Применение fortran-библиотек линейной алгебры в среде delphi

**Человек:** В работе представлен компонент и заголовочные модули, написанные на языке Object Pascal, обеспечивающие простое подключение функций и процедур таких библиотек численного решения задач линей-ной алгебры как BLAS, LAPACK, ARPACK и др., написанных на языке FORTRAN. Использование этих библиотек в современных языках программирования, например в C++ или Object Pascal, сталкивается с существенными проблемами, связанными с методами передачи параметров в подпрограммы, а также со способом представления массивов в оперативной памяти. Более серьезной проблемой является нумерация элементов с 1, принятая в FORTRAN. В Object Pascal учет этого условия может потребовать тщательного анализа и модификации алгоритма решения задачи. Это существенно увеличивает трудоемкость программирования и ведет к появлению скрытых, трудно обнаруживаемых алгоритмических ошибок. Для обеспечения прозрачного обращения к подпрограммам FORTRAN-библиотек из программ на языке Object Pascal среды Delphi разработаны два класса объектов для представления одномерных массивов TVector и двумерных массивов - TMatrix.

**Key words:** fortran, библиотеки, линейная алгебра, delphi, Object Pascal, TVector, TMatrix, программирование, алгоритм, подпрограмма

=================================

**FastText\_KMeans\_Clean:** Для преодоления указанных трудностей и обеспечения прозрачного обращения к подпрограммам FORTRAN-библиотек из программ на языке Object Pascal среды Delphi были разработаны два класса объектов для представления одномерных массивов TVector и двумерных массивов - TMatrix. procedure SetCell(ARow, ACol:integer; v:Double; IsValid:Boolean);. //Объявления используемых модулей. //Объявления массивов. var v, d : TMatrix; workl, workd, resid, ax : TVector;.

**Key words part:** 0.7142857142857143

=================================

**FastText\_KMeans\_Raw/:** Для преодоления указанных трудностей и обеспечения прозрачного обращения к подпрограммам FORTRAN-библиотек из программ на языке Object Pascal среды Delphi были разработаны два класса объектов для представления одномерных массивов TVector и двумерных массивов - TMatrix. constructor SetLen (ASize: integer; AMode : TMode = mDELPHI);. property Item[i:integer] : Double read GetValue write SetValue; default;. procedure SetValue(ARow, ACol:integer; v:Double);. function FirstRow: Integer;. workd := TVector.SetLen(3\*maxn,mFORTRAN);.

**Key words part:** 0.7142857142857143

=================================

**FastText\_PageRank\_Clean/:** Более серьезной проблемой является нумерация элементов с 1, принятая в FORTRAN. Это существенно увеличивает трудоемкость программирования и ведет к появлению скрытых, трудно обнаруживаемых алгоритмических ошибок. Наибольшие затруднения представляют вызовы подпрограмм FORTRAN-библиотек, связанные с обработкой матриц. Определения классов имею вид:. При этом для учета способа хранения элементов во внутреннем массиве используется соответствующий способ расчета индексов. //Объявления используемых модулей. //Объявления массивов. //Приведение массива к стилю Delphi.

**Key words part:** 0.6190476190476191

=================================

**FastText\_PageRank\_Raw/:** Более серьезной проблемой является нумерация элементов с 1, принятая в FORTRAN. Это существенно увеличивает трудоемкость программирования и ведет к появлению скрытых, трудно обнаруживаемых алгоритмических ошибок. Наибольшие затруднения представляют вызовы подпрограмм FORTRAN-библиотек, связанные с обработкой матриц. Определения классов имею вид:. При этом для учета способа хранения элементов во внутреннем массиве используется соответствующий способ расчета индексов. //Объявления массивов. v := TMatrix.SetLen(ldv,maxncv,mFORTRAN);. //Приведение массива к стилю Delphi.

