Matlab

+ Yalmip + Gurobi

助教　董九舟

清华大学电机系

djz23@mails.tsinghua.edu.cn

Energy Intelligence Laboratory
清华大学电机系·智慧能源课题组

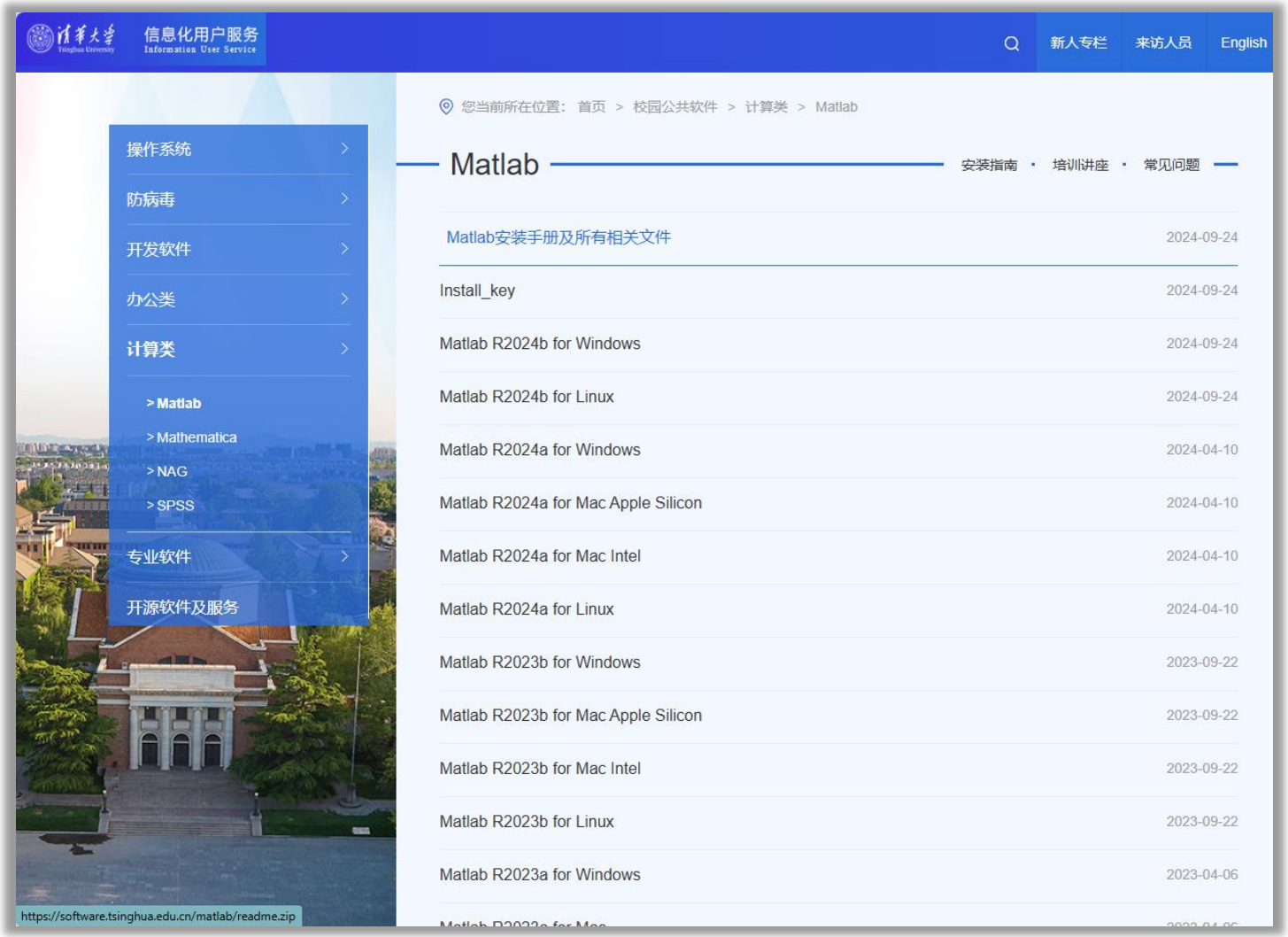
MATLAB 优化工具箱

MATLAB 优化工具箱

➢ Matlab 的安装

学校提供Matlab正版软件：

1. 登录清华大学信息化用户服务平台（https://its.tsinghua.edu.cn/）
2. → 公共软件 → 计算类 → Matlab
3. 下载“Matlab安装手册及所有相关文件”，按照引导进行安装

