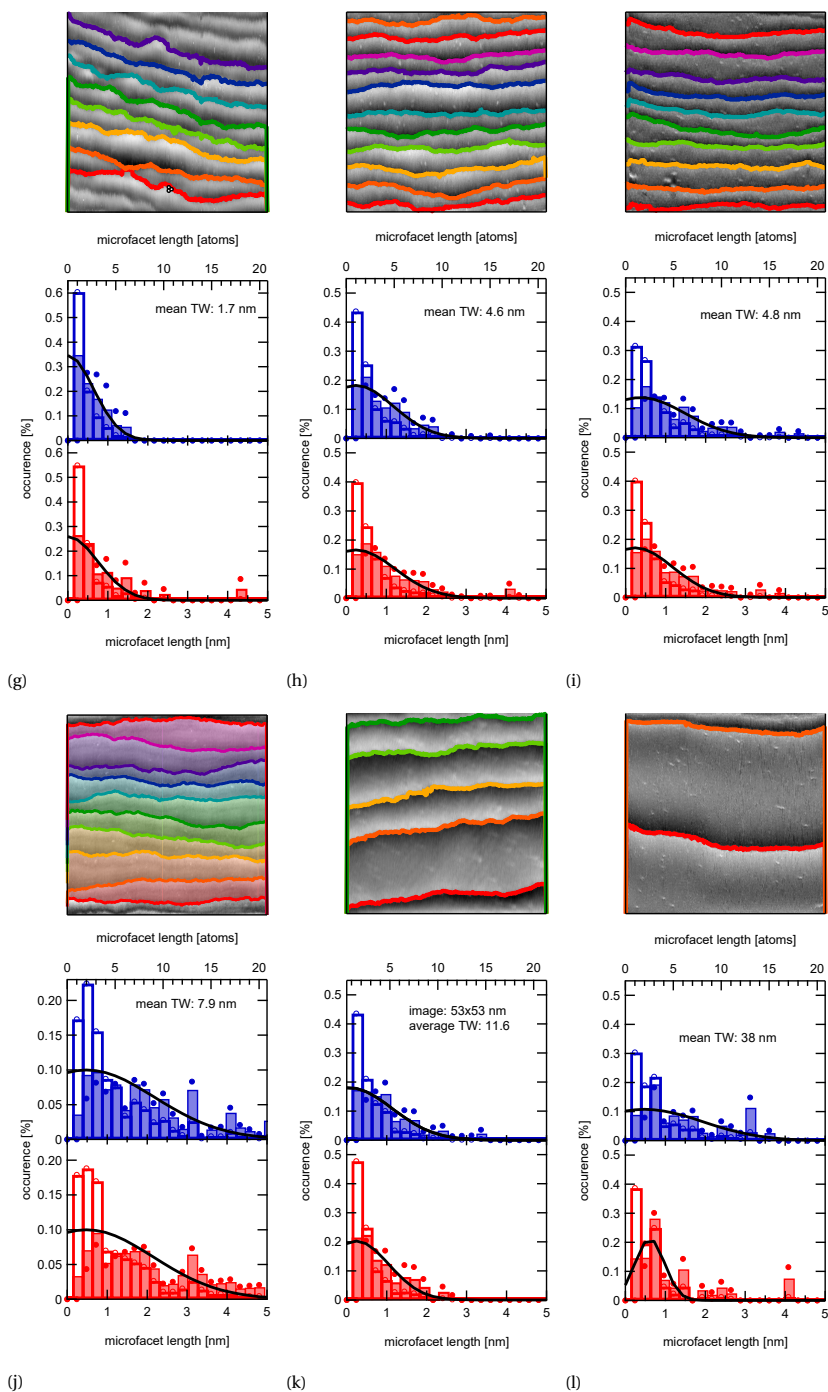
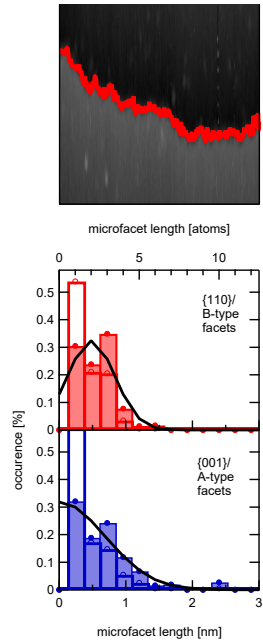


Figure A.1: a)–m) STM images and distributions of segment lengths along the kinked step edges. Kink densities and Gaussian fits are used in figure 3.9.



