
TP-5: Enquête sur la dispersion d'un polluant

Il s'agit ici de déterminer l'origine et amplitude d'une source de contamination atmosphérique, un peu à manière dont ça avait été fait en urgence par Janusz Pudykiewicz (Environnement Canada/Dorval) suite à l'accident de Tchernobyl, tel que discuté en classe,... mais en version simplifiée !

Au coeur du TP: une simulation numérique de dispersion d'un polluant (radioactif ou non...) en deux dimensions spatiales. Le code Python exécutant cette simulation vous est fourni pour le TP via la page web du cours, vous n'avez qu'à y ajouter les commandes graphiques et autres sorties de données dont vous aurez besoin pour le TP même.

Le code solutionne numériquement l'équation d'advection-diffusion en 2D pour un contaminant présent en quantités traces, en présence d'une source $S(x, y, t)$ localisée spatialement et variable temporellement (voir §4.2.1 des Notes de cours, au besoin):

$$\frac{\partial C}{\partial t} = -u_x(x, y, t) \frac{\partial C}{\partial x} - u_y(x, y, t) \frac{\partial C}{\partial y} + D \left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} \right) + S(x, y, t), \quad (1)$$

où C est la concentration du polluant. Le coefficient de diffusion D est supposé constant. En analogie avec le cas Tchernobyl, le terme source $S(x, y, t)$ est considéré très localisé à l'échelle spatiale de la simulation.

1 L'écoulement

L'écoulement utilisé est une forme analytique beaucoup moins complexe qu'un écoulement atmosphérique véritable, mais qui en reproduit néanmoins deux des caractéristiques importantes, soit (1) une variabilité temporelle, et (2) de forts cisaillements causant une grande déformation des structures transportées par l'écoulement.

Travaillant en coordonnées Cartésiennes dans un domaine de simulation $x, y \in [0, 2\pi]$, l'écoulement utilisé dans ce qui suit est défini par:

$$u_x(x, y, t) = u_0 A \cos[y + \varepsilon \sin(\omega t + \phi_0)], \quad (2)$$

$$u_y(x, y, t) = u_0 B \sin[x + \varepsilon \cos(\omega t + \phi_0)]. \quad (3)$$

La partie spatiale de ces expressions décrit une série de cellules d'écoulement contigües, où le sens de rotation du fluide (horaire/antihoraire) alterne d'une cellule à la suivante, comme l'illustre la Figure 1. La dépendance temporelle est imposée via mouvement de "précession" des centres des cellules, de fréquence angulaire ω le long d'un cercle de rayon ε . L'ensemble des cellules valse ainsi à l'unison. Dans tout ce qui suit on posera $A = B = \sqrt{6}$, $\varepsilon = 1$, $\phi_0 = 0$, $\omega = 5$, ce qui laisse le paramètre u_0 déterminer l'amplitude globale de l'écoulement.

Le code qui vous est fourni pour ce TP inclut deux fonctions Python ($u_x(x, y, t)$ et $u_y(x, y, t)$) qui calculent les composantes x et y de l'écoulement 2D au temps t , ces valeurs étant données en argument aux deux fonctions.

