The Simple Harmonic Oscillator

S. Bilbao

The Simple Harmonic Oscillator

Already seen this in Acoustics!

Very simple, but a microcosm of everything oscillatory in musical acoustics

Numerically becomes more complicated than the continuous time SHO! There is just one SHO, but an infinite number of ways to simulate it.

They are not equally good!

Required reading; Numerical Sound Synthesis Chapters 2 and 3

The Simple Harmonic Oscillator

Mass spring system:

$$M\ddot{u} - f = 0 \qquad f = -Ku$$

$u(t)$:	displacement	$M > 0$:	mass
$f(t)$:	force	$K > 0$:	stiffness

First order (symmetric) form:

$$M\dot{v} - f = 0 \qquad \frac{1}{K}\dot{f} + v = 0$$

$v(t)$:	velocity
$f(t)$:	force

NB: f,v are "power-conjugate" (their product has units of Watts)…this has implications for energy analysis

Second Order Form

Second-order Form:

$$\ddot{u} + \omega_0^2 u = 0$$

$$u(t): \quad \text{displacement} \qquad \omega_0 = \sqrt{K/M}: \quad \text{angular frequency}$$

$$f_0 = \frac{1}{2\pi}\sqrt{K/M}: \quad \text{frequency in Hz}$$

Initial conditions (need two!):

$$u(0) = u_0 \qquad \dot{u}(0) = v_0$$

NB:

- **You see first order systems in: electromagnetics, circuit modelling (including virtual analog)**
- **You see second order systems in mechanical engineering (vibration problems)…we'll work with this for the rest of today.**

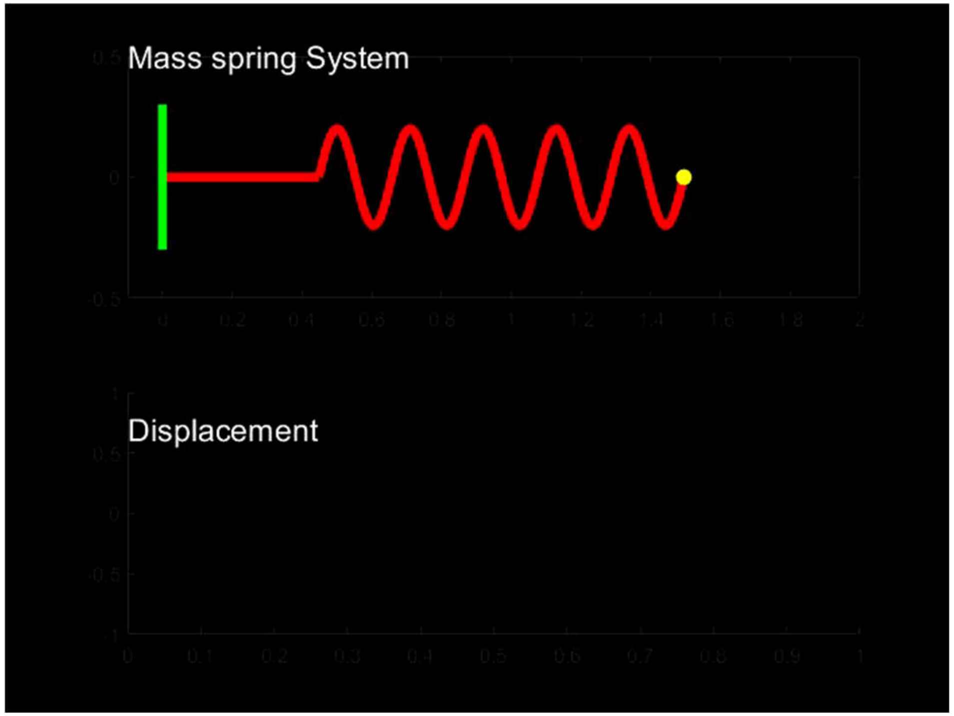
Solution

Solution:

$$u(t) = A\cos(\omega_0 t) + B\sin(\omega_0 t)$$

Constants follow from initial conditions:

$$A = u_0 \qquad B = v_0/\omega_0$$



Energy

Energy: multiply SHO by *du/dt*:

$$\ddot{u} + \omega_0^2 u = 0 \qquad \rightarrow \qquad \dot{u}\ddot{u} + \omega_0^2 \dot{u} u = 0$$

