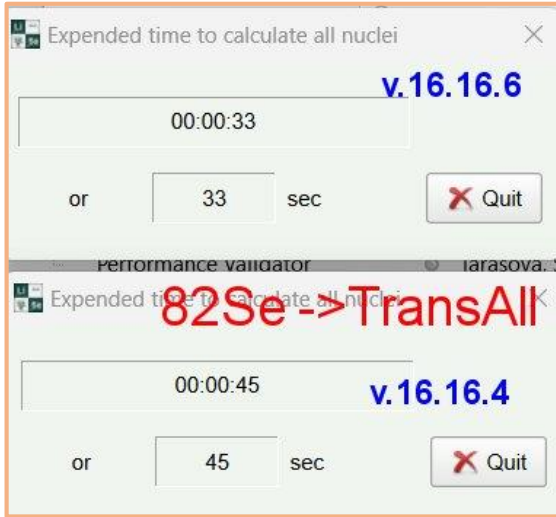


get_direction New Version Optimization



Profiling test changed: +1hr TransAll runtime -> ~5min previously calculated test runtime

1. Inject profiler into LISE running in MSVC debug mode
2. Open test file `1 C:\Users\sasha\OneDrive\Documents\LISEcute\files\for_test.lpp`
3. Disable charge states charge states
4. Calculate

v. 16.16.4

```
LISE++_MSVC.exe:Thu Jul 20 00:18:20 2023
110
111 673,582 107,035.99 => int distribution2::get_direction(void)
112 673.582 106.391.34 {
```

The code changes in the session results after updating the source so the original lines aren't visible, but the line visits and times are.

Not possible to save results in session as that would violate source code security

v. 16.16.6

```
LISE++_MSVC.exe:Thu Jul 20 01:58:11 2023
109
110
111
112 673,582 2,689.49
113 673,582 431.06
114
115 5,852 1,293.84
116 5,852 3.41
117
118
119
```

```
110 111 int distribution2::get_direction(void)
111 112 {
112 - return get_direction_array(x_axis, NN()+1);
113 + if(directionCurrent!=BadValue) return directionCurrent;
114 +
115 + directionCurrent = get_direction_array(x_axis, NN()+1);
116 + return directionCurrent;
113 117 }
```

Child Function (get_direction_array) Results

Running the same task

v. 16.16.4

LISE++_MSVC.exe:Thu Jul 20 00:18:20 2023

```

162 //%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
163 //%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
164 int get_direction_array(double *axis, int points)
165 {
166     double sum_abs=0;
167     double sum_sim=0;
168     double *p1, *p0;
169     p1= &axis[1];
170     p0= &axis[0];
171     for(int i=0; i<points-1; i++)
172     {
173         double dif = *p1 - *p0;
174         sum_abs += dif>0 ? dif : -dif;
175         sum_sim += dif;
176         p0 = p1; p1++;
177     }
178     if(sum_abs==0) return 1;
179     double ratio = sum_sim/sum_abs;
180     double vplus = 1. - ratio;
181     double vminus= 1. + ratio;
182     if(vplus < 1e-7) return 1;
183     if(vminus < 1e-7) return -1;
184     return 0;
185 }
186
187
188
189
190
191

```

Call counts for v. 16.16.4 (highlighted in yellow):

165	673,793	105,044.85
166	673,793	200.56
167	673,793	200.31
171	673,793	199.82
172	673,793	199.40
174	673,793	199.72
176	43,122,752	12,510.30
177	43,122,752	12,515.74
178	43,122,752	12,486.55
179	43,122,752	12,461.72
180	43,122,752	12,440.57
182	673,793	196.33
184	673,793	198.90
185	673,793	200.30
186	673,793	197.53
188	673,793	340.00
189	1,103	0.53

Class::Method	Call Count	Call Cou...	Function	Function %	Children	Children %	Total	Total %
distribution2::get_direction	673,582	3.24%	1,441.39	0.53%	105,594.61	38.68%	107,035.99	39.20%

Functions called by function: distribution2::get_direction

Class::Method	Call Count	Call Cou...	Function	Function...	Children	Children %	Total	Total %
get_direction_array	673,793	3.24%	105,044.85	38.47%	0.00	0.00%	105,044.85	38.47%

v. 16.16.6

LISE++_MSVC.exe:Thu Jul 20 01:58:11 2023

```

162 //%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
163 //%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
164 int get_direction_array(double *axis, int points)
165 {
166     double sum_abs=0;
167     double sum_sim=0;
168     double *p1, *p0;
169     p1= &axis[1];
170     p0= &axis[0];
171     for(int i=0; i<points-1; i++)
172     {
173         double dif = *p1 - *p0;
174         sum_abs += dif>0 ? dif : -dif;
175         sum_sim += dif;
176         p0 = p1; p1++;
177     }
178     if(sum_abs==0) return 1;
179     double ratio = sum_sim/sum_abs;
180     double vplus = 1. - ratio;
181     double vminus= 1. + ratio;
182     if(vplus < 1e-7) return 1;
183     if(vminus < 1e-7) return -1;
184     return 0;
185 }
186
187
188
189
190
191

