ETHZ, BSc RW/CSE
Beispiellösung Prüfung
Numerische Mathematik für RW WS 06/07
Prof. R. Hiptmair

1 **1.** a)

$$
\begin{aligned}
2\sin^2\left(\frac{x}{2}\right) &= 2\left[\frac{1}{2i}\left(e^{\frac{ix}{2}} - e^{\frac{-ix}{2}}\right)\right]^2 = 2\left[-\frac{1}{4}\left(e^{\frac{2ix}{2}} - 2e^0 + e^{\frac{-2ix}{2}}\right)\right] \\
&= 1 - \frac{1}{2}\left(e^{ix} + e^{-ix}\right) = 1 - \cos(x)
\end{aligned}
$$

1

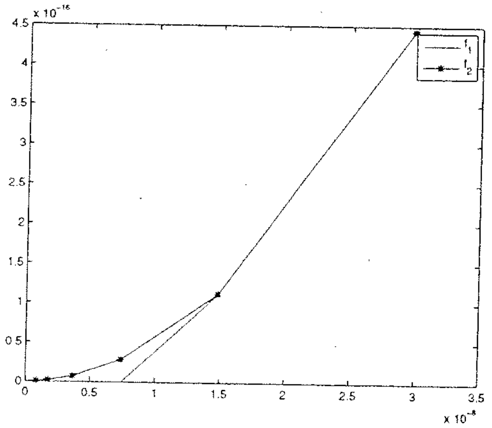
1 b)

```
function Aufgabe1()
x=2.^[-25:-1:-30];
plot(x,f1(x),x,f2(x),'r*-');
legend('f_1','f_2');

function y=f1(x)
 y=1-cos(x)

function y=f2(x)
 y=2*sin(x/2).^2;
```

Kurven korrekt, unterscheidbar, Legende aussagekräftig 1



2 c) Für $x \approx 0$ ist $\cos(x) \approx 1$: Es tritt Auslöschung auf. 1 `f2` liefert den genaueren Wert. 1

4 **2.** Die absolute Konditionszahl von I ist die kleinste obere Schranke des Quotienten:

$$\frac{|I(g)|}{||g||_\infty}.$$

1

Es gilt

$$|I(g)| = |\int_a^b g(x)\,dx| \leq \int_a^b |g(x)|dx \leq ||g||_\infty |b-a|.$$

1 1

Die Schranke ist optimal, da Gleichheit gilt, falls g konstant ist. Die absolute Konditionszahl ist also $|b-a|$.

1

4 **3.** a)

```
function[U,S,V]=prodsvd(A,B)
[QA,RA]=qr(A,0);
[QB,RB]=qr(B,0);
[U,S,V] = svd(RA*RB');
U=QA*U;
V=QB*V;
```

mit 0, beide

Der Aufwand für die QR-Zerlegungen (k Householdertransformationen der Länge n-1, n-2,...,n-k+1) ist je $nk^2 - \frac{1}{3}k^3 + O(nk) = O(n)$ Operationen. Die Singulärwertzerlegung der kxk Matrix braucht $O(n^0) = O(1)$ Operationen und die Multiplikationen benötigen $O(nk^2) = O(n)$ Operationen. Der Aufwand ist also von der Grössenordnung O(n). (Würde ich nicht so explizit verlangen: $O(n)$ ist ausreichend.)

5 b) $'' \Rightarrow''$: $Rang(X) \leq q \Rightarrow X = U\Sigma V^T$, mit $\sigma_i = 0,\ i > q$.
$\Rightarrow U\Sigma V^T = U(:,1:q)\Sigma(1:q,1:q)V(:,1:q)'$.
Mit $A = U(:,1:q)\Sigma(1:q,1:q), B = V(:,1:q)$ folgt die Behauptung.

