C interfaces to GALAHAD SHA

Jari Fowkes and Nick Gould
STFC Rutherford Appleton Laboratory
Thu Jun 22 2023

1 GALAHAD C package sha **1**

1.1 Introduction . . . 1

1.1.1 Purpose . . . 1

1.1.2 Authors . . . 1

1.1.3 Originally released . . . 1

2 File Index **3**

2.1 File List . . . 3

3 File Documentation **5**

3.1 galahad_sha.h File Reference . . . 5

3.1.1 Data Structure Documentation . . . 5

3.1.1.1 struct sha_control_type . . . 5

3.1.1.2 struct sha_inform_type . . . 6

3.1.2 Function Documentation . . . 6

3.1.2.1 sha_initialize() . . . 6

3.1.2.2 sha_information() . . . 7

3.1.2.3 sha_terminate() . . . 7

Chapter 1

GALAHAD C package sha

1.1 Introduction

1.1.1 Purpose

Find an approximation to a sparse Hessian using componentwise secant approximation.

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

April 2013, C interface January 2022.

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

galahad_sha.h 5

Chapter 3

File Documentation

3.1 galahad_sha.h File Reference

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

Data Structures

- struct sha_control_type
- struct sha_inform_type

Functions

- void sha_initialize (void **data, struct sha_control_type *control, int *status)
- void sha_information (void **data, struct sha_inform_type *inform, int *status)
- void sha_terminate (void **data, struct sha_control_type *control, struct sha_inform_type *inform)

3.1.1 Data Structure Documentation

3.1.1.1 struct sha_control_type

control derived type as a C struct

Data Fields

bool	f_indexing	use C or Fortran sparse matrix indexing
int	error	error and warning diagnostics occur on stream error
int	out	general output occurs on stream out
int	print_level	the level of output required. <= 0 gives no output, = 1 gives a one-line summary for every iteration, = 2 gives a summary of the inner iteration for each iteration, >= 3 gives increasingly verbose (debugging) output

Data Fields

int	approximation_algorithm	which approximation algorithm should be used? • 0 : unsymmetric (alg 2.1 in paper) • 1 : symmetric (alg 2.2 in paper) • 2 : composite (alg 2.3 in paper) • 3 : composite 2 (alg 2.2/3 in paper)
int	dense_linear_solver	which dense linear equation solver should be used? • 1 : Gaussian elimination • 2 : QR factorization • 3 : singular-value decomposition • 4 : singular-value decomposition with divide-and-conquer
int	max_sparse_degree	the maximum sparse degree if the combined version is used
int	extra_differences	if available use an addition extra_differences differences
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 sha_inform_type

inform derived type as a C struct

Data Fields

int	status	return status. See SHA_solve for details
int	alloc_status	the status of the last attempted allocation/deallocation.
int	max_degree	the maximum degree in the adgacency graph.
int	differences_needed	the number of differences that will be needed.
int	max_reduced_degree	the maximum reduced degree in the adgacency graph.
int	bad_row	a failure occured when forming the bad_row-th row (0 = no failure).
char	bad_alloc[81]	the name of the array for which an allocation/deallocation error occurred.

3.1.2 Function Documentation

3.1.2.1 sha_initialize()

```
void sha_initialize (
             void ** data,
```

```
            struct sha_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 sha_control_type)
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 sha_information()

```
void sha_information (
            void ** data,
            struct sha_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 sha_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 sha_terminate()

```
void sha_terminate (
            void ** data,
            struct sha_control_type * control,
            struct sha_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 sha_control_type)
out	*inform*	is a struct containing output information (see sha_inform_type)