

# 有感于SP++

## 1. 开门见山

SP++, 即 Signal Processing in C++的缩写, 是一个关于数值计算与信号处理的 C++开源程序库, 该库提供了线性代数与信号处理中常用算法的 C++类模板实现。

SP++最早发表于开源中国社区, 博客地址为: <http://my.oschina.net/zmjerry>, 使用前一定要看看这篇博文“SP++使用指南”, 地址为: <http://my.oschina.net/zmjerry/blog/4173>。后来该项目上传到了Google Code, 地址为: <http://code.google.com/p/tspl/>, 因为名字冲突, 项目名称另取为TSPL(Template Signal Processing Library)。

## 2. 心血来潮

SP++最早的工作始于 2008 年, 而动机则成于 2009 年。

2008 年全年和 2009 年上半年正值本科毕设和研一上课, 在此期间甚是无聊, 于是就想用 C++写点程序, 打发打发时间。本人从事信号处理专业, 因此就把信号处理中的常用算法用 C++写了一遍, 但这些程序各自为阵, 不成体系。

2009 年下半年做项目时要实现一个很复杂的算法, Matlab 程序中调用了许多内部函数, 转 C 语言时遇到了一些困难, 因为该算法在频域实现, 所以各种函数都是基于复数形式。为此上网查了一些 C 和 C++的开源库, 结果有两惊人的发现, 一个令人高兴, 而另一个则让人失望: 高兴的是网上有成千上万的关于数值计算和信号处理的程序库(我最终选用了 GSL); 失望的是这么多的库似乎没有哪个是中国人写的(唉, 其实原因也很简单: 在这个浮躁和功利的学术氛围下, 谁会愿意做这种“义务劳动”的事情呢?)。

经过一年的“烟酒僧”生活, 我有过一些思考: 三年研究生应当怎样度过呢? 我没有保尔柯察金那样的豪迈气魄, 但还是觉得应该做点有意义的事, 于是萌生了一个想法: 写一个关于信号处理的 C++程序库。然而由于信号处理算法中要经常用到矩阵分解, 解线性、非线性方程组、曲线插值与拟合等等数值计算的内容, 因此 SP++实际上可分为两部分, 即数值计算与信号处理。

## 3. 艰苦历程

想着容易做着难啊!

在时间、精力、Money 都有限的情况下, 仅凭我一人完成这样一项工作谈何容易。我又有个毛病: 做什么事都不甘心, 既然动手做了, 就一定尽我所能把它做好! 于是从理论学习, 算法设计, 代码编写、调试、验证, 到最终文档撰写, 一步一步做下去。

首先是时间问题, 平时要忙项目和硕士论文, 所以这项工作只能在晚上和周末有时间的情况下做, 还是鲁迅说的对: “时间就像海面里的水, 要挤还是有的”。其次是精力问题, 现在精力大不如前了, 想当初前一天包夜第二天考试, 而如今熬个通宵几天都缓不过来。最后就是钱的问题了, 做学生难, 在中国做学生更难, 穷得叮当响, 买一本几十块钱的书都要犹豫好长时间, 真是杯具呀!

最艰难的莫过于调程序——推公式——调程序——推公式……要写出正确高效的代码, 必须对算法的理论有深入理解。浅尝辄止、似是而非是远远不够的; 在已有的基础上生搬硬套、照猫画虎, 那就有可能贻笑大方了。

我坚信“有付出就会有回报”，经过2年半的努力，SP++的基本框架和使用文档已经完成。目前实现的算法有：（数值计算部分）向量与矩阵基本运算，矩阵分解，常规、欠定、超定线性方程组求解，非线性方程求根，曲线插值与拟合，非线性无约束优化算法等等；（信号处理部分）Fourier变换，数字滤波器设计，时频分析，小波变换等等。未实现（已有Matlab版本）的算法有：自适应信号处理（比如Weiner滤波器，Calman滤波器，自适应滤波器等），主要是因为还没有参与过关于自适应信号处理的项目，对这部分知识理解不到位（纸上得来终觉浅啊），缺乏实际工程经验，所以不敢误导读者。希望对自适应信号处理有一定造诣，并且对写代码又有一定兴趣的牛人能够补上这一空缺，本人将不胜感激。

