

## LFR-FSW - Bug #3123

Task # 3120 (Closed): Livraison sources 3.2.0.21 pour analyse Logiscope // W. Recart

### Analyse Logiscope LFR\_3.2.0.21 : LFR\_basic-parameters : Tr\_Parenthèses Severity is Medium

03/10/2018 05:32 PM - William Recart

<b>Status:</b>	Closed	<b>Start date:</b>	03/10/2018
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Alexis Jeandet	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Spent time:</b>	0.00 hour
<b>revision:</b>	r0		

#### Description

Rappel de la règle :

Tr\_Parenthèses

Definition:

-----  
In expressions, every binary and ternary operator has to be put in parenthesis, so that the evaluation priorities are not ambiguous.

Use the partpar option to allow the following rules: when the right operand of a + or \* operator uses the same operator, you can omit parenthesis for it.

In the same way, you can omit parenthesis in the case of the right operand of an assignment operator. Moreover, you can omit parenthesis at the first level of the expression.

Parameters:

-----  
The character string "partpar", which, if used, allows programmers not to put systematically parenthesis, according to the rule above.

Justification:

-----  
Removes ambiguity about the evaluation priorities.

Example:

-----  
// do not write  
result = fact / 100 + rem;

// write  
result = ((fact / 100) + rem);

// or write, with the partpar option  
result = (fact / 100) + rem;

// with the partpar option, write  
result = (fact \* ind \* 100) + rem + 10 + (coeff \*\* c);

// instead of  
result = ((fact \* (ind \* 100)) + (rem + (10 + (coeff \*\* c))));

La règle n'est pas respectée dans 122 cas d'après Logiscope:

basic\_parameters.h : lines 81, 94, 114, 123, 124, 127, 128, 144, 165, 174, 175, 178, 179, 196, 201, 202, 203, 211, 212, 215, 219, 239, 241, 251, 268, 270, 281, 295, 313, 319, 339, 349, 365, 376, 378, 382, 400, 423, 429, 456, 466, 481, 520, 552, 554, 555, 562, 563, 570, 572, 573, 580, 581, 587, 589, 590, 597, 598, 604, 606, 607, 614, 615, 621, 623, 624, 631, 632, 638, 640, 641, 648, 649, 655, 657, 658, 665, 666, 672, 674, 675, 682, 683, 689, 691, 692, 699, 700, 706, 708, 709, 716, 717, 750, 759, 760, 763, 764, 775, 795, 804, 805, 808, 809, 820, 840, 849, 850, 853, 854, 865, 886, 895, 896, 899, 900, 911, 931, 940, 941, 945, 946

#### History

#1 - 08/10/2018 12:38 PM - bruno katra

- Status changed from New to In Progress

1- J'ai eu William au téléphone, l'option 'partpar' est bien activée donc on peut faire les modifs en tenant compte de ces règles.

2 - Je suis en train de corriger avec les recommandations que William mais je me pose des questions pour des expressions comme celle-ci :  
e\_cross\_b\_im = compressed\_spec\_mat[(\*NB\_VALUES\_PER\_SPECTRAL\_MATRIX+17)\*k\_coeff\_intercalib[(\*NB\_K\_COEFF\_PER\_BIN+K34\_SX\_IM)] //S34 Re  
+ compressed\_spec\_mat[(\*NB\_VALUES\_PER\_SPECTRAL\_MATRIX+19)\*k\_coeff\_intercalib[(\*NB\_K\_COEFF\_PER\_BIN+K35\_SX\_IM)] //S35 Re  
+ compressed\_spec\_mat[(\*NB\_VALUES\_PER\_SPECTRAL\_MATRIX+5)\*k\_coeff\_intercalib[(\*NB\_K\_COEFF\_PER\_BIN+K14\_SX\_IM)] //S14 Re  
+ compressed\_spec\_mat[(\*NB\_VALUES\_PER\_SPECTRAL\_MATRIX+7)\*k\_coeff\_intercalib[(\*NB\_K\_COEFF\_PER\_BIN+K15\_SX\_IM)] //S15 Re  
+ compressed\_spec\_mat[(\*NB\_VALUES\_PER\_SPECTRAL\_MATRIX+12)\*k\_coeff\_intercalib[(\*NB\_K\_COEFF\_PER\_BIN+K24\_SX\_IM)] //S24 Re  
+ compressed\_spec\_mat[(\*NB\_VALUES\_PER\_SPECTRAL\_MATRIX+14)\*k\_coeff\_intercalib[(\*NB\_K\_COEFF\_PER\_BIN+K25\_SX\_IM)] //S25 Re  
- compressed\_spec\_mat[(\*NB\_VALUES\_PER\_SPECTRAL\_MATRIX+18)\*k\_coeff\_intercalib[(\*NB\_K\_COEFF\_PER\_BIN+K34\_SX\_RE)] //S34 Im  
- compressed\_spec\_mat[(\*NB\_VALUES\_PER\_SPECTRAL\_MATRIX+20)\*k\_coeff\_intercalib[(\*NB\_K\_COEFF\_PER\_BIN+K35\_SX\_RE)] //S35 Im  
- compressed\_spec\_mat[(\*NB\_VALUES\_PER\_SPECTRAL\_MATRIX+6)\*k\_coeff\_intercalib[(\*NB\_K\_COEFF\_PER\_BIN+K14\_SX\_RE)] //S14 Im  
- compressed\_spec\_mat[(\*NB\_VALUES\_PER\_SPECTRAL\_MATRIX+8)\*k\_coeff\_intercalib[(\*NB\_K\_COEFF\_PER\_BIN+K15\_SX\_RE)] //S15 Im  
- compressed\_spec\_mat[(\*NB\_VALUES\_PER\_SPECTRAL\_MATRIX+13)\*k\_coeff\_intercalib[(\*NB\_K\_COEFF\_PER\_BIN+K24\_SX\_RE)] //S24 Im  
- compressed\_spec\_mat[(\*NB\_VALUES\_PER\_SPECTRAL\_MATRIX+15)\*k\_coeff\_intercalib[(\*NB\_K\_COEFF\_PER\_BIN+K25\_SX\_RE)]; //S25 Im