```

Call counts for v. 16.16.6 (highlighted in yellow):

165	6,063	1,320.91
166	6,063	2.50
167	6,063	2.48
171	6,063	2.43
172	6,063	2.43
174	6,063	2.43
176	388,032	153.85
177	388,032	153.62
178	388,032	153.16
179	388,032	152.65
180	388,032	152.68
182	6,063	2.57
184	6,063	2.45
185	6,063	2.45
186	6,063	2.43
188	6,063	3.77
189	1,103	0.68

No change in L_Distr2.cpp source code

Class::Method	Call Count	Call Cou...	Function	Function %	Children	Children %	Total	Total %
distribution2::get_direction	673,582	3.17%	1,405.65	0.57%	1,283.85	0.52%	2,689.49	1.08%

Functions called by function: distribution2::get_direction

Class::Method	Call Count	Call Cou...	Function	Function...	Children	Children %	Total	Total %
get_direction_array	6,063	0.03%	1,320.91	0.53%	0.00	0.00%	1,320.91	0.53%

Chart Reference

for_test (charge disabled)

check again inter2
MakeUniform

Inter2

```

double distribution2::inter2(double x, int method, bo
if(method>4 || method<2)
    method=2; //check stupid warning in s

double CheckValue = inter2_New (x, method, FlagLog,
double
    p = inter2_Old (x, method, FlagLog,

if( (p==0 && CheckValue!=0) || (p!=0 && CheckValu
    || (qFabs(p)>0 && qFabs( (CheckValu
    || (qFabs(CheckValue)>0 && qFabs( (CheckValu
    {
        // double p1 = inter2_Old (
        if(method!=2) p = inter2_New (x, 2, FlagLog, On
        else p = CheckValue*0.99999;
    } //
return p;
    
```

16.16.4 16.16.6

Class::Method	Call Count	Call Cou...	Function	Function...	Children	Children %	Total	Total %
distribution2::inter2	327,340	1.57%	1,324.72	0.49%	167,973.71	61.52%	169,298.44	62.01%

Class::Method	Call Count	Call Cou...	Function	Function...	Children	Children %	Total	Total %
distribution2::inter2	327,340	1.54%	1,936.23	0.78%	92,626.78	37.26%	94,563.01	38.03%

- 28.92% (98.15%) lise++_msvc.exe distribution2::inter2PlusExtrapolate
- 25.48% (88.78%) lise++_msvc.exe distribution2::inter2
- 12.71% (50.26%) lise++_msvc.exe distribution2::inter2_Old
- 7.82% (69.47%) lise++_msvc.exe distribution2::get_direction
- 1.65% (14.63%) lise++_msvc.exe distribution2::get_xmin
- 1.64% (14.60%) lise++_msvc.exe distribution2::get_xmax
- 0.14% (1.25%) lise++_msvc.exe find_line
- 0.01% (0.06%) lise++_msvc.exe mlexp
- 12.58% (49.74%) lise++_msvc.exe distribution2::inter2_New
- 7.83% (68.63%) lise++_msvc.exe distribution2::get_direction
- 1.65% (14.45%) lise++_msvc.exe distribution2::get_xmin
- 1.65% (14.45%) lise++_msvc.exe distribution2::get_xmax
- 0.28% (2.47%) lise++_msvc.exe distribution2::inter2_p2_local

- 13.58% (95.90%) lise++_msvc.exe distribution2::inter2PlusExtrapolate
- 10.08% (75.51%) lise++_msvc.exe distribution2::inter2
- 4.99% (50.55%) lise++_msvc.exe distribution2::inter2_Old
- 1.66% (47.00%) lise++_msvc.exe distribution2::get_xmax
- 1.66% (46.91%) lise++_msvc.exe distribution2::get_xmin
- 0.14% (3.93%) lise++_msvc.exe find_line
- 0.07% (1.97%) lise++_msvc.exe distribution2::get_direction
- 0.01% (0.19%) lise++_msvc.exe mlexp
- 4.89% (49.45%) lise++_msvc.exe distribution2::inter2_New
- 1.66% (44.89%) lise++_msvc.exe distribution2::get_xmax
- 1.66% (44.79%) lise++_msvc.exe distribution2::get_xmin
- 0.28% (7.67%) lise++_msvc.exe distribution2::inter2_p2_local
- 0.10% (2.65%) lise++_msvc.exe distribution2::get_direction

MakeUniform

```

void distrFour::MakeUniform(int /*Mode*/)
if(Uniform) return;

double xmin=d4[Base]->get_xmin_nonzero(d4[e4I]->Re
double xmax=d4[Base]->get_xmax_nonzero(d4[e4I]->Re

if(xmax<=xmin) {
    xmin=d4[Base]->get_xmin();
    xmax=d4[Base]->get_xmax();
}

if(!CheckIntensityDistribution())
{Uniform=false; return;}

d4[Base]->set(xmin,xmax);
for (int i=0; i<N4; i++)
{
    if(i==Base) continue;
    // d4[i]->alignB(d4[Base],i==0? 2 : Mode,
    d4[i]->alignB(d4[Base],2, i==0,false,true);
}

d4[Base]->MakeFromXC();
if(Base != e4I) d4[e4I]->EraseNegativeValues();
    
```

