Two stable schemes for the 2D Fokker-Planck equation

Gianluca Becuzzi

November 23, 2023

1 Introduction

The following system of SDEs describing a particle subject to a deterministic acceleration in a noisy environment

$$\begin{aligned} dx(t) &= v(t)dt \\ dv(t) &= a(x,v,t)dt + \sigma(x,t)dW(t) \end{aligned}$$

corresponds to the (Itô-)Fokker-Planck equation for the transition probability $p = p(x,v,t|x_0,v_0,t_0)$

$$\partial_t p + \partial_x(vp) + \partial_v(ap) = \frac{1}{2}\frac{\partial^2}{\partial v^2}(\sigma^2 p) \tag{1}$$

The first two terms are related to the deterministic advection of the particle in the phase space under a velocity field

$$\mathbf{u} = \big(v, a(x,v,t)\big) \tag{2}$$

that in the cases of interest is not solenoidal. The compression due to the divergence of the velocity field is balanced by the last diffusive term caused by noise.

Since that (1) presents as a sum of differential operators acting on a single variable, it is natural to approach the problem with the fractionary step ([1]) method. This consists in solving a series of PDEs

$$\partial_t p + \partial_v(ap) = \frac{1}{2}\partial_v^2\left(\sigma^2 p\right) \qquad x = x_i \tag{3}$$

$$\partial_t p + \partial_x(vp) = 0 \qquad v = v_j \tag{4}$$

for each value of $x_i = i\Delta x$ and $v_j = j\Delta v$ of a regular grid.

Both (1) and (3-4) represent conservation laws of the whole 2D domain and the 1D line respectively, thus the implemented numerical solutions have the form of a conservative scheme ([2]) given by

$$\overline{P}_i^{n+1} = \overline{P}_i^n - \frac{\Delta t}{\Delta}(F_{i\to i+1} - F_{i-1\to i}) \tag{5}$$

where $\overline{P}_i$ is the average value of the transition probability in the i-th cell of volume ad F is a numerical flux function. Eq. (5) grants that

$$\sum_i \overline{P}_i^{n+1} = \sum_i \overline{P}_i^n - \frac{\Delta t}{\Delta}(F_{1,0} - F_{M-1,M}) \approx \sum_i \overline{P}_i^n$$

if the flux function approaches zero near the domain boundaries, so that the probability norm is conserved.

2 Integration schemes

Formally, both (4) and (3) represent the same advection-diffusion equation, with the former being a special case of the latter when $a = v$ and $\sigma = 0$. Due to this fact, in the following, I will refer to x and v by means of the general variable z and the 1D advection-diffusion equation will be studied.

For dimensional reasons the two adimensional numbers implied in the rest of the treatment are the Fourier number for diffusion and the Courant number for advection:

$$\eta = \frac{\sigma^2}{2}\frac{\Delta t}{\Delta z^2} \qquad \theta = \frac{\Delta t}{\Delta z} \tag{6}$$

2.1 Completely implicit

The fully-implicit finite-differences scheme proposed in [3] consists in the following couple of algebraic equations for the values of $p_i = p(z_i)$:

$$\frac{p_i^{n+1} - p_i^n}{\Delta t} = -\frac{F_{i+1/2}^{n+1} - F_{i-1/2}^{n+1}}{\Delta z} \tag{7}$$

$$F_{i+1/2} = a_{i+1/2}\frac{p_i + p_{i+1}}{2} - \frac{\sigma^2}{2}\frac{p_{i+1} - p_i}{\Delta z} \tag{8}$$

that, if interpreted as a finite volume conservative scheme, correspond to imposing that in the i-th cell the function $p(z)$ is constantly equal to p_i so that

$$p(z) = p_i \quad z \in [z_i, z_{i+1}]$$
$$\overline{P}_i = p_i$$

with a flux given by

$$F_{i\to i+1} = a_{i+\frac{1}{2}}\left(\overline{P}_i + \overline{P}_{i+1}\right) - \frac{\sigma^2}{2}\frac{\overline{P}_{i+1}^{n+1} - \overline{P}_i^{n+1}}{\Delta z} \tag{9}$$

The Von Neumann linear amplification coefficient is

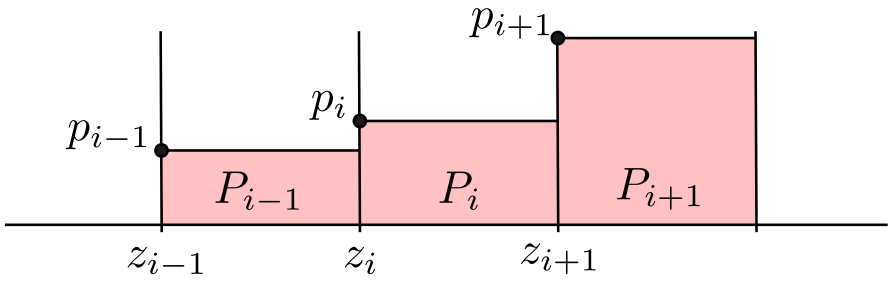


Figure 1: Interpretation of the fully implicit finite difference scheme as a finite-volume conservative one.

