E-fólio B | Folha de resolução para E-fólio

UNIDADE CURRICULAR: FÍSICA GERAL

CÓDIGO: 21048

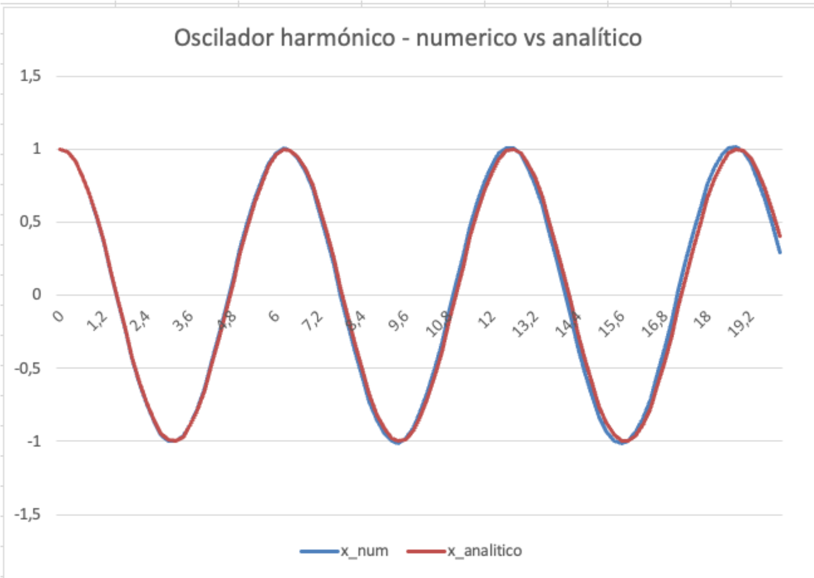
DOCENTE: Nuno Sousa/Ana Valadares

A preencher pelo estudante

ANO LETIVO: 2025-26

TRABALHO / RESOLUÇÃO:

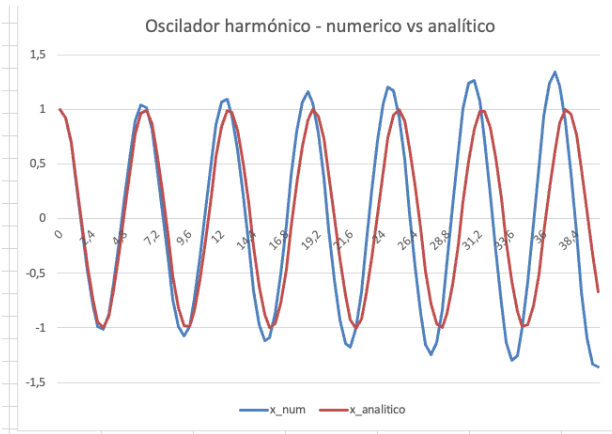
(a) Esta é a parte fácil. Implementando os algoritmos RK2 para equações diferenciais de 2ª ordem (i.e., reduzindo a duas equações de 1ª ordem) obtêm-se, para os valores do enunciado,



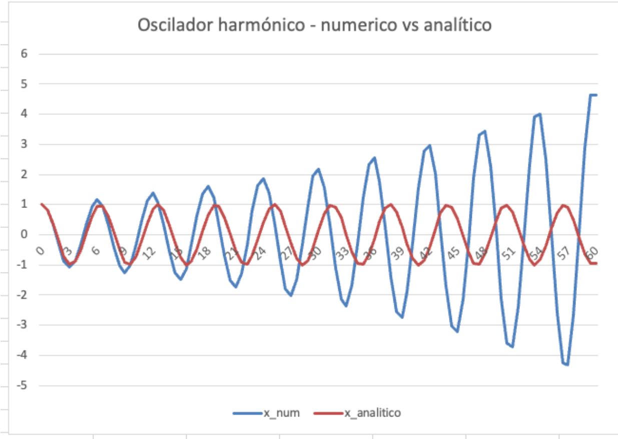
Até aqui tudo normal. Nota-se um ligeiro afastamento no final do intervalo, mas, como a p.17 do texto de apoio 2 sugere, deverão ser apenas acumulações de erros numéricos.

(b) Alterando o passo o resultado começa a flutuar estranhamente:

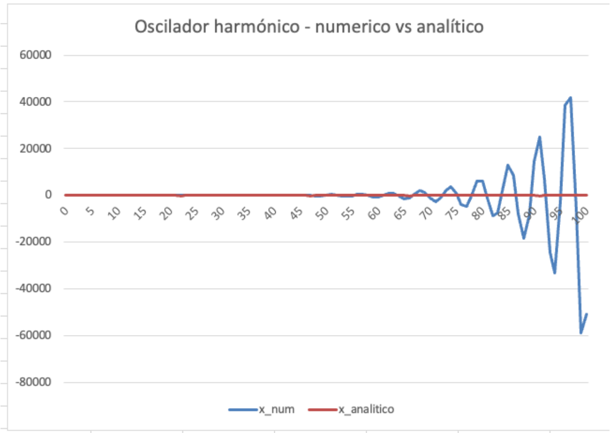
h = 0,4 s



h = 0,6 s



h = 1,0 s



Quando chegamos a passo de 1,0 segundos, vemos claramente que há qualquer coisa de errado com os métodos numéricos. É normal um método numérico afastar-se da solução analítica, mas não é normal afastar-se tanto.

