Gleichmäßige Konvergenz, Existenz der Stammfunktion einer stetigen Funktion

Aufgaben: 4

```
> restart;
  with(plots):
```

Gleichmäßige Konvergenz

MATH: Wie wir früher Zahlenfolgen betrachtet haben, kann man auch Folgen von Funktionen mit gemeinsamem Definitionsbereich betrachten. Für jeden Punkt des Definitionsbereiches bekommt man dann eine Folge im alten Sinne. Sind alle diese Folgen konvergent, so spricht man von **punktweiser Konvergenz**.

BEISPIEL:

$$f_n : [0, 1] \to \mathbb{R} : x \mapsto x^n.$$

```
> limit((1/2)^n,n=infinity);
```

$$0 \qquad \textbf{(1.1.1)}$$

```
> limit((1)^n,n=infinity);
```

$$1 \qquad \textbf{(1.1.2)}$$

Man sieht, dass jedes f_n stetig ist, das die Folge

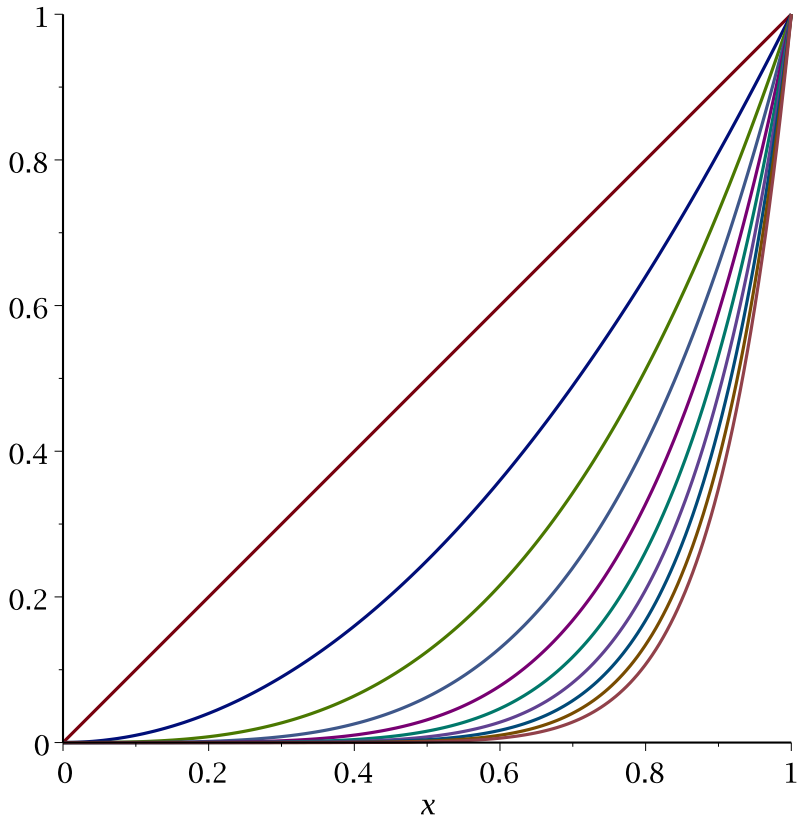
$$f_n(x)$$

für jedes feste $x \in [0, 1]$ konvergiert, also eine Grenzfunktion

$$f : [0, 1] \to \mathbb{R} : x \mapsto \lim_{n \to \infty} f_n(x)$$

existiert, aber dass diese nicht stetig ist.
Schauen wir uns dies am Plot an:

```
> d1:=plot(map(i->x^i,{$1..10}),x=0..1):
  display(d1,scaling=constrained);
```



ÜBUNG [01]:

1) Begründe (kurz!)

$$f_n : [0, 1] \to \mathbb{R} : x \mapsto x^n$$

definiert eine Funktionenfolge auf $[0, 1]$ von stetigen Funktionen.

2) Zeige: Die Folge ist punktweise konvergent gegen eine unstetige Funktion.