$$g_{IMP} = \left(1 + 4\eta\sin^2\left(\frac{k\Delta z}{2}\right) + \frac{\partial a}{\partial z}\Delta t\cos^2\left(\frac{k\Delta z}{2}\right) - 4i\theta a\sin(k\Delta z)\right)^{-1} \tag{10}$$
$$\approx \left(1 + 4\eta\sin^2\left(\frac{k\Delta z}{2}\right) - 4i\theta a\sin(k\Delta z)\right)^{-1}$$

so the scheme is stable as long as it holds true that

$$\gamma\Delta t \ll 4\eta \qquad \gamma = \max_{\Omega}(|\partial_z a|) \tag{11}$$

and the CFL condition for advection

$$\frac{A\Delta t}{\Delta z} < 1 \qquad A = \max_{\Omega}(|a|) \tag{12}$$

2.2 Integral-based scheme

Another possible strategy (which will be referred to as the Tsai scheme, since it is proposed in [4]) consists in interpolating the value of $p(z)$ in the i-th cell with a quadratic function

$$p(z) = a + bz + cz^2 \qquad z \in [z_i, z_{i+1}] \tag{13}$$

that gives for the right and left derivative at the cell boundaries

$$\left.\frac{\partial p}{\partial z}\right|_{right,i} = \frac{4p_{i+1} + 2p_i - 6\overline{P}_i}{\Delta z} \tag{14}$$

$$\left.\frac{\partial p}{\partial z}\right|_{left,\ i} = \frac{-2p_{i+1} - 4p_i + 6\overline{P}_i}{\Delta z} \tag{15}$$

and approximating the advection-diffusion equation with

$$\partial_t \overline{P}_i + \frac{a_{i+1}p_{i+1} - a_i p_i}{\Delta z} = \frac{1}{2\Delta z}\left(\sigma^2_{i+1}\left.\frac{\partial p}{\partial z}\right|_{right,i} - \sigma^2_i \left.\frac{\partial p}{\partial z}\right|_{left,i}\right) \tag{16}$$

that is not a (5)-type integration scheme unless

$$\left.\frac{\partial p}{\partial z}\right|_{right,i-1} = \left.\frac{\partial p}{\partial z}\right|_{left,i} \tag{17}$$

i.e. continuity of the first derivative of $p(z)$ is imposed at each step. Eq. (17) explicitly links the values of $\overline{P}_i + \overline{P}_{i+1}$ to the values of p_i, p_{i+1}, p_{i-1} at each time step.

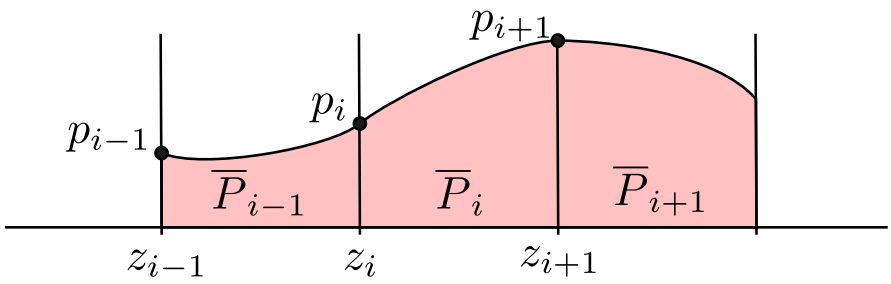


Figure 2: Tsai's integration scheme. Cell averages are evolved explicitly, point values are evolved in a Crank-Nicolson fashion.

To prevent equations from becoming cumbersome, the steps to follow in order to successfully implement the scheme are as follows:

1. Initialise p_i and $\overline{P}_i$
2. Exploit (16) in a Crank-Nicolson fashion together with (17) imposed at the next time step to compute p_i^{n+1} with a tridiagonal system of equations based on $p^n_{i-1,i,i+1}$ and $\overline{P}^n_{i,i-1}$
3. Use (16) to update the values of $\overline{P}_i$ explicitly from $p^{n+1}_{i-1,i,i+1}$, $p^n_{i-1,i,i+1}$

and then iterate step 2 and 3.

This scheme is neither a finite-volume nor a finite-differences scheme, thus it is not straightforward to analytically determine its stability conditions and order of accuracy. However, in [4] it is claimed to be unconditionally stable.

3 Accuracy

In the following, discretised first-order Dirichlet boundary conditions are applied. The problem is considered to be equivalent to the infinite domain one as long as the typical deterministic trajectory is contained in the domain and the (typical) compression in enough to mitigate the diffusion

$$L \gg \sigma^2 \left(\partial_z a\right)^{-1}$$

3.1 1D

For the one-dimensional inviscid case the two schemes are tested upon gaussian wavelets with starting condition

$$p_0(z) = \cos(m\pi\frac{z}{L}) \exp\left(-z^2\right)$$

centered in $z = -cT$ and then evolved at constant speed c for a period of time T. Results (Fig. 3) show that the implicit scheme introduces a k-dependent lag. It is thus sensible to consider reliable the implicit scheme as long as not-too-short scales are involved, i.e. if diffusion dominates over the compression term due to the divergence of the velocity field

$$\sigma^2 \gg \left|\frac{\partial a}{\partial v}\right|$$

In Fig. 4 is shown the stationary solution for the (unphysical) damped noisy harmonic oscillator

$$a(z; x, \omega, \gamma) = -\gamma z - \omega^2 x$$
$$p_\infty(z) \propto \exp\left(-\frac{\gamma}{\sigma^2}\left(z - x\frac{\omega^2}{\gamma}\right)^2\right)$$

Scoring an RMS error of approximately $4 \cdot 10^{-4}$ (Tsai) and $6 \cdot 10^{-4}$ (Implicit).

