

## Optimisation MODLIN

---

Idée : utiliser la structure de famille et l'indépendance du calcul des vraisemblances des sous familles pour optimiser la calcul du gradient utilisé par les optimisateurs (quasi newton, gradient conjugués...). Exemple : lors d'une variation infinitésimale d'une variance d'une familles d'un père  $ip$  , les autres famille de père garde la même valeur de vraisemblance.

Fonctionnement : Actuellement, on écrit une fonction de vraisemblance prenant comme paramètres :

N : le nombre de paramètre a optimiser

X : les valeurs des paramètres à tester

F : le résultat de la vraisemblance globale

sur 1 exemple de 2 familles de père accouplé avec 3 meres (2 avec le pere 1 et 1 avec le pere 2), nous avons comme interprétation de X :

Sig 1	Sig 2	MuGen	PolyPere1	PolyPere2	PolyMere1	PolyMere2	PolyMere3
-------	-------	-------	-----------	-----------	-----------	-----------	-----------

Les 4 sous familles sont influencés par les paramètres grisés (et la variation des autres paramètres n'influence pas le résultat de la vraisemblance de la famille) :

$ip=1, jm=1$

Sig 1	Sig 2	MuGen	PolyPere1	PolyPere2	PolyMere1	PolyMere2	PolyMere3
-------	-------	-------	-----------	-----------	-----------	-----------	-----------

$ip=1, jm=2$

Sig 1	Sig 2	MuGen	PolyPere1	PolyPere2	PolyMere1	PolyMere2	PolyMere3
-------	-------	-------	-----------	-----------	-----------	-----------	-----------

$ip=2, jm=3$

Sig 1	Sig 2	MuGen	PolyPere1	PolyPere2	PolyMere1	PolyMere2	PolyMere3
-------	-------	-------	-----------	-----------	-----------	-----------	-----------

L'idée est donc de donner à l'optimisateur :

- Un filtre de booléen (taille N) pour chaque famille IP,JM indiquant les paramètres de X influençant la vraisemblance de la famille IP-JM
- Une fonction de VRAISSEMBLANCE d'une sous famille à optimiser, générique à l'ensemble des sous famille (FUNCT(IP,JM,N,X,F)).

Note :

Cette optimisation ne peut s'appliquer aux méthodes pour les données pré-calculés puisque le calcul de vraisemblance de la dernière mère à estimer est en fonction des autres mères estimables.

Calcul du gradient

---

Le calcul du gradient optimisé s'écrit donc:

```

XP=X
POUR I=1, N
  XP(I)=X(I)-H
  FMOINH=0
  POUR IP=1, NP
    POUR JM=1, NM
      SI ( FILTRE(IP, JM, I) ) ALORS
        FMOINH=FMOINH+FM(IP, JM) // ON RECUPERE LA DERNIERE VALEUR CALCULE
      SINON
        APPEL FUNCT(IP, JM, N, X, FTEMP)
        FMOINH=FMOINH+FTEMP
      FIN SI
    FIN POUR(JM)
  FIN POUR (IP)

  XP(I)=X(I)+H
  FPLUSH=0
  POUR IP=1, NP
    POUR JM=1, NM
      SI ( FILTRE(IP, JM, I) ) ALORS
        FPLUSH=FPLUSH+FM(IP, JM) // ON RECUPERE LA DERNIERE VALEUR CALCULE
      SINON
        APPEL FUNCT(IP, JM, N, X, FTEMP)
        FPLUSH=FPLUSH+FTEMP
      FIN SI
    FIN POUR(JM)
  FIN POUR (IP)
  //CALCUL DU GRADIENT POUR LE PARAMETRE I
  G(I) = (FPLUSH - FMOINH) / 2*H
  XP=X
FIN POUR (I)