(m)

Figure A.1: a)–m) STM images and distributions of segment lengths along the kinked step edges. Kink densities and Gaussian fits are used in figure 3.9.

B

SUPPLEMENTARY INFORMATION FOR CHAPTER 4: SCALING PT-CATALYZED HYDROGEN DISSOCIATION ON CORRUGATED SURFACES

B.1. INITIAL STICKING PROBABILITIES AT STEP SITES

The slope Σ_0 , as obtained from the linear fits of S_0 at different positions of the curved crystal, has a unit of nm due to the x-axis (step density) being in units of nm^{-1} .

$$S_0(SD) = S_0^{(111)} + \Sigma_0[\text{nm}] \cdot SD[\text{nm}^{-1}] \quad (\text{B.1})$$

An alternative way to quantify the additional reactivity caused by steps is to consider the fraction of the surface area occupied by steps (f^{step}), instead of mere step density:

$$S_0 = S_0^{(111)} \cdot (1 - f^{step}) + S_0^{step} \cdot f^{step} \quad (\text{B.2})$$

Obtaining a unitless S_0^{step} may feel more intuitive when comparing reactivities of the step edge with reactivities of the (111) terrace. However, determining a value f^{step} requires an assumption on the size of a step within the unit cell. Defining it as one atom row at the upper step edge will give a different step size for kinked steps (0.277 nm) than A- and B-type steps (0.240 nm), due to their orientation on the (111) terrace. Considering the upper edge and lower cusp of each step type will yield different step sizes for each type. The same step density may thus correspond to three different values for f^{step} , resulting in a weighing of S_0^{step} purely based on atom arrangement.

We therefore consider the weighing of reactivity by step density in equation B.1 a more appropriate approach when comparing different step types. The unit, nm, in

turn can be interpreted as length over which the step is reactive, i.e. a 'chemical step size'. The implications of Σ_0 as a physical length are further discussed in the main text.

The decreasing fraction of (111) terraces on the stepped surfaces, $(1 - f^{step})$ is neglected there, due to the inherently small value of $S_0^{(111)}$.

B.2. COORDINATION AT THE LOWER STEP EDGE

For atoms at the lower edges, CN and \overline{CN} values for the three step types vary slightly, as listed in table B.1. The coordination number in the lower plane of the A-type step is lower than both the B-type step and any combined contribution of inner and outer kinks. The A-type should thus be most reactive. If only outer kinks contribute to reactivity, the {210} steps should react similarly to the A-type. The B-type should be least reactive.

	upper edge				lower edge			
	A	B	K_{in}	K_{out}	A	B	K_{in}	K_{out}
CN	7	7	8	6	10	11	11	10
\overline{CN}	5.50	5.50	6.17	4.83	8.75	9.50	9.58	8.58

Table B.1: Coordination numbers (CN) and generalized coordination numbers (\overline{CN}) [?] of atoms at the upper and lower edge of the three step types in this study.

B.3. STRUCTURAL ANALYSIS OF THE KINKED {210} STEPS

Table B.2 lists kink atoms (K) and step atoms (A or B) in fully-kinked (a) and longer segments (b,c,d...). In the ultimate case of step faceting, a single {210} step would break into very large segments of the A- and B-type. This reconstruction of the {210} step has a combined total length of 1.16 times that of a straight line connecting the first and last atom of the edge. These ideas form the basis for a model that extracts the average chemical reactivity of inner and outer kinks. We take into account the length and composition of possible step sections, the measured reactivity for A- and B-type steps, and the distribution of A- and B-sections in {210} steps as obtained by STM.

The STM images in figure 4.3b)–e) gives us distributions of segment lengths in averages stretches of step edges. We use those distributions and the atomic compositions of segments laid out in table B.2 and displayed in figure 4.3a) to calculate the number of kink atoms (N_K), as well as intermittent atoms in longer segments of A- or B-type orientation (N_A and N_B):

$$N_A = 0 \cdot a_A + 1 \cdot b_A + 2 \cdot c_A + \dots \tag{B.3}$$

segment	atoms in segment	{001}-oriented		{110}-oriented	
		K	A	K	B
a	2	1	0	1	0
b	4	1	1	1	1
c	6	1	2	1	2
d	8	1	3	1	3
⋮	⋮	⋮	⋮	⋮	⋮

Table B.2: Composition of atoms found in kinked steps with various segment lengths, as displayed in figure 4.3a) of the main text. Atoms at the top of the step edge can be considered either a direct kink atom (K), or a step atom. Depending on the orientation of the segment, step atoms are either A-type (A) or B-type (B).

$$N_B = 0 \cdot a_B + 1 \cdot b_B + 2 \cdot c_B + \cdots \tag{B.4}$$

$$N_K = a_A + a_B + b_A + b_B + c_A + c_B + \cdots \tag{B.5}$$

Here a_A is the fractional occurrence of bin a of the {001} histogram, b_A is the fractional occurrence of segments in bin b of the {001} histogram, a_B is the fractional occurrence of segments in bin a of the {110} histogram, etc...