3.2 2D - stationary

The exact solution of (1) is known only for the damped harmonic oscillator (App. B). The initial value in taken to be the exact solution of the equation at $t = 0.95$, with $\Delta t = \pi/1000$, $\Delta x = \Delta v = 0.1$.

The distribution is evolved for 100 timesteps until the fifth digit of the error w.r.t. the exact solution remains stable. The moments of p are evaluated[1] once the stationary state is (approximately) reached. According to Appendix B they are

$$\begin{aligned} \langle x \rangle = \langle v \rangle = \langle xv \rangle = 0 \\ \omega^2 \langle x^2 \rangle = \langle v^2 \rangle = \frac{\sigma^2}{2\gamma} \end{aligned} \tag{18}$$

and a comparison is shown in 1. For the distance between the distributions $\|e\|_{RMS} = 1.6 \cdot 10^{-4}$ (Implicit) and ??? (Tsai).

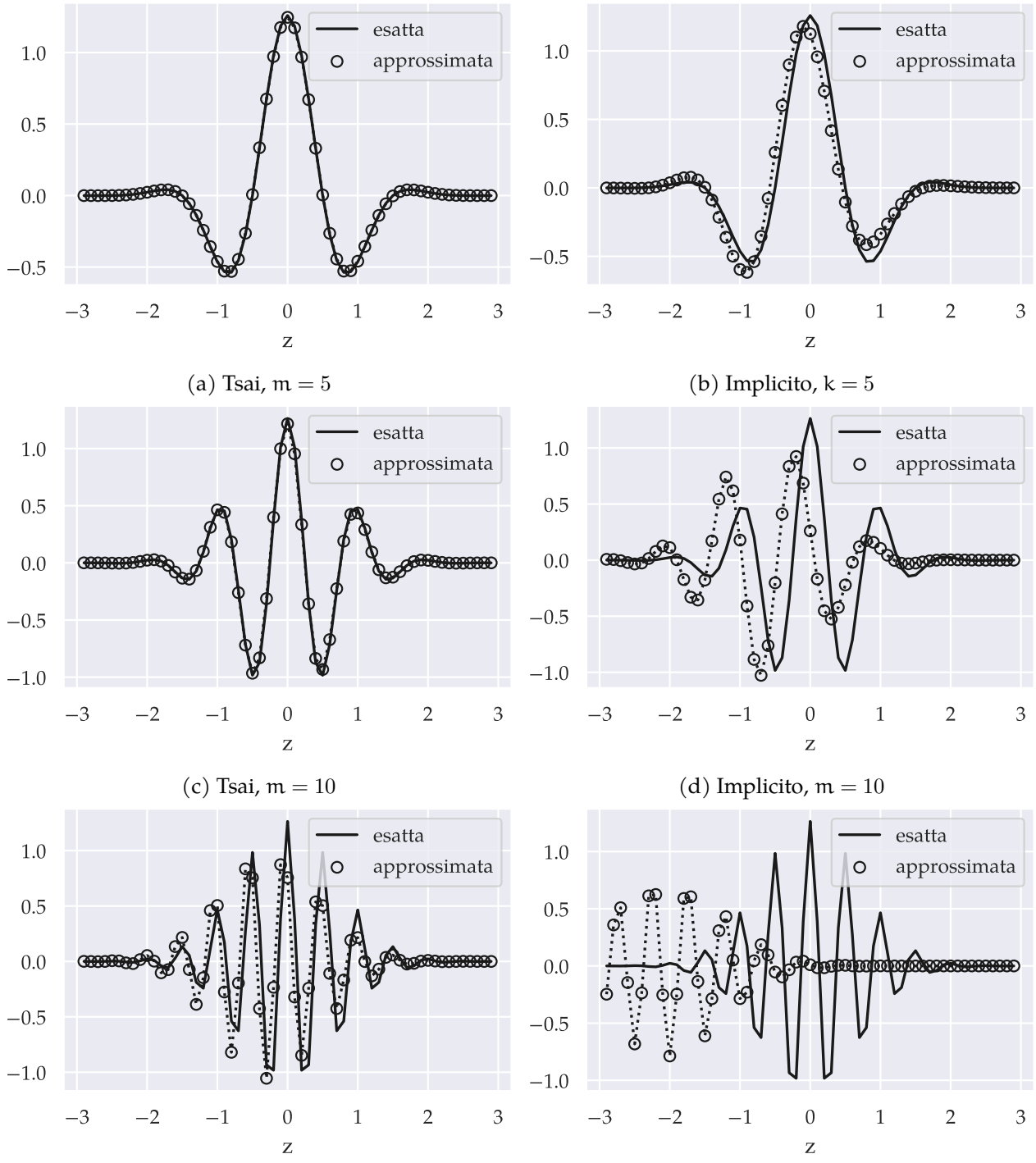
3.3 2D - transient

Results for the mean and variance of x are shown in Fig. 6 and Fig. 7. Contrary to the 1D case, the 2D case shows a lead instead of a lag.

