# 实 验 报 告

课程名称： 惯性导航基础

学 院： 测绘与空间信息学院

专 业：测绘工程(海洋测绘方向)班 级： 2018

姓 名： 黄瑾 学 号： 201801020510

2020年 1 月 10 日

山 东 科 技 大 学 教 务 处 制

**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组 别 | 1 | 姓 名 | 黄瑾 | 同组实验者 | 潘宗龙、孟怡君 |
| 实验项目名称 | 编程实现欧拉角、方向余弦阵、四元数以及等效旋转矢量之间的相互转换 | | | 实 验 日 期 | 2020年11月11日 |
| 教 师 评 语 |  | | | | |
| 实验成绩： | | | 指导教师（签名）：  年 月 日 | | |
| **实验目的：**  利用编程语言实现欧拉角、方向余弦阵、四元数以及等效旋转矢量之间的相互转换。  **实验内容：**  1. 实验环境  基于Visual Studio 2019 版本进行CPP的开发，基于MATLAB 2020a 进行MATLAB 程序的开发，基于PyCharm Professional 2020.3 进行Python 版本程序的开发。  2. 实现内容  通过编程实现四种姿态转换表示方法的计算机实现，编写成为类函数，便于后续编程。  **实验步骤：**  1. 实验原理：  根据课本中四种表示方法之间的转化关系，通过编程实现其转换。程序主要包含attitude\_matrix.h和attitude\_matrix.cpp文件，欧拉角、方向余弦阵、四元数以及等效旋转矢量通过编写类实现，其转换方法通过编写类成员函数实现表示方法相互之间的转化。  2. 算法原理  **1．从欧拉角到姿态阵**  在“东-北-天312”欧拉角定义下，参考式（B-2），可得从地理坐标系（选为导航系，系）到载体坐标系（系）的方向余弦矩阵  （B-3）  式中，表示矩阵的第行列元素，式（B-3）便是根据欧拉角（姿态角）计算方向余弦阵（姿态阵）的公式。  **2．从姿态阵到欧拉角**  如果已知姿态阵，通过观察式（B-3），可得提取姿态角的数值方法如下所述。  （1）当时，有  （B-4）  其中，数值为用户根据具体需求而设定的略小于1的数值；为标准Ｃ语言函数库中的求反正切函数，包含象限判断功能，但两个输入参数和不得同时为零，以为例，它在的第三行向量为单位向量且时是可以保证和不同时为零的。  （2）当时，有，作近似和，则可近似为    由上式可求得  （B-5）  （3）当时，有，作近似和，则可近似为    由上式可求得  （B-6）  式（B-5）和式（B-6）显示，当俯仰角在附近时，横滚角和航向角之间是无法单独分离的，或者说两者都存在多值性，只有在指定其中某一个值之后才能够确定另外一个，比如一般可令。  综合前面（1）~（3）分析，得由姿态阵求解欧拉角的完整算法如下：  （B-7）  **3．从四元数到姿态阵**  参考式（2.4-23），将姿态阵与四元数之间转换关系重写如下：  （B-8）  **4．从姿态阵到四元数**  根据式（B-8）的对角线元素，可得  （B-9）  再由式（B-8）的非对角线元素，可得  （B-10）  若仅根据式（B-9）将难以确定四元数各元素的正负符号。如果已知四元数的某一个元素，则根据式（B-10）可求解其他元素，但须避免该已知元素为0。由四元数归一化条件可知，必然有成立，也就是说，四个元素中必然存在某个。实际应用时，可先根据式（B-9）计算获得某一个较大的元素（不妨取为正值），再根据式（B-10）计算剩余的其他三个元素。  在式（B-9）中，等价于，即；同理，有等价于；以及等价于。由此可得计算四元数各元素的伪代码如下：  （B-11）  **5．从欧拉角到四元数**  在实际惯导的姿态更新算法中经常使用的是四元数，需要涉及四元数和欧拉角的转换问题。根据单位四元数的含义式（2.4-21），在“东-北-天312”欧拉角定义下，由欧拉角求解四元数的公式为  （B-12）  **6．从四元数到欧拉角**  仅根据式（B-12），由四元数直接求解欧拉角并不容易。实际上，可通过姿态阵作为中间过渡量，先由四元数计算姿态阵，再由姿态阵计算欧拉角，分别如式（B-8）和式（B-7），综合一起后其结果为  （B-13）  最后，总结给出欧拉角、方向余弦阵和四元数三种姿态描述之间的相互转换关系，如图B-4所示。    图B-4 三种姿态描述之间的转换关系  后面简要概述等效旋转矢量与四元数之间的关系，在程序实现过程中，即可通过四元数作为中间媒介实现等效旋转矢量与其余三者之间的相互转换。  3. 程序代码  欧拉角类的类架构如下所示：    方形余弦阵类架构的如下所示：    四元数类的架构如下所示：    等效旋转矢量类的架构如下所示：    4. 实验结果  本程序通过首先规定初始的欧拉角，通过将这一欧拉角与其余三个表示方法相互转换，通过表示方法之间的连续转换，最终转换为初始输入结果，验证了转换方法的正确性。    **实验心得：**  在实验过程中，数据的存储采用了Eigen库的一些相关函数，在应用过程中，起初采用了自己编写的矩阵库，但是相较于成熟的开源矩阵库，自主编写的库仍有运行慢、不稳定等问题，还需要加强实践和学习。 | | | | | |

