

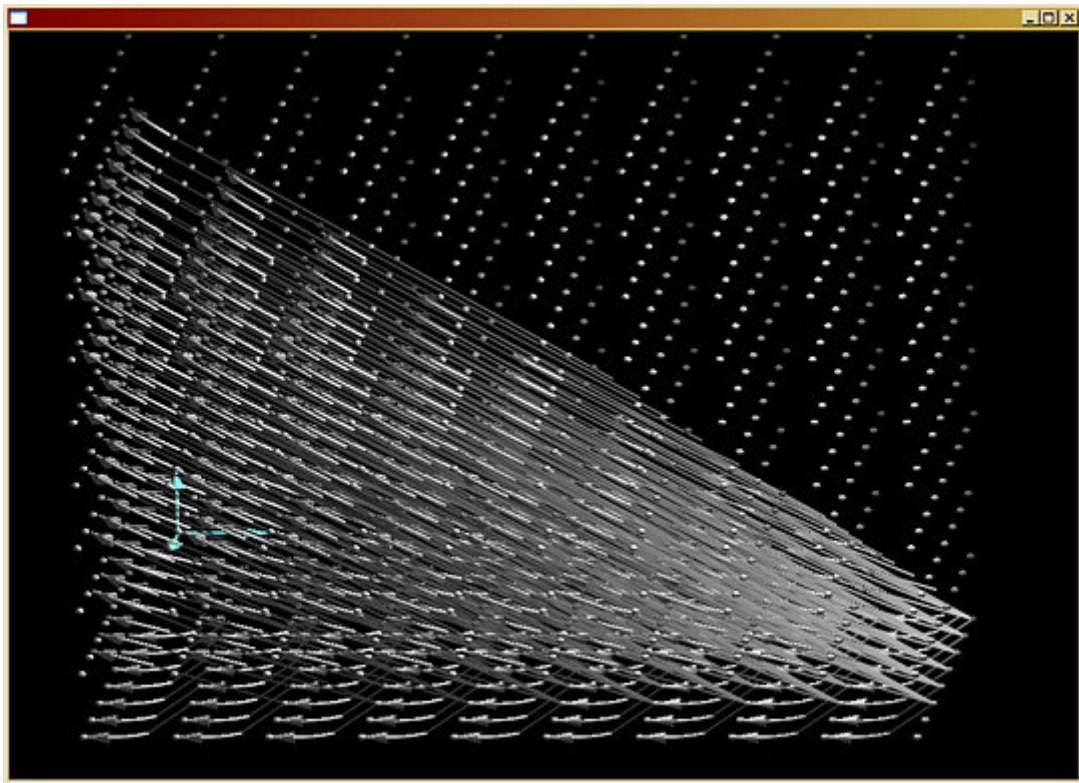
# Трёхмерная визуализация трёхмерного решетчатого графа.

