

TrgSum

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Project "Trigonometrix sums"</b>	<b>1</b>
<b>2</b>	<b>Bug List</b>	<b>3</b>
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>File Documentation</b>	<b>7</b>
4.1	trgsum.c File Reference . . . . .	7
4.1.1	Detailed Description . . . . .	8
4.1.2	Function Documentation . . . . .	8
4.1.2.1	trg_d_s_goertzel() . . . . .	8
4.1.2.2	trg_d_s_reinsch() . . . . .	8
4.1.2.3	trg_d_v_goertzel() . . . . .	10
4.1.2.4	trg_d_v_reinsch() . . . . .	10
4.2	trgsum.h File Reference . . . . .	11
4.2.1	Detailed Description . . . . .	11
4.2.2	Function Documentation . . . . .	11
4.2.2.1	trg_d_s_goertzel() . . . . .	12
4.2.2.2	trg_d_s_reinsch() . . . . .	13
4.2.2.3	trg_d_v_goertzel() . . . . .	13
4.2.2.4	trg_d_v_reinsch() . . . . .	14
	<b>Index</b>	<b>15</b>



## Chapter 1

# Project "Trigonometrix sums"

Author

Przemysław Stpiczynski

The package contains sequential and vectorized implementations of Goertzel and Reinsch algorithms for finding trigonometric sums:

$$C(x) = b_0 \cos(0) + b_1 \cos(x) + b_2 \cos(2x) + \dots + b_n \cos(nx)$$

$$S(x) = + b_1 \sin(x) + b_2 \sin(2x) + \dots + b_n \sin(nx)$$



## Chapter 2

# Bug List

File [trgsum.c](#)

No know bugs.

File [trgsum.h](#)

No known bugs.





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">trgsum.c</a>	Implementation of the package . . . . .	<a href="#">7</a>
<a href="#">trgsum.h</a>	Function prototypes for the package "Trigonometric sums" . . . . .	<a href="#">11</a>



## Chapter 4

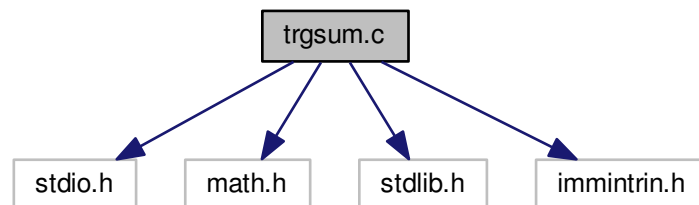
# File Documentation

### 4.1 trgsun.c File Reference

Implementation of the package.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include "immintrin.h"
```

Include dependency graph for trgsun.c:



### Functions

- `__inline void vec8x8_transpose (__m512d *col0, __m512d *col1, __m512d *col2, __m512d *col3, __m512d *col4, __m512d *col5, __m512d *col6, __m512d *col7)`  
*Implementation of basic sequential Goertzel algorithm.*
- void [trg\\_d\\_s\\_goertzel](#) (double x, int n, double \*b, double \*cx, double \*sx)  
*Implementation of basic sequential Goertzel algorithm.*
- void [trg\\_d\\_v\\_goertzel](#) (double x, int n, double \*b, double \*cx, double \*sx)  
*Implementation of vectorized Goertzel algorithm introduced by Stpiczynski.*
- void [trg\\_d\\_s\\_reinsch](#) (double x, int n, double \*b, double \*cx, double \*sx)  
*Implementation of basic sequential Reinsch algorithm.*
- void [trg\\_d\\_v\\_reinsch](#) (double x, int n, double \*b, double \*cx, double \*sx)  
*Implementation of vectorized Reinsch algorithm introduced by Stpiczynski.*

### 4.1.1 Detailed Description

Implementation of the package.

The file contains implementation of all functions.

#### Author

Przemyslaw Stpiczynski

**Bug** No know bugs.

### 4.1.2 Function Documentation

#### 4.1.2.1 `trg_d_s_goertzel()`

```
void trg_d_s_goertzel (
    double x,
    int n,
    double * b,
    double * cx,
    double * sx )
```

Implementation of basic sequential Goertzel algorithm.

#### Parameters

<i>x</i>	Argument for C(x) and S(x)
<i>n</i>	Number of coefficients, b_0,...,b_n
<i>b</i>	Pointer to coefficients
<i>cx</i>	Pointer to computed C(x)
<i>sx</i>	Pointer to computed S(x)

#### Returns

void

#### 4.1.2.2 `trg_d_s_reinsch()`

```
void trg_d_s_reinsch (
    double x,
    int n,
    double * b,
```

```
double * cx,  
double * sx )
```

Implementation of basic sequential Reinschalgorithm.

**Parameters**

$x$	Argument for $C(x)$ and $S(x)$
$n$	Number of coefficients, $b_0, \dots, b_n$
$b$	Pointer to coefficients
$cx$	Pointer to computed $C(x)$
$sx$	Pointer to computed $S(x)$

**Returns**

void

**4.1.2.3 `trg_d_v_goertzel()`**

```
void trg_d_v_goertzel (
    double x,
    int n,
    double * b,
    double * cx,
    double * sx )
```

Implementation of vectorized Goertzel algorithm introduced by Stpiczynski.

**Parameters**

$x$	Argument for $C(x)$ and $S(x)$
$n$	Number of coefficients, $b_0, \dots, b_n$
$b$	Pointer to coefficients (should be allocated using <code>_mm_malloc()</code> )
$cx$	Pointer to computed $C(x)$
$sx$	Pointer to computed $S(x)$

**Returns**

void

**4.1.2.4 `trg_d_v_reinsch()`**

