

---

# PSBLAS-KINSOL interface

---

*Istituto per le Applicazioni del Calcolo “M. Picone”,  
Consiglio Nazionale delle Ricerche  
Pasqua D’Ambra  
Fabio Durastante  
Salvatore Filippone  
PSBLAS 3.6.1 — Interface Version 1*

November 7, 2019

## Contents

<b>1</b>	<b>The NVECTOR_PSBLAS implementation</b>	<b>2</b>
1.1	NVECTOR_PSBLAS accessor macros . . . . .	2
1.2	NVECTOR_PSBLAS functions . . . . .	2

## 1 The NVECTOR\_PSBLAS implementation

The NVECTOR\_PSBLAS implementation of the SUNDIALS NVECTOR module provides an interface to the PSBLAS code for handling distributed dense vectors. Information on the PSBLAS data structures, and functions that are mentioned along the text can be found in [2].

It defines the *content* field of **N\_Vector** to be a structure containing the PSBLAS descriptor for the data distribution, a PSBLAS vector of double, and the PSBLAS communicator (context).

```
struct _N_VectorContent_PSBLAS {
    boolean_t own_data; /*ownership of data*/
    psb_c_descriptor *cdh; /*descriptor for data distribution*/
    psb_c_dvector *pvec; /*PSBLAS vector*/
    int ictxt; /*PSBLAS communicator*/
};
```

! → All the vectors that have to interact needs to be instantiated on the same parallel context ictxt, and on the same data distribution cdh.

The header file to include when using this module is **nvector\_psblas.h**. The installed module library to link to is **sundials\_nvecpsblas.lib** where **.lib** is typically **.so** for shared libraries and **.a** for static libraries.

### 1.1 NVECTOR\_PSBLAS accessor macros

The following macros are provided to access the content of a NVECTOR\_PSBLAS vector. The suffix **\_P** in the names denotes the fact that the data are in distributed memory.

```
#define NV_CONTENT_P(v) ((N_VectorContent_PSBLAS)(v->content))
#define NV_DESCRIPTOR_P(v) (NV_CONTENT_P(v)->cdh)
#define NV_OWN_DATA_P(v) (NV_CONTENT_P(v)->own_data)
#define NV_PVEC_P(v) (NV_CONTENT_P(v)->pvec)
#define NV_ICTXT_P(v) (NV_CONTENT_P(v)->ictxt)
```

NV_CONTENT_P(v)	this macro gives access to the contents of the PSBLAS vector <b>N_Vector</b> .
NV_DESCRIPTOR_P(v), NV_OWN_DATA_P(v), NV_PVEC_P(v)	these macros give instead individual access to the parts of the content of a PSBLAS parallel <b>N_Vector</b> .
NV_ICTXT_P(v)	this macro provides the PSBLAS context used by the NVECTOR_PSBLAS vectors.

### 1.2 NVECTOR\_PSBLAS functions

The NVECTOR\_PSBLAS implementation provides PSBLAS implementations of all the vectors operations listed in Tables 6.2, 6.3, and 6.4 of the original KINSOL library [1]. Following the standard nomenclature of the SUNDIALS library, their names are obtained from the ones listed in Tables 6.2, 6.3, and 6.4 by appending the suffix **\_PSBLAS**. The NVECTOR\_PSBLAS implementation provides the following additional user-callable routines.

<b>N_VNew_PSBLAS</b>	:	This function creates and allocates memory for a parallel vector on the PSBLAS context ictxt with the communicator cdh
----------------------	---	--

Prototype	<b>N_Vector</b> N_VNew_PSBLAS(int ictxt, psb_c_descriptor *cdh);
-----------	--

<b>N_VNewEmpty_PSBLAS</b>	:	This function creates a new PSBLAS vector with empty data array.
---------------------------	---	--

Prototype	<b>N_Vector</b> N_VNewEmpty_PSBLAS(int ictxt, psb_c_descriptor *cdh);
-----------	---

