

# 湖州市地区代理各种工程计算书吊顶计算书墙面挂石材计算书

产品名称	湖州市地区代理各种工程计算书吊顶计算书墙面挂石材计算书
公司名称	金算盘工程造价咨询有限公司
价格	499.00/套
规格参数	工程造价:预结算 标书制作:图纸绘制 全国:全专业
公司地址	山东省济南市市中区七贤街道依山新居29号楼4单元1601
联系电话	15552548594 15552548594

## 产品详情

```
% Hungarian.m

%指派问题的匈牙利算法，输入矩阵，a(ij)为i指派给j,第i人干第j个工作

function [Matching,Cost] = Hungarian(Perf)

%

% 用于在给定MxN边缘的情况下找到小边缘权重匹配的函数

% 使用匈牙利算法的权重矩阵权重。

%

% Inf的边权重表示由其给出的顶点对

% 位置没有相邻的边缘。

%

% MATCHING返回一个MxN矩阵，1表示匹配的位置

% COST返回小匹配的成本

%
```

%

%

% 初始化变量

Matching = zeros(size(Perf));

% ? 通过删除任何未连接的顶点来压缩性能矩阵

% ? 会增加算法的速度

% 找到每个已连接的列中的数字

num\_y = sum(~isinf(Perf),1);%?

% 找到连接的每一行中的数字

num\_x = sum(~isinf(Perf),2);%?

% 找到隔离的列（顶点）和行（顶点）

x\_con = find(num\_x~=0);% find()返回矩阵或向量中非零元素的索引

y\_con = find(num\_y~=0);

% ?汇总凝聚性能矩阵

P\_size = max(length(x\_con),length(y\_con));

P\_cond = zeros(P\_size);

P\_cond(1:length(x\_con),1:length(y\_con)) = Perf(x\_con,y\_con);

if isempty(P\_cond)

Cost = 0;

return

end

% 确保存在完美匹配

%计算边缘矩阵的形式

Edge = P\_cond;

Edge(P\_cond~=Inf) = 0;

% 找到Edge Matrix中的缺陷 ( CNUM )

cnum = min\_line\_cover(Edge);

% 投影额外的顶点和边缘，以便完美匹配

% exists

Pmax = max(max(P\_cond(P\_cond~=Inf)));%不等于无穷值的其他值的大值

P\_size = length(P\_cond)+cnum;

P\_cond = ones(P\_size)\*Pmax;

P\_cond(1:length(x\_con),1:length(y\_con)) = Perf(x\_con,y\_con);

%\*\*\*\*\*

% 主程序：执行步骤的控制

exit\_flag = 1;

stepnum = 1;

while exit\_flag

switch stepnum

case 1 % 找到每行的小数字并且每一行减去它

[P\_cond,stepnum] = step1(P\_cond);

case 2 % 划线

[r\_cov,c\_cov,M,stepnum] = step2(P\_cond);

case 3 % 判断是不是划线覆盖所有值

[c\_cov,stepnum] = step3(M,P\_size);

case 4 % 调整划线，使覆盖所有0

```

[M,r_cov,c_cov,Z_r,Z_c,stepnum] = step4(P_cond,r_cov,c_cov,M);

case 5 % 未划线的0所在行没有星标0

[M,r_cov,c_cov,stepnum] = step5(M,Z_r,Z_c,r_cov,c_cov);

case 6 % 未划线的0所在行有星标0,变换矩阵以增加0

[P_cond,stepnum] = step6(P_cond,r_cov,c_cov);

case 7

    exit_flag = 0;

end

end

%删除所有虚拟卫星和目标并取消激活

%匹配原始性能矩阵的大小。

Matching(x_con,y_con) = M(1:length(x_con),1:length(y_con));

Cost = sum(sum(Perf(Matching==1)));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% STEP 1: 找到每行的小数字并且每一行减去它

% Find the smallest number of zeros in each row

%      and subtract that minimum from its row

function [P_cond,stepnum] = step1(P_cond)

P_size = length(P_cond);% 返回行数和列数较大的那个值

% Loop throught each row

for ii = 1:P_size

    rmin = min(P_cond(ii,:));

    P_cond(ii,:) = P_cond(ii,.)-rmin;

```

end

stepnum = 2;

%\*\*\*\*\*

% STEP 2: 划线：找到一个0，如果这个0所在行没有被星标的0，从这个0开始。重复每个0.