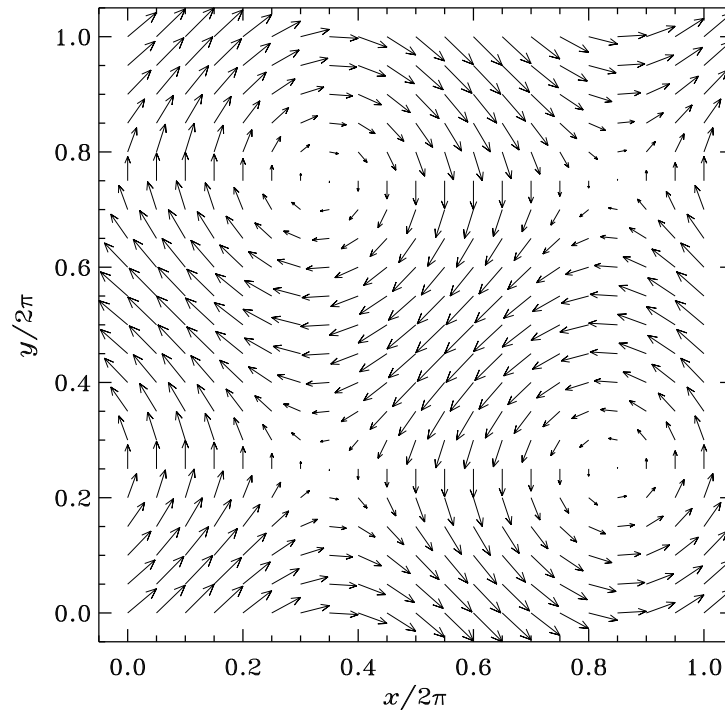


Figure 1: Un instantané des vecteurs vitesses correspondant à l'écoulements décrit par les éqs (2), à $t = 0$. Cet écoulement est composé de cellules contigües de fluide en rotation, le sens de rotation alternant d'une cellule à l'autre. L'écoulement est périodique horizontalement et verticalement. De forts cisaillements sont présents près des points de stagnation (positions x, y où $u_x = u_y = 0$), correspondant ici aux points de contact entre les coins des cellules.

2 Conditions limites

La spécification des conditions limites mérite qu'on s'y attarde. Dans un contexte d'écoulements planétaires à la Tchernobyl, il n'y a pas de frontières: si on attend suffisamment longtemps, un polluant émis en Ukraine et se déplaçant purement vers l'ouest reviendra éventuellement en Ukraine, en provenance de la direction Est.

Ce genre de globalité peut être modélisé sur une maille d'étendue finie en imposant une *condition limite périodique* sur la fonction y étant discrétisée. Autrement dit, ce qui sort par la gauche doit rentrer par la droite, et même chose dans la direction haut/bas¹. On introduit des *couches* de noeuds fantômes en périphérie de la maille. En deux dimensions spatiales, ceci débute avec la définition d'un tableau d'une taille $(N + 2) \times (N + 2)$ pour y représenter la quantité discrétisée sur une maille physique (i.e., couvrant le domaine) de taille $N \times N$. L'algorithme numérique choisi pour l'avancée temporelle, quel qu'il soit, n'est appliqué qu'aux noeuds intérieurs de cette maille agrandie, i.e., les noeuds numérotés (en Python) de 1

¹A strictement parler, une périodicité verticale et horizontale sur une maille cartésienne revient à imposer une topologie de type tore, plutôt que surface sphérique; mais c'est pas trop grave pour la suite...

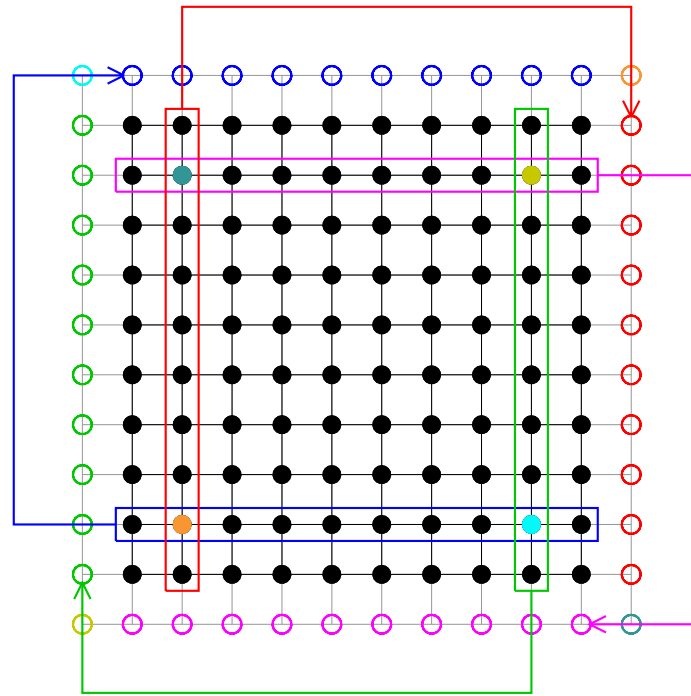


Figure 2: Conditions limites périodiques via les noeuds fantômes. Les valeurs aux points de maille à un intervalle des bords de la maille physique 10×10 (encadrés en couleurs) sont recopiées dans la rangée de noeuds fantômes (cercles vides) du côté opposé du tableau 12×12 , verticalement et horizontalement, tel qu'indiqué par le code couleur. Chaque noeud situé à un intervalle diagonal des coins de la maille physique est recopié au noeud fantôme dans le coin opposé, encore une fois tel qu'indiqué par le code couleur. Ce processus, qui doit être répété à chaque pas de temps, ne change pas les valeurs aux noeuds de la maille originelle 10×10 , dont l'évolution dans le temps est calculée via l'algorithme Leapfrog appliqué aux $c[1:N, 1:N]$ éléments intérieurs du tableau de taille $(N+2) \times (N+2)$.