16.16.4 16.16.6

Class::Method	Call Count	Call Cou...	Function	Function...	Children	Children %	Total	Total %
distrFour::MakeUniform	453	0.00%	15.07	0.01%	168,712.66	61.79%	168,727.73	61.80%

Class::Method	Call Count	Call Cou...	Function	Function...	Children	Children %	Total	Total %
distrFour::MakeUniform	453	0.00%	21.34	0.01%	115,053.71	46.28%	115,075.04	46.29%

- 29.95% (94.20%) lise++_msvc.exe distrFour::ChangeBase
- 29.71% (99.19%) lise++_msvc.exe distrFour::MakeUniform
- 29.68% (99.89%) lise++_msvc.exe distribution2::alignB
- 28.92% (98.15%) lise++_msvc.exe distribution2::inter2PlusExtrapolate
- 25.48% (88.78%) lise++_msvc.exe distribution2::inter2
- 14.67% (91.28%) lise++_msvc.exe distrFour::ChangeBase
- 14.41% (98.21%) lise++_msvc.exe distrFour::MakeUniform
- 14.37% (99.77%) lise++_msvc.exe distribution2::alignB
- 13.58% (95.90%) lise++_msvc.exe distribution2::inter2PlusExtrapolate
- 10.08% (75.51%) lise++_msvc.exe distribution2::inter2

inter2_Old Line Times

for_test (charge disabled)

Line	Time	Value	Code
47		
48			double distribution2::inter2_Old(double x, int method, bool FlagLog, bool OnlyPositive)
49	327,340	43,248.30	{
50			
51	327,340	129.56	if(method==4 && !FlagLog) {return get_spline(x);}
52			
53	327,340	128.49	if(method>2 && NN()<= 1)
54	0	0.00	method=2;
55			
56			
57	327,340	127.24	int i=0;
58			double p; // method = 2 line
59	327,340	126.57	int idelta=-1; // method = 3 parabola
60			int ideltaNext, ideltaNext2;
61			//int method_save=method;
62			//int ideltaPrev; // 11/29/2011
63	327,340	29,405.90	if(x < get_xmin() x > get_xmax()) return 0;
64			
65	327,340	915.53	int direction=get_direction();
66	327,340	130.67	if(direction==0)return 0; //mixing
67			
68			
69	327,340	129.76	if(direction==1)
70	327,340	130.15	for(i=N-1; i>=0; i--)
71	10,553,078	4,023.73	if(x-x_axis[i] >= 0) { idelta=i; break;}
72			
73	327,340	124.87	if(direction== -1)
74	0	0.00	for(i=1; i<=N; i++)
75	0	0.00	if(x-x_axis[i] >= 0) { idelta=i-1; break;}
76			// oleg corrected 01/30/2004
77			
78			
79	304,221	115.46	ideltaNext =idelta+1;
80	304,221	115.72	ideltaNext2=idelta+2;
81			//ideltaPrev =idelta-1;
82			
83	304,221	115.97	if(ideltaNext * idelta == 0)
84	2,381	0.91	{idelta=0; ideltaNext=1; ideltaNext2=2;}
85			
86	304,221	116.17	if((ideltaNext-N) * (idelta-N) == 0)
87	6,492	2.50	{idelta=N; ideltaNext=N-1; ideltaNext2=N-2;}
88			
89			//----- FlagLog - start
90	304,221	117.20	double fi =f[idelta];
91	304,221	116.07	double fin =f[ideltaNext];
92	304,221	116.29	double fin2=f[ideltaNext2];
93			
94	304,221	115.76	if(FlagLog) {
95	19,044	7.36	if(fi<=exp_ln_limit && fin<=exp_ln_limit) return 0; // 4.01.02 oleg
96			
97	19,044	7.27	if(fi<=exp_ln_limit) fi = -99;
98	19,044	8.07	else fi = log(fi);
99			
100	19,044	7.28	if(fin<=exp_ln_limit) fin = -99;
101	19,044	7.62	else fin = log(fin);
102			}
103	304,221	115.80	if(method > 2) {
104	0	0.00	if(FlagLog) {
105	0	0.00	if(fin2 <=exp_ln_limit) method=2;
106	0	0.00	else fin2 = log(fin2);
107			}
108			}
109			}

109		
110	304,221	129.14
111		
112		
113		
114	304,221	1,440.20
115		
116		
117	0	0.00
118		
119		
120		
121	304,221	177.75
122	304,221	118.24
123		
124		
125	304,221	164.25
126	----	----
127		

```

if(qFabs(x_axis[ideltaNext2]-x_axis[ideltaNext]) <1e-3) method=2;
//----- FlagLog - end
if(method==2) p=find_line( x_axis[idelta], fi,
x_axis[ideltaNext], fin, x);
else p=find_parabola(x_axis[idelta], fi, // 3 or 5
x_axis[ideltaNext], fin,
x_axis[ideltaNext2], fin2,x);
if(FlagLog) p=mIexp(p);
if(OnlyPositive) p=qMax(p,0.);
return p;
};