4 Duffing and Van Der Pol

TODO

[1] by means of a trapezoidal integration [5]



(a) Tsai, $m = 5$

(b) Implicito, $k = 5$

(c) Tsai, $m = 10$

(d) Implicito, $m = 10$

(e) Tsai, $m = 20$

(f) Implicito, $m = 20$

Figure 3: Inviscid 1D advection ($\sigma = 0$, $c = const.$) a $T = 1000 \cdot 10^{-3}$ for both schemes tested on gaussian wavelets for $L = 15$ (borders are trimmed).

	Implicit	Tsai	Analytical	Stationary
$E[xx]$	0.15109		0.15238	0.15238
$E[vv]$	0.15010		0.15238	0.15238
$E[xv]$	$3 \cdot 10^{-8}$		$3 \cdot 10^{-8}$	0

Table 1: Confronto tra i momenti secondi di $p(x, v)$ per $\omega = 1$, $\gamma = 2.1$ e $\sigma = 0.8$. Il valore esatto ed il valore teorico (eq. 18) a stazionarietà corrispondono entro la precisione scelta. L'errore della stima numerica è circa lo 0.5%.

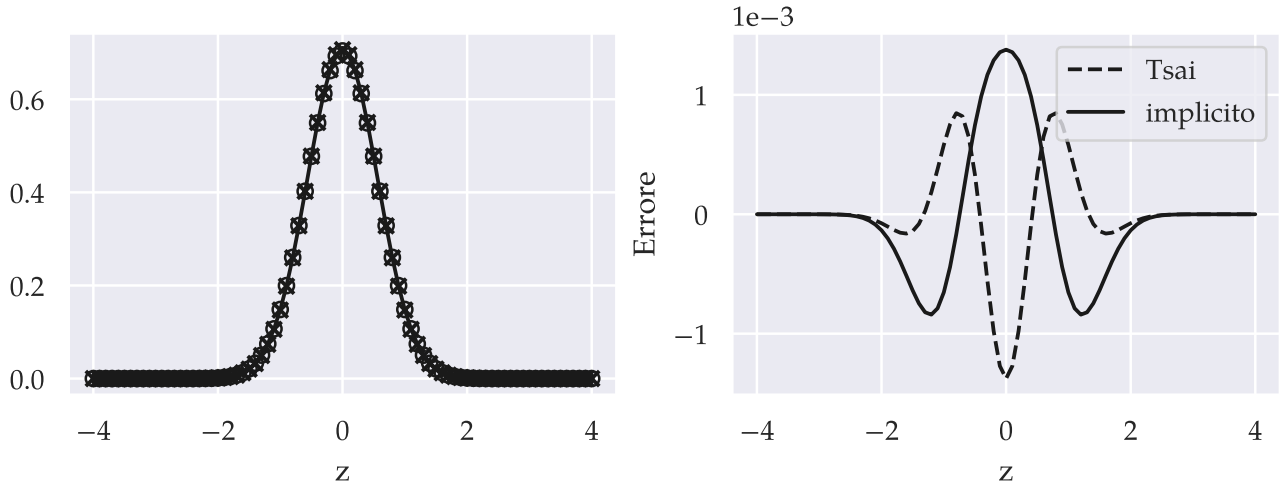


Figure 4: Distribution and error for the stationary state of the 1D damped noisy harmonic osillator.