Федоренко Полина (ранее Шатерникова)  
5 курс 8 группа

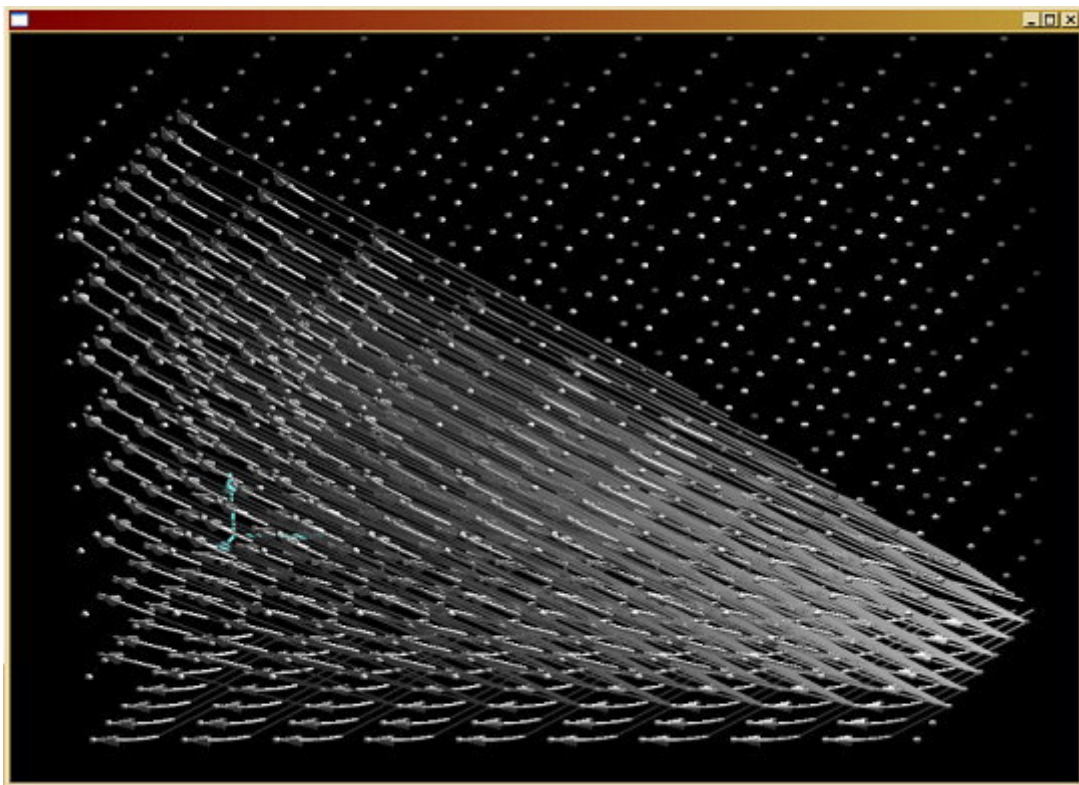
## Примеры.

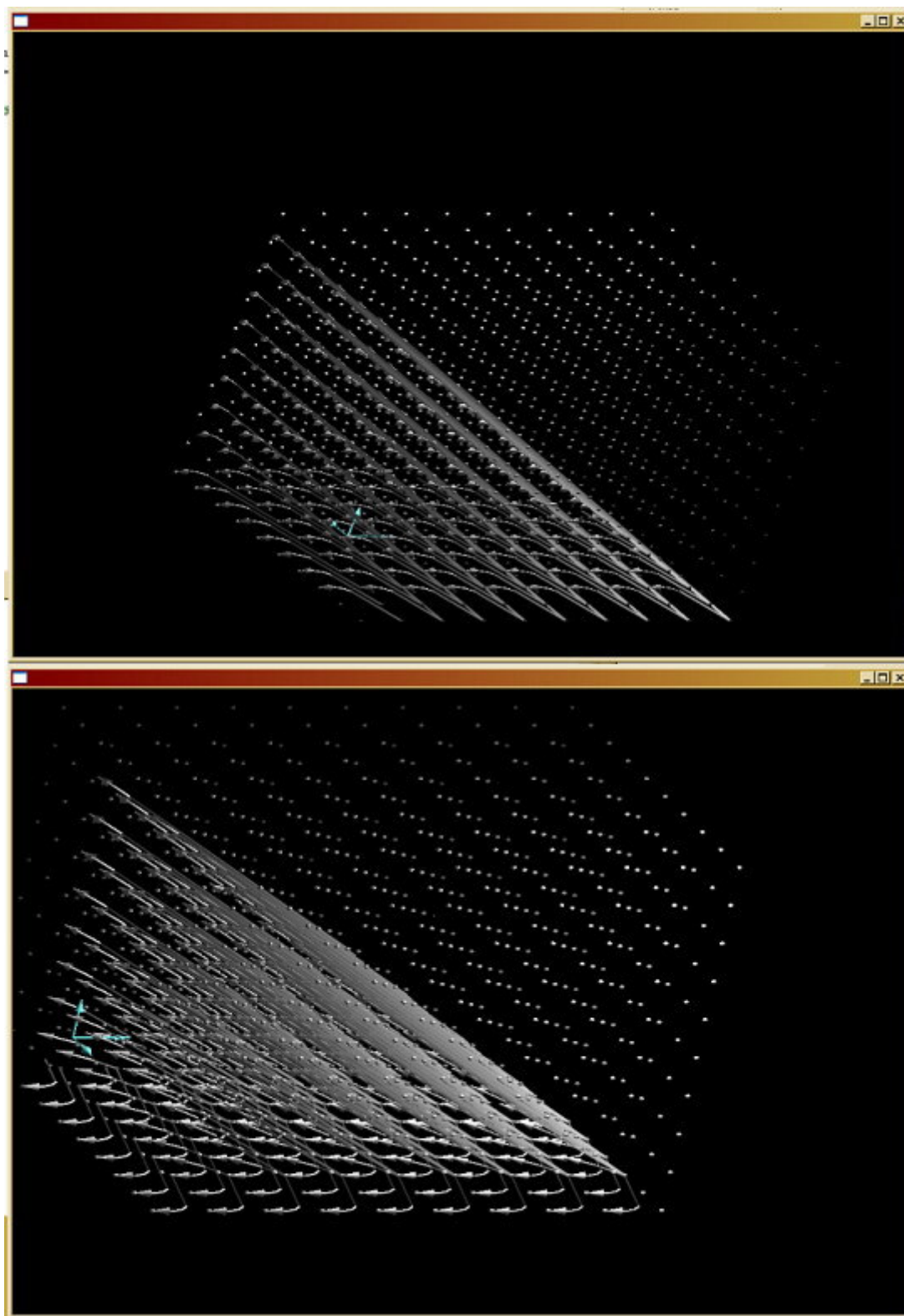
### Пример 1.

```
int b[50][50],a[50][50],x[50],k;  
int main()  
{  
  int i,j,k;  
  for(i=1;i<=10;i=i+1)  
  { for ( j=1;j<=10; j=j+1)  
    { for ( k=1;k<=10; k=k+1)  
      {  
        a[i+k][j]=a[i-1][j+k]+a[i][j-1]+a[i+1][j]+a[i][j+1];  
      }  
    }  
  }  
}
```

