

# Readme by MV

Mirko Vukovic

December 25, 2016

## Contents

<b>1</b>	<b>GSLM to load</b>	<b>2</b>
1.1	Teach GSLM to add MSYS2's GSL DLL's . . . . .	2
<b>2</b>	<b>GSLM Loading errors</b>	<b>3</b>
2.1	<b>DONE</b> Compilation failed in /data/permutation.lisp . . . . .	3
2.2	<b>DONE</b> Compilation style warnings in /special-functions/return-structures.lisp . . . . .	3
2.3	<b>DONE</b> Compilation failed in /solve-minimize-fit/linear-least-squares.lisp . . . . .	3
2.4	<b>DONE</b> Compilation style warnings in /solve-minimize-fit/nonlinear-least-squares.lisp . . . . .	4
<b>3</b>	<b>GSLM Bugs</b>	<b>5</b>
3.1	Bug in (grid:copy permutation) in /data/permutation.lisp	5
<b>4</b>	<b>GSLM Test errors</b>	<b>5</b>
4.1	General . . . . .	5
4.2	CCL crashes . . . . .	5
4.3	Summary of test errors . . . . .	6
4.3.1	<b>DONE</b> Bug in (grid:copy permutation) in /data/permutation.lisp	6
4.3.2	<b>DONE</b> /linear-algebra/lu.lisp . . . . .	6
4.3.3	<b>DONE</b> /calculus/numerical-integration.lisp . . . . .	7
4.3.4	<b>DONE</b> /special-functions/legendre.lisp and /tests/legendre.lisp	7
4.3.5	<b>DONE</b> /sorting.lisp . . . . .	7
4.3.6	<b>DONE</b> Elliptic integrals . . . . .	8
4.3.7	<b>DONE</b> /solve-minimize-fit/minimization-multi.lisp . . . . .	8

Port of master branch to windows 10 & MSYS2 with GSL2.1

Data:

- master is at 0caa036
- `gsl-config --version` returns 2.1
- CCL version is (lisp-implementation-version) "Version 1.11-r16635 (WindowsX8664)"

I proceed in three steps:

- Enable GSSL to load
- Address warnings and errors in all compilation and loading steps as these may cause tests to fail
- Address test errors

## 1 GSSL to load

### 1.1 Teach GSSL to add MSYS2's GSL DLL's

Error in `/init/init.lisp`

Unable to load any of the alternatives:

```
("libgsl-0.dll" "cyggsl-0.dll")
[Condition of type CFFI:LOAD-FOREIGN-LIBRARY-ERROR]
```

To enable GSSL to access MSYS2's GSL DLL's I need to do the following:

- Add to location of DLL's to the Windows Environment Variable PATH. For example, I have `C:\msys64\usr\bin` and `C:\msys64\mingw64\bin` in my PATH
- Add `*msys2-mingw64*` to `:FEATURES`

Finally modify library definitions in `/init/init.lisp`.

- To `(cffi:define-foreign-library libgslcblas add (:msys2-mingw64 "libgslcblas-0.dll"))`
- To `(cffi:define-foreign-library libgsl add (:msys2-mingw64 "libgsl-19.dll"))`

## 2 GSSL Loading errors

### 2.1 DONE Compilation failed in /data/permutation.lisp

- State "DONE" from [2016-12-24 Sat 12:03]  
Added comments to code that the warnings about duplicate function definitions can be safely ignored

```
;Compiler warnings for "c:/mv-program-files/my-lisp-libraries/gssl/data/permutation.lisp"
; In an anonymous lambda form at position 1290: Duplicate definitions of (:METHOD :A)
; In an anonymous lambda form at position 1631: Duplicate definitions of MAKE-PERMUTATION
; Warning: Lisp compilation failed while compiling #<CL-SOURCE-FILE "gssl" "data" "permutation.lisp">
; While executing: UIOP/LISP-BUILD:CHECK-LISP-COMPILE-WARNINGS, in process repl-thread
; Warning: Lisp compilation had style-warnings while compiling #<CL-SOURCE-FILE "gssl" "data" "permutation.lisp">
```

### 2.2 DONE Compilation style warnings in /special-functions/return-structures.lisp

- State "DONE" from [2016-12-24 Sat 12:03]  
Added declarations to suppress warnings about unused variables

```
;Compiler warnings for "c:/mv-program-files/my-lisp-libraries/gssl/special-functions/return-structures.lisp"
; In (TRANSLATE-FROM-FOREIGN (T SF-RESULT-TCLASS)) inside an anonymous lambda form: UNDECLARED-FREE-VARIABLE
; In (TRANSLATE-FROM-FOREIGN (T SF-RESULT-E10-TCLASS)) inside an anonymous lambda form: UNDECLARED-FREE-VARIABLE
; Warning: Lisp compilation had style-warnings while compiling #<CL-SOURCE-FILE "gssl" "special-functions/return-structures.lisp">
```