```

### Construction du filtre

La construction du filtre se fait à partir de la matrice d'incidence réduite (l'estimabilité des paramètres via cholesky nous donne la matrice d'incidence effective utilisée lors de l'optimisation). Elle est automatique et ne nécessite pas a priori (pas besoin de connaître les niveaux à estimer) : Pour chaque colonne de cette matrice (correspondant au ième paramètre de X), on teste si l'ensemble des valeurs est égale à zéro. Si c'est le cas, le ième paramètre de X n'influence pas la vraisemblance de la sous famille ip-jm.

```

filter=.true.
filter(:, :, 1:np+1)=.false.
do ip=1, np
  filter_inc(ip, nmp(ip)+1:nmp(ip+1), ip)=.true.
  do jm=nmp(ip)+1, nmp(ip+1)
    kd1=kd2+1
    kd2=kd2+count(presentc(ic, ndm(jm)+1:ndm(jm+1)))
    filter(ip, jm, np+1:nbnivest)=any(XINCREDUIT(kd1:kd2, :nbnivest)/=0.d0
, dim=1)
  end do
end do

```

## Benchmark

---

Compilation en mode optimisé O4 avec gfortran

3 exemples sont donnés : marie damon,nicolas deschamp et exemple porc:

Chaque exemple présente :

- la taille du jeu de donnée : np,nm,effet de nuisance => nombre de niveau a estime
- 2 prints d'exécution non optimisé et optimisé , chaque ligne correspond à :
  - temps de l'optimisation
  - nombre d'appel effectif du calcul de la vraisemblance total
  - nombre d'appel au calcul de gradient
  - nombre d'appel évité du a la recuperation de la vraisemblance d'une famille
  - le taux d'appel optimisé (nb appel évité/ (nb appel + nb appel evite) )
- Le filtre initialisé sous H0 en guise d exemple (ce filtre étant recalculé à chaque position par la suite)

Voici un tableau récapitulatifs des résultats.

<b>Exemple</b>	<b>Effets de nuisances</b>	<b>Nb niveaux(H1)</b>	<b>Temps cumul non optim</b>	<b>Temps cumul optim</b>	<b>Taux d'optimisation</b>
Marie	0	53	3,34	0,40	91,70%
Nicolas	5	61	8,54	3,77	58,70%
Exemple porc	0	16	0.240	0,07	76,50%



Exemple NICOLAS DESCHAMP

-----  
 5 : NP  
 5 : NM  
 5 : Effets de nuisances  
 61 : NTNIV

\* pas optimisé \*

-----  
 TEMPS UNIT | LRT | NB APPEL FUNCT Famille IP,JM | NB APPEL GRADIENT | APPEL EVITE | NB APPEL EVITE/ APPEL TOTAL | TEMPS CUMULES  
 -----