But:

$$\dot{u}\ddot{u} = \frac{d}{dt}\left(\frac{1}{2}\dot{u}^2\right) \qquad u\dot{u} = \frac{d}{dt}\left(\frac{1}{2}u^2\right)$$

Yields a conserved energy H(t):

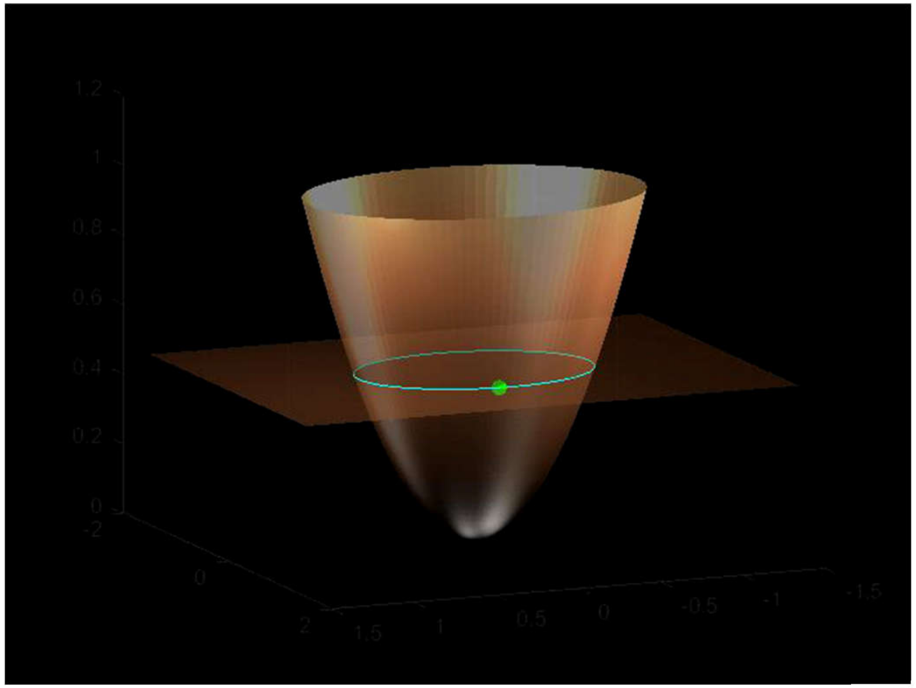
$$\frac{d}{dt}\underbrace{\left(\frac{1}{2}\dot{u}^2 + \frac{\omega_0^2}{2}u^2\right)}_{H(t)} = 0$$

$$H(t) = H(0) \geq 0$$

Solution Bound: $|u(t)| \leq \sqrt{2H(0)}\,/\,\omega_0$

NB: need to scale by M to get physical energy in Joules!

Phase Space: Solution trajectory remains on an elliptical cross section of a paraboloid: thus bounded!



Discrete Time and Difference Operators

- u(t) is approximated by a time series u^n, at multiples of the time step k:

$$u^n \cong u(nk) \qquad k = 1/F_s$$

- Basic operation are unit shifts and the identity:

$$e_{t+}u^n = u^{n+1} \qquad e_{t-}u^n = u^{n-1} \qquad 1u^n = u^n$$

- Various approximations to a first derivative:

Forward $$\delta_{t+} = \frac{1}{k}\left(e_{t+} - 1\right) \quad \rightarrow \quad \delta_{t+}u^n = \frac{1}{k}\left(u^{n+1} - u^n\right) \cong \frac{du}{dt}$$

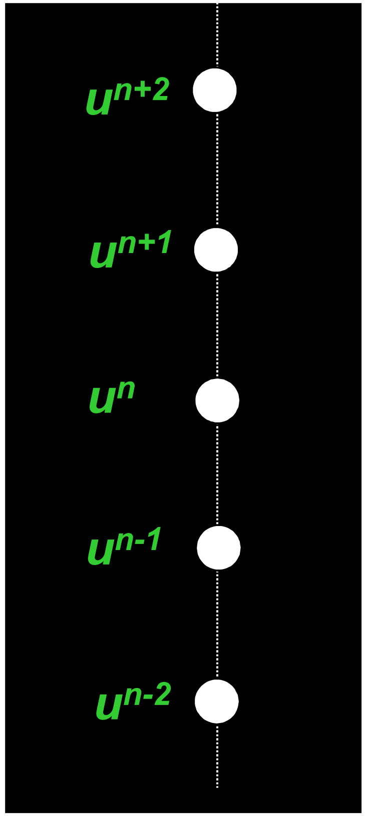
Backward $$\delta_{t-} = \frac{1}{k}\left(1 - e_{t-}\right) \quad \rightarrow \quad \delta_{t-}u^n = \frac{1}{k}\left(u^n - u^{n-1}\right) \cong \frac{du}{dt}$$