$'' \Leftarrow''$: $A, B \in \mathbb{R}^{n,q} \Rightarrow$ Singulärwertzerlegungen $A = U_A\Sigma_A V_A'$, $B = U_B\Sigma_B V_B'$, mit $U_A,\ U_B \in \mathbb{R}^{n,q},\ \Sigma_A,\ \Sigma_B\ V_A,\ V_B \in \mathbb{R}^{q,q}$.
Mit Singulärwertzerlegung von $\Sigma_A\ V_A'\ V_B\Sigma_B = \tilde{U}\tilde{\Sigma}\tilde{V}'$, alle Matrizen $\in R^{q,q}$
$\Rightarrow AB^T = U_A\tilde{U}\tilde{\Sigma}\tilde{V}'U_B'$.
Seien nun $U = [U_A\tilde{U},\ U_2] \in R^{n,n}$, $V = [U_B\tilde{V},\ V_2] \in R^{n,n}$ mit $U_2,\ V_2$ orthogonale Erweiterungen und $\Sigma \in \mathbb{R}^{n,n}$ die mit Nullen erweiterte Matrix $\tilde{\Sigma}$.
$\Rightarrow X = U\Sigma V'$ mit $rang(\Sigma) = q \Rightarrow rang(X) = q$

3 c) Nach Theorem 3.3.21 kann die Rang q Bestapproximation einer Matrix in der Frobeniusnorm erhalten werden, indem die Singulärwertzerlegung der Matrix berechnet wird und aus $\boldsymbol{U}$ und $\boldsymbol{V}$ die ersten q Spalten und aus $\boldsymbol{\Sigma}$ die obere linke qxq Matrix extrahiert wird.
Um die Singulärwert der Matrix $\boldsymbol{X}+\boldsymbol{Y}$ effizient zu berechnen, werden die Rang q Matrixen in $\mathbb{R}^{n,q}$ Matrizen zerlegt $\boldsymbol{X} = \boldsymbol{A}_x\boldsymbol{B}_x^T$, $\boldsymbol{Y} = \boldsymbol{A}_y\boldsymbol{B}_y^T$ und es gilt $\boldsymbol{X}+\boldsymbol{Y} = [\boldsymbol{A}_x\boldsymbol{A}_y][\boldsymbol{B}_x\boldsymbol{B}_y]'$. Darauf kann a) angewendet werden, was die Singulärwertzerlegung von $\boldsymbol{X}+\boldsymbol{Y}$ liefert.

```
function[Az,Bz]=rank_q_approx(Ax,Ay,Bx,By)
 A=[Ax,Ay];
 B=[Bx,By];
 [U,S,V]=svd_ab(A,B);
 q=size(Ax,2);
 Az=U(:,1:q)*S(1:q,1:q);
 Bz=V(:,1:q);
```

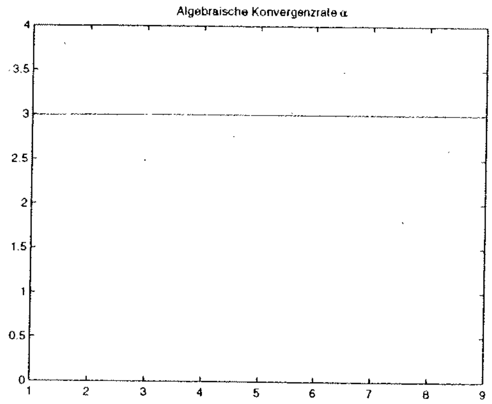
6 **4.** a)

```
function[conv_rate]=Aufgabe4a()
F=@(x) sqrt(1-x.^2).*exp(x);
%Q_ex=quad(F,-2,2,1e-15);
Q_ex=1.775499689212181
tol=1e-4;
n_vec=10:10:1000;
for i=1:length(n_vec);
 n=n_vec(i);
 [x,w]=Gauss_Quad(n);
 Q(i)=sum(F(x).*w)
```

```
  if(i>1)
   alpha(i)=log(abs(Q(i)-Q_ex)/abs(Q(i-1)-Q_ex))/...
           log(n_vec(i)/n_vec(i-1));

   if(abs(alpha(i)-alpha(i-1))<tol)
    conv_rate=-alpha(i);
    return
   end
  end
 end
 disp('no convergence')
 conv_rate=0;
```

2



$\alpha \approx 3$.