```
void trg_d_v_reinsch (
    double x,
    int n,
    double * b,
    double * cx,
    double * sx )
```

Implementation of vectorized Reinsch algorithm introduced by Stpiczynski.

## Parameters

<i>x</i>	Argument for C(x) and S(x)
<i>n</i>	Number of coefficients, b <sub>0</sub> ,...,b <sub>n</sub>
<i>b</i>	Pointer to coefficients (should be allocated using <code>_mm_malloc()</code> )
<i>cx</i>	Pointer to computed C(x)
<i>sx</i>	Pointer to computed S(x)

## Returns

void

## 4.2 trgsum.h File Reference

Function prototypes for the package "Trigonometric sums".

## Functions

- void [trg\\_d\\_s\\_goertzel](#) (double x, int n, double \*b, double \*cx, double \*sx)  
*Implementation of basic sequential Goertzel algorithm.*
- void [trg\\_d\\_v\\_goertzel](#) (double x, int n, double \*b, double \*cx, double \*sx)  
*Implementation of vectorized Goertzel algorithm introduced by Stpiczynski.*
- void [trg\\_d\\_s\\_reinsch](#) (double x, int n, double \*b, double \*cx, double \*sx)  
*Implementation of basic sequential Reinsch algorithm.*
- void [trg\\_d\\_v\\_reinsch](#) (double x, int n, double \*b, double \*cx, double \*sx)  
*Implementation of vectorized Reinsch algorithm introduced by Stpiczynski.*

### 4.2.1 Detailed Description

Function prototypes for the package "Trigonometric sums".

It contains sequential and vectorized implementations of Goertzel and Reinsch algorithms for finding trigonometric sums:

$$\begin{aligned} C(x) &= b_0 \cos(0) + b_1 \cos(x) + b_2 \cos(2x) + \dots + b_n \cos(nx) \\ S(x) &= b_1 \sin(x) + b_2 \sin(2x) + \dots + b_n \sin(nx) \end{aligned}$$

## Author

Przemyslaw Stpiczynski

**Bug** No known bugs.

### 4.2.2 Function Documentation

#### 4.2.2.1 `trg_d_s_goertzel()`

```
void trg_d_s_goertzel (
    double x,
    int n,
    double * b,
    double * cx,
    double * sx )
```

Implementation of basic sequential Goertzel algorithm.



**Parameters**

$x$	Argument for $C(x)$ and $S(x)$
$n$	Number of coefficients, $b_0, \dots, b_n$
$b$	Pointer to coefficients
$cx$	Pointer to computed $C(x)$
$sx$	Pointer to computed $S(x)$

**Returns**

void

**4.2.2.2 trg\_d\_s\_reinsch()**

```
void trg_d_s_reinsch (
    double x,
    int n,
    double * b,
    double * cx,
    double * sx )
```

Implementation of basic sequential Reinschalalgorithm.

**Parameters**

$x$	Argument for $C(x)$ and $S(x)$
$n$	Number of coefficients, $b_0, \dots, b_n$
$b$	Pointer to coefficients
$cx$	Pointer to computed $C(x)$
$sx$	Pointer to computed $S(x)$

**Returns**

void

**4.2.2.3 trg\_d\_v\_goertzel()**

```
void trg_d_v_goertzel (
    double x,
    int n,
    double * b,
    double * cx,
    double * sx )
```

Implementation of vectorized Goertzel algorithm introduced by Stpiczynski.

**Parameters**

<i>x</i>	Argument for C(x) and S(x)
<i>n</i>	Number of coefficients, $b_0, \dots, b_n$
<i>b</i>	Pointer to coefficients (should be allocated using <code>_mm_malloc()</code> )
<i>cx</i>	Pointer to computed C(x)
<i>sx</i>	Pointer to computed S(x)

**Returns**

void

**4.2.2.4 `trg_d_v_reinsch()`**

```
void trg_d_v_reinsch (
    double x,
    int n,
    double * b,
    double * cx,
    double * sx )
```

Implementation of vectorized Reinsch algorithm introduced by Stpiczynski.

**Parameters**

<i>x</i>	Argument for C(x) and S(x)
<i>n</i>	Number of coefficients, $b_0, \dots, b_n$
<i>b</i>	Pointer to coefficients (should be allocated using <code>_mm_malloc()</code> )
<i>cx</i>	Pointer to computed C(x)
<i>sx</i>	Pointer to computed S(x)

**Returns**

void

# Index

- trg\_d\_s\_goertzel
  - trgsum.c, [8](#)
  - trgsum.h, [11](#)
- trg\_d\_s\_reinsch
  - trgsum.c, [8](#)
  - trgsum.h, [13](#)
- trg\_d\_v\_goertzel
  - trgsum.c, [10](#)
  - trgsum.h, [13](#)
- trg\_d\_v\_reinsch
  - trgsum.c, [10](#)
  - trgsum.h, [14](#)
- trgsum.c, [7](#)
  - trg\_d\_s\_goertzel, [8](#)
  - trg\_d\_s\_reinsch, [8](#)
  - trg\_d\_v\_goertzel, [10](#)
  - trg\_d\_v\_reinsch, [10](#)
- trgsum.h, [11](#)
  - trg\_d\_s\_goertzel, [11](#)
  - trg\_d\_s\_reinsch, [13](#)
  - trg\_d\_v\_goertzel, [13](#)
  - trg\_d\_v\_reinsch, [14](#)