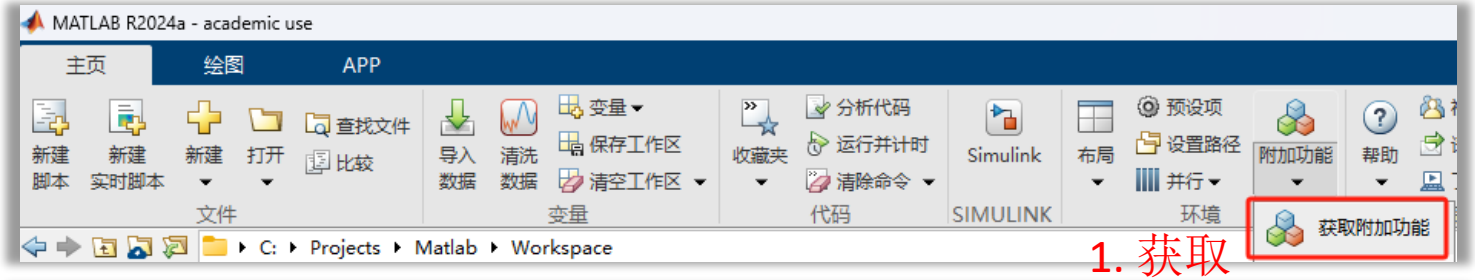


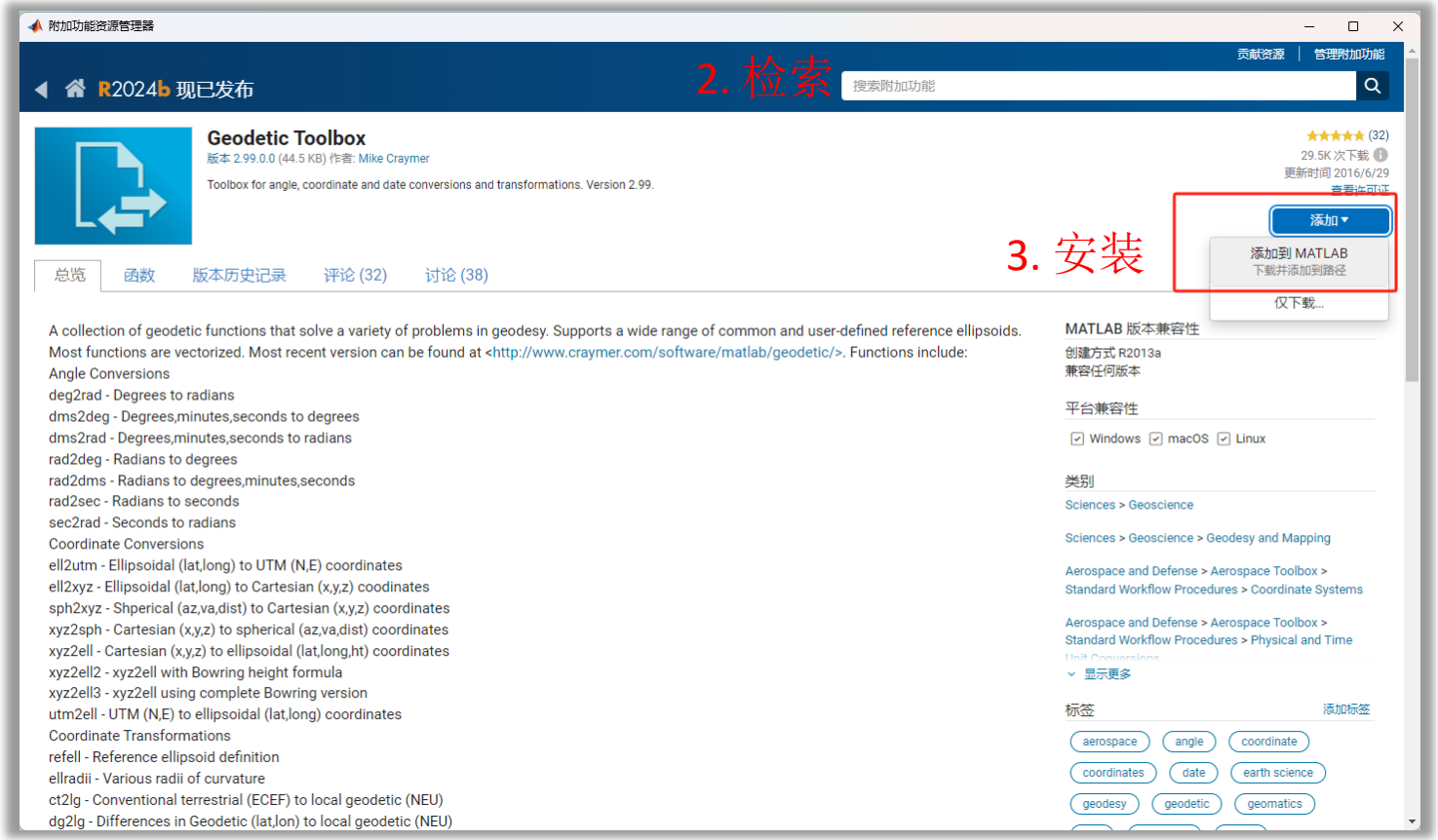
MATLAB 优化工具箱

➢ 优化工具箱简介

Matlab自身可以求解优化问题，需要使用Optimization Toolbox，可在安装Matlab时勾选此工具箱或自行添加

1. → 附加功能 → 获取附加功能
2. 检索“Optimization Toolbox”，点击“添加到MATLAB”（如右图）





MATLAB 优化工具箱

➢ 优化工具箱简介

优化工具箱的使用详见官方说明

https://cn.mathworks.com/products/optimization/features.html

具体函数的参数含义、功能说明、使用示例等可在官方帮助文档中学习

https://ww2.mathworks.cn/help/index.html

YALMIP + GUROBI

YALMIP + GUROBI

➢ 什么是 Yalmip、Gurobi？

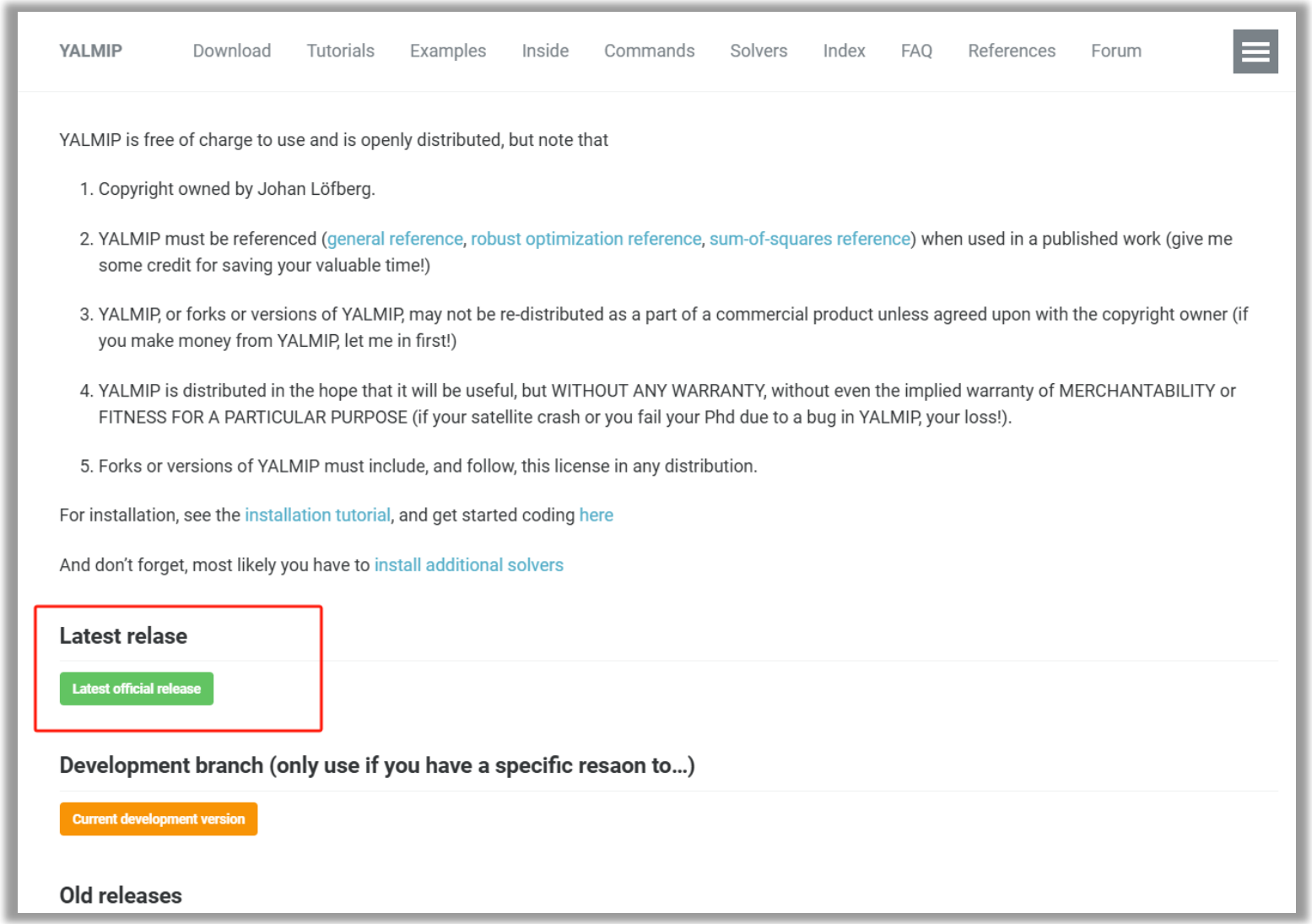
- Yalmip 是一种优化模型**建模语言**
  - 是一个开源的 Matlab 库，提供了调用其他求解器的 API
- Gurobi是商业化的**优化求解器**
  - 商业求解器有多种，不同求解器“擅长”不同类型的优化问题
- Yalmip 与 Gurobi 的关系：