**Key words part:** 0.6190476190476191

=================================

**Mixed\_ML\_TR/:** Вместе с тем использование этих библиотек в современных языках программирования, например в C++ или Object Pascal, сталкивается с существенными проблемами, связанными с методами передачи параметров в подпрограммы, а также со способом представления массивов в оперативной памяти. TMode = (mDELPHI=0, mFORTRAN); //Режим работы массива. fMode : TMode; // mDELPHI V[0.Size-1]; mFORTRAN V[1.Size]. property Ptr[i:integer] : PDouble read GetPointer write SetByPointer;. fRowCount: integer;. function GetPointer(ARow, ACol:integer) : PDouble;. procedure SetPtr(ARow, ACol:integer; p : PDouble);. function GetValue(ARow, ACol:integer):Double;. property Ptr[ARow, ACol: integer]:PDouble read GetPointer write SetPtr;. workl := TVector.SetLen(maxncv\*(maxncv+8),mFORTRAN);.

**Key words part:** 0.6666666666666666

=================================

**MultiLingual\_KMeans/:** Вместе с тем использование этих библиотек в современных языках программирования, например в C++ или Object Pascal, сталкивается с существенными проблемами, связанными с методами передачи параметров в подпрограммы, а также со способом представления массивов в оперативной памяти. TMode = (mDELPHI=0, mFORTRAN); //Режим работы массива. fMode : TMode; // mDELPHI V[0.Size-1]; mFORTRAN V[1.Size]. property Ptr[i:integer] : PDouble read GetPointer write SetByPointer;. fRowCount: integer;. procedure SetPtr(ARow, ACol:integer; p : PDouble);. workl := TVector.SetLen(maxncv\*(maxncv+8),mFORTRAN);.

**Key words part:** 0.6666666666666666

=================================

**Multilingual\_PageRank/:** Вместе с тем использование этих библиотек в современных языках программирования, например в C++ или Object Pascal, сталкивается с существенными проблемами, связанными с методами передачи параметров в подпрограммы, а также со способом представления массивов в оперативной памяти. Передача параметров по адресу (ссылке), принятая в FORTRAN, имеет простой аналог в языке Object Pascal и не является серьезной проблемой. Более серьезной проблемой является нумерация элементов с 1, принятая в FORTRAN. Это существенно увеличивает трудоемкость программирования и ведет к появлению скрытых, трудно обнаруживаемых алгоритмических ошибок. Наибольшие затруднения представляют вызовы подпрограмм FORTRAN-библиотек, связанные с обработкой матриц. Определения классов имею вид:. destructor Destroy; override;. workd := TVector.SetLen(3\*maxn,mFORTRAN);.

**Key words part:** 0.7142857142857143

=================================

**RuBERT\_KMeans\_Without\_ST/:** Вместе с тем использование этих библиотек в современных языках программирования, например в C++ или Object Pascal, сталкивается с существенными проблемами, связанными с методами передачи параметров в подпрограммы, а также со способом представления массивов в оперативной памяти. var tol: TData; resid: PData; var ncv: Integer;. property Ptr[ARow, ACol: integer]:PDouble read GetPointer write SetPtr;. workd := TVector.SetLen(3\*maxn,mFORTRAN);.

**Key words part:** 0.6666666666666666

=================================

**RuBERT\_KMeans\_With\_ST/:** var n: Integer; var which: Char; var nev: Integer;. function GetValid(i:integer):Boolean;. property Ptr[ARow, ACol: integer]:PDouble read GetPointer write SetPtr;. В качестве примера использования предложенных классов приводится выдержка из программы на языке Object Pascal для решения проблемы собственных значений методом Арнольди [2] с применением пакетов BLAS, LAPACK и ARPACK.

**Key words part:** 0.5238095238095238

=================================

**RUBERT\_page\_rank\_Without\_ST/:** Это существенно увеличивает трудоемкость программирования и ведет к появлению скрытых, трудно обнаруживаемых алгоритмических ошибок. Определения классов имею вид:. function FirstIndex: Integer;. function LastIndex:Integer;. fRowCount: integer;.

**Key words part:** 0.4761904761904762

=================================

**RUBERT\_page\_rank\_With\_ST/:** Наибольшие затруднения представляют вызовы подпрограмм FORTRAN-библиотек, связанные с обработкой матриц. write SetValue; default;. При этом для учета способа хранения элементов во внутреннем массиве используется соответствующий способ расчета индексов. //Объявления используемых модулей. //Объявления массивов.