0.50999928	208.862	call funct:	11375	call gradient:	125	avoid call:	0	0.0%	0.510
0.92000103	204.470	call funct:	13875	call gradient:	125	avoid call:	0	0.0%	1.430
0.10000038	204.348	call funct:	1554	call gradient:	14	avoid call:	0	0.0%	1.530
0.09000015	204.215	call funct:	1332	call gradient:	12	avoid call:	0	0.0%	1.620
0.10000038	204.069	call funct:	1554	call gradient:	14	avoid call:	0	0.0%	1.720
0.10000038	203.910	call funct:	1554	call gradient:	14	avoid call:	0	0.0%	1.820
0.09000015	203.739	call funct:	1443	call gradient:	13	avoid call:	0	0.0%	1.910
0.10000038	203.555	call funct:	1554	call gradient:	14	avoid call:	0	0.0%	2.010
0.09000015	203.361	call funct:	1443	call gradient:	13	avoid call:	0	0.0%	2.100
0.12999916	203.155	call funct:	1998	call gradient:	18	avoid call:	0	0.0%	2.230
0.09000015	202.941	call funct:	1443	call gradient:	13	avoid call:	0	0.0%	2.320
0.13000107	202.719	call funct:	1998	call gradient:	18	avoid call:	0	0.0%	2.450
0.08999825	202.491	call funct:	1443	call gradient:	13	avoid call:	0	0.0%	2.540
0.09000015	202.260	call funct:	1443	call gradient:	13	avoid call:	0	0.0%	2.630
0.12000084	202.028	call funct:	1887	call gradient:	17	avoid call:	0	0.0%	2.750
0.10000038	201.798	call funct:	1554	call gradient:	14	avoid call:	0	0.0%	2.850
0.12000084	201.571	call funct:	1887	call gradient:	17	avoid call:	0	0.0%	2.970
0.07999992	201.351	call funct:	1221	call gradient:	11	avoid call:	0	0.0%	3.050
0.09000015	201.141	call funct:	1554	call gradient:	14	avoid call:	0	0.0%	3.140
0.09000015	200.941	call funct:	1443	call gradient:	13	avoid call:	0	0.0%	3.230
0.34000015	200.749	call funct:	5217	call gradient:	47	avoid call:	0	0.0%	3.570
0.09000015	200.576	call funct:	1443	call gradient:	13	avoid call:	0	0.0%	3.660
0.10000038	200.418	call funct:	1554	call gradient:	14	avoid call:	0	0.0%	3.760
0.11999893	200.277	call funct:	1776	call gradient:	16	avoid call:	0	0.0%	3.880
0.07000160	200.152	call funct:	1110	call gradient:	10	avoid call:	0	0.0%	3.950
0.10000038	200.043	call funct:	1554	call gradient:	14	avoid call:	0	0.0%	4.050
0.12000084	199.936	call funct:	1887	call gradient:	17	avoid call:	0	0.0%	4.170
0.09000015	200.079	call funct:	1332	call gradient:	12	avoid call:	0	0.0%	4.260
0.10999870	200.292	call funct:	1665	call gradient:	15	avoid call:	0	0.0%	4.370
0.11000061	200.570	call funct:	1665	call gradient:	15	avoid call:	0	0.0%	4.480
0.14999962	200.964	call funct:	2109	call gradient:	19	avoid call:	0	0.0%	4.630
0.07999992	201.033	call funct:	1110	call gradient:	10	avoid call:	0	0.0%	4.710
0.05999947	201.115	call funct:	777	call gradient:	7	avoid call:	0	0.0%	4.770
0.06999969	201.211	call funct:	999	call gradient:	9	avoid call:	0	0.0%	4.840
0.09000015	201.320	call funct:	1443	call gradient:	13	avoid call:	0	0.0%	4.930
0.07999992	201.444	call funct:	1110	call gradient:	10	avoid call:	0	0.0%	5.010
0.05999947	201.583	call funct:	888	call gradient:	8	avoid call:	0	0.0%	5.070