注：Yalmip本身不具有求解优化问题的能力！

YALMIP + GUROBI

➢ 安装教程-Yalmip

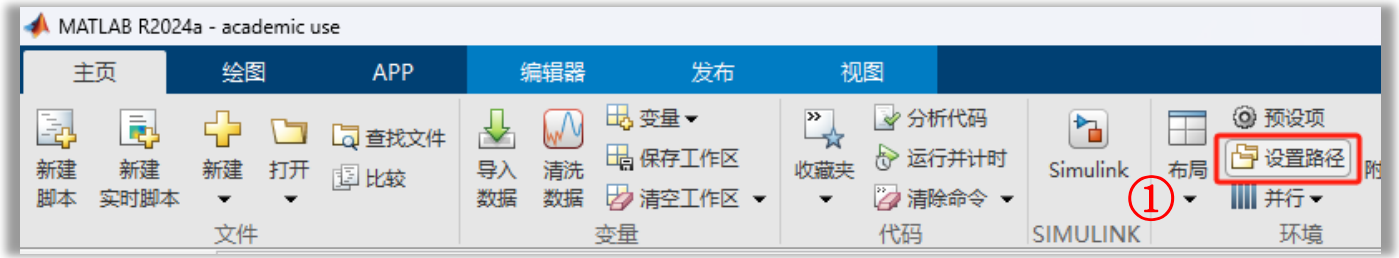
1. Yalmip官网下载
   https://yalmip.github.io/download/
   （下载得到YALMIP-master.zip）
2. 将YALMIP-master.zip解压到合适的路径下（例如MATLAB安装路径下的toolbox文件夹，此步非必要）