Table B.3 lists the results, as well as kink densities and Σ_0 , as described in the main text.

image in fig.4.3	N_A	N_B	N_K	N	N_K/N	Σ_0^K
b)	152.201	122.222	200	474.423	0.422	0.747
c)	139.734	161.9	200	501.634	0.399	0.734
d)	73.6264	106.522	200	380.1484	0.526	0.667
e)	163.158	127.016	200	490.174	0.408	0.758

Table B.3: Numbers of atoms found in kinked steps with various segment lengths, as displayed in figure 4.3a), averaged to 100 segments in {001} orientation and 100 segments in {110} orientation. Atoms are considered either a kink atom (K), or an atom resembling those in type A steps (A) and type B steps (B).

C

SUPPLEMENTARY INFORMATION FOR CHAPTER 5: CHIRAL SURFACE CHARACTERISATION AND REACTIVITY TOWARD H-D EXCHANGE OF A CURVED PLATINUM CRYSTAL

C.1. TERRACE WIDTH ANALYSIS

Figure C.1a) shows the averaged terrace widths for each image, 300 μm on either side of the apex. Blue data with error bars represent each image's average terrace width and standard deviation. The red dots are, for each image, the local maximum terrace width which, together with the standard deviation, provide an idea of the terrace width distribution across the crystal. The black traced lines are the expected terrace widths as described by equation 5.1. We observe a close match with the expected terrace width except for in the $\sim 40 \mu\text{m}$ around the apex. This is more clearly visible in the zoomed-in section of the figure.

By taking the inverse of the terrace widths, we expect to see a linearly increasing step density where the slope is determined by the radius of curvature and interatomic distance. Beyond 400 μm on the (-Y) side of the crystal, terrace width distribution was low enough to simply divide the number of steps in an image by the distance between the first and last step. This data, along with the inverse of the terrace widths previously shown, is plotted in figure C.1b). We can see that all along the imaged regions of the crystal, the step density increases linearly with distance from the apex respecting the expected defect density.