à N , qui correspondent aux noeuds “physiques”. Une fois l'avancée temporelle effectuée sur les noeuds internes, les noeuds fantômes en périphérie sont mis à jour en recopiant les valeurs provenant du groupe de noeuds du côté opposé de la maille physique. Ici une image vaut vraiment mille mots, donc étudiez bien la Figure 2 ci-dessous.

Le code qui vous est fourni pour ce TP inclut une fonction Python (`periodic`) effectuant le processus de copie illustré sur la Figure 2. Notez que cette fonction doit être invoquée à chaque pas temps à l'intérieur de la boucle contrôlant l'itération temporelle.

Opérer de cette façon a l'avantage additionnel que les algorithmes d'avancée temporelle peuvent être appliqués tel quel aux noeuds $[1:N, 1:N]$ du tableau (ici le tableau 2D c) contenant les valeurs discrétisées de la quantité modélisée, sans avoir à se préoccuper du danger de débordement de tableaux.

3 La source de polluant

On utilisera un terme source dont le profil spatial a la forme d'une Gaussienne de demie-largeur w centrée sur une position (x_0, y_0) à l'intérieur du domaine:

$$S(x, y, t) = S_0 f(t) \exp(-r^2/w^2), \quad r^2 = (x - x_0)^2 + (y - y_0)^2. \quad (4)$$

La fonction $f(t)$ contrôle la dépendance temporelle de l'émission, et S_0 son taux instantané. Dans les simulations qui suivent on utilisera une simple fonction carrée, i.e.:

$$f(t) = 1 \text{ si } t_d \leq t \leq t_f, \quad 0 \text{ sinon}, \quad (5)$$

les valeurs t_d et t_f correspondant donc au commencement et fin de l'incident, respectivement. La simulation présentée comme exemple ci-dessous utilise les valeurs numériques $S_0 = 2$, $(x_0, y_0)/2\pi = (0.8, 0.6)$, $w = 0.3$, $t_d = 0.2$, et $t_f = 1.75$.

Le code qui vous est fourni pour ce TP inclut deux fonctions Python, `sourcexy(x0, y0, s)` et `sourcet(t)`, qui calculent les parties spatiales et temporelle de la source, la première pouvant être calculée une seule fois et emmagasinée dans un tableau 2D $s[N, N]$ avant le début de la boucle temporelle, pour sauver du temps de calcul.

4 Quelques résultats représentatifs

Le code Python qui vous est fourni solutionne la forme adimensionnelle de l'équation d'advection-diffusion, avec $L = 1$ comme échelle de longueur, et le temps advectif L/u_0 comme unité de temps. La forme adimensionnelle de l'éq. (1) devient:

$$\frac{\partial C}{\partial t} = -u_x(x, y, t) \frac{\partial C}{\partial x} - u_y(x, y, t) \frac{\partial C}{\partial y} + \frac{1}{\text{Pe}} \left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} \right) + S^*(x, y, t), \quad (6)$$

où ici le nombre de Peclet est défini comme $\text{Pe} = u_0/D$ (attention, on a posé $L = 1$!), l'astérisque sur le terme source indique que ce dernier est maintenant mesuré en unités du temps d'advection, et le coefficient u_0 a maintenant disparu des éqs. (2).

Considérons maintenant une solution représentative obtenue avec ce modèle, à l'aide de la version 2D de la méthode Leapfrog avec pas spatiaux et temporel $\Delta x = \Delta y = 0.196$ (maille 65×65) et $\Delta t = 0.0025$. Les paramètres de l'écoulement et de la fonction source sont fixés aux valeurs citées précédemment, et la valeur du nombre de Peclet est fixée à $\text{Pe} = 300$, impliquant que l'advection domine sur la diffusion à l'échelle globale de la simulation.

La Figure 3 est l'équivalent conceptuel de la Fig. 6.25 des Notes de cours, et montre quatre instantanés de la distribution spatiale de la concentration du contaminant. La position de la source est indiquée par le viseur rouge. On remarque immédiatement comment le nuage de polluant est déformé et étiré par l'écoulement, formant éventuellement une structure très allongée et de forme très complexe (qualitativement comme sur la Fig. 6.25). Bien que le nuage soit initialement assez diffus, l'action de l'écoulement en vient à produire des gradients de concentration très abrupts en sa périphérie.

