Simulation von Differentialgleichungen

Hier geht es um die Echtzeit-Simulation von Differentialgleichungen als eine Folge von infinitesimalen Bewegungen

Früher hat man gerne zur Lösung von komplexeren „gewöhnlichen“ Differentialgleichungen sogenannte Analogrechner eingesetzt – beispielsweise, um die Dämpfung eines PKW damit zu überprüfen.
Allerdings musste die Differentialgleichung dazu aufgestellt und „programmiert“ werden.

Im Zeitalter der digitalen Simulation gibt es andere Möglichkeiten.
Allerdings ist die hier vorgestellte Methode wohl neu.

Es werden zur Simulation gar keine Schwingungsgleichungen benutzt!

Vielmehr wird von grundlegenden physikalischen Gleichungen ausgegangen, die in minimalen zeitlichen Intervallen immer neu angewendet, und für das System (rekursiv) jeweils ein neuer Anfangszustand berechnet wird.

Durchgeführt wurde dies für Feder- und Fadenpendel.

Als Ergebnis ist die Simulation geeignet, um diese Pendel in Echtzeit nachzubilden, ohne eine Winkelfunktion (sin, cos) zu benutzen und ohne die Schwingungsgleichungen zu kennen.

Das Verfahren sollte daher auch geeignet sein, für nichtlineare Dämpfungen gute Simulationsergebnisse zu bekommen.

Das Verfahren könnte genauso (bei genügend Rechenkapazität) benutzt werden, um die Bewegung von Sternen in einer Galaxie zu berechnen oder die Flugbahn einer Kanonenkugel bei Seitenwind.

Das Verfahren sollte daher auch geeignet sein, für nichtlineare Dämpfungen gute Simulationsergebnisse zu bekommen.

Anbei die Erklärung aus der Korrespondenz mit einem Freund.

Hallo Sven,

immer schon wollte ich ausprobieren, ob eine Simulation eines (Feder-), Faden-) Pendels auch ohne die in der Literatur üblichen „hochgestochenen“ Ansätze mit Winkelfunktionen, Differentialgleichungen usw. möglich ist.
Der Ansatz ist recht simpel und gleichzeitig eine gute Einführung in die Differentialrechnung:

Anfangsbedingung:
Masse an Feder im Federkonstanten D,
Masse wird um x ausgelenkt.

Ansatz:

Auf die Masse wirkt durch die Feder eine Kraft F auf die Masse M.
Diese beschleunigt (a) die Masse Richtung Ruhelage:

===> Wiederholung:

```
F=D*x;      // HOOK                    // Force(x)
a=F/M;      // NEWTON: F=a*M           // compute acceleration
```

Dies gelte für einen winzigen Zeitraum dt. Je nach Geschwindigkeit des Rechners werden etwa 10 ms (dt=0.010 s) benutzt.

Die Geschwindigkeit der Masse ist danach:

```
vi = a*dt;                            // change of velocity
v=v+vi;                               // new effective velocity
```

Mit dieser Geschwindigkeit v bewegt sich für einen winzigen Zeitraum die Masse.
Die Wegänderung ist dadurch:

```
ds=v*dt;                              // with actual speed
```

Und der zurückgelegte Gesamtweg seit t=0 ist

```
x=x-ds;
```

Das negative Vorzeichen: Die Auslenkung wird kleiner.

