1 考虑第一类积分方程

$$\int_0^1 e^{ts}x(s)ds = y(t) = \frac{e^{t+1}-1}{t+1}, 0 \le t \le 1$$

的解，该方程的唯一精确解为e^t。对该问题求数值解时，取步长$h = 1/n$，用复合梯形积分公式

$$\int_0^1 e^{ts}x(s)ds = h(\frac{1}{2}x(0) + \frac{1}{2}x(1) + \sum_{j=1}^{n-1} e^{jht}x(jh))$$

近似左端积分。对不同的区间等分数n，最后由线性代数方程组

$$h(\frac{1}{2}x(0) + \frac{1}{2}x(1) + \sum_{j=1}^{n-1} e^{jh*ih}x(jh)) = y(ih), i = 0, 1, ..., n$$

求$x(ih)$的值。由于离散后得到的线性方程组$Kx = y$不适定（随n增大，K求逆不稳定），人们通常采取正则化的方法来求解，即求解方程:

$$(\alpha I + K^T * K)x_\alpha = K^T y$$

将其解x_α作为x的近似，其中α为一给定的小参数。根据我们所学稠密矩阵的算法，并行求解上述方程，最后画出计算结果。其实现要点如下:

- 并行计算$\alpha I + K^T K$，其中有矩阵转置，矩阵相乘，矩阵相加，
- 并行计算$K^T y$，稠密矩阵乘向量，
- 并行求解线性方程组，由于是稠密矩阵，故可采用并行LU分解法求解（可选）
- 将前两步产生的矩阵和向量输出为Matlab文件，然后在Matlab中串行求解线性方程组（可选）。

最后求解线性方程组可以在后两步中根据个人情况选择一种方法。 参数设置为: $\alpha = 0.001, n = 1000$。

2 考虑如下方程组

$$\begin{aligned} \frac{\partial u}{\partial t} - \Delta u &= 0 \text{ in } \Omega \times \{t > 0\} \\ u &= 0, \text{ on } \partial\Omega \\ u(x, y, 0) &= g(x, y) \text{ in } \Omega \end{aligned}$$

其中$\Omega = [0, 1]^2, g(x, y) = \sin(2\pi x)\sin(2\pi y)$。设计求解该问题的并行算法,并实现，给出$t = 1$的结果。 其实现要点如下:

- 时间方向采取显式或隐式格式，空间方向区域分解
- 如果采用显式格式，不需求解线性方程组，要做并行的矩阵向量乘及并行的向量相加

- 如果采用隐式格式，要求解线性方程组，可用Jacobi方法。
- 具体的通信结构可以参考Jacobi方法的例子程序。

基于稳定性的考虑，隐式格式步长可以大一些，而显式格式步长必须较小， 二则各有利弊。

3 利用PHG软件，求解下方程：

$$
\begin{aligned}
-\nabla(a\nabla u) + u &= 1.0 \text{ in } \Omega \\
u &= 0 \text{ on } \partial\Omega
\end{aligned}
$$

其中$\Omega = [0,1]^3, \Omega_1 = [0,0.5]^3$, $a = 2$ in Ω_1, $a = 1$ in $\Omega_2 = \Omega\backslash\bar{\Omega}_1$. 并给出$\int_{\Omega_1} udx$.

实现要点如下：

- 给出初始网格，使得每个单元要么在Ω_1内要么在Ω_2内。
- 参数a可以设置为一个解析函数类型的自由度，在计算线性系统的时候使用。
- 求解线性方程组可以用PCG方法。
- 最后计算$\int_{\Omega_1} udx$的时候，由于并行计算采用的是动态负载平衡，不知道Ω_1是哪些进程负责，故要写一段程序， 通过Reduce的方法得到计算结果。