```

More time consumed

inter2_New Line Times

for_test (charge disabled)

130			//~~~~~
131			double distribution2::inter2_New(double x, int method, bool FlagLog, bool OnlyPositive)
132	327,340	42,608.52	{
133	327,340	125.68	if(method==4 && !FlagLog) {return get_spline(x);}
134			
135	327,340	124.92	if(method>2 && NN()<= 1) method=2;
136			
137			
138	327,340	124.92	int i=0; // method = 2 line
139			// method = 3 parabola
140			
141	327,340	124.71	int istart=-1;
142			
143			//int ideltaPrev; // 11/29/2011
144			
145	327,340	29,418.08	if(x < get_xmin() x > get_xmax()) return 0;
146			
147	327,340	1,291.64	int direction=get_direction();
148	327,340	129.81	if(direction==0)return 0; //mixing
149			
150	327,340	129.39	if(direction==1)
151	327,340	130.12	for(i=N-1; i>=0; i--)
152	10,553,078	4,020.83	if(x-x_axis[i] >= 0) { istart=i; break;}
153			
154	327,340	124.57	if(direction==-1)
155	0	0.00	for(i=1; i<=N; i++)
156	0	0.00	if(x-x_axis[i] >= 0) { istart=i-1; break;} // oleg corrected 01/30/2004
157			
158	327,340	131.49	if(x-x_axis[i]==0) return f[istart];
159			
160	304,221	115.84	if(FlagLog)
161			{
162			if(f[istart]<=0 && f[istart+1]<=0) return 0;
163	19,044	7.44	
164			if(f[istart]<=0 f[istart+1]<=0) return inter2_p2_local(x, istart, FlagLog,Only
165	19,044	7.32	};
166			
167			
168			
169	304,221	2,851.92	if(method!=3) return inter2_p2_local(x, istart, FlagLog,OnlyPositive);
170			
171			//=====
172			
173	0	0.00	int Mode = 0; // two parabolas. : normal
174			
175	0	0.00	istart--; // to correspond fim
176			
177			//int KR = 1 ; // between 1st and 2nd -- start in t-array 0 { 1 x 2 3 }
178			//int KL = 0 ; // between 2nd and 3rd { 0 1 x 2 }
179			
180	0	0.00	double XR, XL, deltaR=0, deltaL=0, aL, aR, WidthD, vs;
181			double ty[4], tx[4];
182			
183			
184	0	0.00	if(istart >= N-2)
185			{
186	0	0.00	Mode = 1; // do not use KR!!!
187	---	---	}
188			else {
189	0	0.00	XR = XI(istart+2);
190	0	0.00	deltaR=qFabs(XR-x);

0 | 0 until end of function

271			
272			
273	0	0.00	if(OnlyPositive) vs=qMax(vs,0.);
274	0	0.00	return vs;
275	---	---	}
276			//~~~~~

More time consumed

inter2_Old & inter2_New Chart

for_test (charge disabled)

Class::Method	Call Count	Call Cou...	Function	Function %	Children	Children %	Total	Total %
distribution2::inter2_Old	327,340	1.36%	12,508.67	5.56%	30,739.63	13.67%	43,248.30	19.23%
distribution2::inter2_New	327,340	1.36%	10,157.07	4.52%	32,451.45	14.43%	42,608.52	18.94%

Functions called by function: distribution2::inter2_Old

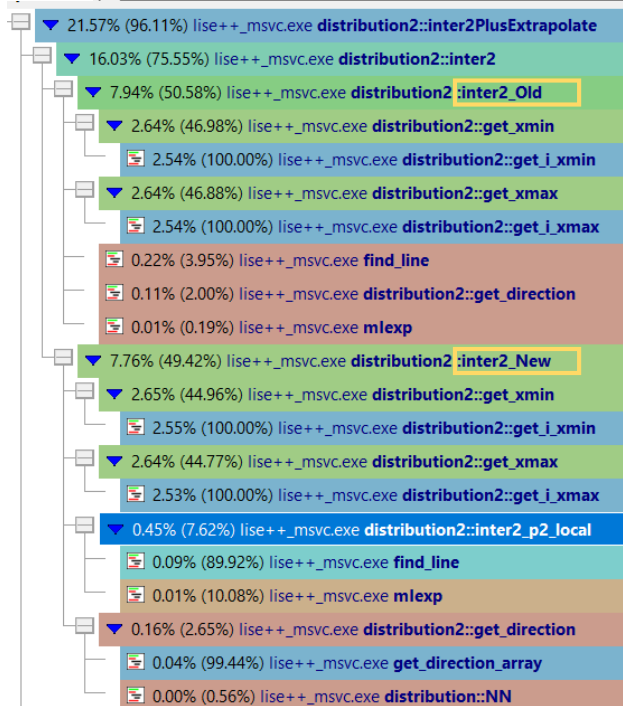
Class::Method	Call Count	Call Cou...	Function	Function...	Children	Children %	Total	Total %
find_line	476,055	1.98%	1,817.49	0.81%	0.00	0.00%	1,817.49	0.81%
distribution2::get_xmax	953,826	3.96%	1,623.05	0.72%	40,173.13	17.86%	41,796.18	18.58%
distribution2::get_xmin	953,902	3.96%	1,562.43	0.69%	40,331.86	17.93%	41,894.29	18.63%
distribution2::get_direction	673,582	2.80%	1,268.91	0.56%	1,212.50	0.54%	2,481.41	1.10%
mlexp	38,088	0.16%	97.05	0.04%	0.00	0.00%	97.05	0.04%

