CS601: Software Development for Scientific Computing

Maximum Points: 40 | End-semester examination | 27/11/2024, 2PM to 5PM

Instructions: This exam has two parts. Part I is open-book, open-notes (printed/written). Calculators are allowed and no other electronic devices are allowed. Part II is take-home. The submission instructions for part II are the same as in programming assignment 1. State your assumptions (if any) clearly.

Part I:

1. The figure below shows two design approaches to achieve the functionality implemented in the `main` function (note that the main function is *exactly* the same in the left half and not shown due to space constraints.). The approach shown in the left is implemented using *compositors*. While the approach shown in the right is the method that we have discussed in class and is the traditional approach. Discuss which approach is better or worse. Explain in detail why you think a particular approach is better or worse. Discussion must touch upon maintainability, readability, extendibility, and the performance aspects. **(10 points)**

```cpp
#include<iostream>
using namespace std;
class MyVec;
class Vadd{
        public: const MyVec *v1, *v2;
        Vadd(const MyVec& _v1, const MyVec& _v2){
                v1=&_v1;v2=&_v2;
        }
};
class MyVec{
public:
        double* data;
        int len;
        MyVec(int _len){data=new double[_len];len=_len;}
        MyVec(const Vadd& _vadd){
                addassign(*(_vadd.v1), *(_vadd.v2), *this);
        }
        void initialize(){for(int i=0;i<len;i++) data[i]=0.;}
        friend MyVec& addassign(const MyVec& lhs, const MyVec& rhs, MyVec& out);
        MyVec& operator=(const Vadd& rhs){
                return addassign(*(rhs.v1), *(rhs.v2), *this);
        }
        ~MyVec(){ delete [] data;len=0;}
};
inline Vadd operator+(const MyVec& lhs, const MyVec& rhs){
                return Vadd(lhs, rhs);
}
MyVec& addassign(const MyVec& lhs, const MyVec& rhs, MyVec& out){
        for(int i=0;i<lhs.len;i++)
                out.data[i]=lhs.data[i]+rhs.data[i];
        return out;
}
```

```cpp
#include<iostream>
using namespace std;
class MyVec{
public:
        double* data;
        int len;
        MyVec(int _len){data=new double[_len];len=_len;}
        void initialize(){for(int i=0;i<len;i++) data[i]=0.;}
        MyVec& operator=(const MyVec& rhs){
                for(int i=0;i<rhs.len;i++)
                        this->data[i]=rhs.data[i];
                return *this;
        }
        ~MyVec(){ delete [] data;len=0;}
};

inline MyVec operator+(const MyVec& lhs, const MyVec& rhs){
                MyVec out(lhs.len);
                for(int i=0;i<lhs.len;i++)
                        out.data[i]=lhs.data[i]+rhs.data[i];
                return out;
}

int main(){
        MyVec u(10000000), v(10000000), w(10000000);
        u.initialize(); v.initialize();
        w=u+v;
}
```

2. Consider the boundary value problem:

$$u_{xx} + u_{yy} = -2 \text{ in the square } 0 < x < 1, 0 < y < 1$$
$$u = xy \text{ on the boundary}$$

Use discretization step size, h = 1/3. Using the five-point stencil approach,

a) Show the grid and the known values at grid points. Formulate the system of equations showing unknown values of grid points. **(3 points)**

b) Solve the system of equations using Jacobi iteration. Show only one iteration. **(2 points)**

c) Solve the system of equations using Gauss-seidel iteration. Show only one iteration. **(2 points)**

3. Consider multiplying Fourier matrix F_4 with a vector $c = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$. The first step is to separate the even and odd numbered components of c to $\begin{matrix} c_0 \\ c_2 \\ c_1 \\ c_3 \end{matrix}$. The second step is to multiply the upper half (c') and lower half (c'') of the separated c vector by F_2 to yield $\begin{bmatrix} F_2 c' \\ F_2 c'' \end{bmatrix}$. The last step is to gather the upper half (y') and lower half(y'') of the vector obtained from the matrix-vector multiplication of the previous step and construct y=$F_4 c$ as: $y_j = y'_j + w_n^j y''_j$ and $y_{j+m} = y'_j - w_n^j y''_j$ for j=0, …, m-1.

   a) Draw the data flow diagram showing the data dependences needed to compute y and backwards all the way to the input c. **(4 points)**
   b) The three steps described correspond to transformation (multiplying) with matrices A, B, and C shown below. Mention which step corresponds to multiplying with which matrix. Show the matrix multiplication A*B*C and compare the product matrix with with F_4. Which six entries of the product matrix require you to know that $i^2 = -1$ so that the product matrix and F_4 are the same? **(5 points)**

A= $\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & i \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -i \end{bmatrix}$ B= $\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$ C= $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

4. consider the following set of points in 2D space: (5.5, 6.5), (1.5, 3.5), (0.5, 2.5), (0.5, 0.5), (2.5, 0.5), (3.5, 0.5),(6.5, 0.5), (6.5, 1.5), and (4.5, 3.5). A point (x, y) denotes x units along X-axis and y units along Y-axis.
   a) Draw an adaptive quad-tree representing the space of points. Assume that the leaf nodes can have at most one point. **(2.25 points)**
   b) Show a numbering of the leaf nodes as per z-order numbering. **(2.25 points)**
   c) Retrieve all points in the range (4,4) to (6,1). If you did this retrieval naïvely for N-points, each in K-Dimensional space, what would be the time complexity? **(1 point)**
   d) Using the quad-tree constructed, propose a scheme to retrieve the set of points (for the range mentioned previously) efficiently. You must write a pseudocode or code snippet. You may assume that a member method bool DoesIntersect(QuadTreeNode* node, Box* b) of class QuadTreeNode is given. Class Box represents the bounding box (a rectangle in this space). Class Point represents a point. **(2.5 points)**

Part II – take home: (6 points) Visit the discussion forum to receive the question paper and instructions.