En effet, la règle ne parle que des signes + et - mais en soit un signe + et un signe - devraient être traité de façon similaire, non ? En particulier pour l'expression ci-dessus?  $(a*b) + (c*d + (e*f) - (g*h) - (i*j) - (k*l))$  est-il acceptable par LOGISCOPE?

Merci

## #2 - 08/10/2018 01:17 PM - William Recart

point 2 : attention, il y a deux soucis :

2.1 : les opérations dans les index;

2.2 : les parenthèses des multiplications

Pour répondre à la question,  $(a*b) + (c*d) + (e*f) - (g*h) - (i*j) - (k*l)$  n'est pas accepté par Logiscope. (la règle parle bien d'un même opérateur)

Pour preuve, cette correction de la ligne 295 qui lève toujours des erreurs :

```
e_cross_b_im =
(compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+17]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K34_SX_IM]) //S34 Re
+ (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+19]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K35_SX_IM]) //S35 Re
+ (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+5]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K14_SX_IM]) //S14 Re
+ (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+7]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K15_SX_IM]) //S15 Re
+ (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+12]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K24_SX_IM]) //S24 Re
+ (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+14]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K25_SX_IM]) //S25 Re
- (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+18]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K34_SX_RE]) //S34 Im
- (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+20]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K35_SX_RE]) //S35 Im
- (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+6]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K14_SX_RE]) //S14 Im
- (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+8]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K15_SX_RE]) //S15 Im
- (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+13]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K24_SX_RE]) //S24 Im
- (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+15]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K25_SX_RE]); //S25 Im
```

La correction stricte qui ne lève plus d'erreur est la suivante :  $(a*b) + (c*d) + (((e*f) - (g*h) - (i*j) - (k*l))$  :

```
e_cross_b_im =
(compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+17]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K34_SX_IM]) //S34 Re
+ (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+19]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K35_SX_IM]) //S35 Re
+ (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+5]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K14_SX_IM]) //S14 Re
+ (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+7]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K15_SX_IM]) //S15 Re
+ (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+12]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K24_SX_IM]) //S24 Re
+ ((((((compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+14]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K25_SX_IM])
//S25 Re
- (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+18]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K34_SX_RE])) //S34 Im
- (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+20]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K35_SX_RE])) //S35 Im
- (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+6]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K14_SX_RE])) //S14 Im
- (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+8]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K15_SX_RE])) //S15 Im
- (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+13]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K24_SX_RE])) //S24 Im
- (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+15]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K25_SX_RE])); //S25 Im
```

La correction suivante ne lève plus d'erreur non plus car les signes dans les parenthèse sont les mêmes:  $(a*b) + (c*d) + ((e*f) - (g*h) - (i*j) - (k*l))$  :

(on peut donc quand même simplifier le parenthésage)

```
e_cross_b_im =
(compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+17]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K34_SX_IM]) //S34 Re
+ (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+19]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K35_SX_IM]) //S35 Re
+ (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+5]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K14_SX_IM]) //S14 Re
+ (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+7]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K15_SX_IM]) //S15 Re
+ (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+12]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K24_SX_IM]) //S24 Re
+ ((compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+14]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K25_SX_IM]) //S25 Re
- (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+18]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K34_SX_RE]) //S34 Im
- (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+20]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K35_SX_RE]) //S35 Im
- (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+6]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K14_SX_RE]) //S14 Im
- (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX)+8]*k_coeff_intercalib[(*NB_K_COEFF_PER_BIN)+K15_SX_RE]) //S15 Im
```