0.07999992	201.736	call funct:	1221	call gradient:	11	avoid call:	0	0.0%	5.150
0.12000084	201.901	call funct:	1776	call gradient:	16	avoid call:	0	0.0%	5.270
0.07999992	202.077	call funct:	1110	call gradient:	10	avoid call:	0	0.0%	5.350
0.12999916	202.260	call funct:	1887	call gradient:	17	avoid call:	0	0.0%	5.480
0.22000122	202.445	call funct:	3330	call gradient:	30	avoid call:	0	0.0%	5.700
0.10000038	202.631	call funct:	1443	call gradient:	13	avoid call:	0	0.0%	5.800
0.11000061	202.811	call funct:	1554	call gradient:	14	avoid call:	0	0.0%	5.910
0.09000015	202.983	call funct:	1332	call gradient:	12	avoid call:	0	0.0%	6.000
0.11999893	203.142	call funct:	1776	call gradient:	16	avoid call:	0	0.0%	6.120
0.22999954	203.916	call funct:	3552	call gradient:	32	avoid call:	0	0.0%	6.350
0.10000038	204.053	call funct:	1554	call gradient:	14	avoid call:	0	0.0%	6.450
0.12999916	204.170	call funct:	1998	call gradient:	18	avoid call:	0	0.0%	6.580
0.07999992	204.268	call funct:	1110	call gradient:	10	avoid call:	0	0.0%	6.660
0.11000061	204.347	call funct:	1665	call gradient:	15	avoid call:	0	0.0%	6.770
0.11000061	204.410	call funct:	1554	call gradient:	14	avoid call:	0	0.0%	6.880
0.15999985	204.457	call funct:	2331	call gradient:	21	avoid call:	0	0.0%	7.040
0.09000015	204.491	call funct:	1332	call gradient:	12	avoid call:	0	0.0%	7.130
0.07999992	204.515	call funct:	1221	call gradient:	11	avoid call:	0	0.0%	7.210
0.12000084	204.529	call funct:	1776	call gradient:	16	avoid call:	0	0.0%	7.330
0.09000015	204.536	call funct:	1332	call gradient:	12	avoid call:	0	0.0%	7.420
0.10999870	204.536	call funct:	1554	call gradient:	14	avoid call:	0	0.0%	7.530
0.09000015	204.532	call funct:	1332	call gradient:	12	avoid call:	0	0.0%	7.620
0.07999992	204.523	call funct:	1221	call gradient:	11	avoid call:	0	0.0%	7.700
0.13999939	204.105	call funct:	2220	call gradient:	20	avoid call:	0	0.0%	7.840
0.07999992	204.105	call funct:	1221	call gradient:	11	avoid call:	0	0.0%	7.920
0.09000015	204.104	call funct:	1332	call gradient:	12	avoid call:	0	0.0%	8.010
0.06999969	204.102	call funct:	1110	call gradient:	10	avoid call:	0	0.0%	8.080
0.06999969	204.099	call funct:	1110	call gradient:	10	avoid call:	0	0.0%	8.150
0.06999969	204.096	call funct:	1110	call gradient:	10	avoid call:	0	0.0%	8.220
0.09000015	204.093	call funct:	1332	call gradient:	12	avoid call:	0	0.0%	8.310
0.06999969	204.089	call funct:	1110	call gradient:	10	avoid call:	0	0.0%	8.380
0.07999992	204.085	call funct:	1221	call gradient:	11	avoid call:	0	0.0%	8.460
0.07999992	204.080	call funct:	1221	call gradient:	11	avoid call:	0	0.0%	8.540
0.19000053	249.937	call funct:	2990	call gradient:	46	avoid call:	0	0.0%	8.730
0.35999870	262.708	call funct:	5642	call gradient:	62	avoid call:	0	0.0%	9.090
0.87000084	204.835	call funct:	14980	call gradient:	140	avoid call:	0	0.0%	9.960