**Key words part:** 0.4761904761904762

=================================

**RUSBERT\_KMeans\_Without\_ST/:** Для преодоления указанных трудностей и обеспечения прозрачного обращения к подпрограммам FORTRAN-библиотек из программ на языке Object Pascal среды Delphi были разработаны два класса объектов для представления одномерных массивов TVector и двумерных массивов - TMatrix. fSize: Integer; //Кол-во элементов. procedure SetValue(i:integer; v:Double);. constructor SetLen (ASize: integer; AMode : TMode = mDELPHI);. workd := TVector.SetLen(3\*maxn,mFORTRAN);.

**Key words part:** 0.7142857142857143

=================================

**RUSBERT\_KMeans\_With\_ST/:** Проблемой здесь является метод сохранения матриц по столбцам, принятый в языке FORTRAN, тогда как в Object Pascal массивы хранятся в памяти по строкам (Рис.1). procedure SetPtr(ARow, ACol:integer; p : PDouble);. function GetValue(ARow, ACol:integer):Double;. v := TMatrix.SetLen(ldv,maxncv,mFORTRAN);.

**Key words part:** 0.5714285714285714

=================================

**RUSBERT\_page\_rank\_Without\_ST/:** var n: Integer; var which: Char; var nev: Integer;. Это существенно увеличивает трудоемкость программирования и ведет к появлению скрытых, трудно обнаруживаемых алгоритмических ошибок. Поэтому прямая передача адреса начального элемента в подпрограмму LAPACK из программы Object Pascal приведет к обработке транспонированной матрицы, либо к ошибке обращения к памяти для неквадратных матриц. for j:=0 to nconv-1 do. Разработанные классы объектов решают все перечисленные проблемы и обеспечивают прозрачный доступ к подпрограммам математических библиотек BLAS, LAPACK и ARPACK из программ на языке Object Pascal.

**Key words part:** 0.6666666666666666

=================================

**RUSBERT\_page\_rank\_With\_ST/:** Передача параметров по адресу (ссылке), принятая в FORTRAN, имеет простой аналог в языке Object Pascal и не является серьезной проблемой. В Object Pascal учет этого условия может потребовать тщательного анализа и модификации алгоритма решения задачи. Это существенно увеличивает трудоемкость программирования и ведет к появлению скрытых, трудно обнаруживаемых алгоритмических ошибок. Поэтому прямая передача адреса начального элемента в подпрограмму LAPACK из программы Object Pascal приведет к обработке транспонированной матрицы, либо к ошибке обращения к памяти для неквадратных матриц. Разработанные классы объектов решают все перечисленные проблемы и обеспечивают прозрачный доступ к подпрограммам математических библиотек BLAS, LAPACK и ARPACK из программ на языке Object Pascal.

**Key words part:** 0.7619047619047619

=================================

**Simple\_PageRank/:** TMode = (mDELPHI=0, mFORTRAN); //Режим работы массива. function GetPointer(ARow, ACol:integer) : PDouble;. procedure SetPtr(ARow, ACol:integer; p : PDouble);. function GetValue(ARow, ACol:integer):Double;. procedure SetValue(ARow, ACol:integer; v:Double);. workd := TVector.SetLen(3\*maxn,mFORTRAN);.

**Key words part:** 0.42857142857142855

=================================

**TextRank/:** Вместе с тем использование этих библиотек в современных языках программирования, например в C++ или Object Pascal, сталкивается с существенными проблемами, связанными с методами передачи параметров в подпрограммы, а также со способом представления массивов в оперативной памяти. Кроме того все массивы в языке FORTRAN в качестве номера первого элемента используют индекс 1, тогда как динамические массивы C++ и Object Pascal принимают в качестве номера первого элемента массива индекс 0. Проблемой здесь является метод сохранения матриц по столбцам, принятый в языке FORTRAN, тогда как в Object Pascal массивы хранятся в памяти по строкам (Рис.1). Для преодоления указанных трудностей и обеспечения прозрачного обращения к подпрограммам FORTRAN-библиотек из программ на языке Object Pascal среды Delphi были разработаны два класса объектов для представления одномерных массивов TVector и двумерных массивов - TMatrix. property Ptr[ARow, ACol: integer]:PDouble read GetPointer write SetPtr;. Разработанные классы объектов решают все перечисленные проблемы и обеспечивают прозрачный доступ к подпрограммам математических библиотек BLAS, LAPACK и ARPACK из программ на языке Object Pascal.