% Find a zero in P\_cond. If there are no starred zeros in its

% column or row start the zero. Repeat for each zero

%\*\*\*\*\*

function [r\_cov,c\_cov,M,stepnum] = step2(P\_cond)

% Define variables

P\_size = length(P\_cond);

r\_cov = zeros(P\_size,1); % 显示是否覆盖行的向量

c\_cov = zeros(P\_size,1); % 显示是否覆盖列的向量

M = zeros(P\_size); %一个掩码，显示一个位置是星号还是已准备好

for ii = 1:P\_size

for jj = 1:P\_size

if P\_cond(ii,jj) == 0 && r\_cov(ii) == 0 && c\_cov(jj) == 0

M(ii,jj) = 1;

r\_cov(ii) = 1;

c\_cov(jj) = 1;

end

end

end

% Re-initialize the cover vectors

r\_cov = zeros(P\_size,1); % A vector that shows if a row is covered

c\_cov = zeros(P\_size,1); % A vector that shows if a column is covered

stepnum = 3;

% STEP 3: 从星标的0开始划线，如果所有列都被覆盖，那么这个匹配是大匹配。

% Cover each column with a starred zero. If all the columns are

% covered then the matching is maximum

function [c\_cov,stepnum] = step3(M,P\_size)

c\_cov = sum(M,1);

if sum(c\_cov) == P\_size

stepnum = 7;

else

stepnum = 4;

end

% STEP 4: 调整划线，使覆盖所有0：找到一个未被覆盖的0，置顶它。置顶0所在行没有

%

星标0，则进入第5步———。否则，有星标0，对这一行划线，并且取消包含这个星标0的列的线。

% 重复这一步骤直到没有剩余的未被覆盖的0。保留这个小覆盖值并且进入步骤6.

% Find a noncovered zero and prime it. If there is no starred

% zero in the row containing this primed zero, Go to Step 5.

% Otherwise, cover this row and uncover the column containing

% the starred zero. Continue in this manner until there are no

```
%      uncovered zeros left. Save the smallest uncovered value and
```

```
%      Go to Step 6.
```

```
function [M,r_cov,c_cov,Z_r,Z_c,stepnum] = step4(P_cond,r_cov,c_cov,M)
```

```
P_size = length(P_cond);
```

```
zflag = 1;
```

```
while zflag
```

```
    % Find the first uncovered zero
```

```
    row = 0; col = 0; exit_flag = 1;
```

```
    ii = 1; jj = 1;
```

```
    while exit_flag
```

```
        if P_cond(ii,jj) == 0 && r_cov(ii) == 0 && c_cov(jj) == 0
```

```
            row = ii;
```

```
            col = jj;
```

```
            exit_flag = 0;
```

```
        end
```

```
        jj = jj + 1;
```

```
        if jj > P_size; jj = 1; ii = ii+1; end
```

```
        if ii > P_size; exit_flag = 0; end
```

```
    end
```

```
% If there are no uncovered zeros go to step 6
```

```
if row == 0
```

```
    stepnum = 6;
```

```
    zflag = 0;
```

```
    Z_r = 0;
```

```

Z_c = 0;

else

% Prime the uncovered zero

M(row,col) = 2;

% If there is a starred zero in that row

% Cover the row and uncover the column containing the zero

if sum(find(M(row,:)==1)) ~= 0

    r_cov(row) = 1;

    zcol = find(M(row,:)==1);

    c_cov(zcol) = 0;

else

    stepnum = 5;

    zflag = 0;

    Z_r = row;

    Z_c = col;

end

end

end

% STEP 5: 调整划线

%    Z0表示第4步中发现的未被覆盖的置顶0；Z1表示Z0所在列的星标0（如果有）；

%    Z2表示Z1所在行的置顶0（总是会有一个）。

%    直到置顶0所在列没有星标0时终止。

%    取消星标这个系列的每个星标0，星标这个系列的每个置顶0

%    擦除所有置顶，取消所有划线。

%    返回步骤3.