<code>N_VMake_PSBLAS</code>	:	Function to create a PSBLAS <code>N_Vector</code> with user data component. This function is substantially a wrapper for the PSBLAS function <code>psb_c_dgeins</code> .
Prototype		<code>N_Vector N_VMake_PSBLAS(int ictxt, psb_c_descriptor *cdh, psb_i_t m, psb_i_t *irow, double *val);</code>
		The PSBLAS context <code>ictxt</code> with the communicator <code>cdh</code> are the one defined for the whole programs, the integer <code>m</code> is the number of rows in <code>val[]</code> to be inserted, the array of integers <code>irow</code> is the indices of the rows to be inserted. Specifically, row <code>i</code> of <code>val</code> will be inserted into the local row corresponding to the global index <code>row index row[i]</code> .
<code>! →</code>		This routine does not assemble the final vector. After the insertion of all the elements has been completed then the vector should be assembled by means of the <code>N_VAss_PSBLAS</code> routine.
<code>N_VAsb_PSBLAS</code>	:	This routine assemble the <code>N_Vector</code> after that all the elements have been inserted into it, i.e., after that all the calls to the <code>N_VMake_PSBLAS</code> routine have been completed. This is substantially a wrapper for the PSBLAS function <code>psb_c_dgeasb</code> .
Prototype		<code>void N_VAsb_PSBLAS(N_Vector v)</code>
<code>N_VCloneVectorArray_PSBLAS</code>	:	This function creates an array of new parallel vectors (by cloning) an array of count parallel vectors <code>v</code> .
Prototype		<code>N_Vector *N_VCloneVectorArray_PSBLAS(int count, N_Vector w)</code>
<code>N_VCloneVectorArrayEmpty_PSBLAS</code>	:	This function creates an array of count new parallel vectors with empty data array on the same communicator and context of the vector <code>w</code> .
Prototype		<code>N_Vector *N_VCloneVectorArrayEmpty_PSBLAS(int count, N_Vector w)</code>
<code>N_VDestroyVectorArray_PSBLAS</code>	:	This function to frees an array of count <code>N_Vectors</code> created with <code>N_VCloneVectorArray_PSBLAS</code>
Prototype		<code>void N_VDestroyVectorArray_PSBLAS(N_Vector *vs, int count)</code>
<code>N_VGetLength_PSBLAS</code>	:	This function returns the <i>global</i> vector length, this is substantially a wrapper for the PSBLAS function <code>psb_c_cd_get_global_rows</code> .
Prototype		<code>sunindextype N_VGetLength_PSBLAS(N_Vector v)</code>
<code>N_VGetLocalLength_PSBLAS</code>	:	This function returns the <i>local</i> vector length, this is substantially a wrapper for the PSBLAS function <code>psb_c_cd_get_local_rows</code> .
Prototype		<code>sunindextype N_VGetLocalLength_PSBLAS(N_Vector v)</code>
<code>N_VPrint_PSBLAS</code>	:	This function prints the local data in a parallel vector to <code>stdout</code> .
Prototype		<code>void N_VPrint_PSBLAS(N_Vector x)</code>
<code>N_VPrintFile_PSBLAS</code>	:	This function prints the local data in a parallel vector to <code>outfile</code> .
Prototype		<code>void N_VPrintFile_PSBLAS(N_Vector x, FILE* outfile)</code>

## References

- [1] Aaron M. Collier et al. *User Documentation for kinsol v4.1.0*. Center for Applied Scientific Computing Lawrence Livermore National Laboratory. URL: [https://computing.llnl.gov/sites/default/files/public/kin\\_guide.pdf](https://computing.llnl.gov/sites/default/files/public/kin_guide.pdf).

- [2] S. Filippone and A. Buttari. *PSBLAS 3.6.1 User's guide. A reference guide for the Parallel Sparse BLAS library*. Cranfield University, Centre for Computational Engineering Sciences. URL: <https://github.com/sfilippone/psblas3/blob/development/docs/psblas-3.6.pdf>.