- (compressed\_spec\_mat[(i\*NB\_VALUES\_PER\_SPECTRAL\_MATRIX)+13]\*k\_coeff\_intercalib[(i\*NB\_K\_COEFF\_PER\_BIN)+K24\_SX\_RE]) //S24 Im  
- (compressed\_spec\_mat[(i\*NB\_VALUES\_PER\_SPECTRAL\_MATRIX)+15]\*k\_coeff\_intercalib[(i\*NB\_K\_COEFF\_PER\_BIN)+K25\_SX\_RE]); //S25 Im

### #3 - 08/10/2018 02:15 PM - bruno katra

Mes commentaires en interligne.

William Recart wrote:

point 2 : attention, il y a deux soucis :  
2.1 : les opérations dans les index;  
2.2 : les parenthèses des multiplications

Tout à fait mais c'est le point 2.2 qui me laisse perplexe pour cet exemple

Pour répondre à la question,  $(a*b) + (c*d) + (e*f) - (g*h) - (i*j) - (k*l)$  n'est pas accepté par Logiscope. (la règle parle bien d'un même opérateur)

Pour preuve, cette correction de la ligne 295 qui lève toujours des erreurs :

```
e_cross_b_im =  
(compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+17]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K34_SX_IM]) //S34  
Re  
+ (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+19]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K35_SX_IM])  
//S35 Re  
+ (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+5]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K14_SX_IM]) //S14  
Re  
+ (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+7]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K15_SX_IM]) //S15  
Re  
+ (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+12]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K24_SX_IM])  
//S24 Re  
+ (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+14]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K25_SX_IM])  
//S25 Re  
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+18]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K34_SX_RE])  
//S34 Im  
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+20]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K35_SX_RE])  
//S35 Im  
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+6]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K14_SX_RE]) //S14  
Im  
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+8]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K15_SX_RE]) //S15  
Im  
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+13]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K24_SX_RE])  
//S24 Im  
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+15]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K25_SX_RE]);  
//S25 Im
```

La correction stricte qui ne lève plus d'erreur est la suivante :  $(a*b) + (c*d) + (((e*f) - (g*h)) - (i*j)) - (k*l)$  :

```
e_cross_b_im =  
(compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+17]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K34_SX_IM]) //S34  
Re  
+ (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+19]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K35_SX_IM])  
//S35 Re  
+ (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+5]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K14_SX_IM]) //S14  
Re  
+ (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+7]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K15_SX_IM]) //S15  
Re  
+ (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+12]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K24_SX_IM])  
//S24 Re  
+ ((((((compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+14]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K25_SX_IM])  
//S25 Re  
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+18]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K34_SX_RE])  
//S34 Im  
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+20]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K35_SX_RE])  
//S35 Im  
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+6]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K14_SX_RE])  
//S14 Im  
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+8]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K15_SX_RE])  
//S15 Im  
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+13]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K24_SX_RE])  
//S24 Im  
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+15]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K25_SX_RE]));  
//S25 Im
```

C'est bien ce que je craignais, d'où ma question. Dans ce cas la correction stricte est absurde : il n'y a pas d'ambiguïté dans une suite d'addition et de soustraction et la multiplication des parenthèses tuent complètement la lisibilité.

La correction suivante ne lève plus d'erreur non plus car les signes dans les parenthèse sont les mêmes: (a\*b) + (c\*d) + ((e\*f) - (g\*h) - (i\*j) - (k\*l)) : (on peut donc quand même simplifier le parenthésage)

```
e_cross_b_im =
(compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+17]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K34_SX_IM]) //S34
Re
+ (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+19]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K35_SX_IM])
//S35 Re
+ (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+5] *k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K14_SX_IM]) //S14
Re
+ (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+7] *k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K15_SX_IM]) //S15
Re
+ (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+12]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K24_SX_IM])
//S24 Re
+ ((compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+14]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K25_SX_IM])
//S25 Re
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+18]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K34_SX_RE])
//S34 Im
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+20]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K35_SX_RE])
//S35 Im
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+6] *k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K14_SX_RE]) //S14
Im
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+8] *k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K15_SX_RE]) //S15
Im
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+13]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K24_SX_RE])
//S24 Im
- (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+15]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K25_SX_RE]));
//S25 Im
```

Ok merci : en effet là on y voit plus clair. Mais il manque une parenthèse avant le "//S25 Re" non?

#### #4 - 08/10/2018 02:48 PM - William Recart

Sur le dernier point, toutes les parenthèses sont présentes et il n'en manque pas (c'est un copier/coller de ce qui fonctionne)  
En gros, tous les "+" n'ont pas de parenthèses et tous les "-" sont dans une seule parenthèse.