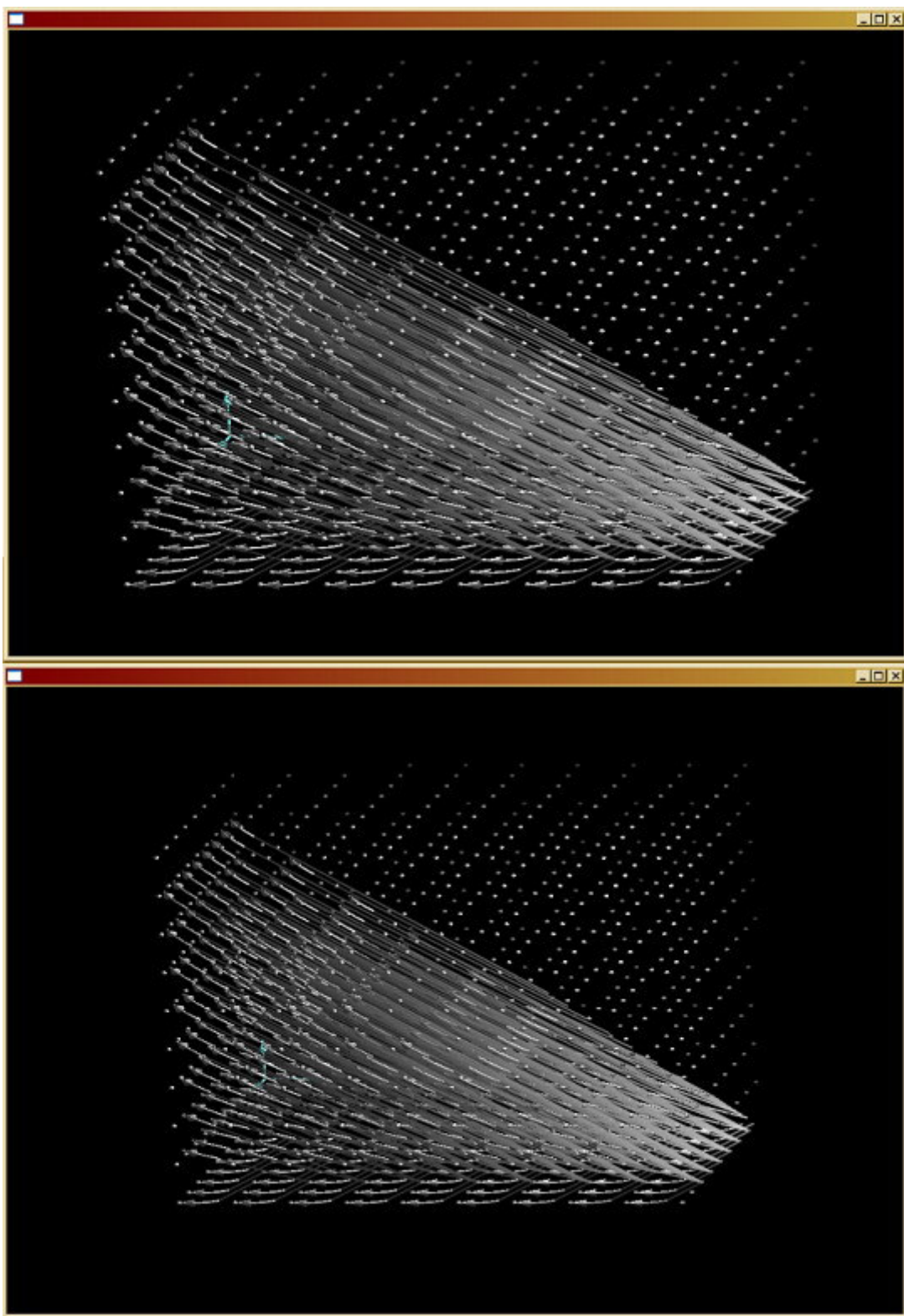


Полученный граф можно вращать при помощи мыши(с нажатием левой кнопки) и стрелок на клавиатуре в различном направлении:



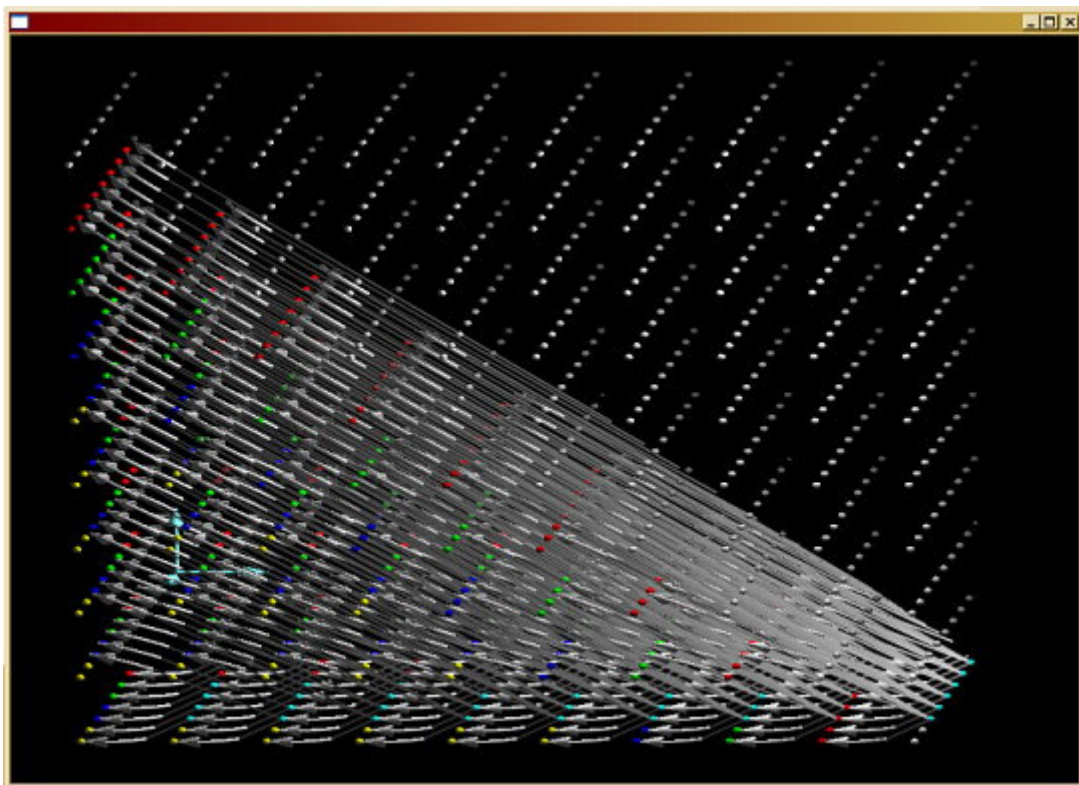


Кроме того, обеспечено удаление и приближение графа. Происходит это с нажатием правой кнопки мыши и её передвижением вверх-вниз. А также кнопками на клавиатуре: Page Up, Page Down.