MATH: Da sich also der punktweise Konvergenzbegriff nicht gut mit der Stetigkeit verträgt, braucht man einen eingeschränkteren Konvergenzbegriff für Funktionenfolgen, damit man wenigstens die Stetigkeit der Grenzfunktion bekommt. Dieses ist die gleichmäßige Konvergenz:

Die Funktionenfolge

$$f_n : \mathrm{D} \to \mathbb{R}$$

konvergiert gleichmäßig gegen

$$f: \mathrm{D} \to \mathbb{R} \quad ,$$

falls zu jedem $\varepsilon > 0$ ein $n_0 \in \mathbb{N}$ existiert mit

$$|f(x) - f_n(x)| < \varepsilon$$

für alle $n > n_0$ und alle $x \in \mathrm{D}$.

MATH: Sind die f_n alle stetig, so auch *f*.

DENKANSTOSS: Formuliere das (notwendige und hinreichende) Cauchy-Kriterium und das (hinreichende) Majorantenkriterium für gleichmäßige Konvergenz.

BEISPIEL: Im obigen Beispiel liegt also keine gleichmäßige Konvergenz vor. Man kann jedoch ein beliebiges abgeschlossenes Teilintervall D von $[0, 1)$ betrachten.
Wie oben gesehen, hat man dort punktweise Konvergenz mit Hilfe von

$$n_0(x, \varepsilon).$$

Man findet jedoch für ein festes ε eine gemeinsame Majorante

$$n_0(\varepsilon) \geq n_0(x, \varepsilon)$$

für alle $x \in \mathrm{D}$. Mit Hilfe dieser Majorante sieht man, dass gleichmäßige Konvergenz vorliegt.

Der Weierstraßsche Approximationssatz und Bernsteinpolynome

MATH: Der Weierstraßsche Approximationssatz besagt, dass für jede stetige Funktion $f: [a, b] \to \mathbb{R}$ eine Folge $(p_n)_{n \in \mathbb{Z}_{\geq 1}}$ von Polynomfunktionen mit $Grad(p_n) \leq n$ existiert, welche gleichmäßig gegen *f* konvergiert.

MATH: Der Weierstraßsche Approximationssatz hat einen sehr expliziten Beweis mit Hilfe der Bernsteinpolynome:
Jede stetige Funktion

$$f: [0, 1] \to \mathbb{R}$$

ist Grenzfunktion der Folge der Bernsteinpolynome $B_{(n, f)}$ von *f*, welche gleichmäßig gegen *f* konvergiert, also $\lim_{n \to \infty} B_{(n, f)}(x) = f$.
Die Bernsteinpolynome sind dabei gegeben als:

$$B_{(n, f)}(x) := \sum_{k=0}^{n} f\left(\frac{k}{n}\right) \cdot \binom{n}{k} \cdot x^k \cdot (1 - x)^{n-k}$$

ÜBUNG [02]:

1) Schreibe eine Maple-Funktion, die für eine stetige Funktion *f* auf dem

Intervall $[-1, 1]$ das approximierende Bernsteinpolynom der Ordnung n bestimmt.
(Hinweis: Benutze z. B. die Transformation

$$x \mapsto 2x - 1,$$

um von dem Intervall $[0, 1]$ auf das Intervall $[-1, 1]$ umzurechnen.)

2) Plotte den Fall

$$f(x) := |x|$$

für einige n.

Potenzreihen als Funktionenfolgen

MATH: Eine sehr wichtige Klasse von gleichmäßig konvergenten Funktionenfolgen werden durch die Potenzreihen geliefert:
Auf jedem Kompaktum, welches im (offenen) Konvergenzkreis liegt, konvergiert die Folge der Teilsummen gleichmäßig gegen eine Grenzfunktion, die somit stetig ist.
Dabei liegt es nahe, von vornherein den Definitionsbereich als Teilmenge von C statt = zu betrachten.

Man kann sich schnell Beispiele aus den Folgen von Taylorpolynomen konstruieren:

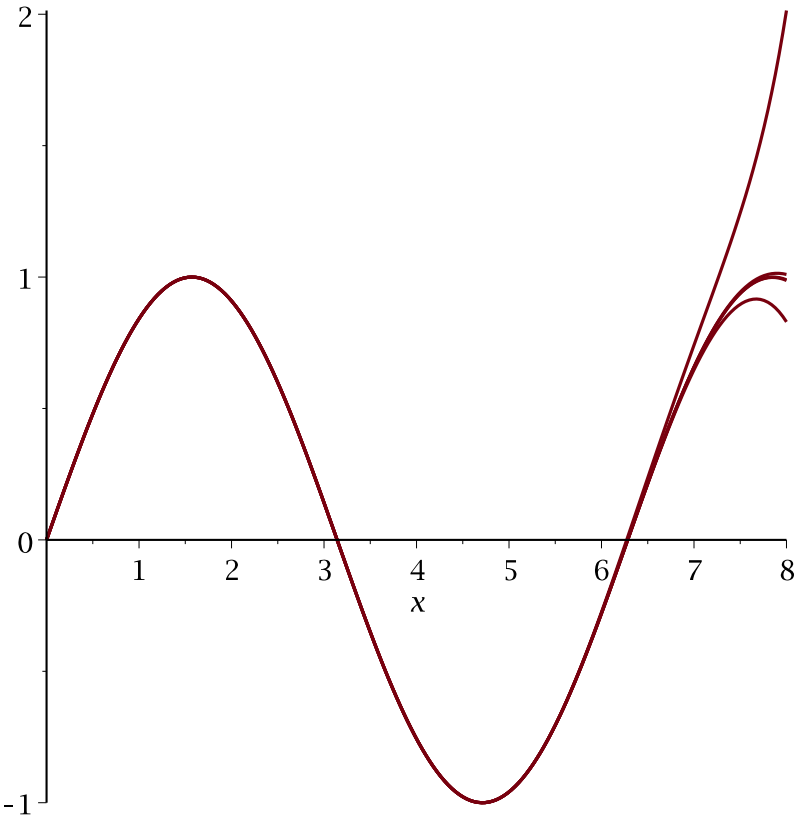
```
> s:=k->add((-1)^(i)*(x)^(2*i+1)/(2*i+1)!,i=0..k);
```

$$s := k \to add\left(\frac{(-1)^i x^{2i+1}}{(2i+1)!}, i = 0..k\right) \quad \textbf{(1.3.1)}$$

```
> s(5);
```

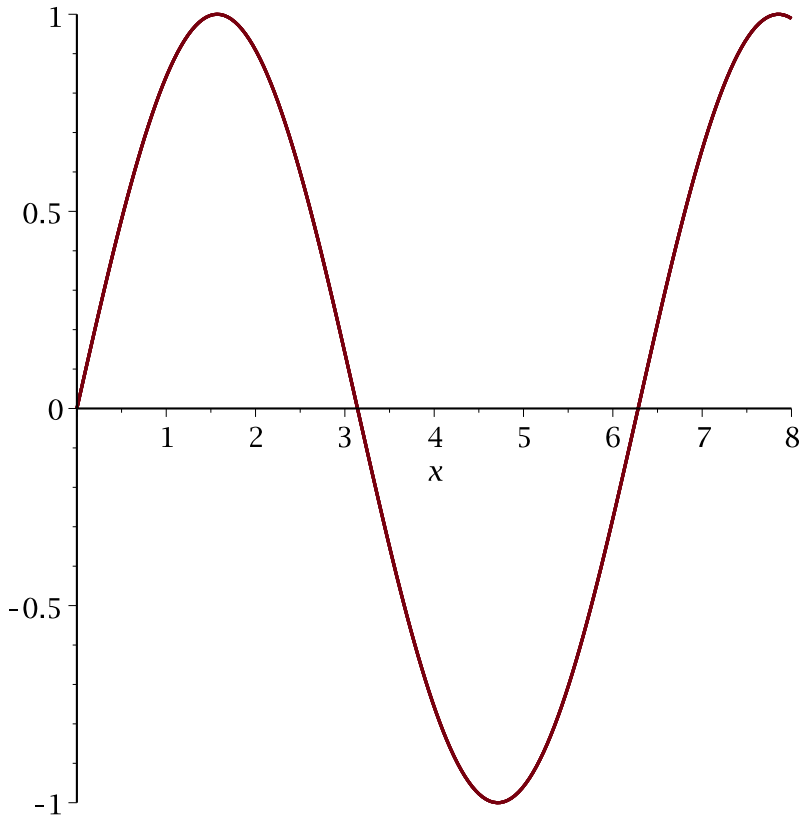
$$x - \frac{1}{6}x^3 + \frac{1}{120}x^5 - \frac{1}{5040}x^7 + \frac{1}{362880}x^9 - \frac{1}{39916800}x^{11} \quad \textbf{(1.3.2)}$$

```
> display(map(k->(plot(s(k),x=0..8)),[$8..12]));
```



DENKANSTOSS: Man experimentiere mit den Intervallgrenzen und dem *k*, um ein Gefühl für die Konvergenz zu bekommen.

```
> display(map(k->(plot(subs(x=x-Pi,-s(k)),x=0..8)),[$8..12]));
```



Es sieht also so aus, als ob unsere Funktionenfolge gegen eine Funktion *f* konvergiert mit

$$f(x-\pi) = -f(x)$$

für alle *x*. Insbesondere wäre *f* dann 2π-periodisch. Dies ist natürlich kein Beweis, es wird lediglich durch die Bilder nahegelegt.