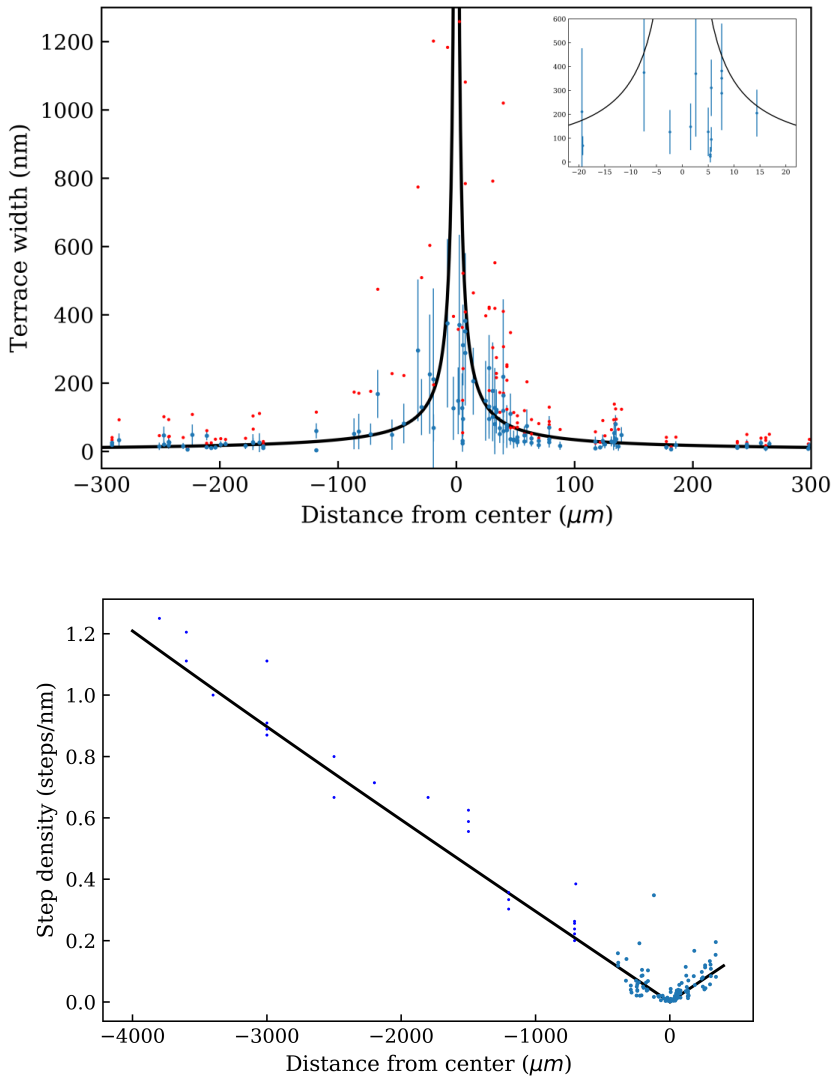


Figure C.1: a) Terrace widths as a function of distance from the apex for 330 nm on either side of the apex. Blue data points indicate the average terrace widths including the standard deviation per STM image and red data points are the maximum terrace widths per image. A subfigure in the top right shows a zoom around the center of the graph where the averages start to deviate from the expected TW curve (in black). b) Step densities as a function of position on the crystal. Black line represents the expected step density taken from the inverse of equation 5.1.

D

STM ANALYSIS PROCEDURE TO DETERMINE OUTLINES OF STEPS IN STM IMAGES WITH IGOR PRO

D.1. PROCEDURE 1

One pixel wide line profiles in vertical direction are created across the entire STM image.

Function alllineprofiles(i,j,name,x)

variable i//first lineprofile

variable j//last line profile

variable x//number of pixels in image

wave name

make/n=2 xxx,yyy

wave xxx ; wave yyy

string srcwave ; string xwave ; string ywave

Display

For (i=i; i<j+1;i +=1)

xxx=i,i ; yyy=0,(x-1)

string num=num2str(i) ; string profile="Prof"+num

ImageLineProfile srcwave=name, xwave=xxx, ywave=yyy, width=3

duplicate W_ ; ImageLineProfile \$profile

```

wave pro=$profile
Appendtograph pro vs W_LineProfileY
string profiledif="Prof"+num+"_dif"
differentiate pro/X=W_LineProfileY/D=$profiledif
wave dif=$profiledif
EndFor
End

```

D.2. PROCEDURE 2

Step edges are found along the horizontal direction, across all line profiles.

Function limitprofiles(i,j,k,x,name,A,B,n)

```

variable i //first x
variable j //last x that you take a lineprofile of
variable A //first point of y values in which to find the step
variable B //last point of y values in which to find the step
variable n //how manyth step in image
wave name //name of image wave
variable k //imagesize (nm)
variable x//number of pixels in image

string st="step"+num2str(n)
make/n=(j+1) $st
wave step=$st

For (i=i; i<j+1;i +=1)
string num=num2str(i) ; string profile="Prof"+num
wave pro=$profile
string profiledif="Prof"+num+"_dif"
wave dif=$profiledif
wavestats/Q/R=(A,B) dif
step[i]=V_maxRowLoc

```

EndFor

```
step[0]=step[1] ; step[j]=step[j-1]
```

```
string stnm="step"+num2str(n)+"_nm"
```

```
duplicate/O step $stnm
```

```
wave step_nm=$stnm
```

```
step_nm=step/x*k
```

```
string stx="stepX_nm"
```

```
duplicate/O step $stx
```

```
wave stepX=$stx
```

```
stepX=p/(j+1)*k
```

AppendtoGraph step vs W_LineProfileY

```
ModifyGraph mode=4,marker=19
```

```
If(n==2)
```

```
ModifyGraph rgb(step2)=(65280,21760,0)
```

```
elseif(n==3)
```

```
ModifyGraph rgb(step3)=(65280,43520,0)
```

```
elseif(n==4)
```

```
ModifyGraph rgb(step4)=(26112,52224,0)
```

```
elseif(n==5)
```

```
ModifyGraph rgb(step5)=(0,39168,0)
```

```
elseif(n==6)
```

```
ModifyGraph rgb(step6)=(0,39168,39168)
```

```
elseif(n==7)
```

```
ModifyGraph rgb(step7)=(0,9472,39168)
```

```
elseif(n==8)
```

```
ModifyGraph rgb(step8)=(19712,0,39168)
```

```
elseif(n==9)
```

```
ModifyGraph rgb(step9)=(52224,0,41728)
```

```
elseif(n==11)
```

```
ModifyGraph rgb(step11)=(65280,21760,0)
```

```

elseif(n==12)
ModifyGraph rgb(step12)=(65280,43520,0)
elseif(n==13)
ModifyGraph rgb(step13)=(26112,52224,0)
elseif(n==14)
ModifyGraph rgb(step14)=(0,39168,0)
elseif(n==15)
ModifyGraph rgb(step15)=(0,39168,39168)
elseif(n==16)
ModifyGraph rgb(step16)=(0,9472,39168)
elseif(n==17)
ModifyGraph rgb(step17)=(19712,0,39168)
elseif(n==18)
ModifyGraph rgb(step18)=(52224,0,41728)
endif

string one=nameofwave(step)+"#1"
removefromgraph/Z $one
End