#### #5 - 08/10/2018 03:16 PM - bruno katra

William Recart wrote:

Sur le dernier point, toutes les parenthèses sont présentes et il n'en manque pas (c'est un copier/coller de ce qui fonctionne)  
En gros, tous les "+" n'ont pas de parenthèses et tous les "-" sont dans une seule parenthèse.

Pour moi il manque la parenthèse fermante de la multiplication . D'ailleurs QtCreator me lève une erreur de parenthèse pour l'expression... étrange

```
+ (( compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+14]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K25_SX_IM]) //S25 Re
```

#### #6 - 08/10/2018 03:28 PM - William Recart

Effectivement, le copier/coller est faux sur cette ligne :

```
+ ((compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX)+14]*k_coeff_intercalib[(i*NB_K_COEFF_PER_BIN)+K25_SX_IM]) //S25 Re
```

J'ai répondu un peu vite, ce n'est pas le même code que j'avais dans Logiscope....

#### #7 - 08/10/2018 04:50 PM - bruno katra

Corrigé, en attente de validation Thomas.

#### #8 - 08/10/2018 09:47 PM - bruno katra

- File *basic\_parameters.h* added

- Assignee changed from *thomas chust* to *William Recart*

Thomas a validé que les sorties du programme sont les mêmes après les modifs LOGISCOPE. William a proposé de faire une passe rapide sur *basic\_parameters.h* pour être sûr que tout est bien corrigé avant de tout recompiler.  
Le fichier corrigé est en PJ.

#### #9 - 09/10/2018 10:47 AM - William Recart

Il reste une erreur ligne suivante :

221 : multiplication à mettre entre parenthèse : proposition de correction :

```
tmp = sqrt( (compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX) + 2]
*compressed_spec_mat[(i*NB_VALUES_PER_SPECTRAL_MATRIX) + 2]) //Im S12
```

```
+(compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX) + 4] *compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX) + 4]) //Im S13
+(compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX) + 11]*compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX) + 11]) //Im S23
);
```

**#10 - 09/10/2018 01:33 PM - bruno katra**

- File deleted (*basic\_parameters.h*)

**#11 - 09/10/2018 01:36 PM - bruno katra**

- File *basic\_parameters.h* added

- Assignee changed from *William Recart* to *thomas chust*

William Recart wrote:

Il reste une erreur ligne suivante :

221 : multiplication à mettre entre parenthèse : proposition de correction :

```
tmp = sqrt( (compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX) + 2]
*compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX) + 2]) //Im S12
+ 4) *compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX) + 4]) //Im S13
(compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX) +
11]*compressed_spec_mat[(*NB_VALUES_PER_SPECTRAL_MATRIX) + 11]) //Im S23
);
```

Mince! Ca m'apprendra à faire 2 choses en même temps... *sad.png* C'est corrigé, j'ai remplacé le fichier en PJ par celui corrigé... Thomas pourrais-tu remplacer le fichier livré hier par celui-ci et faire un push sur le dépôt. Ainsi Alexis pour l'amender et l'intégrer au dépôt parent. J'en ai profité hier pour ajouter les entêtes GPL dans tes fichiers...

**#12 - 10/10/2018 10:08 AM - William Recart**

- Status changed from *In Progress* to *Resolved*

Le nouveau fichier a été analysé et l'erreur n'apparaît plus.  
Cette issue sera close à la prochaine analyse complète.

**#13 - 10/10/2018 10:57 AM - Alexis Jeandet**

- Parent task set to *#3120*

**#14 - 10/10/2018 11:05 AM - thomas chust**

Voilà j'ai revérifier hier avec le dernier fichier envoyé par Bruno et là je viens de pusher sur mon dépôt cette dernière version correspondante des fichiers:

[https://chust@hephaistos.lpp.polytechnique.fr/rhocode/HG\\_REPOSITORIES/LPP/INSTRUMENTATION/USERS/CHUST/LFR\\_basic-parameters](https://chust@hephaistos.lpp.polytechnique.fr/rhocode/HG_REPOSITORIES/LPP/INSTRUMENTATION/USERS/CHUST/LFR_basic-parameters).

Alexis à toi de jouer si je comprends bien ... *smile.png*

**#15 - 10/10/2018 11:06 AM - thomas chust**

- Assignee changed from *thomas chust* to *Alexis Jeandet*

**#16 - 10/12/2018 03:39 PM - William Recart**

- Status changed from *Resolved* to *Closed*

Corrected in source code V3.2.0.24

## Files

---

basic_parameters.h	65.3 KB	09/10/2018	bruno katra
--------------------	---------	------------	-------------