文章编号:1007-6735(2013)02-0147-05

基于GPU的B-S模型下改进的Crank Nicolson算法

王文浩, 邬春学

(上海理工大学 光电信息与计算机工程学院,上海 200093)

摘要:针对Black-Scholes模型及其公式特点进行了理论分析与数学处理,给出了优化的Crank-Nicolson算法,提高了实际期权交易效率.通过使用GPU作为计算平台,结合CUDA架构技术,验证改进后算法的有效性和适用性.在CPU平台下进行横向测试,验证GPU平台运行环境优势.实验表明,改进后的算法在GPU平台下运行所提升的效果显著,运算精度和效率得到提高.

关键词:金融期权计算;B-S模型;改进的C-N算法;GPU;CUDA构架;HPC

中图分类号:TP 302 **文献标志码**:A

The Research of Improved Crank Nicolson Algorithm with B-S Model Based on GPU

WANG Wenhao, WU Chunxue

(*School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China*)

Abstract: Based on the analysis of the famous Black-Scholes (B-S) model, the Crank-Nicolson algorithm was optimized in accordance with the theoretical and numerical analysis resuls. A new way to improve the efficientey and effective ness of actual options tradings was created. By using the GPU as a computing platform, and combining with the CUDA technology, the validity and applicability of the improved algorithm were verified. By doing horizontal test on CPU platform the running environmental advantages of GPV platform were verified. The experiments show that the improved algorithm running on GPU platform can get significant effecs. The operation precision and computational efficiency can be greatly improved.

Key words: *financial option calculation*; *B-S model*; *improved C-N algorithm*; *graphic processing unit*; *compute unified device architecture (CUDA)*; *high performance computing*

随着中国经济的持续发展,金融领域的相关问题愈发被重视,其中,期权以其所具有的降低投资者风险的功能被视为重中之重.针对市场瞬息万变的特点,期权的定价计算要求高效与准确.本文以最为著名的Black-Scholes为依托,根据其特点选取了一种类有限差分算法(C-N算法)并对其进行优化处理,目的是提高其运算效能与精度,应用计算机技术,结合当前计算机硬件的发展特点,选取目前发展

收稿日期:2012-06-01

第一作者:王文浩(1987-),男,硕士研究生.研究方向:计算机体系结构、无线传感器网络、工业控制系统.E-mail:wwh19870131@126.com

通讯作者:邬春学(1964-),男,教授.研究方向:无线传感器网络、网络控制系统、高速网络.E-mail:tyford@126.com

迅猛的GPU作为运算平台并与传统的CPU平台作性能对比.经实验表明,优化后的算法通过在GPU平台上运行,在加强了运算效率且拓展了实用性的同时,运算性能与传统平台相比明显提高,这为今后期权定价问题与计算机技术相结合的研究提供了依据.

由于GPU相对CPU集群有着在相同数量的运算核心情况下成本低、体积小、易于搭建等优点,笔者提出使用图形处理器(GPU)代替目前主流的中央处理器CPU集群(如利用微软HPC、HMPP等)作为运算平台的核心,并通过使用英伟达公司的CUDA架构技术运行改进后的C-N算法的程序,实现了将著名的B-S期权定价模型在GPU平台上运行[2~4]的目的.为了对研究效果作更为直观的了解,在实验中以两套基准作为测试条件(不同向量级数和不同方程数),并对实验结果进行分析.

1 Black-Scholes模型及其优化算法

基于股票期权价格的微分方程为

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial s} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2}\mathrm{d}t = rf \qquad (1)$$

式中,f为期权在t时刻的价格;r为无风险利率;S为标的资产价格;σ为资产价格的波动率.这是一个微分方程,通过求解此方程,得到看涨期权和看跌期权的精确模型(欧式),即Black-Scholes式.该式出现后,通过不断的改进和发展,已被作为期权交易的标准工具.若用X表示执行价格,t_1表示距到期日的时限,其看涨期权的价值为

$$C = SN(d_1) - X\mathrm{e}^{-n}N(d_2)$$

其中,$d_1 = [\mathrm{In}(S/X) + (r + \sigma^2/2)t_1]\sigma t_1^{1/2}$

$$d_2 = [\mathrm{In}(S/X) + (r + \sigma^2/2)t_1]\sigma t_1^{1/2} = d_1 - \sigma t_1^{1/2}$$

$N(x)$为堆积概率分布函数,具有标准正态分布的特点.因为股价的看跌和看涨之间呈平价的关系,能得到看跌期权的价值为

$$P = Xe^{-n}N(d_2) - SN(-d_1)$$

模型由t_1、r、S、X和σ 5个参数直接决定.对波动率而言,其估值尽管在公式中具有局限性,但它却能直接体现执行价格程度,各投行为将交易风险降至最低并保证资产安全,对波动率高度重视.由GARCH模型可知,σ因为具有波动类聚现象,其值虽不断变化,但分布较为紧密,而且为数值,格式简单识别容易,所以B-S公式相比于其它理论模型有其特殊的优势.正如此,投资者在实际的市场交易中往往用波动率替代原绝对价值作为某些期权报价的参考.