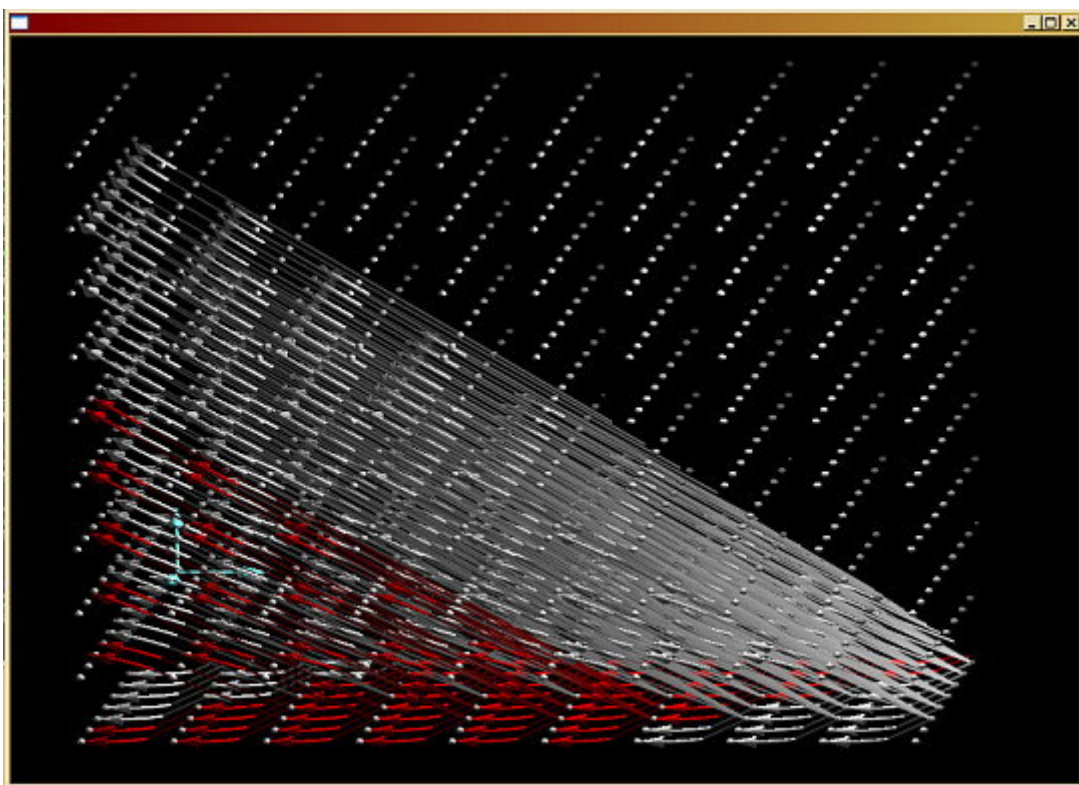


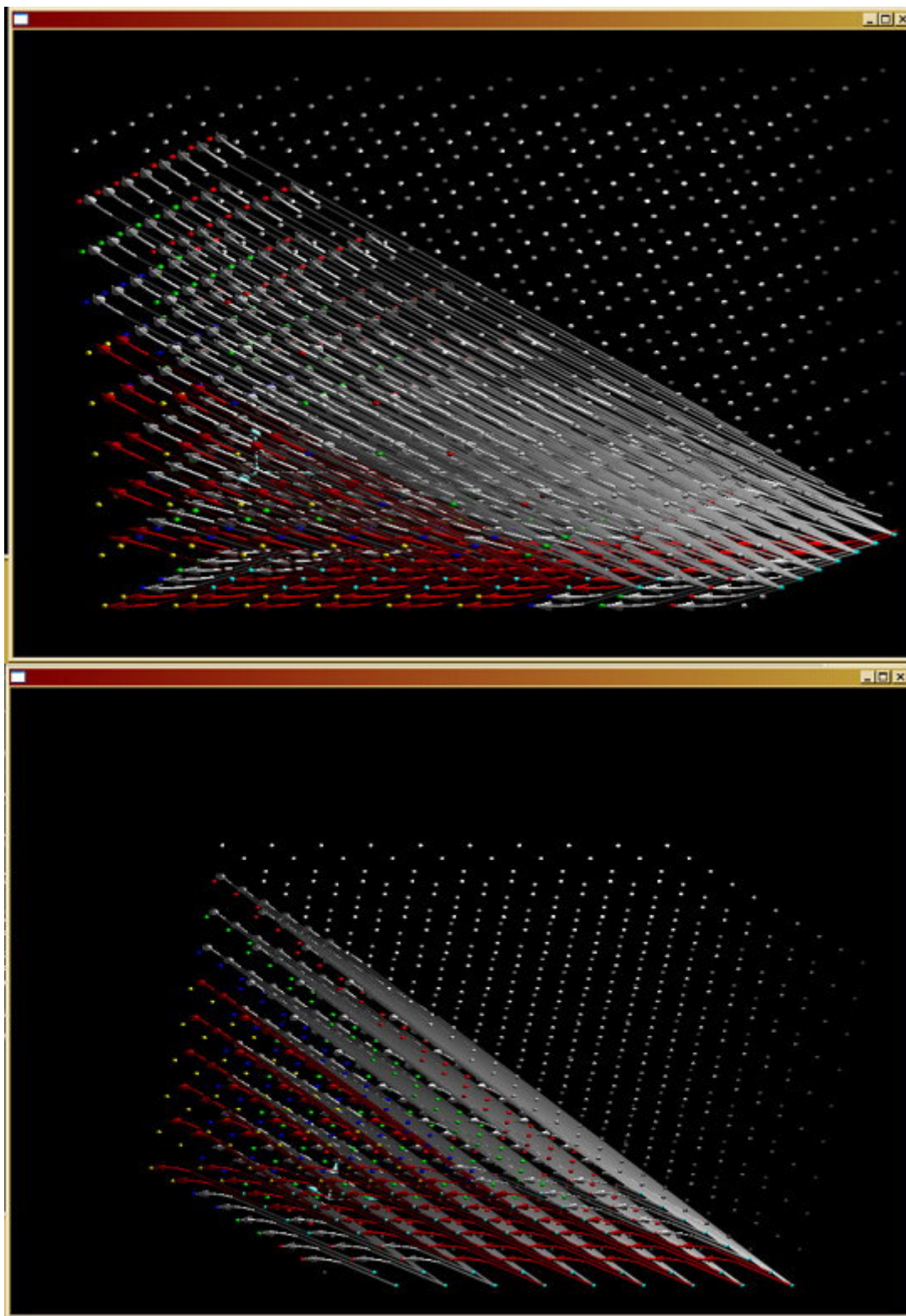
При нажатии кнопки Home вершины графа делятся на слои. А при нажатии кнопки End можно убрать изображение слоев графа, если они уже не нужны.





Кнопка Insert отвечает за построение критического пути на графе. Аналогично, кнопка Delete убирает изображение критического пути с экрана.