\* Gradient Optimisé \*

TEMPS UNIT | LRT | NB APPEL FUNCT Famille IP,JM | NB APPEL GRADIENT | APPEL EVITE | NB APPEL EVITE/ APPEL TOTAL | TEMPS CUMULES

0.23999977	208.862	call funct:	5225	call gradient:	125	avoid call:	6150	54.1%	0.240
0.43999958	204.470	call funct:	6366	call gradient:	139	avoid call:	9062	58.7%	0.680
0.03999996	204.348	call funct:	641	call gradient:	14	avoid call:	912	58.7%	0.720
0.03999996	204.215	call funct:	641	call gradient:	14	avoid call:	912	58.7%	0.760
0.03999996	204.069	call funct:	641	call gradient:	14	avoid call:	912	58.7%	0.800
0.05000019	203.910	call funct:	641	call gradient:	14	avoid call:	912	58.7%	0.850
0.05999947	203.739	call funct:	824	call gradient:	18	avoid call:	1173	58.7%	0.910
0.03000069	203.556	call funct:	503	call gradient:	11	avoid call:	717	58.7%	0.940
0.05999947	203.361	call funct:	870	call gradient:	19	avoid call:	1238	58.7%	1.000
0.04999924	203.156	call funct:	595	call gradient:	13	avoid call:	847	58.7%	1.050
0.04000092	202.942	call funct:	595	call gradient:	13	avoid call:	847	58.7%	1.090
0.04999924	202.719	call funct:	778	call gradient:	17	avoid call:	1108	58.7%	1.140
0.03000069	202.492	call funct:	503	call gradient:	11	avoid call:	717	58.7%	1.170
0.05999947	202.261	call funct:	778	call gradient:	17	avoid call:	1108	58.7%	1.230
0.03999901	202.029	call funct:	595	call gradient:	13	avoid call:	847	58.7%	1.270
0.06000137	201.798	call funct:	870	call gradient:	19	avoid call:	1238	58.7%	1.330
0.02999878	201.571	call funct:	549	call gradient:	12	avoid call:	782	58.7%	1.360
0.04000092	201.352	call funct:	595	call gradient:	13	avoid call:	847	58.7%	1.400
0.06000137	201.141	call funct:	824	call gradient:	18	avoid call:	1173	58.7%	1.460
0.03999901	200.941	call funct:	595	call gradient:	13	avoid call:	847	58.7%	1.500
0.14999962	200.749	call funct:	2106	call gradient:	46	avoid call:	2999	58.7%	1.650
0.05000114	200.576	call funct:	641	call gradient:	14	avoid call:	912	58.7%	1.700
0.05999947	200.418	call funct:	870	call gradient:	19	avoid call:	1238	58.7%	1.760
0.03000069	200.277	call funct:	412	call gradient:	9	avoid call:	586	58.7%	1.790
0.03999901	200.152	call funct:	503	call gradient:	11	avoid call:	717	58.7%	1.830
0.03000069	200.043	call funct:	458	call gradient:	10	avoid call:	652	58.7%	1.860
0.04999924	199.937	call funct:	778	call gradient:	17	avoid call:	1108	58.7%	1.910
0.03999901	200.079	call funct:	549	call gradient:	12	avoid call:	782	58.7%	1.950
0.05000114	200.292	call funct:	641	call gradient:	14	avoid call:	912	58.7%	2.000
0.03999901	200.571	call funct:	687	call gradient:	15	avoid call:	978	58.7%	2.040
0.04999924	200.964	call funct:	778	call gradient:	17	avoid call:	1108	58.7%	2.090
0.03000069	201.034	call funct:	458	call gradient:	10	avoid call:	652	58.7%	2.120
0.02000046	201.116	call funct:	320	call gradient:	7	avoid call:	456	58.7%	2.140
0.03999901	201.211	call funct:	503	call gradient:	11	avoid call:	717	58.7%	2.180
0.02000046	201.320	call funct:	320	call gradient:	7	avoid call:	456	58.7%	2.200