通过对B-S模型的形式进行分析,可以将其视为一个偏微分方程.结合期权交易的实际情况,选择Crank-Nicolson来求解方程.对其整理后,得到

$$p\frac{\partial^2 u}{\partial y^2} + q\frac{\partial u}{\partial y} + lu - \frac{\partial u}{\partial t^2} = 0$$

式中,t_2为分布在时间轴t上的时间;y为状态变量.随着时间t_2的不断变化,如果将系数p、q、l表示为t_2与y的函数,那么由y轴上不同的Y值,就可求得相应节点的U值,根据初始条件分析,$u(y,0)$和$f(y)$相等.图1为以t_2和y组成的坐标环境中创建的一个区域,域中的各坐标点和t_2及y相关联.对于横轴t(时间轴),将其平均分成N个点($n=0,1,\cdots,N$),并以d为间隔.在纵轴y上,通过对区间$[y_{\min}, y_{\max}]$进行分割,由间隔s将变量y分作$i=0,1,I$个点,原则上这个域的范围应当足够大,这样就可以包含数目足够庞大的数值解,如图1所示的锥形区域,各点联系紧密,其中任何一点都可以由其它相关的几点求得,实线和虚线共同表示其中一点与其它各点的联系.

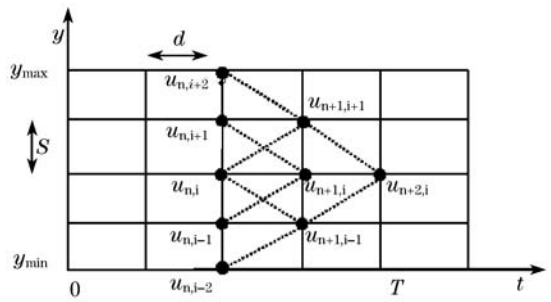


图1 与t和y相关的偏微分方程的坐标区域

Fig.1 Coordinate domain of partial differential equations related with t and y

各点的推导可依据表达式

$$u_{n+1,i} = \left(\frac{d}{s^2}p + \frac{d}{2s}q\right)u_{n,i+1} + \left(1 + dl - \frac{2d}{s^2}\right)u_{n,i} + \left(\frac{d}{s^2}p - \frac{d}{2s}q\right)u_{n,i-1}$$

如果要求$u_{n+1,i}$,只需要求得$u_{n,i}$、$u_{n,i-1}$和$u_{n,i+1}$即可,依次类推,如果要求$u_{n+1,i+1}$,只要求得$u_{n,i+1}$、$u_{n,i}$以及$u_{n,i+2}$即可.至于初始值$u_{0,i}$的求取,只需要利用初始条件求得即可.利用此种递推方法将最终求得所有网点上的函数u,所有状态变量都能够得到表示.实现这种方法虽然较为容易,但由于在大数据量计算的情况下,随着坐标域范围的无限扩大,需要经过无数次的偏导近似处理.任何一次偏导

处理结果和真实值相比总会有极微小的偏差，所以在运算需要经过大数量级偏导处理的情况下，数据值因不断求偏导而产生的偏差会被逐断放大，大到一定程度必定会出现严重的谬误.

为了保证数据的准确性，采取一种提高算法精度的处理方法，同时保持计算简单高效的特点. 针对欧式期权，将方程(1)中标的变量 S 对数化处理，取自然对数 $K=\ln S$，形如

$$\frac{\partial f}{\partial t}+\left(r-\frac{1}{2}\sigma^2\right)\frac{\partial f}{\partial K}+\frac{1}{2}\sigma^2\frac{\partial^2 f}{\partial K^2}=rf \quad (2)$$