Functions called by function: distribution2::inter2_New

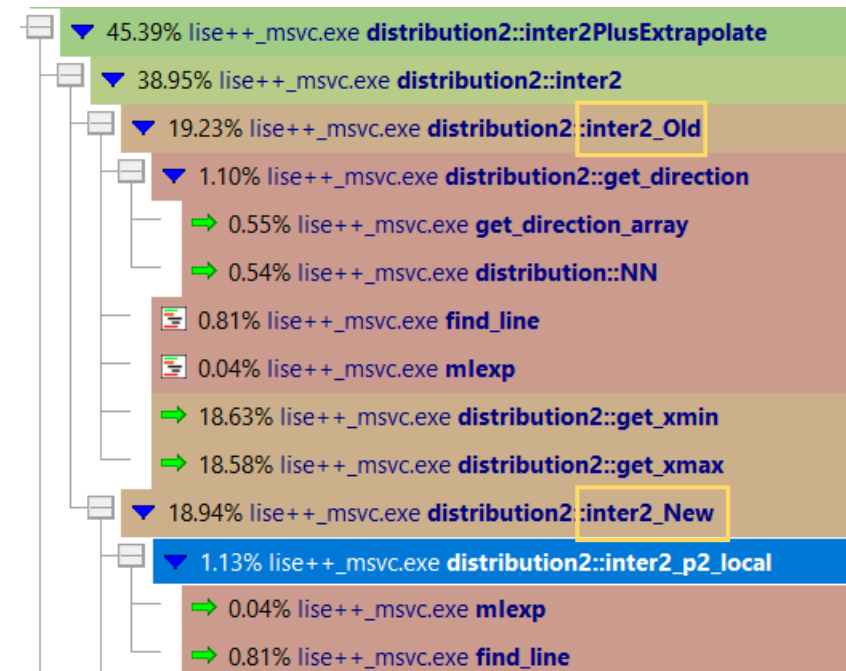
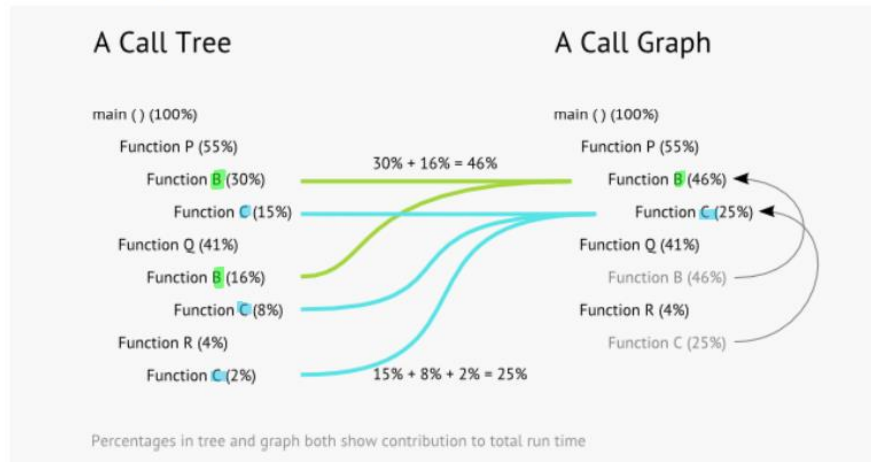
Class::Method	Call Count	Call Cou...	Function	Function...	Children	Children %	Total	Total %
distribution2::inter2_p2_local	304,221	1.26%	1,915.66	0.85%	617.08	0.27%	2,532.74	1.13%
distribution2::get_xmax	953,826	3.96%	1,623.05	0.72%	40,173.13	17.86%	41,796.18	18.58%
distribution2::get_xmin	953,902	3.96%	1,562.43	0.69%	40,331.86	17.93%	41,894.29	18.63%
distribution2::get_direction	673,582	2.80%	1,268.91	0.56%	1,212.50	0.54%	2,481.41	1.10%