4 b)

$$\int_{-1}^{1} \sqrt{1-x^2}e^x\,dx = \int_{0}^{1} \sqrt{1-x^2}e^x\,dx + \int_{-1}^{0} \sqrt{1-x^2}e^x\,dx \qquad 1$$

$$\int_{0}^{1} \sqrt{1-x}\sqrt{1+x}e^x dx = 2\int_{0}^{1} s^2\sqrt{2-s^2}e^{1-s^2}\,ds \qquad 1$$

mit $s=\sqrt{1-x}$, $x = 1-s^2$, $dx = -2sds$.

$$\int_{-1}^{0} \sqrt{1-x}\sqrt{1+x}e^x dx = 2\int_{0}^{1} s^2\sqrt{2-s^2}e^{s^2-1}\,ds \qquad 1$$

mit $s=\sqrt{1+x}$, $x = s^2-1$, $dx = 2sds$.
Also

$$\int_{-1}^{1} \sqrt{1-x^2}e^x\,dx = 2\int_{0}^{1} s^2\sqrt{2-s^2}(e^{s^2-1}+e^{1-s^2})\,ds \qquad 1$$

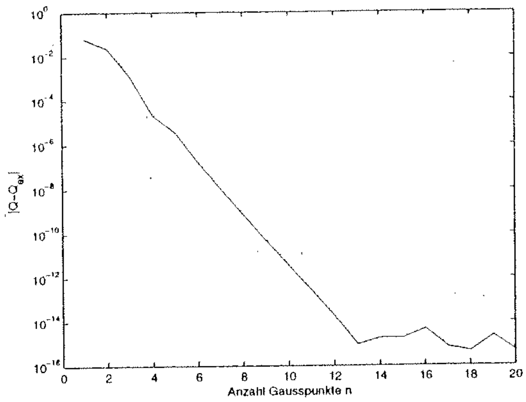
6 c)

```
function[]=Aufgabe4c()
F=@(s) 2*s.^2.*sqrt(2-s.^2).*(exp(1-s.^2)+exp(s.^2-1))   1

Q_ex=1.775499689212181;
n_vec=1:20;
for i=1:length(n_vec);
 n=n_vec(i);
[x,w]=Gauss_Quad(n);
```

```
Q(i)=sum(F(x/2+0.5).*w/2);            2
end
error=abs(Q-Q_ex)
figure(1)
semilogy(n_vec,error,'r')             1
xlabel('Anzahl Gausspunkte n')
ylabel('|Q-Q_{ex}|')                  1
```

(mit Skalierung, an richtigen Stellen ausgewertet)



Exponentielle Konvergenz 1

3 **5.** Aufstellen des Gleichungssystems fuer α und β:

1. Zeile: $x_1\alpha + x_2\beta = b_1$ 1

i. Zeile(i=2,...,n-1): $x_i\alpha + (x_{i-1} + x_{i+1})\beta = b_i$ 1

n. Zeile: $x_n\alpha + x_{n-1}\beta = b_n$ 1

4 a)

```
function[alpha,beta]= Least_Squares(x,b)
X=[x,[x(2:end);0]+[0;x(1:end-1)]];     2
a=X\b;   1
alpha=a(1);
beta=a(2);     1
```

ohne Schleife!

1 b)

```
function  Least_Squares_example
x = [0.1,0.75, 3,4.1,5,4.9]'
b =[0.3,1.54, 2.8,4.6,3.7,2]'
[alpha,beta]=Least_Squares(x,b)
```

$\alpha = -0.1837$, $\beta = 0.5920$ 1

6 + 6 **6.**

```
function draweggcurve()
options=odeset('reltol',1e-5,'abstol',1e-5);   1
[t,y]=ode45(@f,[0 6],[0 1.2],options);   1 1 1
plot(y(:,1),y(:,2))   1
xlabel('x')
ylabel('y')   1


function y=f(t,x)   1
```

```
gradF=4*(x(1)^2+x(2)^2)*x-3*x.^2;
y=[-gradF(2);gradF(1)]./norm(gradF);
```

3

1 1