### 2.3 DONE Compilation failed in /solve-minimize-fit/linear-least-squares.lisp

- State "DONE" from [2016-12-25 Sun 10:04]

Error message:

```
;Compiler warnings for "c:/mv-program-files/my-lisp-libraries/gssl/solve-minimize-fit/linear-least-squares.lisp"
; In LINEAR-MFIT-SVD: Undeclared free variable CHISQ
; In LINEAR-MFIT-SVD: Undeclared free variable RANK
; Warning: Lisp compilation failed while compiling #<CL-SOURCE-FILE "gssl" "solve-minimize-fit/linear-least-squares.lisp">
; While executing: UIOP/LISP-BUILD:CHECK-LISP-COMPILE-WARNINGS, in process repl-thread
; Warning: Lisp compilation had style-warnings while compiling #<CL-SOURCE-FILE "gssl" "solve-minimize-fit/linear-least-squares.lisp">
```

Definition of `linear-mfit-svd` was being incorrectly expanded by the `defmfun` macro. The code was supposed to expand based on `weight` and the expansion was done incorrectly. Instead I did a manual expansion and the file compiles cleanly.

## 2.4 DONE Compilation style warnings in /solve-minimize-fit/nonlinear-least-squares.lisp

- State "DONE" from [2016-12-25 Sun 10:04]  
Improved interface to Jacobian and fixed ls-covariance bug

Error message:

```
;Compiler warnings for "c:/mv-program-files/my-lisp-libraries/gsl1/solve-minimize-fit/  
; In LS-COVARIANCE: In the call to JACOBIAN with arguments (SOLVER),  
; 1 argument was provided, but at least 2 are required  
; by the definition of JACOBIAN visible in the current compilation unit.  
; Warning: Lisp compilation had style-warnings while compiling #<CL-SOURCE-FILE "gsl1"
```

In the definition of ls-covariance a call is made to the Jacobian solver with two errors in it:

- Incorrect number of arguments
- Incorrect handling of the returned value (needs a pointer, not foreign array)

The new definition of the Jacobian requires a second argument, the matrix in which the result will be returned. I modified that by making the second argument optional, and properly initializing it like so:

```
(defmfun jacobian (solver &optional  
  (matrix (grid:make-foreign-array  
'double-float  
:dimensions (list (dim0 solver) (dim1 solver))))  
  ...)
```

In ls-covariance, the code needs the pointer to the Jacobian, not the foreign array itself. I fixed that by applying mpointer to the Jacobian like so:

```
(defmfun ls-covariance  
  (solver relative-error &optional covariance  
&aux (cov (or covariance  
  (grid:make-foreign-array 'double-float  
  :dimensions  
  (list (dim1 solver) (dim1 solver))))))  
  "gsl_multifit_covar"  
  ((mpointer (jacobian solver)) :pointer) ;; mpointer is new  
  ...))
```

The unit test works but still signals an error as it does not meet the required number of significant digits:

```
| Failed Form: (MULTIPLE-VALUE-LIST (NONLINEAR-LEAST-SQUARES-EXAMPLE 40 +LEVENBERG-MA
| Expected ((5.045357801443204D0 0.10404905892045835D0 1.0192487061031013D0))
| but saw ((5.045357787875535D0 0.10404905908953155D0 1.0192487172454892D0))
|
NONLINEAR-LEAST-SQUARES: 0 assertions passed, 1 failed.
```

### 3 GSLL Bugs

#### 3.1 Bug in (grid:copy permutation) in /data/permutation.lisp

The method `grid:copy` on a permutation calls a next-method if `grid` is specified. But there is no next method

Bug was uncovered while looking at compilation errors for this file

### 4 GSLL Test errors

I investigate execution errors in tests.

#### 4.1 General

I limit the number of tries in random number generator tests. Otherwise, some tests did not terminate (within the bounds of my patience).

```
(let ((*pdf-number-of-tries* 3))
  (asdf:test-system :gsll))
```

#### 4.2 CCL crashes

Irreproducably, sometimes CCL crashes, sometimes not. Occasionally, the inferior-lisp buffer will contain a traceback.

One way to try to isolate the cause of the crash is to run CCL from the command line, and trace the calls to `lisp-unit`'s internal function `run-test-thunk`. The last entry in this trace pointed to `laplace pdf` as a cause for the CCL crash. Then the way to run tests but that one is as follows:

```
(asdf:load-local-system :gsll)
(trace lu::run-test-thunk)
(in-package :gsll)
```

```
(lisp-unit:remove-tests '(laplace))
(let ((*pdf-number-of-tries* 3)) (asdf:test-system :gsl1))
```

From within REPL, the code is

```
(let ((*pdf-number-of-tries* 3))
  (lisp-unit:remove-tests '(laplace))
  (asdf:test-system :gsl1))
```

However, at a later date, all the tests were running (including laplace-pdf) without crashing. This issue of CCL crashing is not resolved.