0.04999924	201.444	call funct:	641	call gradient:	14	avoid call:	912	58.7%	2.250
0.05000114	201.583	call funct:	687	call gradient:	15	avoid call:	978	58.7%	2.300
0.03000069	201.736	call funct:	458	call gradient:	10	avoid call:	652	58.7%	2.330
0.04000092	201.901	call funct:	503	call gradient:	11	avoid call:	717	58.7%	2.370
0.06000137	202.076	call funct:	870	call gradient:	19	avoid call:	1238	58.7%	2.430
0.04999924	202.259	call funct:	641	call gradient:	14	avoid call:	912	58.7%	2.480
0.05000114	202.446	call funct:	732	call gradient:	16	avoid call:	1043	58.7%	2.530
0.04000092	202.632	call funct:	549	call gradient:	12	avoid call:	782	58.7%	2.570
0.03999901	202.813	call funct:	549	call gradient:	12	avoid call:	782	58.7%	2.610
0.10000038	202.983	call funct:	1419	call gradient:	31	avoid call:	2021	58.7%	2.710
0.04000092	203.141	call funct:	549	call gradient:	12	avoid call:	782	58.7%	2.750
0.05999947	203.916	call funct:	1007	call gradient:	22	avoid call:	1434	58.7%	2.810
0.04999924	204.054	call funct:	641	call gradient:	14	avoid call:	912	58.7%	2.860
0.05000114	204.171	call funct:	641	call gradient:	14	avoid call:	912	58.7%	2.910
0.02999878	204.269	call funct:	458	call gradient:	10	avoid call:	652	58.7%	2.940
0.09000015	204.346	call funct:	1374	call gradient:	30	avoid call:	1956	58.7%	3.030
0.03999901	204.408	call funct:	549	call gradient:	12	avoid call:	782	58.7%	3.070
0.03000069	204.456	call funct:	549	call gradient:	12	avoid call:	782	58.7%	3.100
0.03999901	204.491	call funct:	595	call gradient:	13	avoid call:	847	58.7%	3.140
0.04000092	204.514	call funct:	549	call gradient:	12	avoid call:	782	58.7%	3.180
0.03999901	204.529	call funct:	595	call gradient:	13	avoid call:	847	58.7%	3.220
0.04000092	204.535	call funct:	595	call gradient:	13	avoid call:	847	58.7%	3.260
0.04000092	204.536	call funct:	595	call gradient:	13	avoid call:	847	58.7%	3.300
0.04000092	204.531	call funct:	595	call gradient:	13	avoid call:	847	58.7%	3.340
0.04000092	204.523	call funct:	549	call gradient:	12	avoid call:	782	58.7%	3.380
0.06999969	204.105	call funct:	961	call gradient:	21	avoid call:	1369	58.7%	3.450
0.03000069	204.105	call funct:	458	call gradient:	10	avoid call:	652	58.7%	3.480
0.03999901	204.104	call funct:	503	call gradient:	11	avoid call:	717	58.7%	3.520
0.04000092	204.102	call funct:	503	call gradient:	11	avoid call:	717	58.7%	3.560
0.02999878	204.099	call funct:	503	call gradient:	11	avoid call:	717	58.7%	3.590
0.03000069	204.096	call funct:	549	call gradient:	12	avoid call:	782	58.7%	3.620
0.02999878	204.093	call funct:	503	call gradient:	11	avoid call:	717	58.7%	3.650
0.04000092	204.089	call funct:	503	call gradient:	11	avoid call:	717	58.7%	3.690
0.03999901	204.085	call funct:	503	call gradient:	11	avoid call:	717	58.7%	3.730
0.04000092	204.080	call funct:	503	call gradient:	11	avoid call:	717	58.7%	3.770
0.18999863	249.937	call funct:	2925	call gradient:	45	avoid call:	0	0.0%	3.960
0.38000107	262.708	call funct:	6097	call gradient:	67	avoid call:	0	0.0%	4.340
0.75000000	204.835	call funct:	13268	call gradient:	124	avoid call:	0	0.0%	5.090