```

D.3. PROCEDURE 3

The terrace widths in between step edges are determined.

Function maketerraces(j,k,x)

```

variable j //number of steps
variable k //imagesize (nm)
variable x//number of pixels in image
variable i ; variable a=0

For (i=1; i<j;i +=1)
string st="step"+num2str(i)
wave step=$st
string st2="step"+num2str(i+1)

```



```

wave step2=$st2
string ter="terrace"+num2str(i)
duplicate/O step $ter
wave terrace = $ter
terrace = step2-step
string ternm="terrace"+num2str(i)+"_nm"
duplicate/O terrace $ternm
wave terrace_nm = $ternm
terrace_nm=terrace/x*k
wavestats/Q terrace_nm
a += V_avg
EndFor

print "avg terrace width [nm]" ; print a/(j-1)
End

```

D.4. PROCEDURE 4

Terrace width distribution histograms are created from the available terrace widths.

Function terracehist(j,k,x)

```

variable j //number of terraces
variable k //image size (nm)
variable x//number of pixels in image
variable l=k*2/0.277
variable i

display

For (i=1; i<j+1;i +=1)
string ternm="terrace"+num2str(i)+"_nm"
wave terrace_nm = $ternm
string hist="terrace"+num2str(i)+"_Hist"
string number=num2str(l)

```

```

Make/N=(l)/O $hist;
wave histo=$hist
Histogram/C/R=[1,(x-2)]/B=0,0.1385,l terrace_nm,histo
AppendToGraph histo

If (i==1)
string hall="terrace_all_hist"
duplicate histo $hall
wave h_all=$hall
appendtograph h_all
ReorderTraces terrace1_Hist,terrace_all_hist
Else
h_all+=histo
endif
Endfor

ModifyGraph mode=5,hbFill=3 ; ModifyGraph tick=2,mirror=1,standoff=0
Label bottom "terrace width (nm) (1 bin = 0.1385 nm = 1/2 atom row)" ; Label left "counts"
End

```

D.5. PROCEDURE 5

Segments along a step edge (in between kink sites) are determined and their length is determined.

```

Function makeblock(step,stepX)

wave step ; wave stepX
variable x//image size
string st=nameofwave(step) //name of step wave
string stx=nameofwave(stepX) // name of x wave of step wave
duplicate/O step $st+"_dif"
wave step_dif = $st+"_dif"
Differentiate/METH=1 step/X=stepX/D=step_dif;
duplicate/O step_dif $st+"_block"

```

```
wave block = $st+"block"
```

```
variable i ; variable j = numpnts(step) //number of points in wave
```

```
For (i=0; i<j; i +=1)
```

```
If(step_diff[i]>0.1)
```

```
block[i]=1
```

```
Elseif(step_diff[i]<-0.1)
```

```
block[i]=-1
```

```
Else
```

```
block[i]=0
```

```
EndIf
```

```
EndFor
```

```
duplicate/O stepX $st+"blockpos"
```

```
wave position=$st+"blockpos"
```

```
duplicate/O block $st+"blockvalue"
```

```
wave value=$st+"blockvalue"
```

```
duplicate/O position $st+"blockpos_fw"
```

```
wave posfw=$st+"blockpos_fw"
```

```
duplicate/O block $st+"blockvalue_fw"
```

```
wave valuefw=$st+"blockvalue_fw"
```

```
duplicate/O position $st+"blockpos_bw"
```

```
wave posbw=$st+"blockpos_bw"
```

```
duplicate/O block $st+"blockvalue_bw"
```

```
wave valuebw=$st+"blockvalue_bw"
```

```
For (i=0; i<j; i +=1)
```

```
if(valuefw(i)==0)
```

```
deletepoints i,1,posfw ; deletepoints i,1,valuefw
```

```
i=i-1
```

```
EndIf
```

```
EndFor
```

```

For (i=numpnts(step); i>0;i -=1)
If(valuebw[i]==0)
deletpoints i,1,posbw ; deletpoints i,1,valuebw
EndIf
EndFor

