

数值模拟简介（2003 年秋）

作业#6-上交时间十月 27

注：请尽快开始动手做！本作业必须在 27 号完成，虽然其有好几个问题。

- 1) 当模型用漂散方程将控制粒子分区之后，典型的平衡分析将会产生下面的非线性位置方程：

$$-\frac{\partial^2 \psi(x)}{\partial x^2} = -(e^{\psi(x)} - e^{-\psi(x)}), \quad (1)$$

这里时间范围取 $x \in [0, 1]$ ，边界条件为 $\psi(0) = -V$ and $\psi(1) = V$ 。

在一个 N 节点的珊格系统中如果用一个简单的有限微分方法来解非线性位置方程，分解后的方程如下：

$$\begin{aligned} 2\psi_i - \psi_{i+1} - \psi_{i-1} + \Delta x^2(e^{\psi_i} - e^{-\psi_i}) &= 0 \\ \text{for } i \in [2, \dots, N-1], \\ 2\psi_1 - \psi_2 - (-V) + \Delta x^2(e^{\psi_1} - e^{-\psi_1}) &= 0, \\ \text{and} \\ 2\psi_N - \psi_{N-1} - V + \Delta x^2(e^{\psi_N} - e^{-\psi_N}) &= 0. \end{aligned}$$

注： $\Delta x = 1/(N+1)$ （并不是 $1/(N-1)$ ），在 $x=0$ 和 $x=1$ 处的节点不包括在分解后的方程中，但是会进入并穿过边界）

- a) 可以证明不管 ψ_i 's 的值是多少，雅各布矩阵式与上述分解方程联立是非奇异的。这就意味着可以使用阻尼牛顿法来解决这问题。
- b) 使用复合的牛顿法来解上面的方程，给定 ψ_i 's 的初始值为零， $N=100$ 。证明你的程序二次收敛，并且当要求确保每一部分的精度都达到 10^6 时，确定需要的牛顿迭代的次数。并在以下两种情况下考察：当 $V=1$ 和 $V=20$ 。如果当 $V=100$ 将会发生什么事情？
- c) 试着使用阻尼牛顿法解 $V=100$ 时的 ψ_i 's。在计算 ψ 时，大体估算一下我们需要多大的 V 可以使这一部分保持在 10^6 的精度，并使参与的误差估计量少与 200（确保能计算出你的函数估算值为多少才能保证衰减牛顿三角）。
- 2) 思考本课讲的雅各布自由法。在这一方法下，牛顿法是用来解 $F(x) = 0$ 的问题，而且 GCR 法是用来解牛顿修正方程的问题 $J_F(x^k)(x^{k+1} - x^k) = -F(x^k)$ 。使用 GCR 法将是暗示 $J_F(x^k)$ 不能分解，但是可以用来求取矩阵向量的乘积， $J_F(x^k)r$ 。通过近似，这一方法可以演化成为

雅各布（或者矩阵）-自由法。

$$J_F(x^k)r \approx \frac{1}{\alpha}[F(x^k + \alpha r) - F(x^k)].$$

雅克布-自由算法名字的由来是基于这只是一种计算 $F(x)$ 的算法，而不能用来计算 $J_F(x)$ 。

- a) 通过作业一中的分解非线性位置方程，我们可以比较标准的牛顿法的收敛速度和雅克布-自由法的收敛速度。为了比较两者的速度，我们可以画出两种方法的 $\|F(x^k)\|/\|F(x^0)\|$ 相对与 k 的对数图，并解释你的结果。对本试验而言，使用 N=100 和 V=10。对 GCR 运算法则，假设当 $\frac{\|r^k\|}{\|r^0\|} < \epsilon$ （这里 r^k 余量， $\epsilon = 0.1$ ）系统收敛。最后，用 $\alpha = 10^{-3}$ （前面已经定义过了）来近似的求解矩阵向量的积。

- b) 修正了 ϵ 和 α 之后，你的雅各布自由法的计算精度会变有什么变化？也就是说， $F(x^k)$ 与零点有多接近？使用标准牛顿法计算， $F(x^k)$ 与零点有多接近？

