

Temps en secondes	NeuroCorr (neuro-stat). Total (Gns)	Version V2 (sans $k_1, k_2$ ). ma-trice G	Version V2 (avec $k_1, k_2$ ). matrice Gk	Version PWC. ma-trice Gpw	Erreur $\ G - Gk\ _\infty$	Erreur $\ G - Gpw\ _\infty$	Erreur $\ G - Gns\ _\infty$	Erreur $\ Gpw - Gns\ _\infty$
M=10 ( $\simeq 50$ spikes/neurone)	0.02	0.00	0.01	0.03	1e-15	4e-15	3e-15	2e-15
M=50 ( $\simeq 50$ spikes/neurone)	0.43	0.06	0.09	0.35	1.1e-15	4e-15	3.1e-15	3.1e-15
M=100 ( $\simeq 50$ spikes/neurone)	1.33	0.19	0.25	1.12	1.1e-15	4e-15	4e-15	3.1e-15
M=200 ( $\simeq 50$ spikes/neurone)	5.44	0.61	1.00	4.45	1.1e-15	1e-14	1e-14	4e-15
M=500 ( $\simeq 50$ spikes/neurone)	33.81	5.62	7.88	27.91	1.1e-15	1e-14	1e-14	9.8e-15
M=1000 ( $\simeq 50$ spikes/neurone)	135.27	29.24	37.17	111.76	1.1e-15	1e-14	1e-14	9.8e-15
M=100 ( $\simeq 300$ spikes/neurone)	3.04	0.74	1.25	6.91	0	3.1e-13	4.2e-13	3.8e-13
M=1000 ( $\simeq 300$ spikes/neurone)	325.88	141.06	167.17	628.26	0	3.3e-13	4.6e-13	5e-13

- $\delta = 0.02$ ,  $K = 5$  (donc  $A = 0.1$ )
- $]T_{\min}, T^{\max}] = ]0, 8]$
- Le vecteur low n'a pas été pris en compte dans le temps calcul de G
- Ici la dernière version V2 est faite en C++ avec armadillo. L'interfaçage avec R (RcppArmadillo) semble donner des temps comparables à C++ armadillo.
- Les mesures ont été faites sur math11 - elles sont relatives car d'autres utilisateurs pouvaient l'utiliser en même temps
- Le temps calcul pour neuro-stat regroupe les temps calcul de  $G$ ,  $\mu_1$ ,  $\mu_2$  et  $\mu_A$
- La version PWC est ma version C++ armadillo avec les fonctions constantes par morceaux