NOTE :

-----  
Les 3 dernier appels sont les tests de nuisance, je n ai pas encore appliquer cette optimisation dessus....

Filtre sur les famille sous H0

	IP	JM	KD1	KD2	FILTRE
1	1	1	198	T F F F F T T F F F T T T T T T T F F F F F F F F F F F F F F T T T F F F F F F T	
2	2	199	394	F T F F F T F T F F F T T F F F T T F T T T T T T F F F F F F F F T T T T T T F F F T	
3	3	395	575	F F T F F T F F T F T F T F F F F F F F T F T F F T T F F F F T T T T T T T T F F T	
4	4	576	774	F F F T F T F F F T T F F F F F F F F F F F F F F T F T F T F F F F T T T T T T T T	
5	5	775	974	F F F T T F F F T F F F F F F F F F F F F F F F F F F T T T T T F F T F T T T T T T	

Exemple PORC

4 : NP  
 16 : NM  
 0 : Effets de nuisances  
 16 : NTNIV (H1)

\* pas optimisé \*

TEMPS UNIT | LRT | NB APPEL FUNCT Famille IP,JM | NB APPEL GRADIENT | APPEL EVITE | NB APPEL EVITE/ APPEL TOTAL | TEMPS CUMULES

TEMPS UNIT	LRT	NB APPEL FUNCT	Famille IP,JM	NB APPEL GRADIENT	APPEL EVITE	NB APPEL EVITE/ APPEL TOTAL	TEMPS CUMULES
0.00000000	104.752	call funct:	475	call gradient:	19	avoid call: 0 0.0%	0.000
0.01999998	100.026	call funct:	1025	call gradient:	25	avoid call: 0 0.0%	0.020
0.00000000	100.182	call funct:	697	call gradient:	17	avoid call: 0 0.0%	0.020
0.00999999	100.027	call funct:	615	call gradient:	15	avoid call: 0 0.0%	0.030
0.01000023	99.836	call funct:	656	call gradient:	16	avoid call: 0 0.0%	0.040
0.00000000	99.623	call funct:	369	call gradient:	9	avoid call: 0 0.0%	0.040
0.00999975	98.302	call funct:	574	call gradient:	14	avoid call: 0 0.0%	0.050
0.01000023	95.634	call funct:	735	call gradient:	18	avoid call: 0 0.0%	0.060
0.00999975	92.542	call funct:	693	call gradient:	17	avoid call: 0 0.0%	0.070
0.00000000	89.819	call funct:	694	call gradient:	17	avoid call: 0 0.0%	0.070
0.00000000	88.151	call funct:	612	call gradient:	15	avoid call: 0 0.0%	0.070
0.00999975	87.539	call funct:	738	call gradient:	18	avoid call: 0 0.0%	0.080
0.01000023	85.114	call funct:	613	call gradient:	15	avoid call: 0 0.0%	0.090
0.00999975	82.337	call funct:	735	call gradient:	18	avoid call: 0 0.0%	0.100
0.00000000	79.264	call funct:	736	call gradient:	18	avoid call: 0 0.0%	0.100
0.01000023	79.241	call funct:	736	call gradient:	18	avoid call: 0 0.0%	0.110
0.00999975	80.986	call funct:	656	call gradient:	16	avoid call: 0 0.0%	0.120
0.01000023	82.044	call funct:	696	call gradient:	17	avoid call: 0 0.0%	0.130
0.00000000	84.790	call funct:	696	call gradient:	17	avoid call: 0 0.0%	0.130
0.00999975	89.367	call funct:	615	call gradient:	15	avoid call: 0 0.0%	0.140
0.01000023	93.955	call funct:	656	call gradient:	16	avoid call: 0 0.0%	0.150
0.00999999	97.184	call funct:	615	call gradient:	15	avoid call: 0 0.0%	0.160
0.00000000	95.365	call funct:	655	call gradient:	16	avoid call: 0 0.0%	0.160
0.00999999	94.530	call funct:	614	call gradient:	15	avoid call: 0 0.0%	0.170
0.00999999	95.378	call funct:	820	call gradient:	20	avoid call: 0 0.0%	0.180
0.00999999	96.179	call funct:	492	call gradient:	12	avoid call: 0 0.0%	0.190
0.00000000	96.215	call funct:	533	call gradient:	13	avoid call: 0 0.0%	0.190
0.00999999	96.635	call funct:	410	call gradient:	10	avoid call: 0 0.0%	0.200
0.00000000	97.462	call funct:	492	call gradient:	12	avoid call: 0 0.0%	0.200
0.00999999	98.462	call funct:	492	call gradient:	12	avoid call: 0 0.0%	0.210
0.00999999	99.325	call funct:	533	call gradient:	13	avoid call: 0 0.0%	0.220
0.00000000	99.863	call funct:	451	call gradient:	11	avoid call: 0 0.0%	0.220
0.00999999	100.522	call funct:	451	call gradient:	11	avoid call: 0 0.0%	0.230
0.00000000	101.187	call funct:	533	call gradient:	13	avoid call: 0 0.0%	0.230
0.00999999	101.664	call funct:	533	call gradient:	13	avoid call: 0 0.0%	0.240

\* Gradient Optimisé \*

TEMPS UNIT | LRT | NB APPEL FUNCT Famille IP,JM | NB APPEL GRADIENT | APPEL EVITE | NB APPEL EVITE/ APPEL TOTAL | TEMPS CUMULES