Old

New



Multiple locations vs. Once



for_test Charge Disabled vs Charge Enabled

Line	Call Count	Call Count	Function	Function %	Children	Children %	Total	Total %
16								
17								
18	327,340	90,503.16	186,172	32,573.49				
19	327,340	132.33	186,172	78.91				
20	0	0.00	0	0.00				
21								
22								
23								
24	327,340	44,335.17	186,172	15,782.31				
25	327,340	44,972.54	186,172	16,124.50				
26								
27								
28								
29								
30								
31	327,340	140.43	186,172	82.54				
32								
33								
34								
35	3	0.00	1	0.00				
36	3	0.00	1	0.00				
37								
38								
39								
40	327,340	165.50	186,172	94.59				
41								
42								
43								

```

double distribution2::inter2(double x, int method, bool FlagLog, bool
if(method>4 || method<2)
    method=2; //check stupid warning in stacks window

double CheckValue = inter2_New (x, method, FlagLog, OnlyPositive);
double p = inter2_Old (x, method, FlagLog, OnlyPositive);

if( (p==0 && CheckValue!=0) || (p!=0 && CheckValue==0)
    || (qFabs(p)>0 && qFabs( (CheckValue-p)/p)
    || (qFabs(CheckValue)>0 && qFabs( (CheckValue-p)/CheckValue)
    )
    {
        // double p1 = inter2_Old (x, method, FlagLog, OnlyPositive);
        if(method!=2) p = inter2_New (x, 2, FlagLog, OnlyPositive);
        else p = CheckValue*0.99999;
    }

return p;

//return CheckValue;
};
    
```

Class::Method	Call Count	Call Cou...	Function	Function %	Children	Children %	Total	Total %
distribution2::inter2	327,340	1.34%	1,875.91	0.79%	88,627.25	37.26%	90,503.16	38.05%

Class::Method	Call Count	Call Cou...	Function	Function %	Children	Children %	Total	Total %
distribution2::inter2	186,172	0.96%	1,020.46	0.64%	31,553.03	19.90%	32,573.49	20.54%

inter2

Charge enabled:

- Less calls
- Less time to execute

Line	Call Count	Call Count	Function	Function %	Children	Children %	Total	Total %
211								
212	453	110,543.09	511	40,871.61				
213	453	0.26	511	0.27				
214								
215	308	27.83	348	17.84				
216	308	52.00	348	32.23				
217								
218	308	0.13	348	0.15				
219	0	0.00	0	0.00				
220	0	0.00	0	0.00				
221								
222								
223	308	17.29	348	12.54				
224	0	0.00	0	0.00				
225								
226	308	42.37	348	26.55				
227								
228	308	0.13	348	0.15				
229								
230	4,620	1.87	5,220	2.24				
231								
232								
233	4,312	110,269.96	4,872	40,701.92				
234	4,312	2.58	4,872	2.61				
235								
236	308	73.75	348	41.14				
237								
238	308	49.15	348	27.88				
239								
240	308	0.18	348	0.21				
241								
242								

```

void distrFour::MakeUniform(int /*Mode*/)
{
    if(Uniform) return;

    double xmin=d4[Base]->get_xmin_nonzero(d4[e4I]->ReturnPtrF());
    double xmax=d4[Base]->get_xmax_nonzero(d4[e4I]->ReturnPtrF());

    if(xmax<=xmin) {
        xmin=d4[Base]->get_xmin();
        xmax=d4[Base]->get_xmax();
    }

    if(!CheckIntensityDistribution())
        {Uniform=false; return;}

    d4[Base]->set(xmin,xmax);

    for (int i=0; i<N4; i++)
    {
        if(i==Base) continue;

        // d4[i]->alignB(d4[Base],i==0? 2 : Mode, i==0,false,true);
        d4[i]->alignB(d4[Base],2, i==0,false,true); //Extrapolate
    }

    d4[Base]->MakeFfromXC();

    if(Base != e4I) d4[e4I]->EraseNegativeValues();

    Uniform=true;
}
    
```

Class::Method	Call Count	Call Cou...	Function	Function %	Children	Children %	Total	Total %
distrFour::MakeUniform	453	0.00%	19.51	0.01%	110,523.59	46.46%	110,543.09	46.47%

Class::Method	Call Count	Call Cou...	Function	Function %	Children	Children %	Total	Total %
distrFour::MakeUniform	511	0.00%	20.73	0.01%	40,850.88	25.76%	40,871.61	25.78%

MakeUniform

Charge enabled:

- More calls
- Less time to execute

for_test Charge Disabled vs Charge Enabled

```

48 //www.....
49 double distribution2::inter2_Old(double x, int method, bool Flag
50 {
51     if(method==4 && !FlagLog) {return get_spline(x);}
52     if(method>2 && NN()<= 1)
53         method=2;
54     int i=0;
55     double p;
56     int idelta=-1;
57     int ideltaNext, ideltaNext2;
58     //int method_save=method;
59     //int ideltaPrev;
60     // 11/29/2011
61     if( x < get_xmin() || x > get_xmax() ) return 0;
62     int direction=get_direction();
63     if(direction==0)return 0; //mixing
64     if(direction==1)
65         for(i=N-1; i>=0; i--)
66             if(x-x_axis[i] >= 0) { idelta=i; break;}
67     if(direction==-1)
68         for(i=1; i<=N; i++)
69             if(x-x_axis[i] >= 0) { idelta=i-1; break;} // oleg c
70     if(x-x_axis[idelta]==0) return f[idelta];
71 }

