HUPP 3b

David Tonderski - davton

1 Uppgift 1

Koden med alla ändringar (alltså inte den som användes för den här uppgiften) bifogas i Appendix. Det koden plottade med ändringarna som jag utförde i uppgift 1 visas i figur 1.

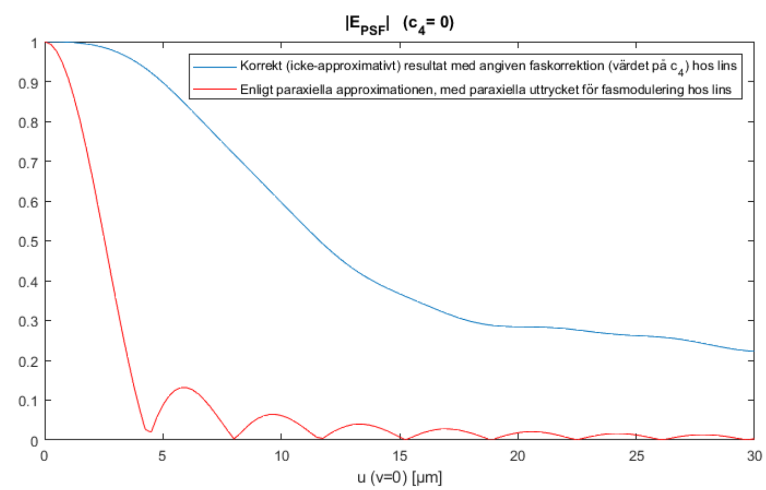

Figure 1: Verklig vs paraxiell med $c_4 = 0$.

2 Uppgift 2

Jag fick att $c_4 = \frac{k}{8f^3} \approx 4.8 \cdot 10^4$. Värdet verkar stämma bra i figur 2, där PSF:en inte längre är blaffig, utan nästan följer den paraxiella approximationen. Det bekräftas med hjälp av *Sampla_om_radiell_till_2D.m*, vilket visas i figur 3.

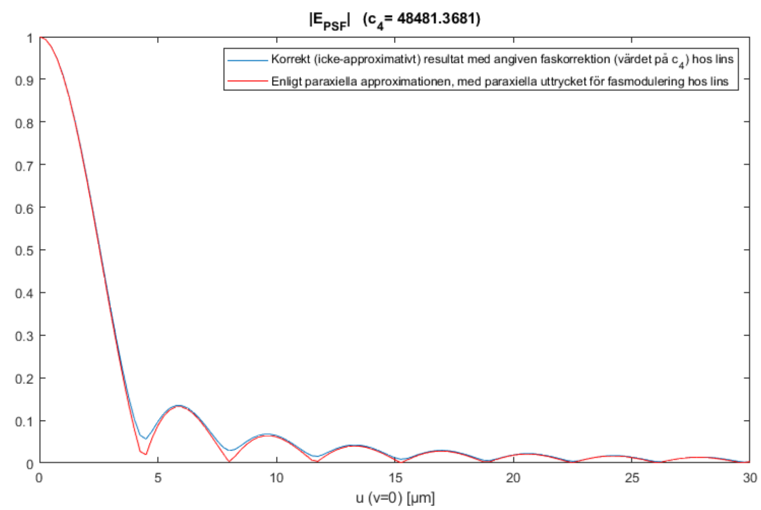

Figure 2: Verklig vs paraxiell med $c_4 \approx 4.8 \cdot 10^4$.

