```
% gda01_03
%
% G for various cases
% Supports Section 1.3.3 and 1.3.4

clear all;

z = [1, 2, 4, 8, 9, 12, 15, 20]';
N=length(z);

% straight line case
M=2;
G=zeros(N,M);
G=[ones(N,1), z];
fprintf('Kernel for linear case\n');
```

```
Kernel for linear case
```

```
G
```

```
G =

     1     1
     1     2
     1     4
     1     8
     1     9
     1    12
     1    15
     1    20
```

```
% quadratic case
M=2;
G=zeros(N,M);
G=[ones(N,1), z, z.^2];
fprintf('Kernel for quadratic case\n');
```

```
Kernel for quadratic case
```

```
G
```

```
G =

     1     1     1
     1     2     4
     1     4    16
     1     8    64
     1     9    81
     1    12   144
     1    15   225
     1    20   400
```

```
% acoustic tomography case
N=8;
M=16;
G=zeros(N,M);
h=1;
```

```
for i = [1:4]
for j = [1:4]
    % measurements over rows
    k = (i-1)*4 + j;
    G(i,k)=h;
    % measurements over columns
    k = (j-1)*4 + i;
    G(i+4,k)=h;
end
end
fprintf('Kernel for tomography case\n');
```

```
Kernel for tomography case
```

```
G
```

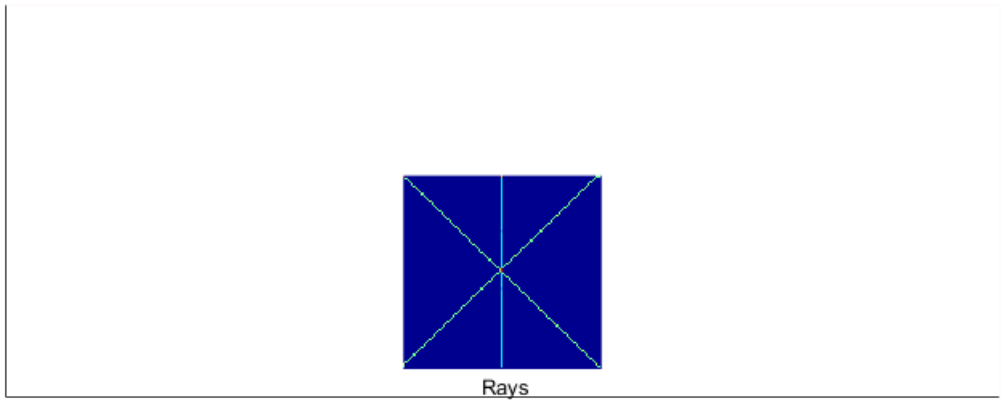
```
G =

     1     1     1     1     0     0     0     0     0     0     0     0     0
     0     0     0     0     1     1     1     1     0     0     0     0     0
     0     0     0     0     0     0     0     0     1     1     1     1     0
     0     0     0     0     0     0     0     0     0     0     0     0     1
     1     0     0     0     1     0     0     0     1     0     0     0     1
     0     1     0     0     0     1     0     0     0     1     0     0     0
     0     0     1     0     0     0     1     0     0     0     1     0     0
     0     0     0     1     0     0     0     1     0     0     0     1     0
```

```
% CAT scan case
% let the image be square, from (0 to 1) in both x and y
% let it be divided into M=KxL pixels which constitute model parameters
% the pixel array is unwrapped using k=i+(j-1)*K
% let N rays be collected with endpoits at (x1,y1) and (x2,y2).
K=100;
L=100;
M=K*L;
N=3;  % only three rays in this example
Dx=1/K; % x dimension of pixel
Dy=1/L; % y dimension of pizel
% endpoints of rays
X1 = [ [0,0]', [0,1]', [0,0.5]' ];
X2 = [ [1,1]', [1,0]', [1, 0.5]' ];
Ds = 0.001;
I = zeros(K,L); % image of rays, for plotting purposes
MAXNONZEROELEMENTS = 3*max(K,L);
G = spalloc( N, M, MAXNONZEROELEMENTS );
for n=[1:N]
    % step along ray
    x1=X1(:,n); x2=X2(:,n); % end points of ray
    R = sqrt((x2-x1)'*(x2-x1)); % length of ray
    t = (x2-x1)/R; % tangent to ray
    Nr = floor( R/Ds ); % number of ray segments
    x=x1; % starting point along ray
    for ir = [1:Nr]
        x = x1 + (ir-1)*Ds*t; % point along ray
        % indices of pixel containing the point
        px = floor( x(1) / Dx );
        if( px<1 ) % check that is withing bounds
            px=1;
        elseif( px>K )
            px=K;
```

```
        end
        py = floor( x(2) / Dy );
        if( py<1 ) % check that is withing bounds
            py=1;
        elseif( py>L )
            py=L;
        end
        % index in unwrapped vector of model parameters
        k=px+(py-1)*K;
        I(px,py) = I(px,py)+Ds; % make inage of rays
        G(n,k)=G(n,k)+Ds; % increment data kernel
    end
end

gda_draw(I,'caption Rays');
```




```
% Figure. Kernel for the CAT-scan case

fprintf('G-transpose (transposed so it fits on the screen)\n');
```

```
G-transpose (transposed so it fits on the screen)
```

```
full(G)'
```

```
ans =

    0.0290         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
```