Un examen attentif de la solution à $t = 2$ révèle de faibles oscillations, devenues clairement visibles à $t = 3$. C'est une conséquence de l'utilisation de Leapfrog sur une maille spatiale et temporelle relativement

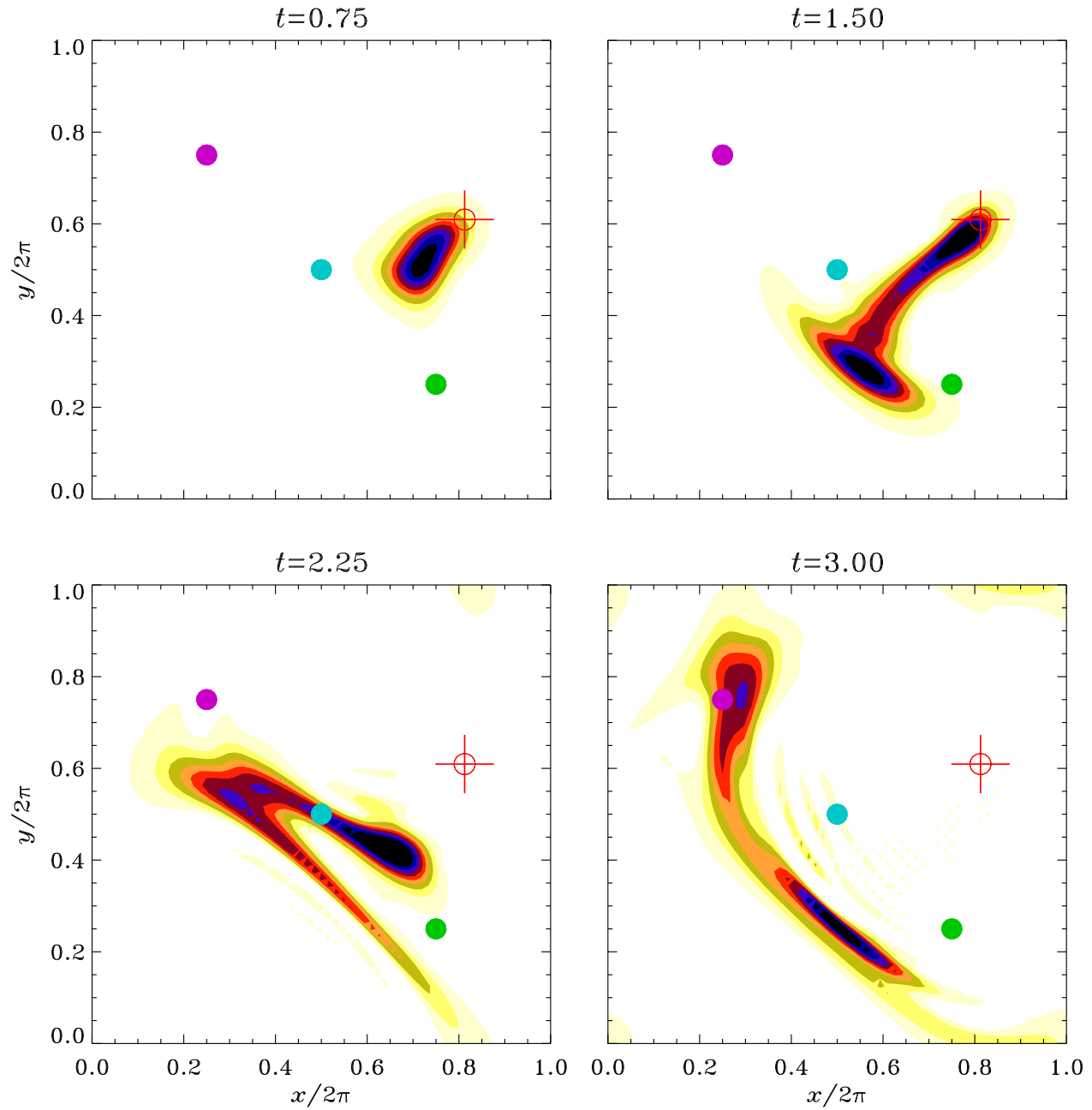


Figure 3: Quatre instantanés de la distribution spatiale d'un polluant dans une simulation à $Pe = 300$. La position de la source de polluant est indiquée par le viseur rouge, et les trois autres points colorés indique les positions d'où sont extraites les séquences temporelles portées en graphique sur la Fig. 4 qui suit. L'injection de polluant se fait à un taux ($S_0 = 2$ dans l'éq. (4)) constant dans l'intervalle $0.01 \leq t \leq 1.75$. Les isocontours de concentration ont un espacement de $\Delta C = 0.05$.