```



```
% Construct a series of alternating primed and starred zeros as
% follows. Let Z0 represent the uncovered primed zero found in Step 4.
% Let Z1 denote the starred zero in the column of Z0 (if any).
% Let Z2 denote the primed zero in the row of Z1 (there will always
% be one). Continue until the series terminates at a primed zero
% that has no starred zero in its column. Unstar each starred
% zero of the series, star each primed zero of the series, erase
% all primes and uncover every line in the matrix. Return to Step 3.
```

```
function [M,r_cov,c_cov,stepnum] = step5(M,Z_r,Z_c,r_cov,c_cov)
```

```
zflag = 1;
```

```
ii = 1;
```

```
while zflag
```

```
% Find the index number of the starred zero in the column
```

```
rindex = find(M(:,Z_c(ii))==1); %置顶0所在列的星标0
```

```
if rindex > 0
```

```
% Save the starred zero
```

```
ii = ii+1;
```

```
% Save the row of the starred zero
```

```
Z_r(ii,1) = rindex;
```

```
% The column of the starred zero is the same as the column of the
```

```
% primed zero
```

```
Z_c(ii,1) = Z_c(ii-1);
```

```
else
```

```
zflag = 0;
```

```
end
```

```
% Continue if there is a starred zero in the column of the primed zero
```

```
if zflag == 1;
```

```
    % Find the column of the primed zero in the last starred zeros row
```

```
    cindex = find(M(Z_r(ii),:)==2);% 上一个星标0所在行有无置顶0
```

```
    ii = ii+1;
```

```
    Z_r(ii,1) = Z_r(ii-1);
```

```
    Z_c(ii,1) = cindex;
```

```
end
```

```
end
```

```
% UNSTAR all the starred zeros in the path and STAR all primed zeros
```

```
for ii = 1:length(Z_r)
```

```
    if M(Z_r(ii),Z_c(ii)) == 1
```

```
        M(Z_r(ii),Z_c(ii)) = 0; %星标0取消星标
```

```
    else
```

```
        M(Z_r(ii),Z_c(ii)) = 1; %置顶0全部星标
```

```
    end
```

```
end
```

```
% Clear the covers
```

```
r_cov = r_cov.*0;
```

```
c_cov = c_cov.*0;
```

```
% Remove all the primes
```

```
M(M==2) = 0;
```

```
stepnum = 3;
```

```
% *****
```

```
% STEP 6: 已划线行每个元素加上未覆盖的小值，然后从未划线的列中减去。不改变星标0、
```

```
% 置顶0和划线，返回步骤4.
```

```
% Add the minimum uncovered value to every element of each covered
```

```
% row, and subtract it from every element of each uncovered column.
```

```
% Return to Step 4 without altering any stars, primes, or covered lines.
```

```
function [P_cond,stepnum] = step6(P_cond,r_cov,c_cov)
```

```
a = find(r_cov == 0);
```

```
b = find(c_cov == 0);
```

```
minval = min(min(P_cond(a,b)));
```

```
P_cond(find(r_cov == 1),:) = P_cond(find(r_cov == 1),:) + minval;
```

```
P_cond(:,find(c_cov == 0)) = P_cond(:,find(c_cov == 0)) - minval;
```

```
stepnum = 4;
```

```
function cnum = min_line_cover(Edge)
```

```
% Step 2
```

```
[r_cov,c_cov,M,stepnum] = step2(Edge);
```

```
% Step 3
```

```
[c_cov,stepnum] = step3(M,length(Edge));
```

```
% Step 4
```

```
[M,r_cov,c_cov,Z_r,Z_c,stepnum] = step4(Edge,r_cov,c_cov,M);
```

```
% Calculate the deficiency
```

cnum = length(Edge)-sum(r\_co

流体力学，作为物理学的一个重要分支，研究的是流体在不同条件下的运动规律和性质。流体可以是液体，也可以是气体，其基本特性在于其粘性和连续性。在自然界和工程领域中，流体力学的应用无处不在，从微观的细胞膜流动到宏观的气象变化、海洋环流和航空航天工程等。

流体力学的基本原理主要包括质量守恒定律和动量守恒定律。质量守恒定律指出，在封闭系统中，流体的质量不会发生变化；动量守恒定律则表明，在封闭系统中，流体动量的总和保持不变。这两个定律是流体力学的基础，为后续的数学模型和分析方法提供了理论依据。

流体力学的数学模型和分析方法多种多样，其中常用的是纳维-斯托克斯方程（Navier-Stokes equations）。这是一组描述流体运动和压力的偏微分方程。通过求解这些方程，可以得到流场的速度、压力等物理量。然而，纳维-斯托克斯方程的求解具有高度的复杂性，目前尚无通用的求解方法。科学家们通常采用数值模拟、实验验证和理论分析等方法来研究流体力学问题。

在流体力学的应用中，数学模型和分析方法起着至关重要的作用。例如，在气象学领域，通过数值模拟和理论分析，科学家们可以预测天气变化和气候趋势；在海洋学领域，流体力学的方法被用于研究海洋环流、海底地形和海洋生态系统的演变。此外，在航空航天、汽车制造和能源开发等领域，流体力学的研究也为优化设计和提高效率提供了重要参考。

然而，流体力学的研究仍面临着诸多挑战。首先，纳维-斯托克斯方程的求解问题仍然是国际数学界的一个难题。其次，随着科学研究和技术发展的深入，流体力学的研究对象越来越复杂，例如高速湍流、多相流、生物流体力学等。这些复杂现象的数学描述和物理机制尚待深入研究。此外，实验技术和设备的更新与发展也为流体力学的研究提供了新的机遇和挑战。