**Key words part:** 0.8571428571428571

=================================

**TF-IDF\_KMeans/:** var n: Integer; var which: Char; var nev: Integer;. fMode : TMode; // mDELPHI V[0.Size-1]; mFORTRAN V[1.Size]. property Ptr[i:integer] : PDouble read GetPointer write SetByPointer;. function GetValid(ARow, ACol:integer):Boolean;. procedure SetValue(ARow, ACol:integer; v:Double);. //Создание массивов в стиле FORTRAN. d := TMatrix.SetLen(maxncv,2,mFORTRAN);. Разработанные классы объектов решают все перечисленные проблемы и обеспечивают прозрачный доступ к подпрограммам математических библиотек BLAS, LAPACK и ARPACK из программ на языке Object Pascal.

**Key words part:** 0.6666666666666666

=================================

**Текст:** Вместе с тем использование этих библиотек в современных языках программирования, например в C++ или Object Pascal, сталкивается с существенными проблемами, связанными с методами передачи параметров в подпрограммы, а также со способом представления массивов в оперативной памяти. Подпрограммы BLAS, LAPACK, ARPACK воспринимают все векторные параметры подпрограмм как адреса соответствующих объектов. Кроме того все массивы в языке FORTRAN в качестве номера первого элемента используют индекс 1, тогда как динамические массивы C++ и Object Pascal принимают в качестве номера первого элемента массива индекс 0.. Передача параметров по адресу (ссылке), принятая в FORTRAN, имеет простой аналог в языке Object Pascal и не является серьезной проблемой. Достаточно объявить вызываемые функции библиотеки LAPACK в стиле, например. var dsaupd\_: function(var ido: Integer; var bmat: Char;. var n: Integer; var which: Char; var nev: Integer;. var tol: TData; resid: PData; var ncv: Integer;. v: PData; var ldv: integer; iparam: PInteger;. ipntr: PInteger; workd, workl: PData;. var lworkl: Integer; var info: Integer ) : integer cdecl;. Более серьезной проблемой является нумерация элементов с 1, принятая в FORTRAN. В Object Pascal учет этого условия может потребовать тщательного анализа и модификации алгоритма решения задачи. Это существенно увеличивает трудоемкость программирования и ведет к появлению скрытых, трудно обнаруживаемых алгоритмических ошибок.. Наибольшие затруднения представляют вызовы подпрограмм FORTRAN-библиотек, связанные с обработкой матриц. Проблемой здесь является метод сохранения матриц по столбцам, принятый в языке FORTRAN, тогда как в Object Pascal массивы хранятся в памяти по строкам (Рис.1). Поэтому прямая передача адреса начального элемента в подпрограмму LAPACK из программы Object Pascal приведет к обработке транспонированной матрицы, либо к ошибке обращения к памяти для неквадратных матриц.. . Для преодоления указанных трудностей и обеспечения прозрачного обращения к подпрограммам FORTRAN-библиотек из программ на языке Object Pascal среды Delphi были разработаны два класса объектов для представления одномерных массивов TVector и двумерных массивов - TMatrix. Определения классов имею вид:. TMode = (mDELPHI=0, mFORTRAN); //Режим работы массива. TVector = class. private. fMode : TMode; // mDELPHI V[0..Size-1]; mFORTRAN V[1..Size]. fSize: Integer; //Кол-во элементов. fFirstItem: PDouble;. protected. function GetPointer(i: integer): PDouble;. procedure SetByPointer(i:integer; p : PDouble);. function GetValid(i:integer):Boolean;. function GetValue(i:integer):Double;. procedure SetValue(i:integer; v:Double);. procedure SetSize(const