**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组 别 | 1 | 姓 名 | 黄瑾 | 同组实验者 | 潘宗龙、孟怡君 |
| 实验项目名称 | 编程实现大地坐标与地心直角坐标相互转换 | | | 实 验 日 期 | 2020年11月18日 |
| 教 师 评 语 |  | | | | |
| 实验成绩： | | | 指导教师（签名）：  年 月 日 | | |
| **实验目的：**  通过编程实现大地坐标与地心直角坐标相互转换。  **实验内容：**  1. 实验环境  基于Visual Studio 2019 版本进行CPP的开发  2. 实现内容  通过编程实现大地坐标与地心直角坐标相互转换。  **实验步骤：**  1. 实验原理  下面给出地表附近同一地点的地理坐标与地心直角坐标之间的相互转换关系。  **1．由求解**  根据式（3.1-15）和图3.1-2，不难求得  （3.1-35）  其中，卯酉圈曲率半径的计算见式（3.1-14）或重写如下：  （3.1-36）  **2．由求解**  首先，由式（3.1-35）中第二式除以第一式，得  （3.1-37）  当和不同时为0时（非极点处），由式（3.1-37）可直接解得经度  （3.1-38）  式中，为计算给定横、纵坐标点的反正切值函数，取值范围。  其次，对于纬度，不能求得其显式表示，通常采用迭代算法，推导过程如下。  由式（3.1-35）中第一式和第二式实施如下运算：  （3.1-39）  由式（3.1-35）中第三式移项整理，可得  （3.1-40）  当时，也即在非极点处，由式（3.1-40）除以式（3.1-39），得  （3.1-41）  将式（3.1-36）改写成，再代入式（3.1-41），得  （3.1-42）  如记，则由式（3.1-42）可构造出求解纬度正切值的迭代公式如下：  （3.1-43）  令迭代初值，一般经过5～6次迭代便可达到足够的数值计算精度，再由求反正切即可获得纬度。  最后，根据式（3.1-39）求解高度，得  （3.1-44）  式（3.1-38）、式（3.1-43）和式（3.1-44）即为由求解的算法。  2. 程序代码  坐标转化的相应函数如下所示：    3. 实验结果  实验验证代码如下，验证结果如下图所示：  Vector3d BLH(deg2rad(20), deg2rad(30), 10000);  cout << BLH << endl;  Vector3d xyz = gcs2gc(BLH);  cout << xyz << endl;  BLH = gc2gcs(xyz);  cout << BLH << endl;    **实验心得：**  课本中的相关转换方法，尤其是地心直角坐标与大地坐标相互转换的过程中，需要进行迭代计算，本次实验中判断跳出迭代的指标是精度指标，即相邻两次迭代间的差小于1e-20即跳出迭代。 | | | | | |