总之，流体力学作为物理学的一个重要分支，其基本概念和原理为我们理解和掌握流体的运动规律提供了理论基础。数学模型和分析方法则为实际应用中解决流体力学问题提供了有力工具。在应对未来挑战的过程中，流体力学将继续为科学技术的进步和发展作出贡献。

### 3. 流体力学在物理学中的应用

流体力学在物理学中的应用广泛且深入，涵盖了诸如气象学、海洋学、航空航天、汽车制造、能源开发等多个领域。在气象学领域，流体力学为天气预报和气候研究提供了理论基础。科学家们通过研究大气和海洋中的流体运动，可以预测天气变化、台风路径以及气候趋势。在海洋学领域，流体力学方法被用于揭示海洋环流、海底地形和海洋生态系统的演变。此外，在航空航天领域，流体力学的研究有助于优化飞行器设计和提高飞行性能，例如研究空气动力学、湍流控制和超音速飞行。在汽车制造领域，流体力学为汽车空气动力学设计和燃料效率提升提供了理论支持。在能源开发领域，流体力学在研究水力发电、石油开采和核能工程等方面发挥着关键作用。总之，流体力学在物理学中的应用为各种实际问题的解决提供了有力支撑。

### 4. 流体力学的实验技术与设备

流体力学的实验技术与设备不断发展，为研究者提供了有力的实验手段。实验技术主要包括以下几种：

激光多普勒测速仪（LDV）、颗粒图像测速（PIV）、热线风速仪、电磁流量计、超声波测速仪等。这些设备可以高精度地测量流体的速度、压力、流量等物理量，为流体力学的研究提供了基础数据。

此外，随着计算机技术的飞速发展，计算流体力学（CFD）也成为流体力学研究中不可或缺的实验手段。通过数值模拟，研究者可以在计算机上构建流场，分析流体的运动和性质。与传统实验相比，计算流体力学具有成本低、周期短、可重复性高等优点。同时，它也为研究者提供了实时可视化流体运动的能力，为实验方案的优化提供了便利。

实验技术与设备的发展为流体力学的研究带来了诸多便利。例如，通过高精度测量设备，研究者可以获取更准确的实验数据，从而验证理论模型和预测结果。同时，实验技术与计算流体力学的结合为流体力学的研究提供了更为广阔的应用空间。在未来，随着实验技术与设备的进一步发展，流体力学的研究将更加深入，为各种实际问题的解决提供更强大的支持。

## 5. 流体力学的发展历程与现状

流体力学作为物理学的一个分支，其发展历程悠久且富有成果。自古以来，人们就对流体的运动和性质产生了浓厚的兴趣。从公元前的阿基米德浮力定律，到17世纪伽利略的研究，流体力学的基础概念逐渐形成。进入18世纪，欧拉和伯努利等数学家对流体力学的发展作出了重要贡献，他们提出了诸如欧拉方程和伯努利定理等经典公式。

近现代以来，流体力学在理论研究和实验技术方面取得了突破性进展。例如，纳维-斯托克斯方程的提出，为流体力学的发展奠定了坚实基础。在实验技术方面，激光多普勒测速仪、颗粒图像测速等高精度测量设备的问世，极大推动了流体力学的研究。此外，计算流体力学的兴起，更是为流体力学的研究提供了强大的计算手段。

现状方面，流体力学在多个领域取得了显著成果。例如，在气象学领域，数值模拟和理论分析已经能够较为准确地预测天气变化和气候趋势；在海洋学领域，流体力学方法助力我们深入了解海洋环流、海底地形等现象。然而，流体力学的研究仍面临诸多挑战，如纳维-斯托克斯方程的求解、复杂现象的数学描述等。

## 6. 流体力学的未来研究方向与挑战

展望未来，流体力学的研究将聚焦于以下几个方向：首先，针对纳维-斯托克斯方程的求解问题，研究者将继续探索更高效、更精确的求解方法。其次，在复杂现象的研究方面，如高速湍流、多相流、生物流体力学等，流体力学将寻求揭示这些现象的物理机制和数学描述。此外，流体力学还将关注多尺度、多物理场的耦合问题，以期在更广泛的领域发挥重要作用。

在实验技术方面，流体力学将不断发展更高精度、更先进的设备。例如，高分辨率粒子图像测速技术、激光雷达测速技术等，将为流体力学的研究提供更丰富、更精确的数据。同时，计算流体力学将继续完善并发展更高效、更可靠的数值模拟方法。

面对未来挑战，流体力学需要跨学科、跨领域的合作。研究者应积极寻求与数学、计算机科学、生物学等其他学科的交融，以期在解决实际问题中发挥更大的作用。此外，培养高素质的研究人才、加强国际合作也将是流体力学发展的关键。总之，流体力学在未来将继续迎接挑战，为人类社会的科技进步和发展作出更大贡献。

v)-sum(c\_cov);