```

```

For (i=1; i<numpnts(valuefw);i +=1)
if(valuefw(i)==valuefw[i-1])
deletpoints i,1,posfw ; deletpoints i,1, valuefw
i=i-1
EndIf
EndFor

```

```

For (i=1; i<numpnts(valuefw);i +=1)
if(posfw(i)-posfw(i-1)<0.12)
deletpoints i,1,posfw ; deletpoints i,1, valuefw
i=i-1
EndIf
EndFor

```

```

For (i=1; i<numpnts(valuefw);i +=1)
if(valuefw(i)==valuefw[i-1])
deletpoints i,1,posfw ; deletpoints i,1, valuefw
i=i-1
EndIf
EndFor

```

```

For (i=0; i<numpnts(valuebw) ;i +=1)
If(valuebw(i)==valuebw(i+1))
deletpoints i,1,posbw ; deletpoints i,1,valuebw
i=i-1
EndIf
EndFor

```

```

For (i=0; i<numpnts(valuebw) ;i +=1)
If(posbw(i+1)-posbw(i)<0.12)
deletpoints i,1,posbw ; deletpoints i,1,valuebw
i=i-1
EndIf
EndFor

```

```

For (i=0; i<numpnts(valuebw) ;i +=1)
If(valuebw(i)==valuebw(i+1))
deletpoints i,1,posbw ; deletpoints i,1,valuebw
i=i-1
EndIf
EndFor

```

```

duplicate/O posfw $st+"_facet_fw"
wave facetfw=$st+"_facet_fw"
For (i=1; i<j;i +=1)
facetfw[i]=posfw[i]-posfw[i-1]
Endfor

```

```

duplicate/O posbw $st+"_facet_bw"
wave facetbw=$st+"_facet_bw"
For (i=0; i<j;i +=1)
facetbw[i]=posbw[i+1]-posbw[i]
Endfor

```

```

duplicate/O facetfw $st+"_plusfacet_fw"
wave plusfw = $st+"_plusfacet_fw"
duplicate/O valuefw $st+"_plusvalue_fw"
wave plusvalfw = $st+"_plusvalue_fw"
duplicate/O facetfw $st+"_minusfacet_fw"
wave minusfw = $st+"_minusfacet_fw"
duplicate/O valuefw $st+"_minusvalue_fw"
wave minusvalfw = $st+"_minusvalue_fw"

```

```

For (i=0; i<j;i +=1)
If(plusvalfw[i]!=1)
deletepoints i,1,plusfw ; deletepoints i,1,plusvalfw
i=i-1
Endif
Endfor

For (i=0; i<j;i +=1)
If(plusfw[i]==0)
deletepoints i,1,plusfw ; deletepoints i,1,plusvalfw
i=i-1
Endif
Endfor

For (i=0; i<j;i +=1)
If(minusvalfw[i]!=-1)
deletepoints i,1,minusfw ; deletepoints i,1,minusvalfw
i=i-1
Endif
Endfor

For (i=0; i<j;i +=1)
If(minusfw[i]==0)
deletepoints i,1,minusfw ; deletepoints i,1,minusvalfw
i=i-1
Endif
Endfor

duplicate/O facetbw $st+"plusfacet_bw"
wave plusbw = $st+"plusfacet_bw"
duplicate/O valuebw $st+"plusvalue_bw"
wave plusvalbw = $st+"plusvalue_bw"
duplicate/O facetbw $st+"minusfacet_bw"

```

```

wave minusbw = $st+"minusfacet_bw"
duplicate/O valuebw $st+"minusvalue_bw"
wave minusvalbw = $st+"minusvalue_bw"

For (i=0; i<j;i +=1)
  If(plusvalbw[i]!=1)
    deletepoints i,1,plusbw ; deletepoints i,1, plusvalbw
  i=i-1
Endif
Endfor

For (i=0; i<j;i +=1)
  If(plusbw[i]==0)
    deletepoints i,1,plusbw ; deletepoints i,1,plusvalbw
  i=i-1
Endif
Endfor

For (i=0; i<j;i +=1)
  If(minusvalbw[i]!=-1)
    deletepoints i,1,minusbw ; deletepoints i,1,minusvalbw
  i=i-1
Endif
Endfor

For (i=0; i<j;i +=1)
  If(minusbw[i]==0)
    deletepoints i,1,minusbw ; deletepoints i,1,minusvalbw
  i=i-1
Endif
Endfor

Make/O/N=200/O $st+"_plus_hist_fw"
wave plh_fw=$st+"_plus_hist_fw"