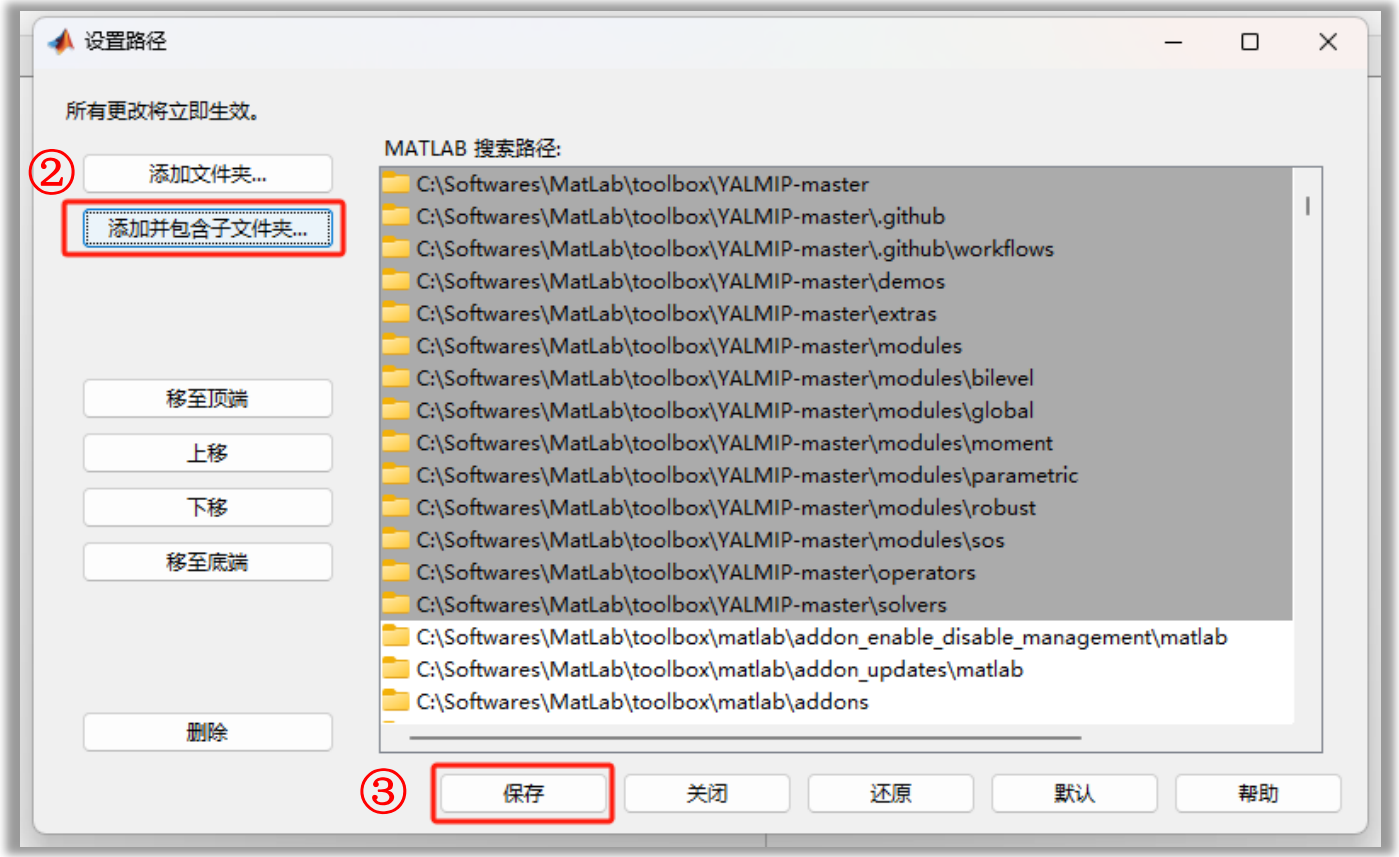


YALMIP + GUROBI

➢ 安装教程-Yalmip

3. Matlab-主页 → 设置路径
4. “添加并包含子文件夹” → 选择解压的YALMIP-master → “保存”



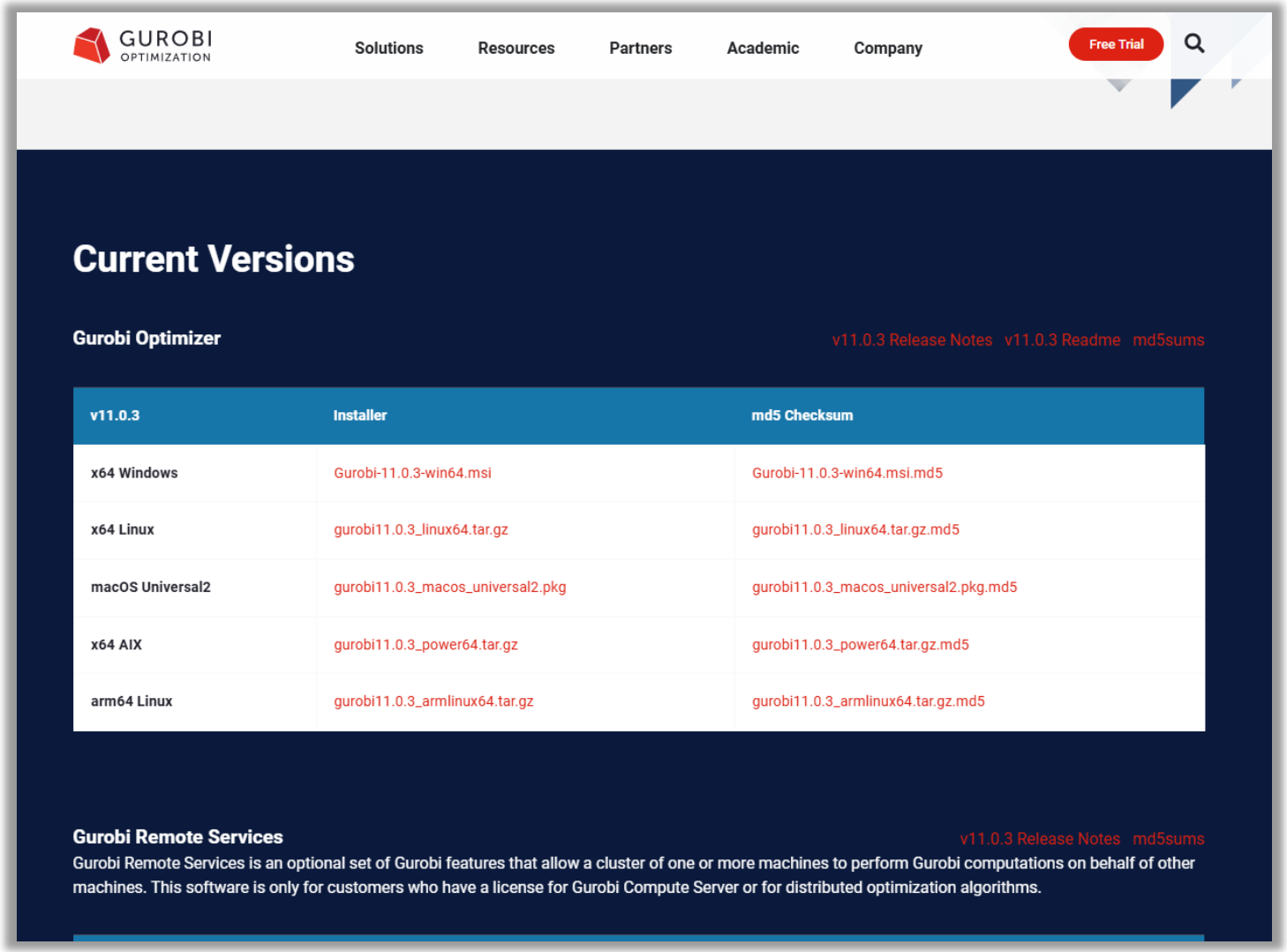


YALMIP + GUROBI

➢ 安装教程-Gurobi

进入官网选择合适的版本下载安装

https://www.gurobi.com/downloads/gurobi-software/



