



IBM ILOG CPLEX Optimization Studio Getting Started with CPLEX for MATLAB

Version 12 Release 7

Note

Before you use this information and the product it supports, read the information in "Notices" on page 19.

Contents

Chapter 1. Introduction	1	Chapter 6. Programming tips	15
Chapter 2. Setting up CPLEX for MATLAB	3	Chapter 7. Using parameters	17
Chapter 3. Integration with MATLAB . . .	5	Notices	19
Chapter 4. Using CPLEX for MATLAB . . .	7	Terms and conditions for product documentation. . .	21
Chapter 5. Overview of the CPLEX for MATLAB APIs	9	Copyright notice	23
CPLEX for MATLAB Toolbox.	9	Additional registered trademarks, copyrights, licenses	25
Cplex Class API	11	Index	27

Chapter 1. Introduction

An overview of CPLEX[®] for MATLAB.

CPLEX for MATLAB is an extension to IBM[®] ILOG[®] CPLEX Optimizers that allows a user to define optimization problems and solve them within MATLAB. Thus a student or practitioner who is using MATLAB can easily solve optimization problems within that framework.

Chapter 2. Setting up CPLEX for MATLAB

Install and configure IBM ILOG CPLEX Optimization Studio before you use the CPLEX connector for MATLAB.

Install CPLEX

Before you begin to use the CPLEX connector for MATLAB, you must first install and configure CPLEX Optimization Studio and CPLEX Optimizer.

By default, the CPLEX Optimization Studio installer automatically installs the CPLEX connector for MATLAB in a standard location. For instance, the installation directory for CPLEX for MATLAB for a Windows 64-bit operating system is `yourC0SHome\cplex\matlab\x64_win64`, where `yourC0SHome` specifies the folder where CPLEX Optimization Studio is installed. Other platforms have similar installation directories, named appropriately.

Verify the service pack

When CPLEX Optimization Studio is installed on Windows operating systems, the appropriate Microsoft Visual C++ Redistributable Package must also be installed. This package installs runtime components of Visual C++ Libraries that are required to run applications that are developed with Visual C++ on a computer that does not have Visual C++ installed. If the CPLEX Optimization Studio installer detects that this package is not already installed, the installer explains how to download the service pack manually. For more information, see the Detailed System Requirements (DSR).

Configure MATLAB

To configure MATLAB to use CPLEX, you must first add the CPLEX connector for MATLAB installation folder to your MATLAB path by using the MATLAB `addpath` command. For instance, the installation directory for 64-bit Windows is `yourC0SHome\cplex\matlab\x64_win64`, where `yourC0SHome` specifies the folder where CPLEX Optimization Studio is installed. In addition, add the folder `yourC0SHome\cplex\examples\src\matlab` in the same manner to support navigational links to examples in the online help from within a MATLAB session.

Chapter 3. Integration with MATLAB

The menu items and windows used to solve optimization models are described.

CPLEX for MATLAB should be integrated into your MATLAB environment in order for you to take full advantage of its features.

When you have installed CPLEX for MATLAB and set the paths as described in the `readme.html` file, a new item is added to the Toolboxes section of the MATLAB **Start Button**. You can use the items on this menu to find more help about using CPLEX.

In addition, the online manuals for CPLEX for MATLAB have been added to the MATLAB Product Help, available from the drop down menu **Help > Product Help**.

Within the MATLAB Command Window, inline help is available for the CPLEX classes and functions. For example typing `'help cplexlp'` will display information about the function `cplexlp`.

Chapter 4. Using CPLEX for MATLAB

Presents an overview of how to solve an optimization problem.

IBM ILOG CPLEX Optimizers provides a tool for solving optimization, or mathematical programming, problems.

The most basic mathematical programming problem is commonly referred to as Linear Programming (LP) problem. The basic form of an LP problem is:

Maximize (or Minimize)	$f \cdot x$
subject to	$A_{eq} \cdot x = b_{eq}$
	$A_{ineq} \cdot x \leq b_{ineq}$
with these bounds	$l \leq x \leq u$

where A_{eq} and A_{ineq} are matrices, f , b_{eq} , b_{ineq} , l and u are vectors such that the upper bounds $u(i)$ and lower bounds $l(i)$ may be positive infinity, negative infinity, or any real number. Both sparse and dense format can be used in all places where matrices/vectors are used.