DENKANSTOSS: Durch welche Formel wird der Konvergenzradius einer Potenzreihe festgelegt?

MATH: Wir hatten bereits früher in dem Reihenworksheet von

$$\exp, \sin, \cos$$

und den Zusammenhängen zwischen den dreien gesprochen. Der Konvergenzradius ist in allen drei Fällen unendlich. Damit folgt, dass alle drei Funktionen stetig sind, denn wir haben gleichmäßige Konvergenz auf jedem Kompaktum. Wir wissen auch schon, dass

$$\exp$$

$2i\pi$-periodisch ist und damit

$$\sin \text{ und } \cos$$

2π-periodisch sind: 2π ist die kleinste positive Zahl mit

$$\exp(i\cdot 2\cdot\pi) = 1,$$

d. h. mit

$$\sin(2\cdot\pi) = 0 \text{ und } \cos(2\cdot\pi) = 1.$$

Die Periodizität für

exp

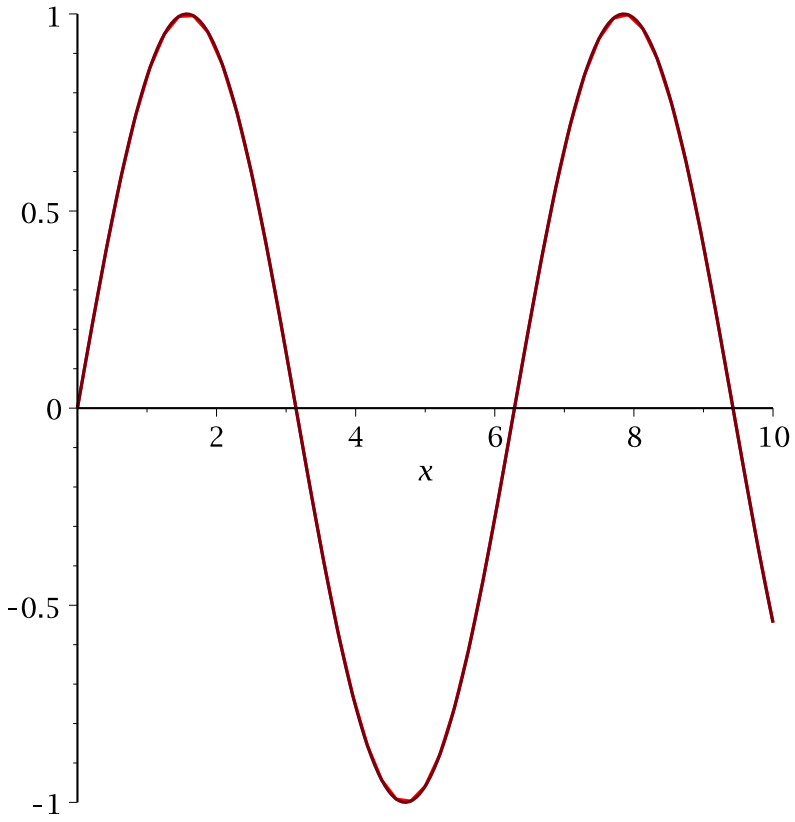
folgt dann aus der fundamentalen Eigenschaft, das exp Summen in Produkte überführt:

$$\exp : (\mathbb{C}, +) \to (\mathbb{C}^*, \cdot)$$

ist Homomorphismus.

Wir wollen betrachten, was beim Verschieben von Potenzreihen passiert.

```
> B:=plot(sin(x),x=0..10):
  display({B,animate(sin(x-a),x=0..10,a=0..Pi)});
```



Beachte: Der Funktionsgraph von sin(x) schiebt sich bei sin($x-a$) um a nach rechts!