Centered $$\delta_{t\cdot} = \frac{1}{2k}\left(e_{t+} - e_{t-}\right) \quad \rightarrow \quad \delta_{t\cdot}u^n = \frac{1}{2k}\left(u^{n+1} - u^{n-1}\right) \cong \frac{du}{dt}$$

- A useful centered approximation to a second derivative:

$$\delta_{tt} = \delta_{t+}\delta_{t-} = \frac{1}{k^2}\left(e_{t+} - 2 + e_{t-}\right) \quad \rightarrow \quad \delta_{tt}u^n = \frac{1}{k^2}\left(u^{n+1} - 2u^n + u^{n-1}\right) \cong \frac{d^2u}{dt^2}$$

and many others…



Simple Scheme for the SHO

The simplest possible scheme for the SHO:

$$\ddot{u} + \omega_0^2 u = 0 \qquad \rightarrow \qquad \delta_{tt} u + \omega_0^2 u = 0$$

Compact "operator" form

Expand out operators to get full recursion:

$$\delta_{tt} u + \omega_0^2 u = 0 \qquad \rightarrow \qquad \frac{1}{k^2}\left(u^{n+1} - 2u^n + u^{n-1}\right) + \omega_0^2 u^n = 0$$

Update:

$$u^{n+1} = \underbrace{\left(2 - \omega_0^2 k^2\right)}_{b} u^n - u^{n-1}$$

Operation count:

1 multiplication, two additions per sample. You can and should precompute "b" outside the run time loop!

Memory:

Requires access to last two computed samples---no more! Can and should discard previously computed samples, and overwrite instead. This becomes very important for PDE systems…

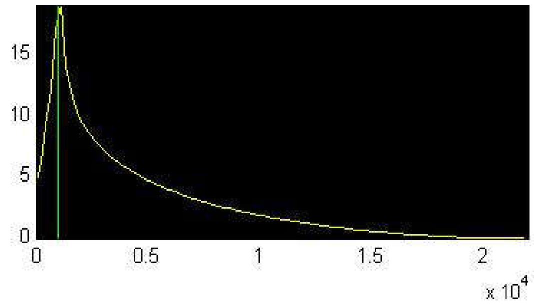
See code example in NSS…

Frequency Warping

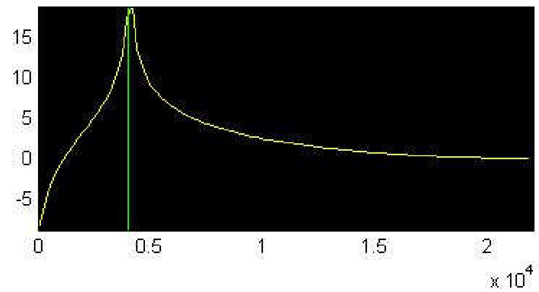
This simple scheme does a good job at lower frequencies of oscillation (relative to the sample rate!). At higher frequencies, frequency warping effects appear.

At a sample rate of 44.1 kHz:

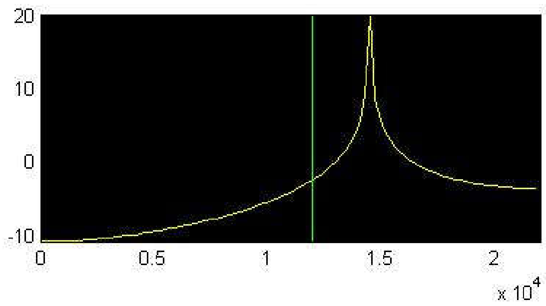
f0 = 1 kHz



f0 = 4 kHz



f0 = 12 kHz



Ultimately, the algorithm becomes unstable!

f0 = 14.04 kHz

Warping and instability are pervasive problems, and get much more serious for PDE systems!