由于在时间轴上分布着大量的时刻 T，现将时间轴分为 N 个小时间段(从零时刻到期权执行时刻)，分法与上文相同. 由于这些小时间段间隔是相等的，则终止时刻 $T=N\Delta t$. 因为期权交易是即时性的，为体现其最后一次的收益情况，需要将最新一次收益的时刻 t_n 放在最后的格点上，但期权交易信息变化极快，任何时候的交易情况都有所变化，所以只能近似的表示为 $t_n\approx M\Delta t$. 由于 $S_{\max}=Su^N$，$N=T/\Delta t$，$K_{\max}=\ln S_{\max}$ ($K_{\max}$表示在纵轴上取得的最大值)，为计算方便，股票价格的对数步长取 $\Delta K=\sqrt{3.5\Delta t}$，则 $M=K_{\max}/\sigma\sqrt{3.5\Delta t}$，如图 2 所示，虚线所框部分反映了改进算法的核心部分，直观地表示出一种递推关系，并且形式更为简单，这样就可以大大提高整个运算的效率，由于前面分析过图中任何一点都可以被和它相关联的几点求得，这样期权价值、时间和资产价格的对数值可以用简单的数值计算方法求得，而且算法的精确度得到大幅度的提高，至于庞大的计算量，可以应用计算机技术来解决. 将(i,j)位置的期权价值表示为 $f(i,j)$，点(i,j)映射了 $i\Delta t$ 时刻以及资产即时价格 $j\Delta K$. 利用已知的边界条件以及这些网格点间的相互关系，连续的偏微分方程就被转化成为一系列的相关联的差分方程. 进行逐次求解后，最终可以得到初始资产价格(零时刻资产价格)所对应格点的期权价格.

为了将偏微分方程(2)转化为差分方程的形态，保证其计算精度，将 C－N 差分格式异型为

$$m_j f_{i-1,j-1}+(n_j-1)f_{i-1,j}+\nu_j f_{i-1,j+1}=-m_j^* f_{i,j-1}+(1-n_j^*)f_{i,j}-\nu_j^* f_{i,j+1}$$

其中

$$\begin{cases} m_j=\dfrac{\Delta t}{2\Delta K}\left(r-\dfrac{\sigma^2}{2}\right)-\dfrac{\Delta t}{2\Delta K^2}\sigma^2 \\ n_j=1+\dfrac{\Delta t}{2\Delta K^2}\sigma^2+r\Delta t \\ \nu_j=-\dfrac{\Delta t}{2\Delta t}\left(r-\dfrac{\sigma^2}{2}\right)-\dfrac{\Delta t^2}{2\Delta K^2}\sigma^2 \end{cases}$$

$$\begin{cases} m_j^*=\dfrac{1}{1+r\Delta t}\left(-\dfrac{1}{2}rj\Delta t+\dfrac{1}{2}\sigma^2 j^2\Delta t\right) \\ n_j^*=\dfrac{1}{1+r\Delta t}\left(1-\sigma^2 j^2\Delta t\right) \\ \nu_j^*=\dfrac{1}{1+r\Delta t}\left(\dfrac{1}{2}rj\Delta t+\dfrac{1}{2}\sigma^2 j^2\Delta t\right) \end{cases}$$

观察此种格式变化处理，可以得知，系数 p，q，l 被表示成了近似于对 m_j 和 m_j^*、n_j 和 n_j^* 以及 ν_j 和 ν_j^* 进行了求差处理，在逐次求解的过程中，这样的操作大大抵消了前一步骤所带来的错误，保证了运算的准确性.

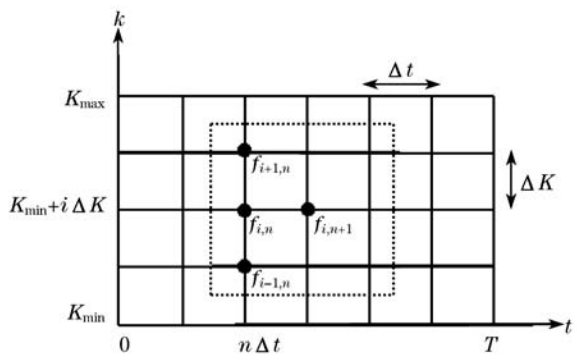


图 2　改进的 C－N 算法所描述的方程的坐标域

Fig.2　Equations coordinate domain decribed by the improved C－N algorithm

2　算法在 GPU 平台上的实现

对 C－N 算法进行改进后，需要选择合适的平台来进行运算. 这里选用硬件和软件发展都较快的 GPU 作为运算平台并以 CUDA 架构技术作为支持.