```

Inter2_Old

```

131 //www.....
132 double distribution2::inter2_New(double x, int method, bc
133 {
134     if(method==4 && !FlagLog) {return get_spline(x);}
135     if(method>2 && NN()<= 1) method=2;
136     int i=0;
137     // method = 3 parabola
138     int istart=-1;
139     //int ideltaPrev;
140     // 11/29/2011
141     if( x < get_xmin() || x > get_xmax() ) return 0;
142     int direction=get_direction();
143     if(direction==0)return 0; //mixing
144     if(direction==1)
145         for(i=N-1; i>=0; i--)
146             if(x-x_axis[i] >= 0) { istart=i; break;}
147     if(direction==-1)
148         for(i=1; i<=N; i++)
149             if(x-x_axis[i] >= 0) { istart=i-1; break;} //
150     if(x-x_axis[istart]==0) return f[istart];
151 }

```

Inter2_New

Class::Method	Call Count	Call Cou...	Function	Function %	Children	Children %	Total	Total %
distribution2::inter2_Old	327,340	1.34%	12,883.92	5.42%	31,749.48	13.35%	44,633.40	18.76%

Inter2_Old

Charge enabled:

- Less calls
- Less time to execute

Class::Method	Call Count	Call Cou...	Function	Function %	Children	Children %	Total	Total %
distribution2::inter2_Old	186,172	0.96%	5,489.90	3.46%	10,457.45	6.60%	15,947.35	10.06%

Class::Method	Call Count	Call Cou...	Function	Function %	Children	Children %	Total	Total %
distribution2::inter2_New	327,340	1.34%	10,499.57	4.41%	33,494.27	14.08%	43,993.84	18.49%

Inter2_New

Charge enabled:

- Less calls
- Less time to execute

Class::Method	Call Count	Call Cou...	Function	Function %	Children	Children %	Total	Total %
distribution2::inter2_New	186,172	0.96%	4,170.11	2.63%	11,435.57	7.21%	15,605.68	9.84%



for_test_charge_states_nonequilibrium

```

110
111 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
112 477,312 1,851.64 int distribution2::get_direction(void) // -1 -
113 477,312 306.04 if(directionCurrent!=BadValue) return directionCurrent;
114
115 6,899 822.10 directionCurrent = get_direction_array(x_axis, NN()+1);
116 6,899 3.98 return directionCurrent;
117 }
118 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    
```

get_direction

Class::Method	Call Count	Call Cou...	Function	Function %	Children	Children %	Total	Total %
distribution2::get_direction	477,312	1.39%	1,042.26	0.41%	809.38	0.32%	1,851.64	0.73%

```

163
164 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
165 7,267 841.63 int get_direction_array(double *axis, int points)
166 7,267 3.20 double sum_abs=0;
167 7,267 3.19 double sum_sim=0;
168
169 double *p1, *p0;
170
171 7,267 3.13 p1= &axis[1];
172 7,267 3.11 p0= &axis[0];
173
174 7,267 3.10 for(int i=0; i<points-1; i++)
175 {
176 232,544 96.59 double dif = *p1 - *p0;
177 232,544 96.73 sum_abs += dif>0 ? dif : -dif;
178 232,544 96.19 sum_sim += dif;
179 232,544 95.93 p0 = p1; p1++;
180 232,544 96.25 }
181
182 7,267 3.22 if(sum_abs==0) return 1;
183
184 7,267 3.06 double ratio = sum_sim/sum_abs;
185 7,267 3.08 double vplus = 1. - ratio;
186 7,267 3.03 double vminus= 1. + ratio;
187
188 7,267 4.46 if(vplus < 1e-7) return 1;
189 1,096 0.68 if(vminus < 1e-7) return -1;
190 return 0;
191 }
    
```

get_direction_array

Class::Method	Call Count	Call Cou...	Function	Function...	Children	Children %	Total	Total %
get_direction_array	7,267	0.02%	841.63	0.33%	0.00	0.00%	841.63	0.33%

for_test_charge_states_nonequilibrium

```

17 //
18 double distribution2::inter2(double x, int method, bool FlagLog, bool OnlyPositive)
19 {
20     if(method>4 || method<2)
21         method=2; //check stupid warning in stacks window
22
23
24     double CheckValue = inter2_New (x, method, FlagLog, OnlyPositive);
25     double p = inter2_Old (x, method, FlagLog, OnlyPositive);
26
27
28
29     if( (p==0 && CheckValue!=0) || (p!=0 && CheckValue==0) //
30         || (qFabs(p)>0 && qFabs( (CheckValue-p)/p) >1 && FlagLog)
31         || (qFabs(CheckValue)>0 && qFabs( (CheckValue-p)/CheckValue)>1 && FlagLog)
32     )
33     {
34         // double p1 = inter2_Old (x, method, FlagLog, OnlyPositi
35         if(method!=2) p = inter2_New (x, 2, FlagLog, OnlyPositive);
36         else p = CheckValue*0.99999;
37     } //
38
39     return p;
40 //return CheckValue;
41 };
42
43