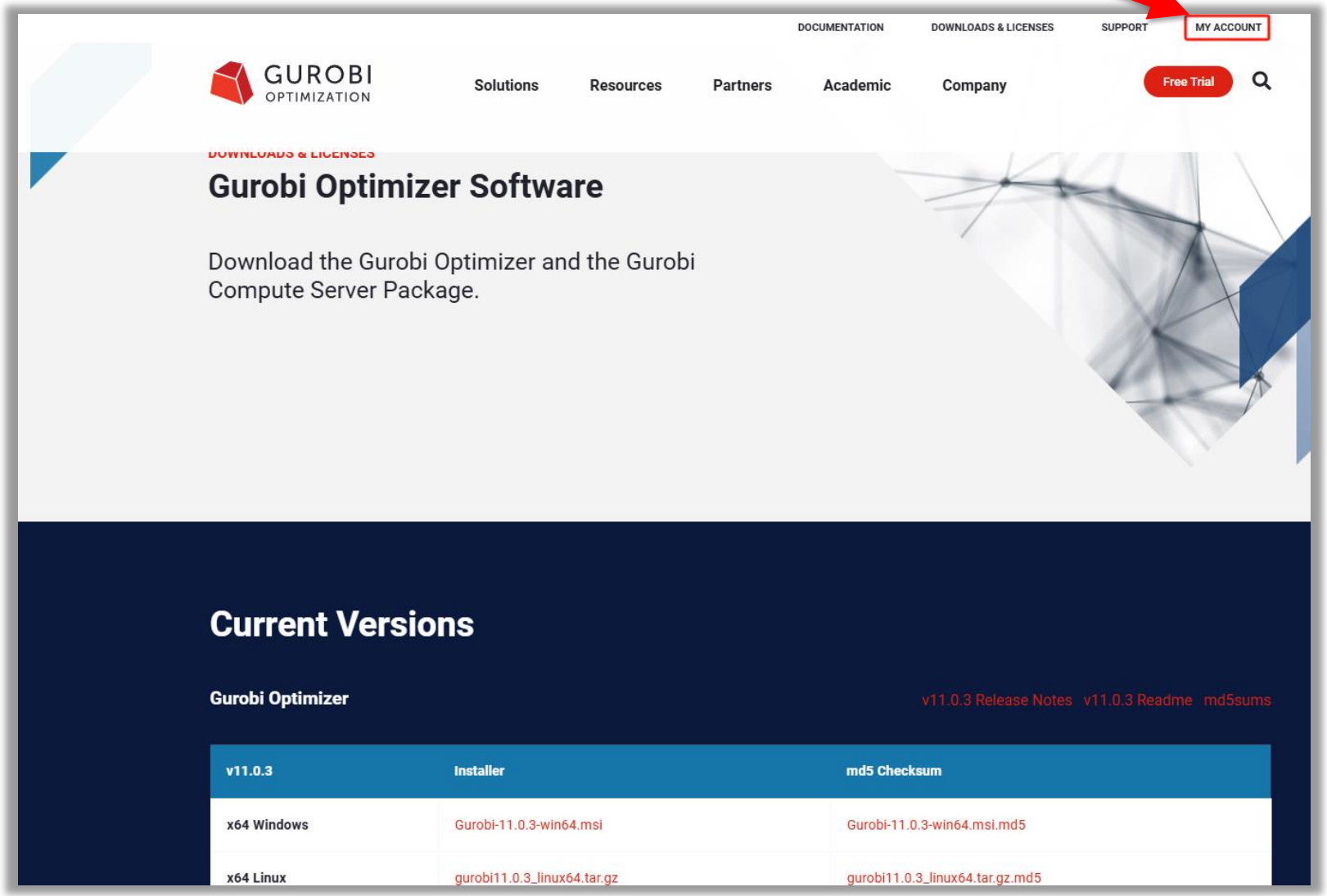
YALMIP + GUROBI

- 若未登录，则此处为REGISTER和LOGIN
- 登录后点击MY ACCOUNT进入个人页面

➢ 安装教程-Gurobi

申请license并激活

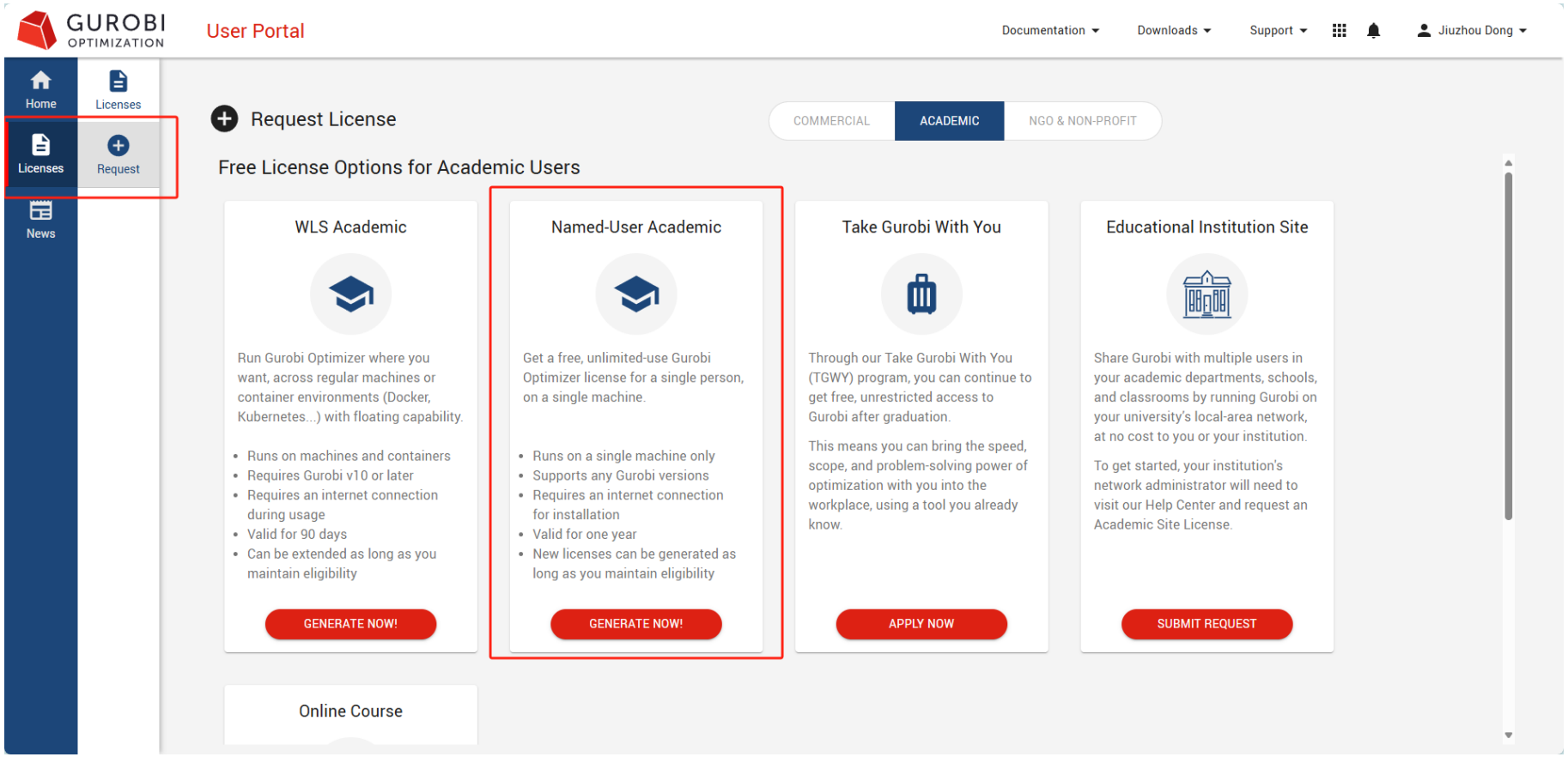
1. 登录或注册账号
2. 转到个人页面，点击 License → Request，申请“Named-User Academic”凭证