O que se passa é que não só a amplitude se altera, crescendo exageradamente, com também o período começa a desfasar-se relativamente à solução exata, tornando-se mais curto.

Claramente estamos a observar um acumular exagerado de erros numéricos, mas porque será?

(c) Perguntando ao chatGPT o que poderá estar a acontecer obtemos este diálogo: (o professor prefere inglês para conversas técnicas)

hi. the harmonic oscillator differential equation can be integrated by the usual reduction to a system of two coupled 1st order DE that can be solved using ordinary 2nd order runge-kutta methods. however, when I do this, there seems to be some numerical instability because the system starts gaining energy and wideining its amplitude. is this known?

Thought for 14s ›

Yes — this is a known and expected behaviour.

What's happening (short version)

If you turn the harmonic oscillator

$$\ddot{x} + \omega^2 x = 0$$

into the first-order system

$$\dot{q} = p, \qquad \dot{p} = -\omega^2 q$$

and apply an ordinary explicit second-order Runge–Kutta (RK2) method, the numeric map that advances the solution has **spectral radius > 1** for purely imaginary eigenvalues of the linearized problem (the harmonic oscillator). In plain language: RK2 is **not** stable on the imaginary axis and will generally *pump energy* into the system, so the amplitude slowly grows.

A tiny linear analysis (why energy grows)

For a linear ODE $\dot{y} = Ay$ a one-step method produces

$$y_{n+1} = R(hA)\, y_n,$$

where $R(z)$ is the stability polynomial of the method. For any order-2 explicit RK method the stability polynomial is

$$R(z) = 1 + z + \tfrac{1}{2}z^2.$$

Plug in $z = i\theta$ (with $\theta = h\omega$). Then

$$R(i\theta) = 1 + i\theta - \tfrac{1}{2}\theta^2,$$

so

$$|R(i\theta)|^2 = (1 - \tfrac{1}{2}\theta^2)^2 + \theta^2 = 1 + \tfrac{1}{4}\theta^4 > 1 \quad (\theta \neq 0).$$

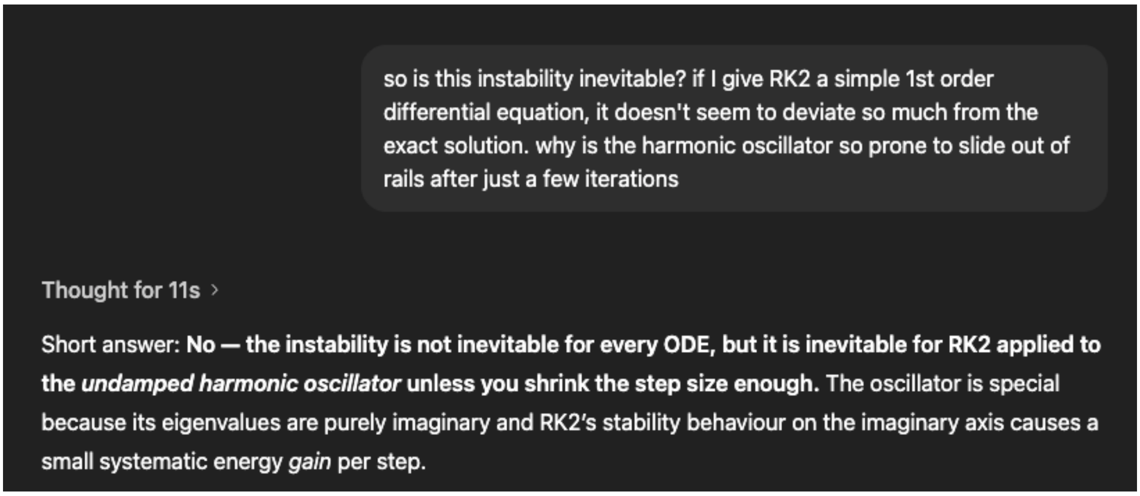
Hence $|R(i\theta)| > 1$ for any nonzero step-size, so errors are amplified and the discrete solution's amplitude (and therefore energy) grows slowly with steps.

(For small θ, $|R| \approx 1 + \frac{1}{8}\theta^4$, so the growth is slow but inexorable unless h is tiny.)

Apesar de toda a linguagem técnica, lendo com cuidado é possível tirar uma conclusão simples: o método RK2 no geral não é estável.