The elements of data you provide as input for this LP problem are:

Data provided as input for an LP problem

Objective function coefficients	f
Constraint coefficients	A_{eq}
	A_{ineq}
Righthand sides	b_{eq}
	b_{ineq}
Upper and lower bounds	u
	l

The optimal solution that CPLEX computes and returns is:

Variables	x
-----------	-----

CPLEX for MATLAB can also solve several extensions to LP:

- Quadratic Programming (QP) problems, where the LP objective function is expanded to include quadratic terms.
- Quadratically Constrained Programming (QCP) problems that include quadratic terms among the constraints. In fact, CPLEX can solve Second Order Cone Programming (SOCP) problems.
- Mixed Integer Programming (MIP) problems, where any or all of the LP, QP, or QCP variables are further restricted to take integer values in the optimal solution

and where MIP itself is extended to include constructs like Special Ordered Sets (SOS), semi-continuous variables, and indicator variables.

- Least Squares (LSQ) problems, where the objective is to minimize a norm. The problem can be constrained linearly or quadratically, and the variables may be restricted to take integer values in the solution.

The standard MATLAB vector and matrix format is used for the elements of data that you need to provide. For example, the CPLEX for MATLAB Toolbox function `cplexlp` solves the problem specified by

```
min      f*x
st.      Aineq*x <= bineq
         Aeq*x   = beq
         lb <= x <= ub
```

where `f`, `bineq`, `beq`, `lb`, and `ub` are MATLAB double vectors, and `Aineq` and `Aeq` are MATLAB double matrices.

The vector `x` returned by the function call

```
x = cplexlp(f,Aineq,beq,Aeq,beq,lb,ub)
```

contains the optimal solution to the specified linear programming problem.

Provided in CPLEX for MATLAB is both a toolbox of functions and a class API. The toolbox contains functions for solving optimization problems, where the input matrices are provided to the function and results returned. With the class API, objects can be created, and those objects carry a state.

The benefits of using the `Cplex` class API include the ability to:

- build up a model by manipulating a `Cplex` object.
- use computation methods such as `Cplex.solve()` and `Cplex.refineConflict()` that modify the object so results can be queried as needed.
- perform restarts after manipulation.
- attach an output parser, a GUI with stop buttons, and other controls.

Chapter 5. Overview of the CPLEX for MATLAB APIs

CPLEX for MATLAB provides two APIs for solving mathematical programming problems, the toolbox functions and the Cplex class.

CPLEX for MATLAB Toolbox

The toolbox provides functions for solving a variety of mathematical programming problems.

The CPLEX for MATLAB Toolbox provides functions for solving a variety of mathematical programming problems. The toolbox functions are designed to take a model description as input and produce a solution as output.

For example:

```
x = cplexlp(f,Aineq,bineq,Aeq,beq,lb,ub)
```

finds the minimum of a linear programming problem specified by

```
min    f*x
st.    Aineq*x <= bineq
        Aeq*x   = beq
        lb <= x <= ub
```

The toolbox provides the functions `cplexlp`, `cplexqp` and `cplexbip` to solve linear programming problems (LP), quadratic programming problems (QP) and binary integer programming problems (BILP).

The toolbox provides functions that support the solution of the basic problem types handled by CPLEX are:

- `cplexlp` for linear programming problems (LP),
- `cplexqp` for quadratic programming problems (QP) and
- `cplexbip` for binary integer programming problems (BILP).

Functions that support the solution of additional problem types handled by CPLEX are provided. These functions are:

- `cplexqcp` for quadratically constrained programming problems (QCP),
- `cplexmilp` for mixed integer linear programming problems (MIP),
- `cplexmiqp` for mixed integer quadratic programming problems (MIQP) and
- `cplexmiqcp` for mixed integer quadratically constrained mixed integer programming problems (MIQCP).