TEMPS UNIT	LRT	NB APPEL FUNCT	Famille IP,JM	NB APPEL GRADIENT	APPEL EVITE	NB APPEL EVITE/ APPEL TOTAL	TEMPS CUMULES
0.00999999	104.752	call funct:	135	call gradient:	19	avoid call: 339 71.5%	0.010
0.00000000	100.026	call funct:	231	call gradient:	24	avoid call: 753 76.5%	0.010
0.00000000	100.182	call funct:	163	call gradient:	17	avoid call: 533 76.5%	0.010
0.00000000	100.027	call funct:	144	call gradient:	15	avoid call: 470 76.5%	0.010
0.00999999	99.836	call funct:	154	call gradient:	16	avoid call: 502 76.5%	0.020
0.00000000	99.623	call funct:	86	call gradient:	9	avoid call: 282 76.5%	0.020
0.00000000	98.302	call funct:	134	call gradient:	14	avoid call: 439 76.5%	0.020
0.00000000	95.634	call funct:	172	call gradient:	18	avoid call: 562 76.5%	0.020
0.00999999	92.542	call funct:	162	call gradient:	17	avoid call: 530 76.5%	0.030
0.00000000	89.819	call funct:	162	call gradient:	17	avoid call: 531 76.5%	0.030
0.00000000	88.151	call funct:	143	call gradient:	15	avoid call: 468 76.5%	0.030
0.00000000	87.539	call funct:	173	call gradient:	18	avoid call: 564 76.5%	0.030
0.01000023	85.114	call funct:	143	call gradient:	15	avoid call: 469 76.5%	0.040
0.00000000	82.337	call funct:	172	call gradient:	18	avoid call: 562 76.5%	0.040
0.00000000	79.264	call funct:	172	call gradient:	18	avoid call: 563 76.5%	0.040
0.00000000	79.241	call funct:	172	call gradient:	18	avoid call: 563 76.5%	0.040
0.00999975	80.986	call funct:	154	call gradient:	16	avoid call: 502 76.5%	0.050
0.00000000	82.044	call funct:	163	call gradient:	17	avoid call: 532 76.5%	0.050
0.00000000	84.790	call funct:	163	call gradient:	17	avoid call: 532 76.5%	0.050
0.00000000	89.367	call funct:	144	call gradient:	15	avoid call: 470 76.5%	0.050
0.00000000	93.955	call funct:	154	call gradient:	16	avoid call: 502 76.5%	0.050
0.00000000	97.184	call funct:	144	call gradient:	15	avoid call: 470 76.5%	0.050
0.00000000	95.365	call funct:	153	call gradient:	16	avoid call: 501 76.5%	0.050
0.00000000	94.530	call funct:	144	call gradient:	15	avoid call: 469 76.5%	0.050
0.00000000	95.378	call funct:	192	call gradient:	20	avoid call: 627 76.5%	0.050
0.00999975	96.179	call funct:	115	call gradient:	12	avoid call: 376 76.5%	0.060
0.00000000	96.215	call funct:	125	call gradient:	13	avoid call: 407 76.5%	0.060
0.00000000	96.635	call funct:	96	call gradient:	10	avoid call: 313 76.5%	0.060
0.00000000	97.462	call funct:	115	call gradient:	12	avoid call: 376 76.5%	0.060
0.00000000	98.462	call funct:	115	call gradient:	12	avoid call: 376 76.5%	0.060
0.01000023	99.325	call funct:	125	call gradient:	13	avoid call: 407 76.5%	0.070
0.00000000	99.863	call funct:	105	call gradient:	11	avoid call: 345 76.5%	0.070
0.00000000	100.522	call funct:	105	call gradient:	11	avoid call: 345 76.5%	0.070
0.00000000	101.187	call funct:	125	call gradient:	13	avoid call: 407 76.5%	0.070
0.00000000	101.664	call funct:	125	call gradient:	13	avoid call: 407 76.5%	0.070

Filtre sur les famille sous H0

IP	JM	KD1	KD2	FILTRE
1	1	1	28	T F F F T T F F T F F F
1	2	29	32	T F F F T T F F F F F F
1	3	33	47	T F F F T T F F F F F F
1	4	48	58	T F F F T T F F F F F F
2	5	59	72	F T F F T F T F F F F F
2	6	73	76	F T F F T F T F F F F F
2	7	77	90	F T F F T F T F F F F F
2	8	91	99	F T F F T F T F F F F F
3	9	100	123	F F T F T F F T F T F F
3	10	124	150	F F T F T F F T F F T F
3	11	151	160	F F T F T F F T F F F F
3	12	161	174	F F T F T F F T F F F F
3	13	175	182	F F T F T F F T F F F F
4	14	183	197	F F F T T F F F F F F F
4	15	198	212	F F F T T F F F F F F F
4	16	213	236	F F F T T F F F F F F T