一个人的力量毕竟有限，如果仅凭一个人把一件事做得尽善尽美，那么此人乃天才也！然而我不相信这个世界存在天才，我们生来都是人才，不同的是经过教育培养，有些人保持着人才的特性（还好，我属于这一类），而有些人则被教育成了蠢才。所以SP++中必然存在着已经发现的和尚未发现的诸多不足，已发现的力争及时改正，未发现的只好争取尽早发现再做改正。

## 4. 有得有失

我这人有点傻，事实证明我确实有些傻，不仅毫无回报地把几乎所有空闲时间都花在了这个“义务劳动”上，而且自己还倒贴了五六百元买书。

“有得有失，有失有得”，我感觉这是老祖宗给我们留下的最大智慧（至少是“之一”）。由于SP++而失去和错过的东西太多，在此没有心情去讲了，还是谈谈所得吧，以勉励他人。有时候在想我付出这么多到底获得了什么？其一是网友的感谢。其二是水平的提高？其三，唉，实在没有了！

最直接的回报当数网友的致谢、赞美和建议、批评了。致谢和赞美固然好，让人有飘飘然的感觉；不过建议和批评也必不可少，使你改正错误，更进一步。

至于水平吗，不敢说有多少提高，只能是在某些方面加强吧了。水平，或者能力的面太广了，学识渊博是一种能力，为人处事也是一种能力，只是在不同领域中、不同场合下能力的各种方面的权重不一样而已。而且某一方面能力的加强，必然会导致另一方面的减少，姑且称之为“能力守恒定律”吧，就看你的选择是什么了。

看过一本书里讲学习的四种境界：学会——会学——会用——被用。真是有一种找到知己的感觉，英雄所见略同啊（大言不惭地自称一下英雄，过过瘾，娃哈哈）！学会很容易，我们从小学（甚至幼儿园）到大学，经过千锤百炼，身经百战，可谓是过五关斩六将的勇士，该学的也都学会了，至少也学得差不多了。会学可就难了，学会与会学就像鱼与渔的关系，可惜据我观察，会学的人不多。会用也相对容易些，但也有好坏之分，用得好，用得恰如其分就能达到事半功倍的效果。最难的要数被用，做到这一点实属不易，要得到别人的认可必须要有过人之处。扯远了，赶快回来！

总而言之，最大的收获，或是对我最大的鼓励莫过于“被用”了。

## 5. 见仁见智

SP++到底是个什么水平？

答：比起老外们写的成熟的库相差甚远，比起网上传的“散兵游勇”式程序要高出那么一点点。毕竟SP++有自己的体系框架，很多算法在项目中经过了实践的检验，而且有些算法根据网上高人指点和好友建议做了不少改进，在此表示忠心感谢！

不过对于一些建议我还是固执己见，没做修改（同样表示忠心感谢），比如某些明显的效率问题。原因有二：其一，既然明显，编译器会帮助我们优化，所以不必担心；其二，部分缺陷是出于开发效率和运行效率的折中，运行速度是效率，开发速度也是效率，如果能够

方便使用，牺牲一些运行效率还是值得的。

**SP++**对我来说到底有什么意义？

答：意义深远，不敢说是比硕士学位重要（当然不敢，以后还得靠这吃饭了），至少比那个本本有意义。一个硕士学位证到底有多少含金量？没上过的不知道，上过的吓一跳——真没多含金量，含水量倒不少。试想，一个硕士论文能被多少人引用？毕业后还有多少人能继续关注曾经被他研究过的东西？不言自明，少之又少，因为我们处在一个文凭与水平毫无关系的时代。所以 **SP++**过万的访问量对我来说确实比那张学位证书更有意义。

是为感

张明，2010年8月于西安