The solution of least square problems is supported through the functions:

- `cplexlsqlin` for linearly constrained least squares problems,
- `cplexlsqmip` for linearly constrained mixed integer least squares problems,
- `cplexlsqbip` for linearly constrained binary integer least squares problems,
- `cplexlsqmiqcp` for quadratically constrained mixed integer least squares problems,
- `cplexlsqqcp` for quadratically constrained programming problems,
- `cplexlsqnonneglin` for nonnegative least squares problems,

- `cplexlsqnonnegmip` for nonnegative mixed integer least squares problems,
- `cplexlsqnonnegmiqcp` for nonnegative quadratically constrained mixed integer least squares problems and
- `cplexlsqnonnegqcp` for nonnegative quadratically constrained programming problems.

The advantage of the toolbox design is that you can reuse your code where you had used MATLAB Optimization Toolbox functions to solve linear programming, quadratic programming, binary integer programming, linearly constrained least squares, and nonnegative least squares problems.

Setting and querying parameters in the CPLEX for MATLAB Toolbox

Options, also called parameters, can be set to control the solution of problems. The toolbox provides two types of options input. One type corresponds to the MATLAB Optimization Toolbox options, and the other type is the CPLEX parameters. You can use either or both of these types of options. If you use both, the CPLEX parameters will override the MATLAB options.

The toolbox options are listed in the following table.

Options corresponding to the MATLAB Optimization Toolbox

Toolbox options	MATLAB Optimization Toolbox options
Display	'off' 'on' 'iter'
MaxIter	refer to <code>cplex.Param.simplex.limits.iterations</code>
Algorithm	'interior-point' 'primal' 'dual' 'auto'
BranchStrategy	refer to <code>cplex.Param.mip.strategy.variableselect</code>
MaxNodes	refer to <code>cplex.Param.mip.limits.nodes</code>
MaxTime	refer to <code>cplex.Param.timelimit</code>
NodeDisplayInterval	refer to <code>cplex.Param.mip.interval</code>
NodeSearchStrategy	refer to <code>cplex.Param.mip.strategy.nodeselect</code>
ToIFun	refer to <code>cplex.Param.simplex.tolerances.optimality</code>
ToIInteger	refer to <code>cplex.Param.mip.tolerances.integrality</code>
ToIRLPFun	refer to <code>cplex.Param.simplex.tolerances.optimality</code>

These options can be set using the toolbox function `optimset`. For example, the following code turns on the optimizer output and sets a node limit of 400,

```
options = cplexoptimset('Display', 'on', 'MaxNodes', 400);
```

Alternatively, these options can be set directly on the fields of the structure. For example, the following code has the same result as the previous one.

```
options = cplexoptimset;
options.Display = 'on';
options.MaxNodes = 400;
```

The current and default values of the options can be queried with the function `optimget`.

If you need to use CPLEX parameters that do not correspond to the options in the MATLAB Optimization Toolbox, you can create a structure which contains all of the CPLEX parameters. For example, to set a node limit of 400 and instruct CPLEX to use traditional branch and cut style search:

```
opt = cplexoptimset('cplex');  
opt.mip.limits.nodes=400;  
opt.mip.strategy.search=1;
```

Tip:

If you are already familiar with the names of parameters in the Interactive Optimizer, then you quickly recognize names of parameters in CPLEX for MATLAB. For example, the command “set mip limits nodes 1” in the Interactive Optimizer corresponds to “opt.mip.limits.nodes = 1;” in the CPLEX for MATLAB Toolbox where opt was created with the line “opt = optimset('cplex');”

To assist in setting parameters, auto-completion of parameter names is available in the MATLAB environment.

Cplex Class API

Describes the Cplex class

While the CPLEX for MATLAB Toolbox functions provide the ability to solve a multitude of mathematical programming problems, the toolbox design does not support restart. To enable users to use decomposition algorithms, the Cplex Class API is also provided in CPLEX for MATLAB.

The Cplex class stores the model and provides methods for the solution, analysis, manipulation and reading/writing of the model file. All of the data associated with the problem is stored in the properties of a Cplex object. These class properties are standard MATLAB data structures and can be manipulated directly within MATLAB. However, modifying the problem using methods provided in the Cplex class enforces consistency, such as ensuring that vectors are of the proper length.

The documentation of the Cplex class provides an introduction of properties and methods of the Cplex class in more detail.