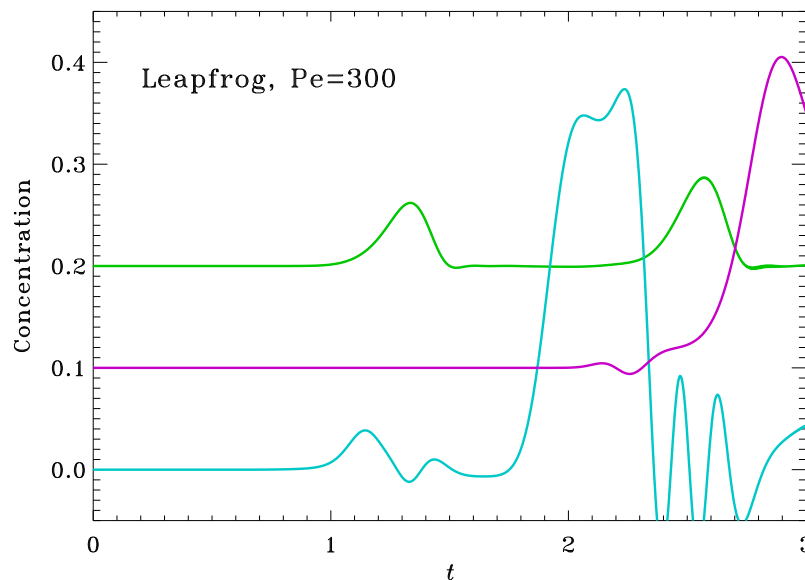


Figure 4: Séquences temporelles de la concentration de polluant, mesurées aux positions indiquées par les points mauve, bleu et vert sur la Figure 3, soit $(x,y)/2\pi = (0.25, 0.75)$, $(0.5, 0.5)$, et $(0.75, 0.25)$. Les courbes mauves et vertes sont décalées artificiellement vers le haut; tout le monde commence avec $C = 0$ à $t = 0$!

grossière. Ces oscillations d'origine purement numériques demeurent de faible amplitude pour $t \simeq 3$, et ne devraient donc pas affecter significativement le déroulement du TP. Ceci demeure cependant un exemple d'un danger très courant dans ce genre de problème, soit distinguer les structures physiques de celles produites artificiellement par l'algorithme numérique utilisé.

La Figure 4 montre trois séquences temporelles de la concentration de polluant, mesurées aux positions indiquées par les points mauve, bleu et vert sur la Figure 3. On notera déjà que ces séquences temporelles sont très différentes l'une de l'autre, même si deux d'entre elles (vert et bleu) proviennent de positions situées à distance presque égales de la source; ceci reflète à la fois la complexité de l'écoulement et la dominance de l'advection sur la diffusion au membre de droite de l'éq. (6).

5 Le TP même...

Dans un contexte environnemental du style Tchernobyl, une question très intéressante est la suivante: connaissant les variations spatiotemporelles de l'écoulement (par observations météo) mais ayant seulement accès aux trois séquences temporelles de la Figure 4 (données provenant de stations de contrôle de la qualité de l'air), est-il possible de déduire la position de la source, et la quantité de polluant libéré ? C'est là le défi auquel vous aurez à faire face dans le cadre de ce TP !

Votre première tâche sera de reproduire l'une ou l'autre (ou les deux) des Figures 3 ou 4, en utilisant les mêmes valeurs de paramètres pour l'écoulement, la fonction source, et le nombre de Peclet. Ensuite, utilisez vos simulations pour déduire la position et l'amplitude du terme source ayant conduit à