统一计算设备架构(compute unified device architecture，)是用以管理和处理 GPU 计算的软件和硬件的统一性架构，将 GPU 看做一个数据并行计算设备，在程序运行过程中将计算通过 GPU 直接实现. 对于求解每个 $f_{N-1,j}$，由各数值之间的连带关系，进行逐步迭代，最后计算得到 $f_{0,j}$ 的值. 由于计算数据量较大，所以特别适合以并行运算[7－8]的方式进行，利用 CUDA 架构并借助 C 语言在 GPU 平台上[9]进行运算.

利用改进后的 C－N 算法，即利用更有效的类有限差分方法进行处理，通过对实际股票期权交易的具体情况进行分析，可以将实际应用以图的方式直观体现.

如图 3(见下页)所示，根据股票期权交易的实际，由于交易状况不断变化，结合 B－S 公式，T 代

表交易终止时间,在实际的运算过程中对 t 采用离散化处理,使函数从理论上能够在域内呈网格化分布.实际执行价格,在理论中存在最大值与最小值,但是这些值不论如何变化,都基本符合标准正态分布.图中弧线部分所囊括的范围即为函数值存在的范围,可以发现此范围大体也呈锥形分布.

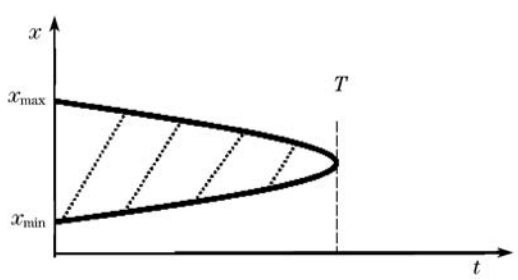


图3 B-S方程解的范围域

Fig.3 Range domain of B-S equations

为了求得从 t_n 时刻到 t_{n+1}时刻的函数值,利用下面的逻辑关系代码可以完成,该算法的结构与脉络为

for all i do

calcul the $Q_{i,n}$, $L_{i,n}$

end for

$N+1 \rightarrow n$

while $n \geqslant 0$ do

for all i do

calculate $U_{i,n}$ from $x_{i,n+1}$, $P_{i,n+1}$, $Q_{i,n+1}$, $L_{i,n+1}$

end for

for all i do

calculate $P_{i,n}$, $Q_{i,n}$, $L_{i,n}$

end for

for all i do

solve the tridiagonal equant $Px = u$

find x

where $U_{i,n}$ vector formed u

end while

将算法在GPU平台上利用CUDA架构进行实现,对于以前在CPU平台上需要进行循环计算的for语句部分,利用CUDA基于GPU的软件与硬件的综合功能并行处理解决.

为了与CPU集群的运行效率进行对比,选取在Windows HPC Server 2008环境下通过使用C语言对传统B-S算法进行编程,然后进行对比测试.利用C语言所编写的程序主体部分可以简单的表示为(省略号为隐去部分)

```
public double BlackScholes(string CallPutFlag,
double S,double X,double T,double r,double v)
{·····
    d1 = (Math.Log(S/X) + (r + v * v/2.0) *
T) /(v * Math.Sqrt(T));
    d2 = d1 - v * Math.Sqrt(T);
    if(CallPutFlag = = "c")
    {
      dBlackScholes = S * CND(d1) - X * Math.
Exp(- r * T) * CND(d2);
    }
    else if(CallPutFlag = = "p")
    {
    dBlackScholes = X * Math.Exp(- r * T) *
CND(- d2) - S * CND(- d1);
    }
    return dBlackScholes;
}
```

通过在CPU集群上运行上述程序,就可以进行与在GPU上运行效率的对比.以保证之前对C-N方法的改进有效而且实用,并为今后的研究提供参考依据.

3 运算的测试结果比较及性能分析

首先进行算法的精度测试以及准确度的比较.利用改进后的C-N算法,对各大投行近年来的一些项目数据进行测试,然后与实际值进行精度与准确度的对比,如表1所示.经过测试后发现,不仅求得的数据精度有着明显的提高,并且与实际值的误差甚微.