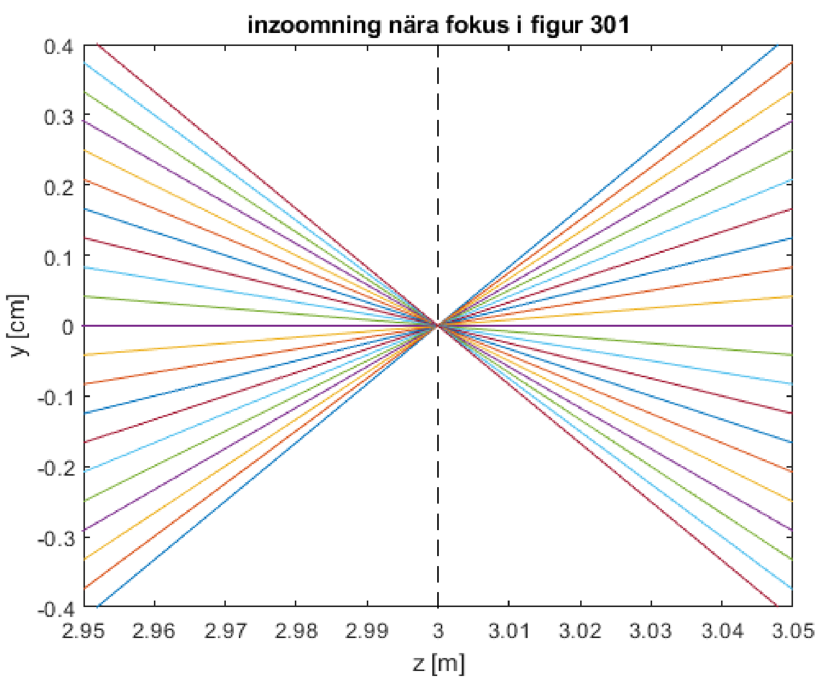

Figure 3: Plot från *rays_HiRISE.m* med $c_4 \approx 4.8 \cdot 10^4$.

3 Uppgift 3

Vi har att $\Delta\phi = k\frac{r^4}{8f^3}$, samt att $\Delta z = \frac{\Delta\phi}{2k}$. Därmed får vi att $\Delta z = \frac{r^4}{16f^3}$.

4 Uppgift 4

Plotten visas i figur 4. Hålet gör märkbar, men inte enorm skillnad, och inget svart hål kommer att uppstå i bilden. Jag tror inte att hålet kommer att göra väldigt stor skillnad i bilden. Det är intressant att även om man ökar hålstorleken till att nästan tecka det inkommande ljuset ($d_{hål} = 23$ cm), så är bilden fortfarande rätt bra; det går att urskilja rovern, men inte fotspåren.

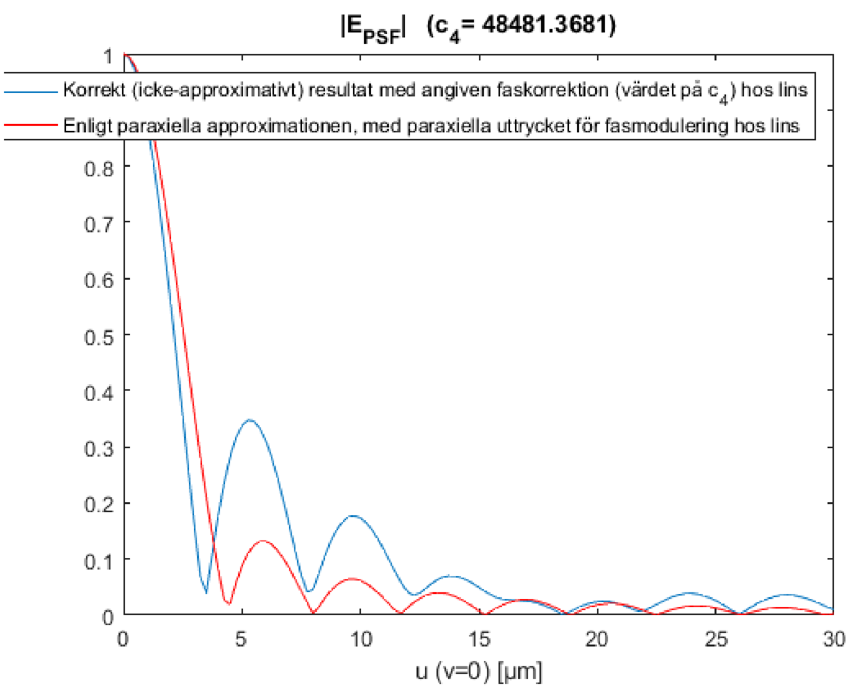

Figure 4: Verklig vs paraxiell med $c_4 \approx 4.8 \cdot 10^4$ och $d_{hål} = 15$ cm.