S. Bilbao

Frequency Domain Analysis

- Analyze using frequency domain analysis (transient solutions): assume solution:

$$u^n = z^n \qquad \text{where} \qquad z = e^{sk}, \qquad s = j\omega + \sigma$$

- NB:

ω_0 : target frequency

ω : numerical frequency of oscillation of the approximate scheme

- This is essentially the same as z transform analysis used in DSP---here, though, there are no samples! We are analysing an algorithm

- Leads to characteristic equation:

$$z + \omega_0^2 k^2 + z^{-1} = 0$$

- Expression for warping:

$$\omega = \frac{2}{k} \sin^{-1}\left(\omega_0 k/2\right)$$

Frequency Domain Analysis

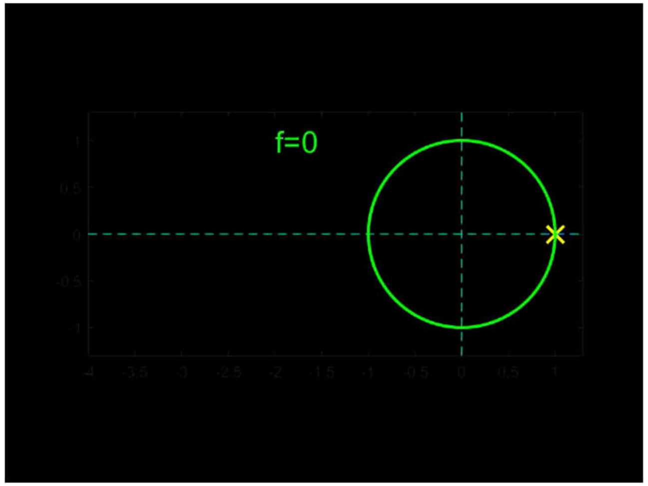
- Need solutions with

$$|z| \leq 1$$

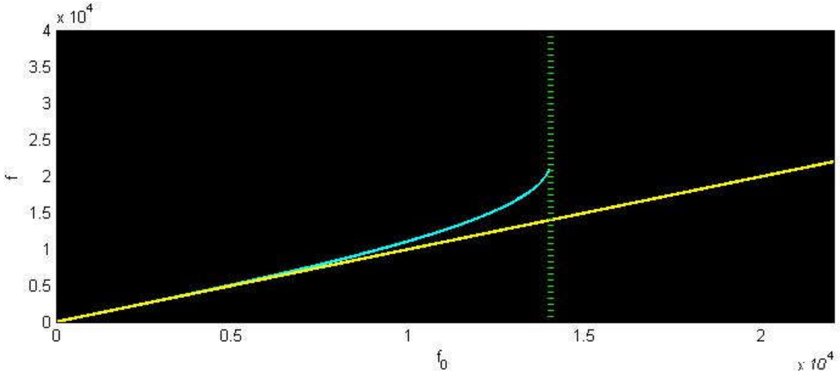
- true under stability condition:

$$k \leq \frac{2}{\omega_0} = \frac{1}{\pi f_0} \qquad \rightarrow \qquad F_s \geq \pi f_0$$

- an upper bound on
  the time step…



- and we can see warping effect explicitly: scheme possesses a numerical frequency distinct from that of the SHO!



Energetic manipulations

- Consider the following products, in continuous time:

$$\dot{u}\ddot{u} = \frac{d}{dt}\left(\frac{1}{2}\dot{u}^2\right) \qquad\qquad u\dot{u} = \frac{d}{dt}\left(\frac{1}{2}u^2\right)$$

- Can perform similar manipulations using difference operators…

$$\begin{aligned}\delta_{t\bullet}u^n\delta_{tt}u^n &= \frac{1}{2k}\left(u^{n+1}-u^{n-1}\right)\frac{1}{k^2}\left(u^{n+1}-2u^n+u^{n-1}\right)\\ &= \frac{1}{k}\frac{1}{2}\left(\left(\frac{u^{n+1}-u^n}{k}\right)^2-\left(\frac{u^n-u^{n-1}}{k}\right)^2\right)\\ &= \delta_{t+}\left(\frac{1}{2}\left(\delta_{t-}u^n\right)^2\right)\end{aligned}$$

$$\begin{aligned}\delta_{t\bullet}u^n u^n &= \frac{1}{2k}\left(u^{n+1}-u^{n-1}\right)u^n\\ &= \frac{1}{k}\frac{1}{2}\left(u^{n+1}u^n-u^n u^{n-1}\right)\\ &= \delta_{t+}\left(\frac{1}{2}u^n e_{t-}u^n\right)\end{aligned}$$

- Total derivatives become total differences…can use this to prove numerical stability.