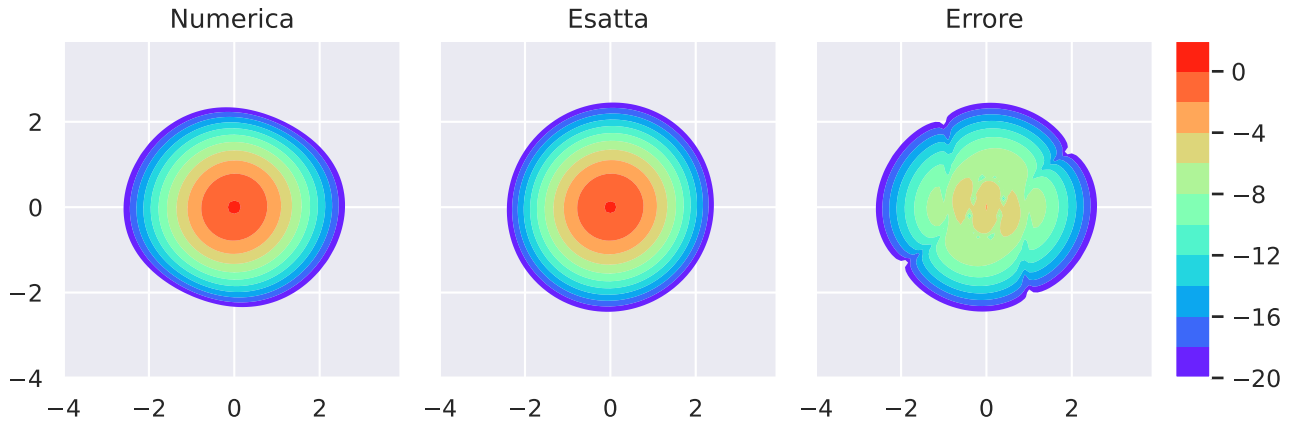


Figure 5: Implicit approximation, exact solution and absolute error for the 2D damped noisy harmonic oscillator after 800 time steps (logarithmic scale)

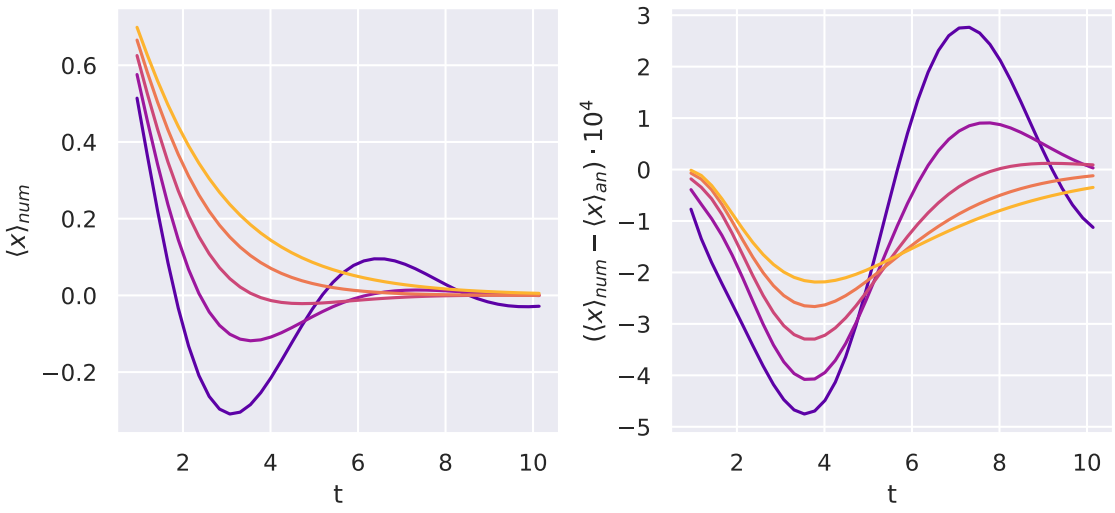


Figure 6: Transient of the expectation value E[x] for the implicit scheme(left) and error w.r.t. analytical solution (right) with $\omega = 1$ for various β. The numerical approximation roughly underestimates the exact result when the latter is decreasing and overestimates it otherwise, displaying an overall lead.

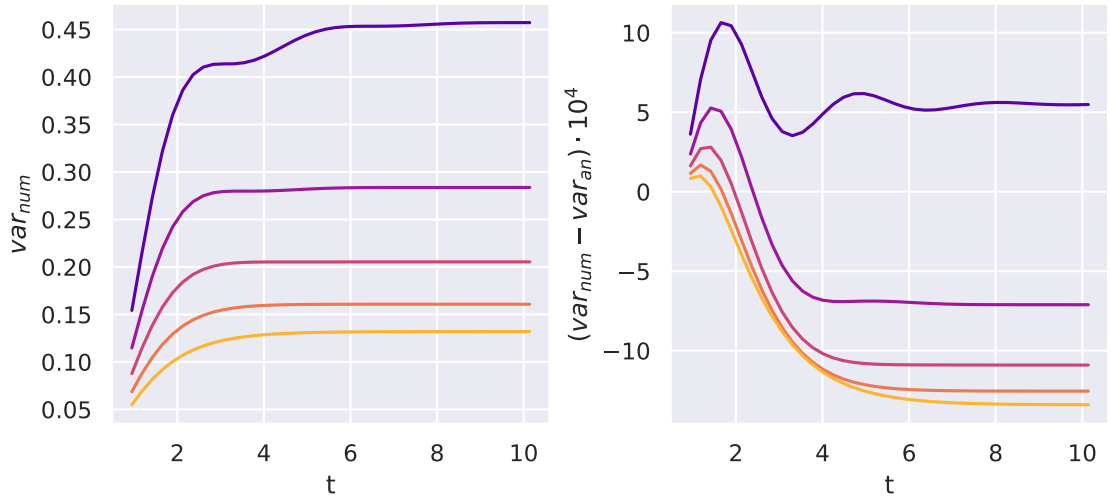


Figure 7: Transient of the expectation value $E[v^2]$ for the 2D damped harmonic oscillator (Implicit).