```

```

Make/O/N=200/O $st+"_minus_hist_fw"
wave mnh_fw = $st+"_minus_hist_fw"
Histogram/C/B=-0.12,0.24,200 plusfw,plh_fw
Histogram/C/B=-0.12,0.24,200 minusfw,mnh_fw
Make/O/N=200/O $st+"_plus_hist_bw"
wave plh_bw=$st+"_plus_hist_bw"
Make/O/N=200/O $st+"_minus_hist_bw"
wave mnh_bw = $st+"_minus_hist_bw"
Histogram/C/B=-0.12,0.24,200 plusbw,plh_bw
Histogram/C/B=-0.12,0.24,200 minusbw,mnh_bw
duplicate/O plh_fw $st+"_plus_hist_avg"
wave plh_avg = $st+"_plus_hist_avg"
duplicate/O mnh_fw $st+"_minus_hist_avg"
wave mnh_avg = $st+"_minus_hist_avg"
plh_avg = (plh_fw+plh_bw)/2
mnh_avg = (mnh_fw+mnh_bw)/2

killwaves block ; killwaves position ; killwaves value ; killwaves facetfw ; killwaves facetbw
killwaves plusvalfw ; killwaves plusvalbw ; killwaves minusvalfw ; killwaves minusvalbw
End

```

D.6. PROCEDURE 6

From this part of the code, distributions of segment lengths are determined separately for {001}- and {110}-oriented segments.

```

Function plotstephistograms(j)

variable j ; variable i

Display

For (i=1; i<j+1;i +=1)

string plus = "step"+num2str(i)+"_nm_plus_hist_avg"
string minus = "step"+num2str(i)+"_nm_minus_hist_avg"

wave plh_avg = $plus
wave mnh_avg=$minus

```



```
string l="L"+num2str(i) ; string b="B"+num2str(i) ; string m="L"+num2str(i+j) ; string t=
"T"+num2str(i)
```

```
Appendtograph/L=$l/B=$b plh_avg
```

```
AppendToGraph/L=$m/T=$t mnh_avg
```

```
variable a = (i-1)/j ; variable c= (i/j)
```

```
ModifyGraph axisEnab($b)=a,c ; ModifyGraph axisEnab($t)=a,c
```

```
ModifyGraph axisEnab($l)=0,0.5 ; ModifyGraph axisEnab($m)=0.5,1
```

```
ModifyGraph freePos($l)=a,kwFraction ; ModifyGraph freePos($m)=a,kwFraction
```

```
ModifyGraph freePos($b)=0,$l ; ModifyGraph freePos($t)=0,kwFraction
```

```
ModifyGraph mode=5,hbFill=4 ; ModifyGraph standoff=0 ; ModifyGraph axisOnTop=1
```

```
string mna = "step_all_nm_min_hist"
```

```
string pla = "step_all_nm_plus_hist"
```

```
If(i==1)
```

```
duplicate/O mnh_avg $mna
```

```
wave mnall=$mna
```

```
duplicate/O plh_avg $pla
```

```
wave plall=$pla
```

```
Else
```

```
mnall += mnh_avg ; plall += plh_avg
```

```
Endif
```

```
Endfor
```

```
string plat="step_all_plus_hist_atoms"
```

```
string mnat="step_all_minus_hist_atoms"
```

```
duplicate/O mnall $mnat
```

```
wave mn_at=$mnat
```

```
duplicate/O plall $plat
```

```
wave pl_at=$plat
```

```
mn_at = mnall*p ; pl_at = plall*p
```

```
End
```


LIST OF PUBLICATIONS

CHAPTER 2

Recent advances in the use of curved single crystal surfaces

Sabine V. Auras and Ludo B.F. Juurlink *Progress in Surface Science*, submitted.

CHAPTER 4

Scaling Platinum-Catalyzed Hydrogen Dissociation on Corrugated Surfaces

Sabine V. Auras, Richard van Lent, Dima Bashlakov, Jessika M. Piñeiros Bastidas, Tycho Roorda, Rick Spierenburg and Ludo B.F. Juurlink *Angewandte Chemie International Edition*, **59** 20973-20979 (2020).