Gehe zu Wiederholung ===>

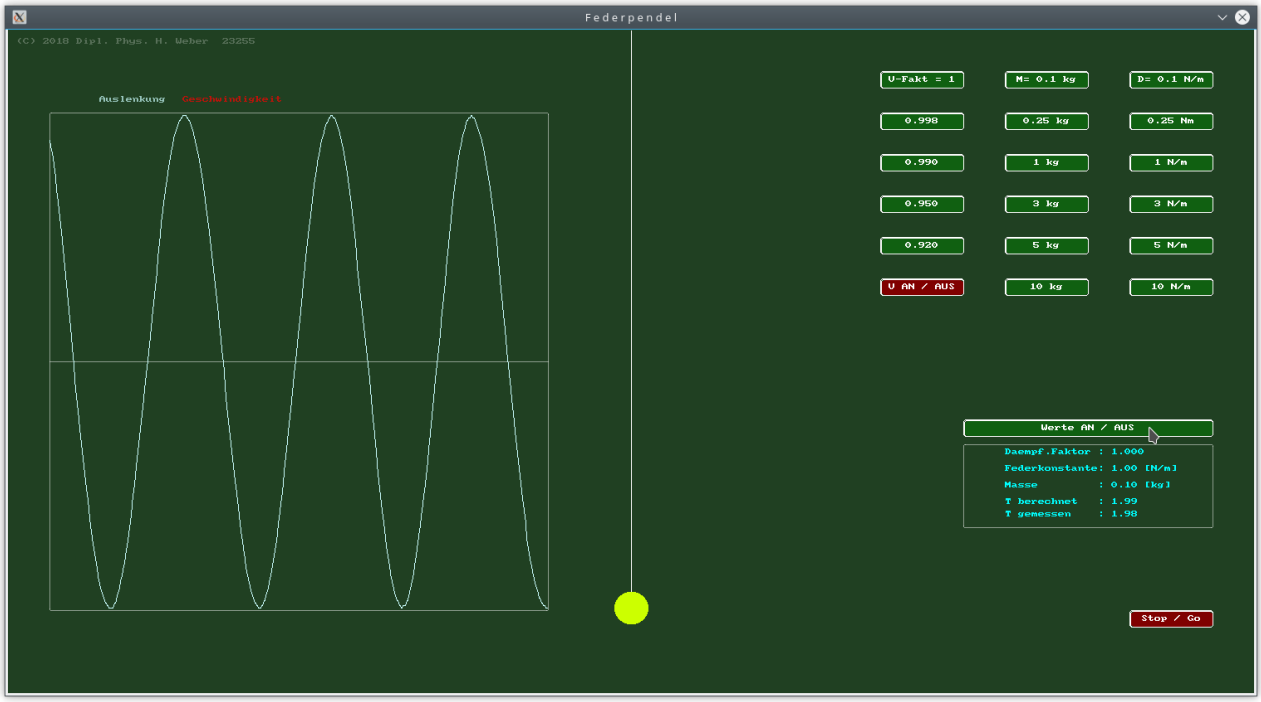
Diese Schritte kann jeder Schüler nachvollziehen und die ersten Werte nachrechnen !

Anfangswerte:

```
double D = 1;                 // spring constant [N/m]
double M = 0.1;               // Mass [kg]
double x = 0.1;               // deflection from NULL [m]  (10 cm)
double t=0;                   // actual time
```

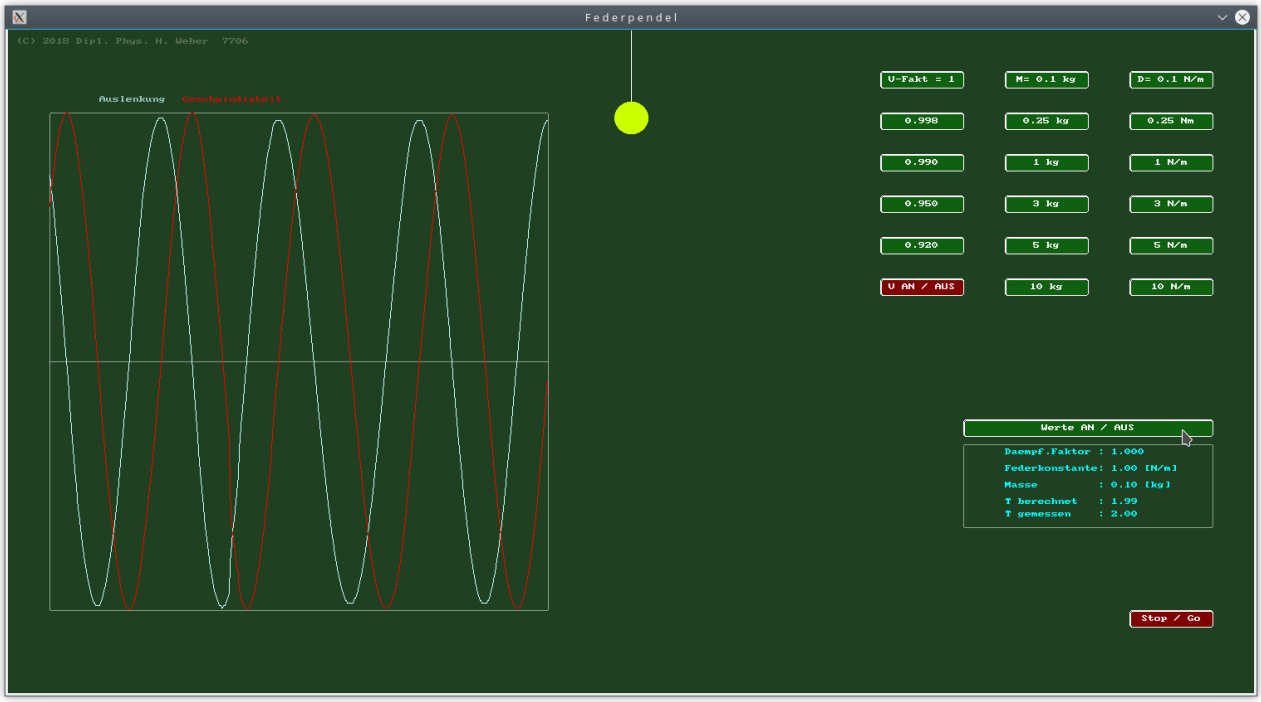
Das Verfahren bedeutet praktisch nichts anderes, als dass ein **Analogrechner digital simuliert** wird, allerdings ohne die Schwingungsgleichungen zu kennen!