Energy-based stability

- Perform same manipulations with scheme for SHO:

$$\delta_{tt}u+\omega_0^2u=0 \quad \rightarrow \quad \delta_{t\bullet}u\delta_{tt}u+\delta_{t\bullet}u\omega_0^2u=0$$

$$\rightarrow \quad \delta_{t+}\underbrace{\left(\frac{1}{2}(\delta_{t-}u)^2+\frac{\omega_0^2}{2}ue_{t-}u\right)}_{h}=0$$

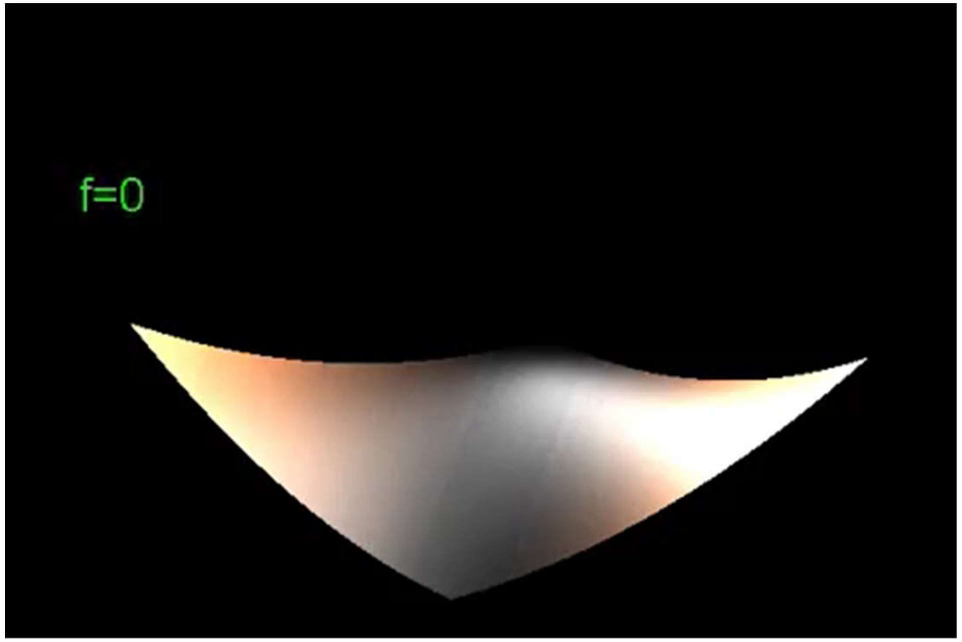
- A discrete conserved energy for the scheme follows:

$$h^n=\frac{1}{2k^2}\left(\left(u^n\right)^2+\left(\omega_0^2k^2-2\right)u^nu^{n-1}+\left(u^{n-1}\right)^2\right)$$

Energy-based stability

$$h^n = \frac{1}{2k^2}\left(\left(u^n\right)^2 + \left(\omega_0^2 k^2 - 2\right)u^n u^{n-1} + \left(u^{n-1}\right)^2\right)$$

- But…discrete conserved energy is not necessarily non-negative!



- Just a quadratic form…non-negative (a paraboloid!) under condition

$$k \le \frac{2}{\omega_0} = \frac{1}{\pi f_0}$$

- Which is the same as the condition arrived at through frequency domain analysis…

Energy vs. Frequency Domain Analysis

- Frequency domain methods:
  - Useful only for linear and time invariant systems
  - Give complete information on stability and warping
- Energy methods:
  - Generally useful for linear and nonlinear systems
  - Give information on stability, but none relating to warping!

Going further: averaging operators and parameterized schemes

Useful to introduce an averaging operator…

$$\mu_{t\bullet} = \frac{1}{2}\left(e_{t+} + e_{t-}\right) \qquad \rightarrow \qquad \mu_{t\bullet} u^n = \frac{1}{2}\left(u^{n+1} + u^{n-1}\right) \cong u$$

Consider the following parameterized scheme…

Operator form | **Update form**

$$\delta_{tt} u = -\omega_0^2\left(\alpha + (1-\alpha)\mu_{t\bullet}\right)u \qquad \rightarrow \qquad u^{n+1} = \frac{\left(2-\alpha\omega_0^2 k^2\right)}{1+(1-\alpha)\omega_0^2 k^2/2} u^n - u^{n-1}$$

Stability condition:

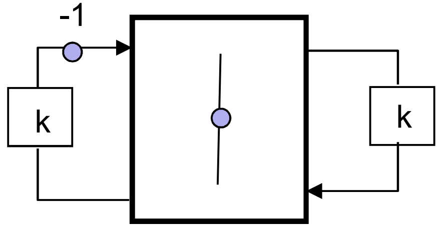
$$k \le \frac{2}{\omega_0\sqrt{2\alpha - 1}} \qquad \textit{when} \qquad \alpha \ge 1/2, \qquad \text{otherwise stable}$$

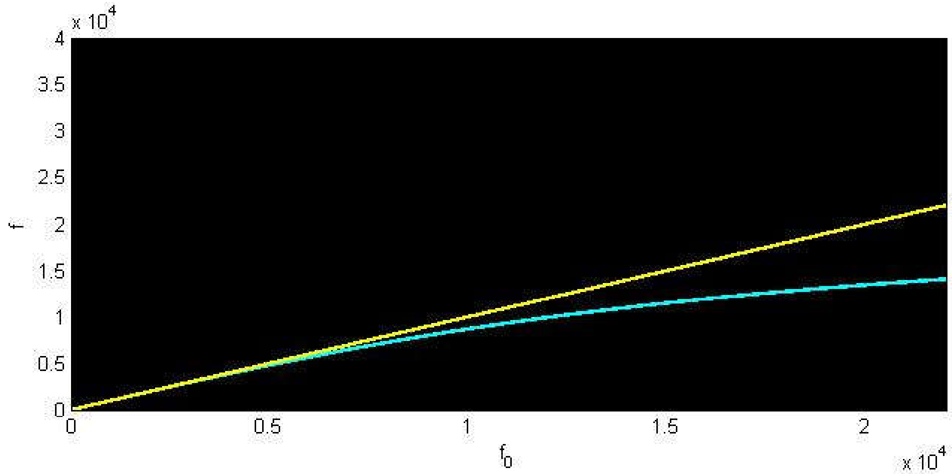
Scheme requires a division…this becomes a linear system solution in PDE systems (implicit!)
NB: reduces to the simple scheme when alpha = 1!

Special Cases: Trapezoid Rule

$\alpha = 1/2$

This is otherwise known as the famous “trapezoid rule” of numerical integration. Used as the basis for wave digital flters, heavily used in virtual analog modelling!

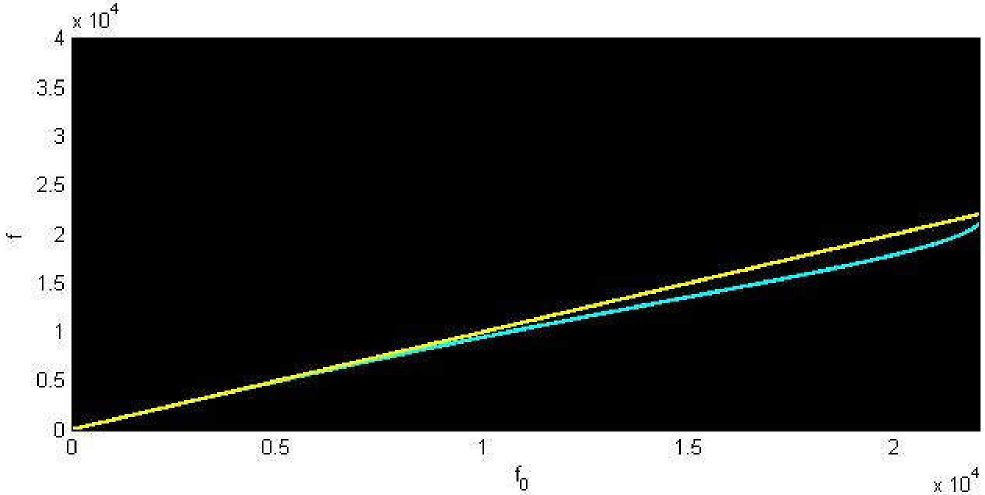




Unconditionally stable! But frequency of oscillation is too low!

Special Cases: Optimised method

$\alpha = 0.7$



Good match over whole frequency range…at virtually no additional cost in terms of operations/memory

Special Cases: Exact Method

$$\alpha = \frac{2}{\omega_0^2} - \frac{\cos(\omega_0 k)}{1-\cos(\omega_0 k)}$$

$$u^{n+1} = 2\cos(\omega_0 k) u^n - u^{n-1}$$

Gives an exact solution!