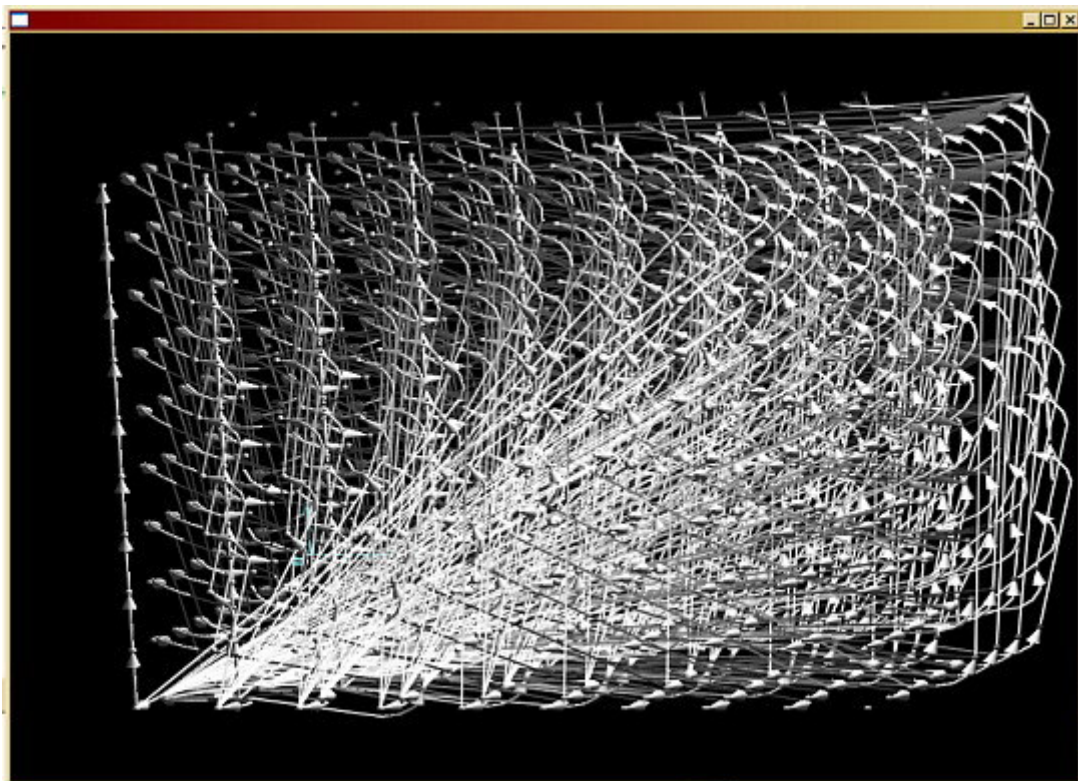




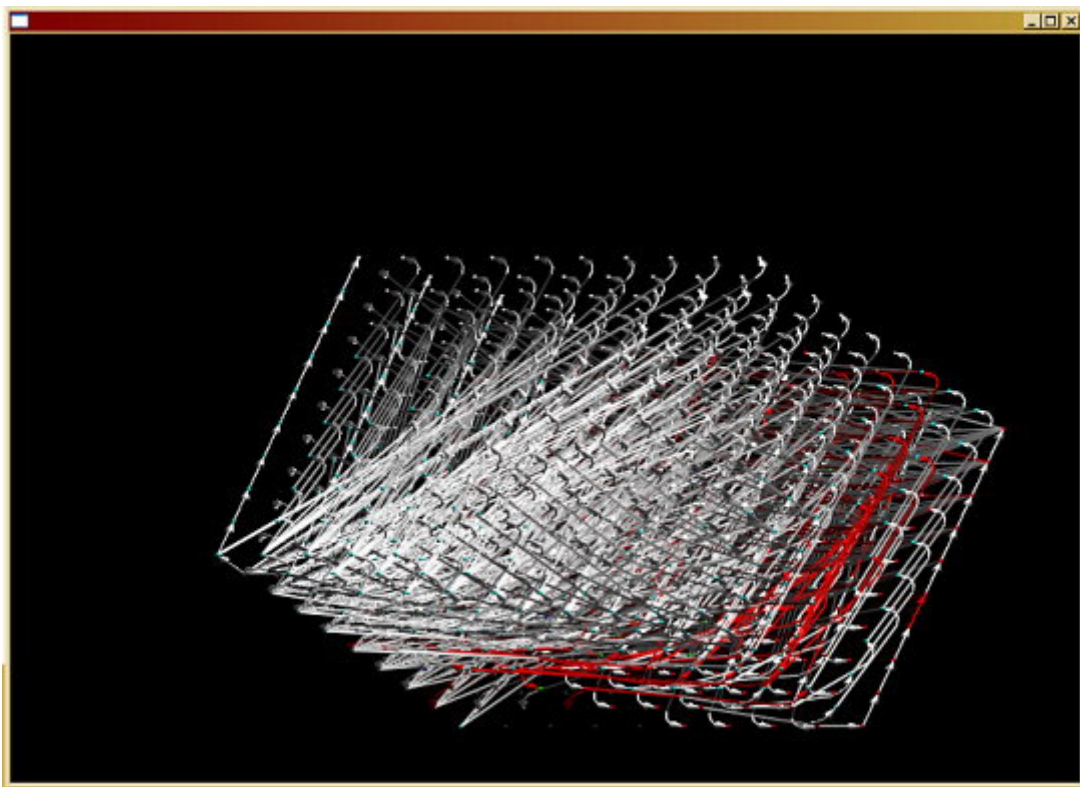
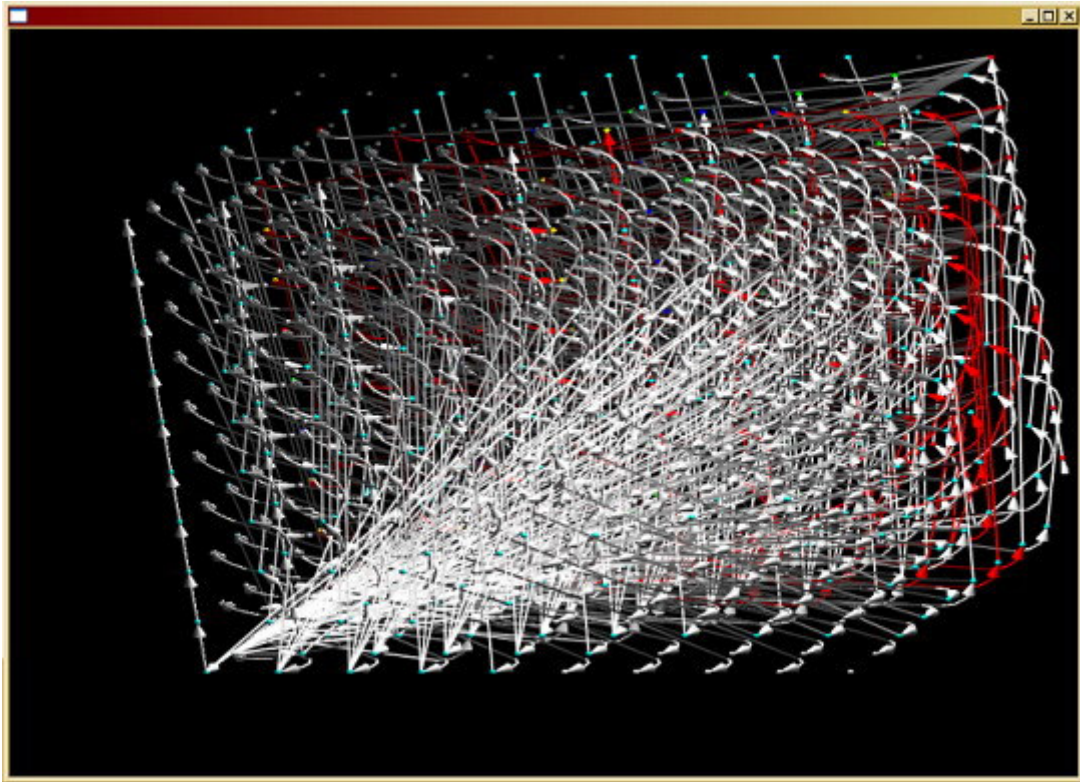
Очевидно, что слои графа и критические пути можно рассматривать одновременно, выбирая для этого различные ракурсы.

## Пример 2.

```
int u[21];  
int main()  
{  
    int i,j,k;  
    for(i=1;i<=10;i=i+1)  
    {  
        for (j=1;j<=10; j=j+1)  
        {  
            for (k=1;k<=10; k=k+1)  
            {  
                u[i+j+k]=u[21-i-j+k];  
            }  
        }  
    }  
}
```







### Пример 3.

```
int latticeGraphDemo;  
int b[50][50],a[50][50],x[50],k;  
int main()
```



```

{
  int i,j,k;
  for(i=1;i<=10;i=i+1)
  { for ( j=1;j<=10;j=j+1)
    { for ( k=1;k<=10;k=k+1)
      {
        a[i][j]=b[i][j];
        b[i+k][j]=a[2*i-7][2*j-7];
        x[i]=b[i-j][j+k];
      }
    }
  }
}

```