表1 C-N改进算法精度与准确度的测试

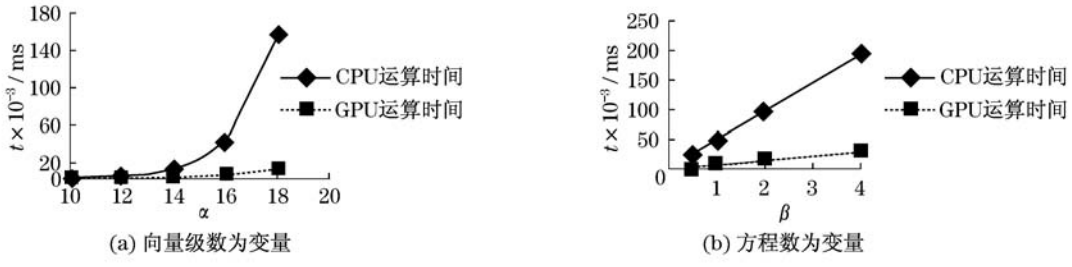
Tab.1 The precision and accuracy test for the improved C-N algorithm

项目	T	S	MIN	MAX	数值解	实际解	误差值
WFC	250	33	30.52	36.81	35.893 5	35.92	0.024 7
CVX	250	60	54.08	75.97	74.227 6	74.28	0.052 4
IBM	250	80.3	72.73	97.40	96.956 3	96.97	0.013 7
HKG	249	160	124.9	187.6	168.884 7	168.90	0.015 3
SHA	241	16	12.12	23.38	13.848 2	13.86	0.011 8
LON	254	26	16.21	21.65	21.385 7	21.39	0.004 3

进而要对不同算法在两种平台上的运算效率进行比较.运行在CPU集群和GPU平台上的算法,在GPU平台上是按照Crank-Nicolson改进后方法来

求解 B－S 方程，在英伟达 GeForce 550M GPU（运算能力 2.1）上进行测试. 作为对比参考，选择在 Windows HPC Server 2008 环境下通过使用 C 语言对程序进行串行运算测试. 将两种平台的测试结果用图表的方式进行对比，两种平台的搭建不必赘述.

通过在 CPU 与 GPU 上进行两种算法的运算测试，最终获得了在不同条件下两种平台的运算时间来进行对比. 这两种测试是相对独立的，一个数据的测试也不会影响其它数据的测试，结果也是独立的. 图 4 为在不同的向量级数 α 和方程数 β 下获得的两种平台运算时间的比较曲线，图中可直观反映两种平台的运算性能.



(a) 向量级数为变量　　(b) 方程数为变量

图 4　不同平台运算时间的对比

Fig.4　CoMParison of computating times on different platforms

其中横坐标的 10 代表向量级数为 α^{10}，18 代表向量级数为 α^{18}，依此类推.

表 2 所示的是 GPU 运算平台相对于 CPU 平台的实际性能增益，表中时间增益通过 CPU 平台与 GPU 平台各自运算时间的比值来衡量. 经过分析可知，在向量级数较小的情况下不论方程数目如何增减，若以运行时间为判断标准，GPU 平台相对于 CPU 平台的运行时间花费要多. 但实际期权交易情况处理的数据量非常大，因而向量级数原则上取足够大的数使其更具备实践价值，由表可知，向量级数越大，GPU 平台相对 CPU 平台的运算速度提升越大.

表 2　GPU 平台运算速度提升参量

Tab.2　GPU platform performance gain parameters

N	P				
	2^{10}	2^{12}	2^{14}	2^{16}	2^{18}
5 000	0.20	1.32	3.92	8.58	14.41
10 000	0.21	1.32	3.95	8.62	14.65
20 000	0.20	1.32	3.96	8.60	14.66
40 000	0.21	1.33	3.98	8.63	14.69

因为 GPU 自身结构及材料和工艺限制等的原因，若将其与 CPU 相比，GPU 显卡首先要将数据从 CPU 中复制到内存中，内存中的全局处理器负责通信于传输，因其无缓存机制，在此过程势必要有一定的时间消耗，因而在面对小数据量的计算时，这一影响体现的后果比较明显，所以运算消耗在内存传输上的时间反而比 CPU 多，GPU 的优势得不到体现. 如果将运算的级数朝着尽可能大的方向调整，由实验证明，运行在 GPU 上的并行算法的运算性能要显著优于 CPU 平台上的串行算法，这是因为在通信与传输数据的过程中，所要消耗时间的比重已微不足道了. 本研究以金融衍生品的定价计算为依托，所以其实际运算的数据量要远大于本研究中的所模拟运算的数据量，因此在实际的期权定价计算中，GPU 所体现的性能优势将会比 CPU 平台更加明显.