**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组 别 | 1 | 姓 名 | 黄瑾 | 同组实验者 | 潘宗龙、孟怡君 |
| 实验项目名称 | 编程实现IMR格式惯导数据的读取与解析 | | | 实 验 日 期 | 2020年11月25日 |
| 教 师 评 语 |  | | | | |
| 实验成绩： | | | 指导教师（签名）：  年 月 日 | | |
| **实验目的：**  使用CPP实现IMR格式惯导数据的读取与解析。。  **实验内容：**  1. 实验环境  基于Visual Studio 2019 版本进行CPP的开发  2. 实现内容  IMR格式惯导数据的读取与解析。  **实验步骤：**  1. 实验原理  IMR 文件主要由两个部分构成，其一是文件头，另一个是文件数据部分，本实验中主要通过建立结构体来实现对文件的读取。具体思路如下：首先通过结构体中各个变量的长度读取二进制数据，将数据保存于结构体中，读取完数据头后，循环读取文件的数据部分，通过seekg函数跳过文件头部分，利用eof判断是否达到文件的结尾。  IMR格式的数据读取即通过给定的IMR格式编写读写程序。其数据结构如下表所示：   | **IMR Header Struct Definition** | | | | | --- | --- | --- | --- | | **Word** | **Size (bytes)** | **Type** | **Description** | | szHeader | 8 | char[8] | “$IMURAW\0” – NULL terminated ASCII string | | bIsIntelOrMotorola | 1 | int8\_t | 0 = Intel (Little Endian), default  1 = Motorola (Big Endian) | | dVersionNumber | 8 | double | Inertial Explorer program version number (e.g. 8.80) | | bDeltaTheta | 4 | int32\_t | 0 = Data to follow will be read as scaled angular rates  1 = (default), data to follow will be read as delta thetas, meaning angular increments (i.e. scale and multiply by *dDataRateHz* to get degrees/second) | | bDeltaVelocity | 4 | int32\_t | 0 = Data to follow will be read as scaled accelerations  1 = (default), data to follow will be read as delta velocities, meaning velocity increments (i.e. scale and multiply by *dDataRateHz* to get m/s2) | | dDataRateHz | 8 | double | The data rate of the IMU in Hz. e.g. 0.01 second data rate is 100 Hz | | dGyroScaleFactor | 8 | double | If *bDeltaTheta* == 0, multiply the gyro measurements by this to get degrees/second  If *bDeltaTheta* == 1, multiply the gyro measurements by this to get degrees, then multiply by *dDataRateHz* to get degrees/second | | dAccelScaleFactor | 8 | double | If *bDeltaVelocity* == 0, multiply the accel measurements by this to get m/s2  If *bDeltaVelocity* == 1, multiply the accel measurements by this to get m/s, then multiply by *dDataRateHz* to get m/s2 | | iUtcOrGpsTime | 4 | int32\_t | Defines the time tags as GPS or UTC seconds of the week  0 = Unknown, will default to GPS  1 = Time tags are UTC seconds of week  2 = Time tags are GPS seconds of week | | iRcvTimeOrCorrTime | 4 | int32\_t | Defines whether the time tags are on the nominal top of the second or are corrected for receiver time bias  0 = Unknown, will default to corrected time  1 = Time tags are top of the second  2 = Time tags are corrected for receiver clock bias | | dTimeTagBias | 8 | double | If you have a known bias between your GPS and IMU time tags enter it here | | szImuName | 32 | char[32] | Name of the IMU being used | | reserved1 | 4 | uint8\_t[4] | Reserved for future use | | szProgramName | 32 | char[32] | Name of calling program | | tCreate | 12 | time\_type | Creation time of file | | bLeverArmValid | 1 | bool | True if lever arms from IMU to primary GNSS antenna are stored in this header | | lXoffset | 4 | int32\_t | X value of the lever arm, in millimeters | | lYoffset | 4 | int32\_t | Y value of the lever arm, in millimeters | | lZoffset | 4 | int32\_t | Z value of the lever arm, in millimeters | | Reserved[354] | 354 | int8\_t[354] | Reserved for future use |     The single header, which is a total of 512 bytes long, is followed by a structure of the following type for each IMU measurement epoch:   | **IMR Record Struct Definition** | | | | | --- | --- | --- | --- | | **Word** | **Size** | **Type** | **Description** | | Time | 8 | double | Time of the current measurement | | gx | 4 | int32\_t | Scaled gyro measurement about the IMU X-axis | | gy | 4 | int32\_t | Scaled gyro measurement about the IMU Y-axis |   2. 实验代码  IMR 文件头的组织架构    IMR 数据结构：    IMR数据读取结构    3. 实验结果  读取的数据内容如下图所示：    **实验心得：**  IMR数据的读取过程中，要注意数据的改正以及高低位问题，需要根据头文件中的信息来判断数据文件的读取过程中是否需要注意高低位数据存取的问题。 | | | | | |