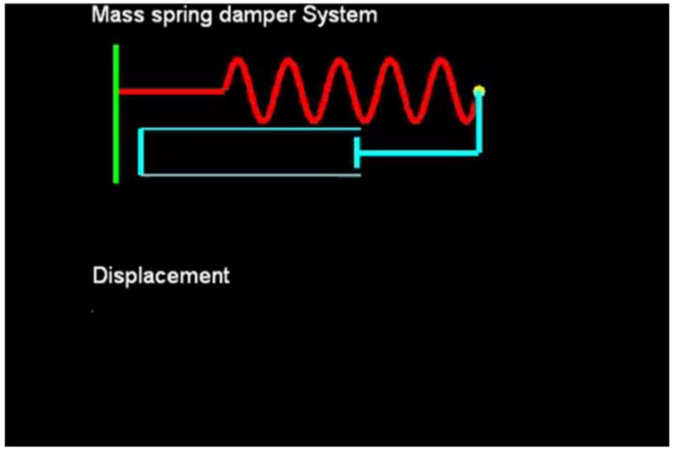
But requires foreknowledge of oscillating frequency…we have the exact frequency in this case, but in more realistic settings we will not.

This is essentially a “modal” method…

Loss

Adding a dashpot…

$$\frac{d^2u}{dt^2} = -\omega_0^2 u - 2\sigma\frac{du}{dt} \qquad \sigma: \ \text{loss parameter}$$



Adjust scheme accordingly…

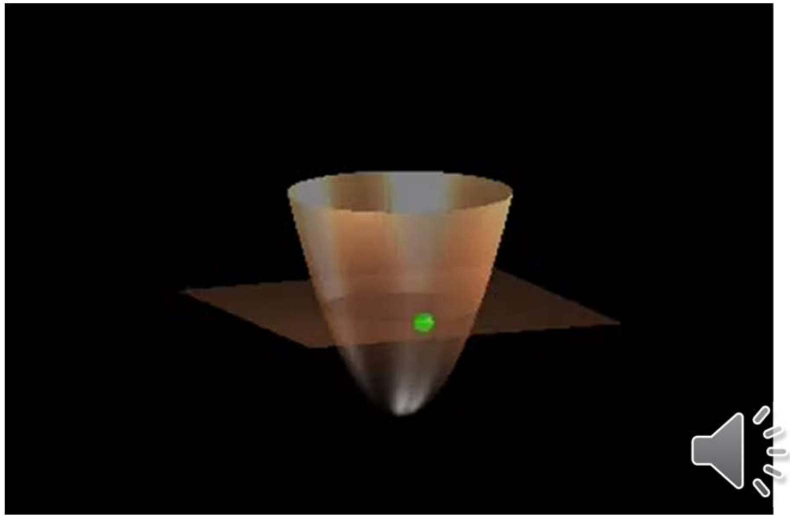
Operator form **Update form**

$$\delta_{tt}u = -\omega_0^2 u - 2\sigma\delta_{t\cdot}u \qquad \rightarrow \qquad u^{n+1} = \frac{2-\omega_0^2 k^2}{1+\sigma k}u^n - \frac{1-\sigma k}{1+\sigma k}u^{n-1}$$

Stable under same conditions as lossless case
Discrete energy is now monotonically non-increasing