The properties of the Cplex class include:

Properties of the Cplex class

Cplex.Model
Cplex.Solution
Cplex.Param
Cplex.Start
Cplex.MipStart
Cplex.InfoCallback
Cplex.Conflict

Cplex.Order
Cplex.DisplayFunc

Function of the properties

stores the data of the model
stores the solution of the model
stores the parameters (options) of the model
stores the start of the LP model
stores the start of the MIP model
pointer to an informational callback
stores the conflict information of a conflicted model

stores the priority order information
pointer to a function which provides control of display of output

The following informative methods are provided:

Cplex classes

Cplex.getVersion
Cplex.getProbType

Use of the Cplex classes

returns the CPLEX version
returns the problem type of the model

The following methods are provided for reading from and writing to files:

Methods for reading from and writing to files

Cplex.readModel
Cplex.writeModel
Cplex.readSolution
Cplex.writeSolution
Cplex.readBasis
Cplex.writeBasis
Cplex.readMipStart
Cplex.writeMipStart
Cplex.readParam
Cplex.writeParam
Cplex.writeConflict

The following methods are provided to solve and analyze the model, solution and mipstart:

Methods for solving and analyzing a model

Cplex.solve
Cplex.populate
Cplex.feasOpt
Cplex.refineConflict
Cplex.refineMipStartConflict
Cplex.terminate

The following methods are provided to set and query parameters:

Methods for managing parameters

Cplex.tuneParam
Cplex.setDefault
Cplex.getChgParam

Although a model can be modified by manipulating the MATLAB data structures directly, the following functions are provided to make modifications easier:

Methods for manipulating the model directory

Cplex.addCols
Cplex.addRows
Cplex.delCols
Cplex.delRows
Cplex.addSOSs
Cplex.addQCs
Cplex.addIndicators

Setting and querying parameters in the CPLEX Class API

To set parameters using the CPLEX Class API, you set the current values of the fields in the Param structure property of the Cplex class. For example, to set a node limit of 400 and instruct CPLEX to use traditional branch and cut style search:

```
cpx.Param.mip.limits.nodes.Cur=400;  
cpx.Param.mip.strategy.search.Cur=1;
```

Tip:

If you are already familiar with the names of parameters in the Interactive Optimizer, then you quickly recognize names of parameters in CPLEX for MATLAB. For example, the command “set mip limits nodes 1” in the Interactive Optimizer corresponds to “cpx.Param.mip.limits.nodes.Cur = 1;” in the CPLEX Class API where cpx is an instance of the Cplex class.

To assist in setting parameters, auto-completion of parameter names is available in the MATLAB environment.

Chapter 6. Programming tips

As you model and solve mathematical programming problems, you may find it useful to refer to the documentation to find answers to your questions.

As you model and solve mathematical programming problems with CPLEX for MATLAB, you may find it useful to refer to the documentation to find answers to your questions.

Some common questions are answered in this documentation as well as the *CPLEX User's Manual* available in your CPLEX distribution.

- **How do I use the MATLAB sparse matrix format with the CPLEX for MATLAB functions and classes?**

Either sparse or dense format can be used in the connector in all places where double matrices and vectors are accepted as arguments.

- **How do I understand and deal with my infeasible (or unbounded) model?**

The section *Infeasibility and unboundedness* of the *CPLEX User's Manual* documents tools to help you analyze the source of the infeasibility in a model: the preprocessing reduction parameter for distinguishing infeasibility from unboundedness, the conflict refiner for detecting minimal sets of mutually contradictory bounds and constraints, and FeasOpt for repairing infeasibilities.

- **How do I obtain a pool of multiple solutions?**

The section *Discrete optimization > Solution pool: generating and keeping multiple solutions* of the *CPLEX User's Manual* introduces the solution pool for storing multiple solutions to a mixed integer programming problem (MIP) and explains techniques for generating and managing those solutions.

- **How do I get this difficult model to solve?**

The section *Programming considerations > Tuning tool* of the *User's Manual* documents the tuning tool, a utility to aid you in improving the performance of your optimization applications, analyzes a model or a group of models and suggests a suite of parameter settings for you to use that provide better performance than the default parameter settings for your model or group of models.

- **How do I invoke parallel threads to solve my model faster?**

The section *Advanced programming techniques > Parallel optimizers* of the *CPLEX User's Manual* documents the CPLEX parallel optimizers.