YALMIP + GUROBI

➢ 安装教程-Gurobi

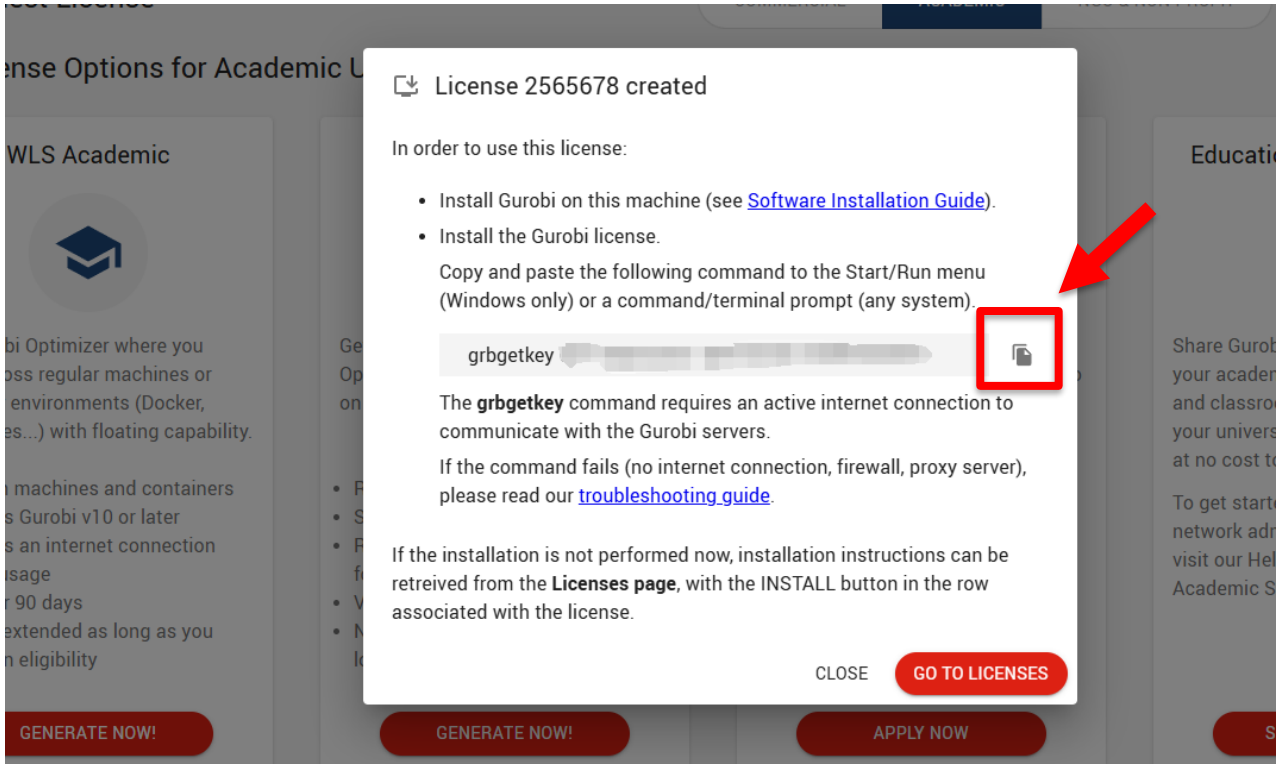
2. 转到个人页面，点击 License → Request，申请“Named-User Academic”凭证