5 Uppgift 5

Det bästa med Hupparna är att man vet att det bästa alltid kommer på slutet.

Samplingsavståndet blir samplingsavståndet på Marsytan (400 m/5000) gånger förstoringen (3m/250km). Vi får alltså att samplingsavståndet blir $0.08 \cdot 1.2 \cdot 10^{-5} = 9.6 \cdot 10^{-7}$ m $= 0.96\,\mu$m.

Efter omsampling till 2D och en hel del in-zoomning ser PSF:en ut så här:

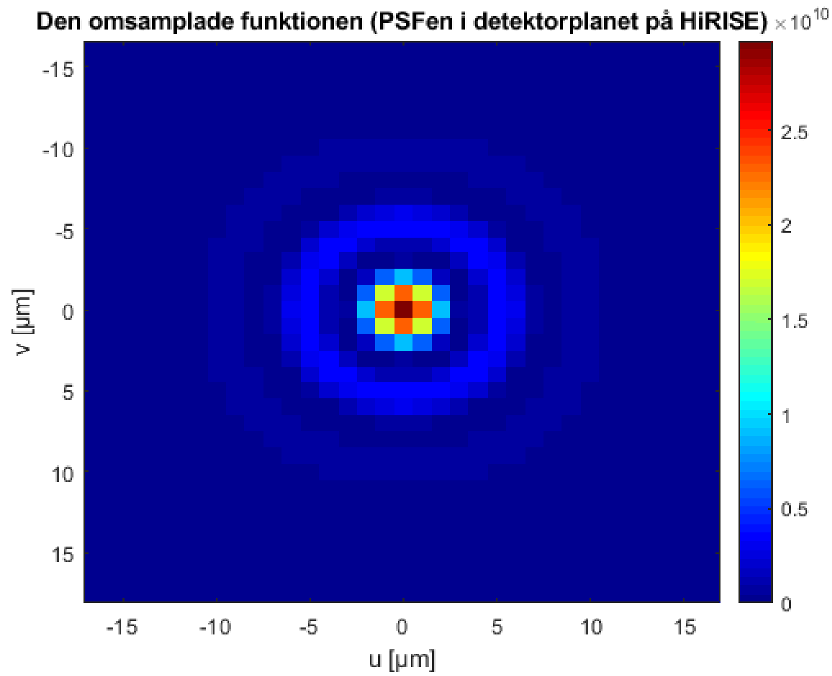

Figure 5: In-zoomad PSF efter omsampling till 2D.

På den perfekta bilden från Mars-ytan ser Rovern samt fot- och hjulspåren ut så här:

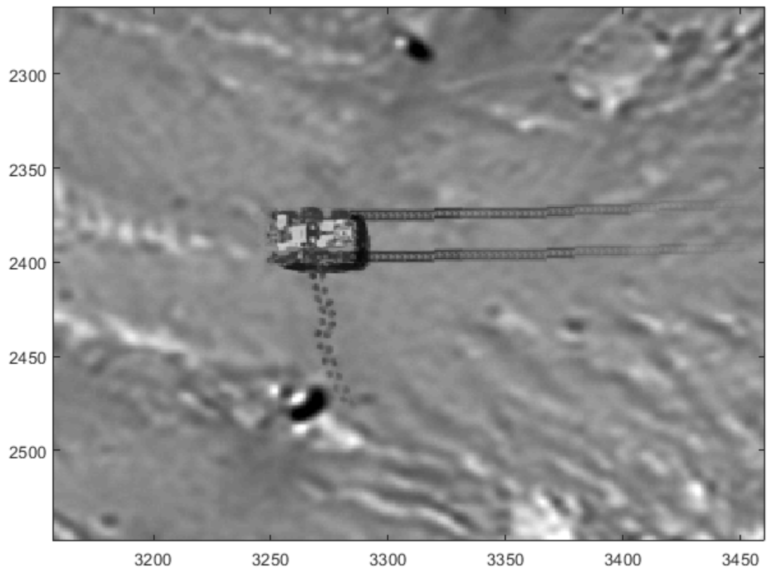

Figure 6: Perfekt bild på Rovern.