N:integer);. public. constructor SetLen (ASize: integer; AMode : TMode = mDELPHI);. destructor Destroy; override;. procedure Clear;. procedure Assign(Source: TVector);. function FirstIndex: Integer;. function LastIndex:Integer;. procedure SetZero;. property Mode : TMode read fMode write fMode;. property Size : Integer read fSize write SetSize;. property Item[i:integer] : Double read GetValue write SetValue; default;. property Ptr[i:integer] : PDouble read GetPointer write SetByPointer;. property FirstItem: PDouble read fFirstItem;. end;. TMatrix = class(TVector). private. fRowCount: integer;. fColCount: integer;. protected. function GetPointer(ARow, ACol:integer) : PDouble;. procedure SetPtr(ARow, ACol:integer; p : PDouble);. function GetValid(ARow, ACol:integer):Boolean;. function GetValue(ARow, ACol:integer):Double;. procedure SetValue(ARow, ACol:integer; v:Double);. procedure SetCell(ARow, ACol:integer; v:Double; IsValid:Boolean);. public. constructor SetLen(ARows, ACols:integer; AMode:TMode=mDELPHI);. procedure Assign(Source: TMatrix);. property Item[ARow, ACol: integer]:Double read GetValue. write SetValue; default;. property Ptr[ARow, ACol: integer]:PDouble read GetPointer write SetPtr;. property Rows: integer read fRowCount;. property Cols: integer read fColCount;. function FirstRow: Integer;. function FirstCol: Integer;. function LastRow: Integer;. function LastCol: Integer;. property Valid[ARow, ACol:integer]:Boolean read GetValid;. end;. Оба класса имеют конструктор SetLen, который обеспечивает выделение памяти для хранения массивов. Физически массив всегда представляется как одномерный, причем для двумерных массивов данные располагаются по столбцам, как это принято в FORTARN. Элементы массивов во внутренней памяти всегда нумеруются с 0-го индекса, однако обращаться к ним следует с учетом режима индексации, заданного свойством Mode. При Mode= mFORTRAN индексом первого элемента является 1. В этом режиме также принято извлечение и запись элементов двумерного массива в стиле FORTRAN – по столбцам. В режиме Mode=mDELPHI обращение к элементам массива выполняется, как это принято в Object Pascal. При этом для учета способа хранения элементов во внутреннем массиве используется соответствующий способ расчета индексов.. В качестве примера использования предложенных классов приводится выдержка из программы на языке Object Pascal для решения проблемы собственных значений методом Арнольди [2] с применением пакетов BLAS, LAPACK и ARPACK.. //Объявления используемых модулей. uses Utils, BLAS, LAPACK, ARPACK, Vector, Matrix;. //Объявления массивов. var v, d : TMatrix; workl, workd, resid, ax : TVector;. . //Создание массивов в стиле FORTRAN. workl := TVector.SetLen(maxncv\*(maxncv+8),mFORTRAN);. workd := TVector.SetLen(3\*maxn,mFORTRAN);. v := TMatrix.SetLen(ldv,maxncv,mFORTRAN);. d := TMatrix.SetLen(maxncv,2,mFORTRAN);. …. //Вызов подпрограммы из пакета ARPACK. dseupd\_( rvec, howmny, @select(1), d.Ptr(1,1), v.Ptr(1,1), ldv, sigma,. bmat, n, which(1), nev, tol, resid.Ptr(1), ncv, v.Ptr(1,1), ldv,. iparam.Ptr(1), ipntr.Ptr(1), workd.Ptr(1), workl.Ptr(1), lworkl, ierr );. //Приведение массива к стилю Delphi. d.Mode := mDELPHI;. for j:=0 to nconv-1 do. for j:=0 to 1 do. … d[j, i] …. …. Разработанные классы объектов решают все перечисленные проблемы и обеспечивают прозрачный доступ к подпрограммам математических библиотек BLAS, LAPACK и ARPACK из программ на языке Object Pascal.