- **How do I set algorithm control parameters?**

The *CPLEX Parameters Reference Manual* lists all of the parameters and explains their settings, and additional information about using parameters is given in Chapter 7, "Using parameters," on page 17. To learn about setting parameters when using the toolbox functions, see the function `cplexoptimset`. For information about setting parameters while using the `Cplex` class API, see `Cplex.Param`.

- **How do I check the consistency of my data?**

There is a consistency check which can be controlled by the `datacheck` parameter. The default value of `datacheck` is off (0), which can improve the performance in the case of valid model data. If the `datacheck` is warn (1) or

assist (2), then a detailed error message will be displayed in the case of invalid data. For information about setting parameters, see `Cplex.Param` and `cplexoptimset`.

- **Can I use single precision for floating point numbers with CPLEX for MATLAB?**

No. All floating arrays, vectors and other constructs must be of type double precision. Using single precision may result in undefined behavior.

Chapter 7. Using parameters

Parameters, accessible and manageable by users, control the behavior of CPLEX.

Though the default settings for CPLEX will prove sufficient to solve most problems, CPLEX offers a variety of parameters to control various algorithmic choices. The methods for accessing and setting parameters differs based on which API you are using.

In the CPLEX for MATLAB Optimization Toolbox, parameters are managed as an options structure. This structure can be passed as an argument to any of the toolbox methods for solving optimization problems. An options structure is created using the method `cplexoptimset`. The values of the parameters in the options structure can be accessed with the method `cplexoptimset`. Once the structure has been created, the parameter fields of the structure can be updated directly.

For example, to set a node limit of 400 and instruct CPLEX to use traditional branch and cut style search, you can use either of the following:

```
options = cplexoptimset('mip.limits.nodes', 400, 'mip.strategy.search', 1);
```

or

```
opt = cplexoptimset('cplex');  
opt.mip.limits.nodes=400;  
opt.mip.strategy.search=1;
```

In order to provide compatibility, the CPLEX for MATLAB Optimization Toolbox also accepts the parameters that are used with the MATLAB Optimization Toolbox. These "compatibility parameters" are listed in the reference manual entry for `cplexoptimset`; listed there are the CPLEX parameters that are equivalent to these compatibility parameters. As an example of using these parameters, to turn on the optimizer output and set a node limit of 400 using the compatibility parameters, you can write:

```
options = cplexoptimset('Display', 'on', 'MaxNodes', 400);
```

Three parameters that are available in the CPLEX for MATLAB Optimization Toolbox do not have equivalents in the other CPLEX APIs. These are:

- The parameter `ExportModel` (or `exportmodel`) takes a file name as an argument. The default is the empty string (""). For example, `opt = cplexoptimset('exportmodel', 'myModel.lp');` can be used to set the name of the exported model.
- The parameter `Display` controls the level of display and takes the following values: 'off' | 'on' | 'iter'.
- The parameter `Algorithm` takes the values: 'interior-point' | 'primal' | 'dual' | 'auto'.

In the Cplex Class API, the structure `Cplex.Param` is provided for managing parameters. Each parameter is a field of the `Cplex.Param` structure. In turn, each field is a structure with six fields, as described in the reference manual entry for `Cplex.Param`.

For example, to set a node limit of 400 and instruct CPLEX to use traditional branch and cut style search, you use:

```
cpex.Param.mip.limits.nodes.Cur=400;  
cpex.Param.mip.strategy.search.Cur=1;
```

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those

websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Copyright notice

Describes general use restrictions and trademarks related to this document and the software described in this document.

© Copyright IBM Corp. 1987, 2017

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other company, product, or service names may be trademarks or service marks of others.

Additional registered trademarks, copyrights, licenses

IBM ILOG CPLEX states these additional registered trademarks, copyrights, and acknowledgements.

Python is a registered trademark of the Python Software Foundation.

MATLAB is a registered trademark of The MathWorks, Inc.

OpenMPI is distributed by The Open MPI Project under the New BSD license and copyright 2004 - 2012.

MPICH2 is copyright 2002 by the University of Chicago and Argonne National Laboratory.

Index

M

MATLAB toolbox 7



Printed in USA