Die Additionstheoreme drücken den verschobenen sin als Linearkombination von sin und cos aus.

```
> expand(sin(y+1));
```

$$\sin(y)\cos(1) + \cos(y)\sin(1) \qquad \textbf{(1.3.3)}$$

```
> subs(y=x-1,%);
```

$$\sin(-1+x)\cos(1)+\cos(-1+x)\sin(1) \qquad \textbf{(1.3.4)}$$

ÜBUNG [03]:

1) Benutze die gerade vorgeführte Rechnung, um
$\sin(x)$
als eine Potenzreihe in $x-1$ darzustellen. (Hinweis: Verschaffe Dir zuerst die Partialsummen der Cosinusreihe analog zu s oben.)
2) Begründe, warum die *k*-te Partialsumme nicht mit
subs(x=x-1,expand(subs(x=x+1,s(k))))
identisch ist.
3)
Sei eine Funktion
$f:(-r,r)\to\mathbb{R}$
dargestellt durch eine Potenzreihe
$$f(x)=\sum_{n=0}^{\infty}a_n x^n$$
in *x* mit Konvergenzradius *r*.
a) Zeige, dass eine Stammfunktion von *f* ebenso als Potenzreihe in *x* dargestellt werden kann.
b) Wie sieht diese Stammfunktion aus?
c) Welchen Konvergenzradius hat diese Stammfunktion?

Existenz der Stammfunktion einer stetigen Funktion

Mit Hilfe der hier vorgestellen Methoden kann man einen Beweis angeben, dass stetige Funktionen integrierbar sind. Dieser Beweis zeigt eine typische Anwendungen der gleichmäßigen Konvergenz an einer einfachen Aussage.

MATH: Die Existenz von Stammfunktionen stetiger Funktionen auf einem Intervall folgt nun sehr leicht aus
1.) der gleichmäßigen Approximierbarkeit stetiger Funktionen auf abgeschlossenen Intervallen durch Polynomfunktionen und
2.) dem folgenden Satz.

MATH: Sei $(f_n)_{n\geq 1}$ eine Folge stetig differenzierbarer Funktionen auf einem beschränktem Intervall I. Die Funktionenfolge $(f'_n)_{n\geq 1}$ konvergiere auf I gleichmäßig gegen eine Funktion $g:I\to\mathbb{R}$. Wenn ein $c\in I$ existiert so, dass $(f_n(c))_{n\geq 1}$ konvergiert, so konvergiert $(f_n)_{n\geq 1}$ gleichmäßig gegen eine differenzierbare Funktion $f:I\to\mathbb{R}$ mit $f'=g$.

ÜBUNG [04]:

Begründe ausführlich mit der hier vorgestellen Beweisidee, warum jede stetige Funktion eine Stammfunktion hat.

Anwendung: RSA-Verfahren in der Kryptografie

Aufgaben: 4

```
> restart;
```

Inhalt

Den chinesischen Restesatz kann man anwenden, um die Korrektheit des RSA-Verfahrens zu beweisen, weshalb wir nun einen kleinen Ausflug in die Kryptographie unternehmen.

Wir brauchen folgenden Satz, welchen wir hier vorerst ohne Beweis akzeptieren wollen:

MATH: Ist M eine endliche abelsche Gruppe, so gilt :

$$|M| \cdot M = \{0\}.$$

Hiermit folgt sofort der berühmte kleine Satz von *Fermat:*

MATH: Ist $a \in \mathbb{Z}$ und $p \in \mathbb{Z}$ prim, so gilt:

$$a^p \equiv a \quad (\mathbf{mod}\, p).$$

ÜBUNG [05]:

Führe diese Folgerung aus. (*Hinweis*: Betrachte die Restklasse von a in $\mathbb{Z}/p\mathbb{Z}$).

Nachdem wir nun die mathematische Vorarbeit geleistet haben kommen wir zum RSA-Verfahren, welches nach seinen Schöpfern Ron Rivest, Adi Shamir und Leonard Adleman benannt wurde.
Das RSA-Verfahren ist ein asymmetrisches Kryptoverfahren (auch Public-Key-Verfahren genannt), das heißt man braucht zwei verschiedene Schlüssel. Einer davon ist privat und nur einem selber bekannt, wohingegen der andere veröffentlicht wird.
Es wird in vielen Bereichen eingesetzt, zum Beispiel zur Signatur von Mails oder deren Verschlüsselung. Ferner wird es im SSL-Protokoll verwendet, also ist jeder der schon mal eine Webseite mit https besucht hat, kam schon mal mit diesem Verfahren in Berührung.