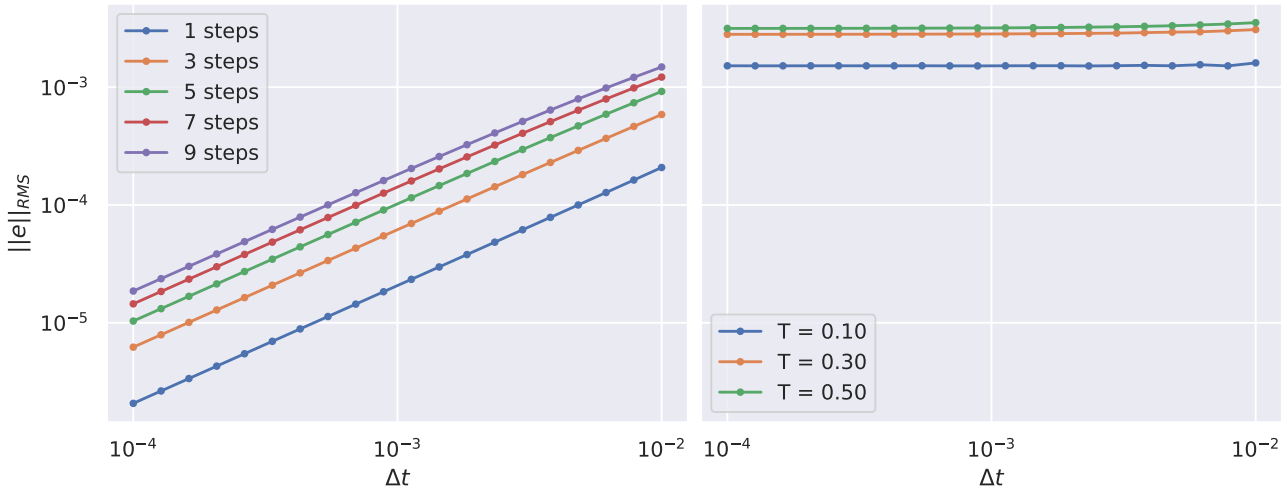


Figure 8: Local truncation error (right) and global error after a time T (right) for the implicit scheme. Contrary to what is stated in [3], the scheme is first-order in time.

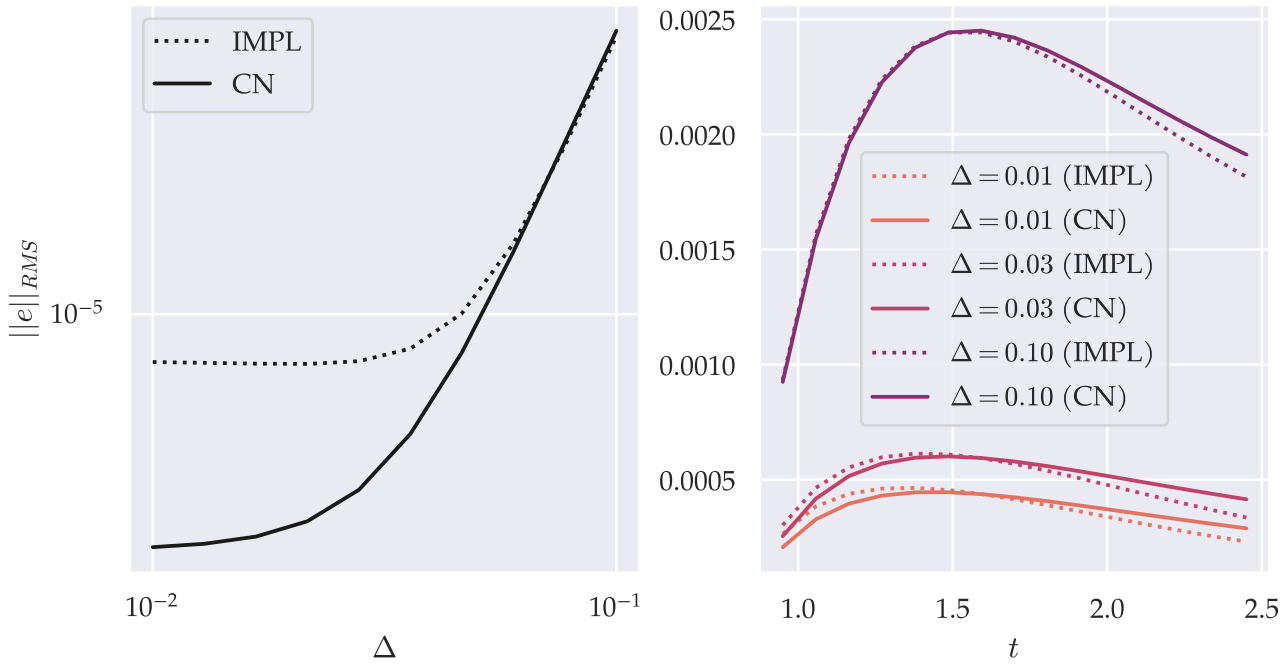


Figure 9: Caption

A Codice e performance (TODO)

Il codice viene implementato in Cython e profilato tramite il pacchetto python `line_profiler`. A sinistra di ogni riga, dove necessario, e' indicato il costo temporale percentuale della singola linea (dove non riportato e' minore del 1%).