Para o problema do oscilador harmónico em particular, o polinómio estabilizador é sempre maior do que zero, ou seja, para cada período é bombeada energia (inexistente, originada pelo método RK2) para o oscilador, resultando no aumento da amplitude e frequência (ou diminuição do período, que é o que se vê dos gráficos). Recordar que a energia do oscilador harmónico é $E = \frac{1}{2} m A^2 \omega^2$, portanto aumento de energia significa aumento de A e ω.

Mas então porque é que isto parece não acontecer com outras equações diferenciais? Novamente é mais rápido ir ao chatGPT do que procurar na enciclopédia:



so is this instability inevitable? if I give RK2 a simple 1st order differential equation, it doesn't seem to deviate so much from the exact solution. why is the harmonic oscillator so prone to slide out of rails after just a few iterations

Thought for 11s ›

Short answer: **No — the instability is not inevitable for every ODE, but it is inevitable for RK2 applied to the *undamped harmonic oscillator* unless you shrink the step size enough.** The oscillator is special because its eigenvalues are purely imaginary and RK2's stability behaviour on the imaginary axis causes a small systematic energy *gain* per step.

Ou seja, tecnicalidades à parte, o problema tem a ver com o facto de a função de estabilidade apontar sempre no sentido de crescimento dos erros. É um caso típico do que se designa por *instabilidade numérica*.

Para outras equações diferenciais a função de estabilidade normalmente sobe e desce, pelo que os erros se vão anulando e a solução numérica segue aproximadamente a exata.

(d) A única solução para o problema dentro do método RK2 é baixar o passo até valores suficientemente baixos para que em todo o intervalo de integração a instabilidade não cresça demasiado.

O chatGPT sugere outras soluções:

Practical consequences & remedies

1. **Don't expect energy conservation from explicit RK2.** It's not symplectic or A-stable for oscillatory systems, so it drifts. This is the reason you see amplitude increasing.
2. **Use a symplectic integrator** if you care about long-term behaviour / energy:
   - **Leapfrog / Stoermer–Verlet (explicit, second order)** — simple, explicit, symplectic, and widely used for oscillatory/Hamiltonian systems. It keeps energy bounded and typically shows small bounded oscillations around the true energy (no secular drift).
   - **Symplectic Euler** — first order but symplectic (may be OK if you need cheap steps).
   - **Implicit midpoint rule** — second order and symplectic (but implicit).
   - **Higher-order symplectic Runge–Kutta (Gauss–Legendre)** — if you need higher order and symplecticness (implicit).
3. **Decrease the step size** h — this reduces the growth rate (since growth $\propto \theta^4$ for small θ), but it doesn't remove the underlying non-symplectic instability.
4. **Energy-preserving or discrete gradient methods** — if you need exact energy conservation (discrete), consider methods specifically designed for that.

Quick example: leapfrog (velocity Verlet) for the oscillator

A simple, explicit symplectic integrator is the leapfrog (velocity Verlet) update:

python Copy code

```python
# assume q, p at time t, step h, omega given
p_half = p - 0.5*h*omega**2 * q        # half-step momentum
q_next  = q + h * p_half               # full-step position
p_next  = p_half - 0.5*h*omega**2 * q_next  # finish momentum
```

This method is second-order and will not exhibit the secular energy blow-up you reported.

Para resolver, ou pelo menos atenuar o problema, haveria que implementar um dos métodos sugeridos acima, sendo que a forma mais simples é mesmo reduzir o passo.

Num caso real normalmente não temos a solução exata e nem podemos usar considerações de energia. Como diagnosticar uma instabilidade então? Aqui ficam algumas sugestões:

so, in the harmonic oscillator i can monitor the energy to know if an instability is developing. but how can i diagnose a numerical instability in general if I dont know the exact solution or there are no energy considerations to rely on?

Thought for 8s ›

Good question — and a very practical one. If you *don't* have an exact solution or a known invariant (energy, mass, etc.), you still have several robust tools to **detect** when a numerical method is becoming unstable or unreliable. Below I give a prioritized checklist (fast-to-run diagnostics first), short explanations of why each works, and concrete recipes / tiny code sketches you can drop into a time-stepper.

Quick checklist (run these each step or every few steps)

1. **Step-doubling (local error estimate)** — universal, easy, and usually the best single test.
2. **Defect (residual) test** — checks how well the numerical step satisfies the ODE.
3. **Solution-norm / growth factor monitoring** — looks for unexpected exponential growth.
4. **Linear stability (Jacobian) test** — linearize and check method/stability polynomial.
5. **Ensemble sensitivity / short Lyapunov test** — detect sensitive / chaotic amplification.
6. **Embedded methods** — use a pair with error estimate (if you aren't using step-doubling).

If any of these signals trouble, either reduce h or switch to a more stable integrator (implicit, symplectic, or an embedded RK with good stability).