**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组 别 | 1 | 姓 名 | 黄瑾 | 同组实验者 | 潘宗龙、孟怡君 |
| 实验项目名称 | 编程实现地理坐标系下的姿态更新 | | | 实 验 日 期 | 2020年12月02日 |
| 教 师 评 语 |  | | | | |
| 实验成绩： | | | 指导教师（签名）：  年 月 日 | | |
| **实验目的：**  编程利用CPP语言实现地理坐标系下的姿态更新  **实验内容：**  1. 实验环境  基于Visual Studio 2019 版本进行CPP的开发  2. 实现内容  编程利用CPP语言实现地理坐标系下的姿态更新  **实验步骤：**  1. 实验原理  选取“东–北–天（E–N–U）”地理坐标系作为捷联惯导系统的导航参考坐标系，后面记为系，则以系作为参考系的姿态微分方程为  （4.1-1）  其中，矩阵表示载体系（系）相对于导航坐标系（系）的姿态阵，由于陀螺输出的是系相对于惯性系（系）的角速度，而角速度信息不能直接测量获得，需对微分方程式（4.1-1）作如下变换：  （4.1-2）  其中，表示系相对于系的旋转，它包含两部分：地球自转引起的导航系旋转，以及惯导系统在地球表面附近移动因地球表面弯曲而引起的系旋转，即有，其中  （4.1-3）  （4.1-4）  式中，为地球自转角速率；和分别为地理纬度和高度。式（4.1-4）来源及含义可参见3.1节式（3.1-34）。  与矩阵微分方程相比，虽然式（4.1-2）也是线性时变的，但它的离散化求解更加麻烦，一般不会直接求解该方程，而是采用如下方法解决姿态阵更新问题。  根据矩阵链乘规则，有  （4.1-5）  式中，角标括号中的符号表示时刻。由于系是绝对不动的惯性参考坐标系，它与时间无关，不需标注时刻；而系和系相对于系都是动坐标系，均跟时间有关，需标注时刻。  根据姿态阵微分方程和，分别可得相对于惯性系的更新算法  （4.1-6）  （4.1-7）  其中，矩阵表示以系作为参考基准，系从时刻到时刻的旋转变化，可由陀螺角速度确定；表示以系作为参考基准，系从时刻到时刻的旋转变化，可由计算角速度确定。  将式（4.1-6）和式（4.1-7）代入式（4.1-5），得  （4.1-8）  其中，和分别表示和时刻的捷联姿态矩阵。若陀螺在时间段内（）进行了两次等间隔采样，角增量分别为和，采用二子样圆锥误差补偿算法，有  （4.1-9）  （4.1-10）  通常在导航更新周期内，可以认为由速度和位置引起的变化很小，即可视为常值，记为，则有  （4.1-11）  式（4.1-8）～式（4.1-11）即为捷联惯导数值递推姿态更新算法。  2. 实验代码      3. 实验结果    **实验心得：**  姿态更新过程中，一个比较重要的关键点就是注意三角函数的计算，将角度与弧度统一在程序中实现。 | | | | | |

**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组 别 | 1 | 姓 名 | 黄瑾 | 同组实验者 | 潘宗龙、孟怡君 |
| 实验项目名称 | 编程实现地理坐标系下的速度和位置更新 | | | 实 验 日 期 | 2020年12月08日 |
| 教 师 评 语 |  | | | | |
| 实验成绩： | | | 指导教师（签名）：  年 月 日 | | |
| **实验目的：**  编程实现地理坐标系下的速度和位置更新  **实验内容：**1. 实验环境  基于Qt Creator5.14.2版本进行CPP的开发  2. 实现内容  编程实现地理坐标系下的速度和位置更新  **实验步骤：**  **1．地理坐标系下的速度更新**  将式（4.1-23）移项，可改写成递推形式  （4.1-26）  下面主要讨论和的数值积分算法。  即使对于诸如飞机等快速运动的运载体，在短时间内其引起的导航坐标系旋转和重力矢量变化都是很小的，因而一般认为的被积函数是时间的缓慢量，可采用区间中点时刻进行近似计算，将式（4.1-25）近似为  （4.1-27）  由于此时尚不知时刻的导航速度和位置等参数，因而式（4.1-27）中时刻的各量需使用外推法计算，表示如下：  （4.1-28）  式中，各参数在和时刻均是已知的。可见，的计算过程比较简单。  在时间段内，对式（4.1-34）两边同时积分，得  （4.1-35）  其中，记  （4.1-36）  （4.1-37）  （4.1-38）  再将式（4.1-43）代入式（4.1-41），便得二子样速度划桨误差补偿算法  （4.1-44）  式（4.1-48）简化为  （4.1-50）  至此，形成了一套完整的捷联惯导速度更新算法，主要包括式（4.1-26）、式（4.1-50）、式（4.1-27）、式（4.1-36）和式（4.1-44）。  **2．地理坐标系下的位置更新**  捷联惯导系统的位置（纬度、经度和高度）微分方程式（3.1-29）～式（3.1-31），重写如下：  （4.1-57）  将它们改写成矩阵形式，为  （4.1-58）  其中，记  ，  ，  ， ，  与捷联惯导姿态和速度更新算法相比，位置更新算法引起的计算误差一般比较小，可采用比较简单的梯形积分法对式（4.1-58）离散化，得  （4.1-59）  式（4.1-59）移项，便得位置更新算法  （4.1-60）  其中，可采用如式（4.1-28）所示的线性外推算法，对矩阵整体进行外推；亦可对矩阵元素中的位置变量，外推，再构造矩阵。  2. 实验代码            3. 实验结果    **实验心得：**  在使用Qt进行编写的过程中，曾将Visual Studio 的代码进行了迁移，但是由于编译器不同，MinGW的编译更为严格，出现了重复定义的问题，在头文件中新增了#program once仍无法解决问题，后发现是在头文件中进行定义导致。 | | | | | |