### 4.3 Summary of test errors

#### 4.3.1 DONE Bug in (grid:copy permutation) in /data/permutation.lisp

- State "DONE" from *[2016-12-25 Sun 16:32]*  
I cannot fix the bug at this time - I don't understand the purpose of call-next-method

The copy permutation calls undefined CALL-NEXT-METHOD

```
| Execution error:
| There is no next method for #<STANDARD-METHOD COPY (PERMUTATION)>
args: (#<PERMUTATION size 4 contents (0 1 2 3)> :GRID-TYPE ARRAY :ELEMENT-TYPE FIXNUM)
|
PERMUTATION: 1 assertions passed, 0 failed, and an execution error.
```

See discussion in Bug in (grid:copy permutation) in /data/permutation.lisp

#### 4.3.2 DONE /linear-algebra/lu.lisp

- State "DONE" from *[2016-12-25 Sun 16:35]*  
This will require a fix in the /tests/lu.lisp. I need to set grid:\*default-element-type\* to (complex double-float)

The last test (test-lu-solve-dim (create-complex-matrix 7)) returns a complex array, on which an error is signaled

```
| Execution error:
| The value #C(24.0717272023734D0 -9.84612797621247D0) is not of the expected type RE.
|
LU: 8 assertions passed, 0 failed, and an execution error.
```

### 4.3.3 DONE /calculus/numerical-integration.lisp

- State "DONE" from [2016-12-25 Sun 11:34]  
Done for now, although the number of failed tests leaves me uneasy about the correctness of the code.

```
| Execution error:  
| Exceeded max number of iterations; maximum number of subdivisions reached in ../../  
|  
NUMERICAL-INTEGRATION: 0 assertions passed, 0 failed, and an execution error.
```

Fixed error in two tests by supplying a double float as required. Updated comments on other tests that fail.

Annotated all errors signaled by GSL. Most tests fail by failing to reach the tolerance or exceeding the number of iterations (specified as 1000)

Improved one test robustness by using `grid:copy` instead of `grid:copy-to`

I could factor the tests into expected and unexpected errors. I tried that but the changes did not register in GSSL test suite, even after recompiling GSSL.

### 4.3.4 DONE /special-functions/legendre.lisp and /tests/legendre.lisp

- State "DONE" from [2016-12-25 Sun 16:14]  
Commented out tests that call functions depreciated in GSL2

```
;Compiler warnings :  
; In an anonymous lambda form inside an anonymous lambda form: Undefined function LE  
; In an anonymous lambda form inside an anonymous lambda form: Undefined function LE  
; In an anonymous lambda form inside an anonymous lambda form: Undefined function LE  
; In an anonymous lambda form inside an anonymous lambda form: Undefined function LE  
| Execution error:  
| Undefined function LEGENDRE-PLM-ARRAY called with arguments (2 0.5D0 #m(0.000000000  
|  
LEGENDRE: 11 assertions passed, 0 failed, and an execution error.
```

Test is calling depreciated functions whose replacement has not been provided. I will only comment out the tests in both `legendre.lisp` and `tests/legendre.lisp`

### 4.3.5 DONE /sorting.lisp

- State "DONE" from [2016-12-25 Sun 16:14]  
Root cause is bug (missing next method) in method `grid:copy permutation`

```
| Execution error:
| There is no next method for #<STANDARD-METHOD COPY (PERMUTATION)>
args: (#<PERMUTATION size 8 contents (0 7 3 5 2 1 4 6)> :GRID-TYPE ARRAY :ELEMENT-TYPE
|
SORT-VECTOR-INDEX: 0 assertions passed, 0 failed, and an execution error.
```

This one fails because of faulty definition of method COPY for permutation

#### 4.3.6 DONE Elliptic integrals

- State "DONE" from *[2016-12-25 Sun 15:35]*  
Fixed error in function call by removing the extra parameter

```
| Execution error:
| The value 1.0D0 is not of the expected type (UNSIGNED-BYTE 32).
|
ELLIPTIC-INTEGRALS: 5 assertions passed, 0 failed, and an execution error.
```

#### 4.3.7 DONE /solve-minimize-fit/minimization-multi.lisp

- State "DONE" from *[2016-12-25 Sun 16:14]*  
I was not able to find the root cause of this error

```
| Execution error:
| Iteration is not making progress towards solution; in ITERATE
|
MINIMIZATION-MULTI: 1 assertions passed, 0 failed, and an execution error.
```

I could not trace the source of the error.

- I manually ran all the code in the tests and none of them triggered an error
- I traced `lisp-unit:run-test-thunk` and `lisps-unit:run-code`. That gave me no insight
- The error message comes from the `no-progress +enopgor+` condition in `/init/conditions.lisp`
- The only place where that condition is used in in the definition of `+powells-hybrid+`

- `+powells-hybrid+` is used only in `/solve-minimize-fit/roots-multi.lisp`  
and in `/tests/roots-multi.lisp`
- I do not see how it is being invoked in the failing tests