$$0 \leq h^n \leq h^{n-1}$$

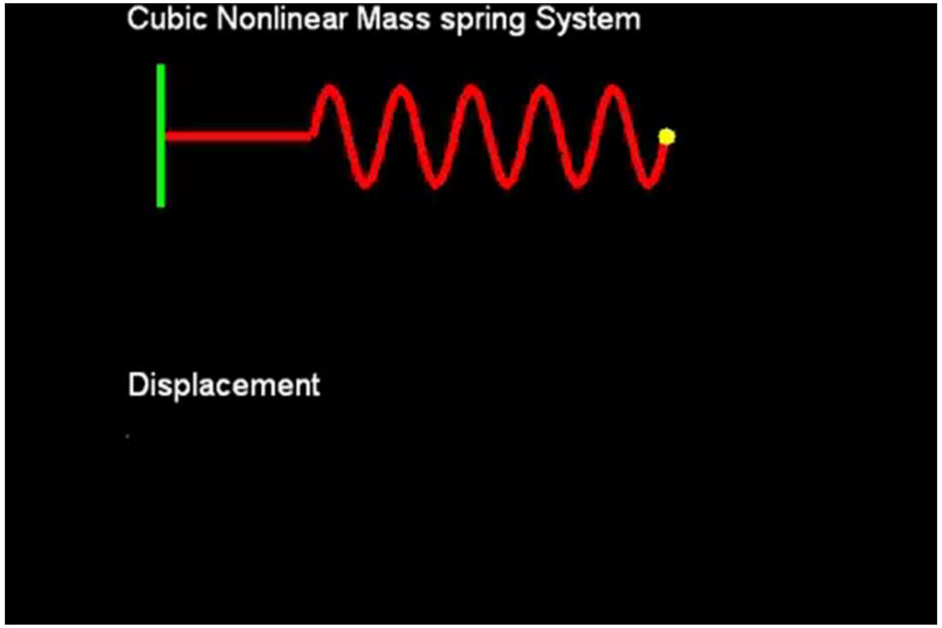
S. Bilbao



Cubic nonlinear oscillator

- A simple nonlinear test problem: $$\frac{d^2u}{dt^2} = -\omega_0^4 u^3$$
- Frequency domain analysis unwieldy (perturbation methods are a possibility!)
- Solution can again be bounded using energy methods

$$H = \frac{1}{2}\left(du/dt\right)^2 + \frac{\omega^4}{4}u^4 \geq 0$$



Cubic nonlinear oscillator

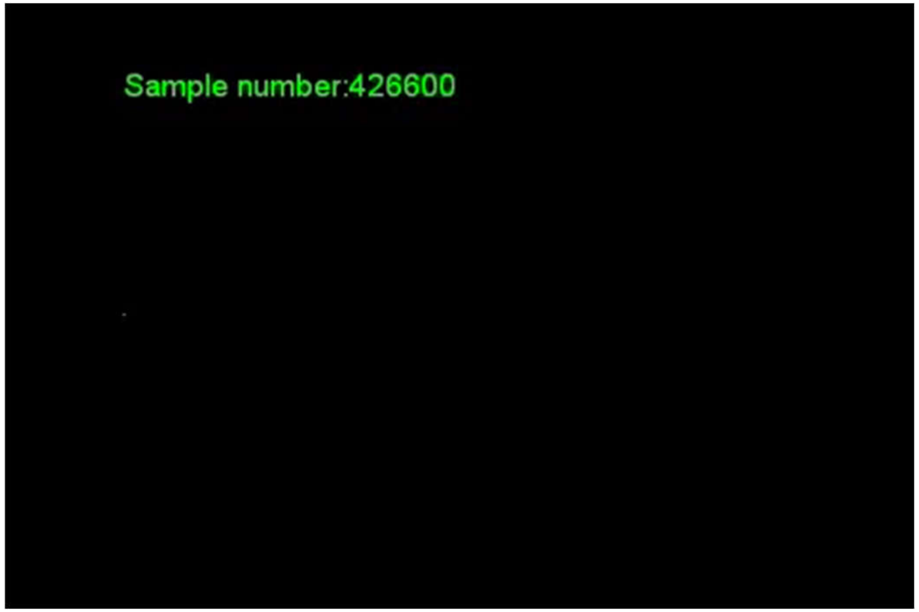
- Very large number of possible methods…with distinct properties!

$$\delta_{tt}u^n = -\omega^4 (u^n)^3 \quad \rightarrow \quad u^{n+1} = 2u^n - u^{n-1} - k^2\omega^4 (u^n)^3 \quad \rightarrow \quad \text{explicit, no stability guarantee!}$$

$$\delta_{tt}u^n = -\frac{\omega^4}{2}(u^n)^2\left(u^{n+1} + u^{n-1}\right) \quad \rightarrow \quad u^{n+1} = \frac{2}{1 + k^2\omega^4 (u^n)^2/2}u^n - u^{n-1} \quad \rightarrow \quad \text{explicit, unconditionally stable!}$$

$$\delta_{tt}u^n = -\frac{\omega^4}{4}\left(\left(u^{n+1}\right)^2 + \left(u^{n-1}\right)^2\right)\left(u^{n+1} + u^{n-1}\right) \quad \rightarrow \quad \rightarrow \quad \text{unconditionally stable, no explicit update!}$$

- Using an “ad hoc” numerical method in this case can lead to disaster…



End