C interfaces to GALAHAD CONVERT

Jari Fowkes and Nick Gould
STFC Rutherford Appleton Laboratory
Wed May 3 2023

Chapter 1

GALAHAD C package convert

1.1 Introduction

1.1.1 Purpose

Given a real matrix A stored in one format, convert it to another

Currently, only the control and inform parameters are exposed; these are provided and used by other GALAHAD packages with C interfaces.

1.1.2 Authors

N. I. M. Gould, STFC-Rutherford Appleton Laboratory, England.

C interface, additionally J. Fowkes, STFC-Rutherford Appleton Laboratory.

Julia interface, additionally A. Montoison and D. Orban, Polytechnique Montréal.

1.1.3 Originally released

June 2014, C interface February 2022.

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

File Documentation

3.1 galahad_convert.h File Reference

```
#include <stdbool.h>
#include <stdint.h>
#include "galahad_precision.h"
#include "galahad_cfunctions.h"
```

Data Structures

- struct convert_control_type
- struct convert_time_type
- struct convert_inform_type

Functions

- void convert_initialize (void **data, struct convert_control_type *control, int *status)
- void convert_information (void **data, struct convert_inform_type *inform, int *status)
- void convert_terminate (void **data, struct convert_control_type *control, struct convert_inform_type *inform)

3.1.1 Data Structure Documentation

3.1.1.1 struct convert_control_type

control derived type as a C struct

Data Fields

bool	f_indexing	use C or Fortran sparse matrix indexing
int	error	unit for error messages
int	out	unit for monitor output
int	print_level	controls level of diagnostic output

Data Fields

bool	transpose	obtain the transpose of the input matrix?
bool	sum_duplicates	add the values of entries in duplicate positions?
bool	order	order row or column data by increasing index?
bool	space_critical	if space is critical, ensure allocated arrays are no bigger than needed
bool	deallocate_error_fatal	exit if any deallocation fails
char	prefix[31]	all output lines will be prefixed by prefix(2:LEN(TRIM(.prefix))-1) where prefix contains the required string enclosed in quotes, e.g. "string" or 'string'

3.1.1.2 struct convert_time_type

time derived type as a C struct

Data Fields

real_wp_	total	total cpu time spent in the package
real_wp_	clock_total	total clock time spent in the package

3.1.1.3 struct convert_inform_type

inform derived type as a C struct

Data Fields

int	status	the return status. Possible values are: • 0 a successful conversion. • -1. An allocation error occurred. A message indicating the offending array is written on unit control.error, and the returned allocation status and a string containing the name of the offending array are held in inform.alloc_status and inform.bad_alloc respectively. • -2. A deallocation error occurred. A message indicating the offending array is written on unit control.error and the returned allocation status and a string containing the name of the offending array are held in inform.alloc_status and inform.bad_alloc respectively. • -3. The restriction n $>$ 0 or m $>$ 0 or requirement that a type contains its relevant string 'coordinate', 'sparse_by_rows', 'sparse_by_columns', 'dense_by_rows' or 'dense_by_columns' has been violated. • -32 provided integer workspace is not large enough. • -33 provided real workspace is not large enough. • -73 an input matrix entry has been repeated. • -79 there are missing optional arguments. • -90 a requested output format is not recognised.
int	alloc_status	the status of the last attempted allocation/deallocation.
int	duplicates	the number of duplicates found (-ve = not checked).
char	bad_alloc[81]	the name of the array for which an allocation/deallocation error occurred.
struct convert_time_type	time	timings (see above).

3.1.2 Function Documentation

3.1.2.1 convert_initialize()

```
void convert_initialize (
            void ** data,
            struct convert_control_type * control,
            int * status )
```

Set default control values and initialize private data

Parameters

in,out	*data*	holds private internal data
out	*control*	is a struct containing control information (see convert_control_type)

Parameters

out	*status*	is a scalar variable of type int, that gives the exit status from the package. Possible values are (currently): • 0. The initialization was succesful.

3.1.2.2 convert_information()

```
void convert_information (
            void ** data,
            struct convert_inform_type * inform,
            int * status )
```

Provides output information

Parameters

in,out	*data*	holds private internal data
out	*inform*	is a struct containing output information (see convert_inform_type)
out	*status*	is a scalar variable of type int, that gives the exit status from the package. Possible values are (currently): • 0. The values were recorded succesfully

3.1.2.3 convert_terminate()

```
void convert_terminate (
            void ** data,
            struct convert_control_type * control,
            struct convert_inform_type * inform )
```

Deallocate all internal private storage

Parameters

in,out	*data*	holds private internal data
out	*control*	is a struct containing control information (see convert_control_type)
out	*inform*	is a struct containing output information (see convert_inform_type)