Efter faltningen blir den lite mer suddig, men allt är fortfarande urskiljbart:

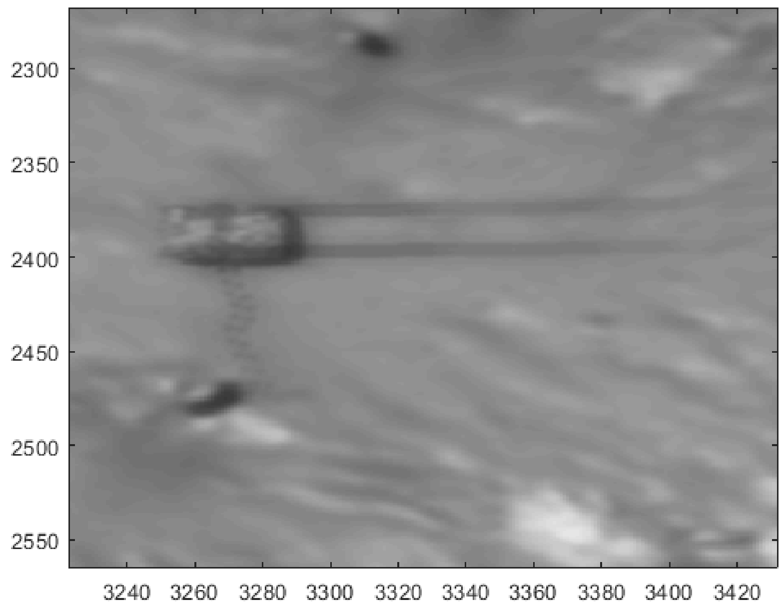

Figure 7: Faltning av PSF och perfekt bild.

Alltså är *The Martian* sann! Det hade jag aldrig trott.