Inaspettatamente il tempo di preparazione delle matrici e' superiore al tempo necessario per risolverle: la risoluzione di entrambi i sistemi tridiagonali occupa il 14% del tempo mentre la preparazione della prima matrice il 55%, la seconda il 9% e il riassestamento dei dati l' 8%.

Trascurando gli overhead di inizializzazione, il singolo step di update ha un costo temporale dato da:

$$\tau_{step} \approx \beta N^2 + \gamma N$$

dove β e' una costante dipendente dalla macchina mentre N è il numero di punti su ciascun asse del dominio, supposto quadrato.

Sulla macchina testata (Intel i5-1135G7) nel caso σ costante e $\partial_\nu a$ costante:

$$\beta \approx 0.13\mu s$$
$$\gamma \approx 4\mu s$$

per cui 1000 step di update su un reticolo 80x80 impiegano circa 1.5 secondi, mentre per il metodo usato in [3]

$$\beta \approx 0.2\mu s$$
$$\gamma \approx 4.4\mu s$$

Per quanto riguarda il caso $\sigma = \sigma(x, t)$ e $\partial_\nu a \neq const.$

$$\beta \approx 0.21\mu s$$
$$\gamma \approx 4.6\mu s$$

.

```
    def funker_plank_original( double [:,:] p0,
                          physical_params,
                          integration_params,
                          save_norm = False,
                          save_current=False
                          ):
      ## Time
      cdef double dt   = integration_params['dt']
      cdef unsigned int n_steps = integration_params['n_steps']
      cdef double t0    = physical_params.get('t0', 0.0)

      ## Space
      cdef double Lx, Lv,dx,dv
      Lx, Lv,dx,dv = map(integration_params.get, ["Lx", "Lv", "dx", "dv"])
      cdef unsigned int N = int(Lx/dx), M = int(Lv/dv)
      cdef double [:] x = np.arange(-int(N)//2, int(N)//2)*dx

      cdef double [:] v = np.arange(-int(M)//2, int(M)//2)*dv

      cdef unsigned int time_index = 0, i = 0, j = 0

      cdef double [:,:] p = p0.copy(), p_intermediate = p0.copy()
      cdef double [:] norm = np.zeros(n_steps)

      cdef double theta = 0.5 * dt/dv
      cdef double alpha = 0.5 * dt/dx
      cdef double eta = 0.5*physical_params['sigma_squared']*dt/dv**2

      # Declarations of the diagonals
      cdef double [:] lower_x, diagonal_x, upper_x, b_x
      cdef double [:] lower_v, diagonal_v, upper_v, b_v

      lower_v, upper_v, b_v = np.ones(M), np.ones(M), np.ones(M)
      lower_x, upper_x, b_x = np.ones(N), np.ones(N), np.ones(N)

      # Diagonal of systems does not change
      diagonal_v = np.ones(M)
      diagonal_x = np.ones(N)

      cdef dict currents = dict(top=np.zeros(n_steps),
                                bottom=np.zeros(n_steps),
                                left=np.zeros(n_steps),
                                right=np.zeros(n_steps))

      for time_index in range(n_steps):
        # First evolution: differential wrt V
        # For each value of x, a tridiagonal system is solved to find values of v
        for i in range(N):

          # Prepares tridiagonal matrix and the constant term
          for j in range(M):
[14%]       diagonal_v[j] = 1 + 2*eta + theta*(a(x[i], v[i] + 0.5*dv, time_index*dt,
    physical_params) - a(x[i], v[i] - 0.5*dv, time_index*dt, physical_params))

[ 6%]       upper_v[j]  = - eta
[10%]       upper_v[j] += theta * a(x[i], v[j] + 0.5 * dv, t0 + time_index*dt,
    physical_params)

[ 6%]       lower_v[j] =  - eta
[10%]       lower_v[j] -= theta * a(x[i], v[j] + dv - 0.5 * dv, t0 + time_index*dt,
    physical_params)

[ 6%]       b_v[j] =  p[j, i]


          ## Dirichlet BCs left
```

```
            lower_v[0] = 0.0
            diagonal_v[0] = 1.0
            b_v[0] = 0.0

            # Dirichlet BCs right
            lower_v[N-2] = 0.0
            diagonal_v[N-1] = 1.0
            b_v[N-1] = 0

            # Solves the tridiagonal system for the column
[15%]       p_intermediate[:, i]= tridiag(lower_v, diagonal_v, upper_v, b_v)


         # Second evolution: differential wrt x
         # For each value of v, a tridiagonal system is solved to find values of x
         for j in range(M):

            # Prepares tridiagonal matrix and constant term
            for i in range(N):
[ 5%]         lower_x[i] = - alpha * v[j]
[ 5%]         upper_x[i] =   alpha * v[j]

[ 5%]         b_x[i] = p_intermediate[j, i]

             ## Dirichlet BCs left
             lower_x[0] = 0.0
             diagonal_x[0] = 1.0
             b_x[0] = 0.0

             # Dirichlet BCs right
             lower_x[N-2] = 0.0
             diagonal_x[N-1] =  1.0
             b_x[N-1] = 0.0

            # Solves the tridiagonal system for the row
[12%]       p[j, :] =  tridiag(lower_x, diagonal_x, upper_x, b_x)


       return p, norm, currents
```

B Soluzione analitica dell'eq. di Kramers a coefficienti lineari

Il seguito è tratto da [6], [7] e [8], in cui il problema viene risolto in uno spazio infinito e non in un quadrato.