Und zu meinem eigenen Erstaunen liefert das hervorragende Werte !
Hier die grafische Umsetzung:



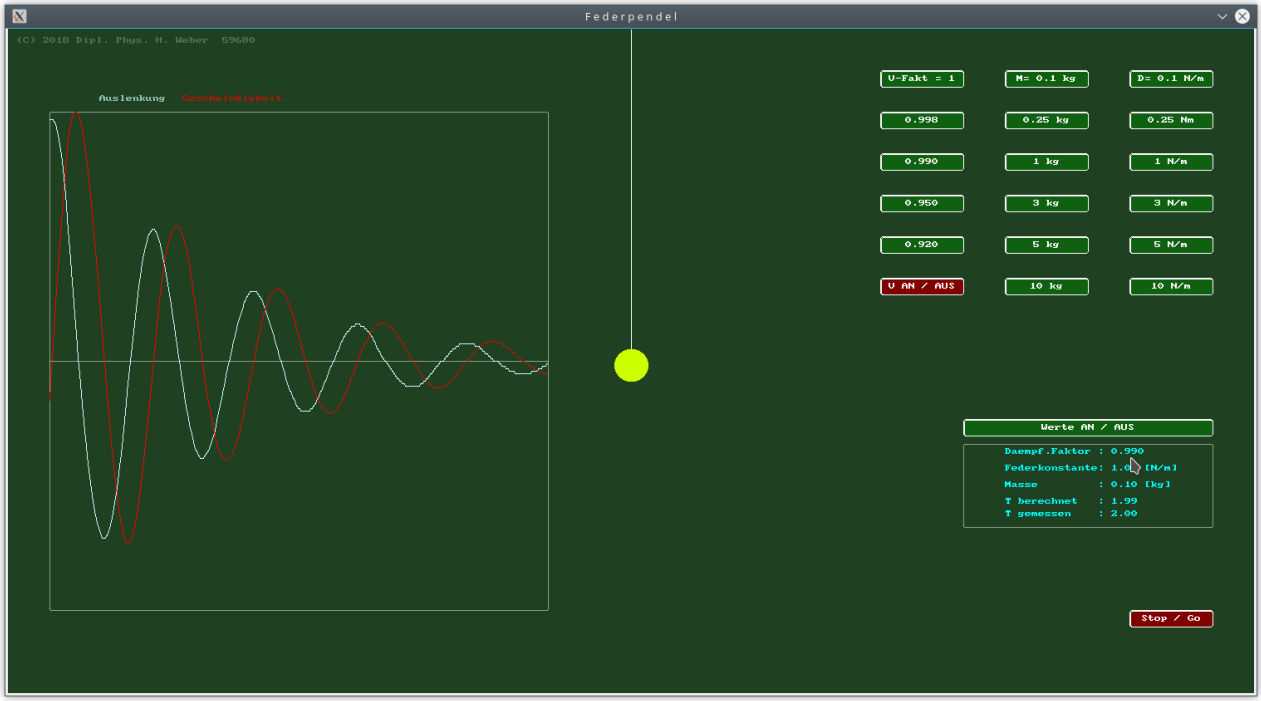
Selbst Pi ist nicht in die Berechnungen eingegangen.
Es kann nach der vorgegebenen Gleichung von T sogar recht gut aus der Simulation errechnet werden. Meines Wissens eine neue Art, Pi zu berechnen ;)
Auch die Werte der Sinusfunktion ergeben sich so, ohne die Sinusfunktion zu kennen.

Selbst die Geschwindigkeit der Masse ist korrekt darstellbar. (Das „Oszilloskop“ läuft vom rechts nach links, um Pendel und Auslenkung direkt nebeneinander zu sehen).