A MATLAB

```
%
% Uppgift 1: Ersätt uttrycket
    to_be_inserted_by_user_uppg_1 (på 2 ställen) med rätt
    uttryck!
%

%
% Uppgift 5: Ersätt uttrycket
    to_be_inserted_by_user_uppg_5 (på 1 ställe) med rätt
    uttryck!
%


close all
clear

D_lins=.5; % Linsdiameter
L=3; % Avstånd mellan Plan 1 och Plan 2
f_lins=L; % Fokallängden på linsen
lambda=600e-9; % Våglängd (Mars är rödaktig så vi antar
    detta värde som en typisk våglängd)
k=2*pi/lambda;

%**** parameter för uppgift 2
c_4=k/(8*f_lins^3); % för korrektion av linsens
    fasmodulering, se uppgift 2 (i uppgift 1 ska c_4=0)
%**** slut parameter för uppgift 2

%**** parametrar för uppgift 5
sampla_om_till_2D=1; % Sätts =1 när du vill sampla om ber
    äknade PSFen till en tvådimensionell funktion (i
    uppgifterna 1-4 ska den vara noll)
if sampla_om_till_2D==1
    N_2D=5000; % matrisstorleken hos omsamplade PSFen:
        ska vara samma som matrisstorleken hos den
        perfekta bilden som du ska falta med
    samplavst_2D=0.96e-6; % samplingsavståndet hos
        omsamplade PSFen: ska vara lika med samplingsavstå
        ndet hos den perfekta bilden i detektorplanet
end
%**** slut parametrar för uppgift 5

% Plan 1 (efter lins)
```

```
N=1024; % matrisstorlek
a=D_lins/N; % samplingsavstånd
xvekt=-N/2*a:a:(N/2-1)*a;
yvekt=xvekt;
[xmat,ymat]=meshgrid(xvekt,yvekt);
rmat=sqrt(xmat.^2+ymat.^2); % radialavstånd i Plan 1

% Linsens fasmodulering
fi_lins=-k*rmat.^2/(2*f_lins)+c_4*rmat.^4; % linsens
    fasmoduleringsfunktion (andra termen är en icke-
    paraxiell korrektion, se uppg 2)
T_lins=exp(1i*fi_lins).*(rmat<=D_lins/2); % linsens
    transmissionsfunktion
T_lins = T_lins.*(rmat>=0.15);

% Välj punkter längs u-axeln i Plan 2 där PSFen ska
    evalueras
steg_i_u_led=0.25e-6;
u_max=30e-6;
u_eval_vekt=0:steg_i_u_led:u_max; % innehåller u-
    positioner för punkterna där PSFen ska beräknas

% Utför fältberäkningen i varje punkt i u_eval_vekt
disp('Beräknar fältet i alla u-positioner')
pause(0.1)
for berpunkt_nummer=1:length(u_eval_vekt) % gå igenom ber
    äkningspunkterna längs u-axeln
     if rem(berpunkt_nummer,10)==0
          disp(['Beräkningspunkt nummer ' num2str(
              berpunkt_nummer) ' av ' num2str(length(
              u_eval_vekt)) ' längs u-axeln'])
     end
     u=u_eval_vekt(berpunkt_nummer); % u-koordinat för
         aktuell beräkningspunkt
     r = sqrt(L^2 + (xmat-u).^2 + ymat.^2); % matris med
         avstånd från alla sampelpunkter i Plan 1 till
         aktuell beräkningspunkt
     E_PSF_i_aktuellt_u=sum(sum(T_lins.*exp(1i.*k.*r)./r))
         ; % fältet i beräkningspunkten, beräknat med HFM-
         integralen
     E_PSF_radiell_ickeparaxiell(berpunkt_nummer)=
         E_PSF_i_aktuellt_u; % spara resultatet i en vektor
end

E_PSF_radiell_paraxiell_teori_paraxiell_linsfasmod=
    besselj(1,k*D_lins/(2*L)*(u_eval_vekt+steg_i_u_led*1e
```

```
    -5))./(u_eval_vekt+steg_i_u_led*1e-5); % Detta är Airy
    -funktionen, som kan beskrivas analytiskt med
    besselfunktioner.
% Detta är resultatet man får om man har lins med
    paraxiella värdet på fasmoduleringen,
% och dessutom antar att paraxiella approximationen är
    giltig (som t.ex i HUPP1).
figure(201)
plot(u_eval_vekt*1e6,abs(E_PSF_radiell_ickeparaxiell)/abs
    (E_PSF_radiell_ickeparaxiell(1)))
hold on
plot(u_eval_vekt*1e6,abs(
    E_PSF_radiell_paraxiell_teori_paraxiell_linsfasmod)/
    abs(E_PSF_radiell_paraxiell_teori_paraxiell_linsfasmod
    (1)),'r')
legend('Korrekt (icke-approximativt) resultat med angiven
     faskorrektion (värdet på c_4) hos lins','Enligt 
    paraxiella approximationen, med paraxiella uttrycket f
    ör fasmodulering hos lins')
xlabel('u (v=0) [um]')
title(['|E_{PSF}|   (c_4= ' num2str(c_4) ')' ]);

PSF_radiell_ickeparaxiell=abs(E_PSF_radiell_ickeparaxiell
    ).^2; % PSFen är intensitetsfördelningen

% Gör om den beräknade radiella PSFen till en 2D-funktion
% samplad i en N_2D x N_2D-matris med samplingsavståndet
    samplavst_2D
if sampla_om_till_2D==1
    [PSF_2D, uvekt, vvekt, umat, vmat]=
        Sampla_om_radiell_till_2D(
        PSF_radiell_ickeparaxiell, u_eval_vekt, N_2D,
        samplavst_2D);
    figure(221)
    imagesc(uvekt*1e6,vvekt*1e6,PSF_2D)
    title('Den omsamplade funktionen (PSFen i 
        detektorplanet på HiRISE)')
    xlabel('u [um]')
    ylabel('v [um]')
    colormap(jet)
    colorbar
end
%%
I_perf = double(imread('
    highly_resolved_photo_of_Marks_surroundings.jpg'));
figure
```

```
imagesc(I_perf)
colormap(gray)
%%
I_faltat = ifft2c(fft2c(I_perf).*fft2c(PSF_2D));
imagesc(I_faltat)
colormap(gray)
```