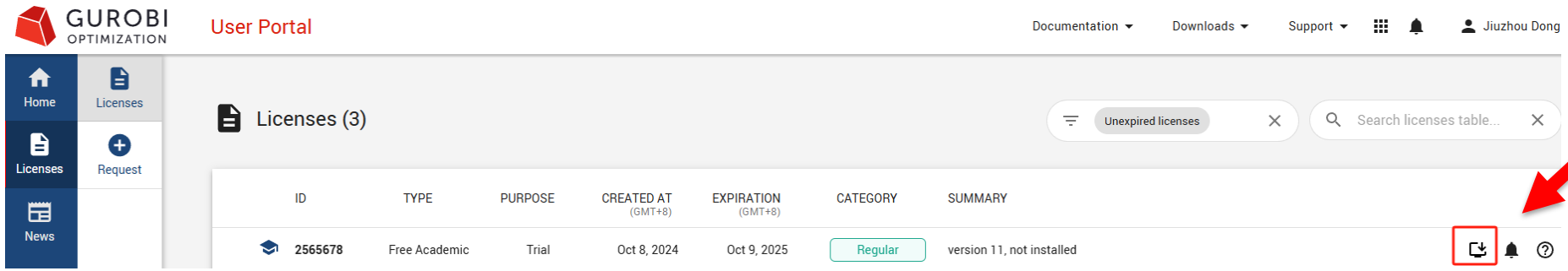


YALMIP + GUROBI

➢ 安装教程-Gurobi

3. 申请成功可获得一个key，点击复制



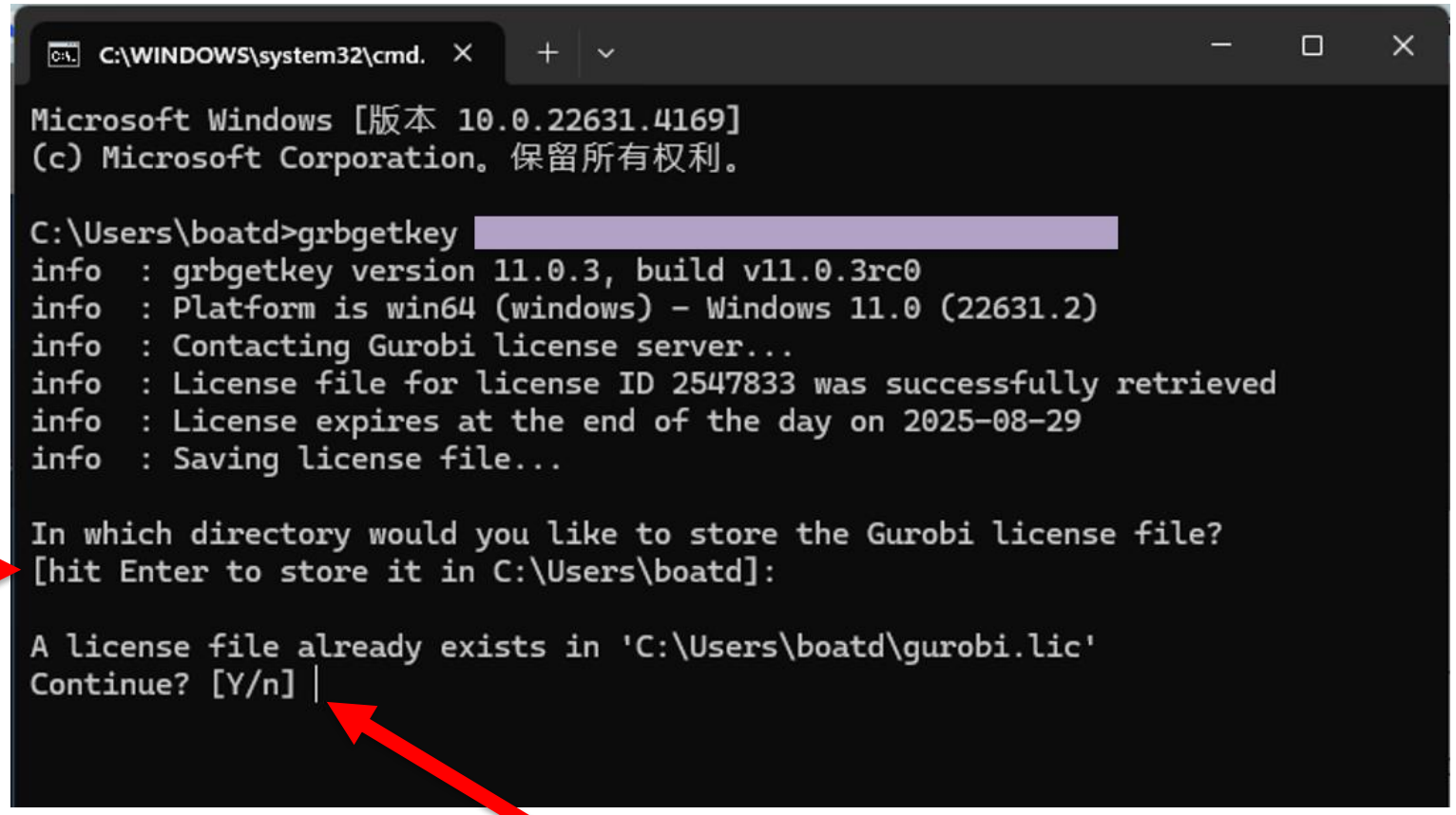


可在license中查看已申请的凭证

YALMIP + GUROBI

➢ 安装教程-Gurobi

4. 打开命令提示符（Win+R键入cmd而后回车），CTRL+V输入复制的key，回车激活
5. 完成安装和激活后，**在Matlab中设置路径**（Matlab-主页→设置路径→“添加并包含子文件夹”→选择<Gurobi根目录\win64\matlab>→“保存”

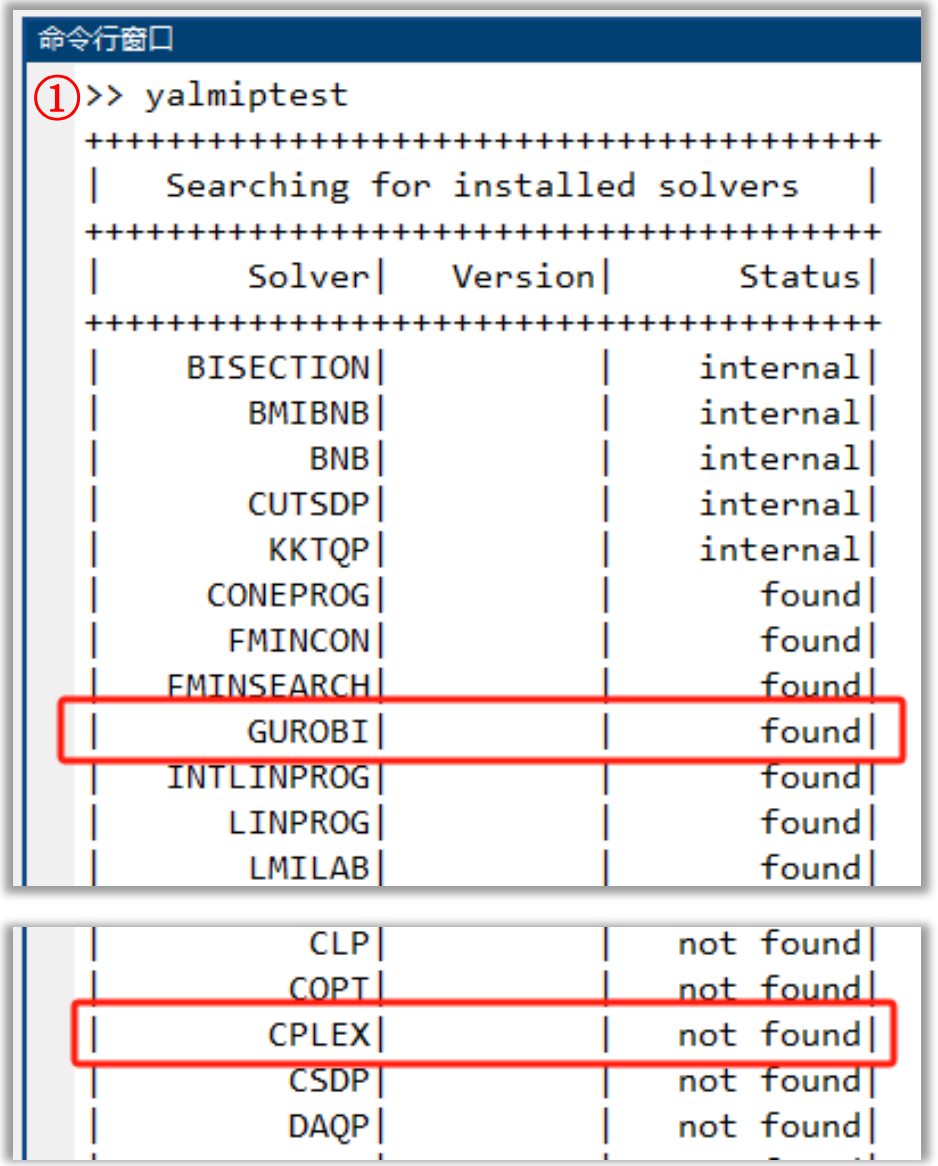


按照提示操作即可，此处因设备已激活过而询问是否覆盖

YALMIP + GUROBI

➢ 安装教程-测试是否安装成功

1. 在Matlab命令行键入“yalmiptest”，会显示已安装的求解器
2. 如右图所示，安装了Gurobi，但未安装Cplex
3. 可按任意键继续，完成测试