```

inter2

Class::Method	Call Count	Call Cou...	Function	Function %	Children	Children %	Total	Total %
distribution2::inter2	226,812	0.66%	1,381.26	0.54%	40,738.83	15.99%	42,120.08	16.53%

```

211 //
212 void distrFour::MakeUniform(int /*Mode*/)
213 {
214     if(Uniform) return;
215     double xmin=d4[Base]->get_xmin_nonzero(d4[e4I]->ReturnPtrF());
216     double xmax=d4[Base]->get_xmax_nonzero(d4[e4I]->ReturnPtrF());
217
218     if(xmax<=xmin) {
219         xmin=d4[Base]->get_xmin();
220         xmax=d4[Base]->get_xmax();
221     }
222
223     if(!CheckIntensityDistribution())
224         {Uniform=false; return;}
225
226     d4[Base]->set(xmin,xmax);
227
228     for (int i=0; i<N4; i++)
229     {
230         if(i==Base) continue;
231
232         // d4[i]->alignB(d4[Base],i==0? 2 : Mode, i==0,false,true);
233         d4[i]->alignB(d4[Base],2, i==0,false,true); //Extrapolate
234     }
235
236     d4[Base]->MakeFfromX();
237
238     if(Base != e4I) d4[e4I]->EraseNegativeValues();
239
240     Uniform=true;
241 }
242 ///////////////////////////////////////////////////////////////////

```

MakeUniform

Class::Method	Call Count	Call Cou...	Function	Function %	Children	Children %	Total	Total %
distrFour::MakeUniform	514	0.00%	22.98	0.01%	44,122.32	17.31%	44,145.30	17.32%



for_test_charge_states_nonequilibrium

```

48 double distribution2::inter2_Old(double x, int method, bool FlagLog, bool OnlyPositive
49 226,812 20,527.41 {
50 if(method==4 && !FlagLog) {return get_spline(x);}
51 226,812 98.85
52 if(method>2 && NN()<= 1)
53 226,812 97.43 method=2;
54 0 0.00
55
56 int i=0; // method = 2 line
57 226,812 96.39 // method = 3 parabola
58 double p;
59 226,812 96.21 int idelta=-1;
60 int ideltaNext, ideltaNext2;
61 //int method_save=method;
62 //int ideltaPrev; // 11/29/2011
63
64 226,812 12,468.84 if( x < get_xmin() || x > get_xmax() ) return 0;
65
66 226,799 743.53 int direction=get_direction();
67 226,799 99.72 if(direction==0)return 0; //mixing
68
69 226,799 98.79 if(direction==1)
70 226,799 98.87 for(i=N-1; i>=0; i--)
71 3,722,472 1,552.74 if(x-x_axis[i] >= 0) { idelta=i; break;}
72
73 226,799 93.94 if(direction==-1)
74 0 0.00 for(i=1; i<=N; i++)
75 0 0.00 if(x-x_axis[i] >= 0) { idelta=i-1; break;} // 0leg corrected 01/30/2004
76
77 226,799 100.84 if(x-x_axis[i]==0) return f[idelta];
78

```

Inter2_Old

```

131 double distribution2::inter2_New(double x, int method, bool FlagLog, bool OnlyPositive
132 226,812 20,211.42 {
133 226,812 95.88 if(method==4 && !FlagLog) {return get_spline(x);}
134
135 226,812 95.37 if(method>2 && NN()<= 1) method=2;
136
137
138 226,812 95.34 int i=0; // method = 2 line
139 // method = 3 parabola
140
141 226,812 95.67 int istart=-1;
142
143 //int ideltaPrev; // 11/29/2011
144
145 226,812 12,492.95 if( x < get_xmin() || x > get_xmax() ) return 0;
146
147 226,799 1,006.50 int direction=get_direction();
148 226,799 99.82 if(direction==0)return 0; //mixing
149
150 226,799 99.00 if(direction==1)
151 226,799 98.68 for(i=N-1; i>=0; i--)
152 3,722,472 1,552.59 if(x-x_axis[i] >= 0) { istart=i; break;}
153
154 226,799 94.00 if(direction==-1)
155 0 0.00 for(i=1; i<=N; i++)
156 0 0.00 if(x-x_axis[i] >= 0) { istart=i-1; break;} // 0leg corrected 01/30/2004
157
158 226,799 101.44 if(x-x_axis[i]==0) return f[istart];
159
160
161 201,416 83.53 if(FlagLog)

```

Inter2_New

Inter2_Old

Class::Method	Call Count	Call Cou...	Function	Function %	Children	Children %	Total	Total %
distribution2::inter2_Old	226,812	0.66%	7,090.20	2.78%	13,437.21	5.27%	20,527.41	8.05%

Inter2_New

Class::Method	Call Count	Call Cou...	Function	Function %	Children	Children %	Total	Total %
distribution2::inter2_New	226,812	0.66%	5,390.81	2.12%	14,820.61	5.82%	20,211.42	7.93%