4　结束语

试验表明，改进后的 C－N 算法提高了运算的精度，经运算所得到的数值接近实际值. 通过与 CPU 平台上常规算法程序的运算性能比较之后，若以运算时间为参考标准，可知性能提高明显，最大约为 14.7 倍，显示了 GPU 计算在处理大规模并行运算数据上的突出优势. 在使用 CUDA 这一最新应用平台的过程中，本研究加深了对 GPU 通用计算技术的理解，同时也为 GPU 的高性能通用计算的应用提供了参考. 近年来 GPU 一直在飞速发展，不论其在硬件性能上还是在可编程功能的拓展方面上都有大幅度的提高，这对继续推动 GPU 在金融计算领域内的应用提供非常大的便利. 由计算机软硬件的发展趋势来看，GPU 的应用会愈加广泛和多样，例如在对冲基金交易和投资组合优化等问题的处理上有实际应用潜力. 伴随着技术的不断发展与完善，GPU 在金融计算领域的作用将更加明显.

（下转第 156 页）

蓄电池系统时,光伏输出功率波动较大;在光伏发电系统出口处加入储能蓄电池系统后,储能蓄电池能够快速吞吐光伏输出功率的波动部分,使注入电网的功率明显平滑.当HHOG系统开启时,其对电网有一定的功率冲击,但影响不大.

6 结束语

储能蓄电池利用单级DC/AC变换器实现功率调节,简化了系统结构.建立了含氢氧发生器的储能光伏并网系统模型,仿真结果表明,氢氧发生器能够准确地模拟其需要的恒定功率.在光照强度变化时,采用四象限变流器控制的蓄电池储能系统能够快速平滑光伏发电系统输出的有功功率的波动.

参考文献:

[1] 裴玮,盛鹏,孔力,等.分布式电源对配网供电电压质量的影响与改善[J].中国电机工程学报,2008,28(13):152-157.

[2] Penner S S. Steps toward the hydrogen economy [J]. Energy,2006,31(1):33-46.

[3] 唐西胜,邓卫,李宁宁,等.基于储能的可再生能源微网运行技术[J].电力自动化设备,2012,32(3):99-103.

[4] 孙自勇,宇航,严干贵,等.基于PSCAD的光伏阵列和MPPT控制器的仿真模型[J].电力系统保护与控制,2009,37(19):61-64.

[5] 姚致清,张茜,刘喜梅.基于PSCAD/EMTDC的三相光伏并网发电系统仿真研究[J].电力系统保护与控制,2010,38(17):76-81.

[6] Shihhido S, Takahashi R, Murata T, et al. Stabilization of wind energy conversion system with hydrogen generator by using EDLC energy storage system [J]. Electrical Engineering in Japan,2009,168(3):10-18.

[7] Tremblay O, Dessaint L A, Dekkiche A A. A generic battery model for the dynamics simulation of hybrid electric vehicles [C]// Proceedings of the 2007 IEEE Vehicle Power and Propulsion Conference, Arlington: 2007,284-289.

[8] 张步涵,曾杰,毛承雄,等.电池储能系统在改善并网风电场电能质量和稳定性中的应用[J].电网技术,2006,30(15):54-58.

[9] 姚勇,朱桂萍,刘秀成.电池储能系统在改善微电网电能质量中的应用[J].电工技术学报,2012,27(1):85-89.

(编辑:金 虹)

(上接第151页)

参考文献:

[1] 郑小映,陈金贤.有交易成本的期权定价方法[J].系统工程,2008,18(5):13-16.

[2] Ralf H, Alexander K, Micheal W. Physically guided animation of trees[J]. Comput Graph Forum,2009,28(2):523-532.

[3] 吴恩华.图形处理器用于通用计算的技术、现状及其挑战[J].软件学报,2004.15(10):1493-1540.

[4] 吴恩华,柳有权.基于提醒处理器(GPU)的通用计算[J].计算机辅助设计与图形学学报,2004,16(5):601-612.

[5] nVidia Company. CUDA Progrmming Guide: CUDA [EB/OL].(2010-8-23)[2010-10-14]. http://www.nvidia.com/object/cuda_home.html.

[6] 张舒,褚艳丽.GPU高性能计算之CUDA[M].北京:水利水电出版社,2010.

[7] 孙世新,卢光辉,张艳,等.并行算法及其应用[M].北京:机械工业出版社,2005.

[8] 邓仰东.Nvdia CUDA超大规模并行程序设计训练课程:性能提升[EB/OL].(2009-02-19)[2010-03-11]. http//cuda.csdn.net/client/CUDA_lecture.Rar.

[9] 左颢睿,张启衡,徐勇,等.基于GPU的并行优化技术[J].计算机应用研究,2009,26(11):4115-4118.

(编辑:金 虹)