**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组 别 | 1 | 姓 名 | 黄瑾 | 同组实验者 | 潘宗龙、孟怡君 |
| 实验项目名称 | 编程实现解析粗对准 | | | 实 验 日 期 | 2020年12月15日 |
| 教 师 评 语 |  | | | | |
| 实验成绩： | | | 指导教师（签名）：  年 月 日 | | |
| **实验目的：**  编程实现解析粗对准  **实验内容：**  1. 实验环境  基于Visual Studio 2019 版本进行CPP的开发  2. 实现内容  编程实现解析粗对准  **实验步骤：**  1. 实验原理  初始对准一般是在运载体对地静止的环境下进行的，即运载体相对地面既没有明显的线运动也没有角运动，且对准地点处的地理位置准确已知，也就是说，重力矢量和地球自转角速度矢量在地理坐标系（初始对准参考坐标系）的分量准确已知，分别如下：  ，  （7.1-17）  其中，，和分别表示当地纬度、重力加速度大小和地球自转角速率大小，且记地球自转角速度的北向分量和天向分量。  根据4.1节的分析，有如下惯导角速度测量关系和比力方程：  （7.1-18a）  （7.1-18b）  在静基座下线运动引起的和都非常小，可以忽略，再考虑到陀螺仪测量误差和加速度计测量误差，  可简写为  （7.1-20a）  （7.1-20b）  其中，为等效陀螺仪测量误差；为等效加速度计测量误差。当测量误差远小于有用信号时，比如并且时，式（7.1-20）近似估计为  （7.1-21a）  （7.1-21b）  一般情况下线运动干扰相对误差小于角运动，所以常常选择作为主参考矢量，根据式（7.1-7）可得姿态阵估计  （7.1-22）  将式（7.1-17）代入式（7.1-22），得  （7.1-23）  由式（7.1-23）可见，根据实际陀螺仪和加速度计测量值即可直接实现姿态阵估计，这一过程表面上与地理位置无关，其实地理纬度信息隐含在两矢量与的夹角之中，即应当有。实际应用中，为了降低传感器高频噪声及高频环境晃动的影响，主要是针对陀螺仪高频噪声和高频角晃动的影响，常常需要采集一段时间的惯性传感器数据，假设角增量为和速度增量为，求解该时间段内的平均角速度为以及比力为，分别代替式（7.1-23）中的和，从而可以估计得。一般情况下，在时间段内当低频晃动角小于且速度变化小于时，就能够求得具有一定近似精度的粗略对准结果。  2. 实验代码    3. 实验结果    **实验心得：**  实现通过双矢量定姿实现解析粗对准，惯导系统初始对准就是确定参考导航坐标系的一个过程。惯导系统刚上电启动时，其载体坐标系相对于参考导航坐标系的各轴指向完全未知或不够精确，无法立即进入导航状态，因此必须先确定载体坐标系相对于导航坐标系的空间方位，从惯导的角度看这等效于寻找参考导航坐标系的一个过程。 | | | | | |