ALGORITHMUS: RSA-Schlüsselerzeugung:
1) Wähle zufällig zwei Primzahlen p und q.
2) Berechne $n = p \cdot q$.

3) Berechne $\varphi(n) = (p-1)\cdot(q-1)$.
4) Wähle $e \in \mathbb{Z}$ mit $1 < e < \varphi(n)$ und $ggT(e, \varphi(n)) = 1$.
5) Berechne $d \equiv e^{-1} \pmod{\varphi(n)}$.

Das Paar $((n, e), (n, d))$ ist das gesuchte Schlüsselpaar. Wir wollen also nun ein solches Paar bestimmen, dafür fragen wir Maple nach Primzahlen:

```
> p:=ithprime(12345);
  q:=ithprime(23456);
```

$$p := 132241$$
$$q := 267649 \qquad \textbf{(2.1.1)}$$

MAPLE: **ithprime(x)** liefert die x-te Primzahl.

```
> n:=p*q;
```

$$n := 35394171409 \qquad \textbf{(2.1.2)}$$

```
> phi_n:=(p-1)*(q-1);
```

$$phi\_n := 35393771520 \qquad \textbf{(2.1.3)}$$

```
> e:=rand(2..phi_n-1)():
  while(igcd(e,phi_n)<>1) do
    e:=rand(2..phi_n-1)():
  end do;
  e;
```

$$e := 15235196455$$
$$e := 17988963612$$
$$e := 34344539878$$
$$e := 7412421907$$
$$7412421907 \qquad \textbf{(2.1.4)}$$

MAPLE: **rand(2..phi_n-1)()** erzeugt eine zufällige natürliche Zahl aus dem Intervall $[2, \varphi(n)]$.

```
> igcdex(e,phi_n,'d','t'):
  d;
```

$$631864603 \qquad \textbf{(2.1.5)}$$

Also erhalten wir das folgende Schlüsselpaar:

```
> public_key:=[n,e];
  private_key:=[n,d];
```

$$public\_key := [35394171409, 7412421907]$$
$$private\_key := [35394171409, 631864603] \qquad \textbf{(2.1.6)}$$

ALGORITHMUS: RSA-Verschlüsselung
Input: Public-Key (n, e) , Nachricht m
Output: verschlüsselte Nachricht c
Algorithmus:

$$c := m^e \pmod{n}$$

Wir können dafür natürlich auch eine Maplemethode schreiben:

```
> encrypt:=proc(key::list(posint),m::posint)
```

```
    if m < 0 or m > key[1] then
     error("Nachricht m verletzt 0 <= m < n");
    fi;
    if nops(key) <> 2 then
     error("Kein gültiger Schlüssel.");
    fi;
    if key[2]=1 then
     return m;
    end if;
    if irem(key[2],2)=0 then
     return encrypt([key[1],key[2]/2],m)^2 mod key[1];
    end if;
    return encrypt([key[1],key[2]-1],m)*m mod key[1];
  end proc:
```

ALGORITHMUS: RSA-Verschlüsselung
Input: Private-Key (n, d) , Kryptotext c
Output: Nachricht m
Algorithmus:

$$m := c^d \pmod{n}$$

Auch hier eine Funktion in Maple:

```
> decrypt:=proc(key::list(posint),c::posint)
    if c < 0 or c > key[1] then
     error("Kryptotext c verletzt 0 <= c < n");
    fi;
    if nops(key) <> 2 then
     error("Kein gültiger Schlüssel.");
    fi;
    if key[2]=1 then
     return c;
    end if;
    if irem(key[2],2)=0 then
     return encrypt([key[1],key[2]/2],c)^2 mod key[1];
    end if;
    return encrypt([key[1],key[2]-1],c)*c mod key[1];
  end proc:
```

Wir wollen nun eine Nachricht versenden:

```
> m:=1234567890;
```

$$m := 1234567890 \qquad \textbf{(2.1.7)}$$

```
> c:=encrypt(public_key,m);
```

$$c := 3067791170 \qquad \textbf{(2.1.8)}$$

Nun können wir diese wieder entschlüsseln und testen, ob es sich um die ursprüngliche Nachricht handelt:

```
> m=decrypt(private_key,c);
  is(%);
```

$$1234567890 = 1234567890$$

(2.1.9)

true **(2.1.9)**

ÜBUNG [06]:

Beweise die Korrektheit des RSA-Verfahrens. (*Hinweis:* Wende den chinesischen Restsatz auf die Situation an.)