CHAPTER 5

Chiral Surface Characterisation and Reactivity Toward H-D Exchange of a Curved Platinum Crystal

Tycho Roorda*, **Sabine V. Auras*** and Ludo B.F. Juurlink *Topics in Catalysis*, **63**, 1558-1568 (2020).

CHAPTER 6

It's not just the defects - a curved crystal study of H₂O desorption from Ag

Sabine V. Auras, Robert A. B. van Bree, Dima L. Bashlakov, Richard van Lent, and Ludo B. F. Juurlink *Physical Chemistry Chemical Physics*, **21** 15422–15430 (2019).

*These authors contributed equally to the manuscript

OTHER PUBLICATIONS

On the correlation of structure and catalytic performance of VPO catalysts

Sebastian Böcklein, Gerhard Mestl, **Sabine V. Auras**, and Joost Wintterlin *Topics in Catalysis*, **60**, (19-20) 1682–1697 (2017).

Step-type and step-density influences on CO adsorption probed by reflection absorption infrared spectroscopy using a curved Pt(111) surface

Anton J. Walsh, Richard van Lent, **Sabine V. Auras**, Michael A. Gleeson, Otto T. Berg, Ludo B.F. Juurlink *J. Vac. Sci. Tech. A* **35**, (3) 03E102 (2017)

Site-specific reactivity of molecules with surface defects – the case of H₂ dissociation on Pt

Richard van Lent, **Sabine V. Auras**, Kun Cao, Anton J. Walsh, Michael A. Gleeson, Ludo B. F. Juurlink *Science* **363**(6423) 155-157 (2019)

CURRICULUM VITAE

Sabine Auras is a trained chemist, who received most of her education at the Ludwig-Maximilians University in Munich, Germany. After a broad undergraduate education (B.Sc. in Chemistry and Biochemistry, 2013), she developed an interest in physical chemistry during her Master studies (M.Sc. in Chemistry in 2015).

During an Erasmus stay at Leiden University in the Netherlands, Sabine had the opportunity to join the group of Dr. Ludo Juurlink for six months and studied the structure of a curved Ag(001) crystal. Subsequently, she returned to Germany to complete her M.Sc. degree. During her master thesis project, she investigated the correlations between structure and catalytic performance of VPO catalysts, under supervision of Dr. S. Böcklein and Dr. G. Mestl at Clariant AG. VPO is the industrial catalyst for the selective oxidation of maleic anhydride from *n*-butane, a reaction similar to ethylene oxidation.

Sabine started her PhD at Leiden University in December 2015, under supervision of Dr. Ludo Juurlink, where she studied the structure and reactivity of curved crystal surfaces with a variety of different techniques. Sabine explored the fundamental aspects of gas-surface dynamics, using multiple curved metal crystals (Ag, Pt, NiAl). A combination of surface structure techniques (LEED, AES, STM), spectroscopy (RAIRS) and mass spectrometry based techniques (TPD, TOF, King-and-Wells) have been the key to unraveling underlying structure dependencies in such reactions. A main focus of the PhD project has been the introduction of STM to the Juurlink group, by reviving an old VT-STM and initiating collaborations with STM-focused groups in Spain and the USA. Namely, Sabine had short-term stays abroad with the groups of Prof. Ortega (CFM, San Sebastian, Spain) and Prof. Killelea (Loyola University, Chicago, USA).

The results of her PhD work are presented in this thesis and resulted in several publications. Sabine also presented her work at various Dutch and international conferences, e.g. the Gordon Research Seminar and Conference on Chemical Reactions at Surfaces in 2017, the Gordon Research Seminar and Conference on Dynamics at Surfaces (2017 and 2019), several Dutch Scanning Probe Microscopy Days and HRSMC Symposia, the Reedijk Symposium (2017), and the AVS Prairie Chapter meeting (2017). In particular, she gave talks at the Gordon Research Seminar on Dynamics at Surfaces in 2019 and the HRSMC Symposium in 2019.

Sabine attended the NIOK course *Catalysis, An Integrated Approach* in 2017 and the HRSMC *Tulip School on Modern Advances in Spectroscopy* in 2018. She was also a

member of the HRSMC PhD platform. Throughout her time in Leiden, Sabine supervised and coached several Bachelor and Master students, and assisted teaching in the *algemene en anorganische chemie* and *practicum basisvaardigheden* courses of the *Molecular Science and Technology* B.Sc. programme of Leiden University and TU Delft.

Sabine will start her PostDoc at the Centro de Física de Materiales in Donostia, Spain in spring 2021.