② 按任意键继续

优化问题一般步骤

针对具体工程问题建立优化设计的数学模型

不等式约束条件表示成$g(X)\geq 0$的形式

建立目标函数文件

文件内容： 输入参数、目标函数表达式等
存储： 以自定义的目标函数文件名存储在文件夹中

建立约束函数文件

文件内容： 输入参数、约束函数表达式等
存储： 以自定义的约束函数文件名存储在文件夹中

建立调用优化工具函数的命令文件

分析优化模型，选择适用的优化工具函数
文件内容： 初始点，设计变量的边界约束条件，运算结果输出等内容
存储： 以自定义的命令文件名存储于文件夹中。

将优化设计的命令文件复制到MATLAB命令窗口中进行运算求解

使用入门

以 Yalmip 例程1进行说明（更多例程详见 https://yalmip.github.io/tutorial/basics/）

1. 创建决策变量
2. 添加约束条件
3. 定义目标函数
4. 设置求解参数并运行

test.m

```
% Define variables
x = sdpvar(10,1);

% Define constraints
Constraints = [sum(x) <= 10, x(1) == 0, 0.5 <= x(2) <= 1.5];
for i = 1 : 7
  Constraints = [Constraints, x(i) + x(i+1) <= x(i+2) + x(i+3)];
end

% Define an objective
Objective = x'*x+norm(x,1);

% Set some options for YALMIP and solver
options = sdpsettings('verbose',1,'solver','quadprog','quadprog.maxiter',100);

% Solve the problem
sol = optimize(Constraints,Objective,options);

% Analyze error flags
if sol.problem == 0
 % Extract and display value
 solution = value(x);
else
 disp('Hmm, something went wrong!');
 sol.info
 yalmiperror(sol.problem)
end
```

使用入门-创建决策变量

主要使用三种决策变量：

- 实数型变量-sdpvar
- 整数型变量-intvar
- 0-1变量-binvar

可将Matlab内置的函数应用于决策变量，例如：

1. 生成对角阵： D = diag(b);
2. 进行矩阵(向量运算)： f = c*b; % 得到2x3矩阵
3. 元素求和： S = sum(d, 'all');
4. ……

创建变量可指定变量的维度：

- 标量： a = sdpvar(1)
- 向量： b = sdpvar(1,3); c = sdpvar(2,1);
- 矩阵： d = sdpvar(3,4); e = sdpvar(2,2);

1. 效果与“e = sdpvar(2)”相同
2. 当创建方阵时，默认为对称矩阵，若要创建非对称的矩阵则用“e = sdpvar(2,2,**'full'**)”

使用入门

以 Yalmip 例程1进行说明（更多例程详见 https://yalmip.github.io/tutorial/basics/）

1. 创建决策变量
2. **添加约束条件**
3. 定义目标函数
4. 设置求解参数并运行

注意：等式约束要用“==”符号

test.m

```
% Define variables
x = sdpvar(10,1);

% Define constraints
Constraints = [sum(x) <= 10, x(1) == 0, 0.5 <= x(2) <= 1.5];
for i = 1 : 7
  Constraints = [Constraints, x(i) + x(i+1) <= x(i+2) + x(i+3)];
end

% Define an objective
Objective = x'*x+norm(x,1);

% Set some options for YALMIP and solver
options = sdpsettings('verbose',1,'solver','quadprog','quadprog.maxiter',100);

% Solve the problem
sol = optimize(Constraints,Objective,options);

% Analyze error flags
if sol.problem == 0
 % Extract and display value
 solution = value(x);
else
 disp('Hmm, something went wrong!');
 sol.info
 yalmiperror(sol.problem)
end
```

使用入门

以 Yalmip 例程1进行说明（更多例程详见 https://yalmip.github.io/tutorial/basics/）

1. 创建决策变量
2. 添加约束条件
3. **定义目标函数**
4. 设置求解参数并运行

注意：Yalmip默认问题为最小化问题

test.m

```
% Define variables
x = sdpvar(10,1);

% Define constraints
Constraints = [sum(x) <= 10, x(1) == 0, 0.5 <= x(2) <= 1.5];
for i = 1 : 7
  Constraints = [Constraints, x(i) + x(i+1) <= x(i+2) + x(i+3)];
end

% Define an objective
Objective = x'*x+norm(x,1);

% Set some options for YALMIP and solver
options = sdpsettings('verbose',1,'solver','quadprog','quadprog.maxiter',100);

% Solve the problem
sol = optimize(Constraints,Objective,options);

% Analyze error flags
if sol.problem == 0
 % Extract and display value
 solution = value(x);
else
 disp('Hmm, something went wrong!');
 sol.info
 yalmiperror(sol.problem)
end
```

使用入门

以 Yalmip 例程1进行说明（更多例程详见 https://yalmip.github.io/tutorial/basics/）

1. 创建决策变量
2. 添加约束条件
3. 定义目标函数
4. **设置求解参数并运行**

```
test.m
1  % Define variables
2  x = sdpvar(10,1);
3
4  % Define constraints
5  Constraints = [sum(x) <= 10, x(1) == 0, 0.5 <= x(2) <= 1.5];
6  for i = 1 : 7
7    Constraints = [Constraints, x(i) + x(i+1) <= x(i+2) + x(i+3)];
8  end
9
10 % Define an objective
11 Objective = x'*x+norm(x,1);
12
13 % Set some options for YALMIP and solver
14 options = sdpsettings('verbose',1,'solver','quadprog','quadprog.maxiter',100);
15
16 % Solve the problem
17 sol = optimize(Constraints,Objective,options);
18
19 % Analyze error flags
20 if sol.problem == 0
21  % Extract and display value
22  solution = value(x);
23 else
24  disp('Hmm, something went wrong!');
25  sol.info
26  yalmiperror(sol.problem)
27 end
```

求解函数的调用，三个输入参数分别为约束、目标、求解参数

注意：主要设置求解器域'solver'，其他选项含义可在Matlab命令行中输入“doc sdpsettings”查看

使用入门

其他常用函数：

- 检查约束条件是否起作用（余量大小）：check
- 查看变量或表达式的值：value
- 给变量赋值（一般用于调试）：assign

Yalmip 机组组合例程

- https://yalmip.github.io/example/unitcommitment/