书本 52 页闭区间套定理和实数集是不可列集的证明。 我懂,但是我对于这两个东西的理解还是非常迷糊的 我不知道怎么形容这个感觉。我觉得自己需要更加高维的角度来看待这个问题我不知道该怎么对这个问题提出质疑,我总是认为这个证明缺少了某些省略了但是非常重要的条件。(这个现象正常吗?)

《陈纪修》定义 5.1.2 p146 页 有关于凹凸函数的定义。 我直接抄下来吧 设函数 f(x) 在区间 I 上定义, 若对 I 中的任意两点  $x_1$  和  $x_2$  和 任意  $\lambda \in (0,1)$ ,都有

$$f(\lambda x_1 + (1 - \lambda)x_2) \leqslant \lambda f(x_1) + (1 - \lambda)f(x_2)$$

则称 f(x) 是 I 上的下凸函数。 若不等号严格成立,则称 f(x) 在 I 上是严格下凸函数。 类似地可以给出上凸函数和严格上凸函数的定义。

开区间的凸函数在定义域内一定连续 这个是可以证明的,以下凸函数为例,我现在直接证明令  $f(x_0)<\lim_{x\to x_0}f(x)$  我们这里直接令左右极限存在且相等 这样我们可以取一个点 $a< x_0$  不妨令  $\lambda=0.5$ 

 $f(\lambda a + (1-\lambda)x_0) = f(0.5a + 0.5x_0) \ \text{这里如果} a \to x_0 \ \text{那么} \ f(0.5a + 0.5x_0) = \lim_{x \to x_0^-} f(x) \ \text{而} \\ \lambda f(a) + (1-\lambda)f(x_0) = 0.5f(a) + 0.5f(x_0) \ \text{这个值显然是小于左极限的 和定义矛盾 如果} \\ f(x_0) > \lim_{x \to x_0^-} f(x)$ 

这个更加好证明了,具体找两个点a,b使 $\lambda a$  +  $(1-\lambda)b$  =  $x_0$  让f(a),f(b) 都小于  $f(x_0)$  就好了这样就和定义矛盾了所以可以得到函数值一定不会大于或者小于左右极限,那么只能是等于了所以我们可以说凸函数一定连续。不过是开区间内,闭区间的那两个端点可以不连续依然满足定义。

可以证明, 开区间的凸函数如果有不可微的点, 只能是左右导数不相等。不可能是跳跃或者别的。

都证明函数一定连续了, 那还说啥呢, 只能是这个情况了

《陈纪修》例 3.2.7 p78 页 Riemann 函数 这个是我发现总是用到这个东西,所以这个要看 这个函数特殊在,他的左右极限都是存在且相等,都为 0,但是函数值会由于其是否是有理数而不同 从而导致连续性的区别

《陈纪修》例 3.4.6 p97 页 Cantor 定理 这个还挺有意思的, 花时间看看

《华师大》第一章 §2二、p5 有关于上下确界的定义。多看几遍,别忘了,上下确界和这个值是否在集合中无关

还有31页的子列的定义和那些性质,都自己多翻翻,挺重要的

陈纪修没有讲有限覆盖定理, 所以我现在把华师大的教材这部分讲一下

《华师大》第十三章 §1p25 这里函数列的极限函数并没有要求函数列连续 这里只是说定义在同一数集上的函数,不代表就连续了 极限函数是通过数列的极限来定义的 这样,原来的函数可以不连续,可以有间断点,反正在那个点可以直接把间断的函数直接剃掉 但是后面可以继续用这个有间断的函数 我现在唯一的疑问就是,这个函数间断点的位置和数量能否被 n 影响?如果可以的话,具体的影响是怎么样的 这部分的反例什么的都比较充足,挺不错的

这个一致收敛的定义还挺奇妙,并不是直接由函数列来找极限函数,而是通过函数列和某个函数的关系来定义一致收敛。但是我有一个疑问,就是这个f(x)一般是怎么找到的?其实一致收敛的极限函数一定是逐点收敛的极限函数把。一致收敛是逐点收敛的充分条件

我有一个疑问,一致收敛出来的极限函数的连续性怎么看?如果函数列统一在某个点间断, 是不同样也可以得到极限函数,并且还是间断的? 这里通过定义一来得到不一致收敛的的结论里要求n'>N 是有道理的 不然任意一个 n 来找这个 x 不是随便找嘛。就算是收敛的,也能找到差值大于 $\varepsilon_0$ 的

函数一致收敛的柯西准则和函数列一致收敛是充要条件 这个柯西准则好,不用找那个极限函数了。直接由函数列本身就能定义一致收敛了

这个定理13.2 我感觉就是定义换了一种说法,这个和定义的意思难道不一样吗?不过他这个推论还挺不错。有点海涅原理的味道。

这部分和函数项级数的关系还挺紧密。 在 p35 页的定理 13.9 上面的 13.8 还挺有意思 极限可以交换顺序的条件是一致连续 不知道普通函数的条件是什么