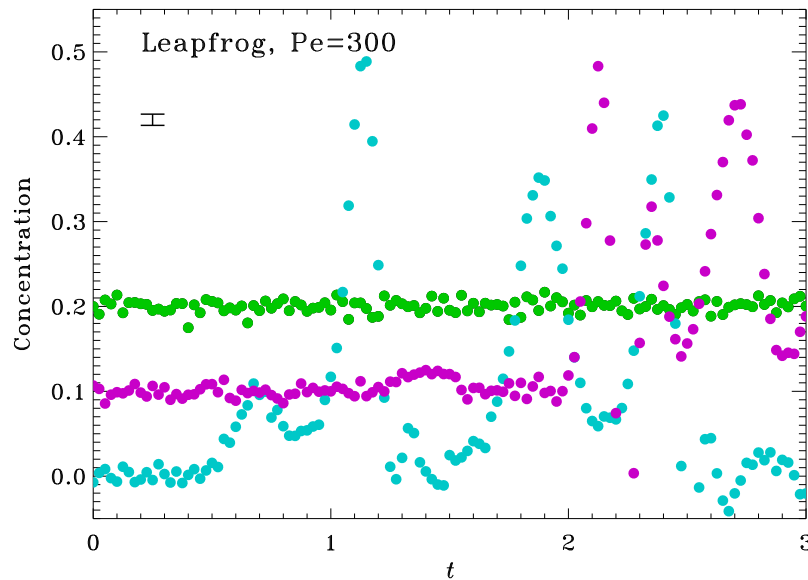


Figure 5: Identique à la Figure 4, sauf que cette fois les paramètres de la fonction source ont changé et les séquences sont échantillonnées à une cadence temporelle de 0.025 unités de temps, comme si il s’agissaient de véritables observations. Ces données, disponibles sur la page web du cours, sont celles avec lesquelles vous devrez travailler dans le cadre du TP. Les positions d’échantillonnage demeurent les mêmes qu’indiquées par les points mauve, bleu et vert sur la Figure 3, soit $(x,y)/2\pi = (0.25,0.75)$, $(0.5,0.5)$, et $(0.75,0.25)$, et un faible bruit aléatoire a été ajouté aux données.

un “incident” dont les seules mesures disponibles sont les séquences temporelles présentées à la Figure 5 ci-dessous, et accessibles sous forme numérique sur la page web du cours. Les positions des points d’échantillonnage demeurent inchangées. Un bruit aléatoire a été ajouté aux données pour simuler les erreurs de mesure. Ce bruit est décrit par une gaussienne centrée à zéro et de demie-largeur à mi-hauteur $\sigma = 0.066$.

Heureusement, les espions du gouvernement qui vous emploie, présents sur le terrain, ont fourni quelques informations utiles:

1. L’incident a débuté à $t = 0.15$ et s’est terminé à $t = 2$;
2. Le taux de libération de polluant (soit le paramètre S_0 dans l’éq. 4)) est demeuré constant tout au long de l’incident.

Vous devez donc déterminer les valeurs du triplet de paramètres (x_0, y_0, S_0) dans l’éq. (4) qui produisent des séquences temporelles aux trois points d’échantillonnage qui ressemblent le plus possible à celles fournies. Pour quantifier cette différence, le plus simple est de calculer le χ^2 normalisé entre les séquences temporelles fournies (C_k^* , disons), et celles produites par votre modèle aux K pas de temps

correspondant, $C(t_k)$:

$$\chi^2 = \frac{1}{K-3} \sum_{k=0}^{K-1} \frac{(C_k^* - C(t_k))^2}{\sigma^2}, \quad (7)$$

et minimiser la somme de cette quantité pour les trois séquences temporelles. Dans cette expression $\sigma = 0.0066$ est la déviation standard, soit la moitié de la barre d'erreur indiquée sur la Fig. 5. Le “3” au dénominateur devant la sommation au membre de droite reflète le fait que votre ajustement compte 3 degrés de liberté, soit les 3 paramètres ajustable du modèle. Normalisé de cette façon, une valeur $\chi^2 \simeq 1$ indique un bon fit, et il n'est pas utile ou nécessaire d'essayer de le pousser plus bas; ce serait alors “fitter le bruit”.

Truc: Si vous ne changez pas le pas de temps dans le code de simulation (et vous ne devriez pas!), alors les données observées peuvent directement être extraites à une cadence de 10 pas de temps.

6 Bibliographie

Pour en savoir plus sur l'algorithme Leapfrog (et plusieurs autres plus précis mais plus gourmands en mémoire et/ou temps de calcul), voir l'excellente présentation de ces sujets se trouvant à la §19.1 de:

Press, W.H., Teukolsy, S.A., Vetterling, W.T., & Flannery, B.P., *Numerical Recipes*, seconde éd., Cambridge University Press (1992);