

GammaLib - OpenMP - # 3

{{lastupdated_at}} by {{lastupdated_by}}

OpenMP

OpenMp is the API used by GammaLib to paralellize some parts of the code.

IMPORTANT: if GammaLib is compiled with parallelization, openmp flag is required (-fopenmp with GCC) for C++ code using it.

Else errors like this will appear:

```
/libgamma.so: undefined reference to `GOMP_critical_end'
/libgamma.so: undefined reference to `omp_get_thread_num'
/libgamma.so: undefined reference to `omp_get_num_threads'
/libgamma.so: undefined reference to `GOMP_sections_start'
/libgamma.so: undefined reference to `GOMP_parallel_start'
/libgamma.so: undefined reference to `GOMP_sections_end'
/libgamma.so: undefined reference to `GOMP_critical_start'
/libgamma.so: undefined reference to `GOMP_parallel_end'
/libgamma.so: undefined reference to `GOMP_sections_next'
/libgamma.so: undefined reference to `GOMP_barrier'
```

documentation :

<http://www.openmp.org/mp-documents/spec25.pdf>

<http://en.wikipedia.org/wiki/OpenMP>

<http://software.intel.com/en-us/articles/getting-started-with-openmp/>

Configuration

It is possible to configure paralellization with environment variable.

The usefull is OMP_NUM_THREADS which sets the maximum number of threads to use during execution. For example:
export OMP_NUM_THREADS=8

To see all environment variables : <https://computing.llnl.gov/tutorials/openMP/#EnvironmentVariables>