Auch eine gedämpfte Schwingung lässt sich so simulieren (Damp < 1):

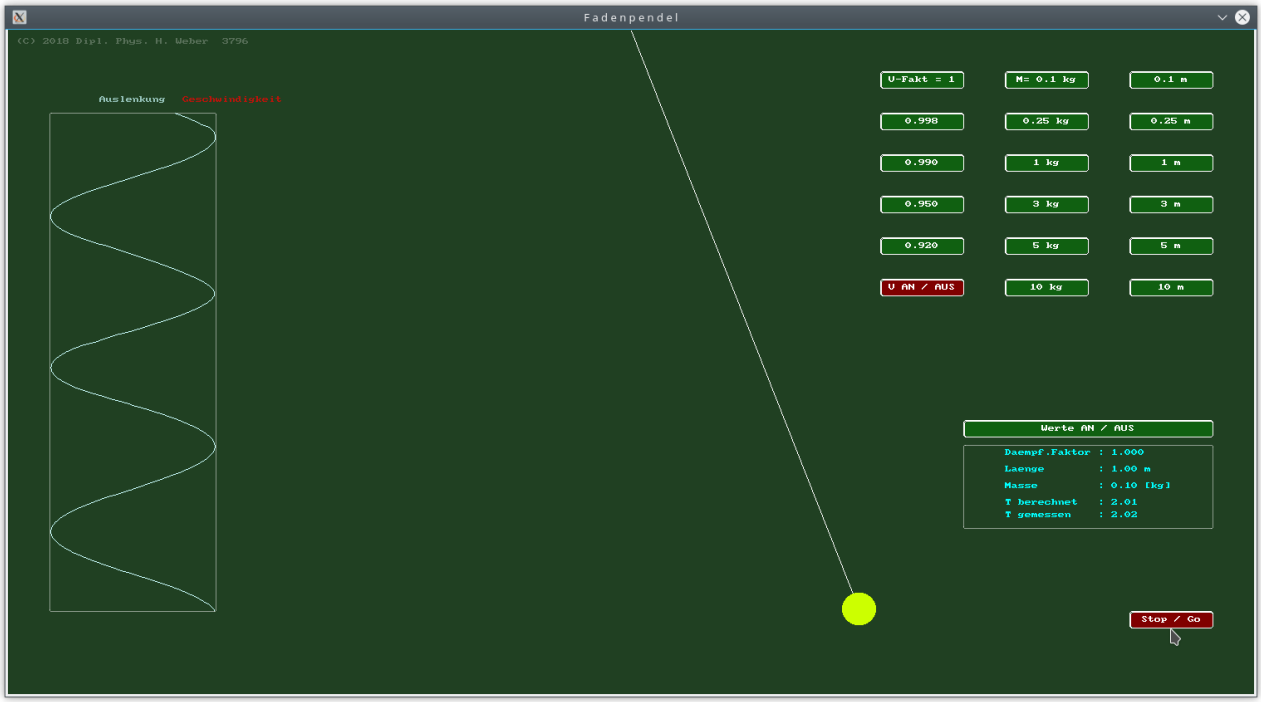
```
 vi = a*dt;                    // change of velocity
 v=v+vi;                       // new effective velocity
 v=Damp*v;                     // damping factor (1 = no damping)
```



Da das Ergebnis so gut ausfällt, habe ich es in gleicher Form mit dem Fadenpendel versucht:
Große Vereinfachung:

```
void doA() {                                    // compute the accelertion
  alpha=x/l;    // Sin(alpha) = alpha for small angles   // for one cycle
  h=l*alpha;                                    // height over NULL
  a=g*x/l;      // F=a*m  ==  m*g * x/l | :m     // compute acceleration

  //printf("A %e\n",a);
}

void doV() {                                    // compute the speed
double vi;
  vi = a*dt;
  v=v+vi;
  v=Damp*v;
}

void doS() {                                    // compute the way for one cycle
  ds=v*dt;
  //printf("ds %e \n",ds);                      // with actual speed
}

void doX() {                                    // compute the distance from NULL
  x=x-ds;
}
```

Auch hier kann sich das Ergebnis sehen lassen:



Diese Methode funktioniert auch bei: (Es fehlt noch die grafische Umsetzung)

*** Entladung eines Plattenkondensators über einen Widerstand.**
*** Elektrischer Schwingkreis mit und ohne Dämpfung**
*** …**

Die simulierten Experimente sind geeignet, um die Schüler eigene Messungen machen zu lassen: 10 Schwingungen ausmessen und für verschiedene Parameter T bestimmen.

Kleine Handexperimente können vorab die Echzeit-Simulation bestätigen:
- 100 g Tafel Schokolade am Gummiband.#
- Murmel am Faden

Dann kann der weitere „Versuchsaufbau“ mit der Simulation folgen, um Messungen vorzunehmen.

Vielleicht konnte ich demonstrieren, wie z. Bsp. aus Hook und Newton die Lösung einer Differentialgleichung 2. Ordnung ohne weiteres Wissen (Winkelfunktionen, Ableitung) auf natürlichem Wege durch Simulation erfolgen kann.

Vielleicht konnte ich Dir ein paar Anregungen geben ;)

Mit besten Grüßen

Helmut