Bleibt die Frage: Wie sicher ist das RSA-Verfahren?
Es ist bisher unbekannt wie schwer es ist eine verschlüsselte Nachricht unter Kenntnis des öffentlichen Schlüsseln zu entschlüsseln. Auf jeden Fall wäre es möglich die Zahl n zu faktorisieren, um dann so den privaten Schlüssel zu berechnen. Zum Glück ist die Faktorisierung ganzer Zahlen (bisher) kein einfaches Problem.

ÜBUNG [07]:

Entschlüssele folgenden Kryptotext

```
> c_2:=462417249216465556509;
```

$$c\_2 := 462417249216465556509 \quad \textbf{(2.1.10)}$$

Hierbei wurde der folgende öffentliche Schlüssel verwendet:

```
> public_key_2:=[105080930338397326003 7,17743887669363763081]
  ;
```

$$public\_key\_2 := [10508093033839732600 37, \quad 17743887669363763081] \quad \textbf{(2.1.11)}$$

Wer nun denkt, dass Entschlüsselung kein Problem sei, der möge sich an folgenden Eingaben versuchen (ABER NUR AN SEINEM EIGENEM RECHNER UND NUR AUF EIGENE GEFAHR!!!):

```
> public_key_3:=
  [74037563479561712828046796097429573142593188889231289084936 2
  32638972765034028266276891996419625117843995894330502127585 37
  01189680982867331732731089309005525051168770632990723963807 86
  71008609696253793465056379635 9,
  65665637630314161885353737197235555906318422322834701562370 9
  40325528212072538528376170113254080984942049287794660850301 81
  39707542366171790845942577383995988169935682541974184530735 18
  85532965717410820210278096419];
```

$$public\_key\_3 := \quad \textbf{(2.1.12)}$$

```
[
74037563479561712828046796097429573142593188889231289 0\
84936232638972765034028266276891996419625117843995894 3\
30502127585370118968098286733173273108930900552505116 8\
```

```
    770632990723963807867100860969625379346505637963 59,
    656656376303141618853537371972355559063184222322834701\
    562370940325528212072538528376170113254080984942049287\
    794660850301813970754236617179084594257738399598816993\
    568254197418453073518855329657174108202102780964 19]
```

```
> c_3 :=
  174558793054456089214780274632018946387229751252799540\
  6047632815262704301168583788814704722341835638038724581819225730396470\
  1587772755443626423797800640313476063999992844398455506103667178299764164631521905971470;
```

$$c\_3 := 174558793054456089214780274632018946387229751252799540\backslash 6047632815262704301168583788814704722341835638038724581819225730396470\backslash 1587772755443626423797800640313476063999992844398455506103667178299764164631521905971470 \quad \textbf{(2.1.13)}$$

Es sei hier noch angemerkt, dass die in der Realität verwendeten Schlüsselpaare noch um einiges größer sind. Der obige Schlüssel hat eine Bitlänge von 704, wohingegen man heutzutage problemlos mit Schlüsseln von 2048 oder sogar 4096 Bits arbeitet.

Wer will kann gerne mit dem obigen Schlüssel einige Nachrichten verschlüsseln:

```
> encrypt(public_key_3,1234567890);
```

```
30534241218373173488318091041161561612660465817648966150 2\ (2.1.14)
    688895190534221574682251225724572759274633514718867634\
    802581214563445535709541567198589515337324161905185149\
    982280102020183717324212571271372672723351683 68
```

Man sieht, dass das Rechnen sehr schnell geht, allerdings ist das Knacken ziemlich schwierig.

Ein weiterer Ansatz RSA zu brechen wäre die Berechnung von $\varphi(n)$. Allerdings ist dieser nicht besser als der vorherige wie die nächste Aufgabe zeigt.

ÜBUNG [08]:

Zeige: Die Berechnung von $\varphi(n)$ ist äquivalent zur Bestimmung von p und q.
Hinweis: Bestimme zuerst $p+q$.