L'equazione

$$\partial_t P = -v\partial_x P + \partial_v((\omega^2 x + \gamma v)P) + \frac{\sigma^2}{2}\partial_v^2 P$$

diventa per la funzione caratteristica $\phi(\mathbf{k}, t) = \langle \exp i\mathbf{k}\cdot\mathbf{x}\rangle$

$$\partial_t \phi(\mathbf{k}, t) = k_x \partial_{k_v}\phi(\mathbf{k}, t) - \omega^2 k_v \partial_{k_x}\phi(\mathbf{k}, t) - \gamma k_v \partial_{k_v}\phi(\mathbf{k}, t) + \frac{\sigma^2}{2}k_v^2\phi(\mathbf{k}, t)$$

per cui si cerca una soluzione della forma

$$\phi(\mathbf{k}, t) = \exp(-i\mathbf{k}\boldsymbol{\mu}(t))\exp(-\frac{1}{2}\mathbf{k}\boldsymbol{\sigma}(t)\mathbf{k})$$

per cui il vettore e la matrice ausiliari $\boldsymbol{\mu}(t)$ e $\boldsymbol{\sigma}(t)$ rispettano

$$\frac{d\boldsymbol{\mu}}{dt} + \boldsymbol{\Gamma}\boldsymbol{\mu} = 0 \tag{19}$$

$$\frac{d\boldsymbol{\sigma}}{dt} + \boldsymbol{\Gamma}\boldsymbol{\sigma} + (\boldsymbol{\Gamma}\boldsymbol{\sigma})^\dagger = 2\mathbf{D} \tag{20}$$

dove la matrice γ e la matrice $\mathbf{D}$ sono

$$\mathbf{\Gamma} = \begin{pmatrix} 0 & -1 \\ \omega^2 & \gamma \end{pmatrix} \tag{21}$$

$$\mathbf{D} = \begin{pmatrix} 0 & 0 \\ 0 & \sigma^2 \end{pmatrix} \tag{22}$$

Cioe' $\boldsymbol{\mu}$ e $\boldsymbol{\sigma}$ sono e soluzioni dell' equazione di Langevin omogenea associata al sistema e rappresentano i momenti primi e i momenti secondi. In particolare, le eqs. (19) e (20) hanno soluzioni

$$\boldsymbol{\mu}(t) = \mathbf{G}(t)\mathbf{x}_0 \tag{23}$$

$$\boldsymbol{\sigma}(t) = \int_0^t \mathbf{G}(\tau)\mathbf{D}\mathbf{G}^\dagger(\tau)d\tau \tag{24}$$

$$\mathbf{G}(t) = \exp(-\mathbf{\Gamma}t) \tag{25}$$

Infine, applicando la trasformata inversa si ottiene per la probabilità di transizione:

$$P(\mathbf{x}, t|\mathbf{x}_0, t_0 = 0) = \frac{1}{2\pi}|\boldsymbol{\sigma}(t)|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \mathbf{G}(t)\mathbf{x}_0)\boldsymbol{\sigma}^{-1}(t)(\mathbf{x} - \mathbf{G}(t)\mathbf{x}_0)\right] \tag{26}$$

References

[1] N. N. Yanenko. *The Method of Fractional Steps: The Solution of Problems of Mathematical Physics in Several Variables*. Ed. by Maurice Holt. Springer, 1971. ISBN: 9783642651106, 3642651100, 9783642651083, 3642651089.

[2] Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002. ISBN: 978-0521009249.

[3] M.P. Zorzano, H. Mais, and L. Vazquez. "Numerical solution of two dimensional Fokker—Planck equations". In: *Applied Mathematics and Computation* 98.2 (1999), pp. 109–117. ISSN: 0096-3003. DOI: https://doi.org/10.1016/S0096-3003(97)10161-8. URL: https://www.sciencedirect.com/science/article/pii/S0096300397101618.

[4] Tung-Lin Tsai, Jinn-Chuang Yang, and Liang-Hsiung Huang. "An Accurate Integral-Based Scheme for Advection–Diffusion Equation". In: *Communications in Numerical Methods in Engineering* 17 (2001), pp. 701–713. DOI: 10.1002/cnm.442.

[5] Richard L. Burden and J. Douglas Faires. *Numerical Analysis*. 9th. Brooks Cole, 2010, p. 236.

[6] N. G. Van Kampen. *Stochastic Processes in Physics and Chemistry*. North-Holland Personal Library. North-Holland/Elsevier Science, 2007.

[7] Hannes Risken. *The Fokker-Planck Equation: Methods of Solution and Applications*. Springer, 1996.

[8] Sau Fa Kwok. *Langevin And Fokker-Planck Equations And Their Generalizations: Descriptions And Solutions*. World Scientific, 2017.