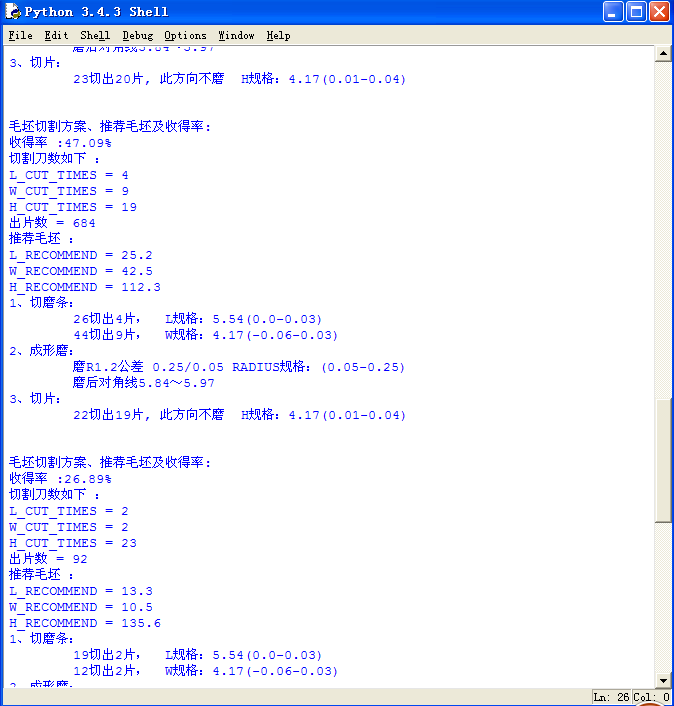
桓博士、张经理：

你们好，我在网上找了下，python是跨平台的，不过在使用之前都需要先安装；是否使用它，这个可能也要列入考量；

另：

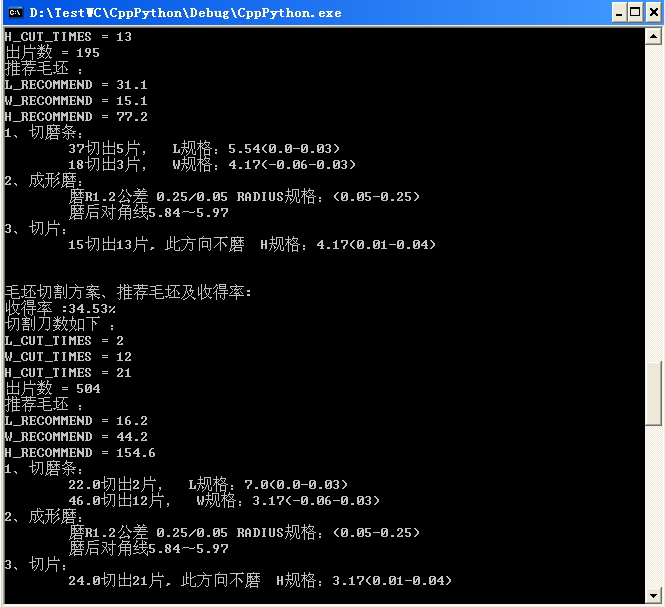
我先说下我自己的想法，昨天讨论的用脚本来执行解决方案确实是可行的；我把您周一发的邮件里的那个excel里描述的工艺，写成一个脚本，用的参数都是文档里的，计算出来的结果也与文档展示的一致，下图里展示的收得率47.09%的就是文档里的那个例子，其他生成的结果是Sheet2毛坯库里的其他毛坯生成的结果，W、L、H等用的都是文档里指定的，用其他参数会生成不一样的结果，计算方式都是一样；

其次就是：

C++里调用其实也很简单，尤其我们这个工艺，传入脚本的参数都比较简单，这个文档里描述的要接收的外部参数是16个，相对来说，规则定义，即脚本的生成就比较复杂，虽然python足够简单，但对绝大多数使用者来说都难免出错；而这个对使用者来说是不好的。我觉得如果可能的话，他们现有工艺的脚本都应该由我们来完成。

在用户熟悉之后，可以自己添加或修改规则。

因为我对这个C++调用python脚本不是很熟悉，测试了很久才正确运行并获得该输出结果，下面34.53%的是通过C++调用函数，修改了参数(L和W)之后的生成的工艺方案：



工艺信息系统一般包含产品、零组件、工艺、工序、工步等核心对象类，并由它们关联出如材料、设备、工装等对象类，各对象类有自己的属性与方法，由此形成面向对象的工艺信息建模体系。

/\*

跟进点：

1.Python是否支持外部函数定义

2.输出XML的定义

3.输出的多对象如何创建

\*/

Python手册里介绍的用c语言扩展python：

最后，python的源代码中附带了一个叫做example\_nt的例子，可以参考一样，完整的扩展代码如下：

#include "Python.h"

static PyObject \* ex\_foo(PyObject \*self, PyObject \*args)

{

    printf("Hello, world/n");

    Py\_INCREF(Py\_None);

    return Py\_None;

}

static PyMethodDef example\_methods[] = {

    {"foo", ex\_foo, METH\_VARARGS, "foo() doc string"},

    {NULL, NULL}

};

PyMODINIT\_FUNC initexample(void)

{

    Py\_InitModule("example", example\_methods);

}

它必须按照这种方式生成动态库，然后把生成的.dll 修改为.pyd，放入python安装目录的lib之后，脚本中才可以import这个模块，然后调用模块里面定义的c函数。简单的说，就是要想引用定义好的外部函数，必须依照这种方式来改写，才可以在脚本里使用。