- c) 假设你对计算的精度要求不高，你仅仅是需要让 $\|F(x^k)\| < 10^{-3}$ 的收敛牛顿法。这样能不能在更少的函数估算值下求得理想精度的 ϵ 值？
- d) 另一种仅仅使用函数估算的牛顿法的例子就是用有限微分法计算近似的雅各布矩阵。那就是

$$(J_F(x^k))_{i,j} \approx \frac{1}{\alpha}[F_i(x^k + \alpha e_j) - F_i(x^k)]$$

其中 e_j 是在 1 和零之间的任何位置的向量。这样的有限微分法在每一步牛顿迭代中需要多少的函数估算值？

3) 在这一问题中，你需要调试基于牛顿算法的 matlab 程序来计算负载结构中结点位置的力平衡性（如果你发现可以有更简单的方法，你可以编写自己的牛顿求解器）。

程序应该遵循下面的格式：

节点：

标号	节点号	结点位置横坐标	结点位置纵坐标
----	-----	---------	---------

这里的“标号”是一种任意的标号，“节点数”是一个整体的节点数量，“结点位置横坐标”和“结点位置纵坐标”是节点在压杆没有受外力之前的 x,y 的坐标。

压杆：

s 标号	节点 1	节点 2	弹性
------	------	------	----

这里“标号”是一种任意的标号，“节点 1”和“节点 2”是一个压杆上所含有的节点数量，“弹性”是压杆的弹性。

载荷：

l 标号	节点号	x 方向的力	y 方向的力
------	-----	--------	--------

这里“标号”是一中任意的排号，“节点号”是负载力施加的节点的排号，“x方向的力”和“y方向的力”是力在x和y方向上的两个分量。

如果你想要使用含有很少的错误的matlab的原稿，在我们本课程的站点，你将会看到一组matlab文件，可以使用这些文件来找到压杆和节点在受到扩展力时的平衡位置。代码使用的是节点分析方程，因此在开始这一问题之前你需要复习一下材料的相关知识。

压杆/节点代码的驱动文件是readsystem.m和newton.m。readsystem.m读取压杆/节点的原始位置，压杆的相互连接关系，施加的力。newton.m文件中包含了牛顿法的所有代码。文件loadNewton.m构造了右手法和关于牛顿法的雅各布矩阵。文件force.m包含了计算压杆受力和派生力的所有的代码。很有必要修改部分或者全部的newton.m, loadnewtom.m和force.m文件。这里还有三个测试文件（他将会给你错误提示）-test1.sys,test2.sys,test3.sys.

- a) 提供一种求解器，来求解test1.sys,test2.sys,test3.sys。作完这个之后，确定你的代码是否工作正常（如果是，那么你可以分析你的结论了，如果不是，你就要解释为什么并修改他）。
- b) 用你的牛顿法求解每一个例子，证明结果的二次收敛性
- 4) 即便是对一个正确的牛顿法而言，还是有不少的测试会使其产生问题
 - a) 试着用你的基于牛顿法的压杆和节点求解器来运行test4.sys文件，看看会有什么事情发生？
 - b) 试着用你的基于牛顿法的压杆和节点求解器来运行test4.sys和test4.sys文件，为什么会产生不同的结果？
- 5)（最晚10月31号上交）挑战传统的6.336牛顿法的竞赛。如果谁得到了一种可以求解所有问题（一个约束的情况）的牛顿法求解器（使用较少的迭代步），我们将给他颁发奖学金
 - a) 根据力与压杆的变形量呈幂指数的关系，修改你的压杆和节点程序。此外，假设压杆内部储存的力由下面公式求出：

$$|f| = e^{\text{elasticity} * (\text{length} - \text{unstretchedlength})} - 1.0$$

注：如果你在长度上的改变量很小，上述形势将会减少你已经运行的模型。用一些简单的例子来检查你的新程序。证明二次收敛性

- c) 无论是用什么思路，试着得到一种收敛的牛顿法来求解从test7.sys到test11.sys问题，看看你最多可以完成几个的求解。事先声明,没有必要把所有的问题都解决。